



HAL
open science

Reconstruction 3D de primitives géométriques par vision active

Samia Boukir, François Chaumette

► **To cite this version:**

| Samia Boukir, François Chaumette. Reconstruction 3D de primitives géométriques par vision active.
| [Rapport de recherche] RR-1722, INRIA. 1992. inria-00076960

HAL Id: inria-00076960

<https://inria.hal.science/inria-00076960>

Submitted on 29 May 2006

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

INRIA

UNITÉ DE RECHERCHE
INRIA-RENNES

Institut National
de Recherche
en Informatique
et en Automatique

Domaine de Voluceau
Rocquencourt
B.P.105
78153 Le Chesnay Cedex
France
Tél.: (1) 39 63 55 11

Rapports de Recherche

1992



ème

anniversaire

N° 1722

Programme 4
Robotique, Image et Vision

RECONSTRUCTION 3D DE PRIMITIVES GÉOMÉTRIQUES PAR VISION ACTIVE

Samia BOUKIR
François CHAUMETTE

Juillet 1992



★ R R - 1 7 2 2 ★

Reconstruction 3D de primitives géométriques par vision active

Samia Boukir, François Chaumette

Publication Interne n°662 - Juin 1992 - 40 pages - Programme 4

Projet TEMIS
IRISA / INRIA Rennes
Campus de Beaulieu
35 042 Rennes-cedex

Résumé

Ce rapport présente une méthode générale de reconstruction 3D des primitives géométriques paramétrables par vision dynamique. Cette méthode est basée sur un concept fondamental : le torseur d'interaction associé aux primitives de l'environnement, qui caractérise entièrement les interactions entre les mouvements de la caméra et la primitive considérée. Cette méthode est appliquée aux primitives les plus représentatives : point, droite, cercle, sphère et cylindre. Nous faisons ensuite appel à la vision active pour améliorer la performance des méthodes mises en œuvre. En effet, des mouvements adéquats de la caméra engendrés par des lois de commande en boucle fermée sur les informations visuelles, permettent d'affiner de manière très significative la reconstruction 3D des primitives. Divers résultats de simulation complètent ce rapport.

3D Reconstruction of geometrical features using active vision

Abstract

This paper presents a general method for 3D structure estimation of geometrical primitives using dynamic vision. This method is based on a fundamental concept: the interaction screw related to environment primitives which fully characterizes the interaction between the camera and the considered primitive. Examples of most representative primitives: points, lines, circles, spheres and cylinders illustrate this approach. Next, active vision is considered to improve the performance of the method. In fact, we show in this paper that adequate camera motions computed by control laws in closed-loop with respect to visual data, provide significative improvements in the structure estimation of the primitives. This paper ends with several simulation results.

1 Introduction

Depuis quelques années, de nombreux travaux menés en vision artificielle se sont fixés pour objectif la réalisation de systèmes puissants capables d'accéder à la géométrie spatiale d'une scène à partir de son observation par une ou plusieurs caméras mobiles. Ces systèmes devraient fournir une description géométrique claire et complète de la scène à partir d'une séquence d'images souvent bruitées et difficilement exploitables. Mais pour l'heure, il n'existe pas de systèmes réellement opérationnels en univers bruité et inconnu. Les études proposées aujourd'hui tentent d'apporter leur contribution au problème de la reconstruction d'environnements assez restreints (objets statiques, information a priori sur la scène, mouvements particuliers...) avec comme ambition la reconstruction de scènes de plus en plus complexes. Deux principales approches ont été mises en œuvre dans la littérature :

- L'approche discrète : dans le sens où les images sont acquises à des instants discrets. Un mouvement relativement important est permis entre deux acquisitions successives d'images. Pour des raisons de simplicité, on suppose généralement que la présence des primitives sélectionnées persiste d'une image à l'autre. Cette approche est confrontée au problème très délicat de la mise en correspondance de primitives [8] [5] [20].
- L'approche continue : dans le sens où les images sont acquises à la cadence vidéo [1] [18] [9] [21]. Cette approche est étroitement liée à l'estimation du champ des vitesses apparentes dans l'image (optic flow) [18] [6] [16] [17] qui est généralement bruité et partiellement incorrect notamment près des contours d'occlusions. Ainsi, le problème fondamental rencontré par ce type d'approche est sa sensibilité au bruit.

L'observabilité du mouvement de la caméra qui est essentielle en reconstruction 3D caractérise un domaine récent appelé "vision dynamique" qui, si le mouvement est en plus commandable, se transforme plus spécifiquement en "vision active". De manière générale, le but de la vision active est de manipuler les contraintes sous-jacentes aux phénomènes observés afin d'améliorer la qualité des résultats de perception [2] [3] [15] [4]. Dans cet article, nous nous sommes confrontés au problème de l'estimation de la structure spatiale de l'environnement par vision active. Nous avons examiné les deux aspects suivants :

- la modélisation qui consiste à établir des méthodes de reconstruction 3D de primitives géométriques par vision dynamique ;
- l'optimisation qui consiste à améliorer par vision active la performance des méthodes mises en œuvre.

Ce rapport est structuré de la manière suivante : nous présentons dans le chapitre 2 une méthode générale d'estimation des primitives géométriques paramétrables par vision dynamique. Nous l'illustrons ensuite par quelques exemples de primitives les plus

représentatives : point, droite, cercle, sphère et cylindre. Le chapitre 3 est consacré à la reconstruction 3D par vision active. Nous montrons que des mouvements adéquats de la caméra permettent de minimiser les effets des erreurs pouvant intervenir dans le processus d'estimation. Des résultats de simulation présentent les performances obtenues par notre approche. Finalement, nous décrivons en annexe de ce rapport un schéma de commande en boucle fermée sur les informations visuelles qui permet de générer automatiquement ces mouvements particuliers de la caméra.

2 Reconstruction 3D de primitives usuelles par vision dynamique

2.1 Introduction

Nous désirons construire une représentation géométrique de l'environnement à partir des images obtenues par une caméra embarquée sur l'effecteur d'un robot. Nous restreindrons notre étude à des scènes statiques constituées d'objets relativement simples dont la structure spatiale peut être approximée par des ensembles de primitives géométriques paramétrables (droites, cylindres...).

2.2 Modélisation

Signalons tout d'abord que le repère de la caméra $(O, \vec{x}, \vec{y}, \vec{z})$ donné Figure 1 sera par défaut celui dans lequel seront exprimées par la suite toutes les grandeurs nécessaires. par exemple, le torseur cinématique, les coordonnées des points, les torseurs d'interaction, etc.

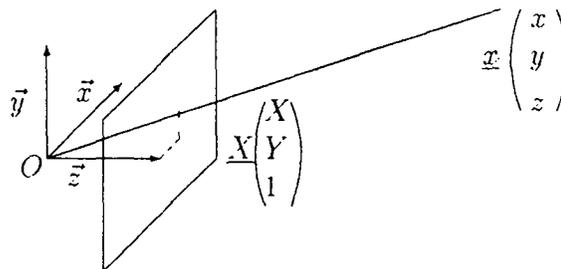


Figure 1 : Modèle de la caméra

La caméra est modélisée par son centre optique O et son plan image P_i (voir figure 1). Sans la moindre perte de généralité, la distance focale est fixée égale à 1 de telle sorte que tout point m ayant pour coordonnées $\underline{x} = (x \ y \ z)$ se projette par projection

perspective sur le plan image en M de coordonnées $\underline{X} = (X Y 1)$ avec :

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

Considérons à présent une primitive géométrique paramétrable \mathcal{P}_s de la scène. Celle-ci est décrite par une équation du type :

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

où h définit la nature de la primitive et \underline{p} sa configuration. Notre objectif est d'estimer la valeur des paramètres \underline{p} afin de reconstruire et localiser la primitive \mathcal{P}_s décrite par h (le type de la primitive à reconstruire est donc supposé connu a priori). Pour ce faire, la représentation de \mathcal{P}_s par l'ensemble des paramètres \underline{p} est choisie complète et unique (quitte à rajouter si nécessaire des contraintes liant certains paramètres \underline{p} entre eux). Ainsi, quelle que soit la configuration de la primitive, il n'existera toujours qu'un seul jeu unique de valeurs de \underline{p} correspondant à cette configuration. Nous noterons par ailleurs n le nombre de paramètres indépendants de \underline{p} .

En divisant l'équation (2) par z (> 0) et en utilisant (1), l'équation (2) devient :

$$h'(\underline{X}, 1/z, \underline{p}) = 0 \quad (3)$$

Le théorème des fonctions implicites assure, sous la condition $\frac{\partial h'}{\partial (1/z)} \neq 0$, satisfaite pour tous les cas non dégénérés (un cas dégénéré se produit par exemple quand une droite se projette dans l'image en un point ou un cercle en un segment), qu'il existe une fonction unique μ autour d'une solution de (2) telle que [7] [10] :

$$1/z = \mu(\underline{X}, \underline{p}_0) \quad (4)$$

où les paramètres \underline{p}_0 , fonction des paramètres \underline{p} , sont choisis pour les mêmes raisons que précédemment de telle sorte que la dimension n_0 de \underline{p}_0 soit minimale ($n_0 < n$).

Soit \mathcal{P}_i la projection de \mathcal{P}_s sur le plan image. En utilisant (4) dans (3), on peut caractériser \mathcal{P}_i par une équation de la forme :

$$h'(\underline{X}, \underline{p}) = 0, \quad \forall \underline{X} \in \mathcal{P}_i \quad (5)$$

qui, après reparamétrisation, peut se réécrire :

$$g(\underline{X}, \underline{P}) = 0, \quad \forall \underline{X} \in \mathcal{P}_i \quad (6)$$

où \underline{P} , fonction des paramètres \underline{p} , est choisi de telle sorte que sa dimension m soit minimale ($m < n$) [7].

- **Remarque :** Pour une primitive plane, la fonction μ représente le plan contenant la primitive. Pour une primitive tridimensionnelle (sphère, cylindre, tore....), la fonction $g(\underline{X}, \underline{P})$ représente l'équation des limbes et définit le contour dans l'image de la projection de la primitive. La relation entre ces points contour et les points 3D correspondants s'exprime par la fonction $\mu(\underline{X}, \underline{p}_0)$ qui représente alors la surface des limbes.

Soit $T = (V(O), \Omega)$ le torseur cinématique de la caméra où $V(O) = (V_x V_y V_z)$ représente sa vitesse de translation et $\Omega = (\Omega_x \Omega_y \Omega_z)^T$ sa vitesse de rotation. La variation de \underline{P} qui permet de lier le mouvement apparent de la primitive dans l'image au mouvement de la caméra peut alors s'exprimer par l'équation fondamentale [7] [10] :

$$\dot{\underline{P}} = L_{\underline{P}}^T(\underline{P}, \underline{p}_0) T \quad (7)$$

où $L_{\underline{P}}^T$, appelée matrice d'interaction associée à \underline{P} , caractérise entièrement les interactions entre la caméra et la primitive considérée. Dans [7] et [10] est décrite une méthode de calcul de la matrice d'interaction associée aux primitives géométriques paramétrables. Les résultats explicites y sont donnés pour les primitives les plus usuelles : point, segment, droite, cercle, sphère et cylindre.

2.3 Reconstruction 3D par vision dynamique

La méthode de reconstruction 3D que nous allons présenter est directement liée au formalisme qui vient d'être rappelé, à savoir la matrice d'interaction associée à une primitive. Elle se situe donc dans la catégorie des approches continues, puisque basée sur la mesure de $\dot{\underline{P}}$ et du torseur cinématique T .

Soit $\mathcal{H}(\underline{P}, \dot{\underline{P}}, \underline{p}_0, T)$ la fonctionnelle suivante définie à partir de (7) :

$$\mathcal{H}(\underline{P}, \dot{\underline{P}}, \underline{p}_0, T) = \dot{\underline{P}} - L_{\underline{P}}^T(\underline{P}, \underline{p}_0) T = 0 \quad (8)$$

Sous la condition $\frac{\partial \mathcal{H}}{\partial \underline{p}_0}$ (de dimension $m * n_0$) de rang plein n_0 , le théorème des fonctions implicites nous permet d'exprimer \underline{p}_0 en fonction des autres variables de la fonctionnelle (8) :

$$\underline{p}_0 = \underline{p}_0(T, \underline{P}, \dot{\underline{P}}) \quad (9)$$

Généralement, comme nous le verrons dans les exemples décrits par la suite, les paramètres \underline{p}_0 s'obtiennent simplement par résolution d'un système linéaire. Il est par ailleurs intéressant de noter que la condition d'utilisation du théorème des fonctions implicites permet de détecter, lorsqu'elle n'est pas vérifiée, les configurations de la primitive et les mouvements de la caméra qui ne permettent pas de remonter à une information tridimensionnelle sur cette primitive.

Connaissant donc la structure du torseur d'interaction associé à la primitive à reconstruire et ayant mesuré le long d'une séquence de deux images la vitesse de la caméra T , l'ensemble des paramètres 2D \underline{P} associés à la primitive \mathcal{P}_i et leur variation $\dot{\underline{P}}$, on peut déterminer les paramètres \underline{p}_0 qui représentent la fonction μ donnée par (4). La profondeur de tous les points de la primitive dont la projection dans l'image vérifie $g(\underline{X}, \underline{P}) = 0$ est alors connue. On peut ensuite calculer les paramètres \underline{p} qui définissent complètement la configuration de la primitive :

- **Cas des primitives planes :** La fonction μ représente dans ce cas le plan contenant la primitive. Les paramètres \underline{p} sont obtenus en effectuant l'intersection de ce plan avec le cône généralisé défini par son sommet au centre optique de la caméra O et la fonction $g(\underline{X}, \underline{P})$.
- **Cas des primitives tri-dimensionnelles :** L'intersection du cône défini par O et g avec la surface des limbes définie par μ permet la reconstruction 3D des limbes (voir figure 2). Des propriétés spécifiques à celles-ci telles que la tangence (voir par exemple le cas du cylindre présenté au paragraphe 2.3.4) permettent ensuite de déterminer l'ensemble des paramètres \underline{p} .

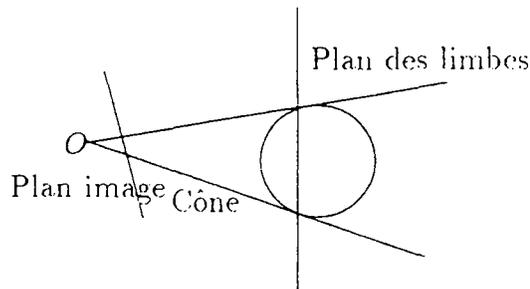


Figure 2 : Vue de dessus d'un cylindre : approche continue

- **Remarque :** L'approche discrète, similaire à celle de la reconstruction 3D par stéréovision, est basée sur la triangulation. Elle consiste à calculer l'intersection de deux cônes généralisés (un pour chaque position de la caméra), soit l'intersection de deux volumes (voir figure 3). L'approche continue consiste, elle, à effectuer l'intersection d'un volume et d'une surface (un plan ou la surface des limbes selon les cas) et s'avère donc de complexité moindre.

Nous allons maintenant nous attacher à reconstruire par la méthode que nous venons de décrire les primitives géométriques qui nous ont paru les plus significatives à savoir le point, la droite, le cercle, la sphère et le cylindre. Signalons que la reconstruction des primitives de type point et segment par la même approche a déjà fait l'objet de nombreux travaux [9] [21].

