DÉTECTION DE CARACTÉRISTIQUES ET DE MOTIFS

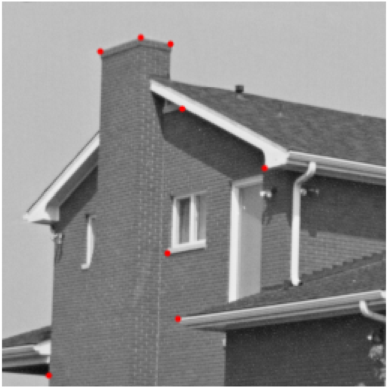
Détection de caractéristiques
(feature detection)

Contours

Arêtes des objets : brusque changement de la luminance.

Détection de caractéristiques

(feature detection)



Coins

Brusque changement de la luminance dans les deux dimensions, intersection de deux contours.

Détection de caractéristiques

(feature detection)

Lignes

Alignements de points (à effectuer après une détection de contours).

Détection de motifs

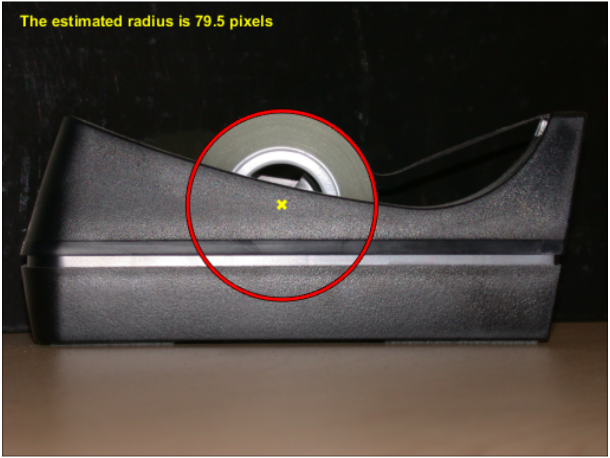
(pattern detection)

Motifs

Imagette connue.

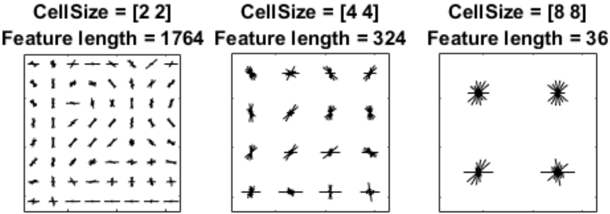
Détection de caractéristiques

Mesurer des objets



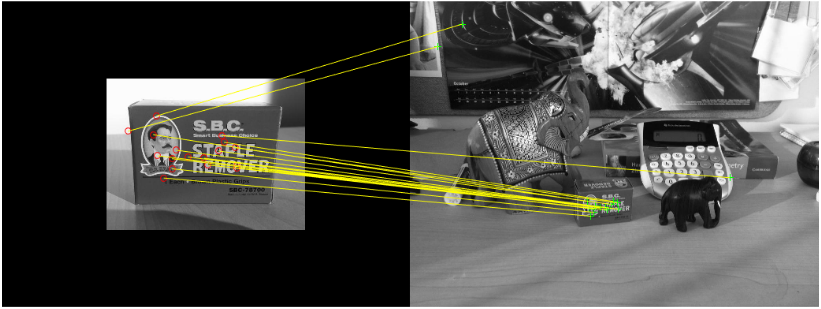
Détection de caractéristiques

Extraction de caractéristiques
(classification pour la reconnaissance de caractères, …)



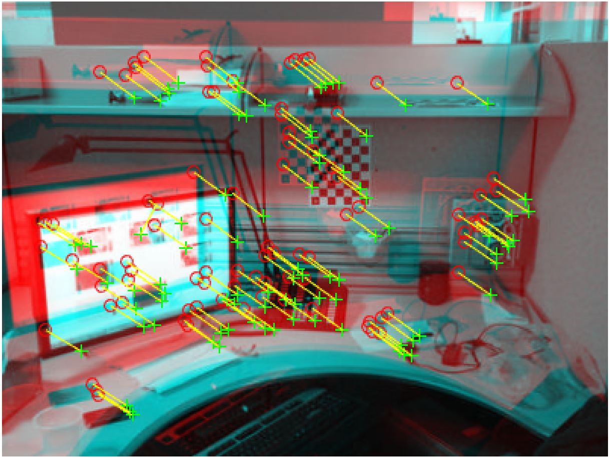
Détection de caractéristiques

Détection d’objets



Détection de caractéristiques

Association d'images
(recalage, stabilisation de vidéo, …)



Sommaire

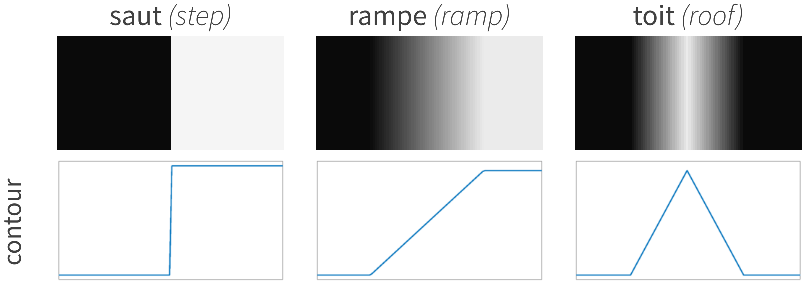
- Détection de contours *(edges)*
- Détection de coins *(corners)*
- Détection de droites *(lines)*
- Détection de motifs *(patterns)*

