HAL
open science

A local approach for matching contour segments

Samia Boukir, Patrick Bouthemy, Didier Juvin, François Chaumette

To cite this version:

Samia Boukir, Patrick Bouthemy, Didier Juvin, François Chaumette. A local approach for matching contour segments. [Rapport de recherche] RR-1792, INRIA. 1992. inria-00077032

HAL Id: inria-00077032

https://inria.hal.science/inria-00077032v1

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.

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

25ème anniversaire

N° 1792

Programme 4
Robotique, Image et Vision

MISE EN CORRESPONDANCE DE SEGMENTS DANS UNE SÉQUENCE D'IMAGES PAR UNE APPROCHE LOCALE

Samia BOUKIR
Patrick BOUTHEMY
François CHAUMETTE
Didier JUVIN

Novembre 1992

RR.1792

IRISA

INSTITUT DE RECHERCHE EN INFORMATIQUE
ET SYSTEMES ALEATOIRES

Campus Universitaire de Beaulieu
35042 - RENNES CEDEX FRANCE
Tél. : 99 84 71 00 - Télex : UNIRISA 950 473 F
Télécopie : 99 38 38 32

Mise en correspondance de segments dans une séquence d'images par une approche locale

Publication Interne n° 685 - Octobre 1992 - 30 pages - Programme 4

Samia Boukir*, Patrick Bouthemy*

François Chaumette*, Didier Juvin**

* IRISA-INRIA Rennes

** CEA-LETI DEIN-SLA Saclay

Résumé

Ce rapport présente une approche locale pour la mise en correspondance de segments dans une séquence d'images. Ce travail complète une étude antérieure sur la reconstruction 3D de cylindres par vision active qui nécessite un algorithme capable de suivre les limbes des cylindres dans une séquence d'images à une cadence proche du temps réel. Nous proposons une approche originale et robuste qui répond parfaitement aux exigences de ce problème. Cette méthode s'appuie sur un algorithme de calcul d'éléments de contours en mouvement qui se distingue par sa robustesse aux aléas de segmentation et d'occlusion, et par sa facilité de mise en œuvre. Nous l'avons ensuite implémentée sur une machine parallèle afin d'atteindre les performances de calcul requises (temps réel). Cette méthode a été validée sur plusieurs séquences d'images et s'est avérée très performante.

CENTRE NATIONAL DE LA RECHERCHE SCIENTIFIQUE (U.R.A. 227) UNIVERSITE DE RENNES I I.N.S.A. DE RENNES
INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE (UNITE DE RECHERCHE DE RENNES)

A local approach for matching contour segments

Abstract

This paper presents a local approach for matching contour segments in an image sequence. This study is related to previous work dealing with the recovery of 3D structure of cylinders using active vision. This issue requires an algorithm able to track in real-time the limbs of cylinders in an image sequence. Therefore, we propose an original and robust approach which is perfectly suitable for this problem. This method relies on an algorithm for determining moving edges which is distinguished by its robustness to segmentation unstability and to occlusion and its easiness to implement. We have implemented this solution on a parallel architecture board in order to satisfy real-time processing constraints. This method has been validated on several image sequences and turned out to be a powerful one.

1 Introduction

1.1 Position du problème

La finalité de notre travail est de remonter, à partir d'informations fournies par une caméra embarquée sur un robot, à une représentation 3D de la scène perçue. Dans notre cas, la scène est supposée constituée de tuyaux, l'application visée étant la navigation et l'inspection à l'intérieur des bâtiments auxiliaires d'une centrale nucléaire. Ils seront modélisés par des cylindres. Une méthode originale de reconstruction de cylindres par vision active a déjà fait l'objet de travaux antérieurs [3], [7] et a donné d'excellents résultats en simulation. La stratégie de reconstruction consiste d'abord à se focaliser sur un seul cylindre (ou plus exactement sur un ou deux limbes du cylindre) de la scène. Ensuite, des mouvements optimaux, engendrés par des lois de commande en boucle fermée sur les informations visuelles, permettent de positionner la caméra de manière à optimiser la reconstruction 3D du cylindre.
Dans ce travail, nous allons aborder le problème délicat de mise en correspondance de segments dans une séquence d'images, sans lequel notre méthode de reconstruction ne pourrait fonctionner en environnement réel. En effet, les limbes d'un cylindre seront perçus dans la séquence d'images comme des segments de droite correspondant à des contours d'intensité. On se pose ici deux questions essentielles auxquelles on tentera d'apporter une réponse :

1. comment se focaliser, dans la première image de la séquence, sur un limbe parmi les limbes des cylindres présents dans la scène ?

2. une fois ce limbe sélectionné, comment l'apparier de manière fiable à ses correspondants dans les images suivantes ?

1.2 Contraintes temps réel

Il est important de souligner ici, avant de rentrer dans les détails de notre approche, que l'élément essentiel qui aura guidé nos choix algorithmiques, sans toutefois négliger les contraintes de robustesse et de fiabilité, est l'aspect temps réel. En effet, nos algorithmes de traitement d'images doivent respecter des contraintes fortes de temps puisque une commande en boucle fermée sur la vision est à la base de notre méthode de reconstruction 3D. Une partie importante du travail a donc consisté à implanter les algorithmes choisis sur la machine parallèle SYMPATI2 [2]. Il s'agit d'une carte multi-processeurs de type SIMD adaptée au traitement d'images, développée par le CEA en collaboration avec l'IRIT, et en cours d'industrialisation par la société CENTRALP Automatismes. Nous reviendrons plus en détail sur cette carte dans la suite de ce rapport.

2 Focalisation sur un segment de la scène

2.1 Introduction

Le traitement des images permet d'extraire de l'image brute, considérée comme une matrice de pixels, l'information pertinente sous forme de contours, de régions,... selon les besoins de l'application envisagée. En ce qui nous concerne, on cherche à détecter une entité image de type segment de contour ou, plus exactement, on se limitera à une chaîne de points contours. La sélection d'une chaîne potentielle est effectuée selon un critère très simple : la longueur des chaînes. Nous allons traiter ce problème en trois phases :

1. la détection des points de contraste.
2. la sélection d'une droite support significative par la transformée de Hough.
3. le repérage de la chaîne de points contours la plus longue sur la direction sélectionnée.

2.2 Détection des points de contraste

Les pixels image correspondant à des variations de niveaux de gris importantes sont appelés les points de contraste. Des recherches récentes ont mis au point des détecteurs de contours très performants tels que par exemple le filtre de Deriche [10] ou le filtre de Shen-Castan [17]. Les deux filtres sont inspirés du filtre de Canny [6] et ont des performances très similaires. Nous avons choisi le second filtre puisque disponible sur la machine SYMPATI2.

Le filtre de Shen-Castan [17] est un filtre exponentiel symétrique permettant l'utilisation de masques de convolution de taille infinie. Les opérateurs differentiels d'ordre un et deux sont déterminés simplement par un algorithme récursif. La détection de contours est alors réalisée en calculant au moyen de ces opérateurs, les passages par zéro des dérivées secondes dans la direction du gradient. Ce filtre est optimal au regard de trois critères :

- il maximise le rapport signal/bruit.
- il localise au mieux le point de contraste.
- il aboutit à un pic unique.

Enfin c'est un algorithme relativement simple et qui est facilement parallélisable sur une machine de type SIMD.

2.3 Transformée de Hough

La transformée de Hough (TH) permet de détecter dans des images la présence de courbes paramétriques appartenant à une famille connue à partir d'un ensemble de points sélectionnés [12], [14]. La TH utilise essentiellement la position dans l'image des points, mais, parfois, tient compte également de la valeur de luminance en ces mêmes points.

2.3.1 Définition

Pour une famille de courbes de la forme $h(\underline{x},\underline{p}) = 0$, on quantifie l'espace des paramètres $\{\underline{p}\}$ et on associe un accumulateur $A(\underline{p})$ à chaque élément de l'espace (cellule) ainsi quantifié.

Pour chaque point sélectionné $\underline{x}_i$, on détermine dans l'espace quantifié l'ensemble $\{\underline{p}\}$ de tous les paramètres possibles $\underline{p}_j$ qui vérifient la relation :

$$h(\underline{x}_i, \underline{p}_j) = 0 \tag{1}$$

et on incrémente les accumulateurs correspondant aux paramètres $\underline{p}_j$ (étape de vote) :

$$A(\underline{p}_j) = A(\underline{p}_j) + 1, \quad \underline{p}_j \in \{\underline{p}\} \tag{2}$$

Les meilleurs paramètres $\underline{p}^*$ sont obtenus par la recherche des maxima locaux dans le tableau $A(\underline{p})$, soit :

$$\{\underline{p}^*\} = Argmax_{\underline{p}}\, A(\underline{p}) \tag{3}$$

2.3.2 Détection de droites

Le paramétrage cartésien d'une droite dans un plan $h(x, y, a, b) = y - ax - b = 0$ n'est généralement pas le meilleur choix car l'espace $\{a, b\}$ est infini et très inhomogène [12]. De plus, ce paramétrage nécessite l'emploi de deux cartes. Un paramétrage polaire $\{\rho, \theta\}$ est donc plus satisfaisant (voir la figure 1).

