

"EXTRACTION SEQUENTIELLE DE CONTOURS; MASQUE $\nabla^2 G$ ""SEQUENTIAL EDGE EXTRACTION; $\nabla^2 G$ MASK"

José C. PINEDA

CENTRO DE INVESTIGACION Y DE ESTUDIOS AVANZADOS DEL INSTITUTO POLITECNICO NACIONAL
DEPARTAMENTO DE INGENIERIA ELECTRICA. A.P. 14-740 07000 MEXICO 14, D.F.

RESUME

Nous présentons ici, un algorithme de poursuite de contours, dont sa formulation en tant qu'automate de "décision", nous permet de tester hiérarchiquement, chaque point susceptible d'appartenir au contours, par rapport au point contour courant. Le test est binaire; à chaque point est associé une fonction booléenne F , tel que dans le cas des images binaires $F=p_{ij}$, où p_{ij} est la valeur (binaire) du pixel de coordonnées (i,j) . Dans le cas des images à niveaux de gris, la fonction F est définie à partir du résultat du filtre $\nabla^2 G$ (opérateur de Laplace appliqué sur le filtre Gaussien bidimensionnel), appliqué localement sur le voisinage, d'un certain nombre de pixels entourant p_{ij} . De ce fait, le poursuiveur du contour, suit les contours, par la simple exploitation des propriétés d'adjacence et de connectivité des images digitalisées, aussi bien dans les images binaires que dans les images à niveaux de gris. Le contour obtenu est mince (d'un pixel) et connecté, représenté comme une chaîne de directions codées.

Cet algorithme est rapide étant donné le peu de calculs à réaliser.

Nous traiterons ici le cas où nos images sont assez bien contrastées. Les améliorations à effectuer dans le cas des images peu contrastées, sont discutées brièvement.

SUMMARY

We describe a sequential edge follower for binary and gray level images. The follower is formulated with a sequential approach, as "decision" automaton. This formulation permits testing hierarchically, those pixels "susceptible" to belong to the object boundary, in relation with the last boundary point identified. The test is binary and for that, we associate a boolean function F , to each pixel. In the case of binary images the binary value of a pixel p_{ij} is taken as a result of the boolean function F , then $F=p_{ij}$, where (i,j) are the pixel coordinate pairs. In the case of gray level images, F is defined from the result of the $\nabla^2 G$ filter (Laplacian operator applied to Gaussian two-dimensional filter), applied locally on the vicinity of any pixels surrounding p_{ij} . Therefore the edge extractor follows the object boundary, by exploiting the adjacency and connectivity properties of digitalized images. We obtain a thin (1 pixel) connected contour as a coded chain.

Since the number of required operations to obtain the contour is small, the resulting follower is fast. In this paper we deal with well contrasted images.

For the case of corrupted images, we present a brief discussion on an improvement of the above algorithm.



I. INTRODUCTION

L'extraction de contours, dans les images digitalisées, est une opération préliminaire très importante, pour la réalisation de tâches de vision artificielle. Ces tâches sont telles que l'inspection industrielle, la reconnaissance de formes, la localisation d'objets dans un plan, (1) (2) etc. Par l'extraction du contour extérieur de l'image d'un objet, on réalise en effet, une segmentation de l'image, dans le but d'une compression de l'information. D'autre part, toute l'information (de forme et de dimensions) qui nous intéresse de l'objet, est contenue dans son contour (3) (4).

Nous allons présenter ici, un algorithme de poursuite de contours, dont sa formulation en tant qu'automate de décision, nous permet de tester hiérarchiquement chaque point susceptible d'appartenir au contour, par rapport au point contour courant.

L'algorithme de poursuite a l'avantage suivant:

- Le poursuiveur du contour, suit les contours, par la simple exploitation des propriétés d'adjacence et de connectivité des images digitalisées, aussi bien dans les images binaires que dans les images à plusieurs niveaux de gris. Le contour obtenu est mince (d'un pixel) et connecté, représenté comme une chaîne de directions codées.

La notation utilisée dans la présentation de l'algorithme, est donnée en structuré, pour rendre plus simple la compréhension et la programmation de celui-ci. Cet algorithme est très simple à implanter dans un minicalculateur du type utilisé dans les systèmes de vision, et ceci dans un langage quelconque, étant donné le peu de calculs à réaliser. L'intérêt de cet algorithme, dans les applications pratiques de vision par ordinateur, son évidente.