Détection de contours

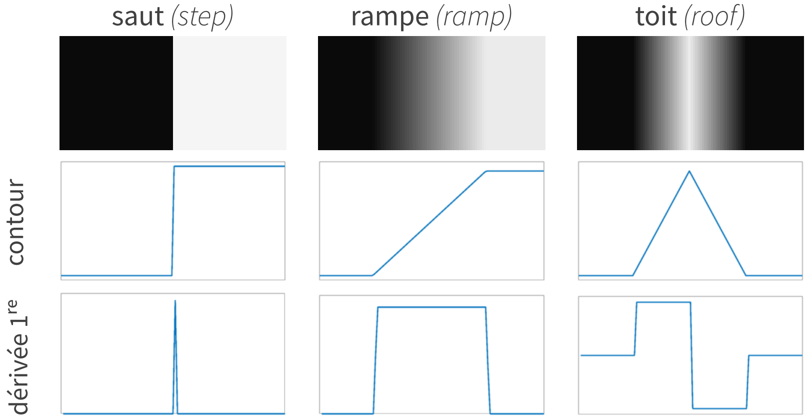
Exemples de contours

lena_edges.m

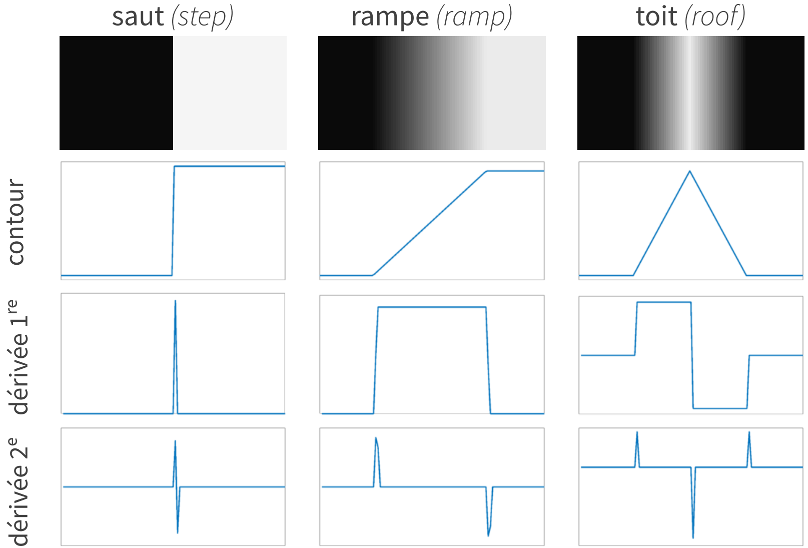
Modèles de contours



Modèles de contours



Modèles de contours



Modèles de contours

La présence d'un contour est détectée…

- en analysant l'amplitude de la dérivée 1^{re}
- ou en déterminant le passage à zéro de la dérivée 2^{e}

…selon le profil d'intensité perpendiculairement au contour.

Gradient et laplacien

- Dérivée 1re (**gradient**)

$$\begin{pmatrix} \dfrac{\partial f(x,y)}{\partial x} \\ \dfrac{\partial f(x,y)}{\partial y} \end{pmatrix}$$

- Dérivée 2e (**laplacien**)

$$\begin{pmatrix} \dfrac{\partial^2 f(x,y)}{\partial x^2} \\ \dfrac{\partial^2 f(x,y)}{\partial y^2} \end{pmatrix}$$

Gradient et laplacien

- Dérivée 1re (**gradient**)

$$\begin{pmatrix} \dfrac{\partial f(x,y)}{\partial x} \\ \dfrac{\partial f(x,y)}{\partial y} \end{pmatrix} = \begin{pmatrix} f(x+1,y) - f(x,y) \\ f(x,y+1) - f(x,y) \end{pmatrix}$$

- Dérivée 2e (**laplacien**)

$$\begin{pmatrix} \dfrac{\partial^2 f(x,y)}{\partial x^2} \\ \dfrac{\partial^2 f(x,y)}{\partial y^2} \end{pmatrix}$$

Gradient et laplacien

- Dérivée 1re (**gradient**)

$$\begin{pmatrix} \dfrac{\partial f(x,y)}{\partial x} \\ \dfrac{\partial f(x,y)}{\partial y} \end{pmatrix} = \begin{pmatrix} f(x+1,y) - f(x,y) \\ f(x,y+1) - f(x,y) \end{pmatrix}$$

- Dérivée 2e (**laplacien**)

$$\begin{pmatrix} \dfrac{\partial^2 f(x,y)}{\partial x^2} \\ \dfrac{\partial^2 f(x,y)}{\partial y^2} \end{pmatrix} = \begin{pmatrix} f(x+1,y) - 2f(x,y) + f(x-1,y) \\ f(x,y+1) - 2f(x,y) + f(x,y-1) \end{pmatrix}$$

Opérateurs du gradient

$$f(x+1,y) - f(x,y)$$

Opérateurs du gradient

$$f(x+1,y) - f(x,y) = \sum_m \sum_n h_x(m,n) f(x-m, y-n)$$

Opérateurs du gradient

$$f(x+1,y) - f(x,y) = \sum_m \sum_n h_x(m,n) f(x-m, y-n)$$

où $\begin{cases} h_x(0,0) = -1 \\ h_x(-1,0) = +1 \\ h_x(m,n) = 0 \quad \text{ailleurs} \end{cases} \quad \Rightarrow \quad h_x = \begin{pmatrix} 0 & +1 \\ 0 & -1 \end{pmatrix}$

Opérateurs du gradient

$$f(x+1,y) - f(x,y) = \sum_m \sum_n h_x(m,n) f(x-m, y-n)$$

où $\begin{cases} h_x(0,0) = -1 \\ h_x(-1,0) = +1 \\ h_x(m,n) = 0 \quad \text{ailleurs} \end{cases} \quad \Rightarrow \quad h_x = \begin{pmatrix} 0 & +1 \\ 0 & -1 \end{pmatrix}$

$$f(x,y+1) - f(x,y)$$

Opérateurs du gradient

$$f(x+1,y) - f(x,y) = \sum_m \sum_n h_x(m,n) f(x-m,y-n)$$

où $\begin{cases} h_x(0,0) = -1 \\ h_x(-1,0) = +1 \\ h_x(m,n) = 0 \quad \text{ailleurs} \end{cases} \Rightarrow \quad h_x = \begin{pmatrix} 0 & +1 \\ 0 & -1 \end{pmatrix}$

$$f(x,y+1) - f(x,y) = \sum_m \sum_n h_y(m,n) f(x-m,y-n)$$

Opérateurs du gradient

$$f(x+1,y) - f(x,y) = \sum_m \sum_n h_x(m,n) f(x-m, y-n)$$

où $\begin{cases} h_x(0,0) = -1 \\ h_x(-1,0) = +1 \\ h_x(m,n) = 0 \quad \text{ailleurs} \end{cases} \Rightarrow \quad h_x = \begin{pmatrix} 0 & +1 \\ 0 & -1 \end{pmatrix}$

$$f(x,y+1) - f(x,y) = \sum_m \sum_n h_y(m,n) f(x-m, y-n)$$

où $\begin{cases} h_y(0,0) = -1 \\ h_y(0,-1) = +1 \\ h_y(i,j) = 0 \quad \text{ailleurs} \end{cases} \Rightarrow \quad h_y = \begin{pmatrix} 0 & 0 \\ +1 & -1 \end{pmatrix}$