Une droite est alors définie par l'équation suivante :

$$h(x, y, \rho, \theta) = x\cos\theta + y\sin\theta - \rho = 0$$

2.3.3 Mise en œuvre de la TH

Pour la sélection d'une droite support dans l'image, nous avons choisi la transformée de Hough car elle se prête aisément à une parallélisation sur une architecture du type SIMD. En effet, chaque opération que nous avons décrite ci-dessus est répétée identiquement sur tout point M considéré dans l'image. D'autre part, c'est une méthode

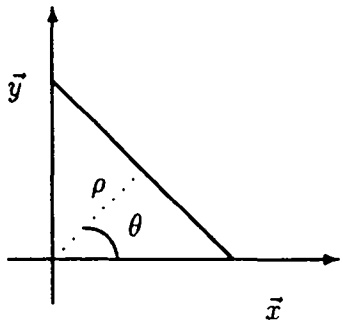


Figure 1 : Droite en représentation polaire

robuste au bruit et aux occlusions.

Si on choisit la représentation (ρ, θ) des droites, l'espace des paramètres sera alors l'espace (ρ, θ) avec $0 \leq \rho \leq \Delta/2$ où Δ est la diagonale de l'image et $0 \leq \theta < 2\pi$. Cet espace est discrétisé en cellules uniformes avec un pas sélectionné de façon expérimentale. La TH que nous avons mise en œuvre procède en deux étapes, toujours dans l'optique d'optimiser le temps de calcul :

- une étape grossière avec un pas de 10 degrés pour la dimension θ et un pas de 6 pixels pour la dimension ρ. Cette étape permet de repérer la partie de l'espace de paramètres la plus dense en points contour. Une fois la procédure de vote effectuée par les points de contraste de l'image, le maxima $(\rho_{max}, \theta_{max})$ est repéré par la cellule du tableau $A(\underline{p})$ ayant accumulé le maximum de votes.

- une étape fine avec un pas de 1 pour les deux dimensions θ et ρ. On aurait pu discrétiser plus finement l'espace des paramètres mais dans notre cas, ce n'était pas nécessaire, cette précision étant suffisante pour explorer, par l'opération de chaînage, la droite support qui sera sélectionnée. Dans cette seconde étape, il suffit de se focaliser sur une fenêtre autour des valeurs $(\rho_{max}, \theta_{max})$ sélectionnées dans l'étape précédente. La recherche est ainsi beaucoup moins exhaustive puisque guidée par le résultat de l'étape grossière. Cette fois, le maxima $(\hat{\rho}, \hat{\theta})$ du tableau $A(\underline{p})$ correspond aux paramètres de la droite support sélectionnée sur laquelle il nous reste à détecter la chaîne de points contours la plus longue.

On peut noter qu'il est possible aussi de retenir d'autres critères de sélection ou de considérer si souhaité des contraintes particulières, comme par exemple que θ soit compris entre deux valeurs données θ_1 et θ_2. La TH offre un moyen très simple de les mettre en œuvre grâce à la matrice d'accumulation.

2.4 Extraction des chaînes de points de contraste

Le processus de chaînage permet de regrouper les points de contraste en lignes de contraste. Ces lignes représentent les lieux de l'image où l'intensité lumineuse varie fortement dans une direction. Chaque ligne correspond dans la scène à une arête d'un objet, une ligne de discontinuité de surface, ou une ligne de changement de luminance. Dans le cas particulier de notre application, elles représentent plus exactement les limbes des cylindres (ou parfois des ombres générées par ces derniers).

L'algorithme de chaînage que nous avons développé est relativement simple. En effet, à l'issue de l'étape précédente de classification par la transformée de Hough, nous disposons d'une information clef sur les points de contraste à chaîner : les paramètres $(\tilde{\rho}, \tilde{\theta})$ de la droite support sélectionnée. Le chaînage s'effectue en trois étapes :

1. création des chaînes : La recherche des points d'une chaîne s'effectue dans un voisinage 8-connexe, le champ de recherche étant limité à une bande centrée sur la droite support définie par les paramètres $(\tilde{\rho}, \tilde{\theta})$;

2. élimination des chaînes trop courtes et fusion des chaînes voisines :

3. sélection de la chaîne la plus longue : elle représente le "segment virtuel" qu'il faudra suivre tout au long de la séquence d'images. En effet, comme cela sera motivé ci-dessous, on ne procède pas à une approximation polygonale des chaînes de contour et on conserve les chaînes de points contours comme niveau de représentation. Par ailleurs, si la chaîne sélectionnée n'est pas satisfaisante (longueur inférieure à un seuil prédéfini), on relance le processus de chaînage en considérant le maxima suivant de la TH.

3 Suivi du segment dans la séquence d'images par une approche locale

3.1 Introduction

La mise en correspondance d'indices bi-dimensionnels entre deux images successives est loin d'être un problème trivial. En effet, autant il est facile pour un être humain d'effectuer cette mise en correspondance, autant l'élaboration d'un algorithme permettant de la réaliser de manière automatique, fiable et robuste est une tâche ardue [8], [11], [16], [18]. La complexité de cette mise en correspondance vient du nombre généralement élevé d'indices dans les images et des ambiguïtés possibles d'appariement.

Des solutions ont été étudiées qui s'appuient soit sur l'utilisation d'information contextuelle, mise en œuvre par des méthodes de relaxation [16], soit sur une prise en compte plus complète de la dimension temporelle du problème par des techniques de filtrage récursif (prédiction/correction) [11], [9]. Ces méthodes sont toutes basées

sur l'extraction explicite des segments de contour après chaînage et approximation polygonale des chaînes de contour obtenues. De plus le critère d'appariement met en jeu des paramètres globaux des segments comme leur longueur et leur orientation, parfois leur contraste moyen. Ces techniques sont assez performantes, mais elles sont par contre sensibles à l'instabilité de la phase de segmentation (fragmentation d'un segment d'une image à l'autre). De plus, elles requièrent l'élaboration d'une carte dédiée pour obtenir une solution temps réel (quand celle-ci peut être obtenue).

Nous avons préféré privilégier une approche locale qui, en fait, ne manipule jamais explicitement de segments de contour. Elle ne nécessite la recherche de correspondants aux points de la chaîne de contour que dans une seule direction. Elle présente les avantages suivants :

- une seule structure de données est considérée : la matrice de pixels ;
- elle est implémentable sur des cartes programmables offrant des opérations de convolution temps réel ;
- elle est robuste aux aléas de segmentation et d'occlusion ;
- elle est adaptée au niveau d'information requis par les étapes ultérieures de la reconstruction 3D tout en étant générale et versatile.

La méthode de mise en correspondance de segments que nous avons mise en œuvre s'appuie sur un algorithme de calcul des éléments de contours en mouvement (ECM) précédemment développé dans l'équipe [4], [5]. Cet algorithme est parfaitement adapté aux besoins de notre application tant au niveau des implications algorithmiques que des considérations de robustesse. Avant de décrire notre méthode qui représente une extension significative de cet algorithme, nous allons rappeler brièvement ses principes de fonctionnement.

3.2 Détermination des appariements locaux de points contours

Elle consiste en une modélisation spatio-temporelle d'un élément de contour en mouvement et utilise un test de rapport de vraisemblance pour en déterminer la localisation, l'orientation spatiale et la composante de déplacement perpendiculaire au contour.

3.2.1 Modélisation d'un élément de contour en mouvement (ECM)

Une séquence d'images est considérée comme un espace à trois dimensions : deux dimensions spatiales (x,y) et une dimension temporelle t. Dans cet espace, un élément de contour en mouvement (ECM) correspond à une portion de surface spatio-temporelle 3D. Cette surface peut être approximée localement par une portion de surface plane avec comme représentation paramétrique (voir la figure 2) :

1. une position (x_0, y_0, t_0) et une direction spatiale $\vec{m}$ de l'elément de contour :
2. un vecteur de déplacement $\vec{V}$ associé à l'ECM.

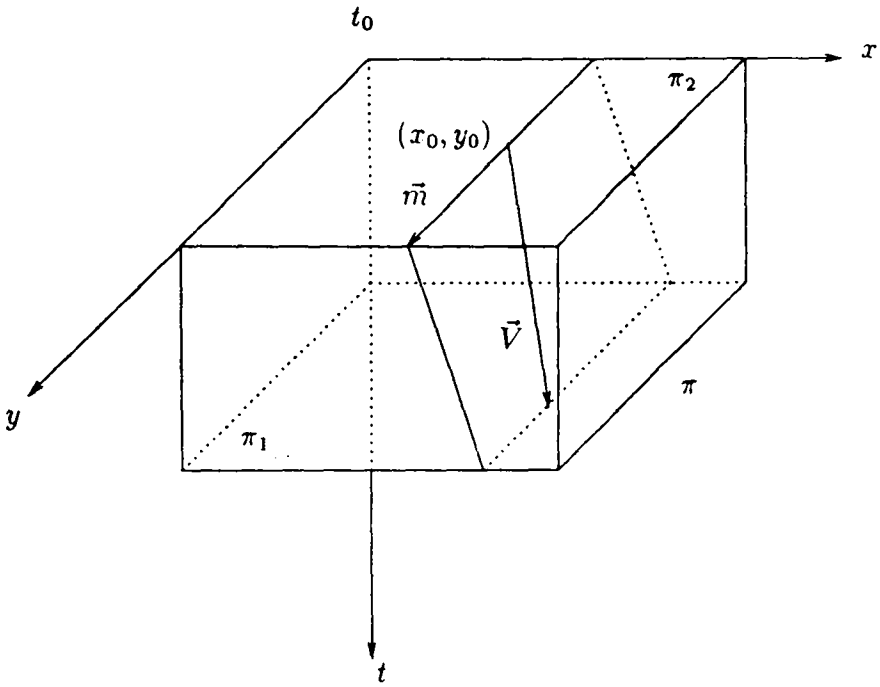


Figure 2 : Modélisation géométrique d'un ECM.

