

SYSTEME DE VISION STEREOSCOPIQUE EN TEMPS REEL D'EXTRACTION DE SEGMENTS

F. DEVILLARD, G. BOUVIER et A. CHEHIKIAN

Laboratoire de Traitement d'Images et Reconnaissance de Formes
I.N.P.G. 46 Av. Félix Viallet 38031 Grenoble Cédex

RÉSUMÉ

Nous proposons un ensemble de vision stéréoscopique embarqué, destiné à la navigation d'un robot mobile. Ce système, basé sur l'extraction de segments, doit être robuste même en environnement perturbé, rapide conformément à la vitesse de déplacement du robot et de faible volume. Toutes ces contraintes, liées à la robotique mobile ne peuvent être respectées que par le développement d'algorithmes et d'architectures spécifiques et parfaitement adaptées aux traitements mis en œuvre.

Nous présentons une réalisation et les résultats obtenus d'une étude faite dans le cadre EUREKA EU110 (MITHRA) de robot mobile.

ABSTRACT

We propose a stereovision package that goes on a mobile robot. This system, dedicated for robot navigation and based on segments extraction, must be robust especially in a perturbed environment. The treatment must be rapid in relation with the speed at which the robot is moving. Further more it must have a reduced volume. All these constraints, emaning from mobile robotics, can only be met in the conjoint development of specific hardware architectures and the adapted software algorithms.

We present the design and results obtained by a study made within the context of the european project EUREKA EU110 (MITHRA) of mobile robot.

1. Introduction

1. Le projet

Le but du projet MITHRA est de réaliser un démonstrateur de robot mobile pourvu de fonctionnalités multiples. On lui confiera des missions de surveillance et d'intervention dans le milieu industriel.

2. La fonction du capteur vision

MITHRA évolue dans un milieu structuré (locaux industriels). Les indices visuels extraits en stéréo doivent permettre une localisation du robot dans son environnement et ainsi pouvoir contrôler sa trajectoire lors de déplacement à vitesse élevée (2 mètres par seconde).

Les performances de ce système doivent être conformes aux contraintes sévères qu'impose la robotique autonome mobile.

- L'insensibilité aux variations de luminance.
- La rapidité de traitement.
- La faible consommation.
- Le volume réduit.

3. Les indices visuels

Le choix du type d'indices visuels à traiter est primordial, il faut choisir les primitives nécessaires et suffisantes à la reconstruction tridimensionnelle (3D).

Afin de répondre au mieux aux contraintes de vitesse, nous nous sommes attachés à extraire des images les informations les plus significatives d'un point de vue

topologique. Dans le cas de sites industriels, l'ensemble des objets présentent des géométries orthogonales telles que les intersections de cloisons, portes, fenêtres, mobiliers, ...

Une recherche des contours verticaux ou horizontaux montre après expérience qu'elles permettent une définition suffisante d'un environnement tout en réduisant la redondance des informations visuelles dans des proportions satisfaisantes.

2. Traitements

1. L'extraction de contours

Les techniques d'extraction de contours sont très diversifiées, nous recherchons une méthode permettant une extraction fidèle, efficace et aisément intégrable matériellement en temps réel. Notre choix s'est porté sur une variante de l'algorithme préconisé par CANNY [4].

Le traitement se compose en plusieurs phases successives.

Le lissage:

Cette étape est réalisée par un filtre itératif de type FIR gaussien; il permet de réduire la bande de fréquence de l'image traitée. Le noyau de convolution élémentaire est de la forme :

$$FIRg = 1/16 \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

Ce filtre est caractérisé par sa variance en fréquence égale à $0.5/N$, N étant le nombre d'itérations.

Le calcul des gradients et des codes contour:

A partir du résultat précédent, nous calculons les composantes horizontale et verticale respectivement G_x et G_y du gradient par l'application du masque de convolution $[1 \ 0 \ -1]$.