Opérateurs du gradient

Filtres de Roberts [Roberts 1965] :

$$h_x = \begin{pmatrix} 0 & +1 \\ 0 & -1 \end{pmatrix} \quad h_y = \begin{pmatrix} 0 & 0 \\ +1 & -1 \end{pmatrix}$$

Opérateurs du gradient

Filtres de Roberts [Roberts 1965] :

$$h_x = \begin{pmatrix} 0 & +1 \\ 0 & -1 \end{pmatrix} \quad h_y = \begin{pmatrix} 0 & 0 \\ +1 & -1 \end{pmatrix}$$

Filtres de Prewitt [Prewitt 1970] (permet de centrer les filtres de Roberts) :

$$h_x = \begin{pmatrix} +1 & +1 & +1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{pmatrix} \quad h_y = \begin{pmatrix} +1 & 0 & -1 \\ +1 & 0 & -1 \\ +1 & 0 & -1 \end{pmatrix}$$

Opérateurs du gradient

Filtres de Roberts [Roberts 1965] :

$$h_x = \begin{pmatrix} 0 & +1 \\ 0 & -1 \end{pmatrix} \quad h_y = \begin{pmatrix} 0 & 0 \\ +1 & -1 \end{pmatrix}$$

Filtres de Prewitt [Prewitt 1970] (permet de centrer les filtres de Roberts) :

$$h_x = \begin{pmatrix} +1 & +1 & +1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{pmatrix} \quad h_y = \begin{pmatrix} +1 & 0 & -1 \\ +1 & 0 & -1 \\ +1 & 0 & -1 \end{pmatrix}$$

Filtres de Sobel [Sobel 1968] (version lissée du filtre de Prewitt) :

$$h_x = \begin{pmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{pmatrix} \quad h_y = \begin{pmatrix} +1 & 0 & -1 \\ +2 & 0 & -2 \\ +1 & 0 & -1 \end{pmatrix}$$

Opérateurs du gradient

- Il existe des variantes diagonales.
- La somme des coefficients est égal à 0.
- Amplitude *(magnitude)*

$$M = \sqrt{(h_x * f)^2 + (h_y * f)^2}$$

- Angle (direction)

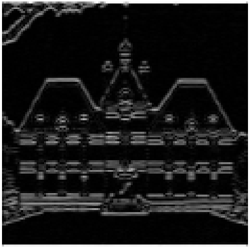
$$A = \operatorname{atan}\left(\frac{h_y * f}{h_x * f}\right)$$

Filtre de Sobel

f

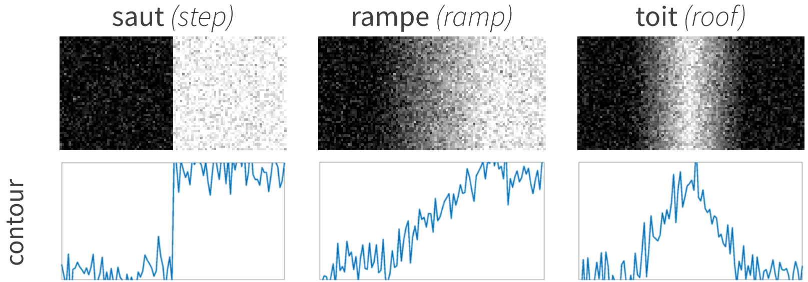
M

$|h_x * f|$

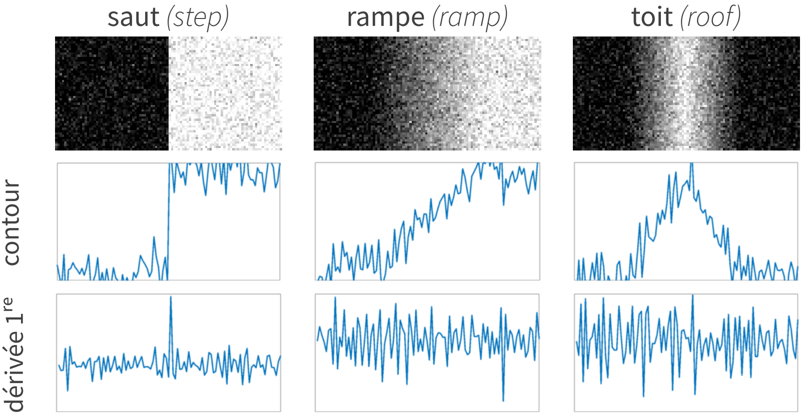


$|h_y * f|$

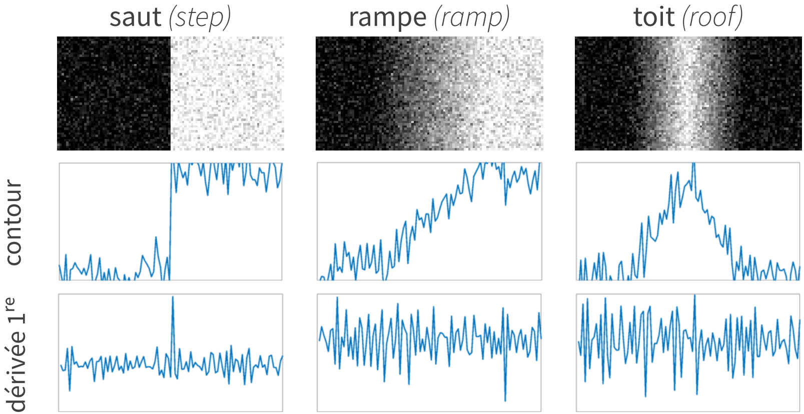
Impact du bruit sur la détection



Impact du bruit sur la détection



Impact du bruit sur la détection



Dérivées très sensibles au bruit ! ⇒ débruitage

Impact du bruit sur la détection

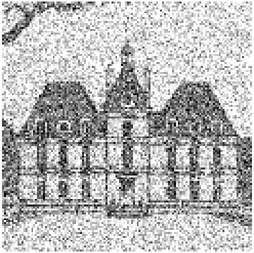


Image bruitée

Sobel

Moyenneur 5×5
+ Sobel

Seuillage du résultat

On peut seuiller l'image $|h_x * f| + |h_y * f|$ pour ne conserver que les grandes valeurs du gradient.

f

Sobel

Sobel + seuil

Techniques avancées de détection de contours

L'objectif est d'améliorer la détection en tenant compte du bruit et de la nature des contours.