Nous traiterons ici le cas où nos images sont assez bien contrastées et peu bruitées. Les améliorations à effectuer dans le cas des images peu contrastées, sont discutées brièvement.

II. DEFINITIONS

Avant de passer à la présentation de l'algorithme il est opportun de faire un rappel sur quelques propriétés topologiques des images digitalisées, telles que l'adjacence et la connectivité. Nous ferons souvent appel à ces dernières notions, dans la suite de ce travail.

La fig. (1) représente les directions codées qui indiquent la position relative d'un pixel (au centre de l'étoile) et de ses huit voisins. Usuellement les quatre pixels voisins dans les directions impaires sont appelés les 4-voisins. Les pixels voisins dans les huit directions sont appelés les 8-voisins (5) (6)

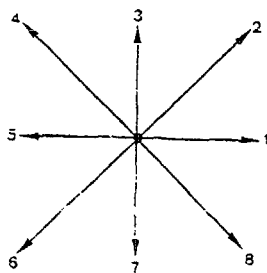


Fig. 1. Position d'un pixel et de ces huit voisins.

Si (i,j) sont les coordonnées du pixel centre de l'étoile fig. (1), les coordonnées des 4-voisins sont donc les suivantes: $(i,j+1), (i-1,j), (i,j-1), (i+1,j)$. Nous dirons que ces pixels sont 4-adjacents à (i,j) . De la même façon, les coordonnées des 8-voisins sont les suivants: $(i,j+1), (i-1,j+1), (i-1,j), (i-1,j-1), (i,j-1), (i+1,j-1), (i+1,j), (i+1,j+1)$. Nous dirons que ces pixels sont 8-adjacents à (i,j) .

Si nous avons une séquence de points $(i_0, j_0), (i_1, j_1), \dots, (i_k, j_k), \dots, (i_n, j_n)$; tous appartenant à un sous-ensemble A, et tel que (i_k, j_k) soit adjacent à (i_{k-1}, j_{k-1}) où $1 \leq k \leq n$, nous dirons que nous avons un chemin ou une "courbe" du point (i_0, j_0) au point (i_n, j_n) et que ces deux points sont connectés (5).

III. FORMULATION DE L'AUTOMATE DE POURSUITE

Avant de donner la formulation de l'automate proprement dite, nous allons donner quelques notions et définitions préliminaires.

- Un contour C 8-connecté est une courbe 8-connectée tel que: $C = \{(i_1, j_1), \dots, (i_k, j_k), \dots, (i_n, j_n)\}$ où (i_k, j_k) est 8-adjacent à (i_{k-1}, j_{k-1}) avec $1 \leq k \leq n$ et $(i_0, j_0) = (i_n, j_n)$

- Soit E une matrice de 8×8 éléments $e_{k,l}$; $k=1, \dots, 8$ et $l=1, \dots, 8$. Chaque élément $e_{k,l}$ ($E(k,l) = e_{k,l}$) est une direction codée selon le schéma de la fig. (1); $e_{k,l} \in \{1, \dots, 8\}$

- Soit ID(K) et JD(K) deux tableaux à 8 éléments $K=1, \dots, 8$ tels que: $ID = \{0, -1, -1, -1, 0, 1, 1, 1\}$ et $JD = \{1, 1, 0, -1, -1, -1, 0, 1\}$. Si nous avons les coordonnées (i,j) d'un pixel quelconque, nous pouvons trouver les coordonnées (i^*, j^*) de son voisin selon la direction $E(k,l)$ par:

$$i^* = i + ID(E(k,l))$$

$$j^* = j + JD(E(k,l))$$

- Soit (x_s, y_s) les coordonnées du point de départ du suiveur (le pixel de coordonnées (x_s, y_s) appartient à C) et κ la direction relative à la recherche préliminaire effectuée pour détecter ce point.
- Soit (x, y) les coordonnées du point courant et λ la direction courante.
- Soit (x_f, y_f) les coordonnées du premier point courant. Ce point est celui qui est pointé par la première direction courante à partir de (x_s, y_s) , dans ce sens (x_s, y_s) est le dernier point courant de C, puisqu'un point courant doit être pointé par une direction courante.
- Soit $ch(K)$ avec $K=1, \dots, N$ le tableau de sortie (résultat) du suiveur; N =nombre d'éléments du tableau=nombre d'éléments du contour.

