

**Institut National Polytechnique de Toulouse**

**Ecole Nationale Supérieure d'Electrotechnique,  
d'Electronique, d'Informatique et  
d'Hydraulique de Toulouse**

**Filière INFORMATIQUE**

Supplément au

Rapport de fin d'étude E.N.S.E.E.I.H.T

en vue de l'obtention du D.E.A informatique

**PROJET R.E.V.E.R.I.E:**

Utilisation d'un système expert pour la segmentation  
d'images en régions et contours.

Vincent RODIN

Septembre 1989

## Sommaire

I	Introduction	page	1
II	Contenu du présent rapport	page	2
III	Représentation symbolique des contours des objets	page	3
	III-1 Structures de données LISP	page	3
	III-2 Principaux algorithmes	page	5
IV	Protocole de communication entre les deux machines	page	8
V	Le système expert et les règles utilisées	page	10
	V-1 Les règles sur les régions	page	10
	V-1.1 Règles de fusion de deux régions	page	10
	V-1.2 Règles de division d'une région en plusieurs	page	16
	V-1.2.1 Histogramme en niveaux de gris d'une région	page	16
	V-1.2.2 Les règles de division de région	page	18
	V-2 Les règles sur les lignes	page	20
	V-2.1 Les extracteurs de contours	page	20
	V-2.2 Les règles proprement dites	page	22
VI	Conclusion	page	27
	BIBLIOGRAPHIE	page	28

## I Introduction

Ce rapport de stage "supplément D.E.A." fait suite au rapport de stage E.N.S.E.E.I.H.T. option informatique (Juin 1989).

Il présente l'utilisation d'un système expert pour réaliser une segmentation en régions et contours "propres" car les régions obtenues par les outils classiques ne permettent pas une reconnaissance directe des objets.

Des règles sur la segmentation de bas niveau y seront exposées et discutées.

MOTS CLÉS: Vision, Traitement d'images et Système Expert.

**REMARQUE**: Ce rapport a été écrit en supposant que le lecteur avait pris connaissance du premier rapport:

Projet R.E.V.E.R.I.E. :

Contribution à la segmentation et à la représentation symbolique des régions pour le système de vision bas niveau.

## II Contenu du présent rapport

Après avoir réalisé quelques représentations symboliques des régions, en vue de l'utilisation d'un système expert, il est nécessaire de définir une représentation des contours obtenus par des extracteurs de contours.

Nous verrons ensuite quel est le protocole de communication entre le prototype et la station SUN. Ce protocole est utilisé lorsque le système expert fonctionne pour qu'il puisse obtenir des renseignements sur toute ou une partie de l'image (attributs).

Lorsque tout ceci fonctionnera, il sera possible de définir des règles sur la segmentation de bas niveau pour modifier les régions et contours obtenus par des outils classiques.  
Nous discuterons ces règles en fonction des résultats obtenus.

### III Représentation symbolique des contours des objets

Nous avons décidé de représenter une ligne sous la forme d'une suite de segments de droite.  
Ces segments de droite sont obtenus par approximation polygonale sur le contour considéré.

La structure de données adoptée pour représenter une ligne est la même que celle adoptée pour représenter les frontières des régions ( voir le rapport E.N.S.E.E.I.H.T. VII Représentations symboliques des régions ) .

#### III-1 Structures de données LISP

##### a) Structure de donnée associée aux points image

Un point est représenté par une S-expression:

(ligne . colonne)

##### b) Structure de données associée aux segments

Un segment est représenté par un couple de 2 points:

((x1 . y1) (x2 . y2))

##### c) Structure de donnée associée aux segments de droite d'une ligne

C'est une liste de segments:

```
(  
  ((x1 . y1) (x2 . y2))  
  ((x~ . y~) (x# . y#))  
  .  
  .  
  ((xn . yn) (xm . ym))  
)
```

d) Structure de données associée aux lignes d'un image

A chaque ligne  $l$  on associe un atome  $g_i$ :

. La valeur de l'atome contient le numéro de la ligne

. La P-liste de l'atome contient toutes les caractéristiques de la ligne qu'il représente.

Les champs de la P-liste sont pour le moment:

- 'Contour: contient le suite de segments de droites représentant la ligne.

- 'Longueur: contient la longueur de la ligne  $l$  en nombre de points.

- 'Gradient: contient la moyenne du module du gradient de la ligne.

Remarque:

Si le module du gradient est élevé alors la probabilité que la ligne considérée soit réellement une ligne est élevée.

- 'Fermé: permet de savoir si la ligne est fermée ou non.

REMARQUES:

. Les champs Gradient et Contour doivent être calculés par le prototype.

. Les champs Longueur et Fermé sont calculés par le programme LISP.

. La liste n'est sûrement pas complète!  
( Régions à droite et à gauche de la ligne  $l$ , direction de la ligne  $l$  ... ).  
Des champs pourront être très facilement ajoutés.

e) Représentation de toutes les lignes d'une image

C'est une liste de lignes.

$ll = (g_i \ g_{i+1} \ .. \ g_n)$

### III-2 Principaux algorithmes

Ces algorithmes vont permettre de compléter les contours des objets.

Ils vont considérer les propriétés bas niveau des contours de objets (i.e: proximité, position relative ).

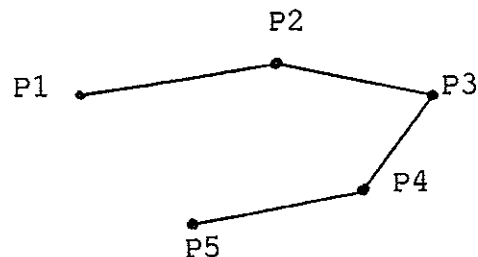
#### a) ligne fermée?

Teste si une ligne boucle sur elle même.

Ceci est utile car lorsqu'une ligne boucle sur elle même c'est qu'elle est complète.

