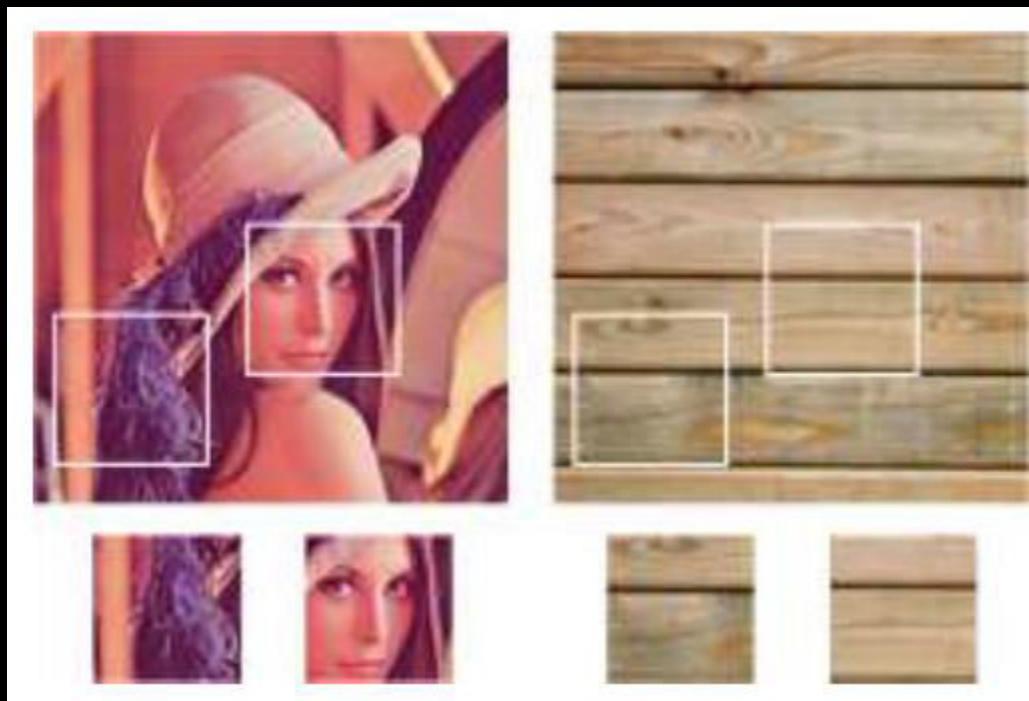
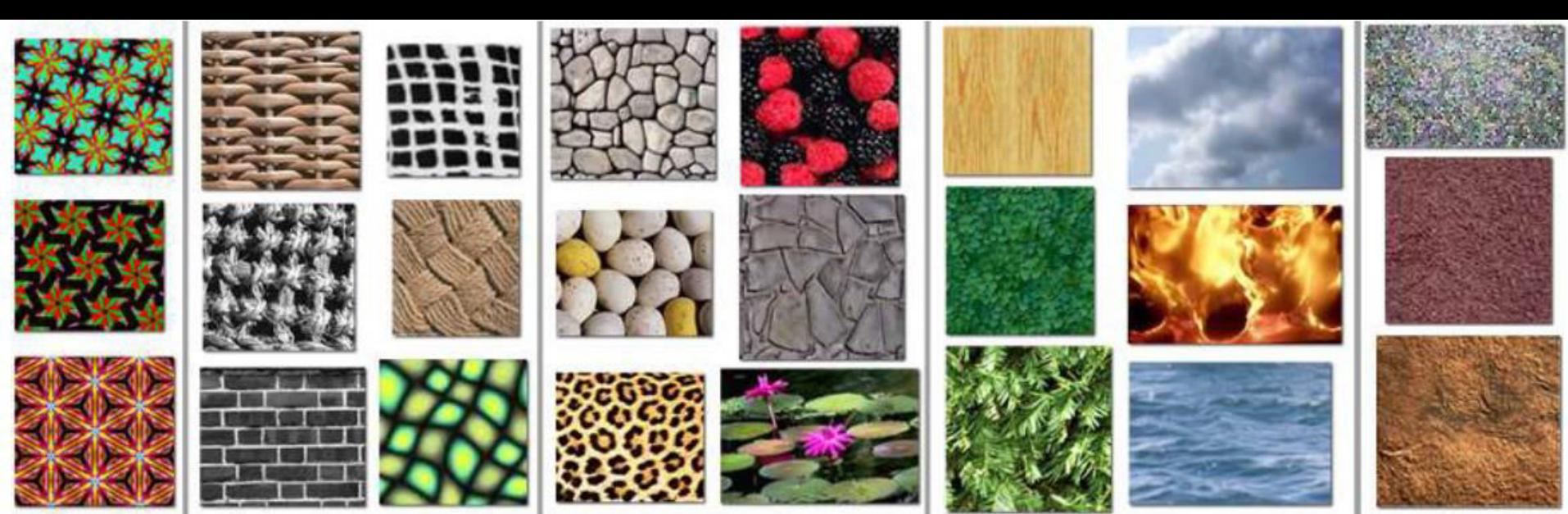


Extraction de caractéristiques

Traitemet d'images cours n°6/6
(suite)

Texture

- Comment analyser une image de texture ?
 - Stats du 1^{er} ordre
 - Matrice de cooccurrence
 - Bancs de filtres (Gabor)
 - Local Binary Patterns (LBP)
- Dans quel cadre utilise-t-on ces outils ?
 - Pour la segmentation, classification, CBIR
 - On peut aussi faire de la synthèse de texture





Les textures : quelle définition ?

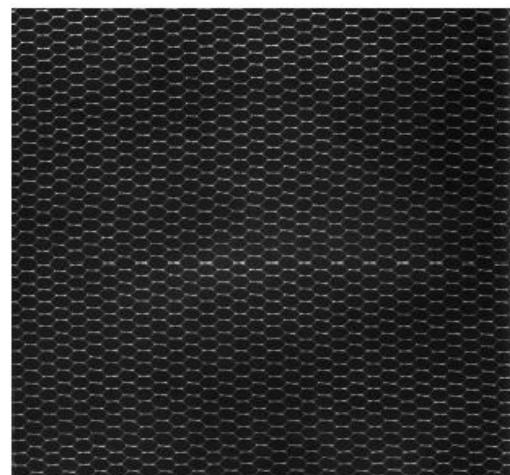
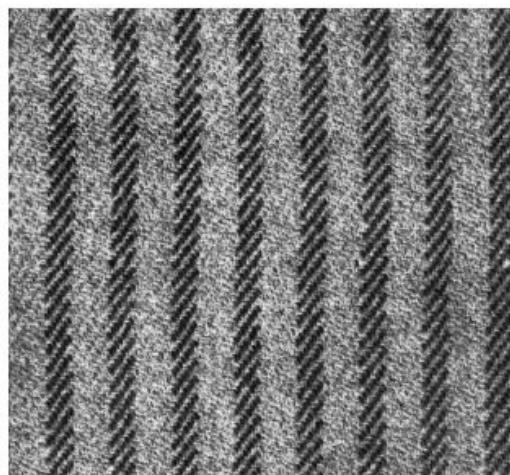
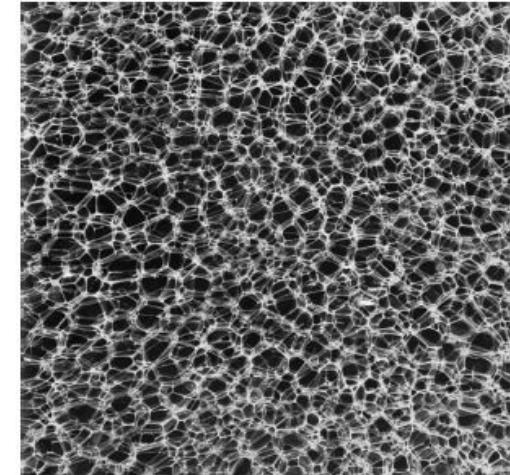
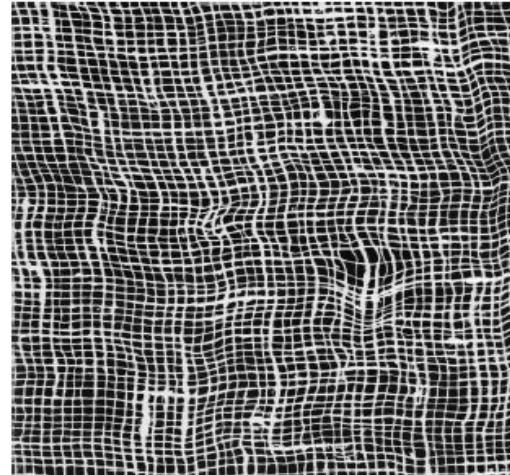
- **Textile** : disposition et mode d'entrecroisement des fils dans un tissage.
- **Culinaire** : Qualité physique des aliments liée à leur densité, leur viscosité, leur caractère homogène, leur dureté.
- **Matériaux** : Caractéristique relative à la dimension, la forme, la disposition des grains dans le métal.
- **Arts** : texture musicale, texture d'un drame.

En vision par ordinateur, il n'y a pas de définition complètement satisfaisante :

- Zone homogène en un certain sens, assemblage d'entités élémentaires formant un tout.
- Description des variations d'intensité lumineuse sur une surface, rendant compte de propriétés telles que la rugosité, la douceur, la granularité, etc.



Des exemples de texture



Exemples de textures de Brodatz



Comment analyser des textures ?

■ Difficultés

- Pas de modèle mathématique suffisement général et manipulable
- Pas de notion satisfaisante de distance entre textures
- Notion fortement dépendante de l'échelle



Applications

- Reconnaissance et classification de matériaux, contrôle de qualité.
- Reconnaissance d'objets, **segmentation**.
- Synthèse et restauration d'images



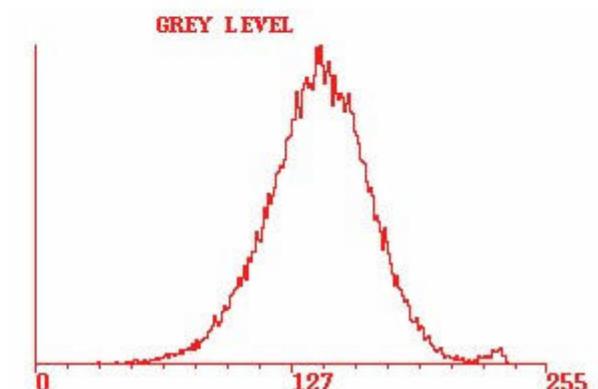
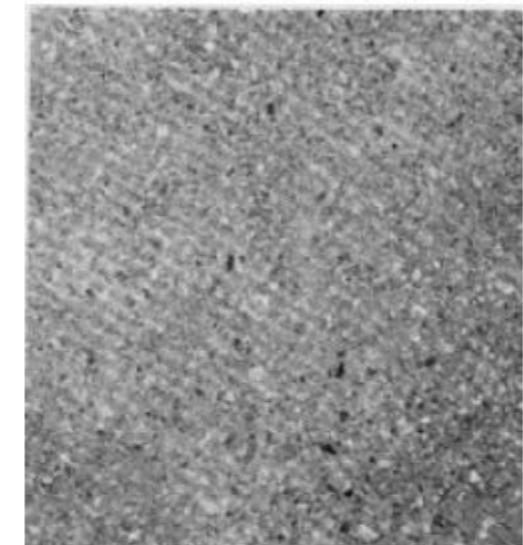
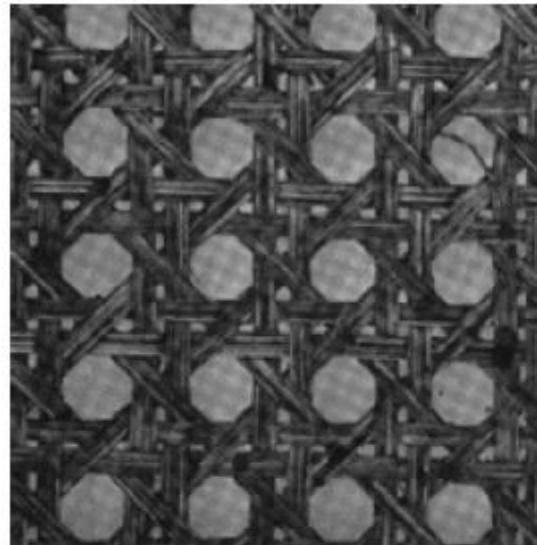
Comment analyser des textures ?

