

# Rapport FNRS 2000 : Vers une synthèse des approches algébriques et géométriques de la robotique mobile

Matthieu Amiguet

Octobre 2000 — document de travail

## Table des matières

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>État de l'art</b>	<b>2</b>
2.1	Représentation a priori . . . . .	2
2.2	Représentation construite par le robot . . . . .	3
2.3	Synthèse . . . . .	4
<b>3</b>	<b>Vers un synthèse des approches</b>	<b>4</b>
3.1	Démarche . . . . .	4
3.2	Le problème . . . . .	6
3.2.1	Discussion . . . . .	6
3.2.2	Formalisation . . . . .	7
3.2.3	Exemples . . . . .	8
3.3	Pistes de résolution . . . . .	9
<b>4</b>	<b>Conclusion et perspectives</b>	<b>9</b>

## 1 Introduction

Pour qu'un robot mobile puisse remplir ses tâches, il est évidemment indispensable qu'il dispose de mécanismes efficaces pour se situer dans son environnement et planifier ses trajectoires. Les algorithmes existants pour peuvent se répartir en deux classes principales : les algorithmes de navigation par carte et ceux de navigation par balises.

Chacune de ces classes d'algorithmes possède ses avantages et ses limitations intrinsèques. Malgré les nombreux travaux sur le sujet, dont on trouvera un aperçu dans le

paragraphe 2, peu d'efforts ont été déployés en direction d'une synthèse de ses approches permettant de bénéficier des avantages de chacune.

Ce papier propose des pistes en direction d'une telle synthèse. Le paragraphe 2 présente un bref survol de l'état de l'art ; Le paragraphe 3 expose ensuite les pistes que nous proposons pour obtenir une synthèse intéressante des deux types d'approche ; le paragraphe 4 termine sur quelques perspectives de recherche dans cette direction.

## 2 État de l'art

Un des éléments importants de la robotique mobile semble être la représentation interne au robot de son environnement. Quelques travaux ont tenté de se passer complètement de cette représentation [Bro86, Bro89, Con90, H<sup>+</sup>99, Thr95], mais les tâches accessibles par ce genre de procédé sont généralement assez spécialisées. Par exemple, [H<sup>+</sup>99] propose un système capable de se déplacer vers une source sonore en évitant les obstacles. Si cette approche possède effectivement des avantages (possibilité de l'utiliser en mauvaises conditions d'éclairage, possibilité de faire l'économie d'une représentation explicite de l'environnement), il est évident que les applications possibles sont assez spécialisées.

La plupart des recherches se concentrent donc sur le cas où le robot dispose d'une représentation explicite de son environnement. Là encore, deux approches sont possibles : il est possible de fournir *a priori* une représentation de l'environnement au robot ; dans ce cas, le problème est de l'exploiter de manière adéquate. Une autre possibilité est de laisser le robot construire lui-même sa représentation de l'environnement ; nous verrons que, si cette approche pose des nouveaux problèmes, elle possède aussi ses avantages propres. Nous revenons maintenant sur les caractéristiques de ces deux approches.

### 2.1 Représentation a priori

Fournir au robot une carte de l'environnement présente de nombreux avantages de simplicité et de rapidité de mise en oeuvre. Cependant, cette approche nécessite de résoudre de nombreux problèmes dont nous citons ci-après les principaux. La représentation fournie au robot dans ce genre d'approche est généralement une carte, au sens habituel du terme.

**Orientation** La localisation d'un système sur une carte donnée consiste non seulement à déterminer sa position, mais aussi son orientation. La plupart des approches traitent d'abord la question de la localisation, puis celle de la position, mais le contraire est aussi possible [RG00].

**Fonction de contact** Les robots sont souvent pauvres en capteurs par rapport à leur morphologie. Cela pose de nombreux problèmes au moment de la planification de la trajectoire pour éviter les collisions. Les solutions peuvent parfois demander des formalismes assez complexes, comme dans le cas par exemple de [P<sup>+</sup>] qui utilise un formalisme inspiré de la biologie moléculaire.

