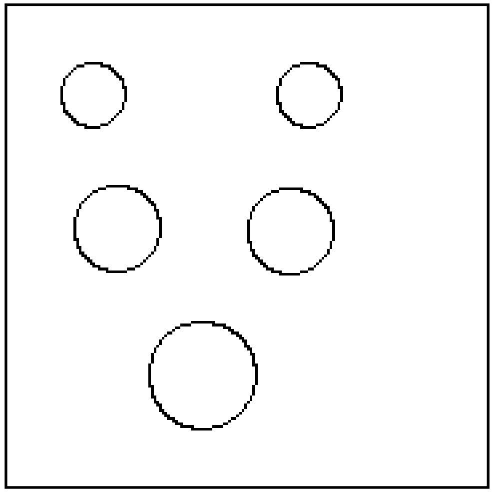
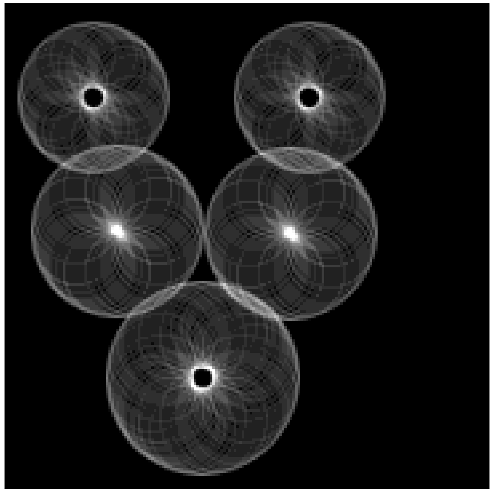
TRANSFORMATION DE HOUGH : extension

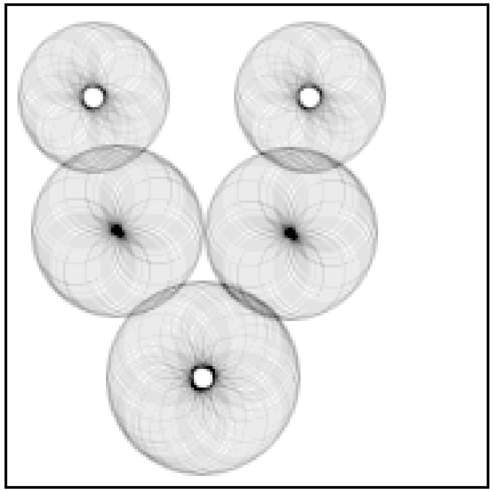
rayons : 12,16 et 20

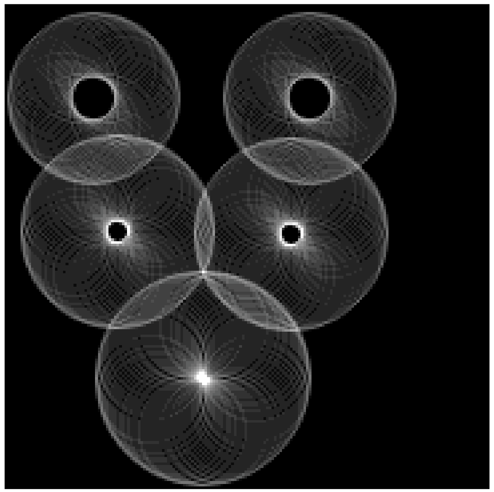


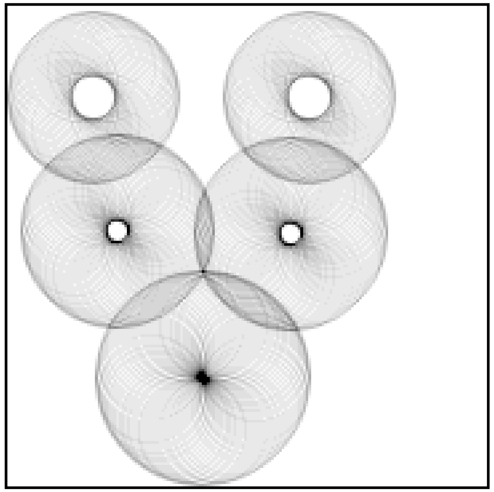
recherche des cercles de rayon 16 (+ augmentation du contraste)



