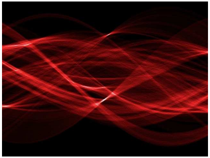
Algorithmie de l'image
Transformée de Hough

Christophe Tilmant - Vincent Barra (tilmant@isima.fr)

ISIMA - Université Blaise Pascal

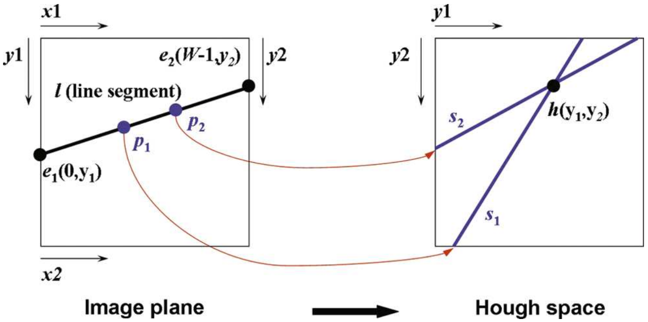
ZZ3 F2-F4



Standard Hough Transform (1962) : Principe

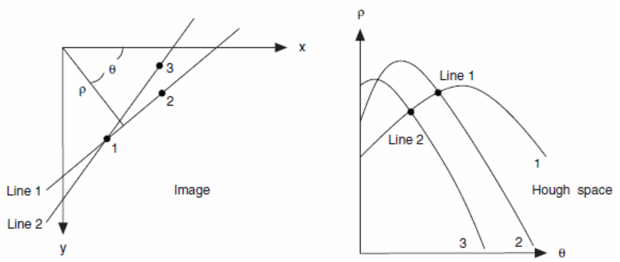
Idée : Détection d'une forme paramétrique basée sur une procédure de vote.

Principe : Après détection d'un point de contour, on vote pour toutes les formes paramétriques pouvant passer par ce point. Finalement, on traite les maximas locaux des "votes".

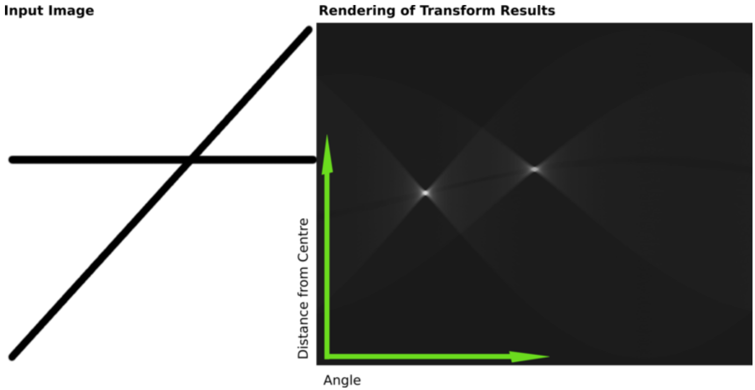


Détection de droites dans une image

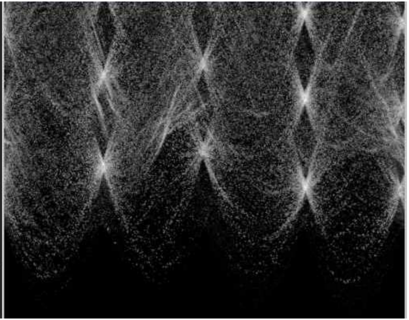
Équation : La représentation $y = a \cdot x + b$ n'est pas intéressante. En effet $a \in [-\infty, +\infty]$ et sera dur à discrétiser, on préfère donc $\rho = x \cdot \cos(\theta) + y \cdot \sin(\theta)$ car $\theta \in [0, 2\pi]$ et $\rho \in [0, \rho_{max}]$.
Quand on détecte un point (x_0, y_0), on vote $\forall \theta$;
$\rho = x_0 \cdot \cos(\theta) + y_0 \cdot \sin(\theta)$: c'est une sinusoïde.