**Fusion multi-capteurs** Le fait de représenter l’environnement sous forme de carte nécessite de projeter toutes les perceptions dans cette carte pour les rendre utilisables. Dans le cas de robots possédant plusieurs modalités perceptuelles, cela pose le problème de la fusion de ces données ([D<sup>+</sup>95, HS95, WD98]).

**Calibrage** De même, la projection des données capteurs sur la carte nécessite généralement un calibrage des capteurs. Ce problème est abordé par exemple dans [HS95].

**Choix des balises** Pour se repérer dans un environnement cartographié, il faut choisir des balises qui soient accessibles et donnent des informations pertinentes. On trouvera des exemples de traitement de ce problème dans [DMM96] et [MA98]. [Bau95] propose une approche originale dans laquelle le robot utilise ses capacités motrices pour affiner sa perception des balises.

**Puissance de calcul** De manière générale, les approches par carte demandent de grandes capacités de calcul. [D<sup>+</sup>95] par exemple, qui présente un ambitieux programme d’exploration planétaire, rencontre déjà de gros problèmes de capacité de calcul même sur des expériences partielles. Différentes approches ont été proposées pour remédier partiellement à ce problème : il est par exemple possible de ne considérer que des cartes locales [WD95, FPCM97], ou encore de travailler à une grande échelle et de ne raffiner la carte que lorsque c’est nécessaire [LN98].

**Structure motrice** Si le robot n’est pas holonome, c’est à dire qu’il existe certaines contraintes à ses mouvements (par exemple, les robots de type voiture), il est bien sûr important de tenir compte de ceci lors de la navigation du robot. Des travaux dans ce sens ont été effectués par [Des96] ou [?]. Nous reviendrons plus tard sur ce dernier travail.

## 2.2 Représentation construite par le robot

La plupart des problèmes ci-dessus sont dus à la différence de l’appareil sensoriel dont disposent le concepteur d’une part et le robot d’autre part. Ainsi, pour éviter d’y être confronté, une possibilité est de faire construire par le robot lui-même sa représentation du monde. Les représentations ainsi obtenues sont souvent beaucoup plus simples et moins riches que dans l’approche précédente, pour des raisons évidentes de réalisabilité. Il s’agit souvent de structures algébriques telles des graphes.

Ce genre d’approche, basé sur la prise en compte des capacités motrices du robot, est en général très inspirée du courant “sensori-moteur” [PS97, Ber00] initié par Piaget et ses successeurs [Pia90].

Pour les raisons évoquées ci-dessus, les approches par graphes [Rod94, D<sup>+</sup>97, Dav86, BD90, KL88] permettent d’éviter élégamment les problèmes de calibrage, fusion, choix des balises, prise en compte de la structure motrice, etc., qui se trouvent intégrés naturellement dans la démarche même. Concernant le calibrage, par exemple, la citation suivante met bien en évidence l’esprit de la démarche<sup>1</sup> :

<sup>1</sup>Tirée d’un ouvrage sur la navigation humaine, cette citation pourrait sembler déplacée ici. Cependant le remarquable ouvrage de Hutchins, par la grande rigueur de son exposé, permet de bien mettre en évidence de nombreux parallèles entre les problèmes rencontrés au cours de la navigation humaine,

Early charts of the Mediterranean showed a pronounced upward tilt in the eastern end. This tilt was produced by the difference in magnetic variation between the western and the eastern reaches of the Mediterranean Sea. If the cartographer uses a magnetic compass to make the chart, and the navigator uses a magnetic compass to determine courses, and if both compasses show the same errors in the same places, why would anyone care and how could anyone ever notice that the charts put the land in the wrong places? [Hut95, p.108]

De plus, ces approches se montrent souvent moins gourmandes en temps de calcul. Par contre, elles sont plus sujettes à des problèmes de type “état caché”. Deux états distincts peuvent en effet être indistingables par le robot.

