Identification des polygones 3D à partir des segments 3D

Ming Xie, Patrick Rives

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

N° 908

IDENTIFICATION DES POLYGONES 3D A PARTIR DES SEGMENTS 3D

Programme 6

Ming XIE
Patrick RIVES

Octobre 1988

RR-0908

IRISA

INSTITUT DE RECHERCHE EN INFORMATIQUE
ET SYSTÈMES ALÉATOIRES

Campus Universitaire de Beaulieu
35042 - RENNES CÉDEX
FRANCE
Téléphone : 99 36 20 00
Télex: UNIRISA 950 473 F
Télécopie: 99 38 38 32

Identification Des Polygones 3D A Partir Des Segments 3D

===

Identification Of 3D Polygons From A Set Of 3D Line Segments

Ming XIE and Patrick RIVES
Publication Interne n°427 - Septembre 1988 - 46 Pages
INRIA/IRISA de Rennes

Campus de Beaulieu
F-35042 Rennes Cedex FRANCE
Tel (33)-99-36-20-00

Résumé

L'interprétation des segments 3D par des polygones 3D est un sujet de recherches important. L'application de la triangulation de Delaunay a déjà été proposée. Cette méthode ne traite le problème que d'une façon très globale, elle perd toute information locale concernant une scène. La méthode présentée ici va traiter ce problème d'une façon à la fois locale mais aussi globale. Nous avons étudié deux techniques : (a) la méthode de la "Prediction/Verification Dynamique" et (b) la méthode de la "Transformée de HOUGH". La stratégie employée consiste à : (a) estimer d'abord toutes les orientations possibles des surfaces planes qui sont les supports physiques des polygones 3D, (b) identifier ensuite toutes les surfaces planes parallèles à une orientation donnée et (c) effectuer le chaînage des segments 3D sur une surface plane estimée afin d'en extraire des polygones 3D présents. Les deux premiérs traitements de notre approche permettent d'avoir

CENTRE NATIONAL DE LA RECHERCHE SCIENTIFIQUE
(L. A. 227)
UNIVERSITÉ DE RENNES 1 I. N. S. A. DE RENNES

INSTITUT NATIONAL DE RECHERCHE
EN INFORMATIQUE ET EN AUTOMATIQUE
(LABORATOIRE DE RENNES)

une information globale sur une scène donnée. De plus ces deux traitements sont généraux, ils peuvent s'appliquer dans d'autres algorithmes d'interprétation des segments 3D. Le traitement de chaînage des segments 3D tente d'avoir une information locale sur une scène. A cette étape de traitement, on peut extraire des informations concernant : (a) les sommets des polygones 3D, (b) les enveloppes convexes des surfaces planes et (c) la triangulation de Delaunay de segments 3D.

Nous consiérons que deux polygones 3D connexes appartiennent à un même objet polyédrique. Cette contrainte permet d'extraire rapidement tous les objets polyédriques dans une scène à partir des connaissances sur la connection des polygones 3D estimés.

Mots Clés : Interprétation des segments 3D, Prediction/Verification, Transformée de Hough, Polygone 3D, Chaînage des segments 3D.

Abstract

How to interpret the 3D line segments by the 3D polygons become a important subject of research. In [1], the author use Delaunay triangulation to interpolate the 3D line segments. This method have a shortcome, it can give only a global information about a scene and loss the local information. In this paper, we present a general method which interpret the 3D line segments by the 3D polygons. We have studied two techniques which can resolve succesfully the problem : (a) the "Dynamic Prediction/Verification" method and (b) the "HOUGH transformation" method. Our algorithm is based on three principal operations : (a) the estimation of all of orientations possibles of plans which are the physical supports of 3D polygones, (b) the identification of parallel plans in a given orientation and (c) the linking of 3D line segments in a estimed plan in order to find the 3D polygones localized in the plan. The first two operations consist of extrating the global information about a scene, they give us a general method that can be applied to the other algorithm of interpretation of 3D line segments. The third operation consist of extrating the local information that can be : (a) the polygones' vertex, (b) the convex envelop of plans and (c) the Delaunay triangulation of 3D line segments.

We consider that two 3D polygons connected are belonged to a same polyhedral object. This assumption allows us to extract rapidely

all of polyhedral objects in a scene from the information about the connection of the estimed 3D polygones.

Key Words : Interpretation of 3D line segments, Prediction/Verification, HOUGH transformation, 3D polygone, Linking of 3D line segments.

Table des matières

1 Introduction

La vision par ordinateur est une discipline qui tente de développer sur des machines informatiques des algorithmes ayant pour but de percevoir et d'interpréter un environnement 3D afin d'en extraire les caractéristiques nécessaires à la résolution d'un problème donné. L'étape de perception consiste à extraire l'information sensorielle. L'étape d'interprétation consiste à établir la liaison entre l'information sensorielle et une description symbolique du modèle de cet environnement. Le résultat de l'interprétation permet au système décisionnel d'éxécuter les actions nécessaires au bon accomplissement d'une tâche donnée.

Il y a deux types de stratégies utilisées afin d'atteindre les buts de la vision par ordinateur : ascendantes et descendantes. Les stratégies ascendantes tentent de construire à partir de l'information sensorielle (primitives 2D) une représentation de plus haut niveau (ex: primitives 3D) interprétable par le processus décisionnel. Les stratégies descendantes tentent de déduire à partir d'un ensemble d'objets connus par le système une description compatible avec les primitives extraites de l'image, le processus décisionnel étant alors basé sur une mesure de distance entre la représentation extraite de l'image et la description issue du modèle. En fait, la combinaison de ces deux stratégies dans un système de vision constitue une approche plus sophistiquée.

1.1 Contexte

Une des tendances récentes de recherches en vision par ordinateur consiste à prendre en compte une séquence d'images au lieu d'avoir une ou deux images dans le cas de vision statique. Cette approche est connue sous le nom de "vision dynamique" (si le mouvement est contrôlable, on l'appelle aussi "vision active") et possède deux avantages fondamentaux :

1. Certains problèmes rencontrés (par exemple: Shap from shading) en vision par ordinateur peuvent être résolus à partir d'équations linéaires. La solution trouvée est unique et stable.

2. La possibilité d'utiliser la cohérence temporelle au sein de la séquence d'images permet de raffiner les estimation des primitives par des méthodes de filtrage (par exemple : Filtre de KALMAN).

Si nous considérons le cas particulier des robots mobiles, l'utilisation de la vision dynamique procure d'autres avantages :

1. Premièrement, elle donne la possibilité au robot d'acquérir en ligne l'information sur son environnement de façon à pouvoir éxécuter correctement sa tâche dans un univers inconnu ou mal modélisé.

2. Deuxièmement, elle donne la possibilité de réaliser une commande en boucle fermée qui permet d'avoir :

   (a) La capacité de contrôler le mouvement de caméra en vue d'améliorer la qualité d'estimation de primitives 3D (vision active).

   (b) La capacité de compenser l'incertitude sur les primitives (estimées) par une commande en boucle fermée ayant de bonne propriétés de robustesse et de stabilité.

Nous avons focalisé nos recherches sur l'utilisation d'une caméra embarquée sur un robot et d'algorithmes de vision dynamique (ou active). Notre objectif de recherche se définit de la façon suivante :

> *En supposant que le robot se déplace avec une vitesse contrôlabe dans un environnement inconnu composé d'objets polyédriques, essayer de reconnaître cet environnement à partir de la séquence d'images observées par une caméra embarquée sur le robot.*

Nous travaillons dans le cadre de la vision monoculaire. Puisque l'application de notre recherche est restreint à un univers polyédrique, nous portons naturellement attention aux primitives de type : segments 2D et segments 3D qui sont les types de primitives plus distinctifs. Nous avons proposé un système de vision qui répond à ce besoin. Les aspects principaux de ce systèmes sont :

1. Estimation des segments 2D en mouvement (dynamiques).
   Il s'agit d'intégrer des techniques d'extraction de contours statiques, d'extraction de contours spatio-temporels et de chaînage des contours

afin d'estimer les segments 2D dynamiques le long d'une séquence d'images.

2. Commande en boucle fermée.
   En vue d'améliorer la qualité de l'estimation des segments 2D dynamiques, on génère un mouvement local selon un critère de minimisation des erreurs d'estimation. Ce mouvement local est superposé au mouvement global de façon à obtenir un mouvement exécutable.

3. Estimation récursive des segments 3D.
   Connaissant le mouvement de la caméra, il est possible de calculer directement les segments 3D à partir des segment 2D dynamiques selon une méthode générale que nous avons étudié. Cette méthode s'applique aussi à la vision polynoculaire.

4. Identification des polygones 3D.
   On essaye d'extraire les polygones 3D interprétables à partir d'un ensemble des segments 3D estimés. Ceci permet de reconnaître et de localiser les objets polyédriques.

La figure 1 montre schématiquement notre système de vision dans le cadre de la reconstruction d'une scène 3D polyédrique. Nous voulons que chaque niveau de traitement dans notre système de vision donne un resultat robuste. Ceci est le point clef pour assurer la meilleure performance d'un système de vision.

1.2 Description Du Problème

Nous traitons dans ce rapport le problème de l'interprétation des segments 3D par des polygones 3D qui est un sujet de recherche important. L'application de la triangulation de Delaunay a déjà été proposée. Cette méthode ne traite le problème que d'une façon très globale, elle perd toute information locale concernant une scène. La méthode présentée ici va traiter ce problème d'une façon à la fois locale mais aussi globale. Nous avons étudié deux techniques :

1. la méthode de la "Prediction/Verification Dynamique".

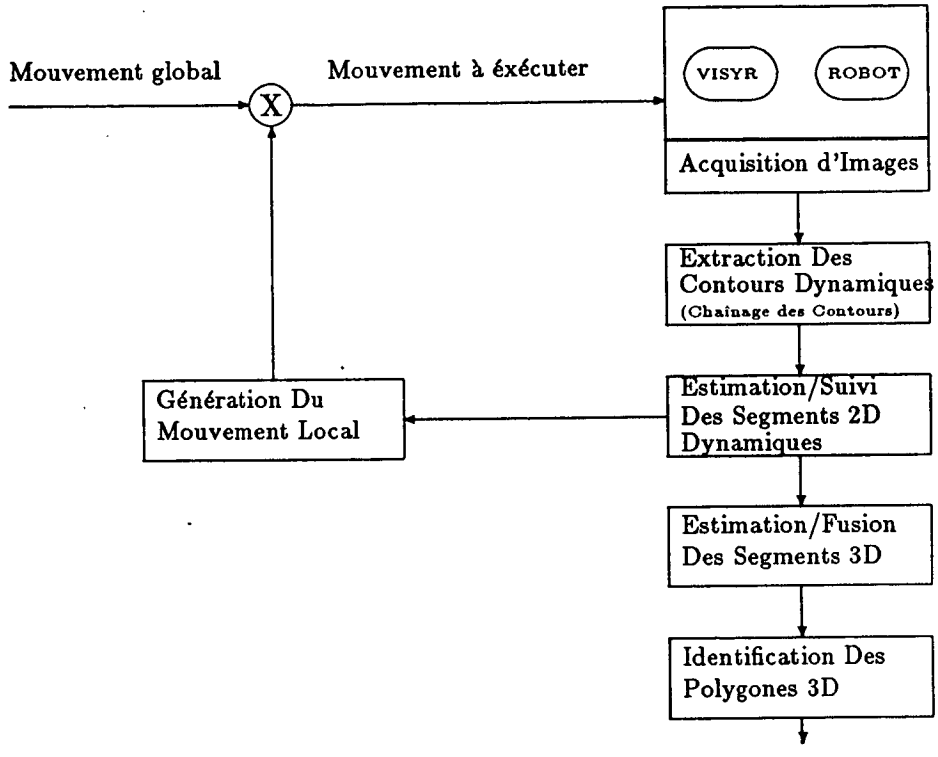


Figure 1 : Un Système pour la Reconstruction d'une Scène 3D Polyédrique

2. la méthode de la "Transformée de HOUGH".

