

# Vision par Ordinateur

James L. Crowley

DEA IVR

Premier Bimestre 2006/2007

Séance 3

17 Octobre 2006

## Déscription d'images par analyse de contraste

### Plan de la Séance :

Description de Contraste .....	2
Le Détecteur de Contraste de Roberts.....	3
Le détecteur de Sobel.....	4
Les filtres de différence .....	5
Lissage : Les Filtres Binomiaux .....	6
La fonction de transfert des filtres binomiaux.....	7
Détection de contrastes par dérivées.....	8
Détection des Pics dans le module du Gradient.....	9
Transformée de Hough .....	10
Espace de Hough .....	11
Généralisation de Transformée de HOUGH.....	12
Les Champs Réceptifs Gaussiens .....	13
Filtres et dérivées.....	13
La fonction Gaussien.....	14
Les dérivées de la Gaussien.....	15
Les operateurs Differentielles.....	15
Les Filtres Numérique Gaussien.....	16
Les Dérivées de l'Image.....	18
L'Orientabilité des Gaussiens.....	18
L'espace d'échelles .....	19
Echelle Intrinsèque (ou caractéristique).....	20

## Description de Contraste

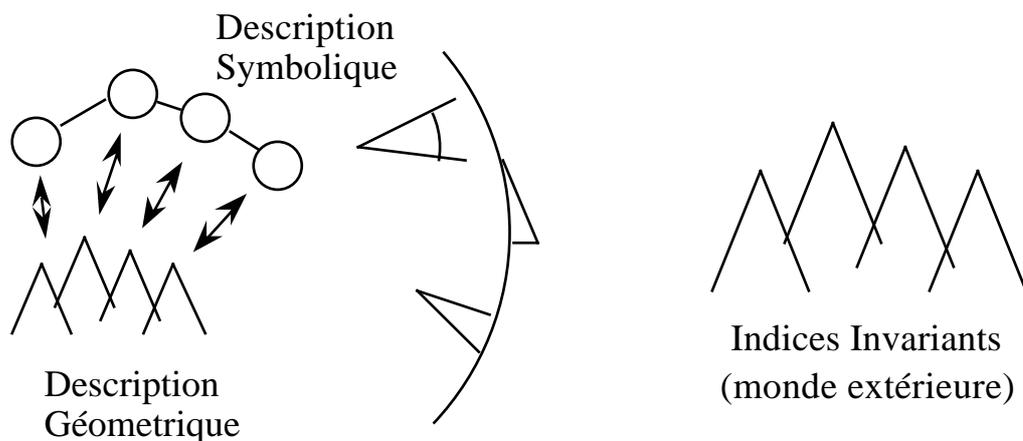
Le "Contraste" est les variations en intensité d'un image.

L'Intensité ou Luminance d'un image est déterminé par l'orientation de la surface.

$$R(i, e, g, ) \quad \text{Cos}(i)$$

Les variations de normale sont traduite par les variations en intensité.

Pour mesurer les variations, il faut mesurer les dérivés.



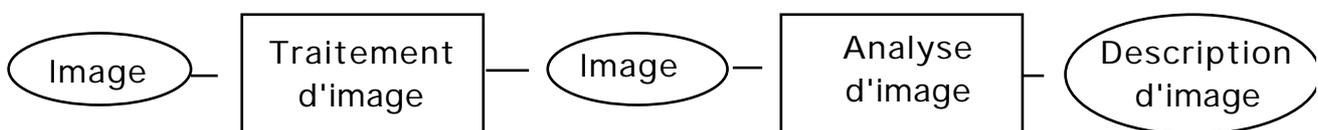
Objectif : Une description stables quel que soit le point de vue et la luminosité ambiante.

La première phase d'un processus de vision est l'extraction des "Indices"

Les Indices d'image servent à :

- 1) Réduire la quantité d'informations (coût mémoire, calcul et communication)
- 2) Simplifier l'interprétation

La détection de point de contraste s'organise en deux étapes :



## Le Détecteur de Contraste de Roberts

2-D : Détecteur de Roberts (1962)

$$m_1(i, j) = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

$$m_2(i, j) = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$$

Corrélation : Pour  $n = 1, 2$

$$E_n(i, j) = p * m_n = \sum_{k=0}^1 \sum_{L=0}^1 p(i-k, j-L) m_n(k, L)$$

Module de contraste :

$$E(i, j) = \sqrt{E_1(i, j)^2 + E_2(i, j)^2}$$

Direction de contraste (phase) :

$$(i, j) = \text{Tan}^{-1}\left(\frac{E_2(i, j)}{E_1(i, j)}\right) + \frac{\pi}{4}$$

Mais le détecteur de Roberts est très sensible au bruit de haute fréquence de source électronique et photo-optique.