Notons que dans certains cas, ceci est acceptable : il arrive que le robot n’aie pas besoin de distinguer ces états pour bien accomplir sa tâche. Le seul problème est que cela rend la tâche plus difficile pour l’observateur humain.

La grosse faiblesse de ces approches algébriques est qu’elles rendent souvent difficile la remise à jour de la représentation dans un environnement changeant ou l’extrapolation sur la base de la “carte” déjà à disposition. Ainsi, si une porte s’ouvre dans l’environnement du robot, lui fournissant ainsi un raccourci, l’approche par graphe ne permettra généralement pas d’inférer ceci, alors que les approches par cartes le permettent généralement.

## 2.3 Synthèse

Les travaux ci-dessus permettent de mettre en évidence les avantages et inconvénients de chacune des méthodes. Nous les résumons dans le tableau 1 :

# 3 Vers un synthèse des approches

## 3.1 Démarche

Comme il semble difficile de combiner tous les avantages des deux approches exposées ci-dessus, nous ferons dans ce texte les choix suivants :

1. Nous visons une approche qui soit applicable lorsque l’environnement n’est pas connu à l’avance.
2. Nous désirons que celle-ci permette de faire du raisonnement géométrique sur les trajectoires.
3. Par contre, nous renonçons à viser une transparence pour le concepteur de la représentation du monde obtenue : si le robot sait bien la mettre en oeuvre, cela nous suffira, même si elle est difficile à gérer pour le concepteur.

En effet, la facilité de compréhension de la représentation n’est utile qu’en phase de développement (débugage, etc.) ou dans des applications très spécifiques, où le but du

---

et ceux rencontrés en robotique (calibrage, choix des balises, localisation, planification de trajectoire, etc.)

Navigation	Avantages	Limites
par cartes	Permet le raisonnement géométrique sur les trajectoires (optimisation, extrapolation, etc.)	Difficile à établir par le robot si l'environnement n'est pas connu à l'avance.
	Bien compréhensible par le concepteur	Difficile à utiliser pour certains robots (lorsque les capteurs ne sont pas adaptés à la perception de certains éléments de la carte)
par balises	Assez facile à mettre en oeuvre dans un environnement qui n'est pas connu à l'avance	Ne permet pas le raisonnement géométrique sur les trajectoires
	Parfois difficile à gérer pour le concepteur (la représentation de l'espace obtenue peut être très différente de notre perception)	Les caractéristiques de l'environnement utilisées pour la navigation sont exactement celles que le robot sait percevoir

TAB. 1 – Avantages et inconvénients des deux méthodes

robot serait effectivement de faire de la cartographie au sens usuel du terme. Dans les autres cas, ce qui nous importe avant tout est d'obtenir un robot autonome et capable de remplir sa tâche, même si sa représentation du monde nous paraît étonnante.

Nous proposons donc la démarche suivante :

1. Pour pouvoir laisser le robot construire une représentation de son environnement qui corresponde à ses facultés perceptives, nous choisirons d'utiliser une approche par balise pendant la première phase d'exploration.
2. Lorsque le graphe de navigation ainsi obtenu semble se stabiliser sur quelque chose d'utilisable, l'idée est de plonger ce graphe dans un espace géométrique continu qui permette le raisonnement.

Cette démarche permet donc de combiner les avantages des deux approches traditionnelles. De plus, le fait que le robot dispose au bout d'un moment de deux structures superposées pour se localiser permet de rendre le processus de localisation plus robuste.

Par contre, nous verrons que le problème de trouver un tel plongement semble assez délicat et n'est pas encore résolu.