La stratégie employée consiste à :

1. estimer d'abord toutes les orientations possibles des surfaces planes qui sont les supports physiques des polygones 3D.

2. identifier ensuite toutes les surfaces planes parallèles à une orientation donnée.

3. effectuer le chaînage des segments 3D sur une surface plane estimée afin d'en extraire les polygones 3D présents.

Les deux premiérs traitements de notre approche permettent d'avoir une information globale sur une scène donnée. De plus ces deux traitements sont généraux, ils peuvent s'appliquer dans d'autres algorithmes d'interprétation des segments 3D. Le traitement de chaînage des segments 3D tente d'avoir une information locale sur une scène. A cette étape de traitement, on peut extraire des informations concernant : (a) les sommets des polygones 3D, (b) les enveloppes convexes des surfaces planes et (c) la triangulation de Delaunay de segments 3D.

Nous considérons que deux polygones 3D connexes appartiennent à un même objet polyédrique. Cette contrainte permet d'extraire rapidement tous les objets polydriques dans une scène à partir des connaissances sur la connection des polygones 3D estimés.

1.2.1 Modélisation De Segment 3D Et De Surface Plane

Un segment 3D est modélisé généralement par ses deux coordonnées :

1. Les coordonnées cartésiennes :

$$\{\vec{p}_1 + \sigma^2\vec{p}_1, \vec{p}_2 + \sigma^2\vec{p}_2\}$$

où $(\vec{p}_1, \vec{p}_2)$ sont les deux points d'extrémité d'un segment 3D.

2. Les coordonnées *plückeriennes* :

$$\{\vec{U} + \sigma^2\vec{U}, \vec{H} + \sigma^2\vec{H}\}$$

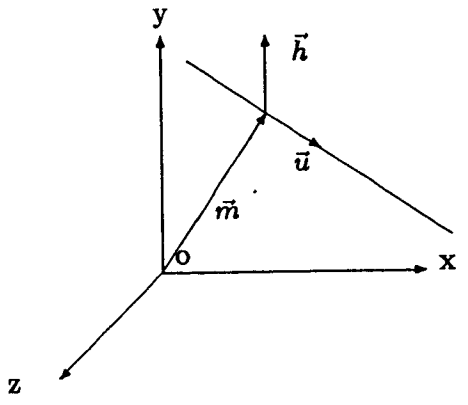


Figure 2 : Modélisation d'un segment 3D

En vue de faciliter l'identification des paramètres des surfaces planes, nous nous intéressons à une autre manière de modélisation pour un segment 3D. Dans la figure 2 nous montrons le modèle de segment 3D que nous employons. Un segment 3D est paramètré par :

$$(\vec{u} + \sigma^2\vec{u},\ \vec{m} + \sigma^2\vec{m})$$

où $\vec{u}$ est le vecteur unitaire représentant la direction du segment 3D, $\sigma^2\vec{u}$ sa matrice de covariance, $\vec{m}$ le vecteur représentant le centre de gravité du segment 3D, $\sigma^2\vec{m}$ sa matrice de covariance.

Une surface plane est paramètrée en principe par :

- Son orientation $\vec{n}$.
- La distance à l'origine ρ.

Sa représentation analytique dans l'espace $\Re^3$ s'ecrit comme :

$$x \bullet \sin\sigma \bullet \sin\theta + y \bullet \cos\theta + z \bullet \cos\sigma \bullet \sin\theta - \rho = 0.$$

où (σ,θ) est la coordonnée sphérique du vecteur $\vec{n}$. Il est evident qu'une surface plane est déterminée uniquement par les trois paramètres:(σ,θ,ρ).

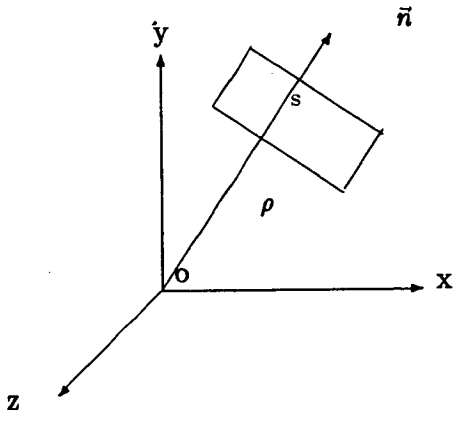


Figure 3 : Modélisation d'une surface plane

Compte tenu du bruit au niveau de la mesure des primitives, nous considérons également dans le modèle la variance relative à chaque paramètre, la surface plane est alors modélisée par :

$$\{\vec{n} + \sigma^2\vec{n}; \rho + \sigma^2\rho\} \quad (1)$$

1.2.2 Position Du Problème

Nous essayons d'utiliser l'information portée par un ensemble de segments 3D pour identifier tous les polygones 3D (surfaces planes) dans une scène 3D polyédrique. Ceci est possible du fait que la plupart des segments 3D mesurés sont issus des arêtes des polygones 3D. Cette remarque nous a conduit à développer une approche qui comporte principalement deux parties :

1. Estimation des surfaces planes : Une surface plane est le support géomètrique d'un polygone 3D. Nous allons étudier deux méthodes qui ont pour but d'identifier les paramètres de toutes les surfaces planes dans une scène à partir de l'ensemble des segments 3D mesurés.

2. Détermination des polygones 3D : Un polygone 3D est une surface plane bornée par des segments 3D qui forme une frontière fermée. Dans cette partie, nous avons exploité deux techniques pour déterminer la frontière de chaque polygone.

Nous rappelons ici que la Sphère Gaussienne est un outil très utile pour représenter l'orientation. Suivant la représentation choisie, une orientation sera figurée sur la sphère de GAUSS soit par un point (normale) soit par un grand cercle (plan perpendiculaire à la normale).

A partir de cette représentation, nous choissisons de représenter les paramètres $(\vec{n}, \rho)$ d'une surface plane dans la Sphére Gaussienne Etendue qui est definie comme l'ensemble des sphères dont :

1. La normale d'un grand cercle représente la direction d'un groupe de surfaces ayant la même orientation.

2. Le rayon d'un grand cercle représente le paramètre ρ de la surface plane.

Pour une scène donnée qui est composée par des surfaces planes, on trouvera une seule configuration représentative relative à cette scène dans la Sphère Gaussienne Etendue, et notre objectif peut être résumé comme la recherche de cette configuration.

Le problème de l'identification des surfaces planes dans une scène polyédrique peut être donc décrit de la façon suivante :

> *Etant donnée un ensemble des segments 3D mesurés qui sont issus des contours des objets polyédriques, estimer les paramètres* $(\vec{n}, \rho)$ *des surfaces planes dans la scène correspondante et leur variances de telle sorte que la configuration construite par l'ensemble des* $(\vec{n}, \rho)$ *estimés est conforme à celle de la scène initiale.*

D'un point de vue mathématique, la relation entre les paramètres $(\vec{n}, \rho)$ d'une surface plane et les paramètres $(\vec{u}, \vec{m})$ de ses segments 3D contenus est très simple, à savoir :

$$\begin{cases} \vec{n} = f(\vec{u}) \\ \rho = g(\vec{n}, \vec{m}) \end{cases} \tag{2}$$

Supposons que $\{(\vec{u}_i, \vec{m}_i)\}_{1 \times l}$ soit l'ensemble de segments 3D contenus par la même surface plane, la détermination des paramètres $(\vec{n}, \rho)$ peut être réalisée en minimisant les deux critères suivants :

1. Critère 1 : La recherche de $\vec{n}$ de telle sorte qu'on minimise le critère :

$$J_n = min\ (\ J(n) = \sum_i^l \|\vec{n} \bullet \vec{u}_i\|^2\ ) \tag{3}$$

avec $\|\vec{n}\| = 1$.

2. Critère 2 : La recherche de ρ de telle sorte qu'on minimise le critère :

$$J_\rho = min\ (\ J(\rho) = \sum_i^l \|\rho - \vec{m}_i \bullet \vec{n}\|^2\ ) \tag{4}$$

2 Méthodologie

Selon les deux critères qu'on vient d'établir, le calcul de ρ dépend de celui de $\vec{n}$, par contre le calcul de $\vec{n}$ ne dépend pas de celui de ρ, cette remarque est très intéressante parce qu'elle permet de décomposer l'estimation des paramètres $(\vec{n}, \rho)$ en deux phases séparables. L'autre point important est qu'en vue de confirmer l'existence d'une surface plane estimée, il nous faudra tester la fermeture des segments 3D sur cette surface plane. L'identification des paramètres des surfaces planes doit donc se dérouler en trois phases qui sont :

- phase 1: L'estimation du vecteur $\vec{n}$ et sa matrice de covariance $\sigma^2\vec{n}$.
- phase 2: L'estimation du paramètre ρ et sa variance $\sigma^2\rho$.
- phase 3: Le chaînage des segments 3D sur la surface plane estimée afin d'obtenir un polygone 3D.

Du fait que l'estimation des paramètres $(\vec{n}, \rho)$ nécessite de savoir à quelle surface plane appartiennent les segments 3D, notre idée de base pour réaliser le traitement correspondant aux deux premières phases consiste à :

1. classifier d'abord les segments 3D mesurés :
   On suppose que la scène est composée par des surfaces planes. L'ensemble des segments 3D mesurés sont issus des contours de ces surfaces

planes. Il est possible d'identifier quel sous-ensemble de segments appartient à telle surface plane. Nous utilisons la méthode de "Prediction/Vérification Dynamique" et de "Transformée de Hough Généralisée" pour classifier l'ensemble des segments 3D en des sous-ensembles dont chacun est contenu dans un groupe de surfaces planes parallèles.

2. estimer ensuite les paramètres des surfaces planes :
   Il s'agit d'abord d'estimer le vecteur de normale pour chaque sous-ensemble de segments, l'orientation du groupe des surfaces planes parallèles relative à ce sous-ensemble étant représentée par ce vecteur. La méthode d'estimation: le " Moindre Carré (classique ou recursive)", le "Filtre de KALMAN" et le " Gradient Stochastique" nous serviront à estimer les vecteurs de normale. Ensuite, on va estimer les paramètres ρ correspondant à chaque groupe de surfaces planes par une méthode de classification des données monodimensionnelles.