- **Quelques exemples de méthodes de caractérisation statistique**
 - Statistiques d'ordre 1
 - Matrices de cooccurrence
 - Analyse par un banc de filtres : filtres de Gabor
 - Décomposition en ondelettes

Statistiques d'ordre 1 et d'ordre supérieur



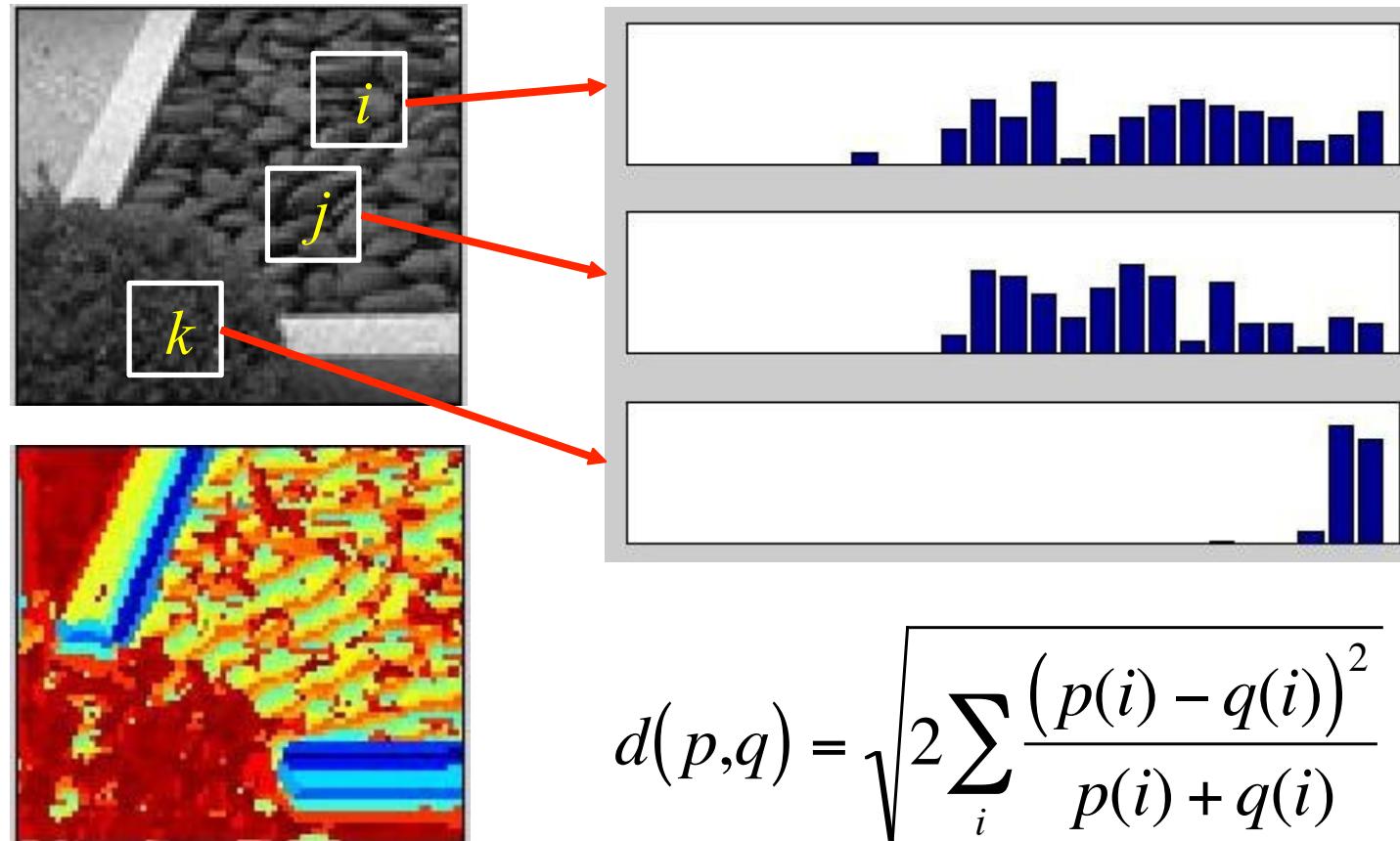
Statistiques d'ordre 1





Statistiques d'ordre 1 : distance entre histogrammes

Distance du Chi² entre histogrammes de textures



$$d(p,q) = \sqrt{2 \sum_i \frac{(p(i) - q(i))^2}{p(i) + q(i)}}$$



Statistiques d'ordre 1: descripteurs

- On peut calculer des **descripteurs** à partir des histogrammes

- Moyenne $M = \sum u_p$

- Variance $\sigma^2 = \sum u_p^2 - M^2$

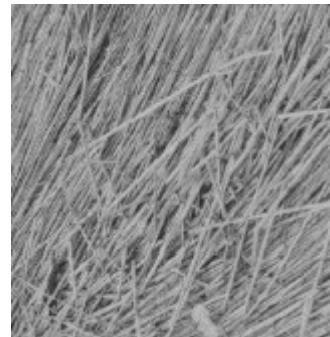
- Kurtosis $\kappa = \frac{\sum (u_p - M)^4}{\sigma^4} - 3$

- Entropie $E = -\sum h(v)\log(h(v))$

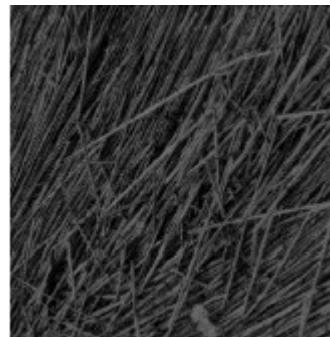
-



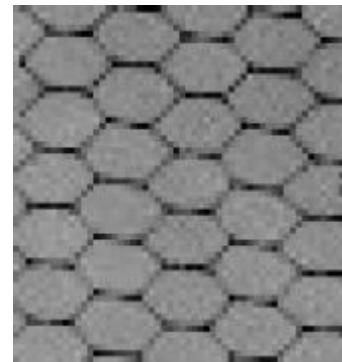
Statistiques d'ordre 1: descripteurs



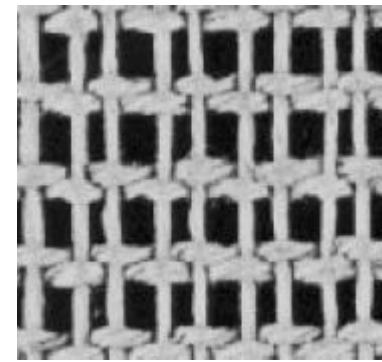
moyenne=137



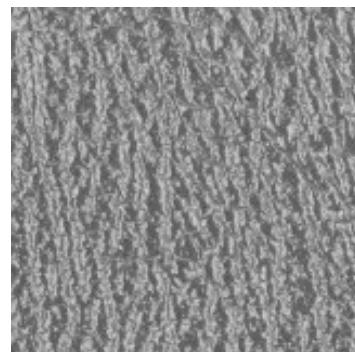
moyenne=50



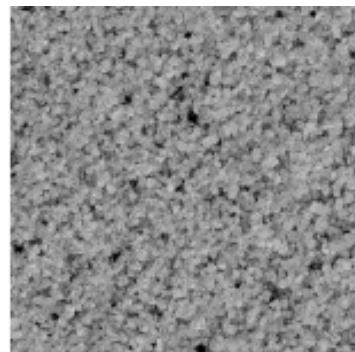
variance=30



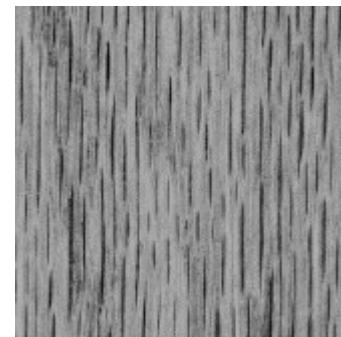
variance=71



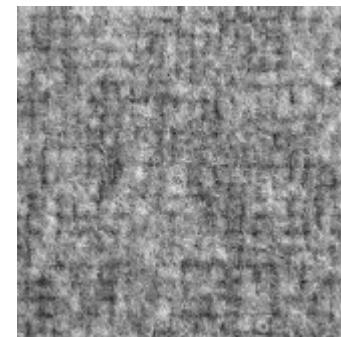
skew=0.25



skew=-1.15



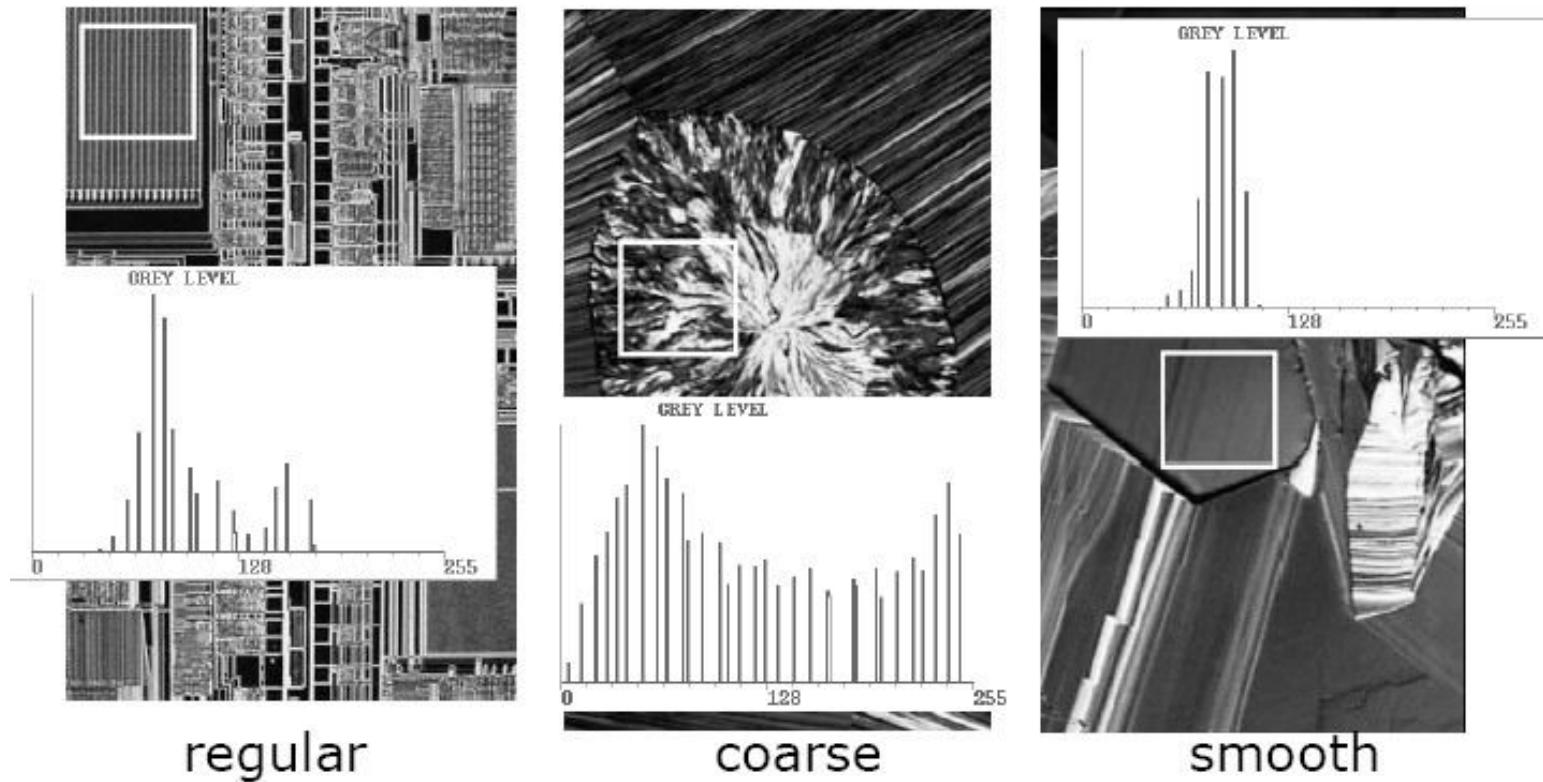
kurt=2.31



kurt=-0.09



Statistiques d'ordre 1: descripteurs



texture	m_1	σ	R_{norm}	μ_3	U	e
regular	99.72	33.73	0.017	0.750	0.013	6.674
coarse	143.56	74.63	0.079	-0.151	0.005	7.783
smooth	82.64	11.79	0.002	-0.105	0.026	5.434