Soit Succ(point) le sommet suivant du sommet point dans la liste des sommets de l'approximation polygonale.

exemple:



Succ (P1) = P2  
Succ (P5) = Nil

#### Algorithme

Premier ← Premier point de la ligne

Suivant ← Succ (premier)

Boucle

    si premier = suivant Alors . La ligne boucle  
  . Sortir

FinSi

    si suivant = Nil Alors . La ligne ne boucle pas  
  . Sortir

FinSi

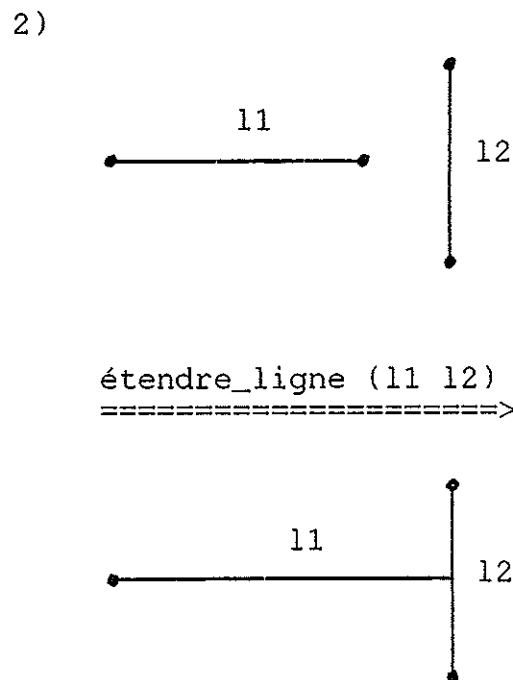
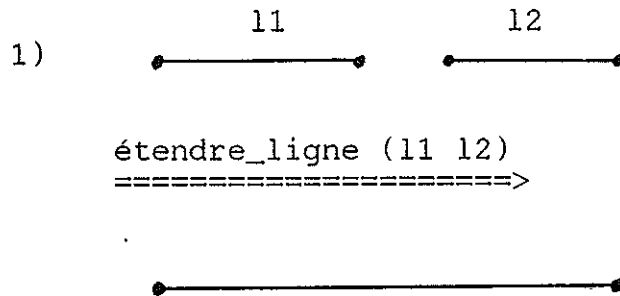
    Suivant ← Succ(Suivant)

FinBoucle

b) Etendre\_ligne

Étend la ligne (l1,l2) , étend la ligne l1 jusqu'à la ligne l2.

Exemples:



Cet algorithme permettra de compléter les contours obtenus par des extracteurs de contours ( voir rapport E.N.S.E.E.I.H.T. )



### Algorithme

Etendre\_ligne (l1 , l2):

/\* étend la ligne l1 jusqu'à la ligne l2 \*/

Si l1 est fermée alors fin

Sinon

. p1 <- 1 extrémité de l1

. p2 <- l'autre extrémité de l1

. d1 <- distance minimal de p1 à l1

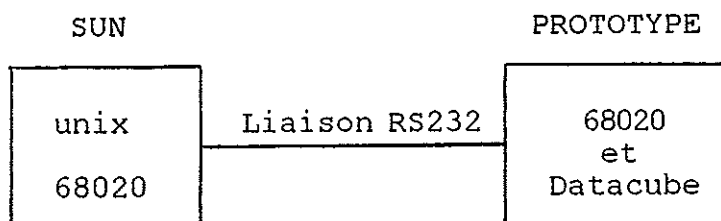
. d2 <- distance minimal de p2 à l1

. d <- min (d1 , d2)

. rajouter à l1 le segment de longueur d nécessaire  
pour prolonger l1 jusqu'à l2

#### IV Protocole de communication entre les deux machines

Nous avons à notre disposition un SUN 3/150 et un prototype avec un 68020 et des cartes de traitement d'images (Datacube). Les deux machines sont reliées via une liaison série RS232C.



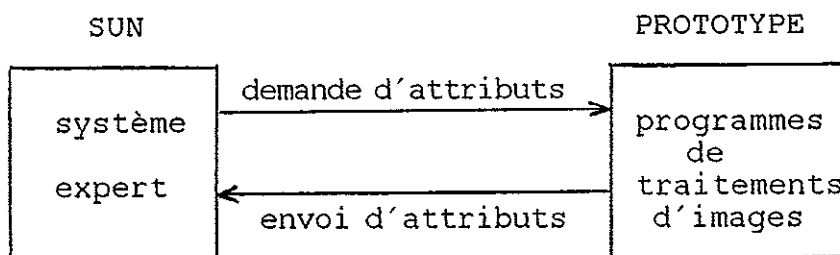
Le système expert fonctionnant sur le SUN aura besoin de renseignements sur le contenu de l'image à analyser.

Exemples :

- quelle est la moyenne de gradient d'une ligne?
- quelle est la moyenne de niveau de gris d'une région donnée.
- etc ...

Il va donc y avoir un échange d'informations entre le SUN et le prototype.

Le système expert s'exécutant sur le SUN va être le maître par rapport au prototype.

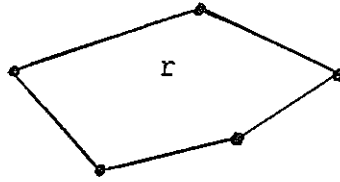


Chaque attribut aura un identificateur particulier ( numéro ).

Le système expert va envoyer un numéro (identificateur d'attribut) et une liste d'objets nécessaire au calcul de l'attribut demandé par le système expert.

Exemple:

Pour calculer la moyenne d'une région il faut envoyer le contour de la région.



Un programme de traitement d'images sera capable de calculer la moyenne d'une région déterminée par ses contours.

### AINSI

pour obtenir des attributs sur les régions et les lignes il faut envoyer le contour (de la région ou de la ligne).

Il peut bien entendu y avoir des attributs qui ne demandent aucune connaissance préalable de la partie de l'image à analyser.

## V Le système expert et les règles utilisées

Actuellement le système expert n'existe pas. C'est une simple boucle LISP prenant en compte toutes les lignes et les régions de l'image.

La version future (avec un système expert) sera beaucoup plus efficace.

### V-1 Les règles sur les régions

Nous avons vu des algorithmes de segmentation en régions (voir le rapport E.N.S.E.E.I.H.T). Ces algorithmes ne donnent pas de résultats idéaux:

- Des faces sont composées de plus d'une région  
=> fusion de 2 régions nécessaire.
- Une région englobe plus d'une face  
=> division nécessaire.

Nous savons réaliser ces opérations grâce à la représentation symbolique des régions (voir le rapport E.N.S.E.E.I.H.T.).

Voyons tout d'abord quelques règles portant sur la fusion de deux régions.

#### V-1.1 Règles de fusion de deux régions

Soit à fusionner 2 régions  $r_1$  et  $r_2$ .

Dans ce qui suit nous adoptons les notations suivantes:

$M(r)$  : la moyenne de la région  $r$

$V(r)$  : la variance de la région  $r$

$P(r)$  : le périmètre de la région  $r$

$S(r)$  : la surface de la région  $r$

$I(i, j)$ : la valeur de niveau de gris de l'image à analyser (au point  $i, j$ ).

a) Voyons tout d'abord une règle générale réalisant la fusion de deux régions.