La figure ci-dessous résume les deux opérations pour l'identification des paramètres des surfaces planes.

3 Outils Mathématique d'Estimation

Avant d'exposer les deux techniques de classification des segments 3D, nous présentons très brièvement les deux méthodes d'estimation que nous avons utilisées.

3.1 Moindre Carré

On pose :

$\vec{u}_i = (u_{ix}, u_{iy}, u_{iz})$ ---- *la direction de segment* i $(1 \ldots k)$.
$\vec{n} = (n_x, n_y, n_z)$ ---- *la normale de plan à estimer.*

L'objectif consiste à trouver $\vec{n}$ qui minimise le critère 3. Pour cela, il suffit de résoudre les équations suivantes :

$$\begin{aligned}
\frac{\partial J}{\partial n_x} &= 0 = a \bullet n_x + m \bullet n_y + n \bullet n_z. \\
\frac{\partial J}{\partial n_y} &= 0 = m \bullet n_x + b \bullet n_y + l \bullet n_z. \\
\frac{\partial J}{\partial n_z} &= 0 = n \bullet n_x + l \bullet n_y + c \bullet n_z. \\
1 &= n_x^2 + n_y^2 + n_z^2.
\end{aligned} \tag{5}$$

où :

$$\begin{aligned}
a &= \textstyle\sum_i^k (u_{ix})^2. \quad m = \sum_i^k (u_{ix} \bullet u_{iy}). \\
b &= \textstyle\sum_i^k (u_{iy})^2. \quad n = \sum_i^k (u_{ix} \bullet u_{iz}). \\
c &= \textstyle\sum_i^k (u_{iz})^2. \quad l = \sum_i^k (u_{iy} \bullet u_{iz}).
\end{aligned} \tag{6}$$

Selon les deux premières équations , on a :

$$\begin{aligned}
\begin{pmatrix} n_x \\ n_y \end{pmatrix} &= \begin{pmatrix} a & m \\ m & b \end{pmatrix}^{-1} \bullet \begin{pmatrix} -n \\ -l \end{pmatrix} \bullet n_z \\
&= \begin{pmatrix} \frac{l \bullet m - n \bullet b}{a \bullet b - m^2} \\ \\ \frac{n \bullet m - l \bullet a}{a \bullet b - m^2} \end{pmatrix} \bullet n_z.
\end{aligned} \tag{7}$$

En employant la quatrième équation, on obtiendra la solution suivante :

$$\begin{cases}
n_x = \frac{l \bullet m - n \bullet b}{a \bullet b - m^2} \, n_z. \\
n_y = \frac{n \bullet m - l \bullet a}{a \bullet b - m^2} \, n_z. \\
n_z = \pm \sqrt{\frac{(a \bullet b - m^2)^2}{(a \bullet b - m^2)^2 + (l \bullet m - n \bullet b)^2 + (n \bullet m - l \bullet a)^2}}.
\end{cases} \tag{8}$$

Cas particuliers :

1. Si $a = 0$, cela signifie que tous les segments se situent dans le plan OYZ ($u_{ix} = 0$), la normale du plan est obtenu directement :

$$\vec{n} = \{1, 0, 0\}.$$

2. Si $b = 0$, cela signifie que tous les segments se situent dans le plan OXZ ($u_{iy} = 0$), la normale du plan est obtenu directement :

$$\vec{n} = \{0,\ 1,\ 0\}.$$

3. Si $c = 0$, cela signifie que tous les segments se situent dans le plan OXY ($u_{iz} = 0$), la normale du plan est obtenu directement :

$$\vec{n} = \{0,\ 0,\ 1\}.$$

4. Si $a \bullet b - m^2 = 0$ ou bien ($\frac{a}{m} = \frac{m}{b}$), cela signifie que $\frac{u_{1y}}{u_{1x}} = \frac{u_{2y}}{u_{2x}} = \ldots = \frac{u_{ky}}{u_{kx}}$, c'est-à-dire que tous les segments se situent dans un plan qui est perpendiculaire au plan OXY, donc $n_z = 0$, la normale du plan est alors calculé par :

$$\begin{cases} n_x = \frac{-m}{a} \bullet n_y. \\ n_y = \pm\sqrt{\frac{a^2}{a^2 + m^2}}. \\ n_z = 0. \end{cases}$$

3.2 Gradient Stochastique

Etant donnée la fonction d'erreur $\varepsilon_k = f(\ X, Z_k)$ où "X" est le vecteur de paramètre à estimer et "Z_k" le vecteur de mesures, la meilleure valeur estimée de "X" qui minimise le critère :

$$J = \frac{1}{2} \bullet E(\ \varepsilon^2\ ) \tag{9}$$

est obtenu par un processus d'estimation récursive , à savoir :

$$X(k+1) = X(k) - \varepsilon_k \bullet \bigtriangledown_x \varepsilon_k \bullet G. \tag{10}$$

où "G" est une matrice de gain qu'on choisit à priori et "$\bigtriangledown_x \varepsilon_k$" la dérivée de ε_k par rapport à X.

Dans le cas de l'estimation de la normale $\vec{n}$ suivant le critère 3, nous établissons la fonction d'erreur telle que :

$$\begin{cases} X(k) &= \vec{n}. \\ \varepsilon_k &= \vec{u}_k \bullet X(k). \end{cases} \tag{11}$$

Nous avons donc :

$$\nabla_x \varepsilon_k = \vec{u}_k.$$

D'apres nos résultats expérimentaux, la matrice de gain G peut être choisie ainsi :

$$G = \begin{pmatrix} 0.05 & 0.01 & 0.01 \\ 0.01 & 0.05 & 0.01 \\ 0.01 & 0.01 & 0.05 \end{pmatrix}.$$

Les deux paragraphes suivants présenteront deux techniques qui ont pour objectif de classifier l'ensemble des segments 3D en des sous-ensembles. Puis nous utiliserons une des méthodes d'estimation présentée précédemment pour estimer toutes les orientations possibles dans la scène selon le critère 3.

4 Méthode I : Prediction / Vérification Dynamique

L'idée de base de cette méthode est de faire une prédiction d'abord et de la valider ensuite afin de classifier , dans le cas de l'estimation des orientations, les segments 3D en des sous-ensembles dont chacun est contenu dans un groupe de surfaces planes parallèles. L'étape suivante sera l'estimation des paramètres de l'orientation relative sur chacun de ces sous-ensembles.

La méthode classique de Prediction/Vérification est réalisée en deux étapes : (a) la création des hypothèses et (b) la vérification des hypothèses. Si la vérification est positive selon un critère choisi, les hypothèses générées sont considèrées vraies avec des degrés de certitude. En vue d'améliorer la performance de cette méthode, nous ajoutons une étape de "modification des hypothèses". Si le degré de certitude d'une hypothèse est inférieur à un seuil donné, l'étape de " modification des hypothèses" va modifier

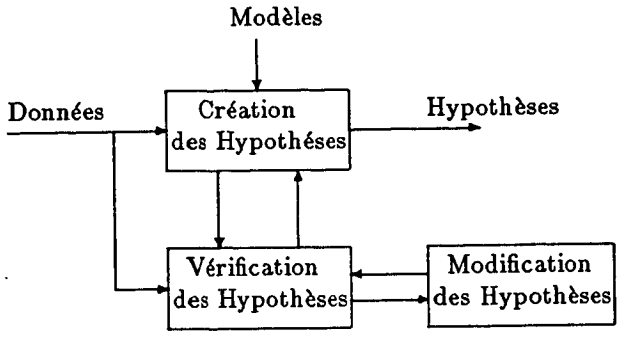


Figure 4 : Prediction / Vérification Dynamique

cette hypothèse pour que l'hypothèse modifiée ait un degré de certitude plus élevé. La méthode améliorée est appellée la "Prediction/ Vérification Dynamique" qui est schématisée par la figure 4.

Dans l'algorithme issu de cette méthode, la prédiction d'une orientation est faite en sélectionnant deux segments 3D non-parallèles , puis, on regroupe en un sous-ensemble les segments 3D qui sont perpendiculaires à cette orientation à un seuil près. A l'étape de vérification, on compare seulement le nombre des éléments dans ce sous-ensemble avec une valeur donnée.

La convergence de cette opération répétitive est assurée par le processus suivant :

On définit trois classes de segments :

- $\{S_0\} \longrightarrow$ l'ensemble des segments 3D n'ayant pas été traités.
- $\{S_1\} \longrightarrow$ l'ensemble des segments 3D appartenant déjà à un sous-ensemble validé.
- $\{S_2\} \longrightarrow$ l'ensemble des segments 3D appartenant déjà à deux ou plusieurs sous-ensembles validés.

L'algorithme utilise les régles suivantes :

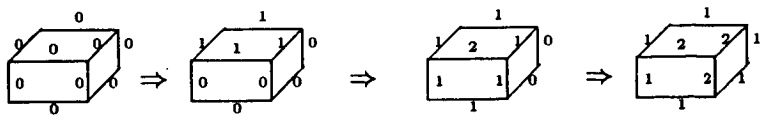


Figure 5 : Exemple de l'identification d'un bloc

1. **La sélection de deux segments pour la prédiction doit être faite parmi $\{S_0\}$.**

2. **La vérification et la création d'un sous-ensemble doit être faite parmi $\{S_0\}$ et $\{S_1\}$.**

3. **L'arrêt du programme s'effectue si toutes les prédictions de l'orientation faites parmi $\{S_0\}$ ne sont pas confirmées au niveau de la vérification, ou bien si $\{S_0\}$ est presque vide.**

La figure 5 montre le déroulement de cet algorithme dans le cas de l'identification des surfaces d'un bloc,le numéro de chaque droite indique à quelle classe appartient ce segment.

5 Méthode II: Transformée De Hough Généralisée

Dans [1], la définition de la T.H.G est la suivante :

> Pour une famille de fonctions (ex: lignes, courbes et plans) de la forme $f(x,p) = 0$, on quantifie l'espace du paramètre p et on associe un accumulateur $A(p)$ à chaque élément de l'espace (" cellule ") ainsi obtenu.
>
> Après avoir effectué un ensemble de mesures $\{X_i\}$, on va déterminer dans l'espace quantifié, pour chaque mesure x_i, tous les paramètres possibles p qui vérifient la relation :
>
> $$f(x_i, p) = 0$$

En même temps, on incrémente les accumulateurs correspondants :

$$A(p) = A(p) + 1$$

Les résultats relatifs aux mesures $\{X_i\}$ sont obtenus par la recherche des maximas locaux dans le tableau $A(p)$. C'est-à-dire :

$$Résultats = \{ \ \{p^*\} \mid A(p^*) = max\{A(p)\} \ \}$$

Au niveau de l'implémentation de la T.H.G, on doit tenir compte les deux problèmes se rapportant à :

1. La diminution de la dimension de l'espace à quantifier qui dépend de la dimension du paramètre à estimer.

2. La précision du paramètre estimé qui dépend de la résolution de quantification.

Il y a plusieurs façons de traiter ces deux problèmes selon les applications entreprises. Dans le cas de l'identification des paramètres d'une surface plane, nous avons pris en compte la considération suivante :

1. Décomposer l'espace des paramètres en des sous-espaces indépendants et estimer les paramètres séparément. Par exemple, dans notre cas, on décompose l'espace des paramètre $(\vec{n}, \rho)$ en deux. La méthode T.H.G va nous servir à identifier la normale $\vec{n}$.

