SEGMENTATION et RECONNAISSANCE

utilisation **d'informations a priori** pour segmenter et reconnaître

Modèles paramétriques

transformée de Hough

MODÈLES PARAMÉTRIQUES TRANSFORMATION DE HOUGH

Principe :
- des méthodes paramétriques

Les contours des objets sont représentés par des courbes décrites à l'aide de fonctions simples (droites, arcs de cercles, …) modélisées par leurs paramètres.

- de la transformée de Hough

On transforme l'image dans l'espace des paramètres et on identifie la courbe dans cet espace.

TRANSFORMATION DE HOUGH

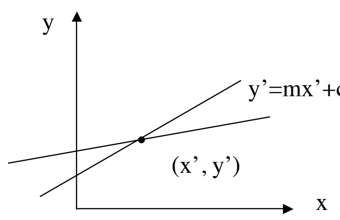
Principe

(Explication dans le cas de la recherche de droites dans une image)

représentation d'une droite : $y=mx+c \quad -\infty<m<+\infty$

Pour un point (x', y') donné, toutes les droites passant par ce point ont la forme

y'=mx'+c pour différentes valeurs de m et de c

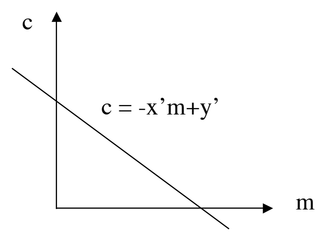


TRANSFORMATION DE HOUGH : principe

Cette équation de la droite, y'=mx'+c, peut s'écrire **c = -x'm+y'**

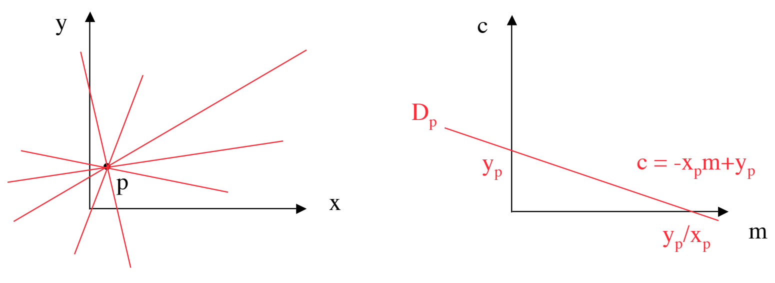
en considérant que x' et y' sont constants et que m et c varient.

C'est l'équation d'une droite dans un espace (m,c)



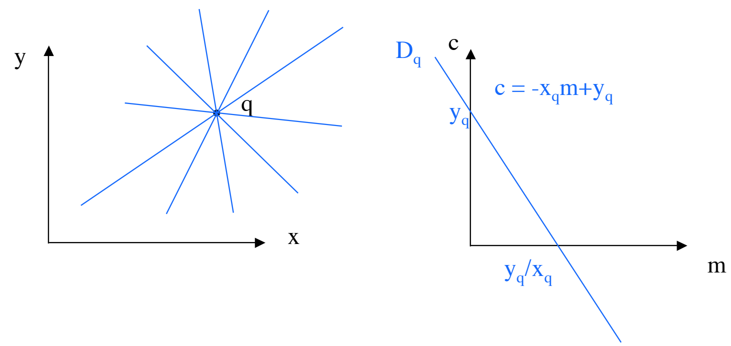
TRANSFORMATION DE HOUGH : principe

Pour chaque point p, toutes les droites passant par ce point correspondent à une seule droite D_p dans l'espace (m,c)



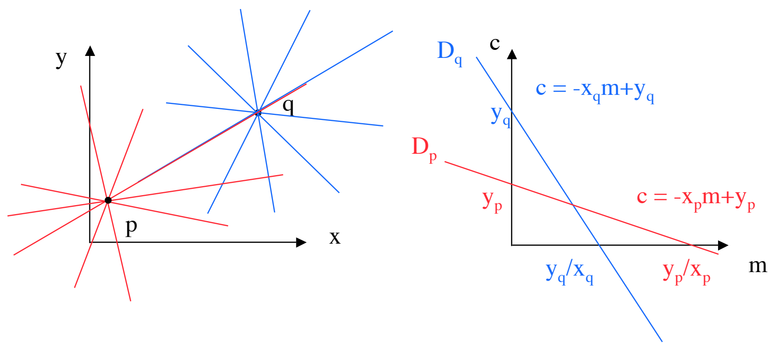
TRANSFORMATION DE HOUGH : principe

Pour chaque point q, toutes les droites passant par ce point correspondent à une seule droite D_q dans l'espace (m,c)



