

THÈSE de DOCTORAT de L'UNIVERSITÉ PARIS 6

Spécialité

Informatique

présentée par

M. Stéphane Gourichon

pour obtenir le grade de

DOCTEUR de L'UNIVERSITÉ PARIS 6

Utilisation d'un compas visuel pour la navigation d'un robot mobile

soutenue le 2 juillet 2004

devant le jury composé de :

Dr. Jean-Arcady Meyer	Directeur de thèse
Dr. Matthias Franz	Rapporteur
Pr. John Hallam	Rapporteur
Dr. Ryad Benosman	Examineur
Pr. Philippe Gaussier	Examineur
Pr. Maurice Milgram	Examineur

Résumé

Dans cette thèse, nous nous intéressons à la navigation et à l'orientation d'un robot à l'aide de mesures d'angles, sans information de distance ni reconstitution de l'environnement.

Des modèles issus d'expériences sur des abeilles ont montré qu'une approche minimaliste suffit pour une tâche de retour au nid à orientation connue. Nous analysons les conséquences d'une erreur d'orientation et les modes de fonctionnement dégradés qui restent possibles mais insatisfaisants.

Nous élaborons une méthode originale et minimale de réorientation, appelée « compas visuel ». L'observation de trois panoramas d'orientation connue et une approximation au premier ordre suffisent pour mémoriser de quoi s'orienter. Nous comparons le compas visuel avec une seconde méthode plus simple et approximative. Nous montrons formellement et expérimentalement que leur précision est suffisante pour naviguer et que leur champ d'action peut être arbitrairement étendu par construction d'une carte cognitive.

Abstract

Using a visual compass for mobile robot navigation

In this thesis, we consider navigation and orientation of a robot using measurements of angles, without any distance information or reconstruction of the environment.

Previous models derived from experiments involving bees showed that a minimalistic approach suffices to perform homing if the orientation is known. We analyse the consequence of an error in orientation and the degraded functionality that remains possible, but is unsatisfactory.

We elaborate an original and minimalistic reorientation method, called “visual compass”. The observation of three panoramas of known orientation and a first-order approximation is sufficient stored information for reorientation. We compare the visual compass with another method, simpler and more approximate. We show formally and experimentally that the precision of each is sufficient to navigate and that their range can be arbitrarily extended via the building of a cognitive map.

Remerciements

Je remercie tout d'abord Jean-Arcady Meyer dans son rôle de directeur de l'AnimatLab et directeur de thèse, qu'il assume de façon à la fois très exigeante et très souple. Je remercie chaleureusement Matthias Franz et John Hallam pour l'intérêt qu'ils ont manifesté pour mon travail de thèse et pour avoir aimablement accepté la charge de rapporteur alors que le français n'est pas leur langue maternelle. Je remercie avec eux Philippe Gaussier, Maurice Milgram et Ryad Benosman d'avoir accepté de m'honorer de leur présence dans le jury malgré leurs nombreuses obligations professionnelles. Je remercie aussi Patrick Pirim : ma thèse aurait été très différente sans la puce de perception visuelle qu'il a inventée et réalisée.

Je salue toute l'équipe AnimatLab, les membres actuels comme les anciens membres, qui illustrent collectivement que la bonne humeur d'un chef d'équipe a une nette influence sur celle de l'équipe entière. Merci à Agnès Guillot et Stéphane Doncieux pour leurs conseils, David Filliat qui a influencé positivement mes premiers pas en navigation, Vincent Cuzin pour avoir pris le relais de la maintenance logicielle du parc informatique de l'AnimatLab et qui devrait peut-être trouver à son tour un successeur, Olivier Sigaud qui a spontanément proposé de relire le mémoire de thèse et m'indiquer quelques fautes. Hors de l'AnimatLab, je remercie le personnel du pôle IA et du LIP6 pour leur travail administratif et technique et leur présence agréable. Je pense aussi à Pierre Andry, Jean-Christophe Baillie, Nicolas Bredèche, Bernard Huguenev, Samuel Landau, Anneli Lenica, Stéphane Sikora.

Je remercie James Crowley qui m'a aidé à faire le point sur les qualités de mon travail, ce qui m'a motivé pour la longue épreuve de rédaction.

Bien qu'il leur reste une grande marge de progression en terme de facilité d'usage, les technologies de l'information n'auraient pu être autant à la fois puissantes et souples sans les standards ouverts, l'Internet et la communauté du logiciel libre. Je remercie collectivement non seulement des acteurs comme l'IETF, le W3C, l'Electronic Frontier Foundation, la Free Software Foundation, le projet Debian ou MandrakeSoft, mais aussi tous ceux qui apportent leur contribution à des projets libres. Toute la thèse a été réalisée avec des logiciels libres, dont : emacs et l'environnement GNU, gcc, Linux, valgrind, CVS, unison, doxygen, L^AT_EX, xfig, drgeo, MagicPoint, ImageMagick, R, KDE, mozilla, xine, le lecteur videolan, kino/dvgrab, transcode, qcad. L'Internet et les logiciels libres auraient pu rester de la pure science fiction pendant très longtemps. Espérons que la société saura s'améliorer grâce aux technologies de l'information et non étouffer leurs potentialités par frilosité ou égoïsme.

Hors l'aspect professionnel, l'entourage personnel est très important. Je remercie mes parents, frère et soeur pour leur soutien, ainsi que les amis qui se reconnaîtront. Stéphanie, maintenant mon épouse, mérite un honneur particulier pour son encouragement quotidien et tout ce que nous construisons ensemble.

À tous les miens.

Table des matières

1	Introduction	17
I	Problématique	23
2	Navigation locale utilisant des panoramas	25
2.1	Structure commune à la plupart de ces modèles	25
2.1.1	Signature d'un lieu	25
2.1.1.1	Notations	26
2.1.1.2	Représentation de l'orientation de l'animat	28
2.1.2	Appariement	28
2.1.3	Calcul de direction	29
2.1.4	Surface de captation (Catchment area) et attracteurs	29
2.1.5	Difficultés supplémentaires possibles	31
2.1.5.1	Rotation mentale	31
2.1.5.2	Désorientation	31
2.1.5.3	Vision non totalement panoramique	32
2.1.6	Comment étalonner les modèles ?	32
2.1.6.1	Obtention de statistiques	33
2.1.6.2	L'appariement	34
2.1.6.3	La direction à suivre : mesure de composante centripète	34
2.1.6.4	La longueur du pas	36
2.1.6.5	L'orientation	36
2.1.6.6	Tout ensemble : mesure synthétique	36
2.1.6.7	Conclusion	36
2.1.7	Généalogie des modèles	37
2.2	Les premiers modèles, Cartwright & Collett	37
2.2.1	Signature d'un lieu	37
2.2.2	Appariement et calcul de direction	38
2.2.2.1	Principe	38
2.2.2.2	Appariement	38
2.2.2.3	Différents essais pour la direction à suivre	40

2.2.3	Conséquences d'une désorientation	42
2.2.4	Conclusion sur les modèles CC	43
2.3	Modèles modifiant le calcul de direction à suivre	43
2.3.1	Méthodes inverse et moyenne, modèle de Hong	43
2.3.1.1	Signature d'un lieu	43
2.3.1.2	Appariement	44
2.3.1.3	Calcul de direction	44
2.3.1.4	Conséquence d'une désorientation	45
2.3.1.5	Conclusion	45
2.3.2	Modèle PV vecteur proportionnel	45
2.3.3	Modèle DV vecteur différence	46
2.4	Simplifier tout, ALV ou « vecteur amer moyen »	47
2.4.1	Signature d'un lieu	47
2.4.2	Appariement et calcul de direction	48
2.4.3	Conséquences d'une désorientation	49
2.4.4	Conclusion	49
2.5	Méthodes d'appariement différentes, Weber et al.	49
2.5.1	Conséquences d'une désorientation	50
2.5.2	Conclusion	50
2.6	Autres approches	51
2.6.1	Association lieu/direction par apprentissage, Gaussier et al.	51
2.6.1.1	Signature d'un lieu	51
2.6.1.2	Appariement	51
2.6.1.3	Calcul de direction	52
2.6.1.4	Commentaires	52
2.6.1.5	Conclusion	52
2.6.2	Rendre le champ de vecteurs plus conservatif, Bianco et al.	53
2.7	Tableau synthétique	53
2.8	Conclusion de la première revue	55
3	Analyse approfondie et généralisation	57
3.1	Composante centripète et environnements testés	58
3.2	Composantes tangentielles et radiale, et désorientation	59
3.2.1	Conséquence d'une désorientation sur la composante tangentielle	60
3.2.2	Indépendance de la composante radiale vis-à-vis de l'orientation	62
3.3	Redéfinition et généralisation	66
3.3.1	Appariement	67
3.3.2	Direction à suivre	68
3.3.3	Méthodes directe, inverse et moyenne	68
3.4	Différences fondamentales	70
3.4.1	Rappel de géométrie : arc capable d'un angle donné	70

3.4.2	Composante tangentielle	71
3.4.2.1	Qualité de guidage	74
3.4.3	Composante radiale	75
3.4.3.1	Zone centrifuge avec un amer	75
3.4.3.2	Zone centrifuge avec plusieurs amers	76
3.4.3.3	Composante radiale centripète	76
3.5	Mise en défaut de la composante radiale	76
3.5.1	Problème de satisfaction des hypothèses	78
3.5.2	Illustration expérimentale	78
3.6	Synthèse	79
3.7	Conclusion de l'analyse	81
4	Navigation locale avec orientation	83
4.1	Techniques d'orientation utilisées par les insectes	83
4.2	Flux optique sur la sphère, Nelson & Aloimonos	84
4.2.1	Découpler pour réduire la complexité du problème	87
4.2.2	Signature d'un lieu et appariement	87
4.2.3	Calcul d'orientation et direction à suivre	88
4.2.4	Conclusion	88
4.3	Fondu enchaîné, Chahl & Srinivasan	89
4.3.1	Rotation	90
4.3.2	Translation	90
4.3.3	Analyse de la méthode	90
4.3.4	Analyse théorique	90
4.3.5	Analyse critique	91
4.3.6	Conclusion	91
4.4	Minimiser la somme des écarts, Möller	92
4.5	Modèle de déformation, Franz	92
4.5.1	Hypothèse de distance aux amers constante	93
4.5.2	Hypothèse de répartition isotropique	93
4.5.3	Signature d'un lieu	94
4.5.4	Appariement, orientation et direction	94
4.5.5	Conclusion sur le modèle de déformation	95
4.6	L'algorithme Panama, Röfer	95
4.6.1	Signature d'un lieu	95
4.6.2	Appariement	95
4.6.3	Orientation	98
4.6.4	Direction à suivre	100
4.6.5	Conclusion	101
4.7	Parcourir les lignes iso-ALV, Ruchti	102
4.8	Reconstitution 3D, Benosman et al.	103

4.8.1	Signature d'un lieu	103
4.8.2	Appariement	103
4.8.3	Calcul de direction	103
4.8.3.1	Calibration préalable	103
4.8.3.2	Estimation du déplacement 3D	105
4.9	Conclusion	107

II Notre modèle **109**

5 Notre modèle, lorsque l'orientation est connue **113**

5.1	Signature d'un lieu : segmenter le panorama en couleurs	113
5.1.1	Utiliser la couleur	114
5.1.2	Maintenir le découpage en éléments distincts	114
5.2	Appariement par programmation dynamique	114
5.2.1	Limites de l'appariement originel	115
5.2.2	Cahier des charges	115
5.2.3	La programmation dynamique	116
5.2.4	Application aux appariements de panoramas	117
5.2.4.1	Matrice de ressemblances	117
5.2.4.2	Remplissage de la matrice des appariements potentiels	118
5.2.4.3	Extraction de l'appariement	119
5.2.4.4	Petit exemple concret	119
5.2.5	Bénéfices de cette méthode	120
5.2.5.1	Prise en compte des informations de l'ensemble des amers	120
5.2.5.2	Prise en compte des couleurs des secteurs à apparier	120
5.2.5.3	Fonctionne avec des panoramas d'orientations différentes	120
5.2.5.4	Fonctionne avec des panoramas incomplets	120
5.2.6	Complexité calculatoire	121
5.2.7	Adaptation au cas circulaire	121
5.2.8	Conclusion sur l'appariement	124
5.3	Direction à suivre : méthode par « élimination »	126
5.3.1	Définition	127
5.3.2	Formalisation de la méthode par élimination	128
5.3.3	Analyse théorique de performances	130
5.3.4	Étude expérimentale	132
5.3.5	Conclusion sur la méthode par élimination	136
5.4	Considérations sur la longueur des pas à effectuer	136
5.4.1	Étude théorique du gain	137
5.4.1.1	Composante tangentielle	137
5.4.1.2	Composante radiale	139

5.4.1.3	Conclusion de l'analyse théorique	141
5.4.2	Agrégation des données de plusieurs amers	142
5.4.3	Algorithme pour déterminer en ligne les gains à appliquer	142
5.4.3.1	Solution pragmatique figée	142
5.4.3.2	Solution adaptative : ralentir sur les courbes	142
5.4.3.3	Ajuster la proportion de réduction des parallaxes	144
5.4.3.4	Conclusion sur le gain	145
5.5	Critère d'arrêt : comment l'animat sait-il quand il est arrivé ?	146
5.6	Conclusion	147
6	Le compas visuel	149
6.1	Trouver séparément l'orientation de l'animat	150
6.2	La méthode de l'étoile polaire	150
6.2.1	Définition de la méthode d'orientation par étoile polaire	150
6.2.1.1	Principe, dans un cas idéal	150
6.2.1.2	Cas réel et implémentation	151
6.2.2	L'étoile polaire suffit pour naviguer dans certains cas	151
6.2.3	Quelques remarques	152
6.2.4	Situation respective de l'étoile polaire et du compas visuel	153
6.3	Fonctionnement interne	153
6.3.1	Découpler l'orientation et la position	153
6.3.2	Vue d'ensemble	154
6.3.2.1	Notations	154
6.3.2.2	Aperçu concret du fonctionnement	154
6.3.3	Étape d'observation	155
6.3.3.1	Approximation de linéarité des parallaxes	155
6.3.3.2	Hypothèses et déroulement	156
6.3.3.3	L'invariant	156
6.3.3.3.1	Définition	156
6.3.3.3.2	Obtention des coefficients	159
6.3.3.4	Conclusion pour l'étape observation	159
6.3.4	Étape exploitation	160
6.3.4.1	Nécessité d'un algorithme d'appariement performant	160
6.3.4.2	Pré-orientation après appariement	160
6.3.4.3	Orientation fine grâce à l'invariant	161
6.3.4.3.1	Calcul de l'orientation de l'animat	161
6.3.4.3.2	Démonstration	161
6.3.5	Illustration par un cas concret simplifié	164
6.3.6	Utilisation de plusieurs observations pour étendre la couverture du compas	165
6.4	Étude théorique	167
6.4.1	Coût computationnel	167

6.4.2	Cas de non-applicabilité	167
6.4.2.1	Un triplet donné n'est pas utilisable partout	168
6.4.2.2	Comment choisir un bon triplet d'amers	169
6.4.3	Quand plusieurs triplets sont disponibles	170
6.4.3.1	Agrégation des estimations par plusieurs triplets	170
6.4.3.2	Sélection de triplets pour améliorer la robustesse	170
6.4.4	Limites	172
6.4.5	Observations qualitatives sur la robustesse	174
6.4.6	Utilisation complémentaire de l'odométrie et du compas	175
6.5	Étude expérimentale	175
6.5.1	Dépendance par rapport au positionnement des vues d'observation	175
6.5.1.1	Dépendance par rapport aux erreurs de mesures d'angles	178
6.6	Conclusion sur le compas visuel	178
7	Carte topologique support du compas	181
7.1	Pourquoi explorer, pourquoi une carte	181
7.2	Fondements et hypothèses	182
7.3	Description précise de l'algorithme	183
7.3.1	Démarrage	185
7.3.2	Choix d'une nouvelle direction d'exploration	185
7.3.3	Exploration avec évitement d'obstacle	187
7.3.4	Vérification d'un lien	188
7.3.5	Étapes d'observation pour le compas visuel	189
7.4	Exemples de cartes construites	190
7.5	Effet d'une perturbation d'orientation	190
7.6	Bilan et extensions possibles	192
7.7	Conclusion	194
III	Expérimentations	197
8	Simulations	201
8.1	Notre simulateur	201
8.1.1	Considérations techniques	202
8.1.2	Monde	203
8.1.3	Vision	203
8.1.4	Mécanique du robot et positionnement	204
8.1.5	Odométrie biaisée	205
8.2	Navigation locale simulée avec visibilité partielle	205
8.2.1	Expérience témoin : modèle original	208
8.2.2	Expérience témoin : modèle original avec programmation dynamique	208
8.2.3	Programmation dynamique et couleur	208

8.3	Navigation avec orientation	209
8.3.1	But visible depuis le point de départ.	209
8.4	Conclusion des expériences simulées	209
9	Expériences robotiques	213
9.1	Matériel et paramètres	214
9.1.1	Robots Pioneer 2	214
9.1.2	Caméra couleur	215
9.1.3	GVPP	216
9.1.4	Obtention d'un panorama via la caméra mécanique	217
9.1.5	Paramètres de la méthode d'appariement	219
9.2	Navigation locale, orientation par odométrie	219
9.2.1	Déroulement de l'expérience	221
9.2.2	Résultats	221
9.2.3	Discussion	222
9.3	Comparaison avec une méthode calibrée	223
9.3.1	Dispositif expérimental	223
9.3.2	Résultats	225
9.3.2.1	Notre méthode de compas visuel	225
9.3.2.2	Méthode calibrée	226
9.3.3	Discussion	226
9.4	Modifications des paramètres	227
9.4.1	Capteur panoramique	227
9.4.2	Salle expérimentale	227
9.4.3	Segmentation par comparaison continue d'écart-type	229
9.4.4	Modifications concernant l'appariement	231
9.5	Test analytique du compas visuel : orientation seule	232
9.5.1	Déroulement de l'expérience	232
9.5.2	Résultats	232
9.6	Test synthétique du compas visuel : navigation avec orientation	234
9.6.1	Méthode expérimentale	234
9.6.2	Déroulement des expériences	234
9.6.2.1	Résultats et interprétation	236
9.7	Limites et viabilité dans d'autres contextes	239
9.8	Conclusion sur les expérimentations robotiques	242
IV	Discussion	243
10	Discussion	245
10.1	Comparaisons avec d'autres approches	245
10.1.1	Considération d'ensemble	245

10.1.1.1	Signature d'un lieu	245
10.1.1.2	Choix d'hypothèses faibles ou fortes	246
10.1.2	Comparaison avec le modèle de Nelson et Aloimonos	247
10.1.2.1	Rapprochement avec la méthode par élimination	247
10.1.2.2	Rapprochement avec le compas visuel : découplage	248
10.1.2.3	Conclusion	248
10.1.3	Complémentarité avec l'ALV	249
10.1.4	Comparaison avec le modèle de Gaussier	249
10.1.4.1	Navigation	249
10.1.4.2	Orientation	250
10.1.4.3	Conclusion	250
10.1.5	Complémentarité avec le modèle de déformation de Franz	251
10.1.6	Comparaison avec la méthode de Benosman et al.	252
10.1.7	Comparaison avec l'algorithme Panama de Röfer	253
10.1.7.1	Signature d'un lieu	253
10.1.7.2	Appariement	254
10.1.7.2.1	Cas des panoramas partiels	254
10.1.7.2.2	Pré-orientation	254
10.1.7.2.3	Coût computationnel	254
10.1.7.3	Calcul d'orientation	255
10.1.7.4	Liens entre Röfer, PV et la méthode par élimination	256
10.1.7.5	Conclusion de la comparaison avec le modèle Panama	256
10.2	Bilan	257
10.3	Perspectives	258
10.3.1	Estimation plus fine de la direction	259
10.3.2	Extensions de l'usage de la carte	260
10.3.3	Alternatives à la segmentation de panoramas en couleurs	260
10.3.4	Retour vers la biologie	261
10.3.5	Le projet Psikharpax	261
11	Conclusion	263
A	Généalogie et classification des modèles	265
A.1	Généalogie	265
A.2	Classification	265

Chapitre 1

Introduction

Il est étonnant de constater les tâches que mènent à bien certains animaux malgré leur petitesse. Une abeille est capable de trouver son chemin en trois dimensions dans un environnement naturel changeant, malgré les mouvements de l'air, les variations d'intensité et de couleur de la lumière et de la végétation, la direction changeante du soleil et donc des ombres. Elle peut ainsi explorer l'environnement en quête de pollen, à une distance de dix mille fois sa taille et par des chemins indirects, puis retrouver le chemin jusqu'à la ruche. Mieux, elle est capable de transmettre à ses congénères la position de la source de nourriture, que d'autres abeilles vont alors exploiter. Pendant ce temps, des abeilles spécialisées se coordonnent, digèrent le pollen, stockent le miel, construisent des alvéoles pour les larves, nourrissent les jeunes, défendent la ruche contre les intrus, ou jouent d'autres rôles.

À l'heure actuelle, certaines réalisations mises en place par l'homme dépassent les meilleurs animaux en vitesse ou puissance, mais celles qui *fonctionnent* sans l'intervention de l'homme ont encore beaucoup à envier aux prouesses animales. Par exemple, les robots industriels réalisent des tâches répétitives à l'infini, mais seulement en environnement extrêmement contrôlé, et ne sont pas plus intelligents que le mécanisme d'une horloge. À l'autre bout du spectre on a cru pertinent de considérer l'ordinateur joueur d'échecs. Or, la recherche de solutions efficaces à ce type de problèmes n'a pas grand chose à voir avec la capacité à évoluer de façon autonome dans un environnement physique non préparé.

Du côté de la robotique autonome, l'autonomie énergétique se mesure typiquement en minutes, au mieux en quelques heures. L'exploration autonome de l'environnement, qui en dépend, ne peut qu'être esquissée. Les robots dont la morphologie permet de se déplacer en forêt ou ne serait-ce qu'emprunter un escalier sont extrêmement rares. L'autonomie motivationnelle complète ne peut émerger qu'avec une longue expérience du monde par la machine, et d'autres pré-requis que nous avons bien du mal à apercevoir.

L'approche animat [MEY 95, MEY 96, MEY 97] nous semble particulièrement intéressante. Elle capitalise sur l'inspiration animale, méthode ayant fait ses preuves dans de nombreux domaines. Elle cherche à des problèmes concrets des solutions simples et efficaces, et grâce au travail de recherche, adaptables et réutilisables dans d'autres contextes. Ainsi, les systèmes de classeurs montrent qu'un

mécanisme d'apprentissage autonome est plus général et parfois plus performant ou plus simple à réaliser qu'un système manuellement prédéterminé et figé [GÉR 02]. Les algorithmes évolutionnistes nous montrent comment des solutions adaptées à un problème donné dans un environnement donné peuvent émerger de l'histoire collective des individus qui ont été sélectionnés, histoire « imprimée » dans un code génétique [DON 03]. Les modèles de navigation utilisant une carte cognitive s'inspirent par exemple de ce qui se passe dans le cerveau d'un rat qui se déplace dans son environnement [FIL 01].

Navigation

Les travaux que nous présentons dans cette thèse ne font pas intervenir d'évolution ni d'apprentissage au sens technique du terme, mais concernent la navigation¹. Si la capacité à se déplacer est une médaille, le problème de la navigation en est le revers indissociable. C'est un des premiers problèmes qui se posent à un robot mobile, avec l'autonomie énergétique et la sélection de l'action. Notons qu'il y a une interdépendance entre ces problèmes. Ainsi, le choix d'une direction à suivre est une forme particulière et restreinte de sélection de l'action. Symétriquement l'espace des états possibles dans un problème général de sélection de l'action, et la nécessité de passer par des états intermédiaires pour réaliser une action de haut niveau n'est pas étrangère à la nécessité de passer par des lieux intermédiaires pour naviguer d'un point à un autre non directement accessible.

La navigation est un problème complexe car elle fait intervenir la structure tridimensionnelle de l'espace environnant : la perception courante de l'animat dépend à la fois de sa position, de son orientation, et des variations de l'environnement, surtout en environnement réel. Pour se déplacer, l'animat doit être équipé d'actionneurs convenables, comme des roues ou des pattes, et la conséquence d'une action dépend de la position et de l'orientation de l'animat. C'est aussi un problème difficile : même si Braitenberg a montré [BRA 84] qu'on peut obtenir des comportements intéressants à l'aide des animats mobiles les plus simples qu'on puisse imaginer, l'obtention de capacités de haut niveau comme la localisation [MEY 03] et la cartographie [FIL 03] simultanées n'a que des solutions partielles à l'heure actuelle.

La navigation peut s'appuyer sur différentes modalités sensorielles, comme le toucher (utilisation de vibrisses), l'ouïe (le principe des sonars utilisés en robotique est comparable à celui des chauves-souris), ou la vue (utilisation d'une caméra). Nous avons choisi de nous focaliser sur cette dernière modalité. La navigation peut se dérouler sur une surface, plane ou non, ou bien en trois dimensions pour un animat volant. Les modèles sur lesquels nous sommes appuyés représentent une simplification à deux dimensions d'idées applicables à la navigation en trois dimensions. Enfin, l'utilisation de la vue seule ne permet pas toujours de déterminer facilement et précisément la distance aux éléments qui composent l'environnement. Constatant que des modèles existants, conçus initialement par des biologistes étudiant la navigation des abeilles, permettaient à un animat de naviguer localement sans connaître aucune distance ni posséder de carte de l'environnement, nous avons exploré plus loin

¹Le mot navigation vient du latin *navis*, navire, et *agere*, conduire.

cette voie de recherche sur la navigation par mesures d'angles.

Afin de préparer le lecteur à l'exposé qui va suivre, nous devons aborder quelques notions relatives à la navigation. Nous nous restreindrons aux seules notions qui sont nécessaires à la suite de l'exposé. Levitt et Lawton [LEV 90] définissent le problème de la navigation en trois points : où suis-je, où sont les autres endroits par rapport à moi, comment m'y rendre ? Pour répondre à ces questions, des sous-problèmes se posent. Le premier sous-problème est qu'il est nécessaire de percevoir des éléments autour de soi et d'en extraire une sélection d'informations pertinentes. Lorsque l'animat perçoit de son environnement, mémorise et compare, fait intervenir des éléments de l'environnement que l'expérimentateur peut identifier, il les nomme en général amers². Le second sous-problème est qu'il est nécessaire de repérer le contexte dans lequel on se trouve, par comparaison entre les informations actuellement perçues et celles qu'on a pu mémoriser par le passé. Nous utilisons tout au long de notre thèse des vues panoramiques. Une vue panoramique est un ensemble de mesures d'informations visuelles réparties circulairement autour de l'animat. Elle sert de signature du lieu d'où elle est prise, et la comparaison avec la vue d'un lieu voisin permet d'estimer la position relative de ces deux lieux. Enfin, le troisième sous-problème est que, muni de cette comparaison, il faut décider de la direction à prendre et envoyer des ordres aux moteurs permettant de réaliser le mouvement voulu. Alors la boucle environnement-perception-action est bouclée et l'animat peut s'animer.

Pour se faire une idée claire des capacités des différentes méthodes de navigation existantes, il est pertinent de s'appuyer sur une classification hiérarchique en cinq niveaux introduite par Trullier [TRU 97] : approche d'un objet, guidage local³, association lieu-action⁴, navigation topologique, navigation métrique. Seul les quatre premiers nous concernent ici.

L'approche d'un objet consiste simplement en la capacité à rejoindre un objet visible. Il suffit de se tourner en direction de l'objet puis d'avancer.

Le guidage local est l'élément essentiel utilisé pour naviguer dans cette thèse. Il s'agit de la capacité à rejoindre un point qui n'est pas directement défini par un objet matériel dont il faut s'approcher, mais par la disposition des amers autour de ce point. Elle nécessite que l'animat soit suffisamment proche du but pour que la comparaison des panoramas actuel et mémorisé soit pertinente.

L'association lieu-action consiste à définir des régions de l'espace – reconnaissable par les éléments de l'environnement qu'on y perçoit – chacune associée à une direction à suivre qui mène au lieu suivant puis finalement au but. Si l'animat arrive en un de ces lieux, il peut parcourir la chaîne et aboutir au but, même si le but n'est pas visible depuis le premier point. Cependant, un lieu appris ne permet de joindre qu'un but et l'animat ne peut pas prendre d'initiative en cas de rupture de la chaîne.

La navigation topologique consiste à mémoriser et représenter les relations de voisinage entre des

²Le mot amer est initialement un terme de navigation maritime : point remarquable, sur une côte par exemple, qui permet de se repérer. En anglais on emploie généralement le terme « landmark ».

³Nous traduisons par « guidage local » le mot anglais « guidance ».

⁴Nous traduisons par « association lieu-action » l'expression « place recognition triggered response ».

lieux, formant ainsi une carte de l'environnement. Parce qu'un lieu n'est plus associé à un but unique, elle permet à l'animat de planifier des déplacements variés en réutilisant les mêmes nœuds. N'importe lequel des lieux représentés dans la carte peut devenir un but. Toutefois, l'absence d'information métrique signifie que l'animat ne peut estimer la distance réelle à parcourir pour rejoindre un point, ou le chemin le plus court parmi plusieurs options, ni planifier un chemin passant par une zone non encore explorée de l'environnement.

Notre contribution

Au cours de notre travail, nous nous sommes tenu à une certaine forme de minimalité. Des hypothèses de travail minimales sont avantageuses, parce qu'ainsi on évite des hypothèses non essentielles qui apportent peu mais peuvent restreindre beaucoup la généralité des techniques employées.

Par ailleurs, nous avons choisi de nous intéresser occasionnellement aux résultats des recherches biologiques comme une source d'inspiration, sans nous contraindre pour autant à réaliser des systèmes biologiquement plausibles.

Notre travail prend appui sur des modèles initialement introduits par les biologistes Cartwright et Collett [CAR 83] et sur les travaux qui les ont suivis. Il s'agissait au départ d'expliquer le comportement des abeilles dans un problème de type « retour au nid »⁵, où une vue panoramique prise depuis un point intéressant est utilisée pour faire du guidage local.

Les modèles initiaux, intéressants par leur simplicité et la plupart de ceux qui ont suivi restent trop schématiques pour être applicables à la robotique. Nous avons donc consolidé les solutions aux sous-problèmes évoqués plus haut (sélection des informations, comparaison, estimation de la direction à prendre).

Nous avons constaté le rôle particulier et important de l'orientation. Il nous a semblé possible d'estimer l'orientation à partir d'informations visuelles sans modèle géométrique de l'environnement ni même d'estimation de distance aux amers environnants, tout en s'abstenant d'hypothèses nécessaires dans d'autres modèles. Nous n'avons pas non plus supposé que les mouvements que l'animat commande sont précisément réalisés, ni en direction ni en longueur.

Cela nous a conduit à développer un compas visuel, librement inspiré par la biologie. Utilisant seulement des vues panoramiques sans information de distance, il n'a pas besoin de télémètre laser ni de compas magnétique ou autres capteurs.

Cette thèse expose les résultats de recherche suivants :

- une analyse des deux mécanismes fondamentaux utilisés par les modèles de retour au nid étudiés. En particulier, une démonstration de viabilité du premier mécanisme en cas de légère désorientation et une contre-démonstration illustrant la faiblesse relative de l'autre mécanisme ;
- une méthode pour apparier des éléments dans deux vues d'un même environnement réel, fondée sur l'exploitation d'informations discriminantes (dans notre cas, essentiellement la couleur)

⁵ « Retour au nid » correspond à « homing », en anglais

et utilisant la programmation dynamique, en particulier dans le cas où les vues ne sont pas de même orientation, voire d'orientation inconnue. Cette méthode permet à l'animat de s'affranchir de la contrainte d'orientation connue voire constante qui caractérise la plupart des modèles sur lesquels nous nous sommes appuyé ;

- l'implémentation et l'expérimentation de cette méthode pour faire naviguer localement un animat simulé puis un robot réel. Les performances de la méthode sont comparées aux performances de l'algorithme initial de Cartwright et Collett ;
- une méthode originale pour calculer avec un minimum d'hypothèses le déplacement à suivre pour revenir à un lieu d'où un panorama a été saisi. À la différence des méthodes précédentes, elle permet de constater si certaines hypothèses implicites dans les modèles précédents (orientation connue, environnement statique) sont vérifiées ;
- l'implémentation et l'expérimentation de cette méthode pour faire naviguer localement un robot simulé. Les performances de la méthode sont comparées avec le modèle PV, un successeur de l'algorithme initial de Cartwright et Collett ;
- deux méthodes de conception et de réalisation simples, mathématiquement fondées et légères en calcul, pour estimer l'orientation actuelle d'un animat dans un contexte préalablement exploré, utilisables pour le retour au nid. La première méthode, appelée « méthode de l'étoile polaire », est approximative, mais nous démontrons qu'elle est suffisante dans certains cas. La seconde méthode, que nous avons appelée compas visuel, peut être considérée comme un raffinement de la première prenant une « empreinte » de l'orientation en trois points d'observation. Cette empreinte permet à l'animat de se réorienter chaque fois qu'il se trouve dans le voisinage de ces trois points, en plus de permettre le retour au nid ;
- l'implémentation et l'expérimentation de cette méthode en simulation et sur robot réel ;
- une méthode d'exploration et de construction d'une carte topologique de l'environnement, illustrant comment l'animat peut, de façon autonome, obtenir des vues permettant au compas visuel de fonctionner et donc maintenir une navigation viable en compensant grâce au compas une dérive de son orientation.
- l'implémentation et l'expérimentation de cette méthode en simulation.

Plan du mémoire

Dans le chapitre 2, nous présentons une revue de méthodes de navigation par mesure d'angles qui font l'hypothèse que l'orientation de l'animat est connue.

Les informations recueillies sont l'occasion d'analyser dans le chapitre 3 les différences essentielles entre deux éléments complémentaires des modèles étudiés jusque-là, de démontrer leurs forces et faiblesses, et d'introduire des éléments sur lesquels nous nous sommes appuyé dans notre modèle.

Dans le chapitre 4, nous continuons notre revue par des modèles apparentés à la navigation par mesure d'angle qui traitent le problème de l'orientation.

Le deuxième partie de la thèse commence par le chapitre 5, où nous commençons à décrire le

sous-ensemble de nos travaux qui ne se préoccupe pas d'orientation.

Dans le chapitre 6, nous exposons le fonctionnement de nos méthodes d'orientation : méthode de l'étoile polaire et compas visuel.

Dans le chapitre 7, nous décrivons notre modèle d'exploration de l'environnement et de construction de carte.

Puis nous abordons la partie expérimentale de la thèse. Dans le chapitre 8, nous décrivons notre simulateur et présentons quelques expériences simulées qui valident la chaîne complète de la perception à l'action.

Dans le chapitre 9, nous rapporterons des expériences réelles, qui nous ont permis de comparer le compas visuel avec une méthode calibrée, et de valider sur robot réel la chaîne complète de la perception à l'action, dans le cas où l'orientation est supposée connue, puis dans le cas où l'animat est soumis à une désorientation qu'il compense à l'aide du compas visuel.

La quatrième et dernière partie de la thèse commence par le chapitre 10, où nous comparerons de façon détaillée nos modèles avec ceux que nous avons présentés dans la première partie.

Enfin, nous concluons. Une annexe présente des schémas d'ensemble permettant de visualiser d'un coup d'œil les relations entre la plupart des modèles présentés dans cette thèse.

Première partie

Problématique

Chapitre 2

Navigation locale utilisant des panoramas

Nous considérons un animat qui perçoit son environnement sous la forme de panoramas. Cet animat est face à un problème de « retour au nid » : une vue panoramique, prise par le passé depuis un point intéressant, est utilisée pour faire du guidage local. Il détermine le chemin à suivre par comparaison des panoramas.

Dans ce chapitre nous dressons une revue critique des modèles pré-existants, qui supposent que l’animat connaît son orientation par ailleurs. Par contraste, dans le chapitre 4 nous passerons en revue des modèles qui incorporent le problème de l’orientation.

Nous avons choisi de présenter les équipes et leurs travaux apparentés dans l’ordre dans lequel ont été introduites les principales améliorations apportées aux modèles.

Avant d’exposer les modèles eux-mêmes, nous commençons par introduire les notions sur lesquelles nous allons nous appuyer.

2.1 Structure commune à la plupart de ces modèles et grille d’analyse

Nous dressons ici le squelette commun à la plupart de ces modèles, afin de clarifier la suite de l’exposé. Nous utiliserons cette structure comme « grille d’analyse », ce qui facilitera la compréhension et la mise en correspondance des différents éléments des modèles étudiés. Le schéma 2.1 résume ce qu’est une tâche de navigation locale.

2.1.1 Signature d’un lieu

Il est nécessaire de définir quels éléments de l’environnement l’animat peut percevoir et mémoriser pour réaliser sa tâche de navigation.

Nous appelons « signature d’un lieu » l’ensemble des éléments retenus lorsque l’animat observe la situation en un point donné. Les modèles que nous étudions ici se concentrent sur des informations

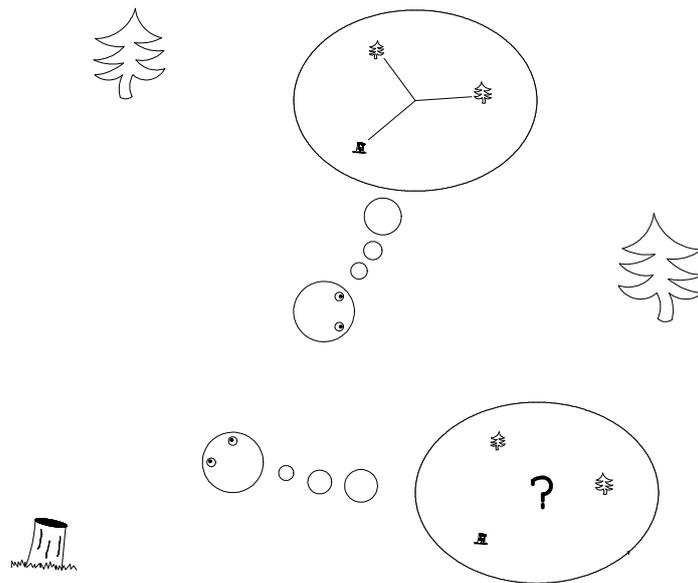


FIG. 2.1: Guidage local. Un animat (au centre de la figure) se trouve en un endroit intéressant pour sa survie. Il observe son environnement et mémorise la signature du lieu où il se trouve. Plus tard, il est (en bas) à proximité du lieu et souhaite le rejoindre. La caractéristique du guidage local est que la perception actuelle de l'animat ressemble suffisamment à la perception mémorisée pour qu'une comparaison des deux suffise à estimer la direction à suivre. La navigation par mesure d'angles est un exemple de guidage local.

visuelles. Selon les modèles, la signature d'un lieu peut être une image bidimensionnelle, ou bien seulement un panorama unidimensionnel. Les informations représentées peuvent être de type binaire (comparaison de la luminance observée à un seuil, indiquant un obstacle ou l'absence d'obstacle), ou bien plus nuancées, en niveaux de gris ou dans divers espaces de couleurs. L'information peut être échantillonnée régulièrement tout autour de l'animat, ou bien regroupée en secteurs de largeurs inégales et qui correspondent aux largeurs apparentes des objets environnants. Dans le premier cas, on parle de panorama échantillonné. Dans le second on parle de panorama segmenté.

L'essentiel, qui justifie cette appellation de « signature », est que cette information mémorisée depuis un point de l'environnement permet à l'animat de revenir plus tard en ce point, grâce à la comparaison entre la vue instantanée actuelle et la vue mémorisée.

Cette signature permettra en général d'identifier des amers. Selon le cas, un amer peut être étendu (correspondre à un secteur dans le panorama), ou bien ponctuel (correspondre à une transition entre deux secteurs du panorama).

2.1.1.1 Notations

Dans la plupart des modèles suivants, l'animat est supposé avoir un système visuel capable de mesurer des angles relativement à son corps, et de mémoriser la position des objets environnants dont l'image se projette sur sa rétine modélisée par un cercle. En outre, il est aussi supposé capable

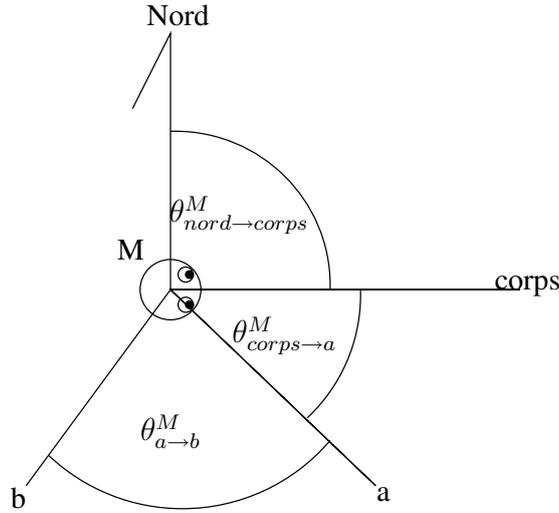


FIG. 2.2: Notation introduite pour les angles, en vue aérienne de dessus. L'animat est représenté au centre. L'angle entre deux directions a et b de sommet M est noté $\theta_{a \rightarrow b}^M$. Les directions a et b peuvent correspondre à deux amers ponctuels ou bien aux deux bords d'un amer étendu. Nous introduisons aussi deux éléments particuliers pouvant intervenir dans les angles : *nord* et *corps*. L'angle entre le *nord* et un élément i est appelé *azimut* de i . Les angles sont signés, avec la convention trigonométrique. Ainsi, les trois angles θ explicités sur le schéma sont négatifs.

de connaître la direction actuelle de son corps (ou bien de maintenir son corps dans une direction constante), de sorte que les angles mémorisés le sont par rapport à une véritable référence, et non simplement relatifs à une direction accidentelle du corps de l'animat.

Pour mesurer des angles, nous utilisons la notation $\theta_{i \rightarrow j}^M$ pour désigner l'angle signé de sommet M et couvrant les directions de i à j (voir la figure 2.2).

Du point de vue de l'animat, cela se traduit ainsi : étant donnée une vue prise depuis un lieu X , nous noterons $\theta_{i \rightarrow j}^X$ l'angle signé entre la direction pointant vers i et la direction pointant vers j , où i et j peuvent désigner des amers comme a ou b , où des directions de référence comme *nord* ou *corps*.

Nous appellerons « azimut d'un point i depuis la position M », la grandeur $\theta_{nord \rightarrow i}^M$, c'est-à-dire la direction de la demi-droite joignant le centre de M (la rétine de l'animat) au point i de l'environnement, relativement au nord.

Pour une vue donnée et un amer étendu i , qui occupe dans le panorama le secteur de a à b de sommet M , nous appellerons « largeur apparente de l'amer i » la grandeur $\theta_{a \rightarrow b}^M$.

Nous appellerons « parallaxe angulaire d'un élément i au cours d'un déplacement de l'animat de M à X » la variation de l'azimut de i causée par ce déplacement. Ainsi, pour un amer i , lors d'un déplacement de M à N , la parallaxe est $D_i^{M \rightarrow N} = \mu(\theta_{nord \rightarrow i}^N - \theta_{nord \rightarrow i}^M)$ où $\mu(x)$ est le nombre le plus proche de zéro parmi l'ensemble $(x \bmod 2\pi)$.

Ce que nous appelons la parallaxe est aussi appelé « discrepancy » [CAR 83], « displacement » [FRA 98b] ou « offset » [HON 91].

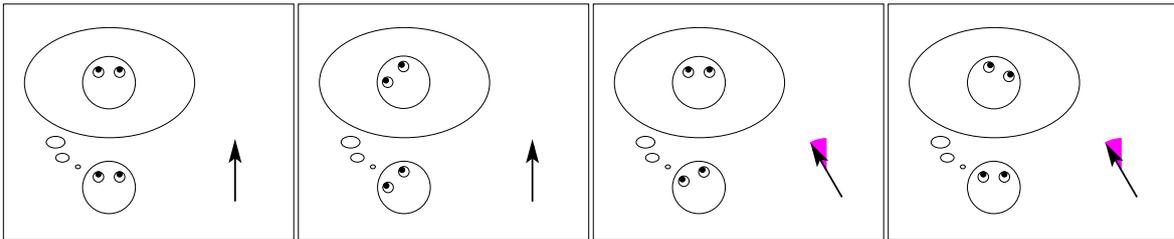


FIG. 2.3: Représentation de la direction de l'animat dans nos simulations. On représente l'animat dans son orientation réelle (indiquée par le regard). On représente aussi dans une bulle l'orientation qu'il estime avoir. À droite de chaque schéma, une flèche et un secteur ombré indiquent s'il y a désaccord entre orientation estimée et orientation réelle ou, ce qui est équivalent, si ce que l'animat croit être le nord est décalé par rapport au nord réel. La flèche noire en harpon est la « boussole mentale de l'animat », ce que l'animat croit être le nord. L'écart qui va du nord réel (haut de la figure) au nord estimé est le même que l'écart qui va de l'orientation estimée à l'orientation réelle.

2.1.1.2 Représentation de l'orientation de l'animat

L'animat ayant sa propre estimation de sa direction et une orientation réelle, il faut, pour que le lecteur puisse interpréter correctement les schémas, une représentation de ces orientations.

Dans toutes nos expériences, les schémas sont représentés avec une direction constante appelée « nord » en haut. Lorsque c'est pertinent, les schémas représentant un panorama ou un appariement de panoramas comprennent des indications sur l'orientation de l'animat, réelle et estimée. Leur interprétation est illustrée en 2.3. Rappelons que les angles sont en convention trigonométrique. Notons l'équivalence entre les propriétés suivantes, où α est commun :

- l'animat connaît correctement son orientation (respectivement, il commet une erreur α dans son estimation) ;
- les deux représentations de l'animat (réelle et estimée), coïncident (respectivement, l'animat dans la bulle est décalé d'un angle α en plus) ;
- la flèche en harpon en bas à droite est alignée avec le haut du schéma, (respectivement, la flèche dérive d'un angle $-\alpha$, mis en évidence par un secteur ombré).

2.1.2 Appariement

Dans cette tâche de retour au nid, ce sont essentiellement les différences entre un panorama mémorisé et le panorama actuel qui permettent à l'animat de déterminer les mouvements à envisager. Certains modèles associent explicitement des éléments des deux panoramas, tandis que d'autres ont recours à des méthodes qui fonctionnent aussi grâce aux différences, mais sans appairer explicitement les éléments des panoramas visuels.

2.1.3 Calcul de direction

Presque tous les modèles étudiés ont pour but de rechercher dans quelle direction se déplacer pour se rapprocher d'un point mémorisé, en choisissant une direction qui tend à faire coïncider la signature courante avec la signature mémorisée.

En passant en revue les correspondances entre panoramas obtenus à l'étape précédente et en comparant les positions angulaires des objets dans le champ de vision, on peut obtenir un ensemble d'indications élémentaires (écarts angulaires) dont la synthèse permet de répondre à la question : « quel(s) déplacement(s) l'animat peut-il effectuer, qui réduirai(en)t les écarts observés ? ». Tous les algorithmes étudiés ici donnent une réponse unique, la direction à suivre. Par contraste, nous introduirons en section 5.3 un algorithme qui donne un intervalle de directions.

2.1.4 Surface de captation (Catchment area) et attracteurs

On dit que le système de navigation réussit lorsque l'animat atteint le but. Mais, comme il s'agit ici de navigation locale, dépendante de la ressemblance de panoramas, la portée est limitée : assez loin du but il existera toujours des points de départ depuis lesquels l'animat n'atteindra pas le but par manque de comparaison pertinente.

On appelle « bassin d'attraction du but »¹ l'ensemble des points de l'environnement depuis lesquels l'animat atteint le but. Il est possible de déterminer approximativement cet ensemble en faisant partir l'animat de nombreux points et en repérant les points depuis lesquels il atteint le but. En pratique, en simulation comme en robotique, on fait partir l'animat de chacun des points d'une grille régulière. Un exemple de bassin de captation est représenté figure 2.4.

Nous ferons référence aux mathématiques des systèmes dynamiques en parlant d'attracteurs et de bassins d'attraction. Nous en donnons ici très brièvement et sans justification quelques propriétés.

Dans tous les modèles étudiés ici, la trajectoire de l'animat résulte d'itérations successives de la procédure faisant un pas dans la direction à suivre à partir de la comparaison entre panorama actuel et panorama mémorisé. C'est donc un système dynamique.

Ce système dynamique est conçu pour que le but ait la propriété particulière d'attirer l'animat : que pour le plus grand nombre possible de points de départ dans l'environnement, la trajectoire de l'animat tende vers le but. On dit que le but est un attracteur. Pour qu'un point soit attracteur, il est nécessaire (mais pas suffisant) que le mouvement commandé en ce point soit nul².

L'ensemble des points qui sont des départs de trajectoires convergeant vers un même point attracteur est appelé bassin d'attraction, ce qui est la raison de la définition plus haut. Idéalement, il y a un attracteur, le but, et son bassin d'attraction est la surface de captation qui nous intéresse.

Malheureusement, il n'y a pas *a priori* de raison que la surface de captation soit convexe ni sans

¹Le terme anglais utilisé dans [CAR 83] est « catchment area ». Une traduction fidèle est « bassin de captation », qui est un terme d'hydrographie (géologie), ou « surface de captation ».

²Nous verrons dans le prochain chapitre l'utilité de cette caractéristique pour analyser le système de navigation.

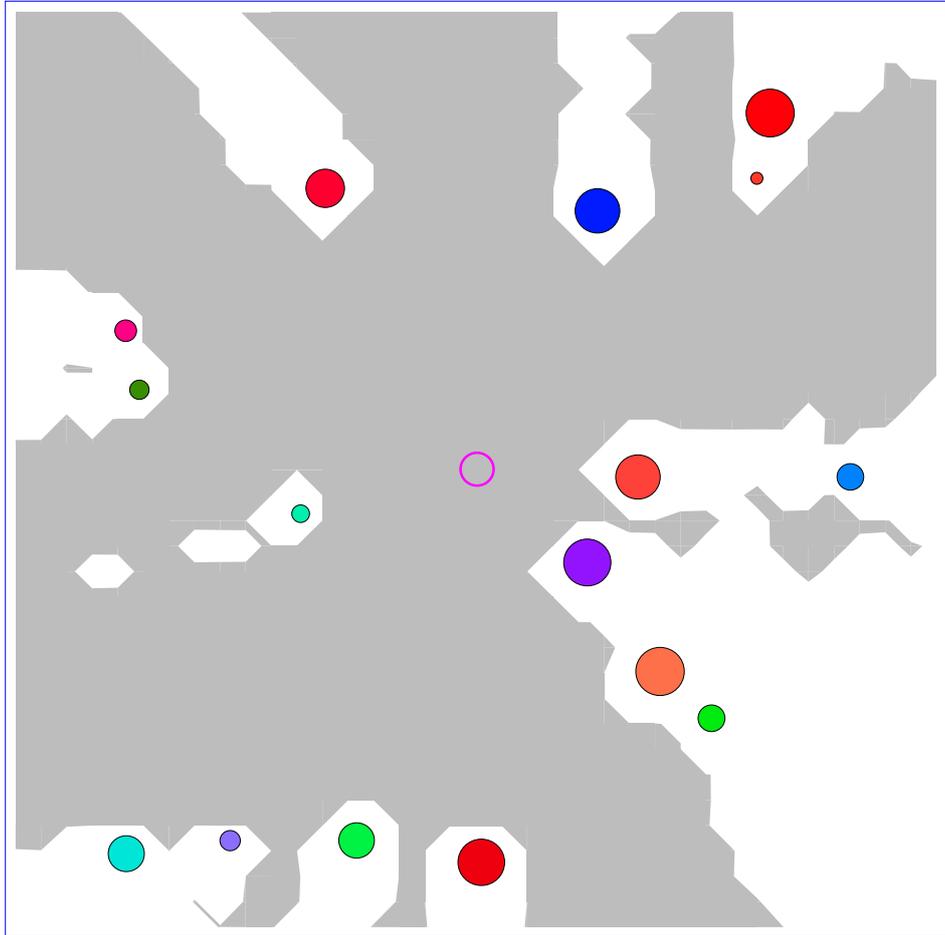


FIG. 2.4: Exemple de bassin d'attraction en simulation. Les disques sont des obstacles dont l'animat perçoit la couleur. Au centre de l'arène, l'animat (cercle) est au nid et a mémorisé un panorama. Puis on a envoyé l'animat en chaque point d'une grille régulière, avec pour tâche de rejoindre le but. On a ensuite entouré de gris les points de départ depuis lesquels l'animat a rejoint le but. La région grise résultante est une représentation approximative du bassin d'attraction de ce but dans cet environnement, avec la méthode particulière utilisée. Chaque amer est entouré de blanc, non parce que la méthode échoue lorsque l'animat est trop près d'un amer, mais parce que la simulation tient compte du diamètre de l'animat. En conséquence, la distance entre le centre de l'animat et le bord d'un amer est toujours supérieure au rayon de l'animat et la zone grise ne touche pas d'amer.

trou. Il peut avec certains modèles exister des points de départ où l'animat se dirigera vers un obstacle ou vers un point qui n'est pas le but mais qui attire à lui toute trajectoire passant à proximité. Ces points sont aussi des attracteurs, mais sont indésirables³.

2.1.5 Difficultés supplémentaires possibles

Comme annoncé, la première revue qui va suivre concerne des modèles dans lesquels on suppose que l'animat a une vision panoramique et connaît son orientation.

L'abandon de l'une ou l'autre de ces hypothèses (ou des deux) perturbe plus ou moins ces modèles. Nous abordons ici brièvement les difficultés qu'une désorientation implique. Au cours de la revue, nous expliquerons pour chaque modèle les implications d'une désorientation. Certains cas seront illustrés plus loin encore par des expériences que nous avons simulées.

2.1.5.1 Rotation mentale

Jusqu'ici nous avons supposé que l'animat pouvait estimer des azimuts et les comparer avec des informations mémorisées. Si l'animat a une direction constante, une mémorisation simple, qui peut même être locale au système visuel et qu'on qualifie alors de *rétinotopique*, peut suffire à mettre en regard deux panoramas. Si l'animat peut tourner sur lui-même, pour pouvoir faire cette mise en regard, il doit soit se replacer dans la même direction pour prendre la vue actuelle, soit être capable de faire une rotation mentale du panorama mémorisé.

2.1.5.2 Désorientation

Nous envisageons le cas où l'animat est désorienté. Dans notre cadre de navigation locale utilisant des panoramas, cela signifie que l'animat ne sait pas comment la direction à laquelle il fait face actuellement se compare avec la direction à laquelle il faisait face au moment où il avait mémorisé la vue du but.

Formellement, cela signifie que l'animat ne pourra plus estimer les azimuts des amers dans la vue actuelle par rapport au même référentiel que dans la vue mémorisée. Les mesures qu'il peut faire sont toutes perturbés (modulo 2π) d'une même grandeur inconnue, par rapport à la même vue prise en connaissant son orientation. De même, l'animat ne pourra estimer les parallaxes vraies dues à son déplacement. Il n'aura accès qu'à des pseudo-parallaxes perturbées de la même façon.

En revanche, les largeurs apparentes ne sont pas perturbées. De façon plus générale, aucune combinaison linéaire d'azimuts dont la somme des coefficients est nulle ne sera perturbée.

³Notons que nous ne parlons pas ici de minimum local, notion pourtant courante. La raison en est qu'un minimum local suppose l'existence d'une fonction potentiel. Or, si le champ de vecteurs n'est pas conservatif, il n'existe pas de fonction potentiel (voir la section 2.6.2 pour quelques mots sur la conservativité).

L'étape d'appariement et l'étape de calcul de la direction à suivre seront perturbées si elles dépendent des parallaxes et pas uniquement, par exemple, des largeurs apparentes. L'appariement peut toujours s'appuyer sur des caractéristiques plus intrinsèques des amers, comme la couleur.

Notons que tout ordre moteur que produit l'animat est défini dans le référentiel de son propre corps. Cela signifie qu'un ordre moteur exprimé par rapport à la vue courante (qui est par définition liée à l'orientation actuelle du corps de l'animat) reste toujours bien défini. En revanche, si l'animat est désorienté, tout ordre moteur exprimé par rapport au référentiel du panorama mémorisé n'est plus bien défini. De façon plus concrète, cela signifie que, si l'animat est désorienté, il doit s'abstenir de faire dépendre ses ordre moteurs des directions du panorama mémorisé. Il ne peut s'appuyer que sur les directions du panorama actuel.

2.1.5.3 Vision non totalement panoramique

Nous considérons brièvement le cas d'une vision non complètement panoramique, c'est-à-dire qui ne couvre pas tout le cercle car, bien que la plupart des modèles étudiés ici en fassent l'hypothèse, la plupart des animaux et des robots n'en disposent pas, et nous n'en disposons pas au début de notre thèse.

Si l'animat a une orientation constante au cours de son déplacement⁴, et que son déplacement est tel que la plupart des amers vus du but sont aussi visibles lors du retour au but, alors certains modèles peuvent fonctionner. Nous avons utilisé cette propriété dans nos premières expériences robotiques.

Dans les autres cas, le problème est plus difficile. En effet, une même position dans l'environnement peut présenter une apparence très différente selon la direction dans laquelle l'animat regarde. Si l'animat ne dispose pas d'une vision panoramique, l'ensemble des amers vus n'est pas invariant par rotation de l'animat. Il se peut même qu'aucun amer ne soit commun entre deux vues prises d'un même point. Cela signifie que, dans certains cas, aucune méthode ne peut, avec deux images pour seules données, déterminer si elles sont prises du même point ou calculer un déplacement à effectuer.

L'hypothèse de vision panoramique évite des difficultés importantes, et nous avons fait le choix de la conserver dans cette thèse. Ceci écarte *a fortiori* le cas combiné d'une vision non totalement panoramique avec orientation inconnue.

2.1.6 Comment étalonner les modèles ?

Afin de comparer les modèles autrement que qualitativement, il est bon de se donner des outils et des mesures.

Il nous est apparu que deux approches étaient possibles pour essayer de mesurer les performances d'un modèle. La première, *analytique*, consiste à mesurer séparément le comportement de chacune des étapes des modèles, en les comparant à des réponses théoriques idéales. La seconde, plus *synthétique*,

⁴Ruchti parle d'*agent directionnel* [RUC 00].

consiste à ignorer les détails et à se concentrer sur l'efficacité d'ensemble, c'est-à-dire la proportion de points de l'environnement où l'animat atteint effectivement le but avec une certaine précision, la longueur du chemin parcouru, et éventuellement d'autres propriétés de la trajectoire.

Cependant, la première méthode pose quelques difficultés. Les mesures les plus naturelles à faire ne sont pas forcément satisfaisantes. Ainsi, il est possible en simulation de vérifier rigoureusement si une méthode d'appariement fonctionne bien ou se trompe, ou bien de comparer la direction indiquée par l'algorithme à la direction qui va droit au but. Ces mesures ne sont pas réellement satisfaisantes, d'une part parce qu'une méthode peut parfois s'accommoder d'un appariement mauvais dans certains cas (dans le sens où sa performance globale n'est pas plus mauvaise), d'autre part parce que la méthode qui va droit au but va parfois buter sur un obstacle ou un ensemble d'obstacles et donc échouer, là où une méthode qui fait des détours aura une meilleure performance globale.

L'esprit des mesures analytiques est bien résumé par la question « Cette méthode estime-t-elle précisément dans quelle direction se trouve le but ? », tandis que l'esprit des mesures synthétiques est mieux résumé par « Cette méthode recommande-t-elle des directions qui permettent à l'animat de rejoindre le but ? ». Notons que s'il n'y a pas d'obstacle sur le chemin, la réponse aux deux questions est la même (le plus court chemin est la ligne droite). Mais s'il y a des obstacles sur le chemin le plus direct, ce n'est plus le cas.

Le cas particulier de la méthode par élimination que nous introduirons en section 5.3 est éloquent : la direction calculée par cette méthode, s'il n'y a pas d'erreur en amont, diminue la distance entre l'animat et le but. Dans les cas où il est nécessaire de faire un détour parce qu'un obstacle se trouve entre l'animat et le but, cette méthode enverra l'animat sur l'obstacle. En d'autres termes, cette méthode peut, s'il y a des obstacles à contourner, obtenir d'excellents résultats analytiques et de mauvais résultats synthétiques. Il peut être nécessaire d'ajouter un évitement d'obstacle extérieur à la méthode pour obtenir des trajectoires qui mènent bien l'animat au but.

Chaque méthode de mesure ayant ses avantages et ses inconvénients, nous avons trouvé que les deux pouvaient être intéressantes et nous avons utilisé l'une et l'autre. Weber et al. [WEB 99] utilisent des statistiques synthétiques, comme nous le verrons en section 2.5. Franz et al. [FRA 98b] utilisent les deux sortes de statistiques (section 4.5).

2.1.6.1 Obtention de statistiques

Pour obtenir des statistiques d'efficacité, on se donne typiquement, d'une part, une méthode fixée et, d'autre part, un ensemble d'environnements générés aléatoirement (de mêmes paramètres, ou de paramètres variés comme le nombre d'amers, leur taille, leur répartition) ainsi qu'un ensemble de points de départ (aléatoires ou alignés sur une grille) et de points d'arrivée (par exemple aléatoires mais un point d'arrivée unique est possible).

On applique ensuite l'élément à tester (appariement, calcul de direction, ou bien trajectoire complète) depuis chacun des points de départ dans chacun des environnements. Dans le cas où l'on fait depuis chaque point une trajectoire complète, on peut obtenir une surface de captation en se donnant

un critère de réussite comme, par exemple, le fait que la distance entre le point d'arrivée effectif et la cible est inférieure à un seuil.

2.1.6.2 L'appariement

Dans une vue donnée, tous les amers ne sont pas forcément visibles. Entre la vue du but et la vue actuelle, certains amers ont disparu et d'autres sont apparus. Pour pouvoir estimer la qualité d'un appariement, on se donne en simulation une méthode particulière d'appariement dite « appariement parfait ». Cette méthode connaît l'identité véritable de chaque amer, se contente de passer en revue les amers de la vue de référence, et apparie chacun avec le même amer dans l'autre vue si et seulement s'il est visible. Bien sûr, cette méthode « triche » par rapport aux informations accessibles à l'animat mais elle fournit une référence.

2.1.6.3 La direction à suivre : mesure de composante centripète

Pour estimer analytiquement la performance d'une méthode de navigation, on ne l'applique qu'une fois depuis chaque point de la grille de test ⁵.

Une mesure intéressante, utilisée par Franz est appelée « composante centripète »⁶ [FRA 98b]. Cette mesure s'intéresse à la composante du déplacement qui approche l'animat du but, c'est-à-dire

$$c = \cos(\theta_{\text{méthode testée}} - \theta_{\text{direction menant droit au but}})$$

c est toujours entre -1 et 1, et les valeurs s'analysent facilement. Voici l'interprétation des valeurs en terme de trajectoire de l'animat :

	interprétation	si c constant le long de la trajectoire
$c = 1$	avancée face au but	ligne droite vers le but
$0 < c < 1$	approche du but	approche en spirale
$c = 0$	à angle droit du but	arc de cercle à distance constante du but
$-1 < c < 0$	éloignement du but	éloignement en spirale
$c = -1$	l'animat avance dos au but	ligne droite

On peut représenter un ensemble de mesures sous la forme d'un nuage de point, représentation que nous trouvons particulièrement claire. Un exemple est donné en figure 2.5.

⁵Par contraste, la mesure synthétique consistera à faire partir de chaque point de la grille une trajectoire complète en appliquant tout le modèle.

⁶Nous traduisons par « composante centripète » l'expression « homeward component », qui se traduirait mot à mot par « composante en direction du nid ».

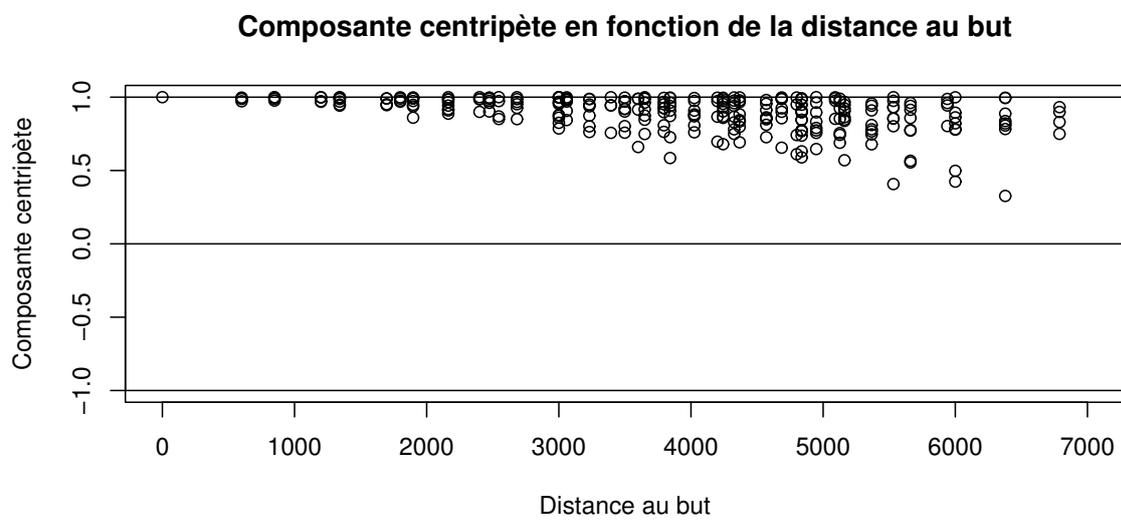


FIG. 2.5: Exemple de mesure analytique de performance : une représentation de la composante centripète. Dans une expérience simulée donnée, pour chaque point de l'environnement où l'animal a fait appel au système de navigation, on place un point dans ce schéma, dont l'abscisse est la distance au but, et l'ordonnée la composante centripète de la direction à suivre calculée par le modèle (voir texte). 1 signifie que la direction va droit au but ; 0 signifie que la direction est orthogonale à celle du but ; -1 signifie qu'elle est à l'opposé de celle du but. Nous avons choisi la moyenne de la composante centripète sur l'ensemble des points comme mesure analytique de la performance d'une méthode de navigation.

2.1.6.4 La longueur du pas

Nous pensons que la longueur du pas est un critère moins important que les autres. En pratique, la longueur du vecteur trouvée par un modèle peut être ignorée et la longueur du déplacement de l'animat calculée par d'autres méthodes comme celle que nous verrons en section 5.4. Nous exposons toutefois brièvement comment nous procéderions pour étudier ce paramètre.

Le principe consisterait à observer s'il existe une relation linéaire entre la distance au but et la longueur du vecteur déplacement calculé par la méthode⁷. Pour cela, on peut représenter la longueur du vecteur en fonction de la distance au but. On peut aussi tracer le rapport longueur du déplacement divisé par la distance en fonction de la distance et observer si ce rapport tend vers une constante pour des petites distances, ou est très dispersé. Cela permet de déterminer si les méthodes peuvent estimer la distance restant à parcourir. Idéalement ce rapport tendrait vers 1.

2.1.6.5 L'orientation

Estimer la performance d'une méthode d'orientation semble simple : on peut étudier la répartition des écarts angulaires entre l'orientation réelle de l'animat et l'orientation estimée par la méthode.

2.1.6.6 Tout ensemble : mesure synthétique

Pour obtenir une mesure synthétique depuis chaque point de la grille définie d'où on fait partir l'animat, on le laisse parcourir une trajectoire complète jusqu'au but, ou jusqu'à ce qu'il se bloque contre un obstacle, ou encore après un nombre défini de cycles.

Dans ce cas, on fait fonctionner un modèle complètement défini : une méthode d'appariement, une méthode donnant la direction à suivre, éventuellement assistée d'une méthode d'évitement d'obstacles.

Les mesures synthétiques qu'on peut obtenir sont : le point d'arrivée (distance au but à l'arrivée notamment), longueur de la trajectoire, *tortuosité* (définie en section 5.4), nombre de pas effectués).

2.1.6.7 Conclusion

On dispose globalement de deux approches pour estimer la valeur d'une méthode de navigation. Les mesures analytiques fournissent des informations théoriques. Les mesures synthétiques fournissent des informations plus pragmatiques. En effet, une méthode peut avoir une efficacité théorique meilleure (aller souvent plus directement dans la direction du but) mais une efficacité pragmatique moins bonne (rester coincée dans certains cas, là où une autre méthode aurait trouvé un contournement). Il sera intéressant de conserver cet aspect présent à l'esprit lors de l'analyse des résultats statistiques.

⁷Le déplacement réellement effectué par l'animat peut dépendre d'autres facteurs comme la variation du gain liant ces deux variables, ou la présence d'un obstacle.

2.1.7 Généalogie des modèles

Afin de faciliter la compréhension, nous avons réalisé une vue synthétique de type généalogique représentant les relations entre les modèles que nous allons étudier dans cette thèse. Elle est présentée en annexe, figure A.1.

Nous pouvons maintenant aborder la revue des modèles proprement dite.

2.2 Les premiers modèles, Cartwright & Collett

Dans les années 80, deux chercheurs anglais de l'Université du Sussex, Cartwright & Collett, ont étudié la façon dont les abeilles retrouvent le chemin menant à un lieu intéressant préalablement rencontré [CAR 83, CAR 87].

En entraînant des abeilles à rejoindre une source de jus sucré dans une salle aux murs blancs où étaient disposés des cylindres noirs verticaux, les chercheurs ont observé où l'abeille cherche la source par la suite, dans les mêmes conditions, puis dans des conditions perturbantes (la position relative, la taille et l'orientation relative des objets environnants modifiés par rapport au modèle appris). Ils ont complété leurs observations par des comparaisons avec des modèles computationnels qu'ils ont développés pour l'occasion. Ces modèles sont habituellement collectivement désignés par « modèles à instantané »⁸, en référence au fait que la signature d'un lieu est un instantané des informations reçues par la rétine.

Bien que ces modèles aient été conçus à l'origine pour tester des hypothèses expliquant les techniques de navigation locale exploitées par les abeilles, ils ont servi de source d'inspiration pour créer des modèles utiles à la robotique. En effet, ils ont un rapport performance/coût remarquable dans la mesure où il leur suffit de quelques mesures directes de capteurs, sans carte fournie *a priori*, par contraste avec des méthodes d'ingénierie classique qui s'appuieraient plutôt sur une carte de l'environnement afin de calculer des positions en trois dimensions.

Dans la suite de cette thèse, nous désignerons collectivement les modèles de Cartwright et Collett par « modèles CC » pour les distinguer des autres modèles utilisant des instantanés. Lorsque nous désignerons un des 5 modèles en particulier, nous utiliserons les libellés CC1 à CC5.

2.2.1 Signature d'un lieu

Dans les premiers modèles, on schématise l'environnement comme un fond clair sur lequel des objets sombres se détachent, ce qui correspond en effet à la situation d'une abeille en extérieur, le fond du ciel étant presque toujours beaucoup plus lumineux que les objets⁹. Le système visuel de l'abeille est schématisé par une rétine circulaire sur laquelle se projettent les objets environnants, l'ensemble est représenté sur la figure 2.6. Mémoriser cette projection permet d'extraire des informations angulaires

⁸Nous traduisons par « modèles à instantané » l'expression « snapshot models ».

⁹Les photographes le savent bien aussi.

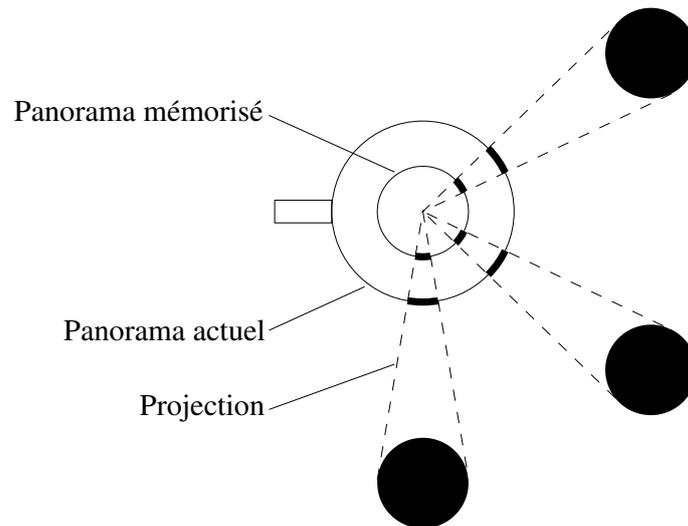


FIG. 2.6: Fondements de la famille de modèles CC : prise d'un instantané. Un animat (représenté par deux cercles concentriques) est équipé d'une rétine (le plus grand cercle) sur laquelle se projettent (arcs épais sur le cercle) les obstacles environnants (disques noirs). Il peut mémoriser un panorama visuel (le petit cercle intérieur). Le rectangle indique l'arrière de l'animat. Les seuls modèles efficaces de cette famille nécessitent que l'animat connaisse son orientation.

qui seront par la suite utilisées pour naviguer. Nous appellerons « secteurs » les parties noires et blanches du champ de vision, respectivement représentées par des arcs de cercles épais et fins sur les schémas.

2.2.2 Appariement et calcul de direction

2.2.2.1 Principe

Les modèles CC ont innové en ce qu'ils fournissent une indication de la direction à suivre pour revenir au but désigné par un panorama mémorisé, sans qu'aucune information de distance ne soit connue. Le principe est illustré figure 2.7.

2.2.2.2 Appariement

Les modèles CC fondent le calcul de la direction à suivre sur la mesure des variations d'angles observés entre deux points de vue. Pour pouvoir estimer comment la direction pointant vers un objet a varié d'une vue à l'autre, le choix est fait dans ces modèles d'apparier explicitement les objets entre les vues, c'est-à-dire d'estimer, pour chaque secteur dans le champ visuel d'une vue, à quel secteur il correspond dans l'autre vue.

Dans tous les modèles CC, l'appariement choisi est de type glouton [COR 98]. Cela signifie qu'il

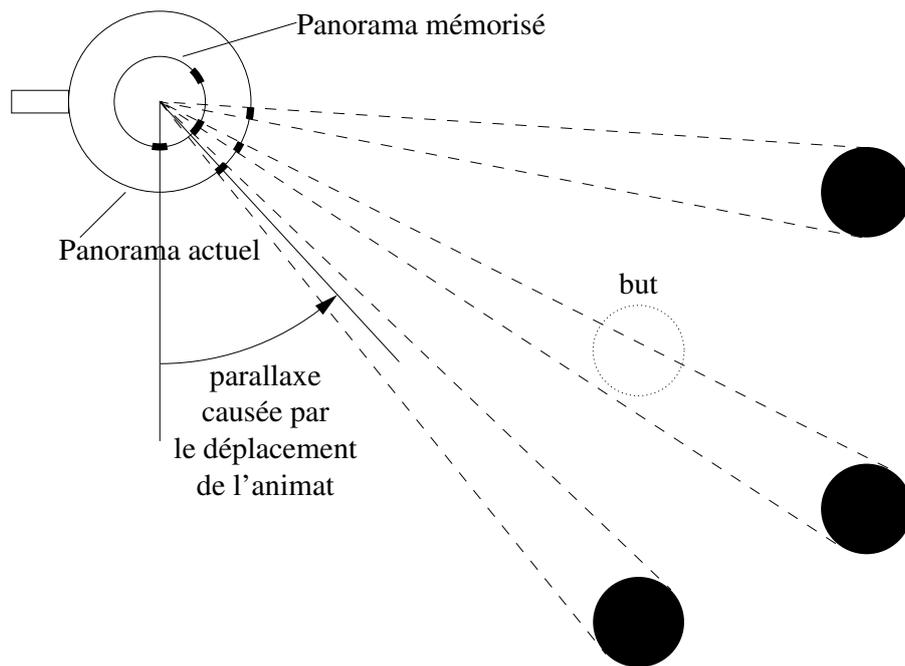


FIG. 2.7: Fondements de la famille des modèles CC : retour au nid. Le but, en pointillé, montre le lieu d'où a été mémorisé le panorama en figure 2.6. Lors du retour à proximité du lieu mémorisé, les obstacles se projettent sur la rétine à des positions différentes. En d'autres termes, le fait que l'animat n'est pas au but cause des parallaxes entre le panorama mémorisé et le panorama actuel. Un exemple de parallaxe est représenté : l'amer le plus au sud est précisément au sud du but, mais pas au sud de l'animat dans sa position actuelle. Les modèles CC permettent à l'animat de rejoindre le but en choisissant une direction qui tend à diminuer les parallaxes. Pour estimer ces parallaxes il faut appairer les éléments mémorisés avec ceux actuellement visibles. Quelques erreurs d'appariement sont tolérées par le modèle.

passé en revue les éléments d'un des panoramas et apparie chacun à l'élément de l'autre panorama qui lui est le plus proche, sans tenir compte des autres panoramas déjà effectués ou à venir, et sans remettre en question les appariements déjà effectués.

Le fait que l'appariement puisse être en partie faux fait partie du fonctionnement normal du modèle. La trajectoire peut en être perturbée mais, si elle aboutit tout de même au but, l'essentiel est réalisé.

2.2.2.3 Différents essais pour la direction à suivre

Les caractéristiques qui distinguent les modèles CC entre eux sont liées au choix des éléments qui sont extraits de la signature d'un lieu. En effet, les éléments retenus pour l'appariement peuvent être les contrastes à chaque transition entre secteurs successifs, ou bien les secteurs eux-mêmes.

Dans un premier cas, chaque contraste de la vue mémorisée est apparié au contraste (de même sens) le plus proche (angulairement) dans l'autre vue et on suppose que l'appariement est correct, c'est-à-dire que le même objet du monde est représenté par les contrastes appariés dans les deux vues. Une information exploitable est la différence entre la direction pointant vers cet objet dans la première vue, et la direction pointant vers cet objet dans la seconde vue (voir figure 2.8 à gauche). On considère alors pour chaque paire un vecteur unitaire représentant une direction à suivre pour réduire cet écart. Le principe est que si un objet était à gauche de mon champ de vision et qu'il est maintenant juste en face de moi, un pas chassé vers la droite (à 90° de la direction où je le vois actuellement) le ramènera vers la gauche de mon champ de vision. La somme de ces vecteurs unitaires est la direction suivie par l'animat. Dans la représentation du panorama en cercle, les flèches partant des projections des amers sont tangentes au cercle, d'où l'appellation de *composante tangentielle* donnée à cette méthode.

Dans un deuxième cas, on considère les secteurs eux-mêmes. On apparie chaque secteur de la vue mémorisée avec le secteur le plus proche (angulairement) dans l'autre vue. On reprend l'algorithme du premier cas, où l'on considèrera le milieu de chaque secteur lors des comparaisons d'angles, mais on dispose aussi de l'information de largeur apparente du secteur. Dans le même esprit, on ajoute d'autres vecteurs pour en tenir compte. Si un secteur apparaît plus large dans la vue actuelle que dans la vue mémorisée, on considère un vecteur unitaire qui fait s'éloigner l'animat de ce secteur et inversement. On note que chacun de ces vecteurs unitaires supplémentaires est radial au cercle représentant la vue, d'où l'appellation de *composante radiale* donnée à cette méthode.

Nous appellerons, dans ces modèles et dans les autres modèles similaires, « vecteurs individuels »¹⁰ les différents vecteurs intervenant dans la somme, et « vecteur résultant » la somme vectorielle.

Avant de détailler les modèles, une distinction doit être précisée. Les modèles CC1 et CC4 considèrent les contrastes entre secteurs et la composante tangentielle, alors que les modèles CC2, CC3 et CC5 considèrent les secteurs eux-mêmes et les composantes tangentielles et radiales. Les deux sont illustrés dans la figure 2.8. Les secteurs blancs comme les secteurs noirs sont appariés et pris en

¹⁰Nous traduisons de façon aussi satisfaisante que possible l'expression « contribution vector » par « vecteur individuel ».

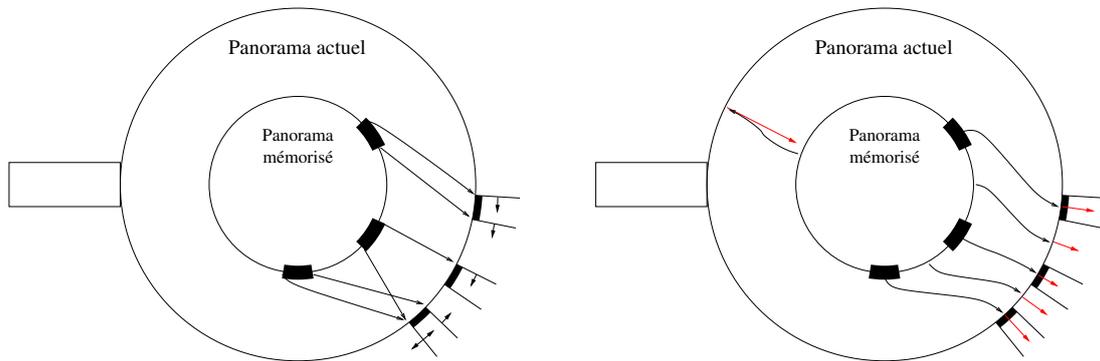


FIG. 2.8: Deux façons d'exploiter les parallaxes expérimentées par Cartwright et Collett. Tous les modèles CC utilisent comme direction à suivre une somme de vecteurs individuels. Ces vecteurs sont obtenus en comparant les azimuts d'objets appariés dans l'une et l'autre vue. À gauche, on apparie les transitions entre secteurs (points de contraste), ce qui fournit des vecteurs individuels tangents au cercle. À droite, on apparie les secteurs eux-mêmes, ce qui fournit des vecteurs individuels radiaux au cercle. Dans le schéma, on montre tous les secteurs (amers et espaces entre amers) et leurs contributions. Les modèles CC1 et CC4 n'exploitent que les vecteurs tangentiels, les modèles CC2, CC3 et CC5 exploitent les vecteurs tangentiels et radiaux. La flèche apparemment isolée en haut à gauche du schéma de droite correspond au vecteur individuel calculé d'après l'appariement entre le secteur sans amer qui occupe deux tiers de la vue mémorisée et presque 5/6 de la vue actuelle.

compte.

Pour finir la présentation des modèles CC, nous les résumons ici en précisant les différences essentielles.

- CC1** apparie les arêtes, somme les composantes tangentielles, se met face à la direction à suivre et avance. Échoue parce que l'animat tourne, ce qui perturbe l'appariement et le calcul des vecteurs individuels.
- CC2** apparie des secteurs, somme les composantes tangentielles et radiales, se met face à la direction à suivre et avance. Échoue pour les mêmes raisons que CC1.
- CC3** procède comme CC2 mais seule la composante radiale sert à calculer le déplacement, la composante tangentielle servant à faire tourner l'animat. Dans un environnement à 3 amers une surface de captation existe, mais les conditions perturbantes produisent des effets très différents de ceux observés chez les abeilles. Ce modèle échoue en environnement plus complexe. Les chercheurs ont conclu que les panoramas doivent être alignés par un moyen extérieur pour pouvoir être appariés correctement.
- CC4** comme CC1 mais les panoramas sont calés en orientation. Fonctionne bien mais les conditions perturbantes produisent des effets très différents de ceux observés chez les abeilles.
- CC5** comme CC2 mais les panoramas sont calés en orientation. Fonctionne bien et les conditions

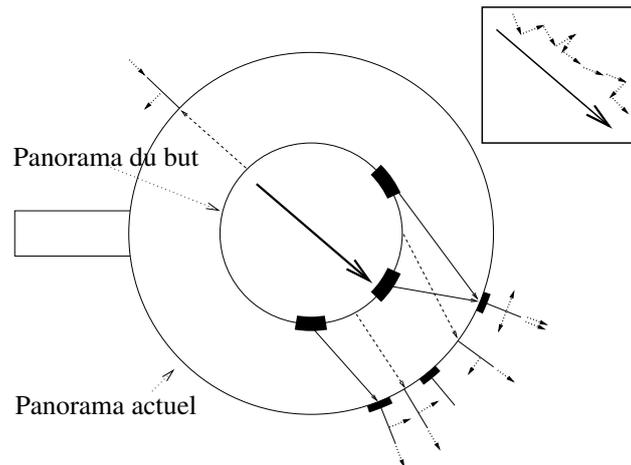


FIG. 2.9: Le modèle CC5 complet. Il utilise les composantes radiales et tangentiels. Le rectangle en haut à droite montre comment les vecteurs individuels (flèches en pointillés) contribuent au vecteur mouvement résultant (flèche en gras). L'animat conserve une orientation constante. Le schéma représente volontairement un cas où l'appariement est imparfait pour illustrer la tolérance du modèle.

perturbantes produisent des effets similaires à ceux observés chez les abeilles. (Cf. figure 2.9.)

2.2.3 Conséquences d'une désorientation

Nous nous focalisons sur les modèles CC4 et CC5 qui fonctionnent.

La méthode d'appariement (dans le cas des secteurs, comme dans le cas des transitions) est fondée sur la minimisation des parallaxes. Elle sera donc perturbée par une désorientation. Ceci a été remarqué par les auteurs, dans la mesure où ils ont étudié le cas où l'animat peut tourner mais n'est pas capable de compenser par rotation mentale (CC1 à CC3), ce qui a le même effet sur l'appariement. La méthode ne prévoit pas de constater une situation d'échec.

Pour le calcul de la direction à suivre, distinguons les deux composantes, et distinguons la direction et le sens des vecteurs individuels.

Considérons la direction des vecteurs individuels. Dans la composante tangentielle comme dans la composante radiale, la direction d'application de chaque vecteur est calculée à partir de l'azimut de l'amer concerné dans le panorama actuel, et ne dépend pas de l'azimut du même amer dans le panorama mémorisé. En cas de désorientation de l'animat, l'azimut actuel reste bien défini (mais l'azimut mémorisé ne l'est plus). L'utilisation des azimuts mémorisés introduirait donc une dépendance en orientation, qui est évitée par l'utilisation exclusive des azimuts actuels. Nous verrons plus loin (section 2.3.1) un modèle qui introduit cette dépendance.

Considérons maintenant le sens des vecteurs individuels, la composante tangentielle utilise les azimuts pour estimer le sens du vecteur individuel. Elle est donc perturbée par une désorientation, dès

que celle-ci est suffisante pour qu'au moins une des pseudo-parallaxes mesurée n'ait plus le même signe que la vraie parallaxe que l'animat mesure s'il n'est pas désorienté. Les parallaxes tendant vers 0 quand l'animat s'approche du but, on peut prédire (et nous l'illustrerons par des simulations en section 3.2.1) que le modèle sera faiblement perturbé loin du but, et plus fortement perturbé près du but.

La composante radiale, elle, n'utilise que les largeurs apparentes pour estimer le sens du vecteur individuel. Elle n'est donc pas perturbée par une désorientation.

Les modèles CC sont donc dépendants d'une bonne connaissance par l'animat de son orientation, à l'étape d'appariement et à l'étape de calcul de la direction à suivre.

2.2.4 Conclusion sur les modèles CC

Ces modèles simples valident expérimentalement le principe consistant à utiliser des variations d'angle pour naviguer. Toutefois, pour une utilisation sur un robot réel, un environnement adapté ou une modification de la signature et de la méthode d'appariement sont nécessaires. En outre, l'étape d'appariement et la composante tangentielle nécessitent que l'animat connaisse son orientation. La composante radiale semble un élément à conserver, nous y reviendrons au prochain chapitre.

2.3 Modèles modifiant le calcul de direction à suivre

Parmi les améliorations qu'on peut faire à la famille des modèles CC, il est possible de changer la façon dont la direction à suivre est calculée.

2.3.1 Méthodes inverse et moyenne, modèle de Hong

Hong et al. [HON 91] semblent avoir trouvé indépendamment une méthode similaire à la composante tangentielle de Cartwright & Collett.

À proprement parler, le modèle de Hong change plus que la direction à suivre, mais à la fois chronologiquement et dans l'ordre des idées, il se place naturellement ici.

2.3.1.1 Signature d'un lieu

Hong utilise comme capteur panoramique une sphère placée au-dessus d'une caméra noir et blanc. De l'image reçue par la caméra on extrait une signature unidimensionnelle à topologie circulaire de 360 valeurs en niveaux de gris en prenant à chaque azimut la moyenne de 5 pixels disposés radialement. Il semble que ce soit le premier exemple d'utilisation de panorama en niveaux de gris dans ce contexte.

Cette signature est segmentée en régions où la luminance varie de façon monotone. Puis, l'algorithme cherche les points de dérivée maximale dans l'image, avec un seuil sur la hauteur du changement pour limiter le nombre de points trouvés. Les points trouvés sont alors classés par ordre de saillance, où la saillance tient compte du contraste du point et de son isolement. Enfin, les 15 points les plus saillants sont conservés, on les appelle les points caractéristiques.

L'image initiale, ou tout au moins un anneau correspondant aux directions proches de l'horizontale, doit être mémorisée car l'étape suivante l'utilise pour une opération de corrélation.

2.3.1.2 Appariement

Hong essaye d'apparier chaque point caractéristique de l'image actuellement vue à tous les points de la signature de destination. Pour calculer la qualité de l'appariement de deux points, il a recours à une mesure de corrélation des voisinages (13 pixels de large, avec une fenêtre de pondération) de ces deux points dans les images 2D mémorisée et actuelle. Il commence par normaliser ces voisinages pour avoir une moyenne et un écart-type constants, afin de ne pas dépendre d'un éclairage constant¹¹.

L'appariement est finalement réalisé en deux temps. Un premier passage apparie les points saillants indépendamment les uns des autres et permet d'estimer une direction à suivre (cf. la section suivante 2.3.1.3). À cela est ajoutée la prise en compte d'une éventuelle petite déviation angulaire du robot. Puis, à l'aide de l'estimation de déviation et de direction à suivre trouvées, un second passage cherche à la fois des valeurs de déviation et de direction à suivre qui minimisent la somme des écarts de corrélations, en parcourant l'espace à deux dimensions des paramètres possibles.

2.3.1.3 Calcul de direction

Le calcul de direction est similaire à celui du modèle CC4 (composante tangentielle), mais il introduit pour les vecteurs individuels la proportionnalité à l'écart qui leur donne naissance. Cet aspect est détaillé plus loin en section 2.3.2.

