3A SRI

Vision et reconnaissance des formes dans les images

philippe.joly@irit.fr

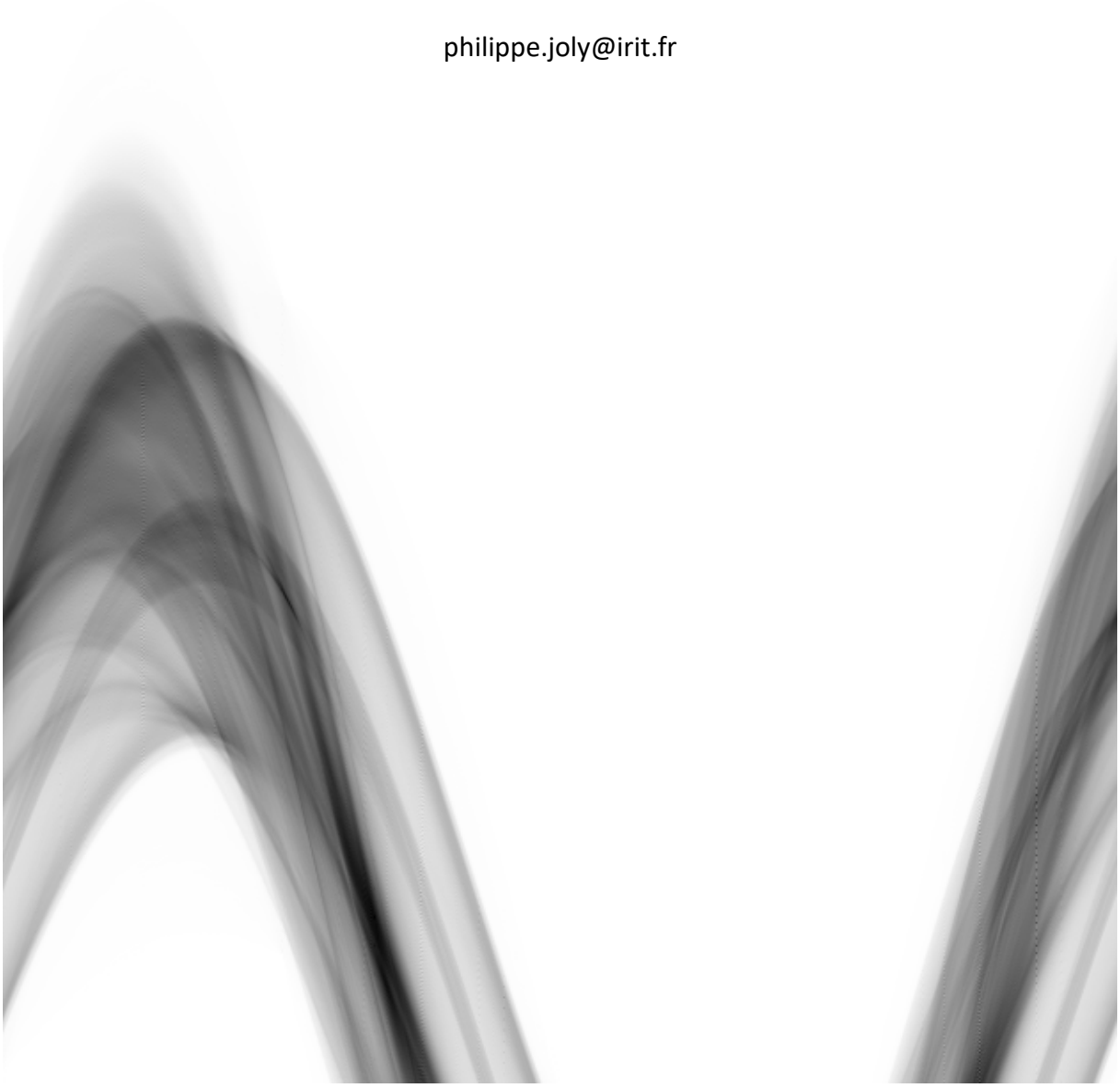


Table des matières

I. Métrologie – Caractérisation

A. Caractérisation de régions

1. Couleur

a) Histogrammes

Cf. cours 2A SRI

b) Espaces de couleur

Cf. cours 2A SRI

2. Texture

a) Transformée de Fourier – Filtres de Gabor

➔ **Transformée de Fourier Discrète directe et inverse,**

Dans le domaine discret, le calcul de la transformée de Fourier s'exprime par :

$$\hat{F}_\omega = \sum_{t=0}^{N-1} F_t e^{-2i\pi\frac{\omega t}{N}}$$

Le calcul de la transformée inverse est donné par :

$$F_t = \frac{1}{N}\sum_{\omega=0}^{N-1} \hat{F}_\omega e^{2i\pi\frac{\omega t}{N}}$$

Exemple :

X	1	3	8	4	5	1	2	6
TFD(x)	30	-1.1716 - 6i	-4 + 6i	-6.8284 + 6i	2	-6.8284 – 6i	-4 - 6i	-1.1716 + i
|TFD(x)|	30	6.1133	7.2111	9.09	2	9.09	7.2111	6.1133

Remarquons que :

- le premier coefficient de la TFD est réel. Il est égale à la somme des échantillon (en pratique c'est N.moyenne(x)). On appelle généralement ce coefficient dans les transformée fréquentielle, le coefficient « DC » ou « composante continue ».

Exercice : Remplir le tableau ci-dessous :

X	1	5	1	5	1	5	1	5
TFD(x)								
|TFD(x)|								

II.3.2.4 Transformée de Fourier 2D

La transformée de Fourier 2D d'une image est obtenue en appliquant la TFD sur chaque ligne indépendamment les unes des autres. Puis on applique la TFD sur chaque colonne du résultat obtenu.

➔ Notions de Systèmes et de Filtres

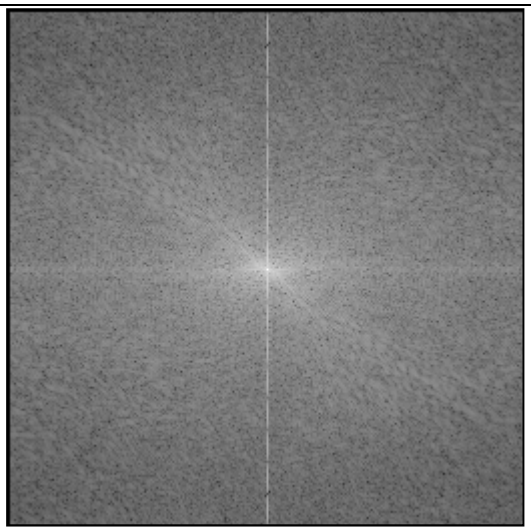
Une fois qu'on a calculé le spectre, il est possible de produire un filtrage fréquentiel en annulant ou en atténuant les amplitudes de certaines des fréquences. On distingue les filtres :

- passe-haut / coupe-bas : qui ne conserve que les hautes fréquences
- passe-bas / coupe-haut
- passe-bande / coupe bande : qui ne conserve (resp. coupe) que les informations d'un intervalle fréquentiel donné.

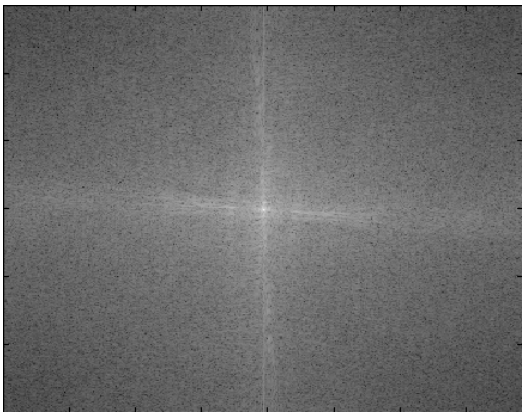
➔ Représentation fréquentielle d'image

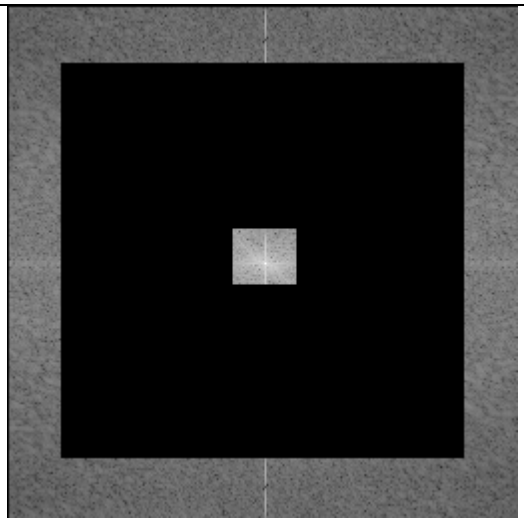
Il est souvent d'usage en traitement d'image d'effectuer une permutation des quadrants pour ramener les basses fréquences (BF) au centre) afin de simplifier les calculs de filtrage notamment.

Image de départ (Lena)



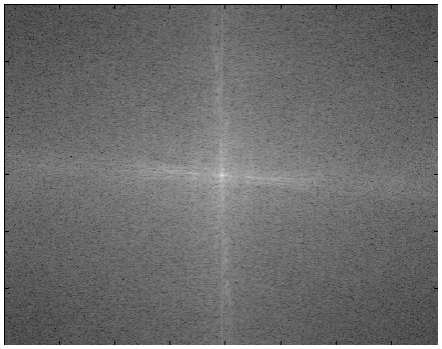
TFD 2D
(log du module normalisé - BF au centre)





Filtre coupe-bande

TFD inverse (image filtrée)



➔ Filtres de Gabor

Les paramètres du filtre sont μ_θ, σ_θ, μ_ω, σ_ω

Un filtre s'exprime sous la forme : $G(\theta,\omega) = e^{\frac{(\theta-\mu_\theta)^2}{2.\sigma_\theta^2}} . e^{\frac{(\omega-\mu_\omega)^2}{2.\sigma_\omega^2}}$

Exemple de valeurs possibles pour les paramètres des filtres :

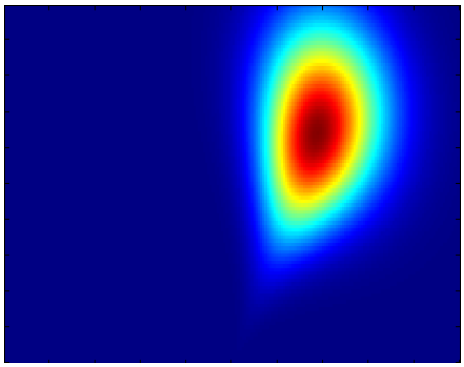
pour i allant de 0 à 4 (= 5 « grains » possibles)

pour j allant de 0 à 5 (= 6 orientations possibles)

μ_ω(i) = 3/4*(max(F) - min(F)).2^{-i}

μ_θ(j) = j*π/6

σ_ω(i) = (max(F) - min(F)).2-(i+1)/sqrt(8.ln(2))



$\sigma_\theta(j) = (\pi/12)/\text{sqrt}(2.\ln 2)$

b) *Transformée en Ondelette – Filtres de Haar*

➔ Transformée en ondelettes continues

$$W(a,b) =< f, \psi_{a,b} >= \int_{t=-\infty}^{+\infty} f(t).\overline{\psi}_{a,b}(t)\partial t \text{ avec } \psi_{a,b}(t) = \frac{1}{\sqrt{|a|}}\psi\left(\frac{t-b}{a}\right).$$

Où $\psi(t)$ est une **ondelette mère** à définir. Le paramètre « a » exprime la « dilatation » (ou l'échelle) de l'ondelette mère. Plus l'échelle (temporelle) est petite » plus la fréquence analysée est élevée : on utilise une petite onde qui oscille donc plus. A l'inverse, une grande échelle permettra d'analyser des basses fréquences (l'onde oscille plus lentement).

« b » exprime sa translation, c'est-à-dire la position sur le signal où on calcule la décomposition en ondelettes.

Pour qu'une fonction puisse jouer le rôle d'ondelette mère, il faut qu'elle soit continue, à valeur complexe (ou réelle), et qu'elle vérifie les conditions suivantes :

- $\int_{t=-\infty}^{+\infty} \psi(t)\partial t = 0$ (ie, dans le domaine discret, si elle est réelle : $\sum \text{coeff} > 0 = -\sum \text{coeff} < 0$)
- $\int_{t=-\infty}^{+\infty} |\psi(t)|^2 \partial t < \infty$

La transformation inverse est obtenue avec :

$$f(t) = \frac{1}{c_\psi^2}\int_a\int_b W(a,b)\frac{1}{a^2}\psi\left(\frac{t-b}{a}\right)\partial b.\partial a \quad \text{avec} \quad c_\psi = \sqrt{2\pi\int_{\omega=-\infty}^{+\infty}\frac{|\hat{\psi}(\omega)|^2}{|\omega|}\partial\omega}$$

Cette reconstruction n'est possible que si $c_\psi < +\infty$. Cette condition est appelée la **condition d'admissibilité**.