(inversée)

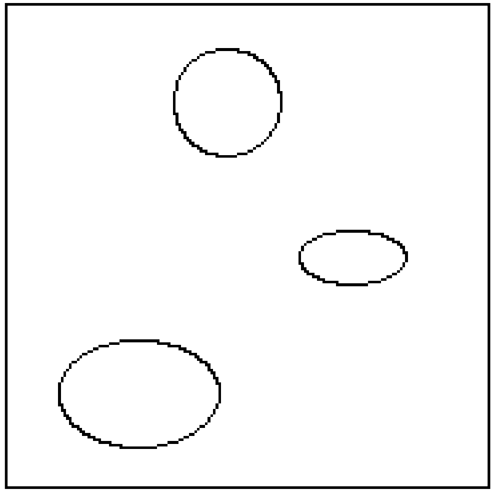






recherche des cercles de rayon 20 (+ augmentation du contraste)

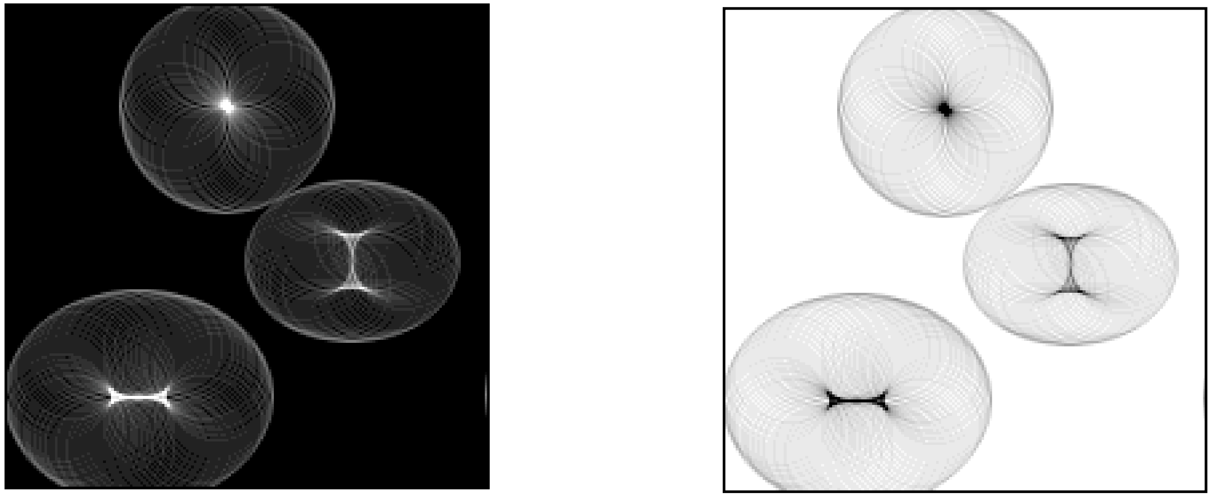
TRANSFORMATION DE HOUGH : extension



cercle de rayon 20

ellipse 20 / 40

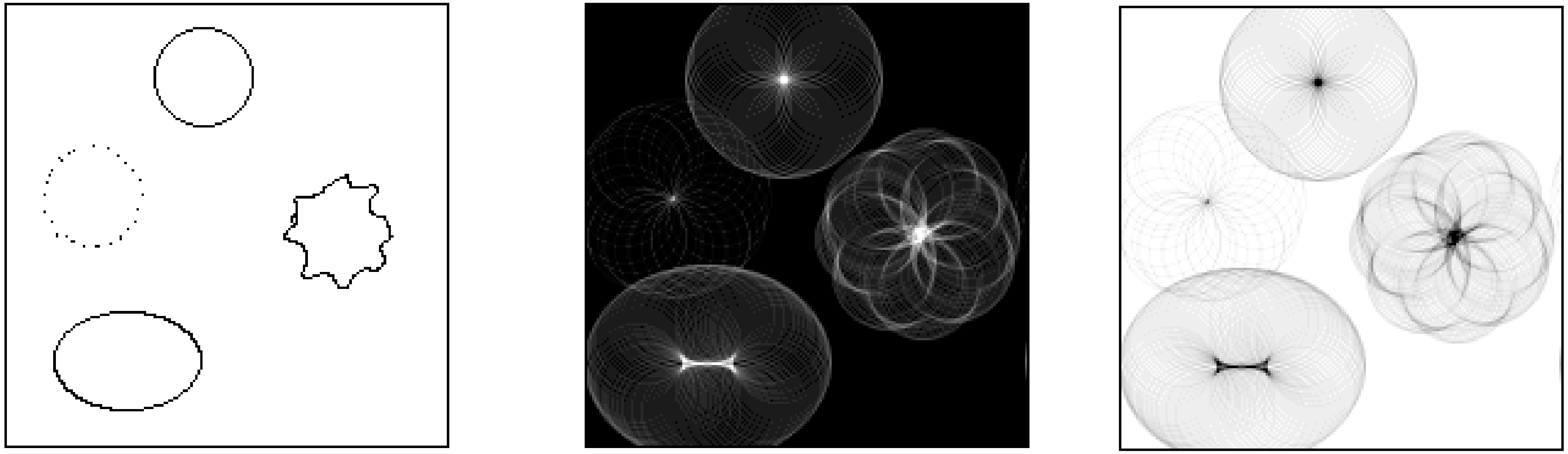
ellipse 40 / 60