## 3.2 Le problème

### 3.2.1 Discussion

Nous supposons que notre robot a à sa disposition un ensemble d'actions  $A_i$  disposant des propriétés suivantes :

1. Chaque action a une intensité standard, linéairement compatible avec le temps : effectuer l'action  $a$  pendant le temps  $t$  à l'intensité  $i$  revient à effectuer cette même action  $a$  pendant le temps  $\frac{t}{n}$  à l'intensité  $ni$  (par exemple, la vitesse de déplacement a cette propriété).
2. Chaque action  $a_i$  possède une action inverse  $-a_i$  : l'action  $a_i$  suivie de l'action  $-a_i$  ramène le robot à son point de départ. Cette connaissance peut être donnée au robot par le concepteur ou apprise en même temps que le graphe.

Sur cette base, le robot aura construit un graphe  $G$  tel que :

1. Les noeuds de  $G$  sont des états instantanés, et ses arcs décrivent les passages possibles d'un état à l'autre. Chaque arc est étiqueté par une combinaison linéaire formelle des éléments de  $A$  représentant l'action nécessaire pour effectuer la transition. Par exemple, un arc étiqueté par  $a_1 + 2a_2$  signifie que le robot doit effectuer simultanément l'action  $a_1$  à son intensité standard et l'action  $a_2$  au double de son intensité standard pendant un temps unitaire déterminé<sup>2</sup>.
2. Le graphe  $G$  est déterministe : deux arcs de même source et de même étiquette sont confondus.

Pour le plongement, nous supposons que le robot a à sa disposition un graphe décrivant l'environnement de manière pertinente pour lui ; en particulier, que deux états distincts dans le graphe doivent rester distinct dans le plongement, et que deux états distinguables par le robot sont effectivement distingués dans le graphe.

Dans ces conditions, notre problème devient de trouver un espace  $E$  et un plongement  $p$  de  $G$  dans  $E$  de manière à ce que

1. Chaque noeud de  $G$  est envoyé sur un point de  $E$ .
2. Chaque arc  $A : e_1 \rightarrow e_2$  de  $G$  est envoyé sur un chemin dans  $E$  allant de  $p(e_1)$  à  $p(e_2)$ .
3.  $p$  doit être injectif sur les noeuds.
4. L'action indexant chaque arc donne une mesure<sup>3</sup> de la "longueur" de l'arc. Nous désirons que cette mesure soit conservée par le plongement, pour pouvoir ensuite l'utiliser pour du raisonnement (optimisation de trajectoire, ...).

Il est nécessaire de revenir ici sur la forme de l'espace que nous visons d'obtenir. Cette démarche étant avant tout destinée à la situation où le robot doit se construire lui-même une représentation d'un environnement inconnu. Dans le cas extrême, le robot ne connaît

---

<sup>2</sup>Par contre, nous n'admettons que des actions constantes sur un arc : une action du type "faire  $a_1$  puis  $2a_2$ " devra être représentée par deux arcs et un état intermédiaire.

<sup>3</sup>Par exemple, en termes d'énergie nécessaire.

même pas la sémantique des commandes qu'il a à disposition. Dans ces conditions, il est improbable que l'espace qu'il construit soit celui que nous percevons en tant qu'humains. Prenons un exemple : si le robot a à sa disposition deux commandes, qu'un observateur extérieur décrirait comme *avancer* et *tourner sur place*, mais que le robot ne connaît pas cette sémantique, il y a toutes les chances pour qu'il perçoive ces deux actions comme étant des déplacements dans l'espace qu'il construit, alors que nous le voyons rester sur place dans le deuxième cas<sup>4</sup>.

Pourtant, même si cela complique la vie de l'observateur, cette représentation a une force au niveau du robot : elle est très opérationnelle. Pour passer d'un état à un autre, le robot doit se déplacer dans cet espace de représentation indépendamment de la sémantique des actions.

L'espace obtenu par notre démarche sera donc résolument plus proche d'un espace de configurations que de l'espace "réel".