Puis à partir de G_x et G_y , nous générons le code polaire du gradient utilisé par la suite pour la détection des contours. Il comporte deux composantes :

-G, le module du gradient donné par:

$$\text{Si } |G_x| \geq |G_y| \text{ alors } G = 1/16 \sqrt{G_x^2 + G_y^2} \text{ sinon } G = 0$$

$-\theta$ quantifie l'angle d'orientation du gradient sur un secteur angulaire situé à $\pm 45^\circ$ autour de l'axe horizontal x [fig 1], avec une précision de $\pi/14$ radian soit environ 12.9° . Pour les autres directions le code est nul.

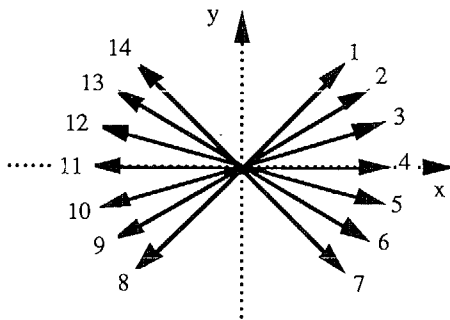


Fig 1 : Le codage de direction de θ .

Toutes les opérations citées jusqu'à présent sont intégrées dans une seule unité de traitements.

La détection de contours :

La détection de contours dans une image de gradient se fait par la recherche du passage à un maximum local du module de gradient.

Le critère de maximum local : le contour est caractérisé par un maximum du gradient dans l'axe d'orientation de ce dernier. Dans notre cas, nous ne gardons que les maxima sur x, expérimentalement cela a pour conséquence que deux pixels ne pourront être contours et voisins sur une même ligne.

Le critère d'intensité : les contours des objets sont définis par des gradients importants de luminance. Chaque pixel sera classé par son intensité comme appartenant ou non à un contour objet selon la valeur du pseudo-gradient $G_m = |G_x| + |G_y|$ par rapport à une frontière haute ThH et une basse ThL .

Les classes sont établies par :

PC : Si $G_m > ThH$ alors *Pixel Contour*.

PPC : Si $G_m > ThL$ et $< ThH$ alors *Pixel Contour Probable*.

PNC : Si $G_m < ThL$ alors *Pixel Non Contour*.

Les classes sont codées sur deux drapeaux ajoutés au code polaire du gradient associé au pixel contour.

Traitement de la classe PPC :

Ce traitement est la dernière étape de la détection de contours. Elle permet une amélioration notable du rapport signal sur bruit sur le résultat final. Il suffit d'observer le voisinage des pixels contours probables : si un de ses voisins

est contour, nous en déduisons qu'il appartient lui aussi à ce contour.

Le traitement de la classe PPC se fait par deux balayages successifs des contours de gauche à droite et du haut vers le bas puis de droite à gauche et de bas vers le haut.

Nous étudions le passé immédiat [Fig 2] de chaque pixel PPC. Si leur passé possède au moins un pixel de type PC alors nous levons l'ambiguïté sur ces pixels en les déclassant, ceci est réalisé par leur passage en type PC. Nous exécutons ainsi une détection des contours flous par étude du voisinage passé.

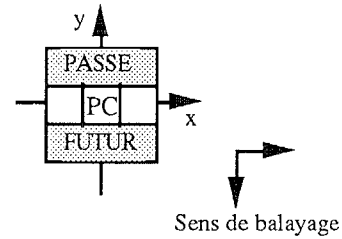


Fig 2 : Masque de connexité.

2. la segmentationLe chaînage :

Le traitement d'image contour est encore trop lourd, il est impératif de réduire l'information par une transformation supplémentaire. Le segment est information économique et puissante de transcription des géométries. Notre choix d'algorithme s'est porté sur une variante de la solution proposée par GIRAUDON [1].

L'algorithme de chaînage proposé par GIRAUDON est proposé dans un contexte de contours omnidirectionnels. Nos contours orientés autour de la verticale et d'épaisseur d'un seul pixel créent des chaînes de forme moins complexes à traiter.

Nous opérons de la même façon. Nous codons nos chaînes avec les coordonnées des extrémités de chaînes et lors d'un seul balayage de l'image nous suivons parallèlement l'évolution des contours.

Nous analysons chaque pixel contour PC avec le masque de connexité [Fig 2]. S'il est contour nous traitons les chaînes en fonction du voisinage détecté par le masque.

