

# Reconstruction et exploration de scènes par vision active

ÉRIC MARCHAND, FRANÇOIS CHAUMETTE

IRISA - INRIA Rennes - Université de Rennes I  
Campus Universitaire de Beaulieu - 35042 Rennes-Cedex  
Email: {marchand, chaumett}@irisa.irisa.fr

## Résumé

*Nous proposons dans cet article une méthode de reconstruction de scènes basée sur un concept de vision active. L'approche retenue est basée sur le contrôle du mouvement de la caméra, celui-ci étant contraint de manière à optimiser la reconstruction de chaque primitive de la scène. Cette approche nécessitant une focalisation sur les objets considérés, nous présentons des stratégies de perception assurant une reconstruction complète. Aucune connaissance sur le nombre, la position et les dimensions des objets constituant la scène n'étant disponible, des processus d'exploration sont mis en œuvre. Une méthode de recherche du positionnement optimal de la caméra utilisant les informations 2D disponibles ainsi que la carte 3D déjà acquise est présentée. Des résultats expérimentaux menés sur une cellule de vision robotique sont finalement présentés.*

## 1 Introduction

Une perception correcte de l'environnement est un élément clé de l'indépendance que l'on souhaite donner aux robots. Depuis quelques années, de nombreux travaux menés en vision artificielle se sont fixés pour objectif la réalisation de systèmes capables d'accéder à la géométrie spatiale d'une scène à partir de son observation par une ou plusieurs caméras mobiles [1][7][11][12][23][22][26]. Ces systèmes ont pour objectif de fournir une description géométrique claire et complète de la scène à partir d'une séquence d'images souvent bruitées et difficilement exploitables. L'étude proposée ici tente d'apporter sa contribution au problème de la reconstruction d'environnements assez restreints (objets statiques, hypothèses sur la nature des objets constituant la scène,...) avec comme ambition la reconstruction de scènes de plus en plus complexes. L'approche retenue pour l'obtention d'une carte précise de l'environnement est basée sur le concept de la vision active [3][4]. La vision active peut être vue comme un processus d'acquisition "intelligent" des données. Les problèmes soulevés lors de la conception de systèmes de vision sont principalement leur sensibilité au bruit et leur faible précision. L'objectif de la vision active est d'élaborer des stratégies de contrôle des paramètres de la caméra (position, mise au point,

vitesse,...) de manière à améliorer la connaissance de l'environnement [3]. Dans cet article, nous utilisons la vision active à deux niveaux différents : un **niveau local**, où les mouvements de la caméra sont contraints de manière à optimiser la qualité des résultats de reconstruction de chaque primitive 3D observée, et un **niveau global**, pour explorer les zones de la scène non encore observées.

L'observabilité du mouvement de la caméra qui est souvent essentielle en reconstruction 3D caractérise un domaine de recherches récent appelé *vision dynamique*. Deux principales approches ont été mises en œuvre dans la littérature pour traiter ce problème. L'approche discrète, dans le sens où les images sont acquises à des instants discrets [7][12][23], et l'approche continue, dans le sens où les images sont acquises à la cadence vidéo. Cette dernière approche est étroitement liée à l'estimation du champ des vitesses apparentes dans l'image (optic flow) qui est généralement bruité et partiellement incorrect notamment près des contours d'occlusions [1][11][22][26]. Une méthode hybride permettant de résoudre le problème de l'estimation de la structure spatiale d'une primitive 3D par vision active a été récemment développée [5][6]. Plus précisément, cette approche consiste à contraindre les mouvements de la caméra de manière à obtenir une estimation précise et robuste de primitives géométriques paramétrables telles les points, les droites, les cylindres, les sphères, etc. L'asservissement visuel [10], qui consiste à considérer le système de vision comme un capteur particulier associé à une tâche et inséré dans une boucle de commande, permet de générer les mouvements de la caméra correspondant à ces contraintes.

Cependant, cette approche ne permet de reconstruire qu'une seule primitive à la fois et ne permet pas d'assurer la complétude de la reconstruction. De plus, une connaissance *a priori* sur la nature de la primitive est nécessaire afin de générer les mouvements adéquats de la caméra. Il faut donc s'abstraire des contraintes imposées par la méthode en définissant des stratégies de perception qui permettent l'acquisition d'une carte précise et complète de la scène. Ce problème d'exploration et de positionnement de la caméra correspond à un **niveau global** dans le cadre de la perception active. Plusieurs approches ont été proposées afin de résoudre le problème du calcul automatique de point de vue [19]. Elles peu-

vent se diviser en deux classes principales en fonction de la connaissance *a priori* sur l'environnement. Le cas où une modélisation géométrique complète de la scène est disponible est traité dans [9][20][21]. Le problème de l'*exploration autonome* est posé dans le cas où la caméra évolue dans un environnement totalement ou partiellement inconnu [8] [17][24][25]. Dans notre cas, nous devons obtenir une carte de l'environnement sans connaissance *a priori* sur le nombre, la position et la dimension des objets de la scène (composée, par hypothèse, de cylindres, de polygones et de segments).

L'article est structuré de la manière suivante : la première partie rappelle la technique que nous avons utilisée pour estimer les paramètres 3D d'une primitive géométrique basée sur l'utilisation d'une caméra mobile commandable. La seconde partie est dédiée aux aspects globaux de notre processus de reconstruction et présente des stratégies autonomes d'exploration.

## 2 Reconstruction de primitives 3D

La méthode utilisée pour reconstruire les primitives 3D est détaillée dans [5] et [6]. Elle permet d'obtenir une estimation précise et robuste des paramètres 3D d'une primitive géométrique à partir de l'analyse d'une séquence d'images acquises par une caméra en mouvement.

### 2.1 Reconstruction 3D par vision dynamique

La caméra est modélisée de manière classique par une projection perspective. Sans perte de généralité, la distance focale de la caméra est fixée égale à 1. Un point  $m$  de coordonnées  $\underline{x} = (x, y, z)^T$  se projette donc en  $M$  de coordonnées  $\underline{X} = (X, Y, 1)^T$  avec :

$$\underline{X} = \frac{1}{z} \underline{x} \quad (1)$$

Soit  $\mathcal{P}_s$  une primitive géométrique paramétrable décrite par une équation de la forme :

$$h(\underline{x}, \underline{p}) = 0, \forall \underline{x} \in \mathcal{P}_s \quad (2)$$

où  $h$  définit la nature de la primitive et  $\underline{p}$  sa configuration. Le but de la reconstruction est d'estimer la valeur des paramètres  $\underline{p}$  afin de pouvoir reconstruire et localiser la primitive  $h$ . À partir de cette équation paramétrique, on peut définir les deux fonctions suivantes [10]:

$$\begin{cases} g(\underline{X}, \underline{P}) = 0, \forall \underline{X} \in \mathcal{P}_i \\ 1/z = \mu(\underline{X}, \underline{p}_0) \end{cases} \quad (3)$$

où  $\mathcal{P}_i$  représente la projection dans le plan image de la primitive  $\mathcal{P}_s$ . Pour des primitives planes, la fonction  $\mu$  représente le plan dans lequel la primitive se situe. Dans le cas plus général de primitives volumétriques (voir Figure 1), la fonction  $g$  représente la projection dans l'image des limbes de la primitive et la fonction  $\mu$

définit la surface 3D à laquelle appartiennent les limbes ( $\mu$  exprime la relation entre les points de  $g$  et les points 3D correspondants). Soit  $T_c = (V(O), \Omega)^T$  le torseur ci-