2.3.1 Les points

Considérons un point de coordonnées (x_0, y_0, z_0) dans le repère de la caméra. Dans ce cas très simple, on a :

$$h(\underline{x}, \underline{p}) : \begin{cases} x - x_0 = 0 \\ y - y_0 = 0 \\ z - z_0 = 0 \end{cases} \quad , \quad g(\underline{X}, \underline{P}) : \begin{cases} X - X_0 = 0 \\ Y - Y_0 = 0 \end{cases} \quad , \quad \mu(\underline{X}, \underline{p}_0) : 1/z = 1/z_0 \quad (10)$$

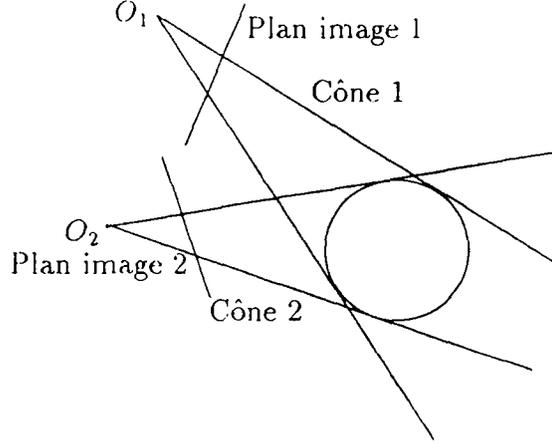


Figure 3 : Vue de dessus d'un cylindre : approche discrète

avec $\underline{p} = (x_0, y_0, z_0)$, $\underline{P} = (X_0, Y_0) = (x_0/z_0, y_0/z_0)$ et $\underline{p}_0 = 1/z_0$.

Les variations des coordonnées d'un point dans l'image en fonction du torseur cinématique de la caméra (appelées également équations d'"optic flow") sont écrites ici sous la forme :

$$\begin{pmatrix} \dot{X}_0 \\ \dot{Y}_0 \end{pmatrix} = L_{\underline{P}}^T T \quad (11)$$

avec :

$$L_{\underline{P}}^T = \begin{pmatrix} -1/z_0 & 0 & X_0/z_0 & X_0 Y_0 & -(1 + X_0^2) & Y_0 \\ 0 & -1/z_0 & Y_0/z_0 & 1 + Y_0^2 & -X_0 Y_0 & -X_0 \end{pmatrix} \quad (12)$$

L'équation (11) permet de calculer la profondeur z_0 du point. En effet, on peut la réécrire sous la forme :

$$A \ 1/z_0 = B \quad (13)$$

avec :

$$A = \begin{pmatrix} V_x - X_0 V_z \\ V_y - Y_0 V_z \end{pmatrix}, \quad B = \begin{pmatrix} \alpha_x \\ \alpha_y \end{pmatrix} = \begin{pmatrix} X_0 Y_0 \Omega_x - (1 + X_0^2) \Omega_y + Y_0 \Omega_z - \dot{X}_0 \\ (1 + Y_0^2) \Omega_x - X_0 Y_0 \Omega_y - X_0 \Omega_z - \dot{Y}_0 \end{pmatrix}$$

La solution aux moindres carrés est alors donnée par $1/z_0 = (A^T A)^{-1} A^T B$ sous réserve que $A^T A$ soit inversible et on obtient :

$$1/z_0 = \frac{\alpha_x (X_0 V_z - V_x) + \alpha_y (Y_0 V_z - V_y)}{(X_0 V_z - V_x)^2 + (Y_0 V_z - V_y)^2} \quad (14)$$

- **Remarque** : La condition $\frac{\partial \mathcal{H}}{\partial \underline{p}_0}$ de rang plein est ici équivalente à $A^T A$ inversible.

On en déduit les mouvements bien connus de la caméra qui ne permettent pas d'estimer la profondeur z_0 d'un point par vision dynamique et qui consistent à avancer dans la direction du point : $V_x = X_0 V_z$ et $V_y = Y_0 V_z$.

2.3.2 Les droites

Nous représentons une droite par un vecteur $\underline{p} = (a_1, b_1, c_1, a_2, b_2, c_2, d_2)$ tel que les points de coordonnées (x, y, z) appartenant à la droite vérifient :

$$h(\underline{x}, \underline{p}) = \begin{cases} a_1x + b_1y + c_1z = 0 \\ a_2x + b_2y + c_2z + d_2 = 0 \end{cases} \quad (15)$$

Une droite est ainsi représentée comme l'intersection de deux plans \mathcal{P}_1 et \mathcal{P}_2 , l'un deux passant par le centre de projection. Cette représentation n'est pas minimale mais elle a l'avantage d'être complète. Elle permet en effet de décrire n'importe quelle droite de l'espace 3D. Pour que la représentation soit unique et non ambiguë, le choix de trois contraintes s'impose (quatre paramètres sont suffisants pour représenter une droite). On choisit bien évidemment les contraintes d'orthonormalité suivantes :

$$\begin{cases} a_1^2 + b_1^2 + c_1^2 = 1 \\ a_2^2 + b_2^2 + c_2^2 = 1 \\ a_1a_2 + b_1b_2 + c_1c_2 = 0 \end{cases} \quad (16)$$

Notons qu'on ne s'intéresse pas au cas dégénéré où la droite passe par le centre de projection ($d_2 = 0$) et qui donne un point pour image de la droite.

La fonction μ définie par (4) s'obtient facilement en utilisant l'équation de projection perspective (1) dans (15) :

$$\mu(\underline{X}, \underline{p}_0) : 1/z = aX + bY + c \quad (17)$$

avec $\underline{p}_0 = (a, b, c) = (-a_2/d_2, -b_2/d_2, -c_2/d_2)$. Par ailleurs, la projection dans l'image de la droite s'exprime par l'équation :

$$a_1X + b_1Y + c_1 = 0 \quad (18)$$

qui peut se réécrire en utilisant la représentation minimale $\underline{P} = (\rho, \theta)$ (voir figure 4) :

$$g(\underline{X}, \underline{P}) = X \cos \theta + Y \sin \theta - \rho = 0 \quad (19)$$

avec :

$$\begin{cases} \cos \theta = a_1 / \sqrt{a_1^2 + b_1^2} \\ \sin \theta = b_1 / \sqrt{a_1^2 + b_1^2} \\ \rho = -c_1 / \sqrt{a_1^2 + b_1^2} \end{cases} \quad (20)$$

- **Remarque** : Les ambiguïtés liées à cette représentation (les jeux de paramètres $(\rho, \theta + 2k\pi)$ et $(-\rho, \theta + (2k + 1)\pi)$, $\forall k$, représentent tous la même droite) sont facilement levées en choisissant pour θ l'orientation modulée à 2π entre l'axe \vec{X} du plan image et la normale de la droite, orientée par convention dans le sens de la diminution de l'intensité lumineuse.

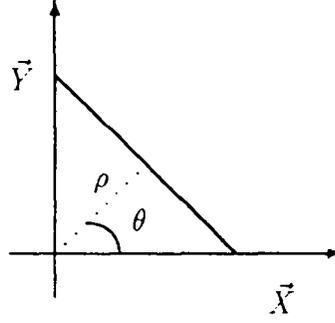


Figure 4 : Représentation (ρ, θ) des droites 2D

Les paramètres a_1, b_1 et c_1 du plan \mathcal{P}_1 passant par le centre de projection se calculent immédiatement à partir des valeurs de ρ et θ . On obtient, puisque $a_1^2 + b_1^2 + c_1^2$ a été fixé égal à 1 :

$$\begin{cases} a_1 = \cos \theta / \sqrt{1 + \rho^2} \\ b_1 = \sin \theta / \sqrt{1 + \rho^2} \\ c_1 = -\rho / \sqrt{1 + \rho^2} \end{cases} \quad (21)$$

Pour calculer les paramètres du second plan caractérisant la droite à reconstruire, nous devons, comme évoqué précédemment, utiliser la matrice d'interaction $L_{\underline{P}}^T$ associé à $\underline{P} = (\rho, \theta)$. Celle-ci s'écrit [7] :

$$L_{\underline{P}}^T = \begin{pmatrix} \lambda_\rho \cos \theta & \lambda_\rho \sin \theta & -\lambda_\rho \rho & (1 + \rho^2) \sin \theta & -(1 + \rho^2) \cos \theta & 0 \\ \lambda_\theta \cos \theta & \lambda_\theta \sin \theta & -\lambda_\theta \rho & -\rho \cos \theta & -\rho \sin \theta & -1 \end{pmatrix} \quad (22)$$

avec :

$$\begin{cases} \lambda_\rho = -(a\rho \cos \theta + b\rho \sin \theta + c) \\ \lambda_\theta = -a \sin \theta + b \cos \theta \end{cases} \quad (23)$$

Les paramètres $\underline{p}_0 = (a, b, c)$ sont obtenus en résolvant le système linéaire suivant construit à partir de (23) et de la contrainte d'orthogonalité entre les plans \mathcal{P}_1 et \mathcal{P}_2 :

$$\begin{cases} -a\rho \cos \theta - b\rho \sin \theta - c = \lambda_\rho \\ -a \sin \theta + b \cos \theta = \lambda_\theta \\ a \cos \theta + b \sin \theta - c\rho = 0 \end{cases} \quad (24)$$

On en déduit :

$$\begin{cases} a = -((1 + \rho^2)\lambda_\theta \sin \theta + \rho\lambda_\rho \cos \theta) / (1 + \rho^2) \\ b = ((1 + \rho^2)\lambda_\theta \cos \theta - \rho\lambda_\rho \sin \theta) / (1 + \rho^2) \\ c = -\lambda_\rho / (1 + \rho^2) \end{cases} \quad (25)$$

où λ_ρ et λ_θ sont obtenus à partir de l'équation (7) dans laquelle on injecte (22) :

$$\begin{cases} \lambda_\rho = \frac{\dot{\rho} + (1 + \rho^2)(\Omega_y \cos \theta - \Omega_x \sin \theta)}{V_x \cos \theta + V_y \sin \theta - \rho V_z} \\ \lambda_\theta = \frac{\dot{\theta} + \rho(\Omega_x \cos \theta + \Omega_y \sin \theta) + \Omega_z}{V_x \cos \theta + V_y \sin \theta - \rho V_z} \end{cases} \quad (26)$$

Enfin, on obtient $d_2 = 1/\sqrt{a^2 + b^2 + c^2}$, $a_2 = -ad_2$, $b_2 = -bd_2$ et $c_2 = -cd_2$. L'ensemble des coefficients représentant la droite est ainsi complètement caractérisé.

- **Remarque** : On peut facilement montrer en étudiant le rang de la matrice $\frac{\partial \mathcal{H}}{\partial \underline{p}_0}$ définie à partir de (8) que les mouvements de la caméra ne permettant pas de reconstruire une droite sont tels que $V_x \cos \theta + V_y \sin \theta - \rho V_z = 0$.

2.3.3 Les cercles

Un cercle peut être représenté comme l'intersection d'une sphère et d'un plan coupant la sphère en son centre. On a alors :

$$h(\underline{x}, \underline{p}) = \begin{cases} h_1 = (x - x_0)^2 + (y - y_0)^2 + (z - z_0)^2 - r^2 = 0 \\ h_2 = \alpha(x - x_0) + \beta(y - y_0) + \gamma(z - z_0) = 0 \end{cases} \quad (27)$$

Un cercle est ainsi caractérisé par un vecteur $\underline{p} = (x_0, y_0, z_0, r, \alpha, \beta, \gamma)$ de dimension 7 où r est le rayon du cercle, x_0, y_0 et z_0 sont les coordonnées de son centre et α, β et γ sont les composantes de la normale au cercle. Cette représentation est complète mais non minimale (six paramètres sont suffisants pour représenter un cercle). On rajoute donc la contrainte suivante :

$$\alpha^2 + \beta^2 + \gamma^2 = 1 \quad (28)$$

La fonction μ définie par (4) est obtenue en substituant (1) dans h_2 :

$$\mu(\underline{X}, \underline{p}_0) : 1/z = aX + bY + c \quad (29)$$

avec :

$$\begin{cases} a = \alpha/(\alpha x_0 + \beta y_0 + \gamma z_0) \\ b = \beta/(\alpha x_0 + \beta y_0 + \gamma z_0) \\ c = \gamma/(\alpha x_0 + \beta y_0 + \gamma z_0) \end{cases} \quad (30)$$

L'image d'un cercle est généralement une ellipse (parfois un cercle) et est définie par une équation du type [7] :

$$g(\underline{X}, \underline{p}) = K_0 X^2 + K_1 Y^2 + 2K_2 XY + 2K_3 X + 2K_4 Y + K_5 = 0 \quad (31)$$

avec :

$$\begin{cases} K_0 = a^2(x_0^2 + y_0^2 + z_0^2 - r^2) + 1 - 2ax_0 \\ K_1 = b^2(x_0^2 + y_0^2 + z_0^2 - r^2) + 1 - 2by_0 \\ K_2 = ab(x_0^2 + y_0^2 + z_0^2 - r^2) - bx_0 - ay_0 \\ K_3 = ac(x_0^2 + y_0^2 + z_0^2 - r^2) - cx_0 - az_0 \\ K_4 = bc(x_0^2 + y_0^2 + z_0^2 - r^2) - cy_0 - bz_0 \\ K_5 = c^2(x_0^2 + y_0^2 + z_0^2 - r^2) + 1 - 2cz_0 \end{cases} \quad (32)$$

- **Remarque** : Pour le cercle, les cas dégénérés sont tels que le plan contenant le cercle passe par le centre optique ($\alpha x_0 + \beta y_0 + \gamma z_0 = 0$) : l'image d'un cercle est alors un segment.

Un paramétrage minimal d'une ellipse \mathcal{E} est obtenu à partir des moments $m_{ij} = \sum_{X \in \mathcal{E}} \sum_{Y \in \mathcal{E}} X^i Y^j$ d'ordre inférieur à 3 ($i + j \leq 2$). On choisit les moments centrés $\underline{P} = (X_c, Y_c, \mu_{20}, \mu_{11}, \mu_{02})$ et on peut établir les relations suivantes :

$$\begin{cases} X_c = m_{01}/m_{00} = (K_1 K_3 - K_2 K_4)/\Delta \\ Y_c = m_{10}/m_{00} = (K_0 K_4 - K_2 K_3)/\Delta \\ \mu_{20} = (m_{20} - m_{00} X_c^2)/4m_{00} = -K_1 K/\Delta \\ \mu_{11} = (m_{11} - m_{00} X_c Y_c)/4m_{00} = K_2 K/\Delta \\ \mu_{02} = (m_{02} - m_{00} Y_c^2)/4m_{00} = -K_0 K/\Delta \end{cases} \quad (33)$$

avec $\Delta = K_2^2 - K_0 K_1$ et $K = K_0 X_c^2 + 2K_2 X_c Y_c + K_1 Y_c^2 - K_5$.