Le choix d'une telle approximation se justifie par le fait que l'on considère un écart de temps très faible entre deux images. Des modèles de surface plus complexes peuvent également être considérés [15]. La méthode décrite ici permet la détection de telles portions de plan et l'estimation de leurs paramètres $\vec{m}$ et $\vec{V}$. Rappelons que seule la composante orthogonale $V^{\perp}$ de $\vec{V}$ pourra être déterminée, la composante tangentielle étant impossible à mesurer dans une fenêtre d'observation locale [4], [5].
Soit π le volume placé au point (x_0, y_0, t_0) dans l'espace (x, y, t). On prend en compte les deux hypothèses de configurations locales suivantes :

1. **Hypothèse H_0 :**
   Il n'existe pas d'ECM dans le volume π. La distribution des intensités lumineuses à l'intérieur du volume π peut être modélisée par un niveau de gris constant c_0 perturbé par un bruit gaussien centré de variance σ^2, soit : $\{c_0, \sigma^2\}$.

2. **Hypothèse H_1 :**
   Il existe un ECM dans le volume π. Cela se traduit par la présence d'une portion de plan divisant le plan π en deux sous volumes π_1 et π_2. Comme dans le cas précédent, la distribution des intensités lumineuses dans ces deux volumes est modélisée par :

- $\{c_1, \sigma^2\}$ pour le sous-volume π_1.
- $\{c_2, \sigma^2\}$ pour le sous-volume π_2 avec $c_1 \neq c_2$.

3.2.2 Test de vraisemblance pour la détermination des ECM :

La distribution des intensités lumineuses dans un volume π suit un modèle de densité Gaussienne :

- $N(c_0, \sigma^2)$ en tout point du volume π pour l'hypothèse H_0.
- $N(c_1, \sigma^2)$, $N(c_2, \sigma^2)$ respectivement en tout point du volume π_1 et du volume π_2 pour l'hypothèse H_2.

A chaque hypothèse peut être associée une fonction de vraisemblance. densité jointe des variables considérées. Ces dernières seront supposées indépendantes.

- Pour l'hypothèse H_0, la fonction de vraisemblance L_0 s'exprime par :

$$L_0 = (2\pi\sigma^2)^{-\frac{n}{2}} \prod_{p_i \in \pi \,:\, i=1}^{n} exp\left\{-\frac{1}{2\sigma^2}(f_{p_i} - c_0)^2\right\} \tag{4}$$

- Pour l'hypothèse H_1, la fonction de vraisemblance s'écrit :

$$L_1 = (2\pi\sigma^2)^{-\frac{n_1+n_2}{2}} \prod_{p_j \in \pi_1 \,:\, j=1}^{n_1} exp\left\{-\frac{1}{2\sigma^2}(f_{p_j} - c_1)^2\right\} \prod_{p_k \in \pi_2 \,:\, k=1}^{n_2} exp\left\{-\frac{1}{2\sigma^2}(f_{p_k} - c_2)^2\right\} \tag{5}$$

où :

  - $f(m)$: note l'intensité lumineuse du point m.
  - n, n_1, n_2 : représentent respectivement le nombre de points dans le volume π, π_1 et π_2.

Un ECM est déterminé à l'aide d'un test de rapport de vraisemblance. La sélection de l'hypothèse H_1contre H_0 s'appuie sur l'expression suivante :

$$\xi = \ln(L_1/L_0) \tag{6}$$

qui dépend de deux types de paramètres, θ étant l'angle que fait $\vec{m}$ avec l'axe des x :

- les paramètres géométriques $\Phi = (x_0, y_0, \theta, V^{\perp})$ qui déterminent la portion de plan modélisant un ECM.
- les paramètres photométriques et statistiques notés α qui déterminent les valeurs moyennes (c_0, c_1, c_2) et la variance du bruit σ^2 sur les intensités lumineuses.

L'hypothèse H_1 est retenue si la valeur maximale du rapport $\xi(\Phi, \alpha)$ est supérieure à un seuil prédéterminé :

$$max_{\Phi}\ max_{c_1,c_2}\ min_{c_0}\ \xi(\Phi,\alpha) \ \geq \ \lambda \tag{7}$$

3.2.3 Dérivation du critère sous forme d'une convolution :

Considérons une géométrie Φ_g prédéfinie, dans le domaine de variation possible des paramètres θ et $V^{\perp}$. Après évaluation des estimateurs optimaux de c_0, c_1, c_2, la racine carrée de la valeur maximale de la fonction ξ peut s'écrire [4], [5] :

$$\zeta(\Phi_g) = \left| \sum_{\eta \in M} a_\eta^g f(p+\eta) \right| \tag{8}$$

avec M un jeu d'indices de masque (a_η^g) tel que $(p+\eta)$ désigne tous les points dans le volume π centré au point p et où les coefficients a_η^g sont donnés par :

$$\begin{cases} a_\eta^g \ = \ \frac{1}{\sigma}\left(\frac{n_2}{2n_1(n_1+n_2)}\right)^{1/2}, & \text{si } \eta \in \pi_1 \\ a_\eta^g \ = \ -\frac{1}{\sigma}\left(\frac{n_1}{2n_2(n_1+n_2)}\right)^{1/2}, & \text{si } \eta \in \pi_2 \end{cases} \tag{9}$$

Le critère de vraisemblance devient alors :

$$max_{\Phi_g}\ \zeta(\Phi_g) \ \geq \ \lambda \tag{10}$$

Ainsi, le calcul du maximum de vraisemblance est réduit à une simple opération de convolution de la fonction intensité avec un jeu de masques associés à une géométrie prédéfinie, qu'on peut donc calculer hors-ligne.

Par ailleurs, une contrainte supplémentaire est introduite pour éviter les fausses détections et pour améliorer la robustesse en présence de bruit ou de forte densité des contours spatiaux. Avant de décider qu'un ECM est détecté ou pas selon le critère (10), la double inégalité suivante doit être vérifiée :

$$\mu_1 \ \leq \ |\zeta_{t_2}(\Phi_g) \ / \ \zeta_{t_2}(\Phi_g)| \ \leq \ \mu_2 \tag{11}$$

où μ_1 et μ_2 sont deux seuils prédéterminés.

Ainsi, la sélection de candidats pour lesquels les contrastes sont proches est favorisée. Enfin, le procédé peut être optimisé en effectuant préalablement une détection de contours spatiaux, les ECM étant avant tout des éléments de contours spatiaux. On aura alors à estimer uniquement les composantes de vitesse perpendiculaires à ces points contours [18]. Notons que dans cette version de base, on teste systématiquement toutes les orientations et les déplacements autorisés.

3.2.4 Discussion

L'intérêt de la méthode que nous venons de décrire réside dans le fait que :

- la modélisation exploite au mieux l'information sur les aspects géométrique et photométrique des pixels contour en mouvement ;
- L'opération de mesure de vraisemblance peut être ramenée à une simple opération de convolution avec des masques prédéterminés. Or, l'architecture SIMD de la machine SYMPATI2 est parfaitement adaptée à ce type d'opération. Cette méthode répond donc à nos objectifs temps réel.

Il faut cependant préciser que la détermination de l'ensemble des ECM d'une image pourrait être relativement lourde (hors temps réel) notamment quand la densité des contours spatiaux est importante. Or, dans le cadre de notre application, il s'agit de traiter un seul segment à la fois dans l'image, donc d'effectuer le calcul des ECM sur une petite partie des contours spatiaux de l'image. En effet, la méthode de reconstruction 3D s'effectue selon une approche de type vision active, un mouvement optimal est appliqué à la caméra et est géré en boucle fermée sur les informations capteur [3], [7]. Elle entraîne donc une focalisation tour à tour sur chacun des cylindres d'intérêt. Cela réduit considérablement la charge de calcul et va nous permettre comme on le verra par la suite, d'aboutir à des temps d'exécution tout à fait raisonnables.

3.3 Application au suivi d'un seul segment

Une fois l'étape de focalisation effectuée, il nous faut mettre en correspondance et suivre le segment "virtuel" sélectionné dans la séquence d'images. Nous allons proposer ici une méthode originale qui exploite une version appropriée de l'algorithme de détermination des ECM. C'est de ce fait, une méthode locale puisqu'opérant au niveau pixel. Elle consiste en les étapes suivantes :

- calcul des masques de convolution.
- détermination de la zone de recherche.
- calcul du test de vraisemblance.

que nous décrivons à présent.