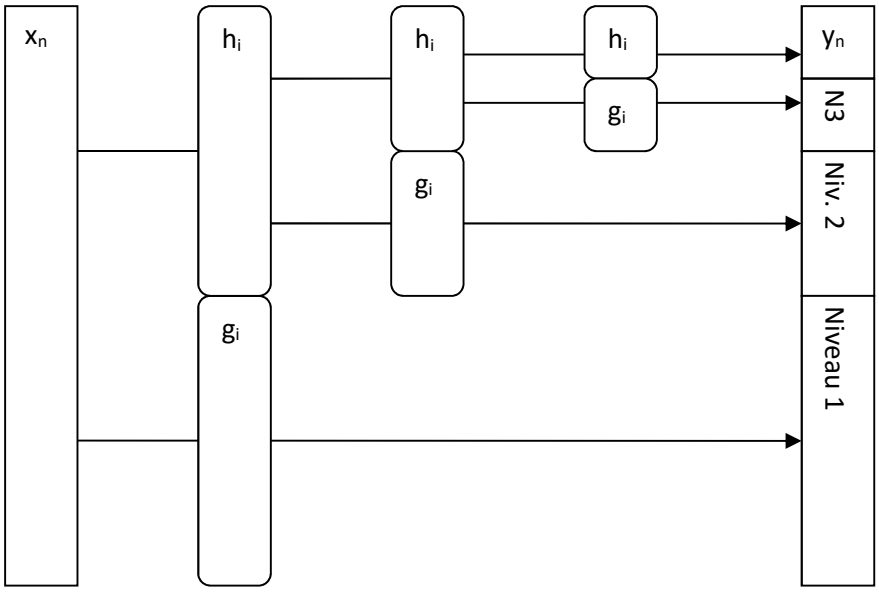
➔ Transformée en Ondelette discrète - Algorithme de Mallat

Le premier problème posé consiste à discrétiser l'espace des paramètres a et b, tout en conservant une expression continue du signal. Pour garantir la reconstruction du signal par transformée inverse, on utilise une « discrétisation logarithmique » du facteur d'échelle - la base du logarithme étant laissée au choix de l'utilisateur. Généralement, on prend un logarithme binaire, ce qui conduit à prendre pour a les valeurs 1, 2, 4, 8, 16, ... On parle alors de **décomposition dyadique** du signal.

On pose $a = a_0^j$ où a_0 dépend de la base logarithmique choisie (ie. 2) et où j exprime le niveau de la décomposition. On pose $b = k.a_0^j.b_0$ où k exprime le multiple du décalage de la fenêtre, et où b0 est relatif au pas de progression de cette fenêtre. En général, on choisit $b_0 = 1$.

On peut exprimer alors la **fonction d'échelle** qui représente l'ondelette à différentes valeurs d'échelle et pour différentes valeurs de translation : $\psi_{j,k}(t) = a_0^{\frac{-j}{2}}.\psi\left(a_0^{-j}t - kb_0\right)$.

La Transformée en Ondelettes Discrète s'exprime par un algorithme décomposant le signal en différentes sous-bandes fréquentielles. Ces bandes sont obtenues par la convolution de ce signal avec un filtre linéaire passe-bas et un filtre linéaire passe haut (complémentaire) à différentes échelles. Ces filtres font office d'ondelettes dans cette décomposition.



Les réponses impulsionnelles de h et g sont dépendantes l'une de l'autre (le filtrage doit être complémentaire). La relation qui relie h et g peut être :

$$g_{L-1-n} = (-1)^n.h_n.$$

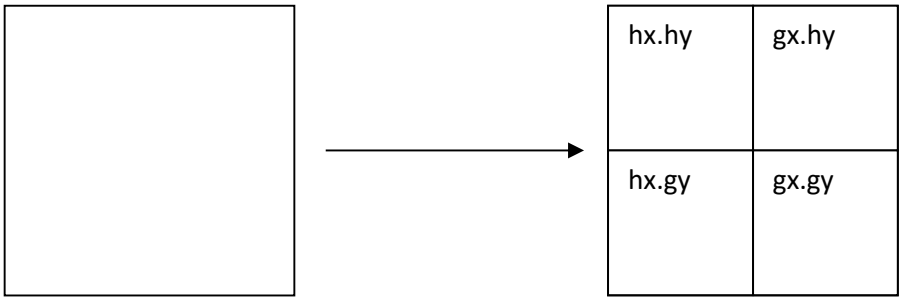
Les filtres qui vérifient cette relation sont appelés des **filtres miroirs en quadrature**.

Le nombre de niveaux (ie. le nombre d'itération du processus sur les basses-fréquences – ou encore l' « ordre ») peut être fixé a priori, ou dépendre d'un seuil sur le nombre de valeurs produites après sous-échantillonnage.

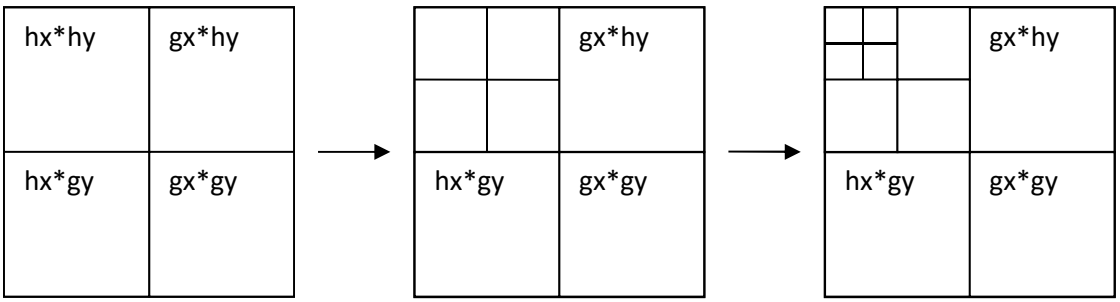
A chaque niveau de la décomposition correspond une résolution du signal (basse fréquence) qui est sous-échantillonné. On dit qu'on effectue une **analyse multirésolution**.

➔ Transformée en ondelettes 2D

On applique à l'image une convolution ligne par ligne avec les filtres gx et hx, puis sur les résultats de cette première convolution, une seconde convolution colonne par colonne avec les filtres g_y et h_y.. Les résultats sont rangés dans une matrice ayant les mêmes dimensions que l'image de la manière suivante :



Le processus est répété sur l'image des basses fréquences obtenue dans la partie (hx.hy)



Exercice 1 : On applique le principe de la transformée en ondelette de Haar sur un signal numérique composé de 8 échantillons avec h=[0.5 0.5] et g=[0.5 -0.5]. On obtient les valeurs 1 1 1 1 1 1 1 1. Quel était le signal de départ ?

Exercice 2

Soit X un signal numérique monodimensionnel. On définit les vecteurs H et G tels que leurs coefficients sont calculés par :

$H_n = X_{n+2} – (G_n + G_{n+2})/4$ (1)
$G_n = X_{n+1} – (X_n + X_{n+2})/2$ (2)

Question 1 : Donner les valeurs des vecteurs H et G obtenues à l'aide de ce procédé pour le signal suivant :

X = [20 16 4 12 36 40 28]

(On ne produira les résultats que pour les valeurs définies de G et de H).

Question 2 : On considère que les définitions de H et G expriment la convolution par un filtre linéaire passe-haut (1) et un filtre linéaire passe-bas (2). Déterminer à partir des formules (1) et (2) la largeur des deux filtres. Quelle technique permet de trouver les coefficients qui définissent les deux filtres ? Calculer ces coefficients.

Question 3 : Donner l'expression de H_n uniquement en fonction des coefficients de X (et non plus de G). Transformer cette expression pour obtenir celle de la convolution de X par le filtre miroir en quadrature de H.

Exercice 3 : On considère une image en niveau de gris de 4x4. Les intensités sont exprimées entre 0 et 255.

Question 1 : Le résultat R de la décomposition dyadique de l'image lorsque le filtre passe-bas (respectivement le filtre passe-haut) vaut h=[0.5 0.5] (resp. g=[-0.5 0.5]) est donné ci-dessous.

4	2	1	1
2	2	1	1
1	1	1	1
1	1	1	1

Le report des convolutions s'effectue selon le procédé suivant :

$* h_x * g_x$	$* h_x * g_y$
$* g_x * h_y$	$* g_x * g_y$

où l'indice x désigne une convolution dans la direction horizontale et y dans la direction verticale.

Quelle est la matrice d'origine ?

Question 2 : On utilise la matrice de quantification suivante :

1	a	b	b
a	a	b	b
b	b	b	b
b	b	b	b

Après quantification de la transformée en ondelettes R, on arrondit les coefficients à la partie entière. Exprimer l'erreur quadratique moyenne entre l'image de départ et l'image reconstruite après quantification lorsque a=2 et b=3.

Question 3 : Discutez de la valeur de l'erreur quadratique de l'image reconstruite à partir de R en distinguant les différents cas possibles en fonction des valeurs de a et de b entiers strictement positifs.

Exercice 4 :

Question 1 : Soit un filtre de convolution $g = \left[\frac{-1}{\sqrt{2}} \quad \frac{1}{\sqrt{2}} \right]$. Donner le filtre h miroir en quadrature de g.

Question 2 : Donner la décomposition dyadique du signal

s= [$16\sqrt{2}$ $4\sqrt{2}$ $16\sqrt{2}$ $12\sqrt{2}$ $12\sqrt{2}$ $8\sqrt{2}$ $8\sqrt{2}$ $4\sqrt{2}$]

produite à l'aide de h et g jusqu'au niveau le plus élevé possible.

Question 3 : Déterminer g1=[-a a], g2=[-b -b b b] et g3=[-c -c -c -c c c c c] où a, b et c sont des réels tels que

[$16\sqrt{2}$ $4\sqrt{2}$] $\otimes$ g1 = 12,

[$16\sqrt{2}$ $4\sqrt{2}$ $16\sqrt{2}$ $12\sqrt{2}$] $\otimes$ g2 = -4$\sqrt{2}$

[$16\sqrt{2}$ $4\sqrt{2}$ $16\sqrt{2}$ $12\sqrt{2}$ $12\sqrt{2}$ $8\sqrt{2}$ $8\sqrt{2}$ $4\sqrt{2}$] $\otimes$ g3 = 8

Le symbole $\otimes$ représente la convolution discrète.

Question 4 : Représenter sur un même graphe les fonctions discrètes g1, g2 et g3. Etablir une rapprochement entre g1, g2, g3 et la base d'ondelettes obtenue pour a_0=2, b_0=0 et j variant de 0 à 2. Il pourra être nécessaire de donner l'expression de l'ondelette mère sous une forme continue morceaux sur un domaine de définition à préciser.