Un tel bruit peut être réduit par un filtrage passe bas .

Pour garder la symétrie (fonction paire), réponse impulsionnelle considérée à  $\pm \frac{1}{2}$

$$\begin{aligned} M_1(u, v) &= \sum_{m=-1/2}^{1/2} \sum_{n=-1/2}^{1/2} m_1(m, n) e^{-j(mu+nv)} \\ M_1(u, v) &= +1 e^{j(0.5u + 0.5v)} - 1 e^{-j(0.5u + 0.5v)} \\ M_1(u, v) &= 2j \sin(0.5u + 0.5v) \end{aligned}$$

## Le détecteur de Sobel

(Duda - Hart 1972) :

Un détecteur de contraste très populaire.

$$m_0(i, j) = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} \qquad m_{90}(i, j) = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

Corrélation : Pour  $n = 1, 2$

$$E_n(i, j) = p * m_n = \sum_{k=0}^1 \sum_{L=0}^1 p(i-k, j-L) m_n(k, L)$$

Comme avec Robert, le module et l'orientation sont calculés par le module et la direction :

Module du contraste :

$$E(i, j) = \sqrt{E_0(i, j)^2 + E_{90}(i, j)^2}$$

Direction de contraste (phase) :

$$\theta(i, j) = \text{Tan}^{-1}\left(\frac{E_{90}(i, j)}{E_0(i, j)}\right)$$

Ce filtre peut être vu comme une convolution de deux composantes :

$$m_0(i, j) = \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} \begin{bmatrix} -1 & 0 & 1 \end{bmatrix} \qquad m_{90}(i, j) = \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix} \begin{bmatrix} 1 & 2 & 1 \end{bmatrix}$$

(lissage)    (dérivée)                      (dérivée)    (lissage)

## Les filtres de différence

Pour une fonction,  $s(t)$

$$\frac{ds(t)}{dt} = \lim_{t \rightarrow 0} \left\{ \frac{s(t) - s(t-t)}{t} \right\}$$

Pour un signal numérique,  $s(n)$ , la limite n'existe pas.

$$n = 2 : \frac{s(n)}{n} = \frac{s(n) - s(n-2)}{2}$$

Ceci est équivalent à  $s(n) * [\frac{1}{2}, 0, -\frac{1}{2}] = s(n) * (\frac{1}{2} \cdot [1, 0, -1])$

$$n = 1 : \frac{s(n)}{n} = \frac{s(n) - s(n-1)}{1}$$

Ceci est équivalent à  $s(n) * [1, -1]$

L'alternatif au dérivées de Gaussienne est les filtres de différence.

Ils ont l'avantage de taille.

$$i = \overline{i} = \begin{bmatrix} -1 & 0 & 1 \end{bmatrix} \quad j = \overline{j} = \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix}$$

$$P(i, j) = \frac{\frac{p(i,j)}{i}}{\frac{p(i,j)}{j}} = \frac{m_0 * p(i,j)}{m_{90} * p(i,j)} = \frac{E_0(i,j)}{E_{90}(i,j)}$$

$$\text{Magnitude} = \| P(i, j) \| = E(i, j) = \sqrt{E_1(i, j)^2 + E_2(i, j)^2}$$

$$\text{Orientation} = (i, j) = \text{Tan}^{-1}\left(\frac{E_2(i, j)}{E_1(i, j)}\right)$$

## Lissage : Les Filtres Binomiaux

Le suite binomial est composé des coefficients du polynôme :

$$(x + y)^n = \sum_{m=-n/2}^{n/2} b_{m,n} x^{n-m} y^m$$

$$b_{m,n} = b_n(m) = [1, 1]^{*n} = \frac{n!}{m!(n-m)!}$$

Les coefficients du suite binomial sont générés par le triangle de Pascal :

n	sum = 2 <sup>n</sup>	μ = n/2	σ <sup>2</sup> = n/4	σ = √(n/2)	Coefficients
0	1	0	0	0	1
1	2	0.5	0.25		1 1
2	4	1	0.5		1 2 1
3	8	1.5	0.75		1 3 3 1
4	16	2	1	1	1 4 6 4 1
5	32	2.5	1.25		1 5 10 10 5 1
6	64	3	1.5		1 6 15 20 15 1
7	128	3.5	1.75		1 7 21 35 35 21 7 1
8	256	4	2	√2	1 8 29 56 70 56 29 8 1

Ces coefficients forment des filtres numérique avec des propriétés remarquables. Ils sont les coefficients de la meilleure approximation du filtre Gaussien sujets aux contraintes de coefficients entier et durée finie.