SI r1 et r2 sont adjacentes

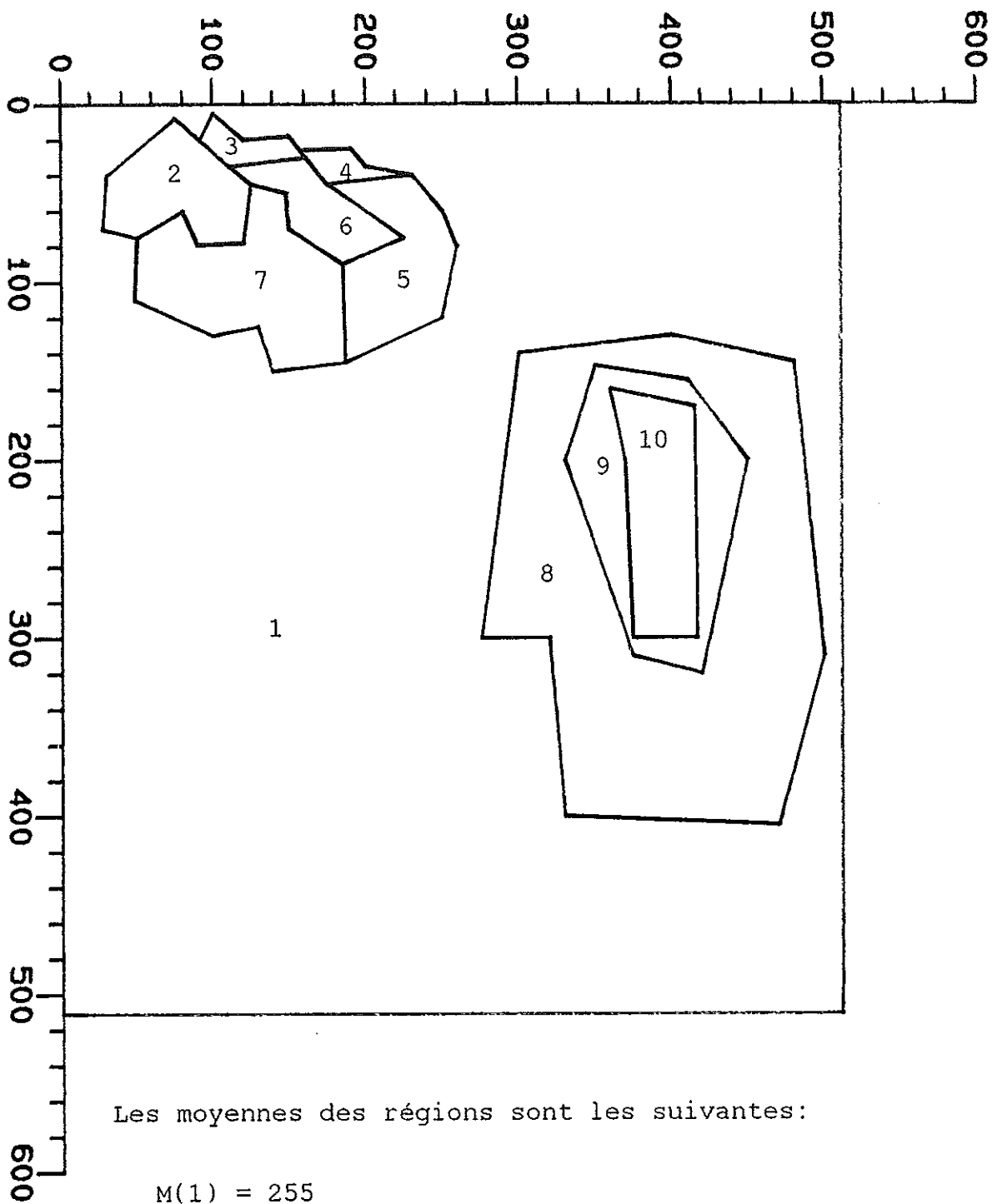
ET  $|M(r1) - M(r2)| < \text{Seuil}$

ALORS fusionner r1 et r2

Cela revient à dire que si deux régions adjacentes ont des moyennes de niveau de gris proches on peut les regrouper.

Cette règle, qui peut paraître assez évidente, permet de résoudre un bon nombre de cas.

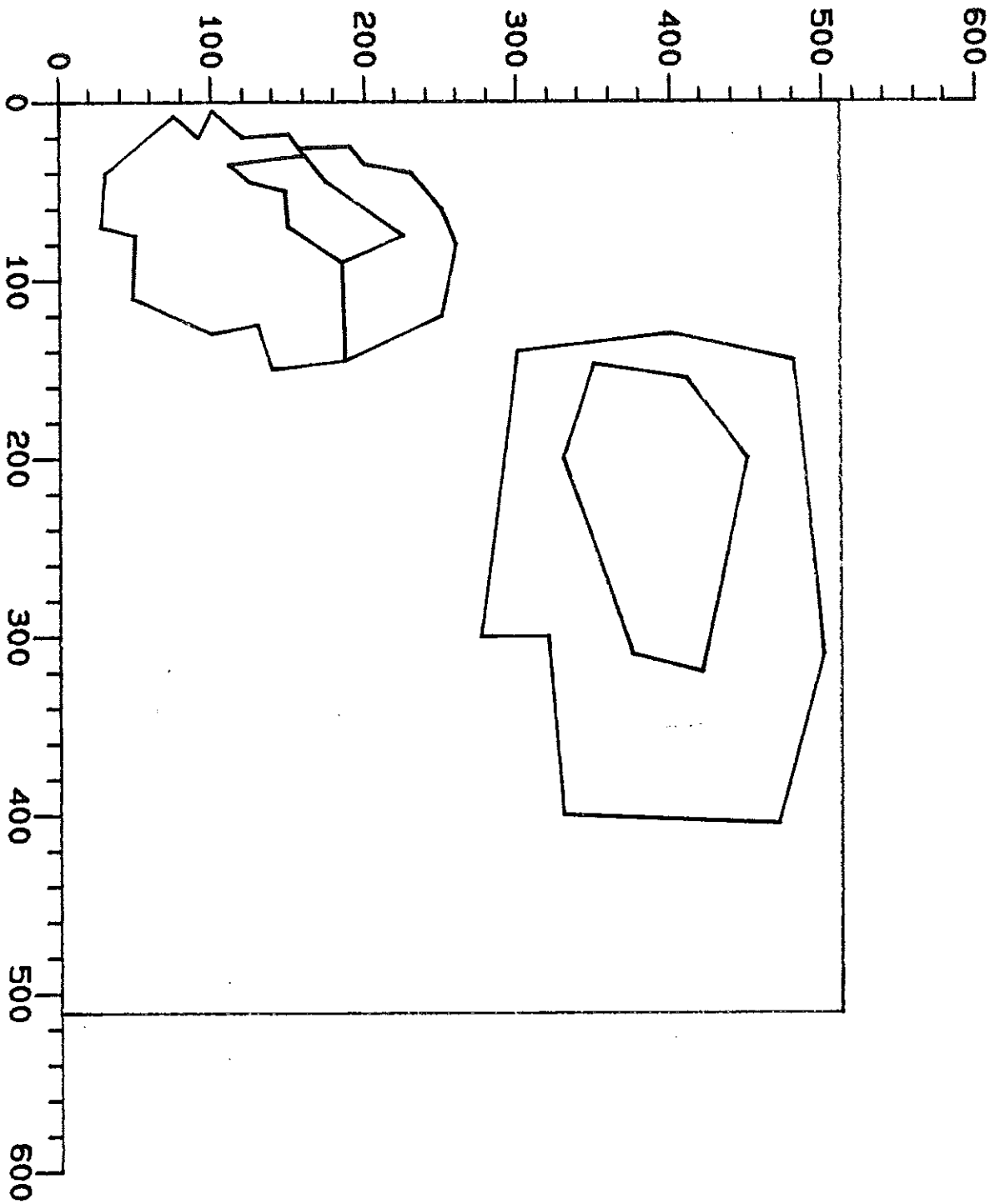
Voyons maintenant un exemple d'application sur l'image suivante:



Les moyennes des régions sont les suivantes:

- M(1) = 255
- M(2) = 30
- M(3) = 47
- M(4) = 70
- M(5) = 85
- M(6) = 120
- M(7) = 51
- M(8) = 100
- M(9) = 130
- M(10) = 145

Le résultat est le suivant:



En prenant un seuil de 20.

b) Nous venons de considérer une règle prenant en compte la proximité des moyennes de niveaux de gris des 2 régions à regrouper.

Mais il est également possible de considérer la Variance de la région correspondant à la fusion des 2 régions de départ [GAG 85].

La règle s'exprime ainsi:

SI r1 et r2 sont adjacentes

ET  $V(r1 \cup r2) < \text{Seuil}$

ALORS Fusionner r1 et r2

Avec:

$V(r)$  la variance des valeurs de points sur l'image initiale:

$$V(r) = \frac{\sum_{(i,j) \in r} (I(i,j) - M(r))^2}{\text{Card}(r)}$$

Cette règle permet de regrouper des régions qui ont des moyennes différentes mais qui une fois regroupées ont une variance faible. Ce qui résout le problème des faces qui ont plusieurs zones différentes.

c) Une autre règle réalisant la même fonction s'exprime ainsi:

SI r1 et r2 sont adjacentes

ET  $\text{Max}(r1 \cup r2) - \text{Min}(r1 \cup r2) < \text{Seuil}$

ALORS fusionner r1 et r2

Avec:

$\text{Max}(r)$  étant la valeur du point appartenant à r qui a le niveau de gris le plus élevé.

et

$\text{Min}(r)$  étant la valeur du point appartenant à r qui a le niveau de gris le plus faible.

Si on prend un seuil très faible (environ 5) on va ainsi regrouper les régions ayant le même niveau de gris.



d) Le problème principal de certains segmenteurs en régions est qu'ils génèrent beaucoup de petites régions dues au bruit résiduel du capteur (caméra).

Si l'on applique la règle suivante:

SI  $r_1$  et  $r_2$  sont adjacentes

ET  $|M(r_1) - M(r_2)| < \text{Seuil } 1$

ET  $((S(r_1) < \text{Seuil } 2) \text{ OU } (S(r_2) < \text{Seuil } 2))$

ALORS fusionner  $r_1$  et  $r_2$

Le Seuil 1 doit être plus grand que dans la règle exposée en a). En effet cette nouvelle règle sera appliquée si la règle a) échoue. Généralement la petite région aura un niveau de gris moyen sensiblement différent de celui de la grande région.

e) Nous n'avons pour le moment considéré que les données de l'image initiale. Nous allons travailler avec l'image gradient de l'image à analyser.

SI  $r_1$  et  $r_2$  sont adjacentes

ET Moyenne du gradient sur la frontière  
entre  $r_1$  et  $r_2 < \text{Seuil}$

ALORS fusionner  $r_1$  et  $r_2$

Si la frontière a un gradient faible c'est qu'elle ne correspond pas à une arête d'objet, il paraît alors tout naturel de fusionner les deux régions.

f) Nous allons considérer maintenant une règle portant sur la forme des régions. [HAC 89] et [BAL 82]

**IMPORTANT:** La règle suivante ne sera appliquée que si toutes les autres échouent.

$$\text{SI } \frac{(S(r1))^{1/2} + (S(r2))^{1/2}}{P(r1) \cdot P(r2)} < \text{Seuil}$$

ALORS Fusionner r1 et r2

On fusionne une région avec une autre si une des deux est étroite (très heuristique).

Ce point peut être assez discutable c'est pour cette raison que l'on essaye cette règle que si les autres règles ne sont pas déclenchées.

### V-1.2 Règles de division d'une région en plusieurs

Nous avons vu jusqu'à présent les règles portant sur la fusion de deux régions. Voyons maintenant quelles règles peuvent être appliquées pour diviser une région en plusieurs.

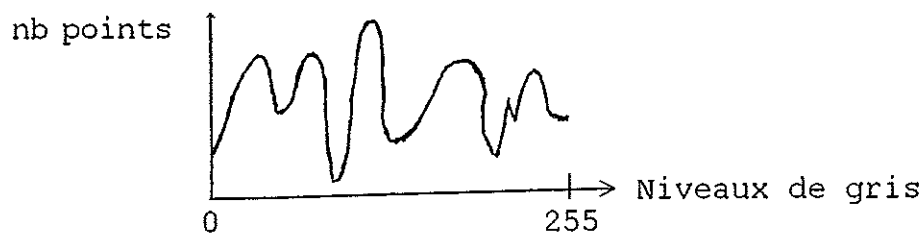
Cette division est utile car dans certains cas les segmenteurs en régions regroupent deux faces (ou plusieurs) d'un objet dans une même région.