Matrices de cooccurrences (Haralick'73)

- **Définition** : Il s'agit de statistiques du second-ordre de l'image. Pour un voisinage (dx,dy) , la matrice de cooccurrence $M(dx,dy)$ est donnée par :

$$M_{[dx,dy]}(u,v) = \frac{1}{(N_x - dx)(N_y - dy)} \sum_{i,j} 1[I(i,j) = u \& I(i + dx, j + dy) = v]$$

- (N_x, N_y) : taille de l'image
- (u,v) : niveaux de gris de l'image (valeur quantifiée)
- Chaque matrice est de taille $G \times G$ où G est le nombre de niveaux de gris (on considère généralement 8 ou 16 niveaux)
- On considère généralement des matrices de cooccurrences pour un ensemble de voisinage (dx,dy) (e.g., $(1,0)$, $(0,1)$, $(1,1)$,)
- **Remarque** : les statistiques de cooccurrences sont les statistiques suffisantes d'un champ de Markov défini sur des cliques d'ordre 2

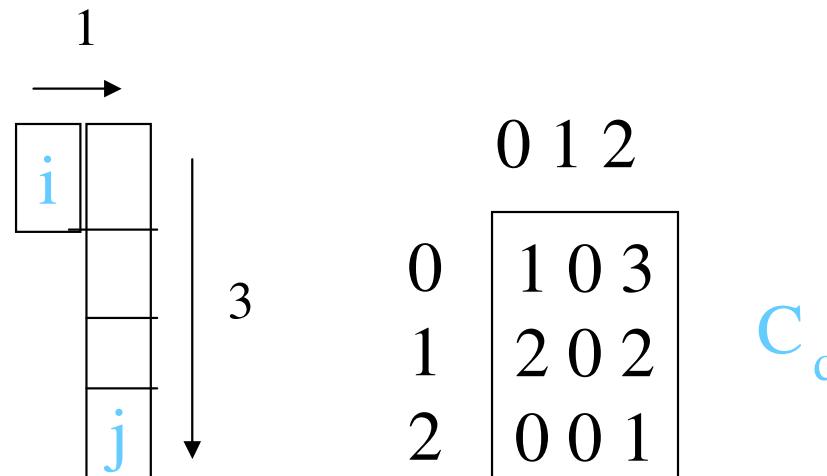
Texture description

Co-occurrence matrices and features

C- matrix(256,256).

$C(i,j)$ =how many times intensity i co-occurs with intensity j in some designated spatial relationship.

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



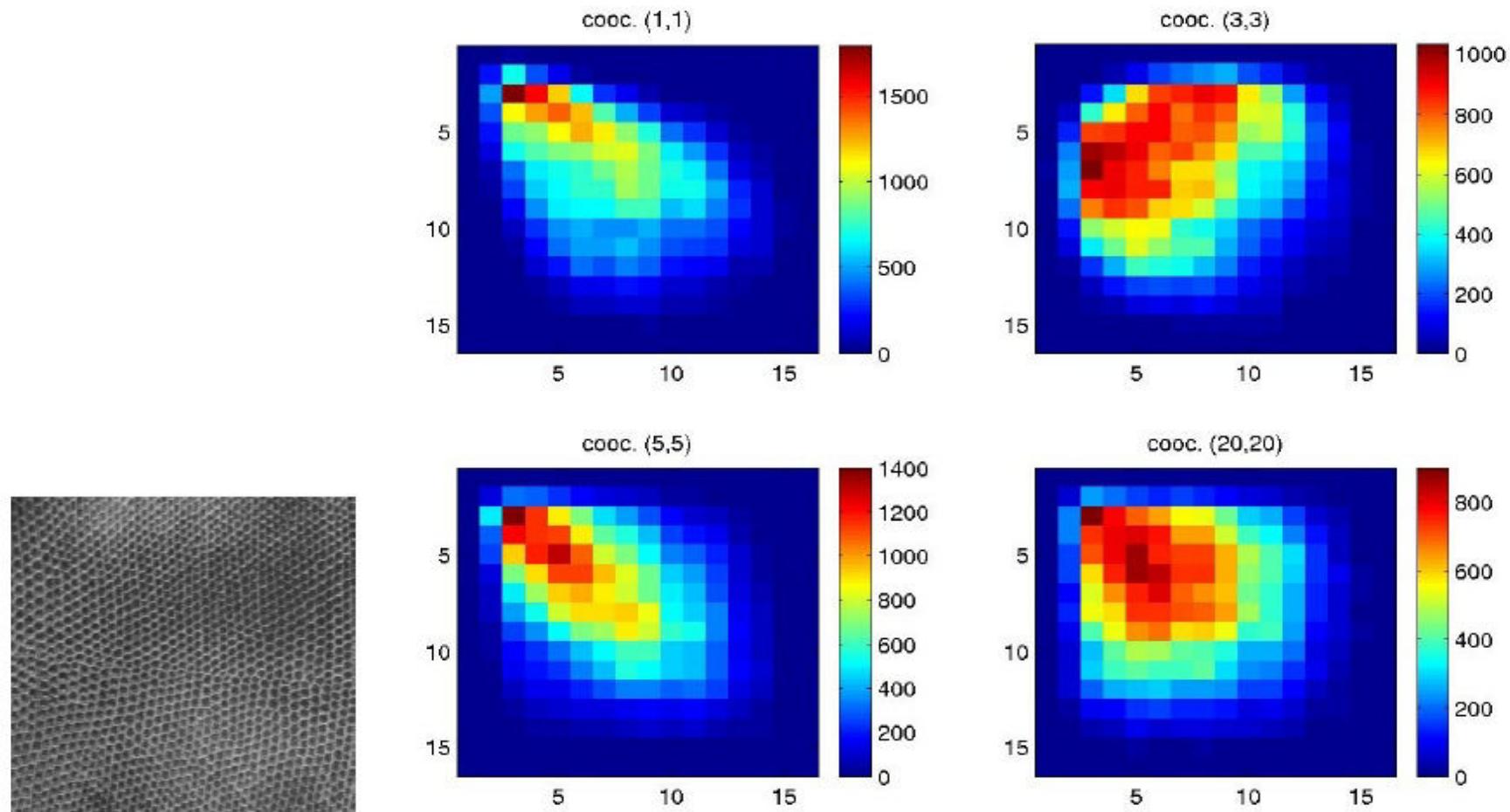
$d = (3,1)$

co-occurrence
matrix



Matrices de cooccurrences (Haralick'73)

Exemples de matrices de cooccurrences





Matrices de cooccurrences (Haralick'73)

- Calcul de descripteurs

$$MOYENNE = \sum_j \sum_i p(i, j)$$

$$VARIANCE_i = \sum_j \sum_j (i - MOYENNE)^2 p(i, j)$$

$$ENERGIE = \sum_j \sum_i (p(i, j))^2$$

$$INERTIE = \sum_j \sum_i (i - j)^2 p(i, j)$$

$$ENTROPIE = -\sum_j \sum_i p(i, j) \log(p(i, j))$$

$$MOMENT DIFFERENTIEL INVERSE = \sum_i \sum_j \frac{1}{1 + (i - j)^2} p(i, j)$$

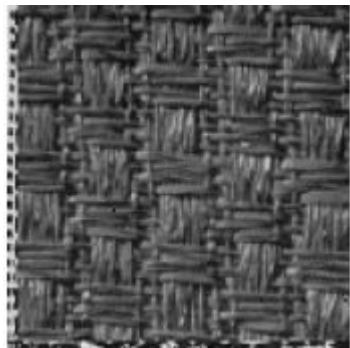
où

(i,j) : coordonnées dans la matrice

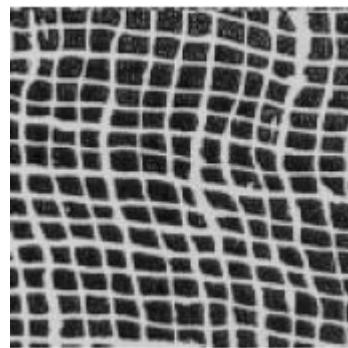
p(i,j) : valeurs normalisées de la matrice



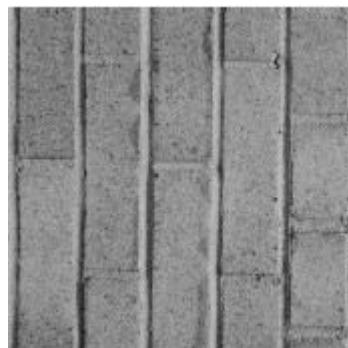
Matrices de cooccurrences (Haralick'73)



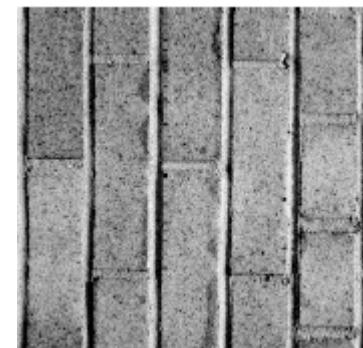
contraste=120



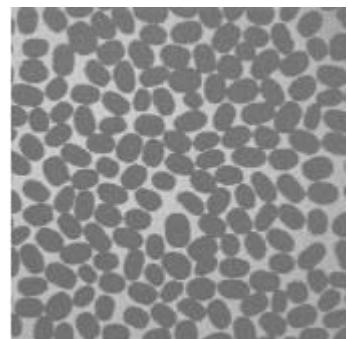
contraste=626



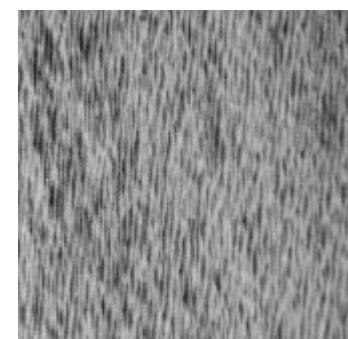
contraste=42



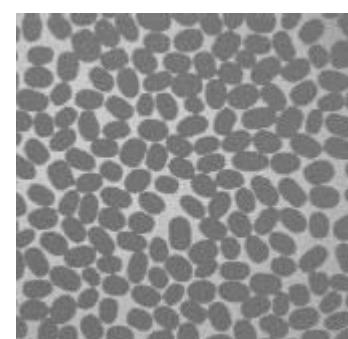
contraste=142



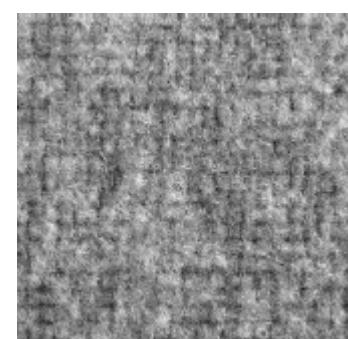
entropie=6.7



entropie=4.7



mdi=0.03



mdi=0.002

Approche par bancs de filtre



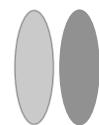
Analyse fréquentielle

- Une texture est une répétition d'éléments avec une certaine **fréquence**
- Filtrage de l'image en utilisant des méthodes spectrales
 - Fourier
 - **Filtres de Gabor**
 - Analyse en ondelette...