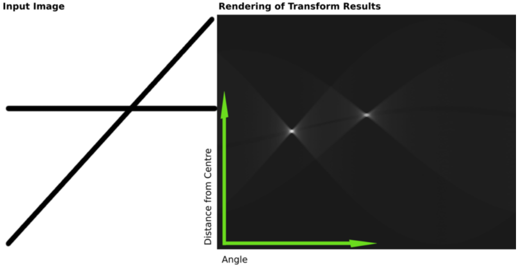
Détection de droites dans une image

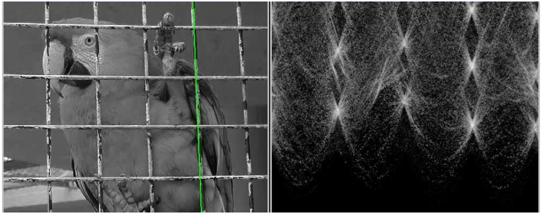


Détection de droites dans une image



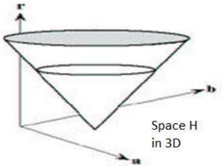
Détection de droites dans une image

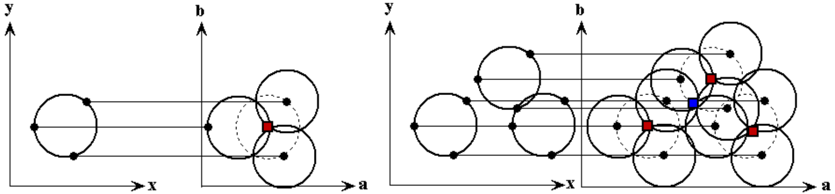




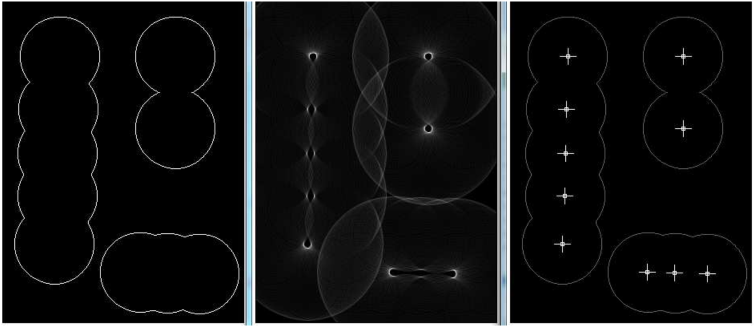
Détection de cercles dans une image

Équation : $r^2 = (x-a)^2 + (y-b)^2$, il y a 3 paramètres.
Quand on détecte le point (x_0, y_0), on vote $\forall\theta$;
$a = x_0 - r\cos(\theta), b = y_0 - r\sin(\theta)$: c'est un cône.

