

## UNE METHODE NOUVELLE DE "CHAIN CODING" PAR SUIVI DE CONTOUR POUR LA SEGMENTATION D'IMAGES BINAIRES COMPLEXES

V. BOMBARDIER, E. LEVRAT, B. HEIT et J. BREMONT

Centre de Recherche en Automatique de Nancy - U.R.A. 821 - Laboratoire d'Electricité et d'Automatique  
Université de NANCY I - Faculté des sciences - BP 239  
54506 Vandœuvre-lès-Nancy Cedex

### RÉSUMÉ

Nous proposons dans cet article une méthode originale de codage en chaîne par suivi de contour dans des images binaires. L'algorithme récursif décrit les formes en 8-voisinage et combine le suivi à un marquage en "ligne" homogène. Associée à un marquage global, cette structure autorise le traitement de formes complexes et imbriquées.

### ABSTRACT

In this paper, we present a new method of Chain Coding by border following in binary images. This recursive algorithm uses 8-connectivity for all shapes, and combines following with "in line" homogeneous labelling. In association with a global labelling, this structure allows the processing of complex and nested shapes.

## 1. INTRODUCTION

L'analyse géométrique et topologique d'une scène, ainsi que la compression d'images cernent les principaux domaines d'intérêt du codage en chaîne. Cette approche de la segmentation d'images par les contours est classiquement abordée de deux façons différentes. On retrouve les méthodes nécessitant une image des contours en niveaux de gris ou binaires, et celles qui travaillent directement sur l'image binarisée. Ces dernières procèdent par description ligne à ligne de l'image (Raster Scan), ou par couplage d'une description ligne à ligne et d'un suivi de contour.

Notre travail s'inscrit dans cette dernière famille avec des méthodes réalisant un bon compromis entre le temps de calcul et l'encombrement mémoire.

Les principales contraintes à respecter sont ici l'invariance des contours par rapport à l'image inverse, et l'interdiction de croisement des contours.

Bien que notre démarche générale reste classique, ses originalités résident dans la description systématique des formes en 8-voisinage (objets, évidements, objets internes), le traitement de scènes complexes (formes imbriquées), une structure de travail récursive et un double marquage homogène.

La robustesse de la méthode est éprouvée sur des applications bio-médicales et en sciences des matériaux.

Après avoir décrit les principales phases de notre algorithme dans la première partie de ce document, nous récapitulons et commentons les différentes caractéristiques de notre méthode et de celles existantes, telles que : le marquage, l'occupation mémoire, le voisinage.

## 2. DESCRIPTIF DE LA METHODE

La méthode que nous proposons remédie aux erreurs de suivi récemment mises en évidence par Danielsson [1] et Haig [2] et se déroule en trois phases successives : la recherche des points de départ, le suivi combiné au marquage en ligne afin de résoudre le

problème de connexité, sans marquage hétérogène des frontières et enfin le marquage final, qui offre la possibilité de traiter des objets troués ou imbriqués (fig. 1).

### 2.1. Recherche des points de départ

On distingue pour cette phase trois types de points de départ, fonction des différentes formes de l'image.

Le point de départ d'un objet est le premier pixel de niveau de gris "1", rencontré en parcourant l'image de gauche à droite, lors de la description ligne à ligne.

Le point de départ d'un évidement, est défini comme étant le premier pixel de niveau de gris "0", rencontré en parcourant la fenêtre de scrutation délimitant l'objet parent par le parallélogramme rectangle circonscrit à l'objet. Ses paramètres sont calculés lors du suivi du contour de l'objet qui est la forme la moins imbriquée. La recherche de ce point s'effectue sur les seuls pixels appartenant à l'objet et repérés grâce au marquage de son périmètre, réalisé durant l'étape de suivi, qui génère les codes d'entrée-sortie de l'objet (§ 2.3.2.).

Le point de départ d'un objet interne est le premier point de niveau de gris "1", rencontré à l'intérieur d'un évidement. Le niveau d'imbrication de la forme décrite repère cette position, et se calcule pendant le parcours d'une ligne en fonction des codes d'entrée-sortie générés lors du suivi de l'évidement.

D'une manière générale, la recherche de la forme suivante est faite à partir du point de départ de la forme précédente pour un même niveau d'imbrication.

### 2.2. Suivi des contours

#### 2.2.1. Principe général : règle du "left most looking"

Dans cette phase, nous pouvons différencier deux cas, le suivi d'objet et le suivi d'évidement pour lesquels, l'algorithme général ne varie pas. Seul change le niveau de gris pris en compte pour la recherche des pixels appartenant au contour.