Voici les trois types d'événements rencontrés, dont la méthode de traitement est expliquée par DISCOURS [5] :

- La création de chaîne.
- L'actualisation de chaîne.
- La fermeture de chaîne.

La création de chaîne a lieu lorsque le pixel ne possède pas de passé, la fermeture quand un pixel est sans futur, l'actualisation est le cas général ou le pixel se trouve encadré dans le passé et son futur.

A ces traitements se greffent la résolution des problèmes de jonctions de chaînes, traitées par une décomposition en autant de chaînes qu'il y a de branches.

L'approximation polygonale :

Nous ne pouvons pas nous contenter du chaînage et prendre les extrémités de la chaîne pour définir un segment équivalent sous peine d'introduire des distorsions géométriques

importantes. Il est impératif lors du suivi des contours de traiter les courbures par approximation polygonale.

Pour cela nous disposons d'une information d'orientation du contour qui est la valeur du code associé [§ 2.1] qui nous fournit l'orientation du gradient local. C'est la variation excessive de ce code d'orientation au cours du chaînage qui nous permettra d'interrompre la chaîne courante par fermeture et d'en réouvrir une autre par création.

A l'ouverture de la chaîne nous effectuons un calcul sur les deux premiers pixels du code de direction probable pour la suite de celle-ci. θ_p est de la forme :

$$\theta_p = \text{INT}[(\theta_1 + \theta_2)/2 + 0.5]$$

Pendant le suivi du contour, l'approximation polygonale est réalisée par comparaison du code θ_p avec le code de direction du contour courant θ_c .

La fermeture de segment est déclenchée par une variation anormale du code engendrée par un changement de pente [Fig 3]. La condition pour qu'un pixel contour de code direction θ_c appartienne au segment courant est :

$$\Delta\theta = |\theta_c - \theta_p| \leq 1$$

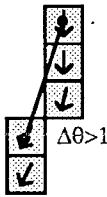


Fig 3 : Approximation polygonale par segments orientés.

3. Système de vision

1. le système vision du robot

L'ensemble comporte 6 modules [Fig 4].

OEC, l'Opérateur acquiert les images et en extrait les contours.

CS, la carte de segmentation gère les tâches de OEC.

CPP, la Carte de Poursuite de Primitives corrèle les listes de segments en temps et en espace. CPP calcule les trajectoires individuelles de chaque segment en stéréoscopie.

CP, la carte Perception élabore le modèle 3D de l'environnement.

CM, la Carte Modélisation, elle est maîtresse du bus VME et de l'Interface Ethernet (IE) permettant la communication avec les autres éléments du robot. Elle s'occupe du recoupement ou la fusion de la vision avec les autres dispositifs de perception.

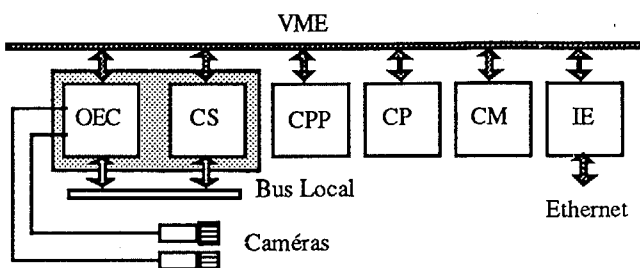


Fig 4 : Le système de vision du robot MITHRA

2. Le système d'extraction de segments

Le système d'extraction de segments comporte 2 cartes notées OEC et CS [Fig 5].

L'Opérateur d'Extraction de Contours : (OEC)

L'Opérateur d'Extraction de Contours acquiert les 2 images qu'il numérise et stocke en mémoire (Md et Mg, 2 plans de 256x256 8 bits), le contenu est traité par une Unité de Traitement (UT) et transféré par le bus local dans les mémoires image (Mi) de la carte de segmentation (CS).

Le tout est géré par une Unité de Contrôle (UC) et une Unité d'Adressage (UA).

L'extraction de contours est traitée en trois passes successives de 6.5 milli-secondes chacune.