Filtres binomiaux :  $b_n(m) = [1, 1]^{*n} = n$  convolution de  $[1, 1]$

Gain :  $S = b_n = 2^n$

Position Centrale (1er Moment)  $\mu_n = \frac{1}{S} \sum_{m=0}^n b_n(m) \cdot m = \frac{n}{2}$

Variance (2ieme Moment) :

$$\sigma_n^2 = \frac{1}{S} \sum_{m=0}^{n-1} b_n(m) (m - \mu_n)^2 = \frac{1}{2^n} \sum_{m=0}^{n-1} b_n(m) (m - \frac{n}{2})^2 = \frac{n}{4}$$

exemple:

$$\text{Var}\{[1, 1]\} = \frac{1}{2} \{ (1 (\frac{-1}{2})^2 + 1 (\frac{1}{2})^2) \} = \frac{1}{2} (\frac{1}{4} + \frac{1}{4}) = \frac{1}{4}$$

## La fonction de transfert des filtres binomiaux

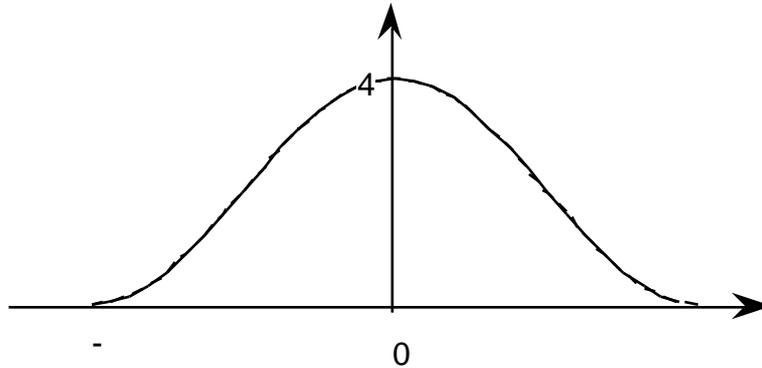
Les filtres binomiaux sont des filtres Gaussiens "finis et discrets" de taille  $n^2 = n/4$ .

Fonction de Transfert :  $B_n(\omega) = [2 \cos(\frac{\omega}{2})]^n$

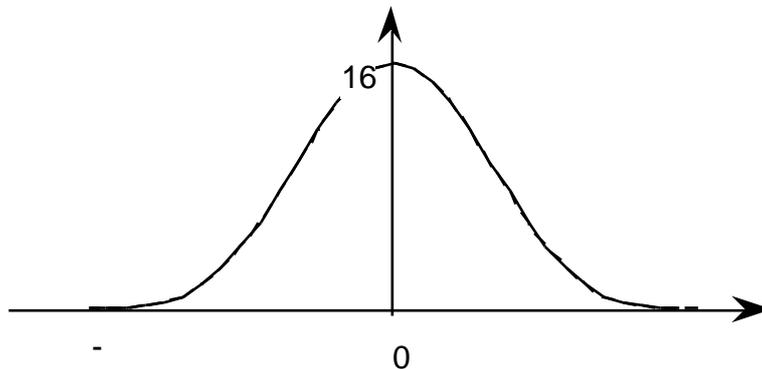
Pour n paire :  $B_n(\omega) = [2 + 2 \cos(\omega)]^{n/2}$

La fonction de transfert des binomiaux peut être calculée facilement à la main  
Exemple

$$\begin{aligned}
 b_2(m) &= [1 \ 2 \ 1] \quad (\text{Deuxième filtre Binomial}) \\
 B_2(\omega) &= 1 e^{j\omega(-1)} + 2 e^{j\omega(0)} + 1 e^{j\omega(1)} \\
 &= 2 + e^{j\omega} + e^{-j\omega} \\
 B_2(\omega) &= 2 + 2 \cos(\omega) = [2\cos(\omega/2)]^2
 \end{aligned}$$



$$\begin{aligned}
 b_4(m) &= [1 \ 1]^{*4} = [1 \ 4 \ 6 \ 4 \ 1] \\
 B_4(\omega) &= 6 + 8 \cos(\omega) + 2 \cos(2\omega) = [2 + 2 \cos(\omega)]^2
 \end{aligned}$$