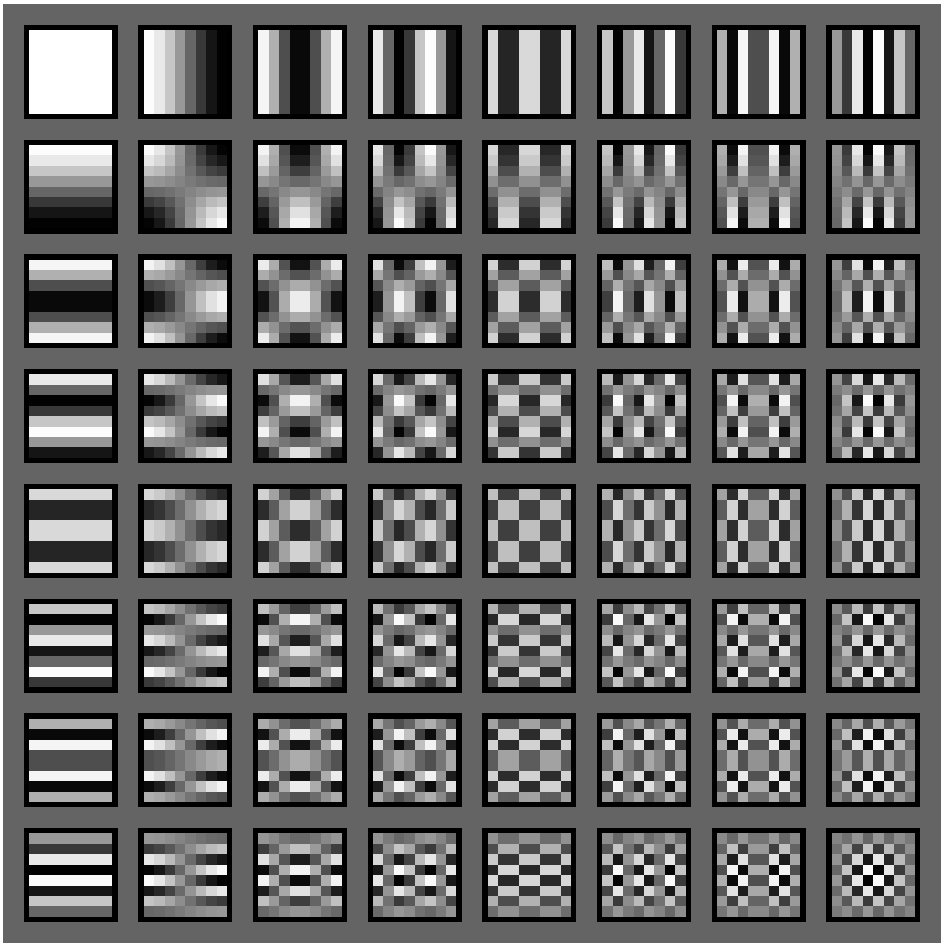
c) Transformée en Cosinus Discrète

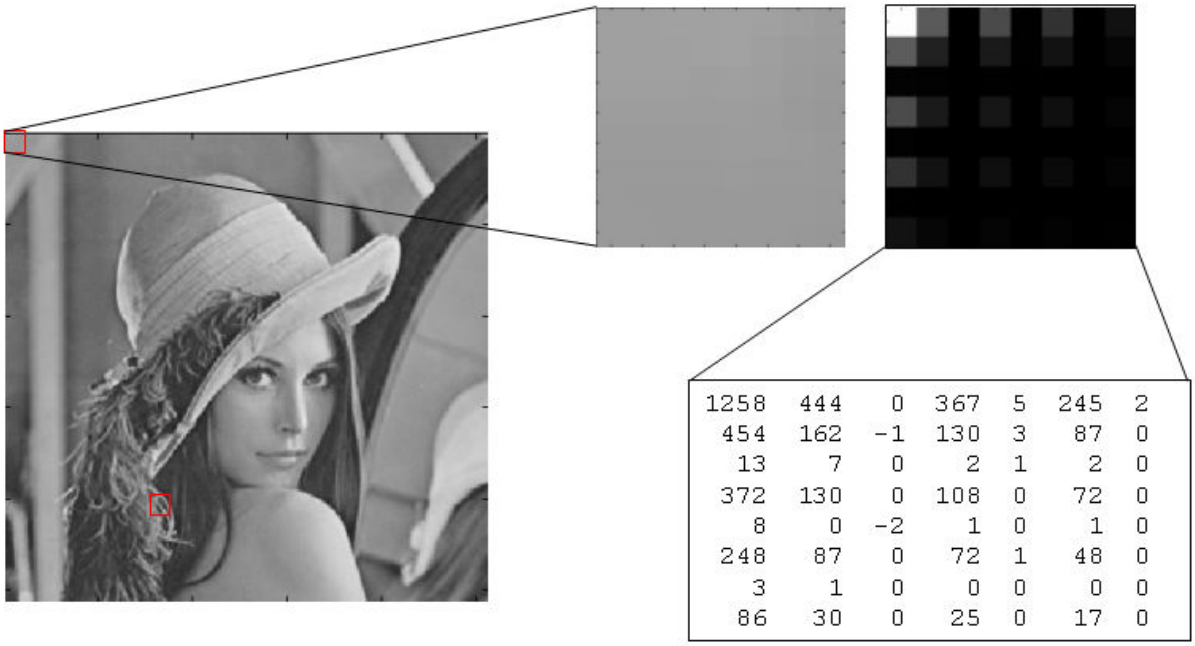
$$DCT(u,v) = \frac{2}{N} c(u).c(v) \sum_{y=0}^{N-1} \sum_{x=0}^{N-1} \cos\left(\frac{\pi}{N} u\left(x+\frac{1}{2}\right)\right) \cos\left(\frac{\pi}{N} v\left(y+\frac{1}{2}\right)\right) M(x,y)$$

avec $c(a) = \begin{cases} \frac{1}{\sqrt{2}} \text{ si a} = 0 \\ 1 \text{ sinon} \end{cases}$

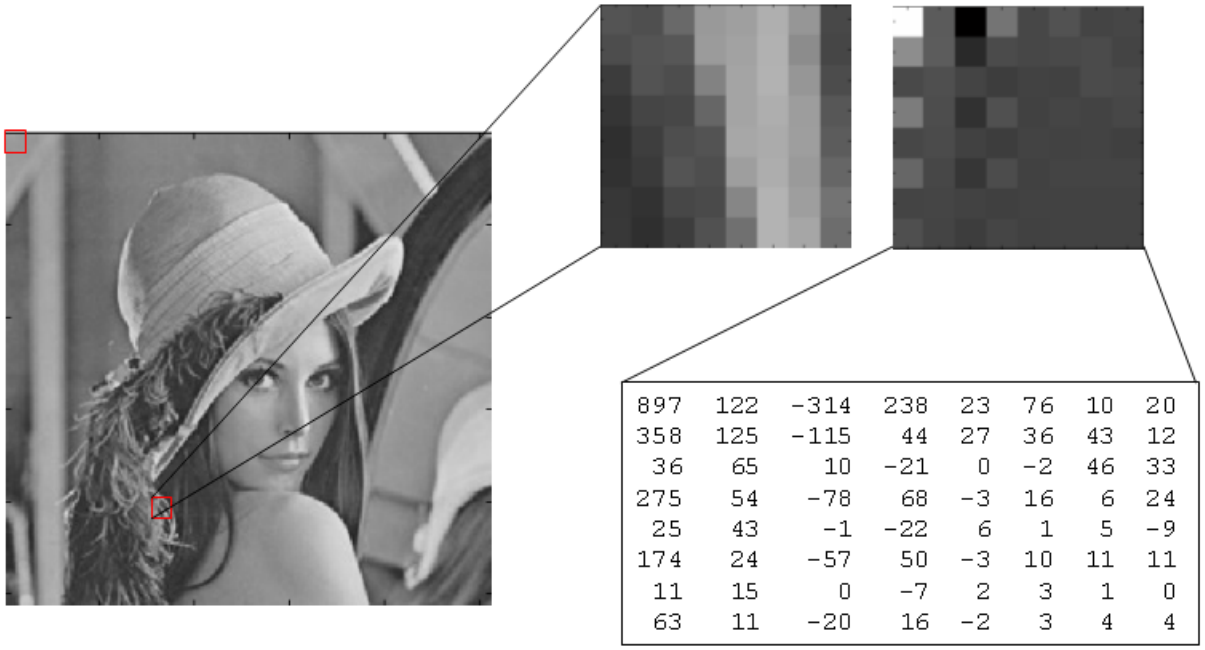
La transformée inverse est obtenue avec :

$$M(x,y) = \frac{2}{N} \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} c(u).c(v).DCT(u,v) \cos\left(\frac{\pi}{N} u\left(x+\frac{1}{2}\right)\right) \cos\left(\frac{\pi}{N} v\left(y+\frac{1}{2}\right)\right)$$





1258	444	0	367	5	245	2
454	162	-1	130	3	87	0
13	7	0	2	1	2	0
372	130	0	108	0	72	0
8	0	-2	1	0	1	0
248	87	0	72	1	48	0
3	1	0	0	0	0	0
86	30	0	25	0	17	0



897	122	-314	238	23	76	10	20
358	125	-115	44	27	36	43	12
36	65	10	-21	0	-2	46	33
275	54	-78	68	-3	16	6	24
25	43	-1	-22	6	1	5	-9
174	24	-57	50	-3	10	11	11
11	15	0	-7	2	3	1	0
63	11	-20	16	-2	3	4	4

d) Matrices de cooccurrence

La matrice de coocurrence NxN (N = nb de niveaux de gris) est définie par :

$$m_R(i,j) = \frac{\#\{(x,y)(x',y') \mid (x,y)R(x',y'), I(x,y)=i, I(x',y')=j\}}{\#\{(x,y)(x',y') \mid (x,y)R(x',y')\}}$$

R : relation spatiale entre 2 pixels (distance et orientation)

I(x,y) : niveau de gris à la position (x,y)

: nombre d'éléments

Les paramètres d'Haralick sont souvent utilisés pour caractériser la distribution des coefficients dans la matrice. Ils sont calculés à l'aide des 14 fonctions suivantes :

notations

$m_x(i)$: somme de la ligne i $\quad m_{x+y}(k)$: somme de la k ième diagonale secondaire

$m_y(i)$: somme de la colonne i $\quad m_{x-y}(k)$: somme de la k ième diagonale principale

moment angulaire d'ordre 2 $\quad f_1 = \sum_i \sum_j m(i,j)^2$

contraste $\quad f_2 = \sum_{n=0}^{N} n^2 \cdot m_{x-y}(n)$

corrélation $\quad f_3 = \dfrac{\sum_i \sum_i \left(i \cdot j \cdot m(i,j) - \mu_x \mu_y\right)}{\sigma_x \sigma_y}$

variance $\quad f_4 = \sum_i \sum_i (i-j)^2 \cdot m(i,j)$

moment des différences inverses $\quad f_5 = \sum_i \sum_i \dfrac{1}{1+(i-j)^2} \cdot m(i,j)$

moyenne des sommes $\quad f_6 = \sum_{k=2}^{2N-1} k \cdot m_{x+y}(k)$

variance des sommes $\quad f_7 = \sum_{k=2}^{2N-1} (k - f_6)^2 \cdot m_{x+y}(k)$

entropie de la somme $\quad f_8 = -\sum_{k=2}^{2N-1} m_{x+y}(k) \cdot \log\left(m_{x+y}(k)\right)$

entropie $\quad f_9 = -\sum_i \sum_j m(i,j) \cdot \log\left(m(i,j)\right)$

variance des différences $\quad f_{10} = \sum_{k=0}^{N} (k - \mu_{x-y})^2 \cdot m_{x-y}(k)$

avec $\quad \mu_{x-y} = \dfrac{1}{N} \sum_{k=0}^{N} m_{x-y}(k)$

entropie des différences : $f_{11} = -\sum_{i=0}^{N} m_{x-y}(i)\log\left(m_{x-y}(i)\right)$

corrélation de l'information : $f_{12} = \dfrac{H_{XY} - H1_{XY}}{\max\left(H_X, H_Y\right)}$

$$f_{13} = \left[1 - \exp\left(-2.0\left(H2_{XY} - H_{XY}\right)\right)\right]^{\frac{1}{2}}$$

avec $H_{XY} = f_9$ $\quad H_X = -\sum_{k=0}^{N} m_x(k)\log\left(m_x(k)\right)$ $\quad H_Y = -\sum_{k=0}^{N} m_y(k)\log\left(m_y(k)\right)$

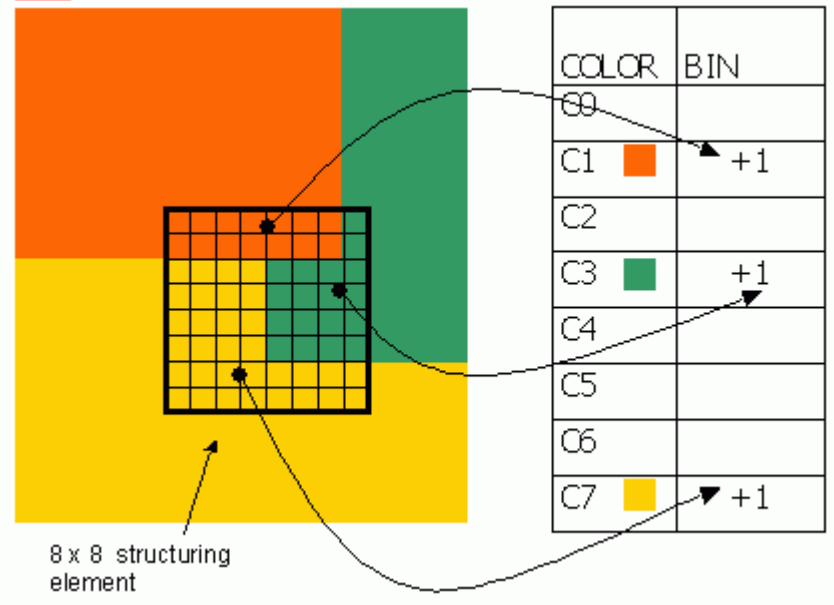
$$H1_{XY} = -\sum_i \sum_j m(i,j)\log\left(m_x(i)m_y(j)\right) \quad H2_{XY} = -\sum_i \sum_j m_x(i)m_y(j)\log\left(m_x(i)m_y(j)\right)$$

corrélation maximum : $f_{14} = \left(2ème \text{ des plus grandes valeurs propres de } Q\right)^{\frac{1}{2}}$

$$Q = \{Q(i,j)\}\begin{cases} i = 0\cdots N \\ j = 0\cdots N \end{cases} \qquad Q(i,j) = \sum_{k=0}^{N} \frac{m(i,k)m(k,j)}{m_x(i)m_y(j)}$$

e) Structure de couleur

On crée un tableau comprenant autant d'entrée qu'il y a de couleurs dans la LUT.
Un élément structurant (ici un carré 8x8) balaye la région.
Pour chaque couleur différente C présente dans la région, T[C] <- T[C] + 1

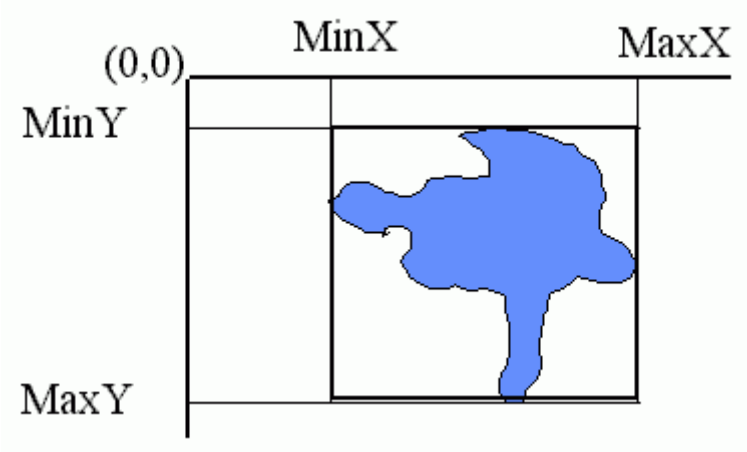


Lorsqu'une couleur est "compacte", elle est associée à une petite valeur dans le tableau par rapport aux nombre de pixels concernés. Lorsqu'une couleur est répartie, elle présente une forte valeur dans le tableau par rapport au nombre de pixels concernés.

3. Forme

a) Caractérisation de la région complète

➔ **les coordonnées de la boîte englobante**



➔ **Centre de gravité - centre géodésique**

Soit $d_R(a,b)$ la distance géodésique dans la région R, c'est à dire la plus petite distance à l'intérieure de X entre le pixel a et le pixel b. On peut définir cette distance comme étant le chemin comportant le moins de pixels en passant de pixel à pixel par un des 4 voisins tout en restant à l'intérieur de R.