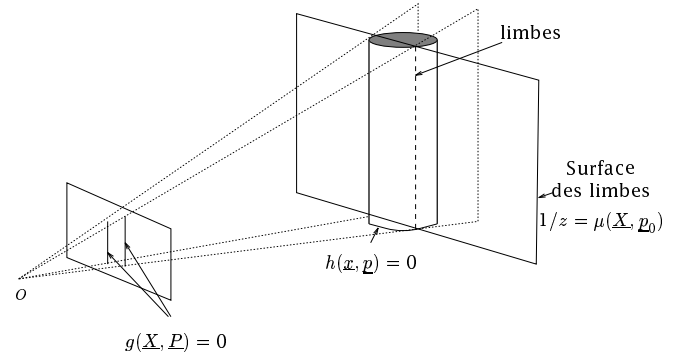


FIG. 1 - Projection de la primitive dans l'image ( $g$ ) et surface des limbes ( $\mu$ ) dans le cas d'un cylindre

nématique de la caméra où  $V(O)$  représente la vitesse de translation de la caméra et  $\Omega$  sa vitesse de rotation. La variation de  $\underline{P}$  qui relie le mouvement apparent de la primitive dans l'image au mouvement de la caméra  $T_c$  peut être calculée explicitement et s'exprime par :

$$\dot{\underline{P}} = L_{\underline{P}}^T(\underline{P}, \underline{p}_0) T_c \quad (4)$$

où  $L_{\underline{P}}^T(\underline{P}, \underline{p}_0)$  représente la matrice d'interaction qui caractérise les interactions entre le capteur et la primitive considérée [10].

L'estimation des paramètres  $\underline{p}$  s'effectue en deux étapes : tout d'abord, les paramètres  $\underline{p}_0$  caractérisant la surface des limbes  $\mu$  sont obtenus à partir de l'équation (4) en utilisant la mesure de  $T_c$ ,  $\underline{P}$  et  $\dot{\underline{P}}$  :

$$\underline{p}_0 = \underline{p}_0(T, \underline{P}, \dot{\underline{P}}) \quad (5)$$

Grâce aux contraintes géométriques sur le type de la primitive à reconstruire (droite, cylindre,...) caractérisées par l'équation  $h(\underline{x}, \underline{p}) = 0$ , la mesure de  $\underline{P}$  dans l'image permet alors de remonter, par intersection du cône de centre  $O$  et de génératrice  $g(\underline{X}, \underline{P}) = 0$  avec la surface des limbes, aux paramètres  $\underline{p}$  qui caractérisent la primitive  $h$  étudiée :

$$\underline{p} = \underline{p}(\underline{p}_0, \underline{P}). \quad (6)$$

**Exemple : le cas de la droite.** Une droite peut être représentée par l'intersection de deux plans orthogonaux :

$$h(\underline{x}, \underline{p}) = \begin{cases} a_1 x + b_1 y + c_1 z = 0 \\ a_2 x + b_2 y + c_2 z + d_2 = 0 \end{cases} \quad (7)$$

$$\text{avec } \begin{cases} a_1^2 + b_1^2 + c_1^2 = 1 \\ a_2^2 + b_2^2 + c_2^2 = 1 \\ a_1 a_2 + b_1 b_2 + c_1 c_2 = 0. \end{cases} \quad (8)$$

Une représentation minimale et complète de la droite 2D correspondante est donnée par :

$$g(X, Y, \rho, \theta) = X \cos \theta + Y \sin \theta - \rho = 0 \quad (9)$$

$$\text{avec } \begin{cases} \cos \theta = a_1 / \sqrt{a_1^2 + b_1^2} \\ \sin \theta = b_1 / \sqrt{a_1^2 + b_1^2} \\ \rho = -c_1 / \sqrt{a_1^2 + b_1^2} \end{cases} \quad (10)$$

De plus, la fonction  $\mu$  définie par (3) est obtenue à partir de (8) de la manière suivante :

$$1/z = \mu(X, Y, a, b, c) = aX + bY + c \quad (11)$$

où  $a = -a_2/d_2$ ,  $b = -b_2/d_2$  et  $c = -c_2/d_2$ . Enfin, la relation entre le mouvement de la droite dans l'image (définie par  $(\dot{\rho}, \dot{\theta})$ ) et la vitesse de la caméra  $T_c$  est donnée par la matrice d'interaction associée aux paramètres  $(\rho, \theta)$  [10]:

$$\begin{pmatrix} \dot{\rho} \\ \dot{\theta} \end{pmatrix} = L_P^T T_c \quad (12)$$

avec

$$L_P^T = \begin{pmatrix} \lambda_\rho \cos \theta & \lambda_\rho \sin \theta & -\lambda_\rho \rho & \alpha \sin \theta & -\alpha \cos \theta & 0 \\ \lambda_\theta \cos \theta & \lambda_\theta \sin \theta & -\lambda_\theta \rho & -\rho \cos \theta & -\rho \sin \theta & -1 \end{pmatrix} \quad (13)$$

et où  $\alpha = (1 + \rho^2)$ ,  $\lambda_\rho = -a\rho \cos \theta - b\rho \sin \theta - c$ , et  $\lambda_\theta = -a \sin \theta + b \cos \theta$ .

A partir des valeurs mesurées de  $\rho, \theta, \dot{\rho}, \dot{\theta}$  et  $T_c$ , il faut maintenant estimer les paramètres des deux plans qui définissent la droite considérée.

Tout d'abord, les paramètres  $a_1, b_1$  et  $c_1$  sont immédiatement déduits de  $\rho$  et  $\theta$ . Ensuite, comme décrit précédemment, les paramètres décrivant la fonction  $\mu$  sont déterminés en utilisant la mesure de la vitesse de la caméra et la vitesse apparente de la droite dans l'image qui en résulte. Ainsi, les paramètres  $a, b, c$  sont donnés par la résolution du système linéaire suivant :

$$\begin{cases} -a\rho \cos \theta - b\rho \sin \theta - c = \lambda_\rho \\ -a \sin \theta + b \cos \theta = \lambda_\theta \\ a \cos \theta + b \sin \theta - c\rho = 0 \end{cases} \quad (14)$$

où  $\lambda_\rho$  et  $\lambda_\theta$  sont obtenus à partir de (12):

$$\begin{cases} \lambda_\rho = \frac{\dot{\rho} + (1 + \rho^2)(\Omega_Y \cos \theta - \Omega_X \sin \theta)}{V_X \cos \theta + V_Y \sin \theta - \rho V_Z} \\ \lambda_\theta = \frac{\dot{\theta} + \rho(\Omega_X \cos \theta + \Omega_Y \sin \theta) + \Omega_Z}{V_X \cos \theta + V_Y \sin \theta - \rho V_Z} \end{cases} \quad (15)$$

On obtient finalement  $d_2 = 1/\sqrt{a^2 + b^2 + c^2}$ ,  $a_2 = -ad_2$ ,  $b_2 = -bd_2$  et  $c_2 = -cd_2$ .

Des résultats similaires peuvent être obtenus pour d'autres primitives géométriques paramétrables. Le cas des points, cylindres, cercles et sphères est par exemple décrit dans [5].