### 3.2.2 Formalisation

Formellement, on peut représenter le problème exposé ci-dessus de la manière suivante :

**Graphe** Un graphe  $G$  est un quadruplet  $\langle G_0, G_1, s, b \rangle$  où  $G_0$  est l'ensemble des noeuds,  $G_1$  l'ensemble des arcs, et  $s, b : G_1 \rightarrow G_0$  sont les fonctions *source* et *but* respectivement.

**Graphe indexé par un ensemble d'actions (GIA)** Un graphe  $G$  *indexé par l'ensemble d'actions*  $A$  (GIA ci-dessous) est un graphe  $\langle G_0, G_1, s, b, a \rangle$  muni d'une fonction supplémentaire  $a : G_1 \rightarrow V(A)$ , où  $V(A)$  est l'espace vectoriel libre sur  $A$ <sup>5</sup>. Cette fonction a donne pour chaque arc l'action correspondante. Pour un arc  $x$  d'un GIA, nous noterons  $\|x\|$  la norme  $\|a(x)\|$  du vecteur  $a(x)$  dans  $V(A)$ .

**Carte d'un GIA** Soit  $G$  un GIA. Nous appellerons *carte de  $G$*  un triple  $\langle M, c_0, c_1 \rangle$  formé d'une variété différentiable  $M$  et de deux fonctions  $c_0, c_1$  vérifiant les axiomes suivants :

1.  $c_0 : G_0 \rightarrow M$  attribue à chaque noeud de  $G$  un point de  $M$ .
2.  $c_1 : G_1 \rightarrow M^{[0,1]}$  attribue à chaque arc de  $G$  un chemin dans  $M$ .
3.  $c_0$  est injective.
4. Pour tout arc  $x : e_1 \rightarrow e_2$  de  $G$ ,  $c_1(x)(0) = c_0(e_1)$  et  $c_1(x)(1) = c_0(e_2)$  (" $c$  respecte la source et le but").
5. Pour tout arc  $x$  de  $G$ , la longueur  $L(x)$  de  $c_1(x)$  dans  $M$  est égale à  $\|x\|$ .

Cette notion de carte nous fournit un plongement tel que nous le désirons, mais qui ne contient pas toutes les informations : l'effet des actions du robot n'y est pas représenté. C'est la raison de l'introduction de la notion de *carte navigable* ci-dessous. Le choix de plonger  $G$  dans une variété  $M$  plutôt que dans  $\mathbb{R}^n$  permet de se limiter à la partie de

<sup>4</sup>Voir le paragraphe 3.2.3 pour un développement de cet exemple.

<sup>5</sup>C'est à dire l'ensemble des combinaisons linéaires formelles des éléments de  $A$ .

l'espace des configuration qui est physiquement pertinente ; ceci peut avoir une grande importance au moment de chercher des solutions d'équations différentielles<sup>6</sup>. L'exigence que  $M$  soit différentiable s'explique par l'apparition d'équations différentielles dans la notion de carte navigable.

**Carte navigable** Soit  $G$  un GIA sur  $A$  ; soit  $n$  la cardinalité de  $A$ . Une *carte navigable* de  $G$  est une carte  $\langle M, c_0, c_1 \rangle$  munie de  $n$  champs de vecteurs unitaires  $v_i : M \rightarrow R^m$ , où  $m$  est la dimension de  $M$ , tels que

1.  $v_i(m)$  est tangent à  $M$  pour tout point  $m$  de  $M$ .
2.  $\forall e \in G_1, \forall t \in [0, 1], a(x) = \sum \lambda_i a_i \Rightarrow \frac{d}{dt} c_1(x)(t) = \sum \lambda_i v_i$ .

Ces deux exigences permettent d'assurer que les champs de vecteurs  $v_i$  représentent bien l'effet des actions  $a_i$  tel qu'il est décrit dans le graphe  $G$ .