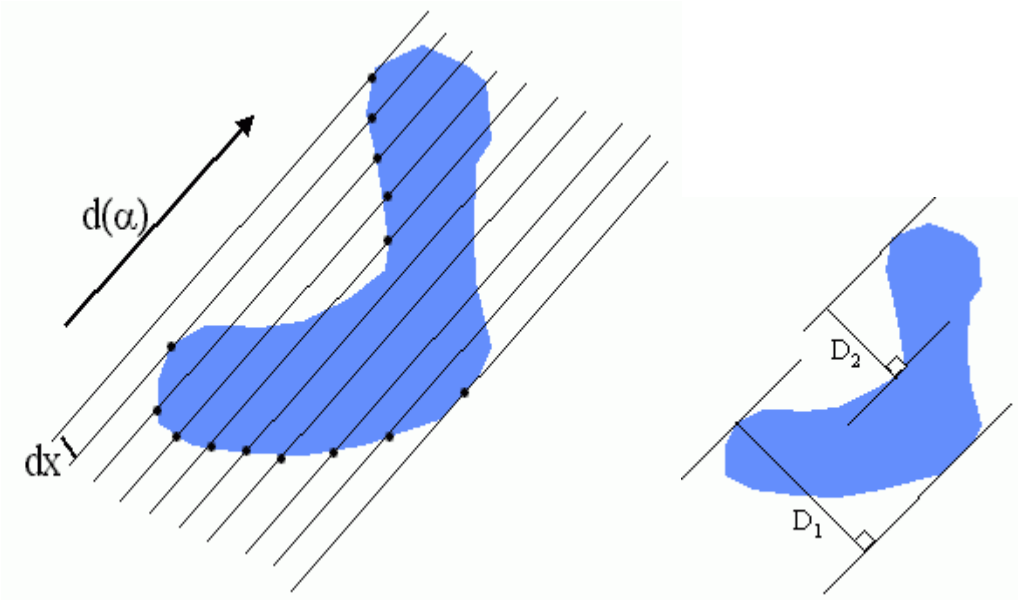
On définit la fonction de propagation P_R par : $P_R(a) = max\ (d(a,b))$, pour tout b appartenant à R.

Le centre géodésique est alors le point produisant la valeur minimale de P_R.

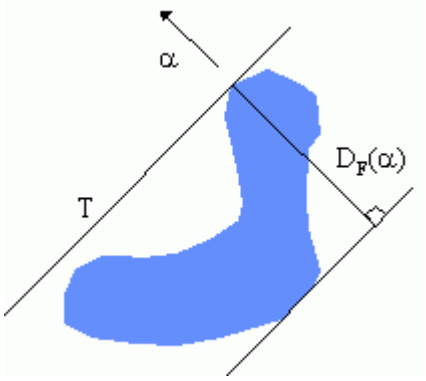
➔ **l'aire**
➔ **le périmètre**
➔ **diamètre - diamètre géodésique - rayon géodésique**

b) Caractérisation du contour

➔ **Le nombre d'intercepts** (ou calcul de la variation diamétrique)



➔ **Périmètre de Crofton**
➔ **Diamètre de Feret**



➔ **Facteur de compacité**
FC = Périmètre(R) /sqrt(4π.Aire(R))

c) VII.4 Moments

On définit le moment cartésien d'ordre p+q par :

$$m_{pq} = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} x^p y^q g(x,y)dxdy$$

$$m_{pq} = \sum_{y=0}^{NL-1}\sum_{x=0}^{NC-1} x^p y^q f(x,y)$$

g(x,y) et f(x,y) sont les fonctions images dans le cas continu et le cas discret. $x^p y^q$ est la "base". On définit l'ensemble complet des moments d'ordre n par : $\{m_{pq} / p+q<=n\}$.

On montre que $\{m_{pq}\}$ est défini de manière unique par f(x,y). L'utilisation des moments consiste donc en général à définir un sous-ensemble de $\{m_{pq}\}$ suffisant pour caractériser une région dans le cadre d'une application donnée.

Pour une image binaire, f(x,y) = 1 si le point appartient à la région et vaut 0 sinon.

On définit à partir des coordonnées des centres de gravité les moments centraux :

$$\mu_{pq} = \sum_{y=0}^{NL-1} \sum_{x=0}^{NC-1} (x-\bar{x})^p (y-\bar{y})^q \cdot f(x,y)$$

➔ Les moments invariants

Ces moments sont invariants en translation, rotation et homothétie (Hu).

$$M(1) = \mu_{20} + \mu_{02}$$

$$M(2) = (\mu_{20} - \mu_{02})^2 + 4\mu_{11}^2$$

$$M(3) = (\mu_{30} - 3\mu_{12})^2 + (3\mu_{21} - \mu_{03})^2$$

$$M(4) = (\mu_{30} + \mu_{12})^2 + (\mu_{21} + \mu_{03})^2$$

$$M(5) = (\mu_{30} - 3\mu_{12})(\mu_{30} + \mu_{12})\left[(\mu_{30} + \mu_{12})^2 - 3(\mu_{21} + \mu_{03})^2\right] + (3\mu_{21} - \mu_{03})(\mu_{21} + \mu_{03})\left[3(\mu_{30} + \mu_{12})^2 - (\mu_{21} + \mu_{03})^2\right]$$

$$M(6) = (\mu_{20} - \mu_{02})\left[(\mu_{30} + \mu_{12})^2 - (\mu_{21} + \mu_{03})^2\right] + 4\mu_{11}(\mu_{30} + \mu_{12})(\mu_{21} + \mu_{03})$$

$$M(7) = (3\mu_{21} - \mu_{03})(\mu_{30} + \mu_{12})\left[(\mu_{30} + \mu_{12})^2 - 3(\mu_{21} + \mu_{03})^2\right] + (\mu_{30} - 3\mu_{12})(\mu_{21} + \mu_{03})\left[3(\mu_{30} + \mu_{12})^2 - (\mu_{21} + \mu_{03})^2\right]$$

Moments invariants en échelle :

$$\eta_{ij} = \frac{\mu_{ij}}{\mu_{00}^{\left(1+\frac{i+j}{2}\right)}}$$

➔ Bitquads

Histogramme de motifs binaires 2x2 comportant au moins un ‘0’ et au moins un ‘1’.

➔ Descripteurs de Fourier

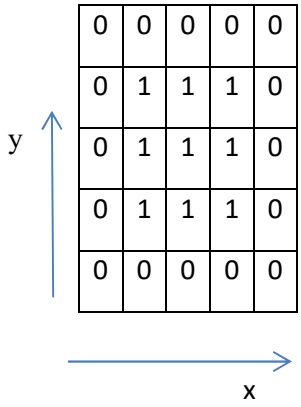
Coefficients fréquentiels issus de la transformée de Fourier discrète appliquée à la fonction de courbure k d’une forme. Cette fonction est définie comme étant la dérivée de la fonction angulaire qui donne en tout point selon une abscisse curviligne l’angle du contour de la forme avec l’axe des abscisses.

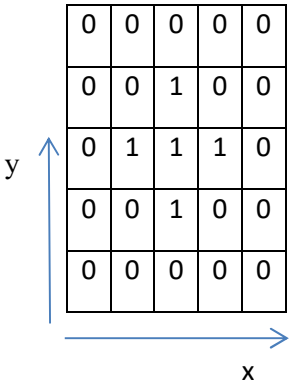
➔ Curvature scale space

Tableau décrivant le nombre d’opérations de lissage à effectuer sur le contour d’une forme pour faire disparaître toutes les concavités.

Exercice : On considère les régions correspondant aux '1' dans les images binaires ci-dessous :

Donnez pour ces deux régions :





1. Le périmètre
2. L'aire
3. Les coordonnées de leur centre de gravité
4. Les coordonnées de leur centre de gravité géodésique
5. Le nombre d'intercepts pour t={0°, 45°, 90°, 135 °}
6. Le diamètre de Feret pour t={0°, 45°, 90°, 135 °}
7. Le facteur de compacité
8. Les moments centraux μ10, μ11, μ01
9. Les bitquads
10. Le Curvature Scale Space

B. Points d'intérêt

1. Principe général

Objectif : mettre en correspondance 2 images ou caractériser un objet pour l'identifier dans toute image où il pourrait figurer.

Principe :

- Detection de zones de saillance dans les images : des points spécifiques qu'on peut retrouver meme si l'image est passablement modifiee
- Association d'un descripteur à chaque point
- Appariement des points entre 2 images en fonction de la position des points et/ou de leur description

2. Détection des points d'intérêt

a) Détecteur de Moravec (1980)

Les points de Moravec sont les maximums locaux de la fonction :

$$SM(x,y) = \min_{(a,b)} \sum_{u,v} (\omega(u,v).|I(x+u,y+v) - I(x+u+a,y+v+b|)$$

Exemple de voisinage exprimé par ω pour un 4-V

u\v	-1	0	1
-1	0	1	0
0	1	1	1
1	0	1	0

Exercice : Appliquez cette méthode sur la matrice ci-dessous pour $(a,b) \in [-1,1]^2$ et un voisinage 8-V.

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

b) Détecteur de Harris (1988)

L'opérateur de Harris vise à réduire quelques problèmes de l'opérateur de Moravec :

- Moravec réagit plutôt bien si les coins sont présents à l'extrémité de contours verticaux, horizontaux ou diagonaux, mais moins bien pour des contours d'orientation autre quelconque – on dit qu'il est anisotropique.
- Harris propose de définir le voisinage sous la forme d'un filtre gaussien 2D : on donne beaucoup de poids au centre du voisinage et l'influence de ce voisinage s'estompe au fur et à mesure qu'on s'éloigne du centre. Un voisinage rectangulaire avec une pondération binaire bruite la réponse de Moravec.
- Le filtre tient compte de toutes les valeurs obtenues pour toutes les translations et non uniquement de la valeur minimale. Ceci permet d'atténuer les réponses sur les contours.

La fonction calculée s'exprime alors par :

$$SH(x,y) = A.*B - C.^2 - k(A+B).^2$$

Avec :

- $A = m\left(\left(\frac{\delta I}{\delta x}\right)^2\right), B = m\left(\left(\frac{\delta I}{\delta y}\right)^2\right), C = m\left(\left(\frac{\delta I}{\delta x}\right)\left(\frac{\delta I}{\delta y}\right)\right),$
- k à fixer expérimentalement
- m(a) est la moyenne pondérée des valeurs dans le voisinage autour du point considéré

Exercice : Même question que précédemment sur la matrice ci-dessous – avec calcul de la moyenne non pondérée sur un support 4V. Déterminer les conditions sur k pour obtenir une « bon comportement de SH sur cet exemple.

0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	5	5	5	5	5
0	0	0	0	5	5	5	5	5
0	0	0	0	5	5	5	5	5
0	0	0	0	5	5	5	5	5
0	0	0	0	5	5	5	5	5

c) SUSAN (1997)

Le détecteur SUSAN considère que les coins sont des points dont les voisins ont peu de valeurs similaires. On définit une zone circulaire autour d'un point (le noyau), à l'aide d' un structurant e 37 pixels.

Puis on considère les pixels de cette zone qui ont un niveau de gris proche de ce noyau. Ces pixels forment une zone appelée USAN ("Univalue Segment Assimilating Nucleus"). En chaque point du disque, on calcule :

$$c((x_0,y_0),(x,y)) = e^{-\left(\frac{I(x_0,y_0)-I(x,y)}{t}\right)^6}$$

où (x_0,y_0) est le nucléus, (x,y) est le point testé et t est le seuil sous lequel on considère que deux pixels ont une couleur proche. Puis on calcule :

$$n(x_0, y_0) = \sum_{(x,y)} c((x_0, y_0), (x, y)).$$

$$R(x_0, y_0) = \begin{cases} g - n(x_0, y_0) & si \quad n(x_0, y_0) < g \\ 0 & sinon \end{cases}$$

Les maximums locaux de R sont des coins ou des contours.

Pour ne retenir que des coins, il faut appliquer 2 filtrages supplementaires. Pour chaque point on calcule le « centre de gravité des USANs ». On élimine alors :

- les points dont les centres de gravité sont trop proches
- les points dont le segment les reliants au centre de gravité et jusqu'au centre n'appartiennent pas à l'USAN.

Exercice :

Question 1 : Donner une valeur (approximée) de la fonction R dans la zone grisée des deux matrices ci-dessous :

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

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

Question 2 : Identifier le ou les points d'intérêt sur les résultats obtenus à la question 1 sur chaque matrice.

d) SIFT (2004)

(Scale Invariant Feature Transform)

Ces descripteurs sont produits en 2 phases :

- **l'extraction de points clés**

  Soit I1 une image. On calcule plusieurs versions de cette image I2, I3, etc, de plus en plus floues en appliquant un filtre passe-bas. Ce filtre passe-bas est un filtre gaussien d'écart-type σ (fixé empiriquement à 1,6)

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y)$$

avec

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

  Pour produire un effet de flou progressif, le paramètre σ_{t+1} permettant d'obtenir l'image I_{t+1} est calculé par $\sigma_{t+1} = k.\sigma$. Si on produit n images, on choisit $k= 2^{1/n}$ On calcule ensuite la différence entre les images successives ainsi obtenues

  On divise ensuite la taille de l'image I1 par 2 et on recommence.

  Chaque groupe d'image de différence d'une même résolution est appelé « octave ». Sur chaque octave, sur chaque image de différence obtenue pour le paramètre σ, on repère les minimums et les maximums locaux. Le voisinage considéré pour tester le caractère min ou max du point est :

  - les 8 voisins
  - les 9 voisins centrés autour du même point obtenus à la même octave pour σ_{t-1}
  - les 9 voisins centrés autour du même point obtenus à la même octave pour σ_{t+1}

Un minimum est un centre d'une région homogène / un maximum est un point saillant sur un contour.