Dans leur expérience sur robot réel, Hong et al.[HON 91] ont introduit une innovation. Elle découle de la remarque que l'application du modèle n'est pas antisymétrique, c'est-à-dire que le vecteur calculé avec M pour point actuel et N pour but n'est pas l'opposé du vecteur calculé avec N pour point actuel et M pour but.

Nous appellerons « méthode directe » la méthode qui consiste à choisir pour chaque amer la direction qui « tend à réduire le plus rapidement l'écart d'azimut de l'amer entre la vue mémorisée et la vue actuelle », pour reprendre l'idée de Hong. Il s'agit donc d'associer à chaque paire d'amers un mouvement orthogonal à la direction où l'amer est vu actuellement, ce qui est aussi le choix des méthodes CC.

¹¹La corrélation ne donne pas de bons résultats si la moyenne et l'écart type des signaux sont différents, il faut normaliser pour compenser les variations d'éclairage locales.

Nous appellerons « méthode inverse » la méthode qui consiste à échanger les deux vues, appliquer la « méthode directe » puis inverser le sens du vecteur obtenu.

Hong souhaite avoir une direction proche de la direction réelle, et décide de faire une moyenne entre le vecteur de la « méthode directe » et le vecteur obtenu par la « méthode inverse ». Nous appellerons ceci « méthode moyenne ».

2.3.1.4 Conséquence d'une désorientation

L'appariement est fondé sur un calcul de corrélation, donc a priori indépendant de l'orientation. Toutefois, les auteurs précisent qu'il n'y a pas de compensation des déformations sphériques, ce qui signifie que même sans désorientation, de grandes parallaxes risquent de perturber la corrélation. Dans l'expérience citée, la distance entre le robot et le but était de l'ordre d'un pied (environ 30 centimètres), ce qui aboutissait à des parallaxes n'excédant probablement pas une vingtaine de degrés. *A fortiori*, en cas de grande rotation ou de désorientation la méthode sera probablement perturbée.

Le calcul de direction à suivre s'appuie sur les parallaxes, et sera donc perturbé par une désorientation.

Par ailleurs, la méthode inverse introduit une nouvelle dépendance vis-à-vis de l'orientation, car elle fait intervenir les directions des amers dans le panorama mémorisé. La méthode moyenne introduit un mélange entre les deux, qui ne peut donc être bien défini lui non plus en cas de désorientation.

2.3.1.5 Conclusion

Le modèle de Hong introduit plusieurs éléments nouveaux : une signature en niveaux de gris avec sélection de points saillants, une méthode d'appariement de ces points saillants, une variante sur le calcul de la direction à suivre dont nous reparlerons, avec la possibilité de compenser une petite déviation angulaire du robot (moins de 3 degrés). Ces éléments sont intéressants.

Hong utilise donc une signature plus riche que les modèles CC. Nous pensons que ce point est important. Notre modèle utilise aussi une signature plus riche, mais différente : nous introduisons la couleur et une méthode de segmentation adéquate.

L'idée de la méthode inverse et moyenne apporte un point intéressant pour la compréhension des modèles. Toutefois, plus loin en section 2.5, nous verrons comment Weber et al. [WEB 99] ont mis en évidence que la méthode directe a une tendance naturelle à éviter les obstacles, tendance qui est perdue si on lui substitue la méthode moyenne. En outre, nous pensons que les méthodes inverses et moyennes sont inapplicables en cas de désorientation.

2.3.2 Modèle PV vecteur proportionnel

Les modèles CC ont une propriété indésirable pour une application robotique : la direction à suivre calculée est une fonction discontinue de la position de l'animat. Si l'animat passe d'un point

– où un amer est vu légèrement plus à gauche qu’au but – à un point proche – où le même amer est vu légèrement plus à droite qu’au but – une discontinuité apparaît, qui peut rendre artificiellement anguleuse la trajectoire d’un robot qui suivrait cet algorithme. En particulier, le vecteur obtenu ne tend pas vers zéro continûment quand l’animat approche du but, alors que ce serait une propriété désirable pour un expérience de robotique réelle. La chute de la norme du vecteur sous un certain seuil pourrait alors être utilisée comme signe de l’arrivée au but.

Le modèle PV¹² « vecteur proportionnel » [MöL 98a, LAM 00] introduit une modification qui produit une réponse plus « lisse ». Il considère des vecteurs individuels proportionnels aux écarts mesurés (parallaxes pour la composante tangentielle, variation de largeur apparente pour la composante radiale) au lieu de vecteurs unitaires. Nous avons trouvé cette modification intéressante et l’avons adoptée.

Nous pensons que cette modification n’introduit pas de changement fondamental vis-à-vis de la viabilité en cas de désorientation : le modèle PV appliqué à la composante tangentielle reste dépendant d’une bonne orientation, alors que le modèle PV appliqué à la composante radiale reste a priori utilisable.

2.3.3 Modèle DV vecteur différence

Le modèle DV¹³ « vecteur différence » [LAM 00] introduit une légère variante par rapport au modèle PV : au lieu de calculer un écart d’angle et de créer un vecteur unitaire dont la norme est cet écart (multiplié par une constante), on réalise simplement une différence entre un vecteur unitaire pointant vers la direction de l’amer vu du but, et un autre vecteur unitaire pointant vers la direction de l’amer vu actuellement (figure 2.10).

La direction calculée par le modèle DV est la même que celle du modèle PV avec la *méthode moyenne*. La norme n’est pas la même toutefois et il tend à tenir un peu plus compte des petits écarts que des très grands, car la norme d’un vecteur élémentaire est alors une fonction non linéaire de l’écart d’angle (un cosinus), et non plus une norme proportionnelle.

Conceptuellement cependant, il porte les prémisses du modèle ALV (décrit à la prochaine section 2.4), en introduisant le fait d’associer un vecteur unitaire à chaque direction conduisant à un obstacle.

Vis-à-vis de la désorientation, le modèle DV introduit la même dépendance que la méthode moyenne de Hong, ce qui le rend inapplicable en cas de désorientation. Nous n’avons pas directement repris d’élément de ce modèle.

¹²PV signifie « Proportional Vector ».

¹³DV signifie « Difference Vector ».

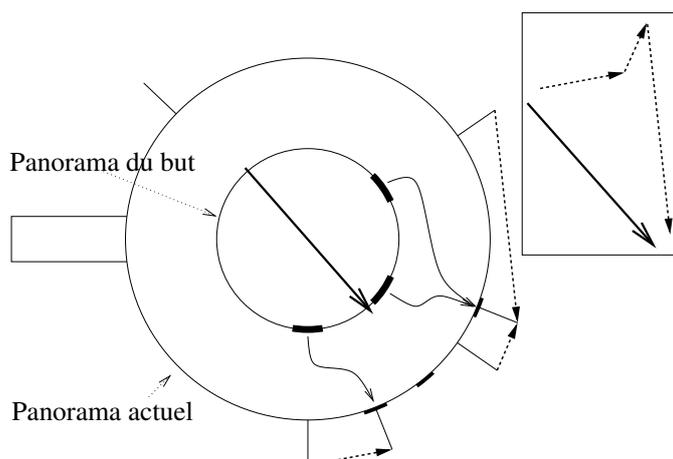


FIG. 2.10: Le modèle DV. Cet exemple représente volontairement un cas où l'appariement (flèches courbes) est inexact (l'amer mémorisé central est mal apparié). Pour chaque amer, le vecteur individuel est calculé comme la différence entre deux vecteurs unitaires pointant respectivement dans la direction de l'amer vu du but et dans la direction actuelle de l'amer. Les vecteurs unitaires ont pour origine le centre des cercles mais pour ne pas surcharger le dessin seule la portion qui dépasse du grand cercle est représentée, sous la forme d'un segment radial. Les flèches autour du grand cercle représentent les différences entre ces vecteurs unitaires. La flèche épaisse au centre représente la direction à suivre, qui est la somme des trois autres vecteurs comme l'indique l'encadré. La direction calculée est modérément perturbée par le mauvais appariement.

2.4 Simplifier tout, ALV ou « vecteur amer moyen »

L'ALV [MöL 99] ou « vecteur amer moyen »¹⁴ peut être considéré comme une simplification du modèle DV. Au lieu de segmenter, d'apparier, puis de faire des différences amer après amer, l'ALV fait un calcul similaire, mais factorisé différemment et sans appariement explicite.

2.4.1 Signature d'un lieu

L'ALV simplifie considérablement l'information à mémoriser comme signature d'un lieu. Dans [LAM 00], l'environnement permet une signature identique à celle des modèles CC, un panorama segmenté succession de secteurs noirs et de secteurs blanc. Mais l'animat ne mémorise pas le panorama complet.

À la place, l'animat va calculer puis mémoriser seulement un vecteur dont la signification apparaît lors du calcul de direction. Ce vecteur est la somme de vecteurs unitaires pointant chacun vers un amer du panorama, voir figure 2.11 à gauche, d'où l'appellation de « vecteur amer moyen ». Les auteurs rapportent avoir tenu compte soit des deux bords des amers (transitions entre secteurs), soit seulement d'un des bords de chaque amer (par exemple une transition noir→blanc).

¹⁴Nous traduisons par « vecteur amer moyen » l'expression « Average Landmark Vector ».

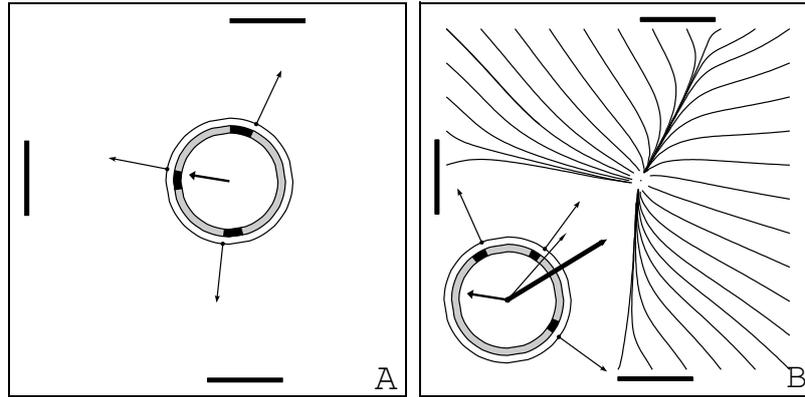


FIG. 2.11: Le modèle ALV. A : L'animat (cercle) est au but. L'anneau grisé représente le panorama vu. L'animat calcule la moyenne des vecteurs dirigés vers les amers (la transition noir-blanc de chaque amer, dans cet exemple), le résultat est le vecteur partant du centre. B : Depuis un autre point, le même calcul aboutit à un autre vecteur (flèche fine partant du centre). La différence entre les deux (flèche épaisse partant du centre) est la direction à suivre. Les courbes montrent des exemples de trajectoires suivies. (Figure reprise de [MöL 99].)

Formellement, si on énumère les amers par i , avec \vec{u}_i^M le vecteur unitaire pointant vers l'amer i :

$$\vec{\text{ALV}}^M = \sum_i \vec{u}_i^M$$

2.4.2 Appariement et calcul de direction

L'ALV, ne mémorisant qu'un vecteur en un lieu donné, ne fait pas d'appariement proprement dit.

Le calcul de la direction à suivre est très simple : si l'animat s'est déplacé de M vers X et souhaite revenir en M , il calcule la signature du point X courant et obtient un autre vecteur. La direction à suivre est celle indiquée par la différence entre le vecteur actuel et le vecteur au but M .

Pour comprendre pourquoi ce vecteur pointe vers le but, il est intéressant de le comparer avec le modèle DV [LAM 00]. Si l'on suppose que l'ensemble des amers énumérés dans la somme est le même dans les deux vues, alors on peut factoriser les termes :

$$\begin{aligned} \vec{\text{ALV}}^X - \vec{\text{ALV}}^M &= \sum_i \vec{u}_i^X - \sum_i \vec{u}_i^M \\ &= \sum_i (\vec{u}_i^X - \vec{u}_i^M) \end{aligned}$$

On reconnaît dans le dernier membre une somme faisant apparaître des différences de vecteurs unitaires pointant vers les amers, c'est-à-dire exactement ce que fait le modèle DV.

Ainsi, sous ces hypothèses, l'ALV calcule sans faire d'appariement explicite le même vecteur que calculerait un modèle DV muni d'un appariement parfait.

L'ALV a été testé en simulation et sur un robot réel en composants analogiques [LAM 98]. Il a aussi donné lieu à diverses études. Ainsi, un réseau de neurones muni de la règle hebbienne d'apprentissage pour trouver la direction à suivre apprend « naturellement » l'ALV [HAF 01, HAF 02].

2.4.3 Conséquences d'une désorientation

On peut d'emblée signaler que la concordance de la direction choisie par l'animat avec celle du modèle DV montre que l'ALV est perturbé par une désorientation. Une analyse directe montre que cela revient à faire la différence de deux vecteurs dont l'un a été tourné d'une grandeur inconnue, en espérant que la direction de cette différence pointe vers le but malgré la rotation.

Il n'est pas surprenant que le fait de ne mémoriser qu'un vecteur au lieu d'un panorama complet rende cette méthode plus fragile. En effet, en cas de modification de l'environnement ou de mauvaise orientation, un modèle qui mémorise un panorama segmenté dispose d'information lui permettant potentiellement de constater la désorientation. Il peut aussi remettre en question l'utilisation d'une partie des amers. Avec l'ALV tous les amers sont pris en compte.

2.4.4 Conclusion

L'ALV est un peu à part dans cet ensemble. Sa simplicité le rend apparemment séduisant. Toutefois, nous remarquons que, s'il a l'avantage d'implicitement utiliser l'appariement correct lorsque l'ensemble des amers est le même dans les deux vues, l'apparition ou la disparition d'un ou plusieurs amers dans l'une des vues perturbe le vecteur et donc la comparaison.

La pauvreté de la signature mémorisée fait que ce modèle n'a vraiment aucun moyen de maintenir un fonctionnement correct en cas de modification de l'environnement ou de perturbation de l'orientation, ni même de constater une telle perturbation. L'ALV ne peut donc probablement être utilisé seul que dans des cas d'école.

L'idée de faire une combinaison linéaire de vecteurs, quoique abandonnée depuis pour une approche plus simple, a été à l'origine une source d'inspiration pour le compas visuel, comme nous le verrons dans le chapitre consacré aux comparaisons entre nos travaux et les travaux précédents (section 10.1.3).

2.5 Méthodes d'appariement différentes, Weber et al.

Weber et al.[WEB 99] ont étudié les performances relatives et les coûts computationnels de quelques variations sur la méthode d'appariement utilisée par les modèles CC, en simulation et avec quelques expériences robotiques.

Dans une première série de simulations, les amers sont modélisés comme des points mathématiques infiniment fins sans occlusions, et les algorithmes sont testés avec ou sans limitation de portée

de la vision (vision limitée à une distance constante, la surface couverte par la vue est donc un disque centré sur l'animat). Puis d'autres simulations font intervenir une modélisation plus fine avec prise en compte des occlusions.

Les résultats sont analysés à l'aide de statistiques analysant les courbes de performance (pourcentage de trajets réussis) en fonction du nombre d'amers.

L'appariement parfait n'est pas envisagé. La variante PV est utilisée, composante tangentielle uniquement. Weber et al. considèrent comme référence pour leurs comparaisons la méthode d'appariement qui consiste en une recherche exhaustive de minimisation des parallaxes $\sum_{i=1}^n |\theta_i - \beta_{p(i)}|^2$, et envisagent 6 autres variantes dont certaines en $O(n)$ et $O(n^2)$. Trois des variantes sont testées en outre avec des vecteurs individuels unitaires.

La modélisation par points mathématiques infiniment fins sans occlusions permet de satisfaire l'hypothèse que tous les panoramas ont le même nombre d'amers, hypothèse nécessaire pour plusieurs des méthodes testées. Les auteurs obtiennent un bassin de captation qui peut couvrir tout l'environnement, notamment avec l'appariement exhaustif et avec une des méthodes en $O(n^2)$.

Par ailleurs, les méthodes directe, inverse et moyenne sont analysées et discutées. Des expériences complémentaires montrent que la méthode directe réalise naturellement un évitement d'obstacle, alors que la méthode moyenne l'empêche, et que la méthode inverse a parfois tendance à envoyer l'animat sur des obstacles (que la modélisation lui permet de traverser sans dommage).

2.5.1 Conséquences d'une désorientation

La situation est similaire aux modèles CC. Tous les appariements envisagés sont fondés sur la minimisation de parallaxes, critère exploitable uniquement si les panoramas sont correctement orientés.

La méthode directe, recommandée par les auteurs, n'introduit pas la dépendance vis-à-vis de l'orientation qu'introduisent les méthodes inverses et moyennes (cf. 2.3.1.4). Toutefois, il s'agit dans tous les cas de la composante tangentielle qui en reste dépendante.

2.5.2 Conclusion

L'approche de Weber et al. consiste à approfondir certaines idées sur la façon d'améliorer l'appariement sans sortir du contexte original des modèles CC, par des méthodes approximatives mais rapides à calculer. Ces méthodes d'appariement approximatives ont permis une navigation effective en simulation dans un monde simple, plus quelques exemples sur robot réel.

Il s'agit toujours de panoramas alternant secteurs noir et secteurs blanc, ce qui est trop pauvre si par la suite on a besoin de distinguer les amers les uns des autres. Les méthodes supposent que les panoramas présentent le même nombre de secteurs, ce qui ne sera pas le cas dans d'autres contextes. D'après leurs résultats, le critère utilisé (le meilleur appariement est celui qui minimise la somme des parallaxes) permet la navigation à orientation connue. Nous pensons que les améliorations proposées

sont difficilement adaptables à d'autres cas.

Nous pensons que pour rendre possible une navigation en environnement réel peu ou pas préparé, ou avec une orientation arbitraire, et *a fortiori* avec ces deux contraintes simultanées, il faut enrichir la perception de l'animat et se donner d'autres critères de qualité des appariements. C'est ce que nous avons fait, à l'aide d'une méthode d'appariement elle aussi en $O(n^2)$ mais applicable à deux panoramas de nombre d'amers et d'orientation quelconques.

L'analyse qui aboutit à recommander la méthode directe, et exclure les méthodes inverses et moyenne a achevé de nous convaincre, même si la méthode moyenne semblait initialement plus judicieuse.

2.6 Autres approches

2.6.1 Association lieu/direction par apprentissage, Gaussier et al.

Gaussier et l'équipe de l'ETIS, à l'ENSEA (Cergy), travaillent depuis plusieurs années à l'étude et l'expérimentation sur robot réel (robot de type Koala) de systèmes de contrôle pour animats, incluant la problématique de navigation [BAB 00].

Leurs réalisations incluent un sous-système d'extraction d'amers visuels, un sous-système réalisant la navigation locale, et un sous-système chargé de faire le lien entre zones locales pour la navigation globale.

Leur direction de recherche est toutefois assez différente de celle de cette thèse, comme nous le verrons ci-dessous. L'équipe ETIS se donne pour contrainte de n'utiliser que les mécanismes « simples » à implémenter dans un réseau de neurones [GAU 00b]. À cette fin, un simulateur générique de réseaux de neurones a été élaboré et utilisé exclusivement.

2.6.1.1 Signature d'un lieu

Une caméra classique, montée sur pivot permet d'obtenir une image presque panoramique (250 degrés) en niveaux de gris. Les colonnes de pixels sont moyennées pour obtenir une signature à une dimension. Une quinzaine de minima et maxima locaux sont sélectionnés comme points d'intérêt. Chacun est mémorisé sous la forme d'une imagerie en niveaux de gris de 32x32 pixels, calculé par anamorphose (moyenne sur y, échelle logarithmique en x) à partir d'une colonne de champ visuel couvrant 7.5° horizontalement.

2.6.1.2 Appariement

Pour appairer deux panoramas, chaque imagerie du premier est comparée avec toutes les images du second. Les meilleures correspondances sont conservées et permettent de calculer des parallaxes.

La somme des valeurs absolues des parallaxes donne une mesure de distance entre les deux panoramas (il ne s'agit donc pas d'une distance métrique, seulement d'une mesure de ressemblance).

2.6.1.3 Calcul de direction

C'est ici que la méthode de Gaussier diffère complètement des modèles étudiés ici.

Des panoramas ont été mémorisés en quittant le but, pendant une phase dite « turn back and look ». L'animat quittant le but a mémorisé un ensemble de panoramas pris depuis des points autour du but, avec le regard dirigé vers ce dernier. Au cours de cette étape, il a associé chaque panorama avec l'ordre moteur rapprochant l'animat du but.

Lors du retour au nid, la procédure d'appariement est en fait utilisée pour déterminer le panorama mémorisé qui ressemble le plus au panorama actuel. Formellement, c'est celui dont la mesure de distance définie plus haut est la plus petite. Les parallaxes ne sont pas utilisées pour autre chose que pour cette mesure.

Une fois le panorama mémorisé identifié, l'animat exécute l'ordre moteur qui a été associé. Comme dans les autres modèles, l'animat exécute plusieurs pas, jusqu'à atteindre le but.

2.6.1.4 Commentaires

L'équipe ETIS a considéré le modèle de Cartwright & Collett comme solution pour guider son robot, sans le trouver satisfaisant toutefois. L'équipe a donc opté pour une solution différente, qui consiste à associer (par apprentissage) à quelques vues prises autour du but, dans lesquelles le robot fait face au but, la direction à prendre pour le rejoindre.

Ainsi, au lieu de descendre un gradient calculé à partir de parallaxes (niveau 1, guidage local, dans la hiérarchie de Trullier [TRU 97]), l'animat apprend à son réseau de neurones à associer un panorama à une direction (niveau 2 association lieu-action). Au niveau de l'implémentation neuronale, cette variante s'appuie sur des mécanismes plus simples et connus d'apprentissage par association. Nous reviendrons sur ce point particulier lorsque nous comparerons notre modèle à celui de Gaussier et al.

Les apports majeurs des travaux de l'équipe ETIS sont plus larges que la simple navigation par mesure d'angles puisqu'ils concernent notamment l'apprentissage et la sélection de l'action.

2.6.1.5 Conclusion

Les travaux de l'équipe ETIS sont intéressants, notamment pour le lien qu'ils font avec des modèles biologiquement plausibles d'apprentissage par association. Toutefois, cela signifie aussi que les mécanismes utilisés par leurs modèles pour déterminer le chemin à suivre sont fondamentalement différents des autres modèles étudiés dans cette thèse. Il s'agit de navigation par panoramas mais pas par mesure d'angles.

2.6.2 Rendre le champ de vecteurs plus conservatif, Bianco et al.

Le travail de Bianco et al. [BIA 00] est, lui aussi, un peu à l'écart par rapport aux autres travaux étudiés ici. Tout d'abord, cette équipe travaille avec l'image directement obtenue par une caméra non panoramique, donc avec un champ de vision très limité, ce qui rapproche ce travail des techniques d'asservissement visuel. Les principes de la composante radiale et de la composante tangentielle sont conservés et appliqués directement, en remplaçant chaque mesure d'un angle par la mesure directe d'une abscisse dans l'image. L'animat détermine donc s'il doit avancer en fonction de la largeur apparente des objets, et s'il doit se décaler sur le côté en fonction de la position des mêmes objets dans le champ de vision.

L'originalité de cette approche se situe dans une méthode particulière de sélection d'amers. Pour la comprendre, il faut se remémorer quelques propriétés des champs de vecteurs. L'ensemble des vecteurs que peut donner le modèle CC (ou une variante) en fonction du point où l'animat se trouve est un champ de vecteurs. Idéalement, chaque vecteur pointe exactement en direction du but, c'est-à-dire qu'il est exactement radial par rapport au but considéré comme centre.

Une caractéristique que Bianco souhaite donner au champ de vecteurs est la conservativité¹⁵. En termes concrets, cette propriété garantit que, si l'animat suit les directions calculées, il arrivera bien quelque part au lieu de tourner en rond.

Dans leurs expériences en intérieur, le robot quitte le but en faisant une danse en zig-zag (cf. schéma 2.12) pour prendre plusieurs vues de plusieurs points de vue de positions connues. Ensuite, un algorithme calcule la composante rotationnelle du champ de vecteurs déduit des mesures, en fonction des amers intervenant dans le calcul (l'existence d'une composante rotationnelle non nulle est directement liée à la non-conservativité). Il est alors en mesure de choisir les amers donnant le champ de vecteurs le plus conservatif.

Il est intéressant de remarquer que l'approche de Bianco et al. se distingue des autres parce qu'elle utilise l'information en provenance de plusieurs points de vues pour extraire des propriétés de plus haut niveau, en l'occurrence la composante rotationnelle du champ de vecteur, suivant les amers utilisés. C'est un point commun avec notre compas visuel qui, lui aussi, s'appuie sur plusieurs vues.

2.7 Tableau synthétique

Le tableau 2.1 résume de façon synthétique les modèles étudiés jusqu'ici.

¹⁵La circulation d'un champ de vecteurs \vec{u} le long d'un chemin est définie par $\int_{A \rightarrow B} \vec{u} \cdot d\vec{l}$. La conservativité signifie que la circulation du champ de vecteurs le long d'un chemin fermé est nulle, ou encore que la circulation d'un point A à un point B ne dépend pas du chemin suivi. C'est aussi équivalent à l'existence d'une fonction potentiel V qu'on peut associer à tout point de la région où le champ de vecteurs est défini, telle que la circulation d'un point A à un point B est $V_A - V_B$. Si le champ de vecteurs est conservatif, il est assuré que l'animat qui suit le gradient monte (ou descend) une fonction potentiel jusqu'à arriver à un extremum.

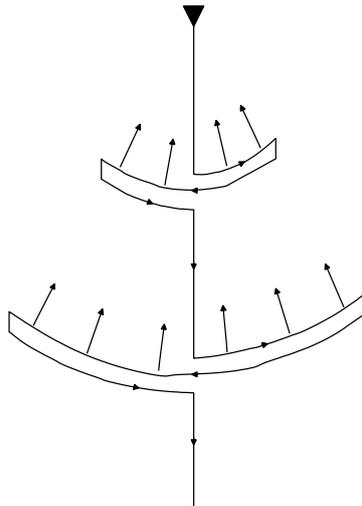


FIG. 2.12: La trajectoire du robot de Bianco et al. lorsque de robot réalise la phase dite « turn back and look ». Cette procédure utilise des vues prises depuis plusieurs endroits (les flèches) pour estimer les amers de l'environnement les plus fiables pour une tâche de navigation. De bons amers aboutissent à des ordre moteurs n'ayant pas de composante rotationnelle. (Figure extraite de [BIA 00].)

Modèle	Signature	Appariement	Direction
CC4	segmentation noir et blanc	glouton local	somme de vecteurs tangentiels unitaires appliqués sur chaque amer vu
CC5	segmentation noir et blanc	glouton local	somme de vecteurs tangentiels et radiaux unitaires appliqués sur chaque amer vu
Hong	segmentation niveaux de gris + image 2D	corrélation, 2 passes	somme de vecteurs tangentiels proportionnels appliqués au milieu de chaque paire (amer mémorisé, amer vu)
PV	segmentation noir et blanc	glouton local	somme de vecteurs tangentiels proportionnels aux écarts appliqués sur chaque amer vu
DV	segmentation noir et blanc	glouton local	somme de différences de vecteurs unitaires joignant position mémorisée et position vue
ALV	segmentation noir et blanc	aucun	comparaison globale de moyennes de vecteurs unitaires
Weber et al.	segmentation noir et blanc	divers	somme de vecteurs tangentiels proportionnels aux écarts appliqués sur chaque amer

TAB. 2.1: Comparaison synthétique des méthodes étudiées jusqu'à ce point de la thèse. Les points que nous conserverons dans la suite de nos travaux sont l'utilisation d'un panorama segmenté, le calcul de la direction à suivre par le modèle PV, vecteurs tangentiels ou radiaux, et la méthode directe (vecteurs appliqués sur chaque amer vu).

2.8 Conclusion de la première revue

Nous avons passé en revue, au travers d'une grille d'analyse commune, un ensemble de modèles de navigation locale par mesure d'angles. Nous avons discuté et comparé leurs caractéristiques et leurs apports respectifs.

Tous ces modèles ont été conçus en supposant qu'une référence extérieure fournit l'orientation de l'animat et font effectivement intervenir au moins une étape qui dépend de l'orientation de l'animat. Rappelons-en les trois raisons.

Tout d'abord, nous avons vu que les appariements étudiés dans le chapitre 2 (CC, Weber) sont perturbés par la désorientation d'une façon telle qu'ils ne sont plus viables.

Ensuite, la plupart des modèles s'appuient de plus sur la composante tangentielle qui introduit une seconde dépendance, alors que la composante radiale n'en introduit pas.

Enfin, les méthodes « inverses » et « moyennes » ajoutent un niveau supplémentaire de dépendance en orientation, seule la « méthode directe » n'en ajoute pas.

À ce stade on peut esquisser un modèle qui serait indépendant de l'orientation. Il devrait être capable d'apparier correctement deux panorama d'orientation quelconque. Il ne ferait intervenir que la composante radiale (exploitation de la variation des largeurs apparentes au cours du déplacement) avec la méthode directe (application des vecteurs individuels sur la position actuellement vue).

Au chapitre suivant nous allons illustrer et analyser plus précisément pourquoi ces modèles ne fonctionnent pas si l'animat ne connaît pas assez précisément son orientation et pourquoi le modèle esquissé ci-dessus est, certes, indépendant de l'orientation mais n'est pas encore satisfaisant.

Chapitre 3

Analyse approfondie et généralisation

Les modèles étudiés jusqu'ici ont exploré quelques variantes possibles de la navigation par panorama, en faisant l'hypothèse que l'orientation de l'animat est exactement connue. Nous avons, au cours de l'étude de ces modèles, indiqué quelles étapes introduisaient des dépendances vis-à-vis de l'orientation.

Nous nous proposons dans ce chapitre de clarifier, d'illustrer et d'analyser les forces et les faiblesses des mécanismes communs à ces modèles. Cet approfondissement se fait en plusieurs étapes.

Il nous paraît nécessaire de préciser au préalable comment interpréter la mesure de performance que nous avons choisie, la composante centripète, et jusqu'à quel point la nécessité d'éviter des obstacles justifie une moindre performance sur ce critère. C'est l'objet de la section 3.1.

Une fois l'interprétation de la composante centripète acquise, nous commencerons l'analyse proprement dite en section 3.2, en précisant et illustrant à quel point ces modèles sont perturbés par une erreur dans l'orientation de l'animat. En effet, alors que tous ces modèles calculent une direction à suivre approximative et parviennent précisément au but si la direction de l'animat est exactement connue, nous devons justifier pourquoi la perte de cette dernière hypothèse est rédhibitoire. Nous illustrerons en simulation les conséquences de la dépendance de la composante tangentielle vis-à-vis de l'orientation. Une comparaison avec la composante radiale montrera expérimentalement que cette dernière est bien indépendante de l'orientation.

Cette comparaison semble plaider en faveur de la composante radiale, ce que nos expériences en général ont contredit. Nous devons alors pousser plus loin dans le détail l'analyse des modèles CC, pour comprendre les raisons de cette contradiction apparente.

La définition de « composante tangentielle » et « composante radiale » dans les modèles CC recouvre en réalité deux différences dont une seule est fondamentale. En décorrélant ces différences en section 3.3, nous ferons apparaître quelques combinaisons qui n'ont pas été testées à notre connaissance et nous donnerons une nouvelle définition plus fondamentale des deux composantes.

À ce point de l'exposé, nous serons en mesure de nous focaliser véritablement sur les différences fondamentales entre les deux composantes. Nous démontrerons en 3.4 que la composante radiale n'a

pas la propriété de toujours rapprocher l'animat du but, contrairement à la composante tangentielle.

Enfin, muni de ces nouvelles informations, nous présenterons en section 3.5 quelques expériences qui illustrent visuellement les conséquences pratiques de ces différences.

Notre conclusion est que l'utilisation de la composante tangentielle est préférable à celle de la composante radiale, à condition que l'animat puisse estimer son orientation. Ceci motive la recherche de solutions permettant de retrouver l'orientation de l'animat, ce qui sera l'objet du chapitre 4.

3.1 Composante centripète et environnements testés

Nous avons annoncé en section 2.1.6.3 que nous utiliserons la composante centripète comme mesure de la performance d'un modèle de navigation. Nous savons la signification des valeurs -1, 0 et 1 de cette mesure, mais nous ne savons pas comment la nécessité de contourner des obstacles peut interférer avec le score d'une méthode.

Si la direction allant droit vers le but ne rencontre aucun obstacle, l'idéal est de la prendre, ce qui équivaut à une composante centripète de 1. Autrement dit, plus un modèle a une composante centripète proche de 1, meilleur il est. Cette analyse est en particulier valable lorsque l'animat est plus près du but que tout obstacle.

En revanche, si l'animat va droit au but et qu'un obstacle intersecte le volume que l'animat parcourt en allant au but en ligne droite, une collision s'ensuit. Une composante centripète égale à 1 en présence d'au moins un obstacle n'est donc pas optimale. Ce cas est représenté en figure 3.1, à gauche. Un bon choix est donc de contourner l'obstacle. La composante centripète peut rester positive.

Un obstacle concave peut nécessiter de s'éloigner momentanément du but, comme illustré au milieu de la figure 3.1. Mais la convexité de l'obstacle ne suffit pas à éviter cette propriété. Comme illustré à droite de la figure 3.1, un obstacle convexe près du but peut aussi nécessiter de s'éloigner momentanément de ce dernier avant de pouvoir le rejoindre.

Notons que la plupart des simulations effectuées dans les travaux que nous avons étudiés [CAR 83, WEB 99, HAF 02, MöL 98a, LAM 00] considèrent des amers circulaires. On peut montrer (mais nous ne l'avons pas fait) que dans ce cas, et à condition que l'espace entre deux amers soit supérieur au diamètre de l'animat, il existe toujours un chemin tel que la distance de l'animat au but décroisse de façon strictement monotone, présentant donc une composante centripète toujours strictement positive.

Ces propriétés de la composante centripète montrent qu'une valeur proche de 1 partout n'est ni nécessaire ni suffisante pour rejoindre le but. Malgré cela, l'observation en nuage de points en fonction de la distance donne une bonne indication, au moins dans certains cas.

Pour la suite de notre exposé, nous retiendrons donc, à propos de la composante centripète c sur une surface de test S , qu'un bon modèle de navigation présente les propriétés suivantes :

- c doit être aussi proche de 1 que possible si S ne contient pas d'obstacle ;
- c n'a pas besoin de devenir négative pour contourner des obstacles circulaires, si l'animat peut

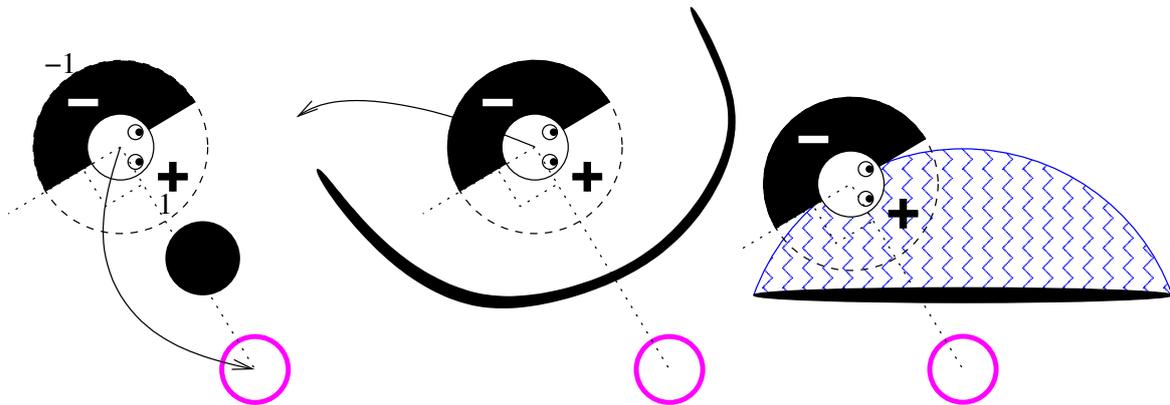


FIG. 3.1: Illustration de la mesure de composante centripète dans la poursuite d'un but. Le cercle représente le but. Autour de l'animat on représente par + et - les direction qui l'approchent et qui l'éloignent du but, respectivement. À gauche : si un obstacle est sur le chemin, l'animat ne peut plus aller droit au but. Une composante centripète de 1 en tout point n'est donc pas une garantie de rejoindre le but s'il y a des obstacles. Un obstacle concave (au milieu) peut obliger l'animat à s'éloigner du but (composante centripète négative) sur une partie de sa trajectoire. Mais un obstacle strictement convexe près du but (schéma de droite, une ellipse) le peut aussi. Si le centre de l'animat entre dans la zone en zig-zag, il ne peut en sortir qu'en s'éloignant momentanément du but. Même dans ce cas, une composante centripète positive n'est donc pas une garantie de rejoindre le but.

se glisser entre deux obstacles (ce sera notre cas).

Une surface S d'intérêt particulier pour notre analyse est le plus grand disque centré sur le but que l'animat peut parcourir sans heurter d'obstacle. Alors, dans la représentation de la composante centripète sous la forme d'un nuage de points en fonction de la distance au but, l'abscisse correspondant au rayon du disque délimite le diagramme en deux portions : à gauche, où le modèle doit avoir une composante centripète aussi grande que possible, et à droite, où le modèle peut avoir une composante centripète plus petite. Dans nos expériences, il n'est jamais utile qu'elle descende sous zéro.

3.2 Composantes tangentielles et radiale, et désorientation

Nous avons expliqué qualitativement au cours de la revue précédente quelles étapes des modèles cités jusqu'ici introduisent une dépendance dans l'orientation. Nous faisons abstraction momentanément de l'étape d'appariement, pour nous focaliser dans cette section sur l'étape de calcul de la direction à suivre. En effet, la simulation permet de placer ces modèles dans des conditions idéales d'appariement parfait.

Nous présentons deux séries de simulations que nous avons réalisées pour illustrer l'importance de la perturbation causée par une erreur de l'animat sur son orientation, dans le cas où il utilise seulement la composante tangentielle, ou bien seulement la composante radiale.

Elles ont été réalisées à l'aide de notre simulateur que nous présenterons plus en détail en section 8.1. La seule information notable pour le moment est que l'animat simulé dispose d'une vision lui permettant des mesures d'angles arrondis à un degré.

L'environnement comporte 10 amers dont les tailles et positions ont été choisies aléatoirement. Un tel environnement est similaire à ceux utilisés dans les expériences apparentées [CAR 83, WEB 99, HAF 02, MöL 98a, LAM 00], dans la mesure où les amers sont répartis sans structure particulière et avec en général un espace assez large pour contourner chaque amer. Ce choix n'est pas surprenant quand on modélise une abeille dans un environnement naturel.

Pour que le test soit représentatif, nous avons éliminé les autres sources de dépendance en orientation, en utilisant l'appariement parfait et la méthode directe. Des expériences préliminaires ont montré que la méthode moyenne ou inverse ne fait qu'empirer la situation, excluant le cas où une interférence inattendue entre ces deux dépendances vis-à-vis de l'orientation aurait abouti à une performance meilleure qu'avec une seule.

3.2.1 Conséquence d'une désorientation sur la composante tangentielle

La composante tangentielle est explicite dans les modèles CC4, PV, DV, Hong, Weber et al, et moins apparente mais présente tout de même dans l'ALV. Nous avons donc étudié en simulation la conséquence d'une désorientation dans un cas concret d'utilisation de la composante tangentielle.

L'ensemble est illustrée par la figure 3.2. Dans le cas d'une faible déviation, on constate que l'animat est attiré, non par le point but, mais par un point voisin. Le modèle échoue encore plus nettement si la direction est très fautive (écart de plus de 60 degrés par exemple) : l'animat vient en général heurter un des obstacles.

Expliquons ces résultats. L'erreur d'orientation de l'animat ajoute une constante à chaque mesure de parallaxe, ce qui perturbe tous les vecteurs tangentiels en conséquence. Les écarts pourraient se compenser si la répartition des amers autour de l'animat était uniforme, mais ce n'est pas le cas en général. Le point but n'a donc plus la propriété (énoncée en section 2.1.4) nécessaire pour caractériser un attracteur : le mouvement commandé en ce point doit être nul. En revanche, si l'erreur angulaire n'est pas trop grande, l'attracteur a dérivé vers un point voisin : c'est le point tel que les vecteurs tangentiels se compensent. Il dépend donc étroitement de la disposition des amers autour de l'animat.

La représentation des mêmes données sous la forme de nuage de points de la composante centripète est donnée en figure 3.3. Cette représentation permet de visualiser la dégradation des performances qui commence près du but dès 10 degrés de perturbation, et devient très claire à partir de 30 degrés.

Il a été prédit lors de l'analyse des modèles CC (section 2.2.3) que la désorientation ne les perturbe que lorsqu'elle est supérieure à la plus petite parallaxe, car c'est alors qu'elle commence à changer le sens des vecteurs individuels. À ce stade il n'est pas prouvé que cette analyse s'applique à la variante PV ici simulée, mais ce sera fait un peu plus loin (section 3.4.2).

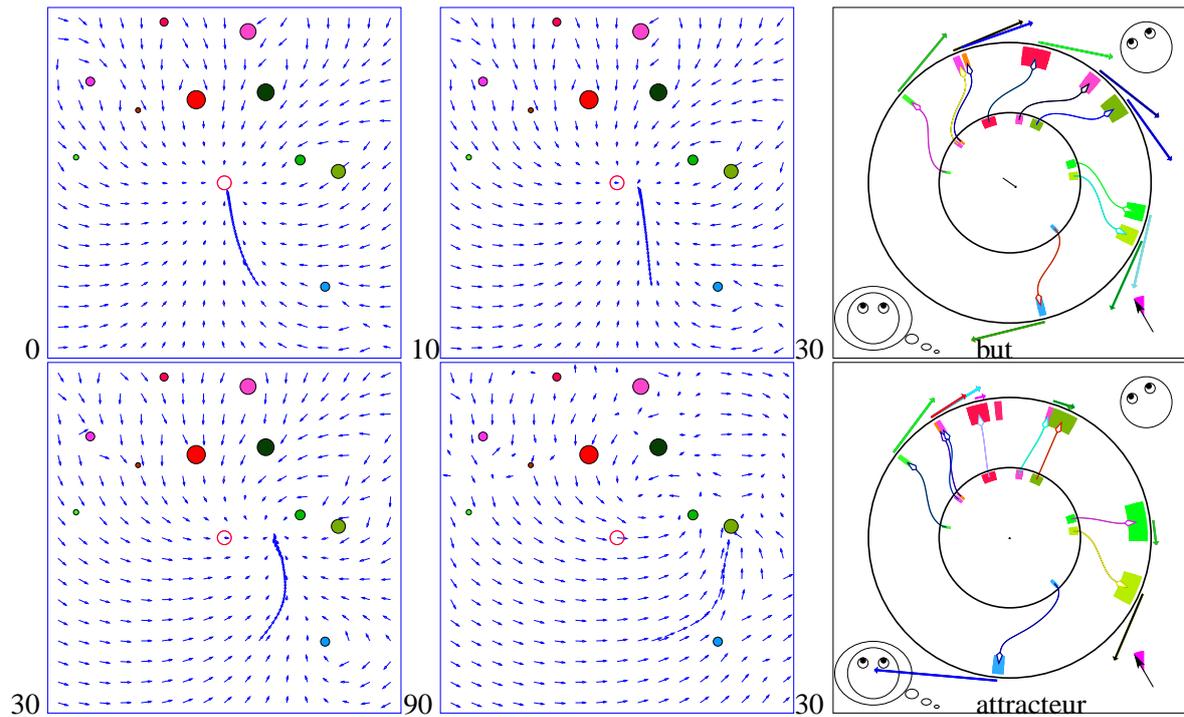


FIG. 3.2: Illustration de la sensibilité de la composante tangentielle à l'orientation. (On utilise l'appariement parfait et la méthode directe pour ne pas introduire d'autre dépendance en orientation). Chaque schéma représente le champ de vecteurs et une trajectoire obtenue avec une perturbation d'angle donnée, indiquée en bas à gauche. On constate que, même pour une erreur faible, le point attracteur dérive et l'animat y parvient en terminant son trajet en spirale. À droite est représentée, pour une perturbation de 30° , la décision que l'animat prend en deux points particuliers. En haut, l'animat a été placé au but. Des pseudo-parallaxes toutes identiques sont caractéristiques d'une position au but avec erreur d'orientation, mais le modèle ne sait pas l'exploiter : la somme vectorielle chasse l'animat vers l'est. En bas, l'animat est au point attracteur, comme le confirme le fait que les pseudo-parallaxes observées ont une somme nulle. Mais elles ne sont pas nulles (cas où l'animat est au but et correctement orienté) ni toutes identiques (animat au but mais mal orienté). La composante tangentielle échoue à ramener l'animat au but s'il est désorienté.

Il n'est donc pas surprenant que loin du but où les parallaxes sont fortes, le modèle rapproche l'animat du but (composante centripète positive) même en cas de désorientation moins importante que les parallaxes.

Même pour de faibles valeurs de désorientation, toutefois, c prend des valeurs négatives près du but, ce qui signifie qu'il existe des lieux au voisinage de l'objectif où le déplacement commandé par le modèle éloigne l'animat du but. Concrètement, le but n'est plus un point attracteur dès que l'animat est désorienté.

Sur les diagrammes, on a fait apparaître une barre verticale correspondant au diamètre du plus grand disque centré sur le but que l'animat peut parcourir sans heurter d'obstacle (voir section 3.1).

De cette figure, on apprend que la composante tangentielle a une très bonne composante centripète quelle que soit la distance au but. Toutefois, dès que l'animat n'est pas assuré de son orientation, la mesure est moins bonne, et l'écart-type augmente beaucoup, en particulier près du but, où cette mesure est particulièrement pertinente. Le problème le plus grave est que le but n'est plus atteint.

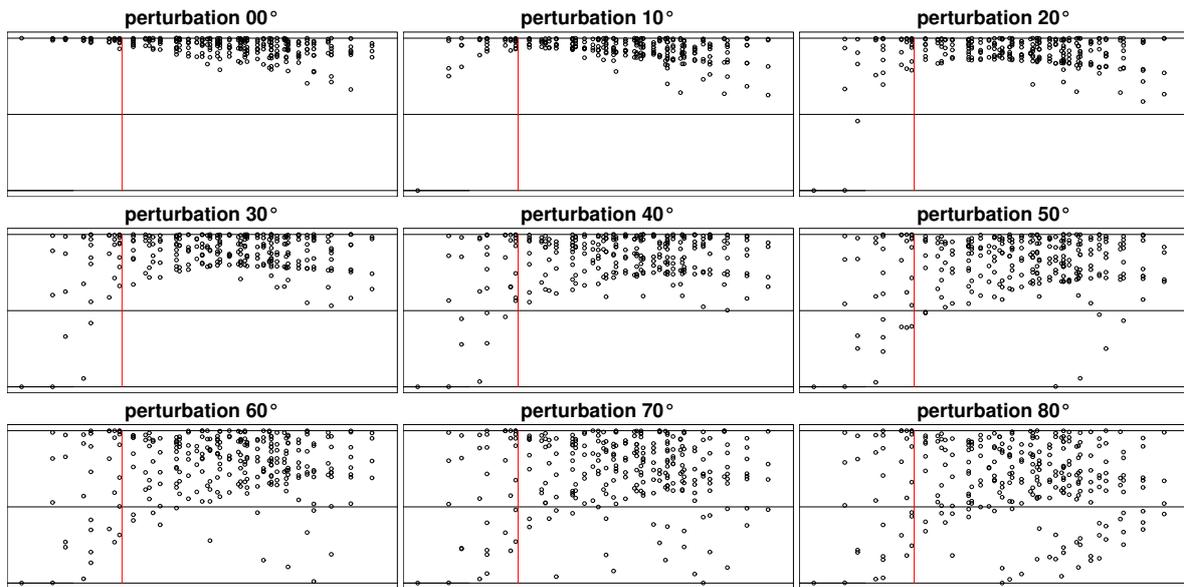
La conclusion de cette étude est que l'utilisation « naïve » de la composante tangentielle est donc bien dépendante du maintien par l'animat d'une bonne orientation.

Plus précisément, dans le cas d'orientation exactement connue pour lequel ils sont prévus, les modèles de la famille CC exploitant la composante tangentielle, ainsi que leurs successeurs, s'appuient sur des approximations qui aboutissent quand même à amener l'animat précisément au but. En revanche, une désorientation, même faible, empêche les mêmes modèles de ramener l'animat au but et même de le reconnaître. En d'autres termes, on ne peut considérer qu'une incertitude finie sur l'orientation de l'animat est une approximation comme les autres dans ces modèles : elle nécessite un traitement particulier.

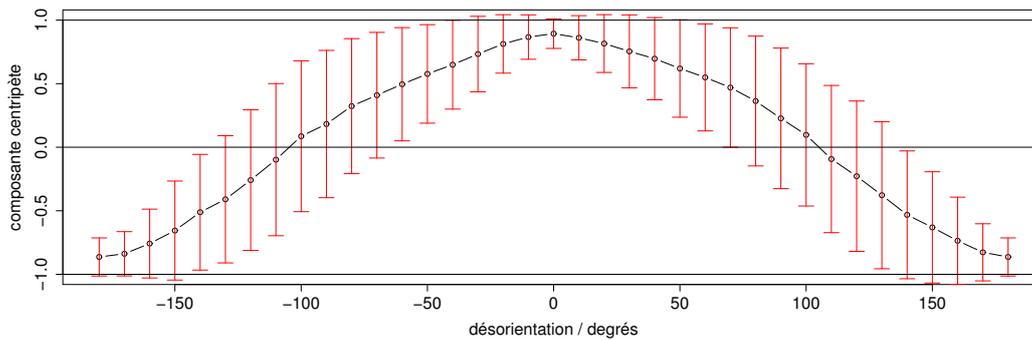
3.2.2 Indépendance de la composante radiale vis-à-vis de l'orientation

Nous avons vu au cours de la revue du chapitre précédent que la composante radiale applique des déplacements qui dépendent seulement d'une comparaison des largeurs apparentes observée dans les panoramas mémorisés et actuels. Si les vecteurs radiaux contribuant au déplacement sont bien appliqués chacun au centre de l'élément correspondant dans la vue actuelle (méthode directe, compatible avec le modèle PV), alors l'étape de calcul de direction est indépendante de l'orientation. Si l'étape d'appariement est aussi indépendante de l'orientation, alors tout peut fonctionner de façon identique quelle que soit l'orientation réelle du robot.

De la même façon que dans la section précédente, nous avons réalisé une simulation idéale (appariement parfait) avec la méthode directe, mais cette fois avec la composante radiale seule. La figure 3.4 illustre ce cas idéal. Elle montre comment, bien que la perception des parallaxes soit perturbée par la désorientation, la taille des amers est indépendante de l'orientation et donc les ordres moteurs exécutés sont en fait les mêmes, dans le référentiel de l'environnement, comme confirmé par la composante centripète en figure 3.3.



Performances de la composante tangentielle sur toute la surface



Performances de la composante tangentielle près du but

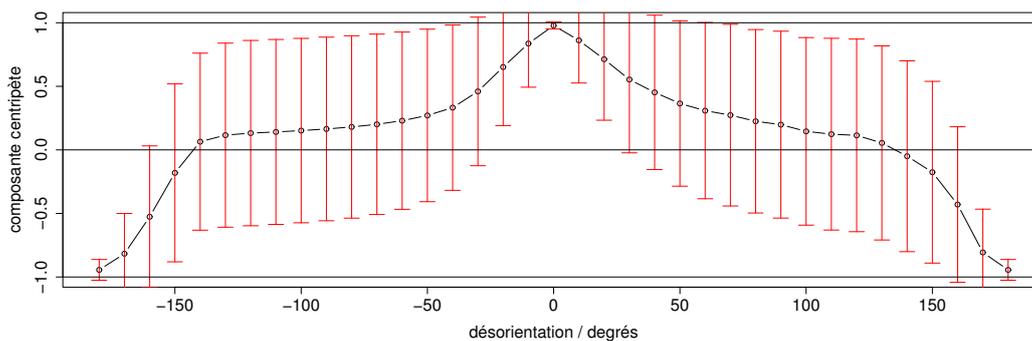


FIG. 3.3: Représentation graphique de la sensibilité de la composante tangentielle à l'orientation. (On utilise l'appariement parfait et la méthode directe pour ne pas introduire d'autre dépendance en orientation). Chaque petit schéma représente la composante centripète c en fonction de la distance au but, avec une barre verticale à la distance du premier amer (voir texte). Le premier grand schéma représente la moyenne sur toute la zone de test de la composante centripète, en fonction de la désorientation. Le second représente la même moyenne sur le plus grand disque centré sur le but et ne touchant aucun amer. Les barres d'erreur ont pour demi-longueur l'écart-type, ce qui explique pourquoi elles débordent parfois. Voir le texte pour l'interprétation détaillée.

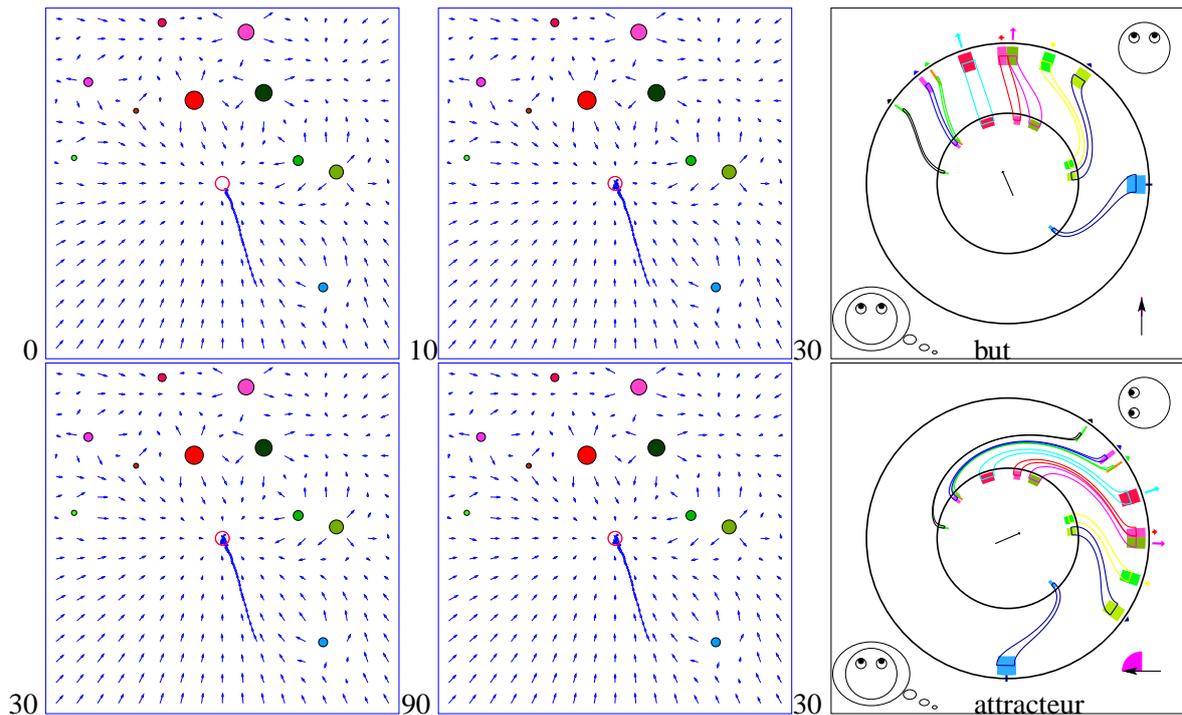
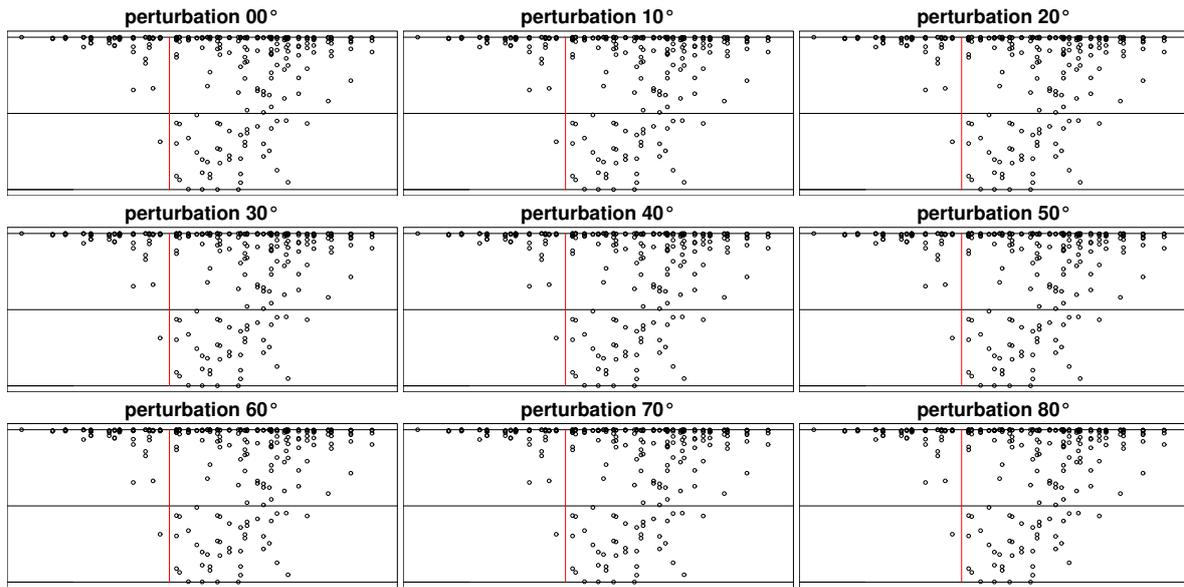
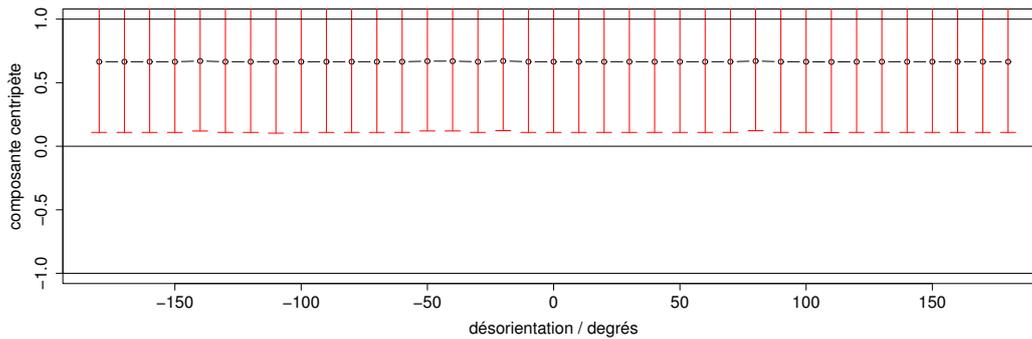


FIG. 3.4: Illustration de l'insensibilité de la composante radiale à l'orientation. (On utilise l'appariement parfait et la méthode directe pour ne pas introduire d'autre dépendance en orientation). Chaque schéma représente le champ de vecteurs et une trajectoire obtenue avec une perturbation d'angle donnée, indiquée en bas à gauche. Champs de vecteurs et trajectoires ne sont pas modifiés, indiquant que, dans ces conditions, l'animat n'est pas perturbé par une mauvaise orientation. À droite sont représentés, pour une perturbation de 0 puis 90° , la décision que l'animat prend au début de la trajectoire. L'animat croit percevoir les amers dans des directions différentes de celle de la vue mémorisée. Toutefois, la direction qu'il prend en réalité est la même que lorsqu'il est bien orienté. Cette expérience n'est possible qu'en simulation car elle suppose l'appariement parfait.



Performances de la composante radiale sur toute la surface



Performances de la composante radiale près du but

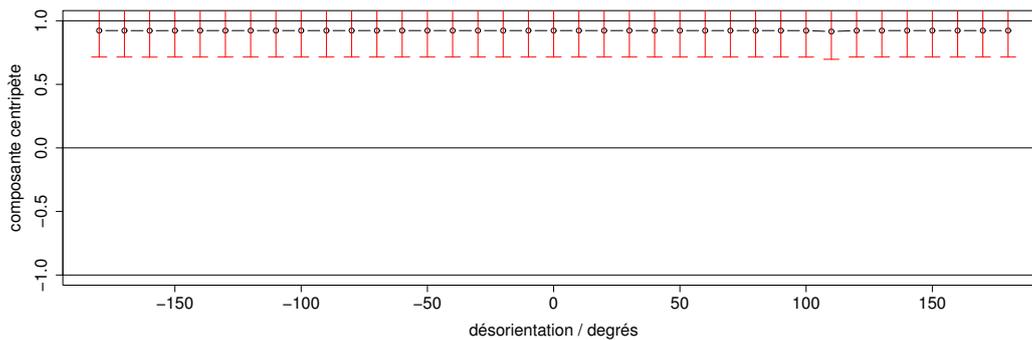


FIG. 3.5: Représentation graphique de l'insensibilité de la composante radiale à l'orientation. (On utilise l'appariement parfait et la méthode directe pour ne pas introduire d'autre dépendance en orientation). Les schémas sont équivalents à ceux de la figure 3.3. Contrairement à la composante tangentielle, la composante radiale n'est pas dépendante d'une bonne orientation. Près du but elle montre en simulation idéale des performances aussi bonnes que la composante tangentielle. Nous démontrons en section 3.5 que la composante radiale n'a pas les propriétés intéressantes de la composante tangentielle, et illustrerons en section 3.4.3 un cas de contournement d'obstacle.

L'observation de la composante centripète semble extrêmement en faveur de la composante radiale. Nous devons toutefois rappeler que la composante centripète n'est qu'une mesure analytique et ne garantit pas à l'animat une trajectoire sûre, même si les grandes valeurs à faible distance du but sont le signe que l'animat se dirige bien vers le but. Par ailleurs, ces résultats s'appuient sur des hypothèses précises que nous résumons : dans une simulation où l'animat peut mesurer des angles avec une précision d'un degré, avec l'environnement présenté, la méthode dite « composante radiale » du modèle CC5, modifiée pour s'appuyer sur l'appariement parfait donné par le simulateur, présente une composante centripète indépendante de l'orientation, et particulièrement élevée lorsque l'animat est plus près du but que tous les amers.

En d'autres termes, il existe des environnements où la composante radiale, dans des conditions idéales, donne d'excellents résultats.

3.3 Redéfinition et généralisation

Nous venons de voir que des mesures analytiques, dans une simulation sans bruit, donnaient un très bon score à la composante radiale. Pourtant, dès les premières expériences préliminaires que nous avons menées (en simulation comme sur robot réel), nous avons constaté que la composante radiale était généralement moins performante que la composante tangentielle. Il devient alors nécessaire de justifier pourquoi la composante tangentielle conserve notre préférence.

La définition des composantes radiale et tangentielle dans les modèles CC n'est pas minimale mais recouvre plusieurs différences. Le but de cette section est de parcourir ces différences.

Cette entreprise permet de se rendre compte que d'autres combinaisons que celles testées jusque-là sont possibles et pose la question de la combinaison à retenir pour des expériences réelles. Nous avons donc fait ce parcours avec à l'esprit l'application à un robot réel, en environnement aussi peu préparé que possible. Ceci nous a permis d'évoquer brièvement quelques ouvertures pertinentes à d'autres possibilités de modèles. Nous invitons le lecteur à se référer au schéma A.2 (annexe A) qui représente sous forme arborescente les choix techniques possibles, en précisant ceux explorés par les modèles existants. Les éléments de cette analyse posent des jalons que nous retrouverons dans la présentation de nos travaux.

Mais, de façon plus importante, ce parcours permet de distinguer, parmi les différences entre composante tangentielle et radiale au sens des modèles CC, laquelle est fondamentale.

Une fois cette distinction faite, nous aurons une définition des deux composantes dont la seule différence est la différence fondamentale. Nous pourrons alors aborder la section suivante où nous démontrons que la composante radiale n'a pas la propriété de composante centripète positive, contrairement à la composante tangentielle.

3.3.1 Appariement

La première des différences entre composantes radiales et tangentielles, dans les modèles CC, est à l'étape d'appariement. Il s'agit d'apparier des transitions ponctuelles, ou des secteurs étendus, comme illustré en figure 3.6.

Les modèles CC utilisent des azimuts de bords de secteurs, ou des largeurs apparentes de secteurs. Le fait de décorréler le type d'appariement du choix de la composante à appliquer signifie qu'il devient possible d'appliquer la composante tangentielle alors qu'on a apparié des secteurs. Il faut donc définir ce qu'est l'azimut d'un secteur. Le plus simple nous a paru de le définir comme l'azimut du centre de ce secteur, autrement dit l'azimut de la demi-droite qui coupe le secteur en deux parties égales.

Les simulations présentées ici utilisent un tel appariement de secteurs. Dans le cas particulier où la segmentation est parfaite et les amers circulaires (cas accessible notamment en simulation), la direction du centre du secteur est aussi la direction du centre de l'amer (s'il n'y a pas d'occlusion partielle). Cela permet d'expérimenter dans des conditions proches de celles des analyses théoriques que nous avons lues, où l'on suppose que chaque amer est associé à un point bien défini. En effet, l'application stricte de la composante tangentielle telle que définie dans les modèles CC fait intervenir deux transitions par amer, une à chaque bord, ce qui paraît moins naturel. Comme dans les travaux cités dans la revue du chapitre précédent, nous confondrons dans nos analyses l'azimut d'un amer avec l'azimut de son point centre.

Il est important de préciser que cela n'invalide pas l'application des résultats théoriques à des expériences utilisant l'appariement de transitions. Simplement, les objets sur lesquels s'appliquent les raisonnements sont des transitions dans le panorama, par exemple la transition entre une porte colorée et le chambranle blanc.

Nous avons, dans nos expériences réelles, retenu l'appariement de transitions. En effet, l'appariement de secteurs suppose qu'à un secteur d'une vue correspond un seul secteur de l'autre vue. Or un même amer peut être représenté par deux secteurs, voire plus. Par exemple, un petit amer peut masquer le milieu d'un gros. Sur un robot réel la segmentation peut avoir été excessive, découpant un même amer en plusieurs secteurs. Dans les deux cas, le centre d'un secteur est donc une variable mal définie, sujette à des erreurs de mesure. L'appariement de transitions ne souffre pas de ce problème : une transition est un contraste visible, de direction bien définie.

L'autre raison pour laquelle nous préférons l'appariement de transition dans les expériences réelles est la spécificité plus grande des appariements de transitions, comme illustré en figure 3.6. Même si, pour des signatures alternant secteurs noirs et secteurs blancs comme dans les modèles CC, il n'y a que deux types de secteurs (noir, blanc) et de transitions (noir-blanc, blanc-noir), cela n'est plus le cas avec des signatures plus riches (niveaux de gris ou couleurs, par exemple). L'information d'une signature plus riche sert à éviter les erreurs lors des appariements. Un appariement de transition fait naturellement intervenir deux paires de couleurs. Il est donc plus sélectif que s'il ne tient compte que d'une paire de couleurs.

En conclusion de ce point, nous raisonnerons en simulation avec appariement de secteurs, mais nos

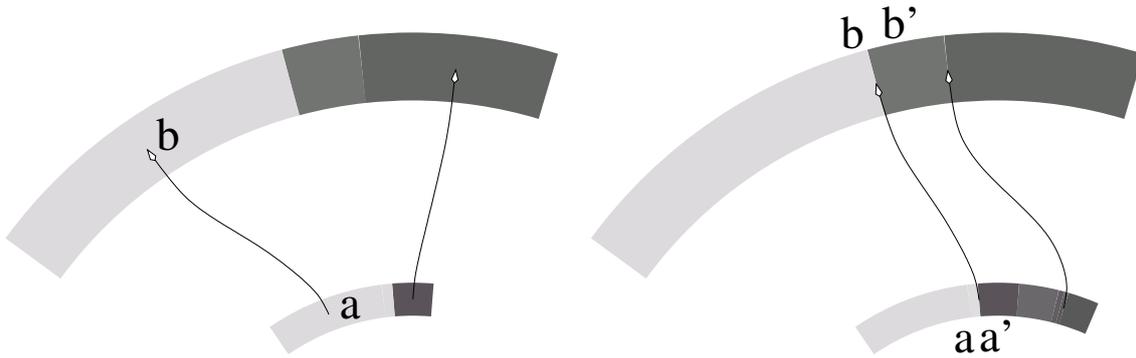


FIG. 3.6: À gauche, appariement de secteurs. Il doit prendre en compte les caractéristiques des deux secteurs a et b . À droite, appariement de transitions. Il doit logiquement prendre en compte les caractéristiques de quatre secteurs : a , a' , b et b' . Ce choix peut être décorrélé du choix de la composante radiale ou tangentielle, alors qu'il est corrélé dans les modèles CC : la composante radiale utilise des appariements de secteurs, la composante tangentielle des appariements de transitions.

expériences réelles utilisent l'appariement de transitions. Toutefois, la distinction entre appariement de secteur et de transition n'est pas la raison fondamentale des différences entre « composante radiale » et « composante tangentielle », au sens des modèles CC.

3.3.2 Direction à suivre

La différence principale entre composantes tangentielles et radiales réside dans le choix d'estimer la direction à l'aide des variations d'azimut (par rapport à une référence extérieure comme le nord), ou bien des variations de largeur apparente (on n'a pas besoin de référence extérieure pour estimer une largeur apparente). Ceci est illustré dans la figure 3.7. C'est essentiellement cette distinction qui cause des vecteurs tangentiels ou radiaux, c'est donc cette distinction que nous désignerons par la suite quand nous emploierons les expressions « composante radiale » et « composante tangentielle ».

Il apparaît aussi que le choix, évoqué par Cartwright et Collett, de tenir compte de tous les secteurs – en particulier ceux représentant l'absence d'obstacle (secteurs blancs dans les modèles CC), ou seulement ceux représentant des obstacles (secteurs noirs) – peut aussi être décorrélé des autres choix. Pour nos expériences réelles, nous avons tenu compte de tous les éléments appariés (secteurs ou transitions).

3.3.3 Méthodes directe, inverse et moyenne

Nous mettons momentanément entre parenthèse la question de l'orientation pour définir une généralisation de la « méthode moyenne » introduite par Hong et al. et expliquée en section 2.3.1 en remarquant qu'elle est un cas particulier d'une idée plus générale. Rappelons que chaque méthode énumère les amers appariés, et associe à chacun un vecteur qui va contribuer au calcul de la direction

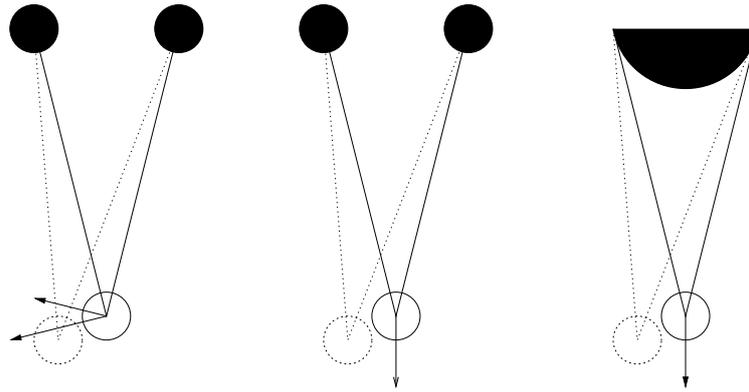


FIG. 3.7: Redéfinition plus précise des composantes radiale et tangentielle. Sur chaque schéma, l'animat au but est représenté en pointillés, la position actuelle en trait plein. À gauche, la composante dite « tangentielle » : pour chaque amer, la parallaxe causée par le déplacement donne naissance à un vecteur indiquant une direction à suivre pour réduire cette parallaxe. Au milieu, la composante dite « radiale » : c'est la largeur apparente entre deux amers qui intervient. La variation de cette largeur au cours du déplacement de l'animat permet d'estimer un vecteur indiquant une direction à suivre pour la réduire. Dans ces deux premiers cas, l'animat apparie les secteurs. À droite, l'appariement de transitions. Parce que chaque objet présente deux bords dans le panorama, la composante radiale peut être appliquée à la largeur apparente d'un seul objet. C'est uniquement cette dernière possibilité qui est utilisée dans les modèles CC, et non la précédente.

à suivre. La méthode d'origine applique chaque vecteur dans la direction dans laquelle l'amer associé se trouve actuellement. La méthode « inverse » revient en fait à appliquer chaque vecteur dans la direction dans laquelle se trouvait l'amer associé dans la vue prise depuis le but. On peut étendre ceci en définissant un paramètre permettant à la fois une interpolation et une extrapolation de cette idée.

L'extension consiste, pour un amer mémorisé en θ_{mem} et vu en θ_{vu} à appliquer le vecteur individuel à un angle

$$\theta = \theta_{mem} + \lambda(\theta_{vu} - \theta_{mem})$$

dépendant d'un paramètre λ pouvant varier continûment¹.

$\lambda < 0$	extrapolation « extrême inverse »
$\lambda = 0$	méthode « inverse »
$\lambda = 0.5$	méthode « moyenne » (Hong et al.)
$0 < \lambda < 1$	interpolation
$\lambda = 1$	Méthode initiale recommandée par Weber et al.
$\lambda > 1$	extrapolation « évitement d'obstacle extrême »

Cette idée est applicable à tous les modèles descendants de CC, avec deux remarques toutefois. Premièrement, le modèle DV utilise naturellement la méthode « moyenne », mais on peut le modifier

¹On peut remarquer l'analogie entre ce paramètre lambda et le paramètre « stéréo » en traitement du signal audio. 0 correspond à mono, 1 à stéréo. Dans les deux cas, les valeurs entre 0 et 1, bornes comprises, ont des résultats raisonnables, alors que des valeurs plus extrêmes peuvent produire des aberrations.

pour y introduire le paramètre λ . Deuxièmement, cette modification n'est pas applicable à l'ALV. Dans la mesure où ce dernier n'apparie pas explicitement, les angles mémorisés et vus ne sont pas disponibles, et les écarts sont implicitement appliqués comme avec le modèle DV par la méthode « moyenne ».

Nous avons vu que seul le cas $\lambda = 1$ n'introduit pas de perturbation en cas de désorientation. Par ailleurs, nous anticipons légèrement la démonstration qui sera faite à propos de la capacité de la composante tangentielle à donner une composante centripète positive, en indiquant dès à présent que la conservation de cette capacité s'applique aux cas $0 < \lambda < 1$.

3.4 Différences fondamentales

Après avoir mis en évidence sur des mesures analytiques dans un environnement d'un certain type la dépendance de la composante tangentielle vis-à-vis de l'orientation, et l'indépendance de la composante radiale, nous illustrons et expliquons plus en profondeur pourquoi la composante radiale est plus limitée que la composante tangentielle, même sans tenir compte de contraintes pratiques (qui seront illustrées dans la section suivante).

3.4.1 Rappel de géométrie : arc capable d'un angle donné

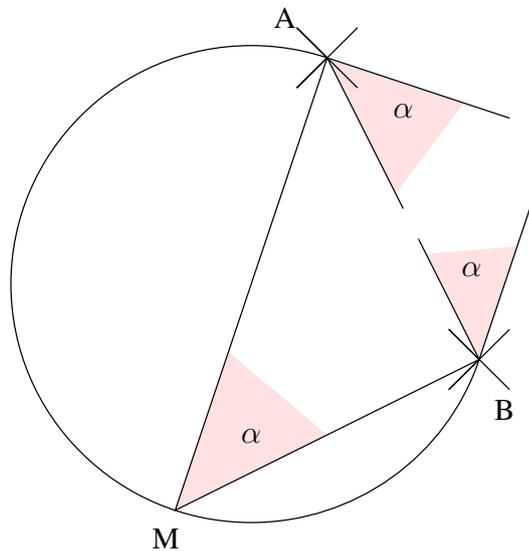


FIG. 3.8: Étant donnés deux points A et B du plan (représentés ici par des croix), l'ensemble des points M du plan depuis lesquels on voit les deux points séparés par un angle apparent α est un arc de cercle passant par les deux points et tel que les tangentes au cercle en A et B font chacune avec le segment $[AB]$ un angle α .

Ce point est illustré en figure 3.8. Étant donnés deux points A et B du plan (représentés ici par des

croix), l'ensemble des points M du plan depuis lesquels on voit les deux points séparés par un angle apparent α est un arc de cercle passant par les deux points et tel que les tangentes au cercle en A et B font chacune avec le segment [AB] un angle α .

3.4.2 Composante tangentielle

Dans cette section, nous présentons une démonstration. Sa formulation introduit des observations que nous retrouverons à deux moments dans la présentation de notre modèle : dans notre méthode de calcul de la direction à suivre (section 5.3), et dans une de nos méthodes d'orientation (section 6.2).

Revenons au cas de la composante tangentielle, dans le cas où l'orientation est connue. Franz a démontré [FRA 98b] que sous les hypothèses suivantes :

- au moins 3 amers non alignés,
- mesures de parallaxes sans erreur (ce qui implique appariement sans erreur),

le modèle PV fournit un vecteur dont la composante centripète est positive.

Nous avons conçu une démonstration différemment formulée et un peu plus générale. Nous supposons aussi que nous pouvons mesurer des parallaxes sans erreur et que l'environnement présente au moins deux (et non trois) amers non alignés avec le but. En effet, l'hypothèse de la présence de trois amers non alignés assure que le but n'est jamais aligné avec tous les amers, mais n'est pas en elle-même nécessaire.

L'observation d'une parallaxe fournit essentiellement une indication de signe. Le signe permet à l'animat de séparer l'espace en deux demi-espaces et de savoir dans lequel de ces demi-espaces se trouve le but. Ceci est illustré dans la figure 3.9.

L'essentiel de la démonstration tient dans le fait que, sous ces hypothèses, chaque vecteur individuel \vec{v}_i^d (voir figure 3.9) fait un angle de moins de 90° avec la direction qui mène au but, et a donc une composante centripète positive. Autrement dit :

$$\vec{\text{MB}} \cdot \vec{v}_i^d \geq 0$$

Le cas d'égalité correspond à une parallaxe de 0 ou un demi-tour. Nous avons éliminé par hypothèse le cas où le but et tous les amers sont alignés, donc l'inégalité est stricte pour au moins un i . Posons pour tout i , $\vec{v}_i = \vec{v}_i^d$. Pour tout ensemble c_i de poids positifs, on peut faire une somme sur tous les amers i et obtenir l'inégalité stricte :

$$\sum_i \left(\vec{\text{MB}} \cdot c_i \vec{v}_i \right) = \vec{\text{MB}} \cdot \underbrace{\left(\sum_i c_i \vec{v}_i \right)}_{\vec{V}} > 0 \quad (3.1)$$

Cela signifie que toute combinaison linéaire \vec{V} des \vec{v}_i à coefficients positifs est un vecteur de composante centripète positive.

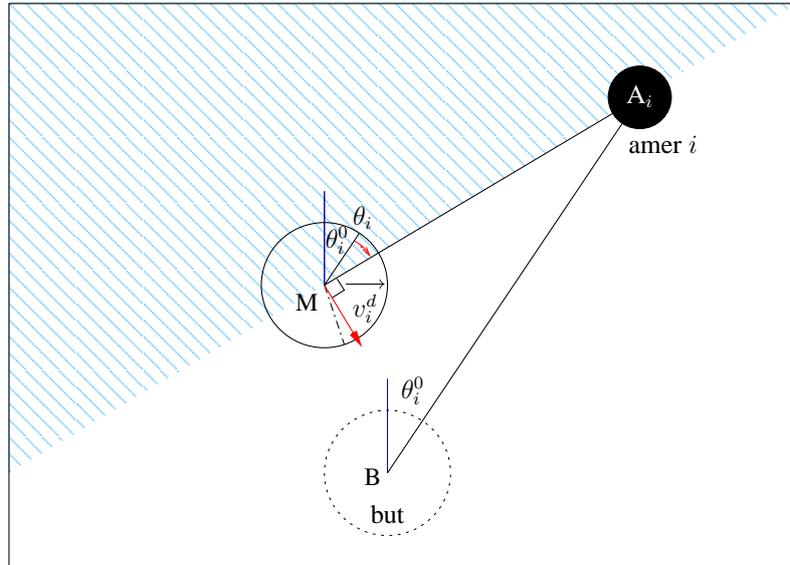


FIG. 3.9: La composante tangentielle, sous certaines hypothèses (voir texte), est garantie d'avoir une composante centripète positive. L'animat (cercle de centre M) compare l'azimut actuel θ_i de l'amer i avec l'azimut vu du but (cercle pointillé de centre B) θ_i^0 . Quelle que soit la position de M (sauf parallaxe nulle ou d'exactly un demi-tour), l'observation du signe de la parallaxe détermine un demi-espace (délimité par la droite MA) dans lequel le but ne se trouve pas (hachures). Dans cet exemple la parallaxe est négative, l'azimut du but est donc entre θ_i et $\theta_i - \pi$ (côté non hachuré). La méthode directe considère un vecteur \vec{v}_i orthogonal à la frontière de ce demi-espace, du côté non hachuré, pointant vers l'azimut $\theta_i - \pi/2$. Autrement dit, \vec{v}_i^d et \vec{MB} sont dans le même demi-plan. Dans le repère orthogonal (\vec{v}_i^d, \vec{MA}) , il est clair que $\vec{v}_i^d \cdot \vec{MB} > 0$: \vec{v}_i^d a une composante centripète positive. Une combinaison linéaire de tels vecteurs, de coefficients positifs, conservant cette propriété (voir texte), les modèles CC4 et PV ont une composante centripète positive. Les méthodes inverses et moyennes (Hong, DV) aussi (voir texte).

Supposons que les $\overrightarrow{v_i}$ sont normés. Si $\forall i, c_i = 1$, alors on reconnaît dans l'expression de \overrightarrow{V} le modèle CC4. Si $\forall i, c_i = |\theta_i - \theta_i^0|$, alors on reconnaît le modèle PV. Les modèles CC4 et PV ont donc une composante centripète positive.

Intéressons nous à la méthode inverse. L'argument de Hong (2.3.1.3), peut être appliqué ici : on inverse les points M et B et on applique la même démonstration. Le résultat est que la méthode inverse a aussi une composante centripète positive. Notons $\overrightarrow{v_i^i}$ les vecteurs individuels de la méthode inverse.

La méthode moyenne fait une moyenne entre les méthodes directe et inverse, ce qui conserve la propriété de composante centripète positive. Formellement, on pose $\overrightarrow{v_i} = \frac{\overrightarrow{v_i^i} + \overrightarrow{v_i^d}}{2}$.

La démonstration couvre ainsi en particulier les cas suivants :

$$\begin{array}{ll}
 \text{CC4 :} & \overrightarrow{v_i} = \overrightarrow{v_i^d} \quad c_i = 1 \\
 \text{PV :} & \overrightarrow{v_i} = \overrightarrow{v_i^d} \quad c_i = |\theta_i - \theta_i^0| \\
 \text{Hong :} & \overrightarrow{v_i} = \frac{\overrightarrow{v_i^i} + \overrightarrow{v_i^d}}{2} \quad c_i = |\theta_i - \theta_i^0| \\
 \text{DV :} & \overrightarrow{v_i} = \frac{\overrightarrow{v_i^i} + \overrightarrow{v_i^d}}{2} \quad c_i = |\sin(\theta_i - \theta_i^0)|
 \end{array}$$

Cette démonstration prouve que les modèles CC4, PV, Hong et DV ont la capacité à approcher l'animat du but jusqu'à l'atteindre, même si dans le détail la trajectoire ne sera pas la même car les c_i dépendent du modèle.

Intéressons-nous maintenant au cas où l'animat subit une légère désorientation et rappelons-nous la remarque faite en section 2.2.3 à propos du modèle CC4 (méthode directe, composante tangentielle seule) en faisant abstraction des problèmes d'appariement. La remarque portait sur le fait qu'une désorientation inférieure à la plus petite parallaxe ne perturbe absolument pas CC4, car chaque vecteur individuel est normé et ne dépend que du signe de la pseudo-parallaxe (section 2.1.5.2) associée. Le cadre de notre démonstration nous permet de formaliser cela par une perturbation des c_i . Si aucune des pseudo-parallaxes n'a changé de signe à cause de la désorientation, la démonstration reste valable malgré la perturbation des c_i . Cela signifie qu'en tout point, le modèle PV est perturbé par une désorientation (la direction calculée change) mais, si la perturbation est plus petite que la plus petite parallaxe, la propriété de composante centripète positive reste garantie.

Est-il donc toujours possible de rejoindre le but malgré une désorientation finie ? L'expérience de la figure 3.2 montrait que non. En effet, rappelons que les parallaxes tendent vers 0 à mesure que l'animat s'approche du but. Une erreur d'orientation constante, quelle que soit son importance, invalide donc cette démonstration (spécifiquement : le fait que tous les c_i soient positifs). À mesure que l'animat s'approche du but, c'est le changement de signe des pseudo-parallaxes (par rapport aux vraies parallaxes) qui finit par perturber le vecteur calculé, au point qu'il amène l'animat vers un point différent. Nous nous servons de cette observation en section 6.2.

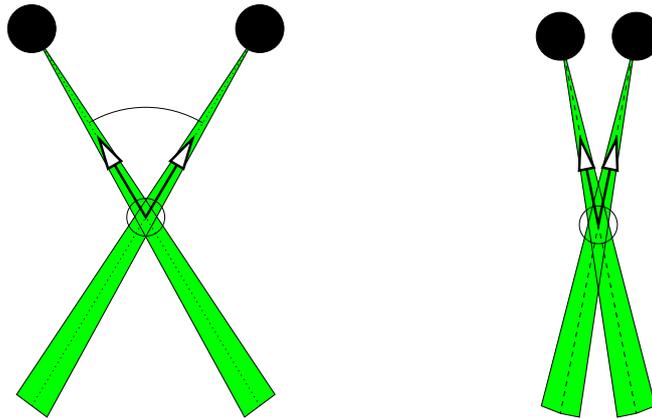


FIG. 3.10: Qualité de guidage. À gauche : il suffit de deux amers visibles suivant un angle qui ne soit ni proche de 180° ni proche de 0° , le cas idéal étant 90° , pour que la composante radiale soit précise malgré de possibles petites incertitudes sur les azimuts. Ceci est illustré par la faible étendue de l'intersection des régions ombrées (le quadrilatère inclus dans le cercle). À droite : si tous les amers sont tous proches d'une même direction (modulo 180°), un déplacement dans cette direction fait peu varier les parallaxes. La même petite incertitude aboutit à une région beaucoup plus étendue le long de cette direction (le quadrilatère déborde). La qualité de guidage, applicable quel que soit le nombre d'amers, est une mesure capable de détecter une situation de mauvaise précision : le déterminant entre les deux vecteurs représentés est proche de 1 dans le premier cas, et faible dans le second cas.

3.4.2.1 Qualité de guidage

Au cours de nos expériences, il nous a paru utile d'introduire une notion et une mesure permettant de chiffrer la qualité d'une situation de guidage. Nous l'avons appelée « qualité de guidage »².

Nous avons vu qu'il est nécessaire de disposer d'au moins deux amers et d'un but non aligné avec eux, pour qu'en théorie l'animat puisse rejoindre le but, en se déplaçant de façon à annuler les parallaxes observées. Si un seul amer est visible, l'animat ne pourra faire mieux que de s'aligner sur la droite joignant le but et l'amer.

En théorie, la présence d'un autre amer non aligné suffit à ramener l'animat au but. Mais en pratique, si les amers sont rassemblés dans une même direction du panorama (modulo 180°), les incertitudes de mesure de parallaxes aboutiront à une mauvaise précision, comme illustré en figure 3.10.

Nous définissons la qualité de guidage d'un panorama utilisé pour rejoindre un point comme le maximum des valeurs absolues de déterminants des vecteurs pointant vers les amers, pris deux à deux. La figure 3.10 le montre. Dans le cas où il y a un seul ou aucun amer, la qualité de guidage est nulle.

Nous définissons la qualité de guidage d'un ensemble panorama mémorisé – vue actuelle de la même façon, mais le calcul est restreint à l'ensemble des amers effectivement appariés entre les deux vues. La qualité de guidage d'une situation est donc inférieure à la qualité de guidage intrinsèque d'un panorama.

²« Qualité de guidage » se traduirait par « guidance quality ».

Muni de cette mesure, lorsque l'animat prend un panorama, il est capable de détecter le cas où une faible qualité de guidage de ce panorama le rend inutilisable pour une tâche de retour au nid.

La qualité de guidage pourrait être utilisée lors de la navigation ou lors de la construction de carte. Nous ne l'avons utilisée que parmi les critères d'arrêt de la trajectoire lors d'une étape de retour au nid : une bonne qualité de guidage est considérée comme un critère nécessaire pour considérer que le but est atteint.

3.4.3 Composante radiale

Nous présentons dans cette section une contre-démonstration, complémentaire de celle de Franz. En effet, nous mettons en évidence que la composante radiale telle quelle ne bénéficie par d'une propriété similaire à celle de la composante tangentielle. Dans certains cas, son application peut éloigner l'animat du but.

3.4.3.1 Zone centrifuge avec un amer

Nous allons faire les mêmes hypothèses que dans la démonstration précédente (pas d'erreur d'appariement ni de mesure d'angle) pour montrer qu'il existe, pour tout but B et tout amer O, une zone de l'environnement dans laquelle l'application de la composante radiale sur cet amer aboutit à un vecteur éloignant l'animat du but.

Nous invitons le lecteur à se référer à la figure 3.11 pour suivre la démonstration. L'animat a pris un panorama d'un point B. On considère un amer O qu'il a vu. Désignons par M le point (quelconque) où l'animat se trouve actuellement. Pour savoir si la composante centripète est positive ou négative, il faut estimer si l'angle \widehat{BMO} est supérieur ou inférieur à 90° . L'ensemble des points où cet angle a une valeur α donnée est un arc de cercle, dit « arc capable de l'angle α ». Dans le cas particulier qui nous intéresse, où $\alpha = 90^\circ$, l'arc est le demi-cercle de diamètre OB, complété par symétrie (cas noté 1 sur le schéma, où A joue le rôle de M). L'angle \widehat{BMO} est donc supérieur à 90° si et seulement si M est à l'intérieur du disque D2 de diamètre BO.