recherche d’un cercle de rayon 20

TRANSFORMATION DE HOUGH : extension

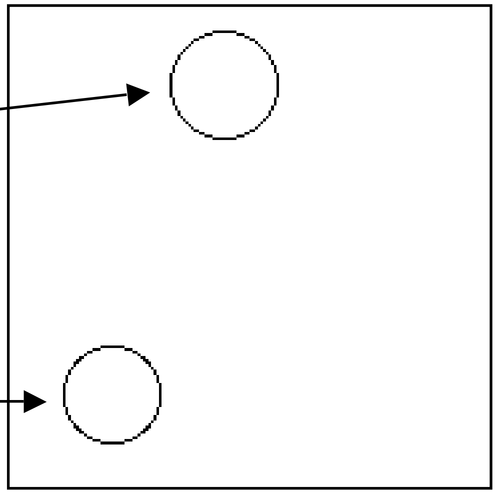
recherche d'un cercle de rayon 20



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

r = 20

r = 18



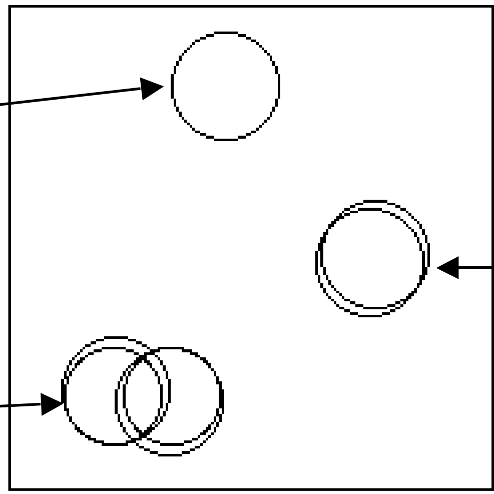
x = 30%

r = 20

2x r = 20