L'algorithme principal de cette partie reste classique et utilise la règle du "Left Most Looking" [3] : le point suivant du contour



est recherché à partir du point courant en testant les directions de Freeman, en fonction de la direction courante moins  $\Pi / 2$ . La description de l'image est faite en 8-voisinage pour une définition plus fine des contours. En outre, la chaîne de codes générée est moins longue puisqu'une direction diagonale remplace deux directions perpendiculaires.

La nature des erreurs de suivi de cette méthode ou des méthodes similaires a été mise en évidence par Danielsson et par Haig et Attikiouzel :

- la mauvaise description des évidements due à la façon de considérer la frontière entre l'objet et l'évidement, qui entraîne la non-invariance par rapport à l'image inversée;
- le croisement des contours dû à la description des contours d'évidement en 8-voisinage.

Notre algorithme évite ces erreurs de description en intégrant la notion de contour d'évidement, c'est-à-dire que ces contours sont suivis sur les pixels à "0" de l'évidement. De plus, le marquage "en ligne" des contours résout le problème du paradoxe de connexité, sans marquage hétérogène des contours. En effet, contrairement à certaines méthodes antérieures, les objets et les évidements peuvent être décrits en 8-voisinage sans risque de croisement.

Enfin, le suivi des frontières d'objets et des frontières d'évidements, se fait toujours dans le sens horaire, ce qui simplifie la procédure de suivi, puisque le même algorithme est utilisé dans les deux cas, contrairement à la méthode de Haig et Attikiouzel.

### 2.2.2. Tests d'arrêt

Un algorithme de suivi de contours peut classiquement rencontrer trois types de terminaison :

- le retour au point de départ,
- l'arrivée sur le cadre de l'image,
- l'impossibilité de trouver un point suivant.

La première condition, seule autorisée, correspond à la fin normale du processus, les deux autres sont gérées de manière à retrouver le point de départ dans tous les cas.

La rencontre du cadre de l'image suppose la description d'un objet partiellement visible. L'algorithme suit alors ce cadre de l'image jusqu'à retrouver le contour de l'objet. Notons qu'un test peut être inclus à ce niveau pour éliminer l'objet incomplet.

La troisième condition apparaît lorsque le suivi parcourt un segment de droite. Comme il n'existe pas, en fin de segment, de pixel non marqué appartenant au contour, la méthode autorise, dans ce cas uniquement, le retour en arrière; c'est-à-dire le passage sur des pixels déjà pris en compte. La description du contour est éventuellement reprise pour des pixels non marqués, si le segment est une singularité de la forme.

### 2.2.3. Fenêtre de scrutation

La fenêtre de scrutation, définie comme le rectangle circonscrit à la forme globale, est utilisée pendant la recherche des points de départ des formes imbriquées et pour le marquage final de l'objet suivi. Elle permet de réduire le temps de traitement.

La fenêtre est mémorisée sous la forme de quatre coordonnées calculées au cours de la phase de suivi et données par les numéros de lignes et de colonnes minimaux et maximaux atteints par l'opérateur de suivi.

### 2.2.4. Cas particulier du point de départ

Le suivi de contour se poursuit jusqu'à retrouver le point de

départ du contour considéré. A ce niveau, si la dernière direction parcourue est 7 ou 8 au sens du code de Freeman (fig. 2), la méthode inclut un test systématique de la direction 6. Ce test est destiné à empêcher une erreur de description sur des formes particulières qui générerait alors deux chaînes de codes pour une seule forme. Seule la méthode de Haig et Attikiouzel fait état de ce cas particulier.

## 2.3. Marquage des formes de l'image

Le marquage de l'objet ou de l'évidement se décompose en deux parties [4] qui sont le marquage des différents contours effectué pendant le suivi du contour correspondant, et le marquage total de la forme en fin de traitement, effectué lorsque celle-ci ne contient plus de formes internes non décrites. Les zones de l'image déjà traitées sont alors différenciées.

### 2.3.1. Compteur d'imbrication

Afin de permettre la recherche des points de départ et un marquage correct de la forme décrite, la méthode utilise un compteur qui précise le niveau d'imbrication d'une forme à partir des codes d'entrée - sortie (§ 2.3.2). Lors du parcours d'une ligne de l'image, il est incrémenté lorsqu'on rencontre une succession de pixels correspondant à l'entrée d'une forme et décrémenté pour une sortie.