### 3.2.3 Exemples

Pour bien illustrer la démarche ci-dessus, nous prenons deux exemples d'architectures de robots et des espaces correspondants. Dans les deux cas, nous supposons que notre robot se situe dans un labyrinthe orthogonal orienté selon les points cardinaux. Nous allons voir que ce même environnement peut donner lieu à des espaces de représentation fondamentalement différents suivant la structure motrice du robot. Nous ne ferons pas ici toute la démarche qui consiste à exhiber un graphe, une variété et le plongement correspondant ; en effet, un la variété utilisée semble, pour un robot donné, en partie indépendante de la forme exacte du graphe. Nous nous contenterons donc d'exposer l'espace qui peut être utilisé pour le plongement d'un tel graphe<sup>7</sup>.

**Le robot NESO** Considérons un robot qui possède 4 actions,  $A = \{N, E, S, O\}$ , correspondant à des déplacements dans la directions des 4 points cardinaux respectivement. Dans ce cas simple, l'espace  $\mathbb{R}^2$  est apte à servir d'espace de plongement pour nos graphes. Les champs de vecteurs d'action sont alors évidemment les champs constants  $v_N = (1, 0)$ ,  $v_E = (0, 1)$ ,  $v_S = -v_N$  et  $v_O = -v_E$ . Ensuite, pour un graphe donné, la condition 2 ci-dessus pour une carte navigable permet d'effectuer sans peine le plongement.

**Le Robot TA** Considérons maintenant un robot dont les deux actions  $A = \{T, A\}$  ont pour sémantique (externe!) *tourner sur place* et *avancer* respectivement. Dans ce cas, l'état du robot est défini par 3 variables : 2 variables de position et une d'orientation. L'espace de plongement sera alors  $\mathbb{R}^2 \times S^1$ , où  $S^1$  est le cercle, représenté par exemple par l'intervalle  $[0, 2\pi]$  ; Le champ de vecteurs pour l'action  $T$  est alors le champ constant  $v_T = (0, 0, 1)$  et celui pour l'action  $A$  prend la forme  $v_A = (\sin t, \cos t, 0)$ .

<sup>6</sup>Dans le cas où les équations ont une solution sur des sous-espaces sans admettre de solution globale.

<sup>7</sup>Il est à noter que nous pouvons prendre ce raccourci parce que nous choisissons un point de vue *externe*, dans lequel nous connaissons la sémantique des actions. Pour construire l'espace de plongement à partir du graphe seul, il est bien sûr indispensable d'utiliser la forme de ce graphe !



Ces deux exemples mettent en évidence un point intéressant : la dimensionnalité de l'espace de plongement est tout à fait indépendante du nombre d'actions à disposition, puisque dans le cas NESO, on a 4 actions et 2 dimensions, alors que le cas TA montre 3 dimensions pour 2 actions.

### 3.3 Pistes de résolution

Le problème ci-dessus est maintenant bien formalisé, et les techniques mathématiques peuvent entrer en jeu pour tenter de le résoudre. Nous n'avons pas encore abordé cette phase de résolution, car elle demanderait, étant donné les compétences présentes dans notre équipe, un travail considérable d'investigation formelle. À première vue, trois branches des mathématiques sont susceptibles d'apporter des éléments de solution. Nous les passons brièvement en revue.

**La géométrie différentielle** La piste la plus évidente est de considérer l'ensemble des contraintes ci-dessus comme un système d'équations différentielles liant les fonctions  $s, b, a, c_0, c_1$  et  $v_i$ . Il s'agirait donc de considérer les conditions de solubilité, de voir si elles imposent des contraintes acceptables pour le problème considéré, et enfin d'étudier les méthodes d'intégration numérique qui permettraient la construction algorithmique de cet espace de représentations par le robot lui-même.

**La théorie des graphes** Cette théorie possède en effet de nombreux outils pour étudier des caractéristiques avancées de certains graphes, comme par exemple la possibilité de les plonger dans le plan en respectant leur structure. Mais en l'état actuel, nous ne savons pas si ces outils permettent d'aborder le problème ci-dessus, qui se situe en dimension quelconque et concerne des graphes "augmentés", avec une structure supplémentaire (arcs indexés par des vecteurs normés).