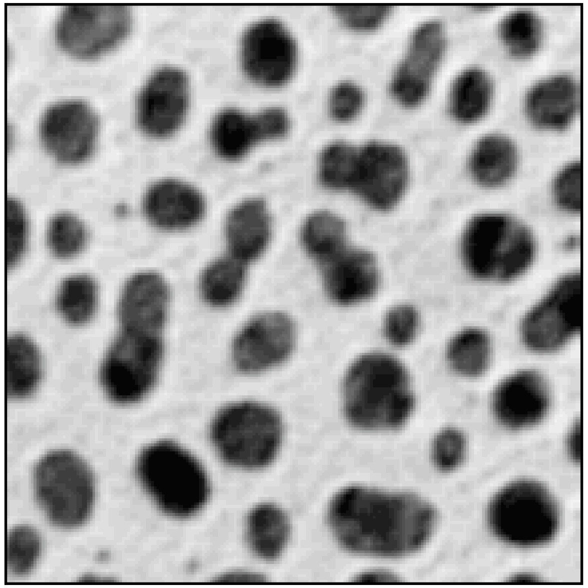
2x r = 18

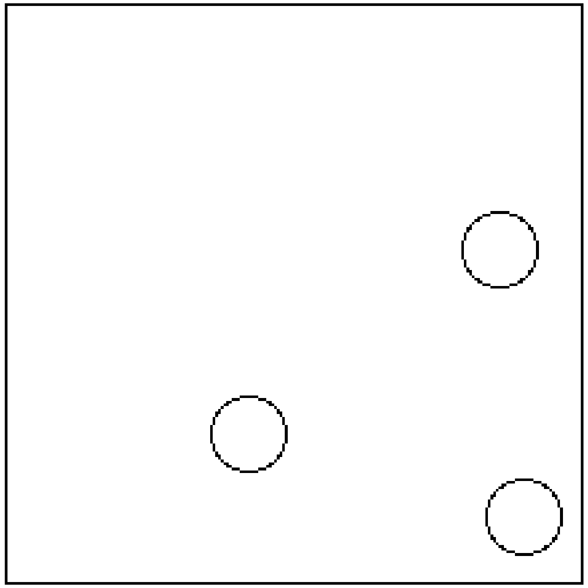
2x r = 20



x=25

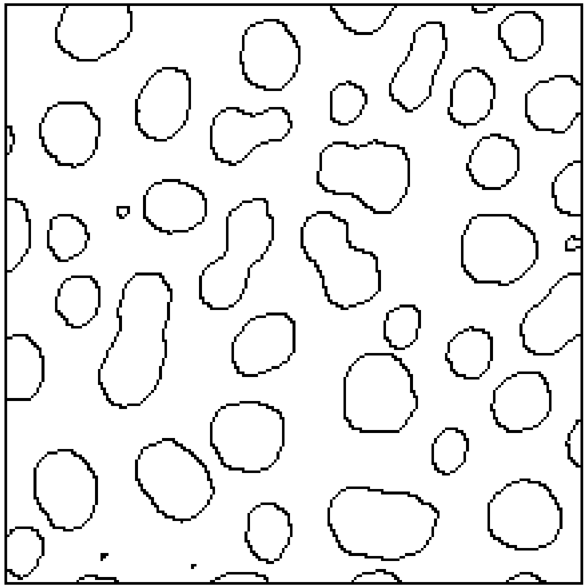
TRANSFORMATION DE HOUGH : extension

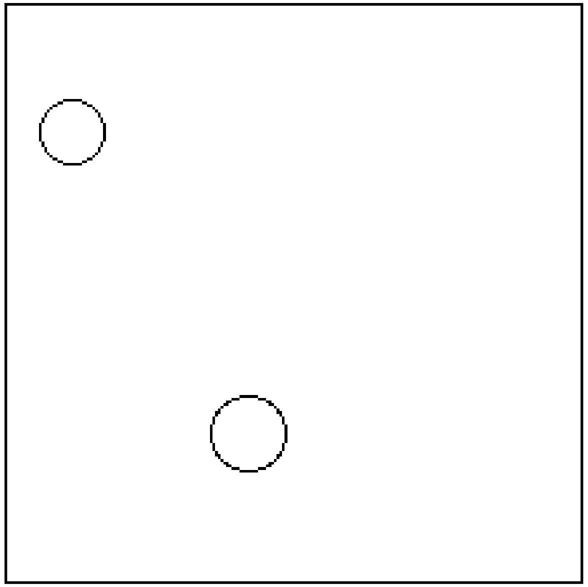




cercles de rayon 14

reconnaissance
à 40% du périmètre



