ALGORITHME D'APPROXIMATION POLYGONALE DEDIE A LA ROBOTIQUE MOBILE

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 Cedex - France

RÉSUMÉ

En robotique mobile, les systèmes de vision embarqués sont sujets à de sévères contraintes de fonctionnement (traitement en temps réel, volume, consommation...). Pour la modélisation 3D de l'environnement, les systèmes de vision doivent utiliser des indices visuels permettant un codage compact, précis et robuste de la scène observée.

En réponse à ce type de problème, nous proposons une méthode d'approximation polygonale permettant d'optimiser la description des contours d'objet.

ABSTRACT

In mobile robotics, the on-board artificial visual systems are exposed to some severe working constraints (real time processing, volume, consumption...). For 3D modelling of the environment, the visions machines must use the visual features allowing a compact, precise and robust coding of the scene.

In order to solve this kind of problems, we propose a method of polygonal approximation optimizing the description of the edges of objects.

1. Introduction

1. Problème

Les algorithmes classiques d'approximation polygonale présentent parfois des résultats qui rendent l'implémentation temps réel d'une modélisation 3D impossible. Bien que performants ces algorithmes ont tendance à fournir des listes de primitives trop importantes.

2. Notre approche

Dans la plupart des applications de la robotique, les robots évoluent à l'intérieur de bâtiments. Les scènes observées présentent une majorité d'objets possédant des géométries orthogonales (portes, intersections de cloisons, mobiliers...). Dans ce cas, l'extraction des contours horizontaux et verticaux montre, après expérience, qu'ils permettent une définition suffisante de l'environnement tout en réduisant les informations visuelles dans des proportions satisfaisantes. Ce genre d'idée a déjà été employée pour réaliser des machines de vision ([KRIEGMAN89] et [CROWLEY90]). Ces machines effectuaient la modélisation 3D d'une scène à partir des segments de droite décrivant les contours verticaux des objets observés.

Nous proposons un algorithme d'approximation polygonale traitant les contours orientés proches de la verticale ou de l'horizontale. Nous décrivons auparavant la méthode employée pour extraire les contours.

2. Détection de contours

La qualité de description des contours d'un objet est tributaire de la méthode de détection employée. Les filtres, à caractéristiques fréquentielles variables, présentent d'excellentes performances même appliqués sur des images bruitées ou de faible contraste. Des solutions ont été proposées par Canny [CANNY86], Deriche [DERICHE87], Marr [MARR79] et Shen [SHEN85].

Notre choix s'est orienté vers l'algorithme qui autorise une implémentation aisée en temps réel et qui nous fournit une information sur l'orientation du contour. La méthode de Canny répond à ces contraintes. Le gradient calculé fournit l'orientation locale du contour et le filtrage FIR utilisé peut-être implémenté simplement et en un minimum de convolutions.

1. Calcul du gradient

On utilise pour le calcul du gradient une variante du filtre optimal de Canny. On applique un filtre quasi-gaussien de lissage en prétraitement sur l'image originale. On dérive l'image résultat en appliquant un filtre passe-haut horizontalement et verticalement afin d'obtenir respectivement les composantes Gx et Gy du gradient.

Les masques de convolutions élémentaires utilisés sont monodimensionnels : b est un filtre binomial approchant un filtre passe-bas gaussien de variance 0.5 et d est le filtre passe-haut de dérivation.

$$b(x) = \frac{1}{4}[1 \quad 2 \quad 1] \qquad d(x) = [1 \quad 0 \quad -1]$$

Les composantes du gradient sont calculées à partir de ces filtres par les convolutions suivantes :

$$G_x(x, y) = b(x)^{\otimes (n-1)} \otimes d(x) \otimes b(y)^{\otimes n} \otimes I(x, y)$$

$$G_y(x, y) = b(y)^{\otimes (n-1)} \otimes d(y) \otimes b(x)^{\otimes n} \otimes I(x, y)$$

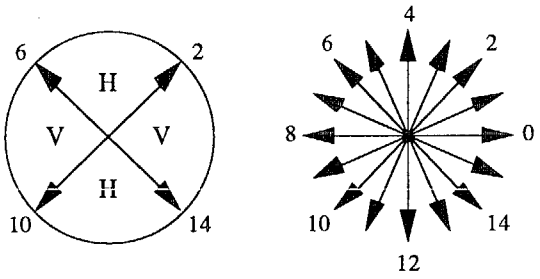
Le coefficient n (n>1) permet d'adapter l'intensité du lissage en fonction du rapport signal sur bruit de l'image I.