2. Exploiter uniquement l'information de regroupement obtenue par la T.H.G et utiliser une méthode de " moindre carré" ou de " gradient stochastique " pour estimer le paramètre relatif à chaque groupe de mesures.

Ici nous utilisons les coordonnées sphériques pour représenter le vecteur $\vec{n}$ sur la sphère de GAUSS, c'est-à-dire :

$$\vec{n} = \begin{pmatrix} \sin(\sigma) \bullet \sin(\theta) \\ \cos(\theta) \\ \cos(\sigma) \bullet \sin(\theta) \end{pmatrix} \tag{12}$$

Du fait de la contrainte de visibilité des surfaces planes, le vecteur $\vec{n}$ se situe dans une demi-sphère. La plage de variation de σ et θ est donc limitée entre 0 et 180 degré.

Compte tenu de la non-homogénéité de l'espace (σ,θ), nous le quantifions en $(18 \times 18 - 17)$ cellules de taille de 10×10 degrés auquelles on associe :

- Un accumulateur qui permet de localiser les maximas locaux.
- Des pointeurs vers les segments 3D qui permettent d'extraire un sous-ensemble de segments 3D contribuant à une orientation déterminée.

La figure 6 montre le résultat de la quantification représentant un hémisphère de l'espace 0xyz. Chaque cellule a une dimension de $10^0 \times 10^0$. Etant donné un segment mesuré, on doit alors incrémenter 18 accumulateurs correspondants.

L'algorithme issu de cette méthode est décrit ainsi :

Algorithme :

1. **Pour chaque segment 3D, incrémenter les 18 accumulateurs correspondants.**
2. **Rechercher un maxima global dans la matrice des accumulateurs.**
3. **Estimer le paramètre $\vec{n}$ relatif à ce maxima par la méthode de "Moindre Carré" ou de " Gradient Stochastique " et éliminer ce maxima.**
4. **Eliminer les segments qui confirment l'existence d'au moins deux orientations et remettre à jour la matrice des accumulateurs.**
5. **Répéter 2), 3), et 4) jusqu'à ce que la valeur du maximum trouvé soit inférieur à un seuil donné.**

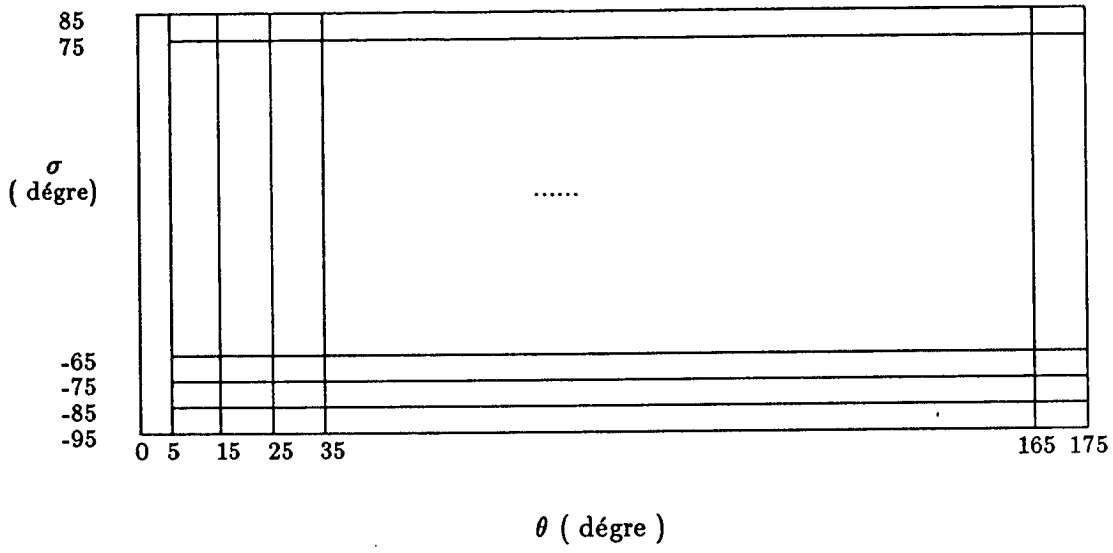
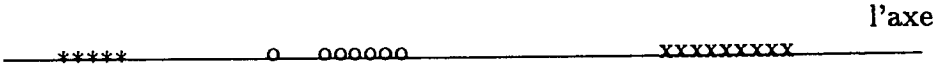


Figure 6 : L'espace (σ, θ) quantifié

6 Estimation Du Paramètre ρ Des Surfaces Planes

Connaissant l'orientation estimée $\vec{n}$ du groupe des surfaces planes parallèles, il nous faut estimer tous les ρ possibles, c'est-à-dire séparer les surfaces planes parallèles selon le critère 4. Remarquons que $\vec{m}_i \bullet \vec{n}$ est une valeur scalaire qui correspond à un point sur un axe A monodimensionel, l'ensemble des points $\{\vec{m}_i \bullet \vec{n}\}$ forme donc un certain nombre de 'nuages' sur cet axe et le point central de chaque 'nuage' donne une valeur ρ_j estimée. Le problème d'estimation du paramètre ρ consiste donc à localiser tous les points centraux des nuages.

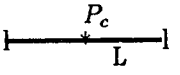


Dans le cadre de "l'Analyse des Données", un certain nombre de méthodes ont été developpées permettant de résoudre les problèmes de classification. Notre problème étant monodimensionel, nous n'avons pas vu l'utilité de les employer du fait de leur lourdeur. Nous utilisons la méthode suivante.

Soit $D_0 = \{d_i\}$ l'ensemble des points 1D et $C = \{C_j \mid C_j = \{d_{jk}\}\}$ l'ensemble de classes relatives à D_0, nous associons à chaque classe C_j un curseur paramètré par :

$$\{Point\ de\ milieu\ P_c, Demi - longueur\ L\}.$$

(voir la figure ci_dessous).

curseur : 

Supposons que l'histogramme de chaque classe C_j est une distribution gaussienne, la dispersion de C_j est donc caractérisée par sa variance, et la quasi totalité ($\simeq 97\%$) des points de C_j se situe dans l'intervale $\{M \pm 3.0 \times V\}$ où M est la valeur moyenne de C_j et V la variance. Le paramètre P_c et L du curseur associé à C_j sont alors determinés par :

$$\begin{cases} P_c &= moyenne\ de\ C_j. \\ L &= 3.0 \times variance\ de\ C_j. \end{cases} \tag{13}$$

Du fait qu'il peut exister plusieurs classes ('nuages') parmi D_0, il est nécessaire d'effectuer une segmentation de D_0 afin de localiser toutes ses classes. Pour cela, notre idée est la suivante :

L'ensemble D_0 des points 1D constitue une image des contours monodimensionels sur l'axe A et que l'ensemble des classes C constitue un ensemble des chaînes monodimensionelles. Donc la segmentation de D_0 revient à faire un chaînage des points 1D sur l'axe A. Ensuite on calcule les paramètres du curseur correspondant à chaque chaîne. Le paramètre P_c de chaque curseur donne un estimé de ρ.

Il est facile de réaliser le chaînage des points 1D. On aligne d'abord les points 1D en croissant. Ensuite on définit une distance maximale de connexité entre deux chaînes. Si deux points sont écartés d'une distance supérieure à la distance maximale, ils sont considérés comme appartenant à deux chaînes différentes. De cette façon, on peut agglomerer les points d'une même chaîne.

Selon cette idée, nous avons réalisé l'algorithme suivant :

Algorithme :

1. **Construire le tableau $D_0 = \{\vec{m}_i \bullet \vec{n}\}$ à partir de $\{\vec{m}_i\}$ et $\vec{n}$.**
2. **Aligner le tableau D_0 en croissant.**
3. **Définir une distance maximale et effectuer le chaînage des points 1D.**
4. **Calculer les paramètres du curseur correspondant à chaque chaîne.**

7 Chaînage Des Segments 3D

Une fois qu'on a estimé une surface plane $(\vec{n}_i, \rho_j)$, il est nécessaire d'effectuer un test pour confirmer la présence de celle-ci dans la scène. Pour cette raison, nous ajoutons une procédure de chaînage des segments 3D dans

notre algorithme. L'objectif de cette procédure étant de tester la fermeture des segments 3D d'une surface plane estimée, on choisit d'éliminer toutes les surfaces dont les segments 3D ne forment pas une frontière totalement fermée. En réalité, les surfaces perçues ayant une frontière non-fermée pourront nous servir à générer un mouvement local de la caméra.

Connaissant la direction de la normale estimée $\vec{n}_j$, il est possible de réaliser le chaînage des segments 3D dans le plan perpendiculaire à cette normale et donc de se ramener, par une rotation appropriée, à un espace 2D sur lequel va travailler l'algorithme de chaînage. Par la suite, un segment 3D sera appellé un segment 2D. Nous signalons ici qu'il ne faut pas confondre le traitement de chaînage des contours sur une image avec le traitement de chaînage des segments 3D dans une scène. Dans le cas de la vision statique, ces deux traitements sont équivalents. Mais dans le cadre de l'estimation des primitives en suivant une séquence d'images d'une scène statique, ces deux traitements ne sont plus équivalents parce qu'une image des segments 2D ne correspond qu'à un sous-ensemble de segments 3D estimés dans une scène.

7.1 Transformation Des Segments 3D

Sur le plan transformé d'une surface plane, on considére un segment 3D comme étant un segment 2D. Ici on s'intéresse simplement aux deux points d'extrémité d'un segment 3D.

Etant donné un segment 2D sur le plan transformé, ses deux extrémités seront notées :

$$\{\vec{p}_d, \vec{p}_f\}.$$

En vue de faciliter le chaînage des segments 2D, on décompose un segment 2D ainsi paramétré en deux segments 2D connexes qui seront modélisés par :

$$\{\vec{p}, \theta\}.$$

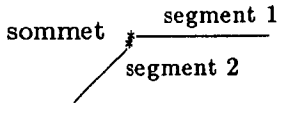
où $\vec{p}$ désigne un point d'extrémité du segment 2D et θ sa direction. On appelle aussi deux segments connexes un couple segment.

Selon la qualité de l'algorithme de chaînage des contours, on peut considérer avec une certitude plus ou moins grande que le point $\vec{p}$ d'un segment 2D correspond à un sommet d'un polygone. Dans les deux algorithmes présentés ci-dessous, l'algorithme I nécessite cette garantie, tandis que l'algorithme II s'adapte au cas général.