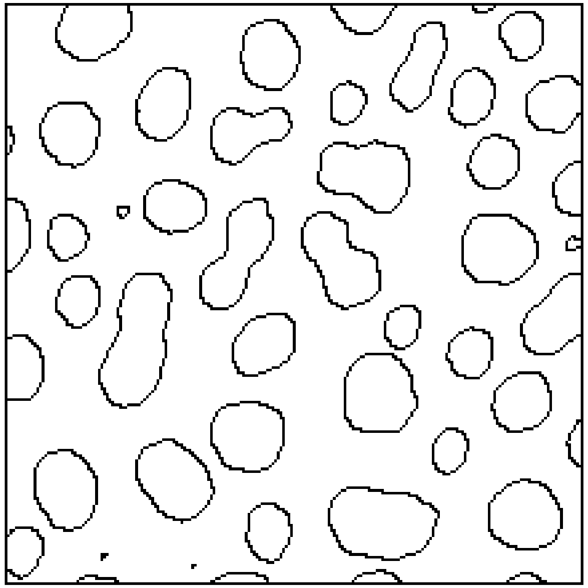


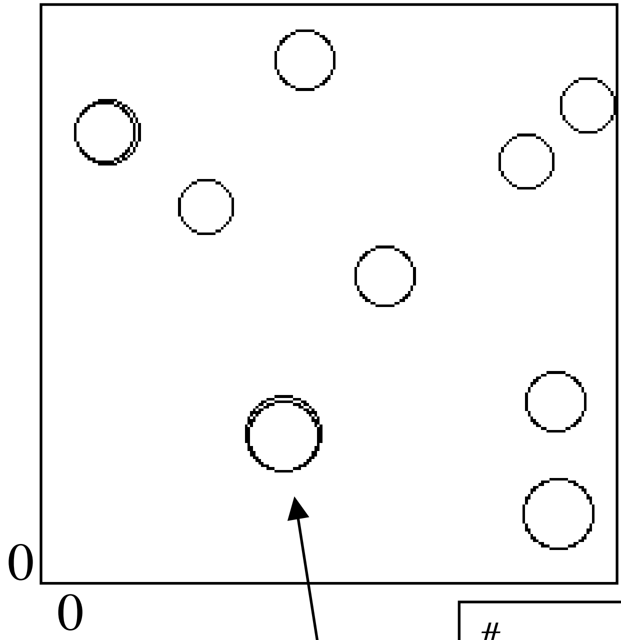
cercles de rayon 12 ou14

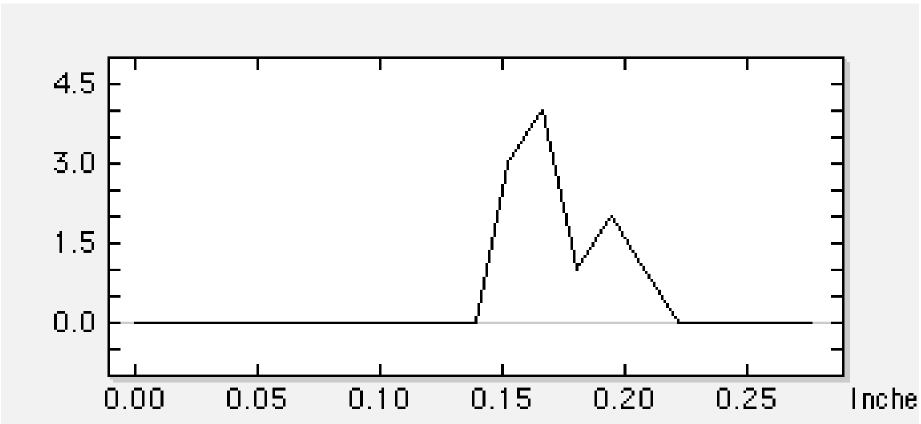
reconnaissance
à 50% du périmètre

contours

TRANSFORMATION DE HOUGH : extension



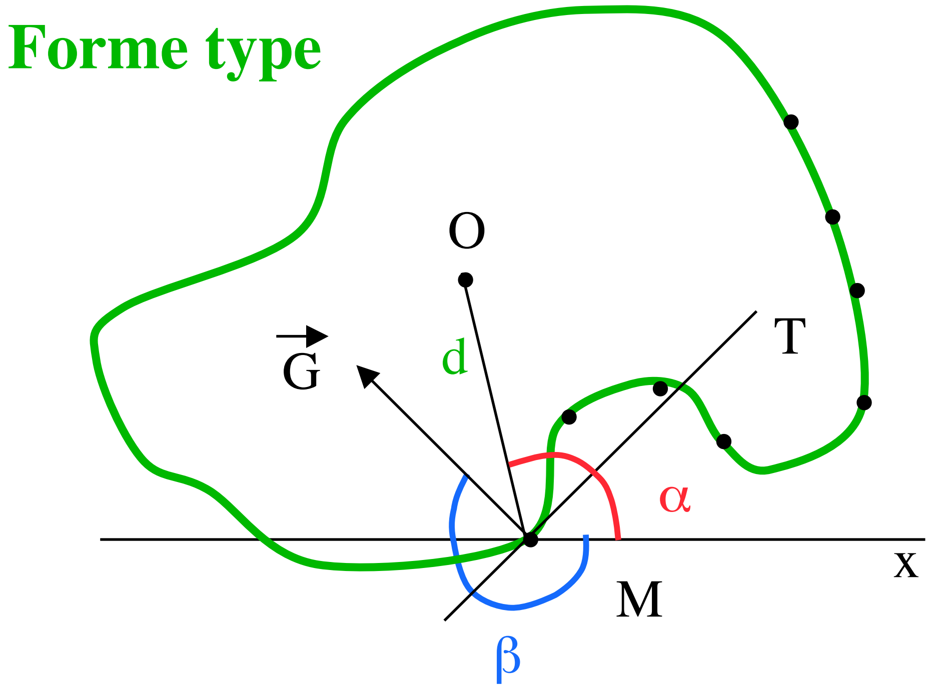




nombre de cercles trouvés / rayon