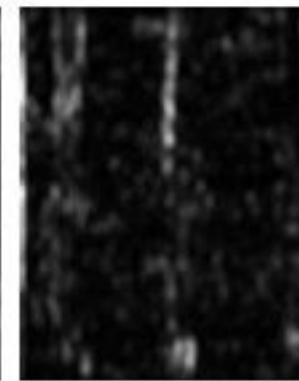
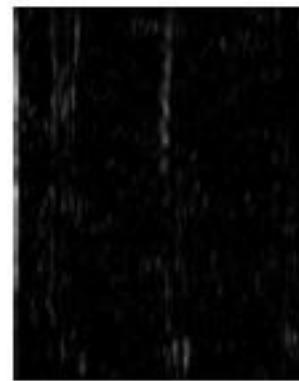


Analyse fréquentielle

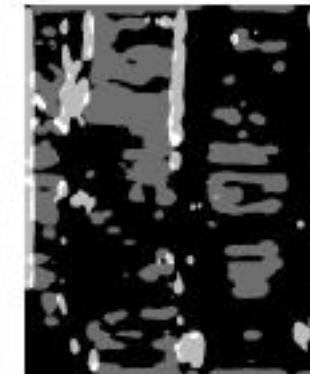
Réponses du carré



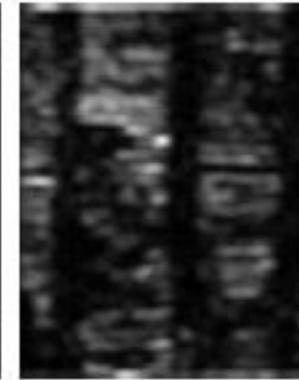
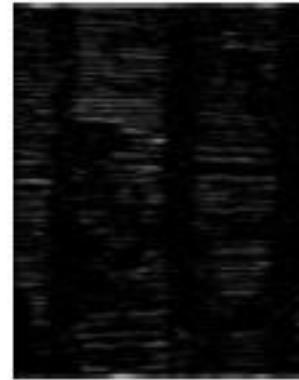
vertical



classification



horizontal

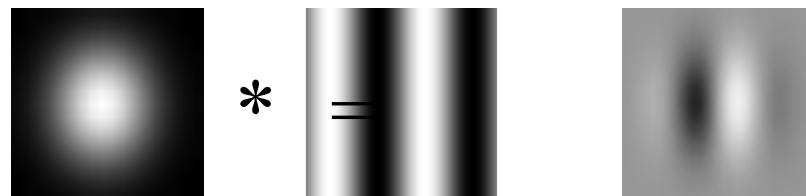


Lissage



Filtres de Gabor

- Les filtres de Gabor sont le produit de Gaussiennes par des sinus ou des cosinus
 - Sinus / cosinus : analyse en fréquences
 - Gaussienne : différents niveaux de lissage (échelles)



Gaussienne * Sinus/Cosinus = Gabor

$$G_s(x, y) = \cos(\omega_x x + \omega_y y) e^{-\frac{x^2+y^2}{2\sigma^2}}$$

$$G_a(x, y) = \sin(\omega_x x + \omega_y y) e^{-\frac{x^2+y^2}{2\sigma^2}}$$



Bancs de filtres : filtres de Gabor

- **Principe de l'analyse de texture par banc de filtres** : La texture est caractérisée par différentes statistiques (moyenne, variance, histogramme,...) des réponses aux filtres (énergie).
- **Filtres de Gabor** : Il s'agit de filtres extrayant une information fréquentielle localisée et orientée. Le filtre de Gabor horizontal en (0,0) de paramètres (σ_1 , σ_2, f_0) est donné par :

$$G_{\theta,\sigma}(x,y) = \exp\left[-\frac{\tilde{x}^2 + \tilde{y}^2}{2\sigma^2}\right] \cos(2\pi\lambda \cdot \tilde{x}/\sigma) \quad \begin{pmatrix} \tilde{x} \\ \tilde{y} \end{pmatrix} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$$

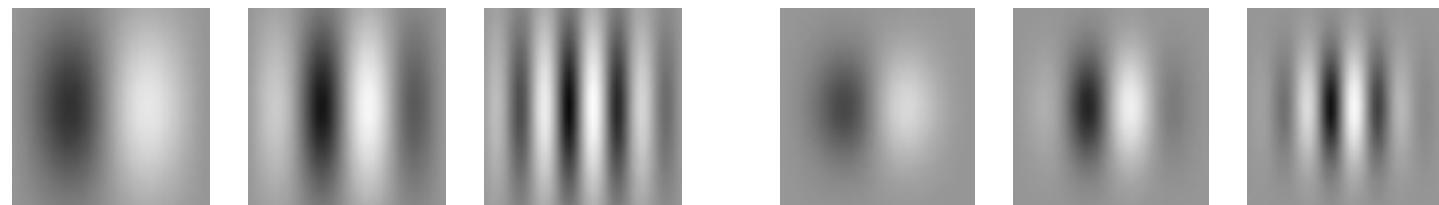
- Application à l'analyse de texture : 5-10 orientations et 5-10 fréquences.



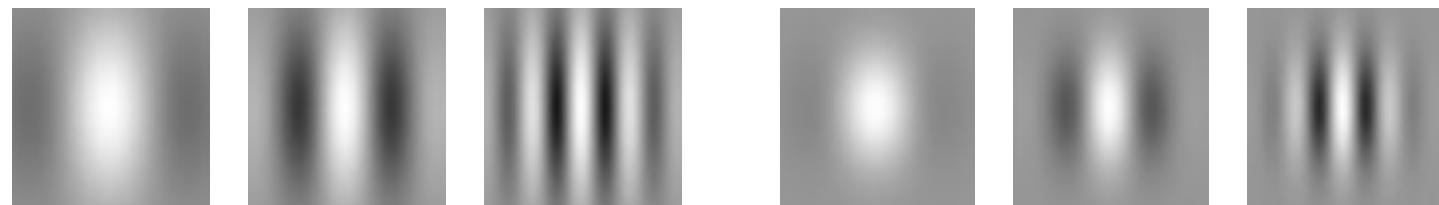
Filtres de Gabor

- Les filtres de Gabor analysent la structure
 - à différentes échelles (fréquences)
 - à différentes orientations (angles)

Filtres
anti-symétriques
(impairs - sinus)



Filtres
symétriques
(pairs - cosinus)



Texture description

How many filters and at what orientation?

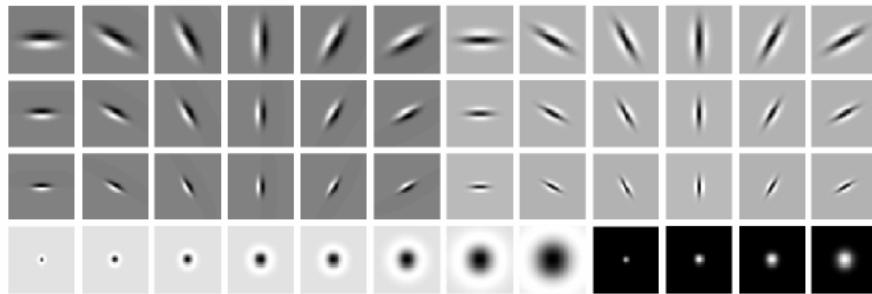


Figure 7. The LM filter bank has a mix of edge, bar and spot filters at multiple scales and orientations. It has a total of 48 filters - 2 Gaussian derivative filters at 6 orientations and 3 scales, 8 Laplacian of Gaussian filters and 4 Gaussian filters.

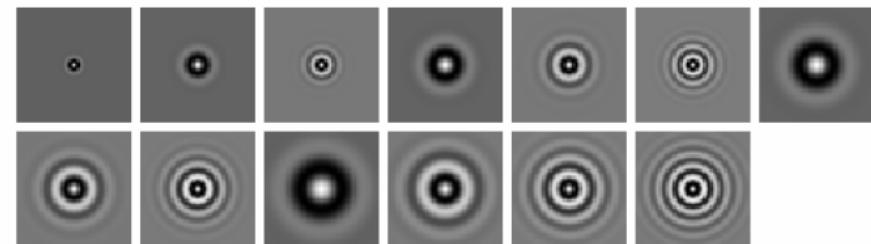


Figure 8. The S filter bank is rotationally invariant and has 13 isotropic, “Gabor-like” filters.

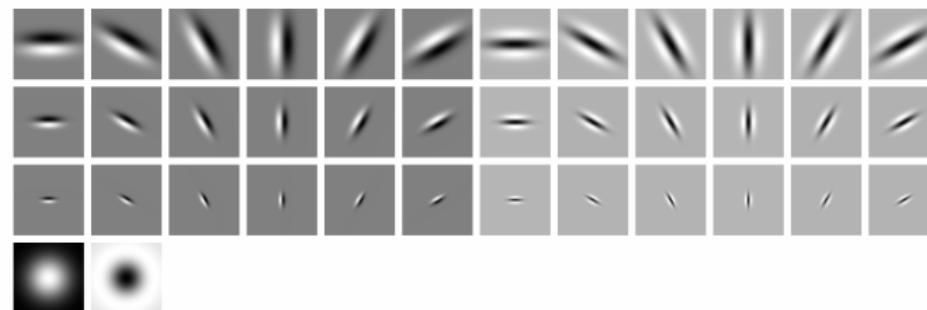


Figure 9. The MR8 filter bank consists of 2 anisotropic filters (an edge and a bar filter, at 6 orientations and 3 scales), and 2 rotationally symmetric ones (a Gaussian and a Laplacian of Gaussian). However only 8 filter responses are recorded by taking, at each scale, the maximal response of the anisotropic filters across all orientations.