La matrice d'interaction associée à cette représentation s'écrit [7] :

$$L_{\underline{P}}^T = \begin{pmatrix} -1/z_c & 0 & X_c/z_c + a\mu_{20} + b\mu_{11} \\ X_c Y_c + \mu_{11} & -1 - X_c^2 - \mu_{20} & Y_c \\ 0 & -1/z_c & Y_c/z_c + a\mu_{11} + b\mu_{02} \\ 1 + Y_c^2 + \mu_{02} & -X_c Y_c - \mu_{11} & -X_c \\ -2(a\mu_{20} + b\mu_{11}) & 0 & 2[(1/z_c + aX_c)\mu_{20} + bX_c\mu_{11}] \\ 2(Y_c\mu_{20} + X_c\mu_{11}) & -4\mu_{20} X_c & 2\mu_{11} \\ -a\mu_{11} - b\mu_{02} & -a\mu_{20} - b\mu_{11} & aY_c\mu_{20} + (3/z_c - c)\mu_{11} + bX_c\mu_{02} \\ 3Y_c\mu_{11} + X_c\mu_{02} & -Y_c\mu_{20} - 3X_c\mu_{11} & \mu_{02} - \mu_{20} \\ 0 & -2(a\mu_{11} + b\mu_{02}) & 2[(1/z_c + bY_c)\mu_{02} + aY_c\mu_{11}] \\ 4Y_c\mu_{02} & -2(Y_c\mu_{11} + X_c\mu_{02}) & -2\mu_{11} \end{pmatrix} \quad (34)$$

où $z_c = 1/(aX_c + bY_c + c)$.

Les paramètres $\underline{p}_0 = (a, b, c)$ sont déterminés directement par la résolution aux moindres carrés du système linéaire défini à partir de (34). On en déduit en utilisant (30)

et (28) les paramètres α, β et γ , composantes de la normale au plan du cercle. On obtient :

$$\begin{cases} \alpha = \pm a/\sqrt{a^2 + b^2 + c^2} \\ \beta = \pm b/\sqrt{a^2 + b^2 + c^2} \\ \gamma = \pm c/\sqrt{a^2 + b^2 + c^2} \end{cases} \quad (35)$$

- **Remarque :** L'ambiguïté logiquement observée sur le sens de la normale au plan du cercle peut facilement être levée en se fixant une contrainte choisie par convention (par exemple $\alpha x_0 + \beta y_0 + \gamma z_0 > 0$).

Considérons à présent le système linéaire suivant construit à partir de (30) et (32) :

$$\begin{cases} ax_0 + by_0 + cz_0 & = 1 \\ (acK_0 - a^2K_3)A - (cK_0 - 2aK_3)x_0 - aK_0z_0 & = K_3 \\ (bcK_0 - a^2K_4)A + 2aK_4x_0 - cK_0y_0 - bK_0z_0 & = K_4 \\ (c^2K_0 - a^2K_5)A + 2aK_5x_0 - 2cK_0z_0 & = K_5 - K_0 \\ (b^2K_0 - a^2K_1)A + 2aK_1x_0 - 2bK_0y_0 & = K_1 - K_0 \\ (abK_0 - a^2K_2)A - (bK_0 - 2aK_2)x_0 - aK_0y_0 & = K_2 \end{cases} \quad (36)$$

avec $A = x_0^2 + y_0^2 + z_0^2 - r^2$. Comme la représentation d'une ellipse par les six paramètres K_i n'est pas minimale, leur valeur est impossible à déterminer de manière unique à partir des valeurs mesurées dans l'image des moments centrés. Par contre, si on pose $A_i = K_i/K_0$, ($i = 1, 5$) sachant que K_0 est toujours différent de 0 pour les cas non dégénérés, ce système se réécrit :

$$\begin{cases} ax_0 + by_0 + cz_0 & = 1 \\ (ac - a^2A_3)A - (c - 2aA_3)x_0 - az_0 & = A_3 \\ (bc - a^2A_4)A + 2aA_4x_0 - cy_0 - bz_0 & = A_4 \\ (c^2 - a^2A_5)A + 2aA_5x_0 - 2cz_0 & = A_5 - 1 \\ (b^2 - a^2A_1)A + 2aA_1x_0 - 2by_0 & = A_1 - 1 \\ (ab - a^2A_2)A - (b - 2aA_2)x_0 - ay_0 & = A_2 \end{cases} \quad (37)$$

où les paramètres A_i sont calculés à partir du changement de variable suivant :

$$\begin{cases} A_1 = \mu_{20}/\mu_{02} \\ A_2 = -\mu_{11}/\mu_{02} \\ A_3 = (\mu_{11}Y_c - \mu_{02}X_c)/\mu_{02} \\ A_4 = (\mu_{11}X_c - \mu_{20}Y_c)/\mu_{02} \\ A_5 = (\mu_{02}X_c^2 + \mu_{20}Y_c^2 - 2\mu_{11}X_cY_c + \mu_{11}^2 - \mu_{20}\mu_{02})/\mu_{02} \end{cases} \quad (38)$$

On peut ainsi calculer A et les expressions des coordonnées x_0, y_0 et z_0 du centre du cercle par résolution aux moindres carrés du système linéaire (37). On en déduit ensuite la valeur du rayon du cercle :

$$r = \sqrt{x_0^2 + y_0^2 + z_0^2 - A} \quad (39)$$

2.3.4 Les cylindres

a) **Modélisation** : L'équation paramétrique d'un cylindre s'écrit [7] :

$$h(\underline{x}, \underline{p}) = (x - x_0)^2 + (y - y_0)^2 + (z - z_0)^2 - (ax + by + cz)^2 - r^2 = 0 \quad (40)$$

où r est le rayon du cylindre, a , b et c les composantes du vecteur directeur de l'axe du cylindre et x_0 , y_0 et z_0 les coordonnées d'un point m_0 de cet axe. Un cylindre est donc représenté par le vecteur $\underline{p} = (x_0, y_0, z_0, r, a, b, c)$. C'est un paramétrage complet mais non minimal. En effet, cinq paramètres sont suffisants pour représenter un cylindre. On rajoute donc les deux contraintes suivantes :

$$a^2 + b^2 + c^2 = 1 \quad (41)$$

$$ax_0 + by_0 + cz_0 = 0 \quad (42)$$

qui fixe notamment le point m_0 comme étant le point de l'axe du cylindre le plus proche du centre optique (voir figure 5).

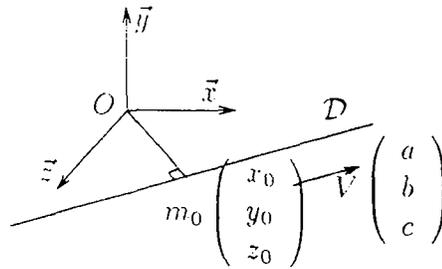


Figure 5 : Point le plus proche du centre optique

En appliquant l'équation de projection perspective (1) sur l'équation (40), on obtient un polynôme du second degré en $1/z$:

$$A 1/z^2 - 2 B 1/z + C = 0 \quad (43)$$

$$\text{avec } \begin{cases} A = x_0^2 + y_0^2 + z_0^2 - r^2 \\ B = x_0 X + y_0 Y + z_0 \\ C = X^2 + Y^2 + 1 - (aX + bY + c)^2 \end{cases}$$

Ce polynôme signifie que pour un point quelconque (X, Y) du plan image, il peut exister :

- soit aucun point d'intersection entre la ligne de vue associée à (X, Y) et le cylindre ;
- soit un point d'intersection ;
- soit deux points d'intersection.

Le cas qui nous intéresse est celui où il n'existe qu'un seul point d'intersection entre les lignes de vue des points du plan image et le cylindre. Ces points appartiennent aux contours (ou limbes) du cylindre. Dans le cas présent, cette propriété est équivalente au fait que le discriminant Δ de (43) soit nul. Après quelques développements décrits dans [7], on peut factoriser Δ sous la forme :

$$\Delta = [rB/\sqrt{A} - (\alpha X + \beta Y + \gamma)][rB/\sqrt{A} + (\alpha X + \beta Y + \gamma)] = 0 \quad (44)$$

$$\text{avec } \begin{cases} \alpha = cy_0 - bz_0 \\ \beta = az_0 - cx_0 \\ \gamma = bx_0 - ay_0 \end{cases}$$

qui donne l'équation de deux droites (une seule si le rayon du cylindre est nul). En utilisant la représentation (ρ, θ) des droites, on obtient :

$$\begin{aligned} D_1 & : X \cos \theta_1 + Y \sin \theta_1 - \rho_1 = 0 \\ D_2 & : X \cos \theta_2 + Y \sin \theta_2 - \rho_2 = 0 \end{aligned} \quad (45)$$

$$\text{avec } \begin{cases} \cos \theta_1 = \frac{rx_0/\sqrt{A}-\alpha}{\sqrt{(rx_0/\sqrt{A}-\alpha)^2+(ry_0/\sqrt{A}-\beta)^2}} & \cos \theta_2 = \frac{rx_0/\sqrt{A}+\alpha}{\sqrt{(rx_0/\sqrt{A}+\alpha)^2+(ry_0/\sqrt{A}+\beta)^2}} \\ \sin \theta_1 = \frac{ry_0/\sqrt{A}-\beta}{\sqrt{(rx_0/\sqrt{A}-\alpha)^2+(ry_0/\sqrt{A}-\beta)^2}} & \sin \theta_2 = \frac{ry_0/\sqrt{A}+\beta}{\sqrt{(rx_0/\sqrt{A}+\alpha)^2+(ry_0/\sqrt{A}+\beta)^2}} \\ \rho_1 = \frac{rz_0/\sqrt{A}-\gamma}{\sqrt{(rx_0/\sqrt{A}-\alpha)^2+(ry_0/\sqrt{A}-\beta)^2}} & \rho_2 = \frac{rz_0/\sqrt{A}+\gamma}{\sqrt{(rx_0/\sqrt{A}+\alpha)^2+(ry_0/\sqrt{A}+\beta)^2}} \end{cases}$$

Par ailleurs, la fonction μ définie par (4) est facilement obtenue à partir de la racine double de (43) correspondant à $\Delta = 0$ et on obtient :

$$\mu(\underline{X}, \underline{p}_0) : 1/z = n_1 X + n_2 Y + n_3 \quad (46)$$

où $\underline{p}_0 = (n_1, n_2, n_3) = (x_0/A, y_0/A, z_0/A)$ sont les composantes de la normale au plan des limbes. La matrice d'interaction $L_{\underline{P}}^T$ avec $\underline{P} = (\rho_1, \theta_1, \rho_2, \theta_2)$ s'écrit comme suit [7] :

$$L_{\underline{P}}^T = \begin{pmatrix} \lambda_{\rho_1} \cos \theta_1 & \lambda_{\rho_1} \sin \theta_1 & -\lambda_{\rho_1} \rho_1 & (1 + \rho_1^2) \sin \theta_1 & -(1 + \rho_1^2) \cos \theta_1 & 0 \\ \lambda_{\theta_1} \cos \theta_1 & \lambda_{\theta_1} \sin \theta_1 & -\lambda_{\theta_1} \rho_1 & -\rho_1 \cos \theta_1 & -\rho_1 \sin \theta_1 & -1 \\ \lambda_{\rho_2} \cos \theta_2 & \lambda_{\rho_2} \sin \theta_2 & -\lambda_{\rho_2} \rho_2 & (1 + \rho_2^2) \sin \theta_2 & -(1 + \rho_2^2) \cos \theta_2 & 0 \\ \lambda_{\theta_2} \cos \theta_2 & \lambda_{\theta_2} \sin \theta_2 & -\lambda_{\theta_2} \rho_2 & -\rho_2 \cos \theta_2 & -\rho_2 \sin \theta_2 & -1 \end{pmatrix} \quad (47)$$

$$\text{avec } \begin{cases} \lambda_{\rho_1} = -(n_1 \rho_1 \cos \theta_1 + n_2 \rho_1 \sin \theta_1 + n_3) \\ \lambda_{\theta_1} = n_2 \cos \theta_1 - n_1 \sin \theta_1 \\ \lambda_{\rho_2} = -(n_1 \rho_2 \cos \theta_2 + n_2 \rho_2 \sin \theta_2 + n_3) \\ \lambda_{\theta_2} = n_2 \cos \theta_2 - n_1 \sin \theta_2 \end{cases} \quad (48)$$

b) **Caractérisation de la fonction μ** : En résolvant le système linéaire (48), on peut facilement déterminer les paramètres $\underline{p}_0 = (n_1, n_2, n_3)$. On obtient :

$$\left\{ \begin{array}{l} n_1 = \frac{(\rho_1 \sin \theta_1 - \rho_2 \sin \theta_2)(\lambda_{\rho_1} - \lambda_{\rho_2}) - (\cos \theta_2 - \cos \theta_1)(\lambda_{\rho_1} - \lambda_{\rho_2})}{(\rho_1 \cos \theta_1 - \rho_2 \cos \theta_2)(\cos \theta_2 - \cos \theta_1) - (\rho_1 \sin \theta_1 - \rho_2 \sin \theta_2)(\sin \theta_1 - \sin \theta_2)} \\ n_2 = \frac{(\sin \theta_1 - \sin \theta_2)(\lambda_{\rho_1} - \lambda_{\rho_2}) - (\rho_1 \cos \theta_1 - \rho_2 \cos \theta_2)(\lambda_{\rho_1} - \lambda_{\rho_2})}{(\rho_1 \cos \theta_1 - \rho_2 \cos \theta_2)(\cos \theta_2 - \cos \theta_1) - (\rho_1 \sin \theta_1 - \rho_2 \sin \theta_2)(\sin \theta_1 - \sin \theta_2)} \\ n_3 = \frac{\rho_1 \rho_2 (\lambda_{\rho_2} - \lambda_{\rho_1}) \sin(\theta_1 - \theta_2) + (1 - \cos(\theta_1 - \theta_2))(\rho_2 \lambda_{\rho_1} + \rho_1 \lambda_{\rho_2})}{(\rho_1 \cos \theta_1 - \rho_2 \cos \theta_2)(\cos \theta_2 - \cos \theta_1) - (\rho_1 \sin \theta_1 - \rho_2 \sin \theta_2)(\sin \theta_1 - \sin \theta_2)} \end{array} \right. \quad (49)$$

où λ_{ρ_i} et λ_{θ_i} ($i = 1, 2$) sont obtenus, comme pour le cas de la droite, à partir de l'équation (7) après y avoir substitué (47) :

$$\left\{ \begin{array}{l} \lambda_{\rho_i} = \frac{\dot{\rho}_i + (1 + \rho_i^2)(\Omega_y \cos \theta_i - \Omega_x \sin \theta_i)}{V_x \cos \theta_i + V_y \sin \theta_i - \rho_i V_z} \\ \lambda_{\theta_i} = \frac{\dot{\theta}_i + \rho_i(\Omega_x \cos \theta_i + \Omega_y \sin \theta_i) + \Omega_z}{V_x \cos \theta_i + V_y \sin \theta_i - \rho_i V_z} \end{array} \right. \quad (50)$$

- **Remarque** : Chaque limbe associé au cylindre nous permet d'avoir deux équations dans le système linéaire précédent (à trois inconnues). En utilisant une seule limbe, une troisième image, correspondant à un mouvement supplémentaire de la caméra, permet également d'obtenir un nombre suffisant d'équations (en l'occurrence quatre).

c) **Estimation des paramètres du cylindre** : Les paramètres \underline{p}_0 étant connus, on peut déterminer l'ensemble des paramètres \underline{p} en se basant sur des contraintes géométriques liées au cylindre. Considérons en effet une vue de dessus du cylindre (voir figure 6) avec :

- P_i le plan image, P le plan des limbes, P_{t1} et P_{t2} les plans tangents au cylindre contenant respectivement les droites D_1 et D_2 , P_{n1} (respectivement P_{n2}) le plan orthogonal à P_{t1} (respectivement P_{t2}) passant par l'intersection de P_{t1} (respectivement P_{t2}) et P ;
- $\underline{V} = (a, b, c)$ le vecteur directeur de l'axe du cylindre, $\underline{N} = (n_1, n_2, n_3)$ la normale au plan des limbes P , $\underline{n}_{t1} = (\cos \theta_1, \sin \theta_1, -\rho_1)$ et $\underline{n}_{t2} = (\cos \theta_2, \sin \theta_2, -\rho_2)$ les normales à P_{t1} et P_{t2} , $\underline{n}_{n1} = (u_1, v_1, w_1)$ et $\underline{n}_{n2} = (u_2, v_2, w_2)$ les normales à P_{n1} et P_{n2} .