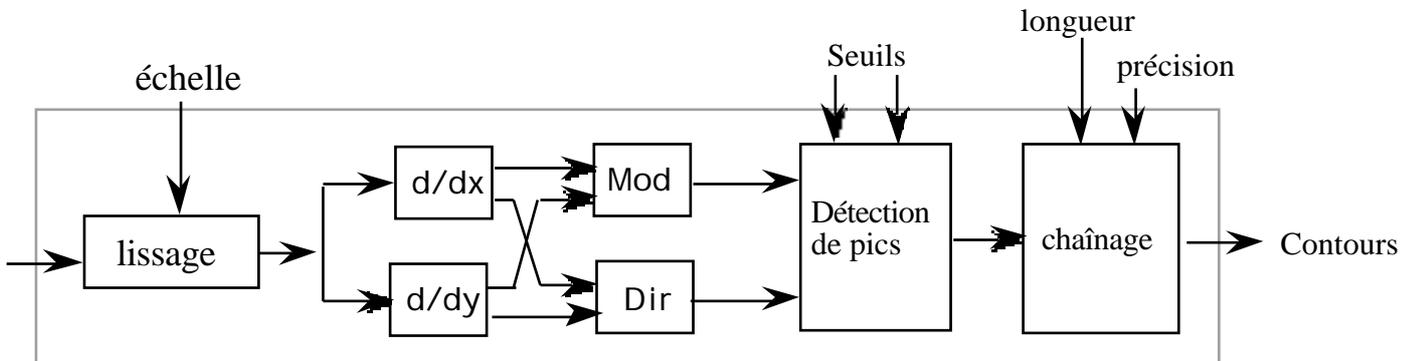


En 2D, les binomiaux sont séparables et symétrique circulaire.

$$\text{en 2-D } b_2(i, j) = \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} \begin{bmatrix} 1 & 2 & 1 \end{bmatrix}$$

## Détection de contrastes par dérivées

Schéma du processus de description de contraste



$$P(i, j) = \frac{\frac{p(i,j)}{i}}{\frac{p(i,j)}{j}} = \frac{m_0 * p(i,j)}{m_{90} * p(i,j)} = \frac{E_0(i,j)}{E_{90}(i,j)}$$

$$\text{Module} = \| P(i, j) \| = E(i, j) = \sqrt{E_1(i, j)^2 + E_2(i, j)^2}$$

$$\text{Direction} = (i, j) = \text{Tan}^{-1}\left(\frac{E_2(i, j)}{E_1(i, j)}\right)$$

### Étapes :

- 1) Lissage : suppression du bruit, conditionnement du signal
- 2) Calcul de gradient magnitude et orientation
- 3) Détection des maxima locaux avec seuillage par hystérésis.
- 4) Chaînage des points
- 5) Approximation polygonale : par aire, orientation, etc...

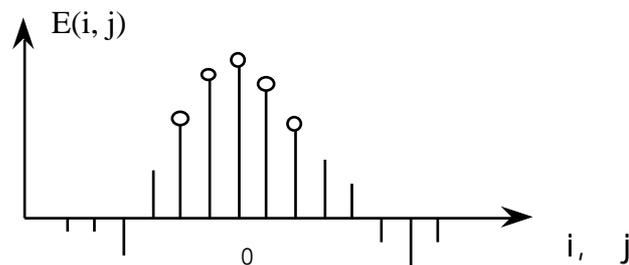
## Détection des Pics dans le module du Gradient

Points de contraste : un extremum local en  $E(i, j)$ .

les points de contraste :  $C(i, j)$   
 pour le gradient de la magnitude  $E(i, j)$  et orientation  $(i, j)$

Sur chaque point :

- 1) Déterminer le vecteur de la direction du maximum du gradient
- 2) Déterminer si le pixel  $E(i, j)$  est un extremum dans la direction du gradient



$$C(i, j) = \begin{cases} 1 & \text{si } F(i, j) > 0 \text{ et} \\ & E(i-1, j-1) < E(i, j) < E(i+1, j+1) \\ 0 & \text{Sinon.} \end{cases}$$

ensuite :

- Construire un graphe des "chaînes" des points de contraste
- Décrire les chaînes avec les segments et courbes.

Techniques:

- 1) Balayage des lignes et colonnes avec une extraction
- 2) Suivi des "crêtes" de contraste
- 3) La transformé de Hough

## Transformée de Hough

(Duda et Hart, 1973)

Une technique "optimale" pour détecter les droites dans les images très bruitées.

Cette technique ne dépend pas de la continuité des droites.

Cependant, elle fournit des droites, pas des segments.

Une équation de droite s'exprime comme :

$$y = m x + c \quad \text{ou} \quad m = \frac{(y_2 - y_1)}{(x_2 - x_1)} \quad c = y_1 - m x_1$$

Pour chaque point  $(x, y)$  de l'image, il y a un ensemble de valeurs possibles pour les paramètres  $m$  et  $c$ .

Cet ensemble forme une droite d'équation  $c = -m x + y$  dans l'espace  $(m, c)$  appelé aussi espace de Hough.

Si des points de contrastes de l'image sont alignés, les droites correspondantes de l'espace de Hough passent toutes par un même point  $(m, c)$ .

### Calcul :

La transformée utilise un tableau  $h(m, c)$  de "cellules".

Les cellules de  $h(m, c)$  sont initialisées à 0.