2. Sélection et conversion polaire des gradients

Dans un premier temps, on sépare les gradients provenant de contours de directions proches des axes horizontaux et verticaux. Les contours horizontaux et verticaux possèdent des gradients dont les composantes cartésiennes vérifient respectivement :

$$|G_y| > |G_x| \qquad |G_x| > |G_y|$$

La figure 1-a illustre les plages d'orientation du gradient déterminant les contours verticaux (V) et l'horizontaux (H).



[a] Sélection des orientations

[b] Codage 16 orientations

Fig 1 : Discrétisation de l'orientation du gradient.

Ensuite, on transforme l'expression cartésienne du gradient en polaire. L'intensité et l'orientation sont obtenues par les formules ci-dessous :

$$G = \sqrt{G_x^2 + G_y^2} \qquad \theta = \mathrm{Arctg}\left(\frac{G_y}{G_x}\right)$$

Le codage de l'orientation θ du gradient est représenté par la figure 1-b. On obtient donc deux images de gradient Gh et Gv représentant respectivement les contours horizontaux et verticaux.

3. Détection des maxima locaux

Cette opération est appliquée sur chacune des deux images de gradient. Dans le cas général, la recherche de la position exacte du contour est réalisée sur le gradient par la détection du maximum d'intensité dans la direction du gradient.

Dans notre cas, le choix des voisins du pixel traité est simple. Les voisins sont les 2 pixels placés de part et d'autre sur la ligne pour l'image Gv et la colonne pour Gh.

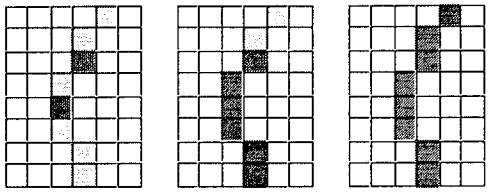
4. Seuillage par hystérésis

Canny utilise le seuil par hystérésis pour améliorer la continuité des contours. Il fait appel à deux valeurs de seuil (basse et haute). Une chaîne de pixels est conservée si tous les pixels possèdent une intensité de gradient supérieure au seuil bas et s'il y a au moins un pixel parmi eux dont l'intensité dépasse le seuil haut.

On impose donc sur les maxima locaux deux seuils Sb et Sh. Les pixels d'intensité inférieure à Sb sont définitivement éliminés. Les pixels d'intensité supérieure à Sh sont attribués contour vrai et enfin ceux qui restent sont attribués contours candidats.

L'hystérésis traite uniquement les pixels candidats. Les candidats possédant un voisin connexe contour vrai le deviennent à leur tour. A la fin de l'hystérésis, les contours attribués vrais constituent l'image de contour finale.

La figure 2 illustre le processus réalisant cette opération sur un contour vertical dans l'image Gv. Le traitement des pixels candidats est effectué en deux balayages de l'image : le premier de gauche à droite et de haut en bas (GDHB) et le second de droite à gauche et de bas en haut (DGBH). Le type du balayage détermine le masque de voisinage utilisé pour tester la présence d'un contour vrai dans le passé [GIRAUDON87]. Le processus appliqué sur un contour horizontal dans l'image Gh utilise les balayages HBGD et BHDG.



[a] Pixels seuillés

[b] Hystérésis GDHB

[c] Hystérésis DGBH

Pixel de contour vrai

Pixel de contour candidat

Fig 2 : Effet de l'hystérésis sur un contour vertical possédant peu de ses points du type contour vrai.