Les gradients et les contours sont calculés pendant les deux premières passes, la troisième effectue la dernière passe de détection de contours et le stockage dans les mémoires de la carte de segmentation.

La Carte de Segmentation : (CS)

Cette carte est interfacée (I) sur le bus VME et communique, sous les directives de la Carte Maître (CM), avec la Carte de Poursuite de Primitives (CPP) par une mémoire double port (Mdp). Le DSP (Digital Signal Processor) par interruptions de l'OEC gère la fréquence d'acquisition et de traitement, une fois le résultat disponible dans Mi, il réalise la segmentation. La partie logiciel est écrite en assembleur optimisé.

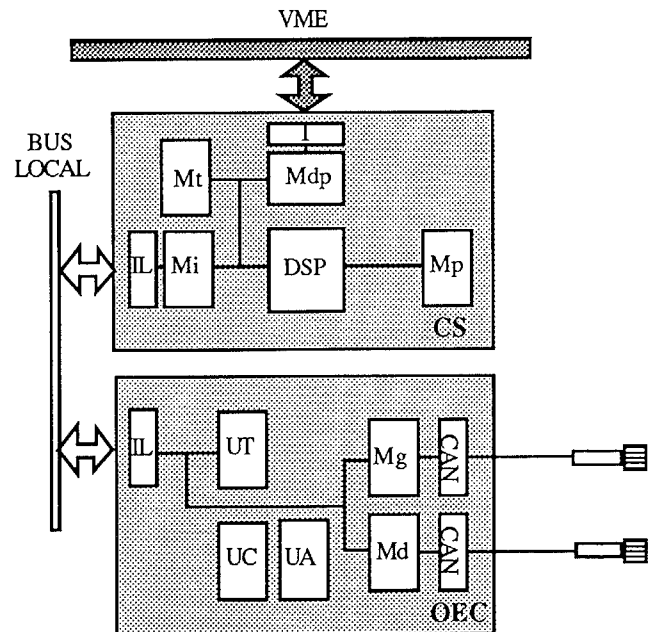
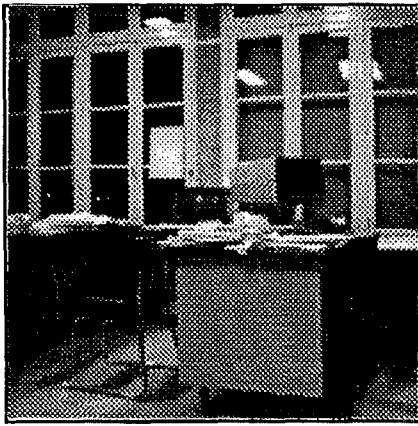


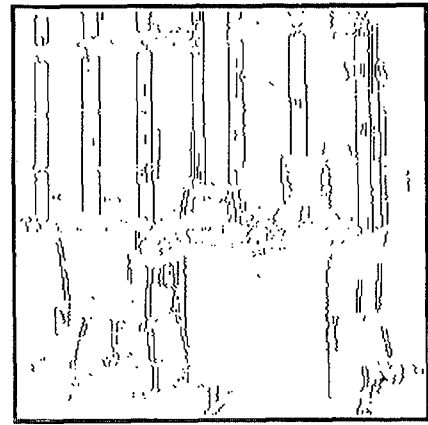
Fig 5 : Acquisition, extraction de contours et segmentation du robot MITHRA.

La puissance du DSP dans ce type d'activités réside dans la nature itérative de l'algorithme de segmentation qui lui permet de fonctionner sur sa mémoire cache. Nous parvenons à lui faire segmenter une image complète en approximativement 35 milli-secondes (12.5Mips). Ce temps d'exécution est une moyenne car il dépend du nombre de pixels contour traités et des fréquences d'appels des différents types de routines.



a : Image originale.

Fig 6 : Exemple "Le bureau", image de l'INRIA.



b : Contours à orientation verticale.ThL=1 ThH=3.

4. Résultats expérimentaux

Nous présentons quelques résultats de traitement obtenus sur l'image fournie par l'INRIA [Fig 6] qui fait référence de test en traitement d'image. Cette image peut être un exemple de scène que rencontrera le robot MITHRA lors de ses déplacements. L'extraction de segments verticaux emploie une méthode qui a été facilement étendue aux horizontales.