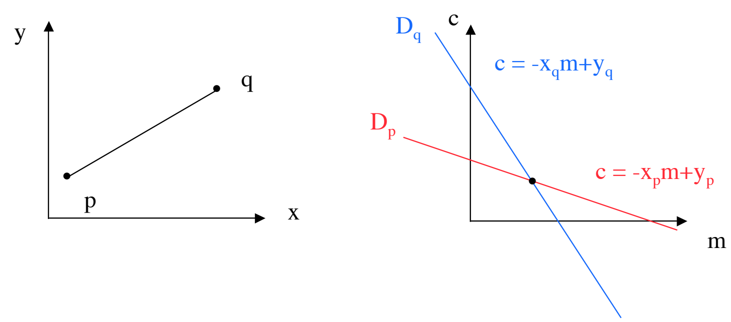
TRANSFORMATION DE HOUGH : principe

Ces deux faisceaux de droites dans l'espace (x, y) ont en commun la droite qui relie les points p et q.
A cette droite (p q) correspond l'intersection des 2 droites D_p et D_q représentant p et q dans l'espace (m, c)



TRANSFORMATION DE HOUGH : principe

Ces deux faisceaux de droites dans l'espace (x, y) ont en commun la droite qui relie les points p et q.
À cette droite (p q) correspond l'intersection des 2 droites D_p et D_q représentant p et q dans l'espace (m, c)

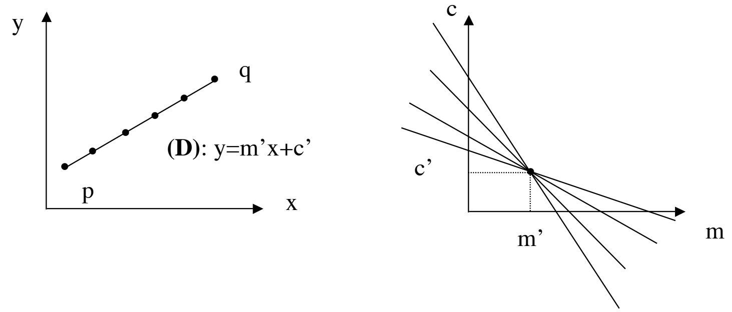


TRANSFORMATION DE HOUGH : principe

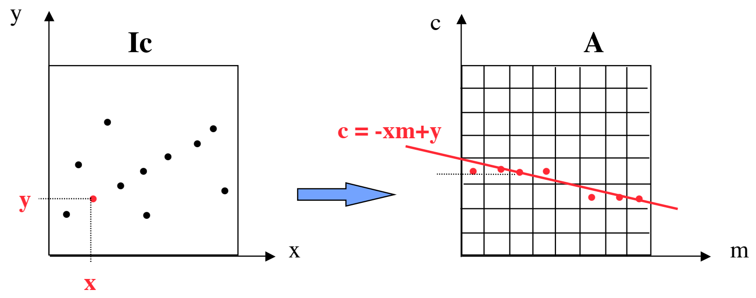
Tous les points situés sur la même droite D dans l'espace (x,y) sont représentés par des droites qui passent toutes par le même point dans l'espace (m,c).

Ce point (m', c') donne les paramètres recherchés de l'équation de la droite D :