Les coordonnées (a, b, c) de \underline{V} représentant la direction du cylindre se calculent en considérant les deux produits vectoriels suivants (voir figure 6) :

$$\underline{V} = \underline{N} \wedge \underline{n}_{t1} = -\underline{N} \wedge \underline{n}_{t2} \quad (51)$$

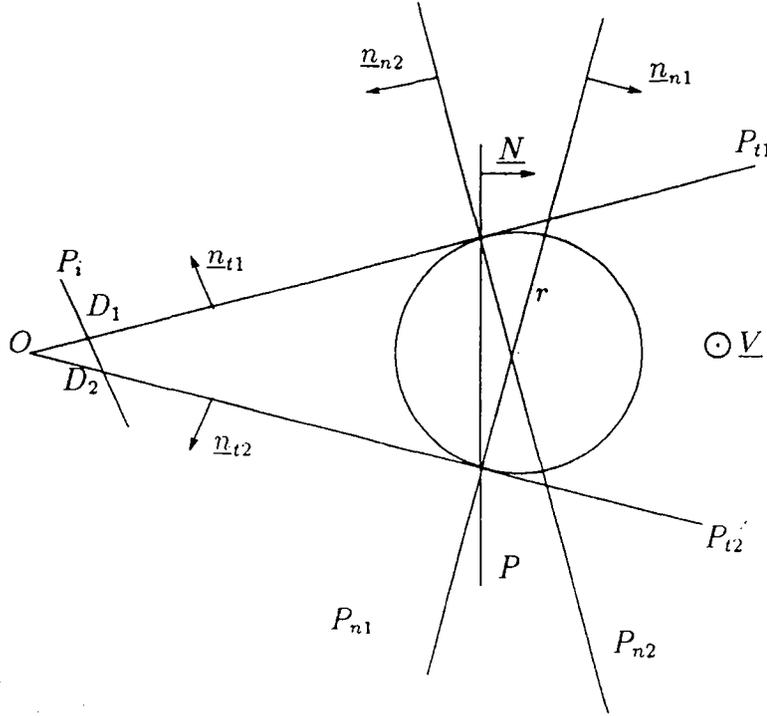


Figure 6 : Vue de dessus d'un cylindre

On en déduit :

$$\begin{cases} a = n_2(\rho_2 - \rho_1) - n_3(\sin \theta_1 - \sin \theta_2) \\ b = n_3(\cos \theta_1 - \cos \theta_2) - n_1(\rho_2 - \rho_1) \\ c = n_1(\sin \theta_1 - \sin \theta_2) - n_2(\cos \theta_1 - \cos \theta_2) \end{cases} \quad (52)$$

qui sont ensuite normalisées conformément à la contrainte (41).

Les coordonnées (x_0, y_0, z_0) de m_0 , point de l'axe de révolution du cylindre le plus proche du centre optique, sont déterminées en utilisant la contrainte (42) et le fait que m_0 appartienne à l'intersection des deux plans P_{n1} et P_{n2} normaux aux cylindre. On obtient une fois de plus un système linéaire :

$$\begin{cases} u_1 x_0 + v_1 y_0 + w_1 z_0 = -l_1 \\ u_2 x_0 + v_2 y_0 + w_2 z_0 = -l_2 \\ a x_0 + b y_0 + c z_0 = 0 \end{cases} \quad (53)$$

où les paramètres (u_i, v_i, w_i, l_i) représentant le plan $P_{ni}, i = 1, 2$ sont obtenus de la manière suivante :

- les coordonnées (u_i, v_i, w_i) de \underline{n}_{ni} en considérant le produit vectoriel suivant (voir figure 6) :

$$\underline{n}_{ni} = \underline{n}_{ti} \wedge \underline{V} \Rightarrow \begin{cases} u_i = b\rho_i + c \sin \theta_i \\ v_i = -c \cos \theta_i - a\rho_i \\ w_i = b \cos \theta_i - a \sin \theta_i \end{cases} \quad (54)$$

- l_i en calculant les coordonnées (x_i, y_i, z_i) du point particulier m_i , appartenant à P_{n_i} tel que $\underline{m}_i \in \{P_{n_i} \cap P\}$ et vérifiant en outre, comme le point m_0 , la contrainte $ax_i + by_i + cz_i = 0$. D'où :

$$\begin{cases} \cos \theta_i x_i + \sin \theta_i y_i - \rho_i z_i = 0 \\ n_1 x_i + n_2 y_i + n_3 z_i = 1 \\ ax_i + by_i + cz_i = 0 \end{cases} \quad (55)$$

dont la résolution donne :

$$\begin{cases} x_i = \frac{c \sin \theta_i + b \rho_i}{\cos \theta_i (bn_3 - cn_2) + \sin \theta_i (cn_1 - an_3) - \rho_i (an_2 - bn_1)} \\ y_i = \frac{-(c \cos \theta_i + a \rho_i)}{\cos \theta_i (bn_3 - cn_2) + \sin \theta_i (cn_1 - an_3) - \rho_i (an_2 - bn_1)} \\ z_i = \frac{b \cos \theta_i - a \sin \theta_i}{\cos \theta_i (bn_3 - cn_2) + \sin \theta_i (cn_1 - an_3) - \rho_i (an_2 - bn_1)} \end{cases} \quad (56)$$

On en déduit :

$$\begin{aligned} l_i &= -(u_i x_i + v_i y_i + w_i z_i) \\ &= -\frac{(c \sin \theta_i + b \rho_i)^2 + (c \cos \theta_i + a \rho_i)^2 + (b \cos \theta_i - a \sin \theta_i)^2}{\cos \theta_i (bn_3 - cn_2) + \sin \theta_i (cn_1 - an_3) - \rho_i (an_2 - bn_1)} \end{aligned} \quad (57)$$

La valeur de x_0, y_0 et z_0 est alors donnée par :

$$\begin{cases} x_0 = \frac{l_2 (cv_1 - bw_1) + l_1 (bw_2 - cv_2)}{u_1 (bw_2 - cv_2) + v_1 (cu_2 - aw_2) + w_1 (av_2 - bu_2)} \\ y_0 = \frac{l_2 (cu_1 - aw_1) + l_1 (aw_2 - cu_2)}{u_1 (bw_2 - cv_2) + v_1 (cu_2 - aw_2) + w_1 (av_2 - bu_2)} \\ z_0 = \frac{l_2 (bu_1 - av_1) + l_1 (av_2 - bu_2)}{u_1 (bw_2 - cv_2) + v_1 (cu_2 - aw_2) + w_1 (av_2 - bu_2)} \end{cases} \quad (58)$$

Finalement, la reconstruction du cylindre se complète par la détermination de son rayon r qui est donné par la distance entre les points m_0 et m_1 (ou m_2) :

$$r = \sqrt{(x_0 - x_1)^2 + (y_0 - y_1)^2 + (z_0 - z_1)^2} \quad (59)$$

2.3.5 Les sphères

L'équation paramétrique d'une sphère de centre m_0 de coordonnées (x_0, y_0, z_0) et de rayon r s'écrit :

$$h(\underline{x}, \underline{p}) = (x - x_0)^2 + (y - y_0)^2 + (z - z_0)^2 - r^2 = 0 \quad (60)$$

Comme pour le cas du cercle, l'image d'une sphère est caractérisée par l'équation d'une ellipse [7] :

$$K_0 X^2 + K_1 Y^2 + 2K_2 XY + 2K_3 X + 2K_4 Y + K_5 = 0 \quad (61)$$

$$\text{avec } \begin{cases} K_0 = r^2 - y_0^2 - z_0^2 \\ K_1 = r^2 - x_0^2 - z_0^2 \\ K_2 = x_0 y_0 \\ K_3 = x_0 z_0 \\ K_4 = y_0 z_0 \\ K_5 = r^2 - x_0^2 - y_0^2 \end{cases} \quad (62)$$

La fonction μ représentant la surface des limbes peut être déterminée par une démarche similaire à celle décrite précédemment dans le cas du cylindre. On obtient [7] :

$$\mu(\underline{X}, \underline{p}_0) : 1/z = aX + bY + c \quad (63)$$

$$\text{avec } \begin{cases} a = x_0 / (x_0^2 + y_0^2 + z_0^2 - r^2) \\ b = y_0 / (x_0^2 + y_0^2 + z_0^2 - r^2) \\ c = z_0 / (x_0^2 + y_0^2 + z_0^2 - r^2) \end{cases} \quad (64)$$

où $\underline{p}_0 = (a, b, c)$ représente ici encore la normale au plan des limbes de la sphère.

Comme pour le cas du cercle, on choisit une représentation de l'ellipse par les moments et les paramètres \underline{p}_0 sont obtenus par la résolution aux moindres carrés du système linéaire défini à partir de (34).

Les relations entre les paramètres $\underline{p} = (x_0, y_0, z_0, r)$ et $\underline{P} = (X_c, Y_c, \mu_{11}, \mu_{20}, \mu_{02})$ vont nous permettre, connaissant \underline{p}_0 , de déterminer \underline{p} . Ces relations sont données par :

$$\begin{cases} X_c = x_0 z_0 / (z_0^2 - r^2) \\ Y_c = y_0 z_0 / (z_0^2 - r^2) \\ \mu_{11} = x_0 y_0 r^2 / (z_0^2 - r^2)^2 \\ \mu_{20} = r^2 [X_c^2 / z_0^2 + 1 / (z_0^2 - r^2)] \\ \mu_{02} = r^2 [Y_c^2 / z_0^2 + 1 / (z_0^2 - r^2)] \end{cases} \quad (65)$$

En additionnant deux à deux les deux dernières équations de (65), on obtient un polynôme du quatrième degré en z_0 :

$$z_0^4 (\mu_{20} + \mu_{02}) + z_0^2 r^2 (\mu_{20} + \mu_{02} + X_c^2 + Y_c^2 + 2) + r^4 (X_c^2 + Y_c^2) = 0 \quad (66)$$

dont le discriminant Δ peut s'écrire, après quelques développements, sous la forme :

$$\Delta = r^4 \Delta' \text{ avec } \Delta' = (\mu_{20} + \mu_{02} - X_c^2 - Y_c^2)^2 + 4(\mu_{20} + \mu_{02} + X_c^2 + Y_c^2) + 4 \quad (67)$$

Le discriminant Δ est strictement positif, (66) admet donc toujours deux solutions réelles qui sont :

$$z_0^2 = \lambda^2 r^2 \quad (68)$$

avec :

$$\lambda = \sqrt{\frac{\mu_{20} + \mu_{02} + X_c^2 + Y_c^2 + 2 \pm \sqrt{\Delta'}}{2(\mu_{20} + \mu_{02})}}$$

On peut vérifier que l'une des deux valeurs possibles de λ est inférieure à 1 quelle que soit la configuration de la sphère, l'autre étant par contre supérieure à 1. Nous rejetons bien évidemment la première solution puisque physiquement irréalisable (le centre de la sphère doit être devant le plan image $\Rightarrow \lambda > 1$).

On peut à présent calculer les paramètres $\underline{p} = (x_0, y_0, z_0, r)$ en utilisant les équations (64). On obtient :

$$\begin{cases} x_0 = az_0/c \\ y_0 = bz_0/c \\ z_0 = [(a^2 + b^2 + c^2)z_0^2 - c^2r^2]/c \end{cases} \quad (69)$$

D'où, en utilisant (68) dans cette dernière équation :

$$\begin{cases} r = c\lambda/[\lambda^2(a^2 + b^2 + c^2) - c^2] \\ z_0 = \lambda r \\ x_0 = a\lambda r/c \\ y_0 = b\lambda r/c \end{cases} \quad (70)$$

2.4 Résultats de simulation

La méthode de reconstruction 3D décrite dans le paragraphe 2.3 a été testée en simulation sur toutes les primitives qui ont été évoquées précédemment : point, droite, cercle, sphère et cylindre. Nous ne présentons ici que les résultats portant sur la reconstruction d'un cylindrique et d'une sphère.

• Cas du cylindre

La figure 7 illustre dans sa partie gauche la position initiale de la caméra par rapport à un cylindre. A droite est représentée l'image correspondante.

A partir de cette position, on applique un mouvement quelconque à la caméra (en l'occurrence $V_x = V_y = V_z = 0.5$ cm/s, $\Omega_x = \Omega_y = \Omega_z = 3$ dg/s) et, à chaque nouvelle acquisition d'image, on mesure les paramètres $\underline{P} = (\rho_1, \theta_1, \rho_2, \theta_2)$ des deux droites observées dans l'image, leur déplacement dans l'image $\Delta\underline{P}$ par rapport à leur valeur précédente ainsi que le déplacement de la caméra. On est alors en mesure d'estimer à chaque itération l'ensemble des paramètres qui caractérisent le cylindre. Rappelons que les paramètres que nous recherchons ici sont les composantes (a, b, c) du vecteur directeur de l'axe du cylindre, son rayon r et les coordonnées (x_0, y_0, z_0) du point de son axe le plus proche du centre optique (voir paragraphe 2.3.4). Les résultats obtenus sont montrés sur la figure 8 : à gauche, les différentes valeurs de a, b, c (en haut) et x_0, y_0, z_0, r en cm (en bas) calculées à chaque itération et exprimées dans un repère de référence fixe (les valeurs devraient donc être constantes puisque le cylindre est fixe) et, à droite, l'erreur entre ces valeurs estimées et les valeurs réelles.