#	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



O : centre de gravité

MT : tangente au contour en M

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

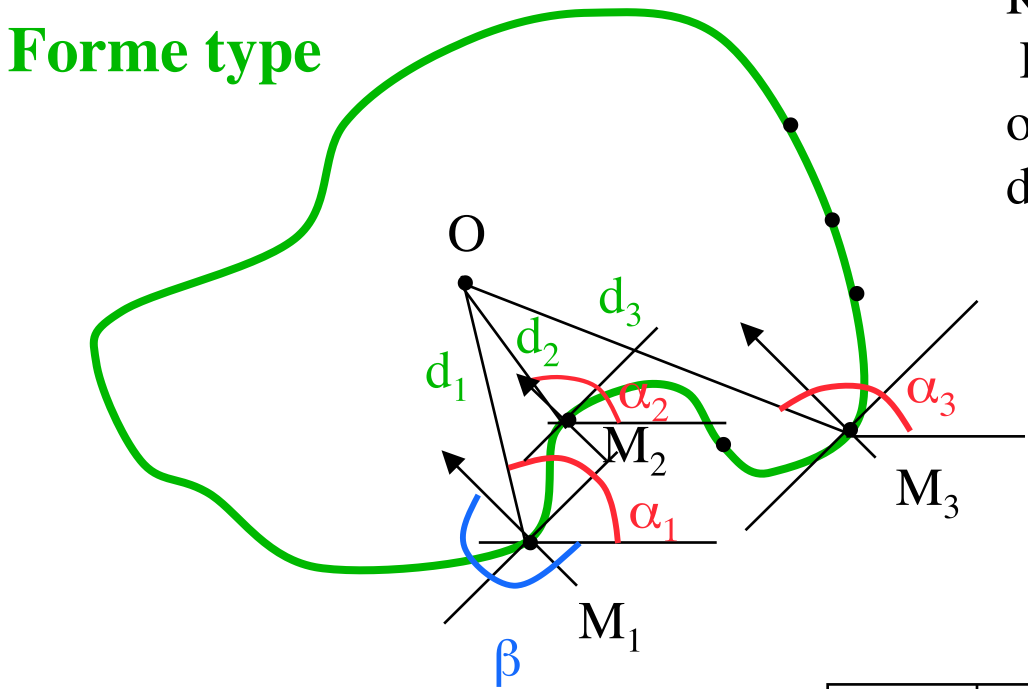
d = $\overline{MO}$

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