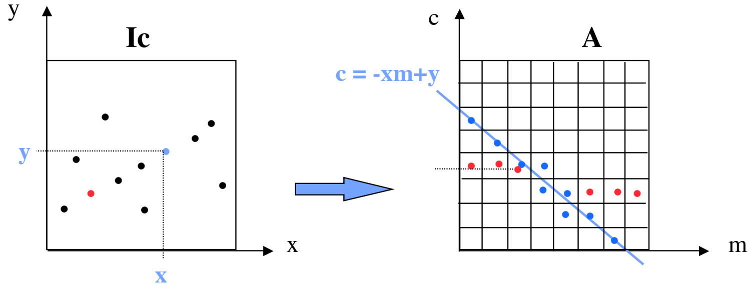
$$y = m'x+c'$$



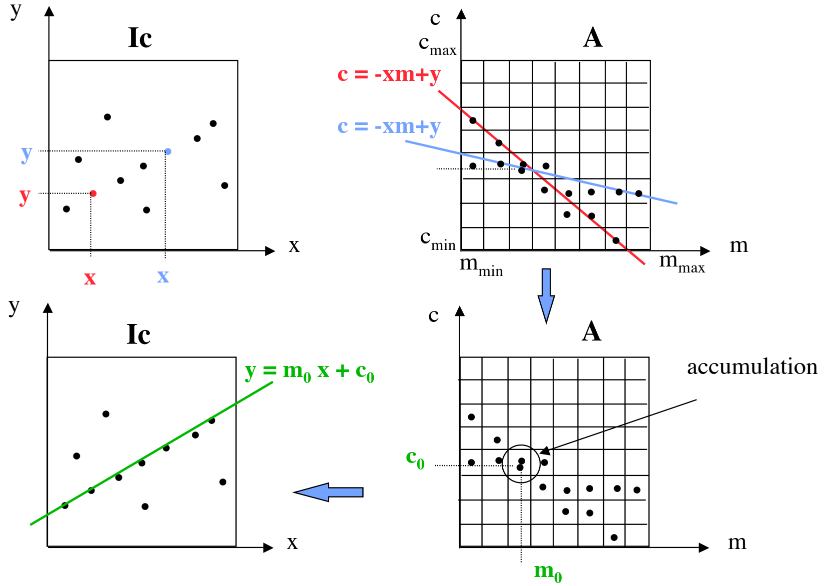
TRANSFORMATION DE HOUGH : algorithme



TRANSFORMATION DE HOUGH : algorithme



TRANSFORMATION DE HOUGH : algorithme



TRANSFORMATION DE HOUGH : algorithme

Algorithme

(On traite une image binaire Ic, résultant par exemple d'une détection de contours)

- Partitionner l'espace (m,c) sous forme d'un tableau A à 2 dimensions.

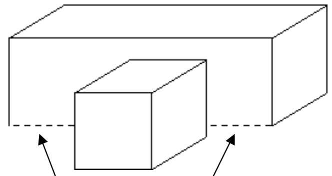
- Initialiser le tableau A à 0

- Pour chaque pixel (x',y') correspondant à un point contour dans l'image Ic, incrémenter toutes les entrées (m,c) de A satisfaisant c= -x'm+y'

- Le valeurs élevées de A, par exemple en (m_0 , c_0) correspondent à l'équation d'une droite y= m_0 x + c_0 pour laquelle beaucoup de pixels de l'image Ic ont voté.

TRANSFORMATION DE HOUGH : évaluation

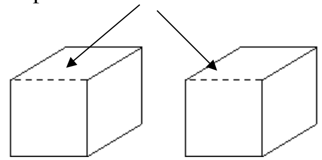
Avantage



les pixels non continus d'une même ligne contribuent à la détermination de l'équation

Inconvénient

Des objets alignés risquent de voter pour la même droite



TRANSFORMATION DE HOUGH : réalisation

Inconvénients de l'approche présentée :

1- fournit l'équation d'une droite infinie au lieu d'un segment de droite

2- problème de discrétisation de l'espace (m,c) :

dimensions de $A(m_{min}: m_{max} , c_{min} : c_{max})$

Si la droite cherchée est verticale => m —> ∞

1 => traitement supplémentaire : parcourir la droite et déterminer le premier et le dernier point effectivement présents