- Détecteur de Marr–Hildreth [Marr & Hildreth 1980]
- Détecteur de Canny [Canny 1986]

Détecteur de Marr–Hildreth

Le détecteur de Marr–Hildreth consiste à :

1. appliquer un filtre gaussien g sur l'image f pour réduire le bruit,
2. calculer le laplacien (dérivée 2$^{\text{e}}$) ℓ sur l'image adoucie,
3. déterminer les passages par zéro du résultat.

Détecteur de Marr–Hildreth

Le détecteur de Marr–Hildreth consiste à :

1. appliquer un filtre gaussien g sur l'image f pour réduire le bruit,
2. calculer le laplacien (dérivée 2^{e}) ℓ sur l'image adoucie,
3. déterminer les passages par zéro du résultat.

wooclap
/mazet

Quelle(s) étape(s) peu(ven)t s'écrire comme une convolution ?

Détecteur de Marr–Hildreth

Le détecteur de Marr–Hildreth consiste à :

1. appliquer un filtre gaussien g sur l'image f pour réduire le bruit,
2. calculer le laplacien (dérivée 2^e^) ℓ sur l'image adoucie,
3. déterminer les passages par zéro du résultat.

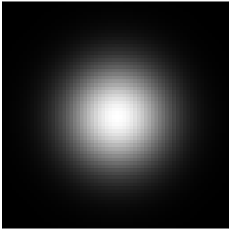
wooclap
/mazet

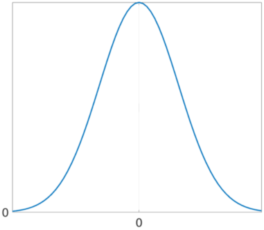
Quelle(s) étape(s) peu(ven)t s'écrire comme une convolution ?

Comme $\ell * (g * f) = (\ell * g) * f$, alors les deux premières étapes sont fusionnées en une seule convolution par $\ell * g$.

Détecteur de Marr–Hildreth

Filtre gaussien :

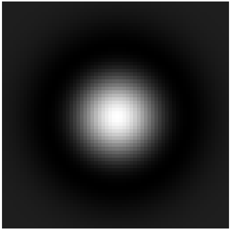


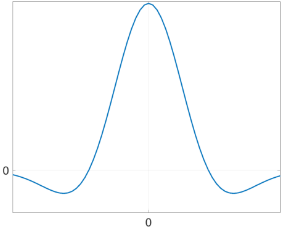


$$g(x,y) = \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right)$$

Détecteur de Marr–Hildreth

Laplacien du filtre gaussien
(appelé LoG *(Laplacian of Gaussian)* ou chapeau mexicain) :



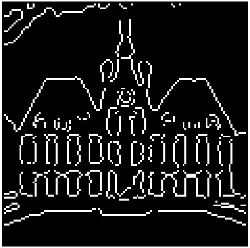


$$\partial^2 g(x,y) = -\left[\frac{x^2+y^2-2\sigma^2}{\sigma^4}\right]\exp\left(-\frac{x^2+y^2}{2\sigma^2}\right)$$

Détecteur de Marr–Hildreth

f

LoG



Marr–Hildreth

Détecteur de Marr–Hildreth

f

LoG

Marr–Hildreth

Détecteur de Canny

Objectifs :

- tous les contours doivent être trouvés
- il doit y avoir un minimum de réponses parasites
- les contours correctement localisés
- l'épaisseur des contours détectés doit être de 1 pixel

Canny a exprimé ces objectifs sous forme mathématique et a proposé des solutions optimales vérifiant ces objectifs.

Détecteur de Canny

Algorithme :

1. lissage de l'image avec un filtre gaussien
2. calcul du gradient (amplitude et angle)
3. suppression des non-maxima sur les amplitudes
4. seuillage par hystérésis

Détecteur de Canny

Lissage

Convolution de l'image f par un noyau gaussien $g(x,y) = e^{-\frac{x^2+y^2}{2\sigma^2}}$:

$$h = f * g.$$

Détecteur de Canny

Lissage

Convolution de l'image f par un noyau gaussien $g(x,y) = e^{-\frac{x^2+y^2}{2\sigma^2}}$:

$$h = f * g.$$

Calcul du gradient

$$M = \sqrt{(h_x * f)^2 + (h_y * f)^2}$$
$$A = \operatorname{atan}\left(\frac{h_y * f}{h_x * f}\right)$$

Détecteur de Canny

Suppression des non-maxima

L'objectif est de réduire les contours trop larges fournis par le calcul du gradient.

Pour chaque pixel (x,y) de l'amplitude M :
  choisir la direction ($\updownarrow$, $\nwarrow\!\!\!\searrow$, $\leftrightarrow$, $\swarrow\!\!\!\nearrow$) la plus proche de $A(x,y)$
  si $M(x,y)$ est plus faible que l'un des deux gradients voisin dans sa direction :
    annuler le gradient : $M(x,y) = 0$

Détecteur de Canny

Seuillage par hystérésis

L'objectif est d'éliminer les faux contours.

Définition de deux seuils tels que $s_{\text{haut}} > s_{\text{bas}}$.

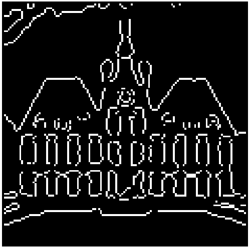
```
pour chaque pixel (x,y) du gradient :
    si M(x,y) > s_haut :
        (x,y) est un point de contour
    si s_bas < M(x,y) < s_haut :
        (x,y) est un point de contour si et seulement
            s'il est voisin d'un point de contour
    si M(x,y) < s_bas :
        (x,y) n'est pas un point de contour
```

Comparaison

f

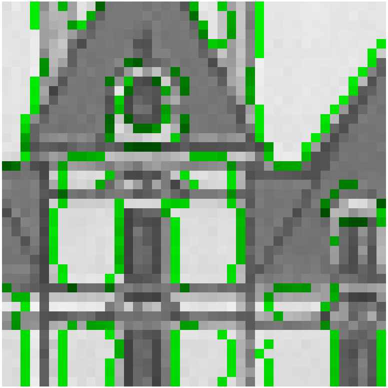
Sobel

Marr–
Hildreth

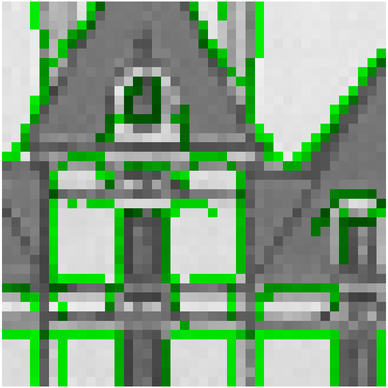


Canny

Comparaison



Marr–Hildreth



Canny

Les contours détectés avec Canny sont mieux localisés.