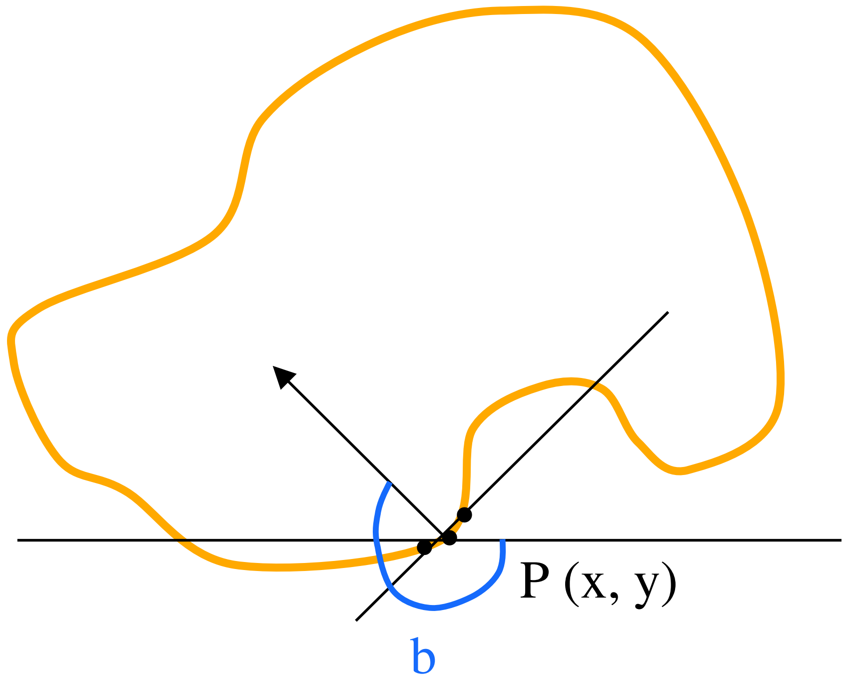
Remarque :
Plusieurs points du contour-modèle ont le même angle β, avec des valeurs différentes pour α_i et d_i .

Pour représenter le modèle, on regroupera les triplets $\{\beta, \alpha, d\}$ 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).