2 = > utiliser une autre représentation d'une droite

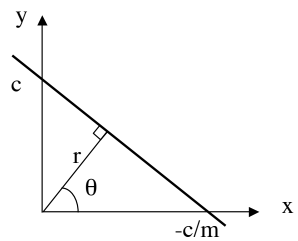
TRANSFORMATION DE HOUGH : réalisation

<u>Autre représentation d'une droite</u>

$$x \cos\theta + y \sin\theta = r$$

$0 < r <$ diagonale de l'image

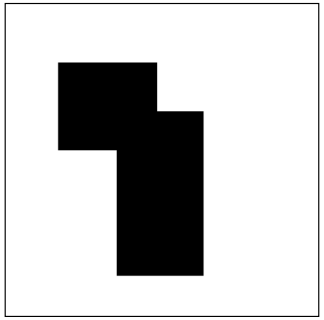
$0 < \theta < 2\pi$

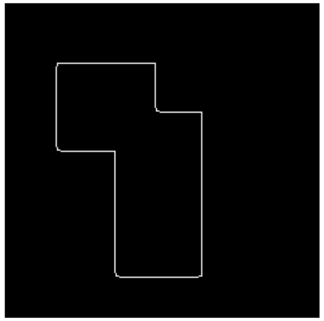


<u>Même principe</u>

<u>espace image (x , y)</u>		<u>espace des paramètres (r, θ)</u>
un point	—>	une courbe
les points d'une même droite	—>	courbes ayant un point d'intersection commun

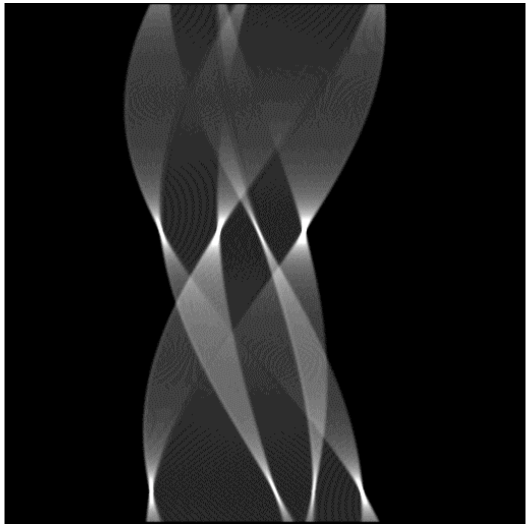
TRANSFORMATION DE HOUGH : exemple (HIPR)



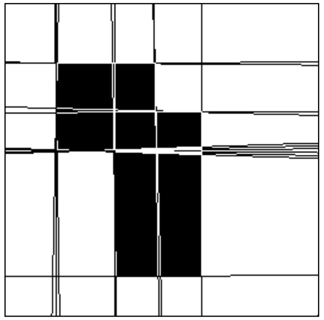


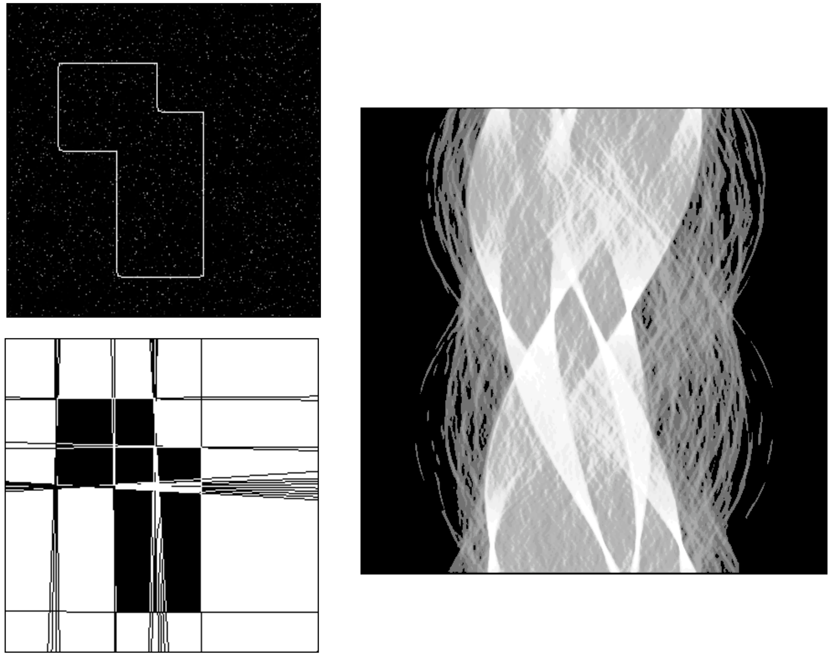
TRANSFORMATION DE HOUGH : exemple (HIPR)