Détection de coins

Détection de coins

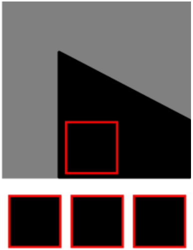
- Jonction de deux contours
- Détecteur de Moravec [Moravec 1980]
- Détecteur de Harris [Harris & Stephens 1988]
- …

Détecteur de Moravec

Principe : observer les changements survenus en décalant légèrement un patch autour d'un pixel d'intérêt. Si les changements sont importants, alors le patch est centré sur un coin.

Détecteur de Moravec

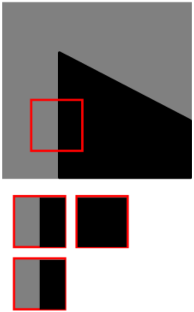
Principe : observer les changements survenus en décalant légèrement un patch autour d'un pixel d'intérêt. Si les changements sont importants, alors le patch est centré sur un coin.



zone plate : pas de changement

Détecteur de Moravec

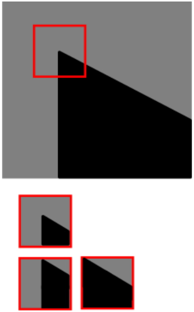
Principe : observer les changements survenus en décalant légèrement un patch autour d'un pixel d'intérêt. Si les changements sont importants, alors le patch est centré sur un coin.



contour : changement significatif dans une seule direction

Détecteur de Moravec

Principe : observer les changements survenus en décalant légèrement un patch autour d'un pixel d'intérêt. Si les changements sont importants, alors le patch est centré sur un coin.



coins : changement significatif dans toutes les directions

Détecteur de Moravec

Les quatre décalages $(x,y) \in \{(1,0), (1,1), (0,1), (-1,1)\}$
sont mesurés en chaque pixel (m,n) de l'image :

$$\forall m,n,x,y \qquad E_{m,n}(x,y) = \sum_{u,v} w_{m,n}(u,v) \left[f(u+x,v+y) - f(u,v)\right]^2$$

où :

- $w_{m,n}$ est une fenêtre rectangulaire autour du pixel (m,n)
- $\left[f(u+x,v+y) - f(u,v)\right]^2$ représente la différence entre le patch $f(u,v)$ et le patch décalé $f(u+x,v+y)$
- $E_{m,n}(x,y)$ est la différence entre les patchs pour un décalage (x,y)

Détecteur de Moravec

Les quatre décalages $(x,y) \in \{(1,0), (1,1), (0,1), (-1,1)\}$
sont mesurés en chaque pixel (m,n) de l'image :

$$\forall m,n,x,y \qquad E_{m,n}(x,y) = \sum_{u,v} w_{m,n}(u,v) \left[f(u+x, v+y) - f(u,v) \right]^2$$

Le minimum de $E_{m,n}(x,y)$ par rapport aux décalages est déterminé :

$$\forall m,n \qquad F_{m,n} = \min_{x,y} E_{m,n}(x,y)$$

Les coins détectés correspondent aux maxima locaux de $F_{m,n}$.

Détecteur de Moravec

$$\forall m,n,x,y \qquad E_{m,n}(x,y) = \sum_{u,v} w_{m,n}(u,v)\left[f(u+x,v+y) - f(u,v)\right]^2$$

0	0	0	0	0	0
0	0	0	0	0	0
0	0	1	1	1	1
0	0	1	1	1	1
0	0	1	1	1	1
0	0	1	1	1	1

Détecteur de Moravec

$$\forall m,n,x,y \qquad E_{m,n}(x,y) = \sum_{u,v} w_{m,n}(u,v)\left[f(u+x,v+y) - f(u,v)\right]^2$$

0	0	0	0	0	0
0	0	0	0	0	0
0	0	1	1	1	1
0	0	1	$\boxed{1}$	1	1
0	0	1	1	1	1
0	0	1	1	1	1

Détecteur de Moravec

$$\forall m,n,x,y \qquad E_{m,n}(x,y) = \sum_{u,v} w_{m,n}(u,v)\left[f(u+x,v+y) - f(u,v)\right]^2$$

0	0	0	0	0	0
0	0	0	0	0	0
0	0	1	1	1	1
0	0	1	1	1	1
0	0	1	1	1	1
0	0	1	1	1	1

Détecteur de Moravec

Problèmes :

- (P1) la réponse du détecteur peut être bruitée car w est une fenêtre binaire
- (P2) seuls des décalages de 45° sont considérés
- (P3) le détecteur est trop sensible aux contours car seul le minimum de E est considéré

Détecteur de Moravec

Problèmes :

- P1 la réponse du détecteur peut être bruitée car w est une fenêtre binaire
- P2 seuls des décalages de 45° sont considérés
- P3 le détecteur est trop sensible aux contours car seul le minimum de E est considéré

$\Rightarrow$ détecteur de Harris.

Détecteur de Harris

Pour éviter une réponse bruitée (problème P1), la fenêtre rectangulaire w est remplacée par une fenêtre w gaussienne dans :

$$E_{m,n}(x,y) = \sum_{u,v} w_{m,n}(u,v) \left[f(u+x, v+y) - f(u,v)\right]^2$$

Détecteur de Harris

Pour étendre le modèle à n'importe quel angle (problème P2), on utilise un développement en série de Taylor de $f(u+x, v+y)$:

$$f(u+x, v+y) \approx f(u,v) + x\,\partial_x f(u,v) + y\,\partial_y f(u,v)$$

Détecteur de Harris

Pour étendre le modèle à n'importe quel angle (problème P2), on utilise un développement en série de Taylor de $f(u+x, v+y)$:

$$f(u+x, v+y) \approx f(u,v) + x\,\partial_x f(u,v) + y\,\partial_y f(u,v)$$

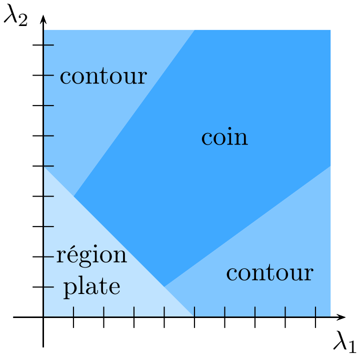
$$\begin{aligned}
\Rightarrow \qquad E_{m,n}(x,y) &= \sum_{u,v} w_{m,n}(u,v)\,\big(f(u+x,v+y) - f(u,v)\big)^2 \\
&\approx \sum_{u,v} w_{m,n}(u,v)\,\big(x\,\partial_x f(u,v) + y\,\partial_y f(u,v)\big)^2 \\
&\approx \begin{pmatrix} x & y \end{pmatrix} M \begin{pmatrix} x \\ y \end{pmatrix}
\end{aligned}$$

où $$M = \sum_{u,v} w_{m,n}(u,v) \begin{pmatrix} (\partial_x f)^2 & \partial_x f\,\partial_y f \\ \partial_x f\,\partial_y f & (\partial_y f)^2 \end{pmatrix}$$