7.2 ALGORITHME I

Si l'algorithme de chaînage des contours peut nous garantir que les points d'extrémité des segments 3D correspondent aux sommets des polygones 3D, dans ce cas, l'objectif de chaînage des segments 2D consiste à détecter, parmi les segments 2D, les points sommets présentés sur le plan transformé et à savoir si ces points sommets forment une frontière fermée (un polygone) ou non.

Un point sommet est défini comme un point de fusion de deux segments 2D (non connexes) dont la distance entre leurs points d'extrémité est suffisament petite et dont leur orientation est suffisament écartée (voir la figure suivante) :



Nous présentons ici un algorithme de chaînage qui permet de résoudre le problème dans ce cas particulier :

Algorithme I :

1. **Chercher un point de départ qui est un point sommet correspondant au point d'extrémité d'un segment 2D.**

2. **Etablir une ligne droite qui passe par le segment 2D contenant ce point sommet de départ et Vérifier l'existence de cette ligne, c'est-à-dire qu'elle doit intersecter d'autres segments 2D si la vérification est vraie.**

3. **Chercher un autre point sommet sur cette ligne qui est le plus proche du point sommet de départ.**

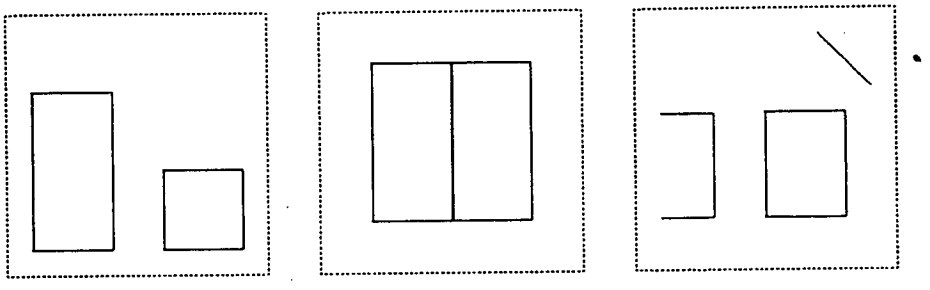
   (a) Si on le trouve, continuer l'étape 2 et 3 en prenant le nouveau point sommet comme le point sommet de départ.

   (b) Sinon, cela signifie qu'on a suivi une frontière non-fermée. Dans ce cas, recommencer par l'étape 1.

4. **Détecter la fermeture de frontière en calculant la distance entre le point sommet initial et le nouveau point sommet. Si cette distance est inférieure à un seuil donné, la frontière est fermée, sinon, elle est ouverte.**

5. **Répèter l'étape 1, 2, 3 et 4 jusqu'à ce que tous les segments 2D soient traités.**

Cette algorithme nous permet de traiter les cas de configuration des segments 2D suivants :



(a) (b) (c)

7.3 ALGORITHME II

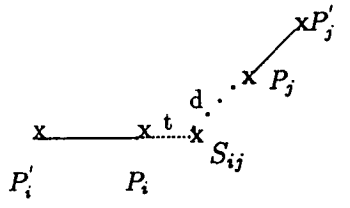
Dans le cas général, au lieu de considérer le point d'extrémité d'un segment 2D comme étant un point sommet, nous définisons un point sommet comme l'intersection de deux segments voisins. La relation de voisinage de deux segments 2D est déterminée de la façon suivante :

Si S_{ij} est le point d'intersection de deux segments P_i et P_j non connexes, on définit la distance entre ces deux segments par :

$$D_{ij} = d + t$$

où (voir la figure ci-dessous) :

d ----- *la distance entre le segment* P_j *et le point* S_{ij}.
t ----- *la distance entre le segment* P_i *et le point* S_{ij}.



Soit :

$y = A_i \bullet x + B_i$ ---- *l'equation du segment* P_i.
$y = A_j \bullet x + B_j$ ---- *l'equation du segment* P_j.

le point sommet S_{ij} relatif à ces deux segments est calculé par la formule suivante :

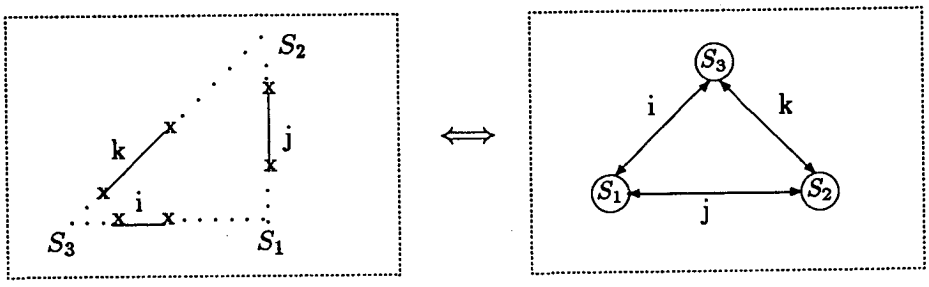
$$\begin{cases} S_x = \frac{-1}{A_i - A_j} \bullet (B_i - B_j). \\ S_y = \frac{A_i B_j - A_j B_i}{A_i - A_j}. \end{cases}$$

On peut donc obtenir la valeur de t et d par les calculs ci-dessous :

$$\begin{cases} t = min\ \{\sqrt{(P_{ix} - S_x)^2 + (P_{iy} - S_y)^2};\ \sqrt{(P'_{ix} - S_x)^2 + (P'_{iy} - S_y)^2}\}. \\ d = min\ \{\sqrt{(P_{jx} - S_x)^2 + (P_{jy} - S_y)^2};\ \sqrt{(P'_{jx} - S_x)^2 + (P'_{jy} - S_y)^2}\}. \end{cases}$$

La relation de voisinage de deux segments non connexes P_i et P_j est donc determinée selon le critère :

$$\begin{cases} D_{ij} \leq A_{seuil}.\ (i,\ j) \subset 1,\ ...,\ n. \\ \overline{P_i S_{ij}} \leq \overline{P'_i S_{ij}}. \\ \overline{P_j S_{ij}} \leq \overline{P'_j S_{ij}}. \end{cases} \qquad (14)$$



Plan Estimé

Graphe Crée

Figure 7 : Creation de Graphe Sommet

Compte tenu du fait qu'il peut exister plusieurs polygones sur un plan transformé, un segment i peut alors avoir plusieurs segments voisins. Pour cette raison, on va établir la liste des segments voisins pour chaque segment i.

A partir des listes de segments voisins, on peut calculer tous les points sommets. On a ainsi obtenu un graphe dont un nœud représente un sommet et une arête représente un segment (aussi son segment connexe) (voir la figure 7). En fait, ce graphe est une combinaison des deux types d'élément de graphe :

1. Type I : un nœud liant deux arêtes.
   Cet élément de graphe exprime le fait qu'un point sommet est l'intersection de deux segments voisins (non connexes).

2. Type II : une arête liant deux nœuds.
   Cet élément de graphe exprime le fait qu'un couple segment (deux segments connexes) peut avoir deux points sommets aux deux côtés opposés. Eventuellement une arête peut être ouverte si elle n'est liée qu'à un nœud.

Pour chaque nœud dans ce graphe, on donne un poids qui correspond au nombre moins un des couples segments différents connectés par ce nœud. Si par exemple un nœud connecte avec trois couples segments différents, il

a donc un poids de "2".

Dans le graphe crée, une frontière fermée (un polygone) se traduit donc par un circuit admissible du graphe dont les numéros des arêtes sont tous différents l'un de l'autre, et une frontière ouverte par un chemin admissible du graphe qui commence par une arête ouverte et termine aussi par une arête ouverte, de plus, les numéros des arêtes sur ce chemin sont tous différents l'un de l'autre. Une fois qu'on a trouvé une frontière (fermée ou non), on va décrémenter les poids associés aux nœuds qui sont chaînés par cette frontière.

En résume, cet algorithme comporte trois opérations principales :

1. Construire la liste des segments voisins pour chaque segment.
2. Calculer tous les points sommets et construire le graphe concerné.
3. Chercher les circuits admissibles et les chemins admissibles.

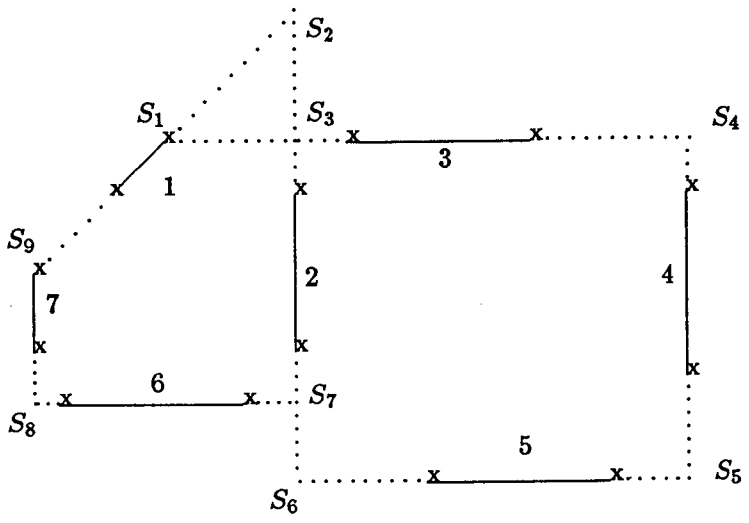
Nous donnons un exemple ci_dessous afin d'illustrer le principe de cet algorithme :

La figure A montre les segments sur le plan estimé. S_1, ... , S_8 sont les huit sommets calculés à partir de la rélation de voisinage entre les segments. Le graphe construit est montré par la figure B. Dans ce graphe, il n'y a que deux circuits admissibles qui correspondent à deux polygones sur la surface plane estimée :

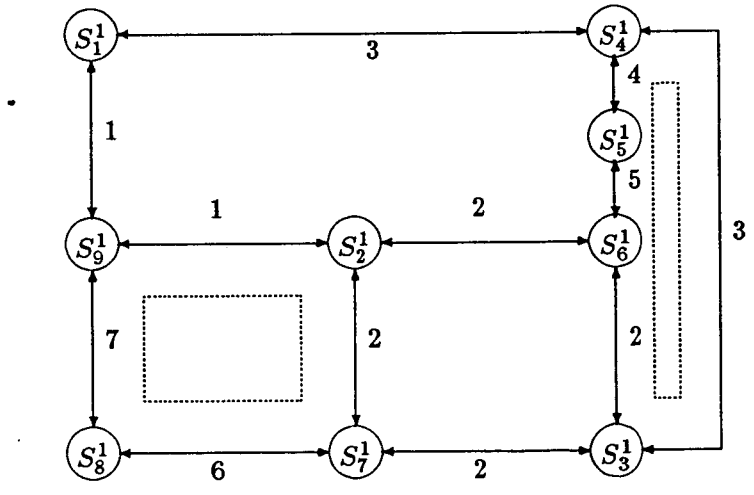
1. polygone 1 : $S_3 - - S_4 - - S_5 - - S_6$.
2. polygone 2 : $S_2 - - S_7 - - S_8 - - S_9$.