Le compteur est initialisé à 0 lors du traitement d'un objet ou objet interne, et à 1 pour un évidement. Son incrémentation, faite de deux en deux pour distinguer les évidements et les objets fait correspondre une valeur paire aux objets ou objets internes et une valeur impaire aux évidements.

### 2.3.2. Marquage "en ligne" :

Cette partie de marquage, active pendant le suivi de contour, génère les codes d'entrée et de sortie de chaque forme pour permettre le marquage final de l'objet ou de l'évidement et l'évolution du compteur d'imbrication.

Le marquage joue un rôle important pour les différentes méthodes dont il est la partie originale les distinguant les unes des autres. Il empêche le croisement des contours et facilite la gestion des entrées-sorties et des imbrications de formes.

Par rapport aux contraintes que nous nous sommes fixées, les divers marquages proposés par les méthodes existantes ne sont pas satisfaisants puisqu'ils limitent, par exemple, le nombre de formes à traiter comme pour Kruse [5] ou Danielsson.

Enfin, contrairement à Haig, nous préférons un marquage homogène des contours. Les marques, différentes pour des formes distinctes, ne varient pas tout au long du contour d'une même forme.

Nous avons donc opté pour un marquage ressemblant à une combinaison de ceux de Kruse et Danielsson, puisque nous marquons à la fois les pixels appartenant aux contours et les pixels situés à l'extérieur de la forme, c'est-à-dire à gauche de la frontière lorsqu'elle est parcourue dans le sens horaire. Ce sont les points testés lors de la recherche du point suivant qui n'ont pas été retenus.

Ce procédé systématique n'implique pas de tests supplémentaires et interdit naturellement le croisement des contours (figure 3-1) sans imposer, comme pour Haig, de tests de croisement.

### - Définition des codes élémentaires du marquage :

Nous définissons quatre codes utilisés pour le marquage en ligne, dédoublés par les deux valeurs que peut prendre l'indice x



et que nous détaillons ultérieurement Ils sont fonction du niveau d'imbrication et de la nature de la forme à traiter : objet ou évidement.

Code Extérieur Frontière Objet (CEFO<sub>x</sub>):

affecté aux pixels appartenant à la frontière extérieure de l'objet et décrit lors de la recherche du point suivant .

Code Frontière Objet (CFO<sub>x</sub>):

affecté aux pixels appartenant au contour de l'objet.

Code Extérieur Frontière d'Evidement (CEFE<sub>x</sub>):

pixels appartenant à la frontière extérieure de l'évidement.

Code Frontière d'Evidement (CFE<sub>x</sub>):

pixels appartenant au contour de l'évidement.

Les codes élémentaires ont une fonction individuelle. Lors de la phase de suivi, le code frontière marque le contour de la forme et autorise le retour en arrière en cas d'impasse (fig 3-1B), le code extérieur empêche le croisement des contours qui survient lors de descriptions de directions diagonales, si l'extérieur de la frontière de la forme parente n'est pas marqué (fig 3-1A).

Grâce au marquage choisi, un pixel extérieur à la forme parente de la forme en cours de traitement ne peut être pris comme pixel suivant (fig 3-2C).

#### - Définition des codes d'entrée-sortie :

Néanmoins, l'utilisation principale des codes élémentaires se fait par leur association sous forme de couple de pixels repérant les entrées et les sorties. Ces couples constituent de nouveaux codes exprimés à partir des définitions précédentes.

Le marquage extérieur de la frontière impose certaines contraintes pour le traitement de formes imbriquées d'épaisseur un pixel. La première conséquence est que chaque code d'entrée ou de sortie est défini par deux couples de codes élémentaires, l'autre conséquence est rencontrée lors du marquage d'une forme interne à une forme d'épaisseur "1" pixel puisque la marque extérieure efface la marque frontière précédente, et entraîne une erreur de gestion des codes d'entrée-sortie (fig. 3-3).

Pour résoudre cette difficulté, nous avons choisi d'effectuer une permutation des codes d'entrée-sortie, liée au niveau d'imbrication des formes (fig. 3-4). Elle se traduit par l'indice x des codes qui prend deux valeurs : i lorsque le niveau d'imbrication est impair et p lorsqu'il est pair. Cela double le nombre de marques utiles pour la génération des codes d'entrée-sortie.