- Le filtrage des points candidats
  Certains de ces points sont éliminés lorsqu'ils ne correspondent pas à un contraste local suffisant. (pour simplifer, lorsque la valeur absolue du point dans l'image de différence est inférieure à 0.03 – plus exactement, on prend une valeur estimée du minimum local à un niveau subpixelique à l'aide des dérivées premières et secondes des valeurs du voisinage pour le test).
  D'autres points sont éliminés lorsqu'ils sont détectés sur des contours (max locaux) qui ne sont pas des points saillants. Ils sont de ce fait peu discriminant pour la suite.
  Un point est retenu si $\frac{(Dxx+Dyy)^2}{Dxx.Dyy-(DxyDyx)} < \frac{(r+1)^2}{r}$ où :
  - Dxx et Dyy sont les dérivées seconde en x et en y ; Dxy et Dyx sont les dérivées en x puis en y et inversement.
  - r est le rapport entre la plus grande et la plus petite des 2 valeurs propres de la matrice hessienne $\begin{bmatrix} Dxx & Dxy \\ Dyx & Dyy \end{bmatrix}$. Si cette valeur est très grande : la premiere valeur propre suffit presque à exprimer les données => il y a une seule orientation dominante de contour => le point n'est pas intéressant. (r fixé expérimentalement à 10).

e) SURF

(Speeded Up Robust Feature)

Principe identique aux SIFT avec accélération des calculs à l'aide :

- Des ondelettes de Haar et du calcul des images intégrales
- Calcul entier du déterminant Hessien (# Harris)
- Réduction de la taille du vecteur de description
  64 valeurs = somme des gradients en x, en y et idem en valeur absolu dans 4x4 zones centrees sur le point d'interet a l'interieur d'une fenetre de taille et d'orientation adaptees a son contexte.

f) Squelette

3. Descripteurs locaux

a) IV.1 SIFT

Chaque niveau produit un certain nombre de points clés. Pour chaque point clé, on produit un descripteur sous la forme de 16 histogrammes de 8 bins de la manière suivante :

- on prend une fenêtre de 16 par 16 pixels, centrée sur le point d'intérêt
- on coupe cette fenêtre en 16 sous fenêtres disjointes de 4 sur 4 pixels.
- Dans chaque sous-fenêtre, en chaque point on calcule l'amplitude et la direction du gradient.
- Pour chaque point d'intérêt, on produit un descripteur de 4x4x8=128 valeurs. Ce vecteur est normalisé par sa norme maximale théorique.

Pour la comparaison des SIFT, on utilise 2 métriques : distance en cosinus et la distance angulaire :

- Cos(Vo,Vr)=Vo.Vr/(|Vo|.|Vr|)
- Dangle=cos-1(Vo.Vr/(|Vo|.|Vr|))

b) SURF

A peu près identique aux SIFT avec :

- La recherche d'une orientation « invariante » à la rotation (estimation de la direction du plus fort gradient à l'échelle retenue)
- La production d'un descripteur de 4 valeurs pour chaque 4x4 sous-regions d'une fenetre centree sur le point dont la taille depend de l'echelle soit 64 valeurs. Les 4 valeurs sont : la somme des derivees locales en x et en y ; la somme des valeurs absolues des dérivées locales en x et en y.

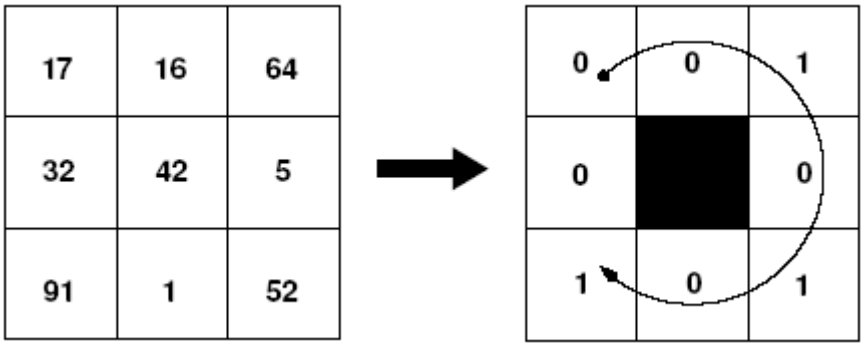
Pour identifier si une image I correspond à une image M indexée, on « apparie » chaque descripteur dI de la première à un descripteur de la seconde. On cherche le descripteur dM le plus similaire. Si la distance euclidienne est inférieure à un seuil, on établit la correspondance.

c) LBP

Le descripteur LBP (Local Binary Pattern) a été introduit dans [Ojala et al., 1996] pour la classification de textures. Il est calculé sur une région de 3x3 pixels.

$$\mathrm{LBP}(x_c, y_c) = \sum_{n=0}^{7} 2^n s(i_n - i_c)$$

avec s(u) = 1 si u > 0 et 0 sinon, (xc; yc) les coordonnées du point où on calcul le descripteur, i_c la valeur de ce point et les i_n parcourant le voisinage comme indiqué sur la figure suivante.



00101010

Un descripteur LBP est invariant par une variation monotone de la valeur des pixels, ce qui est intéressant pour résister aux variations d'illumination.

d) Descripteurs locaux issus de squelettes

1. On assimile le squelette d'une forme à un graphe. Les nœuds et les feuilles sont pris comme étant des points d'intérêt.
2. Les descripteurs couramment associés à ces points sont :
    a. Le nombre d'érosions nécessaires pour obtenir le nœud (ie épaisseur locale de la forme ou encore rayon de la boule maximale centrée sur le nœud)
    b. Degré du nœud
    c. Coordonnées polaires des points dans un repère centré sur la forme dont l'axe des abscisses est l'orientation principale de la forme (invariance à l'orientation). La coordonnée radiale peut être normalisée par rapport à la longueur de l'objet projeté sur cet axe principal. (invariance à l'échelle)

II. Transformées mathématiques

A. Transformée de Hough

1. Objectifs et principes

On dispose d'un modèle paramétrique supposé représenter la manière selon laquelle se distribue des observations. Le but est de déterminer la valeur des paramètres qui permet d'ajuster le modèle aux observations réelles – il s'agit d'un problème de régression. La transformée de Hough est un « algorithme glouton » qui consiste à estimer ces paramètres par une méthode de vote.

A l'origine, cette méthode a été développée pour « vectoriser » les contours dans une image binaire (après seuillage des gradients par exemple). Les contours des objets sont représentés par des courbes décrites à l'aide de fonctions simples (droites, arcs de cercles, …) modélisées par leurs paramètres.

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

Par exemple, les droites passant par un point (x,y) ont toutes une équation de la forme y=m.x+c

Algorithme

Etape 1. Modélisation

- une droite est représentée par une équation paramétrique
- on discrétise l'espace des paramètres (appelé espace de Hough) (on établit un pas pour chacune des 2 dimensions).
- Chacun des points (appelé « accumulateur ») de l'espace discrétisé de Hough se voit attribué le score de 0.

Etape 2. Reconnaissance

- Pour chacun des points de l'image des contours :
  - On établit l'équation paramétrique correspondante dans l'espace de Hough
  - On ajoute 1 au score des points de l'espace de Hough dont les coordonnées vérifient l'équation
- Les droites correspondant à des contours sont données par les points de l'espace de Hough ayant le plus grand coefficient.

Il existe une variante possible pour l'étape 2 :

- Pour chacun des accumulateurs de l'espace de Hough, on détermine l'équation paramétrique correspondante.
- On incrémente l'accumulateur pour chaque pixel à 1 de l'image dont les coordonnées vérifient l'équation.

Formellement, la Transformée de Hough conduit à construire une représentation H(a,b) définie par

$H(a,b) = \sum_x \sum_y I(x,y)\delta(ax+b-y)$ où $\delta(ax+b-y)$ est le symbole de Kronecker (ie.

$$\delta(x) = \begin{cases} 1 \text{ si } x = 0 \\ 0 \text{ sinon} \end{cases}$$

Exercice : Principes de base de la transformée de Hough

Question 1 : On considère une image de 5 x 5 pixels repérés par leurs coordonnées comprises entre 0 et 4. Soit P1 le pixel de coordonnées (2,3). On utilise les paramètres a et b pour représenter l'équation d'une droite sous la forme y = a.x + b. Donner une équation de la droite représentant le point P1 (i.e. l'ensemble des droites passant par ce point) dans l'espace paramétrique associé.

Question 2 :

Soi P2, le point de coordonnées (1,1). Déterminer le point d'intersection de la droite définie à la question 1 avec celle représentant le point P2 dans l'espace des paramètres. En déduire l'équation de la droite passant par P1 et P2. Vérifier le résultat.

Exercice 2 : Mise en oeuvre

L'espace de Hough est représenté par une matrice 11x11 où le coefficient de pente a et le coefficient de translation b peuvent prendre des valeurs entières comprises entre –5 et 5. Donner la configuration de l'espace de Hough lorsque les points de contours de l'image sont les suivants :

1) P1(1,0), P2(2,2), et P3(3,4)

5											
4											
3											
2											
1											
0											
-1											
-2											
-3											
-4											
-5											
b/a	-5	-4	-3	-2	-1	0	1	2	3	4	5

2) P1(1,0), P2(2,2), et P3(0,0)

5										
4										
3										
2										
1										
0										
-1										
-2										
-3										
-4										
-5										

b/a	-5	-4	-3	-2	-1	0	1	2	3	4	5

3) P1(1,0), P2(1,2), et P3(1,4)

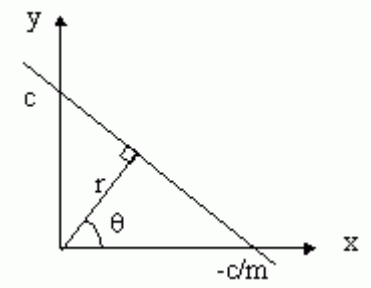
5											
4											
3											
2											
1											
0											
-1											
-2											
-3											
-4											
-5											
b/a	-5	-4	-3	-2	-1	0	1	2	3	4	5

2. Transformée de Radon

Une modélisation des droites sous la forme y=ax+b pose le problème de la discrétisation des paramètres a et b (valeur potentiellement infinie, progression non-linéaire). Pour palier ce problème, on utilise très fréquemment une représentation en coordonnées polaires : $x \cos\theta + y \sin\theta = \rho$

$0 < \rho <$ diagonale de l'image

$0 < \theta < \pi$



On exprime ainsi une extension de la transformée de Hough appelée Transformée de Radon sous la forme :

$$R(\rho,\theta) = \sum_x \sum_y I(x,y)\delta(x\cos\theta + y\sin\theta - \rho)$$

La Transformée de Radon a été développée initialement pour modéliser les informations disponibles dans les images de tomographie (rayons X). Selon le type d'étude, I(x,y) peut être

une image en niveaux de gris. A partir des informations disponibles dans R, il est possible de reconstruire l'image de départ par la « projection » inverse :

$$I(x,y) = \sum_{\theta} R(x\cos\theta + y\sin\theta, \theta)$$

3. Extension aux formes circulaires

$(x - a)^2 + (y - b)^2 = r^2$ => 3 paramètres : a, b, r

Exercice : On s'intéresse à des contours de forme circulaire de rayon 2 que l'on décrit à l'aide de l'équation $(x-c)^2 + (y-d)^2 = 2^2$ où c et d définissent l'espace des paramètres. Donner pour les trois points suivants, la configuration de l'espace de Hough associé :

P1 (0,2), P2(2,0), P3(4,2)

5											
4											
3											
2											
1											
0											
-1											
-2											
-3											
-4											
-5											
c/d	-5	-4	-3	-2	-1	0	1	2	3	4	5

4. Transformée de Hough généralisée

Modélisation

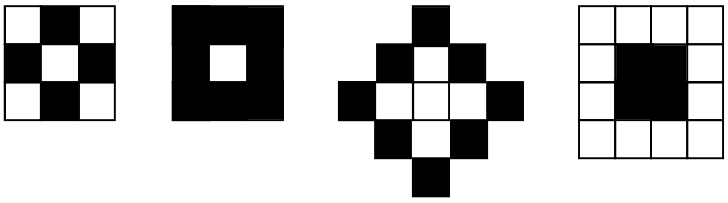
Reconnaissance

Algorithme

Exercice : Le tableau suivant ne contient qu'une partie seulement des données permettant de modéliser une forme (les angles sont exprimés en degré) :

β	**liste des couples (α,d)**
0	(0,1)
90	(90,1)
180	(180,1)
270	(270,1)

Compte-tenu de ces informations, quelle est (quelles sont) parmi les formes suivantes celle(s) qui correspond(ent) au modèle complet ? On considère que le contour de la forme est représenté par les pixels en noir.



Exercice : Une image représentant un circuit imprimé montre en particulier des composants carrés. On suppose qu'idéalement ces composants apparaissent sous la forme de carrés de 3x3 pixels dont les côtés sont alignés selon les axes verticaux et horizontaux (cf. matrice ci-dessous). Afin de détecter leur contour et leur position, on choisit d'utiliser la transformée de Hough. Donner une expression du modèle. Quelles sont les valeurs dans l'espace de Hough lorsque l'image des contours est celle correspondant à la matrice ci-dessous (précisément un carré de 3x3) ?

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

0	0	0	0	0

Vous indiquerez la nature des paramètres utilisés et les raisons qui motivent les différents choix que l'on peut être amené à faire pour implémenter un algorithme de reconnaissance de ces contours carrés, basé sur la transformée de Hough.

B. Transformée en distance

1. Distance, voisinage et connexité

a) Les k-distances et les k-chemins

Une k-distance vérifie les axiomes de définition des distances :

- $d_k(p,p) = 0$;
- $d_k(p,q) > 0$ si p différent de q ;
- $d_k(p,q) = d_k(q,p)$;
- $d_k(p,r) <= d_k(p,q) + d_k(q,r)$.