Selon le principe de cet algorithme, nous avons étudié une méthode qui permet de réaliser efficacement les trois opérations. On la présentera par la suite :

Après avoir établi la liste des segments voisins, on peut définir un point de référence P_r pour deux segments voisins et leurs segments connexes,



(A) Plan Estimé



(B) Graphe Crée

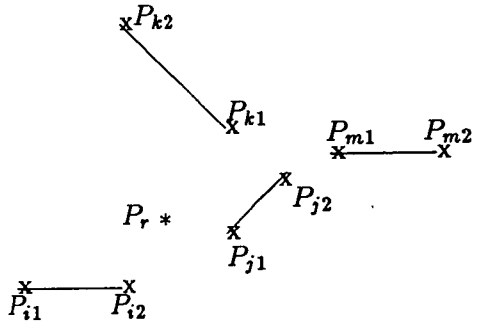
Figure 8 : Exemple de Chainage des Segments par Graphe

c'est-à-dire :

$$P_r = \frac{1}{4}(P_{i1} + P_{i2} + P_{j1} + P_{j2}).$$

où (voir la figure ci_dessous) :

P_{i1} ---- *le segment connexe du segment* P_{i2}.
P_{i2} ---- *le segment voisin du segment* P_{j1}.
P_{j1} ---- *le segment voisin du segment* P_{i2}.
P_{j2} ---- *le segment connexe du segment* P_{j1}.



Ces quatres segments constituent le début d'une frontière à chaîner. Ensuite on commence le chaînage. Le segment qui suit le segment connexe (par exemple P_{j2}) du segment courant (par exemple P_{j1}) au cours du chaînage est déterminé parmi ses segments voisins (par exemple P_{k1} et P_{m1}) selon le critère de "Angle Minimal" :

$$AG(\vec{P_{j2}P_r},\ \vec{P_{l1}P_{l2}}) = min\{AG(\vec{P_{j2}P_r},\ \vec{P_{k1}P_{k2}}), AG(\vec{P_{j2}P_r},\ \vec{P_{m1}P_{m2}})\}.$$

$$l \in (k, m).$$

où $AG(\vec{x},\ \vec{y})$ est l'opérateur calculant l'angle entre deux vecteurs ($\leq 360^0$) et P_{l1} un des segments voisins du segment P_{j2}.

La valeur de l'angle entre deux vecteurs sera évaluée suivant la direction déterminée par les deux vecteurs $\vec{P_{j2}P_r}$ et $\vec{P_{j2}P_{j1}}$:

- La direction est celle du sens de l'aiguille d'une montre si le vecteur $\vec{P_{j2}P_r}$ se situe à droite du vecteur $\vec{P_{j2}P_{j1}}$.

- La direction est celle de l'inverse de l'aiguille d'une montre si le vecteur $\vec{P_{j2}P_r}$ se situe à gauche du vecteur $\vec{P_{j2}P_{j1}}$.

L'intersection du segment suivant avec le segment courant nous fournit un point sommet sur la frontière balayée. Si l'on suppose qu'un couple segment peut éventuellement contribuer à deux frontières (fermées ou non), on définit un poids de "2" pour chaque couple segment. Chaque fois quand on a trouvé une frontière, on va décrémenter les poids des couples segments chaînés par cette frontière.

On résume cette méthode ci_dessous :

Algorithme II :

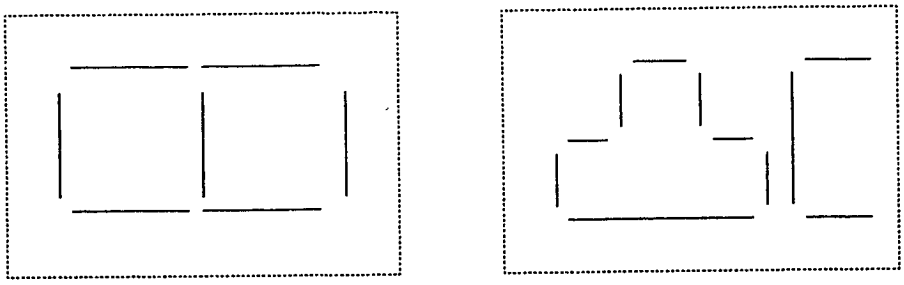
1. **Construire la liste de segments voisins pour chaque segment.**
2. **Construire une frontière de départ en prenant un segment ayant un poids de deux et son plus proche voisin. calculer le point de reference P_r.**
3. **Chercher le segment qui suit le segment courant en utilisant le critère d' "Angle Minimal".**
4. **Tester la fermeture de la frontière balayée:**

   (a) Si le segment pris en compte n'a pas de segment voisin, on a trouvé une frontière ouverte. Décrémenter les poids des couples segments chaînés.

   (b) Si l'un des segments voisins du segment pris en compte est le prenier segment de la frontière, on a trouvé une frontière fermée. Décrémenter les poids des couples segments chaînés.

   (c) Sinon, continuer l'étape 3.
5. **Répéter l'étape 2, 3 et 4 jusqu'à ce que tous les segments aient un poids inférieur à deux.**

Cette algorithme réussit à traiter les cas de configuration des segments suivants :



(a) (b)

8 Résultats Expérimentaux

Au cours du développement de notre projet de recherche, nous nous avons mis au point un système expérimental de vision dynamique intitulé VIDYR (Système de Vision Dynamique pour la Robotique). Les Modules principaux de VIDYR sont presentés à la figure 9 où on trouve :

1. Le noyau de VIDYR.
   Sa fonction est de gérer la communication des données entre les modules et éventuellement l'utilisateur. Les données utilisées dans VIDYR sont généralement : (a) les images, (b) les trajectoires, (c) les paramètres et (d) les résultats des traitements. Pour faciliter cette tâche, nous avons défini un jeu de structures de données adapté aux différents niveaux des primitives de la vision. Le noyau de VIDYR gère aussi l'intégration d'une nouvelle application.

2. L'interface graphique.
   Le système VIDYR est développé sur la station SUN sous UNIX. Nous utilisons les utilitaires de SUNVIEW et de SUNCORE pour créer un environnement graphique destiné à l'interfaçage graphique entre les modules de VIDYR et l'utilisateur, et aussi à l'affichage des résultats de traitements (par exemple: l'image des contours, l'image des segments 2D/3D, etc).

3. VISYR.
   Nous avons à notre disposition un simulateur de vision en mouvement

intitulé VISYR (Vision Synthétique pour la Robotique) développé à l'IRISA. Ce simulateur a été conçu au sein de l'algorithme de "Lancé de Rayons" qui a pour rôle de synthétiser l'image d'une scène spécifiée par le langage LGRC. L'intégration de VISYR dans VIDYR permet de tester les algorithmes de vision avec des données synthétiques. Ceci est très important pour analyser rapidement la validité de certains algorithmes.

4. Le Robot Mobile.
   Nous avons réalisé un banc expérimental de vision en mouvement. Ce banc est constitué d'un robot manipulateur AID sur le poignet duquel est installée une caméra CCD calibrée de marque MICAM. Le manipulateur est sous contrôle d'un processeur de traitement d'images EDIXIA. Ce dernier a une liaison directement avec la station SUN. Le système VIDYR peut commander facilement le robot en vue d'acquérir une séquence d'images de résolution 256×256 pour une tâche robotique donnée.

5. L'application.
   Tous les algorithmes d'applications sont développés au sein de VIDYR. La programmation est basée sur le langage C. En fait, VIDYR représente un outils de recherche pour le développement des algorithmes de vision.

Plus précisément, nous avons utilisé VIDYR pour réaliser le travail concernant la reconstruction d'une scène 3D polyédrique. La figure 10 montre les modules de traitements principaux dans ce contexte.

En tant que module de traitement indispensable, tous les algorithmes décrits dans ce rapport ont été réalisés au sein de VIDYR. Le test de ces algorithmes est fait sur des données synthétiques qui sont imposées d'une erreur de 5%.

9 Conclusion et Discussion

Nous avons étudié une méthode robuste qui traite avec succès le problème de l'identification des polygones 3D à partir des segments 3D. Les deux