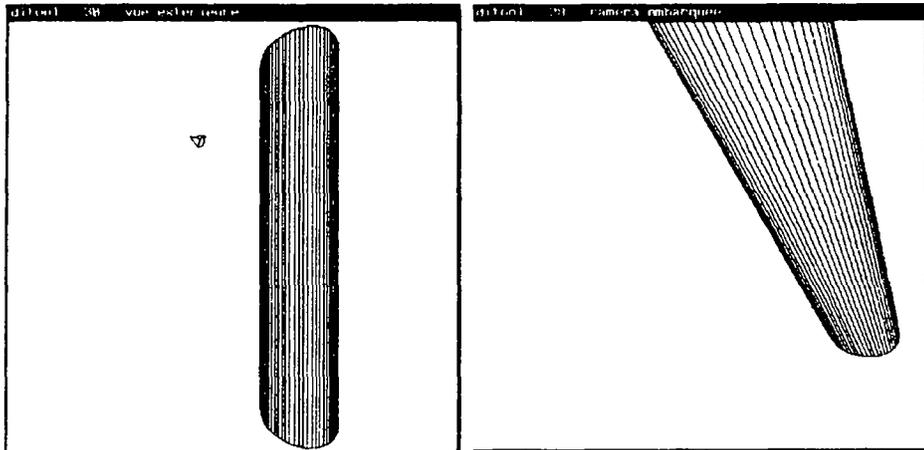


Figure 7 : Position entre la caméra et le cylindre

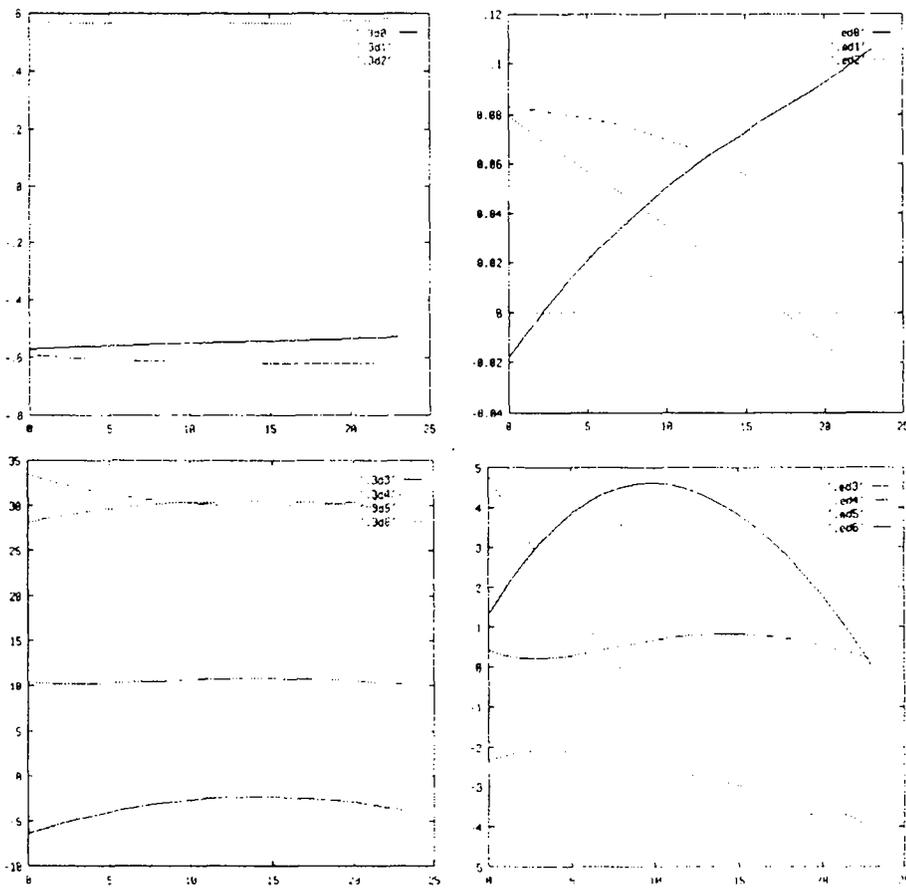


Figure 8 : Reconstruction d'un cylindre par vision dynamique

- Cas de la sphère

Nous montrons sur la figure 9, configurée comme la figure 7, la position initiale choisie entre la caméra et la sphère. Comme pour le cas précédent du cylindre, la

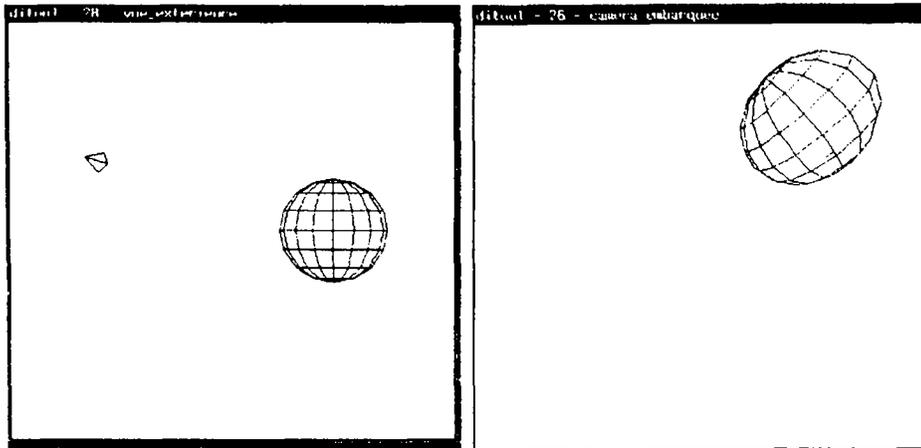


Figure 9 : Position entre la caméra et la sphère

caméra est ensuite animée d'un mouvement quelconque. On estime ici, à chaque itération, les coordonnées x_0, y_0, z_0 du centre de la sphère et son rayon r . Les résultats obtenus sont décrits sur la figure 10, configurée comme la figure 8.

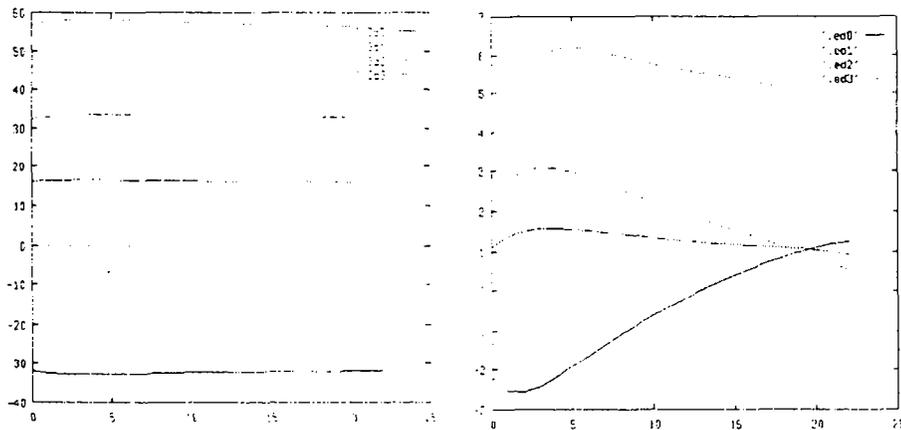


Figure 10 : Reconstruction d'une sphère par vision dynamique

Les erreurs obtenues sont relativement importantes pour l'ensemble des paramètres du cylindre et de la sphère. Notons que le même phénomène a été observé pour la totalité des primitives que nous avons étudiées. Ces erreurs sont dues, comme on le verra par la suite, au phénomène de discrétisation. Il est donc clair que sans stratégies relatives au mouvement de la caméra, notre approche s'avère inefficace pour reconstruire précisément les primitives de la scène. Ceci nous incite à aborder le problème sous un autre aspect : la vision active, qui nous permettra d'améliorer de manière très significative les résultats obtenus.

3 Reconstruction 3D par vision active

3.1 Introduction

La vision active est un concept récent défini comme un processus de perception intelligente de données. Plus précisément, la vision active consiste à élaborer des stratégies de perception afin d'améliorer la connaissance de l'environnement. Une analyse théorique qui démontre les potentialités très prometteuses de la vision active est donnée en [2] et [3]. En ce qui nous concerne, nous désirons appliquer cette approche au problème de la reconstruction 3D des primitives géométriques paramétrables. Notre objectif est de déterminer des positions et mouvements adéquats de la caméra qui permettent une meilleure estimation de la structure spatiale des primitives. La qualité de l'estimation à partir d'une séquence d'images est en effet très sensible aux mouvements successifs de la caméra [9]. Ce problème sera abordé sous les deux aspects suivants :

1. Suppression des effets des erreurs de discrétisation observés dans les résultats précédents ;
2. Minimisation des effets des erreurs de mesure : les erreurs de mesure sont inhérentes à tout algorithme de traitement d'images et, de plus, les imprécisions sur la mesure du mouvement de la caméra sont souvent importantes. Nous montrerons dans le paragraphe 3.3 qu'il existe des configurations et mouvements de la caméra qui permettent une reconstruction plus robuste des primitives.

3.2 Suppression des erreurs de discrétisation

La méthode de reconstruction 3D par vision dynamique que nous avons décrite dans la première partie de ce rapport est notamment basée sur la mesure de $\dot{\underline{P}}$, la vitesse des paramètres représentant la primitive considérée dans l'image. Rappelons en effet que les paramètres de la surface des limbes sont de la forme (voir (9)) :

$$\underline{p}_0 = \underline{p}_0(\underline{P}, \dot{\underline{P}}, T) \quad (70)$$

La mesure de $\dot{\underline{P}}$ est impossible à effectuer exactement puisque les acquisitions d'images sont effectuées à des instants discrets. Seule $\Delta\underline{P}$, la variation des valeurs de \underline{P} entre deux ou plusieurs images, peut être déterminée. L'utilisation de $\frac{\Delta\underline{P}}{\Delta t}$ dans (70) à la place de $\dot{\underline{P}}$ peut entraîner des erreurs sur l'estimation des paramètres \underline{p}_0 . Celles-ci se répercutent bien évidemment sur l'estimation des paramètres \underline{p} représentant la primitive dans la scène, d'où le biais observé dans les résultats du chapitre précédent.

Nous allons donc déterminer les contraintes à fixer sur le mouvement de la caméra afin de supprimer ce biais. En effet, si l'on peut assurer que :

$$\dot{\underline{P}} = \frac{\Delta\underline{P}}{\Delta t}, \quad \forall t \quad (71)$$

la discrétisation effectuée n'aura aucun effet sur les algorithmes d'estimation. Cette condition est satisfaite si et seulement si :

$$\underline{\ddot{P}} = \dots = \underline{P}^{[n]} = 0, \forall t \quad (72)$$

Revenons à présent à l'expression de $\underline{\dot{P}}$. On a d'après (7) :

$$\underline{\dot{P}} = f(\underline{P}, \underline{p}_0, T) \quad (73)$$

d'où :

$$\underline{\ddot{P}} = \frac{\partial f}{\partial \underline{P}} \underline{\dot{P}} + \frac{\partial f}{\partial \underline{p}_0} \underline{\dot{p}_0} + \frac{\partial f}{\partial T} \dot{T} \quad (74)$$

On peut considérer que l'accélération \dot{T} est nulle en choisissant pour T la vitesse moyenne de la caméra, calculée à partir de la mesure des positions de la caméra à chaque acquisition d'image. Une condition générale et suffisante (mais non forcément nécessaire) pour respecter la condition (72) est donc de contraindre les mouvements de la caméra de telle sorte que :

$$\underline{\dot{P}} = \underline{\dot{p}_0} = 0, \forall t \quad (75)$$

Cette condition signifie que l'image et la surface des limbes de la primitive (ou plus précisément les paramètres les représentant) doivent rester constantes le long de la trajectoire de la caméra. Pour le cas d'une sphère qui apparaît sous la forme d'un cercle centré dans l'image par exemple, on peut aisément montrer que cette condition signifie que la caméra doit se déplacer sur une sphère concentrique tout en pointant sur le centre de la sphère. Nous rappelons en annexe de ce rapport qu'il est possible, en utilisant l'approche asservissement visuel, d'établir des lois de commande en boucle fermée sur la vision calculant automatiquement les mouvements de la caméra vérifiant de telles contraintes.

• Remarques :

1. La condition (75) que nous avons établie n'est pas nécessaire et suffisante. En effet, pour chaque primitive, il existe des configurations telles que :

$$\frac{\partial f}{\partial \underline{P}} \underline{\dot{P}} + \frac{\partial f}{\partial \underline{p}_0} \underline{\dot{p}_0} = 0 \text{ avec } \underline{\dot{P}} \neq 0 \text{ et } \underline{\dot{p}_0} \neq 0 \quad (76)$$

Pour le cas du point, on rappelle qu'on a :

$$\dot{z} = -V_z - zY\Omega_x + zX\Omega_y \quad (77)$$

et on obtient en différenciant (11) :

$$\begin{cases} \ddot{X} = \frac{\dot{z}}{z^2}(V_x - XV_z) + \dot{X}\left(\frac{V_x}{z} + Y\Omega_x - 2X\Omega_y\right) + \dot{Y}(X\Omega_x + \Omega_z) \\ \ddot{Y} = \frac{\dot{z}}{z^2}(V_y - YV_z) + \dot{X}(-Y\Omega_y - \Omega_z) + \dot{Y}\left(\frac{V_y}{z} + 2Y\Omega_x - X\Omega_y\right) \end{cases} \quad (78)$$

\ddot{X} et \ddot{Y} s'annulent par exemple si la caméra n'est animée que de mouvements de translation ($\Omega_x = \Omega_y = \Omega_z = 0$). On a alors $\dot{z} = -V_z \neq 0$ si $V_z \neq 0$ et $\dot{X} = (-V_x + X V_z)/z$, $\dot{Y} = (-V_y + Y V_z)/z$ qui ne peuvent s'annuler simultanément pour permettre la reconstruction du point. D'autres cas particuliers ont également pu être déterminés. La recherche de l'ensemble de ces configurations paraît cependant peu intéressante puisque particulière à chaque primitive. On préfère donc se limiter à la condition (75) qui a le mérite d'être valide quelle que soit la primitive considérée.

2. On peut également remarquer que la condition $\dot{P} = 0$ entraîne $\dot{p}_0 = 0$ pour les primitives de type sphère, cylindre et cercle (excepté le cas particulier où l'image du cercle est un cercle centré). Pour ces primitives, la condition $\dot{P} = 0$ est donc suffisante pour supprimer les erreurs de discrétisation. Cela n'est pas le cas, par contre, pour les primitives de type point et droite dont la dimension est inférieure.

Résultats de simulation

• Cas du cylindre :

La figure 11 configurée comme la figure 8 montre les résultats de simulation obtenus en appliquant cette stratégie. Le processus de reconstruction a été déclenché à partir de la même position initiale de la caméra (voir figure 7). On peut observer que les erreurs de discrétisation disparaissent complètement après les quelques premières itérations nécessaires à la convergence de la commande.

Si l'on introduit artificiellement du bruit sur les mesures image (bruit uniforme de 3 pixels pour des images 512×512) et sur la vitesse de la caméra (bruit uniforme de 2.5% sur chaque composante) pour simuler une mauvaise calibration et des erreurs de mesure sur la position de la caméra et les informations extraites des images, on obtient les résultats présentés sur la figure 12.

• Cas de la sphère :

Sur les figures 13 (sans bruit) et 14 (avec bruit) sont représentés les résultats correspondant à la reconstruction d'une sphère en appliquant la même stratégie d'annulation des effets des erreurs de discrétisation.

Ces résultats révèlent l'intérêt d'imposer des contraintes sur le mouvement de la caméra pour estimer la structure des primitives. Signalons que des résultats similaires ont été obtenus pour les primitives de type point, droite et cercle. Nous allons voir à présent que l'on peut améliorer les résultats bruités obtenus en se focalisant sur l'aspect robustesse de la reconstruction.