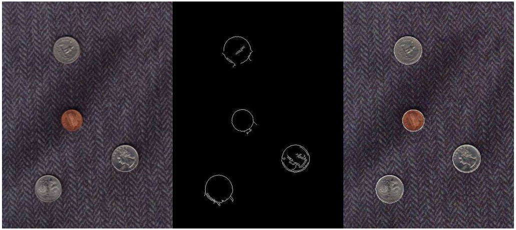


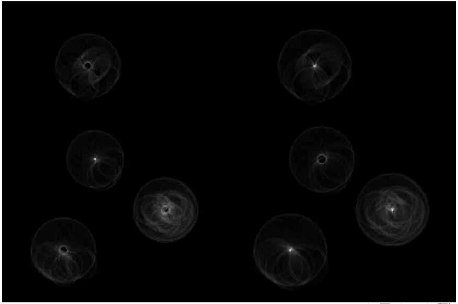


Détection de cercles dans une image

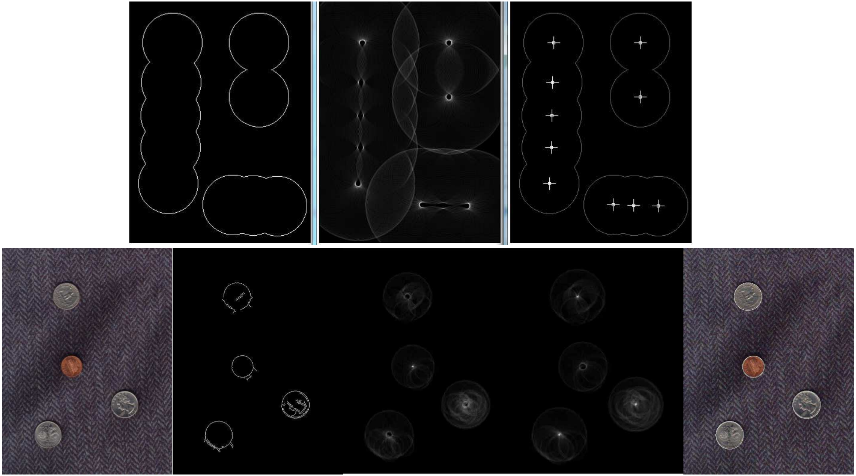


Détection de cercles dans une image



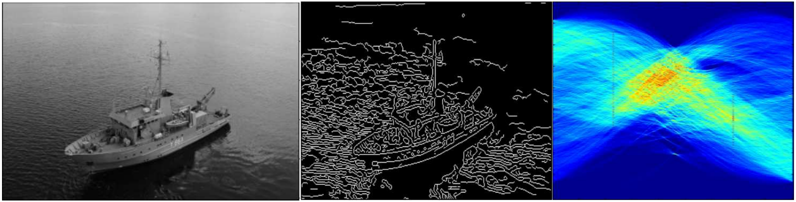


Détection de cercles dans une image



Weighted Hough Transform : Principe

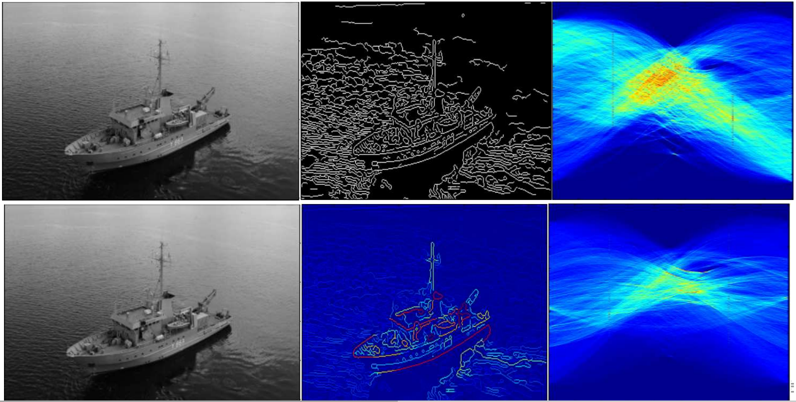
Idée : Ne pas voter en "+1", mais en tenant compte de l'importance du point.
Principe : Comme on utilise un détecteur de contour basé gradient, on pondère le vote par la norme du gradient.



Weighted Hough Transform : Principe

<u>Idée</u> : Ne pas voter en "+1", mais en tenant compte de l'importance du point.

<u>Principe</u> : Comme on utilise un détecteur de contour basé gradient, on pondère le vote par la norme du gradient.

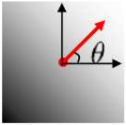


Modified Hough Transform : Principe

<u>Idée</u> : Suivant les structures géométriques locales du point extrait de l'image, on ne vote que pour les formes qui ont les mêmes structures géométriques.

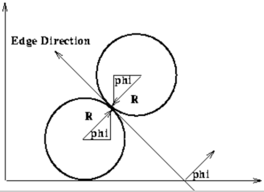
<u>Principe</u> : Suivant l'orientation du gradient du point, on choisit une forme paramétrique avec la même orientation.