## 2.2 Reconstruction 3D par vision active

Les résultats obtenus en utilisant la méthode décrite précédemment sont généralement assez médiocres. En effet, la qualité de l'estimation est très sensible à la nature des mouvements de la caméra [11]. Une solution efficace pour résoudre ce problème est d'utiliser le formalisme de la vision active [3][4]. Contrairement à la

vision dynamique où l'on se contente d'observer le mouvement de la caméra, dans le cas de la vision active, et par analogie avec la vision humaine, le mouvement est commandé. L'asservissement du capteur avec les résultats issus des informations présentes dans les images fait de la vision active un processus de perception intelligente des données. Le principal apport de la vision active est l'inclusion de l'asservissement dans le système ainsi que la collecte des données nécessaires à cet asservissement. On verra que le modèle de l'asservissement visuel [10] est parfaitement adapté à ce formalisme.

On peut montrer [5][6] que, pour obtenir une estimation robuste et non biaisée, il est nécessaire que la caméra réalise une tâche de fixation et une tâche de focalisation. Plus précisément, l'image et la surface des limbes de la primitive doivent rester constantes le long de la trajectoire effectuée par la caméra pour obtenir une estimation non biaisée (fixation) et la primitive doit rester immobile à des positions particulières dans l'image pour obtenir une estimation robuste (focalisation). Si l'on considère le cas du point par exemple, celui-ci doit apparaître en permanence centré dans l'image (*i.e.*  $\dot{X} = \dot{Y} = X = Y = 0, \forall t$ . On doit également assurer  $\dot{z} = 0, \forall t$ ).

**Position spatiale d'une primitive le long de son axe** Le processus de reconstruction optimale décrit précédemment considère que les primitives ont une longueur infinie. De manière à déterminer précisément la position 3D des extrémités de la primitive, celles-ci doivent être successivement observées au centre de l'image, ce qui, dans le cas général, nécessite un mouvement de la caméra. Quand la caméra a atteint la position désirée, la position du point 3D correspondant est calculée par l'intersection de l'axe de la primitive avec l'axe optique de la caméra [15].

## 2.3 Commande de la caméra par asservissement visuel

Les techniques d'asservissement visuel [10] permettent d'effectuer automatiquement les différents mouvements satisfaisant les contraintes nécessaires à une estimation optimale (tâche de fixation, de focalisation, et observation des extrémités des primitives).

L'asservissement visuel consiste en effet à introduire directement et en boucle fermée les informations extraites de l'image dans une boucle de commande [10][14]. Par opposition avec l'asservissement en position classique, le problème est ici spécifié en termes de régulation dans l'image et ne nécessite pas la reconstruction tridimensionnelle de la scène à chaque itération de la boucle de commande.

On peut parfaitement intégrer les techniques d'asservissement visuel dans l'approche fonction de tâche [18]. En effet, les tâches référencées vision peuvent s'exprimer comme la régulation à zéro d'une fonction de tâche combinant une tâche primaire réalisée à chaque instant (dans notre cas, il faut que la primitive apparaisse dans

l'image à la position spécifiée par la tâche de focalisation) et une tâche secondaire (ici un suivi de trajectoire) construite afin de permettre des mouvements de la caméra tout en assurant la contrainte relative à la tâche primaire.

## 2.4 Résultats: reconstruction optimale dans le cas d'un cylindre

Les expérimentations présentées dans cet article ont été réalisées sur la cellule de vision robotique de l'IRISA composée d'une caméra CCD montée sur l'effecteur d'un robot à 6 degrés de liberté. Dans un premier temps, nous avons utilisé la méthode présentée pour estimer les paramètres d'un cylindre (cf figure 2.a). De manière à obtenir une estimation robuste et non biaisée des paramètres, le cylindre doit apparaître centré et vertical ou horizontal dans l'image (cf figure 2.b) pendant le mouvement de la caméra qui consiste en une rotation à distance constante de l'axe du cylindre. Les techniques d'asservissement visuel ont donc été employées pour réaliser cette tâche en temps réel: une itération de la boucle de commande et une estimation sont simultanément effectuées en 100 ms. La figure 3.a montre l'erreur entre la valeur estimée du rayon et sa valeur réelle (*i.e.*,  $r_i - r^*$ ) obtenue en utilisant une estimation basée sur les deux limbes du cylindre.

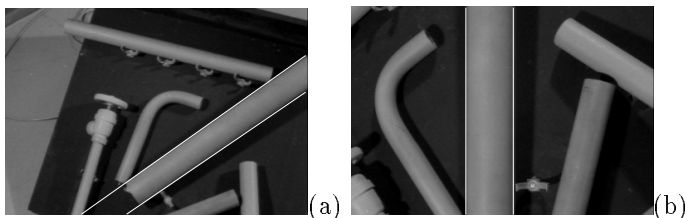


FIG. 2 - Cylindre à reconstruire avant et après la tâche de focalisation

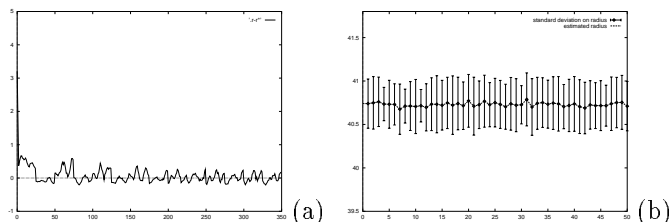


FIG. 3 - (a) estimation du rayon  $r_i - r^*$  (en mm) (b) tests de robustesse

Des tests de robustesse ont également été réalisés. Les paramètres du cylindre ont été estimés 50 fois en partant de positions initiales différentes. Les résultats reportés figure 3.b montrent pour chacune des 50 estimations le rayon  $\hat{r}$  estimé ainsi que l'écart type  $\sigma_{\hat{r}}$  sur cette estimation. Pour chaque estimation, l'erreur entre la valeur estimée et la valeur réelle est inférieure à 0.1 mm et l'écart type sur l'ensemble des moyennes est inférieur à 0.02 mm. Ceci démontre que l'algorithme de reconstruction est robuste, stable et précis.

## 3 Stratégies de perception

Le problème qui nous intéresse maintenant est celui de la reconstruction complète d'une scène contenant plusieurs objets. En effet, si l'approche retenue pour la reconstruction permet une estimation très précise et robuste des paramètres des primitives concernées, elle ne permet cependant de reconstruire qu'une seule primitive à la fois. L'objectif de cette partie est donc de définir des stratégies de perception afin de permettre une représentation 3D précise et complète de la scène. Des phases d'exploration sont notamment nécessaires afin d'assurer la complétude de la reconstruction. De manière schématique, l'approche utilisée consiste à sélectionner automatiquement les informations images pertinentes puis à focaliser successivement la caméra sur les différents objets de la scène afin de les reconnaître et ensuite de les reconstruire.

### 3.1 Un test statistique pour la reconnaissance de primitives

La seule hypothèse effectuée sur la scène, outre les dimensions d'un volume l'englobant, porte sur le fait qu'elle est constituée uniquement de segments et de cylindres. La méthode de reconstruction présentée ci-dessus implique une connaissance *a priori* sur la nature de la primitive observée (segment ou cylindre). Un processus de reconnaissance est donc nécessaire. Pour ce faire, nous supposons dans un premier temps qu'il s'agit d'un cylindre et une estimation des paramètres de la primitive (en particulier de son rayon  $r$ ) basée sur une seule limbe est réalisée. Quand cette estimation est effectuée deux hypothèses sont considérées :

$H_1$  : la primitive observée était en fait un segment.