Quand l'animat est hors du disque de centre O et rayon OB, que nous nommons D1, la composante radiale a toujours le même sens : dirigée vers O. L'angle \widehat{BMO} est inférieur à 90° car D2 est inclus dans D1 et l'animat est hors de D1. On conclut donc que, dans le cas où l'animat doit s'approcher de l'amer concerné, le vecteur individuel radial résultant l'approche aussi nécessairement du but.

Quand l'animat est dans le disque D1, le sens du vecteur radial est inversé : il éloigne l'animat de l'amer O. Si l'animat est en dehors du disque D2, l'angle \widehat{BMO} est toujours inférieur à 90° , mais le changement de sens du vecteur implique que la composante centripète a changé de signe et est donc devenue négative. Cette transition est illustrée en 2 sur le schéma.

Quand l'animat est dans le disque D2, il est aussi dans D1. L'angle \widehat{BMO} dépasse 90° , ce qui change à nouveau le signe de la composante centripète, qui redevient positive (cas 3 sur le schéma).

La composante radiale est donc centripète sur presque tout l'environnement, mais centrifuge sur la zone $D1 \cap \overline{D2}$. Il n'y a donc pas de garantie que l'application de la composante radiale donne toujours un vecteur centripète.

3.4.3.2 Zone centrifuge avec plusieurs amers

La contre-démonstration qui précède ne considère qu'un amer. Or, le fait qu'un amer puisse parfois éloigner l'animat du but ne prouve pas que, s'il y a plusieurs amers, l'effet persiste. Et même s'il persiste, cela ne prouve pas que l'animat ne pourra pas rejoindre tout de même le but. Il faut donc compléter par un exemple montrant un cas où plusieurs amers aboutissent effectivement à une somme vectorielle qui éloigne l'animat du but, et que cela crée un attracteur indésirable. L'exemple est donné en figure 3.12.

3.4.3.3 Composante radiale centripète

Cet exemple est l'occasion de remarquer le sens particulier qu'on peut donner aux cas d'utilisation de la composante radiale. Si l'animat, comparant la largeur apparente d'un amer avec sa valeur vue du but, conclut qu'il doit s'approcher de l'amer, alors cela aboutit nécessairement à le rapprocher du but. C'est même ce cas qui, essentiellement, rapproche l'animat du but. Si, en revanche, il estime devoir s'éloigner de l'amer, il se peut (mais pas toujours) que cela l'éloigne aussi du but. Ce second cas ressemble à de l'évitement d'obstacle.

Si l'animat applique une composante radiale modifiée, dans laquelle il ignore les vecteurs individuels qui l'éloignent des amers pour n'exploiter que ceux qui commandent un rapprochement, alors la méthode est garantie d'avoir une composante centripète positive (sauf si tous les amers ont été éliminés).

Si l'amer et le but sont à l'intérieur d'un même triangle dont les sommets sont des amers alors, en tout point sauf le but, au moins un amer est plus loin de l'animat (et paraît donc moins large). Cette hypothèse est suffisante pour éviter le cas où tous les amers sont éliminés.

Nous appellerons cette méthode modifiée la composante radiale centripète.

3.5 Mise en défaut de la composante radiale

La section précédente donnait des arguments théoriques expliquant pourquoi la composante radiale devrait être moins performante que la composante tangentielle. Dans cette section, nous formulons des remarques plus pratiques et présentons deux expériences qui mettent en évidence comment la composante radiale peut être mise en défaut.

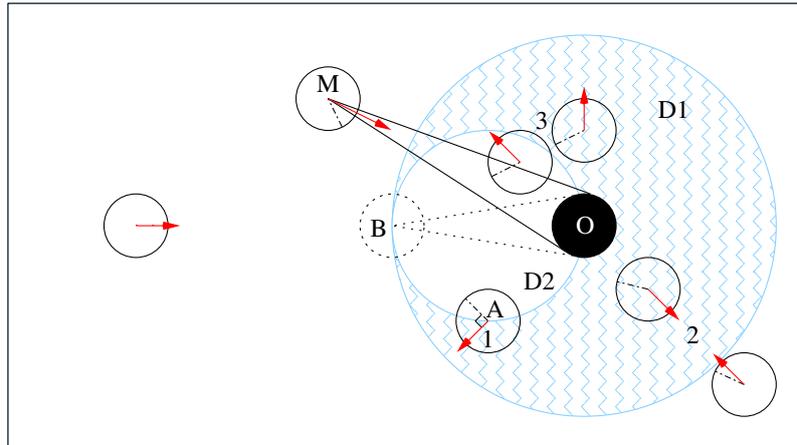


FIG. 3.11: La composante radiale n'a pas de propriété équivalente à celle énoncée en figure 3.9. Au contraire, pour tout amer O et tout but B, il existe une zone de l'environnement (zone de zigzags) où l'utilisation de cet amer par l'animat donne un vecteur centrifuge. Cette zone est délimitée, d'une part, par le cercle de diamètre BO, d'autre part, par le cercle de centre O et de rayon OB (voir texte). En situation, l'animat ne peut savoir où est cette zone. La composante radiale sans modification ne garantit donc pas un vecteur centripète.

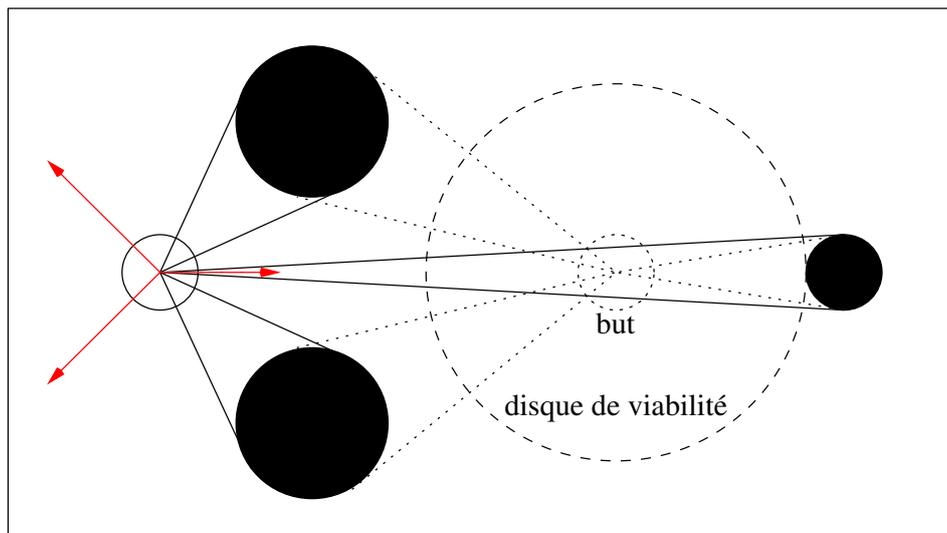


FIG. 3.12: Limitation de la composante radiale à faire une synthèse des informations recueillies sur les largeurs apparentes. Dans cet exemple, la répulsion des deux gros amers compense l'attraction du petit, créant un attracteur indésirable. On peut modifier le modèle pour qu'il ne tienne compte que des amers dont la largeur apparente actuelle est plus petite que la largeur apparente vue du but. L'animat ignorera alors les deux gros amers et se dirigera vers le plus petit. Toutefois, cela signifiera aussi annuler complètement l'évitement d'obstacle dont bénéficiait spontanément le modèle de départ. Ce nouveau modèle est viable, au moins dans le disque centré sur le but et tangent à l'amer le plus proche (intérieur du cercle en tirets), mais l'animat n'a pas de moyen de savoir s'il est dans ce disque. Il est aussi viable dans le triangle formé par les trois amers autour du but. La composante tangentielle évite les obstacles et nous semble viable dans de plus nombreux cas.

3.5.1 Problème de satisfaction des hypothèses

Tout d'abord, même en simulation et avec des appariements parfaits, l'hypothèse de mesure de largeurs apparentes sans erreur est fautive à cause des occlusions entre amers : un amer partiellement occulté semblera présenter une largeur apparente inférieure, que le modèle ne peut distinguer d'un éloignement. Ce défaut concerne la composante radiale normale et centripète. La composante tangentielle, en s'appuyant uniquement sur des azimuts isolés, n'est pas gênée de cette façon par des occlusions.

Ensuite, le même problème prend une tournure plus importante sur un robot réel, où une segmentation défaillante peut aboutir à découper en plusieurs secteurs dans un panorama un amer qui était d'un seul tenant dans un autre. Cette fragmentation intempestive des amers perturbe particulièrement la composante radiale, car la largeur apparente d'un fragment est souvent beaucoup plus petite que l'amer complet. Là aussi, la composante tangentielle n'est pas concernée.

Enfin, un défaut de la composante radiale en pratique (plus apparent en robotique réelle) réside dans sa sensibilité aux erreurs de mesures. En effet, une décision (calcul d'un vecteur individuel) dépend de quatre mesures d'azimut (azimuts des deux bords de l'amer dont on fait la différence, dans chacune des deux vues). Les conséquences des erreurs d'appariement, et les incertitudes de mesures sont donc plus importantes que pour la composante tangentielle où deux mesures suffisent.

Sahin et Gaudio [SAH 98], dans le but d'utiliser la largeur apparente comme une véritable mesure de distance, proposent de traiter le problème des occlusions en s'appuyant sur la connaissance de la distance que le robot parcourt entre deux observations. Il devient alors possible de contrôler la cohérence des mesures de largeur apparente et ignorer les amers présentant des mesures incohérentes. Cela est intéressant, mais nous n'avons pas continué dans cette voie. En effet, cela sort du cadre que nous nous sommes fixé de ne pas supposer que l'animat pouvait faire un lien entre mouvement commandé et mouvement réellement effectué. Nous profitons toutefois d'avoir mentionné les liens entre mesure de largeur apparente et mesure de distance pour signaler que les raisonnements tenus dans ce chapitre, notamment à propos de la composante radiale centripète, sont applicables à un robot équipé d'un capteur capable de mesurer des distances et d'apparier les mesures, comme celui de Einsele [EIN 97, EIN 01]. Un tel robot serait ainsi capable de faire du guidage par mesure de distances, de la même façon que nous faisons du guidage par mesures d'angles.

3.5.2 Illustration expérimentale

Nous présentons en figures 3.13 et 3.14 des simulations qui illustrent ces points. La composante radiale ne fonctionne pas bien dans un environnement où des amers sont groupés près du but : elle présente un attracteur indésirable qui cause un trou dans la surface de captation (non explicitée, mais visible d'après les flèches). Dans cet exemple, la composante radiale centripète aboutit à éliminer tous les amers sur une partie de l'environnement, ce qui diminue finalement la surface de captation. La composante tangentielle, de son côté, présente une surface de captation totale ou presque.

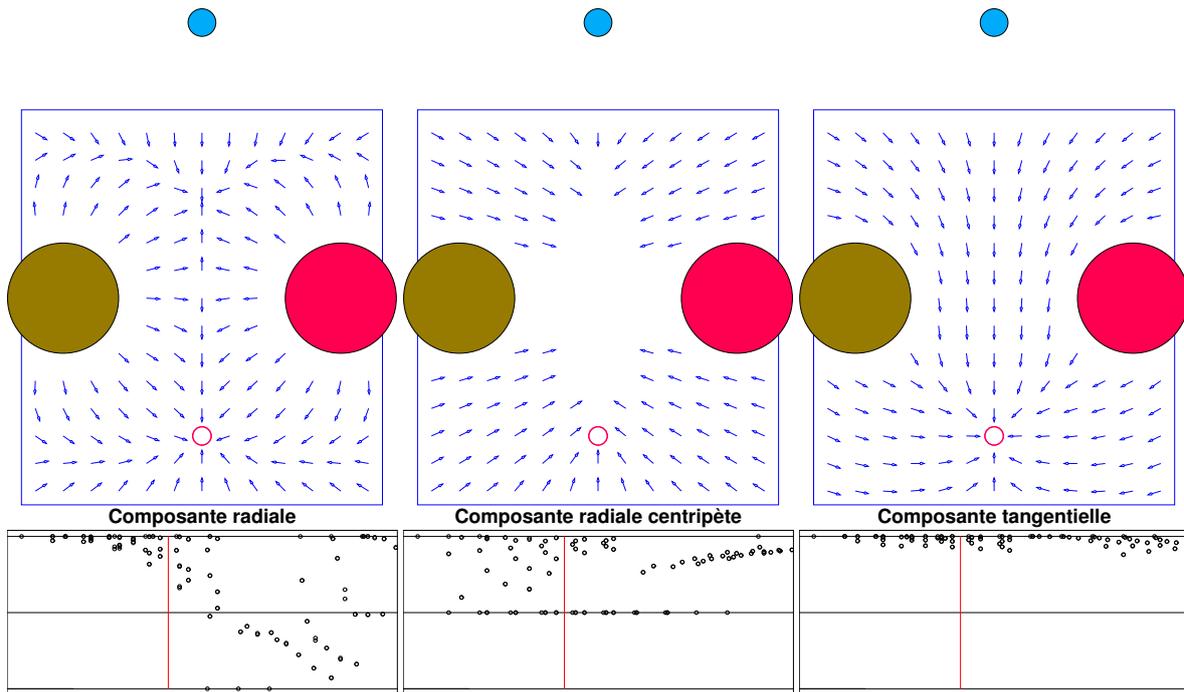


FIG. 3.13: Mise en défaut de la composante radiale (à gauche) et de la composante radiale centripète (au milieu), comparées à la composante tangentielle (à droite). Le but est le cercle ouvert en bas. L'amer au nord est visible. À gauche : l'effet répulsif des deux gros amers crée un attracteur indésirable, presque symétrique du but. Son effet sur la composante centripète est bien visible (portion négative aux deux-tiers de la distance maximale). Au centre, la composante radiale centripète échoue. Elle aurait pu fonctionner, si le but avait été à l'intérieur du triangle représenté par les trois amers, car en tout point l'algorithme aurait utilisé au moins un des amers. Ici, au centre de l'arène, les trois amers sont trop près et donc ignorés. À droite, la composante tangentielle résout élégamment le problème, et la présence de l'amer au nord n'est même pas nécessaire.

Remarquons que la seconde illustration est un cas de situation en miroir, déjà connu [GAU 00b, ZIP 85].

3.6 Synthèse

Cette analyse approfondie des composantes radiale et tangentielle nous a permis de mieux comprendre pourquoi les modèles descendants de CC étudiés jusqu'ici, modèles utilisant la composante tangentielle, ne peuvent pas fonctionner correctement en l'absence d'information sur la direction de l'animat.

L'analyse de la composante radiale, qui semblait une alternative intéressante, a fait apparaître à quels points elle est qualitativement différente dans son fonctionnement de la composante tangentielle et ne présente pas les mêmes garanties que cette dernière. Nous avons illustré par une contre-démonstration et des exemples la différence de performance face à des groupes d'obstacles.

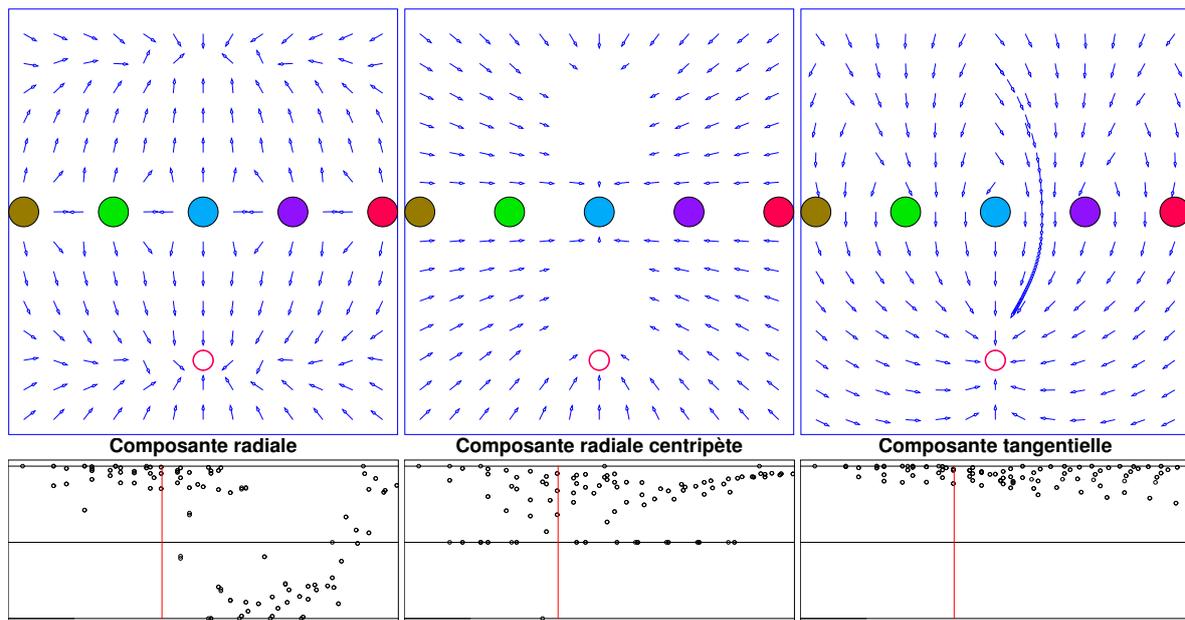


FIG. 3.14: Mise en défaut de la composante radiale (à gauche) et de la composante radiale centripète (au milieu), comparées à la composante tangentielle (à droite). À gauche : le cas est similaire au précédent. Au milieu, la composante radiale centripète rend visible le problème des occlusions : en principe, les point entre le but et la ligne d'amers, en particulier les points très près de l'amer central, ne devraient aboutir à aucun mouvement car l'animat est trop près de chacun des amers. Mais les occlusions partielles aboutissent à des mesures de largeur apparente trop petites, mettant en échec la méthode et les hypothèses qui garantissent une composante centripète positive. À droite, la composante tangentielle réussit là aussi, sans modification. Un exemple de trajectoire de l'animat est inclus.

Cette analyse nous permet de justifier le choix que nous avons fait dans la suite de notre travail, choix qui consiste à déterminer l'orientation de l'animat préalablement à l'application du modèle de navigation. Ceci permet de bénéficier de la performance de la composante tangentielle, et éventuellement de ses variantes (méthode moyenne par exemple), et évite les problèmes pratiques de la composante radiale que nous avons évoqués dans ce chapitre.

Nous n'évoquerons plus la composante radiale dans la suite, sauf dans une tentative d'estimer le gain théorique (rapport entre variation mesurée de la largeur apparente, et norme du vecteur individuel associé) en section 5.4.

3.7 Conclusion de l'analyse

Nous avons fait le choix, pour la suite, de donner à l'animat le moyen d'estimer son orientation, pour pouvoir utiliser un modèle de navigation qui utilise cette information et en tire une meilleure performance.

Le modèle PV a un rôle relativement central parmi les modèles étudiés. Par exemple, il est le dénominateur commun des modèles de Hong, de Weber et al., et le point de départ des modèles DV, Panama, Weber et al. Il peut être considéré comme le point de départ de notre modèle en ce qu'il rassemble proportionnalité des vecteurs individuels aux parallaxes, composante tangentielle et méthode directe.

Dans le chapitre suivant, nous passerons en revue quelques modèles qui prennent en compte le problème de l'orientation et proposent des solutions.

Chapitre 4

Navigation locale avec orientation

Dans le chapitre 2 nous avons étudié des modèles permettant le « retour au nid » d'un animat qui est supposé connaître son orientation. Nous avons, au cours de l'étude de ces modèles, mentionné quelles étapes introduisaient des dépendances dans l'orientation. Nous avons ensuite au chapitre 3 analysé l'importance de ces dépendances, pour conclure qu'il était plus intéressant de retrouver l'orientation de l'animat et avoir le choix des méthodes de navigation que de se restreindre à des méthodes de navigation moins performantes.

Dans ce chapitre nous passons en revue une seconde série de modèles qui, au contraire de la première série, sont conçus pour fonctionner dans les cas où l'animat tourne de façon notable d'un angle éventuellement inconnu.

4.1 Techniques d'orientation utilisées par les insectes

Puisque les modèles de navigation que nous étudions ici ont été inspirés de l'étude du comportement des abeilles, il se peut que la nature puisse aussi inspirer des solutions au problème d'orientation. Après tout, robot ou animal, tout ce qui bouge, perçoit et se déplace de façon autonome dans le monde réel, est confronté au problème de l'orientation.

Diverses études ont été conduites pour essayer de comprendre les solutions utilisées par les animaux. Nous nous contentons de résumer ici quelques pistes rapportées par Ruchti dans un travail dédié à l'orientation [RUC 00]. L'équipe de recherche concernée accorde une place importante à la plausibilité biologique de ses modèles.

Un animal peut conserver une orientation constante tout au long de son déplacement (par exemple vers le sud [COL 94]), ou bien mémoriser une vue avec l'orientation convenable attachée à cette vue [COL 97, FRI 96]. Dans un cas comme dans l'autre, une référence est nécessaire pour retrouver la bonne orientation. Des éléments absolus qui ont été envisagés sont le champ magnétique terrestre [COL 94] et l'utilisation de la polarisation de la lumière du ciel qui permet de retrouver la position du soleil même quand le ciel est presque partout couvert [WEH 96, LIN 96]. On peut aussi

avoir recours à des méthodes sans repère absolu. Certaines façons d’approcher la zone correspondant au but peuvent aussi amener à avoir toujours la bonne orientation, par exemple parce que les lieux sont toujours parcourus suivant la même séquence et qu’on arrive toujours à la même orientation dans la zone du but [COL 92].

Une autre façon de faire peut consister à prendre plusieurs vues dans plusieurs orientations et à mémoriser le tout. Au moment d’exploiter cette information mémorisée, on utilise la vue la plus ressemblante [NIC 99].

De notre côté, conformément à notre approche minimaliste, nous avons choisi de n’exploiter que les informations présentes dans les panoramas, à l’exclusion d’éléments extérieurs supplémentaires. Cela nous évite de dépendre de la présence d’éléments supplémentaires, et nous a conduit à utiliser 3 vues correctement orientées pour construire une référence de direction. Il n’y a pas beaucoup en commun avec les méthodes que nous venons brièvement d’évoquer.

4.2 Flux optique sur la sphère, Nelson & Aloimonos

Nelson et Aloimonos [NEL 88] ont étudié la détermination du mouvement propre instantané d’un mobile à partir de la vision depuis l’élément en mouvement, dans le cas où l’on dispose à la fois d’un champ de vision couvrant toute la sphère visuelle (ou presque) et du flux optique associé.

Nous devons justifier brièvement en quoi cela est relié au problème que nous nous posons. La détermination du mouvement propre instantané n’est, en effet, pas la même chose que la détermination d’une direction à suivre pour rejoindre un but. Les deux problèmes se ressemblent formellement, ce qui apparaît si on remarque que le mouvement propre instantané correspond au mouvement infinitésimal qui s’est produit entre t et $t + dt$. Le mouvement instantané trouvé, dans le modèle de Nelson et Aloimonos, est l’équivalent, dans le problème de retour au nid qui nous intéresse, du mouvement fini qui a pu se produire entre la vue du but (qui équivaut à l’instant t) et la vue actuelle (qui équivaut à l’instant $t + dt$). L’opposé du vecteur vitesse trouvé est l’équivalent de la direction à suivre pour rejoindre le but. Remarquons que le cas de mouvement infinitésimal fait apparaître des simplifications quand $t \rightarrow 0$: il n’y a plus de distinction entre azimuth mémorisé et azimuth vu, donc plus de méthode directe, inverse ou moyenne. Le mouvement propre entre t et $t + dt$ se confond avec le segment infinitésimal reliant les vues en ligne droite, ce qui équivaut au cas où le trajet en ligne droite est le meilleur et aucun obstacle n’obstrue le parcours.

Nelson et Aloimonos expliquent pourquoi un champ de vision limité est essentiellement ambigu pour ce qui est de la détermination du mouvement propre. En effet, sur un champ de vision limité, les effets d’une rotation propre et d’une translation (deux mouvements très différents) peuvent être extrêmement ressemblants, avec pour conséquence que le calcul du mouvement devient très sensible au bruit. Nous en donnons un exemple à l’aide de la figure 4.1. Notons que la solution choisie par la nature chez les mammifères est de ne pas s’appuyer sur la vision seule mais sur les informations données par les capteurs gravito-inertiels situés dans l’oreille interne. En revanche, la vision panoramique facilite beaucoup le problème, comme illustré en figure 4.2.



FIG. 4.1: Illustration de l'effet d'un champ de vision limité sur la difficulté d'estimer le mouvement propre. La photographie du haut est prise verticalement au-dessus d'une table. Pour celle d'en bas à gauche, on a tourné l'appareil sur place pour viser vers la droite. Pour celle d'en bas à droite, l'appareil est à nouveau vertical mais translaté vers la droite. Les différences entre les deux photographies sont faibles (on voit principalement que sur la photographie de gauche les lignes horizontales ne le sont plus tout à fait, surtout sur le bord droit qui apparaît plus petit dans la photographie de gauche que dans celle de droite). Globalement, cela signifie qu'estimer simultanément la position et l'orientation de l'appareil sera très sensible au bruit de mesure.

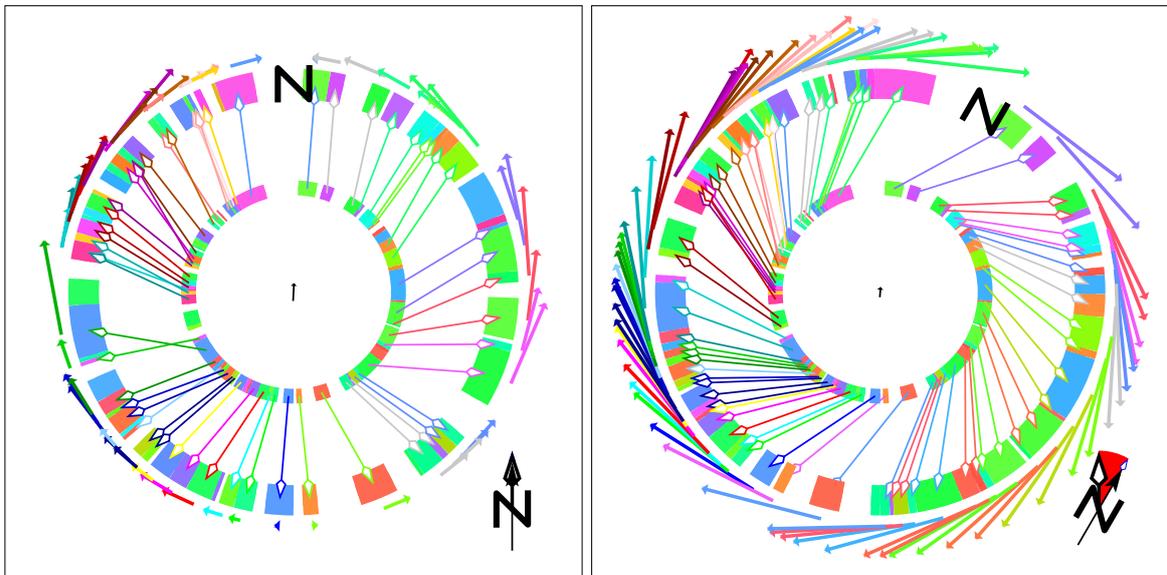


FIG. 4.2: Exemples de flux purement translationnel et purement rotationnel obtenus en simulation en appariant deux images prises en deux positions distinctes (ce n'est donc pas au sens strict un flux optique de type dérivée temporelle comme dans la méthode de Nelson et Aloimonos, mais la différence est faible). Dans le cas translationnel (cadre de gauche), le flux optique est séparable en deux demi-cercles, l'un où il est de sens trigonométrique, l'autre de sens contraire, séparés par le point d'expansion en bas et le point de contraction en haut. Ce dernier indique la direction à suivre pour revenir au panorama central. Les valeurs de flux sont variables et dépendent de la distance des objets environnant. Dans le cas rotationnel (cadre de droite), les valeurs de flux sont constantes tout autour du cercle. Un flux quelconque est la somme d'un flux translationnel et d'un flux rotationnel.

Nelson et Aloimonos expliquent alors comment ramener le problème du mouvement propre 3D à trois déterminations du mouvement propre dans le cas 2D. En effet, moyennant une étape de recombinaison, les résultats des trois déterminations 2D permettent d'obtenir le mouvement 3D.

4.2.1 Découpler pour réduire la complexité du problème

Le flux optique est un champ de vecteurs en chaque point d'une sphère. Chaque vecteur est tangent à la sphère. On n'essaie pas de trouver directement les 5 degrés de liberté du mouvement propre (3 de rotation et 2 de translations, car le vecteur vitesse 3D ne peut être estimé qu'à une homothétie près).

Au lieu de cela, on considère séparément trois plans équatoriaux orthogonaux deux à deux. Chaque plan intersecte la sphère suivant un cercle. Le flux optique le long de ce cercle n'est pas en général dans le plan du cercle (i.e. il a une composante orthogonale à ce plan). On considère la projection de ce flux optique dans le plan. On obtient un flux défini sur le cercle, chaque vecteur étant tangent au cercle. On peut appliquer à ce flux une méthode d'estimation du mouvement propre, mais à deux dimensions. On peut montrer que le mouvement propre obtenu (vecteur translation et vecteur rotation) est la projection du mouvement propre 3D sur le plan.

En appliquant l'analyse du mouvement propre à deux dimensions à chacun des trois plans orthogonaux, on obtient des estimations de mouvement propre qui sont les projections du mouvement propre 3D sur les plans. Le mouvement propre 3D peut être reconstitué à partir de ces projections 2D.

Nous choisissons de nous focaliser sur le cas à deux dimensions pour ne pas nous écartier du contexte dans lequel se placent tous les autres travaux.

La structure de la méthode 2D de Nelson et Aloimonos est assez proche de la structure commune aux autres méthodes étudiées dans cette thèse, de sorte que nous pouvons lui appliquer la grille d'analyse que nous avons définie en 2.1. La différence se situe dans les premières étapes (signature, appariement), car les hypothèses ne sont pas tout à fait les mêmes.

4.2.2 Signature d'un lieu et appariement

Avant découplage du cas 3D vers le cas 2D sur lequel nous avons choisi de nous focaliser, cette méthode est essentiellement fondée sur l'utilisation du flux optique sur la sphère visuelle.

L'article de Nelson et Aloimonos est relativement théorique et n'explique pas de méthode d'extraction de signature d'un lieu, ni d'appariement. Le modèle considère que le flux optique sur lequel il s'appuie est obtenu par une étape en amont qui lui est extérieure.

Par ailleurs, aucune implémentation dans le monde réel n'est évoquée dans [NEL 88]. Une telle implémentation aurait nécessité d'expliquer cette étape en amont.

Toutefois, nous pouvons esquisser comment serait implémentée l'obtention du flux optique dans une application réelle d'après le contexte. Les méthodes traditionnelles de calcul de flux optique utilisent souvent essentiellement des contrastes de luminance et leur déplacement d'une image à

l'autre [BEA 95]. Les contrastes de luminance jouent alors le rôle de signature d'un lieu, et la mise en correspondance entre les deux images est l'équivalent de l'étape d'appariement. Notons cependant que la notion de flux optique suppose deux images obtenues à des instants très proches, de sorte que les mouvements mesurés sont proches des dérivées temporelles théoriques. C'est une différence par rapport aux autres modèles étudiés dans cette thèse, pour lesquels les écarts sont des différences finies et non des dérivées.

En résumé, ce n'est pas cette étape qui nous apprend le plus sur les liens entre le problème que se posent Nelson et Aloimonos et le notre. Le modèle suppose donc dans la suite qu'un flux optique sur la sphère (dans le cas 3D) ou sur le cercle (cas 2D) est disponible.

4.2.3 Calcul d'orientation et direction à suivre

Cette méthode s'appuie principalement sur les caractéristiques d'un flux optique translationnel : il est soit nul, soit caractérisé par un point d'expansion et un point de contraction diamétralement opposés, et le mouvement de chaque côté du cercle va du premier vers le second (voir figure 4.2). En revanche, la composante rotationnelle du mouvement ajoute une constante au champ des vitesses, c'est-à-dire en chaque point un vecteur tangentiel au cercle de norme constante.

Dans ces conditions, trouver l'orientation actuelle se ramène à chercher quelle composante rotationnelle retirer du champ des vitesses pour obtenir un champ translationnel.

Pour ce faire, la méthode se donne une fonction de coût qui, pour une valeur donnée de l'orientation actuelle et de la direction de translation, estime en quoi le flux résiduel ne ressemble pas à un flux translationnel correspondant à ces deux paramètres. La fonction renvoie zéro si elle obtient bien un flux translationnel, et une valeur non nulle s'il y a des valeurs de flux optique dont le signe contredit les paramètres de mouvements testés. La valeur de cette fonction est proportionnelle aux causes de non-nullité.

La technique reste applicable même dans le cas où l'information de flux optique manque sur une portion de panorama, car cela ne perturbe pas forcément le calcul de la fonction de coût. Cette propriété est bienvenue, car disposer du flux optique sur une portion importante de la sphère visuelle n'est pas facile à obtenir avec les capteurs actuellement répandus. Il est toutefois important de disposer de plus d'un demi-tour de cercle, sinon il est impossible de trouver les deux points (expansion et contraction) et la méthode devient alors inapplicable en l'état.

4.2.4 Conclusion

De manière concrète et synthétique, on peut dire que la méthode de Nelson et Aloimonos permet de trouver les paramètres instantanés d'un mouvement (*vitesse* instantanée de rotation et *vitesse* instantanée de translation) dans le cas où l'on dispose de mesures de flux optique sur une grande portion de la sphère visuelle. Elle semble potentiellement adaptable à d'autres cas, comme la reconstitution du mouvement d'ensemble, à condition de disposer d'une séquence continue d'images intermédiaires.

Dans le cas où l'on souhaite faire naviguer un robot avec quelques vues mémorisées éventuellement éloignées les unes des autres (déplacement fini et non infinitésimal), cette méthode ne peut être utilisée telle quelle car le problème de l'appariement se pose alors plus fermement et n'est pas résolu. Le travail de Röfer que nous aborderons en section 4.6 complète ce modèle.

Enfin, bien que nous l'ayons conçue avant de connaître le modèle d'Aloimonos, notre méthode de calcul de la direction à suivre par élimination que nous présentons dans la prochaine partie a des éléments communs, tout en étant applicable aux mouvements finis et aux panoramas segmentés.

4.3 Fondu enchaîné, Chahl & Srinivasan

Chahl & Srinivasan [CHA 96] traitent le problème de l'orientation en introduisant une méthode d'estimation de translation ou de rotation de conception assez simple, applicable à un robot muni de deux capteurs panoramiques. La signature d'un lieu est la même que pour l'ALV : une fonction $f(\theta)$ qui associe à chaque direction un niveau de gris indiquant la luminosité mesurée dans cette direction.

Le principe de cette méthode est de disposer de deux panoramas $f_0(\theta)$ et $f_1(\theta)$ correspondant aux valeurs « extrêmes » de la grandeur à estimer, et d'un panorama $f(\theta)$ correspondant à la situation observée. On considère la fonction

$$\hat{f}(\theta) = (1 - \alpha)f_0(\theta) + \alpha f_1(\theta)$$

qu'on pourrait appeler un « fondu enchaîné » par analogie avec la technique de montage utilisée au cinéma ou en vidéo qui fait passer progressivement d'une image à une autre image en fonction du temps par une simple somme pondérée. De la même façon, le paramètre α fait passer de $f_0(\theta)$ à $f_1(\theta)$. Le principe fondamental de la méthode est qu'il existe une valeur de α qui minimise l'écart entre $\hat{f}(\theta)$ et $f(\theta)$.

Chahl et Srinivasan ont choisi ce qu'on appelle la norme 2, c'est-à-dire l'intégrale du carré de la différence entre fonctions, comme mesure de l'écart. La valeur de α cherchée est celle qui minimise

$$\int_0^{2\pi} [(1 - \alpha)f_0(\theta) + \alpha f_1(\theta) - f(\theta)]^2 d\theta$$

Elle se calcule par l'intégrale suivante :

$$\alpha = \frac{\int_0^{2\pi} [f(\theta) - f_0(\theta)][f_1(\theta) - f_0(\theta)] d\theta}{\int_0^{2\pi} [f_1(\theta) - f_0(\theta)]^2 d\theta}$$

D'après les auteurs, la valeur de α calculée par cette méthode diffère de 5% à 10% de la valeur idéale, dans des conditions réalistes.

4.3.1 Rotation

Un seul capteur est utilisé et on suppose que l'animat tourne autour du centre de ce capteur. On dispose d'un panorama avant rotation, et d'un panorama après rotation. On suppose que l'angle de rotation est compris entre -6° et 6° . On prend pour f_0 le panorama initial tourné de 6° à gauche, et pour f_1 le panorama initial tourné de 6° à droite. La valeur de α indique approximativement l'angle du panorama après rotation entre -6° ($\alpha = 0$) et 6° ($\alpha = 1$).

4.3.2 Translation

L'estimation de translation nécessite que le robot dispose de deux capteurs et que le vecteur translation soit parallèle au vecteur allant du premier au second capteur, de même sens et de norme inférieure.

On prend pour f_0 le panorama pris du premier capteur avant translation, et pour f_1 le panorama pris du second capteur toujours avant translation. La valeur de α indique approximativement la distance parcourue. Les valeurs extrêmes correspondent à l'immobilité ($\alpha = 0$) et à la distance entre les deux capteurs ($\alpha = 1$ correspond au cas où après la translation, le premier capteur se trouve à l'endroit où se trouvait le second).

4.3.3 Analyse de la méthode

4.3.4 Analyse théorique

Nous analysons avec plus de détail le cas de la rotation que celui de la translation car il est formellement plus simple.

Décomposons la fonction f en sa série de Fourier. On choisit la notation complexe car elle simplifie le calcul.

$$f_0(\theta) = \sum_{n=-\infty}^{\infty} A_n e^{in\theta}$$

On introduit β . On suppose que le robot a tourné d'un angle γ tel que $0 < \gamma < \beta$

$$f_1(\theta) = \sum_{n=0}^{\infty} A_n e^{in(\theta-\beta)}$$

La fonction qu'on cherche à minimiser est

$$\underbrace{(1 - \alpha)f_0(\theta) + \alpha f_1(\theta)}_{\hat{f}(\theta)} = \sum_{n=0}^{\infty} A_n e^{in(\theta-\gamma)}$$

Focalisons sur le terme associé à un n particulier. C'est :

$$(1 - \alpha)A_n e^{in\theta} + \alpha A_n e^{in(\theta-\beta)} - A_n e^{in(\theta-\gamma)}$$

On factorise $A_n e^{in\theta}$

$$A_n e^{in\theta} \left((1 - \alpha) + \alpha e^{in(-\beta)} - e^{in(-\gamma)} \right)$$

On factorise α

$$A_n e^{in\theta} \left(-\alpha(1 - e^{in(-\beta)}) + 1 - e^{in(-\gamma)} \right)$$

Si $\|n\beta\| \ll 2\pi$ alors $\|n\gamma\| \ll 2\pi$ aussi et on peut remplacer les exponentielles par leur développement limité au premier ordre.

$$A_n e^{in\theta} (-i\alpha\beta + i\gamma)$$

Lorsque $\alpha\beta = \gamma$, sous l'hypothèse $\|n\beta\| \ll 2\pi$, le $n^{\text{ème}}$ terme de la série de Fourier est nul.

f n'est pas le $n^{\text{ème}}$ terme mais la somme de tous. Or, dans un cas réel, l'essentiel de l'énergie du signal est dans les composantes de basses fréquences, et le bruit dans les hautes fréquences. Autrement dit, les A_n sont importants pour les n faibles. Ou encore, la valeur α qui minimise la fonction d'erreur est proche de celle qui annule les premiers termes, à savoir $\alpha = \gamma/\beta$ qui est bien la grandeur attendue (fraction de la rotation effectuée sur la rotation maximale attendue).

Dans le cas translationnel, on ne peut mener le même calcul mais on estime que les mêmes raisons fondamentales se maintiennent.

Cela explique pourquoi la méthode fonctionne, suggère pourquoi elle n'est applicable qu'aux petits angles, et suggère aussi qu'on obtiendra probablement une meilleure précision en appliquant un filtre passe-bas à toutes les fonctions avant d'appliquer la méthode.

4.3.5 Analyse critique

Cette méthode n'est pas une méthode complète d'estimation d'orientation. Elle ne peut pas donner l'orientation de l'animat ou sa position, ni donner le chemin à suivre depuis un point arbitraire. Elle peut seulement estimer pas à pas les déplacements effectués.

4.3.6 Conclusion

La nécessité de traiter seulement des mouvements de translation ou de rotation cantonne cette méthode à un substitut d'odométrie, avec les défauts qui en résultent (intégration au cours du temps

avec accumulation d'erreurs). Cette méthode ne fournit pas de moyen de retrouver son orientation ni de retourner à un point qui permettrait de recalculer l'information de mouvement propre.

4.4 Minimiser la somme des écarts, Möller

Möller a envisagé une solution pragmatique et simple par rapport à la question de l'orientation [MöL 00]. Partant du modèle PV, on peut faire successivement l'hypothèse que l'animat se trouve dans toutes les orientations possibles et calculer à chaque fois quelle est la direction à suivre et la somme des normes des vecteurs individuels. On considère que l'orientation véritable de l'animat est celle qui minimise la somme des normes des vecteurs individuels. Cette méthode rappelle la première solution envisagée par Röfer pour le calcul de l'orientation de l'animat que nous aborderons en section 4.6.3 même si elle n'est pas identique.

Cette méthode s'attaque au problème d'une façon conceptuellement simple, quoique nécessitant plus de calculs. Dans la lignée des méthodes CC, elle repose de façon pragmatique sur une observation qualitative. En l'occurrence, lorsque l'animat est correctement orienté les vecteurs individuels sont de signes variés et en général plus faibles que si on perturbe l'orientation.

Nous pensons que cette méthode peut fonctionner si on peut faire l'hypothèse d'une répartition relativement uniforme des amers autour de l'animat, mais qu'elle échoue probablement dans les cas très différents. En effet, nous avons vu en section 3.4.2 que la condition essentielle pour s'approcher du but n'est pas une orientation exacte en elle-même, mais suffisamment précise pour que les signes des parallaxes soient justes. Cette méthode, ajustant l'orientation pour que les signes des pseudo-parallaxes soient variés, peut donc obtenir de bons résultats si les parallaxes réelles sont effectivement variées, c'est à dire si des amers sont répartis tout autour de l'animat.

Par ailleurs, nous verrons que notre méthode de l'étoile polaire, lorsqu'elle est applicable, est plus intéressante que celle-ci : sans nécessiter de passer en revue toutes les orientations possibles (donc pour un coût moindre), elle fournit une estimation d'orientation assez précise pour naviguer. Dans le cas où la méthode de l'étoile polaire n'est pas avantageuse, nous pensons que le compas visuel est probablement plus performant que cette méthode.

4.5 Modèle de déformation, Franz

Franz présente un modèle appelé « modèle de déformation »¹ car son principe consiste en la recherche d'une déformation du panorama actuel qui le fait correspondre au panorama vu depuis le but [FRA 98b, FRA 97]. Malheureusement on ne peut calculer cette déformation exactement *a priori* car elle dépend des distances aux objets environnants, qui ne sont pas disponibles dans les signatures utilisées.

¹Nous traduisons par « modèle de déformation » le terme original de « Warping model ».

L'analyse proposée par Franz essaye donc de contourner le fait que les modèles CC ne disposent pas d'assez d'informations pour trouver exactement les inconnues (direction à suivre, distance au but, éventuellement orientation actuelle).

Il explicite les hypothèses que font implicitement les autres modèles de cette famille, notamment l'*hypothèse de répartition isotropique des amers* et l'*hypothèse de distances égales*. Ce sont des hypothèses implicites au sens où, si elles étaient vérifiées, les modèles produiraient des réponses exactes (au lieu de n'être que des approximations biaisées par les écarts à ces hypothèses).

Franz démontre que les modèles similaires à celui de Hong (composante tangentielle et méthode moyenne) commettent une erreur de moins de 90° dans l'estimation de la direction menant au but (sous réserve qu'il n'y a pas d'erreur d'appariement), ce qui implique que l'animat s'approche du but à chaque pas. Ce résultat est intéressant car il exclut en théorie que les modèles puissent aboutir à des comportements indésirables tels que tourner en rond ou s'éloigner. Mieux, si l'animat n'arrête pas sa course trop tôt, il se rapprochera du but jusqu'à l'atteindre effectivement. En théorie, il resterait un cas où l'animat décrirait une spirale. Mais en pratique plus l'animat est proche du but et plus il se dirige précisément vers lui.

4.5.1 Hypothèse de distance aux amers constante

La proposition de Franz, qui conditionne à la fois sa méthode d'appariement et sa méthode de calcul de la direction, consiste à contourner le manque d'informations en supposant explicitement que tous les amers sont à une distance égale de l'animat.

En faisant explicitement l'*hypothèse de distances égales*, on peut reconstruire artificiellement les déformations observées en fonction des paramètres restants (direction du déplacement et longueur, et composante rotation propre).

On peut voir ceci comme une adaptation de la méthode de Nelson et Aloimonos (voir section 4.2), simplifiée et adaptée au cas robotique où les panoramas ne sont pas très proches les uns des autres.

4.5.2 Hypothèse de répartition isotropique

Si les amers étaient répartis régulièrement autour de l'animat (*hypothèse de répartition isotropique des amers*) et à distance égale (*hypothèse de distances égales*), certaines méthodes simples donneraient une réponse exacte. Par exemple, la somme vectorielle de tous les éléments de flux optique pointerait exactement vers le but. Les méthodes fondées sur la composante tangentielle seraient exactes.

La méthode que propose Franz s'appuie plus précisément sur l'*hypothèse de distances égales*, ce qui lui permet de s'affranchir de l'*hypothèse de répartition isotropique des amers* comme nous allons le voir.

4.5.3 Signature d'un lieu

Le modèle de déformation n'introduit pas d'innovation à l'étape de signature d'un lieu. En simulation, la signature est la même que dans les modèles CC et Weber. Pour des expériences réalisées sur robot Khepera [FRA 97] équipé d'un capteur panoramique à miroir conique, un panorama à une dimension circulaire en niveaux de gris est utilisé.

4.5.4 Appariement, orientation et direction

L'étape suivante dans le modèle de déformation combine à la fois appariement, orientation et direction.

Ce modèle introduit une nouvelle méthode d'appariement. C'est la première méthode d'appariement non gloutonne introduite dans ce contexte, puisqu'elle tient compte d'emblée de l'ensemble du panorama pour effectuer l'appariement. Partant du panorama que l'animat avait mémorisé depuis le but, pour chaque valeur possible des paramètres recherchés (direction du déplacement et longueur, et composante rotation propre), on reconstitue artificiellement le panorama théoriquement observé. Parmi tous les jeux de paramètres, il en existe un qui reconstitue un panorama minimisant la distance (norme 2 sur les niveaux de gris) avec le panorama actuellement observé. C'est ce jeu de paramètres qui est conservé.

L'hypothèse de distances égales, en contraignant l'espace des déformations possibles, permet de s'affranchir de l'hypothèse de répartition isotropique, parce que la première hypothèse permet de reconstituer un panorama théorique unique pour chaque jeu de paramètres. La comparaison du panorama théorique et du panorama observé permet d'évaluer la ressemblance. Si un jeu de paramètres convient parfaitement (ou presque), alors les paramètres sont estimés. Aucune approximation n'est introduite, qui dépendrait de la seconde hypothèse pour devenir exacte.

Ce modèle est le plus « compact » dans le sens où il réalise dans la même étape appariement, orientation et calcul de direction. Son coût computationnel est assez élevé puisqu'il imbrique trois boucles pour estimer trois paramètres (orientation de l'animat, direction du déplacement, longueur du déplacement).

Dans [FRA 97], Franz décrit plusieurs approches pour accélérer la méthode. En effet, chaque paramètre qui pourrait être estimé séparément permettrait d'avoir une boucle imbriquée de moins et donc accélérer notablement la méthode. Premièrement, la longueur du déplacement peut être estimée par la valeur de la distance entre images (norme 2 sur les niveaux de gris). Cela fournit une valeur approximative qui aide à réduire l'espace de recherche. Deuxièmement, il a été tenté d'utiliser la méthode pour estimer l'orientation séparément. La conclusion a été que chercher à retrouver l'orientation séparément ne fonctionne pas bien sans avoir déjà une estimation précise de la distance. Enfin, il a aussi été essayé de ne chercher les valeurs optimales que dans un voisinage des valeurs des pas de temps précédents au lieu de parcourir tout l'espace des paramètres (à $t = 0$ il faut toujours appliquer la méthode complète). L'efficacité de la méthode ainsi améliorée lui permet de fonctionner en ligne.

4.5.5 Conclusion sur le modèle de déformation

Avec le modèle de déformation, Franz a proposé une formalisation du modèle CC qui s'appuie sur l'hypothèse que tous les amers environnants se trouvent à la même distance. S'appuyer sur cette hypothèse lui a permis de compenser le manque d'information que représentent des panoramas sans information de distance. Cette hypothèse n'étant pas destinée à être vérifiée strictement dans la pratique, elle devient une approximation dans les expériences. Cependant, elle permet de retrouver une estimation d'orientation et de direction à suivre. Franz présente des expériences sur robot Khepera qui valident expérimentalement le modèle.

Cette approche a le mérite de clarifier les raisons pour lesquelles les modèles CC fonctionnent, et les approximations qu'ils font. Toutefois, l'*hypothèse de distances égales* contraint beaucoup la forme de la mise en correspondance, ce qui empêche l'appariement d'être précis dans le détail. Malgré cette contrainte, l'espace des paramètres (orientation, direction du déplacement, distance) à chercher reste grand. L'approche de notre compas visuel, au contraire de l'approximation de distances égales, utilise implicitement les distances pour estimer l'orientation de l'animat, et peut s'avérer plus légère car elle ne nécessite pas de recherche dans un espace de paramètres.

4.6 L'algorithme Panama, Röfer

Röfer présente dans sa thèse des travaux apparentés aux nôtres [RöF 97b].

Les éléments que nous décrivons dans cette section correspondent à la première partie de sa thèse, où il présente plusieurs essais successifs de modèles d'estimation du mouvement et de l'orientation, avec à chaque fois des comparaisons statistiques en simulation. La suite de sa thèse présente des expériences sur un robot réel, un fauteuil roulant motorisé.

Les modèles de Röfer sont assez directement inspirés du travail de Nelson et Aloimonos que nous avons résumé en section 4.2.

4.6.1 Signature d'un lieu

Le modèle de Röfer utilise un capteur panoramique en couleur. La signature d'un lieu est composée de l'ensemble des niveaux de couleurs (rouge, vert et bleu) mesurés dans toutes les directions. Il n'y a pas de segmentation à ce niveau.

4.6.2 Appariement

Röfer introduit une méthode originale d'appariement inspirée par les réseaux auto-organisés de Kohonen [KOH 82, KOH 01].

Cette méthode réalise un appariement dense de l'environnement. À chacun des n pixels d'un des

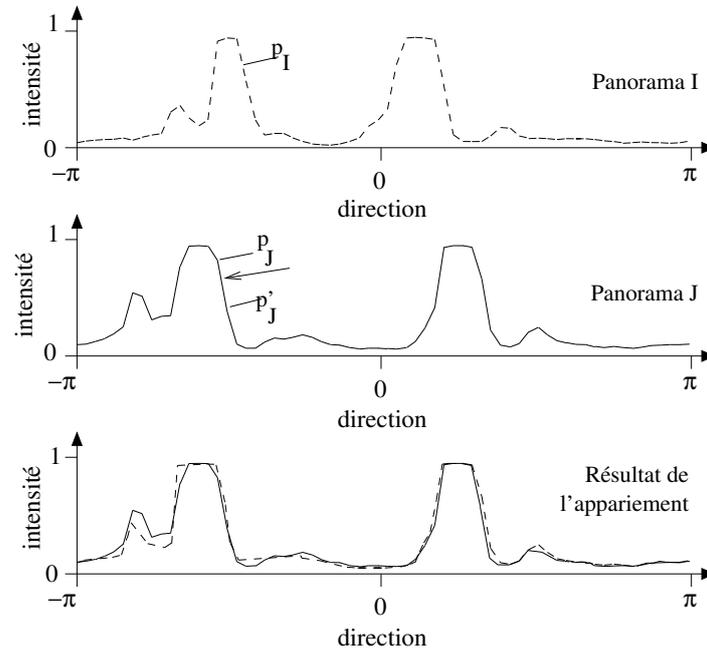


FIG. 4.3: La méthode d'appariement de Röfer permet d'associer à chaque point p_I du panorama I un point qui peut être interpolé entre deux points réellement mesurés p_J et p'_J du panorama J. (Figure adaptée de [RöF 97b].)

deux panoramas, nommons-le I , est associée une position entre deux pixels dans l'autre panorama, nommé J (figure 4.3).

Dans une première étape on calcule une double matrice de similarités entre pixels. La fonction de similarité fait intervenir les composantes de couleurs mais aussi leur dérivées. Röfer se donne une mesure permettant d'associer à un pixel donné de I un pixel de J ou bien une position entre deux pixels de J , de façon continue. La méthode employée considère le segment joignant deux pixels de J dans l'espace des paramètres associés aux pixels, et le point de ce segment le plus proche de I . Röfer n'explique pas pourquoi elle est bien fondée. En fait, elle revient à considérer que le panorama J est un ensemble discret de mesures d'une fonction qui interpolerait linéairement entre chaque point mesuré de J , et on choisit le point le plus proche de I à cette fonction et non au panorama J réellement mesuré. Cette façon de voir les choses permet d'assigner au mieux à des points de I des positions dans J de façon continue, apportant vraisemblablement une meilleure précision au processus d'appariement.

La matrice de similarités est double car elle contient, pour chaque pixel de J , la position continue entre deux pixels de I qui lui ressemble le plus ainsi que la mesure de ressemblance.

La seconde étape est celle inspirée de Kohonen, car elle met en jeu une mise en correspondance itérative de deux structures. On associe à chaque pixel de J une variable réelle représentant la position à laquelle il va être apparié dans I . Cette variable est initialisée à la fonction identité (panoramas I et J appariés sans déformation aucune). Puis, de façon itérative, on apparie un pixel de J (choisi dans une séquence qui couvre le panorama avec une distribution uniforme) avec la position la plus appropriée dans I en déformant tous les autres appariements voisins dans une fenêtre dont la largeur contient

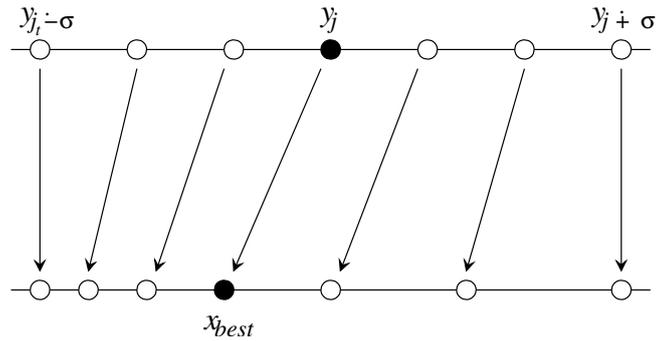


FIG. 4.4: La méthode d'appariement de Røfer est itérative et inspirée par les réseaux de Kohonen. À chaque itération, on applique une déformation dans une fenêtre, afin de mettre en correspondance un point y_j avec le meilleur candidat x_{best} . La largeur de la fenêtre est réduite au cours du temps, jusqu'à stabilisation de l'appariement trouvé. (Figure extraite de [RöF 97b].)

initialement tout le panorama mais se réduit pas à pas (figure 4.4).

Nous conservons dans la suite l'appellation de « flux optique » que Røfer emploie dans son travail. C'est un ensemble de nombres, chacun associé à un pixel de l'image panoramique unidimensionnelle de référence. Ce nombre est proportionnel au déplacement angulaire entre les deux points appariés, positif si le point est vu plus à gauche dans la vue actuelle que dans la vue de référence (convention trigonométrique), négatif dans le cas contraire et nul pour un couple de points en correspondance qui ont le même azimuth dans les deux vues.

La différence entre cette information et un flux optique au sens traditionnel du terme est que les deux images qui ont été appariées pour l'obtenir n'ont pas été prises juste l'une après l'autre mais peuvent correspondre à des vues prises à plusieurs mètres l'une de l'autre, de sorte qu'un grand intervalle de temps peut éventuellement les séparer.

À la fin de cette étape l'appariement est fait, mais il reste un traitement à effectuer. Røfer remarque qu'on ne peut pas encore calculer d'écarts de positions angulaires (le flux optique d'après la définition ci-dessus) à cause des effets de bouclages qui peuvent arriver si une partie des angles dépasse un demi-tour : un calcul naïf aboutirait à des discontinuités d'un demi-tour à certains endroits du flux optique, ce qui perturberait la suite des opérations.

Ce traitement consiste à parcourir le panorama. On compare l'écart d'angle associé à un point avec l'écart d'angle associé au suivant le long du panorama. Normalement la continuité est respectée, donc les deux écarts angulaires sont proches. On peut détecter un bouclage quand les deux écarts angulaires ne sont pas proches mais de signes opposés et chacun plus proche de 180° que de 0° . Alors, on ajoute ou retranche un tour complet pour restaurer la continuité en ce point. On parcourt ainsi tout le panorama.

4.6.3 Orientation

Röfer apporte des idées nouvelles, qui s'appuient sur le fait que le flux optique calculé est dense et régulièrement réparti autour de la vue de référence.

Le but d'ensemble est d'extraire trois paramètres :

- la composante rotationnelle du mouvement du robot ;
- la direction du vecteur déplacement du robot (dans le référentiel de la première vue) ;
- la norme de ce même vecteur.

Nous nous concentrons dans cette section sur l'extraction de la composante rotationnelle.

Röfer reprend le fait (que nous avons déjà illustré en figure 4.2 à l'occasion du modèle de Nelson et Aloimonos) qu'en théorie le flux optique dû à la translation seule est de signe constant à gauche et à droite de la ligne joignant point d'expansion et point de contraction ; et cette ligne est un diamètre du cercle représentant le panorama. À ce champ théorique s'ajoute une composante due à la composante rotationnelle du mouvement, égale à l'angle de la rotation et s'ajoutant sur tous les éléments.

La somme du flux dû à la translation est nulle lorsque l'*hypothèse de distances égales* (cf section 4.5) est vérifiée. La première proposition de Röfer est de calculer la moyenne du flux observé et de retrancher cette moyenne de toutes les valeurs. Malheureusement ceci n'est qu'approximatif car, même sans rotation, les écarts d'un côté et de l'autre peuvent être très différents. Par exemple, si l'animal longe un mur proche situé d'un côté de son champ de vision, alors que l'espace est ouvert de l'autre côté, le flux sera bien plus fort du côté du mur.

Röfer introduit une remarque qualitative intéressante. Même si le flux translationnel est plus important d'un côté que de l'autre, il reste vrai qu'il y a dans le flux translationnel $\frac{n}{2}$ valeurs positives et $\frac{n}{2}$ valeurs négatives. La méthode, illustrée en figure 4.5, consiste donc à trier les écarts pour séparer la moitié des plus grands de la moitié des plus petits. Ensuite, il suffit de retrancher une valeur séparant ces deux ensembles pour retrouver un flux optique translationnel.

Cette dernière méthode semble bien fondée et donner le bon résultat. Röfer rapporte en effet une bonne précision dans des expériences simulées comme réelles. Les simulations qu'il décrit ont une erreur maximale de 4.5° , et en général meilleure (souvent de l'ordre de 1°).

Remarquons que cette précision s'appuie sur la densité de l'appariement obtenu. Si l'environnement ne contient pas assez d'éléments susceptibles de contribuer à un bon appariement, la précision peut s'en trouver dégradée. En fait, cet effet peut être prédit en le comparant à la solution trouvée par Nelson et Aloimonos. Ces derniers trouvent l'orientation en minimisant une fonction mesurant les éléments faisant du flux optique mesuré un flux non purement translationnel. Pour trouver un flux satisfaisant il faut descendre un gradient, donc calculer la valeur de la fonction de coût plusieurs fois. C'est plus coûteux en calcul mais la méthode est applicable même si une partie des informations manque ou est irrégulièrement répartie. En revanche, la méthode de Röfer qui trie les valeurs disponibles est un peu plus légère (si on utilise un algorithme de tri performant) mais elle suppose que toutes les valeurs sont disponibles et pertinentes. Nous y reviendrons au moment de comparer notre méthode de calcul d'orientation avec celle de Röfer en section 10.1.7.3.

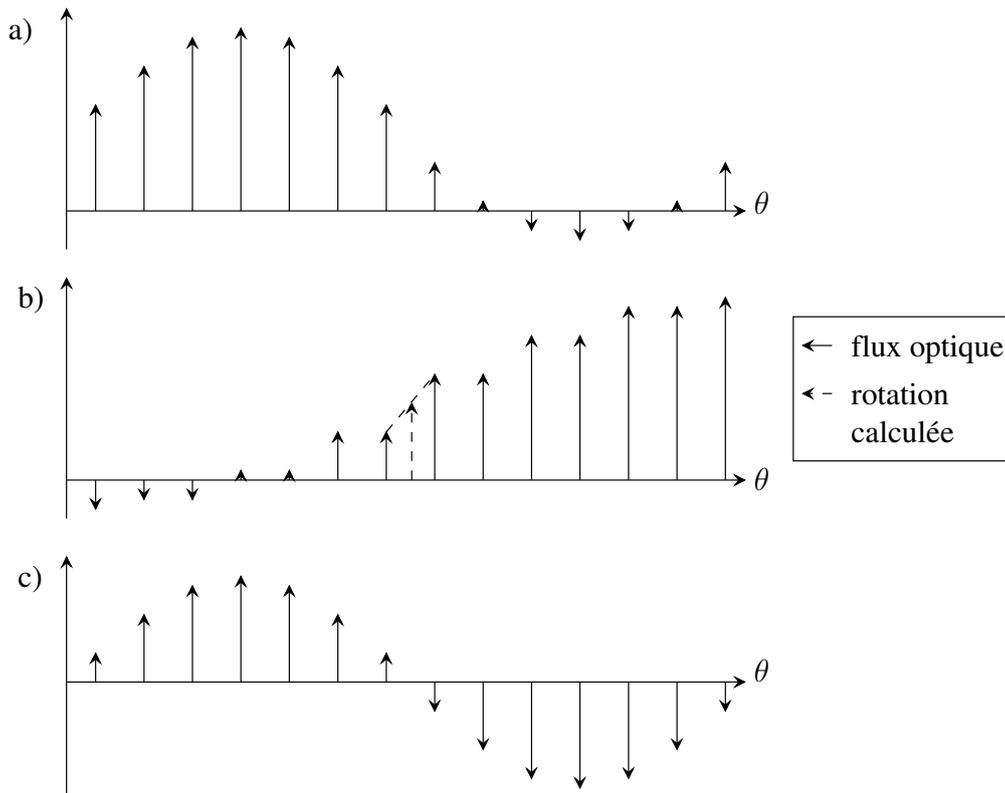


FIG. 4.5: Méthode 2 de Röfer pour extraire la composante rotationnelle du flux optique. a) Le flux optique trouvé, déroulé suivant l'axe des abscisses. b) Puisque le champ translationnel seul contient autant de points de flux dans un sens que de flux dans l'autre sens, il suffit de trier les valeurs du champ composite et repérer une des valeurs séparant le jeu trié en deux sous-ensembles de même nombre d'éléments. C'est la rotation cherchée. c) Après avoir retiré la rotation à tous les éléments du flux initial, on obtient le flux translationnel restant. (Figure adaptée de [RöF 97b].)

4.6.4 Direction à suivre

Une fois obtenu le flux optique dû à la translation, on cherche les points d'expansion et de contraction qui sont normalement opposés.

Röfer introduit là aussi des propositions intéressantes, qui s'appuient sur la disponibilité d'un appariement dense et précis.

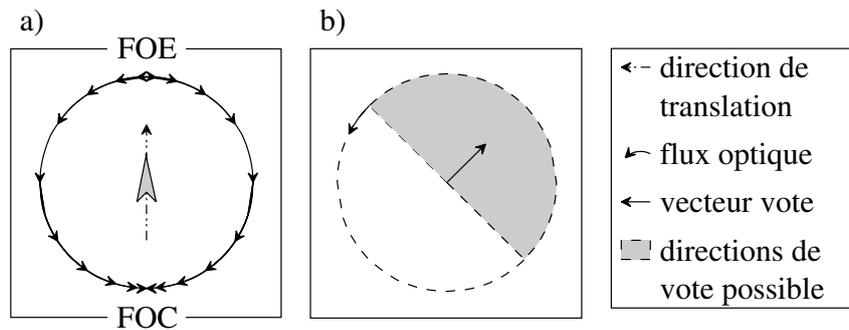


FIG. 4.6: Méthode 1 de Röfer pour extraire la direction à suivre. a) Le flux translationnel, dont on cherche le point d'expansion (FOE) et le point de contraction (FOC). b) Selon sa direction, le sens de chaque flèche du flux optique « vote » sur le côté où se trouve le point d'expansion. Il suffit, si les sens des flèches sont justes, d'associer un vecteur vote à chaque élément de flux. Alors la somme vectorielle des votes pointe exactement vers le point d'expansion. En effet, parce que le flux optique est uniformément dense et tous les vecteurs votes sont unitaires, les écarts se compensent par symétrie. (Figure adaptée de [RöF 97b].)

Röfer expose une première méthode qui consiste à considérer que chaque élément de flux optique est compatible, selon sa direction, avec l'hypothèse que le point d'expansion est dans le demi-plan à gauche ou à droite du pixel considéré. On considère, comme pour les modèles CC, un vecteur unitaire représentant cette information comme un vote (voir figure 4.6). La somme de ces votes est supposée être la direction du point d'expansion. La direction à suivre pour revenir au but est à l'opposé, c'est la direction du point de contraction. Remarquons que ce modèle est proche des modèles CC parce qu'il réalise une somme vectorielle. Cependant il a une particularité nouvelle et importante : la disponibilité d'informations de flux optique sur tout le cercle permet en principe une précision parfaite, si le flux optique est juste.

Malheureusement le sens du flux optique réellement obtenu en amont n'est pas juste en chaque point, surtout sur les petits vecteurs. La première méthode n'est donc pas exacte en pratique. Röfer propose alors une seconde méthode qui ressemble au modèle PV puisqu'on y introduit la proportionnalité de chaque vecteur à la cause qui lui donne naissance. Malheureusement, ce changement rend caducs les arguments qui rendaient la première méthode potentiellement exacte (la symétrie ne compense plus les écarts). On peut lui prédire les mêmes travers que la méthode PV. Et les expériences en simulation ([RöF 97b] chapitre 1) le confirment.

Röfer considère alors une troisième méthode intéressante (puis une quatrième, mais qui n'est qu'un petit raffinement par-dessus la troisième). On calcule la somme des flux optiques sur un demi-

cercle. La position du demi-cercle telle que la somme est maximale est exactement en face de la position telle que la somme est minimale, et la limite entre ces deux demi-cercles opposés est en principe constituée du point d'expansion et du point de contraction.

Il est intéressant de noter que Röfer rapporte à l'aide d'expériences en simulation que ces méthodes, sauf la première, semblent compenser l'erreur sur l'orientation du robot. C'est-à-dire qu'avec ces solutions, si le robot a mal estimé son orientation, l'estimation de la direction vers le but n'en est pas très perturbée.

Röfer présente aussi une méthode pour calculer la « distance » parcourue, tout en précisant qu'elle ne constitue pas vraiment une mesure de distance, mais plutôt une indication qui permet d'estimer par exemple si un panorama A est plus proche de B que de C.

La conclusion des expériences de simulation est qu'en général, le meilleur choix est la solution 2 pour le calcul de l'orientation et la solution 1 pour le calcul de la direction à suivre.

4.6.5 Conclusion

La thèse de Röfer introduit des méthodes intéressantes car bien fondées et appuyées par des expériences.

On regrettera cependant que les expériences simulées dans le chapitre comparant les diverses méthodes concernent un environnement un peu simple et prennent ainsi le risque de ne pas être fidèles à la réalité. On remarquera que l'analyse et la discussion sont fondées sur des statistiques « analytiques » et non « synthétiques » (cf. 2.1.6.7).

Quoiqu'il en soit, il apparaît que si l'environnement est assez riche en détails et si le système visuel est capable de les apparier, alors le modèle de Röfer peut être précis.

Nous estimons cependant qu'il est plus intéressant de partir de panoramas segmentés car ils sont plus légers à traiter et peuvent donner de bons résultats. Les méthodes expérimentées par Röfer pour calculer l'orientation et la direction à suivre s'appuient généralement sur la densité et la régularité des appariements entre panoramas, propriété que n'ont pas des appariements de panoramas segmentés. Ces méthodes ne peuvent donc être reprises telles quelles dans notre cas.

Toutefois, la méthode de calcul de la direction à suivre par élimination, que nous introduisons en 5.3 peut être considérée comme apparentée à la première méthode de Röfer pour calculer la direction à suivre, mais applicable à des panoramas segmentés.

De plus, Röfer s'est donné pour critère d'arrêt d'une trajectoire de l'animat le fait que le mouvement commandé en un point est proche de la direction opposée au mouvement précédent, idée dont nous nous sommes inspiré en 5.5.

4.7 Parcourir les lignes iso-ALV, Ruchti

Ruchti a réalisé des travaux en simulation dans la lignée des études sur les abeilles et les fourmis, notamment celles de Möller et Lambrinos sur les fourmis du désert [MöL 98b], en approfondissant la question de l'orientation [RUC 00],

Il présente une analyse systématique des étapes nécessaires pour des modèles de la famille CC et des variantes. Il introduit des notions pratiques comme celles d'*agent directionnel* et d'*agent aligné*. Un agent aligné se déplace en faisant en permanence face à sa direction d'avancement. Une voiture automobile pilotée par une caméra rigidement fixée est un agent aligné. En revanche, un agent directionnel conserve une direction de son système visuel (ou de son corps entier) constante par rapport à un référentiel extérieur. Un char d'assaut, qui maintient sa tourelle dans une direction constante tout en faisant varier la direction de la base et donc du déplacement, est un agent directionnel.

Ruchti expérimente des variantes des modèles CC, dont la plupart ont déjà été citées et analysées dans le chapitre 2. Il analyse et discute l'utilité de ces variantes et des façons de compenser des écarts ou l'absence de boussole. Ruchti a observé comme nous que la composante radiale n'introduit pas de dépendance en orientation. Ce travail est intéressant par son aspect analytique et systématique.

Il propose une méthode fondée sur le modèle ALV pour retrouver le but malgré une orientation inconnue. Le principe est que, même en cas de désorientation totale, l'animat peut tenter d'appliquer le modèle ALV, qui produit tout de même un vecteur. La norme du vecteur obtenu est correcte dans tous les cas, mais son orientation ne l'est plus forcément. Ruchti considère alors l'ensemble des lieux où l'ALV a la même norme qu'au but [RUC 00]. C'est une ligne. En suivant cette ligne on peut finir par retrouver le but.

Malheureusement, il n'est pas évident de suivre une ligne iso-ALV. La norme de l'ALV dépend de tous les points du panoramas d'une façon que l'animat ne connaît pas. Il doit donc procéder par essais et erreurs dans des directions variées pour trouver des directions qui conservent la norme de l'ALV. Le bruit est particulièrement gênant. Ensuite, même en admettant que suivre la ligne est une opération viable, il reste deux sens de parcours. Or l'animat ne dispose d'aucun élément indiquant dans quelle sens il doit la suivre, n'ayant mémorisé que l'ALV du lieu d'arrivée.

Cette méthode contourne le problème plutôt qu'elle ne le résout, dans la mesure où l'animat évite d'avoir à connaître son orientation. Malheureusement ce contournement a des répercussions importantes sur la façon de naviguer et les performances qu'on peut en attendre. Elle n'est donc pas satisfaisante.

De façon générale, les expériences de Ruchti, bien qu'intéressantes, restent dans une modélisation du monde où les amers ne sont pas distinguables les uns des autres, avec les limitations que nous avons déjà évoquées dans la première revue (section 2.5.2).

4.8 Reconstitution 3D, Benosman et al.

Benosman et son équipe ont adapté et appliqué à un capteur panoramique les méthodes de vision par ordinateur développées initialement par Faugeras [FAU 93]. Il s'agit ici de reconstituer à partir de deux images à la fois le déplacement du capteur dans l'espace (translation et rotation) et la position en trois dimensions de l'ensemble des points d'intérêts extraits des deux images.

Cette technique appartient à un cadre différent de celui de l'approche animat. Son but est de reconstituer l'environnement précisément, et non juste d'extraire les informations nécessaires à la survie d'un animat. Cependant, étant donnée la similitude de structure avec les autres méthodes, nous pouvons lui appliquer notre grille d'analyse (cf. 2.1).

4.8.1 Signature d'un lieu

La signature d'un lieu, dans ce cas, est une image prise par un capteur panoramique (voire figure 4.7). Elle peut être en noir et blanc ou en couleurs.

En premier, une méthode d'extraction de points d'intérêts est appliquée. Pour les applications réelles il s'agit d'une recherche de coins dans l'image, mais pour les expériences internes ont été présentés des disques noirs dont les centres sont extraits.

4.8.2 Appariement

Autour de chaque point d'intérêt est extrait un voisinage en niveaux de gris, en forme de losange dont les diagonales sont parallèles aux axes d'azimut et d'élévation 4.8. Une nouveauté est apportée par l'adaptation des fenêtres de voisinage en fonction de la position dans le champ de vision.

Pour comparer deux voisinages on ajuste leur géométrie pour compenser les déformations dues à la courbure du miroir qui, sinon, perturberaient la mise en correspondance. On calcule leur similarité par une corrélation croisée centrée et normalisée.

On apparie les points de meilleure similarité et on élimine après coup les appariement entre deux amers dont l'écart d'angle est loin de la moyenne.

4.8.3 Calcul de direction

4.8.3.1 Calibration préalable

La méthode nécessite en amont une calibration du capteur panoramique utilisé, illustrée en figure 4.9. À l'issue de cette procédure, on peut associer à chaque pixel de l'image panoramique une direction précise dans l'espace [FAB 02].

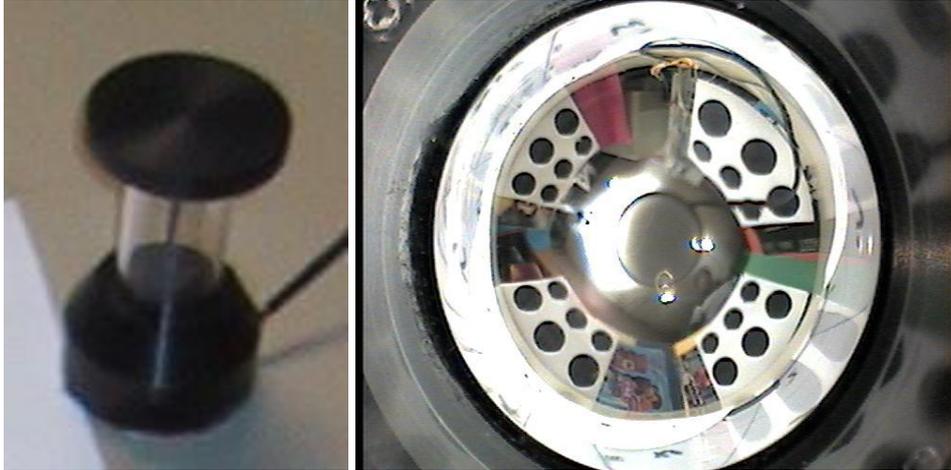


FIG. 4.7: À gauche, un des capteurs panoramiques utilisé par Benosman et al. Son diamètre est de quelques centimètres. À droite, un exemple d'image qu'il permet d'obtenir. On y voit des disques noirs affichés dans l'environnement qui servent de points d'intérêt pour certaines expériences.

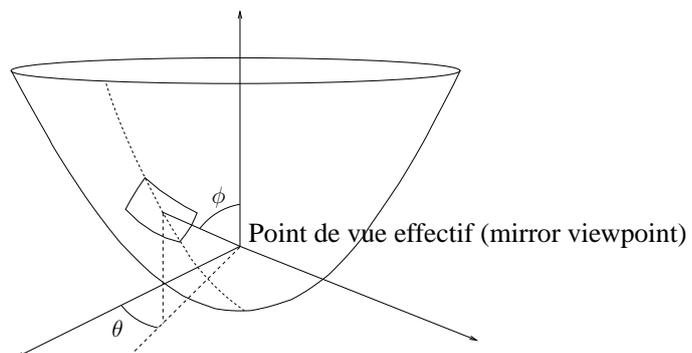


FIG. 4.8: Pour apparier des points entre deux images panoramiques, la méthode de Benosman utilise des fenêtres en forme de losange définies sur la surface du miroir en terme d'azimut et d'élévation. La taille de la fenêtre dépend de sa position pour compenser les déformations dues au miroir.

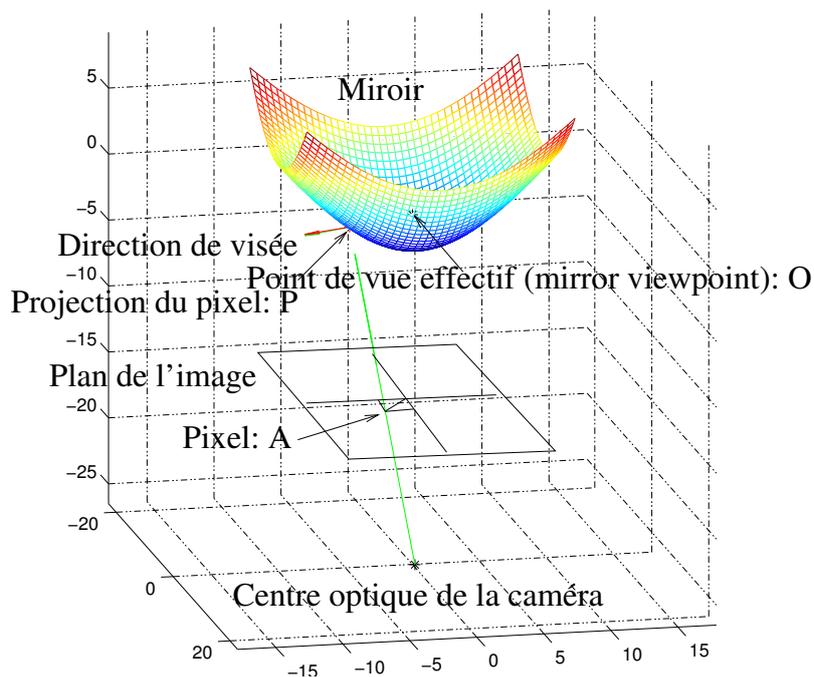


FIG. 4.9: La calibration du capteur panoramique permet d'associer à chaque pixel de l'image panoramique mesurée une direction précise dans le repère du capteur. Cette étape est nécessaire pour permettre une reconstitution 3D avec des rotations arbitraires.

4.8.3.2 Estimation du déplacement 3D

Pour chaque point apparié entre les deux vues, on exprime la contrainte épipolaire, qui est illustrée en figure 4.10. A priori la distance à un point fixe de l'environnement est inconnue dans chacune des deux vues. Mais l'appariement permet de connaître le pixel associé à cet élément de l'environnement dans chacune des deux vues. La géométrie du capteur étant calibrée, on connaît donc dans chacune des deux vues la direction dans l'espace (relative au capteur) qui joint le capteur à l'élément fixe de la scène. Cette connaissance permet d'exprimer une équation linéaire faisant intervenir la position du point dans l'espace et le mouvement du capteur entre les deux vues. Les inconnues sont les coefficients d'une matrice représentant le déplacement du capteur. Pour des raisons de symétrie, le nombre d'inconnues se limite à 7. Cela signifie que de façon théorique, il suffit d'avoir 7 points correctement appariés dans l'environnement pour pouvoir reconstituer les composantes translation et rotation du capteur d'une vue à l'autre. En pratique quelques dizaines à quelques centaines de points peuvent être nécessaires et on peut obtenir une assez grande précision.

Nous désignerons par la suite le modèle de Benosman et al. par l'appellation « méthode calibrée », en référence à la calibration du miroir et à la recherche de l'exactitude, aux dépens des calculs nécessaires.

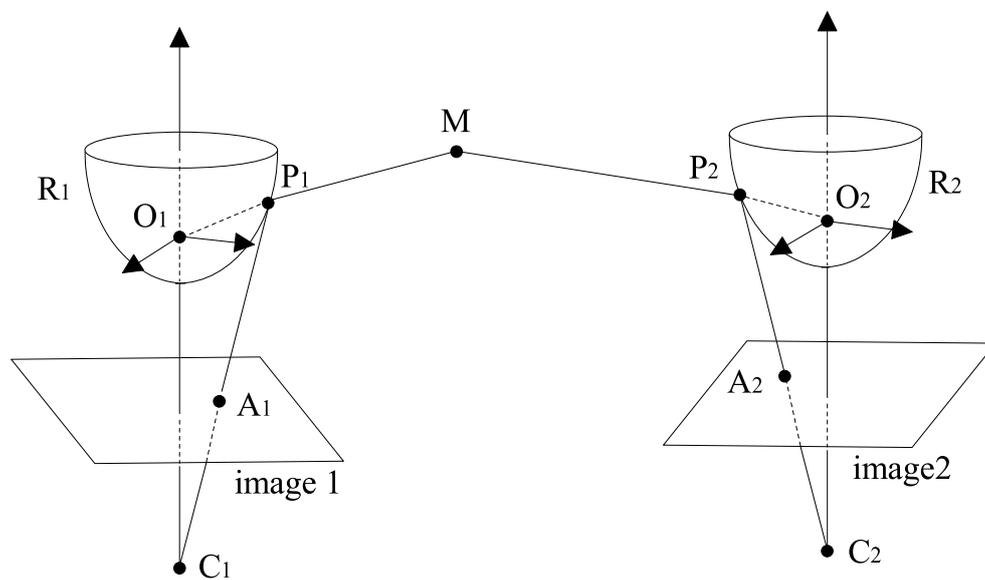


FIG. 4.10: Reconstitution 3D d'une scène à partir de deux vues panoramiques. Après appariement d'un point M de l'espace vu dans les deux images panoramiques, on peut extraire une contrainte reliant la position de M et des deux vues sous la forme d'une équation linéaire. En théorie, 7 de ces points sont suffisants pour reconstituer la position du référentiel R_2 de la seconde vue en fonction du référentiel R_1 de la première vue. En pratique il en faut un plus grand nombre et des méthodes d'élimination des points aberrants pour obtenir un résultat final qui peut être très précis.

4.9 Conclusion

Dans ce chapitre, nous avons passé en revue une seconde série de modèles qui, contrairement à ceux de la première série (chapitre 2), ont en commun de ne pas dépendre d'une estimation d'orientation extérieure, voire d'en fournir une.

Nous retiendrons particulièrement les innovations présentées dans les modèles de Nelson et Aloimonos, de Franz et de Röfer, qui ont des relations avec la technique de calcul de direction par élimination que nous allons introduire au chapitre suivant. Toutefois, le premier n'est pas applicable tel quel à la robotique par manque d'un capteur puis d'une méthode d'appariement. Les deux autres sont intéressants mais relativement coûteux en calculs. Nous ne les reprendrons pas directement car notre approche, plus légère, consiste à estimer seulement l'orientation dans un premier temps, pour pouvoir utiliser les modèles légers du chapitre 2 ensuite.

Nous ne reprendrons pas d'éléments du modèle de fondu enchaîné de Chahl & Srinivasan (équivalent à de l'odométrie avec les problèmes d'intégration), du modèle exhaustif de Möller (nous pensons faire mieux pour moins de calculs), ni du parcours de lignes iso-ALV de Ruchti (ajoute aux limitations de l'ALV la difficulté de parcourir une ligne virtuelle sans connaître sa direction). La méthode de Benosman et al est très différente et beaucoup plus coûteuse en calculs. Elle sera pour nous une occasion de comparaison de performances.

Tous les modèles majeurs traitant l'orientation étudiés dans ce chapitre le font avec des panoramas échantillonnés. Au contraire, l'utilisation d'appariements segmentés a l'avantage d'alléger à la fois la mémoire et la quantité de calculs nécessaires au traitement.

Dans la prochaine partie de cette thèse, nous allons présenter notre modèle, qui réalise la navigation locale en utilisant les propriétés des panoramas segmentés.

Deuxième partie

Notre modèle

Notre modèle

Dans cette partie de la thèse nous allons décrire en détail le modèle que nous avons développé.

Notre modèle s'attaque au problème de la navigation locale par panorama. Il reprend pour fondement le modèle PV, c'est-à-dire essentiellement l'usage de la composante tangentielle (exploitation des parallaxes apparaissant au cours du déplacement) avec la méthode directe (application des vecteurs individuels sur la position actuellement vue).

Il apporte des innovations à plusieurs étapes :

- l'utilisation de panoramas segmentés en couleurs (dans l'espace de teinte et saturation) ;
- une méthode d'appariement utilisant un algorithme de programmation dynamique, qui présente la particularité d'exploiter les éléments distinctifs des amers (ici, la couleur) pour éviter les confusions, ce qui lui permet d'être robuste vis-à-vis de toute orientation de l'animat ;
- une méthode de calcul de la direction à suivre dite « par élimination » qui s'appuie sur des panoramas segmentés appariés et sur la connaissance de l'orientation de ces panoramas ;
- une méthode de calcul de l'orientation, dite « compas visuel », qui s'appuie sur un ou plusieurs triplets de panoramas d'orientation connue et mémorisés par l'animat, pour retrouver l'orientation d'un nouveau panorama pris dans un voisinage ;
- une méthode d'exploration et de construction de carte topologique qui sert de support au compas visuel en lui fournissant des panoramas appropriés.

Nous reprenons dans cette partie une structure similaire à celle de la partie précédente.

Dans le chapitre 5 nous présentons en détail comment nous améliorons les méthodes existantes pour le sous-problème de la navigation avec orientation connue.

Dans le chapitre 6 nous décrivons en détail le compas visuel, qui permet à l'animat de retrouver une estimation de son orientation après avoir préalablement « pris ses marques » lors d'une première exploration.

Nous finirons cette partie par le chapitre 7 en exposant une procédure de construction de carte topologique qui sert de support au compas visuel. En effet, la navigation locale, comme l'orientation locale, ne font que rejoindre un but désigné d'une façon extérieure, à l'aide de marques établies à partir de vues désignées par l'expérimentateur. Un pas suivant dans l'autonomie est la capacité pour l'animat de choisir lui-même au cours d'une exploration les vues qui lui permettent de prendre ses marques, un pas qui est franchi quand l'animat est capable de construire sa propre carte de l'environnement.

Chapitre 5

Notre modèle, lorsque l'orientation est connue

Ce chapitre concerne le sous-problème de la navigation locale dans lequel l'orientation de l'animat est connue.

Pour la clarté de l'exposé, nous appliquons à la description de notre modèle la même grille d'analyse (section 2.1) que lors de la description des modèles étudiés dans la première partie de cette thèse.

Pour chacune de ces étapes nous expliquerons ce qu'elle apporte et la comparerons aux méthodes précédentes.

Nous commençons par l'étape de signature d'un lieu : dans notre modèle, l'animat mémorise de son environnement des panoramas segmentés en couleurs.

5.1 Signature d'un lieu : segmenter le panorama en couleurs

La première étape dans notre grille d'analyse est le choix de ce que l'animat retient du monde qui l'entoure.

Dans les modèles CC, on schématise l'environnement comme un fond clair (comme le ciel) sur lequel des objets sombres se détachent. Cette modélisation, qui a été conservée dans les modèles DV et ALV, n'est pas viable, parce que trop simpliste, pour un robot se déplaçant en environnement intérieur ou même urbain.

Il est donc nécessaire d'enrichir l'information perçue.

Notre choix a été guidé par les éléments techniques qui nous étaient accessibles. Dans un premier temps nous ne disposions pas d'un capteur panoramique, mais de robots équipés chacun d'une caméra motorisée pouvant tourner sur 200°.

5.1.1 Utiliser la couleur

Intuitivement, l'utilisation de secteurs noirs et blancs limite considérablement la perception, avec pour résultat la confusion possible entre des éléments de l'environnement qui pourraient être facilement distingués en utilisant une perception plus riche. Comme nous le verrons par la suite, cet enrichissement de la perception est déterminant pour permettre à l'animat de retrouver son orientation.

Nous avons eu l'opportunité d'utiliser un outil qui a facilité l'exploitation de l'information visuelle et que nous décrirons plus en détails en section 9.1.3. Il s'agit d'une puce de perception visuelle nommée GVPP [PIR 04] qui analyse en temps réel un flux vidéo et s'avère notamment capable de fournir des mesures de couleurs. Nous avons donc exploité l'information de couleur fournie par GVPP. Cette information est codée dans l'espace luminance, teinte, saturation. Cette représentation est avantageuse car la teinte mesurée d'un même objet est souvent bien uniforme malgré la non-uniformité de l'éclairage, ce qui facilite la segmentation. Nous reviendrons sur ce point lors de la comparaison avec le modèle de Röfer en section 10.1.7.

5.1.2 Maintenir le découpage en éléments distincts

Les signatures utilisées par presque tous les modèles étudiés dans ce travail segmentent l'image perçue en secteurs. À l'opposé, les modèles de déformation et Panama n'opèrent pas de segmentation explicite.

Nous avons choisi de conserver un découpage du panorama en objets distincts. Cette étape de segmentation filtre l'information en amont, en réduisant la quantité d'information à traiter par la suite.

Un bon filtrage ne doit sélectionner que ce qui est pertinent pour minimiser l'effort computationnel en aval, mais conserver suffisamment d'informations pertinentes pour arriver à satisfaire la tâche visée.