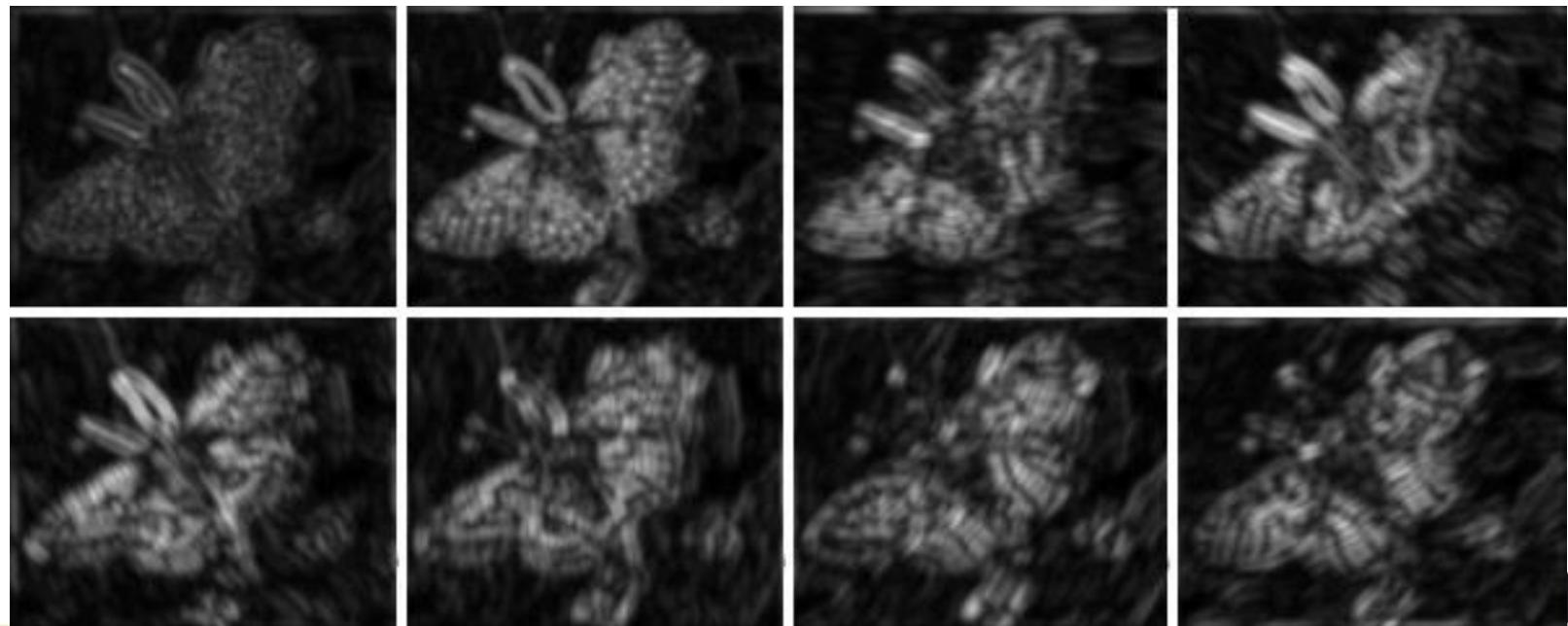


Filtres de Gabor

*Image
d'entrée*



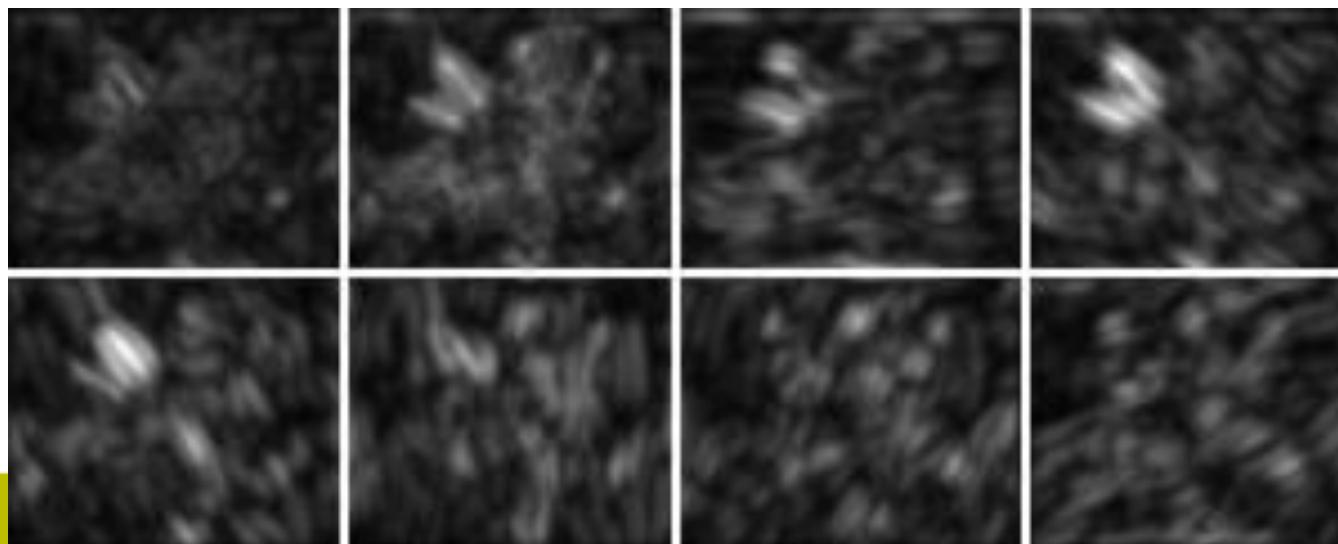
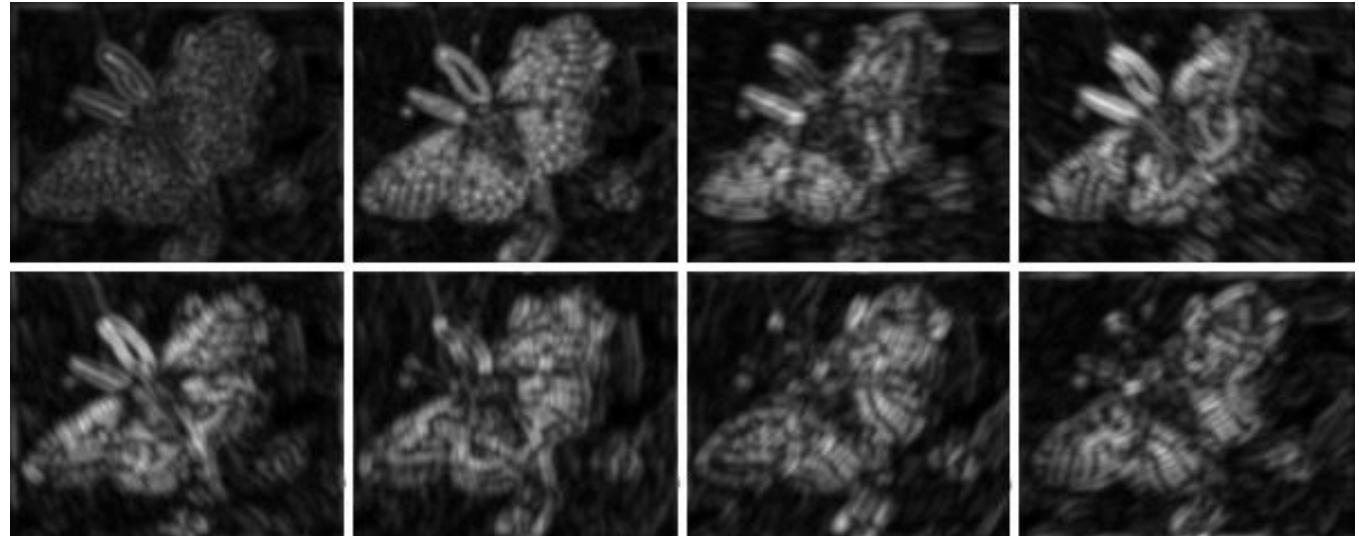
*Banc de
filtres*



Réponse des filtres à une échelle



Filtres de Gabor à différentes échelles



Description de texture par LBP

Local binary partition

- For each pixel p , create an 8-bit number $b_1 b_2 b_3 b_4 b_5 b_6 b_7 b_8$, where $b_i = 0$ if neighbor i has value less than or equal to p 's value and 1 otherwise.
- Represent the texture in the image (or a region) by the histogram of these numbers.

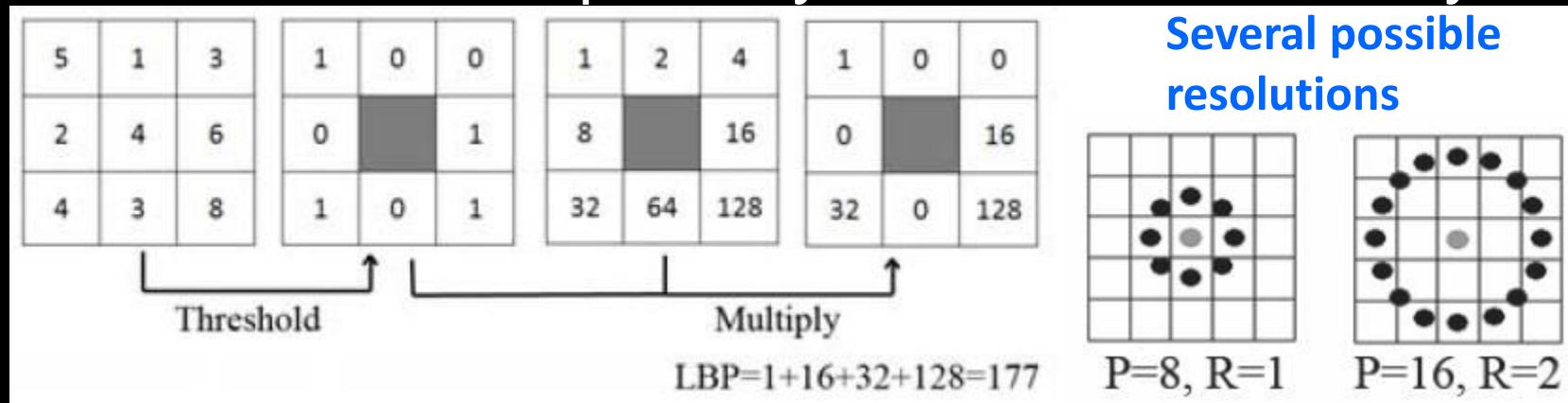
100	101	103
40	50	80
50	60	90



1	1	1	1	1	1	0	0
---	---	---	---	---	---	---	---

Feature extraction: low-level descriptors

- Local Binary Patterns [Ojala, 1996]
 - Texture description by distribution of binary