**La théorie des catégories** Une autre manière de voir le problème ci-dessus est de le considérer comme un problème de changement d'espace mathématiques. Le formalisme approprié serait alors évidemment la théorie des catégories. Cependant, une étude du problème sous cet angle nécessiterait une reformulation importante, puisqu'il faudrait reprendre toutes les définitions et équations ci-dessus et les réécrire sous une forme "element-free", en n'utilisant que les morphismes des différentes structures considérées. Il s'agit d'un travail délicat, mais qui mettrait à disposition tout le puissant arsenal catégoriel.

## 4 Conclusion et perspectives

La démarche exposée ci-dessus est bien entendu incomplète, puisqu'elle se borne à proposer une mathématisation d'un problème sans étudier sa résolution. Cette formalisation semble cependant assez complète pour que le problème se situe maintenant au niveau mathématique, permettant une investigation purement formelle de la question.

Si elle aboutit, cette démarche constituera une synthèse intéressante des approches géométriques et algébriques de la navigation d'un robot en environnement inconnu, puisqu'elle permettra d'allier simplicité d'exploration et puissance du raisonnement géométrique.

De plus, la superposition d'une structure algébrique, discrète et d'une structure géométrique, continue, pour représenter un même espace pourrait fournir des nouveaux outils pour obtenir une navigation robuste mais pas trop lourde au niveau du calcul<sup>8</sup>.

Dans les faits, les conditions imposées ci-dessus pour une carte navigable donnent vraisemblablement toute une famille d'espaces solutions. Dans ce cas, seule une exploration plus poussée de l'espace environnant permettrait au robot de distinguer quel espace est approprié<sup>9</sup>. Une piste de recherche ultérieure serait donc d'étudier les stratégies d'exploration complémentaire qui permettraient de résoudre les ambiguïtés à ce niveau.

Pour finir, notons que ce travail pourrait avoir un intérêt plus épistémologique : nous avons vu dans les exemples du paragraphe 3.2.3 que l'espace de plongement dépend grandement de la structure motrice du robot. En mettant ce fait en évidence dans le cadre contrôlé de la robotique, on pourrait être tenté d'élargir ensuite le cadre à d'autres entités mouvantes établissant des représentations de leur environnement, et en particulier l'humain. En effet, on sait que la représentation euclidienne tridimensionnelle qui nous paraît évidente varie considérablement suivant les époques et les cultures. Poser la question de la justification de cette représentation en termes sensori-moteurs comme ci-dessus pourrait amener un éclairage intéressant sur cette problématique<sup>10</sup>.

## Références

- [Bau95] Rudolf Bauer. Active manoeuvres for supporting the localization process of an autonomous mobile robot. *Robotics And Autonomous Systems*, 16(1) :39–46, November 1995.
- [BD90] K. Basye and T. Dean. Map learning with undistinguishable locations. In M. Henrion, L.N. Kanal, and J.F. Lemmer, editors, *Uncertainty in Artificial Intelligence*, volume 5, pages 331–340. Elsevier, 1990.
- [Ber00] Pierre-Yves Berger. Trying to discern what is in the black box . Technical report, PHITECO, 2000.
- [Bro86] R. Brooks. A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation*, 2(1) :14–23, 1986.
- [Bro89] R. Brooks. A robot that walks : Emergent behaviours from a carefully evolved network. *Neural Computation*, 1(2) :253–262, 1989.

---

<sup>8</sup>On peut par exemple imaginer d'utiliser le graphe tant que "tout va bien", d'où un traitement assez léger, et de passer à l'espace continu lorsqu'une surprise intervient, ou que l'on veut explorer de nouveaux chemins.

<sup>9</sup>Il faudrait probablement aussi introduire un principe d'économie, stipulant que lorsque plusieurs candidats se présentent, le plus simple sera le mieux. Il n'est pas nécessaire de considérer un espace à 6 dimensions si 2 suffisent !