Nous avons utilisé deux algorithmes différents de segmentation, que nous décrirons dans le chapitre dédié aux expériences réelles : un algorithme « naïf » – qui compare à un seuil constant la variation instantanée de la teinte le long du panorama (section 9.1.4) – et un autre – qui travaille dans l'espace YCrCb et compare la variation instantanée de l'information de chrominance à l'écart-type de cette même variable sur les mesures précédentes (section 9.4.3).

5.2 Appariement par programmation dynamique

Pour pouvoir estimer les parallaxes entre signature mémorisée et panorama actuel, nous devons choisir une méthode d'appariement. L'utilisation des parallaxes a essentiellement servi jusqu'ici au calcul de direction à suivre lors du retour au nid.

Nous verrons par la suite que cette étape peut servir à extraire des informations pour d'autres fonctions de plus haut niveau : reconnaissance d'un lieu (de façon similaire à [GAU 00a]), ou estimation

d'orientation comme nous le verrons au chapitre 6.

5.2.1 Limites de l'appariement originel

L'appariement d'origine utilisé par les modèles CC est très fruste. Il était cependant suffisant dans son contexte qui était celui de valider le principe de la navigation par panorama et de comparer son comportement à celui d'abeilles réelles.

Cependant l'utilisation dans notre contexte, faire naviguer un robot réel dans un environnement moins contrasté, fait apparaître les limitations suivantes :

1. algorithme glouton, qui apparie sans tenir compte de l'ensemble de la situation et sans remise en question ;
2. ne tient pas compte des propriétés intrinsèques des amers dans son choix, seulement de leur position visuelle (la pauvreté de la signature dans les modèles CC ne le permet pas) ;
3. suppose implicitement une même orientation des deux panoramas, et dépend effectivement de cette hypothèse ;
4. suppose un champ de vision complet, et dépend effectivement de cette hypothèse.

Les travaux de Weber et al. (section 2.5) apportent des améliorations sur le premier point, mais laissent les autres inchangés. Le modèle de Hong (section 2.3.1) tient compte de l'ensemble quand il sélectionne les 15 points de plus fort contraste.

L'appariement du modèle Panama est intéressant. Il résout bien le point 1. Bien qu'il n'individualise pas les éléments de l'environnement, il répond au point 2 dans la mesure où il exploite la couleur. Il répond complètement au point 3. Le point 4 n'est pas abordé par les modèles précédemment étudiés ; tous supposent le champ de vision complet, sauf celui de Bianco et le modèle de Gaussier.

5.2.2 Cahier des charges

Pour cette étape nous avons besoin d'un algorithme capable d'apparier deux panoramas partiels ou complets comportant des informations de couleurs, et qui potentiellement ne sont pas orientés de la même façon.

Notre contexte est celui d'un robot réel dans un environnement qu'on souhaite le moins préparé possible. Nous cherchons à mettre au point un algorithme capable de satisfaire les éléments suivants :

1. tenir compte de l'ensemble des informations des deux panoramas ;
2. tenir compte des éventuelles différences entre amers, notamment de couleurs, et non seulement de leur position dans le champ visuel ;
3. fonctionner avec deux panoramas pris dans des orientations différentes, voire inconnues ;
4. fonctionner avec des panoramas partiels.

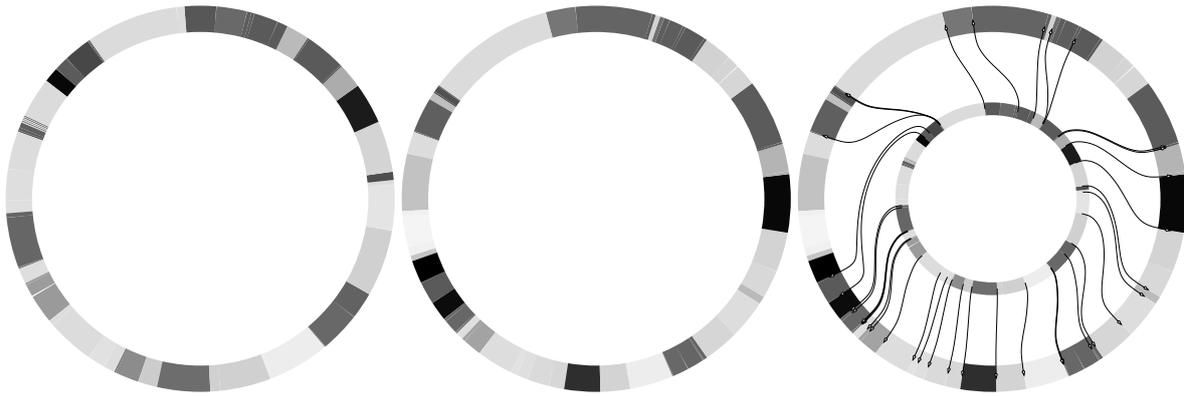


FIG. 5.1: Exemple de panoramas à appairer. À gauche, les deux panoramas, à droite l'appariement trouvé par l'algorithme.

Nous avons utilisé une méthode d'appariement fondée sur le principe de la programmation dynamique. Elle est conçue pour les panoramas segmentés, mais pourrait fonctionner avec des panoramas échantillonnés, comme ceux utilisés par Röfer, en considérant que chaque mesure est un petit segment à elle seule.

5.2.3 La programmation dynamique

La programmation dynamique [BEL 57] est une famille de méthodes qui permet, lorsque son principe est applicable, de réduire le travail nécessaire pour chercher la solution optimale à un problème défini mathématiquement.

Nous énonçons de façon abstraite les hypothèses de la programmation dynamique. Dans le paragraphe suivant, chaque point sera repris de façon concrète. Les hypothèses sont les suivantes :

- un problème d'optimisation global (recherche d'une combinaison de paramètres qui maximise ou minimise une fonction mathématique) ;
- constitué de sous-problèmes élémentaires, qu'on peut regrouper en sous-problèmes partiels ;
- il existe une fonction de coût permettant de mesurer localement la qualité d'une solution à chaque sous-problème élémentaire ;
- la fonction de coût peut être étendue pour mesurer la qualité d'une solution à un sous-problème partiel et au problème global ;
- principe d'optimalité de Bellman : si l'on dispose des solutions optimales aux sous-problèmes partiels, on peut les utiliser pour trouver la solution optimale à un problème complet.

Dans ce contexte, la programmation dynamique permet d'obtenir la solution optimale au problème global sans explorer toutes les combinaisons possibles.

Le principe consiste à enchaîner implicitement les solutions intermédiaires en remplissant un tableau puis, une fois le tableau rempli, à le parcourir pour trouver la solution optimale. Tout l'intérêt de la programmation dynamique tient dans le fait que chaque case du tableau que l'on remplit participe

potentiellement à de nombreuses solutions.

Deux domaines d'application classiques sont la recherche d'un chemin optimal dans un graphe et la recherche de ressemblances entre des séquences d'éléments (sous-séquences communes, notamment).

5.2.4 Application aux appariements de panoramas

Il s'agit dans notre cas :

- d'un problème d'appariement entre deux séquences d'éléments angle/couleur représentant des mesures du panorama entourant le robot, afin de déterminer quels objets sont communs entre les deux panoramas ;
- un sous-problème élémentaire est d'apparier un élément de chaque séquence, un sous-problème partiel est un appariement couvrant une partie de chaque panorama ;
- la fonction score mesure la ressemblance entre deux éléments. Elle vaut 1 pour deux éléments identiques en tout point, et s'approche de 0 pour des éléments différents ;
- on définit le score d'appariement d'une sous-séquence comme la somme des scores élémentaires ;
- parce que la fonction score est additive, la connaissance des solutions optimales à trois sous-problèmes ($i-1$ et $j-1$ éléments, i et $j-1$ éléments, $i-1$ et j éléments) permet de simplement déterminer la solution optimale à i et j éléments, comme nous le verrons ci-dessous.

La programmation dynamique a déjà été utilisée pour apparier des panoramas par Einsele [EIN 97, EIN 01]. Les panoramas en question consistaient en des segments de droites localisés autour d'un robot, reconstitués à partir de mesures de distances. Il n'y avait pas d'information de couleur.

Nous nous sommes inspiré, pour l'implémentation de l'algorithme, d'un exemple appliqué à la recherche de sous-séquences de symboles communs donné dans [COR 98].

5.2.4.1 Matrice de ressemblances

On note respectivement m et n le nombre d'amers des deux panoramas à apparier. On indice les amers par i et j avec $0 \leq i < m$ et $0 \leq j < n$.

Au cours de l'application de la programmation dynamique, chaque élément d'une des deux séquences sera comparé une fois et une seule à chaque élément de l'autre séquence. Pour des raisons pratiques, l'implémentation utilisée commence par remplir un tableau M qu'on appelle *matrice des ressemblances*.

Nous avons choisi de noter la ressemblance entre un amer et un autre sous la forme d'un réel compris entre 0 et 1, en tenant compte des écarts entre différents paramètres mesurés (teinte, saturation, luminance, ou bien distance dans l'espace CrCb). Nous détaillerons les fonctions de ressemblances utilisées dans nos expériences dans le chapitre qui leur est dédié, en sections 9.1.5 et 9.4.4.

On peut souhaiter tenir compte de l'écart d'angle entre deux amers, si les panoramas sont cor-

rectement orientés et si on a des raisons de penser que les bonnes paires ont des parallaxes faibles. Ainsi, si deux secteurs vus j_1 et j_2 ont tous deux des caractéristiques proches du secteur mémorisé i , la fonction de ressemblance favorisera l'appariement de celui qui est le plus proche de i angulairement. Toutefois il est important, dans le cas où on apparie des vues dont l'orientation n'est pas ou mal connue, de ne pas tenir compte des écarts d'angles. Ceci est nécessaire pour l'adaptation au cas circulaire que nous présenterons en section 5.2.7.

5.2.4.2 Remplissage de la matrice des appariements potentiels

On remplit un tableau $\mathbf{D}_{i,j}$, $0 \leq i \leq m$ et $0 \leq j \leq n$ (inégalités larges des deux côtés). $\mathbf{D}_{i,j}$ est le score de l'appariement optimal pour le sous-problème dans lequel on se propose de n'apparier respectivement que les i premiers éléments du panorama mémorisé et les j premiers éléments du panorama vu.

C'est ici que le principe d'optimalité de Bellman est utilisé : le score de la solution optimale à i et j éléments est le meilleur de trois possibilités : soit reprendre l'appariement optimal à i et $j - 1$ éléments inchangé, soit reprendre l'appariement optimal à $i - 1$ et j éléments inchangé, soit reprendre l'appariement optimal à $i - 1$ et $j - 1$ éléments en créant un lien entre les amers i, j .

Le remplissage du tableau \mathbf{D} traduit ce raisonnement. Il se remplit de proche en proche comme suit :

$$\begin{aligned} \mathbf{D}_{0,0} &= 0 \\ \forall i, \mathbf{D}_{i,0} &= 0 \\ \forall j, \mathbf{D}_{0,j} &= 0 \\ \forall i > 0, \forall j > 0, \mathbf{D}_{i,j} &= \max(\mathbf{D}_{i,j-1}, \mathbf{D}_{i-1,j}, \mathbf{D}_{i-1,j-1} + \mathbf{M}_{i,j}) \end{aligned}$$

Pour chaque case d'indices i et j strictement positifs, l'opérateur max intervient, choisissant une valeur entre trois. On note à chaque application de l'opérateur max lequel des trois choix est sélectionné. Cela définit un tableau $\mathbf{C}_{i,j}$ défini pour des indices i et j strictement positifs $0 < i \leq m$ et $0 < j \leq n$:

$$\begin{aligned} \mathbf{C}_{i,j} &= 0 \text{ si c'est } \mathbf{D}_{i,j-1} \text{ qui est retenu} \\ \mathbf{C}_{i,j} &= 1 \text{ si c'est } \mathbf{D}_{i-1,j} \text{ qui est retenu} \\ \mathbf{C}_{i,j} &= 2 \text{ si c'est } \mathbf{D}_{i-1,j-1} + \mathbf{M}_{i,j} \text{ qui est retenu} \end{aligned}$$

Notons qu'en théorie, un cas d'égalité pourrait se produire lors de l'application de l'opérateur max. Cela est possible si plusieurs appariements différents ont exactement le même score optimal. Il serait possible de les énumérer lors de l'extraction de l'appariement mais cela n'est pas important dans notre application. Le cas est en outre tout à fait improbable dans une application réelle, les mesures étant bruitées de façon aléatoire. Nous ne nous y attardons donc pas.

5.2.4.3 Extraction de l'appariement

Nous avons vu que la valeur $D_{i,j}$ représente le score de l'appariement optimal du sous-problème ne faisant intervenir que les i premiers amers du premier panorama et les j premiers amers du second.

Une fois les deux tableaux remplis, on connaît le score de l'appariement optimal complet $D_{m,n}$ mais pas encore l'appariement lui-même. Pour cela, on parcourt en marche arrière le tableau C , en partant de $C_{m,n}$ soit $C_{i,j}$ avec $i = m$ et $j = n$. Alors :

- Si $C_{i,j} = 0$ on décrémente i
- Si $C_{i,j} = 1$ on décrémente j
- Si $C_{i,j} = 2$ on décrémente i et j et on apparie les amers correspondants

Quand i ou j atteint zéro, on a décrit l'appariement complet.

5.2.4.4 Petit exemple concret

Nous prenons pour exemple la matrice des ressemblances suivante, qui signifie essentiellement que l'amer d'indice 1 du premier panorama ressemble à l'amer 1 du second, que l'amer 2 du premier ressemble assez à l'amer 3 du second, et que l'amer 3 du premier ressemble à l'amer 2 du second.

$i \downarrow j \rightarrow$	0	1	2	3	4
0	0.00	0.00	0.00	0.00	0.00
1	0.00	0.70	0.00	0.00	0.00
2	0.00	0.00	0.00	0.55	0.00
3	0.00	0.00	0.79	0.00	0.00
4	0.00	0.00	0.00	0.00	0.00

L'appariement par programmation dynamique respecte l'ordre. Il ne pourra donc retenir les trois paires 1-1 2-3 3-2 qui correspondent à une inversion des amers 2 et 3. Il va choisir en fonction des scores de ressemblances.

L'algorithme remplit de proche en proche les tableaux C et D . Nous représentons le tableau D , augmenté d'un symbole indiquant le cas de la case C concerné de façon plus claire qu'un chiffre. \uparrow correspond au cas 0, \leftarrow au cas 1, et $\swarrow \bowtie$ au cas 2. Le cas 2 dans une case (i, j) correspond à retenir la paire $(i - 1, j - 1)$

$i \downarrow j \rightarrow$	0	1	2	3	4	5
0	0.00	0.00	0.00	0.00	0.00	0.00
1	0.00	\uparrow 0.00	\uparrow 0.00	\uparrow 0.00	\uparrow 0.00	\uparrow 0.00
2	0.00	\uparrow 0.00	$\swarrow \bowtie$ 0.70	\leftarrow 0.70	\leftarrow 0.70	\leftarrow 0.70
3	0.00	\uparrow 0.00	\uparrow 0.70	\uparrow 0.70	$\swarrow \bowtie$ 1.25	\leftarrow 1.25
4	0.00	\uparrow 0.00	\uparrow 0.70	$\swarrow \bowtie$ 1.49	\leftarrow 1.49	\leftarrow 1.49
5	0.00	\uparrow 0.00	\uparrow 0.70	\uparrow 1.49	\uparrow 1.49	\uparrow 1.49

Partant du coin en bas à droite, en suivant les flèches on trouve l'appariement suivant : $i = 1$ apparié à $j = 1$, et $i = 3$ apparié à $j = 2$.

L'algorithme a bien « envisagé » d'apparier $i = 2$ avec $j = 3$ mais cela n'a pas été retenu car le score partiel (1.25) est moins bon que celui de la solution finalement retenue (1.49). On peut considérer que cette décision a été prise dans la case $D_{4,4}$.

5.2.5 Bénéfices de cette méthode

La programmation dynamique apporte des bénéfices par rapport à la méthode originale d'appariement des modèles CC.

5.2.5.1 Prise en compte des informations de l'ensemble des amers

Les appariements obtenus par programmation dynamique tiennent compte de l'ensemble des données présentes. Ceci sera illustré en simulation en section 8.2 et, dans le cas réel, en section 9.2 et correspond aux points 1 et 2 du cahier des charges (5.2.2).

5.2.5.2 Prise en compte des couleurs des secteurs à appairer

Cette méthode d'appariement tient compte des couleurs mais, surtout, elle tient compte naturellement des éventuelles erreurs de mesure de la couleur. Ainsi, même si la teinte d'un amer semble avoir changé d'une mesure à l'autre, ou si deux amers ont des teintes assez proches, l'appariement correct sera trouvé sans qu'il soit nécessaire de définir un seuil arbitraire et fixé de ressemblance. Dans chaque cas particulier il existe un seuil d'écart de teinte tel que l'appariement ne se fait plus, mais ce seuil (qui n'apparaît pas explicitement) est spontanément ajusté en fonction de l'ensemble des amers et de leurs couleurs.

5.2.5.3 Fonctionne avec des panoramas d'orientations différentes

Ce point du cahier des charges n'est pas bien couvert par l'algorithme présenté ci-dessus. En effet, même dans le cas d'un panorama complet qui fait tout le tour sans coupure, la représentation utilisée pour l'appariement introduit une coupure. Les tableaux C et D utilisés ne « bouclent » pas (de $i = m$ à $i = 0$ ou $i = 1$ par exemple, idem pour j), de sorte que les appariements ne peuvent se faire d'un côté à l'autre de la coupure. Nous avons trouvé une solution satisfaisante à ce problème, que nous exposons en section 5.2.7.

5.2.5.4 Fonctionne avec des panoramas incomplets

La programmation dynamique permet d'apparier des panoramas incomplets. Contrairement à la méthode des modèles CC, si une zone de l'environnement est couverte dans un panorama mais pas

dans l'autre, la méthode s'abstient correctement d'apparier les amers qui ne sont visibles que dans l'un des deux. Cette propriété est importante pour un robot dont le capteur ne couvre pas tout le tour et nous l'avons mise en évidence et utilisée en section 9.2.

De même, si un amer a disparu ou est apparu entre deux prises de vues, il sera correctement ignoré lors de l'étape d'appariement.

5.2.6 Complexité calculatoire

La programmation dynamique nécessite essentiellement le remplissage de la matrice des similarités, puis de la matrice des appariements potentiels, deux opérations en $O(mn)$. Typiquement, les deux panoramas ont un nombre d'amers similaire. La complexité de l'algorithme est donc $O(n^2)$, n étant le nombre d'amer d'un panorama.

5.2.7 Adaptation au cas circulaire

Les panoramas pris avec une caméra classique motorisée comme celle des Pioneer 2 ne font pas le tour complet mais se limitent à 200° . En revanche, en simulation comme avec un capteur panoramique, on a accès à des images qui couvrent bien tout le cercle et il est nécessaire de savoir faire des appariements corrects entre ces panoramas. Malheureusement, la programmation dynamique considère l'appariement de deux listes non circulaires d'objets. En pratique, cela signifie qu'un objet tout près d'un bord du premier panorama ne sera jamais apparié avec un objet tout près de l'autre bord de l'autre panorama, parce que tous les autres amers les séparent dans la représentation linéaire du panorama, alors qu'ils sont tout proches dans la topologie circulaire naturelle des panoramas. Ignorer le problème peut aboutir à ne pas apparier jusqu'à la moitié du panorama, dans le cas où on tente d'apparier deux panoramas pris avec des orientations opposées, comme illustré en figure 5.2.

Une adaptation de la programmation dynamique au cas circulaire doit permettre non seulement d'apparier correctement des panoramas pris dans des orientations connues mais différentes (parce que la « coupure » dans la représentation linéaire de chaque panorama ne pose plus problème), mais surtout elle doit permettre d'apparier des panoramas sans connaître leur orientation respective.

Dans ce but, nous supprimons le terme qui dépend des angles dans la fonction de ressemblance définie en 5.2.4.1. La ressemblance entre deux amers ne dépend plus de l'orientation relative dans laquelle sont présentées les panoramas. La matrice sur laquelle opère l'algorithme ne dépend donc plus de cette orientation relative, et l'appariement obtenu non plus.

Nous avons donc cherché une façon d'adapter la programmation dynamique au cas circulaire. Einsele [EIN 97, EIN 01] a utilisé la programmation dynamique avec des panoramas constitués de segments de droites perçus par un télémètre laser. Dans sa thèse, il propose une méthode consistant à essayer toutes les combinaisons de points de départ, en effectuant une rotation des colonnes et des lignes des deux tableaux utilisés par la méthode. Cela augmente le coût de l'algorithme en $O(m^2n^2)$. Il a toutefois introduit des modifications dans l'algorithme adaptées à sa situation, qui permettent de

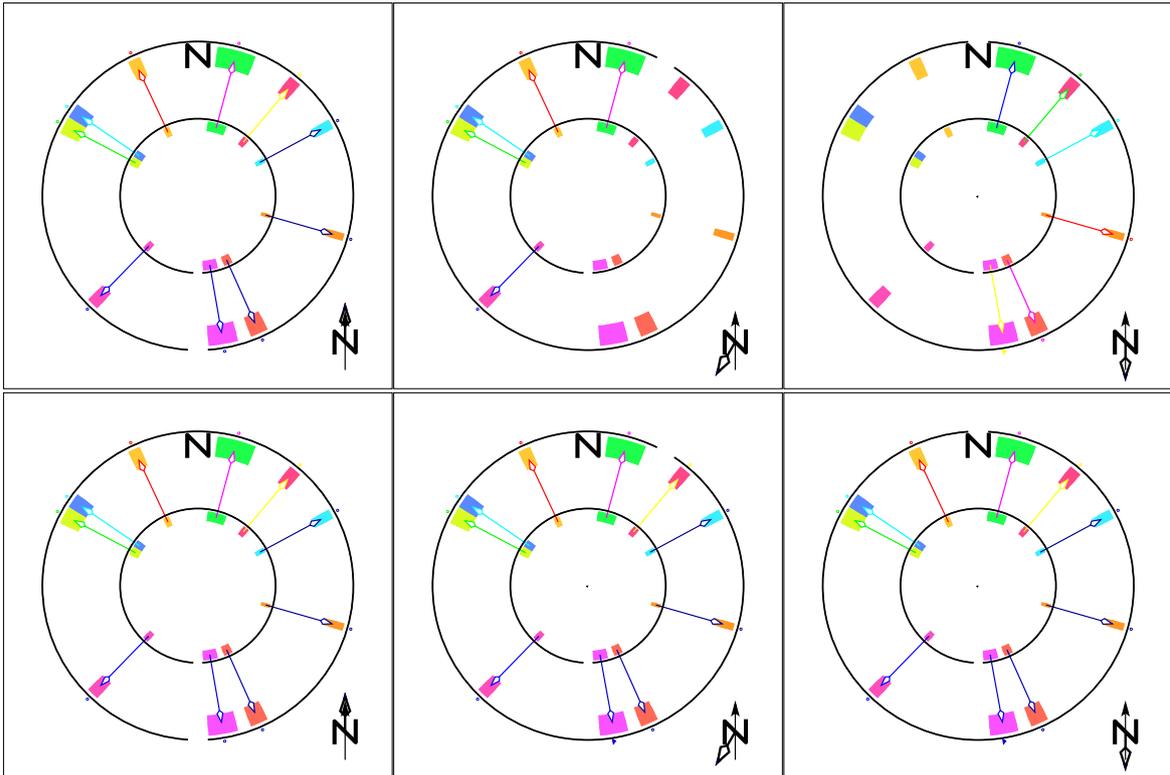


FIG. 5.2: Amélioration apportée par la programmation dynamique circulaire. L'animat prend un panorama et doit l'apparier avec le même panorama mais pris dans une orientation différente. Parce que la programmation dynamique ne traite que des listes d'éléments sans rebouclage, elle peut négliger jusqu'à la moitié des éléments à cause du décalage (en haut). Le cercle noir entourant les vues est interrompu pour montrer où la séquence commence et s'arrête. On constate que, en effet l'algorithme trouve bien la meilleure façon d'apparier ces deux séquences linéaires. La modification que nous proposons est simple : elle permet de tenir compte de la nature circulaire du panorama et d'apparier correctement quelle que soit l'orientation du robot (en bas).

« sauter » des éléments qui se ressemblent peu sans les appairer, et aussi de restreindre le remplissage du tableau d'une façon qui peut faire gagner un facteur 2.

Nous avons essayé de modifier la façon de remplir le tableau afin d'autoriser un bouclage (lui donner une topologie cylindrique voire torique si on boucle les deux dimensions). Le problème est qu'il n'est plus évident que le principe d'optimalité de Bellman (section 5.2.3) s'applique toujours à cette nouvelle situation.

Finalement nous avons opté pour une modification de l'algorithme clairement définie et qui s'est avérée efficace dans toutes nos expériences.

Le principe de cette méthode est de dupliquer l'un des deux panoramas, en le concaténant avec une copie de lui-même, de sorte qu'il fait deux fois le tour, puis à appliquer l'algorithme de programmation dynamique non modifié à ce nouveau problème d'appariement. L'algorithme lui-même n'étant pas modifié, on continue de bénéficier de toutes ses propriétés.

La différence tient dans ce que l'algorithme n'est plus contraint à ne pas franchir la limite circulaire, puisqu'il peut la franchir à l'endroit de la limite entre les deux copies. En pratique, on n'a pas besoin de dupliquer le panorama lui-même, mais seulement l'information fournie à l'algorithme : la matrice des ressemblances. On lui fournit donc M' de dimensions $2m \times n$ au lieu de $m \times n$, en accolant une copie de la matrice M sous elle-même :

$$M' = \begin{pmatrix} M \\ M \end{pmatrix} \quad (5.1)$$

Il faut ensuite interpréter l'appariement obtenu de façon un peu différente. En effet, chaque amer du premier panorama est représenté deux fois dans la matrice. Chacune des deux copies peut potentiellement être appariée avec un amer de l'autre panorama, ce qui contredit le choix que chaque élément d'une vue est apparié à au plus un élément de l'autre vue. Heureusement, ce cas indésirable est facile à caractériser car l'appariement couvre alors plus qu'un tour complet du premier panorama. Mathématiquement cela se traduit par : l'écart entre l'indice du premier amer apparié et du dernier amer apparié dépasse m .

Une façon simple de garantir qu'un amer reste apparié au plus une fois est alors de sélectionner dans le panorama doublé ($2m$ secteurs) une fenêtre de largeur m secteurs, et de regarder le score de l'appariement partiel qui ne tient compte que de cette fenêtre. Nous conserverons la fenêtre qui maximise le score. Concrètement, cela revient à chercher à quel endroit il faut couper le panorama circulaire pour maximiser le score d'un appariement linéaire.

Grâce à l'appariement circulaire que nous avons à ce stade, c'est une opération rapide, il s'agit de calculer des sommes de scores partiels s_i pour les appariements partiels sélectionnés. Désignons par s_i le score partiel de l'amer i : si i a été apparié avec j $s_i = M_{i,j}$, sinon $s_i = 0$.

m additions donnent le score de la fenêtre f_0 couvrant les amers de 0 à $m - 1$:

$$S_0 = \sum_{k=0}^{m-1} s_k \quad (5.2)$$

Pour obtenir le score S_i de la fenêtre f_i , il n'est pas besoin de calculer m sommes de m termes. Il suffit d'appliquer par récurrence $S_i = S_{i-1} - s_i - 1 + s_i + m$, ce qui fait deux opérations par fenêtre, soit au total $3m$ opérations. C'est négligeable devant le coût d'un appariement et a fortiori devant celui de faire tous les appariements. Le i qui donne le plus fort S_i et l'indice de l'amer avant lequel il faut, pour maximiser le score, couper le panorama que nous avons doublé.

Cette méthode a été appliquée avec succès dans toutes nos expériences réelles d'orientation.

Il reste une possibilité de problème, que nous avons pu mettre en évidence en construisant à dessein un cas simulé. Nous avons exprimé mathématiquement que nous voulions maximiser le score de l'appariement, en ne tenant compte que des couleurs des objets et non de leur positions, de sorte que l'appariement qu'on obtient est complètement indépendant de l'orientation relative dans laquelle sont présentés les deux vues. Ce que nous obtenons est exactement ce que nous avons exprimé, mais le résultat est parfois un peu surprenant.

Le cas est illustré par la figure 5.3. Il s'agit d'un environnement à 5 amers de tailles et distances variées. La perspective change l'ordre dans lequel les amers apparaissent dans le panorama selon l'endroit où se trouve l'animat, mettant en difficulté l'appariement. Rappelons que la variante non circulaire tient compte de l'orientation relative des panoramas, non dans la fonction de ressemblance, mais par la représentation linéaire qui fait que, dans chaque panorama, il y a un premier et un dernier amer. Ici, les panoramas commencent au sud et tournent dans le sens horaire. Rappelons aussi que la programmation dynamique apparie toujours les objets en respectant l'ordre : un amer B apparaissant plus loin que A apparié à A' ne sera jamais apparié avec un B' qui apparaît avant A', ce qui se traduit graphiquement par l'absence de croisement entre des flèches qui représentent les paires, dans une représentation rectiligne des panoramas. L'appariement au milieu à droite de la figure semble incorrect car les flèches courbes se croisent. Les appariements sont en fait aA cB eC et il n'y a pas de croisement dans la topologie linéaire qui a été choisie. En fait, aucun des 4 appariements présenté dans la figure n'est erroné. C'est le fait que l'ordre des amers change dans la vue qui cause l'incomplétude des appariements.

La bonne réponse à ce type de problème est de considérer, que si l'appariement peut être incomplet sans causer de tort majeur à l'animat, alors la méthode est suffisante. De plus, nous avons l'intuition (mais nous ne l'avons pas démontré) que le fait que les amers se répartissent sur plus de la moitié du panorama (bas de la figure 5.3) est suffisant pour éviter les croisements des flèches.

En fait, dans une application réelle, si l'ordre d'éléments saillants dans le paysage que perçoit l'animat dépend du déplacement de ce dernier, nous pensons que la bonne solution est de considérer qu'il s'agit de lieux différents et de mémoriser des vues aux points de transitions.

5.2.8 Conclusion sur l'appariement

Nous nous sommes donné une méthode d'appariement adaptée aux panoramas segmentés, mais qui est originale dans ce contexte de navigation par panorama. Elle innove en montrant que la programmation dynamique est applicable à des panoramas n'utilisant que des informations de couleur.

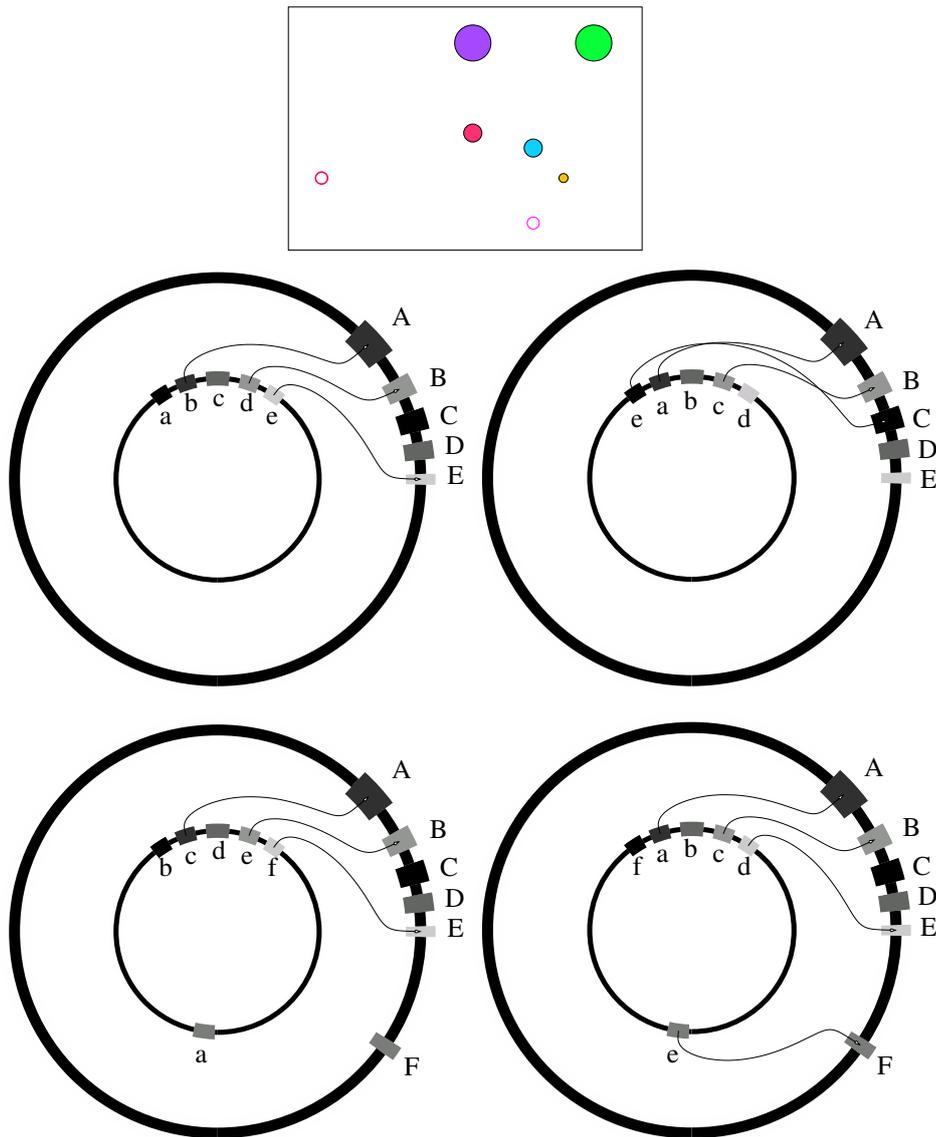


FIG. 5.3: Exemple conçu pour mettre en évidence une limitation de la méthode d'appariement : ces appariements sont toujours justes, mais incomplets si l'ordre des amers change. En haut, une vue aérienne d'un environnement, montrant l'animat (cercle) au sud de 5 objets de couleurs différentes (disques). L'animat, au sud, prend une vue, puis se rend à l'ouest (second cercle), où la perspective change l'ordre des objets. Au milieu à gauche, appariement des panoramas par programmation dynamique non circulaire. Au milieu à droite, appariement des mêmes panoramas par la variante circulaire. Cette variante choisit, en une seule application du premier algorithme sur le double de données, l'endroit où couper la vue mémorisée pour maximiser le score obtenu et trouver le meilleur appariement. Ici, elle a coupé entre le premier et le second amer (e et a), ce qui donne l'impression que l'ordre n'est plus respecté (croisement de flèches). (Les lettres indiquent l'ordre des amers dans la topologie linéaire.) En bas : il suffit d'un amer supplémentaire pour que ce faux problème disparaisse. À gauche, la programmation dynamique non circulaire échoue à appairer l'amer au sud. À droite, la variante circulaire trouve le même découpage pertinent, mais l'appariement est plus intéressant. C'est à ce dernier cas que les cas réels ressemblent.

Notre méthode d'appariement prend en compte en chaque point les informations de l'ensemble des amers, exploite les informations permettant de distinguer les secteurs à appairer les uns des autres (dans notre cas, l'information de couleur), permet d'appairer correctement des panoramas même lorsqu'ils ont été pris dans des orientations différentes et inconnues, et gère raisonnablement le cas des panoramas incomplets. La méthode n'apparie pas tous les amers si leur ordre change d'une vue à l'autre. Toutefois, nous estimons que cela ne pose pas de problème majeur.

Il est intéressant de remarquer que c'est le choix d'ignorer totalement l'information d'angle au cours de l'application de la méthode qui permet de produire un appariement indépendant de l'orientation relative des deux vues. Ceci ouvre la possibilité pour un animat de reconnaître les lieux sans connaître son orientation. Nous avons naturellement exploité cette propriété de notre nouvelle méthode d'appariement dans nos expériences. Dans la suite de cette thèse, l'animat s'appuiera sur cette reconnaissance des lieux pour retrouver son orientation puis son chemin.

L'étape suivante, une fois qu'on dispose de deux panoramas appariés, consiste, selon le contexte, à estimer la direction à suivre pour revenir au point représenté par le panorama de référence ou, si besoin est, à estimer tout d'abord son orientation actuelle. Nous avons choisi de présenter notre méthode d'orientation dans un chapitre dédié, après la section présentant notre méthode de calcul de la direction à suivre.

5.3 Direction à suivre : méthode par « élimination »

Le choix qui est fait dans les modèles CC d'utiliser une somme vectorielle pour combiner les indications des différents amers est plus intuitif que raisonné. Nous avons cherché s'il était possible de faire un choix plus élaboré sans introduire de nouvelle hypothèse.

Un écart de parallaxe représente une information incomplète sur un amer (car on ne sait pas la distance qui nous sépare de l'amer) et, quand on dispose d'un certain nombre de ces informations, une somme vectorielle de vecteurs tangentiels ne garantit pas un traitement « juste », seulement une sorte de moyenne. Par ailleurs, la somme vectorielle utilisée dans la plupart des modèles ne permet pas de détecter d'éventuelle erreur ou incohérence dans les mesures qui perturberait le calcul de la direction à suivre, mais fournit seulement l'information de direction demandée sans la garantir.

Au cours de nos expérimentations, il nous a paru que le choix d'une somme vectorielle n'est en fait qu'une méthode d'agrégation des différentes indications déduites de l'observation des parallaxes, et qu'en cherchant à préciser ce que signifiaient ces indications on pouvait probablement trouver une meilleure méthode d'agrégation.

Le résultat que nous avons obtenu est ce que nous appelons « calcul de direction à suivre par élimination ».

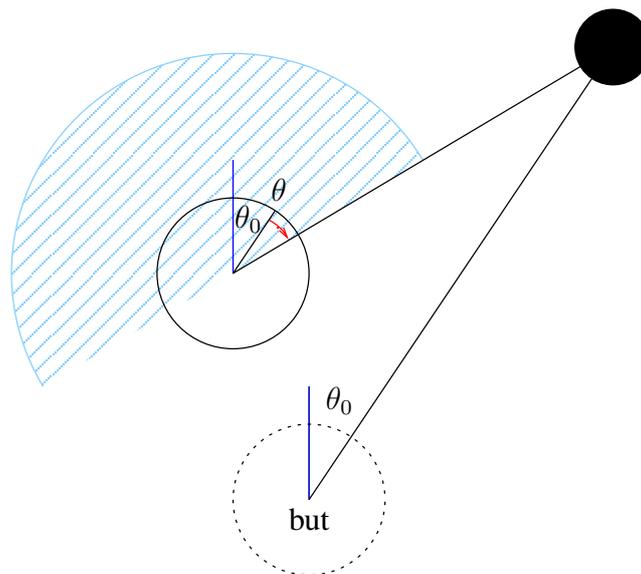


FIG. 5.4: Principe de la méthode par élimination, pour le calcul de mouvement. Même sans connaître aucune distance à un amer, l'animat peut, par observation de parallaxes, éliminer les directions qui l'éloignent du but. On montre ici un exemple avec un seul amer (disque plein noir). Depuis le but, l'animat voyait l'amer sous l'azimut θ^0 . Depuis sa position actuelle (au centre), il le voit sous l'azimut θ , d'où une parallaxe de $\theta - \theta^0$ (petite flèche). La méthode par élimination pénalise les directions vers lesquelles un déplacement augmenterait la largeur de cette parallaxe (demi-disque hachuré). Une fois les pénalités comptées pour tous les amers, n'importe laquelle des directions restantes ne peut que diminuer toutes les parallaxes. On peut démontrer que, s'il y a au moins deux parallaxes de signe opposé, n'importe laquelle des directions restantes ne peut que rapprocher l'animat du but.

5.3.1 Définition

Pour présenter la méthode, nous supposons que nous nous sommes donné : une méthode de signature segmentée, une méthode d'appariement qui fonctionne et, pour ce qui concerne la direction à suivre, le choix de la composante tangentielle dans sa variante méthode directe (2.3.1.3).

La modification que nous proposons part d'une remarque. Chaque parallaxe angulaire observée n'est pas une invitation à faire un pas dans la direction orthogonale à l'azimut de l'amer concerné, même si c'est cette direction qui *localement* fait décroître le plus vite cet écart comme le font remarquer Hong [HON 91] puis Weber [WEB 99]. L'existence de cet écart d'angle indique que, parmi les directions que peut prendre l'animat, la moitié d'entre-elles diminue la parallaxe et l'autre moitié l'augmente. S'il n'y a pas d'erreur d'appariement, on sait avec certitude que le but se trouve du côté qui réduit l'écart (cf. figure 5.4).

Cette approche permet d'agréger de façon simple les informations de plusieurs amers sans interférence : si on détermine qu'il existe un ensemble non vide de directions qui font diminuer une ou plusieurs parallaxes et n'en font augmenter aucune, on convergera nécessairement vers un point où ces parallaxes sont minimales. Idéalement, on convergera vers le point où toutes les parallaxes seront

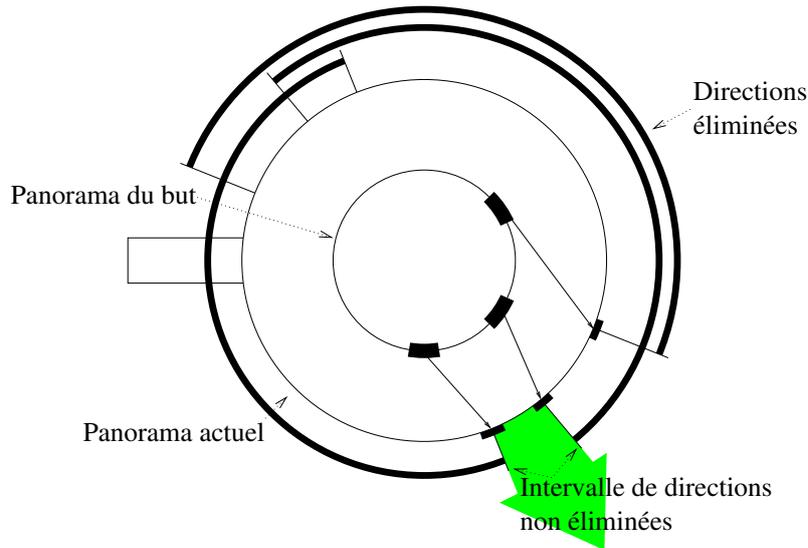


FIG. 5.5: Exemple d'application de la méthode par élimination. Pour chaque amer apparié entre vue mémorisée et vue actuelle, l'animat considère le sens de la parallaxe et élimine le demi-cercle (représenté en noir épais) des directions qui augmenteraient la parallaxe s'il les suivait. Si tout est cohérent, il reste un intervalle non éliminé. L'animat peut suivre n'importe quelle direction de cet intervalle pour se rapprocher du but, par exemple le milieu.

nulles : le but. La méthode est représentée concrètement sur la figure 5.5.

5.3.2 Formalisation de la méthode par élimination

Afin de s'assurer que l'idée est applicable, en particulier en cas de mauvais appariement, nous formalisons cette approche.

Donnons nous une fonction $P(\alpha)$ qui, à chaque direction possible α , associe une pénalité. On part de la fonction nulle $P_0(\alpha)$. On passe en revue les amers appariés. Pour chaque amer i on calcule quel demi-plan est pénalisant, et on ajoute une fonction pénalité à P : $P_{i+1}(\alpha) = P_i(\alpha) + p(\alpha, \theta_i^0, \theta_i)$.

Si la fonction pénalité $p(\alpha, \theta_i^0, \theta_i)$ est positive (du côté pénalisé) et nulle (du côté non pénalisé), alors la valeur de $P(\alpha)$ permet de savoir immédiatement pour une direction donnée si elle a fait l'objet d'au moins une pénalité ou non. Une façon possible de l'implémenter est de construire un tableau associant à n directions à suivre (par exemple, $n = 360$ pour discrétiser par pas de 1°) une valeur qui représente la pénalité associée à cette direction, mais on peut garder à l'esprit la version continue.

Il est facile de prouver que l'ensemble des directions non pénalisées est en général d'un seul bloc. Après application de la première pénalité, il reste un demi-cercle de valeurs non pénalisées, qui est un ensemble contigu (i.e. d'un seul tenant) dont la topologie est celle d'un intervalle de réels. Chaque application d'une nouvelle pénalité fait que le nouvel ensemble de directions non pénalisées est l'intersection de l'ensemble actuel et d'un autre ensemble (l'intervalle des directions non pénalisées par

cette nouvelle opération). Il y a une exception : le cas de deux demi-cercles opposés dont l'intersection se réduit à deux points diamétralement opposés. Dans cette topologie, l'intersection d'un intervalle avec un autre intervalle est, soit l'ensemble vide, soit à nouveau un intervalle. L'ensemble des directions non pénalisées est donc soit un intervalle ou l'ensemble vide soit, dans le cas exceptionnel, deux points diamétralement opposés.

Nous n'avons pas démontré que l'ensemble en question est toujours non vide. Nous pensons que c'est le cas au moins lorsque les circonstances suivantes sont vérifiées :

- il n'y a pas d'erreur d'orientation (l'animat connaît son orientation dans chacune des deux vues) ;
- le monde est statique ;
- il n'y a pas d'erreur d'appariement ;
- les points qui servent à mesurer des angles sont indépendants de la position de l'animat (exemple : le centre d'un amer circulaire est indépendant de la direction de l'animat ; son bord gauche ne l'est pas ; le bord gauche d'une feuille sur un mur l'est) ;
- l'ordre des amers n'a pas changé entre les deux vues (en particulier, il n'y a pas d'obstacle entre le but et la position actuelle).

Nous souhaitons, bien sûr, que la méthode reste utilisable si, l'une de ces conditions n'étant pas vérifiée, toutes les directions finissent par être pénalisées au moins une fois.

Les éléments ci-dessus précisent les directions qui vont être pénalisées, mais ne précisent pas complètement la fonction $p(\alpha, \theta_i^0, \theta_i)$: de combien doit-on pénaliser ? Doit-on pénaliser uniformément ?

Pour répondre à cette question, envisageons des problèmes possibles comme une erreur d'appariement qui peut faire aboutir à une situation où on a pénalisé toutes les directions. Si le cas se produit il serait souhaitable de pouvoir tout de même obtenir une direction la moins fautive possible. On souhaite donc faire que la pénalité tende vers zéro de façon monotone aux bords du demi-cercle pénalisé, plutôt que de lui laisser présenter une discontinuité. Ainsi, si à cause d'erreurs dans les angles, les demi-cercles se recouvrent légèrement, la fonction pénalité présentera un minimum local qui sera choisi comme direction à suivre.

En prévoyance de ce genre de cas, nous avons choisi une fonction pénalité particulière, où la fonction $pente(\theta_i, \theta_i^0)$ sera précisée juste après :

$$\begin{aligned} p(\alpha, \theta_i, \theta_i^0) &= 0 \text{ si } \alpha - \theta_i \cong 0 \pmod{2\pi} \\ p(\alpha, \theta_i, \theta_i^0) &= pente(\theta_i, \theta_i^0) \cdot \sin(\alpha - \theta_i) \text{ si } 0 < \alpha - \theta_i < \pi \pmod{2\pi} \\ p(\alpha, \theta_i, \theta_i^0) &= pente(\theta_i^0, \theta_i) \cdot \sin(\alpha - \theta_i) \text{ si } -\pi < \alpha - \theta_i < 0 \pmod{2\pi} \end{aligned}$$

Elle est conçue avec à l'esprit l'analogie classique du paysage montagneux qui permet de comprendre des méthodes d'optimisation comme la descente de gradient. Cette fonction p décrit que la pénalité associée à une direction est la pente à laquelle devrait faire face l'animat s'il était dans un dièdre (deux plans se croisant suivant une ligne horizontale joignant l'animat et l'amer). L'ensemble est représenté en figure 5.6. Naturellement, la pente du côté non pénalisé est nulle et celle du côté

pénalisé est positive.

Il reste à choisir quelle pente on donne en fonction de la parallaxe observée.

Du bruit dans les mesures d'angles, surtout pour des parallaxes proches de zéro, peut faire se tromper sur le signe d'une des parallaxes, causant une pénalité sur le demi-cercle à ne pas pénaliser et inversement. Le changement de signe d'une parallaxe peut introduire une discontinuité dans la pénalité globale, si la fonction pénalité appliquée ne tend pas vers une fonction symétrique quand la parallaxe tend vers zéro. On souhaite donc que $pente(\theta_i^0, \theta_i)$ soit symétrique quand $\theta_i^0 = \theta_i$.

Par ailleurs, si un amer est exactement bien aligné, c'est-à-dire avec une parallaxe nulle, nous ne souhaitons pas qu'aucune pénalité ne soit appliquée. Cela reviendrait à ignorer l'amer et rien n'inviterait l'animat à essayer de maintenir l'alignement. On souhaiterait que les pénalités tendent à maintenir cet alignement, mais que toutefois cette indication ne prenne trop de priorité. On souhaite donc que quand $\theta_i^0 = \theta_i$, $pente(\theta_i^0, \theta_i)$ pénalise légèrement toutes les directions sauf les deux directions qui maintiennent l'alignement (vers l'amer et à l'opposé).

Pour tenir compte de cela, nous nous sommes donné une parallaxe limite α_{lim} fixée à 2 degrés. Si la parallaxe mesurée vaut moins que la limite, on pénalise un peu les directions qui modifient beaucoup cette parallaxe, ce qui par rebond favorise les directions qui la modifient peu.

Afin de satisfaire toutes ces contraintes, nous nous sommes donné une fonction qui correspond à un dièdre d'angle constant α_d , qu'on fait tourner en fonction de la parallaxe. Elle est donnée formellement ci-dessous et montrée de façon plus visuelle en figure 5.6.

Parallaxe	Pente à gauche de l'amer	Pente à droite de l'amer
$D \leq -\alpha_{lim}$	α_d	0
$-\alpha_{lim} < D < 0$	$\alpha_d \frac{\alpha_{lim} - D}{2\alpha_{lim}}$	$\alpha_d \frac{\alpha_{lim} - D}{2\alpha_{lim}}$
$D = 0$	$\alpha_d/2$	$\alpha_d/2$
$0 < D < \alpha_{lim}$	$\alpha_d \frac{\alpha_{lim} - D}{2\alpha_{lim}}$	$\alpha_d \frac{\alpha_{lim} + D}{2\alpha_{lim}}$
$\alpha_{lim} \leq D$	0	α_d

Remarquons par souci de complétude que cette méthode, ne s'appuyant pas sur une somme de vecteurs comme les autres, ne produit pas un vecteur ayant une norme particulière. Si on a besoin d'un vecteur dont la norme tend vers 0 quand l'animat s'approche du but, comme dans le modèle PV, on peut utiliser la somme des valeurs absolues des parallaxes.

5.3.3 Analyse théorique de performances

Par rapport aux méthodes utilisant une somme de vecteurs, cette méthode apporte une certaine exactitude. En effet, si les hypothèses sont vérifiées, il est certain que le point d'où a été pris le panorama de référence se trouve dans l'intervalle où la pénalité est nulle.

Pour ce qui est de la précision, Franz a démontré (section 4.5) qu'avec les modèles appliquant la méthode moyenne à la composante tangentielle, la direction choisie fait un écart de moins de 90° avec

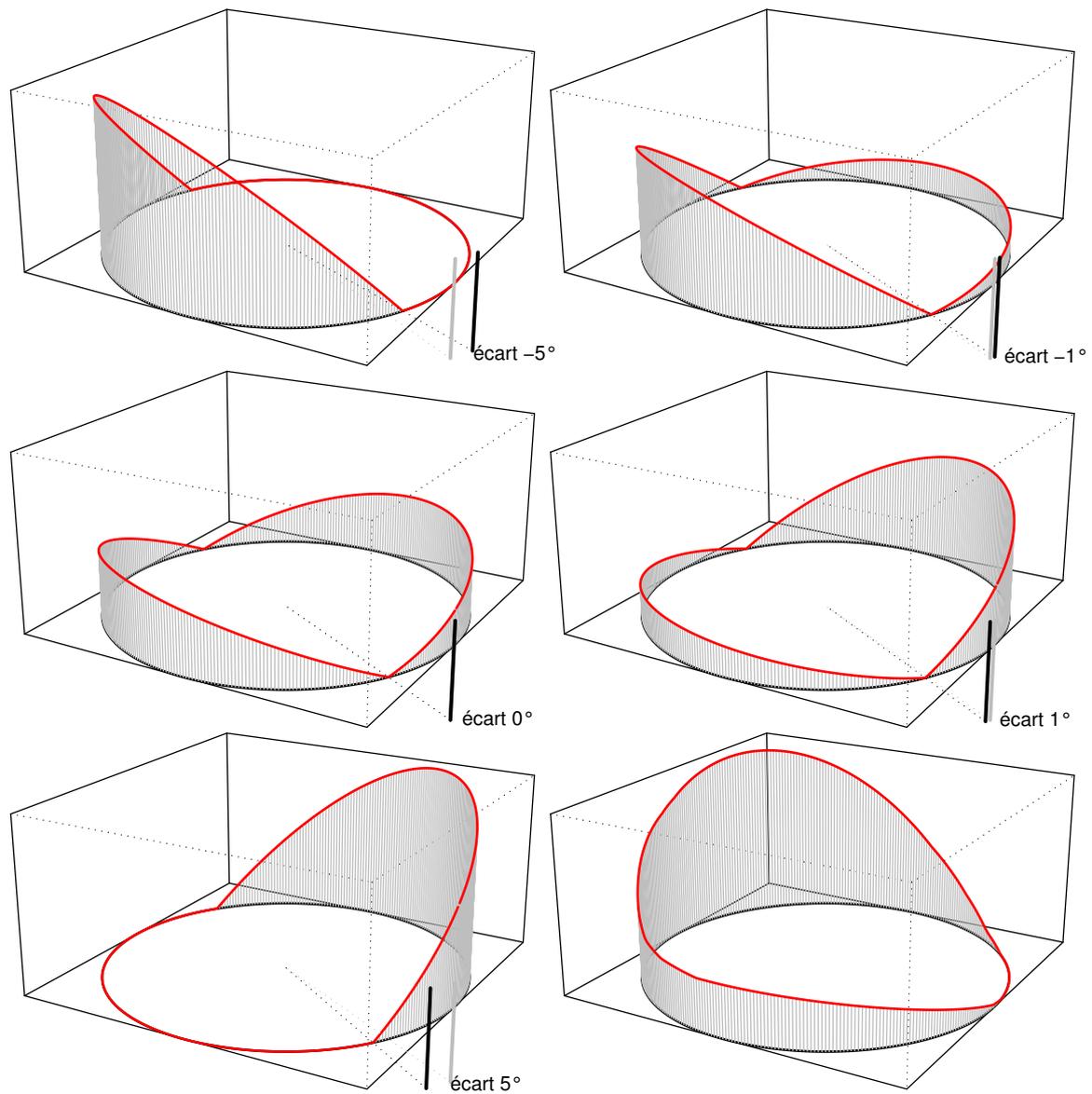


FIG. 5.6: Principe de la méthode par élimination pour déterminer les directions qui rapprochent l'animat du but. Ce schéma est à rapprocher du schéma 3.9. Le segment gris vertical représente la position qu'occupait un amer vu depuis le but. Le segment noir représente la position actuelle de ce même amer. On représente, pour chaque direction autour de l'animat, un segment vertical d'autant plus haut que la direction est pénalisée. Pour de grands écarts (schémas -5° et 5°), on pénalise la moitié des directions. Quant l'écart est suffisamment petit, on peut envisager un raffinement pénalisant les directions qui modifient l'écart, afin de conserver l'alignement (schémas $-1^\circ, 0^\circ$ et 1°). L'animat choisit la direction la moins pénalisée et, si c'est tout un intervalle de direction, il prend le milieu. La dernière figure montre un cas d'utilisation dans un monde à 10 amers.

la direction menant au but en ligne droite. Il est nécessaire d'avoir au moins un amer apparié dans les deux vues.

Notre méthode conduit au même degré de précision avec un seul amer, mais la précision peut être bien meilleure encore. L'incertitude sur la direction à suivre est la largeur de l'intervalle de directions non pénalisées, qui peut être étroit, comme on peut en voir un exemple en figure 5.5.

En fait, la question à laquelle la méthode par élimination répond n'est pas « dans quelle direction faut-il aller pour se rapprocher le plus vite du but », mais « quelles sont les directions qui rapprochent du but sans contourner d'obstacle » ? Cette façon de voir les choses fait apparaître que cet intervalle est plus qu'un intervalle d'erreur puisque n'importe laquelle des directions qu'il couvre mène au but.

Cette méthode permet aussi d'extraire des informations dont on ne dispose pas avec les réponses des autres modèles étudiés : on peut détecter un problème, comme un amer déplacé ou une orientation incorrecte, car ils pourront causer un écart tel que l'intervalle de directions non pénalisées devient vide. On peut alors exploiter l'information des pénalités pour déterminer ce qui ne convient pas. Par exemple, si c'est un amer déplacé, l'examen des pénalités peut montrer qu'un intervalle n'est pénalisé que par un seul amer, et tout redevient cohérent si on l'ignore.

En revanche, les mêmes causes ayant les mêmes effets, comme cette méthode est fondée sur les mesures de parallaxes qui dépendent de la connaissance par l'animat de son orientation propre, cette méthode est en principe aussi dépendante de la direction que la méthode PV.

5.3.4 Étude expérimentale

Nous présentons une comparaison expérimentale de performances qui permet de se rendre compte de la différence qualitative entre la méthode par élimination et la composante tangentielle.

Pour illustrer clairement les changements qu'introduit la méthode par élimination, nous la montrons dans les mêmes conditions que celles qui ont mis en défaut la composante radiale, en section 3.5. C'est l'objet des figures 5.7 et 5.8. La figure 5.9 présente les résultats obtenus avec la méthode par élimination dans les conditions du test de la composante tangentielle en section 3.2.1.

Ces expériences confirment les prédictions théoriques :

- la méthode par élimination est plus précise quand les amers sont plus nombreux, parce que l'intervalle non pénalisé est plus petit ;
- la méthode par élimination est plus précise quand l'animat est loin du but parce que l'intervalle non pénalisé est plus petit ;
- la méthode par élimination ne bénéficie pas de l'évitement d'obstacle spontané de la composante tangentielle.

Ce dernier point est ou non un défaut selon que l'on attend de la méthode un chemin à suivre ou bien seulement une indication de la direction où se trouve le but.

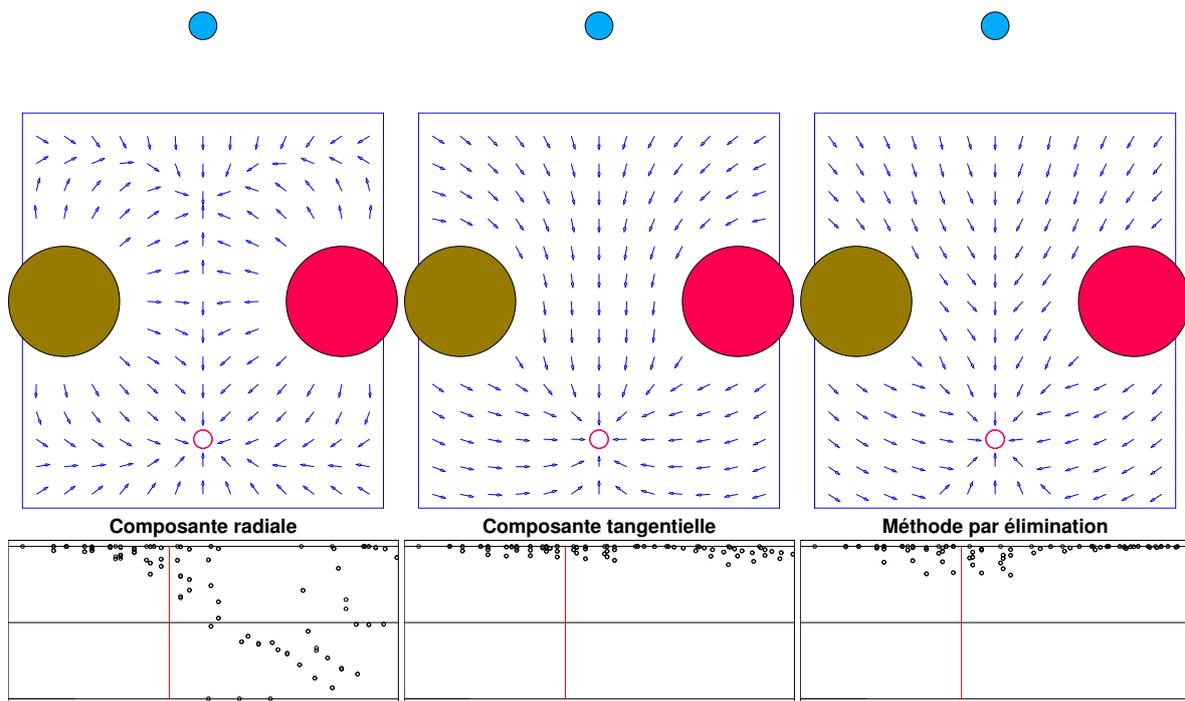


FIG. 5.7: Comparaison de la composante radiale, de la composante tangentielle, et de la méthode par élimination, dans un environnement présentant peu d'amers. On constate sur le nuage de point que la méthode par élimination a une très bonne composante centripète. Le problème est qu'elle est obtenue au détriment de l'évitement d'obstacle. Ainsi, l'animat partant du nord d'un des amers centraux va le heurter sur son chemin ce qui réduit la surface de captation. La méthode par élimination est plus précise lorsqu'il y a plus d'amers.

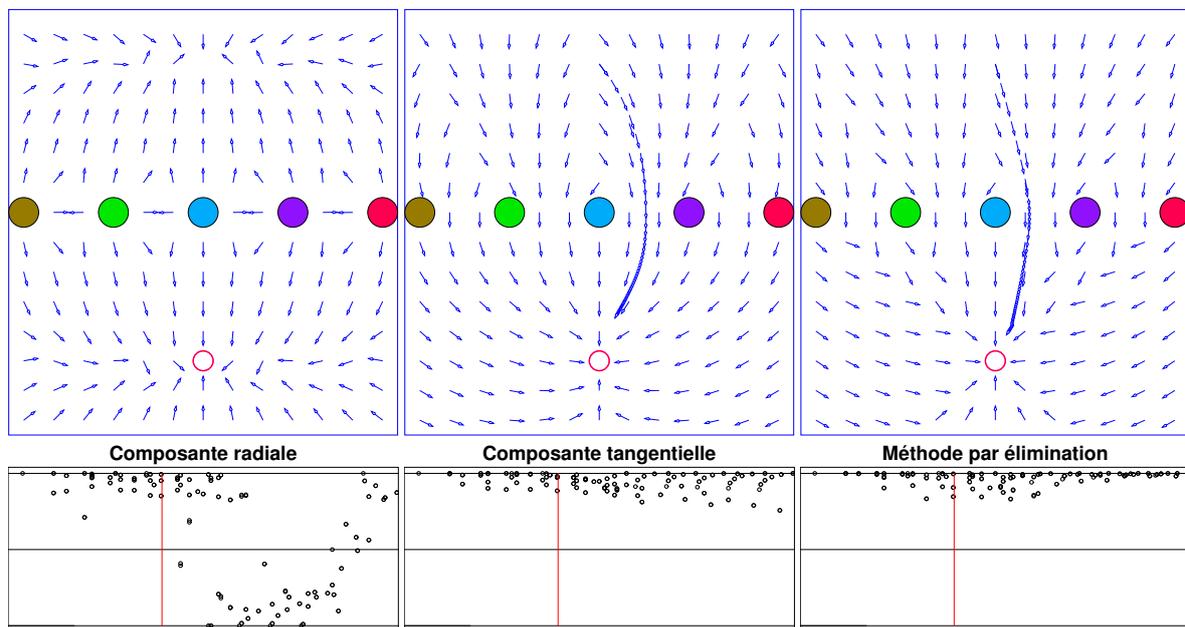


FIG. 5.8: Comparaison de la composante radiale, de la composante tangentielle, et de la méthode par élimination, dans un environnement présentant une symétrie. On constate un comportement similaire à la méthode tangentielle, ce qui est normal puisque les deux méthodes utilisent les parallaxes. On constate sur le nuage de point que la méthode par élimination a une très bonne composante centripète, particulièrement loin du but. Cela n'est pas surprenant car les amers étant plus serrés dans le panorama, l'intervalle de directions non éliminées est plus petit.

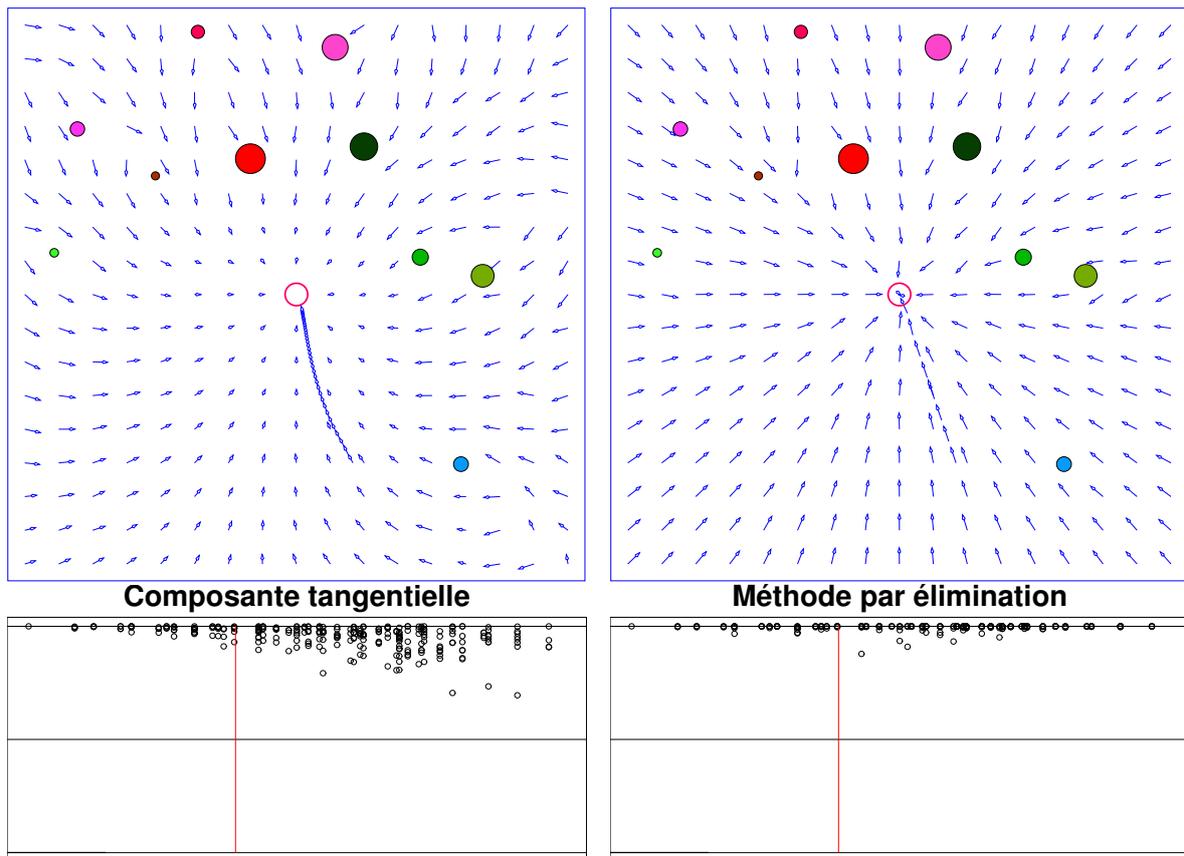


FIG. 5.9: Comparaison entre la composante tangentielle et la méthode par élimination dans un monde à 10 amers. La capacité de la méthode par élimination à estimer précisément la direction du but apparaît nettement, à la fois sur le champ de vecteurs visiblement dirigés vers le but presque partout et sur le nuage de points, beaucoup plus concentré dans des valeurs proches de 1, surtout loin du but. Le manque d'évitement d'obstacle apparaît particulièrement en bas à droite des deux schémas : la composante tangentielle contourne l'amer isolé, pas la méthode par élimination.

5.3.5 Conclusion sur la méthode par élimination

La méthode par élimination pour le calcul de la direction à suivre est conçue pour apporter une réponse plus précise et plus garantie (sous certaines hypothèses) à la question « Compte tenu des informations angulaires extraites du panorama visible actuellement et du panorama mémorisé depuis le but, dans quelle direction se trouve le but ? ».

Sa précision, si les hypothèses sont vérifiées, est en effet meilleure que celle des méthodes fondées sur une somme vectorielle ; elle est capable de détecter certaines anomalies et d'indiquer l'amer qui en est responsable.

La méthode par élimination a une meilleure performance dans les mesures analytiques (cf. section 2.1.6) que les méthodes par somme vectorielle qui, de leur côté, se rattrapent dans les mesures synthétiques. En déterminant de façon plus précise la direction du but, notre méthode a un potentiel intéressant, que nous n'avons pu toutefois développer jusqu'au bout. Par exemple, nous ne maîtrisons pas complètement le cas où la présence d'un ou plusieurs obstacles, ou une désorientation, aboutissent à pénaliser toutes les directions. Dans ce dernier cas, certaines expériences simulées nous ont parfois surpris par l'importance de l'erreur dans le résultat.

Nous estimons que cette méthode peut devenir très intéressante mais demande à être mieux maîtrisée, ce qui implique probablement une modification de la fonction $p(\alpha, \theta_i^0, \theta_i)$. Considérant les problèmes pratiques qu'une expérience robotique supplémentaire implique, et les contraintes de temps correspondantes, nous n'avons pas testé ce modèle sur un robot réel.

5.4 Considérations sur la longueur des pas à effectuer

Une limitation des modèles utilisés jusque-là concerne la non-connaissance par l'animat de la distance aux objets environnants. Cela n'a pas d'incidence quant on aborde ces modèles d'un point de vue mathématique, lorsque la trajectoire de l'animat est la concaténation de segments de longueur infinitésimale dont l'orientation correspond à la direction à suivre calculée par le modèle. Nous appellerons cela le cas continu.

En pratique, cependant, que ce soit en simulation ou sur un robot réel, on fait appel à des segments de longueur finie. Les écarts d'angles mesurés par les diverses méthodes sont homogènes à des radians (c'est-à-dire, en fait, sans dimension), et non pas à des longueurs. Plus concrètement, cela signifie que l'animat observe des écarts angulaires mais ne dispose d'aucune indication sur la distance à parcourir (pour un robot, nombre de tours de roues) qui réduira ces écarts angulaires de façon convenable.

Si des mesures sont possibles en continu pendant que l'animat avance, cela ne pose pas de problème réel : l'animat peut avancer à une vitesse constante convenable et ajuster la direction suivie en fonction de celle prescrite par le modèle en permanence. On se ramène au cas continu.

En revanche, si l'animat doit s'arrêter après chaque pas pour prendre une vue, cela implique qu'il est nécessaire d'introduire un gain permettant de passer d'un angle à une longueur. Nous appellerons

cela le cas discret.

Le gain est homogène à une longueur par unité d'angle, soit des mètres par radian, ou plus simplement des mètres. Nous appellerons γ le gain, et quand il y a distinction entre le gain associé à la composante tangentielle et à la composante radiale, nous les appellerons respectivement $\gamma_{tangentielle}$ et γ_{radial} .

Nous souhaitons que l'animat puisse déterminer de façon autonome le gain au lieu que l'expérimentateur doive le fixer. Pour cela, nous allons analyser de façon théorique les cas simples faisant intervenir un seul amer. Puis nous en déduirons des algorithmes qui réalisent au mieux cette fonction.

5.4.1 Étude théorique du gain

Dans cette section, nous posons les équations qui permettraient à un animat ayant accès à toutes les variables de calculer le chemin optimal. Ainsi, même si l'animat n'a en réalité accès qu'à des angles, nous espérons trouver des indications qualitatives permettant d'exploiter au mieux l'information disponible.

5.4.1.1 Composante tangentielle

Nous considérons le calcul d'un vecteur individuel dans le cas de la composante tangentielle. L'animat omniscient observe un amer à une distance r , un azimut θ . Dans la vue mémorisée, la distance était r_0 , l'azimut θ^0 , voir figure 5.10.

L'application du modèle PV avec la composante tangentielle stipule que, si l'animat observe une parallaxe $\theta - \theta_0$, il ajoute un vecteur orthogonal à la direction menant à l'amer, de norme $\gamma(\theta - \theta_0)$. S'il n'y a qu'un amer, l'animat effectue un pas de même longueur.

L'animat omniscient peut appliquer le théorème d'Al-Kashi et calculer la longueur du segment reliant la position actuelle au but :

$$d = \sqrt{r_0^2 + r^2 + 2r_0r \cos(\theta - \theta_0)} \quad (5.3)$$

Mais cette équation n'est pas facile à exploiter. L'animat réel, lui, peut au mieux utiliser la « méthode directe » définie en section 2.3.1.3. Dans le cas continu, en ne tenant compte que de cet amer, sa trajectoire sera un arc de cercle de rayon r centré sur l'amer. La longueur de cet arc est :

$$l_{arc} = r(\theta - \theta^0)$$

Dans le cas discret, on peut annuler la parallaxe en faisant un seul pas de longueur :

$$l = r \cdot \tan(\theta - \theta_0)$$

Ce n'est pas très satisfaisant : outre que cela fait toujours intervenir la distance à l'amer, si la parallaxe est proche d'un demi-tour, cela aboutit à faire un seul très grand pas, alors que la bonne action à accomplir est probablement de tourner autour de l'amer.

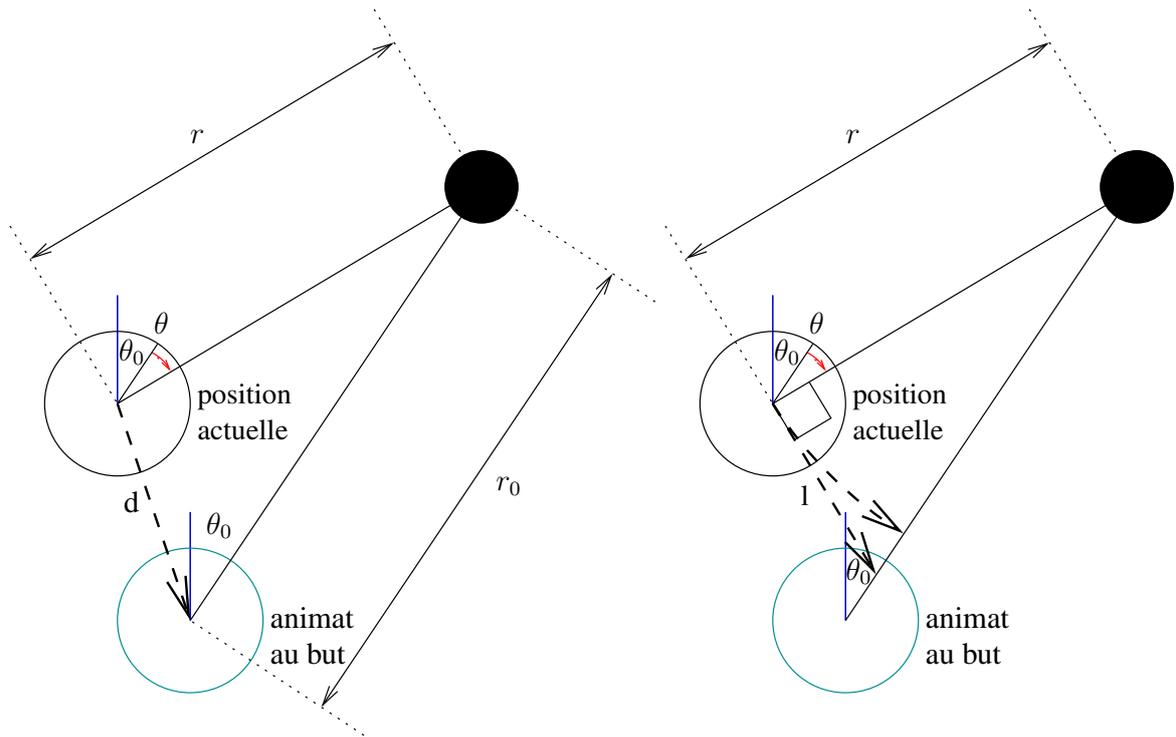


FIG. 5.10: Calcul du gain théorique dans la composante tangentielle. Un animat observe un obstacle d'azimut actuel θ , qui valait θ^0 depuis le but. On cherche une relation entre la longueur d'une trajectoire qui permet de rejoindre le but et la parallaxe $\theta - \theta^0$. Si on la trouve, cette relation permet de calculer le gain optimal de la composante tangentielle. À gauche, la trajectoire optimale en un pas (tirets épais). Elle suppose de connaître les deux distances. À droite (tirets épais), l'arc de cercle que suit l'animat en appliquant de façon continue la composante tangentielle, et le segment de droite qu'il parcourt en un pas dans le cas discret. Dans les deux cas, les équations (voir texte) montrent que le gain fait intervenir les distances de l'animat à l'amer, qui ne sont pas accessibles à l'animat dans notre modèle. Nous concluons que l'animat ne peut calculer directement le gain à partir de la connaissance de deux panoramas.

Dans le cas des petits angles, pour lesquels $\tan(\alpha) \simeq \alpha$, l'expression peut se simplifier et le gain théorique en présence d'un amer devient simplement $\gamma = r$.

La présence d'un amer supplémentaire apporte une nouvelle information d'angle, mais aussi une nouvelle inconnue de distance. Rappelons que la méthode d'agrégation de l'information utilisée par le modèle PV est une somme vectorielle, dont il est difficile de prévoir a priori la longueur et la direction avec peu d'informations. Nous retenons donc qualitativement qu'il est probablement inutile de chercher à compenser en un seul pas de grandes parallaxes.

Il paraît raisonnable de prendre pour gain $\gamma = k \cdot r$, où k est la proportion de la distance théorique restant à parcourir pour annuler l'écart. $k = 0.1$ à $k = 0.25$ paraît raisonnable. En effet, il vaut mieux ne tenter à chaque pas de corriger qu'une partie des écarts et avoir une trajectoire régulière que d'essayer d'approximer en un segment de droite de même longueur une trajectoire qui, de toute façon, est généralement courbe. Toutefois, r n'étant pas connu de l'animat, l'idée n'est pas applicable telle quelle.

Cette brève analyse nous a donné quelques éléments qualitatifs et montré que le gain de la composante tangentielle, si on veut le connaître directement, fait essentiellement intervenir la distance à l'amer. Nous ne chercherons donc pas à calculer directement le gain à partir de la connaissance du panorama du but et du panorama actuel.

5.4.1.2 Composante radiale

Considérons la situation représentée en figure 5.11. L'animat a mesuré les diamètres apparents α et α_0 .

L'animat, n'ayant accès qu'aux diamètres apparents, peut au mieux se replacer à bonne distance, en faisant coïncider le diamètre apparent actuel avec celui mémorisé. En revanche, il ne peut pas corriger l'azimut (ce qui est du ressort de la composante tangentielle).

Pour corriger l'écart, il doit avancer de

$$r - r_0 = d/\alpha - d/\alpha_0 = d \cdot (1/\alpha - 1/\alpha_0)$$

Il est intéressant de remarquer que la longueur qui intervient dans la composante radiale est d , diamètre de l'amer, et non la distance à l'amer comme dans la composante tangentielle. La grandeur qui est accessible à l'animat est $(1/\alpha - 1/\alpha_0)$.

La grandeur qui joue le rôle de gain est donc d le diamètre de l'amer. Le gain à appliquer pour la composante radiale n'est donc pas lié à la distance qui sépare l'animat des amers, mais à la taille des amers eux-mêmes.

L'intervention des termes en $1/\alpha$ nous amène à une remarque : tous les travaux descendant des modèles CC considèrent les écarts de diamètre apparent pour calculer les vecteurs individuel : $\alpha - \alpha_0$. Or cette brève analyse montre qu'il est plus rigoureux, pour estimer de combien on doit s'approcher

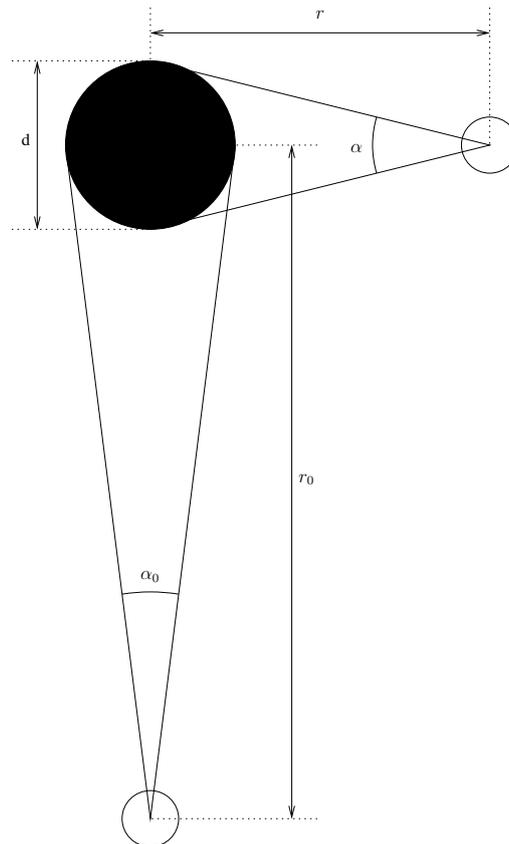


FIG. 5.11: Calcul du gain théorique dans la composante radiale. Un animat observe un obstacle de largeur frontale d , de diamètre apparent α , alors qu'il avait l'angle apparent α_0 depuis le but. On cherche une relation entre la longueur d'une trajectoire qui permet de se retrouver à bonne distance du but et la variation du diamètre apparent $\alpha - \alpha_0$. Si on la trouve, elle permet de calculer le gain optimal de la composante radiale. Comme dans le cas de la composante tangentielle, l'animat ne peut calculer directement le gain à partir de la connaissance de deux panoramas. Il est intéressant de remarquer toutefois que la longueur du trajet n'est pas proportionnelle à la variation du diamètre apparent mais à la variation de l'inverse du diamètre apparent $\frac{1}{\alpha} - \frac{1}{\alpha_0}$.

ou s'éloigner d'un amer, de considérer la différence des inverses des diamètres apparents des amers, au lieu de la différence entre les diamètres apparents.

Le souci avec l'utilisation de la différence des inverses est que, si un amer est partiellement occulté, ce qui est un cas fréquent, son diamètre apparent va se réduire pour pouvoir même atteindre presque zéro et, donc $1/\alpha$ peut devenir très grand. Comme si l'amer semblait tout d'un coup extrêmement loin, le vecteur individuel vers cet amer va être très grand. La conséquence pratique est que l'animat se jetterait sur les amers partiellement occultés et donc, en fait, heurterait souvent le bord de l'amer occultant.

Nous nous trouvons donc dans une situation où une méthode potentiellement plus fidèle s'avère beaucoup moins bonne quand elle est perturbée. Finalement, utiliser l'écart des angles est une bonne chose car cela évite la singularité qui apparaît quand le diamètre apparent d'un amer se réduit.

Cette brève analyse nous permet donc de prédire en théorie quel gain choisir pour la composante radiale, à condition d'appliquer une version plus rigoureuse de cette composante qui fasse intervenir la différence des inverses des diamètres apparents. Ce gain est proportionnel, non pas à la distance aux amers comme pour la composante tangentielle, mais à la taille des amers. Toutefois, nous pensons qu'il est raisonnable en pratique de continuer d'utiliser la différence des diamètres apparents.

Dans ce cas, l'analyse que nous avons faite ne nous donne finalement pas d'indication sur la valeur du gain à appliquer, sauf à introduire une « largeur apparente typique », un paramètre supplémentaire qui permettrait de mettre en correspondance des valeurs attendues de α et de $1/\alpha$. Comme nous verrons pas la suite, nous avons préféré à la place nous tourner vers des méthodes pragmatiques.

5.4.1.3 Conclusion de l'analyse théorique

Cette analyse des composantes radiale et tangentielle explique pourquoi nous avons constaté lors de nos expériences préliminaires que la composante radiale nécessitait des gains différents de ceux de la composante tangentielle. Ils sont en fait décorrélés l'un de l'autre puisque le gain à appliquer à la composante tangentielle dépend essentiellement de la distance de l'animat aux amers, alors que le gain à appliquer à la composante radiale dépend essentiellement de la taille des amers, ce qui n'était pas évident a priori.

Il est intéressant, quoique un peu anecdotique, de se demander si l'on peut en déduire quelque chose de pratique pour un promeneur humain face à une scène. Considérons un promeneur en forêt qui observe des troncs d'arbres à diverses distances. Un mouvement latéral (translater la tête de gauche à droite à orientation constante) permet de se faire directement plutôt une idée de la distance aux troncs, alors que s'approcher ou s'éloigner d'un tronc particulier, même si on ne peut maintenir son orientation au cours de l'opération, permet de se faire une idée plutôt de sa taille. Bien sûr, dans les deux cas, le diamètre apparent permet de faire un lien entre distance et taille, mais il n'était pas évident a priori que le simple fait de choisir de se déplacer latéralement ou de s'approcher donnait des informations différentes.

5.4.2 Agrégation des données de plusieurs amers

Quand plusieurs amers sont disponibles, les méthodes CC calculent traditionnellement la somme des vecteurs individuels, qui grandit linéairement avec le nombre d'amers appariés. Cela implique que, toutes choses égales par ailleurs, l'animat va plus vite dans un environnement où il y a beaucoup d'amers, ce qui n'est pas pertinent. Nous recommandons à la place de faire la moyenne des vecteurs individuels au lieu de la somme. Ceci rend la taille des pas indépendante du nombre d'amers.

5.4.3 Algorithme pour déterminer en ligne les gains à appliquer

Compte tenu du fait que l'animat ne peut déterminer les gains à appliquer aux composantes radiale et tangentielle nous avons eu recours à des méthodes itératives qui tentent d'obtenir par itérations successives des valeurs de gain raisonnables.

5.4.3.1 Solution pragmatique figée

Une première solution est le réglage manuel du gain « en dur » dans le simulateur ou le programme contrôlant le robot. L'expérimentateur doit régler le gain en fonction de l'environnement, le plus simple étant par essai et erreur. Cette solution a l'avantage de la simplicité. Son inconvénient est qu'elle n'est efficace que dans un type d'environnement donné. Si le robot se trouve dans un environnement beaucoup plus grand avec le même gain, il perdra beaucoup de temps à faire des petits pas. Si l'animat se trouve dans un environnement beaucoup plus petit avec le même gain, il va se heurter aux obstacles car il fera des pas trop grands.

5.4.3.2 Solution adaptative : ralentir sur les courbes

L'idée principale de la solution adaptative que nous avons conçue est de trouver un critère pragmatique pour déterminer quelle est une longueur de pas raisonnable par ajustements successifs.

Au début d'une trajectoire, nous partons d'un gain qui produit une longueur de pas raisonnable, comme par exemple la longueur du robot. Nous prenons pour critère de changement de taille de pas la régularité de la trajectoire. L'idée est que si l'animat va toujours tout droit à chaque pas, il est probablement possible d'augmenter la taille du pas pour aller plus vite. En revanche, si la trajectoire fait des zigzags, il semble prudent de réduire la taille du pas pour régulariser la trajectoire.

Plus formellement, nous mesurons et désignons par α l'angle entre deux segments successifs de la trajectoire de l'animat. Nous appelons cette valeur la *tortuosité* locale. En effet, $\alpha = 0$ à un instant donné correspond à faire deux pas dans la même direction exactement, alors que plus α est grand, plus le trajet est tortueux. Dans le cas extrême, $\alpha = 180^\circ$, l'animat fait exactement un demi-tour entre un pas et le suivant.

Nous souhaitons que le gain augmente pour une trajectoire droite, et diminue en cas de trajectoire localement tortueuse. Nous définissons l'angle neutre α_n comme étant l'angle limite tel que, si l'ani-

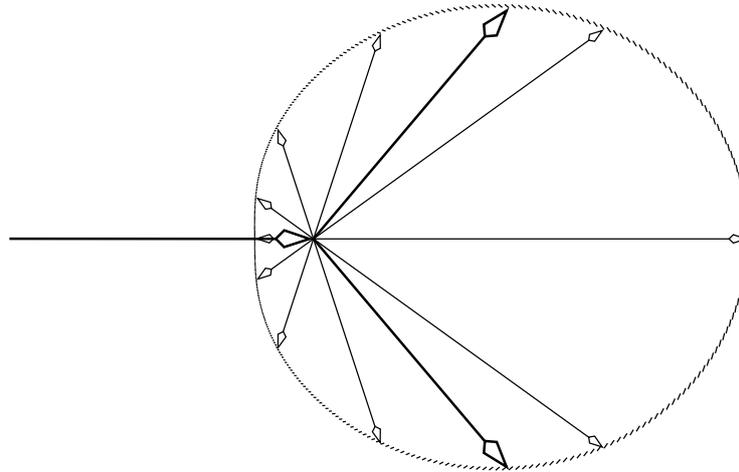


FIG. 5.12: Ajustement automatique du gain en fonction de la régularité de la trajectoire. La flèche à gauche représente un déplacement. De sa pointe partent d'autres flèches correspondant à des virages d'angles variés. Leur longueur est proportionnelle au gain appliqué quand l'animat fait un virage de l'angle représenté. L'idée principale est que, si l'animat avance plutôt en ligne droite, le gain doit croître. Au contraire, si l'animat fait des virages à chaque pas, le gain doit décroître. On modifie le gain à chaque pas d'un facteur dépendant de l'angle entre un pas et le suivant. La courbe représente le lieu des sommets des flèches. Les flèches épaisses correspondent à l'angle neutre choisi (ici 50°) qui, par définition, aboutit à un gain de 1.

mat tourne exactement de cet angle, alors le gain ne variera pas. À chaque pas que fait l'animat, nous multiplions le gain par le facteur suivant :

$$\exp(\cos(\alpha) - \cos(\alpha_n))$$

Ainsi, nous prédisons, et observons effectivement en simulation, que la trajectoire s'adapte bien à la taille d'un environnement donné sans avoir besoin d'introduire une pré-connaissance de l'environnement.