Pour chaque point  $(i, j)$  de l'image

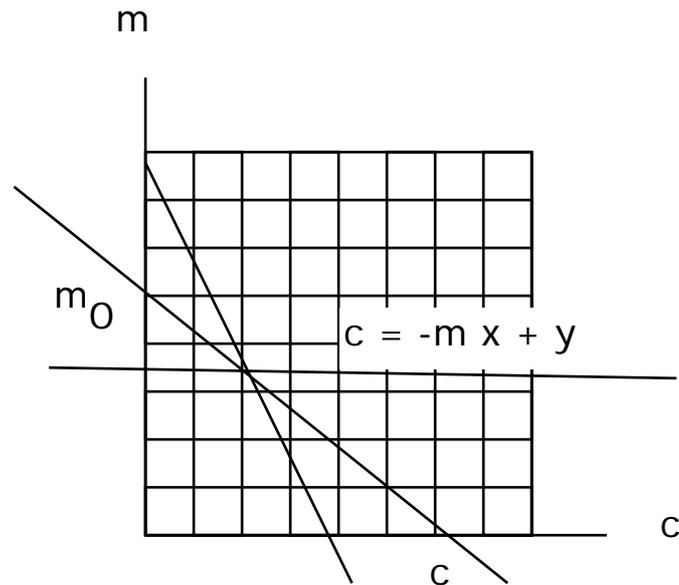
si  $C(i, j) = 1$ , on repère la droite des couples  $(m, c)$  possibles

$$j = m i + c.$$

On ajoute 1 à chaque cellule de  $h(m, c)$  dans laquelle passe cette droite.

Un maximum local de  $h(m, c)$  indique que des points de l'image sont alignés suivant la droite de paramètres correspondants.  $(m, c)$

## Espace de Hough



Algorithme :

Pour chaque (i, j)

Si  $C(i, j) = 1$ :

BEGIN

Calculer les points d'interception avec les axes m, c:

$$m_0 = \frac{-j}{i}, \quad c_0 = j$$

Pour tous les points (m, c) de la droite ((m<sub>0</sub>, 0), (0, c<sub>0</sub>))

$$h(m, c) := h(m, c) + 1$$

END

Les Maximum locaux de  $h(m, c)$  correspondent aux droites (m, c) de l'image.

Problème : les coefficients m, c ne sont pas uniformes en .

Solution : On utilise les paramètres ( , ) de l'équation

$$i \sin( ) - j \cos( ) + = 0.$$

## Généralisation de Transformée de HOUGH

Pour les cercles : une équation de cercle s'écrit :

$$(x - a)^2 + (y - b)^2 = r^2$$

On considère l'espace de Hough  $h(a, b, r)$ .

Chaque point  $(x, y)$  de l'image correspond à un cône de l'espace  $(a, b, r)$ .

Pour un rayon fixé, chaque point  $(x, y)$  correspond à un cercle de l'espace  $(a, b, r)$ .

Idée de l'algorithme :

Pour chaque rayon  $r > 0$  on trace les cercles de l'espace de Hough correspondant aux points de l'image.

Lorsque tous les cercles se coupent en un même point, on a trouvé le bon rayon, les coordonnées  $(a, b)$  de ce point correspondent au centre du cercle.

## Les Champs Réceptifs Gaussiens

### Filtres et dérivées

La produit scalaire est une projection d'un signal sur une fonction.

$$\langle s, f(k) \rangle = \sum_{i=-} s(i) f(i+k)$$

Une evaluation de la produit scalaire à chaque pixel est un "convolution".

$$r(k) = s * f(i) = \langle s(i), f(k-i) \rangle = \sum_{i=-} s(k-i) f(i) = \sum_{i=-} s(i) f(k-i)$$

Pour calculer la dérivé d'un signal numérique, on calcule la fonction de la dérivé d'un noyau,  $g(t)$ . ensuite, on échantillon

$$f(t) = \frac{dg(t)}{dt}$$

$$\frac{ds(n)}{dt} = f(nT_e) * s(n)$$

L'idéal serait une fonction invariante au tranformation projective. Mais ceci n'est pas possible. Mais on peut trouver une noyau invariante aux transformations affines - La fonction Gaussien

**La fonction Gaussien**

La fonction Gaussien est  $G(x, y) = e^{-\frac{1}{2} \frac{x^2+y^2}{z_c}}$

La fonction Gaussienne est invariante à la transformation affine:

$$T_a\{G(x, y)\} = G(T_a\{x\}, T_a\{y\})$$

Rappel en séance 2 on a vu que  $x_r = x_c \frac{F}{z_c}$ .

Donc la "taille" d'un objet est en proportion de  $s = \frac{F}{z_c}$ .

La taille (ou échelle) est une paramètre de la transformation affine.