3.3.1 Calcul des ECM

Etant donné la liste de pixels L_i représentant un segment virtuel S_i d'une image i de la séquence, on cherche à déterminer les correspondants de ces points dans l'image suivante $i+1$ par la méthode des ECM que nous venons de décrire. En fait, il s'agit d'une nouvelle version dérivée pour le problème que nous avons à résoudre ici. L'algorithme que nous avons mis en œuvre procède de la manière suivante :

1. Masque de convolution optimal : La connaissance de l'orientation $\hat{\theta}$ du segment de l'image i (en fait celle de la droite support calculée après moindres carrés comme expliqué plus loin) permet de choisir les masques de convolution les mieux adaptés pour la mesure de vraisemblance. Ces masques de convolution optimaux garantissent l'obtention de résultats nettement plus performants que la version de base. Rappelons que dans [5], il était proposé de considérer un jeu de masques correspondant à une quantification assez grossière (par exemple tous les 45º). En pratique, les masques de convolution correspondant à une direction donnée $0^o \leq \theta \leq 180^o$ sont calculés avec une précision de 1º, ce qui est tout à fait satisfaisant pour la détermination du déplacement 2D apparent.
Soit θ^*, la valeur entière la plus proche de $\hat{\theta}$, pour laquelle on calcule les masques de convolution correspondants $M^1_{|\theta^*}$ et $M^2_{|\theta^*}$ [5]. L'image i (respectivement l'image $i+1$) est alors convoluée avec le masque optimal $M^1_{|\theta^*}$ (respectivement $M^2_{|\theta^*}$). En fait, il s'agit des mêmes masques $M^1_{|\theta^*} = M^2_{|\theta^*}$, lorsque l'on ne considère que des déplacements entiers.

2. Espace de recherche : Pour tout point contour de l'image i , on recherche le point correspondant, dans un espace prédéfini de l'image $i+1$, pour lequel la mesure de vraisemblance est maximale. La recherche la plus efficace doit s'effectuer dans la direction orthogonale au contour (voir la figure 3).

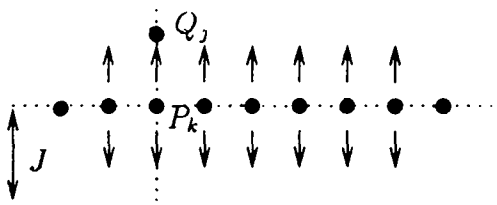


Figure 3 : Recherche des ECMs dans la direction orthogonale au contour.

où les notations introduites ont les significations suivantes :

- $P_k \in L_i$: point contour du segment à l'instant t.
- Q_j : nouvelle position possible du point P_k dans la direction orthogonale au contour à l'intant $t+1$.
- $P_k \overset{?}{\longmapsto} Q_j$, $j \in [-J, J]$: appariement possible entre les points P_k à l'instant t et le point Q_j à l'instant $t+1$, avec J l'amplitude maximale des déplacements recherchés.

Selon l'orientation de la normale à P_k, les déplacements ne correspondent pas toujours à des positions Q_j entières de la matrice de pixels (problèmes d'échantillonnage). L'évaluation en des positions non entières nécessite soit de recalculer les masques de

convolution, soit de procéder à des interpolations sur la fonction intensité. Ces opérations sont trop côuteuses. Nous avons donc opté pour une solution qui, d'une part, satisfait les contraintes temps réel, et d'autre part, ne manque pas d'efficacité quant à la recherche des ECM. En effet, nous avons pris en considération quatre directions principales de recherche : $\delta \in \{0^o, 45^o, 90^o, 135^o\}$ qui, elles, induisent des positions Q_j entières après déplacement (voir la figure 4). Parmi ces quatre directions, on sélectionne la direction δ^* la plus proche de la direction orthogonale à θ^* :

$$\delta^* = \delta_i \ / \ \delta_i \in \{0^o, 45^o, 90^o, 135^o\} \text{ et } \left|(\theta^* + \frac{\pi}{2}) - \delta_i\right| (mod\ \pi) < \frac{\pi}{4} \qquad (12)$$

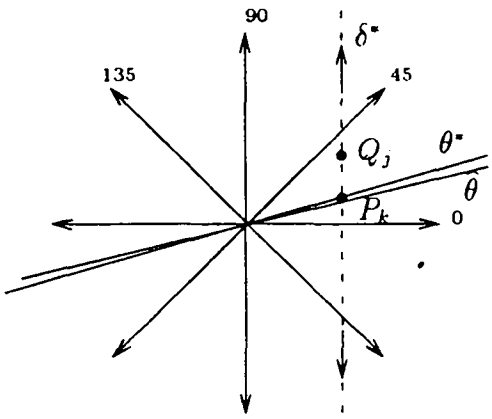


Figure 4 : Choix de la direction de recherche

Il faut souligner que l'objectif est d'apparier deux positions successives d'un limbe d'un cylindre. On exploite ici le fait que l'on n'a nul besoin de connaître le déplacement 2D complet de chaque élément de contour. Seule nous intéresse in fine la détermination des nouvelles valeurs des paramètres ρ, θ de la projection du limbe du cylindre dans l'image $i+1$. La recherche des correspondants des points P_k de la liste L_i est alors effectuée dans la direction δ^*, l'amplitude maximale des déplacements recherchés étant prédéfinie. Si l'on peut prédire la prochaine position du segment, la zone de recherche peut alors être délimitée autour de cette prédiction. Dans le cas de notre approche en boucle fermée, cette information se déduit aisément connaissant le mouvement de la caméra et ayant estimé la structure spatiale 3D du cylindre correspondant [3]. De plus, la zone de recherche sera de plus en plus limitée au fur et à mesure de la convergence du processus.

3. Détermination des ECM selon le test de vraisemblance : Pour chaque point P_k de la liste L_i de l'image i, étant donné un déplacement P_kQ_j dans la direction de recherche δ^*, la valeur de vraisemblance ζ est donnée par la valeur absolue de la somme des deux valeurs issues de la convolution avec les masques correspondants $M^1_{|\theta^*}$ et $M^2_{|\theta^*}$. L'ECM correspondant à la valeur maximale ζ, à condition qu'elle soit supérieure à un

seuil λ prédéfini, est enregistré dans la liste L_{i+1}. On obtient ainsi une nouvelle liste de points qui sont les homologues des points de la liste précédente L_i.

3.3.2 Estimation des paramètres de la droite support des ECM

Connaissant la liste L_{i+1} des ECM ou, plus exactement, des correspondants locaux aux points de la liste L_i, nous utilisons une technique de moindres carrés pour estimer les paramètres $(\widehat{\rho}_{i+1}, \widehat{\theta}_{i+1})$ de la droite support. Ensuite, on calcule les résidus du moindre carré qui vont constituer une mesure de la qualité de l'estimation :

$$\varepsilon_{quad} = \frac{1}{n} \sum_{j \in L_{i+1}} (x_j \cos \widehat{\theta}_{i+1} + y_j \sin \widehat{\theta}_{i+1} - \widehat{\rho}_{i+1})^2 \tag{13}$$

n étant le nombre de points de la liste L_{i+1}.
Si l'erreur ε_{quad} est importante, il s'agit d'un mauvais appariement. Il peut arriver en effet que l'algorithme de calcul des ECM soit en présence d'une ambiguïté très forte : si deux contours, de contrastes de valeurs très proches et de signe identique sont présents dans le champ de recherche, la confusion est possible. En effet, certains points vont s'apparier à un contour, d'autres points avec un second contour. Ces cas rares sont observés en pratique lorsque les images présentent une symétrie de motifs répétitifs.
Par ailleurs, à cause notamment de perturbations, l'algorithme peut détecter ponctuellement quelques "faux ECM". Ces valeurs aberrantes sont éliminées en reparcourant la liste de points L_{i+1} et en vérifiant qu'elles sont compatibles avec les valeurs estimées $(\widehat{\rho}_{i+1}, \widehat{\theta}_{i+1})$:

$$\left| x_j \cos \widehat{\theta}_{i+1} + y_j \sin \widehat{\theta}_{i+1} - \widehat{\rho}_{i+1} \right| < \varepsilon_{aber} \tag{14}$$

La liste L_{i+1} est alors mise à jour et le moindre carré est relancé pour affiner l'estimation des paramètres du segment suivi.
Notons qu'une "maintenance" (ajout de points) de la liste L_{i+1} s'impose ici pour assurer un suivi robuste et fiable du segment virtuel S_{i+1}. En effet, il est possible que les points situés à l'une des extrémités du segment n'aient pas trouvé de correspondants. Ce phénomène est d'autant plus marqué que le mouvement de glissement du segment sur lui-même est important. D'autre part, des occultations partielles du segment peuvent aussi contribuer à "l'altération" de la liste L_{i+1}. Celle-ci doit alors être mise à jour dès que la partie occultée du segment est à nouveau visible. Enfin, il faut substituer aux quelques points Q_j éventuellement déclarés aberrants les véritables points situés sur le support d'origine.

Notons que la méthode d'appariement décrite peut très bien s'appliquer au suivi simultané de plusieurs segments dans la séquence d'images. Il n'y a aucune limitation intrinsèque sur ce point.

3.3.3 Rappel de l'algorithme

(1) Etape de focalisation :

1. détection des contours spatiaux dans la première image.
2. sélection d'une droite significative (ρ_0^*, θ_0^*) (transformée de Hough).
3. repérage du segment le plus long sur cette direction spatiale $\Longrightarrow$ liste de pixels L_0.

(2) Etape de suivi :

1. recherche des correspondants des pixels de la liste L_i dans l'image suivante (méthode des ECM) $\Longrightarrow$ liste L_{i+1}.
2. estimation par moindres carrés des paramètres $(\hat{\rho}_{i+1}, \hat{\theta}_{i+1})$ de la droite support des points de la liste L_{i+1}.
3. mise à jour de L_{i+1}.
4. $i+1 \longrightarrow i$; aller à (1).

4 Présentation des résultats

L'algorithme de suivi de segments qui vient d'être présenté a été testé sur plusieurs séquences d'images. Les différentes scènes considérées ne sont pas forcément constituées d'objets cylindriques, l'objectif étant de montrer les performances de notre approche en environnement quelconque.
Tous les tests ont été réalisés avec les paramètres suivants :

- masques de taille 3×3.
- seuil de vraisemblance $\lambda = 1500$.
- coefficients $\mu_1 = 0.75$ et $\mu_2 = 1.25$.
- amplitude maximale des déplacements $= \pm 10$.