Niveau d'imbrication pair :

CEO<sub>p</sub> = (CEFO<sub>p</sub>, CFO<sub>p</sub>) ou (CEFO<sub>p</sub>, CFE<sub>p</sub>) Code d'Entrée d'Objet.

CSO<sub>p</sub> = (CFO<sub>p</sub>, CEFO<sub>p</sub>) ou (CFE<sub>p</sub>, CEFO<sub>p</sub>) Code de Sortie d'Objet.

CEE<sub>p</sub> = (CEFE<sub>p</sub>, CFE<sub>p</sub>) ou (CEFE<sub>p</sub>, CEFO<sub>i</sub>) Code d'Entrée d'Evidement.

CSE<sub>p</sub> = (CFT<sub>p</sub>, CFE<sub>p</sub>) ou (CEFO<sub>i</sub>, CFE<sub>p</sub>) Code de Sortie d'Evidement.

Niveau d'imbrication impair :

CEO<sub>i</sub> = (CEFO<sub>i</sub>, CFO<sub>i</sub>) ou (CEFO<sub>i</sub>, CFE<sub>i</sub>) Code d'Entrée d'Objet.

CSO<sub>i</sub> = (CFO<sub>i</sub>, CEFO<sub>i</sub>) ou (CFE<sub>i</sub>, CEFO<sub>i</sub>) Code de Sortie d'Objet.

CEE<sub>i</sub> = (CEFE<sub>i</sub>, CFE<sub>i</sub>) ou (CEET<sub>i</sub>, CEFO<sub>p</sub>) Code d'Entrée d'Evidement.

CSE<sub>i</sub> = (CFE<sub>i</sub>, CFE<sub>i</sub>) ou (CEFO<sub>p</sub>, CFE<sub>i</sub>) Code de Sortie d'Evidement.

#### 2.3.2. Marquage final

Le marquage final nécessite la définition de deux codes supplémentaires CO et CT, respectivement Code d'Objet et Code d'Evidement, utilisés pour marquer entièrement les formes déjà décrites.

Il est effectué par description des lignes de la fenêtre de scrutation et repérage les codes d'entrée-sortie de chaque forme, afin de connaître le niveau d'imbrication correspondant. Il se divise en marquage d'objets et marquage d'évidements, et affecte l'ensemble des pixels constituant un objet ou un objet interne, du code CO. Pour les évidements, les pixels sont marqués CT.

La marque des pixels appartenant à la frontière extérieure d'une forme est modifiée pour faciliter le traitement des formes voisines. Les pixels codés précédemment à CEFE<sub>x</sub> ou CEFO<sub>x</sub> sont affectés respectivement des codes CFO<sub>x</sub> et, CFE<sub>x</sub> ou "0" si le niveau d'imbrication est nul.

Ce marquage permet le traitement de scènes comportant plusieurs objets. La marque peut être identique pour chaque objet, ou différencié afin de les étiqueter ou les discriminer (fenêtrage, isolation, ...).

Le marquage final est appliqué pour faciliter le traitement de la forme entière et pour la recherche des points de départ des autres formes non décrites de l'image. En effet, dans le cas où l'on ne choisit pas un marquage différencié, les formes ayant le même niveau d'imbrication ont les mêmes marques. L'une d'elle peut modifier à tort le compteur d'imbrication si elle est partiellement visible dans la fenêtre de scrutation d'une autre forme en cours de traitement. Le marquage final remplace, en fin de traitement, tous les codes d'entrée-sortie de sorte qu'ils ne puissent pas être pris en compte par la suite.

La solution retenue (codage total de l'objet et permutation des quadruplets de codes d'entrée-sortie) ne nécessite que dix codes différents quel que soit le nombre d'imbrications et le nombre de formes dans l'image.

### 3. RESULTATS ET PERSPECTIVES

Les différentes caractéristiques de notre méthode sont résumées dans le tableau de la figure 4. On y retrouve également celles des techniques classiques de compréhension voisine.

Notre travail répond aux différentes contraintes énoncées et présente l'avantage majeur de nécessiter un encombrement mémoire réduit et fixe quelle que soit la complexité de l'image à traiter.

Les performances de notre méthode reposent sur l'originalité du procédé de marquage, clé de voûte de toute démarche de codage en chaîne.

La robustesse de notre algorithme est prouvée par deux applications ayant abouti à la commercialisation de logiciels en Génie Biomédical pour l'expression d'un index de neurotisation (Neurofib) et en Sciences des Matériaux pour l'estimation de l'orientation de fibres de verre dans une matrice de polypropylène.