La fonction Gaussienne est invariante à la transformation d'échelle :

$$T_s\{G(x, y)\} = G(T_s\{x\}, T_s\{y\}) = G(sx, sy)$$

Si on divise  $z_c$  (distance entre la caméra et l'objet) par deux, on double la taille.

$$G(x, y) = G(2x, 2y)$$

En 2-D, la Gaussienne symétrique circulaire est  $G(x, y) = e^{-\frac{x^2+y^2}{2A}}$  avec  $A = \frac{z_c^2}{2F^2}$

La Gaussien est la fonction unique qui est symétrique circulaire et séparable.

On note que  $G(x, y) = e^{-\frac{x^2+y^2}{2A}} = e^{-\frac{x^2}{2A}} * e^{-\frac{y^2}{2A}}$

Ceci nous offre beaucoup d'intérêt pour la vitesse de calcul.

## Les dérivées de la Gaussien

$$G(x, y) = e^{-\frac{1}{2} \frac{x^2 + y^2}{\sigma^2}}$$

$$G_x(x, y) = -\frac{x}{\sigma^2} G(x, y)$$

$$G_{xx}(x, y) = \left( \frac{x^2}{\sigma^4} - \frac{2}{\sigma^2} \right) G(x, y)$$

$$G_{xxx}(x, y) = -\frac{3x^3 - x}{\sigma^6} G(x, y)$$

On note que le "gain" d'un noyau est son intégral.

$$A = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} e^{-\frac{1}{2} \frac{x^2 + y^2}{\sigma^2}} dx dy = \sqrt{2\pi} \sigma^2$$

La propriété de l'invariance à l'échelle demande que  $A = 1$ .

## Les opérateurs Différentielles

Pour un signal en deux dimensions, les opérations différentielles sont le gradient et la Laplacien <sup>2</sup> :

Le gradient est un vecteur :

$$\nabla = \begin{pmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \end{pmatrix}$$

La Laplacien est un scalaire :  $\nabla^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$

Pour une fonction  $s(x, y)$  :

Le Gradient :

$$\nabla s(x, y) = \begin{pmatrix} \frac{\partial s(x, y)}{\partial x} \\ \frac{\partial s(x, y)}{\partial y} \end{pmatrix}$$

La Laplacien

$$\nabla^2 s(x, y) = \frac{\partial^2 s(x, y)}{\partial x^2} + \frac{\partial^2 s(x, y)}{\partial y^2}$$

si  $s(x,y)$  est filtré par  $G(x,y; \sigma)$  :  $s * G(x,y; \sigma) = \int_{s=-\infty}^{\infty} \int_{t=-\infty}^{\infty} s(x-s, y-t) G(x-s, y-t) dx dt$

La Gradient :  $(s * G(x,y; \sigma))_x = s * \frac{\partial G(x,y; \sigma)}{\partial x}$   
 $(s * G(x,y; \sigma))_y = s * \frac{\partial G(x,y; \sigma)}{\partial y}$

La Laplacien  $\Delta(s * G(x,y; \sigma)) = s * \Delta G(x,y; \sigma) = s * \left( \frac{\partial^2 G(x,y; \sigma)}{\partial x^2} + \frac{\partial^2 G(x,y; \sigma)}{\partial y^2} \right)$

### Les Filtres Numérique Gaussien

Parce que l'image est un signal échantillonné, il faut échantillonner  $G(x, y, \sigma)$

On obtient les filtres numériques par un simple échantillonnage de la fonction Gaussienne sur un intervalle  $[-R, R]$ .

On remplace  $x$  par  $nT_e$ , ou  $T_e$  est un

$$G(n) = G(nT_e, \sigma) = e^{-\frac{1}{2} \frac{n^2 T_e^2}{\sigma^2}}$$

$T_e$  est la pas d'échantillonnage.

Par convention l'on considère  $T_e = 1$ .

Donc, la forme numérique est  $G(n, \sigma) = e^{-\frac{1}{2} \frac{n^2}{\sigma^2}}$

$$A = \int_{n=-R}^R e^{-\frac{1}{2} \frac{n^2}{\sigma^2}} dx \sqrt{2\pi} \sigma$$

Il y a deux facteurs à maîtriser :

- a) La taille de la "support"  $N = 2R+1$ .
- b) la ratio  $\sigma/T_e$

Pour a:

Pour  $N = 7$ , les "ondes" de  $W_R(f)$  dominant le spectre.

Pour  $N = 9$ , les ondes, on peut d'effet.