Les règles de division utilisent la notion d'histogramme en niveaux de gris. Voyons donc tout d'abord comment l'on peut utiliser ces histogrammes.

#### V-1.2.1 Histogramme en niveaux de gris d'une région

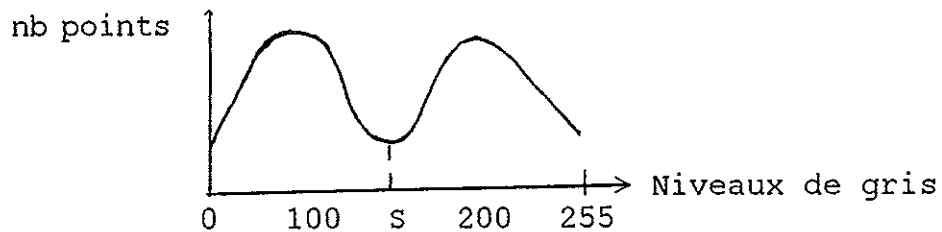
L'histogramme en niveaux de gris d'une région représente la dispersion des niveaux de gris des points de la région.

Exemple:



On voit tout de suite dans cette définition que si l'histogramme est bimodal ( histogramme à 2 bosses ), c'est que deux groupes de niveaux de gris se détachent. On est donc très probablement en présence de deux zones distinctes dans la région.

Exemple:

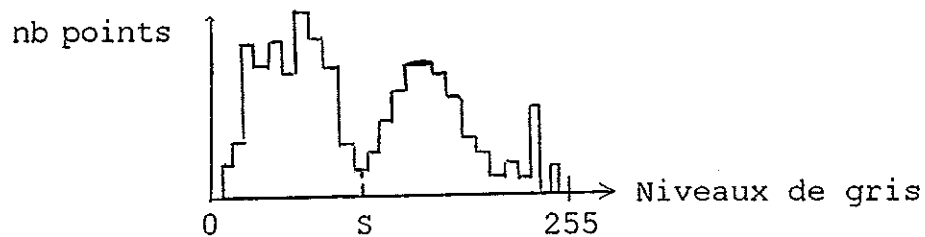


On pourra donc séparer la région de départ selon cet histogramme. Tous les points de la region  $< S$  seront affectés à une région. Les autres point seront affectés à une autre région.

L'étude de cas réels montre que ce n'est pas si simple :

- L'histogramme peut ne pas être lisse.

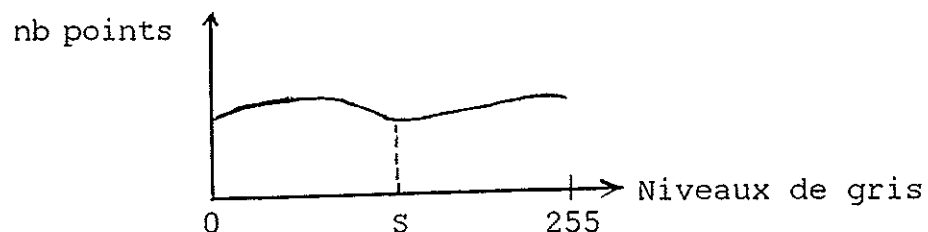
Exemple:



La détermination du seuil  $S$  peut donc poser un problème.

- Les deux bosses de l'histogramme ne sont peut-être pas très nettes.

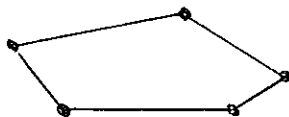
Exemple:



- Si on divise une région selon son histogramme de bimodalité, rien n'assure que seules deux régions vont être trouvées.

Exemple:

soit la région:



si on découpe cette région selon son histogramme de bimodalité, on peut obtenir comme résultats:



Ceci est dû au fait que les points ne sont peut-être pas parfaitement répartis dans la région.

Il est donc possible, si l'on divise une région selon l'histogramme de bimodalité, de commettre des erreurs (diviser en plus de deux régions).

Ce n'est pas très grave puisqu'il y aura toujours la possibilité d'appliquer des règles de fusion sur les régions générées.

Voyons maintenant quelles sont les règles que l'on peut utiliser pour diviser une région en plusieurs.

#### V-1.2.2 Les règles de division de région

Soit à diviser la région  $r$ .

a) La première règle peut paraître assez évidente après l'exposé du V-1.2.1 .

Voici cette règle:

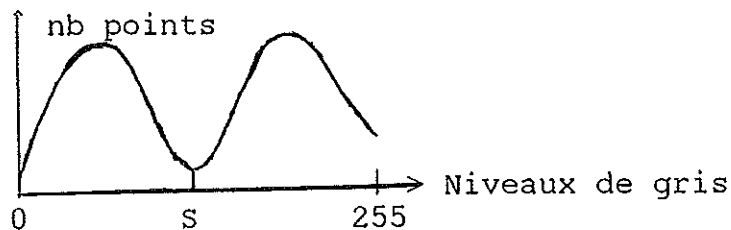
SI L'histogramme de  $r$  est bimodal

ALORS Diviser  $r$  selon cet histogramme

L'application de cette règle va permettre de faire ressortir un bon nombre de faces des objets (sachant que plusieurs faces pouvaient être présentes dans une même région).

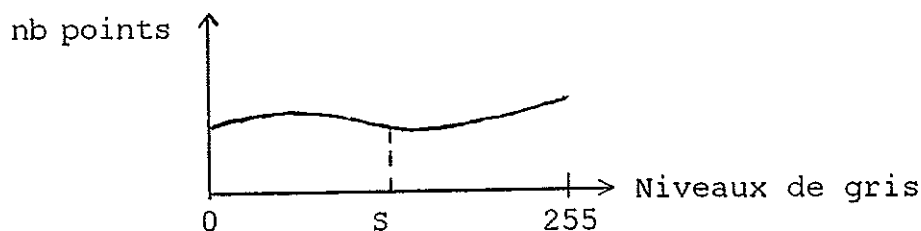
Le problème de cette règle c'est qu'elle ne prend pas en compte le fait que les bosses de l'histogramme apparaissent nettement ou pas.

Si on a



On est sûr de pouvoir diviser.

Par contre si on a



Doit-on réellement diviser?

b) Pour résoudre ce problème nous avons la règle suivante.  
[NAZ 84]

SI L'histogramme de r est bimodal

ET Les bosses de l'histogramme sont grandes

ALORS Diviser r selon cet histogramme.

c) Il peut être intéressant de ne pas diviser des régions trop petites car elles seront regroupées par les règles de fusion de régions.