L'automate de poursuite ou poursuiveur du contour, doit être capable de trouver tous les éléments de C.

Cet automate sera défini par un ensemble d'états D, un ensemble de vecteurs booléens B {chaque élément de chaque vecteur booléen $f \in B$, est associé aux pixels suivants, susceptibles d'appartenir au contour par rapport au point contour courant (pixels 8-adjacents au point courant)}, un ensemble R de résultats, et une application \underline{p} de $D * F$ dans $D * R$. L'automate ainsi défini, ressemble au type d'automate défini par Selkow (7) (8).

Soit d_0 l'état initial, le fonctionnement de l'automate de poursuite peut être schématisé comme suit:



"EXTRACTION SEQUENTIELLE DE CONTOURS; MASQUE $\nabla^2 G$ "

"SEQUENTIAL EDGE EXTRACTION; $\nabla^2 G$ MASK" José C. PINEDA

```

d ← d0
(x,y) ← (xf,yf) premier point courant.
f ← f0 vecteur booléen relatif au 1e point-contour
courant
Faire
    (d,r) ← (d,f)
    (x,y) ← nouvel point-contour courant
    f ← vecteur booléen relatif au point-contour
    courant
Jusqu'à - ce que (x=xf, y=yf et d=d0)
Fin faire
    
```

Notre but sera de nous servir des notions données précédemment pour construire cet automate, pour cela faire nous établirons une correspondance entre d et les directions codées. Avant de passer à la description de l'application \underline{P} , nous allons donner encore quelques définitions.

Nous définirons une image binaire comme étant l'ensemble de pixels S pour les quels leur valeur binaire est égale à 1 (image de l'objet), et de son complément \bar{S} (le fond, ensemble de pixels dont la valeur binaire est égale à 0).

Définition 1: Le contour 8-connecté d'un ensemble S, est la courbe C 8-connectée (C ∈ S), tel que chaque élément de C a au moins un pixel qui lui est 4-adjacent et qui appartient à \bar{S} .

Définition 2: Le contour 8-connecté dans une image à niveaux de gris est la courbe C à n éléments $(C = \{(i_0, j_0), \dots, (i_k, j_k), \dots, (i_n, j_n)\})$ où (i_k, j_k) est 8-adjacent à (i_{k-1}, j_{k-1}) avec $1 \leq k \leq n$ et $(i_0, j_0) = (i_n, j_n)$, tel que si un opérateur "0" de détection de contour est appliqué sur les éléments de C, noté... $C \cdot \underline{P}$ (l'opérateur 0 est appliqué localement sur le i_k, j_k voisinage carré d'un nombre pair de pixels entourant P_{i_k, j_k}), il nous donne une réponse (valeur) qui remplit i_k, j_k une relation donnée. Cette relation sera établie plus loin.

IV. DESCRIPTION DE L'APPLICATION \underline{P} ET ORGANISATION DE L'ALGORITHME.

IV.1 Description de l'application \underline{P}

En établissant une correspondance entre d et les 8 direction codées (Fig. 1), le résultat r sera aussi une des ces 8 directions, d'après les définitions précédentes. De ce fait l'état "présent" de l'automate est la direction courante sur le contour, et l'état suivant est la direction suivante sur le contour. A chaque état présent on associe un vecteur booléen qui applique par \underline{P} , l'état suivant d et le résultat r de la façon suivante: soit E la matrice des directions codées définie plus haut et donnée ci-dessus.

	1	2	3	4	5	6	7	8
1	6	7	8	1	2	3	4	5
2	7	8	1	2	3	4	5	6
3	8	1	2	3	4	5	6	7
4	1	2	3	4	5	6	7	8
5	2	3	4	5	6	7	8	1
6	3	4	5	6	7	8	1	2
7	4	5	6	7	8	1	2	3
8	5	6	7	8	1	2	3	4