$H_2$  : la primitive observée était effectivement un cylindre de rayon  $r_1$  ;

Un test au maximum de vraisemblance est alors utilisé pour déterminer laquelle de ces deux hypothèses est correcte. Notons  $L_0$  et  $L_1$  les fonctions de vraisemblance associées aux hypothèses  $H_0$  et  $H_1$ . En supposant que les estimations du rayon du cylindre suivent une loi Gaussienne de moyenne  $r$  et de variance  $\sigma^2$ , nous obtenons :

$$L_0 = \left(\frac{1}{2\pi\sigma^2}\right)^{\frac{N}{2}} e^{-\frac{\sum_{i=1}^N r_i^2}{2\sigma^2}}, \quad L_1 = \left(\frac{1}{2\pi\sigma^2}\right)^{\frac{N}{2}} e^{-\frac{\sum_{i=1}^N (r_i - r_1)^2}{2\sigma^2}} \quad (16)$$

Le choix de l'hypothèse  $H_1$  contre l'hypothèse  $H_0$  est basé sur l'optimisation du rapport de vraisemblance  $\xi$  donné par  $\xi = \log \frac{L_1}{L_0}$ . En substituant dans cette équation les expressions données dans (16), on arrive à :

$$\xi = -\frac{1}{2\sigma^2} \left( \sum_{i=1}^N (r_i - r_1)^2 - \sum_{i=1}^N r_i^2 \right) \quad (17)$$

Le critère pour déterminer la nature de la primitive peut alors être défini par :

$$\max_{r_1} \xi \geq \lambda$$

où  $\lambda$  est un seuil prédéterminé.

Le paramètre optimal  $\hat{r}_1$  doit satisfaire la relation  $\frac{\partial \xi}{\partial r_1} = 0$ , ce qui amène à  $\hat{r}_1 = \bar{r}$ . En utilisant cette relation,  $\xi$  peut finalement s'exprimer de la manière suivante :

$$\xi = \frac{N \bar{r}^2}{2\sigma^2} \quad (18)$$

L'hypothèse  $H_1$  (cylindre) est donc retenue contre l'hypothèse  $H_0$  (segment) si la valeur obtenue du rapport de vraisemblance  $\xi$  est plus grande que  $\lambda$ . En fait, quand la primitive est un segment, le processus de reconstruction d'un cylindre donne une valeur très faible du rayon, avec un écart type assez élevé (ce qui implique une valeur petite de  $\xi$ ). *A contrario*, quand la primitive est réellement un cylindre, la valeur estimée du rayon est très proche de sa valeur réelle avec une faible variance (ce qui implique une valeur importante de  $\xi$ ). Si la primitive est reconnue comme étant un cylindre, une estimation de ces paramètres basée sur les deux limbes est ensuite réalisée.

### 3.2 Bases de données 2D

À part les hypothèses générales de base sur les dimensions maximales de la scène et la nature des objets (segments et cylindres), aucune information supplémentaire sur la localisation ou le nombre d'objets présents n'est disponible. Les seules données dont l'on dispose sont celles fournies par les images 2D acquises par la caméra. Les stratégies de perception proposées sont principalement basées sur l'utilisation de ces informations 2D. L'une des étapes fondamentales de notre algorithme est donc la création de bases de données répertoriant ces informations. Les bases de données principalement employées, notées  $\omega_{\phi_t}$  (où  $\phi_t$  représente la position de la caméra), contiendront la liste des segments correspondant à la projection des objets de la scène visibles dans l'image depuis la position  $\phi_t$ .

Les algorithmes de traitement d'images que nous utilisons pendant les phases d'asservissement visuel et de reconstruction n'autorisent le suivi en temps réel que d'un petit nombre de segments. De ce fait, et afin d'assurer la contrainte temps réel, ces bases de données ne peuvent être créées à chaque itération du processus d'estimation, mais seulement à la fin de chaque processus de reconstruction d'une primitive.

D'autre part, une seconde base de données, notée  $\Omega_{\Phi_t}$ , est également utilisée.  $\Omega_{\Phi_t}$  contient tous les segments qui n'ont pas encore fait l'objet d'une reconstruction à l'instant  $t$ , ainsi que la position de la caméra depuis laquelle ils ont été observés.

### 3.3 Reconstruction incrémentale

Depuis un point de vue  $\phi_t$  et en utilisant les informations 3D déjà recueillies il est possible de calculer la zone observée  $V(\phi_t)$ . Notons  $\mathcal{V}(\Phi_t)$  la zone de l'espace observée par la caméra depuis le début du processus de reconstruction (*i.e.* les primitives 3D et l'espace libre connus). On a :

$$\mathcal{V}(\Phi_t) = \bigcup_{i=1}^t V(\phi_i) \text{ avec } \Phi_t = \bigcup_{i=1}^t \phi_i$$

En utilisant les informations observées (*i.e.* la bases de données  $\Omega_{\Phi_t}$  et la carte partielle de l'environnement), nous pouvons alors définir des stratégies de positionnement de la caméra qui assureront une reconstruction complète de la scène. Ce processus d'exploration est constitué de deux niveaux distincts :

- une **exploration locale** est réalisée si un segment correspondant à une nouvelle primitive apparaît dans le champ de vision de la caméra ou si un tel segment a été précédemment observé depuis une autre position de la caméra. Dans ce cas, un calcul explicite de point de vue n'est pas nécessaire.
- En revanche, dans les autres cas, quand tous les segments précédemment observés ont été traités, une stratégie plus complexe doit être mise en œuvre de manière à focaliser la caméra sur des zones de la scène n'ayant pas encore été observées. Nous parlerons alors d'**exploration globale**.

### 3.4 Exploration locale

Afin de minimiser le déplacement de la caméra, il apparaît souhaitable de mettre en œuvre une stratégie *locale* utilisant de façon explicite la connaissance courante sur la scène [15]. Cette stratégie est pilotée par des événements détectés dans l'image et par les informations 3D précédemment acquises.

La démarche s'appuie sur les hypothèses suivantes : la scène est constituée d'objets 3D liés par des relations topologiques ; la projection des objets 3D dans l'image peut être représentée par un graphe où les nœuds sont les jonctions multiples et les arcs les contours. Chaque arc de ce graphe correspond à un segment 2D de la base de données courante  $\omega_{\phi}$ . Ces arcs sont valués en fonction de la connaissance éventuelle que l'on a sur la primitive 3D correspondante (si il correspond à la projection dans l'image d'une primitive déjà reconstruite ou non, voir Figure 4) et de sa position dans l'image. Si plusieurs arcs de ce graphe se révèlent être la projection de primitives non reconstruites, un choix doit être effectué pour savoir lequel reconstruire. Ce choix qui permet de minimiser la distance parcourue par la caméra, dépend de la valeur associée à l'arc, ainsi que des relations de voisinage avec le segment courant (le dernier reconstruit et sur lequel la caméra est focalisée) [15].

Dans le cas où tous les arcs du graphe correspondent à des segments qui ont déjà été reconstruits, nous re-

cherchons, dans la base de données globale  $\Omega_{\Phi_t}$ , un segment non traité. La caméra se déplace jusqu'à la position depuis laquelle il avait été observé (phase de retour arrière), et une reconstruction de la primitive associée à ce segment est alors effectuée. Quand cette base de donnée  $\Omega_{\Phi_t}$  est vide, une exploration globale est nécessaire.