Nous avons choisi de prendre $\alpha_n = 50^\circ$ après quelques essais simulés. Ce choix permet un facteur d'amplification maximum de $\exp(1 - \cos(\alpha_n)) = 1.43$, et un facteur minimum de $\exp(-1 - \cos(\alpha_n)) = 0.19$. C'est le cas représenté en figure 5.12. Ainsi, la longueur des pas croît exponentiellement assez vite si la trajectoire est très rectiligne, mais peut décroître exponentiellement très vite lorsque c'est nécessaire.

Si on choisit un angle neutre trop faible, l'animat a trop tendance à freiner. Notamment, quand l'animat est presque arrivé au but et fait des petits ajustements, ceux-ci ont des directions variées et tendent à faire s'arrêter l'animat en un point non optimal alors qu'une meilleure précision est encore possible. Si on choisit un angle neutre trop grand, le gain devient vite trop grand, la trajectoire devient chaotique et la performance moindre.

On peut se donner une distance minimum à parcourir, par exemple une fraction de la taille de l'animat, pour éviter de le voir piétiner surtout en fin de trajectoire.

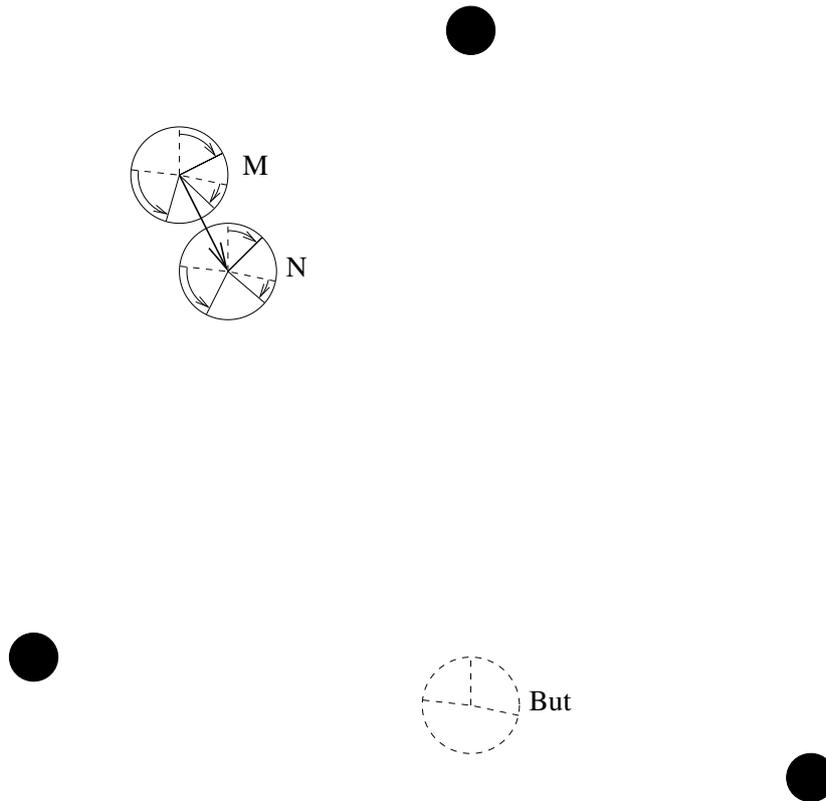


FIG. 5.13: Ajustement automatique du gain. Après chaque pas, l'animat peut comparer les nouvelles (en N) parallaxes (flèches courbes) avec les anciennes (en M). Il peut ainsi estimer la proportion de parallaxes qui a été réduite suite à ce déplacement. S'il constate que la proportion est trop faible (l'animat fait des pas trop courts, ou bien les amers sont tous très lointains), il peut augmenter le gain pour faire de plus grands pas. Inversement, s'il constate que la proportion est trop forte (par exemple, au moins une parallaxe a changé de sens au cours d'un déplacement), il est bon de diminuer le gain.

Nous avons utilisé avec succès cette solution adaptative en simulation. Elle a un défaut toutefois : que la valeur du gain ne converge pas. Le gain est toujours remis en question et fluctuant. Par ailleurs, nous avons trouvé que la marge était faible entre des valeurs de l'angle neutre trop petites et trop grandes. Nous ne sommes pas convaincu que cette méthode fonctionnerait bien dans d'autres environnements.

5.4.3.3 Ajuster la proportion de réduction des parallaxes

Cette solution utilise une idée différente : mesurer l'effet de son déplacement sur les parallaxes, et ajuster la taille des pas pour que cet effet soit dans une proportion adéquate.

L'animat fait un pas dans la direction indiquée par le modèle. Il considère pour chaque amer i la proportion f_i de la parallaxe $D_i^{But \rightarrow M}$ qui a été réduite en faisant ce pas. On ne tient pas compte dans ce calcul des amers i pour lesquels la parallaxe $D_i^{But \rightarrow M}$ n'est pas assez grande devant les incertitudes

de mesure.

$$f_i = -\frac{D_i^{M \rightarrow N}}{D_i^{But \rightarrow M}}$$

Cette proportion peut aller de 0 si la parallaxe n'a pas changé (le pas était beaucoup trop petit pour mesurer une modification, ou bien le pas était dans la direction de l'objet ou dans la direction opposée) à 1 (on est directement arrivé à annuler cette parallaxe), voire plus (on a dépassé en un pas l'écart qu'on devait corriger, il y a donc un écart dans l'autre sens désormais). On considère la plus grande des proportions f_{max} , car elle correspond (si on fait l'hypothèse d'un monde statique) à un objet proche, comme dans la stratégie d'évitement d'obstacle par équilibrage du flux optique expérimentée par Duchon [DUC 96].

Alors on augmente ou on diminue γ de façon à rapprocher f de la proportion k souhaitée. On peut par exemple multiplier γ par k/f , ou bien appliquer $\gamma(t+1) = \gamma(t)^{0.9} * (k/f)^{0.1}$ pour lisser un peu les variations qui peuvent apparaître. Ainsi on ajuste la taille des pas de sorte que chaque pas tende à corriger une proportion k constante de la parallaxe qui varie le plus vite à ce moment. Ainsi, l'animat avance vite quand il est loin du but et de tout obstacle ; et il ralentit quand il passe près d'un obstacle, ou arrive près du but.

Cette méthode a l'avantage de ne pas dépendre de la forme de la trajectoire mais uniquement de l'effet observé visuellement. Nous pensons qu'elle est plus stable et moins sensible à la valeur précise du paramètre.

Par ailleurs, alors que cette section cherchait à résoudre un problème lié au cas discret, cette méthode est applicable et nous semble intéressante pour le cas continu.

5.4.3.4 Conclusion sur le gain

Le problème que nous cherchons à résoudre dans cette section ne se pose véritablement que si un animat évolue dans des environnements où sa distance aux amers varie beaucoup au cours du temps. Sinon, un gain constant ad-hoc suffit, et c'est ce dernier cas qui nous concernait dans nos expériences.

La première méthode (gain fixe) permet à la navigation de fonctionner pour un environnement donné, à condition que l'expérimentateur règle le gain.

La seconde apporte l'indépendance par rapport à la taille de l'environnement. Elle remplace toutefois la recherche du paramètre gain fixé par le réglage d'un autre paramètre, l'angle neutre.

La dernière méthode, qui tente d'ajuster la proportion de parallaxes corrigée en un pas, nous semble la plus intéressante. Toutefois, nous n'avons pu la tester faute de temps.

5.5 Critère d'arrêt : comment l'animat sait-il quand il est arrivé ?

À la fin d'une trajectoire, un critère permettant à l'animat d'interrompre le cycle prise de panorama – déplacement est nécessaire. De même que pour le calcul du gain, ce problème ne se pose pas véritablement dans de courtes expériences, car l'expérimentateur peut se contenter de donner un nombre fixé de pas. Si ce nombre est trop petit, la trajectoire s'interrompt trop tôt. S'il est trop grand, l'animat piétinera autour du but avant de s'arrêter. C'est cette solution que nous avons choisie dans notre test de navigation avec compas visuel (section 9.6), parce que c'est la plus simple et qu'elle garantit que l'expérience se termine. Mais d'autres sont souhaitables de façon plus générale. Dans toutes nos expériences, un critère de nombre maximal de pas était imposé, assez large, complété ou non par un autre.

Un critère externe est possible. Par exemple, dans les expériences de retour d'un robot réel à une station de recharge que nous présenterons en section 9.2, nous avons utilisé le fait que le robot était directionnel (c.f. 4.7) et n'avait pas d'objet proche derrière lui pendant sa trajectoire, sauf quand il arrivait au but. Il utilisait alors les sonars arrière après chaque déplacement pour détecter s'il était arrivé au but. Un tel critère externe reste spécifique à une situation.

Un critère simple, mais pas toujours satisfaisant, est la norme du vecteur indiquant la direction à prendre pour rejoindre le but, dans le modèle PV. Bien qu'apparemment très satisfaisante, cette information ne doit pas être utilisée seule car, si l'animat arrive sur un mauvais attracteur dû à une défaillance de la méthode (par exemple mauvais appariement ou erreur sur l'orientation courante, ce dernier cas a été illustré en section 3.2), ce vecteur peut s'annuler et pourtant l'animat ne sera pas arrivé au but.

On peut penser à un critère fondé sur la ressemblance de la vue actuelle avec la vue mémorisée. Par exemple, énumérer les parallaxes observées pour extraire la plus grande en valeur absolue. Si elle est inférieure à un seuil, on estime que l'animat est arrivé. La difficulté en pratique avec ce genre de critère est qu'il faut choisir un seuil. La valeur choisie sera typiquement quelques degrés. Si on prend un seuil trop grand, l'animat s'arrêtera trop tôt et la précision sera mauvaise. Si on prend un seuil trop petit, l'animat piétinera et n'estimera jamais être arrivé. Ce critère est sensible aux erreurs d'orientation : le seuil doit être supérieur aux erreurs d'orientation qui risquent de frapper l'animat.

On peut utiliser un critère d'arrêt lié à la trajectoire, comme un seuil de tortuosité locale. En termes plus concrets, si l'angle entre deux éléments de trajectoires est proche d'un demi-tour, c'est que l'animat vient probablement de dépasser le but. Cette méthode a été utilisée par Röfer. Elle est applicable si la trajectoire ne fait pas de zig-zag ailleurs que près du but, ce qui est normalement le cas.

Un bon critère d'arrêt combine une information de type ressemblance de vues (seuil sur la plus grande des parallaxes observées) avec une information en rapport avec la trajectoire (norme du vecteur indiquant la direction à suivre, ou bien seuil de tortuosité). En effet, si le second critère indique un arrêt, mais que le premier indique sans ambiguïté que l'animat n'est pas au but, alors l'animat sait qu'il est coincé dans un attracteur indésirable. Ce genre d'association de critère peut devenir important

dans des expériences où l'on donne de plus en plus d'autonomie à l'animat. En permettant à l'animat de détecter le cas où la méthode de navigation a échoué, on lui donne la possibilité de prendre une décision, un changement de stratégie.

Le critère que nous avons conservé, notamment pour les expériences de construction de carte est une combinaison d'un nombre maximal absolu de cycles, d'un critère de parallaxe maximale, et d'un petit délai accordé à l'animat pour qu'il affine la trajectoire.

- Il y a un nombre limité de cycles qu'on ne peut en aucun cas dépasser. Il est fixé généralement à 20 ou 50. Si la trajectoire est interrompue à cause de ce critère, l'animat considère le trajet comme un échec à rejoindre le point voulu ;
- Si la qualité de guidage (cf. section 3.4.2.1) est inférieure à 0.5, on continue la trajectoire et on annule un éventuel compte à rebours ;
- Si la parallaxe maximale à un instant donné est inférieure à 8 degrés, on démarre un compte à rebours qui arrêtera la trajectoire au bout de 5 pas ;
- Si la parallaxe maximale à un instant donné est inférieure à 4 degrés, on démarre un compte à rebours qui arrêtera la trajectoire au bout de 3 pas ;
- Si un compte à rebours atteint 0, la trajectoire est arrêtée et le point visé réputé atteint.

Cette combinaison a l'avantage de toujours finir, n'arrête pas la trajectoire trop tôt – ce qui serait nuisible à la précision – et évite à l'animat de piétiner à l'arrivée.

5.6 Conclusion

Nous avons présenté les unes après les autres nos contributions à la question de la navigation locale par panorama dans la cas où l'orientation est connue.

Nous avons choisi pour notre animat l'utilisation de panoramas segmentés, qui sont plus légers à traiter car plus compacts que les panoramas échantillonnés. Le système visuel utilisé (que nous détaillerons dans la prochaine partie dédiée à l'expérimentation) fournit la couleur dans l'espace luminance, teinte, saturation, ce qui facilite la segmentation.

Nous utilisons une méthode de programmation dynamique qui exploite les informations distinguant les amers, et dont les propriétés permettent d'apparier des panoramas d'orientations variées, voire inconnues.

Une fois l'appariement effectué, et si l'orientation relative des deux panoramas est connue, l'animat peut comparer les azimuts des amers dans les deux vues, et estimer la direction à suivre pour passer du point actuel au but en utilisant notre méthode par élimination. Cette méthode est plus précise que les méthodes fondées sur une somme vectorielle et permet de vérifier la cohérence des informations obtenues. Sous certaines hypothèses, elle indique aussi un intervalle de direction approchant avec certitude l'animat du but.

Nous avons montré que l'animat ne peut déterminer a priori le gain des composantes tangentielles ou radiales à appliquer pour éviter une disproportion entre les mouvements commandés et la taille de

l'environnement. En revanche, nous avons proposé deux méthodes pour ajuster le gain sans connaissance préalable de la taille de l'environnement. Nous avons aussi évoqué la nécessité pour l'animat de déterminer quand arrêter la trajectoire. Nous avons utilisé une méthode composite qui donne un bon compromis entre précision et nombre de pas.

Revenons sur la problématique d'apparier correctement des panoramas d'orientation arbitraire voire inconnue, car elle est importante. Nous avons vu que l'animat a besoin de connaître son orientation pour naviguer efficacement. S'il a aussi besoin de connaître son orientation pour pouvoir réaliser des appariements, il est en mauvaise posture dès qu'il est désorienté. Au contraire, s'il est capable, même désorienté, de reconnaître les environs grâce à une méthode d'appariement suffisamment performante, l'information perçue va lui permettre de retrouver son orientation. Nous ne connaissons pas de méthode de navigation par mesure d'angles dans des panoramas segmentés, et préalable à la nôtre, qui en soit capable, en particulier sur un robot réel.

Nous arrivons maintenant, pour continuer notre exposé, à la présentation de notre méthode d'estimation de l'orientation.

Chapitre 6

Le compas visuel

Ce chapitre aborde maintenant notre contribution personnelle au problème de l'estimation de son orientation par un robot mobile.

Nous présentons dans ce chapitre deux méthodes. Elles ont en commun des hypothèses relativement faibles : pour l'orientation locale dans une zone, elles ont besoin de trois vues correctement orientées et non alignées prises dans la zone. Ces vues servent à extraire une information par un calcul à faire une seule fois. Ensuite, l'une comme l'autre méthode permettent d'estimer l'orientation de toute vue prise dans la zone.

La première méthode, que nous présentons dans la première section de ce chapitre, est simple et approximative mais nous démontrerons théoriquement et expérimentalement qu'elle est suffisante dans certains cas pour faire fonctionner les modèles de navigation utilisant la composante tangentielle.

La seconde est plus élaborée et plus précise. Elle peut être considérée comme un raffinement de la première, utilisant des relations linéaires entre azimuts. Nous avons étudié comme l'étendre en combinant plusieurs zones de proche en proche. C'est cette seconde méthode que nous avons appelé le « compas visuel ».

À notre connaissance c'est la première fois qu'une telle méthode d'orientation exploitant des relations linéaires entre azimuts dans des panoramas circulaires segmentés est utilisée.

Nous précisons que le compas visuel n'a pas besoin, pour estimer l'orientation de l'animat, de connaître la position de ce dernier par rapport à un repère lié à l'environnement, ni de connaître celle des amers utilisés, ni même de connaître les positions relatives des trois vues sur lesquelles il s'appuie. Il ne vise pas à reconstituer toutes ces informations à partir des vues, mais seulement à déterminer l'orientation de l'animat.

6.1 Trouver séparément l'orientation de l'animat

Nous avons vu précédemment que les modèles CC et la plupart des modèles qui en découlent supposent que l'animat connaît la direction à laquelle il faisait face lors de chaque acquisition de panorama. L'algorithme Panama, l'algorithme de déformation, ainsi que la méthode par minimisation de Nelson et Aloimonos, résolvent le problème en modifiant l'algorithme qui décide de la direction à suivre de sorte qu'il trouve l'orientation actuelle au cours de la même étape.

En ce qui nous concerne, nous conservons les modèles de navigation par mesures d'angles comme base intéressante pour le cas où la direction de l'animat est connue. Dans le nouveau cas où l'orientation de l'animat n'est plus connue, le compas visuel permet de retrouver l'orientation de façon indépendante de la position. Alors, l'animat se retrouve dans le cas précédent d'orientation connue et nous pouvons à nouveau appliquer les méthodes qui ont fait leurs preuves jusque-là.

6.2 La méthode de l'étoile polaire

La méthode de l'étoile polaire est une méthode « naïve » d'orientation. Son intérêt est néanmoins triple :

- méthode suffisante pour naviguer avec la composante tangentielle dans certains cas ;
- aide à la compréhension de la description du compas visuel ;
- méthode « témoin » montrant l'intérêt apporté par la méthode du compas visuel, notamment lors du test robotique.

6.2.1 Définition de la méthode d'orientation par étoile polaire

Le principe de la méthode d'orientation par l'étoile polaire est très simple. Son nom provient de l'étoile qui, située presque exactement au pôle nord céleste, reste toujours à la même place dans la sphère céleste et permet à celui qui l'observe de s'orienter simplement.

6.2.1.1 Principe, dans un cas idéal

Considérons notre animat. S'il remarque que dans son environnement se trouve un élément qui semble être dans une direction fixe, il lui devient très facile de retrouver sa propre orientation. Il lui suffit de se souvenir de l'élément en question et d'un contexte qui permet de le reconnaître. Appelons cet élément « étoile polaire », que nous abrégerons parfois en « étoile ».

Alors, si l'animat souhaite réestimer son orientation, il cherchera l'étoile dans son environnement actuel. S'il la trouve, elle lui indiquera une direction fixe.

L'animat pouvait déjà avoir une estimation de son orientation, et donc une direction dans laquelle il s'attendait à trouver l'étoile. Si celle-ci est visible dans la direction attendue, l'animat est déjà bien

orienté. Si elle semble décalée par rapport à la direction attendue, le décalage mesurable est l'écart entre l'orientation que l'animat pensait avoir et celle qu'il avait réellement.

Enfin, si l'étoile est invisible, l'animat ne peut évidemment l'utiliser pour se réorienter.

6.2.1.2 Cas réel et implémentation

Pour fonctionner dans un cas réel, l'algorithme sélectionne comme étoile polaire l'amer qui semble le moins bouger dans le panorama quand l'animat se déplace. Précisons comment.

L'animat prend trois vues d'orientations connues, depuis des points formant un triangle dont le plus grand angle fait au maximum 135° . Cette condition est facile à obtenir : en terme concrets, les points ne doivent pas être alignés ou presque. Il réalise trois appariements, un entre chaque couple de vues, ce qui lui permet de calculer trois parallaxes pour chaque amer. Il est bon de filtrer les appariements par cohérence (voir section 6.4.3.2), pour éliminer les amers peu fiables.

L'animat calcule pour chaque amer apparaissant dans les trois vues l'écart-type de sa parallaxe. L'amer associé à l'écart-type minimal est sélectionné comme étoile.

Expliquons pourquoi il y a une contrainte sur la forme du triangle des points de vues. Si les trois points de vue sont presque alignés, les mouvements de l'animat lors de la prise des trois vues auront des directions similaires. En conséquence, les amers situés dans le prolongement de l'alignement vont présenter des parallaxes faibles, et l'un d'entre eux sera sélectionné alors que cela n'est pas pertinent. Pour cette raison, il est important que le triangle des trois points de vue ne soit pas aplati. Autrement dit, aucun des trois angles du triangle ne doit être proche de 180° . L'idéal étant un triangle équilatéral (trois angles de 60°), nous estimons très bon que le plus grand des angles atteigne 90° , bon à 120° et acceptable jusqu'à 135° . Le plus facile est que l'animat prenne une vue, avance, prenne une seconde vue, tourne de 90° environ, avance à nouveau de la même longueur environ, puis prenne une troisième vue.

L'utilisation de l'étoile est la même que dans le cas idéal : quand il veut se réorienter, l'animat considèrera cet amer comme exactement fixe. C'est cette façon de procéder qui fait qualifier cette méthode de naïve.

6.2.2 L'étoile polaire suffit pour naviguer dans certains cas

Plaçons nous dans un contexte de retour au nid. Supposons que l'environnement comporte au moins trois amers qui satisfont aussi aux hypothèses de notre démonstration de la section 3.4.2. Alors que nous avons montré que deux amers non alignés avec le but suffisent si l'orientation est connue, nous supposons ici l'existence d'au moins un troisième amer.

L'animat, en quittant le nid, a pris trois vues et choisi son étoile. Il doit maintenant revenir au nid.

Si l'étoile est extrêmement loin, alors la parallaxe de l'étoile due au déplacement de l'animat est nulle, la réorientation est exacte et on se ramène au cas d'orientation connue. La démonstration de la

section 3.4.2 s'applique.

Nous considérons alors le cas où la distance à l'étoile est finie et montrons que la démonstration de la section 3.4.2 s'applique toujours. En s'orientant avec l'hypothèse que l'étoile est dans une direction fixe, l'animat commet une erreur d'orientation égale à la parallaxe de l'étoile. Or cette étoile a été choisie parce que sa parallaxe était la plus petite au moment de l'observation. Supposons qu'au cours de la navigation, cela soit encore le cas. Nous reviendrons sur cette hypothèse un peu plus loin.

Parce que l'erreur d'orientation est plus petite en valeur absolue que toutes les parallaxes, cela signifie que son effet perturbe les parallaxes, mais aucune n'est perturbée au point de changer de signe. La démonstration de la section 3.4.2 est donc applicable. La conclusion est que la méthode de l'étoile polaire suffit, malgré l'erreur qu'elle commet, à garantir une composante centripète positive comme dans le cas d'orientation connue.

Pour achever cette démonstration, il nous faut déterminer des situations dans lesquelles l'étoile choisie conserve bien une parallaxe plus petite.

Le déplacement de l'animat en trois points situés en triangle (i.e. loin d'être alignés) assure qu'un amer ne peut présenter de parallaxe faible que s'il est loin, et pas seulement parce qu'il se trouve aligné avec la direction du déplacement de l'animat. L'amer qui a été choisi est donc plus loin de l'animat que les autres amers. Si lors du retour au nid il conserve cette propriété, alors la démonstration est applicable.

Le résultat de ce raisonnement, et nous ne sommes pas surpris de ce résultat, est qu'il suffit que l'animat soit, au cours du déplacement, plus loin de l'étoile polaire que des autres amers pour que cette méthode de réorientation garantisse une composante centripète positive.

Ainsi, une condition suffisante pour que la méthode de l'étoile polaire soit assez précise pour permettre de naviguer avec la composante tangentielle est que l'animat évolue dans une zone telle que la distance de l'étoile à tout point de la zone soit plus grande que le diamètre de la zone.

6.2.3 Quelques remarques

Quelques remarques simples peuvent être faites. Elles illustrent des points qui réapparaîtront dans les sections suivantes.

L'amer jouant le rôle d'étoile polaire est extérieur à l'animat. La méthode met donc en évidence la nécessité de plusieurs étapes. Premièrement, trouver un amer capable de jouer ce rôle. Deuxièmement, mémoriser le contexte pour pouvoir le reconnaître. Troisièmement, le reconnaître en situation et, en particulier, pouvoir reconnaître son absence.

Pour trouver l'amer en question, l'animat doit dans une première étape se déplacer, observer comment les positions angulaires des amers varient dans le panorama en fonction de ses déplacements propres, puis choisir l'amer le plus stable dans le panorama.

La méthode ne peut donner de bons résultats que si un élément de l'environnement est suffisamment lointain pour conserver une parallaxe plus petite que celle des autres amers, même à l'étape de

retour au nid.

6.2.4 Situation respective de l'étoile polaire et du compas visuel

La méthode de réorientation par étoile polaire fait un peu figure d'« élément nul » des méthodes de réorientation, puisqu'elle suppose qu'un amer donne directement une direction fixe. Elle fonctionne idéalement si un amer bien reconnaissable est extrêmement loin de l'animat.

Le compas visuel peut être vu comme un raffinement de la méthode de l'étoile polaire applicable au cas où aucun amer n'est extrêmement loin.

Au lieu de chercher un amer particulièrement immobile dans le paysage, le compas visuel cherche des relations linéaires entre les parallaxes des différents amers visibles.

Dans le cas où un amer relativement lointain bouge relativement peu, l'approche du compas visuel peut être décrite ainsi : chercher si le mouvement apparent de cet amer peut être mis en relation linéaire simple avec le mouvement apparent d'autres amers. Si une telle relation existe, cela signifie que l'animat peut compenser le mouvement apparent de cet amer à l'aide du mouvement apparent d'autres amers de l'environnement. Ainsi, l'animat peut estimer son orientation de façon plus précise qu'en supposant simplement, et à tort, que l'amer relativement lointain est fixe dans l'environnement.

Comme nous allons le voir dans la description du compas visuel, une telle relation fait intervenir trois amers et n'est pas restreinte au cas d'un amer lointain et deux amers proches. Au contraire, les trois amers jouent des rôles équivalents.

Garder à l'esprit la parenté entre méthode de l'étoile polaire et compas visuel permet de mieux comprendre ce dernier. Nous enchaînons donc sur la description de la méthode d'orientation par le compas visuel.

6.3 Fonctionnement interne

Nous présentons d'abord une brève vue d'ensemble du compas visuel avant d'entrer plus avant dans les détails.

6.3.1 Découpler l'orientation et la position

Le principe du compas visuel est apparenté au découplage appliqué par Nelson et Aloimonos, mais appliqué un cran plus loin. En effet, alors que la méthode de Nelson et Aloimonos découplait exactement un problème à 5 degrés de liberté (position 3D à un facteur d'échelle près, et rotation 3D) en trois problèmes à deux degrés de liberté (direction du déplacement et orientation), notre méthode consiste à découpler à nouveau ce dernier problème en introduisant une approximation.

6.3.2 Vue d'ensemble

6.3.2.1 Notations

Nous travaillons désormais avec des panoramas d'orientations variées. Comme la figure 6.1 l'illustre, le panorama mémorisé peut avoir été pris avec une orientation différente du panorama actuellement vu.

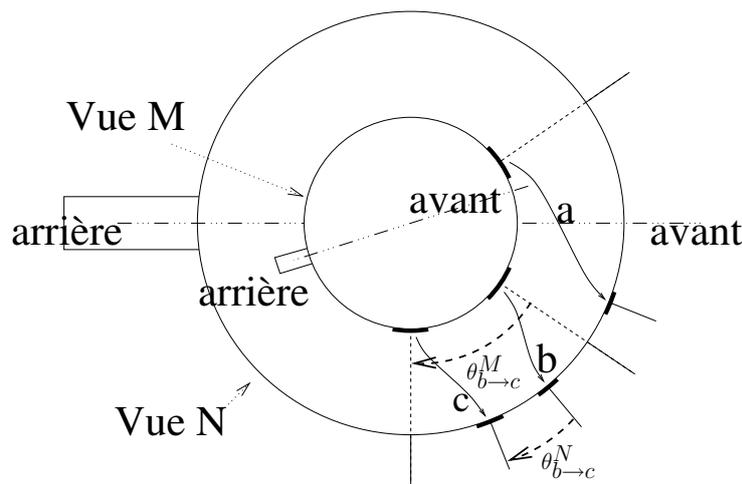


FIG. 6.1: Notation à l'usage du compas visuel. Comme précédemment, le cercle intérieur représente la projection du panorama à la position mémorisée de l'animat (vue M), et le cercle extérieur la projection du panorama à la position actuelle (vue N). $\theta_{b \rightarrow c}^M$ et $\theta_{b \rightarrow c}^N$ sont indiqués par des flèches en arc de cercle. Les flèches courbes représentent l'appariement réalisé entre les projections de trois amers a, b, c dans les vues M et N . La principale nouveauté est que l'orientation du corps de l'animat dans le panorama mémorisé est représentée par un second rectangle tangent au cercle intérieur.

6.3.2.2 Aperçu concret du fonctionnement

La figure 6.2 permet de visualiser concrètement et simplement le principe de fonctionnement du compas visuel avant de rentrer dans les détails formels.

Le compas travaille en deux phases : une étape dite d'observation, qui détermine un invariant à partir des mesures d'azimuts de 3 amers dans 3 vues, et une étape d'exploitation répétable à volonté, qui utilise cet invariant pour retrouver l'orientation dans un voisinage des 3 vues qui ont été utilisées lors de l'observation.

Comme nous le détaillerons, parce que le découplage sur lequel repose la méthode s'appuie sur une approximation au premier ordre, l'invariant n'est qu'approximativement invariant dans une région qui couvre les trois vues d'observation. Il dérive graduellement à mesure que l'animat s'en éloigne, et la précision du compas diminue en conséquence. Cependant, cette dérive n'interdit pas l'usage du

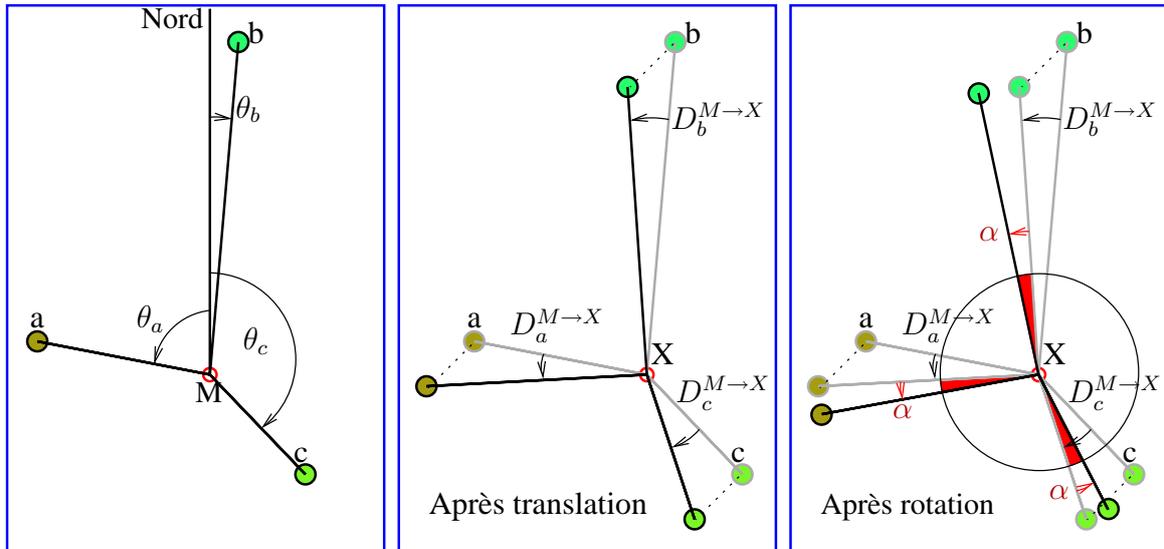


FIG. 6.2: Illustration concrète du principe du compas visuel. À gauche : depuis un point M , l'animat est supposé pouvoir mesurer les azimuts des amers l'entourant, ici a , b et c . Au milieu : si l'animat se déplace vers un point X , son déplacement cause des parallaxes $D_i^{M \rightarrow X}$. Le principe du compas visuel est de se donner une combinaison linéaire des trois $D_i^{M \rightarrow X}$ qui a la propriété d'être constante au premier ordre dans un voisinage du point M . Cela signifie que, tant que l'animat se déplace sans tourner, cette combinaison linéaire est invariante (au premier ordre) vis-à-vis du point d'observation. À droite : utilisation du compas. Si l'animat subit une dérive angulaire α , la combinaison linéaire sera perturbée d' α fois la somme des coefficients, ce qui permet de trouver très facilement α , devenue seule inconnue.

compas en pratique, ainsi que cela sera démontré plus loin.

6.3.3 Étape d'observation

6.3.3.1 Approximation de linéarité des parallaxes

L'approche que nous avons développée avec le compas visuel est parallèle à, et complémentaire du, modèle de déformation de Franz (voir section 4.5). Elle est parallèle car nous y introduisons aussi une approximation, mais qui ne repose pas sur une hypothèse de distance constante. Elle est complémentaire car, au lieu de supposer que les amers sont à la même distance, elle utilise au contraire implicitement les différentes distances. Elle peut aussi être complémentaire de l'algorithme Panama de Röfer, notamment dans le cas où l'environnement contient peu d'amers exploitables, et nous reviendrons plus loin sur cet aspect.

L'approximation que nous introduisons peut être exprimée de plusieurs façons. Concrètement, nous supposons que le déplacement d'un amer i dans le champ de vision causé par le mouvement de l'animat (la parallaxe angulaire) est une fonction linéaire du déplacement.

Formellement, il s'agit simplement de linéariser au voisinage d'un point M la fonction donnant

l'azimut d'un amer en fonction du déplacement de l'animat, en confondant cet azimut avec son approximation au premier ordre. Ainsi, pour un déplacement au voisinage d'un point M , en un point X variable nous obtenons l'expression :

$$\theta_{nord \rightarrow i}^X = \theta_{nord \rightarrow i}^M + \frac{d\theta_{nord \rightarrow i}^M}{d\overrightarrow{MX}} \cdot \overrightarrow{MX} + o(\overrightarrow{MX})$$

Notre approximation de linéarité des parallaxes revient à ignorer le terme en $o(\overrightarrow{MX})$ dans cette équation.

L'essentiel de cette étape d'observation est de récolter suffisamment d'informations pour exploiter cette linéarité et construire un invariant au premier ordre.

6.3.3.2 Hypothèses et déroulement

Nous supposons que l'animat peut évoluer dans un voisinage de sa position actuelle, que nous appellerons région d'observation. Nous supposons la présence d'au moins trois amers que l'animat perçoit dans toute cette région d'observation. On peut énumérer des triplets d'amers (s'il y a n amers il y a $n(n-1)(n-2)/6$ triplets), et considérer le cercle circonscrit à chaque triplet. Nous supposons qu'au moins un de ces cercles circonscrits n'intersecte pas la région d'observation. Nous verrons pourquoi à la section suivante.

L'étape d'observation consiste en la mémorisation de trois panoramas pris depuis trois points M, N, P situés dans la région d'observation. La connaissance de ces trois panoramas va permettre de calculer des coefficients λ_i dont dépendra la propriété d'invariance.

Remarquons que l'animat doit connaître sa direction pendant cette étape, mais les positions des points de prises de vues M, N, P dans l'environnement n'ont pas à être connues.

Ces points de prises de vues ne doivent pas être alignés. De façon similaire au choix d'un amer jouant le rôle d'étoile polaire 6.2.1.2, il est bon que le plus grand des angles du triangle MNP soit inférieur à 135° . Les panoramas doivent avoir au moins 3 amers i, j, k communs vus dans chacun des 3 panoramas, ce qui fait 9 azimuts mesurés.

6.3.3.3 L'invariant

Sous les hypothèses que nous nous sommes fixées, nous allons construire une fonction des azimuts observés qui sera un invariant (au premier ordre) par rapport au déplacement en translation de l'animat. La disponibilité de cet invariant permettra ensuite de découpler la translation et la rotation, dans le cas d'un déplacement faisant intervenir les deux variables.

6.3.3.3.1 Définition La disponibilité des trois panoramas nous donne accès à des mesures de parallaxes. À l'aide de ces mesures nous allons calculer ce que nous appelons une « marque » (d'après

l'expression « prendre ses marques »¹).

Une « marque » est un ensemble de trois nombres réels $\lambda_a, \lambda_b, \lambda_c$ associés au triplet d'amers (a, b, c) et à la vue M , que nous utiliserons à l'étape « exploitation » pour estimer la direction de l'animat *via* l'invariant.

Pour $(i, j, k) \in \{(a, b, c), (b, c, a), (c, a, b)\}$, nous définissons (et justifierons très bientôt) :

$$\lambda_i = D_j^{M \rightarrow N} D_k^{M \rightarrow P} - D_j^{M \rightarrow P} D_k^{M \rightarrow N} \quad (6.1)$$

Nous définissons alors l'invariant $I^{a,b,c,M,N,P}$ attaché au triplet comme étant une combinaison linéaire des trois azimuts, avec les notations introduites en section 2.1.1.1 :

$$\begin{aligned} I^{a,b,c,M,N,P}(X) &= \sum_{i \in \{a,b,c\}} \lambda_i \mu(\theta_{nord \rightarrow i}^X - \theta_{nord \rightarrow i}^M) \\ &= \sum_{i \in \{a,b,c\}} \lambda_i D_i^{M \rightarrow X} \end{aligned} \quad (6.2)$$

La somme des trois λ n'est pas proche de zéro parce que nous éliminons par hypothèse le cas où le cercle circonscrit aux trois amers passe à proximité du triangle MNP . Nous verrons précisément pourquoi en 6.4.2.1.

Tous les triplets sont ainsi énumérés et les marques correspondantes sont créées, comme détaillé plus loin (section 6.4.3.1). Pour l'instant on peut simplement supposer qu'on a trouvé un triplet convenable, parce qu'on a fait l'hypothèse qu'au moins un des triplets avait un cercle circonscrit ne passant pas à proximité du triangle MNP .

Un exemple d'invariant est donné à la figure 6.3. Cette figure montre l'intérêt et les limites de l'approximation de linéarité sur laquelle repose la propriété d'invariance. Les trois disques pleins près des bords sont trois amers. Les trois cercles sont les trois points d'observations reliés par le chemin qu'a pris l'animat. Les dégradés de fond et les lignes de niveau représentent les valeurs de l'invariant sur tout le rectangle. Les petits chiffres sur les lignes de niveaux indiquent les valeurs effectivement prises par l'invariant, divisées par la somme des coefficients, ce qui correspond aussi à l'erreur angulaire due à l'approximation, en degrés. L'aspect polygonal des lignes de niveau est dû à la taille de la grille où les valeurs sont échantillonnées. L'irrégularité des lignes de niveau est due à la granularité de la rétine simulée de l'animat qui limite la précision des mesures d'angles à 1 degré. On constate que l'invariant est exactement nul aux points d'observation (aux erreurs de mesures près), ce qui est vrai par construction. Les lignes de niveaux très espacées montrent que les valeurs sont faibles, non seulement à proximité des points mais sur une zone plus étendue vers le centre du diagramme. En revanche, l'erreur devient forte quand on s'approche des amers car l'approximation de linéarité ne tient plus (cf. 6.4.4), causant des zones noires là où la densité de lignes de niveaux devient très forte.

¹En anglais « prendre ses marques » se dit “to get (ou to find) one's bearings”, d'où le choix du terme « bearing » quand nous publions en anglais.

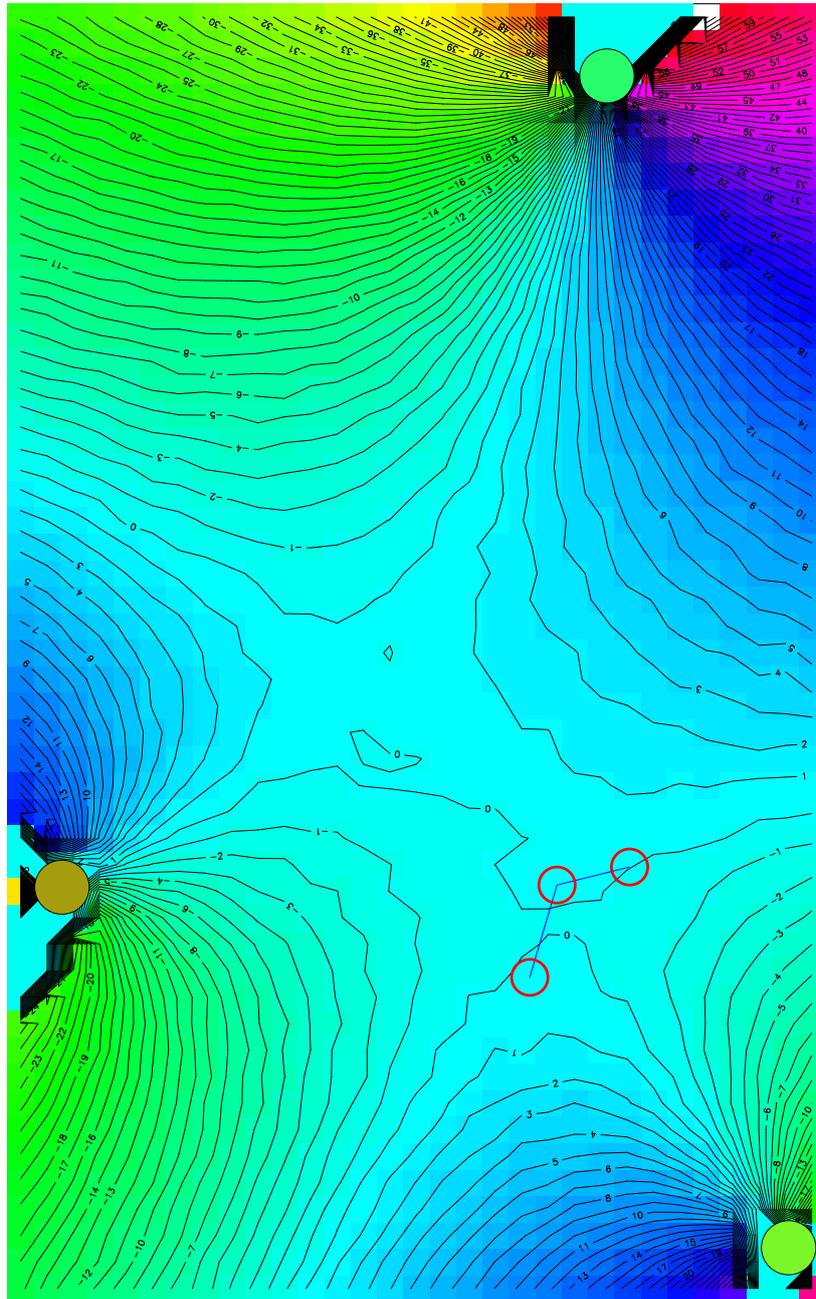


FIG. 6.3: Un exemple de valeurs obtenues pour l'invariant. (Plus d'explications sont incluses dans le texte.) Comme en témoignent les lignes de niveau, l'invariant n'est pas strictement nul. En revanche, les lignes de niveau sont très espacées au voisinage des trois points d'observation (cercles ouverts) mais aussi dans une région plus étendue vers le centre du diagramme, ce qui signifie que l'invariant varie très peu dans cette région. La signification pratique de cette figure est de représenter, en chaque point du rectangle couvert, l'erreur en degrés que ferait l'animat en utilisant le compas visuel en cet endroit. On constate qu'avec à peine 3 amers, il existe une zone étendue où l'animat peut retrouver son orientation avec une bonne précision. La force de cette méthode est dans son compromis entre ses résultats et ses faibles hypothèses.

6.3.3.3.2 Obtention des coefficients Par définition de l'invariant, $I^{a,b,c,M,N,P}(M) = 0$ car les parallaxes $D_i^{M \rightarrow M}$ sont toutes nulles. Les formules 6.1 de calcul des λ_i ont été obtenues à partir des contraintes suivantes :

$$\begin{aligned} I^{a,b,c,M,N,P}(M) &= I^{a,b,c,M,N,P}(N) \\ &= I^{a,b,c,M,N,P}(P) \\ &= 0 \end{aligned} \quad (6.3)$$

Expliquons l'équation 6.1 en interprétant géométriquement le système 6.3 dans l'espace \mathcal{R}^3 .

La définition de l'invariant par l'équation 6.2 est formellement un produit scalaire entre un vecteur $\Lambda = (\lambda_a, \lambda_b, \lambda_c)$ et un vecteur $\Delta^{M \rightarrow X} = (D_a^{M \rightarrow X}, D_b^{M \rightarrow X}, D_c^{M \rightarrow X})$. On cherche un vecteur Λ non nul qui soit orthogonal à chacun des deux vecteurs $\Delta^{M \rightarrow N}$ $\Delta^{M \rightarrow P}$. Le produit vectoriel $\Delta^{M \rightarrow N} \times \Delta^{M \rightarrow P}$ fournit un tel vecteur. Ses composantes sont précisément les coefficients de l'équation 6.1 :

$$\Lambda = \Delta^{M \rightarrow N} \times \Delta^{M \rightarrow P} \quad (6.4)$$

La vérification est immédiate :

$$\begin{aligned} I^{a,b,c,M,N,P}(M) &= \overrightarrow{0} \cdot \Lambda = 0 \\ I^{a,b,c,M,N,P}(N) &= \Delta^{M \rightarrow N} \cdot \Lambda = \Delta^{M \rightarrow N} \cdot (\Delta^{M \rightarrow N} \times \Delta^{M \rightarrow P}) = 0 \\ I^{a,b,c,M,N,P}(P) &= \Delta^{M \rightarrow P} \cdot \Lambda = \Delta^{M \rightarrow P} \cdot (\Delta^{M \rightarrow N} \times \Delta^{M \rightarrow P}) = 0 \end{aligned}$$

La nullité de l'invariant en ces trois points entraîne sa nullité sur tout un voisinage (moyennant l'approximation au premier ordre). En effet, nos hypothèses de linéarité reviennent à confondre $I^{a,b,c,M,N,P}(X)$ avec sa différentielle en M . Celle-ci est une forme linéaire dépendante de deux variables (les coordonnées de X), qui s'annule sur trois points non colinéaires (M, N, P), et est donc nulle partout.

$I^{a,b,c,M,N,P}(X)$ est donc bien nul au premier ordre en M , c'est-à-dire que les λ calculés sont ceux qui minimisent la valeur effective de l'invariant dans un voisinage des points d'observation.

Pour être rigoureux, un invariant nul au premier ordre au sens strict serait obtenu en faisant tendre N et P vers M . Par commodité, nous nous contentons de l'approximation où N et P sont à distance finie de M , d'autant plus que, pour obtenir en conditions réelles une mesure de parallaxe $\Delta^{M \rightarrow X}$, X doit être suffisamment loin de M pour que les parallaxes mesurées ne soient pas noyées dans le bruit de mesure.

6.3.3.4 Conclusion pour l'étape observation

L'approximation de linéarisation des parallaxes par rapport aux déplacements nous a permis de définir une combinaison linéaire des azimuts qui est un invariant au premier ordre par rapport aux déplacements en translation de l'animat.

En observant trois vues non alignées et appariées entre elles, l'animat peut se donner un ensemble de marques, chacune associée à un triplet d'amers, et enregistrant les coefficients de la combinaison linéaire qui réalise l'invariance des parallaxes des amers observés. Les marques obtenues sont attachées aux vues M , N , P .

L'animat n'a pas besoin de savoir où les trois vues ont été prises. Il suffit qu'elles soient d'orientation connue, suffisamment voisines l'une de l'autre pour présenter des amers communs (sans toutefois être trop proches, pour que les parallaxes soient mesurables).

6.3.4 Étape exploitation

Une fois l'opération d'observation faite pour un triplet d'amers donné depuis un point donné, la marque obtenue est acquise. En général, il y aura plusieurs triplets donc plusieurs marques.

Chaque marque qui a été obtenue pendant une étape d'observation des vues M , N et P est attachée à ces vues. Cela signifie que l'animat saura qu'il est à proximité de la zone d'observation de M , N et P chaque fois que le panorama courant X ressemblera à l'une de ces vues. Quand cela se produit, l'animat peut extraire de sa mémoire les marques acquises, et les exploiter pour obtenir une estimation de sa direction actuelle au point X . Nous allons détailler cette étape.

6.3.4.1 Nécessité d'un algorithme d'appariement performant

Une première étape dans un problème d'orientation se pose au moment de l'appariement. En effet, si l'animat est complètement désorienté, les méthodes d'appariement utilisées par les modèles CC ou Hong ainsi que celles expérimentées par Weber ne peuvent fonctionner convenablement, car elles s'appuient essentiellement sur un critère de minimisation des parallaxes lors de la recherche d'appariements potentiels. Elles sont donc très dépendantes de l'orientation préalable des panoramas d'après la même référence et, par là même, de la connaissance préalable par l'animat de son orientation. Ces méthodes d'appariement ne sont donc pas utilisables pour appairer correctement des vues d'orientation relative inconnue.

En revanche, un appariement exploitant une information plus riche permettant de distinguer les amers les uns des autres, comme celui que nous avons exposé au chapitre précédent, est en mesure de fournir un résultat fiable même en cas de désorientation totale. Nous notons que l'algorithme d'appariement de Panama possède aussi cette propriété, mais il opère sur des panoramas échantillonnés, et non des panoramas segmentés comme les nôtres.

6.3.4.2 Pré-orientation après appariement

Cette étape et celle qui suit sont, pour une vue donnée, applicables séparément à chaque marque disponible.

L'étape d'exploitation d'une marque nécessite des calculs de parallaxes et l'évitement des effets

de bouclages dus à la définition des angles modulo 2π . Le problème est illustré en figure 6.4. Pour cette raison, on pré-oriente les panoramas.

Le nouveau panorama perçu au point X est d'abord artificiellement tourné de façon que l'amer auquel la marque attribue le λ le plus grand retrouve le même azimut apparent après rotation que dans la vue M attachée à la marque (autrement dit : on le tourne de façon que la pseudo-parallaxe de cet amer soit nulle). On note de combien on a tourné le panorama au cours de cette étape et on appelle cette grandeur $\theta_{corps \rightarrow preorient}^X$ (car le panorama était initialement aligné avec le corps de l'animat, alors qu'il a maintenant son orientation propre).

Cette étape est optionnelle parce qu'il n'est pas essentiel que l'amer en question semble avoir une parallaxe nulle dans le panorama pré-orienté. Si l'animat connaît déjà à peu près son orientation à un écart près, les trois pseudo-parallaxes sont dans la zone de linéarité, et on peut sauter cette étape. Pour faciliter la compréhension on peut complètement ignorer cette étape de pré-orientation dans un premier temps.

6.3.4.3 Orientation fine grâce à l'invariant

6.3.4.3.1 Calcul de l'orientation de l'animat On calcule le dénominateur attaché à la marque.

$$d = \lambda_a + \lambda_b + \lambda_c$$

La formule de l'invariant est appliquée *comme si la vue X était correctement orientée*. Comme dans le cas général elle ne le sera pas, on parle de pseudo-parallaxe et on note I' le pseudo-invariant calculé. Formellement, la désorientation a pour effet que la référence « corps » n'est pas la même dans les vues M et X dans l'équation ci-dessous :

$$I'(X) = \sum_{i \in \{a,b,c\}} \lambda_i \underbrace{\mu(\theta_{corps \rightarrow i}^X - \theta_{corps \rightarrow i}^M)}_{\text{pseudo-parallaxe}} \quad (6.5)$$

Alors l'orientation actuelle du corps de l'animat $\theta_{nord \rightarrow corps}^X$, estimée par cette marque, est simplement :

$$\theta_{nord \rightarrow corps}^X = \theta_{nord \rightarrow corps}^M - \frac{I'(X)}{d} \quad (6.6)$$

6.3.4.3.2 Démonstration La pseudo-parallaxe serait une vraie parallaxe si et seulement si les vues M et X avaient la même référence d'orientation, c'est-à-dire si $\theta_{nord \rightarrow corps}^X = \theta_{nord \rightarrow corps}^M$. Les trois λ ont été calculés pour que la combinaison linéaire des parallaxes soit constante et nulle, au premier ordre, dans le voisinage des points d'observation. En conséquence, si la vue X est déjà correctement orientée, $I'(X) \simeq 0$. Rappelons que cette valeur est obtenue exactement par construction pour les cas $X = M$, $X = N$ et $X = P$.

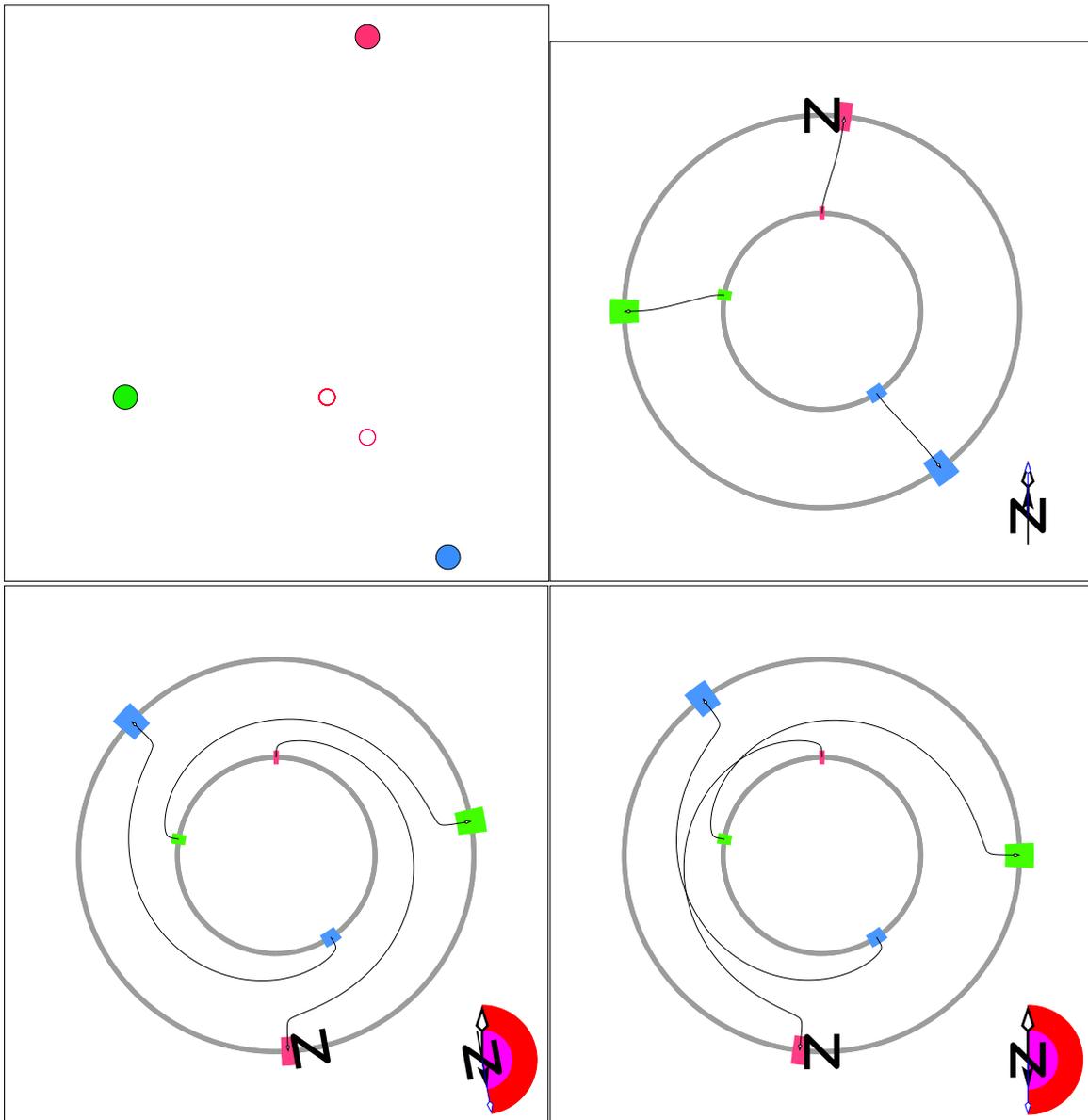


FIG. 6.4: Le compas a besoin pour fonctionner que les pseudo-parallaxes dépendent linéairement du déplacement de l'animat. Quand une pseudo-parallaxe dépasse 180° parce que l'animat est très désorienté, cette linéarité est détruite. La pré-orientation évite ce problème. En haut à gauche, un environnement avec trois amers (disques) et deux points (cercles) où l'animat a pris des vues. En haut à droite, l'appariement entre les deux vues orientées qui permet de calculer des parallaxes. En bas, l'animat est désorienté de presque un demi-tour (à gauche) et un demi-tour (à droite). À gauche le calcul des pseudo-parallaxes est correct, mais l'amer situé au nord a une pseudo-parallaxe de presque 180° . Le compas fonctionne encore sans pré-orientation. À droite, la pseudo-parallaxe a dépassé 180° . La valeur accessible à l'animat est la mesure principale de l'angle, qui devient proche de -180° (illustrée par la flèche courbe qui croise les deux autres). Le compas ne peut plus fonctionner directement. La pré-orientation ramène les pseudo-parallaxes dans la zone de linéarité et permet au compas de fonctionner quelle que soit la désorientation actuelle de l'animat.

Formellement :

$$\begin{aligned}
 I'(X) &= \sum_{i \in \{a,b,c\}} \lambda_i \mu(\theta_{corps \rightarrow i}^X - \theta_{corps \rightarrow i}^M) \\
 &= \sum_{i \in \{a,b,c\}} \lambda_i \mu(\theta_{corps \rightarrow nord}^X + \theta_{nord \rightarrow i}^X - \theta_{corps \rightarrow nord}^M - \theta_{nord \rightarrow i}^M) \\
 &= \sum_{i \in \{a,b,c\}} \lambda_i \mu(\underbrace{\theta_{corps \rightarrow nord}^X - \theta_{corps \rightarrow nord}^M}_{\text{désorientation}} + \underbrace{\theta_{nord \rightarrow i}^X - \theta_{nord \rightarrow i}^M}_{\text{parallaxe vraie}})
 \end{aligned}$$

La pré-orientation assure que la désorientation n'est pas trop forte et permet de sortir les termes de la fonction μ :

$$\begin{aligned}
 I'(X) &= \sum_{i \in \{a,b,c\}} \lambda_i \mu(\underbrace{\theta_{corps \rightarrow nord}^X - \theta_{corps \rightarrow nord}^M}_{\text{désorientation}}) + \sum_{i \in \{a,b,c\}} \lambda_i \mu(\underbrace{\theta_{nord \rightarrow i}^X - \theta_{nord \rightarrow i}^M}_{\text{parallaxe vraie}}) \\
 &= \underbrace{\left(\sum_{i \in \{a,b,c\}} \lambda_i \right)}_d \mu(\theta_{corps \rightarrow nord}^X - \theta_{corps \rightarrow nord}^M) + \underbrace{I(X)}_{\simeq 0}
 \end{aligned}$$

En réarrangeant les termes, et parce que d est non nul :

$$\theta_{nord \rightarrow corps}^X = \theta_{nord \rightarrow corps}^M - \frac{I'(X)}{d}$$

En d'autres termes, la mesure des pseudo-parallaxes permet de retrouver l'angle dont a tourné le corps de l'animat entre les vues M et X , valeur que nous cherchons. Parce que la combinaison linéaire $I(X)$ est conçue pour valoir 0 quand il y a translation sans rotation, la valeur du pseudo-invariant calculé $I'(X)$ est l'angle recherché multiplié par d , la somme des λ , qui est connue. D'où l'équation 6.6.

Nous invitons le lecteur à se référer à la figure 6.2 (page 155) dont la structure correspond à celle du raisonnement tenu ici.

Quant à la pré-orientation, elle n'est là que pour éviter des effets de bouclage qui ne se produisent pas toujours. Si aucune des pseudo-parallaxes n'est proche de 180° , la pré-orientation est complètement superflue. Par contre, si une des pseudo-parallaxes est proche de 180° , une petite rotation peut la faire basculer de l'autre côté (à -180°) et détruire la réponse linéaire sur laquelle repose la méthode.

Pourquoi le choix de pré-orientation exposé en section 6.3.4.2 plutôt qu'un autre ? L'amer ayant le plus grand λ est celui qui était le plus stable dans le champ de vision au moment de l'observation. En pré-orientant la vue de façon à lui donner une pseudo-parallaxe de zéro, on donne au panorama pré-orienté une direction proche de la vraie direction et les pseudo-parallaxes sont proches des vraies parallaxes. Alors on évite les effets de bouclage.

On remarque que, dans le cas particulier où l'amer en question est infiniment loin, la pré-orientation aboutit déjà à l'orientation correcte.

6.3.5 Illustration par un cas concret simplifié

Afin de clarifier l'exposé et de le rendre plus concret, nous présentons ici un exemple dans lequel la pré-orientation n'est pas nécessaire, avec les valeurs obtenues pour l'invariant.

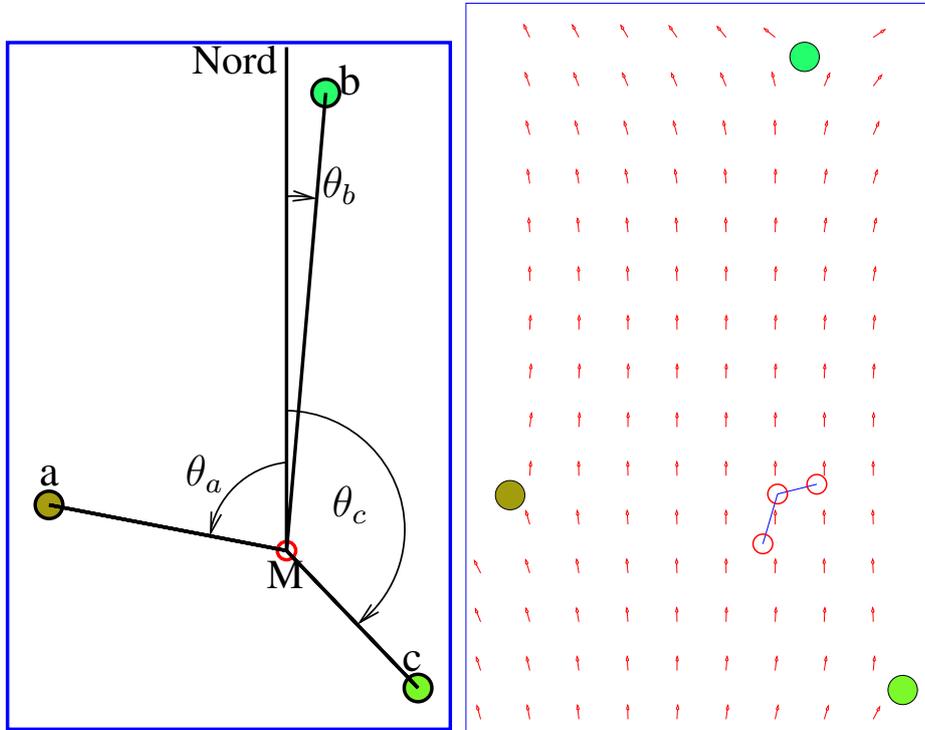


FIG. 6.5: Illustration concrète du principe du compas visuel, légèrement simplifié. Les disques ombrés sont des amers. Au cours de son mouvement le long des trois cercles reliés (à droite), l'animat observe que $43.25 \cdot \theta_{nord \rightarrow a} + 92 \cdot \theta_{nord \rightarrow b} + 55.25 \cdot \theta_{nord \rightarrow c}$ est approximativement constant (constant au premier ordre). Parce que cette expression est linéaire en les θ , il est très facile d'en extraire $\theta_{nord \rightarrow corps}$ (voir texte) et donc de se réorienter. Le champ de vecteurs à droite montre la direction du nord estimée par le compas, en chaque point d'une grille régulière. Elle dérive progressivement à mesure que l'on s'éloigne des points d'observation, à cause de l'approximation linéaire sur laquelle repose la propriété d'invariance. En réalité, cette expression telle quelle serait perturbée par les effets de bouclage des angles, avec pour conséquence que l'étape d'exploitation du compas ne fonctionnerait pas dans toutes les orientations. C'est pourquoi des parallaxes sont en réalité utilisées au lieu d'azimuts, et une nouvelle vue à orienter est d'abord pré-orientée pour aligner l'amer le plus stable (celui qui a le plus grand λ , voir le texte).

La figure 6.5 montre dans un environnement simple un exemple de mouvement de notre animat qui permet une observation aboutissant à un invariant, et l'estimation de direction que peut fournir cet invariant en divers points de l'environnement. L'invariant obtenu est $43.25 \cdot \theta_{nord \rightarrow a} + 92 \cdot \theta_{nord \rightarrow b} + 55.25 \cdot \theta_{nord \rightarrow c} \simeq 0$.

En tout point X , on peut prendre une nouvelle vue, qui fournit des $\theta_{corps \rightarrow i}^X$. On utilise, pour chaque i , l'égalité $\theta_{nord \rightarrow i} = \theta_{nord \rightarrow corps} + \theta_{corps \rightarrow i}$. Par substitution dans l'expression de l'invariant,

on isole très facilement $\theta_{nord \rightarrow corps}$ qui est la seule inconnue.

Dans cet exemple on trouve le résultat concret suivant : si l'animat se déplace et/ou tourne sur lui-même et prend une nouvelle vue X , l'orientation de la vue X est donnée par

$$\theta_{nord \rightarrow corps}^X = - \frac{43.25 \cdot \theta_{corps \rightarrow a}^X + 92 \cdot \theta_{corps \rightarrow b}^X + 55.25 \cdot \theta_{corps \rightarrow c}^X}{43.25 + 92 + 55.25}$$

C'est ce calcul, et celui-là seulement, (3 multiplications et deux additions) qui est nécessaire pour retrouver l'orientation, à chaque fois que l'animat voudra estimer l'orientation d'une nouvelle vue X prise dans le voisinage de la zone d'observation. Le dénominateur de cette fraction est la variable qu'on a introduite et appelée dénominateur en 6.3.4.3.1.

Notons que les λ ont été calculés à partir de vues prises au cours d'un mouvement particulier. Mais à aucun moment l'animat n'a besoin de connaître ce mouvement. On a seulement besoin de connaître l'orientation des trois vues d'observation. C'est dans ce sens que cette méthode découple l'orientation et le déplacement. Autrement dit, l'invariant est une combinaison linéaire qui compense le déplacement, ce qui permet d'extraire l'orientation seule.

Pour clarifier l'exemple et la figure, les équations présentées dans cette section sont simplifiées et font intervenir des azimuts θ au lieu de parallaxes. En conséquence, l'invariant simplifié représenté ici ne fonctionne plus lorsque l'animat passe au nord d'un des amers, car alors $\theta_{nord \rightarrow i}$ passe de $-\pi$ à π ou inversement. L'utilisation de parallaxes déplace cette discontinuité de façon qu'elle ne gêne plus.

Dans un cas réel on disposera probablement de plus de 3 amers. On utilisera alors plusieurs triplets, donc plusieurs marques, comme expliqué en 6.4.3.1.

6.3.6 Utilisation de plusieurs observations pour étendre la couverture du compas

À partir d'une seule observation, l'animat ne peut s'orienter correctement que dans la zone où l'approximation de linéarité sur laquelle s'appuie cette observation est tenable (nous donnerons des exemples en section 6.5).

Mais cela ne signifie pas que l'animat ne peut pas étendre la couverture du compas à une surface plus grande. En effet, un ensemble de marques est associé à une vue. Si l'animat peut mémoriser plusieurs vues correspondant à plusieurs buts, il peut exploiter lors de sa navigation les marques associées à plusieurs vues, même éloignées les unes des autres.

Par exemple, supposons que l'animat soit dans une zone A (le nid) de la figure 6.6, zone dans laquelle il a réalisé une étape d'observation. Muni du compas, il est capable de retourner au nid depuis un point proche. Supposons qu'il explore son environnement. La figure 6.6 présente une façon possible d'étendre le compas à une zone que nous appellerons B, par une méthode accessible à l'animat.

On remarque qu'il n'est pas nécessaire de tenir l'indication du compas visuel pour absolue. Il suffit qu'elle soit relative à une zone de l'environnement donnée. L'essentiel en effet est que l'animat puisse naviguer de façon fiable entre A et B. Cela signifie que, fondamentalement, l'animat peut réaliser

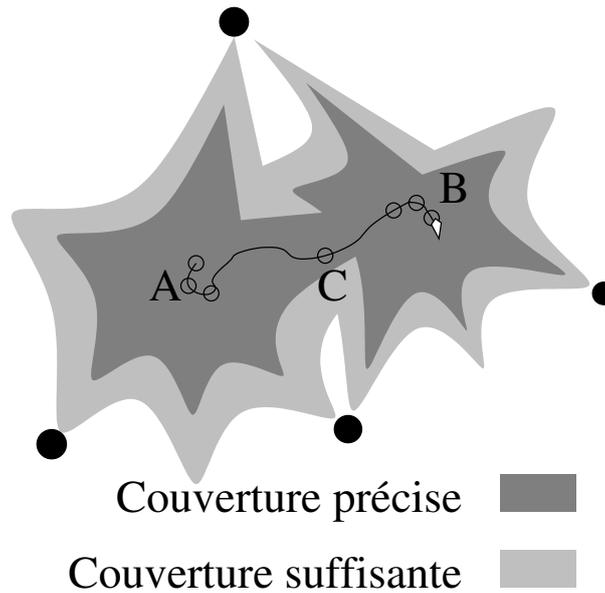


FIG. 6.6: Extension du compas à plusieurs zones. A gauche, une zone A de l'environnement, couverte par le compas (zone grisée autour de A). L'animat explore son environnement aux alentours de A (flèche sinueuse), en s'appuyant sur le compas pour ne pas perdre son orientation. Il peut estimer la confiance qu'il a en l'utilisation des marques de A via, par exemple, le pourcentage de marques toujours visibles ou leur cohérence (cf.6.4.3.1). Si l'animat estime que l'erreur d'orientation va dépasser un seuil, il peut décider d'augmenter la couverture du compas. Ainsi, il réalise une nouvelle étape d'observation en B, de sorte qu'il peut naviguer entre A en B sans se perdre. Dans la nouvelle zone de couverture (grisée autour de B) l'animat bénéficie du compas visuel via les marques acquises en B. Il peut étendre ainsi la couverture du compas de proche en proche en même temps qu'il se construit une carte topologique de l'environnement. Il n'est toutefois pas nécessaire d'avoir une étape observation pour chaque but. Par exemple, l'animat peut rejoindre le point C en restant orienté grâce aux marques de A.

une étape d'observation en B même si son orientation par rapport à la zone A à ce moment n'est pas bien connue. Quand l'animat voudra rejoindre le point B, il utilisera les marques associées à B pour s'orienter. Quand il voudra rejoindre le point A, il utilisera les marques associées à A. Cette façon de faire est suffisante si l'animat n'a pas besoin d'information métrique à plus grande échelle. Une information métrique serait utile pour se construire un plan d'ensemble de son environnement, surtout s'il contient une boucle à grande échelle, ce qui dépasse de trop loin notre propos. Toutefois nous présenterons au chapitre 7 une méthode de construction de carte topologique intégrant le compas.

On peut voir l'indication du compas comme une version « gelée et localement étendue » de l'information que l'animat connaissait sur son orientation au moment de l'étape d'observation. Elle est « gelée » car l'animat obtiendra la même réponse du compas sans dérive au cours du temps, par opposition à l'odométrie qui dérive avec la distance parcourue par l'animat. Elle est « localement étendue » car elle ne se limite pas aux trois points qui ont permis de calibrer (ce qui ne serait pas très intéressant, puisqu'il faudrait rejoindre ces points pour savoir comment on est orienté). Au contraire, elle couvre un voisinage des points suffisamment vaste pour que l'animat puisse connaître son orientation avec

suffisamment de précision pour naviguer et rejoindre ses buts.

6.4 Étude théorique

6.4.1 Coût computationnel

À l'étape d'observation, après avoir réalisé les appariement $M - N$, $M - P$ (et éventuellement $N - P$, comme nous le verrons en 6.4.3.2), il faut énumérer les triplets pour calculer les λ et enregistrer les marques. Cette étape est en $O(n^3)$, n étant le nombre d'amers communs aux trois panoramas. Rappelons que l'étape d'observation n'est en principe nécessaire qu'une fois en un lieu donné et que les informations enregistrées pourront être utilisées de nombreuses fois par la suite.

À chaque occurrence de l'étape d'exploitation, considérons d'abord le cas local, c'est-à-dire le cas où l'animat sait déjà dans quelle région il est, comme nous l'avons fait pour la navigation. En général, la vue qui porte les marques utilisées est la vue qui est le but local de la navigation. Alors, pour une marque donnée, une estimation d'orientation élémentaire peut être produite chaque fois que les trois amers concernés sont visibles. Elle peut alors être obtenus en $O(1)$ puisque le calcul d'une estimation ne nécessite que quelques multiplications et additions (cf. exemple concret en section 6.3.5).

En général, plusieurs marques attachées à la vue du but sont disponibles, et peuvent être implémentées sous la forme d'une liste qu'il suffit de parcourir (on détaillera la façon d'utiliser plusieurs marques en section 6.4.3.1). Alors le coût global de l'utilisation du compas est en $O(k)$, k étant le nombre de marques attachées à la vue du but.

Cependant, si le nombre d'amers dans l'environnement est grand, il se peut que l'animat trouve un grand nombre de marques à l'étape d'exploitation. On peut alors ajouter des filtres de sélection similaires à ceux que nous exposerons en 6.4.3.2 ou d'autres filtres sélectionnant seulement un nombre fixé d'amers parmi les plus saillants, les plus susceptibles d'une bonne ré-acquisition ou d'un bon appariement futur, pour limiter le nombre de marques effectivement utilisées.

On peut résumer ces considérations en disant que le coût d'une étape d'exploitation est borné en pratique. La seule étape vraiment coûteuse est en réalité l'appariement de la vue actuelle avec la vue portant les marques, qui est en $O(n^2)$, mais était déjà nécessaire dans le cadre de la navigation et ne constitue donc pas un surcoût apporté par le compas visuel.

6.4.2 Cas de non-applicabilité

La méthode du compas visuel, comme dit en introduction de ce chapitre (6.3.1), consiste essentiellement à découpler la composante translation du mouvement de la composante rotation, à l'aide de l'invariant. Malheureusement, pour tout triplet d'amers, il existe une région où cette séparation est impossible.

6.4.2.1 Un triplet donné n'est pas utilisable partout

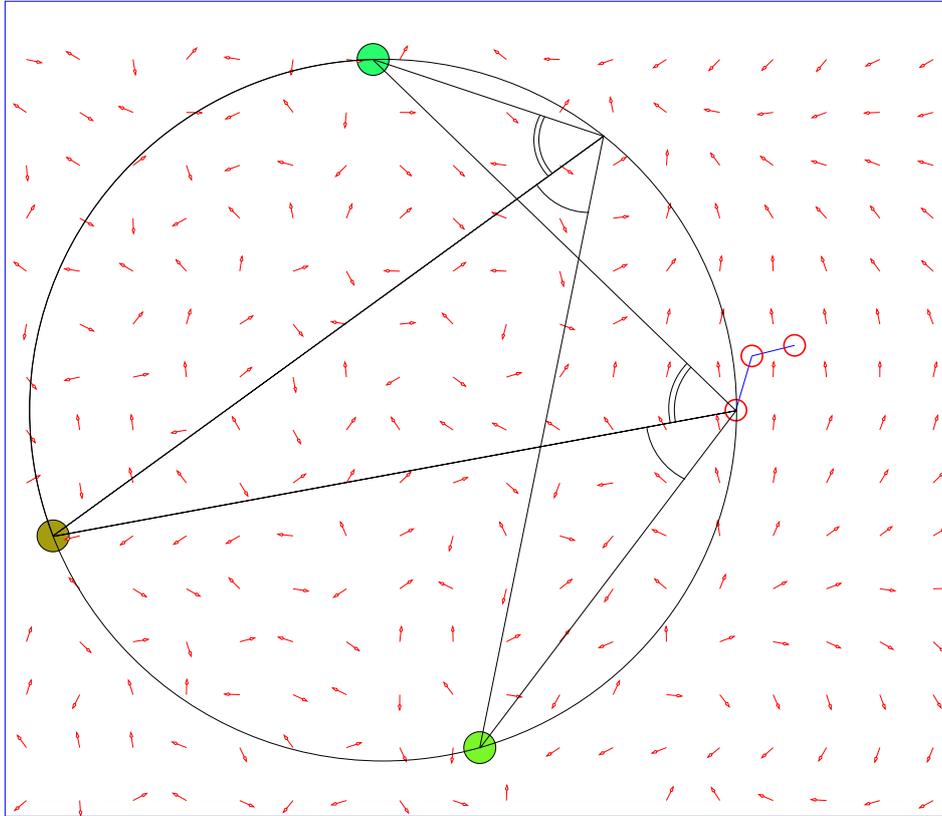


FIG. 6.7: Exemple illustrant un cas dégénéré et une limitation du compas visuel dans le cas où un seul triplet est disponible. Le cercle passe par les trois amers, ce qui le découpe en trois arcs. Chacun de ces arcs a la propriété géométrique que les angles apparents entre amers sont les mêmes depuis n'importe quel point (ils sont représentés ici à deux positions). Du point de vue de l'animat, cela signifie qu'en ces points, l'orientation (tourner sur soi-même) ne peut être découplée du déplacement le long de l'arc. En conséquence, si les points d'observation sont trop proches de ce cercle, le triplet correspondant ne pourra produire de résultat fiable. Dans cet exemple les paramètres trouvés à l'étape d'observation sont $\lambda_a = -31$, $\lambda_b = 14$, $\lambda_c = 16$ et $d = -1$. d est plus petit (en valeur absolue) que chacun des λ , ce qui est le signe mesurable d'un triplet inutilisable dans cette région (voir texte).

Considérons le cercle passant par trois amers a , b , c distincts. Il est partitionné par les amers eux-mêmes en trois arcs de cercles. En chaque point M de l'un de ces arcs, les angles apparents séparant les amers a , b , c sont constants si on se déplace le long de l'arc, en vertu des propriétés rappelées en section 3.4.1. Cela signifie que, si l'animat se déplace le long de l'arc, il mesurera exactement les mêmes écarts angulaires entre amers. En effet, le long de ce chemin, les parallaxes de tous les points du cercle varient de la même grandeur et l'écart angulaire, différence de deux parallaxes, est constant. Le compas visuel s'appuyant précisément sur les écarts angulaires entre amers pour retrouver l'orientation de l'animat, il ne pourra compenser la translation par la combinaison linéaire des parallaxes. Voir la figure 6.7. Dans une situation réelle, le compas aurait ignoré ce triplet en faveur

d'autres (faisant éventuellement intervenir un ou deux des trois mêmes amers).

La situation se traduit sur les équations de la façon suivante. Considérons que deux points d'observation M et N sont sur le cercle. Cela se traduit par :

$$D_a^{M \rightarrow N} = D_b^{M \rightarrow N} = D_c^{M \rightarrow N} \quad (6.7)$$

Annuler l'invariant revient à chercher Λ tel que :

$$\begin{aligned} I^{a,b,c,M,N,P}(N) &= \Lambda \cdot \Delta^{M \rightarrow N} = 0 \\ 0 &= \Lambda \cdot (1, 1, 1) D_a^{M \rightarrow N} \\ 0 &= dD_a^{M \rightarrow N} \end{aligned}$$

Comme $D_a^{M \rightarrow N} \neq 0$ parce que M et N sont distincts sur le cercle, $d = 0$.

Il suffit donc que deux points d'observation soient sur le cercle circonscrit aux trois amers pour que le dénominateur associé à ce triplet soit nul, et le triplet inutilisable.

Les écarts d'angles, les λ et donc le dénominateur attaché au triplet étant des fonctions continues des trois positions d'observation, le dénominateur est faible devant les λ et le triplet inutilisable lorsque l'animat est proche du cercle.

Heureusement, l'animat n'a pas besoin de reconstituer la géométrie de la scène pour prendre sa décision. Il lui suffit de calculer les λ et le dénominateur attaché au triplet pour décider de le garder ou non, à condition bien sûr qu'il reste dans l'environnement d'autres triplets disponibles.

Par ailleurs, il existe un critère simple et suffisant d'acceptabilité : si un triplet occupe plus de 180° dans le panorama du robot, c'est que ce dernier est dans le triangle et le cas cocyclique indésirable est exclu. Ceci illustre l'avantage d'un champ de vision aussi large que possible, beaucoup plus large qu'un demi-cercle.

6.4.2.2 Comment choisir un bon triplet d'amers

Remarquons que le problème de la cocyclicité, énoncé ci-dessus, est attaché au triplet et non aux amers qui composent le triplet. Un triplet produisant une très bonne observation peut avoir un ou deux amers communs avec un triplet non éligible.

À moins que tous les amers soient sur un même cercle passant par l'animat, il existe au moins un triplet d'amers dont les angles relatifs changent au cours du déplacement de l'animat. Bien sûr, les cas proches de celui-ci sont moins favorables au compas.

Remarquons le parallèle entre ces observations et celles de Nelson et Aloimonos [NEL 88] à propos de la nécessité de disposer d'une vision panoramique pour estimer son mouvement propre. En effet, ces auteurs expliquent que, plus on dispose d'un champ de vision réduit, plus il est difficile de faire la différence entre certains mouvements, notamment entre certaines rotations et certaines translations. Dans notre cas, nous remarquons que, si l'animat occupe une position cocyclique avec

les amers d'un triplet, alors ces triplets sont nécessairement compris dans une moitié du panorama délimitée par la droite tangente au cercle au point où l'animat se trouve. Il est suffisant (mais pas nécessaire) que l'animat choisisse des amers qui sont répartis autour de lui sur plus de 180° pour être dans le bon cas. Ce critère simple indique que le mécanisme de découplage sur lequel nous nous appuyons est au moins valable dans l'enveloppe convexe des amers visibles. Dans le cas d'un triplet, l'enveloppe convexe est l'intérieur du triangle.

6.4.3 Quand plusieurs triplets sont disponibles

6.4.3.1 Agrégation des estimations par plusieurs triplets

En général, parmi l'ensemble des amers qui ont été utilisés à l'étape d'observation, plus de trois sont encore visibles et appariés avec la vue actuelle. Cela signifie donc que, parmi les marques créées à l'étape d'observation, plus d'une marque est disponible pour l'étape d'exploitation. Une façon simple d'en tenir compte est d'appliquer séparément l'étape d'exploitation à chaque marque actuellement utilisable.

Chaque marque, donc chaque invariant exploité, vote pour une direction. On agrège alors les résultats pour obtenir la meilleure estimation possible de la direction actuelle réelle.

La méthode la plus simple d'agrégation de ces résultats individuels est d'associer à chaque marque exploitée un vecteur unitaire pointant dans la direction votée. On calcule la somme de ces vecteurs unitaires. La direction du vecteur global est la direction retenue. La longueur du vecteur global est utilisée comme mesure de confiance : elle est maximale quand toutes les marques votent pour la même direction, et réduite quand les différentes estimations de direction se contredisent. On peut normaliser le vecteur global en le divisant par la somme des normes des segments. La norme du vecteur indique alors la cohérence du faisceau de direction trouvées, avec un maximum à 1 si toutes les marques votent pour la même direction.

Nous avons vu que les marques de meilleure qualité sont associées à des dénominateurs les plus grands. Au lieu de considérer chaque marque à égalité, on peut pondérer les marques par la valeur absolue de leur dénominateur². Nous avons constaté que ce raffinement avait un résultat spectaculaire, aboutissant à ignorer presque complètement les marques les moins fiables, dès que quelques marques fiables sont disponibles à l'étape d'exploitation. Ce point est illustré en figure 6.8. Ceci suggère une amélioration : pour alléger la quantité d'information à mémoriser, on peut envisager dès l'observation de ne conserver qu'un petit nombre de marques, associées aux dénominateurs les plus grands.

6.4.3.2 Sélection de triplets pour améliorer la robustesse

Si les appariements étaient toujours parfaits, il suffirait de réaliser les appariement $M - N$ et $M - P$, et d'utiliser tous les triplets résultants.

²Les expériences de navigation sur robot réel ont été menées sans ce raffinement : la même importance était donnée à toutes les marques.

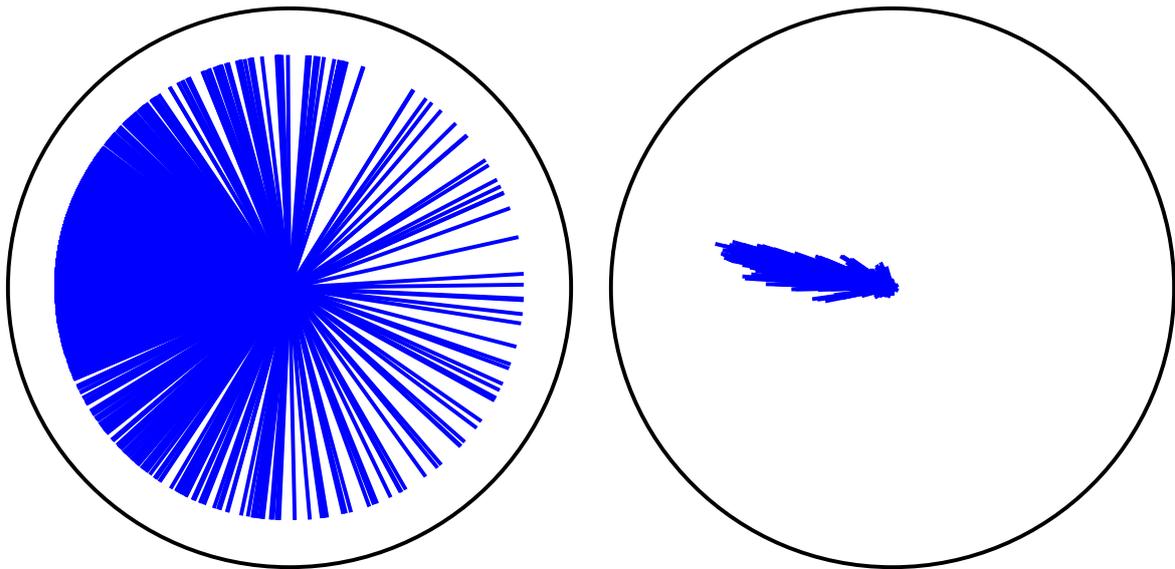


FIG. 6.8: Illustration d'un cas réel où plusieurs triplets d'amers sont disponibles. Une étape d'observation avait apporté 2600 marques dont 2024 sont utilisables dans la vue actuelle. Le schéma de gauche correspond à pondérer toutes les marques de la même manière, celui de droite à pondérer chaque marque en proportion de son dénominateur. Dans les deux cas, l'animat estime son orientation : un ensemble de segments indique les estimations de direction trouvées par autant de marques. À gauche, tous les segments sont normés et on constate qu'ils se répartissent suivant presque toutes les directions. À droite, les marques sont les mêmes mais chaque segment a une norme proportionnelle au dénominateur de la marque associée. La différence est flagrante : les marques de plus grand dénominateur donnent des estimations beaucoup plus groupées. Autrement dit, seules les marques de dénominateurs très faibles votent pour des directions variées. L'orientation trouvée par sommation des vecteurs (respectivement 81.4° et 79.9°) diffère peu entre les deux cas, et reste proche de la meilleure estimation disponible, celle donnée par l'odométrie (82.5°). La mesure de cohérence est respectivement de 0.70 et 0.99. Ceci suggère que l'animat pourrait se contenter de ne mémoriser qu'un nombre fixé de marques, celles de plus grands dénominateurs.

En pratique, les appariements ne sont pas toujours parfaits. Des erreurs d'appariement peuvent causer des invariants aberrants et donc des estimations d'orientation erronées.

À l'étape d'observation, l'utilisation d'un appariement supplémentaire, $N - P$, permet d'éliminer facilement les amers probablement mal appariés.

Considérons un élément a dans la vue M . Il est apparié à un élément b dans la vue N , et à un élément c dans la vue P . Si dans $N - P$, l'élément b est effectivement apparié à c , alors l'ensemble a, b, c est apparié de façon cohérente, et réputé correspondre à un amer de la scène. En revanche, si b n'est pas apparié à c ou pas apparié du tout, on élimine d'office a, b et c lors de l'énumération des amers disponibles pour réaliser des triplets. La figure 6.9 montre dans un exemple réel les trois appariements et les triplets sélectionnés.

À l'étape d'exploitation, un filtrage similaire peut être réalisé. Au lieu de simplement appairier M et X pour calculer les pseudo-parallaxes, on apparie aussi $N - X$ et $P - X$. Si l'élément x de X est bien apparié aux bons éléments a, b et c , alors il est utilisé, sinon il est éliminé.

L'avantage de ce type de filtre est qu'il n'a besoin d'aucun paramètre parce qu'il contrôle simplement la cohérence des appariements. La cohérence n'est pas une garantie de justesse, mais on s'attend à ce qu'un appariement faux ait beaucoup plus de chances d'être incohérent qu'un appariement juste. On estime donc éliminer plus d'erreurs que de éléments bien appariés par ce filtre. Il peut arriver aussi que des amers soient souvent bien appariés mais parfois « manqués ». Ceux-ci sont aussi éliminés.

On pourrait imaginer un raffinement qui consisterait à appairier simultanément M, N et P en tenant compte de la cohérence, plutôt que de filtrer a posteriori, ce qui malheureusement demanderait de repenser l'algorithme d'appariement et serait plus complexe.

6.4.4 Limites

Toutes les méthodes peuvent échouer lorsque les hypothèses sur lesquelles elles s'appuient ne sont pas satisfaites. Dans la mesure où le compas visuel s'appuie sur la linéarité des variations des azimuts par rapport au déplacement de l'animat, il échoue lorsque cela n'est que très approximativement vrai. Ceci se produit en particulier si l'animat s'approche trop d'un amer : l'azimut associé varie beaucoup plus sur de courtes distances que pendant l'étape d'observation parce que l'amer est plus près, et la combinaison linéaire sur laquelle se fonde l'invariant n'est plus bonne.

Cela dit, dans une application robotique réelle, une technique d'agrégation assez performante (cf. section 6.4.3.1) ignorerait simplement cet amer quand l'animat s'en trouverait trop proche car les estimations de direction auxquelles il participe ne seraient pas cohérentes avec celles des autres triplets disponibles. Cependant, ce critère d'élimination ne fonctionnerait que si suffisamment d'amers sont visibles, sans quoi on risquerait de tout éliminer.