Pour b:

Il vaut mieux que  $\sigma/T_e = 1$

Les dérivées de la Gaussienne numérique sont :

$$G_x(n, ) = -\frac{n}{2} G(n, )$$

$$G_{xx}(n, ) = \frac{n^2 - 2}{4} G(n, )$$

$$G_{xxx}(n, ) = -\frac{n^3 - n}{6} G(n, )$$

Pour la Gaussien en 2D.  $G_x(i, j, )$

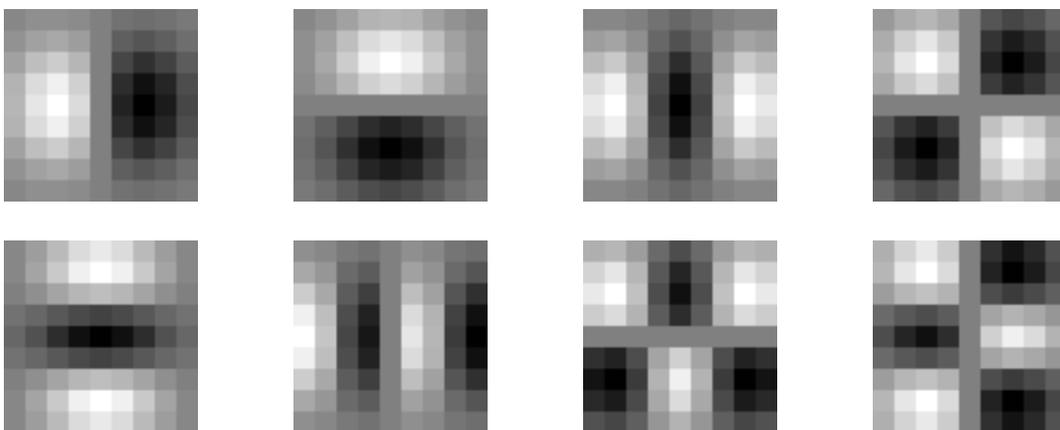
$$G_x(i, j, ) = -\frac{i}{2} G(i, j, ) = -\frac{i}{2} G(i, ) * G(j, )$$

$$G_{xy}(i, j, ) = \frac{i}{2} \frac{j}{2} G(i, j, ) = -\frac{i}{2} G(i, ) * -\frac{j}{2} G(j, )$$

Un vecteur de champs réceptifs forme une base de Taylor

$$G_a = (G_x, G_y, G_{xx}, G_{xy}, G_{yy}, G_{xxx}, G_{xxy}, G_{xyy}, G_{yyy})$$

Ceci donne la famille de champs réceptifs Gaussien



Les champs réceptifs Gaussien  $G_x, G_y, G_{xx}, G_{xy}, G_{yy}, G_{xxx}, G_{xxy}, G_{xyy}, G_{yyy}$ .

Note qu'il y a une paramètre  $\sigma$ . Ceci est la paramètre d'echelle. Ce détermine la limite de la résolution d'une description.

## Les Dérivées de l'Image

Pour l'image  $p(i, j)$ ,  $\nabla p(i, j)$  est calculé par  $\nabla G(i, j) * p(i, j)$ .

$$\text{ou } \nabla p(i, j) = \begin{pmatrix} G_x(i, j) \\ G_y(i, j) \end{pmatrix}$$

$$\text{Gradient: } \nabla p(x, y) = \begin{pmatrix} G_x(x, y) * p(x, y) \\ G_y(x, y) * p(x, y) \end{pmatrix}$$

(Le gradient est un vecteur).

$$\text{Laplacien: } \nabla^2 p(i, j) = G_{xx}(i, j) * p(i, j) + G_{yy}(i, j) * p(i, j)$$

(Le Laplacien est une scalaire.)

## L'Orientabilité des Gaussiens

Pour chaque pixel, on peut calculer une orientation "intrinsic"

$$\theta(x, y) = \text{Atan2}\{\langle A(x, y), G_y \rangle, \langle A(x, y), G_x \rangle\}$$

Les réponses de filtres orientées peuvent être calculé par une somme de réponse des filtres de base, pondérée par les sinus et cosinus.

$$G_1^\theta = \cos(\theta) G_x + \sin(\theta) G_y$$

$$G_2^\theta = \cos(\theta)^2 G_{xx} + 2 \cos(\theta) \sin(\theta) G_{xy} + \sin(\theta)^2 G_{yy}$$

$$G_3^\theta = \cos(\theta)^3 G_{xxx} + 3 \cos(\theta)^2 \sin(\theta) G_{xxy} + 3 \cos(\theta) \sin(\theta)^2 G_{xyy} + \sin(\theta)^3 G_{yyy}$$

## L'espace d'échelles

La fonction Gaussian est invariante à la transformation d'échelle :

$$T_s\{G(x, y)\} = G(T_s\{x\}, T_s\{y\})$$

On a vu que la "taille" d'un objet est en proportion de  $\frac{F}{z_c}$ .

Si on multiplie par "s" la taille.

$$G(sx, sy) = G(x, y)$$

Mais, on concrète, quelle échelle faut-il utiliser ? tous !