Les colonne de E sont imposée par l'état présent ou direction courante (à toute direction courante correspond un point courant, donné par la ligne qui joint le point précédent et le point courant selon la Fig. 1). Si l'on associe une fonction booléenne F a chaque pixel 8-adjacent au point courant, nous obtenons le vecteur booléen f associé a l'état "présent" de l'automate. Puisque la fonction F doit nous renseigner sur l'appartenance ou non au contour (point suivant) d'un des pixels voisins au point courant, f doit donc être construit (par l'ensemble des résultats de F, associée a chaque pixel concerné), en suivant la "hierarchie" sur les pixels voisins au point courant, donnée par la colonne imposée par la direction ou état courant. Cela veut dire, que l'évolution de l'automate pour atteindre l'état suivant, est visualisé sur la colonne de E imposée par l'état présent, en partant de la ligne k=1 avec la sequence $k=1 \rightarrow k=2 \dots k=n_1$, jusqu'à ce que la "décision" sur l'appartenance au contour du point suivant ($F=1$) soit effectuée (l'état suivant est la direction qui pointe au point suivant depuis le point courant). A ce moment là les conditions suivantes: son vérifiées: $n_1 \leq 8$, $f = \{0,0, \dots, 1\}$ et l'état suivant qui deviendra a son tour l'état présent dans l'étape suivante, sera l'élément $e_{n_1, 1}$ de E, le résultat r (rest également $e_{n_1, 1}$) est $e_{n_1, 1}$ enregistré dans le tableau ch. La Matrice E nous permet donc de visualiser l'évolution de l'automate, définie par la donnée de P; $(d,r) \underline{P} (d,f)$. En général on initialise k=2 sur une colonne donnée, car le point suivant pointé par $e_{2,1}$ n'appartiendra jamais au contour.

IV.2 Organization de l'algorithme de poursuite du contour.

Avec ce que nous avons dit plus haut, nous pouvons tirer que le poursuiveur du contour doit être décomposé en deux phases à savoir:

- 1) Initialisation et recherche d'un point de départ
- 2) Avancement de la poursuite avec test d'arrêt.

Nous avons essentiellement discuté la phase 2 du poursuiveur, la phase 1 est effectuée comme suit:

La recherche peut être effectuée sur une colonne (de haut en bas $\kappa = 7$ ou dans l'autre sens $\kappa = 3$) ou sur une ligne de l'image (de gauche à droite $\kappa = 1$ ou dans l'autre sens $\kappa = 5$). Le point de coordonnées (i,j) du premier pixel rencontré par cette recherche, pour lequel $F=1$ (pixel de coordonnées (i,j)) et qui n'est pas un point isolé, appartient a C. On note que si nous recherchons (x_s, y_s) selon une direction paire, le premier point non isolé rencontré pour lequel $F=1$, n'appartient pas nécessairement a C (voir définition 1). F est la fonction booléenne associée aux pixels. Nous considérons que nous nous déplaçons toujours du fond vers l'objet.

Avec ses définitions et en considérant que le suiveur avance dans le sens trigonométrique, nous allons donner l'algorithme suivant:

Poursuite séquentielle d'un contour 8-connecté dans une image digitalisée.

Entrées. Image, matrice E, tableau ID, tableau JD, Operateur 0(matrice (n,n), dans le cas des Images à niveaux de gris)
 Sorties. Tableau ch(K) ; $K=1, \dots, N+2$.
 Variables. La variable "a" est un pointeur des lignes de la matrice E, le long d'une colonne imposée. La variable "m" est le pointeur du tableau ch. Les autres variables utilisées ont été déjà définies.

Remarque:

- Le bloc faire jusqu'à ce que s'effectue au moins une fois (première fois).



"EXTRACTION SEQUENTIELLE DE CONTOURS; MASQUE $\nabla^2 G$ "

"SEQUENTIAL EDGE EXTRACTION; $\nabla^2 G$ MASK"

José C. PINEDA

POURSUITE DU CONTOUR

1. DEBUT POURSUITE

1.1 Début bloc 1.1, initialisation

- 1 - Initialiser pointeurs: $m=1$, $a=1$
- 2 - Trouver point de départ (x_s, y_s) et K
- 3 - $ch(m) = x_s$; incrémenter m
- 4 - $ch(m) = y_s$
- 5 - Initialiser $i=x_s+ID(E(a,K))$ et $j=y_s+JD(E(a,K))$

Fin bloc 1.1.

1.2 Début bloc 1.2 ; recherche du premier élément de la chaîne codée.

Faire tant que $F=0$

- 1 - Incrémenter a
- 2 - $i = x_s + ID(E(a,K))$
- 3 - $j = y_s + JD(E(a,K))$

Fin bloc 1.2.