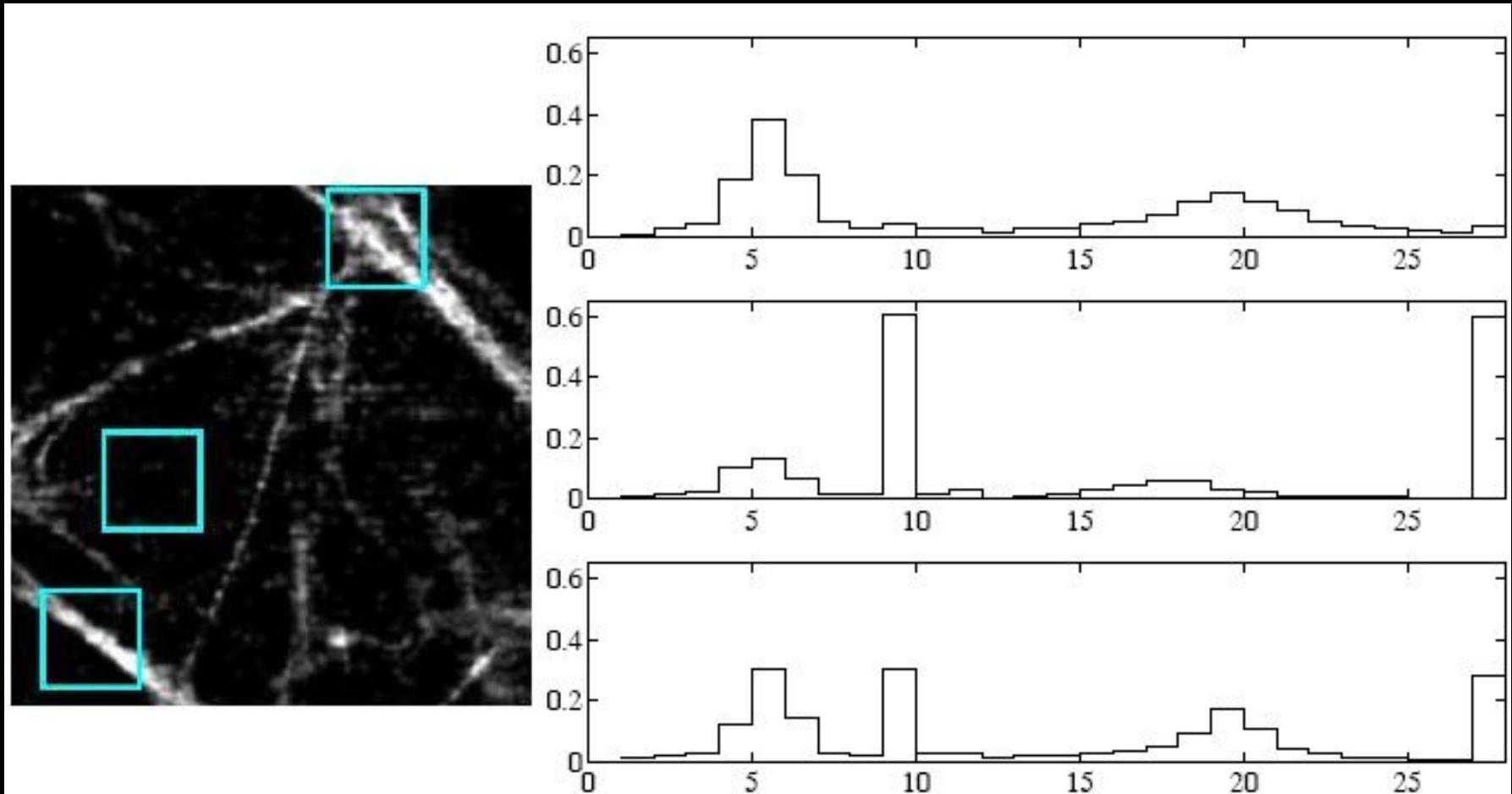
$$LBP_{P,R} = \sum_{p=0}^{P-1} s(g_p - g_c) \cdot 2^p$$

$$VAR_{P,R} = \frac{1}{P} \sum_{p=0}^{P-1} (g_p - \mu), \text{ where } \mu = \frac{1}{P} \sum_{p=0}^{P-1} g_p$$

Gray-scale and rotation invariant
spatial structure of image texture

Texture contrast

Descripteur de texture LBP

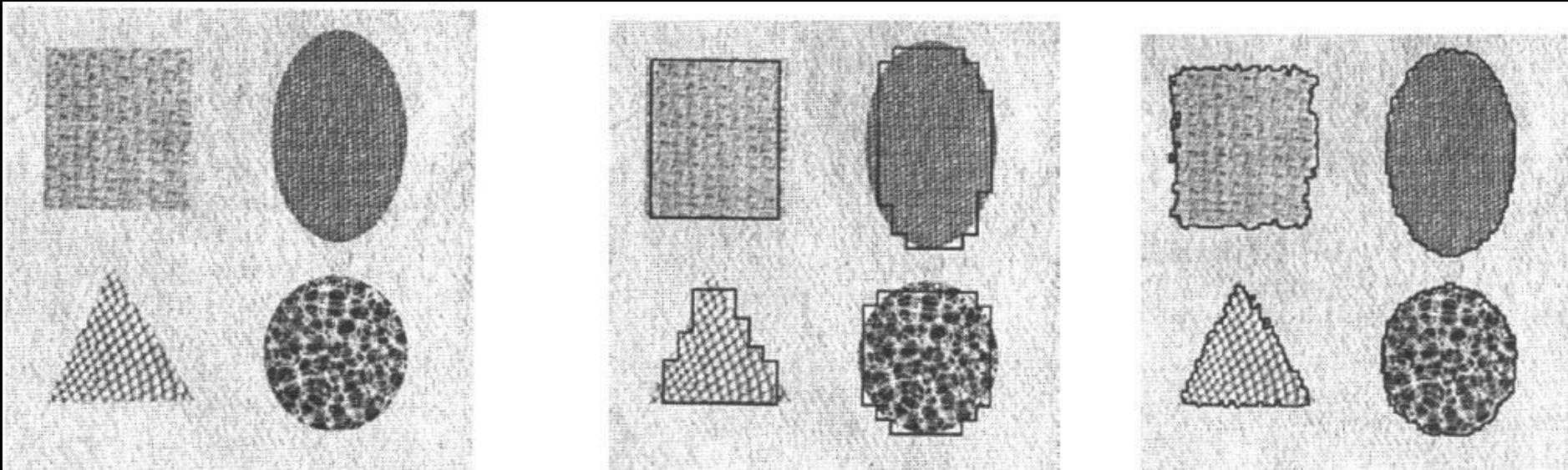


Texture par LBP

- Pour chaque pixel, on calcule son « pattern » correspondant.
- Pour la résolution P , il y a $P+2$ patterns possibles.
- Pour chacun des $P+2$ patterns, on compte le nombre de pixels ayant ce pattern (distribution).
- L'image est caractérisée par l'histogramme des patterns.

Utilisation de la texture

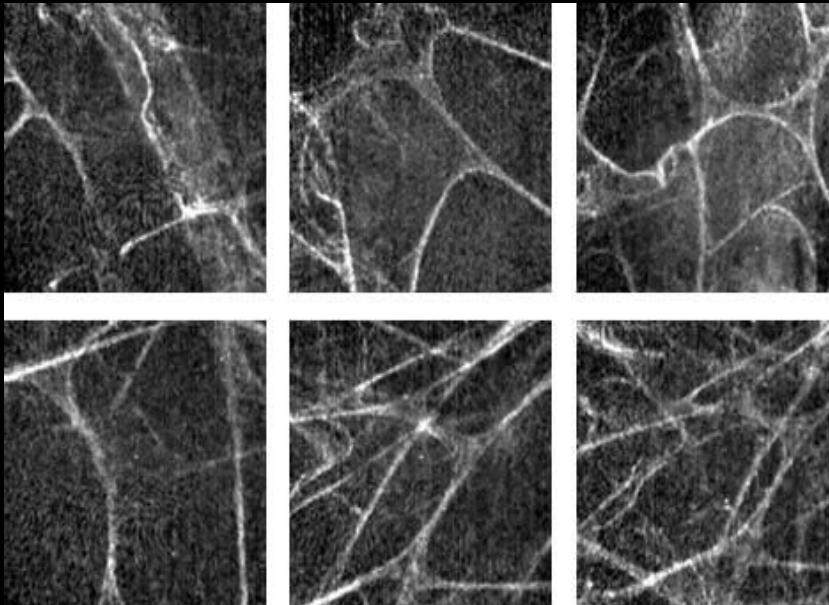
- Pour la segmentation
 - Exemple avec split and merge



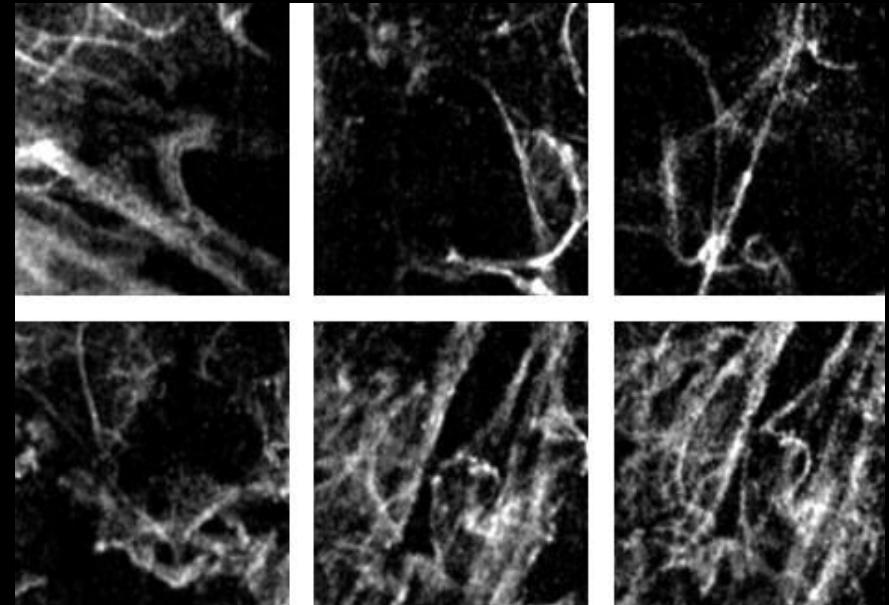
Utilisation de la texture

- Pour la classification d'images

Patient sain

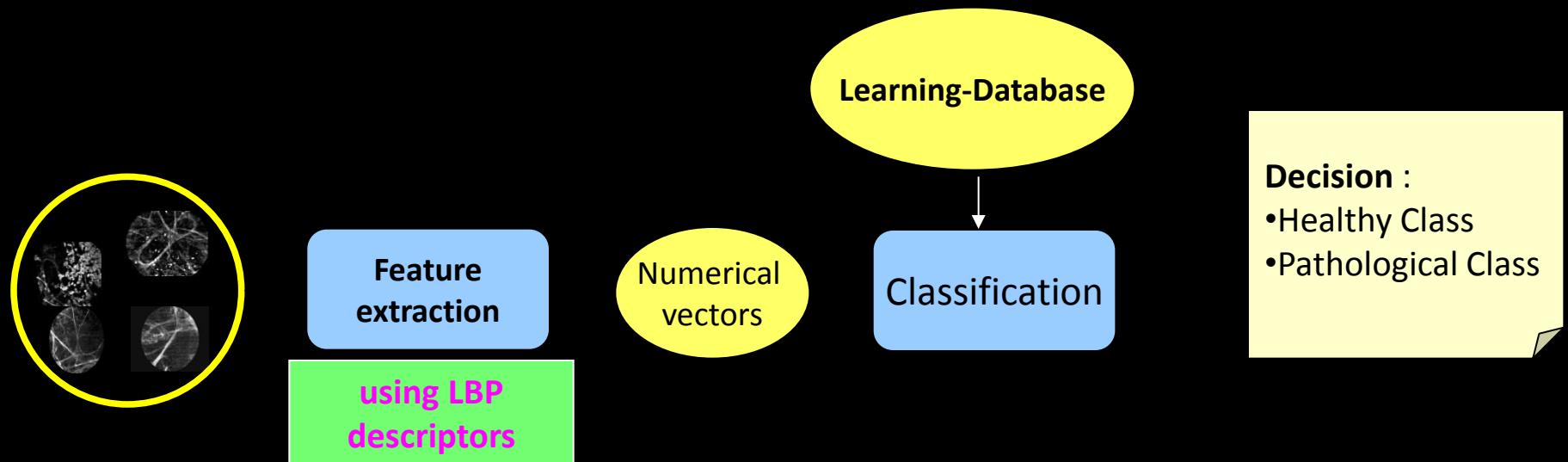


Patient malade



Système de classification

■ Classification sain/malade



Utilisation de la texture

- Dans les systèmes de requête par le contenu
(CBIR = content-based image retrieval)

Fids (Flexible Image Database System) is retrieving images similar to the query image using LBP texture as the texture measure and comparing their LBP histograms