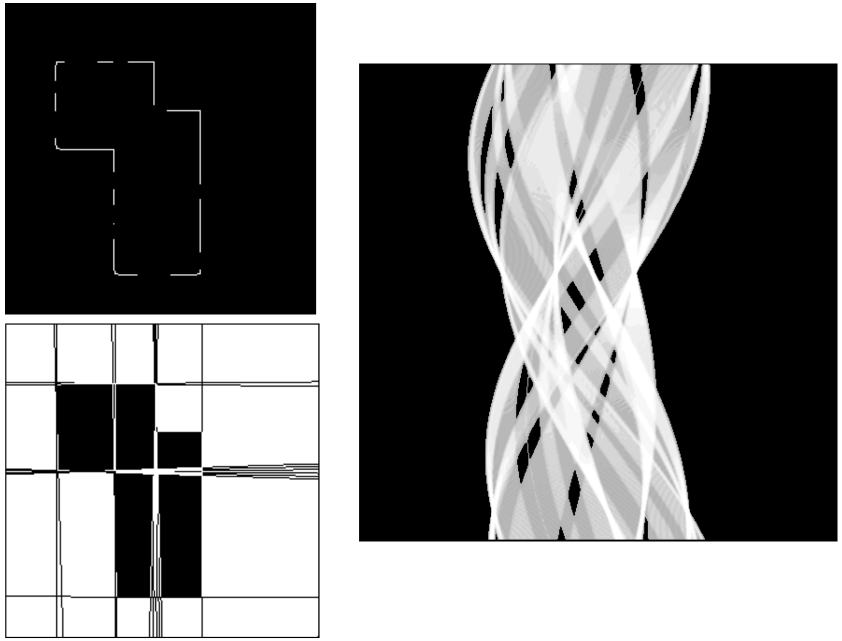
TRANSFORMATION DE HOUGH : exemple (HIPR)



TRANSFORMATION DE HOUGH : exemple (extrait de HIPR)







TRANSFORMATION DE HOUGH : extension

Détection de formes circulaires

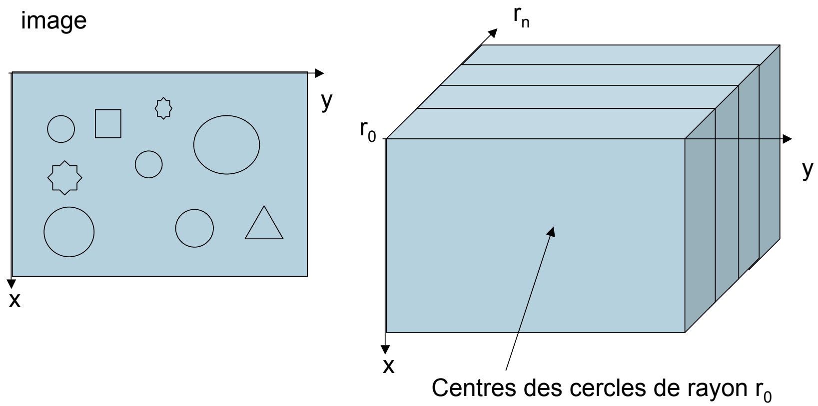
$$(x - a)^2 + (y - b)^2 = r^2 \Longrightarrow \text{3 paramètres : a, b, r.}$$

(x,y) dans l'espace image —> surface dans l'espace (a , b , r)

=> tableau 3D : A(a,b,r),
Pour un point contour (x,y), on incrémente tous les points (a,b,r) qui satisfont l'équation du cercle : $(x - a)^2 + (y - b)^2 = r^2$

On sélectionne les maxima de A

HOUGH : Détection de formes circulaires



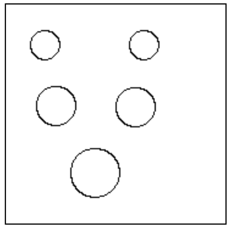
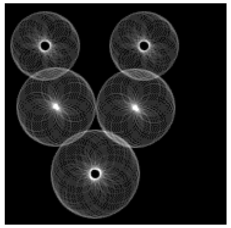
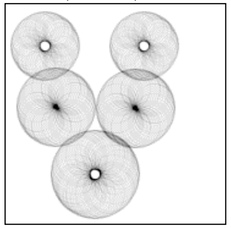
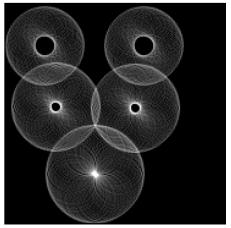
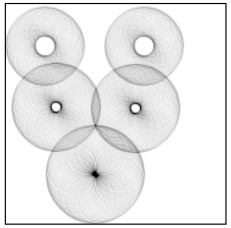
Chaque point contour (x,y) de l'image vote pour l'ensemble des centres de cercles de rayon r pouvant passer par (x,y)