1.3 Début bloc 1.3 ; Enregistrement 1er élément de la chaîne et 1er point courant.

- 1 - Incrémenter m
- 2 - $ch(m) = E(a,K)$; premier élément de la chaîne codée
- 3 - Les coordonnées du point courant sont les dernières valeurs de (i,j) initialiser donc $x=i$, $y=j$.
- 4 - Enregistrer $x_f=x$, $y_f=y$; coordonnées du 1er point courant.

Fin bloc 1.3.

FIN BLOC DEBUT POURSUITE

2. DEBUT BLOC AVANCEMENT DE LA POURSUITE AVEC TEST D'ARRET.

2.1 Début bloc 2.1 ; avancement et test d'arrêt Faire, jusqu'à ce que $(x=x_f$ et $y=y_f$ et $ch(m)=ch(3))$

2.1.1 Début bloc 2.1.1

- 1 - $\lambda = ch(m)$
- 2 - $a = 2$ (afin d'éviter un test inutile)
- 3 - $i = x + ID(E(a, \lambda))$
- 4 - $j = y + JD(E(a, \lambda))$

Fin bloc 2.1.1.

2.1.2 Début bloc 2.1.2

Faire tant que $F=0$

- 1 - Incrémenter a
- 2 - $i = x + ID(E(a,\lambda))$
- 3 - $j = y + JD(E(a,\lambda))$

Fin bloc 2.1.2.

2.1.3 Début bloc 2.1.3.

- 1 - Incrémenter m
- 2 - $ch(m) = E(a,\lambda)$
- 3 - $x=i$, $y=j$ (mise à jour du point courant)

Fin bloc 2.1.3.

Fin bloc 2.1

2.2. Début bloc 2.2, sortie

- 1 - Décrémenter m
- 2 - $N1 = m$; $N = N1 - 2$
- 3 - Afficher N
- 4 - Tracer le contour?

Fin bloc 2.2

FIN BLOC AVANCEMENT DE LA POURSUITE AVEC TEST D'ARRET

FIN POURSUITE DE CONTOUR.

Dans le cas des images binaires la valeur binaire du pixel P_{ij} est prise ici comme le résultat d'une fonction booléenne F tel que $F=P_{ij}$.

Dans le cas des images à niveaux de gris, nous nous servons pour définir F , du masque $\nabla^2 G$ que nous avons obtenue directement à partir de l'opérateur $\nabla^2 G$ (9), lequel sera brièvement présenté.

Il a été montré en (9) qu'il existe un filtre optimisant la relation, $\Delta x \Delta \omega > \frac{1}{4} \pi$, et en conséquence les conditions d'une part dans le domaine spatial et d'autre part dans le domaine fréquentiel sur les caractéristiques du filtre à utiliser, pour le traitement à donner à une image dans le but de la détection des contours. Ce traitement doit donner comme résultat, une réduction de l'échelle de variations sur laquelle les changements d'intensité ont lieu. Ce filtre est appelé Gaussien et sa réponse impulsionnelle bidimensionnelle est donnée par la relation:

$$G(x,y) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x^2+y^2)}{2\sigma^2}}$$

Dans ces conditions, la détection de contours va s'effectuer par la détection des passages à zéro de la fonction $z(x,y)$ tel que:

$$z(x,y) = \nabla^2 G(x,y) * I(x,y)$$

$$\text{ou } \nabla^2 G(x,y) = -\frac{1}{\sqrt{2\pi}\sigma^3} \left(1 - \frac{x^2+y^2}{2\sigma^2}\right) e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (9)$$

La fenêtre de filtrage (masque) $\nabla^2 G$ que nous proposons et que nous appellerons par la suite simplement opérateur $\nabla^2 G$, travaille sur le voisinage carré d'un nombre pair de pixels entourant le point sur lequel l'opérateur est appliqué. Ceci afin de garder une uniformité par rapport aux masques du type gradient ou moyennage. Nous avons considéré que l'image est une fonction bidimensionnelle $I(i,j)$ échantillonnée spatialement avec une période d'échantillonnage unitaire aussi bien sur l'axe X que sur l'axe Y ($i=1,2,\dots,N$; $j=1,2,\dots,N$). Ceci nous a conduit, afin d'avoir une interprétation simple des résultats, à échantillonner $\nabla^2 G$ avec la même période d'échantillonnage, pour les différentes tailles de la fenêtre de filtrage (masques). Il est évident que pour une taille donnée de la fenêtre, nous pouvons faire varier le paramètre σ , ce qui nous permet de faire varier la bande passante du filtre ($\sigma \uparrow \Delta \omega \downarrow$).