4.1 Séquence "table"

La scène enregistrée est constituée d'une table sur laquelle est posée une boite blanche. La séquence a été acquise en faisant des rotations de 1º entre chaque image autour d'un axe fixe aligné avec la verticale de l'image. La boite a été simultanément déplacée approximativement dans la direction de la table. La figure (5) représente respectivement à gauche la première et à droite la dernière image de la séquence en niveaux de gris.

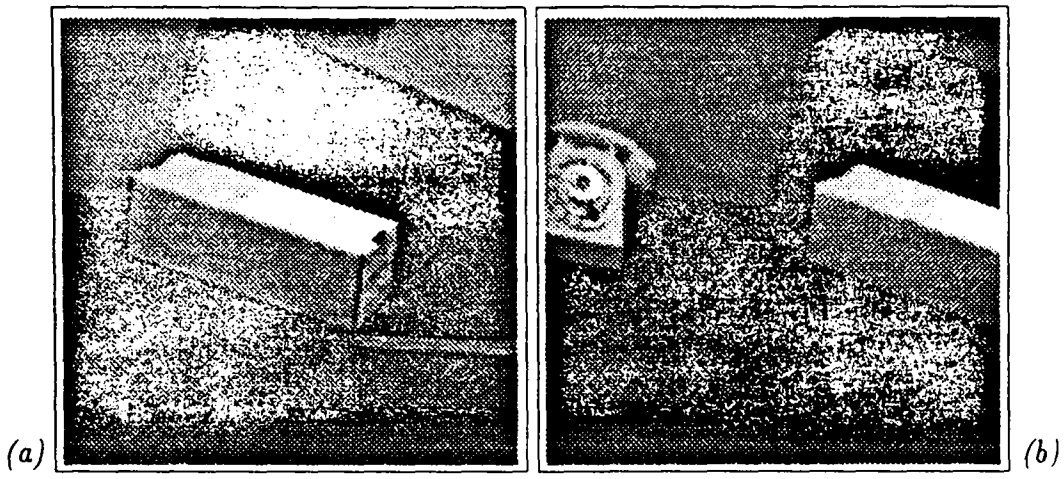


Figure 5 : (a) Première image de la séquence "table"; taille : 256 × 256. (b) Dernière image de la séquence "table".

La figure (6) montre en gras le segment sélectionné initialement par la transformée de Hough. Nous illustrons alors le suivi de ce segment dans la séquence d'images par la figure (7). A gauche est représentée la séquence de segments correspondant au segment initial sélectionné et à droite, est mise en évidence la droite support de la position finale estimée du segment en accord avec le segment extrait de cette image. On peut remarquer que le suivi du segment est parfaitement maitrîsé par notre algorithme tout au long de la séquence d'images.

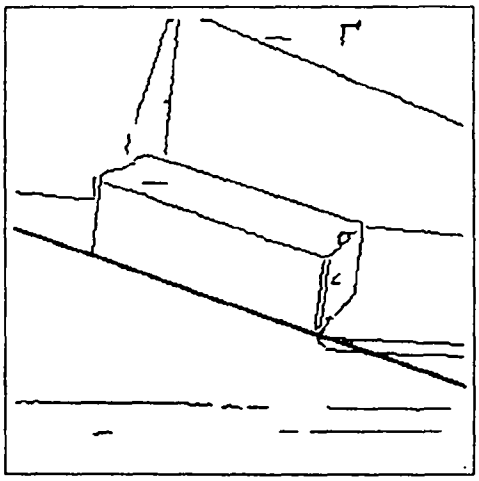


Figure 6 : Droite support du segment sélectionné dans la première image.

Performances sur SYMPATI2 :

(1) Etape de focalisation : L'étape de focalisation est relativement gourmande en temps de calcul et dépend de la densité des contours contenus dans la première image de la séquence. Le temps d'exécution de cette étape est de 232.5 ms soit :

- 40 ms pour la détection des contours spatiaux (Shen-Castan) de la première image.
- 184 ms pour la transformée de Hough.
- 8.5 ms pour le chaînage.

Ce temps de calcul peut être réduit de manière très significative en privilégiant par exemple les droites descendantes (pente négative) dans l'étape grossière de la transformée de Hough. L'espace des paramètres est alors réduit à l'espace (ρ, θ) avec $0 \leq \rho \leq \Delta/2$ et $0 \leq \theta \leq \pi/2$ ou $\pi \leq \theta \leq 3\pi/2$, ce qui divise par deux le temps d'exécution de cette étape. Le temps de calcul de la phase de focalisation est alors réduit à 152.5 ms.

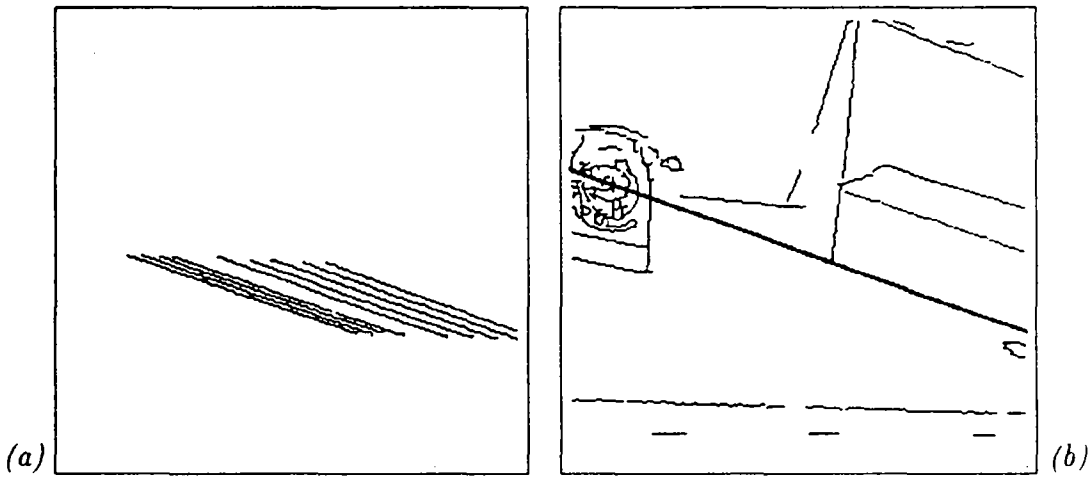


Figure 7 : (a) Séquence de segments correspondant au segment sélectionné. (b) Position finale du segment estimé en accord avec le segment extrait de cette image.

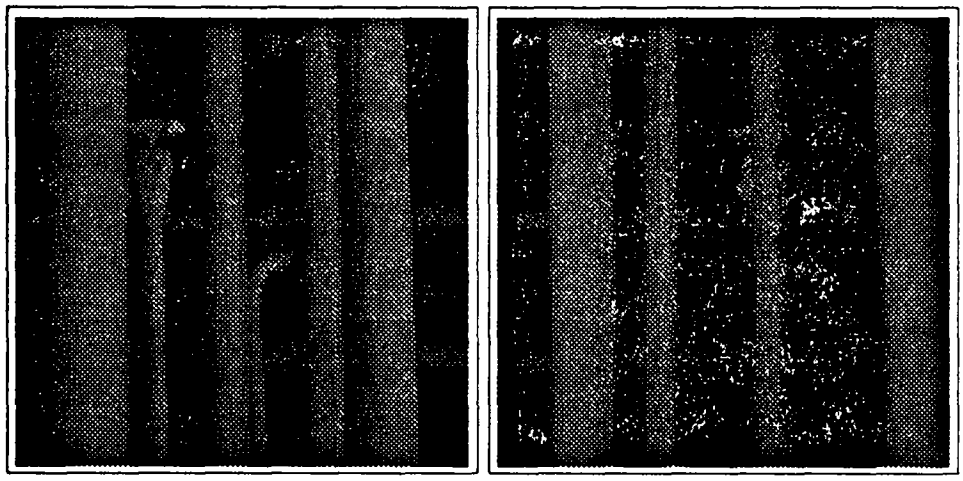
Notons que la lourdeur relative de cette étape ne compromet en aucun cas les performances de notre algorithme de reconstruction 3D en boucle fermée, vu qu'elle ne s'effectue qu'une seule fois au début du suivi d'un segment. Par contre, il est nécessaire que l'étape de suivi du segment qui, elle, est répétitive, soit proche du temps réel. Il ne s'agit pas nécessairement de la cadence vidéo mais plutôt de la fréquence d'échantillonnage de la boucle fermée pour une réalisation adéquate de la commande, soit une cadence minimale de l'ordre de 10 HZ.

(1) Etape de suivi : Le temps de calcul relatif à une itération de suivi du segment sélectionné de la séquence "table" est en moyenne de 54 ms. Ce temps dépend de la longueur du segment, de l'amplitude des déplacements et aussi de l'orientation du segment relativement au segment de PEs (horizontal). Rappelons que l'amplitude maximale des déplacements recherchés est ici de ± 10. Or, la zone de recherche sera de plus en plus restreinte au cours de la convergence du processus. On pourra alors se

limiter à une amplitude de ± 2 par exemple qui réduit ce temps à 35 ms seulement. Ce temps, bien en dessous des 100 ms est tout à fait satisfaisant pour notre application.