TRANSFORMATION DE HOUGH : extension

rayons : 12,16 et 20

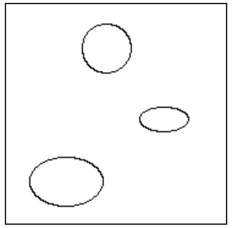
recherche des cercles de rayon 16 (+ augmentation du contraste)

(inversée)

recherche des cercles de rayon 20 (+ augmentation du contraste)

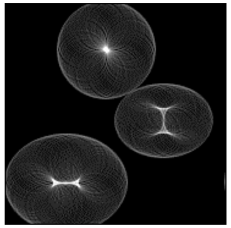
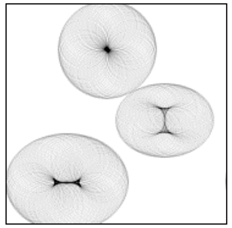
TRANSFORMATION DE HOUGH : extension



cercle de rayon 20

ellipse 20 / 40

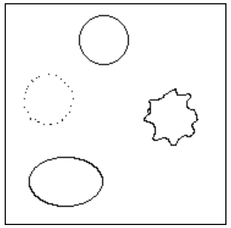
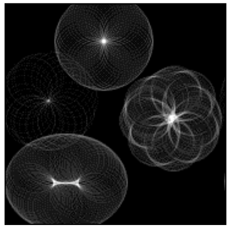
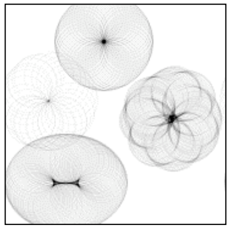
ellipse 40 / 60

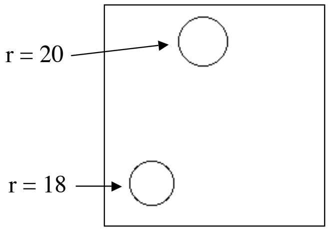
recherche d’un cercle de rayon 20

TRANSFORMATION DE HOUGH : extension

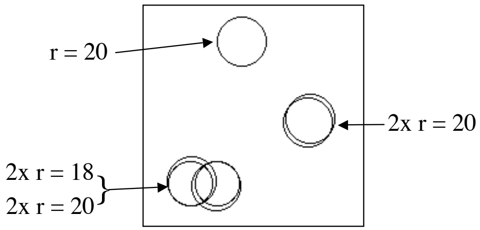
recherche d’un cercle de rayon 20

sélection des cercles de rayon 18, 20 ou 22, reconnus par x% de points du périmètre

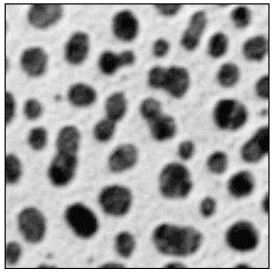
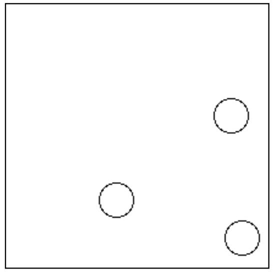


x = 30%



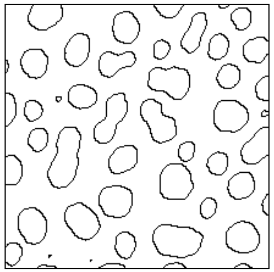
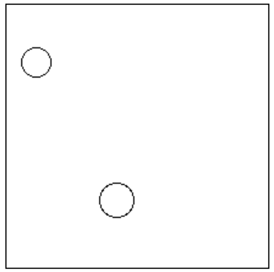
x=25