Nous allons maintenant définir une fonction booléenne F , à partir du résultat de l'opérateur $\nabla^2 G$ appliqué localement sur P_{ij} , comme suit:

$$F = \nabla^2 G \cdot P_{ij} \begin{cases} \text{si } \nabla^2 G \cdot P_{ij} \geq 0 & \text{alors } F=0 \\ \text{si } \nabla^2 G \cdot P_{ij} < 0 & \text{alors } F=1 ; P_{ij} \text{ est un point contour} \end{cases}$$

V. PRESENTATION DES RESULTATS

La fig. (2) nous montre l'opérateur $\nabla^2 G$ et son spectre fréquentiel. La fig. (3) nous montre la convolution de l'image originelle (3a) par l'opérateur $\nabla^2 G$ dans la condition de la figure (2). La taille de la fenêtre dans ce cas est de 5×5 .



"EXTRACTION SEQUENTIELLE DE CONTOURS; MASQUE $\nabla^2 G$ "

"SEQUENTIAL EDGE EXTRACTION; $\nabla^2 G$ MASK"

José C. PINEDA

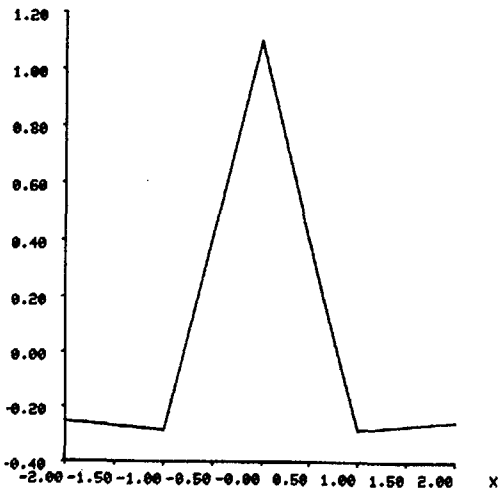
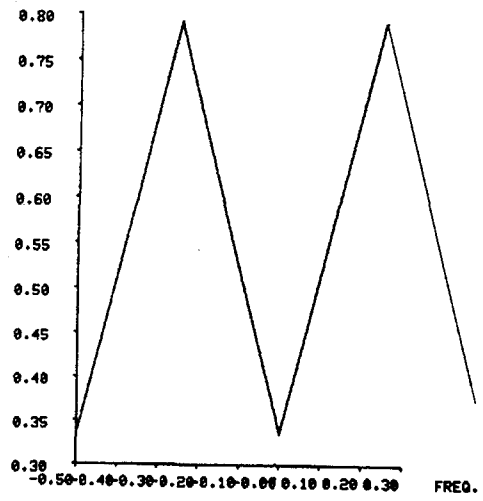


Fig. 2a. Masque $\nabla^2 G$.



2b. Spectre fréquentiel (projection) de $\nabla^2 G$ a $\sigma=0.8$ sur masque 5x5.

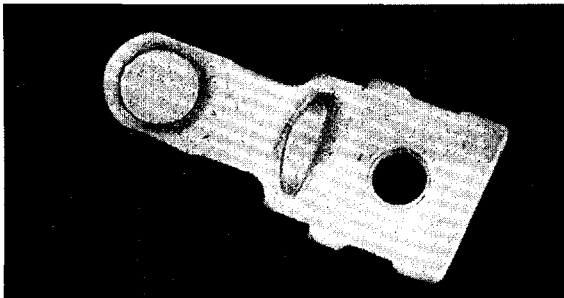
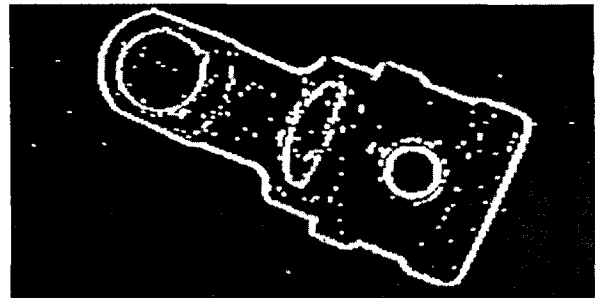


Fig. 3a Image d'une pièce mécanique.



3b. Convolution de 3.a par le masque $\nabla^2 G$ masque 5x5 et $\sigma=0.85$.