On obtient ainsi la règle suivante:

SI  $S(r) > \text{Seuil}$  (  $S(r)$  étant la surface de r )

ET L'histogramme de r est bimodal

ET Les bosses de l'histogramme sont grandes

ALORS Diviser r selon cet histogramme.

d) On peut considérer un autre style de règles prenant en compte la variance de la région que l'on veut diviser.

Soit la règle:

SI  $V(r) > \text{Seuil}$  (  $V(r)$  la variance des points  
de la région )  
ET L'histogramme est bimodal  
ALORS Diviser  $r$  selon cet histogramme

Dans le cas où la variance de  $r$  est forte, il n'est pas nécessaire de prendre en compte la hauteur des bosses de l'histogramme car la variance assure de la présence de plusieurs zones distinctes dans la région.

## V-2 Les règles sur les lignes

Avant d'exposer ces règles, voyons de quelle façon peuvent être détectés les contours d'objets.

Nous verrons ensuite quelles règles appliquer pour améliorer les arêtes obtenues par les extracteurs de contours.

### V-2.1 Les extracteurs de contours

Les extracteurs de contours déterminent les changements brusques de contraste pour essayer de mettre en évidence les arêtes des objets.

En fait ils calculent l'image gradient de l'image à analyser. Sachant que le gradient en un point  $p$  correspond à la différence de niveaux de gris entre ses 8 voisins.

Le gradient d'une image correspond à l'application d'un masque de convolution sur l'image de départ:

Soit  $h$  un masque  $(2m+1) \times (2m+1)$ ,  
 $f$  l'image initiale  
et  $g$  l'image gradient

$$g(x,y) = \sum_{i=-m}^{i=+m} \sum_{j=-m}^{j=+m} f(x+i,y+j) \cdot h(i,j)$$

Exemples de masques:

. Sobel diagonal :

$$h_1 = \begin{vmatrix} 2 & 1 & 0 \\ 1 & 0 & -1 \\ 0 & -1 & -2 \end{vmatrix} \quad \text{et} \quad h_2 = \begin{vmatrix} 0 & 1 & 2 \\ -1 & 0 & 1 \\ -2 & -1 & 0 \end{vmatrix}$$

. Sobel horizontal-vertical :

$$h_1 = \begin{vmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{vmatrix} \quad \text{et} \quad h_2 = \begin{vmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{vmatrix}$$

Si le gradient en un point  $p$  est fort, c'est que ce point a de grandes chances d'appartenir à un contour.

Par contre si le gradient en  $p$  est faible, le point n'est probablement pas sur un contour.

On voit ainsi apparaître la notion de *seuil de gradient*.

A partir de quelle valeur peut on être sûr d'être en présence d'un point appartenant à un contour?

C'est en fait un problème très délicat car un seuil trop bas ferait apparaître des contours qui ne correspondent pas à des arêtes d'objets.

Par contre un seuil trop haut éliminerait certains contours.

On voit donc que de toutes façons il y aura des problèmes:

- Des arêtes seront absentes,
- Des arêtes seront en trop.

Voyons maintenant quelles règles peuvent permettre de résoudre ces problèmes.

## V-2.2 Les règles proprement dites

Soit à prolonger l1 jusqu'à l2.

a) La règle suivante peut paraître assez évidente:

SI l1 est non fermée

ET la distance entre l1 et l2 est faible

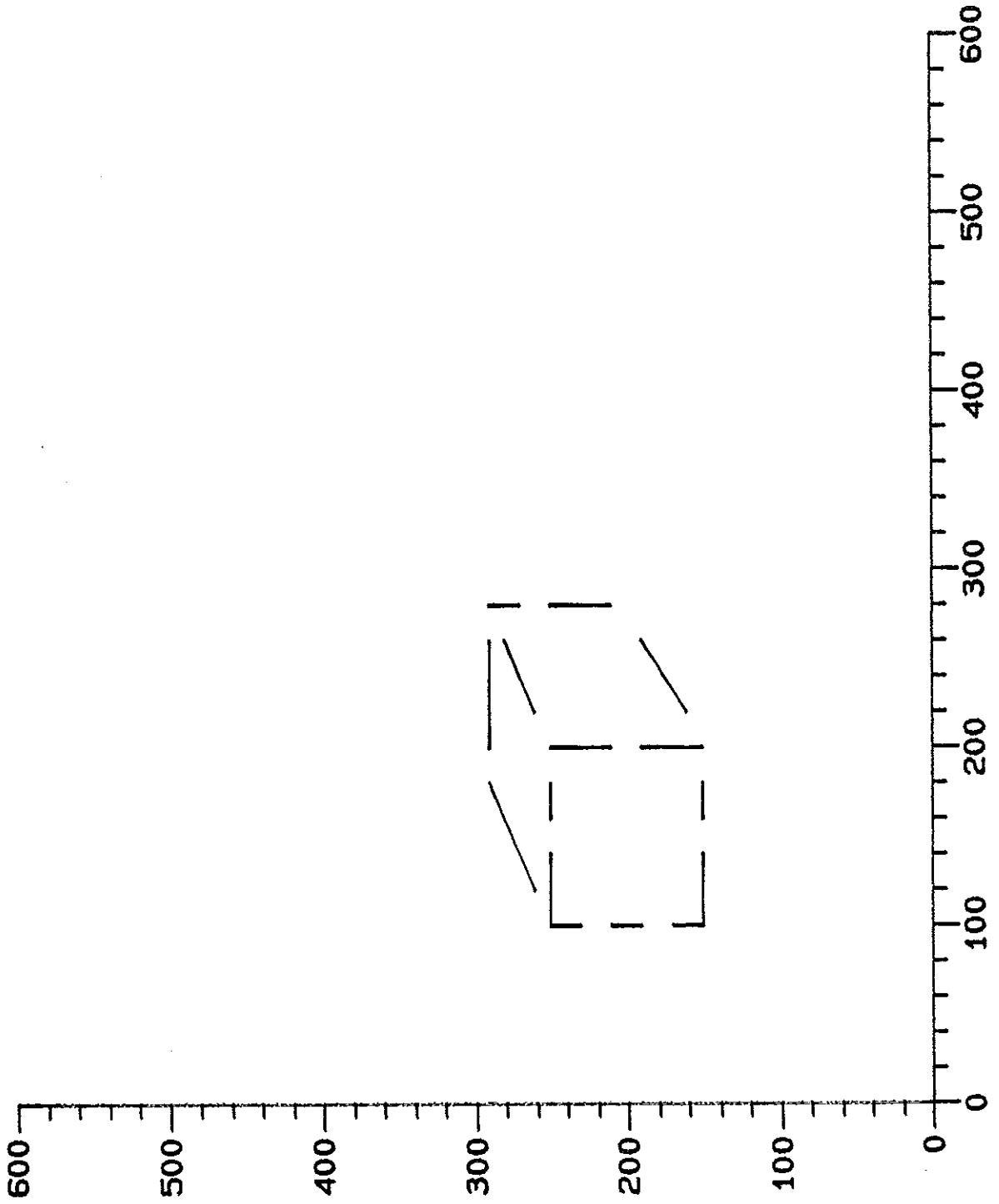
ALORS Prolonger l1 jusqu'à l2 avec le segment  
de longueur minimale

Cette règle prend en compte la notion de proximité des lignes.