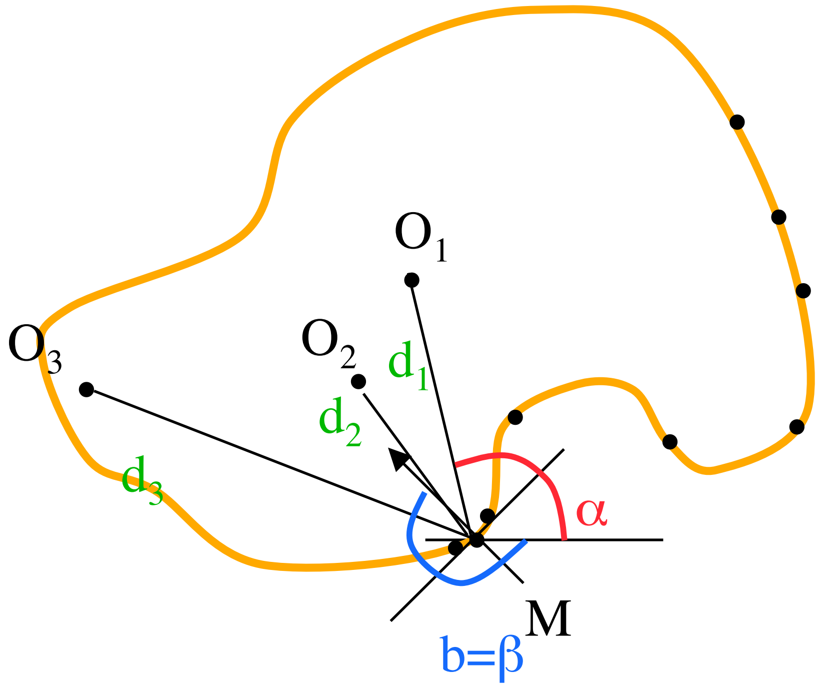
On parcourt la forme modèle et on cherche les triplets {b, α, d}. Ce sont les points de contours du modèle qui ont une normale parallèle à celle de P. On peut donc dire que la portion de contour centrée sur P vote pour un centre de gravité dont les coordonnées sont données par α et d :

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

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})

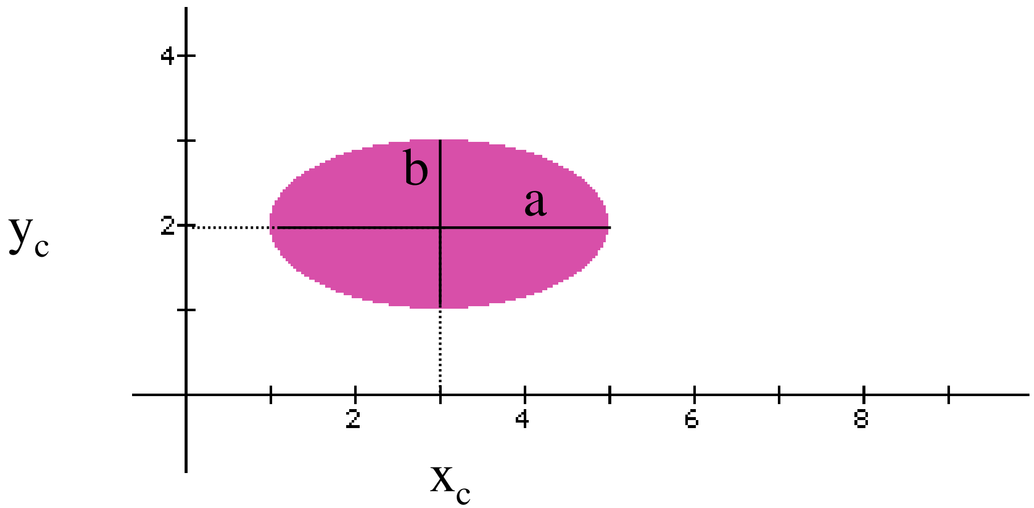
B- Reconnaissance

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 ($b=\beta$) 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]

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.