Une autre possibilité, à la place ou en complément, consisterait à exploiter la complémentarité entre les propriétés du compas visuel (observer une fois, réutiliser plusieurs fois même après un temps arbitrairement long) et celles d'autres sources d'information. Par exemple, l'odométrie (utilisée à

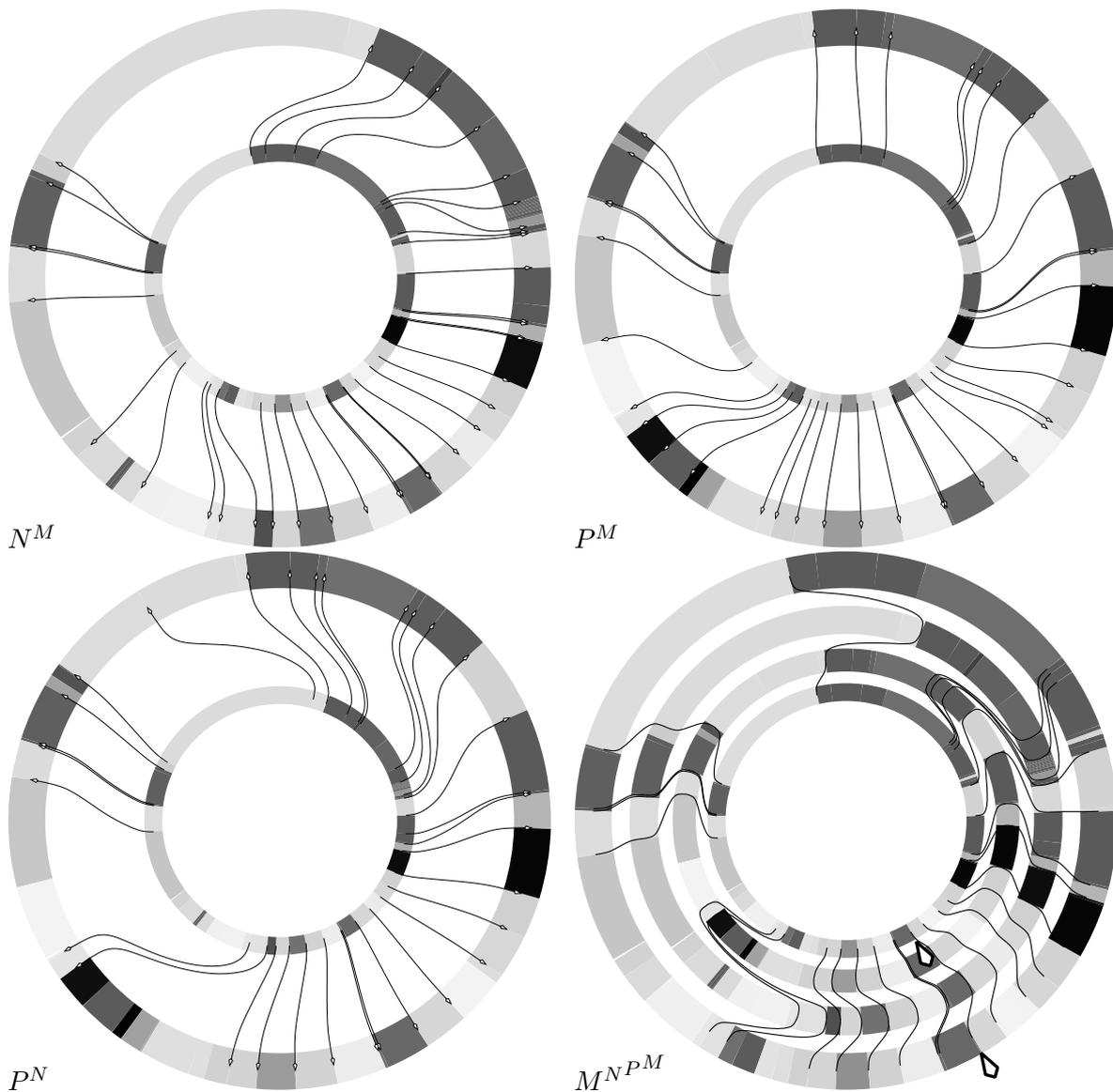


FIG. 6.9: Exemple de filtrage d'appariement par cohérence sur des données réelles. On apparie des transitions. Les trois premiers schémas représentent les trois appariements deux à deux. Le dernier schéma montre l'ensemble des paires conservées, reliées par des lignes courbes. On constate que presque toutes les paires de façon générale ont été conservées, parce que cohérentes entre les trois appariements. Par contre en haut, par exemple, une seule paire a été trouvée cohérente sur 3 à 4 candidates. Ce type de sélection a l'avantage de ne demander aucun paramètre, et d'éliminer les amers qui ne sont pas traités de façon reproductible par la segmentation ou l'appariement. Toutes les paires conservées sont ensuite parcourues pour trouver des triplets d'amers pour une étape d'observation du compas visuel. On montre aussi par une flèche en bas à droite du dernier schéma l'amer sélectionné comme étoile polaire.

courte distance pour éviter les dérives) permet de faire d'autres observations et d'étendre de proche en proche le champ couvert par le compas visuel, en stockant l'information dans une carte topologique comme l'a fait [FRA 98a] avec une méthode d'inspiration différente.

6.4.5 Observations qualitatives sur la robustesse

Nous avons observé durant nos expériences que le compas visuel est sensible aux erreurs de segmentation et d'appariement alors que les algorithmes de navigation étudiés dans cette thèse peuvent généralement fonctionner correctement malgré des erreurs de segmentation et/ou d'appariement. Nous expliquons ici cette différence qui s'avère peu gênante en pratique.

Considérons les mesures synthétiques de la capacité de navigation. Il y a une assez grande liberté de chemins possibles qui finissent par converger vers le but, même si une erreur assez importante est commise à chaque pas (il suffit qu'elle soit inférieure à $\pm 90^\circ$, cf. sections 3.4.2 et 4.5). Les mesures analytiques concernant la navigation permettent de départager des modèles qui, pour l'essentiel, réussissent de toute façon à rejoindre le but, et nous avons vu que certains modèles réussissent mieux en donnant plus de priorité à l'évitement d'obstacle qu'au rapprochement le plus direct du but (section 2.5).

En revanche on juge le compas visuel différemment. Du côté des mesures analytiques on ne s'estime satisfait que si l'erreur est beaucoup plus faible : 10° ou même 20° est beaucoup plus faible que 90° . De leur côté, les mesures synthétiques portent sur la capacité à donner une direction suffisamment précise pour restaurer une navigation efficace. Aux défaillances possibles du système de navigation s'ajoutent celles du compas visuel. Une éventuelle erreur de segmentation ou d'appariement aura des répercussions sur les deux niveaux. Une mauvaise estimation d'orientation peut fausser suffisamment la navigation pour éloigner l'animat du but.

Pour être plus précis, rappelons que le succès des modèles de navigation par mesure d'angle dépend d'un signe correct des parallaxes. Considérons un cas où un animat navigue à l'aide du modèle PV. Une segmentation imprécise perturbe les mesures de parallaxes mais, si elle n'est pas importante au point d'en changer le signe, le mouvement reste garanti de rapprocher l'animat du but. Par exemple, une erreur de 5° avec des amers à 3 mètres aboutit à une incertitude d'environ 25cm autour du but. Dans le cas où l'on cherche à la fois l'orientation et la direction à suivre avec la même information, les erreurs s'accumulent. En admettant (estimation très approximative) que l'orientation est perturbée d'autant, la dispersion est plus que doublée, si toutefois l'animat n'a pas déjà heurté un obstacle.

Nous pensons qu'il y a un parallèle intéressant avec les résultats de [SVO 96] montrant, dans un contexte de caméra projective, que du bruit dans les estimations des paramètres intrinsèques de la caméra perturbent les estimations de rotation plus que les estimations de translation.

6.4.6 Utilisation complémentaire de l'odométrie et du compas

Puisque le compas visuel ne sera pas toujours en mesure de donner des indications fiables, et parfois pas d'indication du tout, il faut pouvoir composer avec ces éventualités.

Sur sol plat et sur de courtes distances, l'odométrie est un bon indicateur des mouvements relatifs. Il est alors naturel d'utiliser le compas visuel comme correctif à l'odométrie, pour l'empêcher de dériver avec le temps, tout en continuant de s'appuyer sur cette dernière localement si le compas est incertain ou silencieux. Détaillons comment.

On maintient une information d'angle qui représente la meilleure estimation d'orientation propre actuelle. On note que cette fonctionnalité est souvent incluse dans les logiciels de contrôle de robot de niveau suffisamment élevé. En général cette information est maintenue à jour à l'aide des seules informations de mouvement instantané en provenance des capteurs odométriques. Nous ajoutons le compas visuel comme autre source de maintien de la meilleure estimation d'orientation actuelle.

Quand le compas visuel ne donne aucune indication, le robot continue de maintenir cette variable en s'appuyant exclusivement sur l'odométrie. Le compas visuel fournit une information de direction à partir du moment où au moins une marque est exploitable dans la vue actuelle. Cependant, si le nombre de marques trouvées est faible, l'indication donnée par le compas est peut-être imprécise. Il faut alors fusionner les informations en provenance des deux sources.

Nous avons choisi une fonction de fusion assez simple. Le principe est que, lorsque le compas trouve beaucoup de marques, l'information qu'il renvoie est réputée exacte et devient la meilleure estimation d'orientation actuelle. Lorsqu'on craint que l'information donnée par le compas soit imprécise, on choisit de n'appliquer qu'une correction partielle. Cependant la correction ne doit pas être trop faible, car alors le compas serait inefficace. Plutôt qu'une proportion linéaire, nous avons choisi de calculer un facteur f :

$$f = 5 \sqrt{\frac{\text{marques exploitées}}{\text{marques mémorisées}}}$$

Si $f \geq 1$, on remplace l'estimation d'orientation par celle donnée par le compas. Sinon, on applique la correction donnée par le compas en proportion de f . Cela signifie que la correction indiquée par le compas est appliquée complètement dès qu'au moins une marque sur 25 (soit 4%) est utilisable dans la vue actuelle. La proportion est représentée en figure 6.10.

6.5 Étude expérimentale

6.5.1 Dépendance par rapport au positionnement des vues d'observation

Voir figure 6.11.

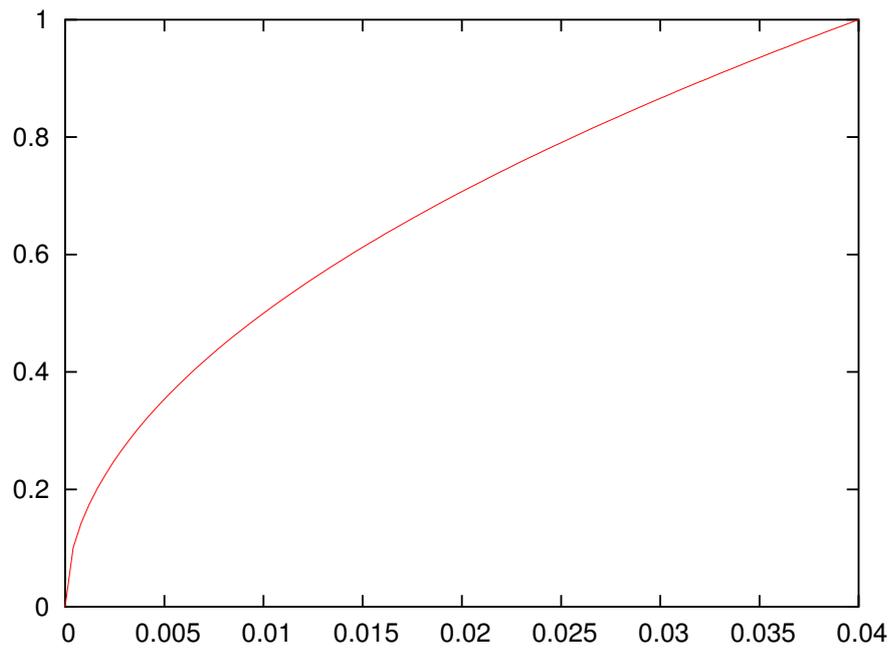


FIG. 6.10: Application partielle de la correction du compas visuel lorsque le compas ne parvient à exploiter qu'une faible proportion des marques acquises. En abscisse, la proportion du nombre de marques exploitées sur le nombre de marques mémorisées. En ordonnée, la proportion de la correction appliquée : à 0 on ne tient aucun compte du compas visuel et l'estimation d'orientation du robot reste sous le contrôle de l'odométrie ; à 1 l'orientation estimée du robot devient exactement celle donnée par le compas visuel ; dans les cas intermédiaires on réalise une somme pondérée. Lorsque le compas ne parvient à exploiter que peu de marques, son indication est probablement moins précise et on évite de lui faire aveuglément confiance. Cependant, il ne sert à rien d'appliquer des corrections trop faibles, c'est pourquoi nous avons choisi une fonction non linéaire.

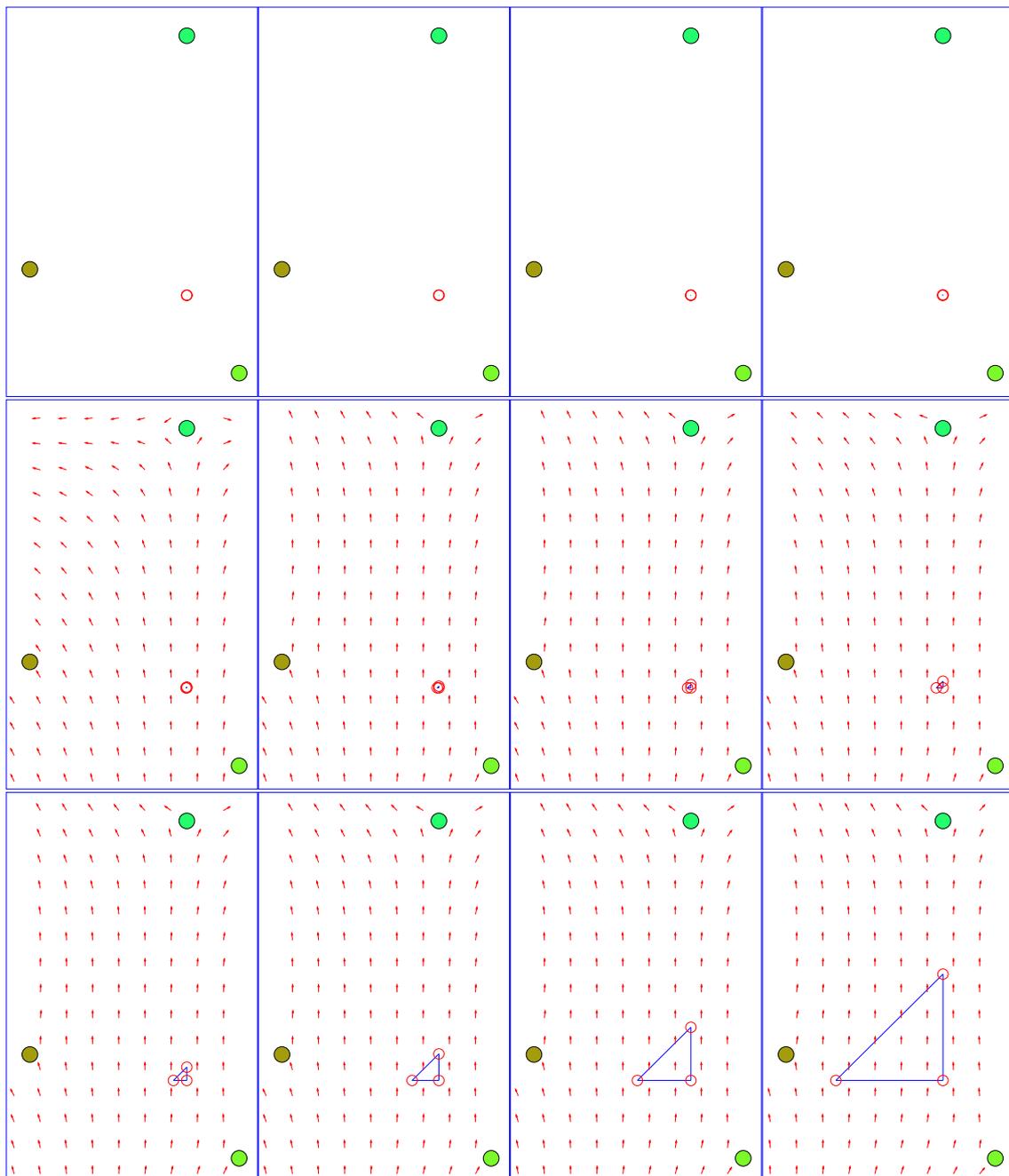


FIG. 6.11: Étude expérimentale de la dépendance du compas visuel par rapport aux points d'observation. On réalise une série d'expériences dans laquelle on utilise des points d'observation très proches, puis à chaque fois de plus en plus éloignés d'un facteur 2. La précision des mesures d'angles est limitée à 1° mais aucun bruit n'est ajouté. Les premiers champs de vecteurs sont vides car les images sont trop proches et les parallaxes sont nulles. On constate que la précision est plutôt meilleure pour de grands écarts.

6.5.1.1 Dépendance par rapport aux erreurs de mesures d'angles

Voir figures 6.11, 6.12 et 6.13.

6.6 Conclusion sur le compas visuel

Nous avons présenté un « compas visuel », une méthode originale de calcul de l'orientation d'un animat dont le principe est de découpler les effets d'une translation et d'une rotation, pour estimer cette dernière malgré la non-connaissance par l'animat de sa position exacte ni de celle des amers.

Cette méthode s'appuie sur la capacité par l'animat de mesurer les azimuts d'amers visibles, de parcourir de *courtes* distances sans perdre son orientation, et de réaliser des appariements fiables entre des vues d'orientation relative inconnue.

L'information est acquise pendant une étape d'observation nécessitant trois vues d'orientation connue prises depuis des lieux non alignés. L'information acquise peut être exploitée ensuite, arbitrairement longtemps après et éventuellement plusieurs fois, pour estimer l'orientation de l'animat quand il se trouve à nouveau dans un voisinage des lieux d'observation, sans nécessiter un passage du robot par les points précis où une observation a été faite.

Plusieurs observations indépendantes peuvent être faites, étendant potentiellement autant que souhaitée la portée du compas visuel.

La méthode ne nécessite pas un grand nombre d'éléments visibles dans l'environnement, car elle peut fonctionner à partir de 3 amers, tout en étant plus précise si davantage d'amers sont disponibles.

Un avantage de notre méthode de calcul d'orientation est sa simplicité conceptuelle et calculatoire, résultat du découplage cité plus haut. Le compas visuel peut n'apporter aucun surcoût calculatoire notable à une méthode de navigation utilisant déjà des appariements.

Grâce à cette méthode, des modèles de navigation robotique inspirés par les travaux sur les insectes, comme le modèle PV ou notre méthode par « élimination », peuvent désormais fonctionner sans indication extérieure d'orientation, mais seulement à l'aide d'informations visuelles.

La méthode est plus sensible aux erreurs d'appariement et de mesures d'angle que les méthodes de navigation dont elle tire son inspiration. Les raisons en ont été analysées mais ne menacent pas la viabilité de la procédure.

Au chapitre suivant nous décrirons un algorithme de construction de carte qui permet à l'animat de se donner de façon autonome des vues permettant de calibrer le compas visuel.

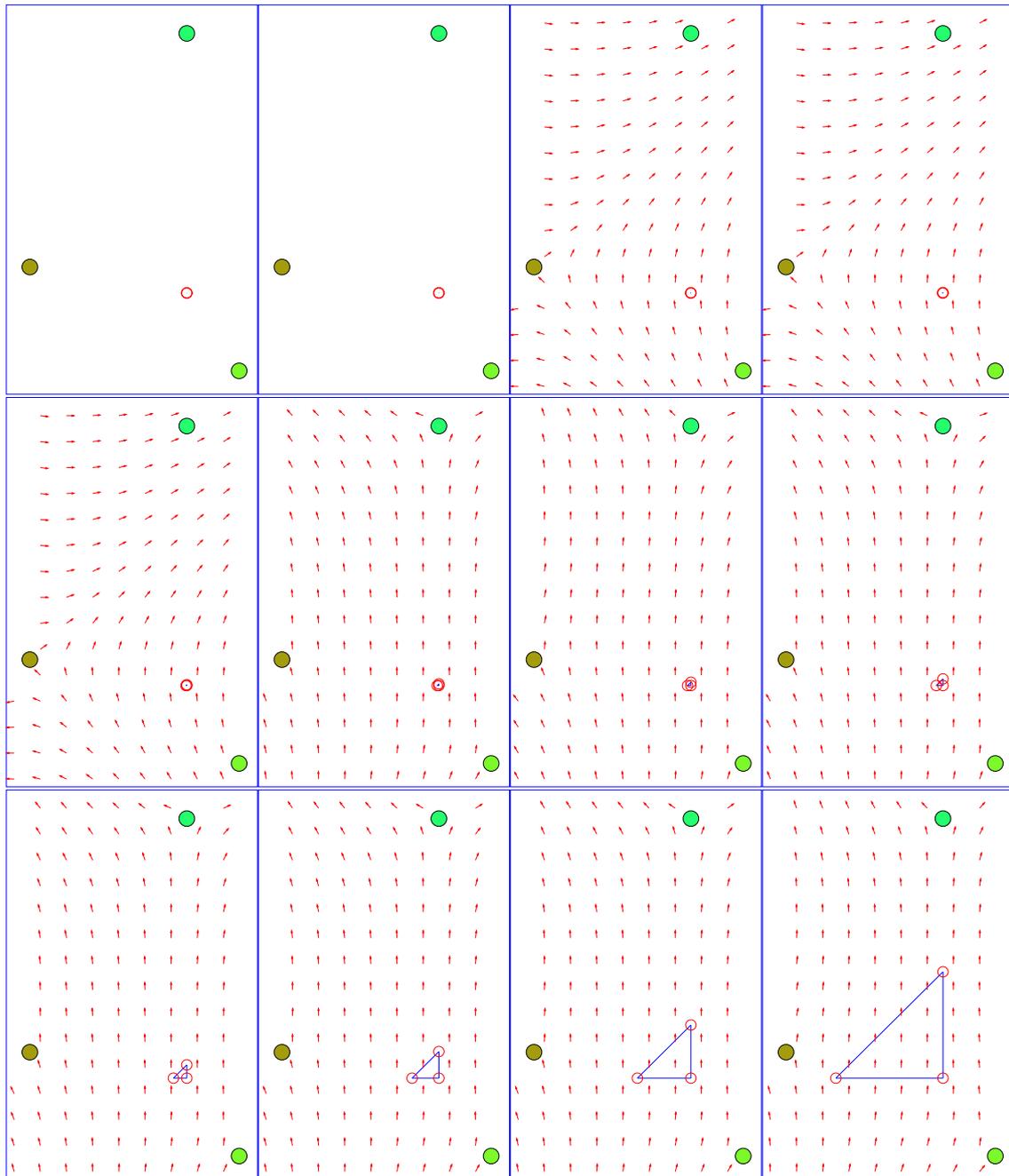


FIG. 6.12: Étude expérimentale de la résistance du compas visuel au bruit dans les mesures d'angles. On réalise une série d'expériences dans laquelle on utilise des points d'observation très proches, puis à chaque fois de plus en plus éloignés d'un facteur 2. La précision des mesures d'angles est limitée à 1° mais chaque mesure d'angle est entachée d'un bruit de 1° . Les premiers champs de vecteurs sont vides car les images sont trop proches et les parallaxes sont nulles. Puis un champ de vecteur apparaît, mais de mauvaise qualité. On constate que la précision est plutôt meilleure pour de grands écarts, ce qui n'est pas surprenant car de plus grandes parallaxes signifient un bruit relatif moindre.

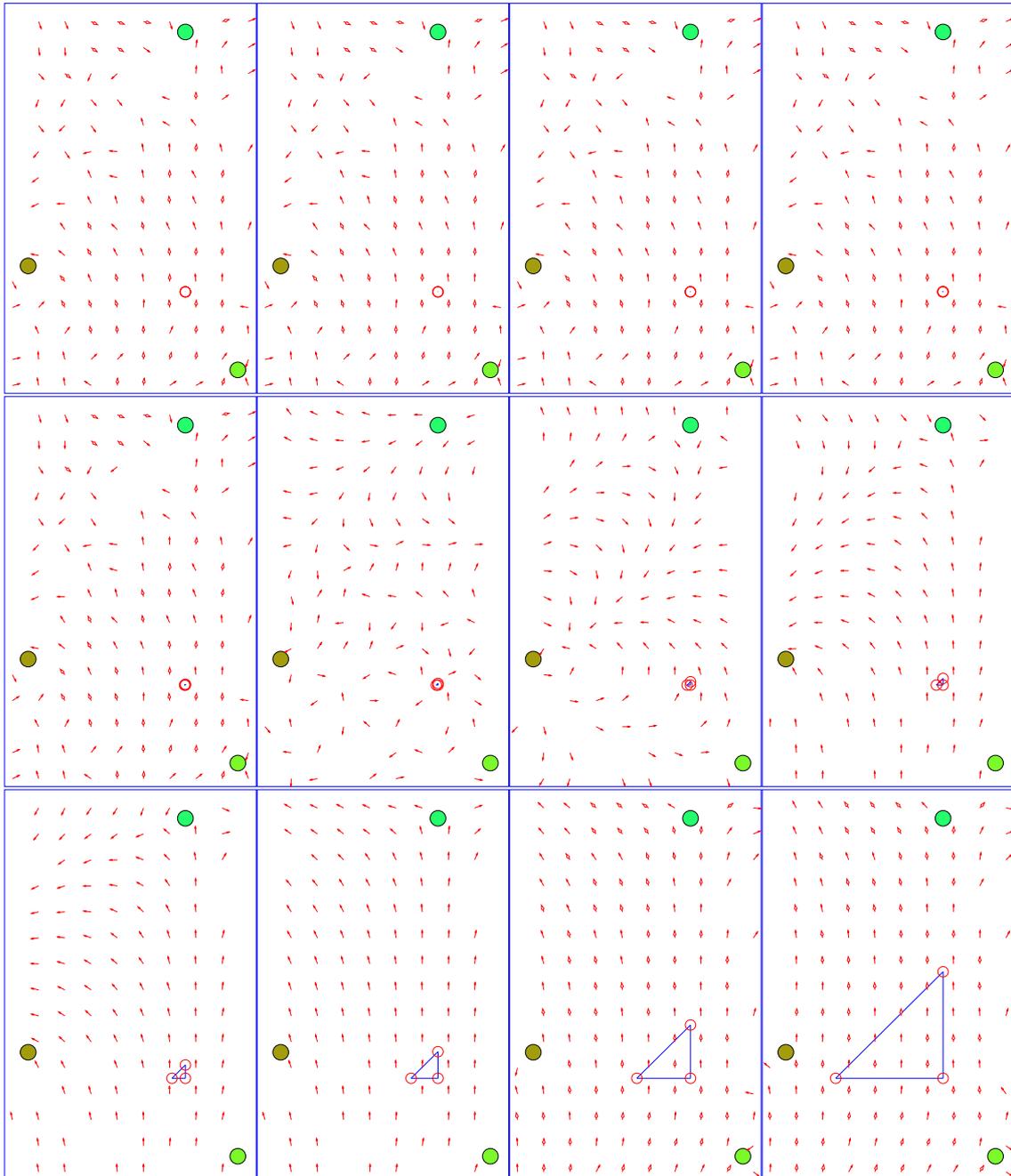


FIG. 6.13: Étude expérimentale de la résistance du compas visuel au bruit dans les mesures d'angles. On réalise une série d'expériences dans laquelle on utilise des points d'observation très proches, puis à chaque fois de plus en plus éloignés d'un facteur 2. La précision des mesures d'angles est limitée à 1° mais chaque mesure d'angle est entachée d'un bruit de 5° . Dans les premiers cas, le bruit est suffisamment fort pour créer des parallaxes aboutissant à une observation aberrante mais retenue par le modèle. Un écartement suffisant des vues d'observation finit par surmonter ce niveau de bruit et par aboutir tout de même à un champ de vecteurs convenables.

Chapitre 7

Carte topologique support du compas

Dans le chapitre précédent, nous avons exposé le fonctionnement du compas visuel et expliqué comment, à partir de trois panoramas convenables, un invariant est construit. Cependant, nous n'avons pas abordé la façon dont l'animat peut se donner lui-même des panoramas convenables en environnement *a priori* inconnu. Pour cela, une procédure d'exploration de l'environnement est nécessaire et ce chapitre décrit la procédure que nous nous sommes donnée. Dans ce chapitre, nous considérerons le compas visuel, mais les éléments restent valables pour la méthode de l'étoile polaire.

7.1 Pourquoi explorer, pourquoi une carte

Si, en une région de l'environnement, l'animat parvient à trouver trois panoramas convenables M , N et P , alors une étape d'observation est possible et l'animat pourra potentiellement retrouver son orientation dans cette région. S'il ne s'agissait que de cela, il suffirait de prendre trois vues en triangle comme cela a été expliqué en section [6.2.1.2](#) et [6.3.3.2](#).

Cependant, nous souhaitons montrer de façon constructive la possibilité, évoquée de façon théorique en section [6.3.6](#), que l'animat se donne de façon autonome les moyens d'étendre la portée du compas sur une plus grande surface que celle couverte par une étape d'observation. Pour cela, l'animat doit mémoriser plusieurs marques à partir de plusieurs étapes d'observation. Mais il doit surtout être capable de savoir dans quelle région de son environnement il se trouve, pour choisir quelles marques utiliser.

La réponse naturelle, à consulter la littérature sur la navigation autonome des animats, est la réalisation d'une carte topologique [[FIL 03](#), [MEY 03](#)].

7.2 Fondements et hypothèses

Franz [FRA 98a] a développé une méthode simple de construction de carte pour un robot utilisant des panoramas. Nous avons choisi de reprendre sa méthode et de l'adapter aux besoins du compas visuel. Nous précisons les différences au fil des explications.

Pour la commodité de l'exposé, nous noterons de la même façon un panorama et le point d'où il a été pris.

Autour de chaque panorama A que l'animat peut prendre, c'est la méthode de navigation qu'on se donne qui définit la forme et l'étendue de la surface de captation de ce panorama. La navigation locale permet de rejoindre A depuis tout point de la surface de captation, par définition de cette dernière. En pratique il existe une zone intermédiaire où la navigation réussit parfois mais pas toujours, auquel cas on restreint la définition à la portion fiable. En d'autres termes, on suppose qu'il existe autour de chaque panorama A un voisinage où la méthode de navigation permet de rejoindre A de façon fiable.

Nous dirons que deux panoramas A et B sont « situés dans la surface de captation l'un de l'autre » si et seulement si l'animat est capable d'aller de A à B et de B à A à l'aide de la navigation locale. Cette relation est réflexive (A est en relation avec A) et symétrique (on peut échanger A et B, la relation reste la même).

Maintenant considérons que l'animat explore son environnement. Nous souhaiterions qu'il puisse prendre des panoramas en divers points tels qu'il soit possible de façon fiable d'aller de n'importe lequel à n'importe quel autre, en bénéficiant du compas visuel.

Partant d'un point où il a pris un panorama A, supposons que l'animat soit capable d'estimer s'il est encore dans la surface de captation de A et quand il arrive à une limite où la fiabilité n'est plus assurée. Avant d'atteindre cette limite, il peut prendre un nouveau panorama B. Plus précisément, nous supposons nous nous donnons un critère déterminant une surface autour de A. Cette surface est telle que l'animat peut prendre un panorama B dès qu'il sort de cette surface, que B est dans la surface de captation de A et que A est lui-même dans la surface de captation de B. Alors l'animat sait se donner A et B situés dans la surface de captation l'un de l'autre, au sens défini plus haut. Nous appellerons la surface ainsi définie « surface de reconnaissance ».

Partant de A ou B, l'animat peut explorer dans des directions variées. Le critère défini plus haut lui permet régulièrement d'estimer le moment où son exploration l'a fait sortir de la surface de reconnaissance du point qu'il a quitté, mais pas encore la surface de captation associée. C'est le critère principal utilisé pour créer des nœuds et se construire progressivement une carte couvrant l'environnement. Un nœud de la carte correspond à une surface de reconnaissance autour d'un point où un panorama est mémorisé. Un lien entre deux nœuds signifie que l'animat peut utiliser la navigation locale pour passer de l'un à l'autre. Par précaution, il peut être bon de vérifier au moins une fois que le lien fonctionne dans chaque sens et permet de rejoindre effectivement le lieu reconnu.

Alors, pour passer d'un nœud à un autre quelconque de la carte, l'animat peut planifier des déplacements par des nœuds intermédiaires, utilisant à chaque fois la navigation locale. Le passage par

des sous-buts intermédiaires permet de définir que deux points d'une telle carte sont « joignables via la carte ». Cette relation hérite des propriétés de réflexivité et symétrie de la relation « situés dans la surface de captation l'un de l'autre ». Le passage par des sous-buts ajoute la transitivité : si deux points quelconques A et B sont joignables par la carte d'une part, et si B et C sont aussi joignables par la carte d'autre part, alors A et C le sont, en passant par B. Nous venons donc de définir une relation d'équivalence.

Pour revenir à la problématique du compas, nous introduisons un élément important : le choix d'une nouvelle direction d'exploration qui tend à créer des nœuds dont la position relative est compatible avec les hypothèses d'une étape d'observation (nœuds non alignés, les détails ont été exposés en sections 6.2.1.2 et 6.3.3.2). Chaque nœud de la carte satisfaisant à ces hypothèses peut faire l'objet d'une étape d'observation. Cela signifie que chacun de ces nœuds est non seulement utilisable pour la navigation locale, mais aussi que l'animat peut s'appuyer sur le compas quand il tente de rejoindre un tel nœud.

Après cette explication de principe, nous abordons dans la section suivante une description détaillée de l'algorithme suivi.

7.3 Description précise de l'algorithme

Nous avons implémenté un algorithme qui peut être décrit comme une machine à états qui opère sur la carte, assortie de quelques variables. Il y a quatre états et deux variables : la première indique la région actuellement reconnue, abrégée en RA, et la seconde le panorama qu'on cherche à joindre, abrégée en PJ. Les états et les transitions sont représentés dans le schéma 7.1.

Nous décrivons maintenant les 4 états, leur but, les détails et les raisons de leur implémentation.

Les nœuds de la carte topologique que nous construisons ont un rôle particulier : chaque nœud M représente à la fois un point précis de l'environnement (le point d'où a été prise la vue associée au nœud), et la surface de reconnaissance associée, ensemble des points reconnus comme proches (par la mesure des parallaxes) de M et ressemblant plus à M qu'à tout autre point. La navigation locale permet, si tout se passe bien, de passer de n'importe quel point de cette surface associée au point M , autrement dit la région couverte par un nœud est incluse dans la surface de captation du panorama associé au nœud.

Pour bien comprendre les explications suivantes, il sera important de conserver à l'esprit qu'à chaque nœud est associé un point et une région. En particulier, la variable RA pointe sur un nœud de la carte et représente toujours la région dans laquelle l'animat estime être. La variable PJ pointe sur un autre nœud de la carte, et représente le point qu'on veut joindre (le point précis où le panorama de ce nœud a été pris).

Le premier état dans lequel se trouve l'algorithme est naturellement le démarrage.

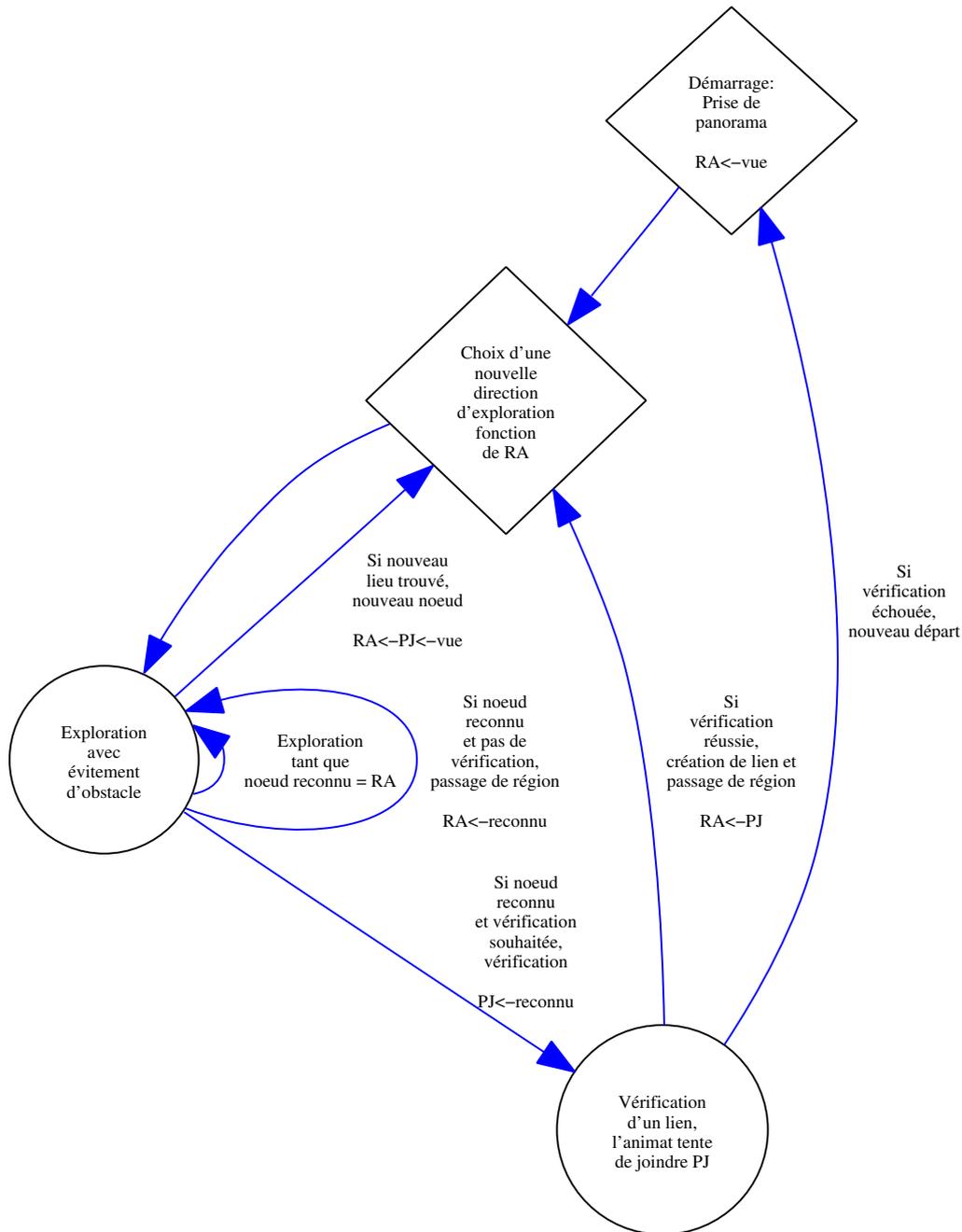


FIG. 7.1: Machine à états utilisée pour explorer l'environnement et construire la carte qui sert de soutien au compas visuel. Les nœuds en losange représentent des états ponctuels, les nœuds circulaires représentent des états prolongés dans le temps. Une légende de lien est toujours à droite du lien concerné. Les modifications des variables sont indiquées : RA=Région Actuelle (le nœud du graphe, surface de captation attachée à un panorama, où l'animat estime se trouver), PJ=Panorama à Joindre (le nœud du graphe, point où le panorama a été pris, que l'animat estime devoir rejoindre). En sortie de l'étape d'exploration, une vérification est souhaitée quand l'animat n'a jamais réussi de trajet entre RA et PJ ; sinon l'animat suppose simplement qu'il est entré dans le bassin d'attraction de PJ. L'animat ne connaît jamais précisément la surface de captation, mais il utilise un critère assez strict qui aboutit à créer un nouveau nœud avant qu'il risque de sortir de cette surface (section 7.3.3).

7.3.1 Démarrage

Pour commencer la carte, l'animat prend un premier panorama. La région où il se trouve est naturellement celle représentée par ce dernier panorama. On fait donc pointer RA sur ce panorama pour signifier ce fait.

Après le démarrage, l'animat passe à l'état dans lequel il choisit une nouvelle direction d'exploration.

7.3.2 Choix d'une nouvelle direction d'exploration

Cet état a un rôle plus important dans notre application que dans celle de Franz. Du choix d'une nouvelle direction d'exploration, après avoir découvert un nouveau lieu ou après être retourné en un lieu connu, dépendent la façon dont la carte va s'étendre et la répartition relative des nœuds de la carte. Dans notre cas, nous avons besoin de nœuds non alignés mais faisant un angle.

Suivant l'exemple de Franz, nous avons testé en premier la méthode qui consiste à tourner d'un angle fixé de 90° à chaque passage par cet état. Cette méthode convient pour commencer une carte en environnement assez ouvert. Nous avons aussi essayé la méthode qui consiste à prendre une direction complètement au hasard.

Nos expériences préliminaires nous ont montré que ces solutions simples ne suffisent pas. Le modèle de Franz affine le choix quand l'animat part d'un lieu qui possède déjà au moins deux liaisons avec des nœuds voisins. Il choisit la direction la plus éloignée des directions de ces liaisons, c'est-à-dire le milieu de l'angle le plus ouvert entre les liaisons existantes. En environnement assez ouvert ce choix semble bon. Mais des expériences préliminaires dans notre environnement nous ont suggéré que ce choix pouvait conduire à réessayer plusieurs fois d'explorer une direction qui n'aboutit pas à la création d'un nouveau nœud.

Nous avons donc conçu et testé une procédure originale qui consiste à explorer des directions variées, qu'elles aboutissent ou non à la création de nœuds. Le principe est de noter les directions déjà empruntées, et d'en choisir une relativement différente de toutes celles déjà empruntées.

Les règles suivantes sont appliquées pour la mémorisation des directions déjà explorées depuis un point donné (illustrées en figure 7.2) :

- l'animat mémorise pour chaque nœud de la carte les directions qui ont déjà été explorées ;
- la mémorisation se fait sous la forme d'un anneau de pénalité similaire à celui utilisé dans la méthode par élimination (section 5.3) ;
- quand l'animat quitte pour la première fois ce point, les pénalités sont à zéro dans toutes les directions ;
- à chaque fois qu'une exploration est lancée dans une direction choisie, un profil de pénalité est ajouté à cet anneau ;
- le profil ajouté pénalise particulièrement la direction qui vient d'être choisie, mais ne pénalise pas les directions situées à $+120^\circ$ et -120° .

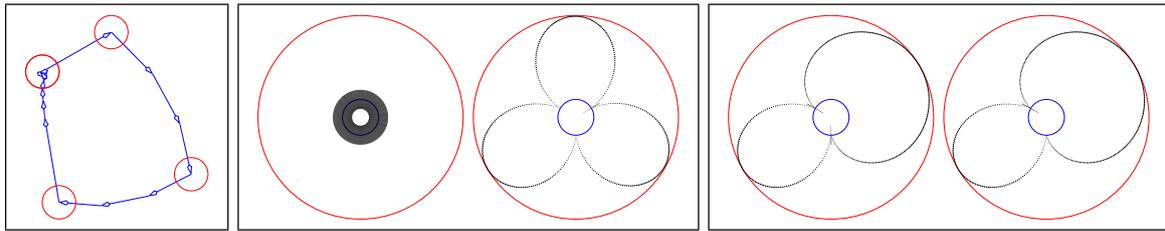


FIG. 7.2: Illustration du choix d'une nouvelle direction d'exploration. Cadre de gauche : le contexte. L'animat part du point de gauche, explore vers le nord-est, le sud-est, le sud, et revient au départ pour explorer à nouveau. Autres cadres : représentation polaire de directions pénalisées, servant à choisir une nouvelle direction d'exploration, dans le cas du premier nœud. En une direction donnée, la distance au cercle central est la pénalité normalisée de sorte que la direction la plus pénalisée touche le cercle extérieur. Un trait coupant le petit cercle distingue les directions de pénalité exactement nulle. Cadre du milieu : premier départ. Cercle de gauche : aucune pénalité mémorisée. À droite : on ajoute une (légère) pénalité pour tenir compte de ce que l'animat est tourné vers le nord, invitant à plutôt tourner à gauche ou à droite de 60° . La direction choisie est de tourner à droite de 60° (est-nord-est). Cadre de droite : après le retour au point de départ, second choix d'une direction d'exploration. À gauche : la direction prise la première fois est désormais pénalisée. Deux directions ne sont pas pénalisées du tout : ouest-nord-ouest, et sud. À droite : le choix est départagé par le fait que l'animat est revenu par le sud ; il prend ouest-nord-ouest. Les deux schémas semblent identiques parce que la pénalité ajoutée pour départager est très petite devant celle qui considère les directions explorées.

Quand l'animat a besoin de choisir une nouvelle direction d'exploration, il prend une décision qui dépend essentiellement de ce profil. La première fois, le profil de pénalité est nul, et on souhaite que l'animat ait tendance à tourner de 60° . Pour cela, la direction choisie dépend aussi de la direction de laquelle l'animat provient. On fait une copie du profil de pénalité, à laquelle on ajoute les éléments suivants :

- les directions autres que correspondant à un virage de 60° , -60° ou 180° (faire un demi-tour) sont légèrement pénalisées ;
- les directions correspondant à tourner de plus de 90° sont très légèrement pénalisées.

Compte tenu de l'ensemble, la direction la moins pénalisée est choisie.

Ce système tend à favoriser une structure avec des branches à 120° les unes des autres, tout en s'adaptant aux contraintes de l'environnement.

La différence la plus importante avec la solution de Franz est qu'une direction d'exploration, une fois choisie, est pénalisée pour les fois suivantes qu'elle ait abouti ou non à la création d'un nouveau nœud. Nous pensons que notre choix permet une plus grande variété des directions choisies, en évitant de retenter en boucle des directions ne menant pas à la création d'un nouveau nœud.

Après le choix de la nouvelle direction d'exploration, l'animat passe à l'état où il explore.

7.3.3 Exploration avec évitement d'obstacle

Le but de cette étape d'exploration est de quitter un lieu pour en découvrir de nouveaux, tout en évitant les obstacles. Cet état peut se maintenir de façon prolongée dans le temps.

La partie évitement d'obstacle utilise un sonar simulé : l'animat évite les directions dans lesquelles des obstacles sont relativement proches. Secondairement si possible, il évite aussi les directions qui impliquent des virages de plus de 60° de la direction à laquelle il fait face actuellement. Cela lui permet de suivre les couloirs mais aussi de faire demi-tour s'il arrive dans un cul-de-sac.

La partie exploration mérite plus d'explications. L'animat doit estimer à chaque instant s'il est toujours dans la surface de captation du lieu qu'il quitte, ou bien s'il est dans un autre lieu qu'il connaît, ou en un lieu inconnu. Selon les cas, cet état se maintiendra, ou bien on basculera dans un autre état. Ce nouvel état peut être le choix d'une nouvelle direction (si l'animat estime être dans un nouveau lieu et créer un nouveau nœud) ou bien la vérification d'un lien (si l'animat estime être passé à une autre région et le vérifie en tentant de joindre le point associé au nœud estimé reconnu). Nous avons défini un algorithme pour distinguer ces situations les unes des autres, que nous présentons ici.

Tant que l'état « exploration avec évitement d'obstacle » se maintient, après chaque déplacement l'animat prend un panorama et va déterminer, de la façon définie ci-dessous, si le nouveau panorama ressemble suffisamment à l'un des panoramas connus, ou non.

Pour cela il faut passer en revue des panoramas connus. Pour des raisons de performance, on pourrait définir un critère permettant d'extraire rapidement un sous-ensemble de panoramas mémorisés susceptibles de correspondre au lieu actuel (par exemple en comparant une petite signature extraite de chaque panorama, faisant intervenir les quelques amers les plus saillants). Si la carte est petite on peut ignorer cette étape de présélection et considérer tous les nœuds de la carte comme candidats potentiels que l'on passe donc en revue.

Pour effectuer une comparaison entre un panorama mémorisé et le panorama actuel on commence par les appariers. Si le panorama mémorisé n'a pas au moins 4 amers appariés avec le panorama actuel, il est ignoré et on reprend avec le candidat suivant, sinon on continue. On parcourt alors les parallaxes entre amers appariés, en valeur absolue. Si au moins une des parallaxes est supérieure à un seuil que nous avons fixé à 45° , ce panorama mémorisé est ignoré. Ce dernier critère est important. C'est lui qui limite la surface de reconnaissance, assurant que cette dernière est dans la surface de captation. C'est aussi une différence par rapport à la méthode de Franz. Ce dernier utilise une définition différente de la distance entre deux panoramas, définie directement sur les mesures de luminosité (niveaux de gris) et non sur des parallaxes.

Si le panorama mémorisé n'a pas été éliminé, on note la parallaxe maximale qui est associée. Ainsi, à la fin du passage en revue, on disposera d'une sélection de panoramas et de leur « parallaxe maximale ».

Après avoir passé en revue les panoramas mémorisés, il y a deux possibilités : soit il y a au moins un panorama qui a satisfait à tous les critères, soit il n'y en a aucun. S'il y en a au moins un, on

sélectionne celui qui a la plus petite « parallaxe maximale », et on ignore les autres¹.

S'il y a un panorama sélectionné Y , et que c'est celui que désigne la variable « région actuelle » RA , alors l'animat considère qu'il est toujours dans la région et continue son exploration.

S'il y a un panorama sélectionné Y , et s'il n'est pas celui associé à la région actuelle RA , alors l'animat considère qu'il est peut-être arrivé dans la région associée à Y . Si l'animat n'a jamais effectué un trajet avec succès d'un point de la surface de reconnaissance du nœud RA au point Y , il va essayer de rejoindre le point Y en passant dans l'état « vérification de lien ». Si au contraire un trajet $RA \rightarrow Y$ a déjà été réalisé avec succès, alors l'animat estime directement être dans la région de Y et continue son exploration. Cela lui permet, quand la carte commence à être dense, de traverser librement la zone cartographiée sans refaire une étape de vérification à chaque fois qu'il passe d'une région de la carte à une région voisine.

Sinon, l'animat est dans le cas où toutes les vues ont été éliminées : aucune vue ne contient au moins 4 amers appariés avec la vue actuelle tous montrant une parallaxe inférieure au seuil.

Avant de traiter ce cas, précisons que, lorsque l'état « exploration avec évitement d'obstacle » est activé, l'animat sait dans quelle région il se trouve (indiqué par RA), soit parce qu'il l'a reconnue, soit parce qu'il démarre une nouvelle carte. Les premiers panoramas qu'il observe doivent avoir beaucoup en commun avec le panorama mémorisé représentant la région actuelle, puis de moins en moins à mesure qu'il s'éloigne.

On distingue alors deux sous-cas. S'il y avait des panoramas qui avaient suffisamment d'amers en commun, mais qui ont été éliminés pour cause de trop grandes parallaxes, on considère que l'animat risque de quitter la surface de captation (même s'il y est probablement encore). Il crée immédiatement un nouveau nœud sur la carte, et passe dans l'état de choix d'une nouvelle direction d'exploration. Il reste l'autre sous-cas : celui de l'absence de panoramas ayant suffisamment d'amers en commun.

Le dernier sous-cas signifie, soit que l'animat arrive dans une zone très pauvre en amers, soit qu'il franchit un seuil où l'environnement change subitement (seuil de porte par exemple). Dans nos expériences simulées, nous avons choisi de laisser l'animat continuer d'explorer sans créer de nouveau lieu. Nous n'avons pas expérimenté le cas où l'environnement change subitement. Une solution raisonnable nous semble de créer un nouveau nœud et de s'assurer au moins une fois qu'on peut revenir dans la région précédente en faisant un pas en arrière et en essayant de revenir à la dernière région reconnue (état « vérification d'un lien »).

7.3.4 Vérification d'un lien

L'animat arrive dans cet état lorsqu'il estime, d'après le dernier panorama observé, qu'il est proche d'une autre région que RA , région qu'il associe alors à la variable PJ . Le critère qui lui a permis d'arriver ici est que, parmi les panoramas qu'il a comparés à celui visible actuellement – qui ont au

¹Le but de cette notion apparemment paradoxale de « plus petite parallaxe maximale » est de trouver le panorama ayant les plus petites parallaxes. Or, pour un panorama donné, pour savoir par combien on peut majorer les parallaxes, on regarde la parallaxe la plus grande dans ce panorama.

moins 4 amers en commun, tous de parallaxe inférieure au seuil critique – celui qui a les plus faibles parallaxes n'est pas celui associé à RA . En outre la vérification a été déclenchée seulement si l'animat n'a jamais effectué de trajet $RA \rightarrow PJ$ avec succès.

Cet état se prolonge dans le temps, car il consiste à lancer une procédure de navigation locale pour rejoindre le point dont on pense être proche. La navigation locale est couverte par les chapitres 5 et 6. Pour garantir que la navigation ne se prolonge pas trop, un nombre maximum de pas est défini. Ce nombre est fixé pour une expérience donnée.

L'issue de cette procédure peut être la réussite, si l'animat atteint le point demandé dans le nombre de pas impartis (cf. section 5.5). Alors, un nouveau lien est créé dans la carte topologique et l'animat note qu'il est arrivé dans la région visée.

Si la tentative de rejoindre le point qu'on pensait avoir reconnu échoue (cf. section 5.5), alors on ne crée pas le lien espéré. Ce cas est difficile car l'animat est peut-être perdu ou, si c'est un robot réel, mécaniquement coincé. Nous avons repris la solution utilisée par Franz : l'animat passe dans l'état « démarrage ». Il prend alors un nouveau panorama sur place et continue d'explorer.

Nous pensons que cette étape pourrait être affinée, car le nouveau nœud créé à cette occasion est rarement pertinent. Par exemple, si au cours de la vérification d'un lien de A vers B, l'animat estime ne plus être dans la surface de reconnaissance de B mais d'un autre panorama C, la vérification de lien pourrait être interrompue. Cette variante n'a pas été testée.

7.3.5 Étapes d'observation pour le compas visuel

Nous expliquons dans cette section quand et comment l'animat sélectionne des vues pour y appliquer une étape d'observation du compas.

Nous avons défini une procédure appliquée, au moment où on crée une liaison entre deux nœuds A et B, à chacun des deux panoramas associés. Le principe est de trouver un troisième panorama satisfaisant aux critères d'une étape d'observation. Pour cela, il est nécessaire de savoir dans quel azimut se trouve B vu de A. Une approximation de cette direction est donnée par la méthode de navigation locale : c'est la direction dans laquelle l'animat ferait le premier pas en partant de A pour rejoindre B.

L'animat parcourt la liste des nœuds reliés à l'extrémité A. Pour chaque nœud (et panorama) C trouvé on estime de la même façon son azimut vu depuis A. Si l'angle \widehat{BAC} est entre 45 et 135 degrés, une étape d'observation est réalisée.

Remarquons que d'autres panoramas que ceux disponibles comme nœuds de la carte peuvent être utilisés pour les étapes d'observation. Par exemple, à la création d'un nœud A, on peut considérer dans le rôle de B l'un des derniers panoramas pris pendant l'exploration avec évitement d'obstacle qui a mené à A et, dans le rôle de C, un des premiers panoramas pris après à nouveau quelques instants d'exploration.

Remarquons que l'approche choisie est latente, c'est-à-dire que l'animat n'attend pas d'avoir be-

soin de rejoindre un point pour réaliser une étape d'observation de compas. Nous aurions pu simplement construire la carte, et effectuer des observations pour le compas à mesure de la nécessité de naviguer : pour rejoindre un point auquel aucune marque n'est associée, on commence par parcourir l'ensemble des liens avec les nœuds voisins pour réaliser des étapes d'observation.

Il y a d'autres possibilités. L'essentiel ici est d'avoir montré de façon constructive qu'il est possible de déterminer automatiquement des vues satisfaisantes pour des étapes d'observation, afin que les nœuds de la carte comprennent aussi des marques permettant à l'animat de conserver son orientation.

7.4 Exemples de cartes construites

Nous présentons dans cette section quelques exemples de cartes construites à l'aide de l'algorithme décrit dans ce chapitre.

Un cercle représente un nœud dans la carte, un point d'où l'animat a mémorisé un panorama. Il y a deux circonstances où cela se produit : au démarrage de la carte (y compris un redémarrage consécutif à un échec à rejoindre un point) et lorsque l'animat estime au cours d'une exploration qu'il arrive en limite de surface de reconnaissance. Souvent, un anneau est dans le cercle. Un tel nœud est associé à des marques utilisables par le compas visuel. Dans certains schémas, on a désigné par des polygones ombrés les triplets de vues ayant servi à l'étape d'observation. Un exemple est donné en figure 7.3.

Une flèche d'un nœud vers un autre signifie que l'animat est au moins une fois parti explorer depuis le premier panorama et a tenté de rejoindre le second avec succès en utilisant la méthode de navigation locale. Il y a en général une flèche dans chaque sens entre deux nœuds.

Il peut y avoir des nœuds non reliés. Cela signifie que l'animat n'a jamais essayé de passer de ou vers un tel nœud par navigation locale. Il peut s'agir d'un chemin que l'animat n'a fait qu'explorer sans même le reprendre en sens inverse. Ou bien, il s'agit d'un nœud créé suite à une tentative échouée de vérification de lien, et jamais revisité (nous avons vu plus haut que ce cas pourrait être évité).

Les exemples sont présentés dans les figures 7.4 à 7.6.

7.5 Effet d'une perturbation d'orientation

Une estimation d'orientation telle que donnée par l'odométrie sur un sol plan et régulier a pour caractéristique que l'animat peut naviguer pendant une certaine distance avec une bonne précision sur la connaissance de son orientation. En revanche, les erreurs s'accumulant, l'estimation devient de plus en plus biaisée à mesure que la distance parcourue s'accumule.

S'appuyer sur l'odométrie est donc suffisant pour obtenir les premières marques à condition que la distance à parcourir soit suffisamment faible. Dans notre contexte, il s'agit typiquement de parcourir quelques mètres, alors que la dérive devient appréciable au bout de quelques dizaines de mètres. Après ces quelques mètres, les premières marques sont acquises. L'animat en bénéficie lorsqu'il rejoint le

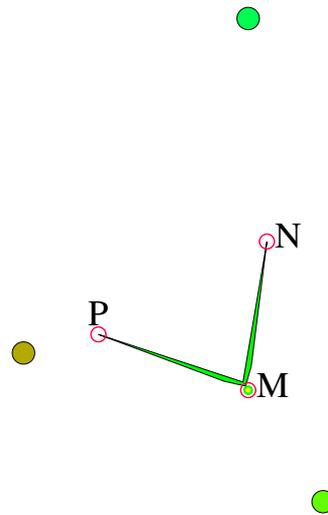


FIG. 7.3: Exemple de représentation d'une étape d'observation. Une représentation par un triangle ou par des flèches est trop symétrique et ne permet pas de distinguer lorsque une ou deux vues sont communes à plusieurs étapes d'observation. Nous avons donc choisi de montrer un polygone : il permet de bien distinguer la vue M (du côté épais) et les vues N et P (pointes).

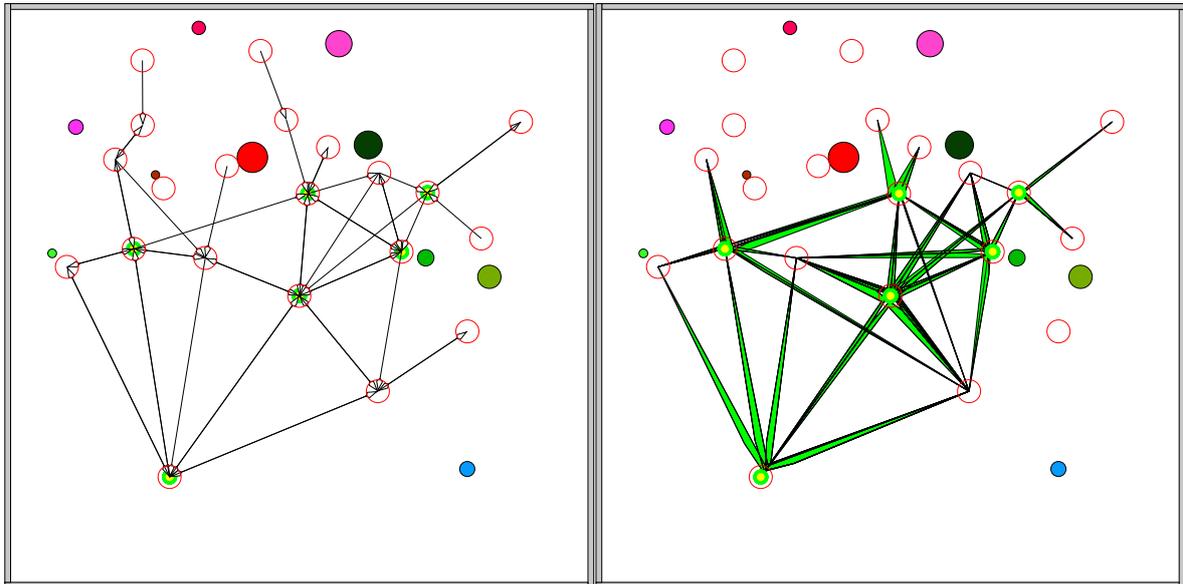


FIG. 7.4: Exemple de carte construite, dans un monde à 10 amers. Voir le texte pour la signification des cercles, disques et lignes. La version de droite montre les triplets de vues qui ont servi aux étapes d'observation. Il est intéressant de constater comme les nœuds se répartissent tout autour des amers, alors qu'ils sont clairsemés dans l'espace vide au sud-ouest. Cet effet, observé aussi par Franz, est intéressant car il correspond précisément au fait que la parallaxe d'un amer proche dépend fortement de la position de l'animat. Ainsi, les nœuds sont plus rapprochés là où ils ont besoin de l'être, en particulier pour le compas visuel.

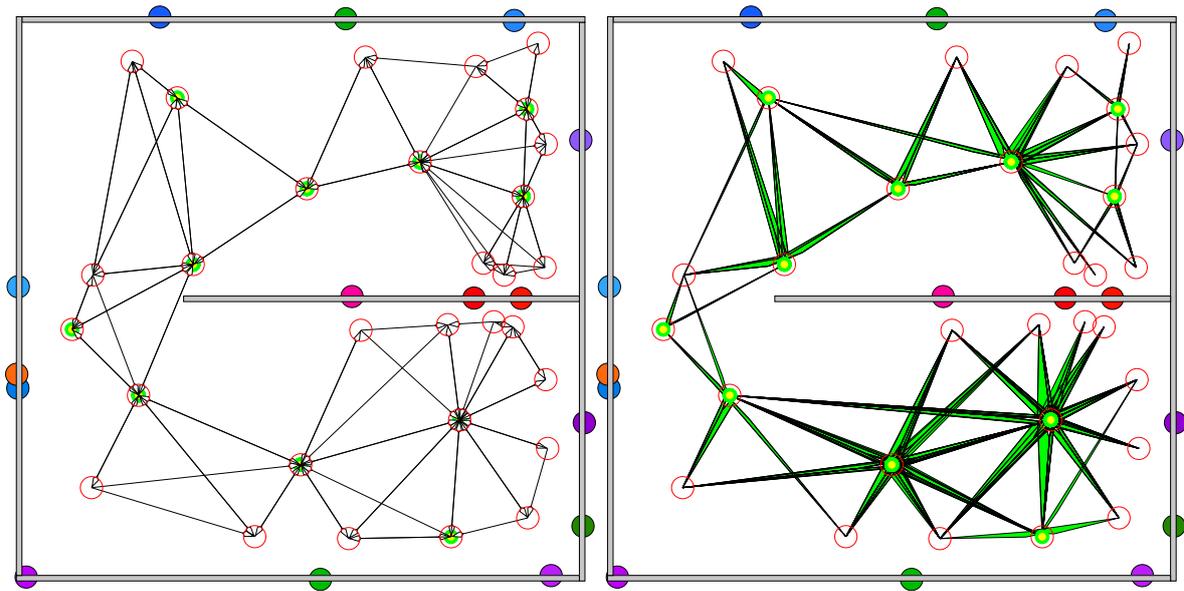


FIG. 7.5: Exemple de carte construite dans un monde structuré en deux parties. Là encore, on constate que les nœuds sont plus denses près des amers.

point associé ou simplement lorsqu'il se traverse la région couverte (la surface de reconnaissance). L'animat recalcule donc en continu son estimation d'orientation et évite ainsi le problème de dépendre d'une estimation qui dérive progressivement. Lorsqu'il explore, il crée un nouveau nœud dès qu'il sort de la surface de reconnaissance. Il s'agit, là aussi, d'un cas où quelques mètres à parcourir suffisent.

La procédure actuelle ne garantit pas que la distance parcourue hors de la couverture du compas soit bornée. Il peut être nécessaire en principe d'introduire un mécanisme qui tend à ramener l'animat régulièrement vers une zone couverte par le compas, pour éviter qu'il n'explore trop longtemps, au risque de laisser se dégrader sa connaissance de son orientation. Ceci ne remet pas en cause la capacité à construire une carte en s'appuyant localement sur l'odométrie.

7.6 Bilan et extensions possibles

La procédure de construction de carte que nous avons décrite ici montre que l'animat peut se donner lui-même, au cours de son exploration de l'environnement, une carte munie de nœuds lui permettant de naviguer avec le compas visuel. La problématique de la construction de carte et de la navigation par carte peut être menée plus loin, comme nous l'évoquons dans cette section.

La procédure décrite aboutit à la création de liens entre nœuds lorsque l'animat vérifie la possibilité d'appliquer la procédure de navigation locale directement d'un point à l'autre. Mais aucun mécanisme ne favorise cet événement. Il peut rester pendant longtemps des liens non vérifiés. Il pourrait être intéressant qu'un mécanisme d'exploration à l'échelle de la carte guide l'animat vers des points où peu de directions ont été explorées.

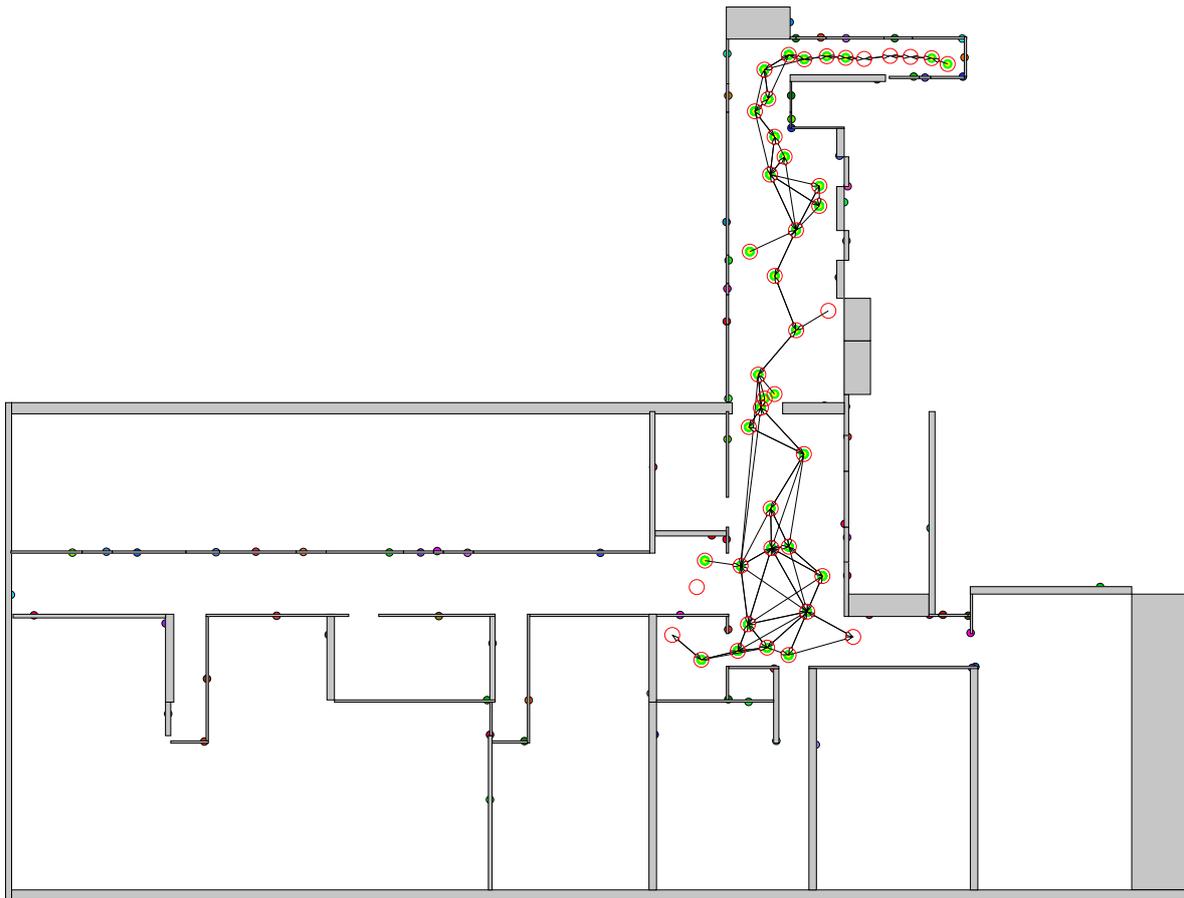


FIG. 7.6: Exemple de carte construite dans un environnement simulé qui correspond approximativement aux couloirs du laboratoire. La carte couvre bien la surface explorée par l'animat, mais ce dernier n'a pas exploré tout l'environnement. En laissant l'expérience se dérouler plus longtemps, ou bien en modifiant les paramètres de l'exploration avec évitement d'obstacle, l'animat peut construire une carte couvrant toute la surface qui lui est accessible.

De même, aucun mécanisme n'assure que la carte couvre tout l'environnement. Elle ne grandit essentiellement qu'en fonction de la trajectoire que l'animat prend lors de l'exploration avec évitement d'obstacles. On pourrait par exemple mémoriser en chaque nœud s'il reste de l'espace libre non exploré, soit à l'aide du sonar, soit, pour rester dans l'idée de n'utiliser que des angles, parce que des amers lointains sont reconnaissables à leur faible parallaxe quand l'animat se déplace. Alors, un mécanisme d'exploration à l'échelle de la carte pourrait guider l'animat vers des points où se trouveraient des espaces non explorés.

Il n'existe pas de circonstance où un nœud de la carte est détruit (oublié). Cela n'est pas très important dans la mesure où l'algorithme ne crée pas de nœud s'il n'est pas déjà suffisamment loin des nœuds existants. En principe, il ne peut donc pas créer un grand nombre de nœud inutiles. L'exception est la création d'un nœud après échec de vérification d'un lien, point qui pourrait être remis en question.

7.7 Conclusion

Nous avons conçu, implémenté et testé une variation sur la méthode de Franz de construction de carte topologique, afin de servir de support au compas visuel.

Elle démontre de façon constructive qu'il est possible pour l'animat de se donner au cours de l'exploration de l'environnement des vues utilisables pour l'étape d'observation du compas visuel.

Nous avons expliqué que l'animat peut s'appuyer sur l'odométrie pour construire la carte et qu'il bénéficie du compas dès que la première étape d'observation a permis d'acquérir des marques. Ceci permet de construire la suite de la carte pendant une durée possiblement longue sans craindre de dérive due à l'odométrie.

Il aurait été intéressant de tester cela, en particulier sur robot réel, mais cela n'a pas été possible faute de temps.

Conclusion sur la présentation de notre modèle

Nous venons d'achever la présentation détaillée de notre modèle.

Nous avons successivement présenté notre choix d'utiliser des panoramas segmentés en couleurs, décrit la méthode d'appariement que nous avons utilisée, comment nous avons adapté l'algorithme de la programmation dynamique à la topologie circulaire de nos appariements. Nous avons expliqué notre façon de calculer la direction à suivre par élimination. Nous avons aussi expliqué comment nous avons résolu les questions de longueur des pas à effectuer et comment notre animat peut estimer s'il est arrivé au but.

Ensuite, nous avons exposé notre méthode d'orientation, dite « compas visuel », qui réalise, *via* une approximation de linéarisation, un découplage entre déplacement et rotation, et permet ainsi d'estimer l'orientation de l'animat notamment lorsqu'il cherche à rejoindre un point.

Nous avons montré de façon constructive comment l'animat peut se construire une carte topologique de l'environnement, lui permettant de bénéficier du compas visuel sur des surfaces étendues.

Dans cette partie, nous avons cité les travaux antérieurs sur lesquels nous nous sommes appuyés, mais nous n'avons pas réalisé de comparaison détaillée de notre modèle avec ceux que nous avons cités, celle-ci viendra en partie [IV](#). C'est pourquoi nous abordons maintenant la partie expérimentale de la thèse, dans laquelle nous décrivons comment nous avons validé nos travaux à la fois en simulation et sur robot réel.

Troisième partie

Expérimentations

Introduction à la partie expérimentale

Après avoir exposé et expliqué le fonctionnement interne de chaque élément de la chaîne réalisant la navigation, nous abordons maintenant la validation expérimentale de nos travaux.

Dans un premier chapitre nous décrivons le simulateur dédié que nous avons développé et les expériences simulées que nous avons effectuées.

Dans un second chapitre nous décrivons quelques-unes de nos expériences sur robot réel.

Chapitre 8

Simulations

La simulation permet de voir fonctionner les algorithmes que nous concevons dans des conditions plus faciles et reproductibles qu'avec un robot réel. Une fois qu'un simulateur est disponible, on peut réaliser facilement des séries d'expériences et éventuellement générer des statistiques.

Dans la première section 8.1, nous allons décrire le simulateur que nous avons utilisé.

Le simulateur a été pour nous un outil de travail important, qui nous a servi à tester de nombreuses intuitions en continu le long de notre travail. Le nombre d'expériences que nous avons réalisées est bien trop grand pour les présenter toutes ici. C'est pourquoi nous nous limiterons à présenter quelques expériences significatives pour la validation de notre travail.

La première, en section 8.2, a servi essentiellement à valider l'utilisation de panoramas en couleurs et leur exploitation effective par la méthode d'appariement par programmation dynamique, dans le cas de panoramas ne faisant pas le tour complet.

La section suivante 8.3 aborde des expériences simulées qui permettent de valider la navigation à l'aide du compas visuel et d'un capteur panoramique. Cette série d'expériences valide notre algorithme d'appariement dans le cas de panoramas complets, avec l'estimation de l'orientation de l'animat par le compas visuel.

8.1 Notre simulateur

Nous nous sommes orienté vers une simulation simple et proche du modèle de navigation.

Il fallait choisir entre développer un simulateur simple à partir de rien ou réutiliser un simulateur existant.

8.1.1 Considérations techniques

Parfois un logiciel proche de l'outil désiré existe déjà, avec possibilité de l'adapter à ses besoins. Typiquement, il s'agit de logiciel libre sous licence GPL. La réutilisation d'un tel code extérieur offre des avantages :

- fonctionnalités nécessaires déjà présentes, donc économie du temps de développement correspondant ;
- fonctionnalités plus évoluées déjà présentes, donc bénéfice d'utilisation d'outils qu'on n'aurait pas pris la peine de développer ;
- évolution continue par la communauté du logiciel libre, qui fait croître les avantages avec le temps.

Cependant, ceci n'est applicable que si un logiciel proche existe bel et bien. Ensuite, il y a un coût de prise en main du logiciel, lié aux questions de compatibilité de choix techniques : un simulateur très complet mais très lourd et qui impose de programmer dans un langage qu'on ne connaît pas bien aura un coût élevé.

Nous avons envisagé plusieurs hypothèses :

- ODE [SMI 04], un moteur en C++, qui permet de réaliser des simulations physiques et éventuellement obtenir une visualisation 3D de la scène simulée. ODE n'a pas été retenu car le coût de prise en main a été estimé bien trop élevé. Une simulation mécanique fidèle du type de ce qu'ODE réalise est assez complexe et non nécessaire dans notre cas. En outre, le logiciel semblait insuffisamment mature et exigeait beaucoup de connaissances spécifiques avant de pouvoir l'utiliser ;
- `xbraitenberg` [KOH 99] a des caractéristiques intéressantes : simple, léger, rapide, monde à deux dimensions avec animats et objets visibles. Il aurait pu être utilisé mais nous ne le connaissions pas à l'époque. Il aurait fallu vérifier si l'organisation interne du programme se prêtait bien à des extensions pour des capteurs du type désiré ;
- Nous souhaitions pouvoir faire fonctionner le même programme à la fois sur le robot réel et en simulation. Un choix possible aurait été d'utiliser le simulateur de `Saphira` [ACT 99], le système fourni par le fabricant des robots Pioneer. Cependant, ce logiciel a le défaut, important à nos yeux, de n'être prévu que pour simuler en temps réel, et non en temps accéléré. Devoir attendre aussi longtemps pour des simulations que pour un robot réel aurait annihilé une grande partie de l'intérêt de la simulation.

Nous avons donc choisi de réaliser un programme indépendant en C, qui pouvait utiliser les routines `Saphira` pour l'interfaçage avec le robot réel, tout en ayant un noyau indépendant qui pouvait faire tourner sa propre simulation simple. Le programme est portable facilement vers une autre plate-forme robotique.

Le reste de cette section décrit donc les caractéristiques de notre simulateur.

8.1.2 Monde

Le monde simulé est en deux dimensions. Deux types d'objets peuvent y être incorporés : des disques (repérés par leurs centre et rayon) et des rectangles dont les côtés peuvent être parallèles aux axes (repérés par les coordonnées de deux coins opposés). Chaque objet peut avoir une couleur (teinte, saturation, luminance) ou bien être considéré « non-coloré ».

8.1.3 Vision

La vision de l'animat fonctionne sur le principe du lancer de rayon à une dimension sans calcul d'ombrage, ce qui a pour avantages d'être assez simple à implémenter et de gérer les occlusions. La rétine de l'animat est modélisée comme une suite de facettes disposées en un polygone régulier, chacun sensible à la couleur de l'objet situé exactement en face de lui, c'est-à-dire à la première intersection entre la demi-droite de visée et les objets du monde, comme illustré figure 8.1.

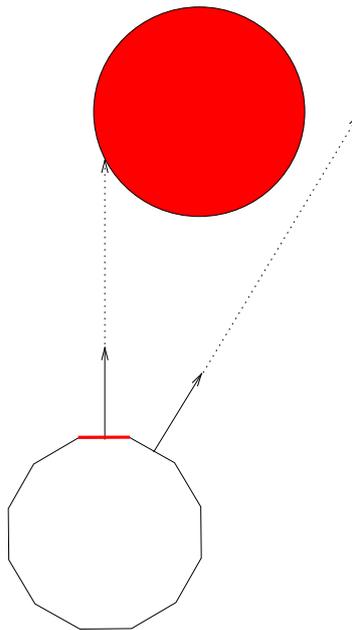


FIG. 8.1: Principe de fonctionnement des capteurs simulés. On représente ici un capteur à 12 facettes. Celui de l'animat a en réalité 360 facettes. Du centre de chaque facette, le simulateur fait partir un rayon infiniment fin (une demi-droite). La facette perçoit la couleur du premier objet qui intersecte le rayon. Le fait de ne pas tenir compte de tout l'angle que couvre une facette introduit des artefacts : avec cette implémentation simple, les objets trop petits ou trop lointains peuvent ne pas être détectés car ils « passent » entre deux rayons. Une vision de portée infinie aurait été mathématiquement plus parfaite mais nous aurait paru moins réaliste.

8.1.4 Mécanique du robot et positionnement

Le simulateur est fondé sur un modèle de robot à un essieu porté par deux roues parallèles motrices indépendantes, en supposant que la stabilité est assurée par un autre moyen non modélisé (typiquement une roue folle) et que le sol est plan. Ceci est illustré sur la figure 8.2. Par exemple, les robots Khepera, Pioneer 2 (modèle d'intérieur) et Pekee se conforment à ce modèle.

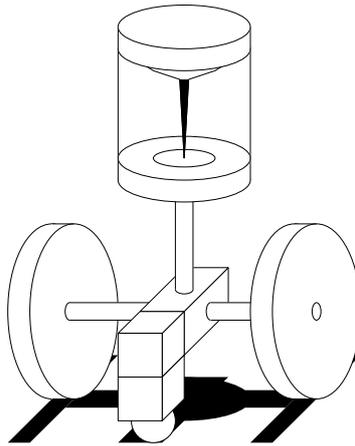


FIG. 8.2: Schéma symbolique de la mécanique du robot simulé. Le robot s'appuie sur un essieu dont la stabilité est assurée par une roue folle. Chaque roue dispose d'un moteur indépendant. Le capteur panoramique est représenté symboliquement. On suppose que son axe est vertical et qu'il passe par le centre de l'essieu.

Ce modèle permet de représenter informatiquement la position du robot par les coordonnées à deux dimensions de son centre et l'angle de rotation par rapport à un nord arbitraire. Il permet surtout de calculer simplement le déplacement (translation du centre, rotation) du robot au cours d'une rotation élémentaire de chacune des roues. On programme le calcul du déplacement dans le cas où les deux roues tournent à vitesse angulaire constante. Une intégration de type méthode d'Euler avec des pas de temps suffisamment petits permettrait de calculer le cas général. Dans nos expériences, toutefois, les ordres moteurs sont des successions finies d'intervalles de temps où la vitesse des roues est constante, ce qui économise une étape d'intégration. Le but n'est pas d'avoir une modélisation fine de problèmes mécaniques mais plutôt un modèle idéalisé simple et clairement défini.

On choisit la même convention de coordonnées que Saphira : repère orthonormé direct, avec le nord le long de l'axe des x . L'unité de base est le millimètre.

On néglige l'épaisseur des roues, modélisées comme des disques. Les paramètres mécaniques dans nos expériences sont les suivants :

longueur de l'essieu 300 mm

rayon nominal des roues 100 mm

Ces paramètres ne sont pas extrêmement importants en eux-même, mais nécessaires pour modéliser proprement la façon dont nous basons l'odométrie dans nos simulations.

8.1.5 Odométrie biaisée

Afin de vérifier la capacité de l'animat à s'orienter et à rejoindre un but quand la relation entre les ordres moteurs et leurs conséquence est perturbée, nous avons besoin de simuler une perturbation.

Certaines perturbations informatiquement simples, mais artificielles, étaient possibles, comme par exemple ajouter une erreur à la position et à l'orientation à chaque déplacement.

Une cause courante des problèmes d'odométrie, qui concerne notamment les robots Pioneer 2, est la dissymétrie des roues qui cause des erreurs d'orientation systématiques dès que le trajet total du robot dépasse 10 à 20 mètres¹.

Nous avons choisi de perturber le déplacement de l'animat par une modélisation très simple de ce phénomène : la simulation du mouvement est faite en utilisant un diamètre différent pour les roues du robot, sans modifier les ordres moteurs. Ainsi, quand un ordre moteur commande d'aller tout droit, les deux roues tournent à la même vitesse angulaire, mais la trajectoire effective est un arc de cercle. Nous avons choisi les paramètres suivants :

rayons de la roue gauche 99mm

rayons de la roue droite 101mm

Les modèles de navigation que nous étudions étant sensibles aux erreurs d'orientation, mais n'utilisant pas d'intégration des données odométriques au cours du temps, nous avons estimé que cette modélisation était suffisante. En effet, elle perturbe l'orientation d'une façon simple et déterministe et, comme nous le verrons, suffisante pour faire apparaître les mêmes problèmes que dans le cas réel. En revanche, elle ne contient pas de composante aléatoire non systématique. Cela l'empêche de simuler une perturbation réaliste lorsque, par exemple, l'animat réalise de nombreux petits mouvements au cours desquels les roues tournent alternativement dans un sens puis dans l'autre. Ni notre modèle ni le comportement du robot ne s'appuyant sur de tels comportement, nous estimons que cette modélisation suffit.

Les effets sur la trajectoire du robot sont illustrés en figure 8.3.

8.2 Navigation locale simulée avec visibilité partielle

Après avoir décrit l'environnement simulé, nous abordons les expériences qu'il nous a permis de mener. La première d'entre elles a pour but de valider l'utilisation de panoramas en couleurs et leur exploitation effective par la méthode d'appariement par programmation dynamique dans le cas de panoramas ne faisant pas le tour complet. L'orientation est supposée exactement connue sans faire intervenir le compas visuel [GOU 03].

¹Ce syndrome est parfois désignée par l'appellation « banana problem ». Considérons un robot qu'on envoie explorer un couloir rectiligne et qui suit le couloir en utilisant un simple évitement d'obstacle (maintien d'une distance au mur à l'aide de sonars ou d'infrarouge). Sa trajectoire réelle à grande échelle va être rectiligne. Mais si on trace la trajectoire obtenue par intégration des données odométriques, on aboutit à une courbe en arc de cercle rappelant la forme d'une banane.

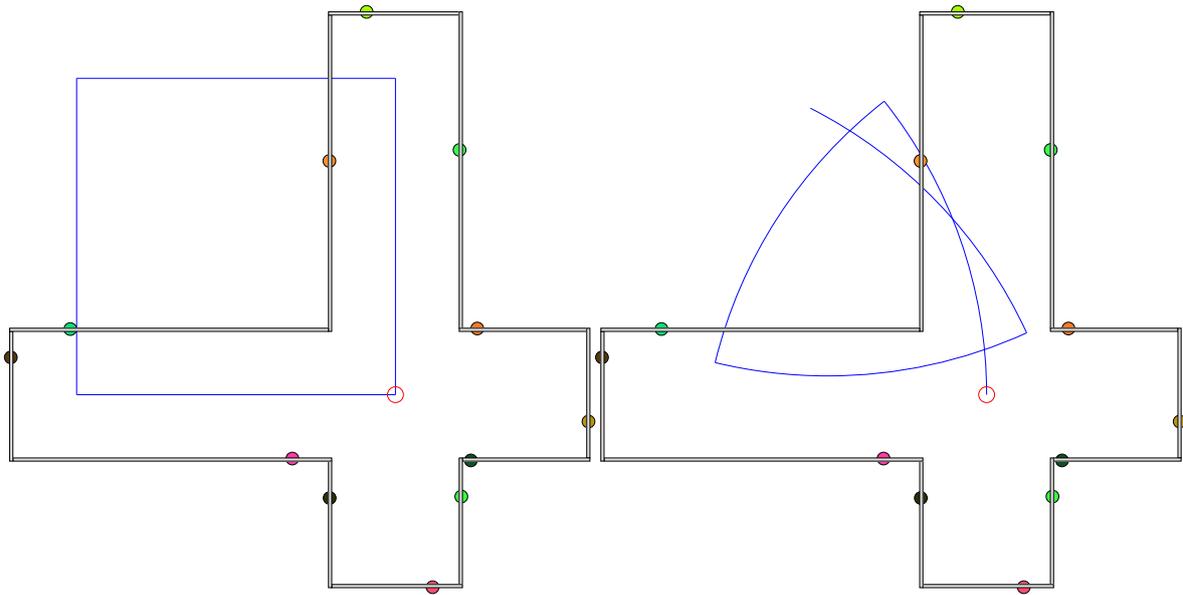


FIG. 8.3: Exemple de trajectoire avec les roues symétriques et dissymétriques. Le robot simulé, guidé par l'odométrie, essaye de tracer un carré de côté 10 mètres. À gauche, roues nominales, à droite roues dissymétriques. Les paramètres sont les mêmes que dans les expériences simulées de navigation avec mauvaise odométrie. Pour rendre la figure plus intuitive, nous avons superposé l'image d'un environnement avec des couloirs de 4m de large et d'une longueur totale de 18m.

On compare la performance pour une tâche de navigation locale dans trois situations. Le même environnement est utilisé dans les trois cas. La figure 8.4 illustre l'expérience. Les amers étiquetés a et c sont d'une même couleur, tandis que les deux amers étiquetés b près du centre du terrain sont d'une autre couleur.

Le robot simulé dispose d'un champ de vision de 200° . Il se déplace suivant le mode directionnel (cf. 4.7), faisant toujours face à la partie droite du terrain. Ce choix d'une direction unique permet de définir un unique champ de vecteurs et de maximiser les amers communs entre panoramas.

La tâche de navigation locale consiste à revenir en un point d'où 5 amers sont visibles, avec orientation connue et la méthode PV (donc composante tangentielle seulement).

Dans la première situation, l'animat utilise l'appariement glouton des méthode originales CC. Dans la seconde, la programmation dynamique est introduite mais sans information de couleur (amers indistinguables). Dans la troisième, l'information de couleur est donnée et permet à la programmation dynamique de fournir un appariement plus pertinent.

Les champs de vecteurs des trois parties de la figure représentent les vecteurs déplacement calculés en chaque point d'une grille régulière. La zone ombrée représente le bassin d'attraction.