3.3 Minimisation des effets des erreurs de mesure

Intuitivement, on peut penser que la configuration d'une primitive dans l'image influence sur la qualité de son estimation. Un moyen naturel d'aborder ce problème consiste à

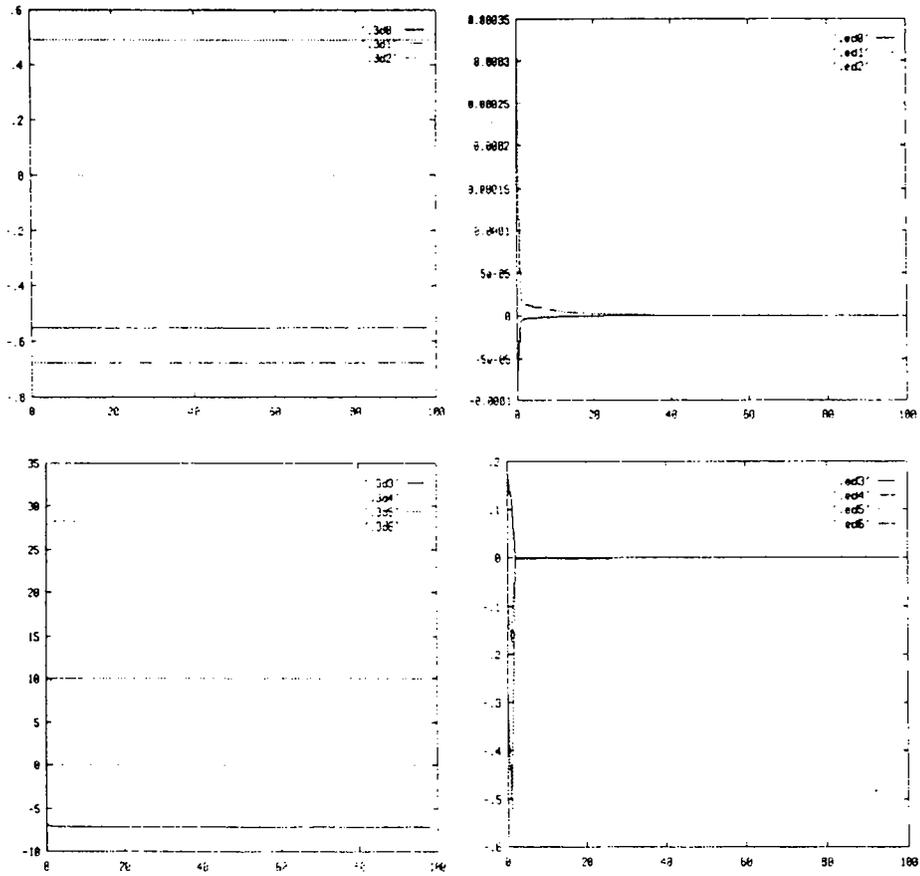


Figure 11 : Reconstruction d'un cylindre en supprimant l'erreur de discrétisation (sans bruit)

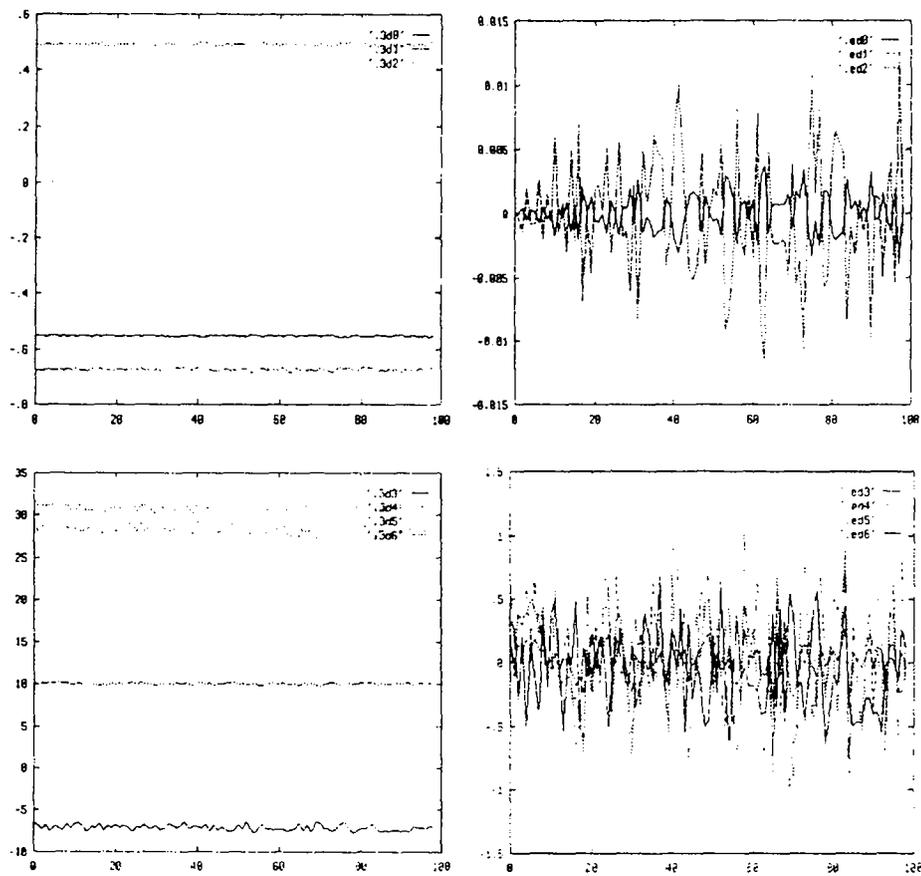


Figure 12 : Reconstruction d'un cylindre en supprimant l'erreur de discrétisation (avec bruit)

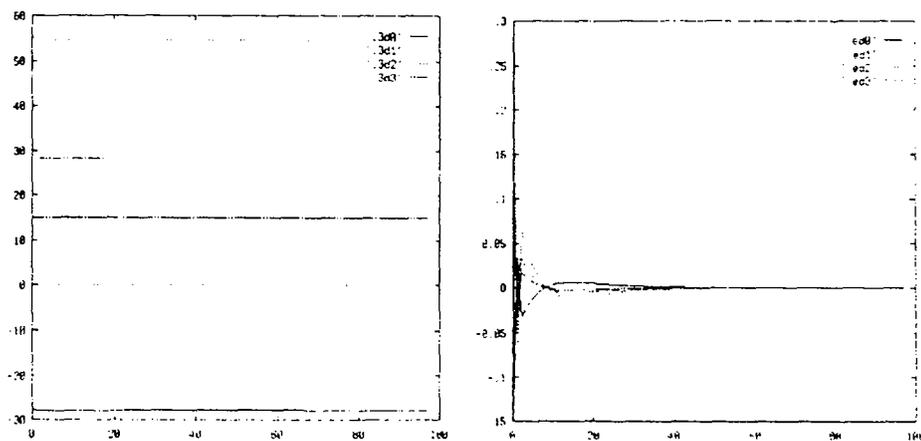


Figure 13 : Reconstruction d'une sphère en supprimant l'erreur de discrétisation (sans bruit)

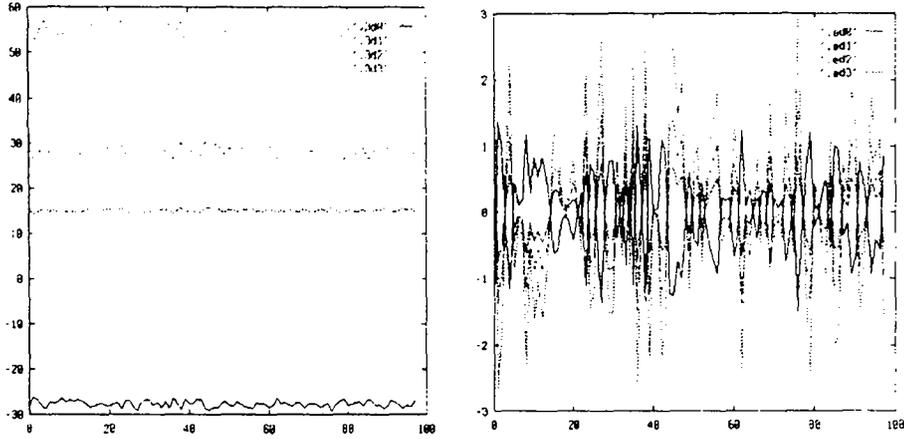


Figure 14 : Reconstruction d'une sphère en supprimant l'erreur de discrétisation (avec bruit)

minimiser les erreurs d'estimation sur les paramètres \underline{p} , fonction des erreurs de mesure effectuées. Rappelons que \underline{p} est de la forme :

$$\underline{p} = \underline{p}(\underline{P}, \dot{\underline{P}}, T) \quad (79)$$

Les erreurs sur \underline{p} sont donc étroitement liées aux erreurs de mesure sur les informations visuelles \underline{P} et $\dot{\underline{P}}$ et sur la vitesse de la caméra T . Soit p un des paramètres représentant la primitive à reconstruire. Si l'on suppose que les erreurs de mesure sur \underline{P} , $\dot{\underline{P}}$ et T sont décorréelées, l'incertitude σ_p sur l'estimation de p peut s'exprimer sous la forme suivante :

$$(\sigma_p)^2 = \sum_{i=1}^m \left(\frac{\partial p}{\partial P_i} \right)^2 (\sigma_{P_i})^2 + \sum_{j=1}^m \left(\frac{\partial p}{\partial \dot{P}_j} \right)^2 (\sigma_{\dot{P}_j})^2 + \sum_{k=1}^6 \left(\frac{\partial p}{\partial T_k} \right)^2 (\sigma_{T_k})^2 \quad (80)$$

Notons $p_{a_i} = \left(\frac{\partial p}{\partial a_i} \right)^2$, $a_i \in \{\underline{P}, \dot{\underline{P}}, T\}$. La minimisation de $(\sigma_p)^2$ par rapport à une configuration \underline{P} de la primitive dans l'image se ramène alors à trouver l'ensemble des paramètres P_j ($j = 1$ à m) tels que :

$$\left(\frac{\partial p_{a_i}}{\partial P_j} \right) = 0, \quad \forall a_i \text{ et } \forall j = 1 \text{ à } m \quad (81)$$

La résolution explicite de tels systèmes non linéaires s'avère extrêmement délicate aussi avons-nous procédé de la manière suivante :

1. on sélectionne les configurations de la primitive dans l'image qui semblent intuitivement entraîner une meilleure estimation (celles qui, par exemple, simplifient au mieux les expressions analytiques) ;
2. on vérifie ensuite si chaque équation du système (81) est satisfaite. On peut ainsi trouver des configurations optimales permettant une reconstruction robuste.

Cas du point : Nous cherchons à minimiser ici l'incertitude σ_z sur la profondeur z du point à reconstruire. Celle-ci peut être calculée à partir de l'équation (14) en utilisant la relation (80) avec $\underline{P} = (X, Y)$. La configuration choisie comme candidat à fournir une estimation de z robuste est bien évidemment telle que le point apparaisse centré dans l'image, soit $\underline{P} = (0, 0)$. On peut montrer que :

$$X = Y = 0 \Rightarrow \begin{cases} \frac{\partial p_{a_i}}{\partial X} = 0 \\ \frac{\partial p_{a_i}}{\partial Y} = 0 \end{cases} \iff V_z = \Omega_z = 0 \quad (82)$$

Outre les contraintes sur le mouvement de la caméra pour conserver en permanence la configuration $X = Y = 0$, des contraintes supplémentaires ($V_z = \Omega_z = 0$) sont donc nécessaires pour que cette configuration minimise les effets des erreurs de mesure sur la reconstruction du point. Signalons que s'il nous a été impossible de prouver que cette configuration est le seul minimum (en raison de la complexité du système (81)), nous avons pu vérifier que certaines configurations (par exemple $X = 1, Y = 0$) n'annuleraient pas, quels que soient les mouvements de la caméra, l'ensemble des coefficients $\frac{\partial p_{a_i}}{\partial P_j}$ et ne minimisaient donc pas l'incertitude sur z .

Autres primitives : Le même principe a été appliqué aux autres primitives que nous avons étudiées et nous avons obtenu les résultats suivants :

- la reconstruction d'une droite est optimale quand celle-ci apparaît centrée horizontale ou verticale dans l'image ($\rho = 0, \theta = k\frac{\pi}{2}$) avec en outre $V_z = 0$ et $V_x = \Omega_y = 0$ si la droite est horizontale ou $V_y = \Omega_x = 0$ si elle est verticale.
- la reconstruction d'un cylindre est optimale quand celui-ci se projette en deux droites symétriques parallèles verticales ou horizontales dans l'image ($\rho_1 = -\rho_2, \theta_1 = \theta_2 = k\frac{\pi}{2}$) avec en outre $V_x = 0$ si les droites sont horizontales ou $V_y = 0$ si elles sont verticales.
- la reconstruction d'un cercle ou d'une sphère est optimale quand leur projection dans l'image est un cercle centré ($X_c = Y_c = \mu_{11} = 0, \mu_{20} = \mu_{02}$) si, en outre, $\Omega_z = 0$.

L'approche asservissement visuel (voir l'annexe de ce rapport) est ici encore parfaitement adaptée pour générer automatiquement les mouvements de la caméra permettant d'obtenir des configurations particulières entre la caméra et la primitive considérée (tels par exemple que l'image d'un point soit au centre de l'image ou l'image d'une sphère soit un cercle centré).

Résultats de simulation

- Cas du cylindre :

Nous avons choisi comme position optimale pour la reconstruction celle qui donne pour image du cylindre deux droites symétriques verticales (voir figure 15). A partir de la position initiale de la figure 7, une loi de commande détermine les mouvements de la caméra permettant de converger vers la position désirée. La figure 16 montre les résultats de la reconstruction du cylindre obtenus pendant et après la phase de convergence (celle-ci est assurée aux alentours de l'itération 20).

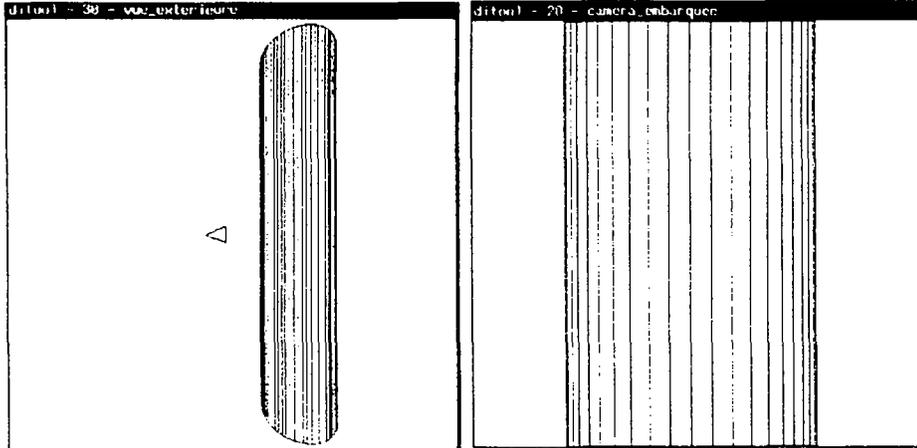


Figure 15 : Position optimale entre la caméra et le cylindre

- Cas de la sphère :

Le même principe est appliqué dans le cas de la sphère pour amener la caméra de la position visualisée figure 9 à celle visualisée figure 17 telle que l'image de la sphère soit un cercle centré. Les résultats de la reconstruction sont représentés sur la figure 18.