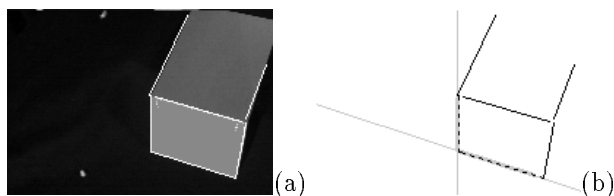


FIG. 4 - (a) Image acquise (b) Base de données 2D (segments non traités et segments mis en correspondance avec la scène reconstruite)

La stratégie *locale* que nous avons développée assure une reconstruction efficace de toute primitive ayant été observée lors du processus de reconstruction. Elle ne fait pas appel à un calcul explicite de nouveaux points de vue et minimise localement le déplacement de la caméra. Cependant, elle ne donne pas l'assurance d'une reconstruction complète de la scène. Pour résoudre ce problème, des stratégies *globales* doivent être mises en œuvre. Quand toutes les primitives observées pendant les phases d'exploration locale ont été reconstruites, de nouveaux points de vue doivent donc être calculés de manière à obtenir un maximum d'informations supplémentaires sur la scène.

### 3.5 Exploration globale : calcul de points de vue

La plupart des travaux antérieurs traitant du problème de placement de caméra supposent connu un modèle complet de la scène [9][20]. Cowan et Kovesi [9] se proposent de sélectionner automatiquement les points de vue permettant d'observer une ou plusieurs faces d'un objet. Ils définissent une série de contraintes sur les paramètres de la caméra qui permettent la réalisation de la tâche visuelle. Le problème traité par Tarabnis, Tsai et Allen dans [20] repose sur le même principe mais le système optique est plus général et permet d'obtenir un point de vue généralisé dans un espace à 8 dimensions. Une fonction d'optimisation permet de caractériser la qualité de la solution calculée en se basant sur les mesures liées aux contraintes. Dans le cas d'un environnement partiellement connu ou dynamique, Al Chami et Laugier [2] ont proposé une méthode pour déterminer automatiquement les positions d'une caméra embarquée sur un robot, afin de réaliser un ensemble d'opérations sensorielles qui seront utilisées par la suite pour construire des représentations de l'environnement. Cette méthode consiste à rechercher un positionnement de la caméra qui minimise une fonction d'énergie dont les paramètres représentent les contraintes imposées par la tâche et par l'environnement. L'optimisation de la

fonction d'énergie est réalisée par un processus de recuit simulé. Cette méthode est reprise par Triggs et Laugier dans [21] mais, ici, l'optimisation de la fonction d'énergie repose sur une méthode de recherche probabiliste de l'optimum de la fonction d'évaluation. Les contraintes prises en compte sont de type optique, géométrique et mécanique (accessibilité). Le problème est cependant différent si les informations sur la scène sont incomplètes ou nulles (c'est-à-dire si le capteur évolue dans un environnement inconnu). Cela revient à effectuer une tâche d'exploration autonome. De nombreux articles [8][25][17] traitent ce problème sous des angles différents. Connolly [8] se propose, en utilisant un capteur laser, de déterminer un modèle complet de la scène à partir d'un ensemble de points de vue. Il décrit l'algorithme dit du *planétarium* qui utilise une décomposition en *oct-tree* de la scène. La caméra se déplace sur la surface d'une sphère régulièrement échantillonnée et circonscrite à la scène. Le point de vue permettant de dévoiler la zone non vue la plus importante est sélectionné. Maver et Bajcsy [17] utilisent les occlusions pour déterminer les différents points de vue nécessaires pour acquérir les informations 3D des parties cachées de l'environnement. Le capteur est composé d'une caméra et d'un plan laser. Dans [25], Wixson décrit des stratégies pour rechercher un objet connu dans une zone encombrée. Il explore différentes stratégies pour l'exploration d'un monde 2D à partir d'un capteur 1D. Il propose des stratégies basées sur l'algorithme du *planétarium* [8]. Deux variantes sont proposées : la première basée sur la recherche du point de vue offrant le maximum de visibilité des zones encore inconnues, la deuxième favorisant le point de vue qui minimise le déplacement du capteur. Une autre stratégie étudiée est basée sur l'utilisation des arêtes d'occlusion. L'intérêt principal de cette méthode est l'évaluation des coûts et des bénéfices des opérations de perception effectuées.

Contrairement aux approches précédentes qui utilisent des hypothèses sur la structure de la scène ou qui sont basées sur l'utilisation d'autres types de capteurs qu'une caméra (laser), notre objectif est de reconstruire tous les objets de la scène sans connaissance *a priori* sur leur nombre, leur position et leur dimension. Nous devons donc déterminer les points de vue permettant d'amener de nouvelles primitives dans le champ de vision de la caméra. De tel points de vue sont calculés en utilisant la connaissance courante sur la géométrie spatiale de la scène.

**Sélection de points de vue** La recherche du point de vue s'effectue en minimisant une fonction d'énergie  $\mathcal{F}(\phi_{t+1})$  qui doit respecter les contraintes imposées au système et représenter le niveau de satisfaction apporté par le point de vue sélectionné. Cette fonction intègre un ensemble de mesures indépendantes à valeur dans  $R^+$  représentant le gain attendu par une position  $\phi_{t+1}$ , le coût résultant du déplacement vers cette position, ainsi que des contraintes mécaniques dues aux butées articulaires

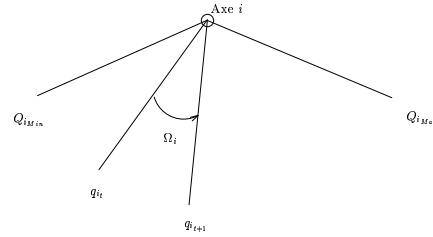
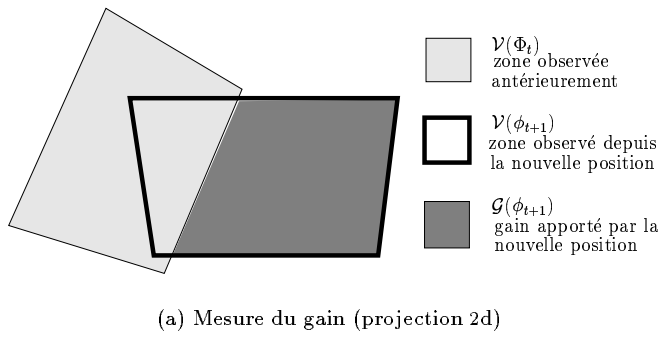


FIG. 5 - Calcul de la fonction de gain et de coût

du robot.

- Le gain apporté par une nouvelle position  $\phi_{t+1}$  est défini par le volume de la zone inobservée qui apparaît dans le cône de vision de la caméra quand celle-ci se déplace de  $\phi_t$  à  $\phi_{t+1}$ . La zone découverte depuis cette position correspond à la zone  $\mathcal{G}(\phi_{t+1})$  définie par (voir Figure 5.a) :

$$\mathcal{G}(\phi_{t+1}) = \mathcal{V}(\phi_{t+1}) - \mathcal{V}(\phi_{t+1}) \cap \mathcal{V}(\Phi_t) \quad (19)$$

La mesure du gain associé à la position  $\phi_{t+1}$  est définie à partir de la surface au sol  $\mathcal{S}$  de  $\mathcal{G}(\phi_{t+1})$  et  $\mathcal{V}(\phi_{t+1})$ <sup>1</sup> :

$$g(\phi_{t+1}) = 1 - \frac{\mathcal{S}(\mathcal{G}(\phi_{t+1}))}{\mathcal{S}(\mathcal{V}(\phi_{t+1}))} \quad (20)$$

**Remarque :** La zone  $\mathcal{G}(\phi_t)$  correspond en fait à une zone potentiellement découverte. En effet, dans le cas où un nouvel objet apparaît dans le champ de vision, les occlusions dues à cet objet font que la zone réellement observée est de taille inférieure ( $\mathcal{G}'(\phi_t) \subseteq \mathcal{G}(\phi_t)$ ).