3. Extraction de segments de droite

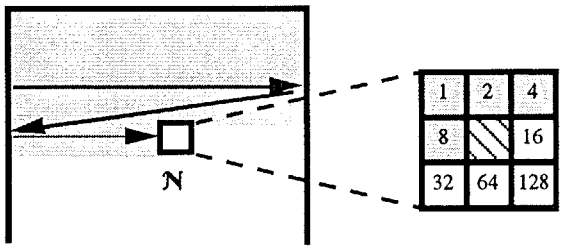
L'extraction de segments de droite appliquée à une image de contour est constituée de deux étapes qui sont le chaînage et l'approximation polygonale.

L'algorithme, décrit ci-dessous, regroupe le chaînage et l'approximation polygonale dans un même processus. Dans un souci d'implémentation temps réel, l'extraction de segments de droite est réalisée sur une image de contour en un seul passage.

1. Le chaînage

Le chaînage consiste à relier tous les pixels de contour connexes entre eux sous forme de chaînes. La coupure d'une chaîne se trouve aux pixels de jonction, c'est-à-dire les pixels se situant à l'intersection de plusieurs chaînes.

Nous procédons comme Giraudon [GIRAUDON87] par un balayage de l'image de contour. Pour chaque pixel de contour, on applique un traitement en fonction du voisinage. La figure 3 montre le balayage appliqué sur les contours verticaux. Le traitement des contours horizontaux utilise le même procédé mais avec l'image au préalablement tournée de 90°.



[a] Balayage de l'image des contours

[b] Masque de voisinage $\mathcal{N}$

Fig 3 : Principe du chaînage par balayage de l'image de contours.

Le problème pour chaque pixel de contour traité est de déterminer la position des pixels de contour connexes. La figure 3-b illustre le principe, on affecte un poids à chaque pixel voisin. Ces poids vont nous permettre de calculer un nombre $\mathcal{N}$ déterminant les connexités. Ce nombre caractérisant le voisinage nous permettra de choisir le traitement de chaînage adéquat à appliquer sur le pixel de contour courant.

Le traitement des contours orientés verticaux nous permet de ne considérer qu'un nombre restreint et sans ambiguïté de voisinage que l'on présente ci-dessous.

• Pixel de chaîne

$\mathcal{N} = 0$
Aucune opération de chaînage car le pixel est isolé.
$\mathcal{N} = 1, 2$ ou 4
Ouvrir une chaîne depuis le pixel.
$\mathcal{N} = 32, 64,$ou 128
Fermer la chaîne aboutissant au pixel.
$\mathcal{N} = 33, 65, 129, 34, 66, 130, 36, 68, 132$
Actualiser la chaîne passant par le pixel.

• Pixel de jonction

$\mathcal{N} = 5, 160, 161, 162, 164, 37, 69, 133$ ou 165
Fermer les chaînes aboutissant au pixel (s'il y a lieu) et ouvrir les chaînes partant du pixel (s'il y a lieu).

2. L'approximation polygonale

L'approximation polygonale consiste à approcher les chaînes par des segments de droite.

Notre algorithme d'approximation polygonale utilise les principes suivants :
- Le segment de droite est décrit par les pixels d'extrémité de la chaîne qu'il représente.
- L'ensemble des pixels de la chaîne approximée par un même segment de droite respecte certains critères.

• Le critère d'orientation :

On part du principe que si un ensemble de pixels de contour connexes appartient à un même segment alors le vecteur de gradient associé à ces pixels possède une orientation identique.

On propose la méthode employée par Rungsunseri qui utilise pour le test de l'orientation du gradient une table de compatibilité de code [RUNSUNGSERI92]. Cette table autorise le chaînage d'un ensemble de pixels connexes si, et seulement si, leurs codes d'orientation ne sont dispersés que sur 3 codes d'orientation adjacents. Le test de compatibilité de code est effectué par la lecture d'une table à deux entrées. La première entrée reçoit le code d'orientation du pixel courant et la seconde entrée reçoit le code moyen de la chaîne. Si le code du pixel courant est compatible, on obtient, en sortie de cette table, le code moyen résultant de la chaîne. Dans le cas contraire, la table retourne -1. Pour n codes d'orientation, la dimension de la table est $3n^2$ soit 768 codes dans notre application avec le codage d'orientation présenté figure 1-b.