Nos travaux s'orientent actuellement vers une méthode inspirée de la précédente, s'appliquant à des images à plusieurs niveaux de gris, et conservant notamment les avantages du procédé de marquage. Cette extension s'impose lorsque l'image ne possède pas une qualité suffisante pour être binarisée.

### 4. BIBLIOGRAPHIE

- [1] Danielsson P.E. : "An improved Segmentation and Coding Algorithm for Binary and non-Binary Images", IBM J. Res. Developpement, novembre 1982, vol. 26, n° 6, pp. 698-707.
- [2] Haig T.D. et Attikiouzel Y. : "An improved algorithm for border following in binary image", European Conference on



- Circuit Theory and Design, Brighton UK, 5-8 septembre 1989.
- [3] **Gonzalez C. et Wintz P.** : "Digital Image Processing", Addison Wesley, 1987, pp. 253-260.
- [4] **Bombardier V., Levrat E., Bremont J.** : "Un algorithme robuste de suivi de contour. Application au comptage de fibres nerveuses", ISMM Int. Symposium, Lugano (Suisse), 19-21 juin 1990.
- [5] **Kruse B.** : "A fast algorithm for segmentation of connected components in binary images", Proceeding of first Scandinavian conference on image analysis studentlitteratur, Suède, janvier 1980, pp. 57-63.
- [6] **Suzuki S. et Abe K.** : "Topological Structural Analysis of digital Binary Image by border Following", Computer Graphics and Image Processing, IEEE, vol. 30, 1985, pp. 32-46.

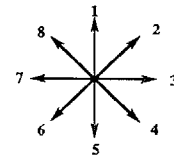


Figure 2 : Codes de Freeman

**Valeurs des différents codes de la figure 3 :**

- CEFOp = 3 CFOP = 2 CEFEp = 4 CFEp = 5  
 CEFOi = 7 CFOi = 6 CEFEi = 8 CFEi = 9  
 CEOp = (3,2) ou (3,4) CSOp = (2,3) ou (4,3) CEEp = (4,5) ou (4,7) CSEp = (5,4) ou (7,4)  
 CEOi = (7,6) ou (7,8) CSOi = (6,7) ou (8,7) CEEi = (8,9) ou (8,3) CSEi = (9,8) ou (3,8)

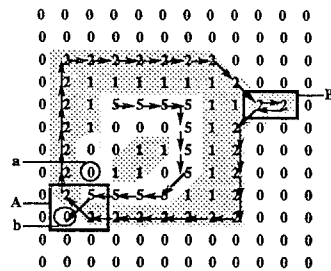


Figure 3-1 : Sans marquage extérieur

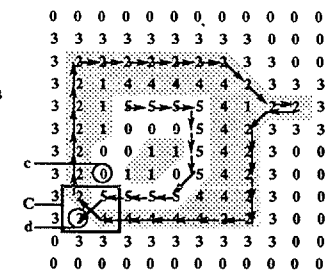


Figure 3-2 : Avec marquage extérieur

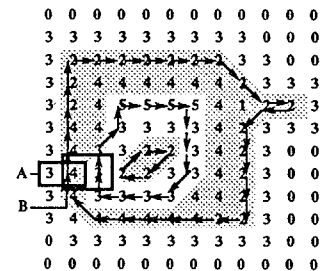


Figure 3-3 : Sans permutation

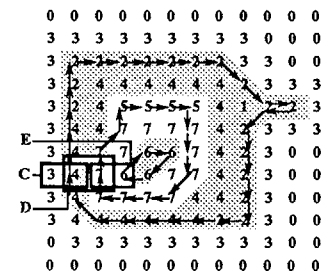


Figure 3-4 : Avec permutation

Figure 3-3 : Lors du marquage de l'objet interne, la permutation n'est pas appliquée. Si A représente le code d'entrée dans l'objet, B est confondu avec le code de sortie de ce même objet alors qu'il représente le code d'entrée dans l'évidement.

Figure 3-4 : Cette figure représente l'image après la description de l'objet interne, mais avant le marquage final de ce dernier. La permutation évite l'erreur précédente sans augmenter considérablement le nombre de marques : C représente l'entrée dans l'objet, D l'entrée dans l'évidement et E l'entrée dans l'objet interne.