Query image

Ordered list of
best matches



Decreasing
response
vector
similarity



Synthèse de texture

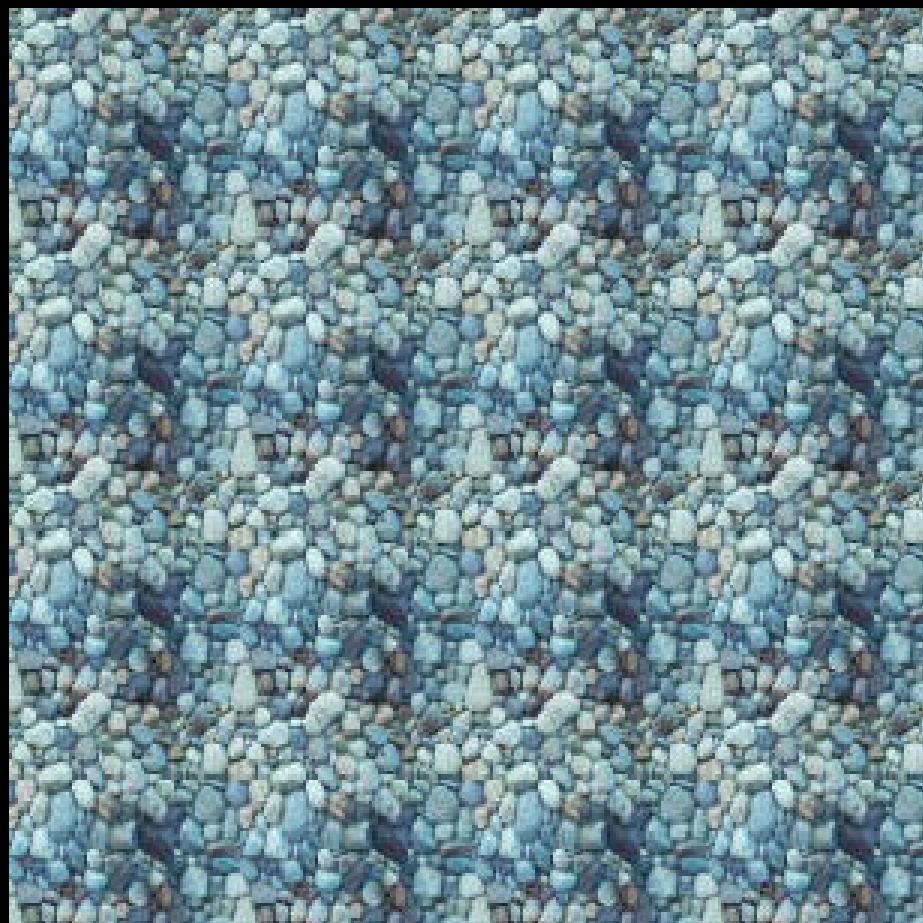


Synthèse de texture

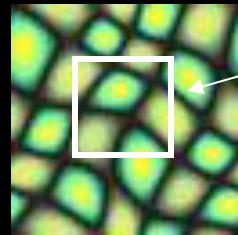
Répétition du pattern (=motif)



Photo



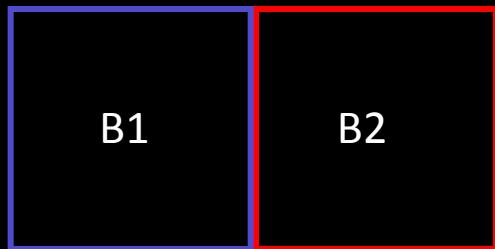
Pattern Repeated



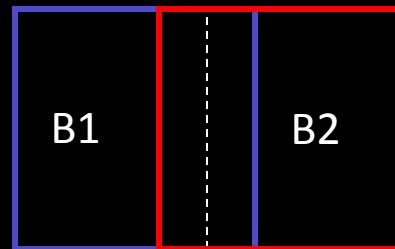
block

Input texture

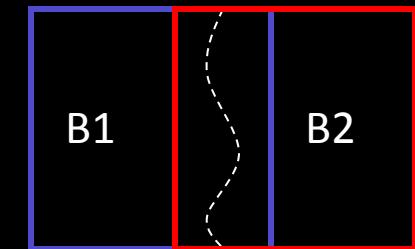
L'approche de Efros et Leung ('99)



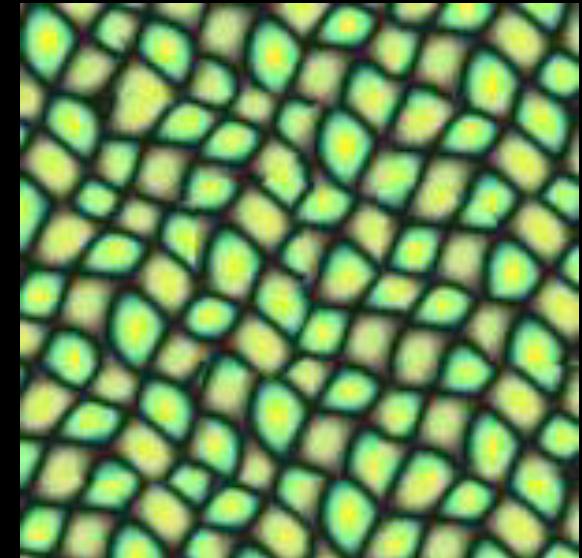
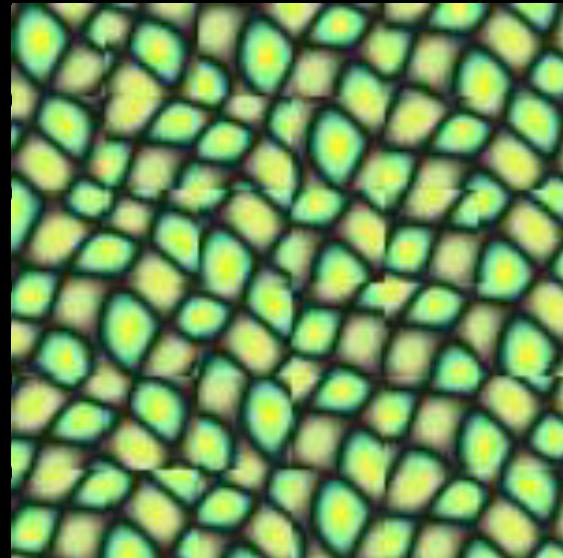
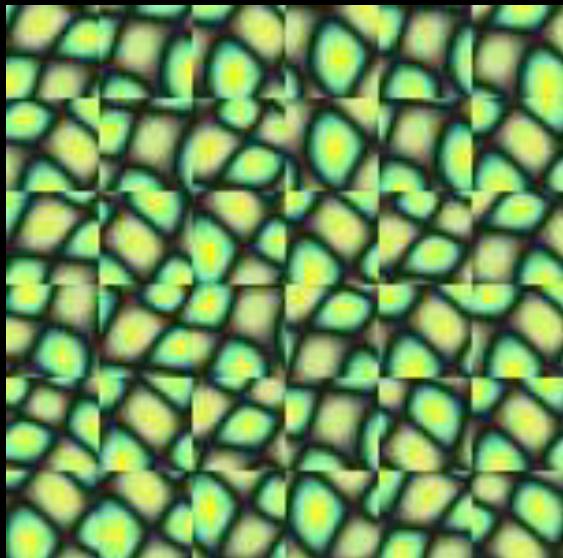
Random placement
of blocks



Neighboring blocks
constrained by overlap

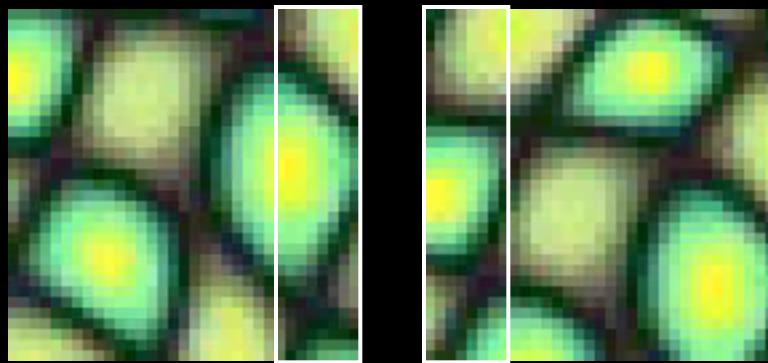


Minimal error
boundary cut

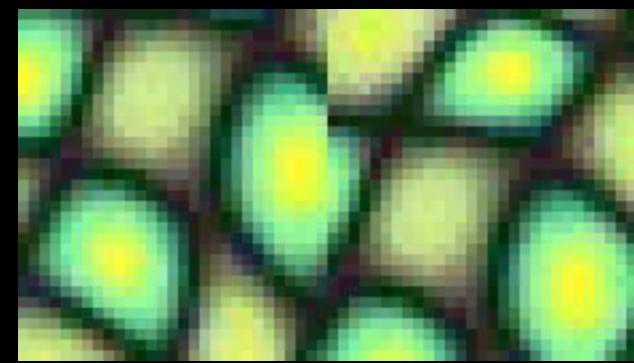


Minimal error boundary

overlapping blocks

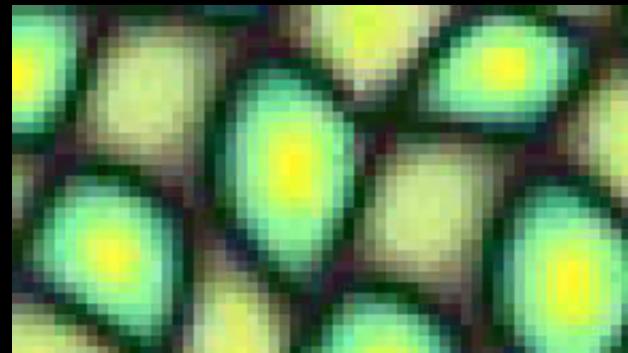


vertical boundary



$$\left[\begin{array}{c|c} \text{overlapping blocks} & \\ \hline \text{---} & \end{array} \right]^2 = \text{overlap error}$$

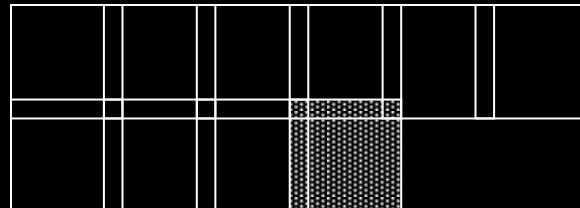
The diagram shows the calculation of 'overlap error'. It consists of two overlapping blocks of a heatmap. A bracket groups the two blocks together, and a square symbol indicates the calculation of the squared difference between them. This result is then equated to a vertical red line with a jagged, stepped pattern, representing the 'overlap error'.



min. error boundary

Algorithm

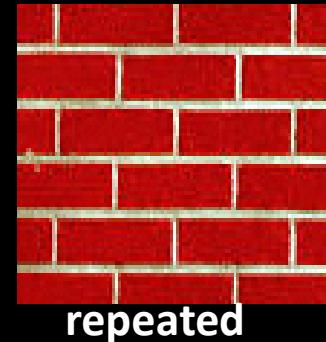
- Pick size of block and size of overlap
- Synthesize blocks in raster order



- Search input texture for block that satisfies overlap constraints (above and left)
 - Easy to optimize using NN search [Liang et.al., '01]
- Paste new block into resulting texture
 - use dynamic programming to compute minimal error boundary cut