4.2 Séquence "tuyaux"

La scène dans ce cas est constituée de plusieurs tuyaux et d'une vanne. Elle illustre ainsi adéquatement l'intérieur des bâtiments d'une centrale nucléaire. Cette séquence a été acquise en faisant effectuer de petits déplacements à la caméra autour de la scène qui, elle, reste fixe. La figure (8) montre la première et la dernière image de la séquence en niveaux de gris. Sur la figure (9) est représenté en gras le segment qui a été sélectionné. Le résultat du suivi de ce segment est montré sur la figure (10) qui révèle une fois de plus l'efficacité de notre algorithme. Notons qu'une technique basée sur la segmentation polygonale aurait difficilement suivi ce segment sans être confrontée (entre autres) à une confusion avec le second limbe du cylindre ou à des fragmentations instables de l'approximation polygonale. Par contre, notre algorithme qui prend en compte l'information photométrique des ECM, différencie aisément les deux segments et tout risque de confusion est ainsi écarté. De même, l'approche locale qui a été choisie montre ici tout son intérêt. En effet, réaliser l'appariement global entre les deux segments requiert seulement qu'un nombre suffisant d'appariements locaux soit effectué (donc robustesse aux ambiguïtés locales, aux occlusions et même possibilité de sous-échantillonner le nombre d'éléments de contour pris en compte dans la chaîne extraite).



(a) (b)

Figure 8 : (a) Première image de la séquence "tuyaux", taille : 256 × 256. (b) Dernière image de la séquence "tuyaux".

Le temps de calcul de la phase de focalisation est de 303 ms dont 40 ms pour la détection des contours spatiaux, 253 ms pour la transformée de Hough et 10 ms pour le chaînage. Ce temps est, comme on pouvait facilement le prévoir, plus important que dans le cas de la séquence "table" puisque la densité des contours spatiaux de la

première image de la séquence est ici plus importante.

Le temps de calcul d'une étape de suivi du segment sélectionné était initialement de 150 ms en moyenne. Ce temps assez contraignant est dû comme on peut le voir sur la figure (10) à la taille importante du segment (255 pixels) et surtout à son inclinaison pratiquement verticale (91º). En effet, le segment des 32 PEs de la carte SYMPATI2 (voir Annexe) parcourt l'image de façon linéaire. Par conséquent, le suivi d'un segment vertical est un cas très pénalisant. On ne profite quasiment pas de la disponibilité des 32 PEs. Pour y remédier, on propose une procédure très simple qui permet d'améliorer les performances de SYMPATI2 sur cette étape de suivi. Elle consiste à effectuer préalablement pour tout segment proche de la verticale une simple transposition de l'image contenant la chaîne de points correspondant au segment courant ainsi que des deux images de niveau de gris. Cette opération très peu côuteuse en temps de calcul (4 ms) permet alors de diviser pratiquement par trois le temps de calcul (temps de transposition compris) soit 51 ms en moyenne. Pour une amplitude maximale des déplacements recherchés de ± 2, ce temps est encore plus intéressant (38 ms). Remarquons que ce temps reste du même ordre de grandeur que pour la séquence précédente malgré la différence très significative de taille des segments considérés. Cela s'explique par le fait que cette fois on se trouve dans un cas de figure très favorable où on exploite au maximum les 32 PEs disponibles puisque le segment à suivre est horizontal. Il est important de souligner ici que notre approche de reconstruction 3D optimale de cylindres consiste justement à se ramener à des configurations particulières où le cylindre apparaît dans l'image sous la forme de deux droites verticales ou horizontales symétriques par rapport au centre de l'image [3]. Cela permet donc d'assurer outre l'optimisation de l'estimation de la structure spatiale 3D du cylindre, des performances de calcul tout à fait remarquables sur SYMPATI2.

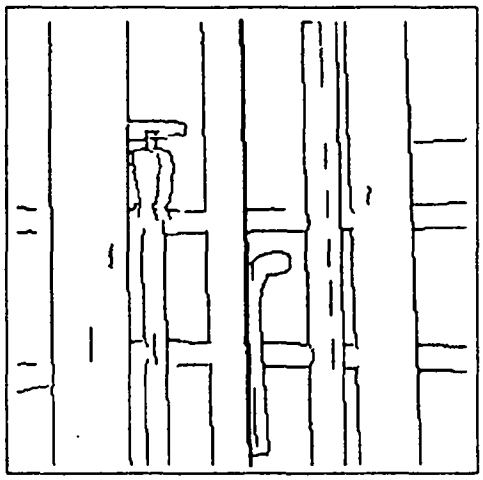


Figure 9 : Droite support du segment sélectionné dans la première image.

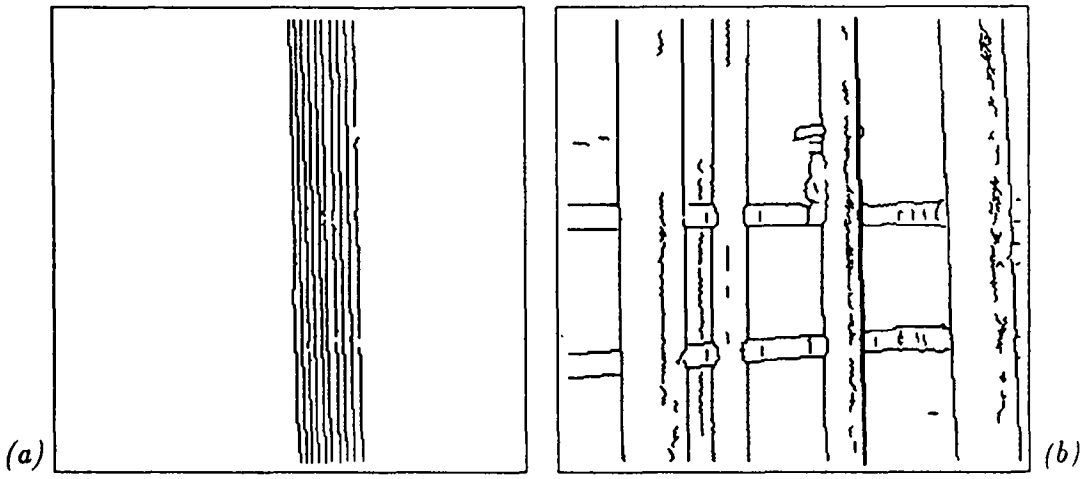


Figure 10 : (a) Séquence de segments correspondant au segment sélectionné. (b) Position finale du segment estimé en accord avec le segment extrait de cette image.

4.3 Séquence "mire"

La scène enregistrée est constituée d'un cylindre vertical et d'une mire qui restent fixes tout au long de l'acquisition de la séquence. Entre la première et la dernière image, la caméra a effectué une translation de 500 mm dans un plan horizontal et une rotation de 35º autour de l'axe vertical de la caméra, la scène étant située à environ 1m. La figure (11) représente la première et la dernière image de la séquence en niveaux de gris.

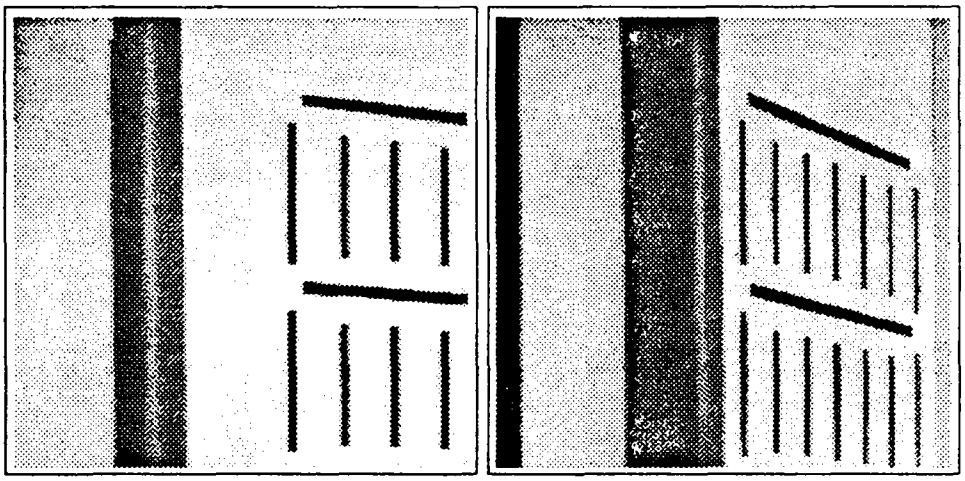


Figure 11 : (a) Première image de la séquence "mire"; taille : 256 × 256. (b) Dernière image de la séquence "mire".

La figure (12) montre en gras le segment sélectionné initialement. Ici la TH n'a été lancée que sur la partie droite de l'image (si l'on considère toute l'image un des segments du cylindre est évidemment sélectionné).

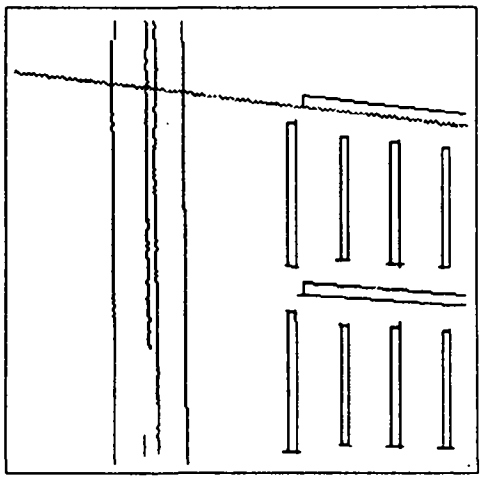


Figure 12 : Droite support du segment sélectionné dans la première image.