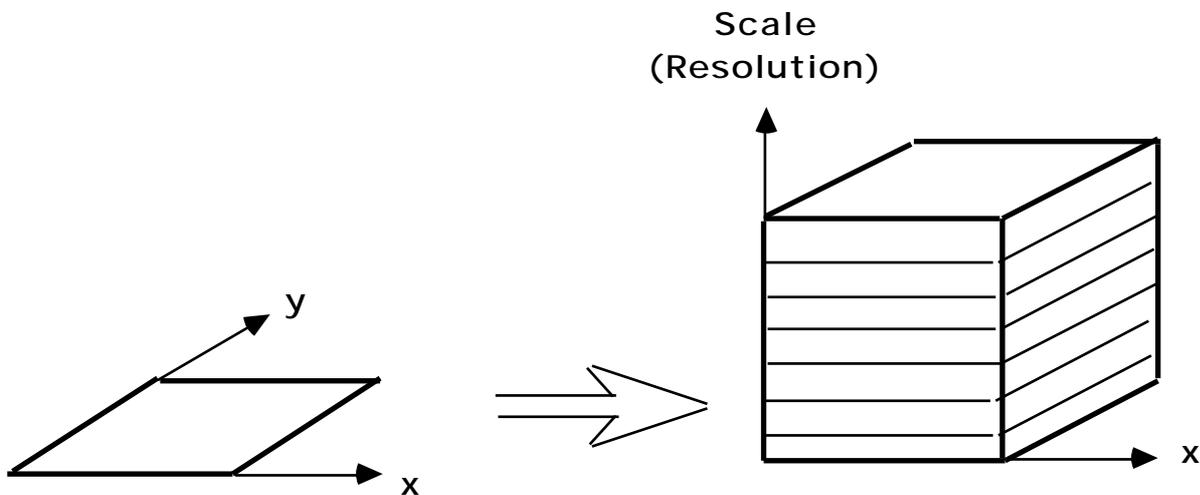
Un espace d'échelle,  $P(x, y, s)$  est défini par un noyau,  $g(x, y; s)$  avec le paramètre  $s$  libre

$$P(x, y, s) = g(x, y, s) * p(x, y)$$

pour  $X_{\min} \leq x \leq X_{\max}, Y_{\min} \leq y \leq Y_{\max}, s_{\min} \leq s \leq s_{\max}$ ,

ou  $s$  est le paramètre d'échelle.

Pour une description invariant à l'échelle, il faut un axe logarithmique pour  $s$ .



La description de l'image se trouvent à toutes les échelles.

L'espace d'échelle facilite la recherche de correspondance grâce à la décomposition des formes à divers niveaux de résolution :

- |                    |   |
|--------------------|---|
| Résolution basse : | peu de formes,<br>vue globale,<br>détails grossiers |
| Résolution haute : | nombreux formes,<br>vue locale,<br>détails fins.    |

Propriétés de l'espace d'échelle :

- Invariance (Equivariance) aux changements de taille
- Le bruit de numérisation se trouve dans les hautes fréquences (donc petit  $\sigma$ ).
- Les points de contraste dans les moyennes fréquences sont souvent les plus stables,

L'espace d'échelle est une idéal mathématique. Pour calculer il faut travailler sur les valeurs numériques. Ceci implique une échantillonnage en  $x$ ,  $y$ , et  $\sigma$ .

Le Gaussian est une solution de la Equation de Diffusion.

Equation de Diffusion:  $\frac{\partial^2 G(x,y; \sigma)}{\partial x^2} = -\frac{G(x,y; \sigma)}{\sigma^2}$

En conséquence,  $\frac{\partial^2 G(x,y; \sigma)}{\partial x^2} = G(x,y; \sigma_1) - G(x,y; \sigma_2)$

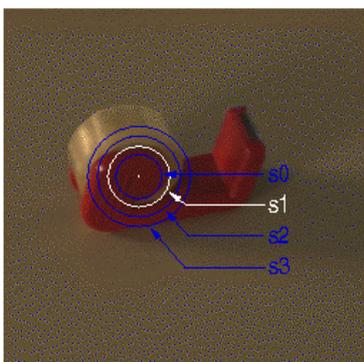
pour  $\sigma_1 = \sqrt{2} \sigma_2$

### Echelle Intrinsèque (ou caractéristique)

Considère la Laplacienne pour à la pixel  $x$ ,  $y$ , en fonction de

$$\Delta^2 p(i,j) = \langle G_{xx}(i, j, \sigma), p(i, j) \rangle + \langle G_{yy}(i, j, \sigma), p(i, j) \rangle$$

A chaque point de l'image il y a quelques valeurs de  $\sigma$  pour laquelle la  $\Delta^2 p(i,j)$  sont maximales.



zero crossing of Laplacian at  $s_1$