TRANSFORMATION DE HOUGH : extension

cercles de rayon 14

reconnaissance
à 40% du périmètre

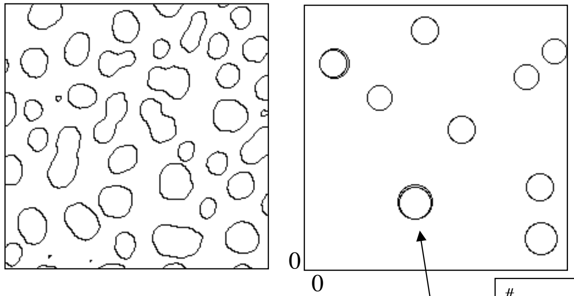



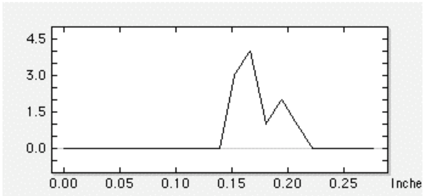
cercles de rayon 12 ou14

reconnaissance
à 50% du périmètre

contours

TRANSFORMATION DE HOUGH : extension



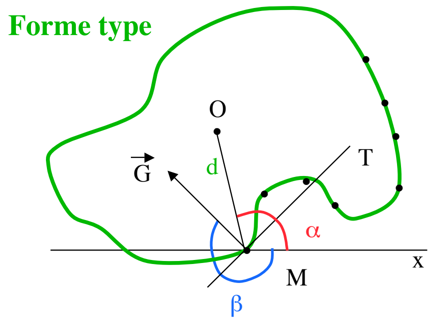


nombre de cercles trouvés / rayon

#	X	Y	Size
1.	2.833	2.472	21
2.	2.521	2.181	21
3.	0.847	1.944	21
4.	1.361	2.708	23
5.	0.319	2.340	23
6.	1.778	1.583	23
7.	2.667	0.931	23
8.	0.333	2.333	25
9.	**1.250**	**0.750**	**27**
10.	2.681	0.347	27
11.	**1.250**	**0.764**	**29**

HOUGH GÉNÉRALISÉE 1- MODÉLISATION

Forme type



O : centre de gravité

MT : tangente au contour en M

$\vec{G}$: normale à MT (direction du gradient)

$d = \overline{MO}$

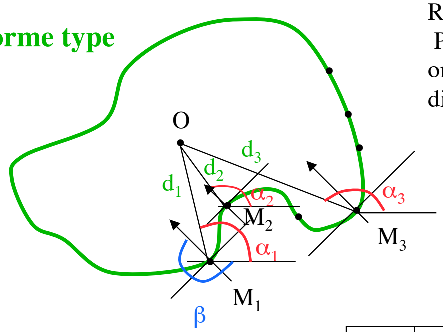
α :angle sous lequel est vu le centre de gravité

β : angle de la normale au contour avec l 'horizontale

Modèle = liste de triplets {β, α, d }

HOUGH GÉNÉRALISÉE 1- MODÉLISATION

Forme type



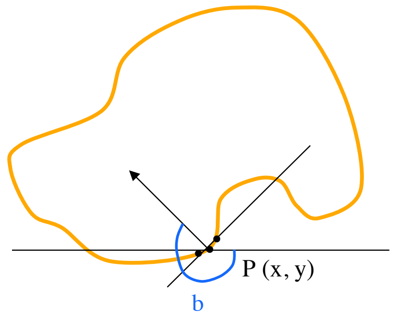
Remarque :
Plusieurs points du contour-modèle ont le même angle β, avec des valeurs différentes pour α_i et d_i .

Pour représenter le modèle, on regroupera les triplets {β, α, d } ayant la même valeur β.

β_1	(α_{11},d_{11}) (α_{12},d_{12}) … (α_{1j},d_{1j})
β_2	(α_{21},d_{21}) (α_{22},d_{22}) … (α_{1k},d_{1k})
β_n	(α_{n1},d_{n1}) (α_{n2},d_{n2}) … (α_{nr},d_{nr})

HOUGH GÉNÉRALISÉE 2- RECONNAISSANCE

Forme à reconnaître



L 'espace des paramètres est celui des coordonnées du centre de gravité O

Pour chaque point P, on calcule l 'angle **b** que fait la normale au contour en P avec l 'horizontale (on estime la direction du contour à l 'aide des 2 points adjacents à P sur le contour).