Voyons quels sont les résultats obtenus en appliquant cette règle.

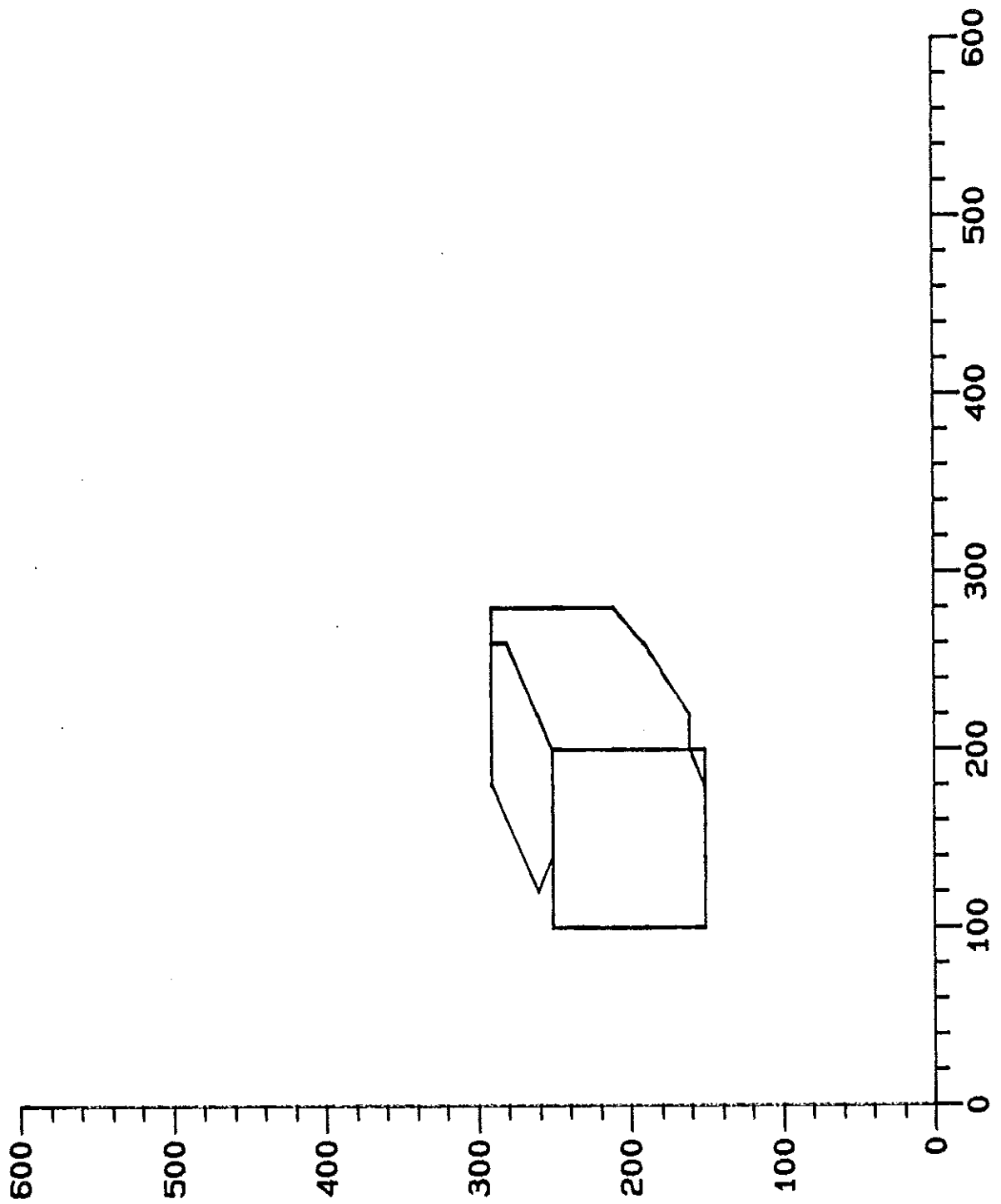


Soit l'image de départ correspondant à un cube dont on connaît partiellement les arêtes:



un cube incomplet

Après application de la règle a) on obtient l'image suivante:



cube complete ( critere : distance minimale )

Les résultats obtenus sont très encourageants mais il y a encore des problèmes au niveau du coin du cube.

Ces problèmes sont dûs au fait que l'on n'a pas tenu compte de la direction des droites que l'on prolonge.

Ceci nous amène donc à la règles suivante:

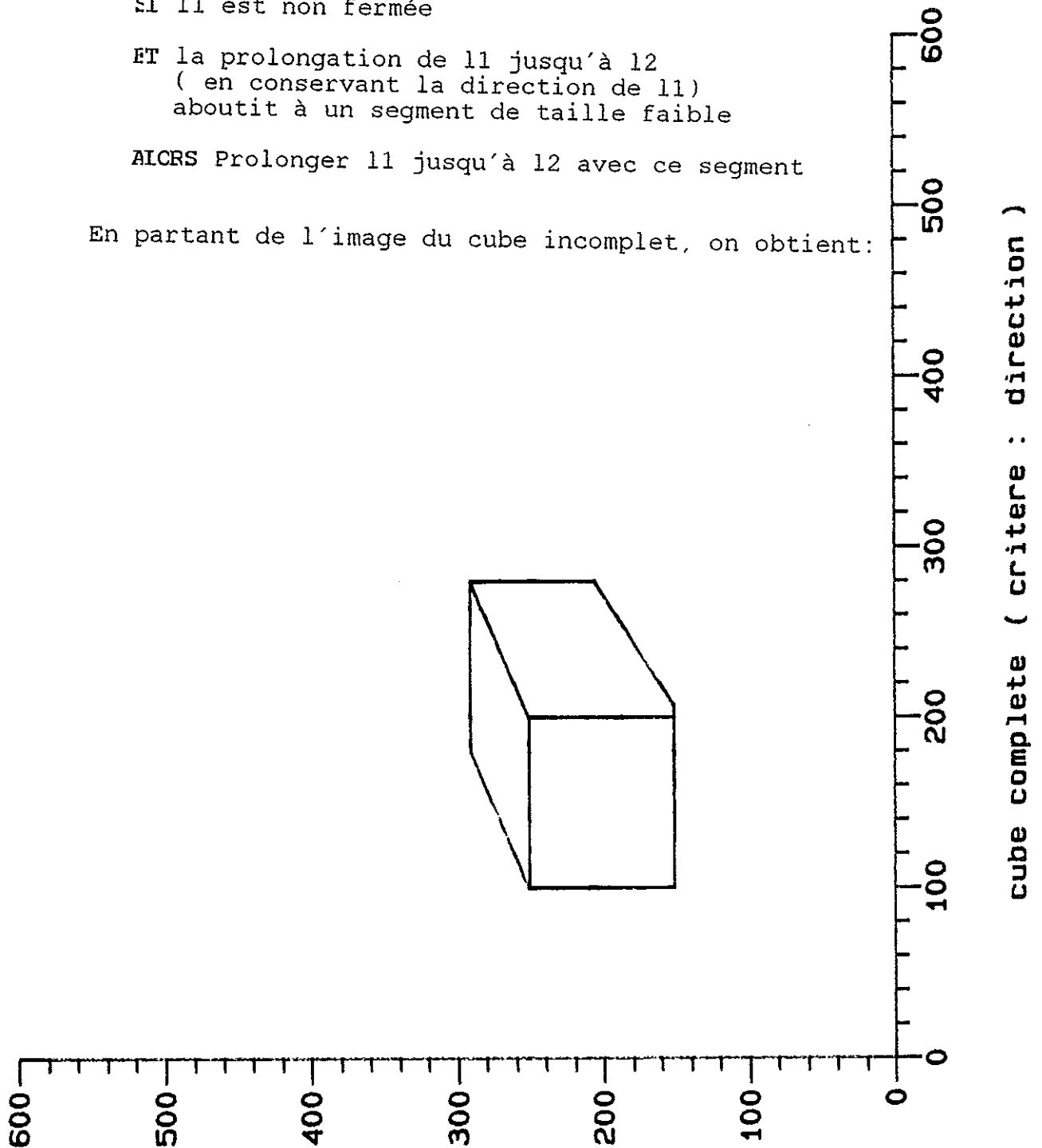
b)

SI l1 est non fermée

ET la prolongation de l1 jusqu'à l2  
( en conservant la direction de l1 )  
aboutit à un segment de taille faible

ALCRS Prolonger l1 jusqu'à l2 avec ce segment

En partant de l'image du cube incomplet, on obtient:



Les résultats sont nettement meilleurs que dans le cas de la règle a). Il peut encore y avoir des problèmes dus à l'ordre d'analyse des lignes.

Si les deux règles précédentes n'aboutissent pas, il faut considérer d'autres règles:

c) La règle suivante permet de prolonger une ligne sans avoir besoin de la connaissance des lignes voisines:

SI l1 est non fermée

ET le gradient est fort dans la direction de l1

ALORS Prolonger l1 dans sa direction tant que le gradient reste fort.

Si le gradient est supérieur à un certain seuil, il n'est pas absurde de penser que cette droite se continue (même si la prolongation ne rejoint aucune autre ligne!!) .

AINSI peut-être, après application de cette règle, les règles précédentes pourront être déclenchées.

d) La règle suivante va permettre de supprimer les lignes de petites tailles:

SI l1 est petite

ET la moyenne du gradient sur l1 est faible

ALORS supprimer la ligne l1.

Il faut faire attention à 2 choses:

. Si le gradient de la ligne est fort on ne supprime pas la ligne car elle doit correspondre à une réalité.

. On n'applique cette règle que si toutes les autres non pas été déclenchées car le fait de supprimer une ligne fait perdre de l'information!

## VI Conclusion

Participant à la partie vision du projet R.E.V.E.R.I.E, j'ai développé une méthode originale permettant d'obtenir des contours et des faces suffisamment précis des objets se trouvant dans une image.

C'était une étape obligée pour entreprendre la reconnaissance d'objets dans de bonnes conditions.

Mais le projet R.E.V.E.R.I.E. n'est pas encore terminé.

Plusieurs modules doivent être réalisés:

. Un module de reconnaissance d'objets (avec possibilité d'apprentissage).

Ce module va permettre d'identifier des objets dans une scène.

. Un module de planification de tâches pour commander un robot:

En partant d'un but à atteindre, quel objet faut-il utiliser pour atteindre ce but et comment l'utiliser ?

Tout ceci implique une description fonctionnelle des objets en fonction de leurs géométries (longs, courts, etc...), de leurs propriétés physiques (durs, mous, solides, cassables, etc...), de leurs poids (lourds, léger), etc...

Bibliographie:

[NAZ 84] :

"LOW LEVEL IMAGE SEGMENTATION: AN EXPERT SYSTEM."  
IEEE Transactions on pattern analysis and machine  
intelligence, VOL. PAMI-6 No.5, Sept 84.

[BAL 82] :

"Computer Vision"  
Dana H. Ballard  
Christopher M. Brown  
PRENTICE-HALL 1982.

[MIN 82] :

"Segmentation of images using minimum spanning trees"  
Minsoo suk and Tai Hoon Cho  
PROCEEDING of SPIE  
VOLUME 397  
Applications of Digital Image Processing  
April 19-22, 1983  
Geneva Switzeland.

[GAG 85] :

"A hierarchical segmentation algorithm"  
by  
André Gagalowicz  
Olivier Monga

INRIA

56 / SPIE Vol 595 Computer Vision for robot (1985)  
(2-6 Décembre 85 Cannes, France)

[HAC 89] :

"Segmentation using intensity and range data"  
Jay K. Hackett  
Mubarak Shah

SPIE Optical Engineering June 1989  
Vol 28 No 6 page 667

[BRA 85] :

"Describing Surface"

Michael Brady, Jean Ponce, Alan Yuille,  
and Haro Asada.

Artificial Intelligence Laboratory,  
Massachusetts Institute of Technology, CAMBRIDGE.

Computer Vision, Graphics and image processing  
Vol 32 Number 31 October 1985 page 1