- La mesure du coût du déplacement entre deux positions  $\phi_t$  et  $\phi_{t+1}$  est définie en calculant la distance entre les deux positions sur chacun des axes du robot. Plus précisément, ce coût est donné par (voir Figure 5.b) :

$$\mathcal{C}(\phi_t, \phi_{t+1}) = \frac{1}{N_{ddl}} \sum_{i=1}^{N_{ddl}} \beta_i \frac{|q_{i_t} - q_{i_{t+1}}|}{|Q_{i_{Max}} - Q_{i_{Min}}|} \quad (21)$$

où

- $N_{ddl}$  représente le nombre de degrés de liberté du robot ;
- $q_i$  représente la coordonnée articulaire de l'axe  $i$  ( $\phi = (q_0, q_1, \dots, q_{N_{ddl}})$ ).
- $|Q_{i_{Max}} - Q_{i_{Min}}|$  est la distance entre les deux butées articulaires de l'axe  $i$ .

1. Un calcul en fonction du volume de la zone observée est bien sur possible et a également été mis en œuvre

- De plus, des contraintes supplémentaires sont associées à certaines positions du robot. Ces contraintes tendent à :

- éviter les positions inatteignables pour la caméra en raison des butées articulaires du robot. Une pénalité infinie est donnée à une position si celle-ci n'est pas accessible :

$$\mathcal{A}(\phi) = \begin{cases} 0 & \text{si } \phi \text{ est accessible} \\ \infty & \text{sinon} \end{cases} \quad (22)$$

- éviter des positions trop proches des butées articulaires :

$$\mathcal{B}(\phi) = \frac{1}{N_{ddl}} \sum_{i=1}^{N_{ddl}} \frac{4(q_i - \frac{Q_{i_{Max}} + Q_{i_{Min}}}{2})^2}{(Q_{i_{Max}} - Q_{i_{Min}})^2} \quad (23)$$

La mesure associée à cette contrainte sera optimale (égale à 0) si les coordonnées articulaires  $q_i$  se trouvent au milieu de l'axe d'extension des butées.

La fonction d'énergie  $\mathcal{F}(\phi_{t+1})$  est définie par la somme pondérée de ces différentes mesures :

$$\mathcal{F}(\phi_{t+1}) = \mathcal{A}(\phi) + \alpha_1 g(\phi_{t+1}) + \alpha_2 \mathcal{C}(\phi_t, \phi_{t+1}) + \alpha_3 \mathcal{B}(\phi) \quad (24)$$

La détermination des coefficients  $\alpha_i$  dans un problème d'optimisation de ce type est un problème non trivial [13]. Nous nous sommes contentés de choisir ces coefficients de manière empirique. Cependant, leur valeur fixe l'ordre de priorité associé à chacun des critères. L'accessibilité étant bien sûr prioritaire et la découverte de nouvelles zones à explorer étant notre objectif, nous avons choisi  $\alpha_1 > \alpha_2 > \alpha_3$ .

**Optimisation de la fonction de choix** Chaque position  $\phi \in SE_3$  peut *a priori* être solution de ce problème d'optimisation. Cependant, de manière à contraindre le problème, nous autorisons la caméra à se déplacer sur la surface d'une sphère circonscrite à la scène. La position de la caméra peut alors être décrite par un vecteur à 5 paramètres  $(\theta, \varphi, \Omega_x, \Omega_y, \Omega_z)$  où  $\theta$  et  $\varphi$  représentent la latitude et la longitude de la caméra sur la sphère et où  $\Omega_x, \Omega_y$  et  $\Omega_z$  représentent l'orientation de la caméra.

Pour minimiser  $\mathcal{F}(\phi)$ , nous avons choisi d'utiliser une méthode déterministe classique de type gradient conjugué avec un pas de descente à plusieurs niveaux : nous utilisons tout d'abord des incréments importants afin de déterminer la région de l'espace des paramètres où l'optimum de la fonction  $\mathcal{F}(\phi)$  est probablement situé. Puis, nous itérons le processus depuis cette nouvelle position avec un incrément plus faible. Contrairement aux méthodes stochastiques de type recuit simulé, nous ne pouvons assurer que la convergence s'effectue vers le minimum global de la fonction. Cependant, le gain en temps de calcul est très important et les expériences ont montré qu'un optimum correct est toujours atteint en un faible nombre d'itérations. De plus, l'intérêt de trouver un minimum global de la fonction d'énergie ne nous a pas paru fondamental dans la mesure où une position apportant un complément d'informations important est trouvée.

### 3.6 Contrôle de tâches

L'approche utilisée pour la reconstruction ne permet de reconstruire qu'une primitive à la fois, un séquençement de sous-tâches d'asservissement visuel est donc nécessaire pour passer de la reconstruction d'une primitive à une autre, et ainsi obtenir, par construction incrémentale, une représentation complète de la scène. Nous nous sommes intéressés à la conception d'un automate hiérarchique parallèle capable de gérer l'ensemble du processus de reconstruction d'une scène complexe. Cet automate sélectionne et gère les actions à effectuer en fonction des événements 2D perçus dans l'image, des connaissances acquises et répertoriées au fur et à mesure de la reconstruction, ainsi que de la détection de la fin d'une tâche d'asservissement visuel (on retrouve ici l'aspect *événementiel*). La mise en œuvre de cet automate a été réalisée avec le langage synchrone SIGNAL [16].

### 3.7 Résultats : stratégies de perception

La scène considérée est composée d'un cylindre et de plusieurs polygones disposés dans des plans différents. L'image 6 montre une vue extérieure de la scène et des différents objets qui la composent.

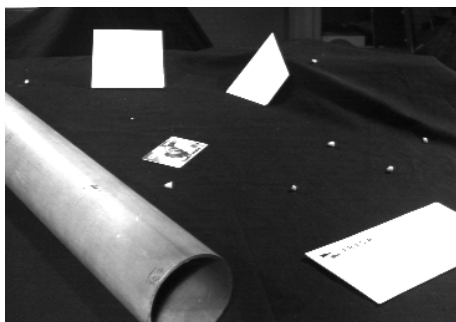


FIG. 6 - Vue extérieure de la scène

**Exploration locale.** La figure 7 représente les images acquises avant chaque reconstruction optimale. À cha-

cune de ces images est associée la base de données 2D correspondante. Les lignes pleines montrent les éléments de la base de données qui n'ont pas encore été traités. Les lignes en pointillés représentent les segments correspondant à des primitives 3D déjà reconstruites. La