Détecteur de Harris

Le problème P3 peut être évité en considérant une nouvelle mesure de la présence d'un coin : on peut obtenir d'autres informations sur le changement d'intensité dans la fenêtre en analysant les valeurs propres λ_1 et λ_2 de la matrice M.

$$E(x,y) = \begin{pmatrix} x & y \end{pmatrix} M \begin{pmatrix} x \\ y \end{pmatrix}$$



Détecteur de Harris

Le calcul des valeurs propres de M pouvant être difficile, une alternative est de calculer :

$$R = \det(M) - k(\mathrm{trace}(M))^2 = \lambda_1\lambda_2 - k(\lambda_1 + \lambda_2)^2$$

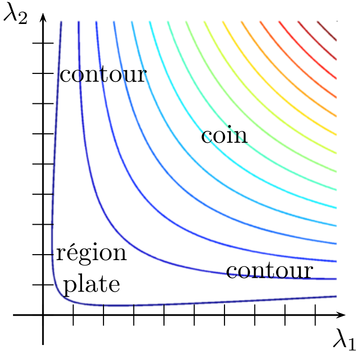
avec $0{,}04 < k < 0{,}06$.

Détecteur de Harris

Le calcul des valeurs propres de M pouvant être difficile, une alternative est de calculer :

$$R = \det(M) - k(\text{trace}(M))^2 = \lambda_1\lambda_2 - k(\lambda_1 + \lambda_2)^2$$

avec $0{,}04 < k < 0{,}06$.

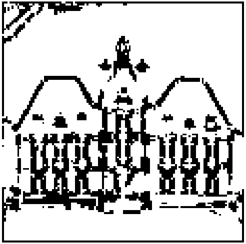


Les valeurs de R sont :

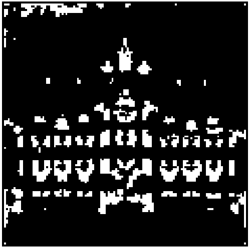
- faibles dans une région plate,
- négatives sur un contour,
- positives sur un coin.

Détecteur de Harris

$R < -10^8$



$|R| < 10^8$

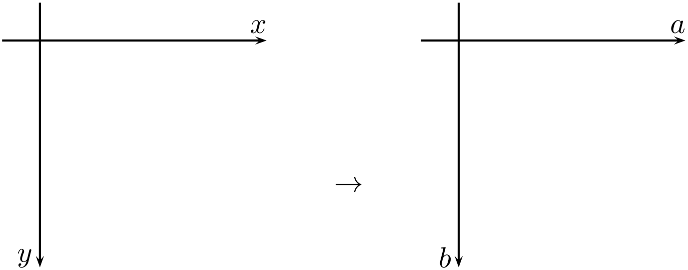


$R > 10^8$

Détection de droites

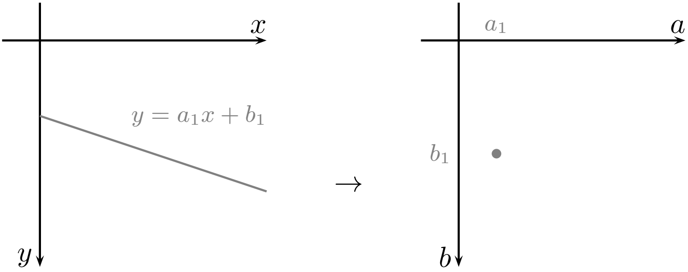
Transformée de Hough pour les droites

L'idée de la transformée de Hough [Hough 1962] est de représenter une droite de l'*image* en un point dans l'*espace des paramètres*.



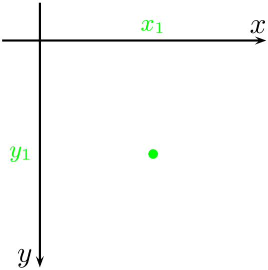
Transformée de Hough pour les droites

L'idée de la transformée de Hough [Hough 1962] est de représenter une droite de l'*image* en un point dans l'*espace des paramètres*.

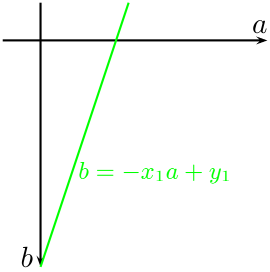


Transformée de Hough pour les droites

L'idée de la transformée de Hough [Hough 1962] est de représenter une droite de l'*image* en un point dans l'*espace des paramètres*.

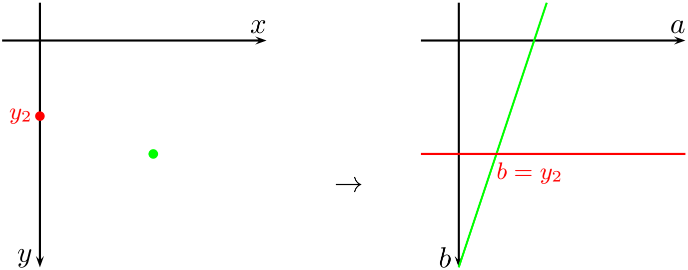


$\rightarrow$



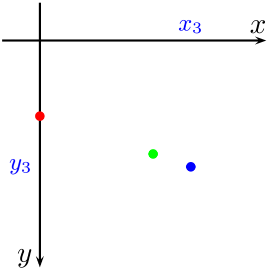
Transformée de Hough pour les droites

L'idée de la transformée de Hough [Hough 1962] est de représenter une droite de l'*image* en un point dans l'*espace des paramètres*.