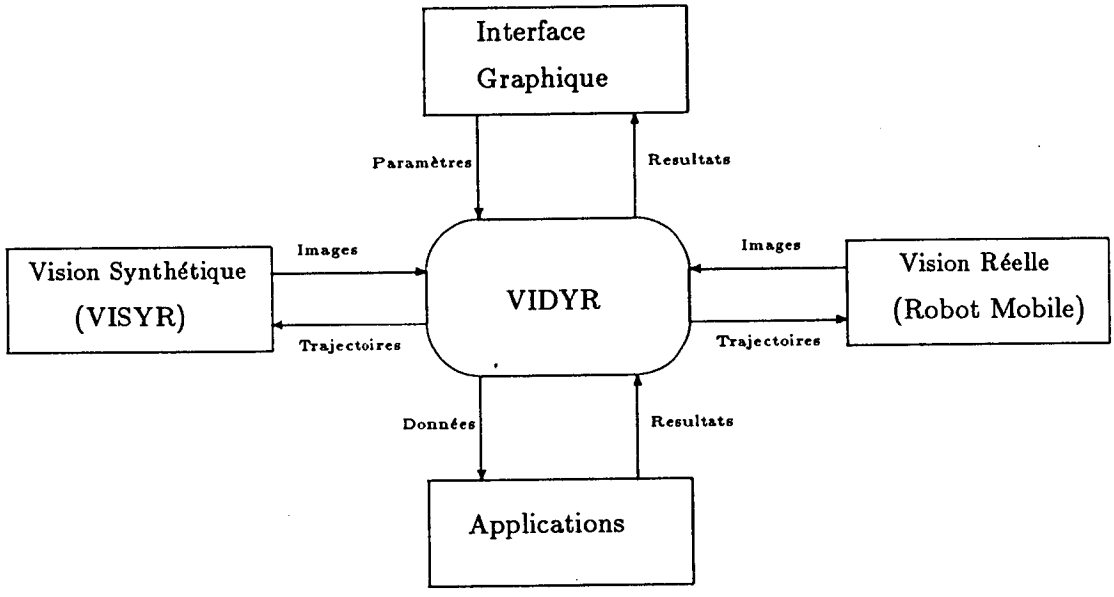


Figure 9 : Description De VIDYR

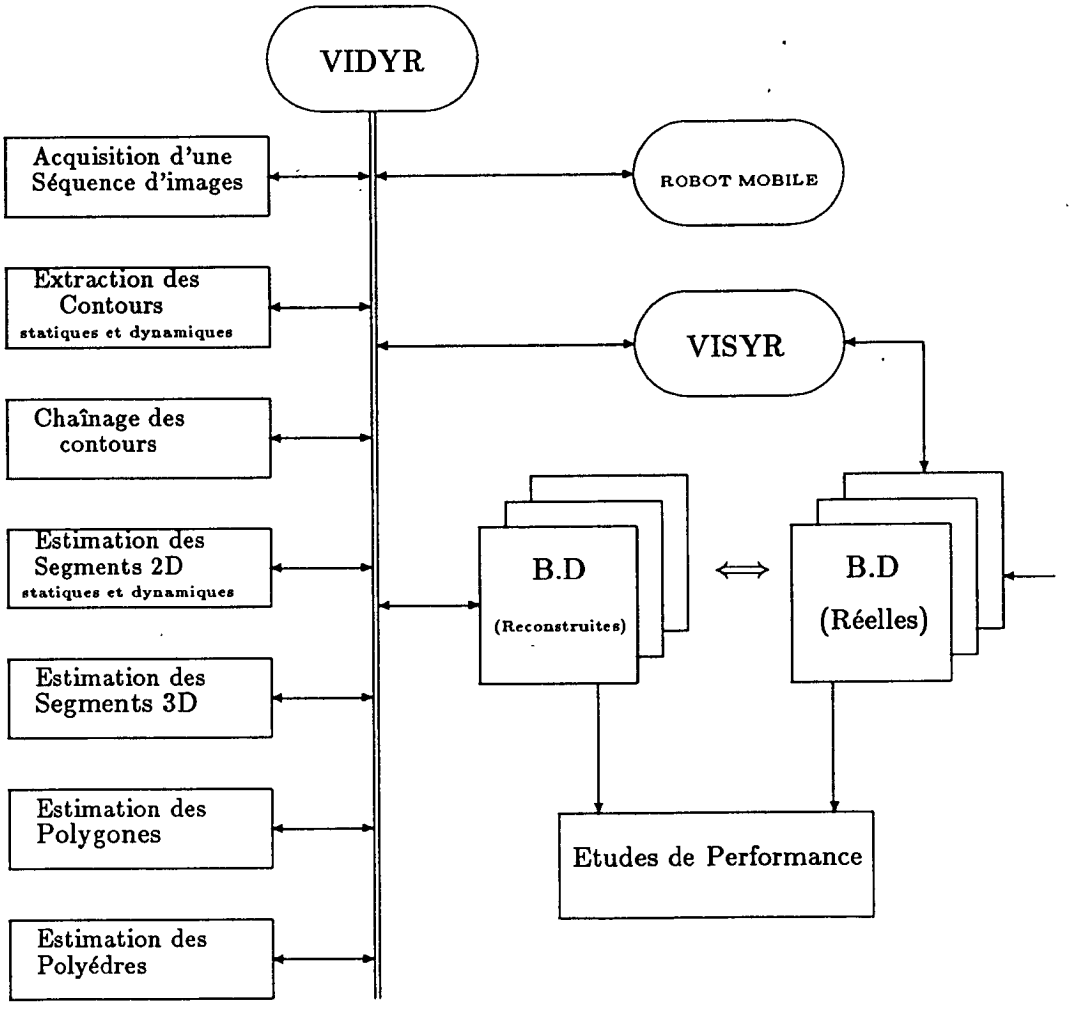


Figure 10 : Application de VIDYR pour la Reconstruction d'une Scène 3D Polyèdrique

techniques: La Prediction/Vérification Dynamique et La Transformée de Hough Généralisée permettent d'identifier effectivement les paramètres des surfaces planes qui sont des supports physiques des polygones 3D à estimer. Ensuite nous avons développé un algorithme de chaînage des segments 3D en vue d'obtenir les arêtes et les sommets des polygones 3D. Dans le cas où les segments 3D estimés sont très bruités, ce dernier traitement peut être remplacé par la triangulation de Delaunay ou la recherche des enveloppes convexes des segments 3D, etc.

La connaissance sur les polygones 3D dans un univers polyédrique est indispensable pour l'exécution des tâches robotiques de haut niveau. Les résultats obtenus par notre méthdoe seront exploitables directement pour de futures recherches dans le domaine de la vision artificielle appliquée au guidage automatique d'un robot mobile.

Bibliographie

[1] BALLARD.D.H : " Generalizing The HOUGH Transform to Detect Arbitrary Shapes ", *Pattern Recognition Vol.13,No.2* , pp.111-122, 1981.

[2] BOISSONNAT.J.D, O.D.FAUGERAS and E.LEBRAS : "Représentation des donnée stéréo par la triangulation de Delaunay", *Congrès AFCET-RFIA*, Antibes/FRANCE, Novembre, 1987.(in french)

[3] BORGNE.M.L : " QUATERNIONS ET CONTROLE SUR L'ESPACE DES ROTATIONS ", *Rapport Intern de l'IRISA*, No.377, Octobre, 1987.

[4] BOUTHEMY.P : " Estimation Of Edge Motion Based on Local Modelling ", *SPIE Vol.595, Computer Vision For Robotis* pp.162 - 169 , Cannes, December, 1985.

[5] DERICHE.R : "Using Canny's Criteria to Derive a Recursively Implemented Optimal Edge Detector", *International Journal of Computer Vision*, P167-187,1987.

[6] ESPIAU.B and RIVES.P : " Closed-Loop Recursive Estimation Of 3D Features For a Mobile Vision System ", *IEEE, International Conference on Robotics and Automation* March 30 - Aril 3 , Raleigh, 1987.

[7] FREEMAN.H and DAVIS.L.S : "A corner-finding algorithm for chain-codes curves", *IEEE Trans. Comput. 26(3)*, P297-303, 1977.

[8] GIRAUDON.G : " Chaînage Efficace de Contour ", *Rapport de Recherche No.605*, centre SOPHIA ANTIPOLIS de l'INRIA, 1987.

[9] HEGRON.G, ARNALDI.B and PRIOL.T : "VISYR : A simulation tools of synthetic vision for robotics", *Pro. of MARI87, Vo.II* , Paris, mai 1987.

[10] KUROZUMI.Y and DAVIS.W.A : "Polygonal Approximation by the Minimax Method", *Computer Graphics and Images Processing 19* , 248-264, 1982.

[11] LEU.J.G and CHEN.L : "Polygonal approximation of 2D shapes through boundary merging", *Pattern Recognition letters 7* , 231-238, 1988.

[12] LOWE.D.G : "Three-Dimensional Object Recognition from Single Two-dimensional Images", *Artificial Intelligence 31* , 355 - 395, 1987.

[13] PAVLIDIS.T : "SURVEY : A Review of Algorithms for Shap Analysis", *Computer Graphics and Images Processing 7*, 243-258, 1978.

[14] RIVES.P : "Dynamic Vision: Theoretic Capability and Practical Problems" *Nato Workshop on Kinematic and Dynamic Issues in Sensor Based Control*, IL CIOCCO, TUSCANY Italy, October 25-31, 1987.

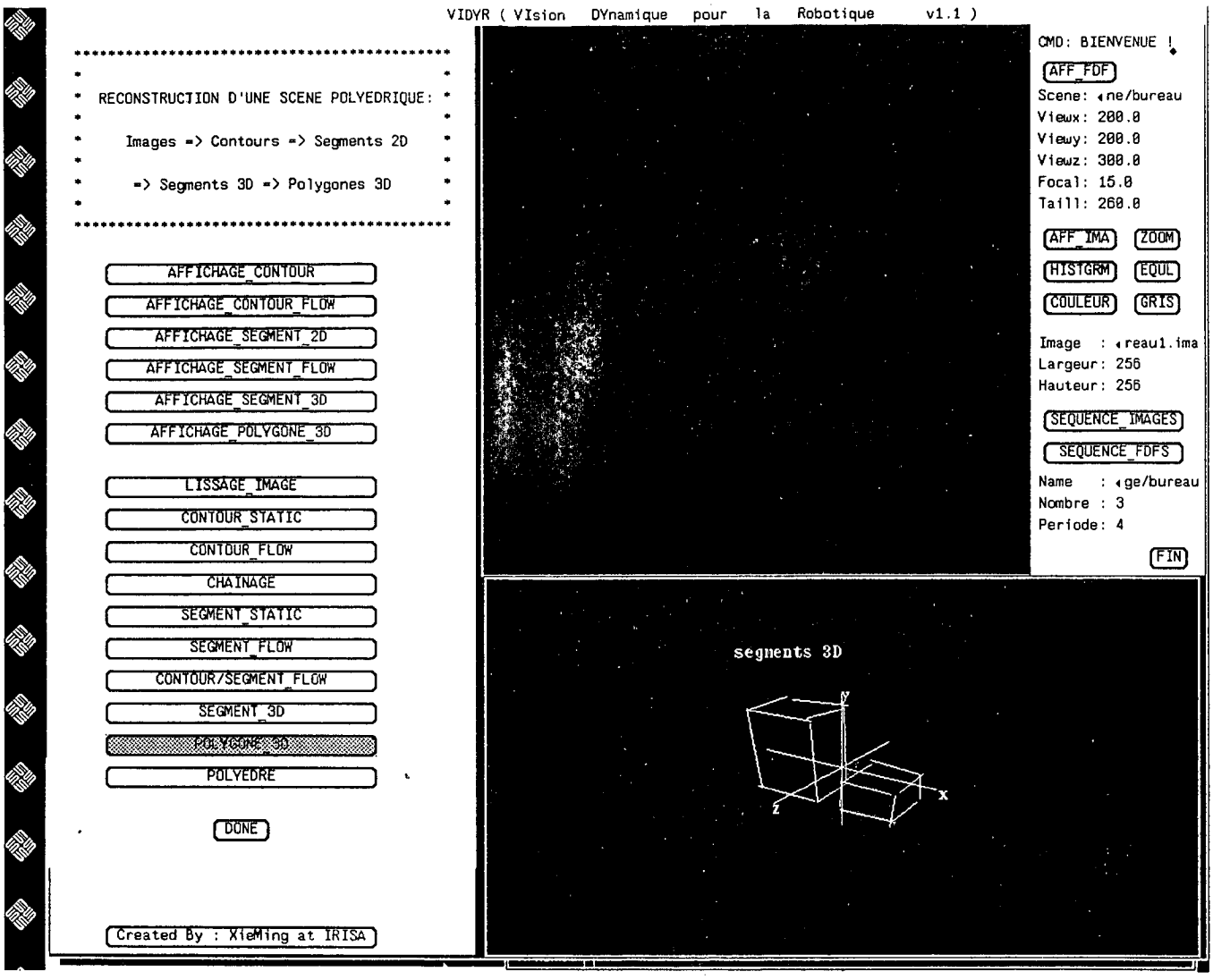


Figure 11 : L'ensemble des segments 3D synthétiques. On a ajouté des bruits uniformes de 5%.