Comme dans le cas précédent, le résultat du suivi du segment sélectionné dans la séquence est montré sur la figure (13). Ici aussi notre algorithme donne d'excellents résultats. De même, on peut relever la robustesse de notre algorithme qui n'a eu aucune difficulté à distinguer le bon segment de celui situé juste au dessus (voir la figure (13)).

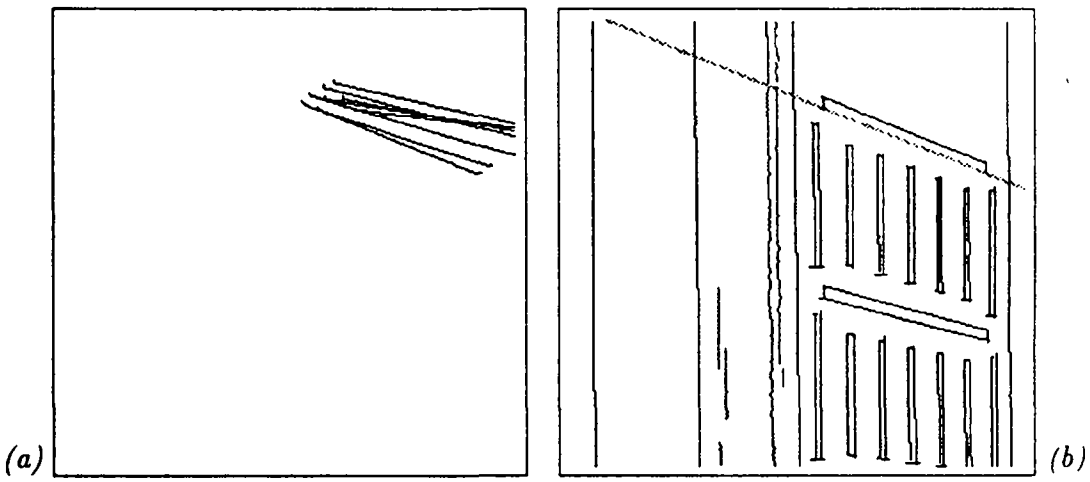


Figure 13 : (a) Séquence de segments correspondant au segment sélectionné. (b) Position finale du segment estimé en accord avec le segment extrait de cette image.

Les temps de calcul sur Sympati2 sont du même ordre de grandeur que pour les deux séquences précédentes pour la même amplitude maximale des déplacements recherchés. En effet, une itération de suivi du segment sélectionné est de l'ordre de 47 ms (33 ms dans le cas d'une amplitude maximale de ± 2). C'est encore une très bonne performance tout à fait compatible avec les exigences de notre approche de reconstruction 3D par

vision active.

Notons qu'une version simplifiée de l'algorithme d'appariement de segments que nous avons présenté a aussi été porté sur la carte EDIXIA-1000 dont dispose l'IRISA. Les performances préliminaires obtenues sont de l'ordre de 2 s pour la phase de focalisation et de 100 ms pour la phase de suivi.

5 Conclusion

Nous avons présenté dans ce rapport une approche locale pour la mise en correspondance de segments dans une séquence d'images, opération indispensable pour la reconstruction 3D de cylindres qui est l'objectif final de cette étude. Nous avons en effet déjà mis au point une méthode permettant de recouvrer la structure 3D de primitives cylindriques par vision active. Des mouvements adéquats de la caméra, engendrés par des lois de commande en boucle fermée sur les informations visuelles, permettent d'optimiser la reconstruction 3D des cylindres. Cette approche n'est évidemment réalisable en pratique que si des contraintes temps réel au niveau de la phase d'analyse d'images sont respectées. Il fallait donc, outre les aspects de robustesse et de fiabilité, prendre en compte l'aspect temps réel dans le choix d'une méthodologie appropriée pour l'appariement des limbes des cylindres dans la séquence d'images.

Nous avons établi une méthode originale parfaitement adaptée aux besoins de notre application tant en ce qui concerne les implications algorithmiques que les aspects de robustesse. Cette méthode comprend un noyau basé sur une modélisation spatio-temporelle d'un élément de contour en mouvement (ECM) dont une version améliorée a été conçue. Un test robuste de vraisemblance permet de détecter et de suivre les ECM. La mise en œuvre effective de ce noyau est équivalente à un simple détecteur de contour spatial de type gradient-convolution. Les autres étapes de la méthode ne demandent que très peu de calcul. Une machine parallèle particulièrement adaptée à ce type d'opérations est la machine SYMPATI2, une carte multi-processeurs de type SIMD, adaptée au traitement d'images. Notre algorithme de suivi de segments a donc été porté sur cette machine et a atteint des performances de calcul qui répondaient tout à fait aux exigences de notre approche de reconstruction 3D en boucle fermée. Outre ses performances temps réel, cet algorithme s'est avéré très efficace lors de sa validation sur plusieurs séquences d'images.

Notre technique, dont les motivations initiales étaient des applications en environnement nucléaire, est évidemment utilisable pour d'autres types de scènes. La méthode peut également être assez directement adaptée à d'autres primitives que des segments de droite. En effet, puisqu'elle ne manipule explicitement que des chaînes de contour, d'autres courbes paramétrées comme des arcs de cercle ou d'ellipses peuvent être considérées. Un aspect novateur de cette approche est qu'elle exploite réellement et effica-

cement le fait que toute problématique vision n'est pas une fin en soi mais est couplée à une action ou à une décision. De plus, ceci est mis en forme de façon bien posée et relativement générale.

Remerciements:
- Cette étude a reçu le soutien du PRC Communications Homme-Machine / pôle Vision (contrat MRT 91S0269) et du CEA (convention d'étude n^o SAV16849/VCH).
- Les auteurs remercient M.Viala et J.P.Guérin, notamment pour la version de la transformée de Hough portée sur Sympati2.

Annexe : Le système multiprocesseurs Sympati2

6 Introduction

SYMPATI2 est un processeur ligne. La structure de base est un anneau constitué de m processeurs élémentaires (PEs) et m bancs mémoire, avec m pouvant varier en puissance de 2, de 16 à 256 PEs. L'ensemble PEs et bancs mémoire fonctionnent en mode SIMD (Single Instruction Multiple Data) sous le contrôle d'une unité de contrôle programmable. Le mode SIMD signifie que tous les processeurs élémentaires effectuent, de façon synchrone, les mêmes opérations sur des données différentes.

7 Architecture de SYMPATI2

SYMPATI2 est composé de deux blocs : une unité de commande (UC) et un bloc de calcul (16 à 256 processeurs élémentaires). Le rôle de l'UC est d'alimenter en instructions les PEs et d'assurer la communication entre le calculateur hôte et le processeur ligne. Le bloc calcul est formé de l'ensemble des PEs et de leurs bancs mémoire associés. Le PE se décompose en trois parties : une partie traitement, une partie adressage et la partie interconnexion [13], [2].

- la partie traitement est constituée d'une unité arithmétique et logique (ALU) de 16 bits, d'un multiplicateur de 8 bits, d'un module de décalage et d'un ensemble de registres de travail.

- la partie adressage permet deux types d'adressage : un adressage hélicoïdal de la mémoire, où une adresse commune est envoyée à tous les PEs par l'UC. Un adressage tabulaire où chaque PE accède à un des bancs mémoire à partir d'une adresse engendrée par la partie traitement.

- la partie interconnexion permet l'accès des PEs à un grand voisinage de l'image.

Notons que le segment de PEs peut être choisi horizontal ou vertical. Dans les deux cas, deux types de balayage peuvent être utilisés, le balayage linéaire et le balayage par bandes [1].

7.1 Arrangement hélicoïdal des données

La matrice de données est distribuée de manière hélicoïdale dans les bancs mémoire des PEs pour que les points d'un segment ligne ou colonne se trouvent dans des bancs mémoire distincts, à la même adresse. Ceci, afin d'éviter les conflits d'accès de m points consécutifs à une ligne ou une colonne de la matrice (m étant le nombre de PEs). La figure (14) illustre un exemple d'arrangement hélicoïdal des données d'une image (8*8) dans 4 PEs (le segment est alors formé de 4 pixels).

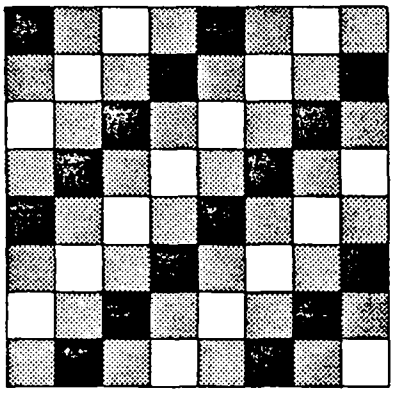


Figure 14 : Arrangement hélicoidal des données

Chaque grisé représente le banc mémoire d'un PE.

Parfois, ce type d'arrangement n'est pas parfaitement adapté notamment à certains algorithmes de traitement d'images de moyen ou de haut niveau. SYMPATI2 offre donc un autre mode d'arrangement des données : le mode tabulaire dans lequel la matrice de données est divisée en bandes (voir figure 15).