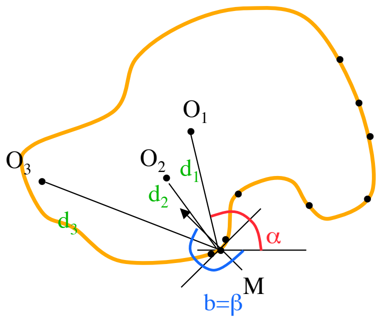
On parcourt la forme modèle et on cherche les triplets {b, α, d}. Ce sont les points de contours du modèle qui ont une normale parallèle à celle de P. On peut donc dire que la portion de contour centrée sur P vote pour un centre de gravité dont les coordonnées sont données par α et d :

$$x0 = x + d.\cos(\alpha)$$
$$y0 = y + d.\sin(\alpha)$$

Si beaucoup de points de la forme votent pour ce même point, la forme sera reconnue.

HOUGH GÉNÉRALISÉE 2- RECONNAISSANCE

Forme à reconnaître



L 'angle b au point M de la forme correspond à l 'angle β du modèle. Donc si M appartient à une forme correspondant au modèle, il vote pour des centres de gravité qui seraient en O_1 ou O_2 ou O_3 .

HOUGH GÉNÉRALISÉE : Algorithme

A- Modélisation

1. On regroupe les triplets du modèle ayant la même valeur β et on les range dans un tableau T dont les lignes sont indicées par les différentes valeurs de β.

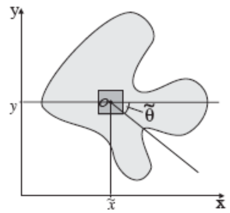
T

β_1	(α_{11},d_{11}) (α_{12},d_{12}) … (α_{1j},d_{1j})
β_2	(α_{21},d_{21}) (α_{22},d_{22}) … (α_{1k},d_{1k})
β_n	(α_{n1},d_{n1}) (α_{n2},d_{n2}) … (α_{nr},d_{nr})

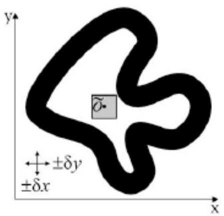
B- Reconnaissance

2. On initialise un tableau A $[x_{min}.. x_{man}; y_{min}.. y_{man}]$
3. Pour chaque point P (x,y) du contour,
   - on calcule l 'angle b, ce qui fournit une entrée (b=β) dans la table T
   - pour chaque couple de la ligne β de T,
     - calculer $x_0 = x + d.\cos(\alpha)$
       $y_0 = y + d.\sin(\alpha)$
     - incrémenter $A[x_0 , y_0]$
4. Déterminer le couple (x,y) qui maximise A[x, y]

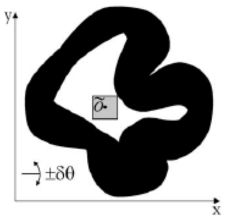
Optimisation algorithmique (hiérarchique)



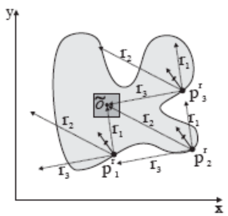
Valeur approchée
(1er niveau)

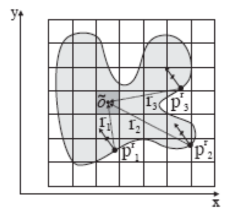


translation



rotation

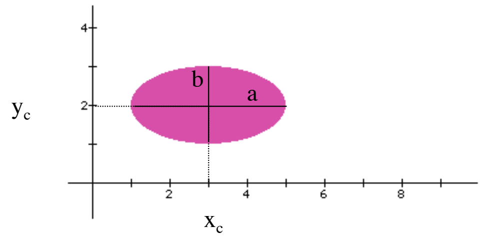




Exercice : recherche d 'ellipses

Soit l'équation d'une ellipse de centre (x_c,y_c) et d'axes 2a et 2b :

$$\frac{(x-x_c)}{a^2}+\frac{(y-y_c)}{b^2}=1$$



Ecrire un programme permettant de détecter la présence d 'ellipse de taille donnée (cad dont a et b sont connus), et de localiser leur centre.