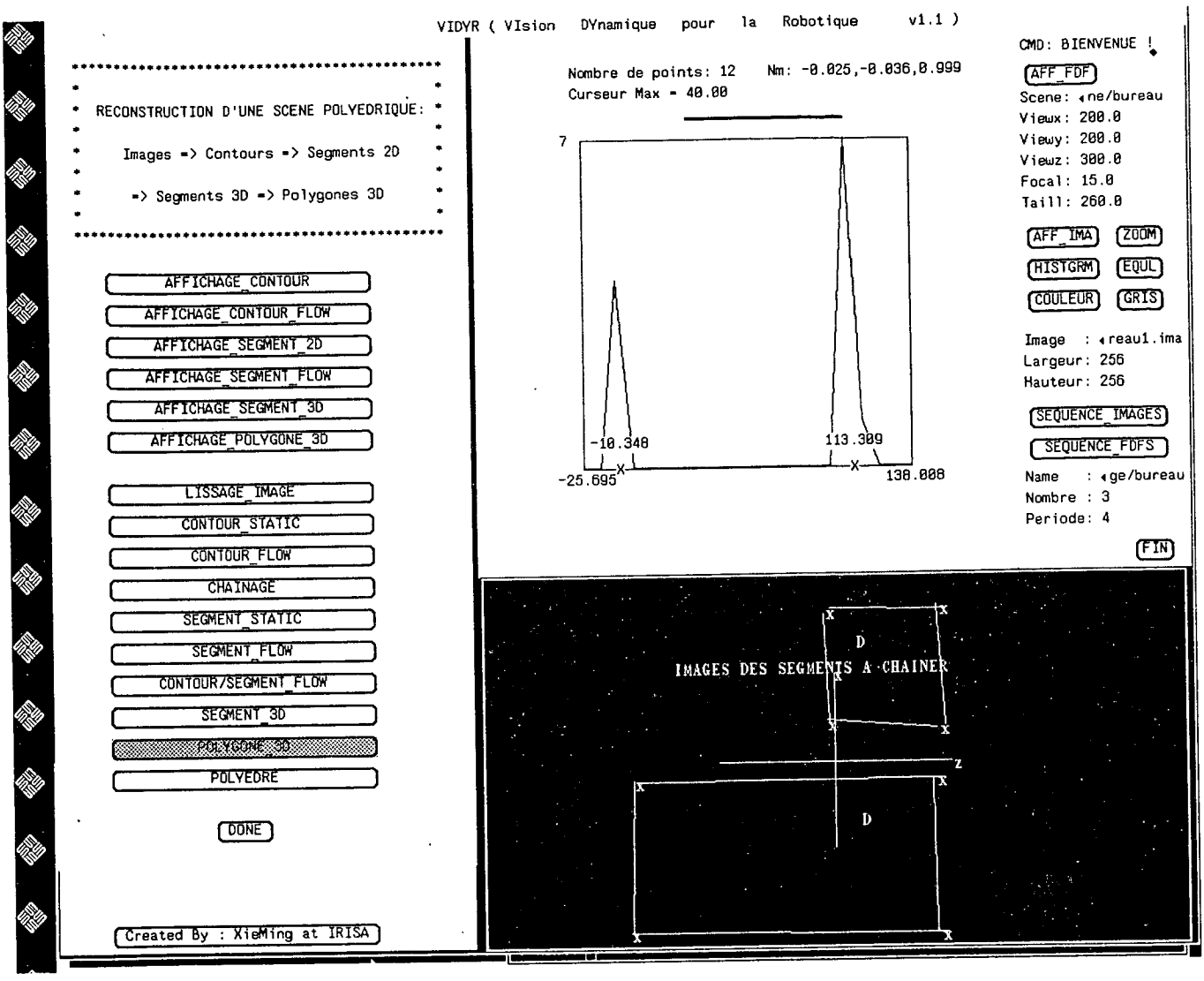


Figure 12 : L'image en haut est un exemple de résultats de l'estimation des paramètres ρ, les croix indiquent les positions de ρ estimés dans son repère. L'image en bas est un exemple de résultats du chaînage des segments 3D, les croix indiquent les sommets des polygones 3D estimés

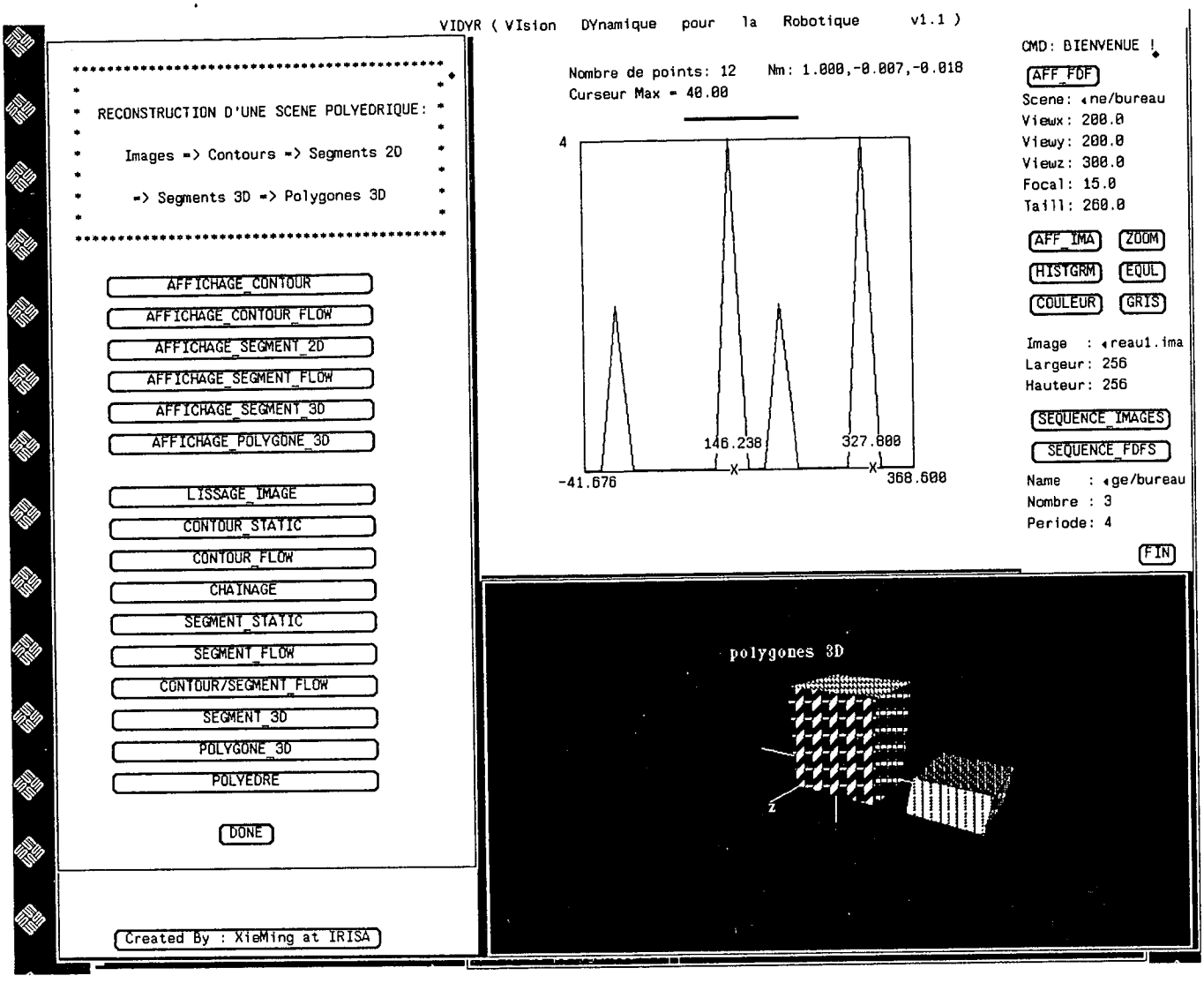


Figure 13 : Les polygones 3D estimés

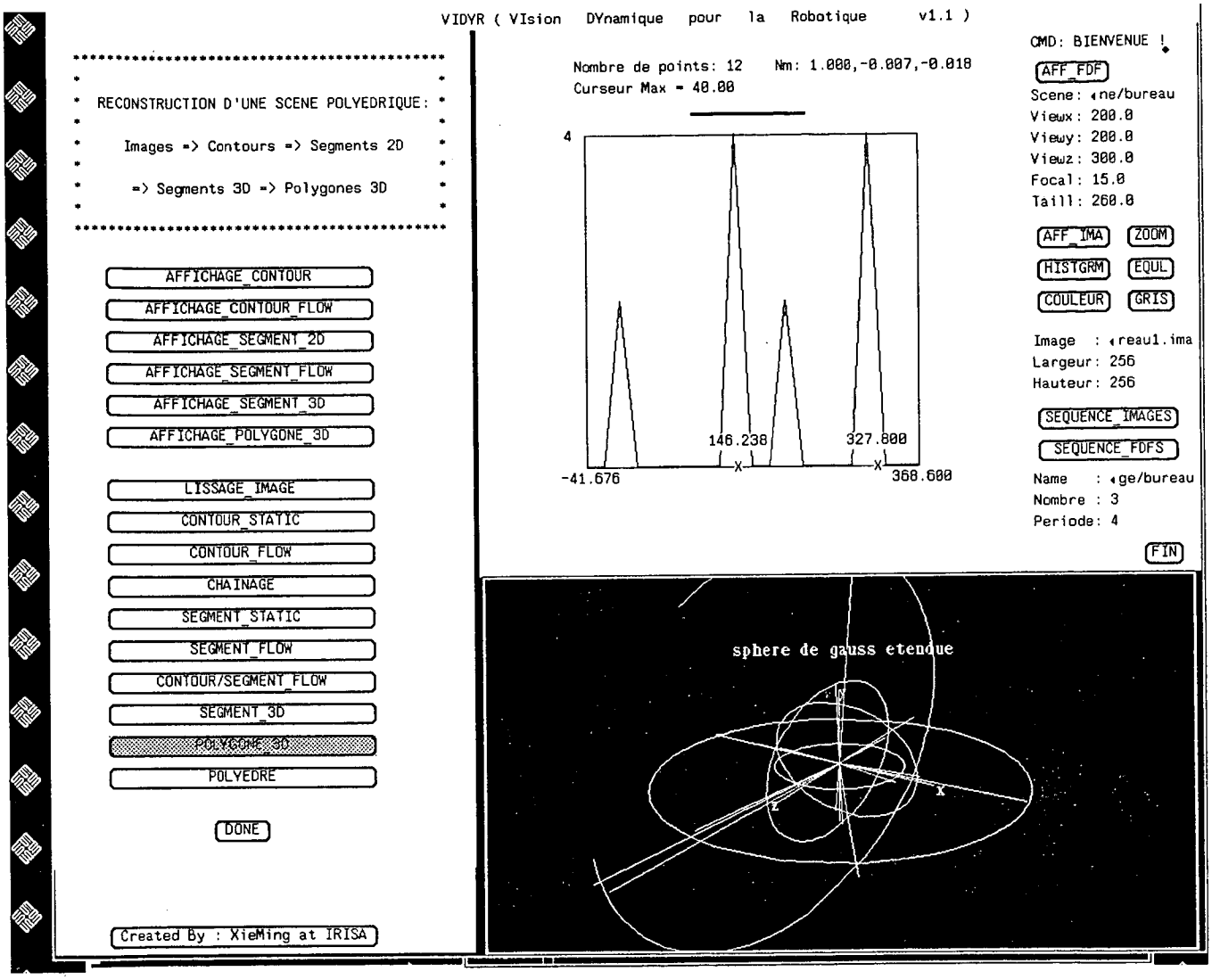


Figure 14 : La représentation de la scène polyédrique identifiée dans le Sphère Gaussienne Etendue. Les orientations des cercles indiquent les directions des polygones 3D. Les rayons des cercles indiquent les paramètres ρ des polygones 3D

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