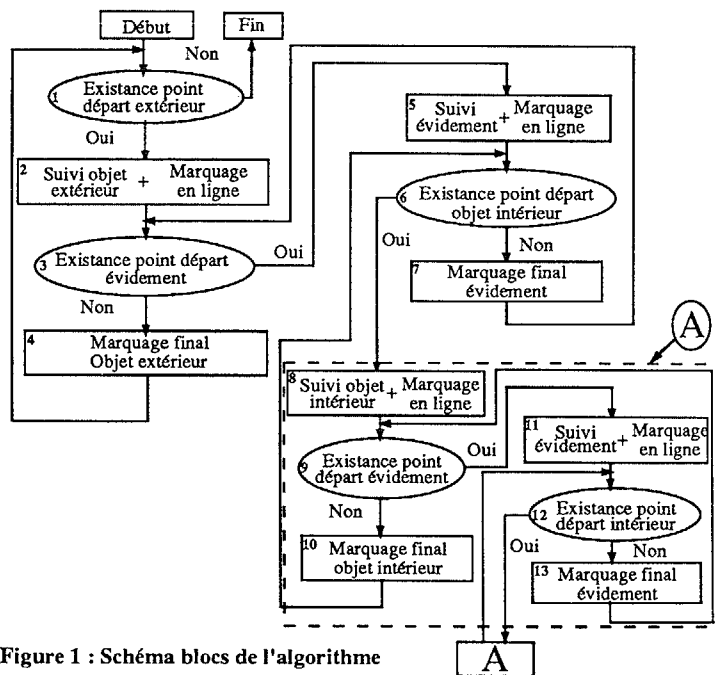


Figure 1 : Schéma blocs de l'algorithme

Figure 4 : Tableau récapitulatif des caractéristiques des différentes méthodes

	Technique de marquage utilisée	Place mémoire occupée	Description des contours		Invariance des contours	Particularités Remarques
			Voisinage	Sens		
Méthode de KRUSE (1980)	Marquage des formes complètes et repérage des sorties de formes 1 marque par forme + 1 repère par forme.	Image initiale (N niv de gris) + pile (marque) + repère	Evidement 4 voisins Objet 8 voisins	Trigo	Non	Nombre de formes totales dans l'image limité par la mémoire image multi-niveaux (une marque par forme) et un bit supplémentaire occupé en mémoire pour les repères de fin de formes. Marquage homogène des contours et des formes
Méthode de DANIELSSON (1982)	Marquage des pixels situés à l'Est du contour. 1 marque par forme	Image initiale (N niv de gris) + indice de parité	Evidement 4 voisins Objet 8 voisins	Horaire	Non	Version simplifiée de la méthode de KRUSE (plus de repère). La pile est remplacée par un indice de parité. La forme entière n'est pas marquée. Le nombre de formes pouvant être traitées dans une image est doublé mais toujours limité par le marquage. Marquage hétérogène et non systématique des contours.
Méthode de SUZUKI ABE [6] (1985)	Marquage des pixels du contour 2 marques par forme (entrée et sortie)	Image initiale (N niv de gris) + repère d'imbrication	Evidement 8 voisins Objet 8 voisins	Trigo	Non	Nombre de formes pouvant être décrites limité par le marquage comparable à celui de la méthode de Kruse. Les contours d'évidement sont en fait les contours intérieurs d'objets. Marquage hétérogène des contours.
Méthode de ATTIKIOUZEL HAIG (1989)	Marquage des pixels du contour + codage de la forme entière après suivi 4 marques pour toute l'image.	Image initiale (N niv de gris) + contours parents de la forme en cours de traitement	Evidement 8 voisins Objet 8 voisins	Evid. Trigo Objet Horaire	Oui	Méthode récente. Bonne description des contours. Problème de croisements résolu par un test de blocage. Mémorisation des contours nécessaire. Réservation mémoire difficile. Marquage hétérogène et non systématique des contours. Structure entièrement récursive de l'algorithme. Prise en compte des contours d'évidement.
Méthode proposée (1990)	Marquage des pixels du contour (marquage en ligne) + marquage final de la forme en cours 10 marques pour toute l'image.	Image initiale (N niv de gris) + coordonnées de la fenêtre de scrutation + compteur d'imbrication	Evidement 8 voisins Objet 8 voisins	Horaire	Oui	Marquage sur image initiale. Bonne description des contours. Problèmes de croisement résolu par marquage en ligne. Occupation mémoire réduite grâce à l'utilisation d'une fenêtre de scrutation. Structure entièrement récursive de l'algorithme. Prise en compte des contours d'évidement.