On peut observer que ces résultats sont assez médiocres pendant la phase de convergence. Ceci s'explique par le fait que, pendant cette phase, les mouvements de la caméra ne respectent pas les contraintes nécessaires pour annuler les effets de discrétisation. La contrainte $\dot{\underline{P}} = 0$ ne peut en effet être respectée puisque l'on souhaite amener \underline{P} à une valeur désirée différente \underline{P}_d . Par contre, lorsque la convergence est assurée, les mouvements de la caméra sont contraints pour conserver $\underline{P} = \underline{P}_d$ (d'où $\dot{\underline{P}} = 0$). De plus, il est également possible d'assurer $\dot{\underline{p}_i} = 0$ ainsi que les contraintes supplémentaires établies précédemment pour minimiser les effets des erreurs de mesure (par exemple $\Omega_z = 0$ dans le cas de la sphère). On peut vérifier que les résultats obtenus après la phase de convergence sont nettement plus robustes que ceux obtenus en appliquant la stratégie précédente où l'on supprimait l'erreur de discrétisation pour une configuration quelconque entre la caméra et la primitive à reconstruire (voir figure 12 et 14).

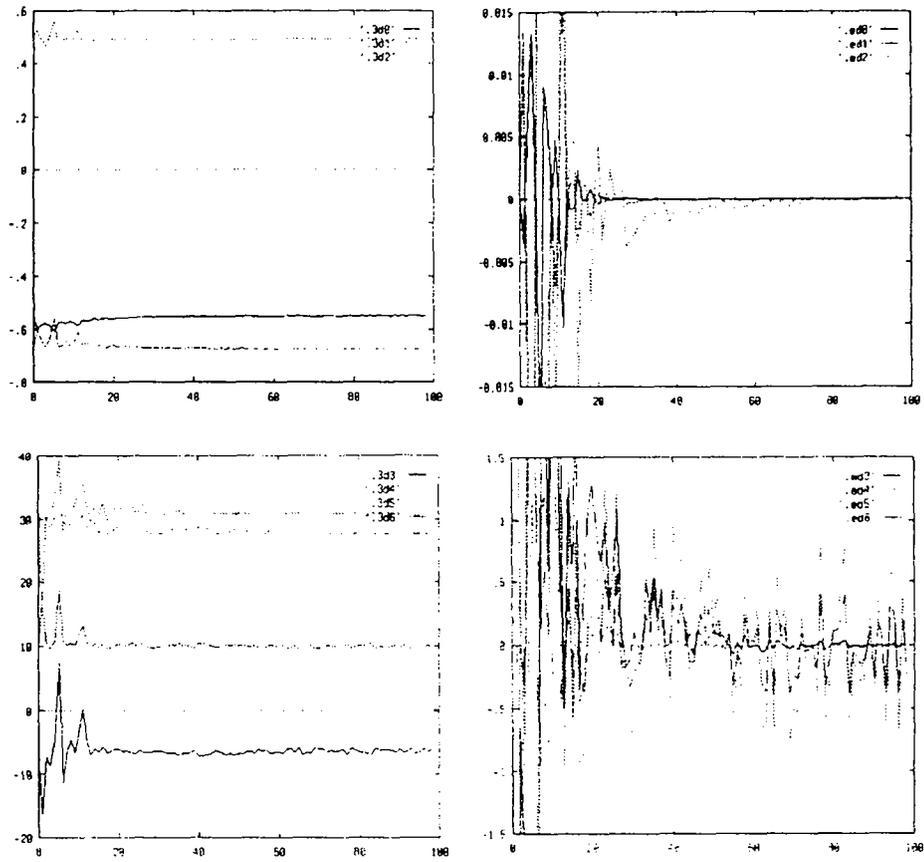


Figure 16 : Reconstruction d'un cylindre en minimisant les effets des erreurs de mesure (avec bruit)

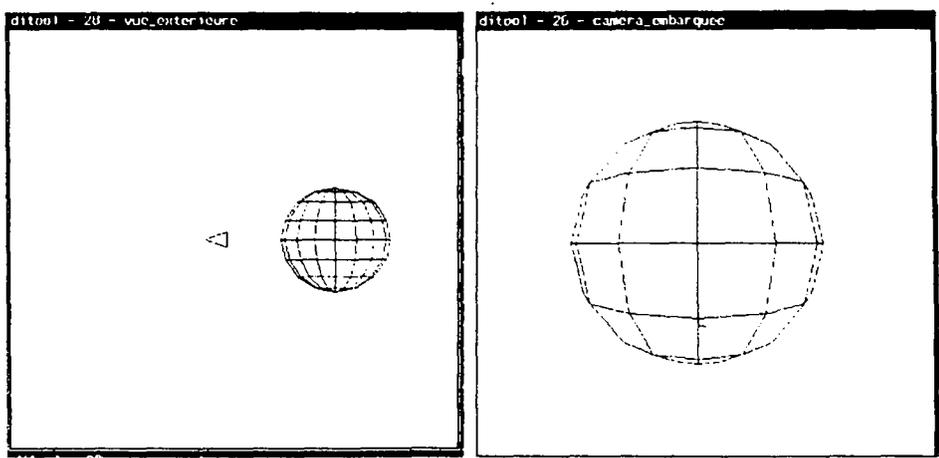


Figure 17 : Position optimale entre la caméra et la sphère

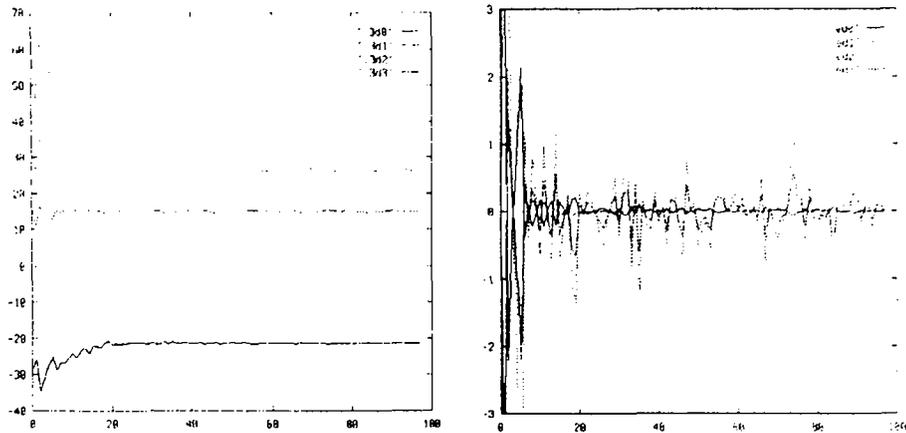


Figure 18 : Reconstruction d'une sphère en minimisant les effets des erreurs de mesure (avec bruit)

• **Remarques :**

1. Des résultats similaires ont été obtenus pour les primitives de type point et droite. Par contre, cette stratégie n'est pas applicable au cas du cercle. En effet, en raison d'une singularité isolée dans la matrice d'interaction associée au cercle lorsque l'image du cercle est un cercle centré, il n'est pas possible d'établir de loi de commande permettant de faire converger la caméra vers une position telle que le cercle se projette en un cercle centré dans l'image [7] [10].
2. Soulignons enfin qu'il ne nous a pas semblé nécessaire de faire appel à un schéma d'estimation linéaire optimal tel que le filtre de Kalman. Un tel filtrage aurait en effet ralenti la convergence de notre méthode d'estimation vers les valeurs réelles des paramètres 3D puisque la loi de commande, en convergeant vers une configuration optimale, permet d'améliorer les estimées des paramètres au cours du temps. Un simple filtre moyenneur, par exemple sur les trois ou cinq dernières valeurs, semble plus adéquat pour robustifier encore davantage les résultats obtenus.

3.4 Stratégie de reconstruction optimale

Les résultats de simulation obtenus précédemment nous ont permis de dégager une stratégie de reconstruction robuste et efficace combinant les deux stratégies précédentes. Elle consiste en trois étapes :

- reconstruire en conservant le motif initial constant dans l'image pour annuler les effets de discrétisation et ce jusqu'à avoir une estimée des paramètres 3D satisfaisante pour obtenir une loi de commande performante (voir en annexe).

- utiliser cette loi de commande pour amener la caméra vers une des configurations particulières de la primitive permettant de minimiser les effets des erreurs de mesure, l'estimation des paramètres 3D n'étant pas effectuée durant cette étape.
- relancer le processus de reconstruction afin de raffiner l'estimation sur les paramètres 3D de la primitive.

Nous présentons les résultats obtenus en appliquant cette stratégie sur les figures 19 pour la reconstruction du cylindre et 20 pour la reconstruction de la sphère.

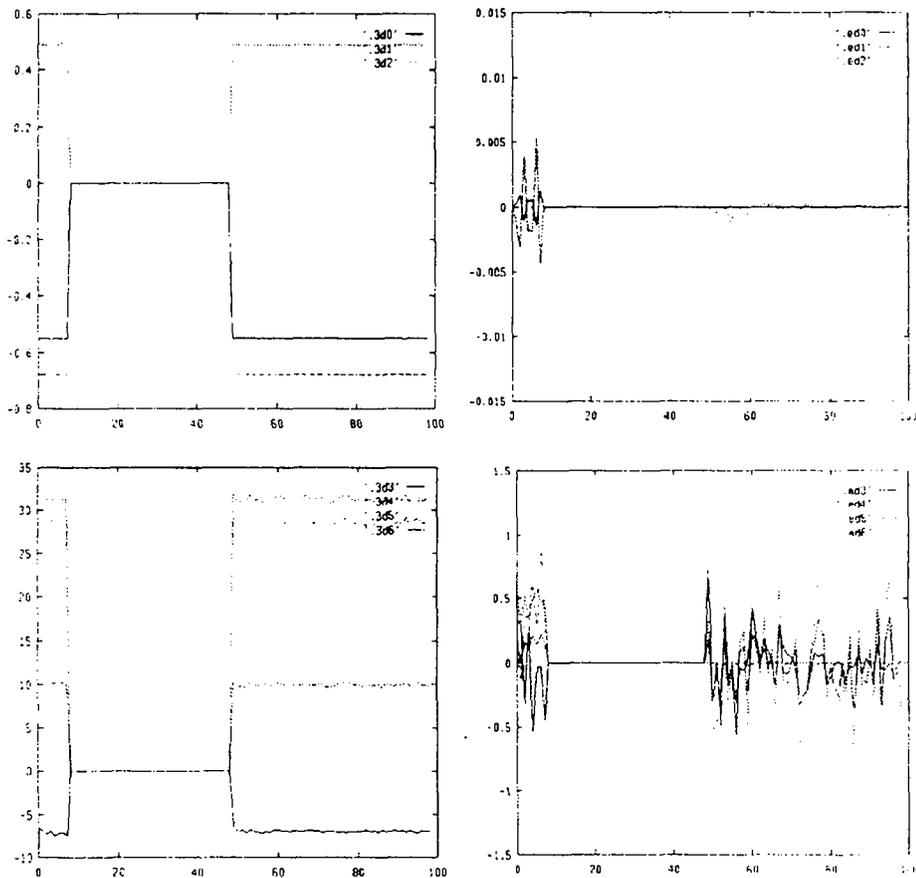


Figure 19 : Reconstruction optimale d'un cylindre

4 Conclusion

Nous avons présenté dans ce rapport une méthode générale de reconstruction des primitives géométriques paramétrables par vision dynamique. Dans notre démarche, nous avons généralisé les résultats déjà acquis concernant les primitives de type point et segment à des primitives plus complexes. Plus précisément, l'utilisation du torseur d'in

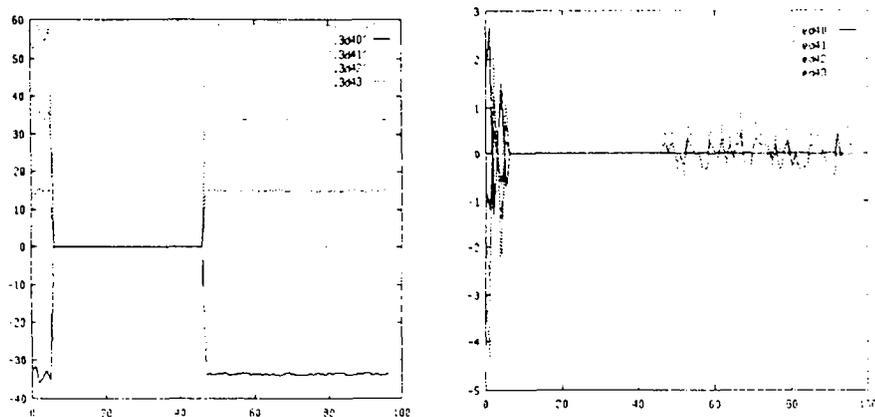


Figure 20 : Reconstruction optimale d'une sphère (avec bruit)

teraction associé aux primitives de l'environnement a permis d'établir une méthode s'appliquant aisément aux primitives telles que cylindres, sphères, etc.

Puisque la nature du mouvement de la caméra affecte la précision des résultats, nous nous sommes particulièrement intéressés à cet aspect critique de la vision dynamique. Nous avons montré qu'en contrôlant les mouvements de la caméra à l'aide de lois de commande en boucle fermée sur la vision, on pouvait obtenir de nettes améliorations dans la reconstruction et la localisation des primitives. Les résultats obtenus en simulation confirment ainsi le point de vue de travaux antérieurs sur l'avenir très prometteur de la vision active [4] [2] [3] [15]. L'implémentation des algorithmes que nous avons décrits dans ce rapport sur une cellule expérimentale de vision dynamique active constituée d'une caméra embarquée sur un robot à six degrés de liberté est en cours de réalisation. Des résultats expérimentaux devraient prochainement être obtenus. On envisage notamment la reconstruction d'un environnement similaire à celui d'une zone encombrée de nombreux tuyaux d'une centrale nucléaire.

Remerciements : Ce travail est effectué en collaboration avec l'équipe de Mr Juvin du CEA-LETI / DEIN-SLA dans le cadre de la convention 1 91 C 244 00 31315 01 1.

Annexe : Rappels sur la commande référencée vision

La vision active vise notamment à améliorer la connaissance d'une scène par des mouvements adéquats de la caméra. Une commande en boucle fermée sur des informations visuelles telle la commande référencée vision [19] [11] [10] [7] paraît donc parfaitement adaptée pour générer automatiquement ces différents mouvements. Nous n'en rappelons ici que les principes de base et renvoyons le lecteur à [14] pour une étude complète de la commande des robots manipulateurs par l'approche fonction de tâche, à [13] pour

l'application de cette approche à la commande référencée capteurs et à [10] ou [7] pour son application à la commande référencée vision.