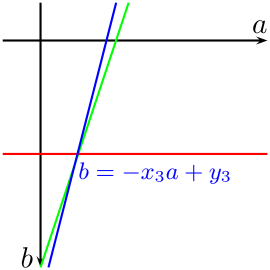


Transformée de Hough pour les droites

L'idée de la transformée de Hough [Hough 1962] est de représenter une droite de l'*image* en un point dans l'*espace des paramètres*.

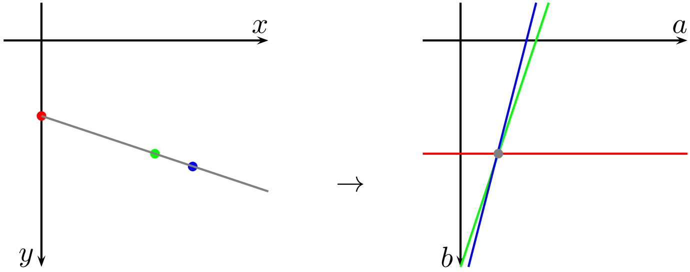


$\rightarrow$



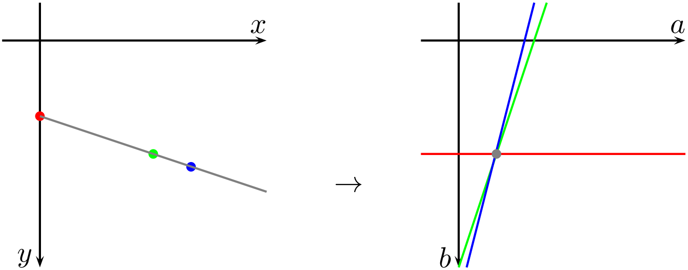
Transformée de Hough pour les droites

L'idée de la transformée de Hough [Hough 1962] est de représenter une droite de l'*image* en un point dans l'*espace des paramètres*.



Transformée de Hough pour les droites

L'idée de la transformée de Hough [Hough 1962] est de représenter une droite de l'*image* en un point dans l'*espace des paramètres*.



Les points d'une droite $y = ax + b$ dans l'image deviennent des droites qui se coupent en (a,b) dans l'espace des paramètres.

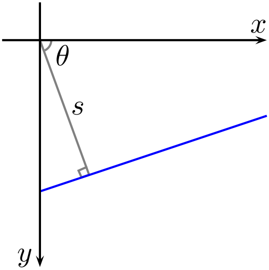
Nouvelle paramétrisation

L'inconvénient de la paramétrisation (a,b) est que l'espace des paramètres doit être borné et discrétisé $\Rightarrow$ une droite verticale ($a = \infty$) ne peut pas être représentée.

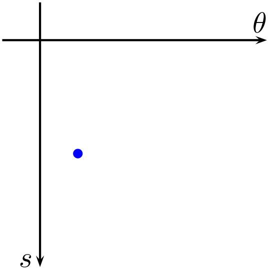
Nouvelle paramétrisation

L'inconvénient de la paramétrisation (a, b) est que l'espace des paramètres doit être borné et discrétisé $\Rightarrow$ une droite verticale $(a = \infty)$ ne peut pas être représentée.

$\Rightarrow$ Nouvelle paramétrisation



$\rightarrow$

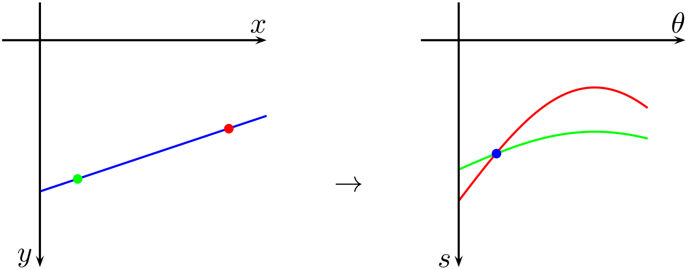


$x = s\cos(\theta)$ et $y = s\sin(\theta) \quad \Rightarrow \quad s = x\cos(\theta) + y\sin(\theta)$

Nouvelle paramétrisation

Pour chaque point (x_i, y_i) de l'image, une sinusoïde est associée dans l'espace (θ, s) :

$$s = x_i \cos(\theta) + y_i \sin(\theta)$$



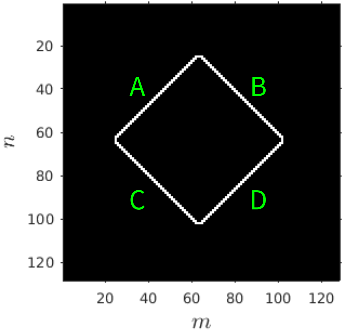
Les sinusoïdes correspondant aux points d'une même droite se coupent au point (s^*, θ^*) paramétrisant cette droite.

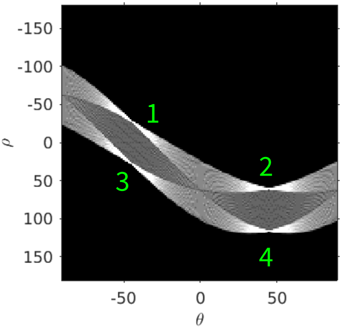
Algorithme

Appliquer une détection de contours
Définir un accumulateur (= espace des paramètres discrétisé)
Pour chaque point des contours :
- Déterminer la sinusoïde correspondante au point
- Incrémenter l'accumulateur le long de cette sinusoïde

Rechercher les maxima de l'accumulateur
En déduire les paramètres des lignes

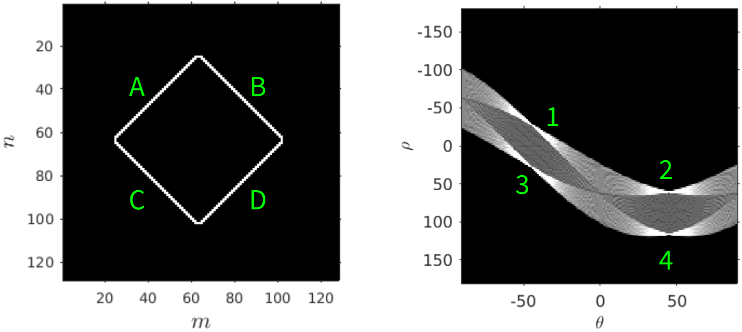
Exemple





wooclap
/mazet

Exemple



/mazet

A→2   B→3   C→1   D→4

Transformée de Hough

Avantages :

- robuste au bruit
- robuste aux occlusions (peut détecter des objets partiellement recouverts)
- Extensible à tout objet paramétré (cercles, ellipses, ...)
  [Duda & Hart 1972]