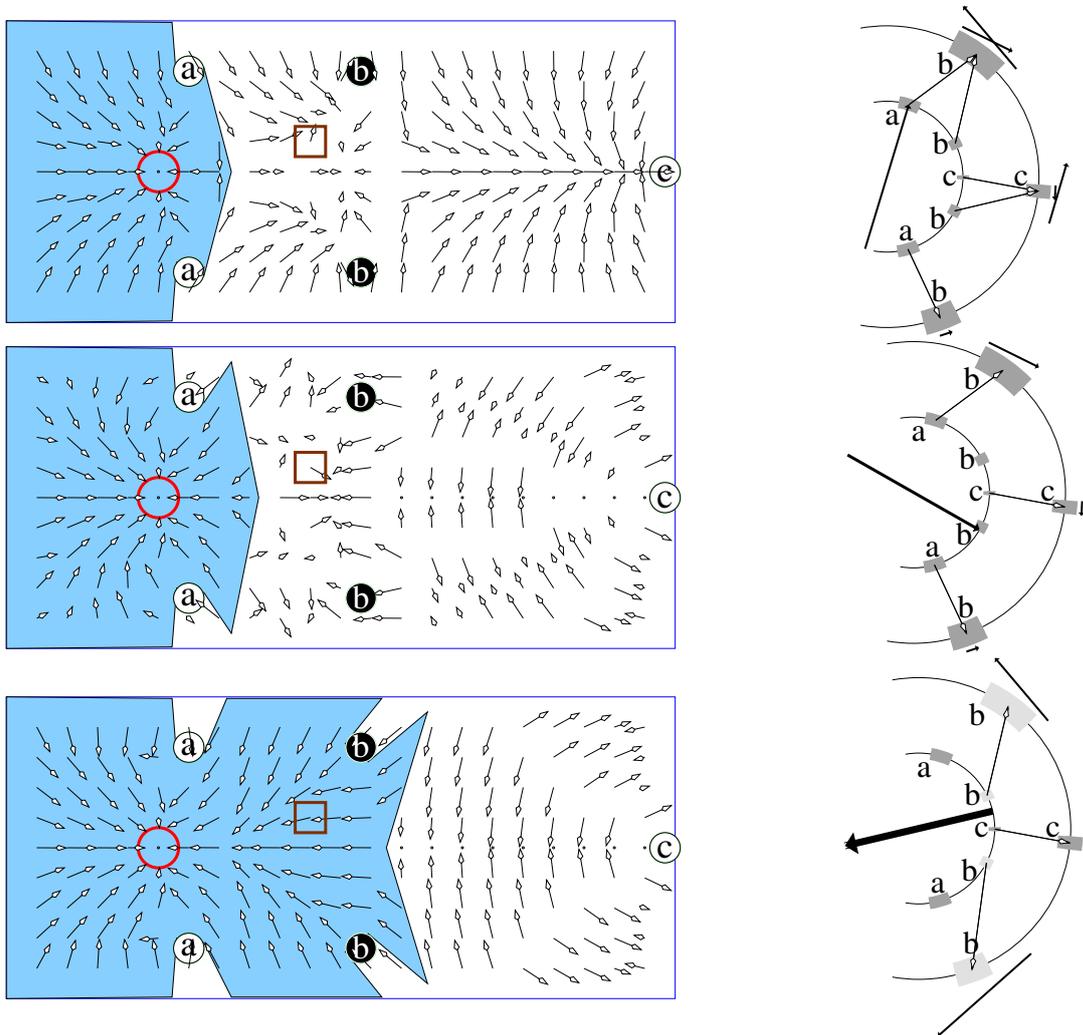


FIG. 8.4: Comparaison expérimentale des méthodes d'appariements en simulation face à la disparition d'amers due au champ de vision limité du robot. On utilise le modèle PV pour calculer la direction à suivre. 1^{er} cas (en haut) : appariement original utilisé par les modèles CC. 2^{ème} cas (au milieu) : appariement utilisant la programmation dynamique, sans information de couleur. 3^{ème} cas (en bas) : appariement utilisant la programmation dynamique, avec information de couleur disponible. À gauche : les champs de vecteurs montrent le déplacement indiqué par chaque algorithme pour rejoindre le but indiqué par le cercle rouge. Pour la clarté du schéma, nous avons limité la longueur des flèches. Les surfaces ombrées représentent approximativement les bassins d'attraction. À droite : représentation de l'appariement et du calcul effectué aux positions marquées par des carrés à gauche. Les vecteurs individuels (à l'extérieur du grand cercle) contribuent au vecteur mouvement résultant (au centre du cercle, représenté d'autant plus épais que le déplacement calculé est important).

8.2.1 Expérience témoin : modèle original

Les trois situations produisent des résultats similaires à gauche du terrain, où tous les amers sont dans le champ de vision du robot. Des différences sont visibles en revanche à droite, où deux, voire quatre amers, peuvent sortir du champ de vision. Avec l'algorithme d'appariement original (cas 1), un point attracteur indésirable existe au centre du terrain. En effet, l'algorithme d'appariement glouton original peut appairer plusieurs secteurs de la vue mémorisée avec le même secteur de la vue actuelle. Ceci a tendance à approcher l'animat de l'amer c ; nous y reviendrons au chapitre sur les expériences robotiques (9.2). Ceci ne se produit plus dans les cas 2 et 3 car la programmation dynamique ne permet pas à un secteur de s'appairer avec plus d'un autre secteur. Comme nous n'utilisons dans cette expérience que la composante tangentielle, le robot peut s'aligner avec l'amer c quand il est le seul amer vu, mais ne peut pas estimer s'il doit s'approcher ou s'éloigner de l'amer.

8.2.2 Expérience témoin : modèle original avec programmation dynamique

Dans le cas 2, la programmation dynamique sans information de couleur ne supprime pas l'attracteur central, mais agrandit légèrement le bassin d'attraction.

8.2.3 Programmation dynamique et couleur

Dans le cas 3 on utilise la programmation dynamique, qui dispose de l'information de couleur. Le bassin d'attraction est largement agrandi.

Ceci illustre l'intérêt de distinguer les amers les uns des autres. Dans les cas 1 et 2, les bassins d'attraction étaient limités par l'attracteur indésirable au milieu du terrain, où les deux amers b étaient incorrectement appariés avec les deux amers a du panorama mémorisé. Dans le cas 3, l'information de couleur permet un meilleur appariement. Les amers b du panorama mémorisé ne sont plus appariés incorrectement. S'ils sont toujours visibles dans le nouveau panorama, ils sont correctement appariés. Sinon ils sont ignorés. L'appariement plus correct a pour conséquence un plus grand bassin d'attraction.

Remarquons que l'algorithme d'appariement d'origine (cas 1) ne peut généralement pas produire de bon résultat si le champ de vision ne couvre pas les 360° . Les différentes méthodes expérimentées dans [WEB 99] ne peuvent pas non plus faire aussi bien que la programmation dynamique avec information de couleur (cas 3).

La capacité à distinguer des amers en utilisant une information supplémentaire comme la couleur est nécessaire pour éviter d'appairer incorrectement des amers qui n'apparaissent que dans une des deux vues, ce qui est crucial quand le champ de vision ne couvre pas 360° .

8.3 Navigation avec orientation

8.3.1 But visible depuis le point de départ.

Cette expérience valide ponctuellement l'utilisation du compas dans une tâche de navigation locale.

Dans l'environnement montré en figure 8.5, on fait passer l'animat par une série de 7 points p_i , selon une trajectoire en zigzag, d'où on lui fait prendre des panoramas.

On a choisi une disposition en zigzag pour fournir au compas des vues non alignées. Les 5 triplets de vues successives qu'on peut former à partir de cette liste sont utilisés pour le même nombre d'étapes d'observation pour le compas visuel. Dans chacun des cas on utilise la vue centrale du triplet dans le rôle de la vue M : $N = p_{i-1}$, $M = p_i$, $P = p_{i+1}$. Un tel triplet est déclaré utilisable pour que l'animat estime son orientation quand il rejoint le point p_i .

Alors, partant du point le plus au nord sur la figure, l'animat a pour tâche de rejoindre le point au centre de la jonction des couloirs, dans 4 situations successives : d'une part, avec ou sans compas et, d'autre part, avec des roues symétriques ne causant pas d'écart angulaire, ou avec des roues dissymétriques causant une erreur d'orientation d'autant plus grande que l'animat avance d'une plus grande distance.

La figure 8.5 permet de constater visuellement les points suivants :

- l'animat parvient à suivre la séquence sans compas s'il n'y a pas d'erreur d'orientation (haut gauche) ;
- l'animat échoue à suivre la séquence sans compas s'il y a une erreur d'orientation (haut droite). Seul le premier segment est nettement couvert ;
- l'animat parvient à suivre la séquence en utilisant le compas pour compenser l'erreur d'orientation (bas droite) ;
- l'animat parvient aussi à suivre la séquence en utilisant le compas même s'il n'y avait pas d'erreur d'orientation à compenser (bas gauche).

On conclut de cet exemple que le compas visuel peut effectivement corriger, dans ce cas du moins, l'erreur d'orientation introduite par une dissymétrie des roues et que l'erreur que le compas lui-même introduit n'empêche pas une navigation satisfaisante.

Ces conditions sont toutefois relativement contrôlées et on souhaite tester le compas dans des situations un peu plus difficiles.

8.4 Conclusion des expériences simulées

Les expériences simulées nous ont permis de tester nos algorithmes et d'en tirer des conclusions relativement générales.

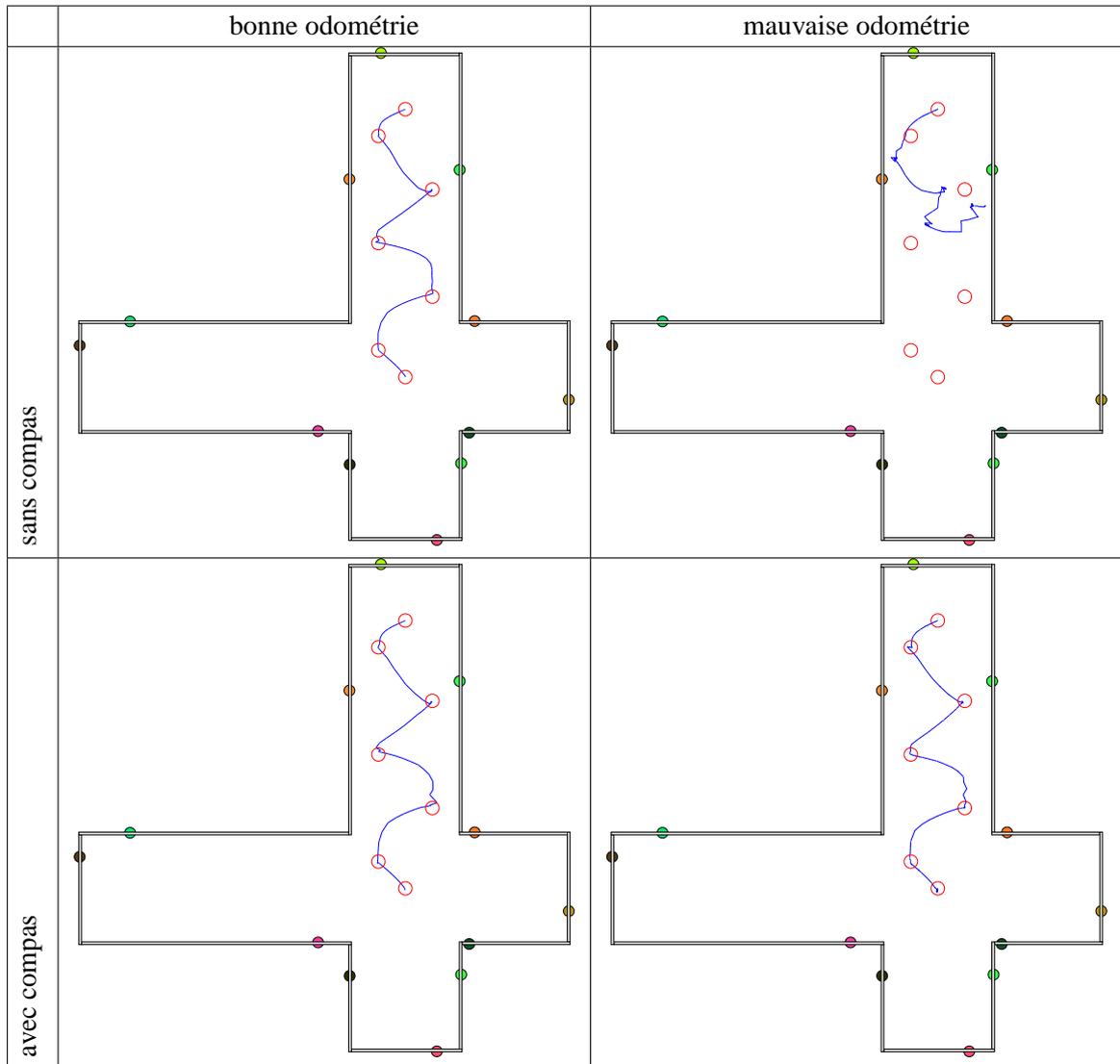


FIG. 8.5: Navigation simulée de proche en proche en utilisant ou non le compas pour compenser une mauvaise odométrie éventuelle. Le robot est d'abord mené du centre de la croix au nord, pour prendre des panorama (cercles). Puis il doit revenir par le même chemin mais en sens inverse, du point le plus au nord vers le centre. En haut, l'animat n'utilise pas le compas visuel, en bas il l'utilise. À gauche, l'odométrie est bonne, à droite elle est mauvaise. Le compas visuel permet de rattraper les effets d'une mauvaise odométrie, sans introduire de perturbation dans le cas où son utilisation n'est pas nécessaire.

Nous avons présenté une sélection des expériences de simulations significatives pour la validation de nos algorithmes : utilisation de panoramas en couleur, appariement par programmation dynamique, navigation avec utilisation du compas visuel.

En conclusion des expériences présentées dans ce chapitre, il apparaît que :

- L'utilisation de panoramas en couleurs, lorsqu'elle est exploitée par notre algorithme de programmation dynamique, permet un meilleur appariement en éliminant des ambiguïtés qui se produisent quand les amers se sont beaucoup déplacés entre deux vues. Cela permet aussi de tenir compte de ce que certains amers peuvent n'être visibles que dans l'une des deux vues, en particulier dans le cas d'un champ de vision qui ne couvre pas tout le cercle ;
- L'utilisation du compas visuel a été validée dans un cas simple où les panoramas utilisés pour l'étape d'observation sont donnés, pour une tâche de navigation sur une succession de points.

Après ces expériences en simulation, nous allons présenter des expériences qui valident nos travaux en environnement réel.

Chapitre 9

Expériences robotiques

La validation d'un modèle sur robot réel est une étape importante. Elle représente un test sérieux car elle nécessite une chaîne complète et fonctionnelle, de la perception à l'action, et met la robustesse des méthodes à l'épreuve des erreurs de mesures. Elle est aussi la source de nombreuses difficultés pratiques qu'il faut surmonter.

Dans ce chapitre, après une présentation de l'environnement robotique dont nous disposons (section 9.1), nous détaillerons les expériences que nous avons réalisées.

La première expérience est l'équivalent réel de la première expérience simulée que nous avons rapportée en section 8.2. Son but est de valider à la fois l'utilisation de panoramas en couleurs et la méthode d'appariement par programmation dynamique, sur robot réel cette fois.

La seconde expérience est une comparaison entre notre méthode d'orientation par le compas visuel et la méthode de Benosman et al, que nous avons résumée en section 4.8.

Ensuite, nous expliquons les modifications que nous avons apportées au robot et aux paramètres expérimentaux pour rendre possible l'application du compas visuel au robot. En effet, il a fallu réaliser un capteur panoramique adapté à nos contraintes, et modifier les paramètres de la segmentation et de l'appariement pour les rendre suffisamment robustes et plus génériques que dans les premières expériences.

Dans la troisième expérience, le compas visuel est appliqué au robot. Il s'agit d'un test analytique du compas, dont la performance est comparée à celle de la méthode de l'étoile polaire introduite avant le compas visuel (section 6.2).

Enfin, la dernière expérience fait intervenir le compas visuel dans une tâche réelle de navigation, de durée relativement étendue. Elle fait l'objet d'une comparaison entre trois méthodes de maintien de l'orientation face à une perturbation cumulative : la méthode témoin, la méthode de l'étoile polaire, et compas visuel.

Enfin, nous terminerons cette partie en expliquant les difficultés que nous avons eues en tentant de faire fonctionner nos modèles dans les couloirs du laboratoire et en explicitant les recommandations

que nous pouvons tirer de notre expérience.

9.1 Matériel et paramètres

9.1.1 Robots Pioneer 2



FIG. 9.1: Un robot Pioneer 2 connecté à une station de recharge. Le contact est assuré par deux larges lames de cuivre, au-dessus et en dessous d'un becquet. Le robot peut se connecter sans intervention manuelle à condition qu'il vise suffisamment bien.

Nous avons à notre disposition quelques robots de modèle Pioneer 2 fabriqués par ActivMedia Robotics.

Mécaniquement, ce robot s'appuie essentiellement sur un essieu à deux roues, plus une « roue folle » (dont la direction n'est pas fixée) pour faire un troisième point de contact avec le sol nécessaire pour la stabilité et l'horizontalité.

Ses senseurs comprennent une caméra motorisée en couleurs (modèle EVI-D31 de Sony), des émetteurs/récepteurs d'ultrasons qui permettent des mesures de distances aux obstacles, des encodeurs dans les roues qui permettent de mesurer l'angle de rotation (avec une alerte détectant le blocage de

roues) et un capteur de tension électrique aux bornes des batteries.

Le robot fonctionne sur une à trois batteries au plomb qu'il ne faut pas trop décharger sous peine de les détériorer. Il ne peut malheureusement pas s'éteindre spontanément, ce qui serait utile lorsqu'il détecte que ses batteries sont trop faibles.

Au niveau informatique, il contient un PC (PC AT sur carte PC 104) équipé d'un processeur Pentium MMX cadencé à 233 MHz, 32Mo de RAM, disque dur 4Go, avec sortie VGA, prises clavier, souris et RS232, carte d'acquisition vidéo couleurs (pour la caméra), liaison Ethernet (réseau) sans fil à la norme 802.11 réalisant des débits jusqu'à 200ko/s. Cette liaison permet de suivre et commander le robot sans devoir s'agenouiller à côté pour brancher un écran et un clavier. Grâce à l'utilisation de Gnu/Linux comme système d'exploitation et à son exemplaire intégration avec le fonctionnement en réseau, toutes les opérations peuvent être menées à distance de façon transparente via la liaison réseau sans fil, sans brancher d'écran ni de clavier sur le robot¹.

Malgré ses quelques limitations et parfois des soucis de maintenance, le robot Pioneer est dans l'ensemble une assez bonne plate-forme, notamment grâce à l'utilisation de composants matériels relativement standards, ce qui offre une certaine souplesse. Par exemple, il nous été relativement facile de connecter la puce de perception visuelle GVPP à la caméra du robot en entrée, à un émetteur vidéo radio en sortie, et d'alimenter notre capteur panoramique, parce que des connecteurs standard pour vidéo et alimentation étaient accessibles dans le robot et parce que la carte mère du robot fournissait jusqu'à 4 liaisons RS232-C par le simple ajout d'une nappe et de connecteurs.

Dans toutes nos expériences avec le robot Pioneer 2, la perception et les calculs ont été réalisés de façon embarquée, la liaison vidéo radio permettant de contrôler visuellement le bon fonctionnement de GVPP et la liaison Ethernet radio servant à observer le bon fonctionnement des étapes de navigation, sous formes de diagrammes d'appariements et de directions à suivre.

9.1.2 Caméra couleur

La caméra disponible sur les Pioneer 2 est le modèle EVI-D31 de Sony, capable de tourner mécaniquement de -100° à $+100^\circ$ horizontalement, et de -25° à $+25^\circ$ verticalement, et équipée d'un zoom optique x12 ($f=5.4$ à 64.8mm). Lorsque la caméra est montée sur le robot, l'axe optique est à 31 cm au-dessus du sol. Toutes les fonctions de la caméra peuvent être commandées par une liaison série (norme RS-232, protocole VISCA).

Cette caméra nous a paru le capteur tout désigné pour faire percevoir son environnement au robot. Nous parlerons de panoramas dans les sections suivantes bien que, avec cette caméra, ils aient été limités à 200° .

¹Même si on considère cette fonctionnalité comme normale et acquise, il est bon d'en expliciter les bénéfices pour inviter les lecteurs à utiliser les mêmes moyens.

9.1.3 GVPP

La disponibilité d'une caméra motorisée en couleurs n'était pas forcément suffisante pour faire de la vision robotique dans des conditions satisfaisantes. De fait, les performances de la carte d'acquisition intégrée étaient insuffisantes puisqu'il fallait jusqu'à 200ms, soit le temps de 5 images vidéo, pour faire l'acquisition d'une seule image, sans compter qu'ensuite il fallait parcourir cette dernière pour faire les traitements requis.

Pour remédier à ces difficultés, nous avons eu l'opportunité d'utiliser des puces GVPP [PIR 04]. GVPP est un acronyme pour Generic Visual Perception Processor. C'est une puce qui réalise l'analyse d'images vidéo en temps réel² malgré un volume et une consommation étonnamment faibles. Les avantages d'utiliser cette puce sont doubles : tout d'abord, c'est un appareil temps réel qui garantit de ne perdre aucune image vidéo et, en outre, le processeur de l'ordinateur reste libre pour les autres calculs nécessaires (partie navigation), GVPP fournissant les informations au PC par l'intermédiaire d'une liaison série (norme RS-232).

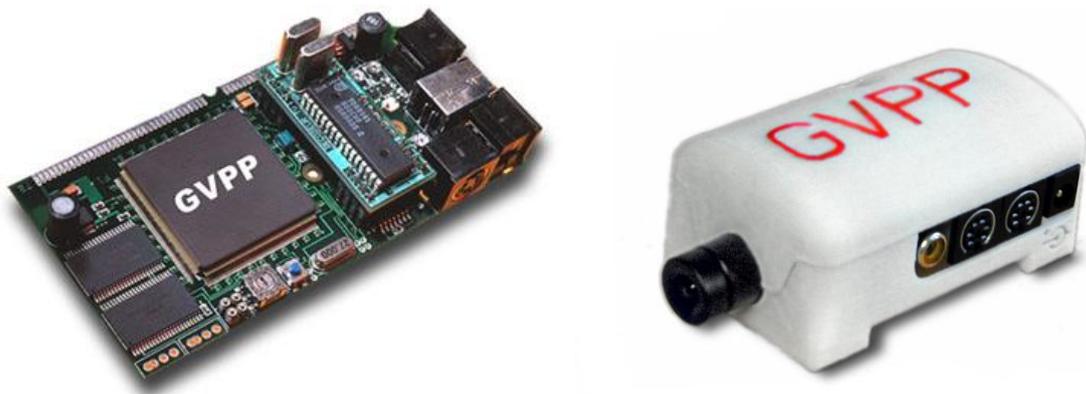


FIG. 9.2: Deux versions de la carte de perception visuelle GVPP. À droite, la version la plus récente a pour dimensions 80 * 48 * 38 mm.

La plupart des expériences apparentées [RöF 97b, FRA 98a, MöL 98a, WEB 99] n'ont pas accès aux objets n'ayant pas d'intersection avec le plan horizontal. La cause est parfois située en amont et dépend de la construction du capteur utilisé ; elle peut être située en aval parce que l'expérimentateur fait le choix de minimiser la surface d'image à traiter face au coût élevé que cela implique. Dans la mesure où GVPP peut traiter à cadence vidéo toute l'image, il nous a été possible dans nos expériences avec une caméra non panoramique de détecter et d'exploiter des objets jusqu'à 20° au-dessus et en-dessous du plan horizontal de la caméra. (En revanche, le capteur panoramique que nous avons utilisé pour valider le compas visuel ne permettait pas ce bénéfice.) Nous pensons que cet avantage est non négligeable lorsqu'on souhaite préparer le moins possible l'environnement du robot.

²Plus d'information sur cette puce conçue par Patrick Pirim est disponible sur <http://bev.lu/>.

9.1.4 Obtention d'un panorama via la caméra mécanique

Pour percevoir un panorama, la méthode que nous avons choisie consiste à commander à la caméra de tourner horizontalement à vitesse angulaire constante, d'une butée à l'autre, pendant que GVPP analyse le flux image par image.

Cette façon de procéder permet potentiellement une analyse très fine du panorama, car les objets visés sont présentés séquentiellement dans le temps et peuvent occuper chacun une grande portion de l'écran. Par contre, elle présente l'inconvénient d'être dépendante d'un balayage complet de la caméra, opération mécanique et donc relativement lente. Il est aussi nécessaire de connaître à chaque instant l'orientation actuelle de la caméra.

GVPP est programmée pour travailler, dans chaque image vidéo (cf figure 9.3), sur les pixels :

- dont la saturation est supérieure à un seuil (i.e. apparaissant suffisamment colorés) ;
- situés dans une fenêtre verticale dans l'image.

La largeur de cette fenêtre est calculée pour que la portion d'espace balayée pendant une image vidéo recouvre légèrement celle balayée à l'image vidéo suivante. Sur cet ensemble de pixels, la puce extrait un intervalle des valeurs les plus représentées des paramètres de teinte, saturation et luminance. Elle envoie cette information et le nombre de pixels concernés au robot.

Initialement, la caméra était connectée au microcontrôleur du robot, mais le logiciel de ce dernier ne permettait pas d'interroger la caméra sur son orientation actuelle. Nous avons donc connecté la caméra directement à la carte-mère du PC en la pilotant avec un logiciel trouvé sur le web, écrit par Thomas Moeslund [MOE 98]. Ce logiciel permet d'interroger la caméra sur son orientation, mais ne rend la main que lorsque la caméra a fini son mouvement et n'est pas prévu pour la programmation parallèle (multithread). Nous l'avons donc modifié pour pouvoir interroger la caméra mécanique au cours de son mouvement, afin de corréliser informations visuelles et angulaires. Afin de contribuer en retour, nous avons contacté l'auteur initial et publié le code modifié sous licence GNU GPL³.

Pendant l'acquisition du panorama, l'ordinateur exécute répétitivement la séquence suivante :

- attendre de recevoir une indication de couleur de GVPP (parce que cette indication est périodique) ;
- interroger la caméra sur son orientation actuelle ;
- noter en mémoire le couple angle-couleur observé.

À la fin du balayage, on opère une segmentation sur les données reçues. Dans les expériences sur le Pioneer 2, nous avons opté pour une segmentation très simple avec un seuil de teinte : on parcourt les données enregistrées de gauche à droite en maintenant une estimation de la moyenne des teintes, saturations, luminances rencontrées jusque là. Si le nombre de points saturés passe au-dessus ou en-dessous du seuil de 500, on considère qu'il y a une transition entre objet coloré visible et absence d'objet. S'il y a un objet et que la teinte change au-delà d'un seuil fixé à 12 niveaux (sur 256), on considère qu'il y a une transition entre deux objets de couleurs différentes.

À l'issue de ce traitement, on obtient un panorama segmenté. Un exemple de tel panorama et

³Le pilote modifié est disponible sur <http://animatlab.lip6.fr/Gourichon/SonyPTZCamEVID31/>.

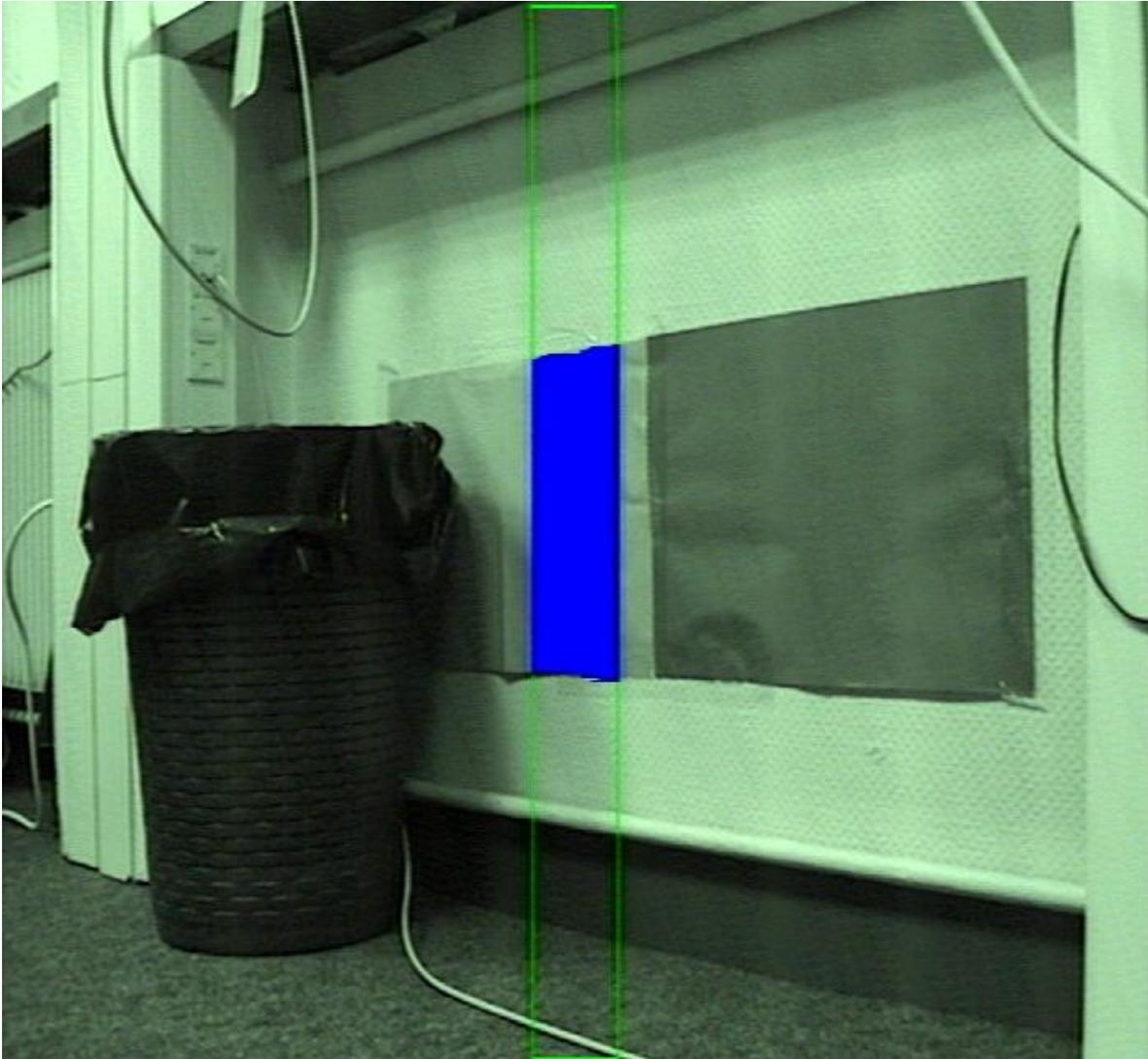


FIG. 9.3: Exemple d'image perçue pendant une prise de panorama par la caméra mécanique. La caméra balaye l'espace comme un phare. Dans chaque image, à cadence vidéo, GVPP analyse une fenêtre verticale (rectangle au centre occupant toute la hauteur). Elle sélectionne les pixels suffisamment colorés (haute saturation), tels que ceux qui sont associés sur cette image à un papier coloré sur un mur blanc. Les objets sombres (poubelle) ou non colorés (murs blancs) ne sont pas vus. Dans l'ensemble sélectionné, le nombre de points, ainsi que des intervalle des luminances, teintes et saturations les plus représentées, sont calculés. Ces informations et elles seules sont envoyées au robot avant l'arrivée de l'image suivante.

d'appariement sera donné en figure 9.4.

9.1.5 Paramètres de la méthode d'appariement

Nous détaillons ici la fonction de ressemblance (c.f. section 5.2.4.1) utilisée dans les expériences qui vont suivre.

Nous apparions les secteurs (et non les transitions entre secteurs). On note respectivement h_i, s_i, l_i et a_i la teinte moyenne, la saturation moyenne, la luminance moyenne, et l'azimut du centre du secteur i du premier panorama, et respectivement h_j, s_j, l_j et a_j pour le secteur j du second panorama. La case i, j du tableau \mathbf{M} est le score de l'appariement élémentaire, la ressemblance entre les deux amers.

Nous avons choisi comme fonction de ressemblance :

$$\mathbf{M}_{i,j} = \exp \left(- \left(\frac{h_i - h_j}{\sigma_h} \right)^2 - \left(\frac{s_i - s_j}{\sigma_s} \right)^2 - \left(\frac{l_i - l_j}{\sigma_l} \right)^2 - \left(\frac{a_i - a_j}{\sigma_a} \right)^2 \right) \quad (9.1)$$

où les σ représentent des tolérances. Une telle fonction répond 1 lorsqu'on compare deux amers en tous points identiques, et s'approche de 0 pour deux amers très dissimilaires. De façon générale il est bon de choisir des tolérances en proportion des écarts-types des variables mesurées si l'on veut tenir compte équitablement des différentes variables. Nous les avons fixés à $\sigma_h = 5$, $\sigma_s = 15$, $\sigma_l = 15$ et $\sigma_a = 90$ (les trois premières variables étant des mesures comprises entre 0 et 255, et la dernière représente un écart d'angle en degrés de -180 à 180).

9.2 Navigation locale, orientation par odométrie

Dans cette expérience, notre animat, un robot Pioneer 2, n'est pas encore équipé du compas visuel ni du capteur panoramique, mais de la caméra mécanique. Parce que son champ de vision limité à 200° signifie qu'une occlusion de 160° se produit à l'arrière du robot, nous avons fait le choix de conserver une direction de la caméra constante pour maximiser le nombre d'amers communs entre la vue du but et une vue actuelle. Autrement dit, le robot se comporte comme un agent directionnel d'après la typologie de Ruchti (cf. 4.7).

Le but de cette expérience est de valider la chaîne que nous avons mise en place :

- utilisation de GVPP et de la caméra mécanique pour l'acquisition de panoramas en couleurs ;
- appariement par programmation dynamique ;
- utilisation de l'odométrie restreinte à une indication de la direction du robot ;
- modèle PV pour la direction à suivre (méthode moyenne) ;
- exécution par le robot des ordres moteurs ;
- validation de l'arrivée au but par connexion physique à une station de recharge.

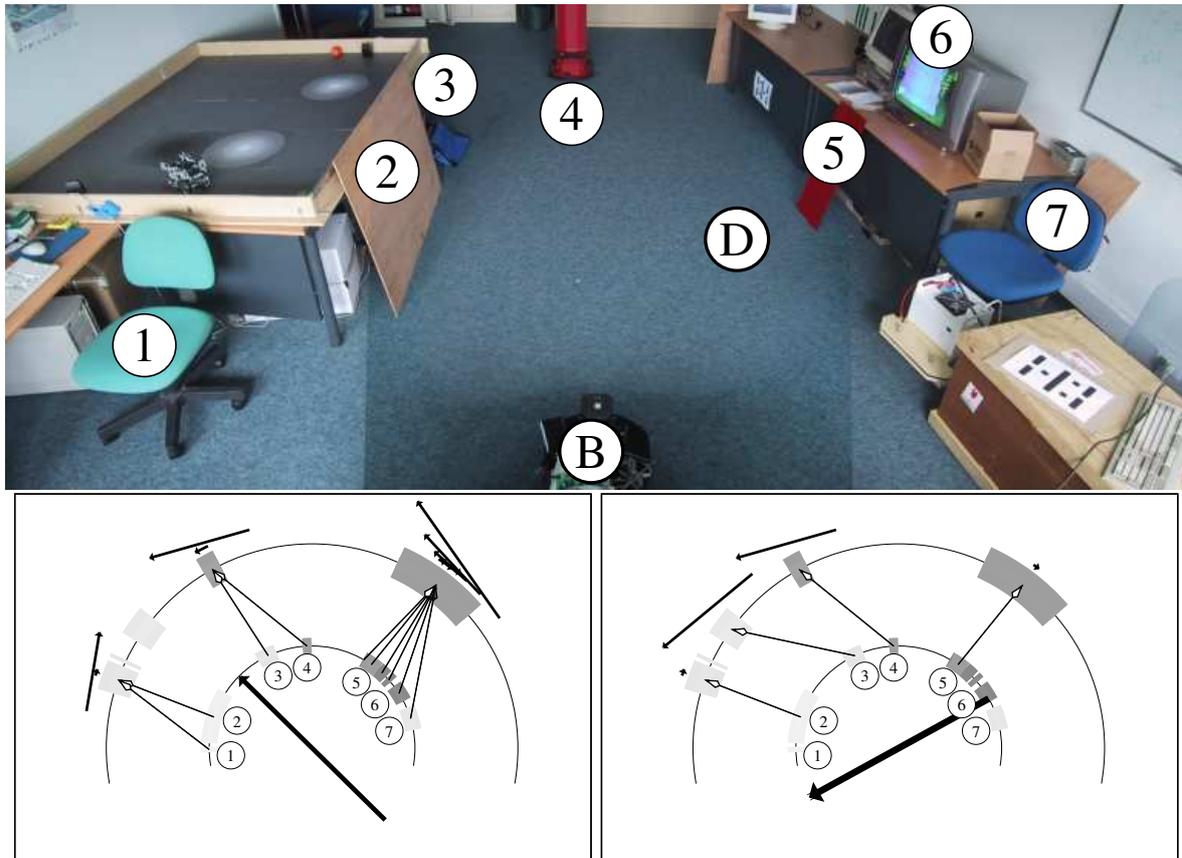


FIG. 9.4: Expérience de navigation locale sans compas visuel en environnement réel dans la salle robotique de notre laboratoire (en haut). Les points numérotés indiquent les objets utilisables par le système visuel. Les points B et D indiquent respectivement le but et un des points de départ. En bas, comparaison de deux procédures d'appariement. De chaque côté, l'arc intérieur représente le panorama vu du but, l'arc extérieur le panorama actuel, acquis par la caméra mécanique. Les arcs épais représentent les objets colorés, les niveaux de gris correspondant à la teinte, paramètre principal pouvant les discriminer. Les flèches d'un arc à l'autre représentent les appariements. Dans chaque cas, les flèches extérieures aux arcs sont les vecteurs tangentiels contribuant au mouvement calculé, la flèche centrale représentant la résultante de l'ensemble des vecteurs, direction que prendra le robot au prochain pas. À gauche : l'algorithme d'appariement glouton des modèles CC ne tient pas compte des différences de couleurs et peut appairer plusieurs amers mémorisés avec le même amer vu. Le champ de vision limité du robot a pour conséquence que certains amers ne sont plus visibles du point D, ce qui amplifie ce défaut. Le nombre important de mauvais appariements aboutit à une direction qui éloigne le robot du but, vers un point attracteur indésirable. À droite : notre algorithme de programmation dynamique avec couleurs produit une direction convenable qui mène le robot vers le but. Seuls sont appariés la planche de bois (2), le sac bleu (3), l'autre robot (4), la plaque rouge (5), et le vecteur résultant est convenablement dirigé vers le but. Grâce à l'information de couleur, le modèle s'abstient avec raison d'appairer la chaise verte à gauche (1), le poste de télévision (6) et la chaise bleue à droite (6) qui sont visibles depuis le but mais pas de la vue actuelle.

9.2.1 Déroulement de l'expérience

Une station de recharge est placée dans la salle robotique de notre laboratoire. Le robot connecté à la station prend une vue, puis est déplacé vers un point voisin de quelques mètres et a pour tâche de revenir à la station. Un essai est déclaré réussi lorsque le robot parvient à se connecter physiquement à la station, le critère étant que la recharge des batteries soit réellement possible. La station de recharge est toujours derrière le robot, de sorte qu'il ne la voit jamais et est contraint de s'appuyer sur d'autres éléments visuels de l'environnement. Une station de recharge est visible sur la photo en figure 9.1. La précision nécessaire pour se connecter est d'une trentaine de centimètres.

Cette expérience est l'équivalent de la première expérience simulée que nous avons rapportée en section 8.2 où l'on a fait varier la perception et l'algorithme d'appariement utilisé afin d'évaluer les bénéfices de la perception de la couleur et son utilisation par l'appariement par programmation dynamique. Nous avons comparé les performances dans deux des trois conditions qui étaient réalisées en simulation : la première, où l'appariement glouton des méthodes CC est utilisé, et la troisième, où la programmation dynamique utilise l'information de couleur pour améliorer l'appariement et donc au final la direction suivie par le robot.

9.2.2 Résultats

Une photographie de la salle d'expérience est en figure 9.4, avec un exemple d'appariement et de direction calculée pour les deux modèles comparés. En surveillant cette expérience, nous avons constaté qu'il y a deux cas principaux pouvant causer la disparition d'un objet : les objets vus du but peuvent devenir invisibles ailleurs – parce qu'ils passent derrière le robot – et d'autres objets peuvent devenir invisibles – quand le robot s'approche trop d'eux, parce qu'ils sont situés trop haut ou deviennent cachés, par exemple sur un bureau (rappelons que l'axe optique de la caméra est à 31cm du sol).

La figure 9.5 montre une vue aérienne et compare les surface de captation des deux modèles. La surface de captation est visiblement plus grande pour le second modèle, avec 48 essais réussis contre 15 dans des conditions par ailleurs similaires. Nous notons aussi que la surface de captation de la seconde méthode couvre entièrement celle de la première : il n'y a pas d'endroit où la première méthode aurait réussi et pas la seconde.

L'analyse des résultats nous conduit à penser que la taille de la surface de captation de la première méthode est limitée par la possibilité d'apparier plusieurs secteurs avec le même secteur dans l'autre vue (la figure 9.4 l'illustre). En conséquence, dès qu'on s'éloigne du but, on arrive à des situations où la première méthode produit régulièrement des appariements très erronés. Au lieu d'attirer le robot vers le but où les écarts s'annulent, ils l'attirent vers des points où les vecteurs individuels ne s'annulent pas mais où leur somme est le vecteur nul. Nous avons observé un tel attracteur du côté du poste de télévision et de la plaque rouge. Il semble que cet attracteur, ainsi qu'un autre du côté de la planche de bois, soient responsables de la forme grossièrement triangulaire de la surface de captation en figure 9.5.

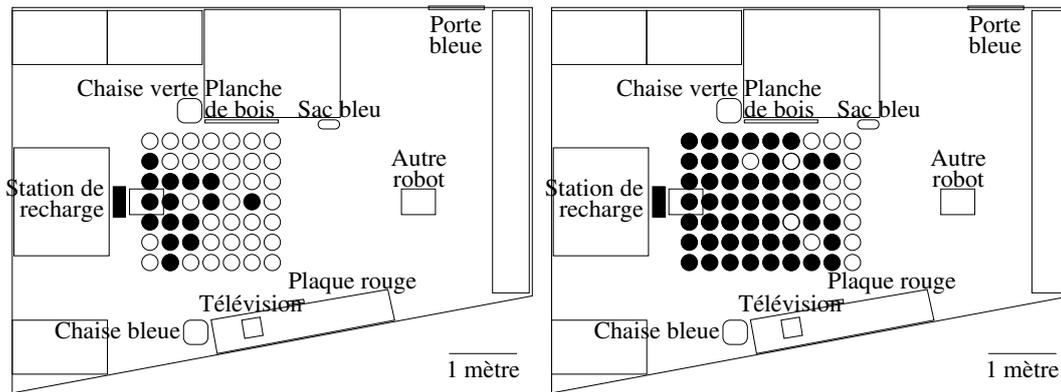


FIG. 9.5: Comparaison de performances entre l'appariement glouton d'origine des modèles CC, à gauche, et l'appariement par programmation dynamique prenant en compte la couleur, à droite, dans la salle robotique de notre laboratoire (photographie en figure 9.4). Le robot a pris une vue à la station de recharge (rectangle rempli de noir) puis a été déplacé vers l'un des points d'une grille régulière de 30 centimètres de côté avec pour tâche de revenir se connecter à la station de recharge. Chaque disque noir indique un essai réussi, chaque disque blanc un essai raté. La grille a été prolongée vers la droite jusqu'à ce que tous les essais d'une même colonne échouent. La méthode utilisant la programmation dynamique donne une surface de captation plus grande, avec un total de 48 essais réussis contre 15.

Le facteur principal limitant la taille de la surface de captation dans la seconde méthode semble être la similarité de couleurs. La chaise bleue n'est visible que près du but, du côté droit du champ de vision. Quand le robot est éloigné de la station de recharge, la porte bleue devient visible sur sa gauche. La chaise et la porte ont des couleurs très similaires de sorte que, lorsqu'il n'y a pas suffisamment d'autres objets pour lever l'ambiguïté, l'algorithme apparie les secteurs correspondants comme s'il s'agissait du même objet physique. Le résultat est un vecteur dirigé vers la gauche de la porte, qui éloigne l'animat encore plus du but. Parce que la programmation dynamique prend en compte l'ensemble de la scène, la présence d'autres objets dans la vue, comme la plaque rouge par exemple, permet de retrouver un appariement correct. Dans notre exemple, chaque fois que la plaque rouge était bien vue, la porte bleue n'était pas appariée par erreur avec la chaise bleue.

9.2.3 Discussion

Les résultats montrent que la chaîne d'éléments réalisée parvient à faire fonctionner la navigation locale sur un robot réel et qu'elle fonctionne beaucoup mieux quand l'algorithme d'appariement tient compte des couleurs et de l'ensemble de la scène, comme le fait la programmation dynamique.

L'analyse que nous avons faite a mis en évidence que le fait d'apparier un même secteur vu avec plusieurs secteurs mémorisés a tendance à attirer le robot vers ce secteur vu. Cette propriété peut être souhaitable dans le cas où quelques amers sont rassemblés dans un espace ouvert et par ailleurs vide d'amers, ce qui est le cas dans les expériences de Weber et al. [WEB 99]. Alors, quel que soit le point de départ, le comportement du robot est de s'approcher du groupe d'amers jusqu'à ce que la

répartition des amers dans le champ visuel se modifie, permettant un appariement plus précis. Dans un environnement non préparé avec des amers naturels, toutefois, cette situation est peu probable.

L'utilisation de signatures en couleurs (les couleurs fournissant l'information nécessaire pour discriminer les amers), couplée à l'utilisation de la programmation dynamique, non seulement évite d'apparier plusieurs secteurs mémorisés avec le même secteur vu, mais permet de correctement gérer le cas de la disparition d'amers. Ce point est particulièrement important quand la vision de l'animat n'est pas panoramique.

Les limitations de cette expérience concernent en premier lieu l'utilisation de l'odométrie pour maintenir fixée la direction de la caméra d'une vue à l'autre, et le temps passé à faire l'acquisition d'un panorama.

En effet, l'erreur d'orientation croît avec la distance totale parcourue par le robot. Lorsque cette dernière dépasse environ 20 mètres, ou lorsque les transitions entre mouvements de rotation et de translation se succèdent en nombre, l'erreur angulaire commence à mettre en péril la validité du modèle.

L'utilisation de la caméra mécanique freine beaucoup la vitesse de l'expérience, car il faut attendre que la caméra parcourt mécaniquement tout le panorama à raison de 25 mesures de couleurs par seconde, ce qui prend typiquement 10 à 20 secondes. L'utilisation d'un capteur panoramique peut simplifier la perception, en permettant directement une vue d'ensemble de la scène.

9.3 Comparaison avec une méthode calibrée

Nous rapportons ici une expérience de comparaison entre notre méthode de compas visuel et la méthode de Benosman [GOU 03] pour le calcul du mouvement propre.

9.3.1 Dispositif expérimental

Nous avons comparé expérimentalement, dans des conditions similaires, notre méthode d'orientation par invariant avec la méthode de Benosman.

Bien que les deux méthodes aient été conçues pour fonctionner dans des environnements non préparés, nous les avons validées dans un environnement contraint, préparé, pour deux raisons. La première est qu'un environnement contraint facilite la résolution des problèmes expérimentaux et permet d'espérer des résultats reproductibles. La seconde est que les deux méthodes n'utilisent pas les mêmes techniques d'extraction de points significatifs dans l'image d'entrée. En environnement non préparé, nous risquons en conséquence de voir des différences dues essentiellement aux répartitions différentes des points utilisés.

Le dispositif expérimental est représenté dans la figure 9.6. Autour de la surface plane rectangulaire de la table, des amers ont été disposés. Pour notre méthode, il s'agit de rectangles de papier coloré. Les transitions entre couleurs pouvaient ainsi être assez facilement extraites après segmenta-

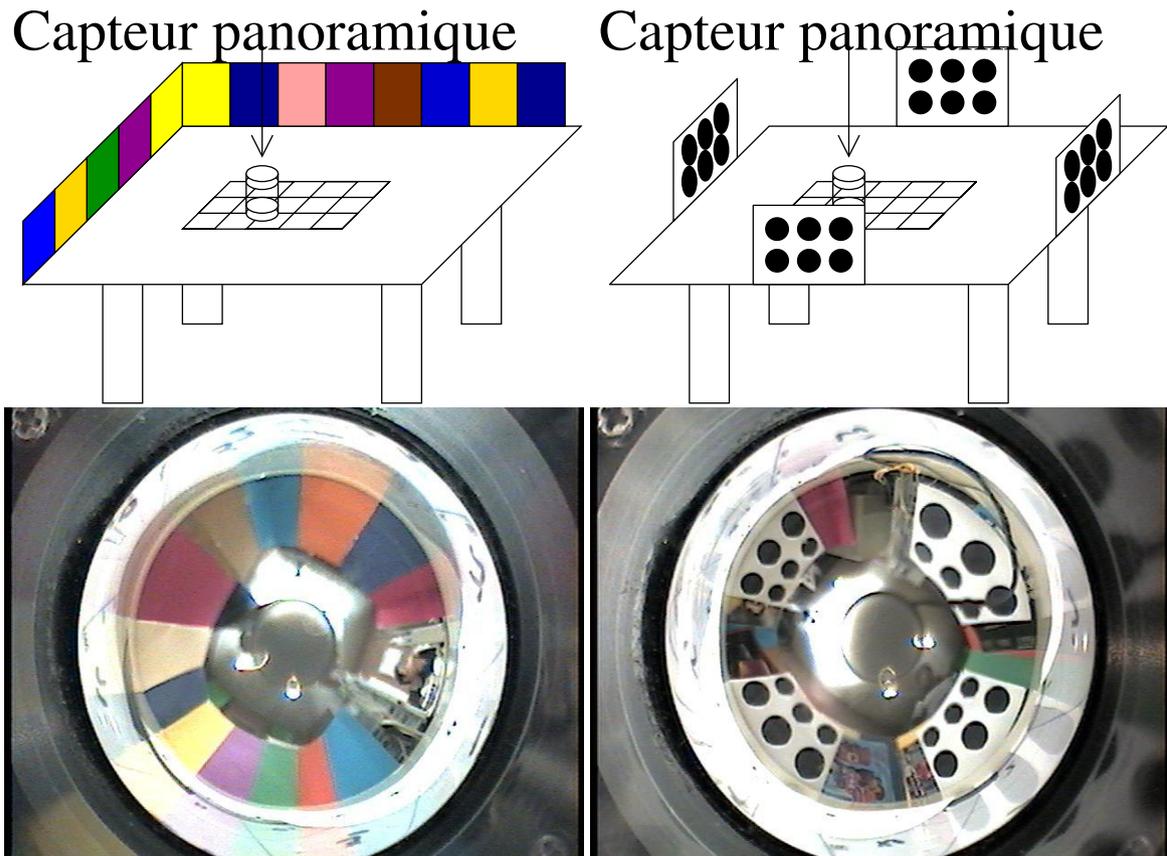


FIG. 9.6: Dispositif expérimental servant à comparer les performances de notre méthode d'orientation par compas visuel (gauche) avec la méthode calibrée de Benosman (droite). Pour notre méthode, des cartons colorés ont été attachés autour d'une table pour fournir des éléments colorés faciles à segmenter. Pour la méthode calibrée, des disques noirs étaient présentés pour garantir que des points étaient disponibles à différentes élévations.

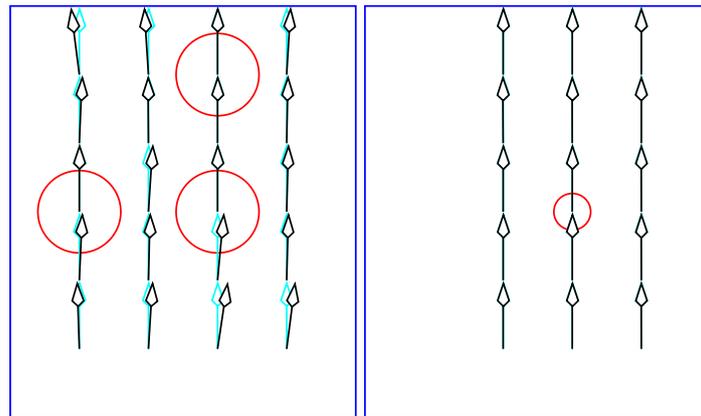


FIG. 9.7: Résultats expérimentaux pour la comparaison entre deux méthodes d'estimation de l'orientation. À gauche, notre méthode de compas visuel. Les cercles indiquent les trois points de vue utilisés pour l'étape d'observation. Chaque flèche noire montre l'orientation calculée à son point origine. Les flèches grisées montrent la vraie direction. À droite : méthode calibrée. L'erreur est trop petite pour pouvoir être visible. Seules trois rangées d'images ont été prises.

tion pour obtenir des azimuts. Pour la méthode de Benosman, des feuilles portant des disques noirs sur fond blanc ont été disposées sur les quatre côtés.

Une grille à deux dimensions a été définie sur la surface. Depuis chaque point de la grille, nous avons enregistré une image. Nous disposons donc d'un ensemble d'images prises suivant les différents points de vue. L'orientation du capteur est restée la même pour toutes les vues.

Chacune des deux méthodes a été appliquée à son ensemble d'images. La méthode de Benosman fournit à la fois une estimation des composantes de translation et de rotation du mouvement propre, alors que la nôtre ne cherche que la rotation.

Puisque l'orientation du capteur est restée en réalité constante (autant que possible), les variations dans les valeurs résultant de chacune des méthodes reflètent les erreurs commises par les méthodes. Cette expérience est donc essentiellement une comparaison d'erreurs et de dispersions.

Cette expérience s'appuie sur le fait que les développements mathématiques à l'oeuvre dans chacune des deux méthodes fournissent une estimation dont la qualité ne dépend pas de l'orientation de l'image perçue. Une alternative aurait été de tourner le capteur en chaque point dans une direction différente et connue, mais expérimentalement il était plus simple de conserver une orientation constante.

9.3.2 Résultats

9.3.2.1 Notre méthode de compas visuel

Après acquisition de toutes les vues, nous en avons choisies trois pour l'étape d'observation, indiquées sur le schéma de gauche en figure 9.7. Exploitant les triplets d'amers convenables dans ces

trois vues, cette étape a fourni 120 marques.

Ensuite, pour chaque vue disponible (y compris celles utilisées pour l'étape d'observation), nous avons appliqué l'étape d'exploitation pour estimer l'orientation qu'avait le capteur à la prise de vue. Les résultats sont montrés sur la partie gauche de la figure 9.7.

Dans cette expérience, l'erreur moyenne a été de 1.39 degrés et l'écart-type de 3.54 degrés.

9.3.2.2 Méthode calibrée

Après acquisition de toutes les vues, une des vues a été choisie comme référence (elle est montrée en rouge sur la partie droite de la figure 9.7).

Ensuite, pour chaque vue disponible nous avons appliqué la méthode calibrée qui estime l'orientation d'une vue par rapport à la vue de référence.

Dans cette expérience, l'erreur moyenne est de 0.155 degrés, et l'écart-type de 0.156 degrés.

9.3.3 Discussion

Dans les expériences que nous avons réalisées pour comparer ces deux méthodes, le traitement d'image en amont pour le compas visuel est un peu simpliste. La méthode calibrée est généralement utilisée avec en amont des algorithmes plus robustes – quoique computationnellement plus lourds – d'extraction de coins. Cependant, ces derniers n'ont pas été utilisés pour nos expériences. À la place, un algorithme d'extraction de centre a été appliqué à des disques noirs placés dans la scène. Ces choix ne remettent pas en cause la validité de la comparaison car cette dernière ne porte pas sur les calculs en amont (capacité à percevoir les éléments d'une scène) mais sur la viabilité intrinsèque des calculs effectués (capacité à trouver une orientation correcte, sachant la disponibilité d'informations angulaires extraites de la scène).

Dans ce test nous avons obtenu des résultats bien plus précis pour la méthode calibrée (écart-type 0,16 degrés contre 3,54 degrés) que pour le compas visuel. Naturellement, des expériences complémentaires seraient nécessaires pour en tirer des conclusions plus générales, d'autant plus qu'aucune des deux méthodes n'est ici testée dans le cas d'un environnement non préparé, pour lequel elles ont été pourtant conçues.

Malgré les limitations et les contraintes d'une telle comparaison, nous retiendrons tout de même une information. Il est plutôt normal que la méthode calibrée produise un résultat bien plus précis que le compas visuel et nous pensons que ce résultat est probablement représentatif.

Nous ferons une comparaison plus détaillée des deux méthodes en section 10.1.6.

9.4 Modifications des paramètres

La première validation expérimentale du compas visuel, qui a fait l'objet de la section précédente, a été rendue possible par l'utilisation d'un capteur panoramique. Nous avons eu recours à la mise en place d'une couverture continue de papiers colorés aux bords très nets et faciles à segmenter, car le but n'était pas de tester la segmentation mais la précision du compas visuel.

Ne pouvant utiliser le même capteur sur notre robot, nous avons construit un autre capteur pour valider le modèle dans une situation de navigation.

Nous nous sommes alors donné une méthode de segmentation plus robuste que celle utilisée jusque-là. Nous avons aussi affiné les paramètres de la méthode d'appariement.

Cette section détaille ces changements.

9.4.1 Capteur panoramique

Tardivement dans la thèse, nous avons réalisé un autre capteur capable de mesurer des angles sur un panorama complet de 360° ⁴.

Le capteur est représenté en figure 9.8. Une caméra classique (la même caméra Sony en l'occurrence) est pointée verticalement vers le haut. L'axe optique de la caméra, vertical, passe par un trou au centre d'un disque horizontal puis rencontre un miroir incliné à 45 degrés par rapport à la verticale. Le miroir est solidaire du disque horizontal. Ce dernier peut tourner autour de l'axe vertical, entraîné par un moteur pas à pas commandé par le robot via une liaison RS232. Ainsi, à mesure que le disque et le miroir tournent, la direction visée par la caméra balaye un plan horizontal passant par l'intersection de l'axe optique de la caméra avec le miroir.

Un interrupteur magnétique permet au robot de détecter le passage du disque à la position origine, afin de repérer le début et la fin d'une prise de panorama.

Pour prendre un panorama, le robot fait tourner le disque sur un tour complet tout en mémorisant les informations de couleur que GVPP extrait de l'image vue par la caméra. Un tour complet correspond à environ 1400 pas de moteur.

9.4.2 Salle expérimentale

La salle disponible pour nos expériences est représentée en figure 9.9.

Pour assurer la viabilité de nos expériences, nous avons dû aménager un peu l'environnement. Nous avons placé des feuilles de papier A3 colorées à la hauteur balayée par le capteur panoramique, pour assurer la présence autour de l'animat d'éléments distinguables par leur couleur.

Rappelons que, tant le système de navigation que le modèle d'orientation testés ici, s'appuient

⁴Nous remercions Patrick Pirim pour son aide lors de cette étape.

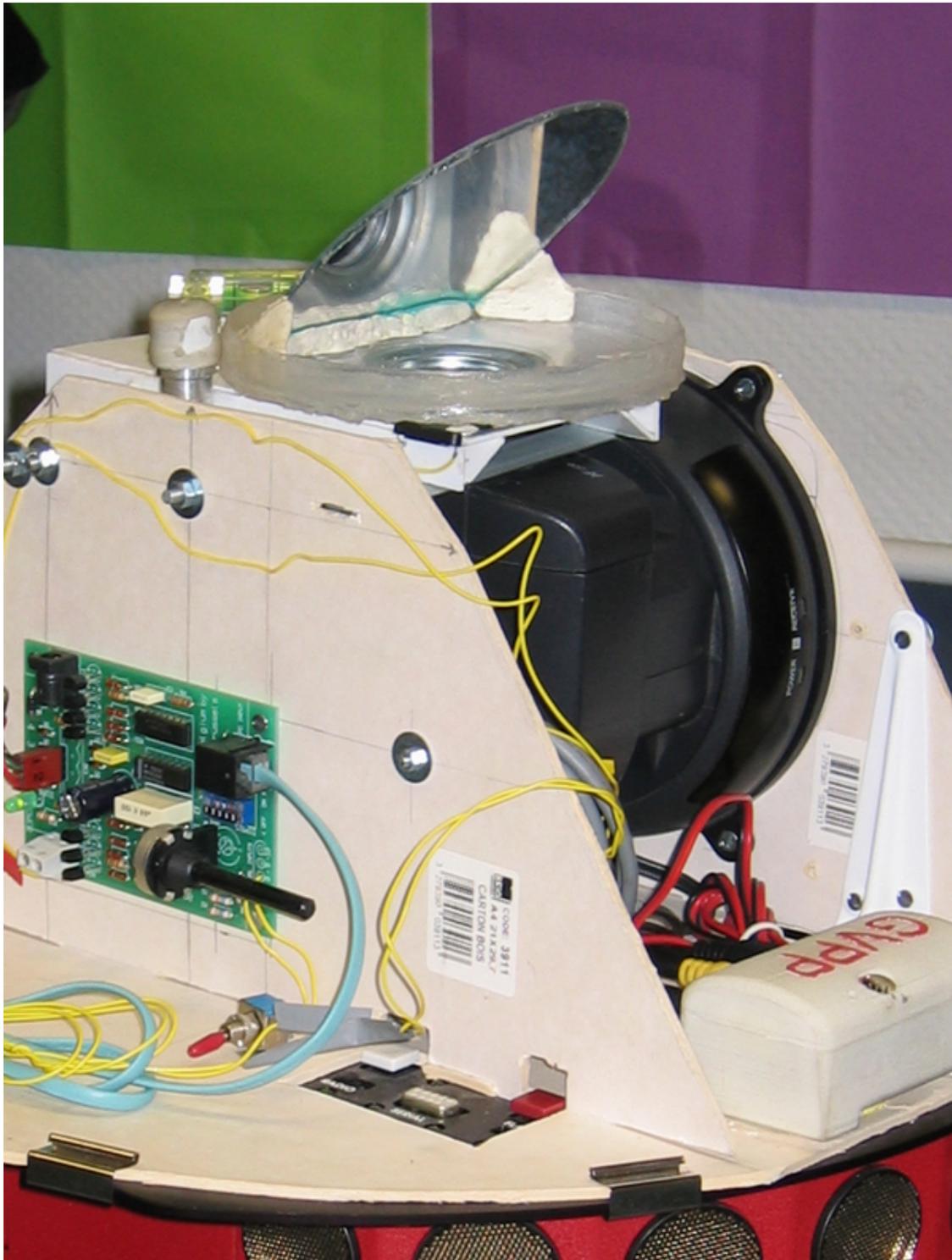


FIG. 9.8: Le capteur panoramique utilisé. Une caméra classique est dirigée verticalement vers le haut. Au-dessus, un disque supporte un miroir incliné à 45 degrés. Pour prendre un panorama, le robot fait tourner le disque sur un tour complet tout en mémorisant les informations de couleur que GVPP extrait de l'image vue par la caméra.

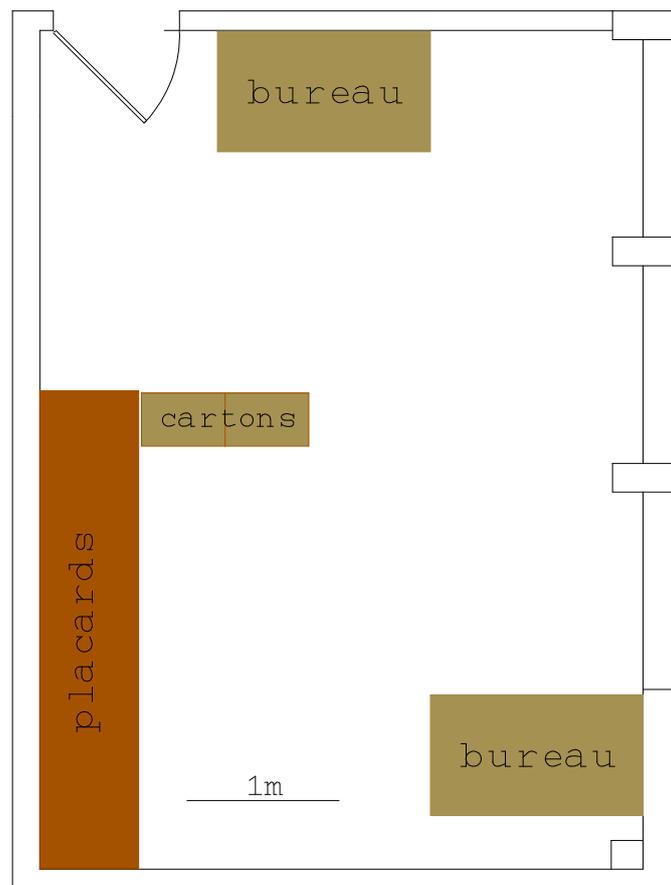


FIG. 9.9: La salle robotique où le compas visuel a été testé (échelle 1/50è).

sur des mesures d'azimuts et d'angles, et sur leur variation au cours des déplacements de l'animat. Le fait que les objets soient repérés par leur couleur ou par une autre modalité sensorielle a des implications sur les méthodes de segmentation et d'appariement, pas sur les fondements des méthodes de navigation et d'orientation. Le point important pour nos expériences est qu'il y ait compatibilité entre les possibilités du système sensoriel du robot et la richesse de l'environnement. Un animat équipé de senseurs plus performants et de systèmes plus sophistiqués pour détecter et reconnaître des amers pourrait évoluer de façon viable dans un environnement moins, voire pas du tout, préparé.

9.4.3 Segmentation par comparaison continue d'écart-type

L'algorithme de segmentation que nous avons utilisé dans nos premières expériences a été décrit en section 9.1.4. Il avait l'avantage d'être simple et suffisant pour cette série. Il permettait une segmentation efficace si le seuil de saturation dans GVPP était ajusté de sorte que les objets colorés soient sélectionnés mais non les murs blancs.

Cette méthode de segmentation nous a semblé insuffisante lors de la mise au point du compas visuel. La nécessité de régler manuellement des seuils spécifiques à un niveau d'éclairage ne nous

satisfaisait pas. Les transitions entre couleurs peu contrastées (moins que le seuil de 12 unités sur la teinte) n'étaient pas vues. En baissant le seuil on obtenait un nombre excessif de transitions non pertinentes. Les azimuts des transitions trouvées étaient très variables. Conséquence de la segmentation imparfaite, l'appariement devenait une tâche difficile. Les erreurs d'appariement causaient de grandes erreurs dans les parallaxes.

Comme nous l'avons montré en première partie de cette thèse (section 3.4.2), pour la navigation, seul le signe des parallaxes est important. Une erreur sur leur valeur précise n'empêche pas l'animat de s'approcher du but du moment que leur signe est conservé. Cela n'est plus vrai en général pour le compas visuel, où une petite incertitude sur une des parallaxes peut, par le jeu de la combinaison linéaire, signifier une grande erreur sur l'orientation.

Nous avons donc conçu et utilisé une méthode différente de segmentation. Le principe de cette méthode est de parcourir les données en calculant en continu la moyenne et l'écart-type.

Avec le capteur panoramique, le champ perçu par la caméra balaye l'espace comme un phare, mais la caméra est réglée pour couvrir un petit champ de vision. Une zone relativement petite de l'espace, généralement de couleur et d'éclairage uniforme, est donc représentée par un grand nombre de pixels dans l'image. L'ajustement automatique de sensibilité de la caméra lui permet d'obtenir une mesure de couleur assez précise même dans un environnement à l'éclairage contrasté. De son côté, GVPP travaille sur une zone carrée au centre de l'image, d'environ 40000 pixels. GVPP ajuste en continu le seuil de saturation pour sélectionner environ la moitié des pixels de cette région. Il n'est donc plus nécessaire de régler ce seuil à la main. Chaque donnée extraite d'une trame vidéo représente une information de teinte, de saturation et un nombre de pixels.

Pour chaque donnée de couleur (une par trame vidéo), l'ordinateur calcule la position de l'information de couleur (chrominance) dans l'espace CrCb. Il maintient à jour, à mesure du parcours des données, des variables représentant la somme, la somme des carrés et le nombre de pixels, ce qui permet efficacement de connaître à chaque instant moyenne et écart-type de chaque variable. Pour chaque nouvelle valeur, on compare sa distance avec la valeur précédente dans l'espace CrCb (les points dans cet espace sont dans un disque de centre l'origine et de rayon 1). Si cette distance est supérieure à 1.6 fois l'écart-type (racine de la somme des variances sur Cr et Cb), on note qu'il y a une transition et on réinitialise les compteurs.

Le compteur de nombre de points est initialisé à zéro, mais pas la somme des carrés des Cr et Cb, qui prend une valeur de 80. Ce procédé biaise l'écart-type calculé : ce n'est plus vraiment un écart-type mais une valeur très élevée au début et après chaque transition, qui tend vers l'écart-type vrai (quand la somme des carrés devient très supérieure à 80). L'effet est d'inhiber la création de transitions trop rapprochées les unes des autres. Sans cet effet, la grande incertitude sur l'écart-type au démarrage et après chaque transition aboutit à de nombreuses transitions intempestives.

Nous avons initialement essayé la segmentation par comparaison continue d'écart-type dans l'espace teinte-saturation (celui des données reçues). Mais la teinte n'étant pas définie pour des objets gris, elle peut varier énormément pour des objets simplement peu colorés. Nous n'obtenions pas de segmentation satisfaisantes. L'espace CrCb a résolu ce problème et les segmentations qu'il fournit

sont précises, reproductibles, et restent pertinentes quand l'animat se déplace.

9.4.4 Modifications concernant l'appariement

On conserve la méthode d'appariement par programmation dynamique, qui implique le remplissage d'une matrice de ressemblance entre les éléments à appairer. L'appariement ne se fait plus sur les secteurs mais sur les transitions. La fonction de ressemblance fait donc intervenir les deux côtés de chaque transition, comme cela a été évoqué en figure 3.6. Elle estime pour deux éléments donnés (ici, des transitions) le score qu'apporterait la création d'une paire les reliant.

Avant de tenir compte de ce qu'il s'agit d'appairer des transitions nous avons adapté la fonction de ressemblance entre secteurs, donnée en section 9.1.5, à l'espace CrCb : on calcule l'écart en couleur comme la distance entre les deux points correspondant aux deux secteurs, dans l'espace CrCb, au lieu de deux termes correspondant à la teinte et à la saturation.

Après avoir essayé cette fonction de ressemblance, nous avons constaté que souvent des petits secteurs dus à des défauts de segmentation étaient appariés de façon non pertinente, parfois avec de grands secteurs. Nous avons donc introduit deux facteurs dépendant de la largeur des secteurs (un pour chaque secteur). Chaque facteur pénalise (sans l'interdire totalement) l'appariement des secteurs très étroits, ce qui minimise leur impact. On note l la largeur angulaire d'un secteur et $l_{caractéristique}$ une constante fixée à 10° . Le facteur du à la taille est :

$$\exp\left(1 - \frac{l}{l_{caractéristique}}\right) \quad (9.2)$$

Ainsi, les secteurs très petit devant la largeur caractéristique sont très défavorisés, alors que les secteurs plus larges que la largeur caractéristique ne sont pas perturbés par ce facteur qui est alors proche de 1.

La fonction décrite ci-dessus, qui détermine la ressemblance entre deux secteurs, est utilisée pour estimer la ressemblance entre deux transitions. La règle est que la ressemblance d'une transition est le produit de la ressemblance entre les deux secteurs d'un côté de la transition par la ressemblance entre les deux secteurs de l'autre côté de la transition.

Nous avons constaté qu'avec une telle règle des transitions peu contrastées, voire entre secteurs de même couleur, étaient appariées. Or nous souhaitons qu'une transition dans un panorama soit appariée en priorité si elle correspond à une transition de fort contraste dans la scène. Au contraire, une transition entre deux secteurs de même couleur est probablement un artefact de segmentation.

Nous multiplions donc la fonction de ressemblance par un facteur qui pénalise les transitions non contrastées. Ce facteur est défini par le produit des distances dans l'espace CrCb. Ainsi, une transition qui sépare deux secteurs de même chrominance obtiendrait un score nul si elle était appariée avec toute autre.

Le produit de l'ensemble des facteurs décrit est le score d'appariement de deux amers. Ce jeu de

paramètres nous a fournis en simulation comment pour nos expériences robotiques des appariement robustes et pertinents.

9.5 Test analytique du compas visuel : orientation seule

Le but de l'expérience présentée dans cette section est de tester le compas visuel de façon analytique (par opposition à synthétique, cf. 2.1.6), c'est-à-dire contrôler si localement l'information fournie par le compas est proche de la direction réelle du robot. Les mesures synthétiques sont dans la prochaine section. Cette expérience met en évidence d'autres aspects :

- capacité de la partie perception (capteur panoramique, GVPP et logiciel de contrôle) à récolter des données pertinentes et précises ;
- capacité de la méthode de segmentation par comparaison continue d'écart-type ;
- capacité de la méthode d'appariement par programmation dynamique adaptée au cas circulaire ;
- capacité de la méthode de l'étoile polaire d'une part, du compas visuel d'autre part, à estimer l'orientation de l'animat.

9.5.1 Déroulement de l'expérience

Dans cette expérience l'animat, un robot Pioneer 2, est équipé du capteur panoramique décrit en 9.4.1. Le but est de comparer la précision permise par le compas visuel et par la méthode de l'étoile polaire (section 6.2). L'expérience est représentée en figure 9.10. Dans une première phase, l'animat prend trois vues pour réaliser une étape d'observation du compas et pour choisir l'amer qui servira d'étoile polaire. Ensuite, il parcourt une grille au sol et en chaque point applique le compas visuel et la méthode de l'étoile polaire pour estimer son orientation. Les distances traversées étant courtes, on s'appuie sur l'odométrie comme référence.

9.5.2 Résultats

La partie de gauche de la figure 9.10 montre le résultat. Rappelons que la méthode de l'étoile polaire consiste à supposer que l'amer dont l'azimut a le moins varié pendant l'étape d'observation donne une direction de référence fixe, ce qui n'est exact que s'il est infiniment loin. Les directions obtenues par la méthode de l'étoile polaire donnent un champ de vecteurs non parallèles (ce qui serait le cas si l'amer était infiniment loin).

Au contraire, le compas visuel, en exploitant les relations linéaires obtenues lors de l'étape d'observation, enregistrées dans les marques, parvient à donner une estimation d'orientation plus constante, comme le montre le champ de vecteurs quasiment parallèles obtenu.

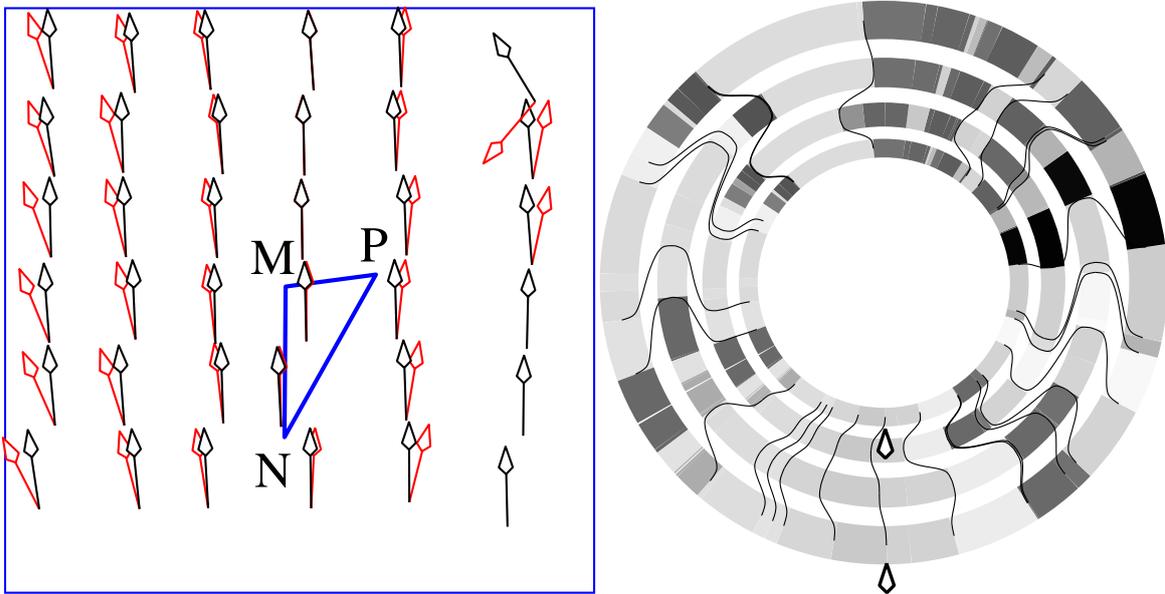


FIG. 9.10: Test du compas visuel et comparaison de performance avec la méthode de l'étoile polaire. L'animat a pris trois vues (triangle au centre du diagramme de gauche) et réalisé une étape d'observation pour prendre ses marques, représentée de façon synthétique à droite. Les quatre anneaux représentent, de l'extérieur vers l'intérieur, les vues M,N,P,M. Chaque zigzag coloré représente un amer apparié de façon cohérente dans chacun des trois appariements, et donc sélectionné par l'algorithme du compas visuel pour l'énumération des triplets et la construction des marques. La pointe en bas du diagramme est l'amer qui a servi d'étoile polaire. Muni de ces éléments, l'animat a parcouru une grille carrée montrée à gauche. Sur chaque point de la grille, il a pris une nouvelle vue et estimé l'orientation à l'aide des deux méthodes. Le résultat est représenté sous la forme d'une flèche. Quand l'accord avec la référence est parfait, la flèche est vers le haut, sinon elle est décalée de l'erreur d'angle. Les flèches rouges correspondent à la méthode de l'étoile polaire, les noires au compas visuel. La méthode de l'étoile polaire n'est pas très bonne car il n'y a aucun amer lointain dans le panorama, et donc tous les amers bougent dans le champ de vision. Au contraire, le compas visuel trouve une bonne orientation même en des points relativement éloignés des points de calibration. On note un point aberrant dans le coin nord-est, dû à une défaillance locale de la méthode de segmentation. Trois flèches rouges manquent dans la colonne de droite. C'est le cas quand la procédure d'appariement n'a apparié l'étoile polaire à aucun des amers actuellement visible.

9.6 Test synthétique du compas visuel : navigation avec orientation

Après les mesures analytiques, le but de cette section est de tester le compas visuel de façon synthétique (cf. 2.1.6), c'est-à-dire de contrôler si, pour une tâche de plus haut niveau, l'information fournie par le compas rend plus viable la méthode de navigation.

9.6.1 Méthode expérimentale

Notre recherche est motivée par le souhait de rendre l'animat aussi autonome que possible. Dans le cas présent, il s'agit de lui permettre de naviguer sans perdre son orientation sur de longues périodes. Pour des raisons matérielles (batteries notamment) et pratiques (liées en particulier aux durées d'expériences, à leur nombre et leur reproductibilité), nous ne pouvons nous permettre de laisser l'animat se déplacer très longtemps jusqu'à pouvoir mettre en évidence l'insuffisance de l'orientation par odométrie et, ainsi, le bénéfice du compas.

Pour obtenir des résultats équivalents, nous avons volontairement introduit une perturbation de l'orientation donnée par l'odométrie, dont l'importance est réglable. Nous avons fait le choix de perturber cette orientation de façon systématique et cumulative : à chaque déplacement de l'animat, elle est décalée d'une grandeur constante définie pour la durée de l'expérience.

On estime l'ordre de grandeur de l'erreur spontanée de l'odométrie à un degré pour un à quelques mètres de déplacement, mais cette erreur dépend de facteurs tels que le gonflement et l'usure des roues du robot et la planéité du sol, notamment. La perturbation que nous imposons est plus grande que l'erreur naturelle d'odométrie. Nous l'avons fait varier de 1 à 20 degrés par pas, un pas faisant toujours moins de 50cm.

Cette approche nous permet de comparer les performances de chaque méthode d'orientation à perturbation égale, et d'observer comment la durée de vie de l'animat varie en fonction de l'importance de la perturbation, tout en s'affranchissant des autres facteurs qui interviendraient dans des expériences plus longues.

9.6.2 Déroulement des expériences

Nous avons défini une expérience qui met en évidence plusieurs caractéristiques de la capacité de navigation dont est équipé l'animat :

- navigation de proche en proche entre plusieurs sous-buts ;
- gestion des occultations (l'ensemble des amers visibles n'est pas le même le long de la trajectoire) ;
- utilisation de plusieurs ensembles de marques d'orientation de sous-but en sous-but.

Le schéma 9.11 montre le déroulement concret d'une expérience. L'animat doit faire des allers-retours sur un parcours en forme de C sans heurter d'obstacles. En préliminaire à chaque expérience, le robot utilise son odométrie pour parcourir une trajectoire programmée, en mémorisant 10 vues qui

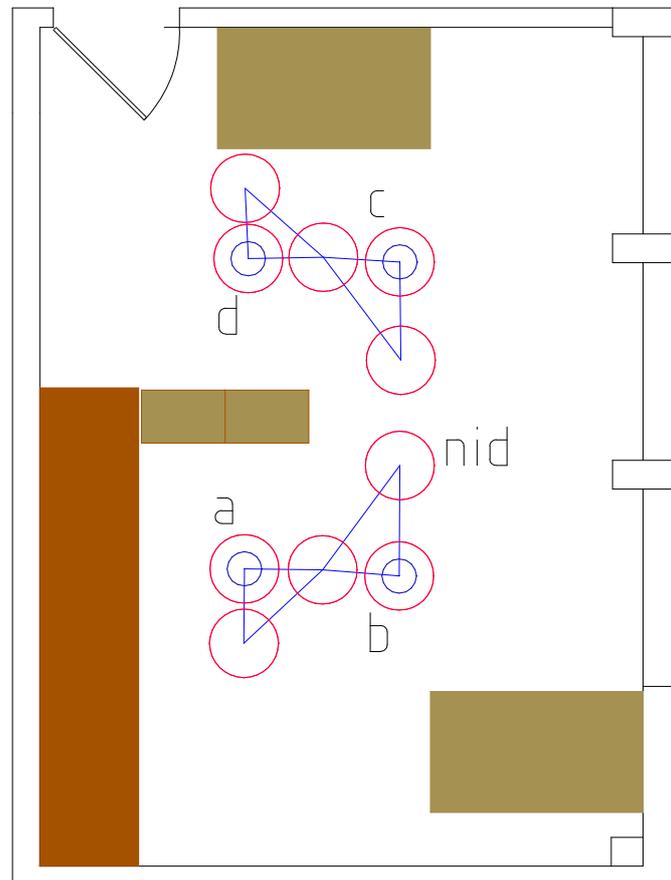


FIG. 9.11: Validation robotique : navigation avec orientation. Le robot doit faire des allers-retours sur un parcours en forme de C sans heurter d'obstacles. En préliminaire à chaque expérience, le robot utilise son odométrie pour parcourir une trajectoire programmée, en mémorisant 10 vues, représentée chacune par un grand cercle. Quatre étapes d'observation par le compas visuel sont réalisées, représentée chacune par un triangle reliant les trois vues observées et un petit cercle qui indique la vue à laquelle sont attachés les marques. (Idéalement, l'animat aurait déterminé des vues de façon autonome suite à une exploration de l'environnement.) Ceci fait, le test de navigation commence. On mesure pendant combien de temps l'animat réussit à faire des allers-retours entre ces quatre points en séquence (d-c-b-a-b-c-d) sans heurter d'obstacle. On répète l'expérience avec divers niveaux de perturbation d'orientation, avec et sans compas visuel. Dans les expériences où le compas est disponible, l'animat utilise à chaque instant les marques attachées à la vue qu'il tente de joindre.

permettent quatre étapes d'observations par le compas visuel. Ensuite, l'animat fait des allers retours entre ces 4 points. L'expérience est arrêtée si le robot se bloque contre un obstacle ou après un nombre défini d'allers-retours. On tolère que l'animat touche un obstacle sans se bloquer.

Précisons comment se passent les allers-retours. L'animat avance pas à pas. Un pas représente une mesure du panorama coloré (qui prend environ 30 secondes) puis un déplacement suivant un segment, ordonné par le système de navigation (environ 5 secondes). Les déplacements sont filmés en vidéo. Le système de navigation est configuré ainsi : méthode PV, composante tangentielle, méthode directe, gain constant.

L'animat dispose de 3 pas pour aller d'un point mémorisé à un autre, sauf entre les points b et c où il dispose de 4 pas. Le premier aller parcourt les trajets d-c-b-a, le retour les mêmes dans l'ordre inverse. Un aller-retour comprend donc $(3 + 4 + 3) \times 2 = 20$ pas. Après 4 allers-retours réussis, l'animat doit retourner au nid par le trajet d-c-nid en $3 + 3$ pas. La durée maximale de l'expérience est donc de 86 pas. Une fois revenu au nid, l'animat doit se replacer dans la direction qu'il avait au tout début de l'expérience. Cette dernière étape donne un moyen de plus de contrôler si l'expérience s'est bien déroulée.

Cette expérience est répétée trois fois pour chaque niveau de perturbation étudié : pas de correction de la perturbation, correction par la méthode de l'étoile polaire, correction par le compas visuel. La correction d'orientation est appliquée le cas échéant à chaque pas de déplacement. La méthode de l'étoile polaire est ainsi appliquée en « tout ou rien » : si l'amer servant d'étoile polaire est apparié avec la vue actuelle, l'indication donnée est appliquée complètement, sinon aucune mise à jour de la meilleure estimation d'orientation n'est appliquée. Lorsque c'est le compas qui est utilisé, on applique la fusion avec les données odométriques comme décrit en 6.4.6.

Cette expérience, en faisant intervenir toute la chaîne depuis la perception, met en évidence d'autres aspects :

- capacité de la partie perception (capteur panoramique, GVPP et logiciel de contrôle) à récolter des données pertinentes et précises ;
- capacité de la méthode de segmentation par comparaison continue d'écart-type ;
- capacité de la méthode d'appariement par programmation dynamique adaptée au cas circulaire ;
- capacité de la méthode de l'étoile polaire d'une part, du compas visuel d'autre part, à estimer l'orientation de l'animat de façon suffisamment précise pour restaurer la viabilité de la navigation à l'aide de la composante tangentielle.

9.6.2.1 Résultats et interprétation

Le but de cette expérience était d'évaluer le bénéfice que le compas visuel apporte à un robot réel dans une tâche de navigation, face à une désorientation importante. La figure 9.12 rend compte des performances de l'animat. En abscisse est indiqué de combien l'animat est perturbé à chaque pas. En ordonnée, le nombre de pas qu'a duré l'expérience (86 au maximum, ou le nombre de pas effectués avant que l'animat ne se bloque contre un obstacle). La courbe rouge représente les expériences sans

Durée de vie de l'animat en fonction de la perturbation d'orientation.

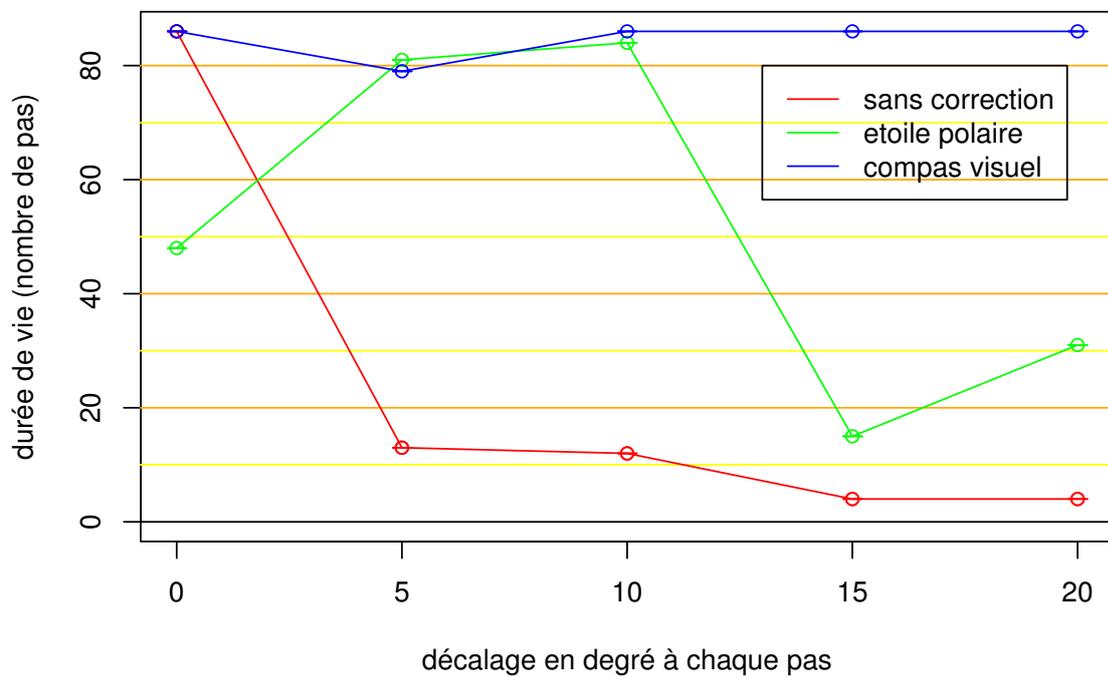


FIG. 9.12: Navigation avec une méthode d'orientation disponible sur la totalité du trajet. On compare la méthode du compas visuel, la méthode de l'étoile polaire, et l'absence de méthode de réorientation. Sans compas, l'animat n'arrive à parcourir tout le trajet que s'il n'y a aucune perturbation. La méthode de l'étoile polaire permet à l'animat de survivre plus longtemps. Le compas visuel donne les meilleurs résultats.

réorientation, la courbe verte la méthode de l'étoile polaire, la courbe bleue le compas visuel.

La différence de durée de vie est très nette. On constate que, sans compas visuel, l'animat n'a pu parcourir tout le trajet que quand il n'y avait aucune perturbation. Il a toujours échoué avant de finir le premier-aller-retour (perturbations de 5 et 10 degrés), voire avant le premier aller (à 15 et 20 degrés). La méthode de l'étoile polaire permet des résultats meilleurs mais inégaux. Enfin, équipé du compas, l'animat a presque toujours la durée de vie maximale. L'analyse du point qui fait exception (à 5 degrés) montre qu'il est dû à un problème de segmentation non imputable au compas⁵.

Il est intéressant de se demander si la variation de performance de l'animat selon l'intensité de la perturbation est normale ou bien si la performance devrait en être indépendante. En effet, à chaque pas où la méthode de l'étoile polaire fonctionne, l'orientation estimée par l'animat est celle donnée par la méthode, donc indépendante de la perturbation. De même, en ce qui concerne le compas, lorsque plus de 4% des marques sont utilisées à un pas donné (cf 6.4.6), l'orientation estimée par l'animat est celle donnée par le compas, donc indépendante de la perturbation.

Considérons les cas où localement l'estimation d'orientation dépend de l'intensité de la perturbation. Dans le cas de l'étoile polaire, il s'agit des pas où cette méthode a échoué parce qu'aucun amer n'a été apparié à celui désigné comme étoile polaire. Dans le cas du compas visuel, il s'agit des pas où moins de 4% des marques ont pu être exploitées.