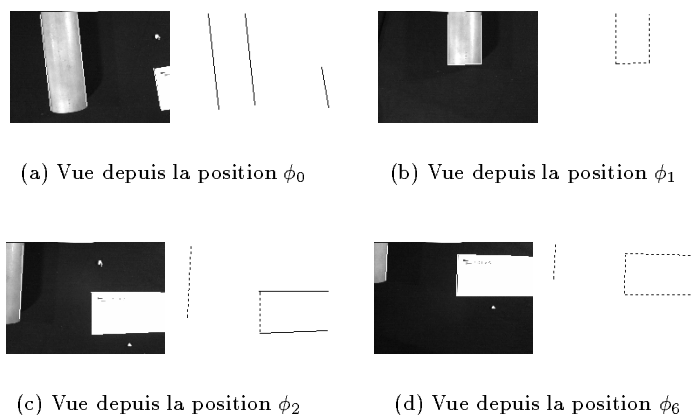


FIG. 7 - Exploration locale de la scène

figure 7.a montre l'image acquise depuis la position  $\phi_0$  de la caméra. Aucune reconstruction n'a encore été effectuée et trois segments seulement sont visibles depuis cette position. On peut noter que la scène complète n'est pas visible depuis cette position de la caméra. Le segment extrait de la base de données  $\omega_{\phi_0}$  est celui correspondant au limbe de droite du cylindre. Après la phase de reconnaissance et de reconstruction du cylindre, la caméra est positionnée en  $\phi_1$  (image 7.b). Les segments de la base de données  $\omega_{\phi_1}$  ont tous été traités. Après consultation de la base de données globale  $\Omega_{\Phi}$ , on constate qu'un segment a été observé depuis la position  $\phi_0$  et n'a pas encore été reconstruit. La caméra se déplace donc en  $\phi_0$  et se focalise sur le segment retenu. Après l'estimation de cette primitive, la caméra est positionnée en  $\phi_2$  (image 7.c). Deux segments correspondant à des

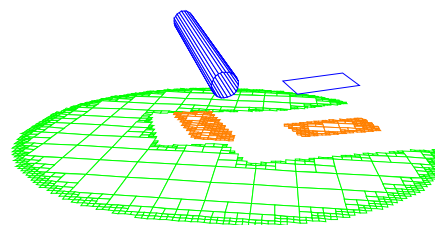


FIG. 8 - Résultat de l'exploration locale (positions de la caméra, scène reconstruite, et projection sur un plan fictif de la zone non observée)

primitives non reconstruites apparaissent dans la base de données  $\omega_{\phi_2}$ . Le segment le plus proche du centre de l'image est sélectionné et reconstruit. Ce processus est renouvelé jusqu'à ce que toutes les primitives observées pendant cette phase d'exploration locale aient été estimées (image 7.d correspondant à la position  $\phi_6$  de la caméra). Notons que des primitives qui n'apparaissent pas dans le champ de vision initial de la caméra



ont été découvertes et reconstruites. La scène estimée à ce stade du processus de reconstruction est présentée sur la figure 8.

**Exploration globale.** La Figure 9 montre les différentes étapes nécessaires à l’exploration complète de la scène. Chacune des figures décrit la scène reconstruite, la trajectoire de la caméra jusqu’à sa position courante, et une visualisation de la zone non observée. La Figure 9.a correspond à la position  $\phi_6$  de la caméra obtenue à l’issue de l’exploration locale décrite dans le paragraphe précédent. Les premiers déplacements de la caméra (voir Figure 9.b et 9.c) permettent de diminuer la zone de la scène non encore explorée. À la position  $\phi_{13}$ , représentée

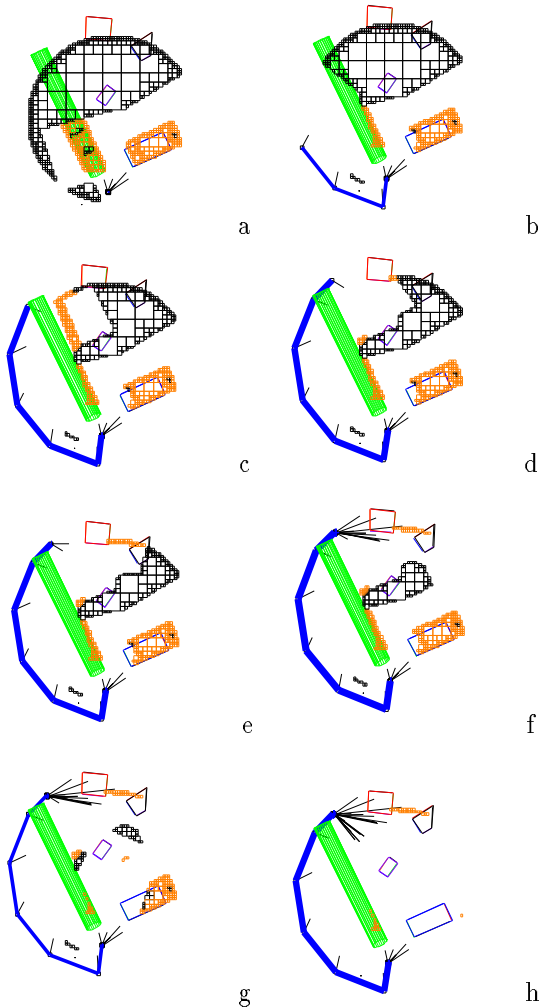


FIG. 9 - Différentes étapes de l’exploration globale (trajectoire de la caméra, scène finale reconstruite, et projection sur un plan fictif de la zone non observée)

Figure 9.d, une nouvelle primitive est détectée marquant le début d’une seconde phase d’exploration locale qui se termine à l’étape 24 (Figure 9.f). Les deux polygones en haut de la scène sont alors reconstruits. Une nouvelle exploration globale amène la caméra en  $\phi_{25}$  (Figure 9.g) où un segment appartenant au dernière objet (une carte de téléphone) apparaît. Après une dernière exploration

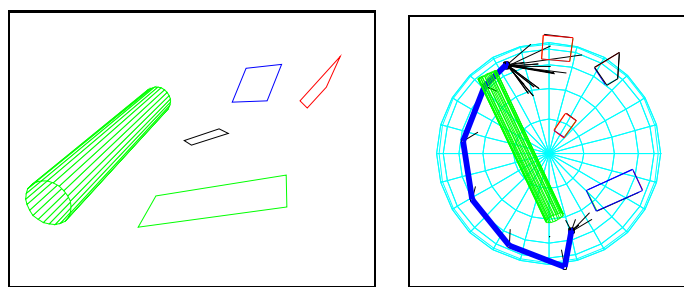


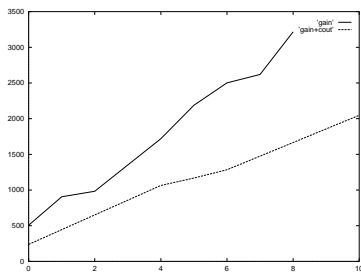
FIG. 10 - Visualisation de la scène reconstruite et vue polaire de la trajectoire finale de la caméra pendant l’exploration globale

locale, permettant la reconstruction des quatre cotés de la carte, la caméra est en position  $\phi_{30}$  (Figure 9.h). À ce stade, 99% de l’espace a été observé assurant ainsi une reconstruction complète de la scène. La Figure 10 montre une visualisation 3D de la scène telle qu’elle a été reconstruite, ainsi que la trajectoire de la caméra pendant l’exploration globale.