5. Conclusions

Nous avons conçu et réalisé un système d'extraction de primitives du robot mobile regroupant les performances ci-dessous.

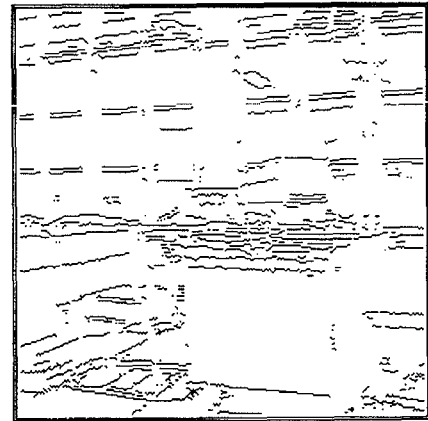
L'ensemble est intégré sur 2 cartes de faible consommation (technologie CMOS) et fonctionne dans une plage importante d'éclairage par la possibilité du contrôle externe des caméras.

Il fonctionne à une cadence de 33 images par seconde dans sa version la plus performante. Nous échantillons et traitons 2 images toutes les 60 milli-secondes (3 trames), ce qui représente un recalage du robot tous les 12 centimètres lorsqu'il se déplace à vitesse maximale.

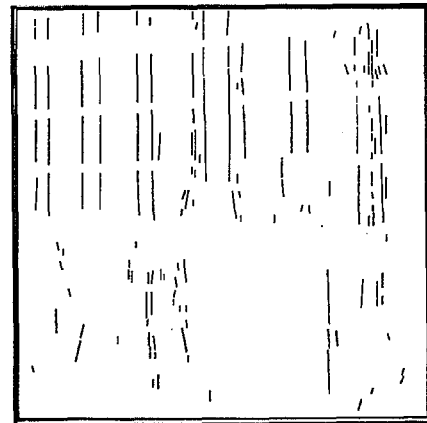
Cette efficacité n'est possible que par un traitement parallèle de l'extraction de contours et de la segmentation. Ces différentes phases réclamaient des algorithmes aussi performants qu'adaptés aux contraintes d'utilisation du matériel.

6. Références

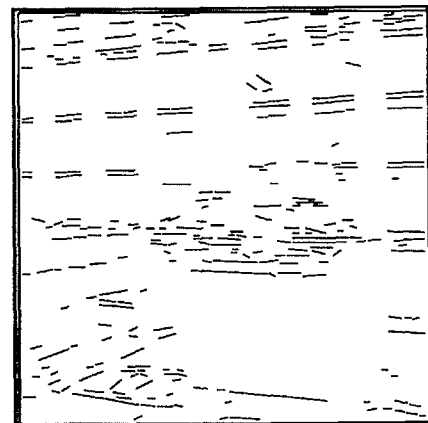
- [1] Gérard Giraudon, "Chaînage Efficace de Contour" Technical Report 605, INRIA, fév. 1987.
- [2] Indranil Chakravarty, "A single-pass, chain generating algorithm for region boundaries", *Computer Graphics and Image Processing*, vol. 15, 1981, pp. 182-193.
- [3] Augustin Lux, "Algorithmique et Contrôle en Vision par Ordinateur", Thèse d'état USMG/INPG, sept. 1985.
- [4] Canny J. F., "A computational approach to edge detection", *IEEE Trans. Patt. and Mach. Intell.* PAMI-8, 1986, pp. 679-698.
- [5] Christophe Discours, "Analyse du mouvement par mise en correspondance d'indices visuels", Thèse INPG, nov 1989.
- [6] Monga O., Wrobel-Dautcourt B., Berthod M., Derich R., Giraudon G. et Shen J., "Segmentation & robotique mobile", Premières journées nationales du GRECO-PRC. Communication Homme-Machine, nov 1988.



c : Contours à orientation horizontale.ThL=1 ThH=3.



d : Segments à orientation verticale.ThL=1 ThH=3 Lmin=5.



e : Segments à orientation horizontale.ThL=1 ThH=3 Lmin=5.