Nous avons défini dans ce rapport des contraintes sur le mouvement de la caméra afin d'obtenir une meilleure reconstruction des primitives géométriques. Plus précisément, pour supprimer les erreurs de discrétisation, les mouvements de la caméra doivent être tels que $\underline{P} = 0, \forall t$, ce qui peut s'exprimer sous la forme $\underline{P}(t) - \underline{P}_d = 0, \forall t$ où $\underline{P}(t)$ est la valeur mesurée à chaque itération des m paramètres représentant la primitive dans l'image et où \underline{P}_d est la valeur de \underline{P} mesurée à la première itération (on aurait également pu choisir pour \underline{P}_d la valeur de \underline{P} mesurée à l'itération précédente). De même, pour minimiser les effets des erreurs de mesure, les mouvements de la caméra doivent assurer $\underline{P}(t) - \underline{P}_d = 0, \forall t$ où \underline{P}_d est ici la configuration optimale choisie de la primitive dans l'image (par exemple, pour le cas du point, $X = Y = 0$). Dans les deux cas, la fonction de tâche correspondante s'exprime sous la forme [14] [10] [7] :

$$\underline{e} = W^+ C (\underline{P}(t) - \underline{P}_d) + (\mathbb{I}_3 - W^+ W) \underline{e}_2 \quad (83)$$

où :

- \underline{e} est la fonction de tâche que l'on souhaite réguler à zéro (\underline{e} est de dimension 6 puisque l'on considère le cas général où la caméra a six degrés de liberté) ;
- C est une matrice de dimension $m \times m$ qui, idéalement, s'écrit :

$$C = W L_{\underline{P}}^{T+} (\underline{P}, \underline{p}_0) \quad (84)$$

avec :

- $L_{\underline{P}}^{T+}$, de dimension $6 \times m$, la pseudo-inverse de la matrice d'interaction associée à \underline{P} (qui correspond grossièrement au jacobien inverse de la tâche référencée vision à accomplir) ;
- W , de dimension $m \times 6$, une matrice de rang plein telle que $\text{Ker } W = \text{Ker } L_{\underline{P}}^T$.

Remarque : Les matrices $L_{\underline{P}}^{T+}$ et W dépendent des paramètres \underline{P} qui sont mesurés dans l'image mais également des paramètres \underline{p}_0 , a priori inconnus, représentant la surface des limbes de la primitive. La méthode de reconstruction que nous avons détaillée dans le paragraphe 2.3 de ce rapport permet d'en avoir une estimation $\hat{\underline{p}}_0$ qui est bien entendu utilisée pour calculer à chaque itération $L_{\underline{P}}^{T+}$ et W .

- \underline{e}_2 est une tâche secondaire (dans notre cas un suivi de trajectoire) construite afin de permettre des mouvements de la caméra tout en assurant la contrainte $\underline{P}(t) = \underline{P}_d$. Des déplacements de la caméra sont en effet en permanence indispensables pour pouvoir estimer la structure des primitives. Par exemple, pour le cas de la sphère (qui doit apparaître sous la forme d'un cercle centré dans l'image), la tâche secondaire va consister à déplacer la caméra sur une sphère concentrique afin de reconstruire la sphère tout en l'observant sous la forme d'un cercle centré

dans l'image. Cette tâche secondaire permet également d'assurer les contraintes supplémentaires que nous avons établies, à savoir $\dot{p}_0 = 0$ pour supprimer les erreurs de discrétisation et celles du type $V_z = \Omega_z = 0$ (pour le cas du point) qui fournissent une estimation plus robuste.

- W^+ et $\mathbb{I}_6 - W^+W$ sont des opérateurs qui assurent la compatibilité de la tâche secondaire \underline{e}_2 avec la contrainte $\underline{P}(t) = \underline{P}_d$

La fonction de tâche \underline{e} étant construite, une commande simplifiée, qui calcule la vitesse désirée de la caméra T_c et assure une décroissance exponentielle de \underline{e} , est donnée par [10] :

$$T_c = -\lambda \underline{e} - \frac{\partial \underline{e}}{\partial t}, \quad \lambda > 0 \quad (85)$$

où λ est un gain fixé par l'utilisateur dépendant de la fréquence d'échantillonnage de la boucle de commande et où $\frac{\partial \underline{e}}{\partial t}$ peut ici être approximée par $(\mathbb{I}_6 - W^+W) \frac{\partial \underline{e}_2}{\partial t}$ puisque la scène est supposée fixe (on suppose en outre que W et C sont lentement variables de telle sorte que leur variation soit négligeable).

Nous détaillons à présent la construction de la fonction de tâche correspondant à la reconstruction du point. Dans ce cas, on a $\underline{P} = (X, Y)$ et (voir (12)) :

$$L_{\underline{P}}^T = \begin{pmatrix} -1/z & 0 & X/z & XY & -(1+X^2) & Y \\ 0 & -1/z & Y/z & 1+Y^2 & -XY & -X \end{pmatrix} \quad (86)$$

$L_{\underline{P}}^T$ est toujours de rang plein 2 aussi choisit-on $W = L_{\underline{P}}^T$. On peut par ailleurs facilement calculer la pseudo-inverse $L_{\underline{P}}^{T+}$ de $L_{\underline{P}}^T$ (vérifiant $L_{\underline{P}}^T L_{\underline{P}}^{T+} = \mathbb{I}_2$) et on obtient :

$$L_{\underline{P}}^{T+} = W^+ = \frac{1}{D} \begin{pmatrix} -z(1+Y^2) & zXY \\ zXY & -z(1+X^2) \\ zX & zY \\ 0 & z^2E \\ -z^2E & 0 \\ z^2YE & -z^2XE \end{pmatrix} \quad (87)$$

avec $D = E(1+z^2E)$ et $E = (1+X^2+Y^2)$. La tâche secondaire \underline{e}_2 est ici choisie égale à :

$$\underline{e}_2 = \begin{pmatrix} x - x_0 - V_1 t \\ y - y_0 - V_2 t \\ z - z_0 + (XV_1 + YV_2)t \\ 0 \\ 0 \\ 0 \end{pmatrix} \quad (88)$$

où x_0, y_0 et z_0 sont les positions initiales de la caméra et où V_1 et V_2 sont deux constantes choisies par l'utilisateur afin d'effectuer un mouvement à vitesse constante dans les directions \vec{x} et \vec{y} du repère de la caméra (l'un de ces deux mouvements au moins

est nécessaire pour permettre la reconstruction du point). Le choix de la troisième composante de \underline{e}_2 permet, lui, d'assurer la contrainte supplémentaire $\dot{z} = 0$ supprimant les erreurs de discrétisation. En effet, si l'on suppose pour simplifier les calculs que la tâche est parfaitement réalisée ($\underline{e} = 0$), la consigne de la caméra prend pour valeur :

$$T_c = \begin{pmatrix} V_x \\ V_y \\ V_z \\ \Omega_x \\ \Omega_y \\ \Omega_z \end{pmatrix} = -(\mathbb{I}_6 - W^+W) \frac{\partial \underline{e}_2}{\partial t} = (\mathbb{I}_6 - W^+W) \begin{pmatrix} V_1 \\ V_2 \\ -XV_1 - YV_2 \\ 0 \\ 0 \\ 0 \end{pmatrix} \quad (89)$$

avec :

$$\mathbb{I}_6 - W^+W = \frac{1}{D} \begin{pmatrix} D-1-Y^2 & XY & X & 0 & -zE & zYE \\ XY & D-1-X^2 & Y & zE & 0 & -zXE \\ X & Y & D-X^2-Y^2 & -zYE & zXE & 0 \\ 0 & zE & -zYE & (1+z^2X^2)E & z^2XYE & z^2XE \\ -zE & 0 & zXE & z^2XYE & (1+z^2Y^2)E & z^2YE \\ zYE & -zXE & 0 & z^2XE & z^2YE & (1+z^2)E \end{pmatrix}$$

et on peut alors vérifier que $\dot{z} = -V_z - zY\Omega_x + zX\Omega_y = 0$.

Pour le cas particulier du point centré dans l'image ($X_d = Y_d = 0$), on a lorsque la tâche est parfaitement réalisée ($X = Y = 0$) :

$$T_c = \frac{1}{1+z^2} \begin{pmatrix} -z^2 & 0 & 0 & 0 & -z & 0 \\ 0 & -z^2 & 0 & z & 0 & 0 \\ 0 & 0 & 1+z^2 & 0 & 0 & 0 \\ 0 & z & 0 & 1 & 0 & 0 \\ -z & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1+z^2 \end{pmatrix} \begin{pmatrix} V_1 \\ V_2 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} = \frac{z}{1+z^2} \begin{pmatrix} -zV_1 \\ -zV_2 \\ 0 \\ V_2 \\ -V_1 \\ 0 \end{pmatrix} \quad (90)$$

qui permet d'animer la caméra d'un mouvement à vitesse constante dans la direction \vec{x} d'amplitude $-z^2V_1/(1+z^2)$ (respectivement $-z^2V_2/(1+z^2)$ dans la direction \vec{y}) compensé par une rotation autour de Ω_y (respectivement Ω_x) pour conserver l'image du point au centre de l'image. La contrainte $\dot{z} = 0$ est bien entendu assurée, de même que $V_z = \Omega_z = 0$ qui minimisent les effets des erreurs de mesure.

Bibliographie

- [1] G. Adiv : *Determining Three-Dimensional Motion and Structure from Optical Flow Generated by Several Moving Objects*, IEEE Trans. on PAMI, Vol 7, n. 4, pp. 384-401, Juillet 1985.
- [2] J. Aloimonos, A. Bandyopadhyay : *Active Vision*, 1st Int. Conf. on Computer Vision, pp. 35-54, Londres, GB, Juin 1987.

- [3] R. Bajcsy : *Active Perception*, Proc. of the IEEE, Vol 76, n. 8, pp. 996-1005, Août 1988.
- [4] A. Bandyopadhyay, B. Chandra, D.H. Ballard : *Egomotion using Active Vision*, Int. Conf on CVPR 86, pp. 498-503, Miami Beach, Floride, USA, Juin 1986.
- [5] B. Boukarri : *Reconstruction 3D récursive de scènes structurées au moyen d'une caméra mobile. Application à la robotique*, Thèse de l'Université d'Orsay, CEA, France, Octobre 1989.
- [6] P. Bouthemy : *A Maximum Likelihood Framework for Determining Moving Edges*, IEEE Trans. on PAMI, Vol 11, n. 5, pp. 499-511, Mai 1989.
- [7] F. Chaumette : *La relation vision-commande: théorie et application à des tâches robotiques*, Thèse de l'Université de Rennes I, France, Juillet 1990.
- [8] C.H. Chien, J.K. Aggarwal : *Model Construction and Shape Recognition from Occluding Contours*, IEEE Trans. on PAMI, Vol 11, n. 4, pp. 372-389, Avril 1989.
- [9] B. Espiau, P. Rives : *Closed-Loop Recursive Estimation of 3D Features for a Mobile Vision System*, IEEE Int. Conf. on Robotics and Automation, Raleigh, North Carolina, USA, Vol. 3, pp. 1436-1443, Avril 1987.
- [10] B. Espiau, F. Chaumette, P. Rives : *A new approach to visual servoing in robotics*, Rapport de recherche IRISA . n. 542, France, Juillet 1990 également à paraître dans IEEE Trans. on Robotics and Automation, Vol 8, n.3, Juin 1992.
- [11] J.T. Feddema, O.R. Mitchell : *Vision-Guided Servoing with Feature-Based Trajectory Generation*, IEEE Trans. on Robotics and Automation, Vol. 5, n. 5, pp. 691-700, Octobre 1989.
- [12] K. Price : *Multi-frame Feature-Based Motion Analysis*, 10th Int. Conf. on Pattern Recognition, Vol 1, pp. 114-117, Juin 1990.
- [13] C. Samson, B. Espiau : *Application of the Task Function Approach to Sensor-Based-Control of Robot Manipulators*, 11th IFAC World Congress, Vol 9, pp. 286-291, Tallin, Estonie, URSS, Août 1990.
- [14] C. Samson, B. Espiau, M. Le Borgne : *Robot Control: the Task Function Approach*, Oxford University Press, 1991.
- [15] G. Sandini, M. Tistarelli : *Active Tracking Strategy for Monocular Depth Inference over Multiple Frames*, IEEE Trans. on PAMI, Vol 12, n. 1, pp. 13-27, Janvier 1990.
- [16] B.G. Schunck : *Image Flow Segmentation and Estimation by Constraint Line Clustering*, IEEE Trans. on PAMI, Vol 11, n. 10, pp. 1010-1027, Octobre 1989.

- [17] M. Subbarao : *Interpretation of Image Flow: A Spatio-temporal Approach*, IEEE Trans. on PAMI, Vol 11, n. 3, pp. 266-278, Mars 1989.
- [18] A.M. Waxman, B.K. Parsi, M. Subbarao : *Closed-form Solutions to Image Flow Equations for 3D Structure and Motion*, Int. Journal of Computer Vision, Vol 1, n. 3, pp. 239-258, Octobre 1987.
- [19] L.E. Weiss, A.C. Sanderson : *Dynamic Sensor-Based Control of Robots with Visual Feedback*, IEEE Journal of Robotics and Automation, Vol 3, n. 5, pp. 404-417, Octobre 1987.
- [20] J. Weng, T.S. Huang, N. Ahuja : *Estimation and Structure from Line Matches: Performance Obtained and Beyond*, 10th Int. Conf on Pattern Recognition, Vol 1, pp. 168-172, Juin 1990.
- [21] M.Xie, P. Rives, C. Labit : *Reconstruction of 3D polyhedra scene by a moving camera*. Congrès PIXIM 89, Paris, France, Septembre 1989.

LISTE DES DERNIERES PUBLICATIONS INTERNES PARUES A L'IRISA

- PI 653 MADMACS : UN OUTIL DE PLACEMENT ET ROUTAGE POUR LE DESSIN DE MASQUES DE RESEAUX REGULIERS
Eric GAUTRIN, Laurent PERRAUDEAU, Oumarou SIE
Avril 1992, 16 pages.
- PI 656 CAUSALITY ORIENTED SHARED MEMORY FOR DISTRIBUTED SYSTEMS
Michel RAYNAL, Masaaki MIZUNO, Mitch NEILSEN
Avril 1992, 8 pages.
- PI 657 ALGORITHMES PARALLELES POUR L'ANALYSE D'IMAGE PAR CHAMPS MARKOVIENS
Etienne MEMIN, Fabrice HEITZ
Mai 1992, 74 pages.
- PI 658 MULTISCALE SIGNAL PROCESSING : ISOTROPIC RANDOM FIELDS ON HOMOGENEOUS TREES
Bernhard CLAUS, Ghislaine CHARTIER
Mai 1992, 28 pages.
- PI 659 ON THE COVARIANCE-SEQUENCE OF AR-PROCESSES. AN INTERPOLATION PROBLEM AND ITS EXTENSION TO MULTISCALE AR-PROCESSES
Bernhard CLAUS, Albert BENVENISTE
Mai 1992, 36 pages.
- PI 660 EXCEPTION HANDLING IN COMMUNICATING SEQUENTIAL PROCESSES DESIGN, VERIFICATION AND IMPLEMENTATION
Jean-Pierre BANATRE, Valérie ISSARNY
Mai 1992, 38 pages.
- PI 661 REACHABILITY ANALYSIS ON DISTRIBUTED EXECUTIONS
Claire DIEHL, Claude JARD, Jean-Xavier RAMPON
Juin 1992, 18 pages.
- PI 662 RECONSTRUCTION 3D DE PRIMITIVES GEOMETRIQUES PAR VISION ACTIVE
Samia BOUKIR, François CHAUMETTE
Juin 1992, 40 pages.
- PI 663 FILTRES SEMANTIQUES EN CALCUL PROPOSITIONNEL
Raymond ROLLAND
Juin 1992, 22 pages.

Imprimé en France
par
l'Institut National de Recherche en Informatique et en Automatique

ISSN 0249 - 6399