Sur 257 utilisations de la méthode de l'étoile polaire, 200 ont effectivement fourni une estimation d'orientation, soit 78% des cas. Dans les 22% restant où la méthode n'a fourni aucune estimation, la perturbation a bien eu une incidence sur la trajectoire. Il n'est donc pas surprenant que la performance soit globalement moins bonne en cas de grande perturbation.

Le compas visuel a été exploité 496 fois dans cette série d'expériences. 450 fois (soit 83% des cas), plus de 4% des marques ont été trouvées, et l'animat a donc considéré l'indication du compas comme suffisamment sûre pour constituer l'orientation vraie du robot. 44 fois (soit 9% des cas), moins de 4% des marques ont été trouvées, une correction partielle a été appliquée. Enfin, 2 fois (0.004% des cas), le compas n'a pu exploiter aucune marque. On peut donc s'attendre à ce que la performance soit, elle aussi, moins bonne en cas de grande perturbation. Elle est restée excellente toutefois.

Nous estimons que la différence de performance entre les deux méthodes dans nos expériences est certes due à la moindre précision de la méthode de l'étoile polaire, mais principalement au fait que, plus souvent que le compas visuel, elle n'a pas été en mesure de donner une estimation d'orientation.

En résumé, on constate que la méthode de l'étoile polaire, malgré sa rusticité, permet à l'animat de compenser des erreurs d'orientation mais que sa performance est inégale, ce qui se constate particulièrement en cas de grande perturbation. Le compas visuel permet des résultats nettement meilleurs, en petite comme en grande perturbation.

⁵Un examen de plus près montre que la segmentation de trois des amers les plus saillants de la scène s'est mal déroulée, en conséquence l'appariement était suffisamment perturbé pour que peu de marques soient visibles.

9.7 Limites et viabilité dans d'autres contextes

Nous aurions beaucoup apprécié de pouvoir laisser le robot naviguer de façon autonome dans les couloirs du laboratoire pendant de longues périodes. Les expériences que nous avons réalisées nous permettent de résumer ici les limitations et difficultés qui ont empêché cette entreprise, et de proposer des solutions aux problèmes correspondants.

Le schéma 9.13 rappelle la chaîne d'éléments nécessaires. Nous les passons en revue en tenant compte des caractéristiques du lieu et en expliquant les difficultés que cela représente pour le système que nous avons développé dans cette thèse.

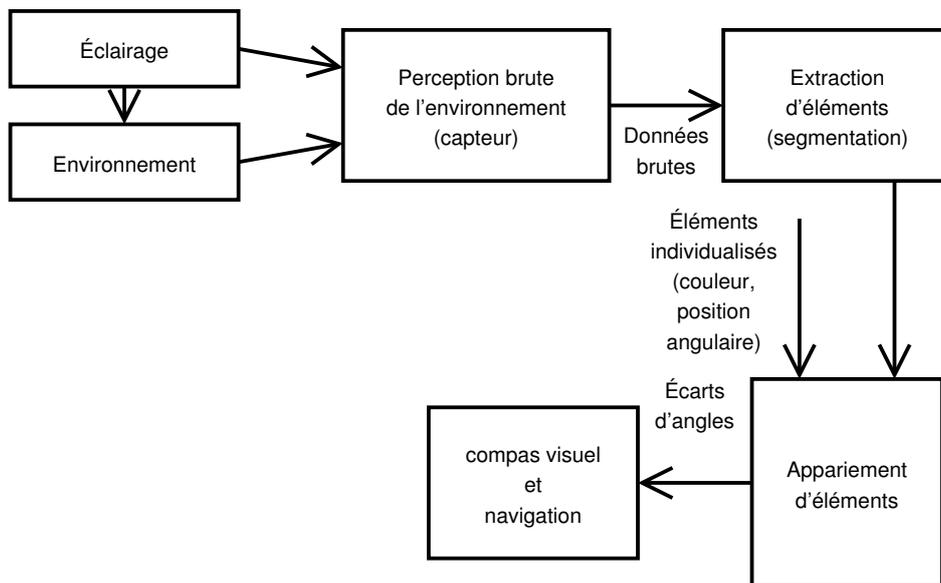


FIG. 9.13: Étapes fonctionnelles sur lesquelles s'appuie notre implémentation du compas visuel et de la navigation par mesure d'angles.

L'environnement dans lequel nous travaillons est représenté en figure 9.14. Le couloir principal, orienté sud-est nord-ouest, fait 40 mètres de long et (en général) 1,70m de large. À l'extrémité sud-est, il fait 3 mètres de larges sur une longueur de 7,5m. Le côté sud-ouest présente 14 portes, une tous les 3m. Le côté nord-est présente 5 petits couloirs.

Éclairage de l'environnement Nous avons choisi d'utiliser la lumière naturelle pour nos expériences. Bien que changeante, la lumière naturelle a un spectre riche qui permet une bonne répartition des couleurs dans le signal récupéré et donc une bonne reconnaissance lors de l'appariement. L'éclairage artificiel de la salle expérimentale, d'une intensité d'environ 400 à 500 lux au sol permet aussi aux expériences de fonctionner. En revanche, l'éclairage dans les couloirs n'est que de 100 lux environ.

Richesse de l'environnement Les panoramas de la salle expérimentale ont une certaine structure : mur, bureaux, placards. L'ajout de papier coloré a introduit des éléments de couleurs plus variées, plus saillants et donc plus facilement reconnaissables, facilitant d'autant la tâche de l'ani-

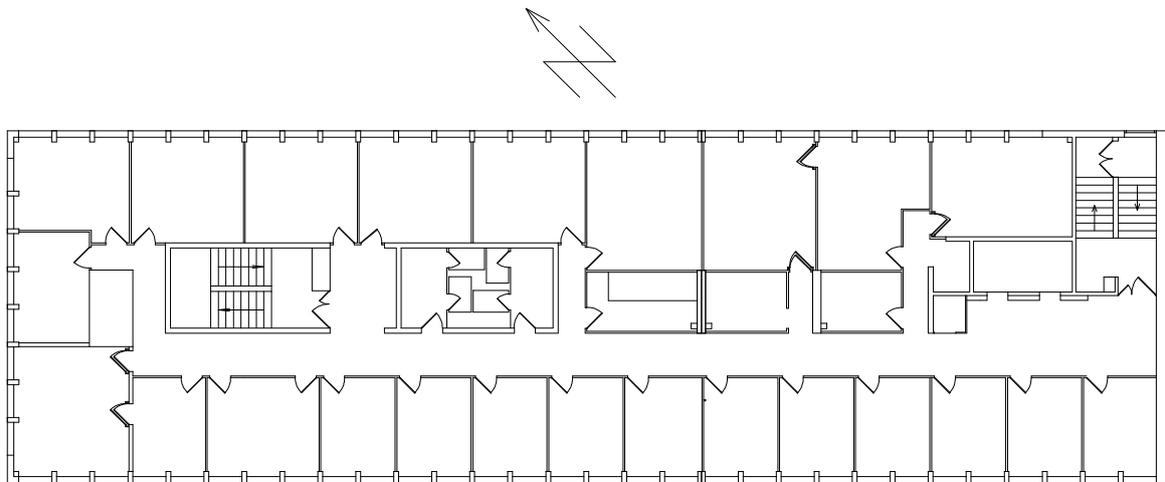


FIG. 9.14: Les couloirs du laboratoire.

mat. En revanche les murs des couloirs, à la hauteur du robot, sont nus. Les autres éléments sont des portes bleues toutes identiques, des ascenseurs gris et des portes vitrées menant aux escaliers. Des éléments tels que les chambranles des portes, ou les arêtes de murs, sont trop peu contrastés et donc inaccessibles à un capteur circulaire comme le nôtre.

Hauteur du robot À la hauteur où le robot évolue se trouvent particulièrement peu d'éléments. Les éléments tels qu'affiches et posters sont tous entre 1 mètre et 2 mètres de hauteur. Le robot devrait, soit être à une hauteur adaptés aux objets qu'il peut voir, soit disposer d'une vue capable de bien percevoir des objets placés nettement plus haut que lui, ce qui est plus difficile.

Le capteur La qualité et la sensibilité du capteur sont déterminants. La caméra que nous avons utilisée est un produit grand public de bonne facture. Cette caméra, de type CCD, peut s'accommoder d'une assez grande variation de luminosité. En outre, comme le miroir tournant ne montre à la caméra qu'une partie de l'environnement à un instant donné, cette dernière peut s'ajuster pendant la prise de panorama. Cependant des tests que nous avons effectués dans les couloirs montrent que les couleurs sont mal perçues dans cet environnement. Compte tenu du choix que nous avons fait d'utiliser la couleur, l'environnement et l'éclairage de la salle robot sont au-dessus de la limite de viabilité, ceux des couloirs bien en dessous pour ce capteur. Par ailleurs, nous avons eu l'occasion d'utiliser un capteur panoramique utilisant un miroir de révolution et un imageur CMOS. Malheureusement cet imageur a une faible dynamique : les zones de forte lumière, comme celles de basse lumière, sont inexploitable. Ce type de capteur a l'avantage de produire un panorama instantané mais, pour être viable, il nécessiterait un imageur beaucoup plus sensible (l'idéal serait d'avoir une dynamique indépendante pour chaque pixel) et probablement plus gros. Par ailleurs, l'extraction d'autres éléments de l'environnement que la couleur serait probablement nécessaire. Ainsi, l'extraction de coins dans une image 2D est probablement plus robuste que l'utilisation de panoramas à une dimension.

La segmentation La segmentation nous a posé beaucoup de problèmes en fin de thèse. Nous avons dû abandonner la segmentation très simple que nous avons utilisée en milieu de thèse, car elle né-

cessitait un réglage manuel de seuil et des amers dédiés. Pour la validation robotique avec orientation, nous avons développé en fin de thèse une segmentation plus robuste et tenant compte de plus d'informations, modifications nécessaires pour des appariements fiables face à une orientation arbitraire. Le fonctionnement est bon avec un éclairage naturel, ou avec l'éclairage de la salle robot. Mais dans l'environnement des couloirs avec le capteur actuel, la segmentation n'est que correcte pour une feuille de couleur vive ou une porte placée à proximité (1 mètre ou moins). Pour les autres éléments, la segmentation les accroche parfois, mais n'est pas du tout fiable ni assez précise.

L'appariement L'appariement utilisé distingue les éléments de l'environnement par leurs couleurs. Il échoue si les couleurs ne suffisent pas à différencier ces éléments. C'est le cas des portes et des murs d'un environnement de bureau. Une centaine de feuilles de couleurs variées, réparties le long des couloirs, permettrait à un robot équipé d'un capteur suffisamment sensible d'apparier correctement.

Le compas visuel Si les étapes amont fonctionnent correctement, les informations que reçoivent le système de navigation et le compas visuel ne sont que des mesures de variations d'angles, indépendamment du détail des éléments de l'environnement. À ce niveau, cela n'a pas d'importance que les amers se distinguent par leur couleur ou non. Si, dans quelques zones réparties dans l'environnement, le robot peut utiliser de façon fiable au moins 5 éléments (strict minimum 3), la réorientation devrait fonctionner.

La navigation locale Si l'orientation est connue, il suffit qu'à chaque instant 2 parallaxes soient mesurables de façon fiable (à l'exclusion du cas où tous les amers soient alignés avec le but) pour que la navigation fonctionne. En pratique, 3 parallaxes et plus sont préférables et, de toute façon, le compas visuel les exige.

La navigation avec carte La construction d'une carte en situation réelle – et sous réserve que les étapes de perception et de segmentation fonctionnent bien – permettrait de réduire la contrainte de couleurs variées en compensant la ressemblance entre une porte et une autre porte par le suivi en continu de la position du robot dans la carte. On entre alors dans le problème de la localisation et cartographie simultanées (souvent abrégé en SLAM pour Simultaneous Localization And Mapping) comme cela a été étudié par Filliat [FIL 01, MEY 03, FIL 03]. En une phrase, il s'agit de pouvoir représenter plusieurs hypothèses sur la position du robot dans la carte, et d'éliminer à mesure du déplacement du robot les hypothèses qui deviennent incompatibles avec les nouvelles observations.

En résumé, notre animat dispose de capacités perceptives et cognitives très limitées. Si des éléments distinguables et suffisamment nombreux sont visibles, il sait où aller. Mais les couloirs du laboratoire – sombres, où toutes les portes se ressemblent et ne permettent pas de mesures de parallaxes précises – ne permettent pas de retrouver son chemin avec l'équipement sensoriel actuellement disponible.

Au sortir de ces considérations, il est possible d'émettre une série de recommandations pour obtenir un robot autonome dans les couloirs du laboratoire. Puisqu'il y a incompatibilité entre le capteur

utilisé, le choix de s'appuyer essentiellement sur la couleur des objets environnants et la densité d'objets colorés présents dans les couloirs, les éléments suivant sont à modifier, de préférence dans cet ordre :

- utiliser un capteur plus sensible ;
- exploiter d'autres éléments en complément de la couleur ;
- placer la caméra plus haut (si possible) pour bénéficier des éléments visibles à hauteur d'homme ;
- ajouter des objets colorés sur les murs ;
- incorporer la localisation et cartographie simultanées.

Plus on améliorerait ces points, meilleure devrait être la viabilité du résultat.

9.8 Conclusion sur les expérimentations robotiques

Nous avons successivement validé en situation robotique réelle les différents éléments de la chaîne nécessaire pour réaliser la navigation par mesure d'angles, puis l'orientation par mesure d'angles. Le succès de ces expériences montre qu'à la fois le principe de la méthode, mais aussi les éléments matériels mis en place, sont suffisants dans les conditions de nos expériences.

Les expériences analytique (champs de vecteurs) montrent que le compas visuel fournit une estimation de la direction de l'animat bonne, et nettement plus juste que la méthode de l'étoile polaire plus simple.

Les expériences synthétiques (navigation avec orientation) montrent que le compas visuel permet à l'animat de naviguer malgré des perturbations odométriques importantes, plus fortes que celles supportables sans méthode de réorientation. La méthode de l'étoile polaire améliore aussi la durée de vie de l'animat, mais ses performances sont inégales.

Enfin, souhaitant non seulement valider le principe de nos méthode, mais aussi leur utilisation dans un environnement réaliste, nous avons reconsidéré, à la lumière des expériences réalisées, les choix que nous avons faits pour les faire fonctionner matériellement. Le choix d'utiliser des panoramas circulaires en couleurs, guidé par le matériel dont nous disposions, nous a permis de progresser dans la suite des modèles que nous avons étudiés. Nous avons constaté les limites de ces choix face à un environnement plus difficile comme les couloirs du laboratoire et avons donné des indications sur les voies à explorer pour dépasser ces difficultés.

Quatrième partie

Discussion

Chapitre 10

Discussion

Après avoir présenté les travaux sur lesquels nous nous sommes appuyé, ensuite nos propres travaux, de façon d'abord théorique puis leur validation expérimentale, il est temps de détailler comment nos choix se comparent aux choix qui ont été faits dans les travaux similaires aux nôtres.

Ce chapitre couvre donc des comparaisons avec les modèles dont nous avons repris des éléments précis ou qui nous ont simplement inspiré. Nos comparaisons, lorsque cela est pertinent, seront orientées suivant deux axes. D'une part, les aspects relatifs aux méthodes de navigation, indépendamment des considérations d'orientation, d'autre part les aspects relatifs à l'orientation.

Puis nous dresserons un bilan de nos travaux et évoquerons quelques perspectives possibles à la suite de ce travail.

Il sera alors temps de conclure.

10.1 Comparaisons avec d'autres approches

Nous commençons notre comparaison par des considérations d'ensemble, faisant éventuellement intervenir plusieurs autres modèles, avant de détailler la comparaison à des modèles individuels.

10.1.1 Considération d'ensemble

10.1.1.1 Signature d'un lieu

Notre signature d'un lieu, avec la caméra mécanique (mais pas avec le capteur panoramique), tient compte des objets situés au-dessus et en dessous de l'horizon jusqu'à des angles importants (de -25° à $+25^\circ$), alors que les capteurs utilisés par la plupart des autres chercheurs (Röfer, Franz mais aussi Möller et Lambrinos) sont restreints à une fine frange centrée sur l'horizontale.

Cette différence n'est pas forcément importante pour des expériences de test mais peut faire une différence, notamment dans des situations d'environnements réels non préparés.

10.1.1.2 Choix d'hypothèses faibles ou fortes

Nous définissons ici une dichotomie que nous réutiliserons dans la suite de notre discussion : hypothèses faibles et hypothèses fortes.

Nous appellerons modèle à hypothèse fortes un modèle qui se donne un cadre théorique précis auquel les données mesurées doivent se conformer, sous peine de rejet. Par opposition, un modèle à hypothèses faibles définit une procédure, appliquée aux données disponibles, et dont le fonctionnement est conçu pour ne pas être trop perturbé par de possibles données aberrantes. Un modèle à hypothèses faibles ne se donne pas les moyens de déterminer complètement si les mesures sont cohérentes. En contrepartie, il gagne en simplicité.

Ainsi, pour l'estimation de la direction à suivre, la méthode par élimination repose sur des hypothèses fortes (cohérence entre la théorie et toutes les mesures retenues), comme celle de Nelson et Aloimonos [NEL 88], par opposition aux méthodes réalisant simplement une approximation fondée sur une somme vectorielle (CC, PV, DV, ALV), à partir d'une procédure issue de considérations qualitatives.

Pour l'estimation de l'orientation, la méthode de Benosman et al. [GOU 03] repose sur des hypothèses fortes (cohérence entre la théorie et toutes les mesures retenues), par opposition à notre compas visuel. En effet, ce dernier est seulement fondé sur l'observation de parallaxes et sur une interpolation linéaire, ayant fait le choix de ne pas réaliser de reconstruction précise de l'espace autour de l'animat. Il ne repose pas non plus explicitement sur l'existence de points d'expansion et de contraction (comme celle de Nelson et Aloimonos).

Le choix d'hypothèses faibles ou fortes a des conséquences positives et négatives.

L'avantage des hypothèses fortes est qu'on peut obtenir des réponses plus précises. Ainsi la méthode par élimination indique un intervalle de directions possibles, garanti sous hypothèses, et peut détecter des anomalies, alors que les méthodes par somme vectorielle donnent une direction en principe à moins de 90° d'erreur, mais fausse en cas de mauvais appariement et sans possibilité de détecter des anomalies.

L'avantage des hypothèses faibles est dans la simplification qu'elle permet aux modèles. Des hypothèses faibles sont plus facilement réalisées et peuvent donc s'accommoder de situations qui seraient rejetées par les modèles à hypothèses fortes¹.

Illustrons-le par un exemple qui compare la méthode par élimination avec le compas visuel. La première suppose que les azimuts mesurés représentent des points précis de l'espace. Si on prend comme azimut les milieux des secteurs, c'est approximatif quoique acceptable si l'objet n'est pas trop proche. On peut prendre comme azimut les transitions en espérant une définition plus rigoureuse. L'azimut d'une transition verticale plane ou anguleuse (bord d'un poster, d'une porte) correspond en effet théoriquement à un point fixe, mais c'est faux pour le bord d'un obstacle de forme arrondie,

¹ Il y a un parallèle avec les méthodes évolutionnistes où l'on renonce à une analyse précise d'un problème pour laisser l'évolution trouver des solutions, toujours fondées sur des mécanismes plus faibles que ceux qu'on aurait mis en oeuvre suite à une analyse précise.

car le point exact faisant la transition dépend de la position de l'animat. Les hypothèses du compas visuel sont compatibles avec les deux cas, alors que les hypothèses plus « dures » de la méthode par élimination peuvent la mettre en difficulté dans le second cas.

Notons que cette notion est relative : on peut en général trouver des solutions telles que l'une fait des hypothèses plus forte qu'une seconde mais plus faibles qu'une troisième. Ainsi, la méthode de l'étoile polaire fait sur les mesures des hypothèses encore plus faibles que celles du compas visuel.

Nous pouvons maintenant entrer dans les détails de la discussion, détails spécifiques à chacun des modèles étudiés dans cette thèse.

10.1.2 Comparaison avec le modèle de Nelson et Aloimonos

Rappelons que le modèle de Nelson et Aloimonos [NEL 88] réalise une estimation de mouvement propre instantané (direction de translation, et rotation) à partir du flux optique supposé connu sur presque toute la sphère visuelle.

Sa réalisation pratique est difficile, notamment dans la construction et l'exploitation d'un capteur capable de fournir un tel flux optique. L'adaptation à un cas réel à deux dimensions nécessiterait l'introduction de capteurs et l'ajout d'une méthode d'appariement, qui le rapprocheraient des autres modèles étudiés ici.

Le problème résolu concerne le mouvement instantané et non le problème de retour au nid qui nous intéresse, même si nous avons déjà expliqué en section 4.2 la similitude entre les deux problèmes et pourquoi celui de retour au nid peut être plus difficile.

10.1.2.1 Rapprochement avec la méthode par élimination

La méthode de Nelson et Aloimonos, comme la méthode par élimination, s'appuie sur un modèle à hypothèses fortes. La méthode suppose la disponibilité de mesures de flux optique presque partout. Les étapes à deux dimensions qui interviennent après le découplage font explicitement intervenir une recherche des points de contraction et d'expansion, y compris dans le cas où leur recherche est compliquée par une rotation.

La méthode par élimination, de son côté, procède de façon similaire en cherchant la minimisation d'une fonction de coût dans le cas 2D. Toutefois elle est issue de considérations différentes. Elle est conçue dans un contexte de déplacement fini, où la rotation éventuelle est connue et déjà compensée et où seulement quelques mesures de parallaxes sont disponibles. Au-delà de ces différences de contexte, la méthode par élimination revient, elle aussi, à trouver le point de contraction, qui est la direction à suivre pour revenir au but.

Un problème important qui se pose à la méthode par élimination, et que le modèle de Nelson et Aloimonos élimine en s'appuyant sur le mouvement instantané, est celui d'une occlusion du point de contraction. La méthode par élimination, travaillant à chaque instant sur une comparaison entre

le panorama actuel et le panorama vu du but, peut se trouver dans le cas où un amer vient occulter le point de contraction, problème dont nous considérons qu'il doit être résolu par l'utilisation d'une carte.

10.1.2.2 Rapprochement avec le compas visuel : découplage

Un autre point de comparaison pertinente porte sur l'aspect découplage qui intervient pour réduire le nombre de dimensions et la complexité.

Le découplage utilisé par Aloimonos et expliqué en section 4.2.1 est exact : le mouvement propre 2D qu'on calcule à l'aide de la projection sur chacun des trois plans est exactement, on peut le démontrer, la projection du mouvement propre 3D sur ce plan. Grâce à ce découplage, on passe d'un problème à 5 dimensions à trois problèmes à 2 dimensions (une de rotation, une de translation, car le vecteur vitesse en 2D ne peut être estimé qu'à une homothétie près). Ce découplage est très intéressant car on passe d'un problème insoluble (espace de paramètres beaucoup trop grand) à trois problèmes solubles.

Le découplage introduit dans notre compas visuel se situe une étape plus loin : partant du cas à deux dimensions, il découple rotation et translation, pour estimer la rotation seule. Comme l'autre découplage, le nôtre réduit encore le coût computationnel puisqu'il permet d'estimer la rotation seule. Toutefois, à sa différence, notre découplage s'appuie sur une approximation, parce que l'invariant que nous avons introduit est fondé sur une approximation au premier ordre.

Une fois rapproché le fait qu'un découplage intervient dans les deux modèles, mais à des moments différents, l'estimation d'orientation par le compas visuel n'a rien d'autre de commun. Comme déjà énoncé en section 10.1.1.2, le compas visuel est un modèle de type « hypothèses faibles ».

10.1.2.3 Conclusion

La méthode de Nelson et Aloimonos résout de façon théorique avec un modèle à hypothèses fortes le problème de l'estimation du mouvement propre, translation et rotation. Sa réalisation pratique complète n'est pas facile. Une méthode de calcul du mouvement propre instantané ne résout pas certaines difficultés d'un problème de retour au nid.

La méthode par élimination présente des similitudes avec la méthode de Nelson et Aloimonos, dans la recherche de minimum d'une fonction calculée à partir des parallaxes. Le compas visuel a en commun de faire opérer un découplage entre problèmes, pour réduire la complexité calculatoire.

La méthode par élimination, d'une part, le compas visuel d'autre part, sont applicables à des situations réelles de retour au nid, et plus légers en calculs.

10.1.3 Complémentarité avec l'ALV

L'ALV résout le problème du retour au nid. Mais nous avons fait remarquer en section 2.4.4 qu'il ne peut pas constater, et encore moins remédier à, un problème d'orientation. Il n'y a pas de comparaison pertinente à faire entre l'ALV et la méthode par élimination, mais il y a paradoxalement des liens entre l'ALV et le compas visuel.

C'est l'ALV qui initialement nous a servi de source d'inspiration dans l'élaboration du compas visuel. En effet, l'ALV calcule une somme de vecteurs pointant dans les directions des amers. Dans nos premières recherches pour trouver un compas visuel, nous avons essayé de trouver un jeu de coefficients et de leur associer une somme vectorielle de trois vecteurs dirigés vers trois amers, dont les normes auraient été ces coefficients. Ces coefficients auraient été choisis de façon que la direction de ce vecteur soit constante au premier ordre. Ainsi l'animat, même désorienté, s'il avait été capable de reconnaître les amers, aurait pu retrouver son orientation car il aurait disposé d'un vecteur d'orientation constante au premier ordre.

Mais cette approche posait problème. Pour exprimer que l'orientation est constante on doit faire appel à des produits scalaires et donc introduire des non-linéarités (termes carrés) dans le système d'équations. Après investigation nous nous sommes tournés vers un calcul directement dans l'espace des angles, parce que les équations résultantes étaient linéaires et donc beaucoup plus simple à résoudre. Nous avons eu l'intuition que résoudre les équations non-linéaires revenait en fait à résoudre le problème complet du calcul de la distance aux objets, mais nous ne l'avons pas démontré.

En conclusion, l'ALV est un modèle très simple, dont l'intérêt essentiel est dans la compréhension des liens qu'il a avec le modèle DV, et utilisable pour des cas d'école. Il ne s'attaque pas au problème de l'orientation, supposé résolu par ailleurs. Sa proximité avec le compas visuel se limite à l'idée d'une combinaison linéaire de vecteurs pointant vers des amers, qui finalement est devenue combinaison linéaire de parallaxes.

10.1.4 Comparaison avec le modèle de Gaussier

La méthode utilisée par Gaussier et al. [GAU 00b] pour le retour au nid consiste essentiellement à associer (par apprentissage) à quelques vues prises autour du but la direction qu'il faut prendre pour le rejoindre. Pour rejoindre effectivement le but, l'animat estime à laquelle des vues mémorisée la vue actuelle ressemble, et se dirige dans la direction apprise.

10.1.4.1 Navigation

Nous passons brièvement sur la comparaison des aspects d'implémentation robotique. La différence principale se situe dans le fait que la définition des amers, une imagerie en niveaux de gris pour chaque amer, est beaucoup plus spécifique que la perception de la couleur. Le résultat est que la méthode d'appariement est plus simple : après calcul de la matrice de ressemblances, on crée les paires qui correspondent aux termes les plus importants de la matrice. Remarquons que le coût est le même

qu'avec la programmation dynamique : $O(n^2)$.

En ce qui concerne le calcul de la direction à suivre, la comparaison des modèles montre un cas de différence entre hypothèse faible et hypothèse forte. La plupart des algorithmes étudiés dans cette thèse font une hypothèse forte : pouvoir identifier des points de l'environnement et utiliser les parallaxes dues au mouvement. Dans le modèle de Gaussier et al., bien que des parallaxes soient utilisées pour estimer à quel panorama mémorisé le panorama actuel ressemble le plus, aucune parallaxe n'est utilisée pour estimer la direction à suivre, puisque cette dernière est directement associée à la vue reconnue. Le coeur de la méthode de navigation est dans l'association lieu-direction. La reconnaissance du lieu pourrait être effectuée d'une façon ne faisant pas intervenir de parallaxe. Du moment qu'une vue, même modifiée, peut être reconnue correctement, la méthode peut fonctionner, justifiant le classement comme méthode à hypothèses faibles.

Étant dans le contexte de l'utilisation de parallaxes pour la navigation, nous pensons que la méthode de Gaussier conduit à une exploration plus longue avant de pouvoir rejoindre un point (elle nécessite plusieurs vues prises tout autour du but) qu'avec l'utilisation directe des parallaxes. En outre, l'ensemble des directions que peut suivre le robot est quantifié : une par vue reconnue (typiquement 3 à 8 autour d'un but). Cela conduit souvent le robot à une trajectoire en zig-zag où il reconnaît alternativement deux vues différentes.

10.1.4.2 Orientation

Le modèle que nous venons de comparer ne traite pas le problème de l'orientation : il considère qu'elle est connue. Toutefois, par souci de complétude, nous devons signaler que des travaux plus récents de Gaussier et al. mentionnent la possible utilisation de leur architecture de reconnaissance de lieux et d'objets comme d'un « compas visuel » [LEP 00]. Toutefois, il ne s'agit pas d'un mécanisme de compas conçu pour être utilisable tout le long d'une trajectoire de retour au nid, mais seulement en un point. Et en effet, il n'est pas fait mention de son utilisation pour s'orienter le long d'une trajectoire.

L'architecture de reconnaissance d'objets de Gaussier et al, utilisée dans ce cas pour une orientation sur place, pourrait servir de point de départ à une implémentation alternative, biologiquement plus plausible, de la méthode de l'étoile polaire ou du compas visuel, en sélectionnant et reconnaissant les amers et leurs directions.

10.1.4.3 Conclusion

Nous pensons que notre animat est plus direct dans son exploitation des informations disponibles et donc dans sa trajectoire de retour au nid qu'avec un modèle comme celui de Gaussier.

L'architecture de Gaussier et al. s'attaque plus généralement à des problèmes qui dépassent notre cadre comme la reconnaissance d'objets isolés, ou la sélection de l'action. Il est normal que les problèmes communs avec nos travaux ne soient pas traités de la même manière.

10.1.5 Complémentarité avec le modèle de déformation de Franz

Par rapport au modèle de Franz [FRA 98b, FRA 97] exposé en section 4.5, notre modèle présente une certaine complémentarité. Les deux font une hypothèse qui repose sur une approximation. Cependant les deux hypothèses sont très différentes.

En ce qui concerne le calcul de la direction à suivre (la problématique de retour au nid), il n'y a pas de proximité directe avec notre méthode par élimination, sinon par l'intermédiaire de la méthode d'Aloimonos, dont le modèle de déformation peut être considéré comme une adaptation à la robotique à l'aide d'une approximation. Nous estimons que Franz a eu besoin de l'*hypothèse de distances égales* parce qu'il ne s'est pas donné une méthode d'appariement en amont. En revanche, Röfer [RöF 97b], comme nous, s'est donné une telle méthode et n'a pas eu besoin de cette hypothèse.

En ce qui concerne la problématique de connaître à chaque instant son orientation, le compas visuel n'a pas besoin de l'*hypothèse de distances égales*. Il utilise même précisément les écarts de distance (*via* les parallaxes). Il n'a pas besoin non plus de l'*hypothèse de répartition isotropique des amers*.

Le modèle de déformation et le compas visuel ont une certaine forme de complémentarité. Les deux s'appuient sur des hypothèses quasiment incompatibles entre elles : l'hypothèse que les amers sont tous à la même distance, pour le modèle de déformation, et l'hypothèse que l'amplitude différente des parallaxes des différents amers, précisément dues à leurs distances différentes, peut être utilisée pour estimer son orientation.

Le modèle de déformation passe par une sorte d'appariement global. Il estime à la fois la direction à suivre et l'orientation. Son coût computationnel est élevé, mais il a pu être allégé, en tenant compte de l'estimation du pas de temps précédent, pour fonctionner en temps réel. Mais le coût reste étroitement liée à la précision des panoramas mesurés. Franz utilisait des panoramas comportant une centaine de mesures, ce qui limite la précision à environ 3° tout en permettant à la version accélérée d'être assez rapide pour être utilisée en ligne.

Au contraire, le compas visuel ne traite pas tout à la fois. C'est une méthode essentiellement liée à la disponibilité d'un nombre éventuellement petit de parallaxes. Une façon de les obtenir peut, comme nous l'avons fait, passer par la segmentation d'un panorama comportant un grand nombre de données ; celle-ci se fait en $O(n)$. Le nombre d'éléments intervenant en aval, dans l'appariement et le compas, est beaucoup plus petit. Mais toute autre façon, éventuellement plus légère, d'obtenir des azimuts et des parallaxes permet d'utiliser le compas visuel. En outre, si beaucoup de parallaxes sont disponibles, le compas n'a pas besoin de les utiliser toutes pour produire une estimation de direction.

En résumé, le modèle de déformation est un modèle « en un bloc » qui fait face à un compromis entre, d'une part, la finesse des panoramas et la précision obtenue et, d'autre part, le coût computationnel. Le compas visuel n'a pas à faire face à ce compromis. Il peut fonctionner avec une bonne précision et s'avère donc favorisé lorsque peu de mesures de parallaxes sont disponibles.

	compas visuel	méthode calibrée
translation	calculée ailleurs	calculée
rotation	un angle	trois angles
amers repérés par	azimuts	azimuts et élévations
points d'intérêt	segmentation 1D simple	centre de disques noirs, ou coins
ensemble minimal d'amers	3 régions	7 points
critère de concordance	couleur (surtout la teinte)	voisinage 2D (luminance)
procédure d'appariement	programmation dynamique 1D	ré-échantillonnage multi-résolution et corrélation
agrégation	simple somme vectorielle	méthodes robustes de minimisation

TAB. 10.1: Comparaison du compas visuel avec la méthode de Benosman et al. La plupart des différences sont dues aux différences entre les contextes dans lesquels les deux méthodes ont été développées.

10.1.6 Comparaison avec la méthode de Benosman et al.

La comparaison entre les deux méthodes porte sur l'aspect orientation, bien que la méthode de Benosman permette une estimation complète du déplacement 3D entre deux vues.

Le tableau 10.1 présente les différences principales entre les deux méthodes telles qu'elles ont été testées.

Une partie de la simplicité qui caractérise notre compas visuel provient des hypothèses un peu plus restrictives sur l'ensemble des mouvements possibles. Les modèles bio-inspirés étudiés dans cette thèse, comme le compas visuel qui les prolonge, ne peuvent dans leur forme actuelle traiter que les mouvements plans. Cette restriction apporte aussi des simplifications dans le contexte calibré, comme le montrent Faugeras et al. [FAU 98] avec un modèle utilisant une caméra classique (non panoramique).

Le choix de ne s'appuyer que sur les azimuts donne au compas visuel l'avantage d'être indépendant de la forme exacte du miroir, l'essentiel étant que des azimuts soient mesurables et que le capteur perçoive une certaine frange au-dessus et au-dessous de l'horizon, sinon les amers ne resteraient pas visibles sur des distances assez grandes. L'étape de calibration nécessaire dans le modèle calibré est inutile pour le compas visuel.

Le compas visuel a besoin au strict minimum de 3 amers individualisés visibles depuis 3 points de vues différents, alors que la méthode calibrée nécessite au strict minimum 7 points communs. Dans les deux méthodes, plus grand est le nombre d'amers visibles, meilleure est la précision de l'estimation. Cette différence est en relation directe avec le nombre de paramètres estimés.

Malgré leur différence de complexité, du point de vue conceptuel comme computationnel, les

deux méthodes semblent produire des résultats raisonnables pour leur contexte. La méthode calibrée a bénéficié d'une précision d'un ordre de grandeur meilleure que le compas visuel. Cela montre que les techniques utilisées par la méthode calibrée justifient le coût computationnel qu'elles occasionnent. Nous estimons que la robotique mobile peut se satisfaire de la précision obtenue par la méthode légère (compas visuel).

À cause de son hypothèse de rotation autour d'un axe vertical, la méthode bio-inspirée se rend vulnérable aux situations où le mouvement ne rentre pas dans ce cadre. Elle échouera dans le cas d'un mouvement quelconque. En revanche, la méthode calibrée bénéficiera de l'analyse plus approfondie qu'elle effectue, et devrait pouvoir traiter n'importe quelle rotation. Dans les deux contextes, la gravité peut permettre une simplification du problème si le corps de l'animat est arrangé pour en tirer bénéfice. Par exemple, un système mécanique peut exploiter la gravité pour stabiliser spontanément la tête de l'animat en position telle que le capteur panoramique soit correctement aligné par rapport à la verticale.

En conclusion, la méthode calibrée est préférable pour les situations de mouvement 3D quelconque. Elle bénéficie d'une meilleure précision, au prix de calculs plus lourds. Le compas visuel est probablement utilisable en dehors du contexte précis de retour au nid à l'aide de parallaxes, mais ne peut traiter des mouvements autres que plans sans être repensé complètement.

10.1.7 Comparaison avec l'algorithme Panama de Röfer

L'algorithme Panama, à l'instar du modèle de déformation, estime à la fois la direction à suivre pour le retour au nid et l'orientation de l'animat. Le modèle Panama se prête à une comparaison assez détaillée avec le nôtre, aussi bien sur le retour au nid et sur l'orientation que sur les aspects en amont. Nous présentons cette comparaison avec la grille d'analyse que nous avons utilisée dans la revue.

10.1.7.1 Signature d'un lieu

Nous avons expliqué en section 5.1.1 que, pour notre application, le codage de la couleur en luminosité, saturation et teinte présentait des avantages. Notamment, la teinte est assez indépendante des ombres sur un objet donné pour pouvoir la considérer comme constante et faciliter la segmentation. Röfer utilise la couleur aussi [RöF 97a], mais en composantes rouge, verte et bleue. Compte tenu de la méthode d'appariement qu'il utilise sans segmentation, ce choix est raisonnable. En effet, sa méthode d'appariement fonctionne en appariant de nombreux petits points de contraste locaux et non des objets segmentés. Il n'y gagnerait donc probablement pas en principe à se donner des variables faisant apparaître l'uniformité de la couleur des objets au-delà des variations locales d'éclairage.

La signature qu'utilise Röfer n'est pas segmentée. Elle représente donc une information d'un peu plus bas niveau que celle que nous utilisons, et il est alors tout à fait logique qu'elle représente un plus grand volume de données.

10.1.7.2 Appariement

La méthode d'appariement de Röfer répond aux trois premiers points du cahier des charges établi en section 5.2.2 : tenir compte de l'ensemble, tenir compte des différences entre éléments à apparier, fonctionner avec des panoramas d'orientations arbitraires.

10.1.7.2.1 Cas des panoramas partiels Röfer n'a pas explicité le cas du panorama partiel et il n'est pas dit que son modèle puisse fournir un appariement convenable. En effet sa méthode apparie forcément chaque point avec un autre, le meilleur pris dans un voisinage (cf. section 4.6.2).

Notons que Röfer cite l'existence de cas où certains secteurs d'une vue n'ont pas d'équivalent dans l'autre vue, notamment à cause d'occlusions entre objets. Il rapporte une bonne précision dans le cas d'une expérience simulée mais ne montre pas les appariements obtenus. Nous pensons qu'il est important qu'une méthode puisse s'abstenir d'apparier certains secteurs, pour être plus proche de la réalité et éviter de perturber les étapes qui utiliseront l'appariement. Or la méthode de Röfer ne permet pas cela.

Illustrons ce point en considérant la situation (section 9.2) de notre expérience robotique avec la caméra mécanique couvrant un champ de vision de 200° (et donc absence d'information sur le côté occulté). On se replace dans le contexte de la section 4.6.2 expliquant la méthode d'appariement de Panama. Si on représente l'absence d'information par une valeur arbitraire comme du noir, l'appariement résultant inclura les secteurs indiquant l'absence de donnée sous forme de points normalement appariés. Alternativement, on peut modéliser l'absence d'information comme un point à ne pas apparier, dans J , et comme un point très dissimilaire à tout autre, dans I . Dans ce second cas, on s'abstiendra d'apparier les secteurs occultés. Mais, dans les deux cas, la méthode appariera tous les points des deux champs de vision, en tassant à la limite entre côté vu et côté occulté les points qui n'auraient pas dû être appariés, à cause de la double contrainte de conserver l'ordre des points et de tout apparier.

10.1.7.2.2 Pré-orientation On remarque un parallèle entre la dernière étape d'appariement de Röfer et la pré-orientation du panorama avant l'étape d'exploitation du compas. Dans les deux cas, il s'agit d'éviter les effets de bouclage dus à la définition des angles modulo 2π .

Il n'est donc pas étonnant d'y retrouver des propriétés similaires : dans un grand nombre de cas (robot ayant tourné d'un angle faible par exemple), cette étape ne change rien car les valeurs données par l'étape précédente ne souffrent pas d'effet de bouclage.

10.1.7.2.3 Coût computationnel Notre méthode d'appariement fournit un relativement petit nombre d'éléments de flux optique, attachés à des objets précis, alors que la méthode de Röfer en fournit un nombre constant, certains attachés à des appariements précis et d'autres interpolés, sans distinction.

Du point de vue de la complexité calculatoire, ces deux méthodes semblent de la même classe

puisque Röfer rapporte $O(t_{max}n)$ avec des valeurs typiques de $n = 256$, le nombre de pixels d'une image et $t_{max} = 4096$, le nombre de pas d'adaptation. On peut raisonnablement considérer que le nombre de pas d'adaptation est en général proportionnel au nombre de pixels dans l'image, soit $t_{max} \simeq 16n$. En comparaison, notre méthode (cf. section 5.2.6) est en $O(mn)$, m et n étant le nombre d'amers dans les deux panoramas obtenus par la segmentation. m et n sont bien plus petits que le nombre de pixels, les valeurs dépassant rarement quelques dizaines, et n'ayant jamais atteint 100 dans nos expériences. Une estimation chiffrée très grossière suggère que notre méthode d'appariement est $\frac{256 \cdot 4096}{64 \cdot 64} = 256$ fois plus rapide que la méthode de Röfer, essentiellement grâce à la segmentation en amont.

On peut résumer en affirmant que notre méthode, qui bénéficie d'une signature d'un peu plus haut niveau parce que segmentée, demande moins de calculs, leur nombre exact étant dépendant de la complexité de l'environnement, alors que chez Röfer le temps est constant. Toutefois, nous confondons dans notre analyse le temps consommé par la méthode globale et celui de l'appariement, car c'est cette étape qui est la plus consommatrice, les autres étant presque toutes linéaires en fonction du nombre d'éléments, ou en tout cas plus rapide qu'en $O(n^2)$.

10.1.7.3 Calcul d'orientation

La méthode de Röfer semble intéressante et assez robuste. Cependant, elle est adaptée au cas où l'appariement obtenu est dense et régulier. L'étape du tri des valeurs pour obtenir l'orientation actuelle (section 4.6.3) – une simplification de la méthode d'Aloimonos – est clairement dépendante d'une répartition uniforme des appariements.

Dans le cas contraire, on peut adapter la méthode de tri des valeurs en reformulant le paragraphe correspondant de la section 4.6.3 comme suit.

Il y a dans le flux translationnel un demi-panorama de valeurs positives (pas forcément au nombre de $\frac{n}{2}$) et l'autre demi-panorama de valeurs négatives (pas forcément au nombre de $\frac{n}{2}$). La méthode consiste donc à trier les écarts pour y trouver un demi-panorama des plus grands et un demi-panorama des plus petits. Alors, il suffit de retrancher une valeur séparant ces deux ensembles pour retrouver un flux optique translationnel.

Le paragraphe ci-dessus spécifie clairement ce qu'il faut faire. Remarquons seulement que cela est un peu plus difficile à implémenter par le fait qu'il est moins facile de trouver un demi-panorama quand on ne sait pas à l'avance combien de valeurs il contiendra. L'astuce que Röfer avait introduite consistant à trier les valeurs ne fonctionne plus. Il faut parcourir l'ensemble des valeurs, un peu comme dans la solution 3 pour le calcul de la direction à suivre.

Il semble alors que notre méthode de calcul de l'orientation soit plus intéressante que celle de Röfer dans le cas d'un environnement où des amers sont bien identifiables, même si peu nombreux.

10.1.7.4 Liens entre Röfer, PV et la méthode par élimination

On peut remarquer une ressemblance entre le calcul de direction par élimination et les solutions utilisées par Röfer pour le calcul de direction (section 4.6.4).

Nous focalisons notre comparaison sur la solution 1 que Röfer a finalement retenue comme plus performante et nous montrons que la méthode par élimination est plus générale (pas d'hypothèse de densité et régularité du flux optique).

La solution 1 que Röfer a retenue et que nous avons résumée en section 4.6.4 ressemble à notre méthode par élimination. Les méthodes ont en commun de déduire de chaque information de parallaxe disponible une information sur la direction à prendre, en fonction du signe de la parallaxe. Rappelons que, chez Röfer, les informations de parallaxe sont des éléments de flux optique denses et régulièrement répartis alors que, dans notre modèle, il s'agit des parallaxes à raison d'une par secteur du panorama segmenté.

La méthode par élimination est qualitativement plus avancée que la solution de Röfer. En effet, cette dernière réalise une somme vectorielle, comme le modèle PV. Ce qui lui permet d'obtenir le bon résultat en théorie, c'est la régularité du flux optique, comme expliqué en section 4.6.4. Quand à la précision, elle est due à la densité du flux disponible. En revanche, parce que la méthode par élimination tient compte de façon plus fine de la signification d'une information de parallaxe, elle n'a pas besoin d'un ensemble d'informations de parallaxes réparties régulièrement. La méthode par élimination est donc strictement plus générale car elle est applicable à tous les panoramas pour lesquels la solution de Röfer est applicable. Remarquons que la solution 1 de Röfer appliquée à un panorama segmenté revient exactement au modèle PV, méthode directe.

On peut maintenant comparer les précisions des deux méthodes. Pour le cas prévu par la solution 1 d'un flux optique dense et régulier, la précision obtenue par les deux méthodes sera la même dans les conditions nominales. Si le flux optique contient des erreurs, il ne paraît pas facile de déterminer *a priori* comment l'une et l'autre des méthodes vont se comporter. Dans le cas de panoramas segmentés, PV a déjà été comparé à la méthode par élimination, en faveur de cette dernière pour ce qui est des mesures pragmatiques.

10.1.7.5 Conclusion de la comparaison avec le modèle Panama

Notre système considère comme distinctes les deux étapes de détection des éléments visibles de l'environnement (associée à la partie « signature d'un lieu ») et d'appariement de ces éléments. Nous nous sommes doté d'une façon simple et pragmatique de déterminer un petit nombre d'amers individualisés. En résumé, notre méthode est plus simple et analytique, alors que le modèle de Röfer est plus lourde et synthétique.

Ce choix initial d'une signature segmentée ou non détermine assez fortement la suite. Comme ce choix conditionne l'applicabilité des méthodes utilisées en aval, on ne peut faire que des parallèles entre les deux méthodes. Par exemple, on ne peut pas immédiatement appliquer la méthode d'appa-

riement de l'une sur la signature de l'autre avec l'un ou l'autre choix de calcul d'orientation et de direction.

La méthode de Röfer peut prétendre à une meilleure précision que la nôtre, lorsque les hypothèses sur lesquelles elle s'appuie sont bien vérifiées.

La chaîne d'étapes que nous avons élaborée pour la navigation semble plus adaptée au cas où un petit nombre d'objets est vraiment identifiable ; ce qui n'est pas étonnant car c'est dans cet esprit qu'elle a été conçue.

10.2 Bilan

Nous avons présenté au cours de cette thèse un ensemble d'améliorations apportées à des modèles de navigation à l'aide de panoramas.

Nous avons présenté une première série de modèles s'attaquant à la navigation locale, principalement : les modèles CC de Cartwright et Collett, le modèle de Hong, les variantes PV, DV et ALV de Möller et al., les variations sur l'appariement de Weber et al. Tous ces modèles supposent que l'animat connaît son orientation.

Nous avons analysé en quoi les modèles qui dépendent de l'orientation en dépendent vraiment pour certaines étapes, mais pas pour toutes.

Ainsi, les méthodes d'appariement utilisées par les modèles CC et leurs successeurs directs, ainsi que les variantes étudiées par Weber et al., sont très dépendantes d'une bonne orientation. Mais nous avons montré que des méthodes d'appariement indépendantes étaient possibles comme celle de Röfer ou comme celle que nous proposons par programmation dynamique.

Nous avons fait remarquer que la composante radiale, dans la variante « méthode directe », ne dépendait pas de l'orientation actuelle de l'animat, en regrettant toutefois qu'elle ne puisse donner d'indication toujours précise et menant directement au but.

Nous avons alors présenté et étudié une seconde série de modèles qui s'attaquent à la navigation locale par panoramas dans le cas où l'animat ne sait pas *a priori* son orientation par rapport aux panoramas mémorisés. Il s'agit principalement des modèles de Nelson et al., de Franz et de Röfer.

Puis nous avons présenté notre modèle :

- Nous avons utilisé des panoramas segmentés en couleur ;
- Nous avons présenté une méthode d'appariement par programmation dynamique, pouvant gérer des orientations arbitraires des deux panoramas, de complexité calculatoire moindre que celle d'Einsele, similaire à celle de Röfer, mais plus rapide en pratique grâce à la légèreté des panoramas segmentés ;
- Nous avons présenté une méthode originale pour estimer la direction à suivre pour rejoindre le panorama mémorisé, dite « méthode par élimination ». Nous avons montré pourquoi elle tenait plus précisément compte des informations que les méthodes précédentes et l'avons illustré sur

des exemples ;

- Nous avons aussi ajouté quelques contributions à des points semblant secondaires mais techniquement utiles pour une implémentation robotique efficace, comme la longueur des pas à effectuer et le critère d'arrivée.

Ensuite, nous avons présenté le compas visuel, une méthode originale, à hypothèses de type faible, pour permettre à un animat d'estimer son orientation dans une région de son environnement où il a pu au préalable prendre des vues correctement orientées. Cette méthode a les caractéristiques suivantes :

- nécessite trois vues correctement orientées et appariées, faisant figurer au minimum trois amers communs à toutes les vues, pour une étape d'observation qui consiste essentiellement à construire des invariants découplant translation et rotation ;
- permet d'estimer à l'avenir l'orientation de tout panorama situé dans un voisinage des points d'où ont été prises les vues d'observation ;
- est computationnellement légère, proportionnelle au nombre d'invariants exploitable en situation (l'étape la plus lourde est l'appariement, qui est de toute façon nécessaire à la navigation).

Nous avons comparé notre modèle aux autres modèles directement pertinents.

Nous retenons de façon générale que l'appariement est souvent l'étape la plus gourmande en calcul. Cependant il est plus léger avec des panoramas segmentés.

Globalement, notre méthode pour estimer orientation et direction à suivre est légère : $O(\text{nombre de marques})$ pour estimer l'orientation, puis $O(\text{nombre d'amers appariés})$ pour trouver la direction à suivre avec la méthode par élimination.

Nous retenons aussi que, pour pouvoir naviguer en orientation quelconque, il est essentiel de disposer d'une méthode d'appariement qui fonctionne correctement quelle que soit l'orientation relative des vues, comme la nôtre fondée sur la programmation dynamique (nous considérons que le modèle de déformation de Franz rentre dans ce cadre en réalisant un appariement global).

10.3 Perspectives

Notre travail n'est pas, espérons-le, la dernière brique sur un édifice achevé. Des voies de recherches futures nous sont apparues intéressantes. Le paragraphe suivant donne un très bref résumé des points qui sont traités dans les sous-sections qui vont suivre.

Tout d'abord, des points internes aux modèles que nous avons développés peuvent être améliorés, par exemple dans le compas visuel. Un premier cas est l'étape d'agrégation des directions estimées par les différentes marques. Un autre s'appuierait sur le fait de tirer parti d'une certaine complémentarité entre compas visuel et méthode par élimination. Ensuite, des points à la périphérie de notre travail peuvent être renforcés et étendus, comme l'algorithme de construction de carte qui, en l'état actuel, est essentiellement là pour soutenir le compas visuel. Par ailleurs, nos travaux étant inspirés de recherches biologiques, un retour vers la biologie serait potentiellement intéressant. Enfin, dans le cadre du projet Psikharpax, notre thèse pourra compléter d'autres travaux de recherche menés à

l'AnimatLab. En particulier, de façon ponctuelle et pour le moment anecdotique, la généralisation du modèle de navigation et du compas visuel au cas à trois dimensions serait intéressante pour un animat volant.

10.3.1 Estimation plus fine de la direction

Lors de la présentation du compas visuel, nous avons évoqué la faiblesse de la méthode d'agrégation utilisée. En effet, au cours de l'estimation de direction par le compas visuel, chaque marque disponible vote pour une direction. Aucune de ces marques en général ne donne la direction exacte, parce que chaque estimation est calculée en s'appuyant sur un invariant dont nous avons montré qu'il est fondé sur une approximation linéaire du premier ordre.

Une limitation du compas visuel est donc qu'avec la méthode d'agrégation actuelle il peut parfois donner une indication franchement fautive, en cas de mauvais appariement par exemple. En effet, ce dernier cas invalide les garanties que nous avons sur la précision suffisante de son estimation. Certes, le nombre de marques disponibles par rapport au nombre total de marques donne une indication, mais ne permet pas de répondre à la question : « la direction calculée est-elle crédible ? ».

Pour améliorer ce point, il faudrait chercher parmi les méthodes mathématiques d'estimation de variables, notamment les algorithmes de regroupement². Les outils de filtrage classiques (filtrage de Wiener, de Kalman) sont conçus pour filtrer un bruit aléatoire, ce qui ne s'applique pas à notre cas. Un outil de regroupement bien choisi et paramétré pourrait tenir compte au mieux des marques pertinentes pour une situation, tout en ignorant sélectivement les marques qui dévient trop par rapport au groupe. Pour une revue de méthodes d'estimation de paramètres, on peut se référer à [ZHA 97]. Idéalement, on pourrait aussi déterminer ainsi les amers qui donnent peu ou pas de bonnes estimations et utiliser ces informations comme heuristiques pour influencer en amont la perception et la sélection des amers.

Nous pensons qu'une complémentarité est possible entre la capacité de la « méthode par élimination » à détecter les incohérences, et la capacité du compas à proposer des directions. Plus précisément, nous avons imaginé comment combiner ces deux derniers aspects pour éliminer les éléments incohérents et ainsi sélectionner les plus robustes. L'idée consiste à passer en revue chacune des marques disponibles. Pour chaque marque, on applique l'étape d'exploitation du compas qui produit une estimation d'orientation. On applique momentanément cette estimation d'orientation à la vue actuelle. La vue orientée résultante est envoyée à l'algorithme de calcul de direction à suivre par élimination. Si ce dernier a éliminé toutes les directions, c'est probablement que la direction trouvée n'est pas acceptable et on pénalise donc la marque en question. De façon similaire à la méthode d'élimination des paires incohérentes (section 6.4.3.2), cette méthode n'a besoin d'aucun paramètre car elle se fonde uniquement sur une vérification de cohérence. Nous estimons qu'une telle méthode a des chances d'éliminer les marques peu fiables, soit parce que la position relative des amers est proche d'un cas défavorable (section 6.4.2.1), soit parce que les amers intervenant dans cette marques ne sont pas bien segmentés ou appariés, de sorte que la mesure de leur azimut n'est pas fiable. Nous pensons qu'il est

²Nous traduisons par « regroupement » le mot anglais « clustering ».

possible ainsi d'améliorer la précision du compas.

10.3.2 Extensions de l'usage de la carte

La méthode de construction de carte que nous avons développée est conçue pour être suffisante pour fournir des vues d'observation au compas visuel. Mais une carte cognitive peut être un support possible pour d'autres problématiques de recherche. Un premier domaine est la planification des déplacements, où l'on peut utiliser la connaissance de l'environnement que représente une carte cognitive pour rejoindre un but unique le plus efficacement possible. Par exemple, de deux chemins pour rejoindre un même point, l'animat pourra déterminer le plus court ou le plus sûr à l'aide de la carte. Un autre domaine est la sélection de l'action, où l'on peut utiliser la carte cognitive pour déterminer la meilleure action à effectuer. Par exemple, si notre animat a faim il tentera de rejoindre une source de nourriture. S'il a faim et un peu soif, de deux chemins qui mènent à une source de nourriture, il pourra décider de choisir celui qui passe à proximité d'une source de boisson pour maximiser le bénéfice. Dans ce contexte il serait intéressant d'utiliser des algorithmes plus élaborés de construction et de maintenance de carte, qui puissent gérer efficacement des buts multiples et annoter les lieux visités en fonction de leur intérêt vis-à-vis des motivations de l'animat.

10.3.3 Alternatives à la segmentation de panoramas en couleurs

Nous avons développé nos modèles et conduit nos simulations, comme nos expériences de robotique réelle, sur l'extraction d'un panorama de couleurs que nous avons segmenté. Nous avons fait ce choix car c'était le prolongement le plus naturel, dans notre contexte, des modèles utilisant des panoramas en noir et blanc et car cela convenait bien à des expérimentations dans l'environnement – certes artificiel – du laboratoire, où des objets plus ou moins colorés se détachaient des murs presque blancs.

Les expériences telles que nous les avons menées ne fonctionneraient pas aussi bien dans d'autres environnements : rue, campagne, forêt, environnement aérien ou aquatique, désert ou autre planète. Pourtant, après l'étape d'appariement, les modèles d'estimation de direction à suivre et d'orientation ne s'appuient que sur des azimuts et leurs variations, n'ayant plus besoin de faire référence à la nature ni aux propriétés des objets concernés.

Il serait intéressant de tester la méthode par élimination et le compas visuel en aval d'autres systèmes de perception de l'environnement, eux aussi capables de percevoir, reconnaître, et mesurer des azimuts des objets environnants.

Pour des applications particulières, on peut penser à des systèmes artificiels (balises colorées, ou infrarouges ou radio en utilisant d'autres systèmes de mesures) qui facilitent le bon fonctionnement mais nécessitent de préparer l'environnement.

Si l'on ne souhaite pas préparer l'environnement, on peut enrichir la perception par des algorithmes de reconnaissance de forme, ou d'autres algorithmes extrayant une signature caractéristique

d'objets environnants, sans forcément les reconnaître comme nous le faisons. Ainsi, en intérieur, l'animat pourrait différencier une porte bleue d'une chaise de la même couleur et repérer des objets qui ne se distinguent pas nettement par leur couleur mais qui sont très caractéristiques, comme une prise de courant. Dans cette optique, l'appariement par programmation dynamique pourrait être conservé mais en incorporant les signatures des objets. En environnement urbain, des éléments pertinents comme fenêtres, feux de signalisation, arbres et panneaux pourraient être capturés puis reconnus par des systèmes génériques extrayant des signatures. Pour les autres environnements, il est difficile de faire un diagnostic *a priori*, l'essentiel étant qu'il y ait une adéquation entre les éléments saillants du paysage et les capacités perceptives de l'animat.

10.3.4 Retour vers la biologie

Les modèles que nous avons étudiés dans cette thèse se sont appuyés sur des recherches biologiques. À l'origine, celles de Cartwright & Collett sur les abeilles, mais aussi celles de Möller et al. sur les fourmis du désert *Cataglyphis fortis* par exemple.

Dans cette thèse, nous nous sommes intéressés à l'obtention de méthodes efficaces pour un robot, et non à la modélisation fidèle de mécanismes biologiques. Pourtant, il serait intéressant d'investiguer des parallèles possibles entre la façon dont le compas visuel exploite les pseudo-parallaxes de quelques amers pour estimer la direction actuelle, et d'éventuels mécanismes similaires chez des insectes ou des vertébrés.

10.3.5 Le projet Psikharpax

Le projet Psikharpax de construction d'un rat artificiel [GUI 02], initié par l'AnimatLab, a vocation à intégrer dans une même réalisation des travaux de recherches sur différents sujets, dont la thèse de Filliat [FIL 01] et celle de Girard [GIR 03] sur la sélection de l'action. Notre thèse peut compléter la thèse de Filliat à deux niveaux.

Le modèle de navigation de Filliat est capable de réaliser une carte topologique dense de son environnement en surmontant le problème de l'ambiguïté perceptive³. Son modèle paraît très prometteur. Cependant, une limitation importante qui limite sa viabilité est l'absence de système de recalage de la direction. Le robot n'ayant pas mieux que l'odométrie, le modèle finit par échouer si aucun humain n'intervient pour recalibrer la direction.

Les perspectives de la thèse de Filliat invitent à ajouter un moyen de maintenir une estimation de direction. Le compas visuel pourrait fournir cette indication. Un travail d'intégration serait toutefois nécessaire, pour que les desiderata du compas – en matière de vues convenables pour des étapes d'observation – puissent guider le robot lors de son exploration de l'environnement.

Un autre aspect cité dans la thèse de Filliat est le fait que la carte topologique est une représentation de granularité relativement élevée : un nœud de la carte représente une zone relativement

³Nous traduisons par « ambiguïté perceptive » l'expression « perceptual aliasing ».

floue, sans possibilité de rejoindre un point précis. La navigation locale par panorama peut réaliser, si l'environnement fournit des amers convenables pour des mesures d'angles, la capacité à rejoindre un point plus précis que ne le fait la carte topologique de Filliat. Il suffirait, pour l'animat, de prendre un panorama lors du premier passage en un point intéressant pour être en mesure de le rejoindre depuis le voisinage. Depuis un point quelconque de l'environnement, le modèle de Filliat ramènerait l'animat au voisinage du point visé, et la navigation locale prendrait le relais pour affiner les derniers mouvements.

Il sera intéressant aussi de connecter le compas visuel au modèle de Degris [DEG 04a, DEG 04b] qui réalise une intégration biologiquement plausible au cours du temps de la rotation instantanée d'un animat. La connexion avec ce modèle permet de fusionner à chaque instant la direction intégrée avec l'information en provenance du compas, au lieu d'avoir recours à une intégration mathématique biologiquement moins plausible. La comparaison de la performance d'un tel système avec celle des cellules de direction de la tête observée chez les rats permettra de mieux comprendre les mécanismes à l'oeuvre chez ce dernier.

Chapitre 11

Conclusion

Nous avons développé et présenté un système de navigation pour un robot autonome capable de réaliser des tâches de navigation locale et d'explorer son environnement pour pouvoir en permanence maintenir une estimation de sa direction. Nous nous sommes appuyé essentiellement sur des mesures d'angles dans des vues panoramiques prises par l'animat, sans chercher à nous donner accès à des informations de distance ni à un plan des lieux connu *a priori*. Le modèle exploite des comparaisons d'azimuts entre le panorama actuel et le panorama mémorisé au but pour, d'une part, connaître l'orientation actuelle et, d'autre part, estimer la direction à suivre pour rejoindre le but.

Notre méthode d'estimation de l'orientation, que nous avons appelée le compas visuel, est computationnellement très légère. À partir des azimuts de trois amers dans trois panoramas dont les orientations – mais pas les localisations – sont connues, un invariant est construit qui découple translation et rotation de l'animat, *via* une approximation au premier ordre. La connaissance de l'invariant donne l'orientation de toute vue prise dans le voisinage des points utilisés.

De son côté, la méthode d'estimation de la direction à suivre s'appuie sur les informations de parallaxe, et fournit une réponse plus précise sur la direction du but que les méthodes auxquelles elle est apparentée. Pour que l'animat se donne de façon autonome des panoramas convenables pour l'estimation d'orientation, nous avons développé une méthode de construction de carte.

Nous avons fait le choix, pour mesurer des azimuts dans l'environnement, de segmenter des panoramas par secteurs de couleur. Nous avons expliqué comment ce choix, tout en nous ayant permis de valider nos modèles, est un facteur limitant leur utilisation dans des environnements variés. Ce problème serait résolu en se donnant d'autres critères de perception et de reconnaissance des amers environnants.

Enfin, les fonctionnalités que notre modèle présente ont vocation à compléter celles de divers modèles en cours de développement à l'AnimatLab, dans le cadre du projet Psikharpax qui vise à la mise au point d'un rat artificiel.

Annexe A

Généalogie et classification des modèles

A.1 Généalogie

La figure [A.1](#) présente une vue synthétique de type généalogique représentant les relations entre les modèles étudiés dans cette thèse. L'intérêt principal de ce schéma est de visualiser les relations « historiques » entre les modèles.

Les différents types de liens représentent les différentes classes de nouveautés apportées par les modèles successifs : nouveauté à propos de la signature des lieux, de l'appariement, de la direction à suivre, ou autre type de nouveauté.

Nous appelons « famille CC » les modèles CC1 à CC5, et « modèles descendants de CC » les autres modèles représentés sur ce schéma.

A.2 Classification

Le schéma [A.2](#) donne une vue d'ensemble analytique des modèles étudiés dans cette thèse.

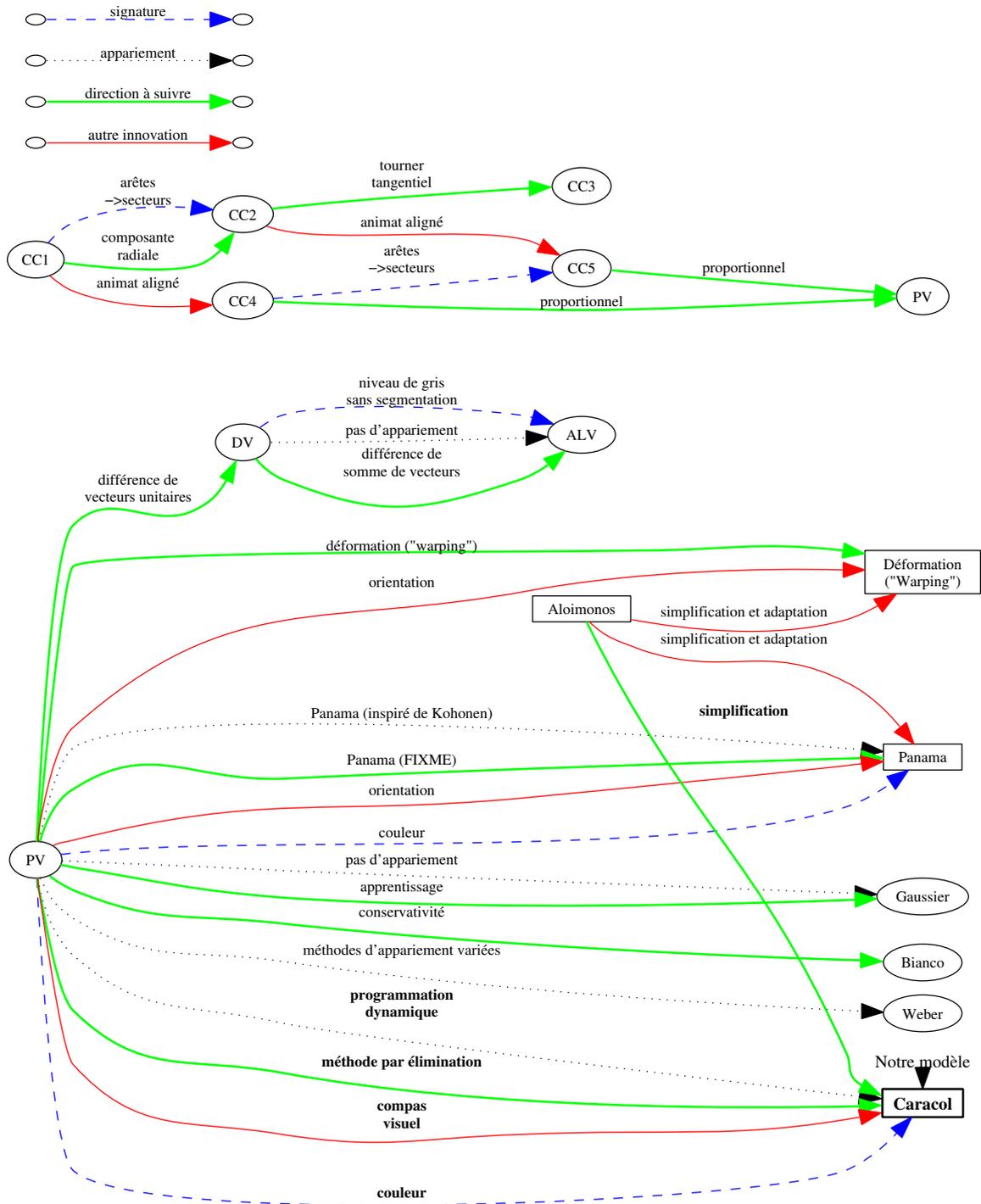


FIG. A.1: Généalogie des modèles issus en ligne directe des travaux de Cartwright et Collett, voir texte. Par exemple, on peut lire sur le schéma que le modèle par déformation descend du modèle PV, que son innovation principale porte sur le calcul de la direction à suivre, et qu'il fournit à l'animat l'estimation de son orientation.

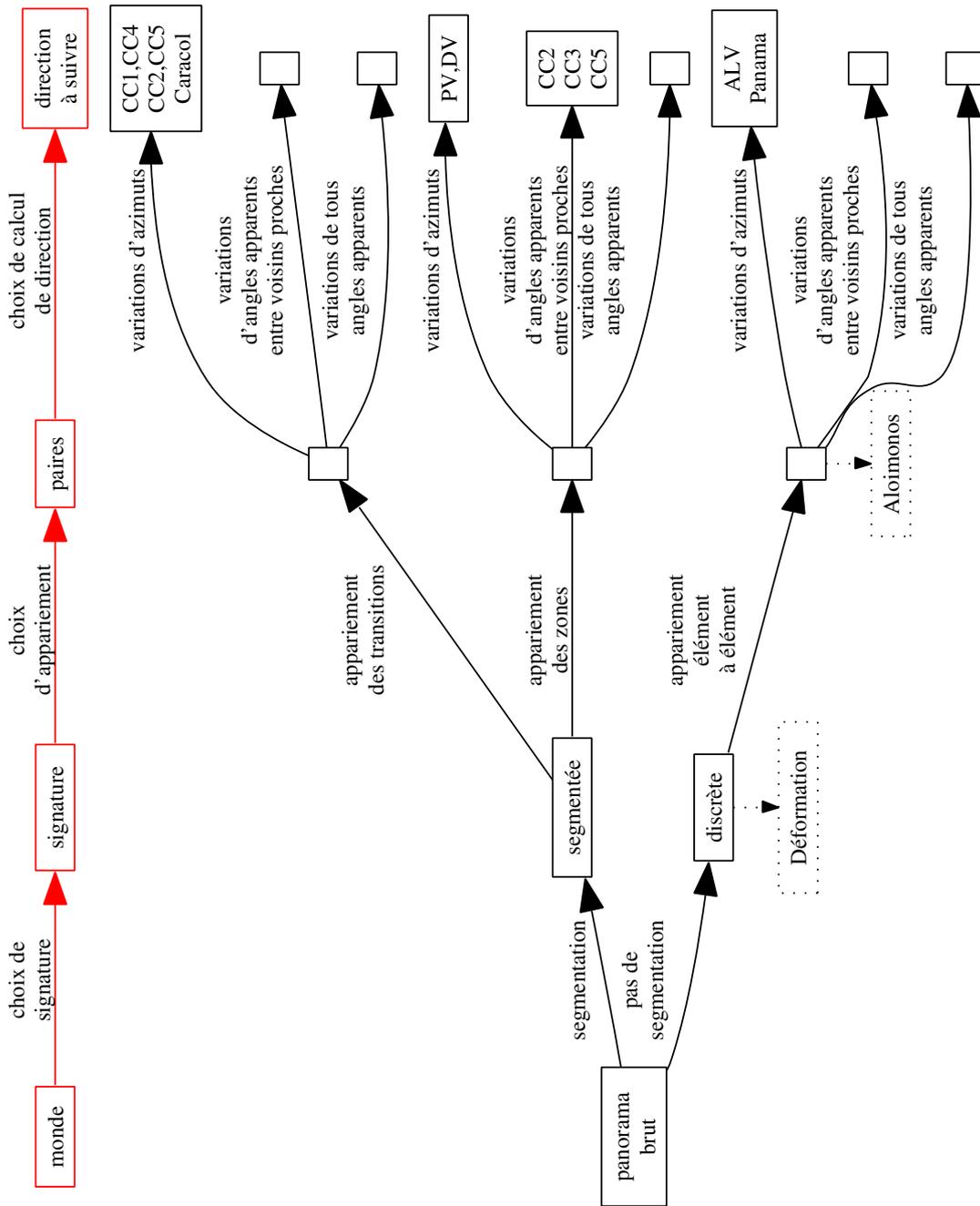


FIG. A.2: Schéma résumant les choix possibles du type de signature, du type des objets sujets à appariement, du type d'information angulaire utilisée pour calculer le mouvement à faire. Certaines combinaisons ont été explorées par les modèles précédents, mais pas toutes, ce qui explique les cases vides. Les noms des modèles sont alors indiqués dans la case correspondante. Cette représentation n'est toutefois pas complètement fidèle car souvent, le modèle indiqué a été conçu et testé dans les conditions indiquées, mais sa contribution essentielle est ailleurs et il pourrait potentiellement être testé dans plusieurs cadres. Par exemple, le modèle PV apporte essentiellement la proportionnalité des vecteurs individuels aux causes qui leur donnent naissance, et ce principe est applicable dans tous les cas.

Bibliographie

- [ACT 99] ACTIVMEDIA. *Pioneer 2 Mobile Robot. Saphira Manual*, 1999. [202](#)
- [BAB 00] V. BABEAU, P. GAUSSIER, C. JOULAIN, A. REVEL ET J.P. BANQUET. Merging visual place recognition and path integration for "cognitive" map learning. In J.A. MEYER, A. BERTHOZ, D. FLOREANO, H.L. ROIBLAT ET S.W. WILSON, Eds, *Sixth International Conference on simulation of adaptive behavior : From Animals to Animats. SAB2000. Proceedings Supplement.*, volume 6. MIT Press, 2000. [51](#)
- [BEA 95] S. S. BEAUCHEMIN ET J. L. BARRON. The computation of optical flow. *ACM Computing Surveys*, 27(3) :433–467, 1995. [88](#)
- [BEL 57] R. E. BELLMAN. *Dynamic Programming*. Princeton University Press, Princeton, NJ, 1957. [116](#)
- [BIA 00] GIOVANNI BIANCO, ALEXANDER ZELINSKY ET MIRIAM LEHRER. Visual landmark learning. In *Proceedings of the 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2000. [53](#), [54](#)
- [BRA 84] VALENTINO BRAITENBERG. *Vehicles*. MIT Press, Cambridge MA, 1984, 1984. [18](#)
- [CAR 83] B. A. CARTWRIGHT ET T. S. COLLETT. Landmark learning in bees. *J. Comp. Physiol*, 151 :521–543, 1983. [20](#), [27](#), [29](#), [37](#), [58](#), [60](#)
- [CAR 87] B. A. CARTWRIGHT ET T. S. COLLETT. Landmark maps for honeybees. *Biol. Cybern.*, 57 :85–93, 1987. [37](#)
- [CHA 96] J.S. CHAHL ET M. V. SRINIVASAN. Visual computation of egomotion using an image interpolation algorithm. *Biological Cybernetics*, 74(5) :405–411, 1996. [89](#)
- [COL 92] T.S. COLLETT, S.N. FRY ET R. WEHNER. Visual landmarks and route following in desert ants. *Journal of Comparative Physiology A*, 170 :435–442, 1992. [84](#)
- [COL 94] T.S. COLLETT ET J. BARON. Biological compasses and the coordinate frame of landmark memories in honeybees. *Nature*, 368 :137–140, 1994. [83](#)
- [COL 97] T.S. COLLETT ET J. ZEIL. The selection and use of landmarks by insects. In M. LEHRER, Ed., *Orientation and Communication in Arthropods*, pages 41–65. Birkhäuser, Basel, 1997. [83](#)
- [COR 98] ROBERT CORI ET JEAN-JACQUES LÉVY. *Algorithmes et programmation*, chapitre 8, pages 179–183. École Polytechnique, Palaiseau, France, 1998. [38](#), [117](#)

- [DEG 04a] T. DEGRIS, L. LACHÈZE, C. BOUCHENY ET A. ARLEO. A spiking neuron model of head-direction cells for robot orientation. In *Proceedings of the Eighth International Conference on Simulation of Adaptive Behavior*, 2004. 262
- [DEG 04b] T. DEGRIS, O. SIGAUD, S. I. WIENER ET ARLEO A. Rapid response of head direction cells to reorienting visual cues : A computational model. *Neurocomputing*, 58-60(C) :675–682, 2004. 262
- [DON 03] S. DONCIEUX. *Evolution de controleurs neuroneux pour animats volants : méthodologie et applications*. Thèse de doctorat, LIP6/AnimatLab, Université Pierre et Marie Curie, France, 2003. 18
- [DUC 96] ANDREW P. DUCHON. Maze navigation using optical flow. In P. MAES, M. MATARIC, J. A. MEYER, J. POLLACK ET S. W. WILSON, Eds, *From animals to animats 4. Proceedings of the Fourth International Conference on Simulation of Adaptive Behavior (SAB-96)*. The MIT Press, 1996. 145
- [EIN 97] T. EINSELE. Real-time self-localization in unknown indoor environments using a panorama laser range finder. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS-97)*, pages 697–703, 1997. 78, 117, 121
- [EIN 01] TOBIAS EINSELE. *Localization in Indoor Environments using a Panoramic Laser Range Finder*. Dr.-ing., Technische Universität München, 2001. 78, 117, 121
- [FAB 02] J. FABRIZIO, J.-P. TAREL ET R. BENOSMAN. Calibration of panoramic catadioptric sensors made easier. In *Proceedings of IEEE Workshop on Omnidirectional Vision (Omnivis'02)*, pages 45–52, Copenhagen, Denmark, 2002. <http://www-rocq.inria.fr/tarel/omnivis02.html>. 103
- [FAU 93] O. FAUGERAS. *Three dimensional computer vision : a geometric viewpoint*. MIT press, 1993. 103
- [FAU 98] OLIVIER FAUGERAS, LONG QUAN ET PETER STURM. Self-calibration of a 1d projective camera and its application to the self-calibration of a 2d projective camera. In *Computer Vision - ECCV'98, 5th European Conference on Computer Vision, Freiburg, Germany, June 2-6, 1998, Proceedings, Volume I*, volume 1406 of *Lecture Notes in Computer Science*. Springer, 1998. 252
- [FIL 01] DAVID FILLIAT. *Cartographie et estimation globale de la position pour un robot mobile autonome*. Thèse de doctorat, Université Pierre et Marie Curie (Paris VI), 2001. 18, 241, 261
- [FIL 03] D. FILLIAT ET J.-A. MEYER. Map-based navigation in mobile robots i. a review of localization strategies. *Journal of Cognitive Systems Research*, 4(4) :243–282, 2003. 18, 181, 241
- [FRA 97] M.O. FRANZ, B. SCHÖLKOPF ET H.H. BÜLTHOFF. Homing by parameterized scene matching. In P. HUSBANDS ET I. HARVEY, Eds, *Proceedings of the 4th European Conference on Artificial Life*, pages 236–245, Cambridge, 1997. MIT Press. 92, 94, 251
- [FRA 98a] M. FRANZ, B. SCHOLKOPF, P. GEORG, H. MALLOT ET H. BULTHOFF. Learning view graphs for robot navigation. *Autonomous Robots*, 5 :111–125, 1998. 174, 182, 216

- [FRA 98b] MATTHIAS O. FRANZ, BERNHARD SCHÖLKOPF, HANSPETER A. MALLOT ET HEINRICH H. BÜLTHOFF. Where did i take that snapshot ? scene-based homing by image matching. *Biological Cybernetics*, 79 :191–202, 1998. [27](#), [33](#), [34](#), [71](#), [92](#), [251](#)
- [FRI 96] H.J. FRIER, E. EDWARDS, C. SMITH, S. NEALE ET T.S. COLLETT. Magnetic compass cues and visual pattern learning in honeybees. *Journal of Experimental Biology*, 199 :1353–1361, 1996. [83](#)
- [GAU 00a] P. GAUSSIER, C. JOULAIN, J.P. BANQUET, S. LEPRÊTRE ET A. REVEL. The visual homing problem : an example of robotics/biology cross fertilization. *Robotics and Autonomous Systems*, 30 :155–180, 2000. [114](#)
- [GAU 00b] PHILIPPE GAUSSIER, C. JOULAIN, J. P. BANQUET, S. LEPRÊTRE ET A. REVEL. The visual homing problem : An example of robotics/biology cross-fertilization. *Robotics and Autonomous Systems*, 30 :155–180, 2000. [51](#), [79](#), [249](#)
- [GIR 03] BENOÎT GIRARD. *Intégration de la navigation et de la sélection de l'action dans une architecture de contrôle inspirée des ganglions de la base*. Thèse de doctorat, Université Paris 6, 2003. [261](#)
- [GOU 03] STÉPHANE GOURICHON, JEAN-ARCADY MEYER, SIO-HOÏ IENG, LAURENT SMADJA ET RYAD BENOSMAN. Estimating ego-motion using a panoramic sensor : Comparison between a bio-inspired and a camera-calibrated method. In HORST HOLSTEIN ET FRED LABROSSE, Eds, *AISB03 Symposium on Biologically Inspired Vision, Theory and Application*, pages 91–101. AISB, April 2003. [205](#), [223](#), [246](#)
- [GUI 02] A. GUILLOT ET J.-A. MEYER. Psikharpax ou l'ambition d'être un rat. *La recherche*, Numéro spécial : Les nouveaux robots, 350 :64–67, 2002. [261](#)
- [GÉR 02] P. GÉRARD. *Apprentissage latent et apprentissage par renforcement dans les systèmes de classeurs*. Thèse de doctorat de l'Université PARIS VI, 2002. [18](#)
- [HAF 01] VERENA VANESSA HAFNER ET RALF MÖLLER. Learning of visual navigation strategies. In M. QUOY, P. GAUSSIER ET J. WYATT, Eds, *Proceedings of the European Workshop on Learning Robots (EWLR- 9)*, pages 47–56, Prague, 2001. [49](#)
- [HAF 02] VERENA V. HAFNER. Learning of a visual homing strategy on a mobile robot. In TONY PRESCOTT ET BARBARA WEBB, Eds, *Robotics as theoretical biology - A SAB'02 Workshop*, pages Poster B–2, 2002. [49](#), [58](#), [60](#)
- [HON 91] J. HONG, X. TAN, B. PINETTE, R. WEISS ET E.M. RISEMAN. Image-based homing. In *Proc. IEEE Intl. Conf. on Robotics and Automation*, pages 620–625, 1991. [27](#), [43](#), [44](#), [127](#)
- [KOH 82] T. KOHONEN. Self-organized formation of topologically correct feature maps. *Biological cybernetics*, 43 :59–69, 1982. [95](#)
- [KOH 99] EDDIE KOHLER. Xbraitenberg - a braitenberg vehicle simulator with pseudo-physics for unix/x11. Site web, 1999. <http://www.lcdf.org/eddielwo/xbraitenberg/>. [202](#)
- [KOH 01] TEUVO KOHONEN, M. R. SCHROEDER ET T. S. HUANG. *Self-Organizing Maps*. Springer-Verlag, New York, 2001. [95](#)

- [LAM 98] DIMITRIOS LAMBRINOS, RALF MÖLLER, ROLF PFEIFER ET RÜDIGER WEHNER. Landmark navigation without snapshots : the average landmark vector model. In N. ELSNER ET R. WEHNER, Eds, *Proc. 26th Göttingen Neurobiology Conference*, volume 1, page 221. Thieme Verlag, 1998. 49
- [LAM 00] D. LAMBRINOS, R. MÖLLER, T. LABHART, R. PFEIFER ET R. WEHNER. A mobile robot employing insect strategies for navigation. *Robotics and Autonomous Systems, special issue : Biomimetic Robots*, 30 :39–64, 2000. 46, 47, 48, 58, 60
- [LEP 00] SACHA LEPRÊTRE, PHILIPPE GAUSSIER ET JEAN-PIERRE COQUEREZ. From navigation to active object recognition. In J.-A. MEYER, A. BERTHOZ, D. FLOREANO, H. ROITBLAT ET S. W. WILSON, Eds, *From animals to animats 6 : Proceedings of the sixth international conference on simulation of adaptive behavior*, 2000. 250
- [LEV 90] TOD S. LEVITT ET DARYL T. LAWTON. Qualitative navigation for mobile robots. *Int. J. Artificial Intelligence*, 44 :305–360, 1990. 19
- [LIN 96] URS LINGG ET RÜDIGER WEHNER. Landmark-based navigation in desert ants *cataglyphis fortis* : use of an external compass reference. In *Proc. Neurobiol. Conf. Göttingen*, page 332, 1996. 83
- [MEY 95] J. A. MEYER. The animat approach to cognitive science. In H. ROITBLAT ET J. A. MEYER, Eds, *Comparative Approaches to Cognitive Science*. MIT Press, 1995. 17
- [MEY 96] J. A. MEYER. Artificial life and the animat approach to artificial intelligence. In M. BODEN, Ed., *Artificial Intelligence*. Academic Press, 1996. 17
- [MEY 97] J. A. MEYER. From natural to artificial life : Biomimetic mechanisms in animat designs. *Robotics and Autonomous Systems*, 22 :3–21, 1997. 17
- [MEY 03] J.-A. MEYER ET D. FILLIAT. Map-based navigation in mobile robots - ii. a review of map-learning and path-planning strategies. *Journal of Cognitive Systems Research*, 4(4) :283–317, 2003. 18, 181, 241
- [MOE 98] THOMAS MOESLUND. Thomas moeslund's evi-d31 sony camera home page. Web page, 1998. [http ://www.vision.auc.dk/ tbm/Sony/](http://www.vision.auc.dk/tbm/Sony/). 217
- [MöL 98a] RALF MÖLLER, DIMITRIOS LAMBRINOS, ROLF PFEIFER, THOMAS LABHART ET RÜDIGER WEHNER. Modeling ant navigation with an autonomous agent. In R. PFEIFER, B. BLUMBERG, J.A. MEYER ET S.W. WILSON, Eds, *From Animals to Animats 5. Proceedings of the Fifth International Conference on Simulation of Adaptive Behavior*, pages 185–194, Cambridge, MA, 1998. The MIT Press/Bradford Books. 46, 58, 60, 216
- [MöL 98b] RALPH MÖLLER ET DIMITRIOS LAMBRINOS. Modeling ant navigation with an autonomous agent. In ROLF PFEIFER, BRUCE BLUMBERG, JEAN-ARCADY MEYER ET STEWART WILSON, Eds, *From Animals to Animats 5, Proceedings of the Fifth International Conference on Simulation of Adaptive Behavior*, volume 5 of *From Animals to Animats*, pages 185–194. MIT Press, 1998. 102
- [MöL 99] RALF MÖLLER. Visual homing in analog hardware. *International Journal of Neural Systems*, 9(5) :383–389, October 1999. 47, 48

- [MöL 00] RALF MÖLLER. Modeling the landmark navigation behavior of the desert ant *cataglyphis*. Rapport technique IFI-AI-00.24, Artificial Intelligence Lab, Dept. Computer Science, University of Zurich, 2000. [92](#)
- [NEL 88] R.C. NELSON ET J. ALOIMONOS. Finding motion parameters from spherical motion fields. *Biological Cybernetics*, 58 :261–273, 1988. [84](#), [87](#), [169](#), [246](#), [247](#)
- [NIC 99] D.J. NICHOLSON, S.P.D. JUDD, B.A. CARTWRIGHT ET T.S. COLLETT. Learning walks and landmark guidance in wood ants *formica rufa*. *Journal of Experimental Biology*, 202 :1831–1838, 1999. [84](#)
- [PIR 04] PATRICK PIRIM. Gvpp : Generic visual perception processor, an embedded vision system. Site web, 2000–2004. <http://bev.lu/gvpp>. [114](#), [216](#)
- [RUC 00] S. RUCHTI. Landmark and compass reference in landmark navigation. Master's thesis, Artificial Intelligence Lab, Institut für Informatik, Universität Zürich, March 2000. [32](#), [83](#), [102](#)
- [RöF 97a] T. RÖFER. Controlling a wheelchair with image-based homing. In *Spatial Reasoning in Mobile Robots and Animals, AISB-97 Workshop*, pages 66–75. Manchester University, 1997. [253](#)
- [RöF 97b] THOMAS RÖFER. *Panoramic Image Processing and Route Navigation*. Thèse de doctorat, Universität Bremen, January 1997. [95](#), [96](#), [97](#), [99](#), [100](#), [216](#), [251](#)
- [SAH 98] EROL SAHIN ET PAOLO GAUDIANO. Visual looming as a range sensor for mobile robots. In R. PFEIFER, B. BLUMBERG, J.A. MEYER ET S.W. WILSON, Eds, *From Animals to Animats 5. Proceedings of the Fifth International Conference on Simulation of Adaptive Behavior*, pages 185–194, Cambridge, MA, 1998. The MIT Press/Bradford Books. [78](#)
- [SMI 04] RUSSELL SMITH ET AL. Open dynamics engine. Site web, 1995–2004. <http://ode.org/>. [202](#)
- [SVO 96] T. SVOBODA ET PETER STURM. What can be done with a badly calibrated camera in ego-motion estimation? Rapport technique CTU-CMP-19, Center for Machine Perception, Czech Technical University, Prague, Nov 1996. [174](#)
- [TRU 97] O. TRULLIER, S. WIENER, A. BERTHOZ ET J. A. MEYER. Biologically-based artificial navigation systems : Review and prospects. *Progress in Neurobiology*, 51 :483–544, 1997. [19](#), [52](#)
- [WEB 99] KEVEN WEBER, SVETHA VENKATESH ET MANDYAM SRINIVASAN. Insect-inspired robotic homing. *Adaptive Behavior*, 7(1) :65–96, Winter 1999. [33](#), [45](#), [49](#), [58](#), [60](#), [127](#), [208](#), [216](#), [222](#)
- [WEH 96] RÜDIGER WEHNER, BARBARA MICHEL ET PER ANTONSEN. Visual navigation in insects : coupling of egocentric and geocentric information. *Journal of Experimental Biology*, (199) :129–140, 1996. [83](#)
- [ZHA 97] ZHENGYOU ZHANG. Parameter estimation techniques : A tutorial with application to conic fitting. *Image and Vision Computing Journal*, 15(1) :59–76, 1997. [259](#)
- [ZIP 85] D. ZIPSER. A computational model of hippocampal place fields. *Behavioral Neuroscience*, (99) :1006–1018, 1985. [79](#)