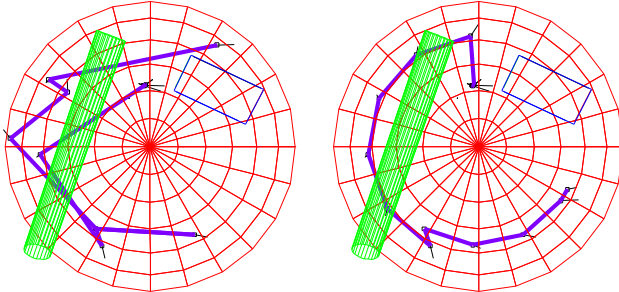
Nous montrons finalement les résultats de l’exploration d’une scène composée d’un cylindre et d’un polygone obtenus en suivant deux stratégies différentes. La première stratégie ne tient pas compte de la distance parcourue par la caméra et est donc principalement basée sur une maximisation de la zone découverte pour chaque point de vue (le coefficient  $\alpha_2$  dans l’équation (24) est nul). La deuxième stratégie tient compte de cette distance et tend donc à faire diminuer la distance totale parcourue par la caméra. La figure 11.a montre la distance cumulée parcourue par la caméra pour les deux stratégies. Les figures 11.b et 11.c montrent les différentes trajectoires effectuées menant à une exploration totale de la scène. On notera que si la distance parcourue n’est pas prise en compte, la caméra effectue une trajectoire en “vol d’abeille”, avec un nombre de points de vue calculés moins important. D’autre part, on peut noter que la deuxième stratégie a tendance à favoriser les mouvements suivants les 3 axes de rotation par rapport aux mouvements suivant les 3 axes de translation. Ceci est dû au fait que les rotations étant moins coûteuses, elles sont moins pénalisées que les translations.

## 4 Conclusion

Dans cet article, nous avons proposé une méthode permettant la reconstruction d’un environnement 3D en utilisant une séquence d’images acquises par une caméra mobile. Nous avons décrit un processus de reconstruction permettant une estimation précise et robuste des paramètres d’une primitive géométrique. Cette méthode étant basée sur des mouvements particuliers de la caméra, des stratégies perceptives permettant d’effectuer une succession de reconstructions indépendantes assurant la complétude de la reconstruction ont été proposées. Des expérimentations menées sur une cellule robotique ont démontré la validité de notre approche (résultats de reconstruction précis, robustes et stables, algo-



(a) Distance parcourue par le robot



(b) Trajectoire du robot sans tenir compte du coût de déplacement

(c) Trajectoire du robot en tenant compte du coût de déplacement

FIG. 11 - Exploration globale de la scène

rithmes d'exploration de la scène simples mais efficaces) mais aussi ses limitations: les contraintes fortes sur les mouvements de la caméra impliquent un séquencement fort des tâches de reconstruction et n'autorisent pas l'estimation des paramètres de plusieurs primitives en parallèle. C'est pourquoi il apparait utile en perspective de ce travail de déterminer les mouvements optimaux de caméra nécessaires à la reconstruction simultanée de plusieurs primitives, et ce sans dégradation notable de la qualité de la reconstruction.

## Remerciements

Cette étude a reçu le soutien du projet inter PRC VIA (Vision Intentionnelle et Action) et du MESR (Ministère de l'enseignement supérieur et de la recherche).

## Références

- [1] G. Adiv. – Inherent ambiguities in recovering 3D motion and structure from a noisy flow field. – *IEEE Trans. on PAMI*, 11(5):477–489, Mai 1989.
- [2] O. Al Chami et C. Laugier. – Stratégie perceptive pour positionner une caméra. – *Congrès AFCET Reconnaissance des Formes et Intelligence Artificielle*, volume 1, pages 617–622, Paris, France, 1994.
- [3] Y. Aloimonos. – Purpose and qualitative active vision. – *ICPR*, pp 346–360, New Jersey, 1990.
- [4] R. Bajcsy. – Active perception. – *Proc. of the IEEE*, 76(8):996–1005, Août 1988.
- [5] S. Boukir. – *Reconstruction 3D d'un environnement statique par vision active*. – Thèse de l'Université de Rennes I, IRISA, Octobre 1993.

- [6] F. Chaumette, S. Boukir, P. Bouthemy, et D. Juvin. – Optimal estimation of 3D structures using visual servoing. – *CVPR'94*, pp 347–354, Seattle, USA, Juillet 1994.
- [7] C. Chien et J.K. Aggarwal. – Model construction and shape recognition from occluding contour. – *IEEE Trans. on PAMI*, 11(4):372–389, Février 1989.
- [8] C. Connolly. – The determination of next best views. – In *Proc. of IEEE Int. Conf. on Robotics and Automation*, pp 432–435, St Louis, Missouri, Mars 1985.
- [9] C.K. Cowan et P.D. Kovesi. – Automatic sensor placement from vision task requirements. – *IEEE Trans. on PAMI*, 10(3):407–416, Mai 1988.
- [10] B. Espiau, F. Chaumette, et P. Rives. – A new approach to visual servoing in robotics. – *IEEE Trans. on Robotics and Automation*, 8(3):313–326, Juin 1992.
- [11] B. Espiau et P. Rives. – Closed-loop recursive estimation of 3D features for a mobile vision system. – *IEEE Int. Conf. on Robotics and Automation*, pp 1436–1443, Raleigh, North Carolina, Avril 1987.
- [12] O. Faugeras. – *Three-dimensional computer vision: a geometric viewpoint*. – MIT press, 1993.
- [13] M.A. Gennert et A.L. Yuille. – Determining the optimal weights in multiple objective function optimization. – *ICCV*, pp 87–94, 1988.
- [14] K. Hashimoto, editeur. – *Visual Servoing: Real Time Control of Robot manipulators based on visual sensory feedback*. – World Scientific Series in Robotics and Automated Systems, Vol 7, World Scientific Press, Singapour, 1993.
- [15] E. Marchand et F. Chaumette. – Active visual 3D perception. – *IEEE Workshop on Vision for Robots*, Pittsburgh, USA, Août 1995.
- [16] E. Marchand, F. Chaumette, et E. Rutten. – Real time active visual reconstruction using the synchronous paradigm. – *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, IROS'95*, Pittsburgh, USA, Août 1995.
- [17] J. Maver et R. Bajcsy. – Occlusions as a guide for planning the next view. – *IEEE Trans. on PAMI*, 15(5):417–433, Mai 1993.
- [18] C. Samson, B. Espiau, et M. Le Borgne. – *Robot Control: the Task Function Approach*. – Clarendon Press, Oxford, Angleterre, 1991.
- [19] K. Tarabanis, P.K. Allen, et R. Tsai. – A survey of sensor planning in computer vision. – *IEEE Trans. on Robotics and Automation*, 11(1):86–104, Février 1995.
- [20] K. Tarabanis, R. Tsai, et P.K. Allen. – The MVP sensor planning system for robotic vision tasks. – *IEEE Trans. on Robotics and Automation*, 11(1):72–85, Février 1995.
- [21] B. Triggs et C. Laugier. – Automatic camera placement for robot vision. – In *Proc. of IEEE Int. Conf. on Robotics and Automation*, Nagoya, Japon, Mai 1995.
- [22] A.M. Waxman, B.K. Parsi, et M. Subbarao. – Closed-form solutions to image flow equations for 3D structure and motion. – *International Journal of Computer Vision*, 1(3):239–258, Octobre 1987.
- [23] J. Weng, T.S. Huang, et N. Ahuja. – Estimation and structure from line matches: Performance obtained and beyond. – *ICPR*, pp 168–172, Juin 1990.
- [24] P. Whaitte et F. Ferrie. – Autonomous exploration: Driven by uncertainty. – *CVPR*, pp 339–346, Seattle, USA, Juillet 1994.
- [25] L.E. Wixson. – Viewpoint selection for visual search. – *CVPR*, pp 800–805, Seattle, USA, Juin 1994.
- [26] M. Xie et P. Rives. – Toward dynamic vision. – *IEEE Workshop on Interpretation of 3D Scenes*, Austin, Texas, Novembre 1989.