Exercice : Etant donné deux pixels p(i,j) et q(i',j'). Indiquer laquelle des trois distances d_4, d_8 et d_e renvoie la valeur la plus grande et laquelle renvoie la valeur la plus petite. Justifier en identifiant l'ensemble des pixels équidistants d'un pixel p central au sens des 3 distances.

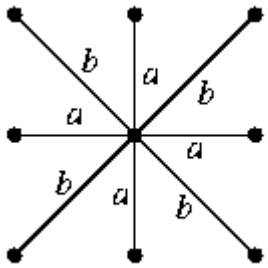
b) Les k-connexité

Une région X est dite k-connexe s'il existe un k-chemin inclu dans X reliant tout couple de points (p,q) de X.

2. Chanfrein 3x3

a) Définition

Pour réduire l'écart entre la distance euclidienne et les k-distances, on propose de pondérer l'"adjacence" entre un pixel et ses k-voisins à l'aide de deux poids différents :



b	*a*	*b*
a	0	*a*
b	*a*	*b*

Exercice : Proposer une valeur pour a et b de manière à retrouver la 4-distance et la 8-distance. Déterminer l'ensemble des points équidistants d'un pixel p par la distance de chanfrein.

Cette matrice de coefficients est appelée le "masque de chanfrein". Des contraintes sur les valeurs de a et b ont été définies (sous le nom de "Conditions de Montanari") pour garantir des propriétés minimales de la distance comme la convexité du cercle unité. Ces contraintes sont : a > 0 et a<= b <= 2a.

b) *Chanfrein de Borgefors*

Borgefors a ajouté à ces conditions deux critères :

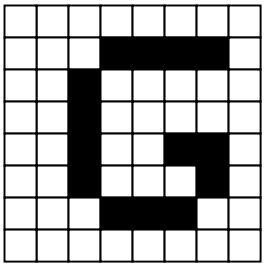
- la réduction de l'écart entre distance de chanfrein et et distance euclidienne.
- les coefficients du masque de chanfrein doivent être des nombres rationnels ayant de petits numérateurs et dénominateurs entiers (pour des raisons de vitesse de calcul).

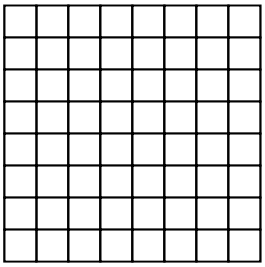
Le premier critère est vérifié de façon optimale en choisissant a = 0,95509 et b = 1,336930. L'approximation de ces valeurs par des rationnels "simples" peut être la suivante : a = 1 et b = 4/3.

c) *Mise en œuvre*

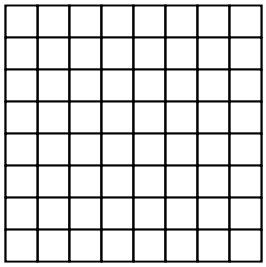
Pour calculer la TFD, on décompose le masque de chanfrein en deux sous masques : le masque postérieur M_+ et le masque antérieur M_- (associé au voisinage postérieur V_+ ou antérieur V_-)

Exercice : Calculer le résultat de l'application de la TFD lorsque le marqueur est la région en noir dans l'image suivante, lorsqu'on utilise la $d_{3,4}$ de Borgefors :

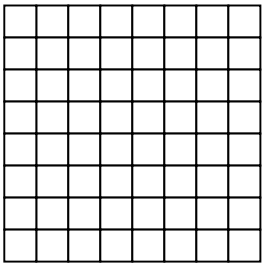




initialisation

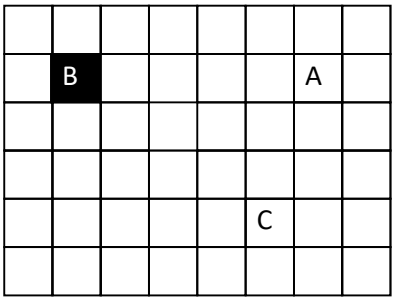


balayage descendant



balayage ascendant

Exercice : On souhaite analyser la précision de différentes transformées en distance. Pour cela, on considère



sur une image binaire trois points A, B, et C.

Seul le pixel correspondant au point B est à 1, tous les autres (y compris les pixels correspondant à A et C) sont à 0.

Question 1 : Montrer que AB = BC en distance euclidienne.

Question 2 : Calculer la transformée en distance de l'image en utilisant tour à tour :

- D1 : la 8-distance ($d_{1,1}$)
- D2 : la 4-distance ($d_{1,2}$)
- D3 : la $d_{3,4}$ de Borgefors

Question 3 : en fonction des résultats de la question 1, indiquer laquelle de ces trois distances vérifie "le mieux" l'égalité de la question 1. Quelle distance vérifie "le moins" cette égalité ?

Question 4 : Que se passe-t-il si on compare ces trois distances à l'aide de l'exemple suivant ?

Exercice :

Soit l'image binaire I :

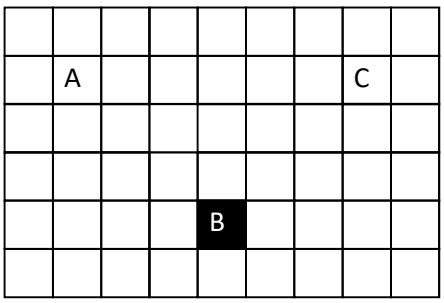
$$I = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

On définit un masque de chanfrein 3x3 par $\begin{bmatrix} b & a & b \\ a & 0 & a \\ b & a & b \end{bmatrix}$

Question 1 : donnez le résultat de la transformée en distance à l'aide de ce masque lorsque a=2 et b=4. Identifiez le cercle de rayon 6 sur le résultat.

Question 2 : même question avec a=4 et b=2.

Question 3 : Montrez que si b > 2a alors la solution d'une transformée en distance par masque de chanfrein est identique à celle obtenue lorsque b=2a.



Une fois la TFD appliquée, de nombreux traitement peuvent être appliqués. En particulier, il est possible de déterminer directement la distance à l'objet le plus proche à partir d'un pixel quelconque de l'image. On peut ainsi construire des relations de dépendance ou d'influence entre les objets.

Exercice 1 : Transformée en distance

Question 1 : Donnez la transformée en distance de l'image binaire ci-dessous en utilisant le chanfrein de Borgefors.

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

Question 2 : Soit une matrice de chanfrein 3x3 construite à l'aide de 2 coefficients a et b vérifiant les conditions de Montanari :

b	a	b
a	0	a
b	a	b

Soit une image binaire de LxH pixels comportant au moins un pixel à 1. Donnez l'expression formelle de la plus grande valeur possible figurant dans le résultat de la transformée en distance calculée avec ce chanfrein. Cette expression formelle ne pourra comporter que les paramètres a et b, les opérateurs arithmétiques de base (+, -, x, /) et les fonctions min et max.

Question 3 : en déduire si il est possible d'obtenir la valeur 567 dans la matrice de la transformée en distance calculée sur une image de 400 x 500 pixels avec le chanfrein suivant :

5	4	5
4	0	4

x 1/4

3. Application

a) Application à la reconnaissance d'objets

Exemple : OCR par appariement avec une TFD :

10	7	4	3	3	3	3	4	7	10
9	6	3	0	0	0	0	3	6	9
9	6	3	0	0	0	0	3	6	9
10	7	4	3	0	0	3	4	7	10
12	9	6	3	0	0	3	6	9	12
12	9	6	3	0	0	3	6	9	12
12	9	6	3	0	0	3	6	9	12
12	9	6	3	0	0	3	6	9	12
10	7	4	3	0	0	3	4	7	10
9	6	3	0	0	0	0	3	6	9
9	6	3	0	0	0	0	3	6	9
10	7	4	3	3	3	3	4	7	10

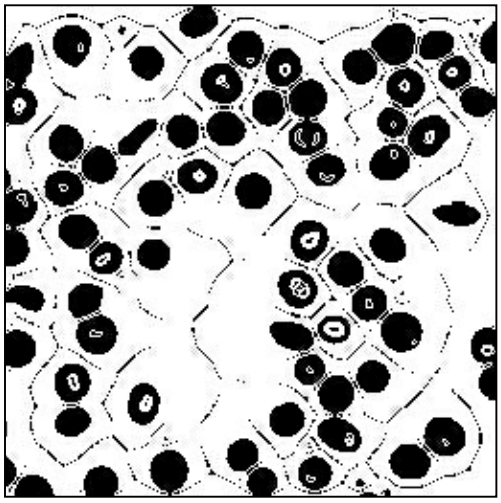
Score « I » \ I = 6 Taux = 26/28

4	3	3	3	4	4	3	3	3	4
3	0	0	0	3	3	0	0	0	3
3	0	0	0	3	3	0	0	0	3
4	3	0	0	3	3	0	0	3	4
6	3	0	0	3	3	0	0	3	6
6	3	0	0	0	0	0	0	3	6
6	3	0	0	0	0	0	0	3	6
6	3	0	0	3	3	0	0	3	6
4	3	0	0	3	3	0	0	3	4
3	0	0	0	3	3	0	0	0	3
3	0	0	0	3	3	0	0	0	3
4	3	3	3	4	4	3	3	3	

Score « I » \ H = 15 Taux = 23/52

b) *Régions d'influence*

Une région d'influence est la zone située autour d'un objet dans laquelle l'influence de l'objet est prépondérante *sur toute autre influence issue d'autres objets. Il existe différents mécanismes pour déterminer les régions* d'influence. Dont les **Diagrammes de Voronoï.**



Le diagramme de Voronoï d'un ensemble de sites (d'objets) se compose de :

- régions composées de l'ensemble des pixels de l'image plus proches d'un objet que de tous les autres (1 région par objets). Elles déterminent les régions d'influence.
- côtés composés de l'ensemble des pixels de l'image équidistants de 2 sites;
- sommets composés de l'ensemble des pixels de l'image plus proches d'au moins 3 sites.

L'algorithme que l'on peut appliquer pour retrouver le diagramme de Voronoï est le suivant :

```
1. Calculer la TFD sur une image
2. Appliquer la ligne de partage des eaux sur le résultat
```

III. Techniques d'analyse de contenus vidéo

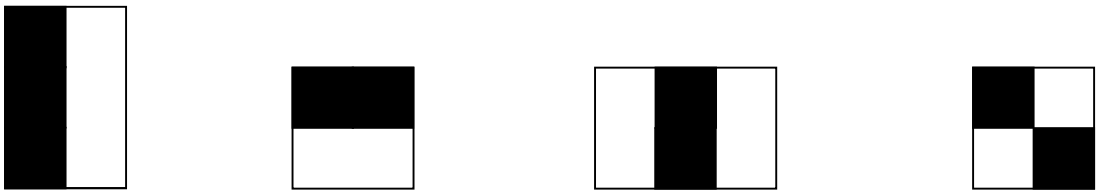
A. Analyse de la personne

1. Detection du visage

a) 1.1.1 Introduction

b) 1.1.2 Filtres de Haar

—Filtres inspirés des filtres de Haar
•4 motifs de base
–A, B, D : surface zone noire = surface zone blanche
–C : surface zone blanche = 2 x surface zone noire



A B C D

Exercice : calculer toutes les features sur le bloc 3x3 ci-dessous :

1	0	1
0	0	0
0	1	0

—Calcul de l' « image intégrale »

$$ii(x, y) = \sum_{x' \leq x, y' \leq y} i(x', y')$$

—Calcul de la somme des coef d'un bloc

c) 1.1.3 Adaboost

—Processus « gourmand » pour la pondération des features utiles pour la détection lors de l'apprentissage
—Construction du classifieur
—Repose sur la définition de « classifieurs faibles ». Dans le principe, chaque feature prise indépendamment, comparée à un seuil est un classifieur faible.

$$h(x, f, p, \theta) = \begin{cases} 1 & \text{if } pf(x) < p\theta \\ 0 & \text{otherwise} \end{cases}$$

Mise en œuvre de l'apprentissage

- Calculer les valeurs d'une feature donnee sur tous les exemples (tous les blocs 24x24 annotés)
- Ordonner les valeurs de la feature
- Definir le meilleur seuil theta pour separer au mieux les exemples positifs et les exemples négatifs.
- En deduire la valeur de p.

Le seuil est choisi comme minimisant le coût de l'erreur defini par :

$$e = \min\left(S^+ + (T^- - S^-), S^- + (T^+ - S^+)\right)$$

où T+ est la somme des poids des exemples positifs (1), où T- est la somme des poids des exemples négatifs (0), ou S+ est la somme des exemples positifs jusqu'à la position du seuil testé, et S- la somme des exemples négatifs jusqu'à la position du seul testée.

Problème : En prenant T=200, on obtient des résultats acceptables, mais des temps de calculs pour la classification trop important.

Exercice : Nous considérons 2 motifs de base pour calculer des filtres de Haar de la forme :

Motif A

Motif B

Nous traitons des images fortement pixélisés de taille 2x2 pixels.

Question 1 : Combien de caractéristiques peut-on extraire d'une image avec des filtres de Haar répondant à l'un ou à l'autre de ces motifs ? Donnez les valeurs de ces caractéristiques sur l'image suivante. Donnez également l'expression de l'image intégrale.