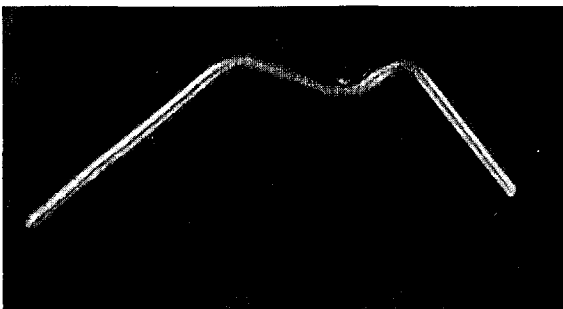
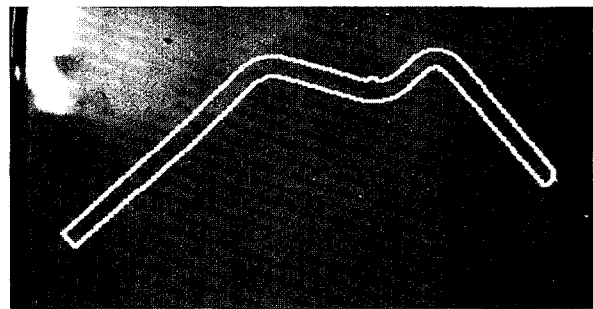


Fig. 4a. Image d'une pièce mécanique.



4b. Contour de la pièce 4a obtenue par l'algorithme présenté sur fenetre 5x5 et $\sigma=0.85$.



"EXTRACTION SEQUENTIELLE DE CONTOURS; MASQUE $\nabla^2 G$ "

"SEQUENTIAL EDGE EXTRACTION; $\nabla^2 G$ MASK"

José C. PINEDA

VI. CONCLUSIONS.

Nous avons présenté un algorithme de poursuite de contours dans une image digitalisée. L'automate de poursuite ou poursuiveur trouve tous les points-contour par l'exploitation des propriétés d'adjacence et de connectivité des images digitalisées. Dans le cas des images à niveaux de gris, on utilise le masque $\nabla^2 G$ obtenue à partir de l'opérateur $\nabla^2 G$, pour tester l'appartenance ou non des points au contour. L'algorithme est rapide et simple à implementer.

Lorsque nous nous trouvons dans le cas de pièces très réfléchissantes (pièces métalliques fig. (3), il n'est pas garanti que le contour soit bien démarqué. Le suiveur tel que nous l'avons décrit, peut se "figer en non décision" ou bien suivre les reflets. Une première solution à ce problème ne concernant pas le logiciel, consiste à choisir un fond absorbant bien contrasté par rapport à la pièce. Dans le cas où cette solution ne suffit pas, il faut donc soit, avoir une connaissance a priori de l'objet (10), soit compliquer l'algorithme présenté en lui ajoutant des procédures interpolatives qui nous permettent de "sauter" les trous dans les contours dans la direction de la poursuite (11).

REFERENCES BIBLIOGRAPHIQUES

- (1) PERKINS W.A., A model-based vision system for industrial Parts. IEEE. T-C V. C-27, No. 2 1978.
- (2) DUDA R.O. and HART P.E., Pattern Classification and scene Analysis. John Wiley and Sons.
- (3) PAVLIDIS T. Algorithms for shape analysis of contours and waveforms. IEEE T. PAMI-2(4). 1980.
- (4) PINEDA J.C. and HORAUD P., An improved method for high-curvature detection with applications to automated inspection. Signal Processing 5 (1983).
- (5) ROSENFELD A. and KAK A.C., Digital Picture Processing Academic Press 1976.
- (6) PINEDA J.C., On a sequential contour following algorithm for binary images on square and hexagonal grids. Submitted paper, copies available for the autor.
- (7) SELKOW S.M. One pass complexity of digital picture. JACM. April 72, V-19, No. 2.
- (8) VEILLON F., Une méthode de calcul en un passage de plusieurs propriétés topologiques et géométriques d'objets dans des images digitalisées. RR No. 67, Janvier 77, ENSIMAG, France.
- (9) MARR D. and HILDRETH E. Theory of edge detection. Proc. R. Soc. London B. 207, 1980.
- (10) GRIFFITH A.K., Edge detection in simple scene using a priori information. IEEE T.C. C-22, 1973
- (11) DESSIMOZ J.D., Traitement de contours en reconnaissance de formes visuelles: Applications en Robotique. Thèse, EPFL. 1980.