Ce critère possède un comportement excellent lorsqu'il est appliqué à des contours polyèdriques. Il est sensible aux fortes courbures dans les chaînes qui caractérisent la position des coins d'objet.

• Le critère de surface :

Le critère de surface utilise l'erreur d'approximation polygonale estimée par la surface comprise entre le segment et la chaîne approximée. Un segment de droite constitue une bonne approximation d'une chaîne si la surface reste inférieure à un seuil prédéterminé. Wall [WALL84] propose une méthode de calcul incrémentale de cette surface illustrée figure 4.

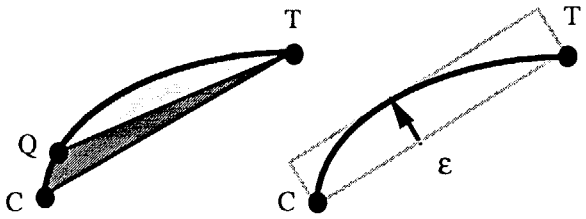


Fig 4 : Calcul incrémental de la surface d'erreur d'approximation polygonale.

TC est la chaîne de pixels à approximer par un segment. Q est le dernier pixel de contour fusionné dans la chaîne. Si on ajoute le pixel C à la chaîne la surface d'erreur A est incrémentée de ΔA qui est l'aire du triangle TQC. Cette aire est calculée par la norme du produit vectoriel suivant :

$$\Delta A = \left\| \frac{1}{2} \left[\vec{QC} \wedge \vec{QT} \right] \right\|$$

Wall montre que la distance orthogonale maximale ε entre la chaîne et le segment ne dépasse jamais 2k pour les chaînes convexes et 4k pour les quelconques dans les conditions énoncées ci-dessous :

$$|A + \Delta A| \leq k \; TC$$

Ce critère est complémentaire de celui de l'orientation. Il est sensible aux faibles courbures dans les chaînes qui caractérisent les portions de chaîne courbes mal traitées par le critère d'orientation.

4. Résultats expérimentaux

L'ensemble des images regroupées sur la figure 5 résume les traitements décrits précédemment.

L'image [a] est l'image de l'INRIA présentant une pièce avec un bureau. Cette image est composée de 256*256 pixels codés sur 8 bits. Les images [b] et [c] sont les images de contour obtenues respectivement à partir des gradients Gv et Gh. Chaque pixel code l'orientation du vecteur gradient associé sur 4 bits. L'image [d] représente les contours par des segments de droite. Cette image est obtenue par le groupement des résultats d'extraction de segments de droite des images [b] et [c]. Le nombre de segments total est de 552.

Ce nombre est réduit par l'utilisation d'un coefficient k choisi plus grand (ex : avec k=5 on n'observe plus que 494 segments). L'introduction d'une longueur minimale Lmin sur les segments extraits permet aussi de réduire le nombre de segments dans la mesure où l'on ne perd pas de contours significatifs (ex : on observe dans cette image 155 segments mesurant 2 et 3 pixels de long).

5. Conclusion

Nous avons présenté un algorithme d'extraction de segments de droite opérant sur deux images complémentaires représentant les contours verticaux et horizontaux d'objet.

La distinction des deux directions nous apporte des avantages considérables :

- L'implémentation de la chaîne de détection de contour est considérablement plus simple.
- L'extraction de segments de droite dans les images de contour est rapide. Le chaînage du fait de la simplicité de la représentation des contours se traite aisément. On s'affranchit notamment de la procédure de fusion de chaînes nécessaire quand on traite une image de contour en un seul balayage [GIRAUDON87].

Les résultats obtenus avec des images structurées prises à l'intérieur de bâtiments sont intéressants. Sur ces images, notre algorithme présente des résultats proches des algorithmes classiques avec un nombre de segments nettement inférieur. A titre indicatif, Rungsunseri [RUNGSUNSERI92] extrait dans des conditions similaires 774 segments.

6. Bibliographie

[CANNY86] J. F. CANNY : "A computational approach to edge detection". IEEE Transactions on Pattern Analysis and Machine Intelligence, vol 8 (6), 1986.