2	5
5	8

Question 2 : On dispose d'une base d'apprentissage composée de 4 images 2x2 et d'une vérité terrain. Elle a été créée pour distinguer les images ayant des pixels de même intensité sur la première diagonale de celles qui ont des pixels ayant la même intensité sur la seconde diagonale :

	2	5	8	6	3	6	4	8
	5	8	6	4	0	3	6	4
Vérité terrain	0		0		1		1	

On effectue un apprentissage par ADABOOST avec un seul classifieur faible. Les caractéristiques utilisées sont les mêmes que pour la question 1. Donnez l'expression du classifieur faible et du classifieur fort (identifiez les problèmes s'il y en a).

d) 1.1.4 Cascades de Haar

Algorithme :

F_0 = 1 % Taux de faux positifs

D_0 = 1 % Taux de bonnes détections

i = 0

Tant que $F_i > F_{cible}$

    i = i + 1

    n_i = 0

    $F_i = F_{i-1}$

    Tant que $F_i > f * F_{i-1}$

        $n_i = n_i + 1$

        Utiliser P et N pour entraîner un classifieur fort avec n_i descripteurs de Haar à l'aide de l'algorithme Adaboost

        Evaluer la cascade avec le classifieur fort ainsi produit pour déterminer D_i

        Diminuer le seuil du classifieur fort de manière à ce que $D_i >= d * D_i-1$

        Evaluer F_i avec le classifieur fort ainsi obtenu

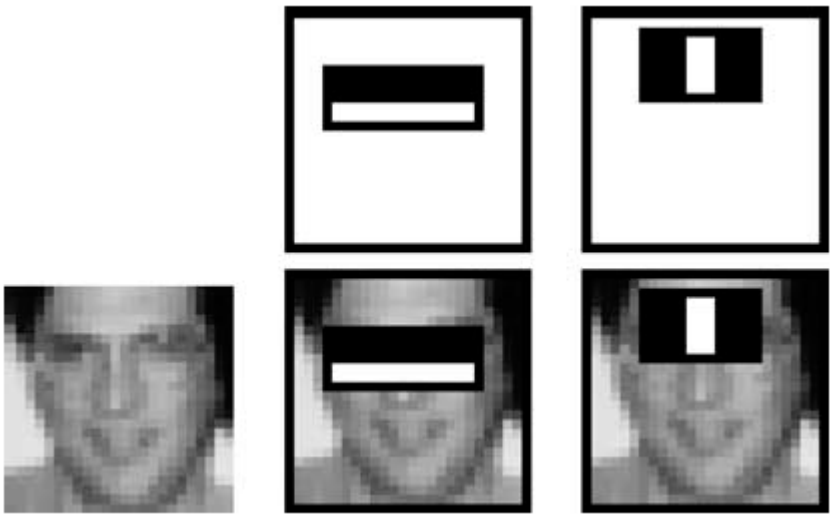
    N = {}

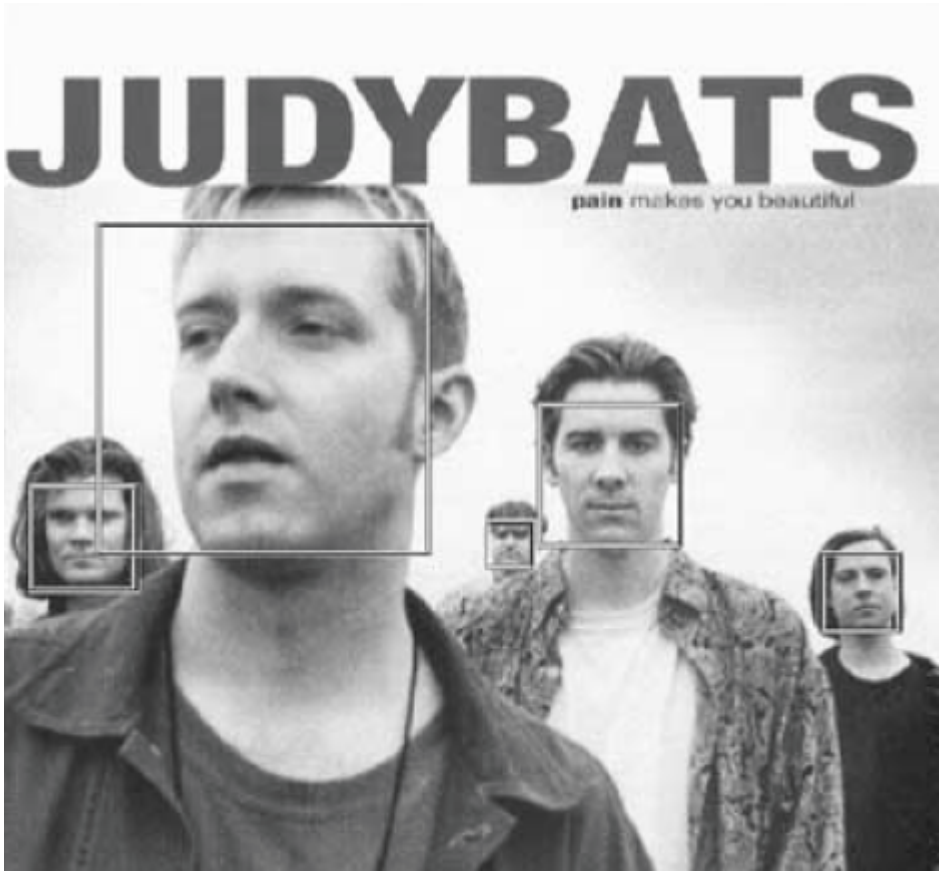
    Si $F_i > F_{cible}$ alors évaluer la cascade sur l'ensemble des exemples négatifs et ajouter les fausses détections dans N

Commentaires :

- le premier while permet d'estimer chaque classifieur de la cascade. Dans l'implantation OpenCV, le premier while a bouclé 38 fois.
- Le second while permet de définir 1 classifieur fort de la cascade. Dans l'implantation, le premier classifieur fort comporte 2 features

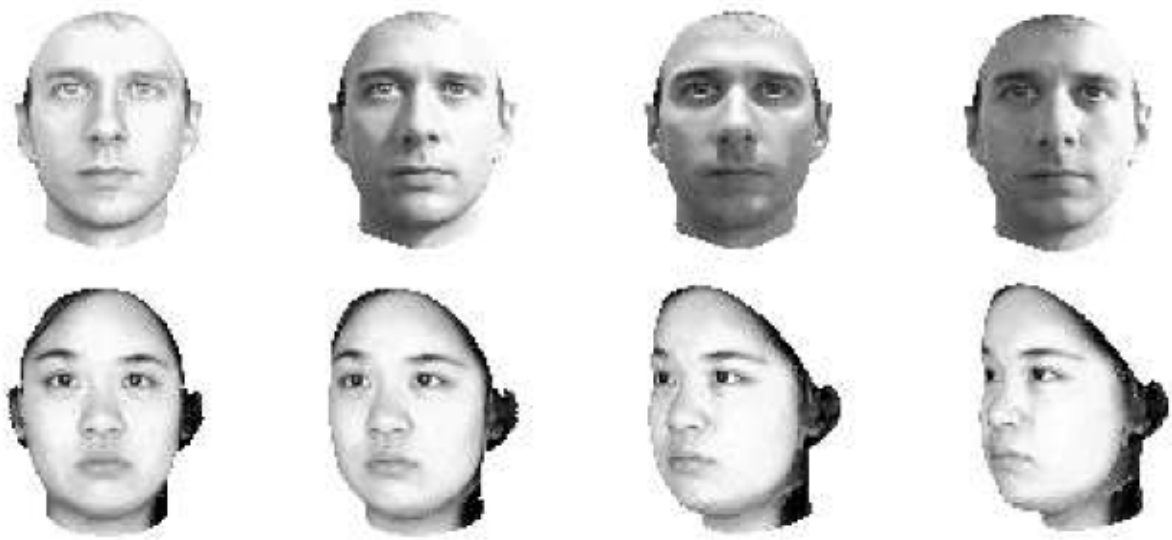
Les 2 premieres features extraites sont :




JUDYBATS
pain makes you beautiful

2. Identification par le visage

a) Introduction



b) ACP / SVD

Apprentissage :

1. Pour chaque personne à identifier, produire un ensemble de K images (appelées Gamma i)
    a. Linéariser les images (
    b. Calculer l' « image moyenne »
    c. Calculer les vecteurs propres uk (Eigenvectors) et les valeurs propres (Eigen value) de la matrice de covariance des (observations – la moyenne)

Reconnaissance :

1. Soit Γ l'image du visage à identifier
2. Pour chaque personne connue
    a. Calculer les coordonnées de Gamma dans l'espace des vecteurs propres obtenus pour cette personne

$$\omega_k = \mathbf{u}_k^T(\mathbf{\Gamma} - \mathbf{\Psi})$$

où Ψ est le centre de gravité (l'image moyenne de la personne) et uk est la kième EigenFace. Omega k est la coordonnée sur ce vecteur.

    b. Calculer l'image reconstruite uniquement à l'aide de ces coordonnées
    IR = sigma sur k de ω_k x u_k

    c. Calculer la distance (erreur quadratique moyenne) entre (Γ-Ψ) et IR
    d. Si l'erreur quadratique est inférieure à Sp alors inférer l'identification de la personne.

Exercice :

Soient 4 images de visage fortement quantifiées … sur 3 pixels. V1=[1 1 4] ; V2=[1 4 1] ; V3=[4 1 1] ; V4=[3 3 3]. On apprend un modèle du visage à l'aide des vecteurs V1, V2 et V3. On teste si le visage V4 correspond bien à la même personne. Pour cela, on reconstruit le visage à l'aide du premier vecteur propre. On estime que le visage est reconnu si l'erreur quadratique de reconstruction est strictement inférieure à 1. V4 représente-t-il le visage de la personne à reconnaître ?
(Rq : les valeurs données permettent de simplifier les calculs.)

Base d'images de visage

Moyenne des observations

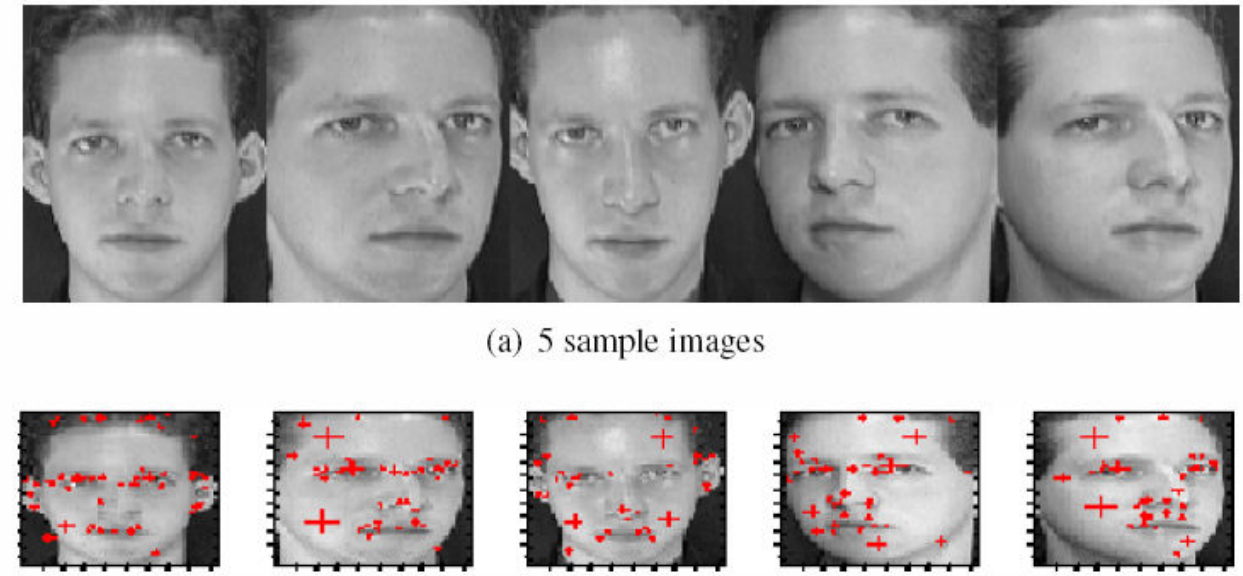
7 premiers vecteurs propres

Visage à traiter et visage reconstruit

c) Points d'intérêt

Application directe des SIFT :

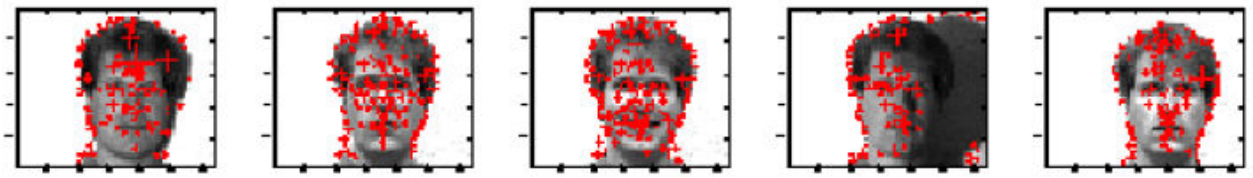
Exemple : Base de visages AT&T (10 images de 40 personnes = 400 images de 112x92 pixels) – environ 70 SIFT par image



(a) 5 sample images

Base de visage de Yale (11 images de 15 personnes = 165 images de 243x320 pixels) – environ 230 SIFT

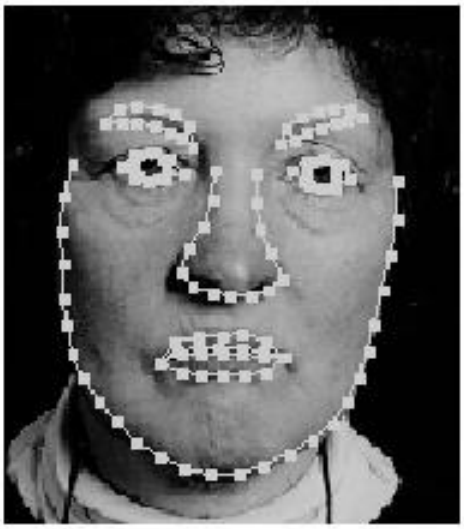
(a) 5 sample images for a subject



Résultats :