Transformée de Hough

Avantages :

- robuste au bruit
- robuste aux occlusions (peut détecter des objets partiellement recouverts)
- Extensible à tout objet paramétré (cercles, ellipses, …)

  [Duda & Hart 1972]

  Exemple pour la détection de cercles :

$$(x-a)^2 + (y-b)^2 = r^2 \quad \rightarrow \quad \text{3 paramètres}$$

Transformée de Hough

Avantages :

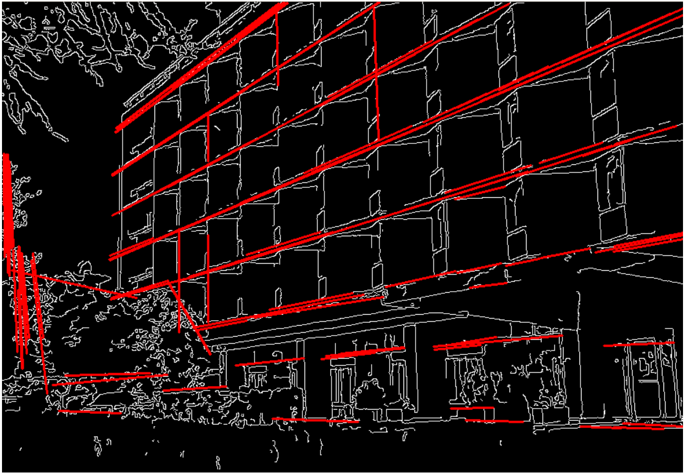
- robuste au bruit
- robuste aux occlusions (peut détecter des objets partiellement recouverts)
- Extensible à tout objet paramétré (cercles, ellipses, …)
  [Duda & Hart 1972]
  Exemple pour la détection de cercles :

$$(x-a)^2 + (y-b)^2 = r^2 \quad \rightarrow \quad \text{3 paramètres}$$

Inconvénient :

- la dimension de l'accumulateur est égal aux nombres de paramètres $\Rightarrow$ le temps de calcul et la mémoire utilisée deviennent vite conséquents

Transformée de Hough



Transformée de Hough

Détection de formes par filtrage adapté

Filtre adapté

La *corrélation croisée* permet de détecter un objet g parfaitement connu (appelé « motif ») dans une image x :

$$R_{x,g}(u,v) = \sum_{m,n} x(m,n)\, g(u+m, v+n)$$

Elle peut s'écrire comme une convolution $\rightarrow$ « Filtre adapté ».

Filtre adapté

La *corrélation croisée* permet de détecter un objet g parfaitement connu (appelé « motif ») dans une image x :

$$R_{x,g}(u,v) = \sum_{m,n} x(m,n) g(u+m, v+n)$$

Elle peut s'écrire comme une convolution $\rightarrow$ « Filtre adapté ».

Généralement, x et g sont remplacées par les images normalisées :

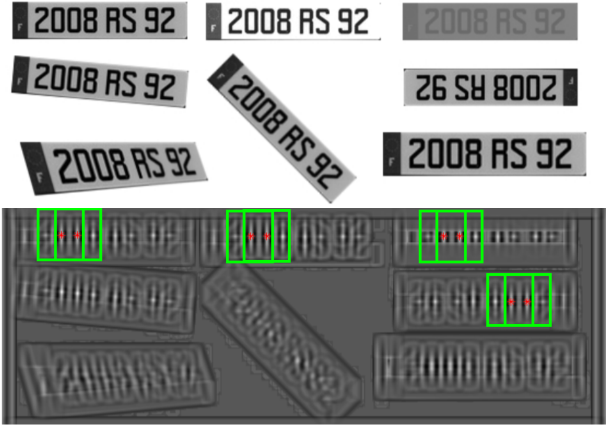
$$x(m,n) \rightarrow \left(x(m,n) - \mu_x\right)/\sigma_x \qquad g(m,n) \rightarrow \left(g(m,n) - \mu_g\right)/\sigma_g$$

où μ_x et σ_x sont respectivement la moyenne et l'écart-type de l'image x. Cela aboutit à la *corrélation croisée normalisée* qui est insensible aux changements d'amplitude.

Filtre adapté

Corrélation croisée normalisée avec le motif **7**.

Filtre adapté



Corrélation croisée normalisée avec le motif 0 .

Filtre adapté

L'inconvénient du filtrage adapté est qu'il est sensible aux variations d'orientation, de taille, etc.

On peut alors appliquer plusieurs filtres représentatifs de toutes les variations du motifs $\Rightarrow$ très coûteux en temps de calcul !

Alternative : trouver des caractéristiques aux objets et effectuer une classification.

Conclusion

Conclusion

Détection de caractéristiques : approches différentes en fonction de la caractéristique cherchée !

- Contour $\rightarrow$ filtrage de l'image en utilisant le gradient ou le laplacien (Roberts, Prewitt, Sobel, Canny ...)
- Coin $\rightarrow$ mesurer les changements d'intensité dans le voisinage des pixels (Moravec, Harris ...)
- Ligne, cercle $\rightarrow$ représenter l'image dans l'espace des paramètres (Hough ...)
- Motif bien connu $\rightarrow$ calculer une intercorrélation (filtre adapté)

Bibliographie

- L.G. Roberts, « Machine Perception Of Three-Dimensional Solids », *Computer Methods in Image Analysis* IEEE Press, 1965.
- J.M.S. Prewitt, « Object enhancement and extraction », *Picture Processing and Psychopictorics*, Academic Press, 1970.
- I. Sobel et G. Feldman, « A 3×3 Isotropic Gradient Operator for Image Processing », In *Stanford Artificial Intelligence Project*, 1968.
- D. Marr et E. Hildreth, « Theory of Edge Detection » *Proceedings of the Royal Society of London* vol. 207, 1980.
- J. Canny, « A Computational Approach To Edge Detection », *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, 1986.
- R.O. Duda et P.E. Hart, « Use of the Hough Transformation to Detect Lines and Curves in Pictures », *Comm. ACM*, 15, p. 11–15, 1972.
- C. Harris, M. Stephens « A combined corner and edge detector », *actes de l'Alvey Vision Conference*, p. 147–151, 1988.
- P.V.C. Hough, *Method and means for recognizing complex patterns*, US Patent 3,069,654, 1962.
- H. Moravec, *Obstacle Avoidance and Navigation in the Real World by a Seeing Robot Rover*, rapport technique, Carnegie–Mellon University, Robotics Institute, 1980.