- Droite : on choisit <u>la</u> droite dont la tangente correspond à l'orientation du gradient ;
- Cercle : pour chaque θ, on vote pour les <u>deux</u> cercles possédant des normales colinéaires au gradient.



$$\nabla f = \left[\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\right]$$

$$\theta = \tan^{-1}\left(\frac{\partial f}{\partial y} / \frac{\partial f}{\partial x}\right)$$



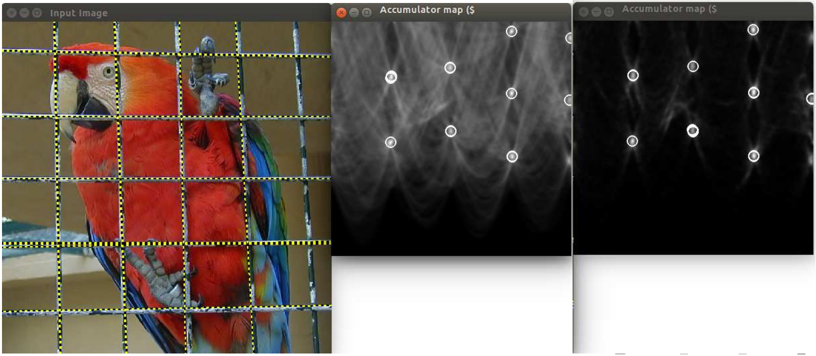
Modified Hough Transform : Principe

Idée : Suivant les structures géométriques locales du point extrait de l'image, on ne vote que pour les formes qui ont les mêmes structures géométriques.
Principe : Suivant l'orientation du gradient du point, on choisit une forme paramétrique avec la même orientation.

Modified Hough Transform : Principe

Idée : Suivant les structures géométriques locales du point extrait de l'image, on ne vote que pour les formes qui ont les mêmes structures géométriques.

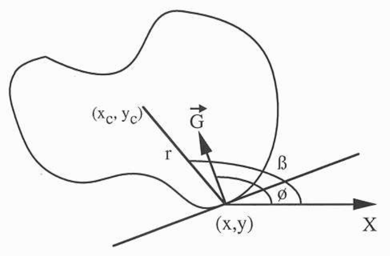
Principe : Suivant l'orientation du gradient du point, on choisit une forme paramétrique avec la même orientation.



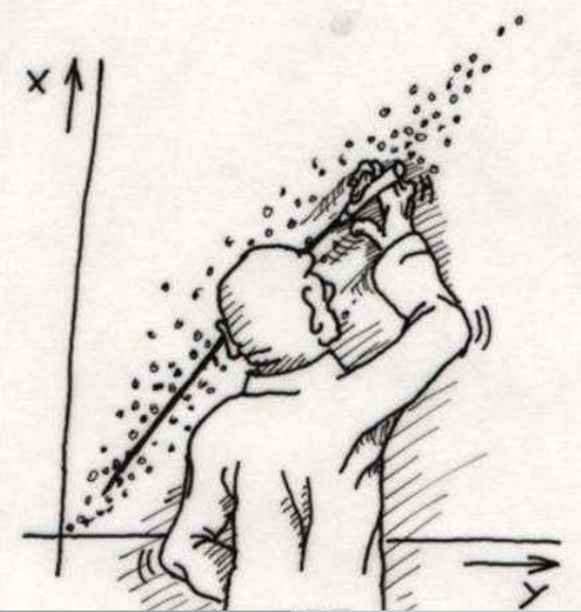
Generalized Hough Transform (1981) : Principe

Idée : détecter une forme arbitraire (et non plus paramétrique) toujours avec un système de vote.

Principe : En partant d'une modèle de forme on créé une signature de la forme (*R-table*), puis on remplit une matrice d'accumulation.



$\phi_1 = 0$	$(r,\beta)_{1_1}$	$(r,\beta)_{1_2}$	$\cdots$	$(r,\beta)_{1_{n_1}}$
$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
ϕ_j	$(r,\beta)_{j_1}$	$(r,\beta)_{j_2}$	$\cdots$	$(r,\beta)_{j_{n_1}}$
$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
$\phi_k = \pi$	$(r,\beta)_{k_1}$	$(r,\beta)_{k_2}$	$\cdots$	$(r,\beta)_{k_{n_1}}$


X
Y