	Eigenfaces					
	Voisin le plus proche			Descripteur moyen le plus proche		
	De	Dm	Dcos	De	Dm	Dcos
AT&T	89.3	92.9	89	74.7	87.1	73.7
Yale	68.4	72	68	57.7	72.1	59.4
	SIFT					
	Voisin le plus proche					
	Dcos	Dangle				
AT&T	93.7	96.3				
Yale	85.8	91.7				

d) 1.2.3 Active shape model (ASM)



(d'apres B. Russel) Active Shape model (1995)

e) 1.2.4. Active appearance model (AAM)

Active Appearance Model

3. Analyse du corps

a) Quelques méthodes et contraintes

b) Extraction de l'image d'arrière-plan

c) Filtres de Haar spatio-temporels

d) Classification des parties du corps humain

Classification des différentes parties du corps et reconstruction du squelette 3D (Microsoft / kinnect)

Utilisation de l'image de profondeur (l'intensité d'un pixel correspond à la distance de l'objet filmé à la caméra)

Random Forest

Fusion tardive d'arbres de décision

Exercice :

On considère un ensemble de 9 images dont 5 seulement représentent des visages. On annote manuellement ces images en indiquant par un '1' la presence d'un visage et par un '0' le cas contraire. On effectue sur ces images 3 mesures k1, k2 et k3 reportées dans le tableau ci-dessous :

X	1	2	3	4	5	6	7	8	9
Y	0	0	0	0	1	1	1	1	1
K1	1	2	3	1	2	3	1	2	3
K2	1	1	1	1	1	2	2	2	2
K3	2	2	2	1	1	1	1	1	1

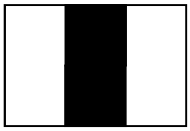
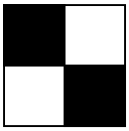
Question 1 : Construire une « random forest » de 4 arbres équilibrés (1 racine 2 sommets intermediaires et 4 feuilles) en utilisant le critère de pureté et un générateur de nombre aléatoire renvoyant successivement les valeurs 1 2 3 2 3 1 3 1 2.

Question 2 : Donnez l'expression du classifieur fort produit selon l'algorithme ADABOOST avec 2 classifieurs faibles. Pour simplifier les calculs, on considerera que tous les poids sont initialisés à 1/9 lors de la première boucle.

Exercice : On souhaite utiliser les forêts aléatoires pour réaliser la reconnaissance de formes prédéfinies. La base d'apprentissage est composée de blocs de 3x3 pixels.

Pour la caractérisation, on utilise des filtres dérivatifs du type de ceux mis en œuvre dans la méthode initiale de Viola et Jones.



Question 1 : Combien de valeurs sont utilisées pour caractériser chaque bloc ? Donnez ces valeurs qui caractérisent le bloc suivant :

1	2	3
2	3	1
3	1	2

On définit 3 caractéristiques pour l'ensemble des blocs et on effectue un apprentissage sur 3 blocs notés x1 , x2, et x3. Le tableau ci-dessous donne les valeurs de la vérité terrain pour chaque bloc et celles des 3 caractéristiques :

Bloc	x1	x2	x3
Vérité terrain	0	1	1
Caractéristique 1	1	3	3
Caractéristique 2	1	1	3
Caractéristique 3	3	3	3

On cherche à produire un arbre de hauteur 2.

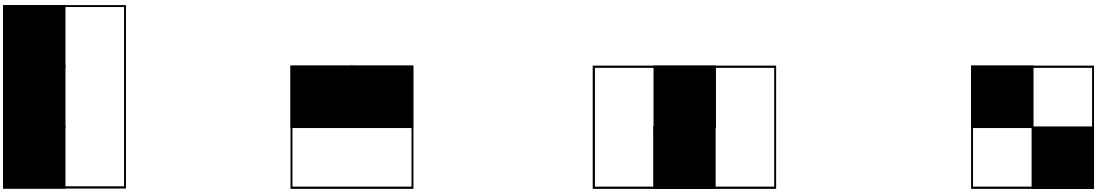
Question 2 : que se passe-t-il si le générateur aléatoire donne la séquence de caractéristiques suivante : 1 2 3 ? Donnez l'arbre résultant et indiquez quelle est la probabilité d'erreur d'une reconnaissance appliquée sur la base d'apprentissage.

Question 3 : même question pour la séquence 3 1 2. Explicitez vos choix.

Question 4 : On construit une forêt composée de tous les arbres différents de hauteur 2 qu'il est possible de construire à l'aide du générateur aléatoire. Combien d'arbres strictement différents est-on susceptible de construire ? Quelle est la probabilité d'erreur de cette forêt appliquée sur la base d'apprentissage ? Argumentez vos réponses.

Exercice : On souhaite utiliser les forêts aléatoires pour réaliser la reconnaissance de formes prédéfinies. La base d'apprentissage est composée de blocs de 3x3 pixels.

Pour la caractérisation, on utilise des filtres dérivatifs du type de ceux mis en œuvre dans la méthode initiale de Viola et Jones.



Question 1 : Combien de valeurs sont utilisées pour caractériser chaque bloc ? Donnez ces valeurs qui caractérisent le bloc suivant :

1	2	3
2	3	1
3	1	2

On définit 3 caractéristiques pour l'ensemble des blocs et on effectue un apprentissage sur 3 blocs notés x1 , x2, et x3. Le tableau ci-dessous donne les valeurs de la vérité terrain pour chaque bloc et celles des 3 caractéristiques :

Bloc	x1	x2	x3
Vérité terrain	0	1	1
Caractéristique 1	1	3	3
Caractéristique 2	1	1	3
Caractéristique 3	3	3	3

On cherche à produire un arbre de hauteur 2.

Question 2 : que se passe-t-il si le générateur aléatoire donne la séquence de caractéristiques suivante : 1 2 3 ? Donnez l'arbre résultant et indiquez quelle est la probabilité d'erreur d'une reconnaissance appliquée sur la base d'apprentissage.

Question 3 : même question pour la séquence 3 1 2. Explicitez vos choix.

Question 4 : On construit une forêt composée de tous les arbres différents de hauteur 2 qu'il est possible de construire à l'aide du générateur aléatoire. Combien d'arbres strictement différents est-on susceptible de construire ? Quelle est la probabilité d'erreur de cette forêt appliquée sur la base d'apprentissage ? Argumentez vos réponses.

B. Analyse du mouvement

1. Flot optique

Hypothèse du flot optique.

a) Algorithmes d'appariement de blocs

2 outils sont nécessaires pour calculer l'appariement de blocs : un critère de qualité, une méthode de recherche

- **Recherche exhaustive**
- **Recherche en 3 étapes**
- **Recherche logarithmique 2D**
- **Recherche en croix**
- **Recherche binaire**
- **Recherche en 4 étapes**
- **Recherche orthogonale**
- **Recherche "1 à la fois"**
- **Appariements de blocs hiérarchique**

Comparaisons

Comparaison de la distorsion (moyenne de la valeur absolue la différence des minimums)

Rech. exhaustive	3 étapes	2D log	4 étapes	1 à la fois	Rech. orthogonale
16.7	19.3	23.8	19.4	20.1	19.7

Comparaison du temps de calcul

Rech. exhaustive	3 étapes	2D log	4 étapes	1 à la fois	Rech. orthogonale

1642.1	65.2	21.3	37.19	15.4	38.5

Exercice : Codage du mouvement

On encode dans un format pseudo-MPEG1/2 une séquence d'images en niveau de gris de résolution 4x4 pixels. Les deux premières images de la séquence sont :

1	0	0	0
1	0	0	0
0	2	0	0
0	2	0	0

0	1	0	0
0	1	0	0
0	0	2	0
0	0	2	0

Question : l'image 2 est encodée de manière prédictive (P) à partir de l'image 1. Exprimez un meilleur appariement possible des blocs de I1 et de I2 selon le procédé utilisé dans l'encodage MPEG. En fonction de votre réponse, indiquez quelle est l'erreur résiduelle (c'est-à-dire la différence) entre l'image telle qu'elle peut être prédite et l'image I2.

b) Methodes differentielles

- **Horn et Schunk**
- **Lukas et Kanade**

Exercice : Donner la valeur du vecteur de déplacement obtenu avec la méthode de Lucas et Kanade en pour les trois pixels sur fond grisé en utilisant un 8-voisinage pour les calculs. Donner une valeur de seuil sur les valeurs propres pour que seul le déplacement du point à 1 soit retenu.

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

Image t

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

Image t+1

c) Corrélation de phase

Exercice : Donner selon cette méthode la valeur du vecteur de déplacement entre les 2 images suivantes :

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

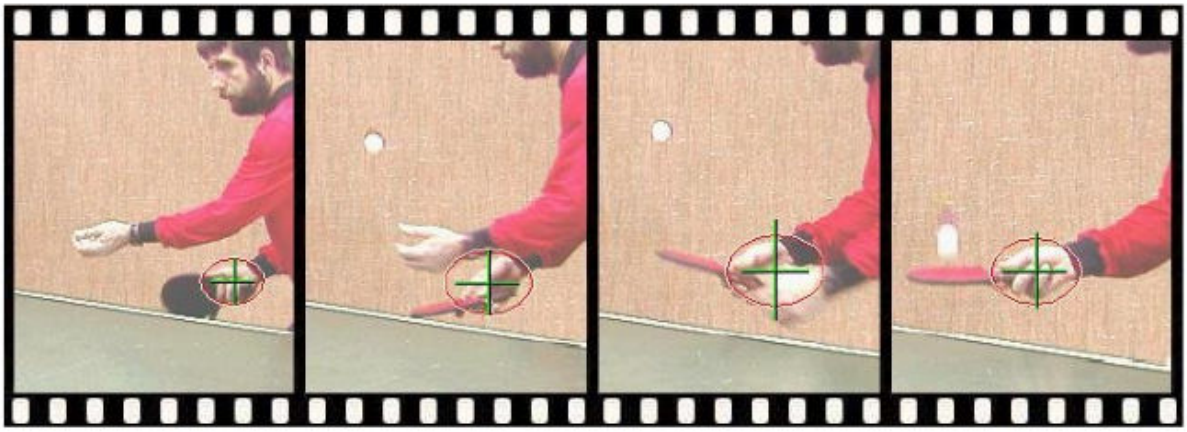
Image t

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

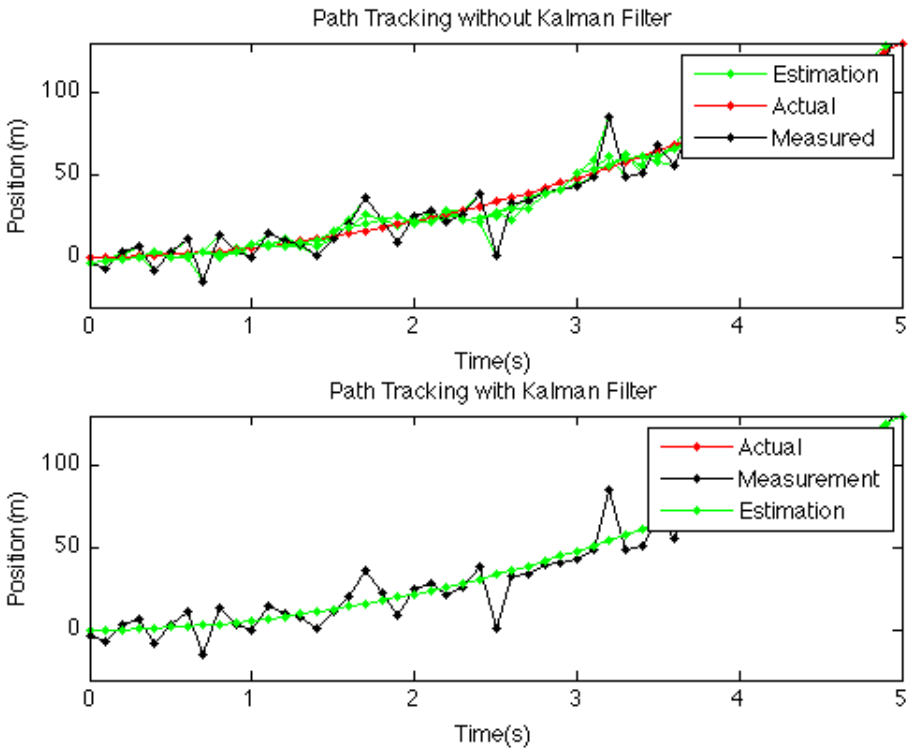
Imaget+1

2. Tracking

a) Mean shift



b) Filtre de Kalman



Modélisation de l'évolution d'un état : $X_{k+1} = A_k X_k + U_k$

Soit Q la moyenne de plusieurs matrices de covariance de $U_k.U_k^t$

Modélisation de l'évolution d'une mesure : $Y_k = HX_k + V$

Soit R la matrice de covariance centrée de V (estimée de la même manière que Q)

Algorithme :

Etape 1 : Prédiction

$$\hat{X}_k = A_k X_k$$

$$\hat{P}_k = A_k P_k A_k^t + Q$$

Etape 2 : Mise à jour

$$K_{k+1} = \hat{P}_k.H^t\left(H.\hat{P}_k.H^t + R\right)^{-1}$$

$$P_{k+1} = \left(I - K_{k+1}.H\right)\hat{P}_k\left(I - K_{k+1}.H\right)^t + K_{k+1}.R.K_{k+1}^t$$

$$X_{k+1} = \hat{X}_k + K_{k+1}\left(y_{k+1} - H.\hat{X}_k\right)$$