Avec une machine composée de 32 PEs, la mémoire peut contenir une image 1024*1024 pixels. On obtient 32 bandes de 32 pixels. Chaque bande est divisée en 32 sous-bandes, soit une sous-bande par PE (1*1024).

7.2 Interconnexion entre les PEs

La partie interconnexion est constituée d'un ensemble de chemins de données permettant, d'une par,t aux différents PEs de communiquer entre eux, et, d'autre part, de transférer les données entre les différents modules (bancs mémoires, PEs et UC). On distingue donc deux types de transfert :

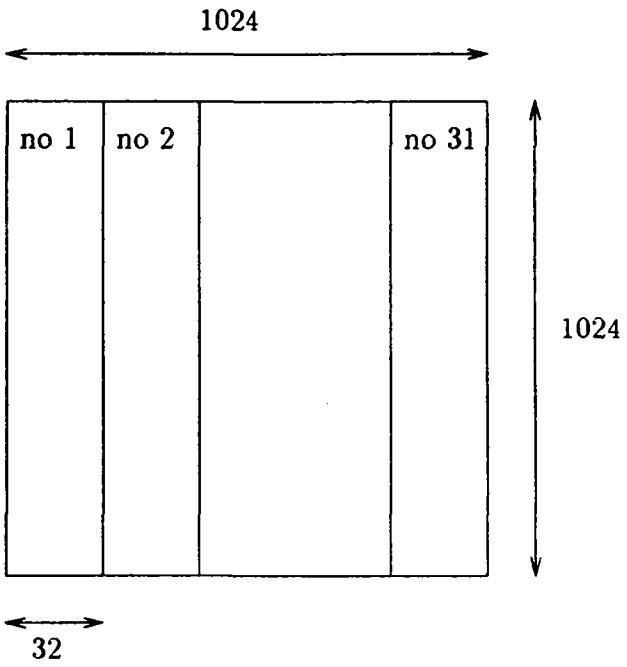


Figure 15 : Découpage en bandes de l'image avec 32 PEs

- transferts entre les PEs : chaque PE peut effectuer des échanges (chargement de registres) avec les PEs de gauche et de droite à une distance maximale de trois.
- transferts entre les PEs et les bancs mémoires : chaque PE peut lire dans les bancs mémoire des PEs de gauche et de droite à une distance maximale de 2.

8 Environnement de programmation

L'environnement de programmation est constitué d'un langage parallèle. le langage 4LP (Low Level Language for Line Processor), de son compilateur et d'un debugger. Le langage 4LP est un langage conçu pour une structure SIMD, dont le parallélisme est totalement transparent pour le programmeur. La philosophie du langage consiste à focaliser le traitement sur un pixel de l'image. Ensuite, le travail est répété pour l'ensemble des pixels de l'image. Pour plus de précisions sur ce langage le lecteur est prié de se reférer au manuel d'utilisation de 4LP [1].

8.1 Application au traitement d'images

Les traitements privilégiés sur la machine SYMPATI2 sont les traitements d'images de bas et moyen niveaux, donc l'ensemble des opérations qui s'effectuent au niveau pixel. Les traitements de moyen niveau correspondent aux algorithmes qui extraient des informations de l'image (contours, textures,...). Par contre la machine est beaucoup

moins performante pour des algorithmes de haut niveau, qui analysent des informations extraites du bas niveau, car ils sont souvent séquentiels.

Bibliographie

[1] P.Adam : *Manuel d'utilisation du langage assembleur de la machine Sympati2*, Rapport Interne CEA, n. 88.34, 1988.

[2] J.L.Basille, H.Essafi, D.Juvin, J.Y.Latil : *SYMPATI II : A 1.5D Processor Array for Image Application*, Proc. 4th Conf. EUSIPCO, Grenoble, 1988.

[3] S. Boukir, F.Chaumette : *Reconstruction 3D de primitives géométriques paramétrables par vision active*, Publication Interne IRISA, N. 662, Juin 1992.

[4] P. Bouthemy : *Estimation et structuration d'indices spatio-temporels pour l'analyse du mouvement dans une séquence d'images*, Traitement du Signal Vol 4, N. 3, pp. 239-257, 1987.

[5] P. Bouthemy : *A Maximum Likelihood Framework for Determining Moving Edges*, IEEE Trans. on PAMI, Vol 11, N. 5, pp. 499-511, Mai 1989.

[6] J.F.Canny : *A computational Approach to Edge Detection*, IEEE Trans on PAMI, 8(6), pp.679-698, 1986.

[7] F.Chaumette, S.Boukir : *Structure from motion using an active vision paradigm*, 11th IAPR Int. Conf. on Pattern Recognition, The Hague, The Netherlands, Vol 1, pp. 41-44, Septembre 1992.

[8] H.H.Chen, T.S.Huang : *Matching 3-D line segments with applications to multiple-object motion estimation*, IEEE Trans. on PAMI, Vol. 12, N. 10, pp. 1002-1008, Octobre 1990.

[9] J.L.Crowley, P.Stelmaszyk, P.Puget : *Measurement and integration of 3-D structures by tracking edge lines*, Int. journal of Computer Vision, Vol. 8, N. 1, pp. 29-52.

[10] R.Deriche : *Using Canny's criteria to derive an optimal edge detector recursively implanted*, International Journal of Computer Vision, N. 2, pp. 167-187, Avril 1987.

[11] R.Deriche, O.Faugeras : *Tracking line segments*, Image and Vision Computing Journal, Vol. 8, N. 4, pp. 261-270, Novembre 1990.

[12] R.O.Duda, P.E.Hart : *Use of the Hough transformation to detect lines and curves in pictures*, Communication of the ACM, 15, N. 1, pp. 11-15, Janvier 1972.

[13] H.Essafi : *Les processeurs ligne en traitement d'image*, Thèse de 3ième cycle, Univ. Paul Sabatier de Toulouse, N. 307, Juin 1988.

[14] H.Maitre : *Un panorama de la transformée de Hough*, Traitement du Signal, Vol 2, N. 4, pp. 306-317, 1985.

[15] L.Marcé, P.Bouthemy : *Determination of a depth map from an image sequence*, Proc. of the 3rd Int. Conf. on Advanced Robotics ICAR, pp 221-232, Versailles, France, Octobre 1987.

[16] G.Medioni, R.Nevatia : *Matching images using linear features.* IEEE Trans. on PAMI, Vol. 6, N. 6, pp. 675-685, Novembre 1984.

[17] J.Shen, S.Castan : *An optimal linear operator for step edge detection*, CVGIP : Graphical models and Image Processing, Vol 54, N. 2, pp. 13-17, Mars 1990.

[18] 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 674 A CALCULUS OF GAMMA PROGRAMS
Chris HANKIN, Daniel LE METAYER, David SANDS
Juillet 1992, 32 pages.

PI 675 EVALUATION DES PERFORMANCES D'UN NOYAU DE SIMULATION REPARTIE
Philippe INGELS, Carlos MAZIERO
Septembre 1992, 36 pages.

PI 676 FONT METRICS
Jacques ANDRE
Septembre 1992, 20 pages.

PI 677 GRIF ET LES INDEX ELECTRONIQUES
Hélène RICHY
Septembre 1992, 40 pages.

PI 678 ETUDE DE QUELQUES ORGANISATIONS D'ANTEMEMOIRES
Nathalie DRACH, André SEZNEC
Octobre 1992, 44 pages.

PI 679 AN ADAPTIVE SPARSE UNSYMMETRIC LINEAR SYSTEM SOLVER
Miloud SADKANE, Roger B. SIDJE
Octobre 1992, 28 pages.

PI 680 BRANCHING BISIMULATION FOR CONTEXT-FREE PROCESSES
Didier CAUCAL, Dung HUYNH, Lu TIAN
Octobre 1992, 36 pages.

PI 681 DEADLOCK MODELS AND GENERAL ALGORITHM FOR DISTRIBUTED DEADLOCK DETECTION
Jerzy BRZEZINSKI, Jean-Michel HELARY, Michel RAYNAL
Octobre 1992, 26 pages.

PI 682 LE MODELE DE MARKOV CACHE (MMC) A DEUX NIVEAUX : APPLICATION A LA MODELISATION DE LA DUREE DANS UN DISPOSITIF DE RECONNAISSANCE DE PAROLE
Nelly SUAUDEAU
Octobre 1992, 94 pages.

PI 683 TARGET TRACKING BY VISUAL SERVOING
Aristide S. SANTOS, François CHAUMETTE
Octobre 1992, 50 pages.

PI 684 UNE DESCRIPTION LINEAIRE COMPLETE ET IRREDONDANTE DU POLYTOPE ASSOCIE AU PROBLEME DU VOYAGEUR DE COMMERCE ASYMETRIQUE A 6 SOMMETS
Reinhardt EULER, Hervé LE VERGE
Octobre 1992, 30 pages.

PI 685 MISE EN CORRESPONDANCE DE SEGMENTS DANS UNE SEQUENCE D'IMAGES PAR UNE APPROCHE LOCALE
Samia BOUKIR, Patrick BOUTHEMY, François CHAUMETTE, Didier JUVIN
Octobre 1992, 30 pages.

PI 686 FROM EQUATIONS TO HARDWARE. TOWARDS THE SYSTEMATIC MAPPING OF ALGORITHMS ONTO PARALLEL ARCHITECTURES
François CHAROT, Patrice FRISON, Eric GAUTRIN, Dominique LAVENIER, Patrice QUINTON, Charles WAGNER
Octobre 1992, 18 pages.

ISSN 0249-6399