<sup>10</sup>Même si la structure sensori-motrice de l'être humain est infiniment plus complexe que celle des robots que nous avons considéré dans ce texte !

- [Con90] J. Connell. *Minimalist Mobile Robotics : A Colony-style Architecture for an Artificial Creature*. Academic Press, Boston, MA, 1990.
- [D<sup>+</sup>95] Michel Devy et al. On autonomous navigation in a natural environment. *Robotics And Autonomous Systems*, 16(1) :5–16, November 1995.
- [D<sup>+</sup>97] G. Dudek et al. Map validation and robot self-location in a graph-like world. *Robotics and Autonomous Systems*, 22(2) :159–178, November 1997.
- [Dav86] E. Davis. *Representing and Acquiring Geographic Knowledge*. Pitman, London, 1986.
- [Des96] Guy Desaulniers. On shortest paths for a car-like robot maneuvering around obstacles. *Robotics And Autonomous Systems*, 17(3) :139–148, May 1996.
- [DMM96] Xioatie Deng, Evangelos Milios, and Andy Mirzaian. Landmark selection strategies for path execution. *Robotics And Autonomous Systems*, 17(3) :171–185, May 1996.
- [FPCM97] N. Le Fort-Piat, I. Collin, and D. Meizel. Planning robust displacement missions by means of robot-tasks and local maps. *Robotics And Autonomous Systems*, 20(1) :99–114, April 1997.
- [H<sup>+</sup>99] Jie Huang et al. A model-based sound localization system and its application to robot navigation. *Robotics And Autonomous Systems*, 27(4) :199–209, June 1999.
- [HS95] Joachim Horn and Günther Schmidt. Continuous localization of a mobile robot based on 3D-laser-data, predicted sensor images and dead-reckoning. *Robotics And Autonomous Systems*, 14(2–3) :99–118, May 1995.
- [Hut95] Edwin E. Hutchins. *Cognition in the Wild*. MIT Press, 1995.
- [KL88] B. Kuipers and T. Levitt. Navigation and mapping in large-scale space. *AI Magazine*, pages 25–43, 1988.
- [LN98] Oliver Lemon and Ulrich Nehmzow. The scientific status of mobile robotics : Multi-resolution mapbuilding as a case study. *Robotics And Autonomous Systems*, 24(1–2) :5–15, August 1998.
- [MA98] Claus B. Madsen and Claus S. Andersen. Optical landmark selection for triangulation of robot position. *Robotics And Autonomous Systems*, 23(4) :277–292, July 1998.
- [P<sup>+</sup>] John W. Perram et al. Modelling an autonomous agent in a multiple agent world. COST 13 Project Working Paper.
- [Pia90] Jean Piaget. *Morphismes et catégories*. Delachaux & Niestlé, 1990.
- [PS97] Rolf Pfeifer and Christian Scheier. Sensory-motor coordination : The metaphor and beyond. *Robotics And Autonomous Systems*, 20(2–4) :157–178, 1997.
- [RG00] Antonio Reina and Javier Gonzalez. A two-stage mobile robot localization method by overlapping segment-based maps. *Robotics and Autonomous Systems*, 31(4) :312–225, June 2000.

- [Rod94] Miguel Rodriguez. *Approche Constructiviste de l'Architecture de Contrôle et de la Représentation des Connaissances*. PhD thesis, Université de Neuchâtel, September 1994.
- [Thr95] Sebastian Thrun. An approach to learning mobile robot navigation. *Robotics And Autonomous Systems*, 15(4), October 1995.
- [WD95] F. Wallner and R. Dillmann. Real-time map refinement by use of sonar and active stereo-vision. *Robotics And Autonomous Systems*, 16(1) :47–56, November 1995.
- [WD98] P. Weckesser and R. Dillmann. Modeling unknown environments with a mobile robot. *Robotics And Autonomous Systems*, 23(4) :293–300, July 1998.