[CROWLEY90] J. CROWLEY, P. BOBET et K. SARACHIK : "Dynamic world modelling using vertical line stereo". Computer Vision, ECCV'90 : Proceedings of the first european conference on computer vision, Antibes, France, Avril 1990.

[DERICHE87] R. DERICHE : "Using Canny's criteria to derive a recursively implemented optimal edge detector". International Journal of Computer Vision, vol 1, n°2, 1987, p 167-187.

[DEVILLARD91] F. DEVILLARD, G BOUVIER et A. CHEHIKIAN : "Système de vision stéréoscopique en temps réel d'extraction de segments". 13ème colloque GRETSI, vol 2, Juan-les-Pins, Septembre 1991.

[GIRAUDON87] G. GIRAUDON "Chaînage efficace de contour". Machines, Réseaux Intelligents. Paris, La Villette, 18-22 mai 1987.

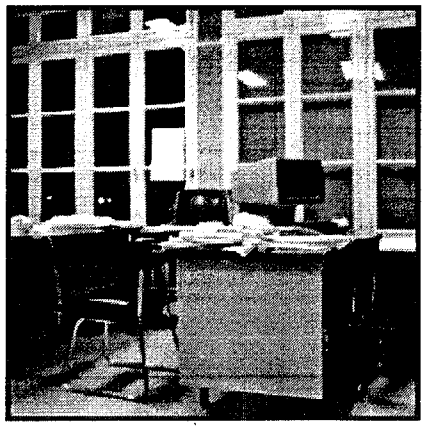
[KRIEGMAN89] D. J. KRIEGMAN, E. TRIENDL et T. O. BINFORD : "Stereo Vision and Navigation in Buildings for Mobile Robots". IEEE Transactions on Robotics and Automation, vol 5 (6) , 1989.

[MARR79] D. MARR et E. HILDRETH : Theory of edge detection. Proceedings of the Royal Society of London, vol. B 207, 1979, p 187-217.

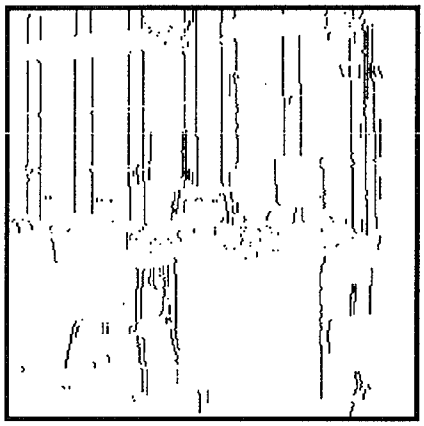
[RUNSUNGSERI92] Y. RUNGSUNSERI : "Système temps réel dédié à la description d'image par segments de droite". Thèse INPG, Novembre 1992.

[SHEN85] J. SHEN, S. CASTAN : "Un nouvel algorithme de détection de contours". AFCET INRIA, 5ème congrès, 1985.

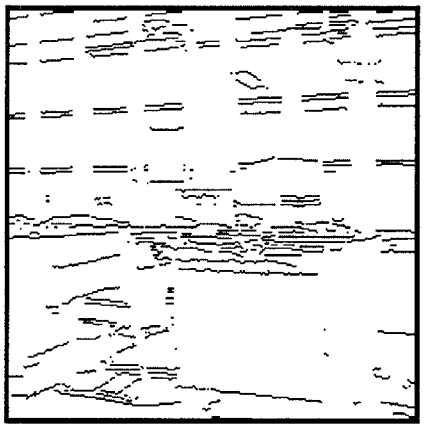
[WALL84] K. WALL et P. E. DANIELSSON : "A fast sequential method for polygonal approximation of digitized curves". CVGIP, vol 28, 1984, p 220-227.



a : Image originale - "BUREAU".



b : Extraction des contours verticaux - Sb=1 Sh=4 n=2.



c : Extraction des contours horizontaux - Sb=1 Sh=4 n=2.

d : Extraction des segments de droite - k=1.

Fig 5 : Traitement de l'image du "BUREAU".