Un paramètre important : la taille du voisinage

The size of the neighbourhood window is a parameter that specifies how stochastic the user believes this texture to be



repeated

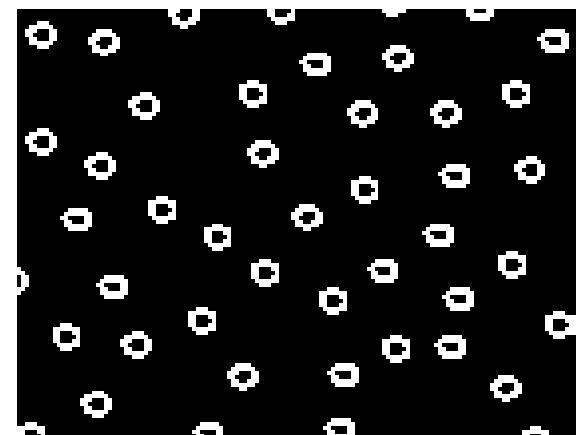
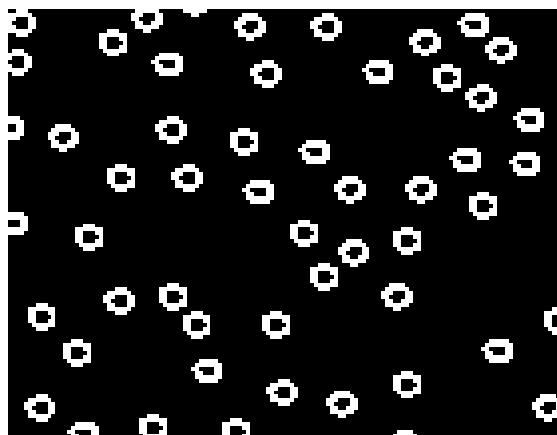
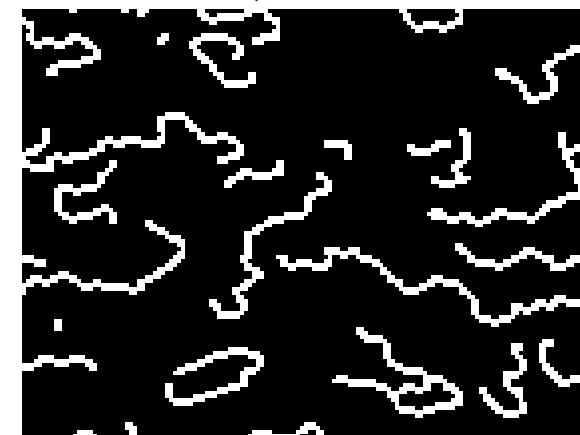
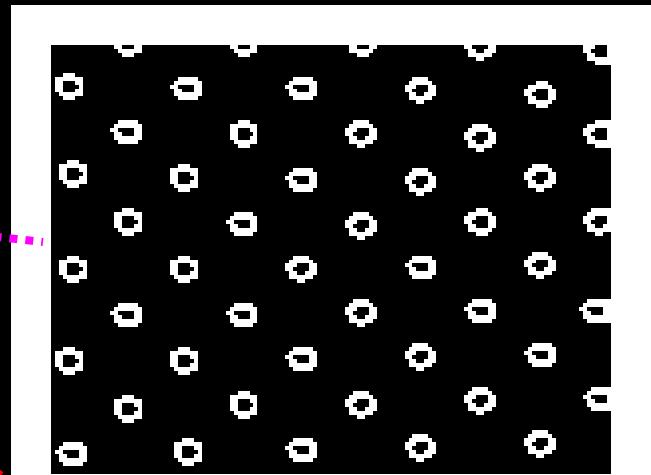
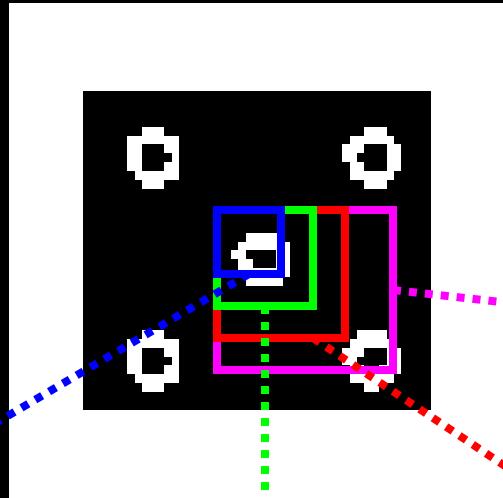
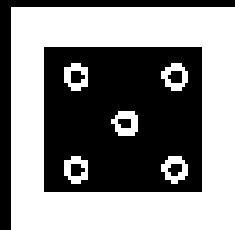


stochastic

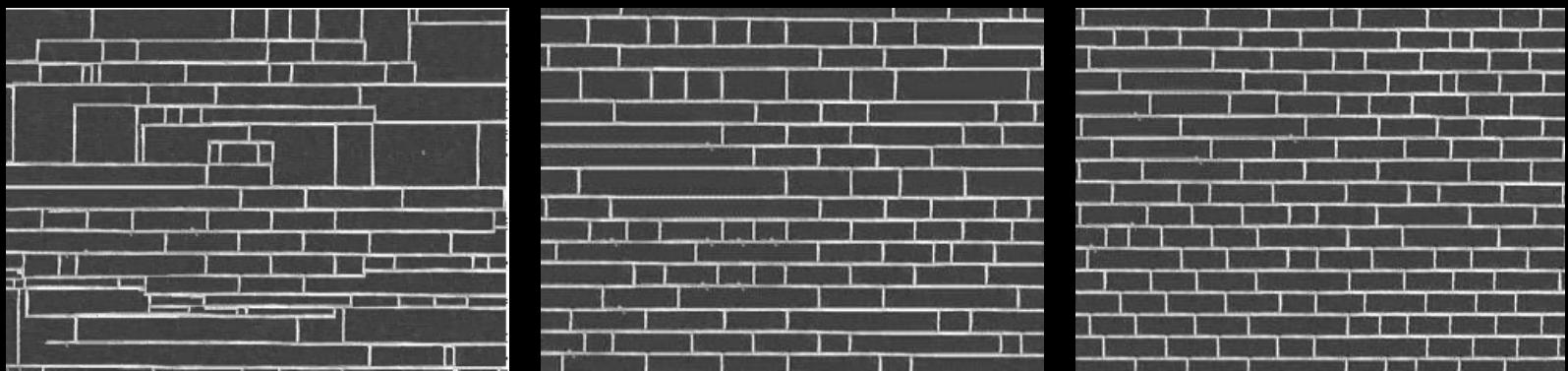
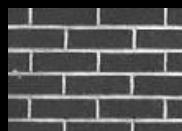
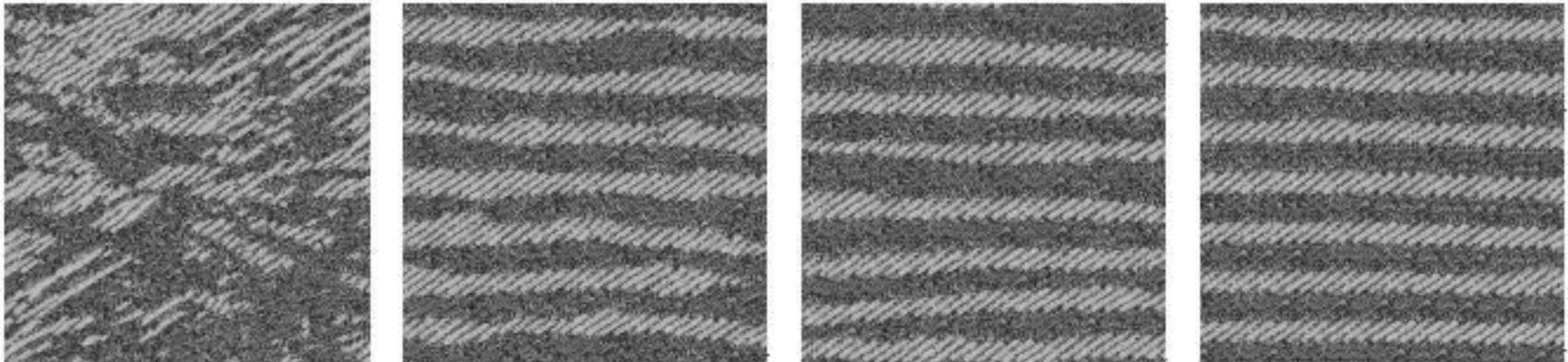


Both?

Un paramètre important : la taille du voisinage



More Synthesis Results

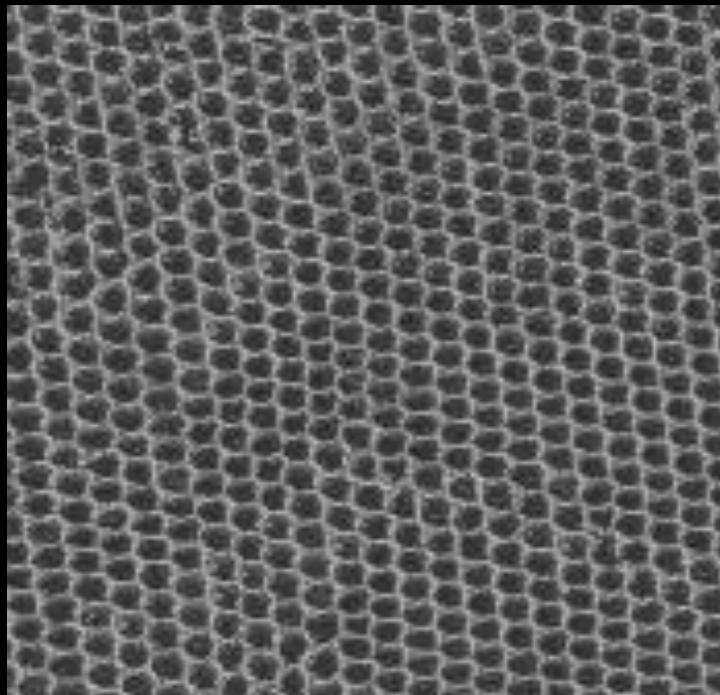
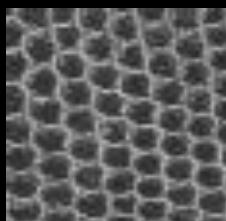


Increasing window size

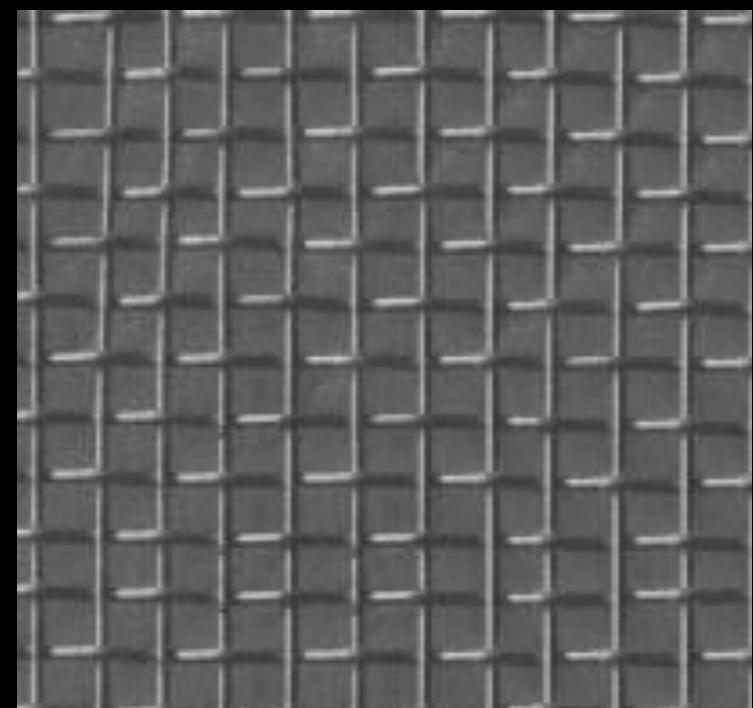
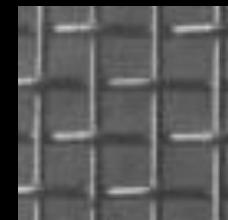


Brodatz Results

reptile skin

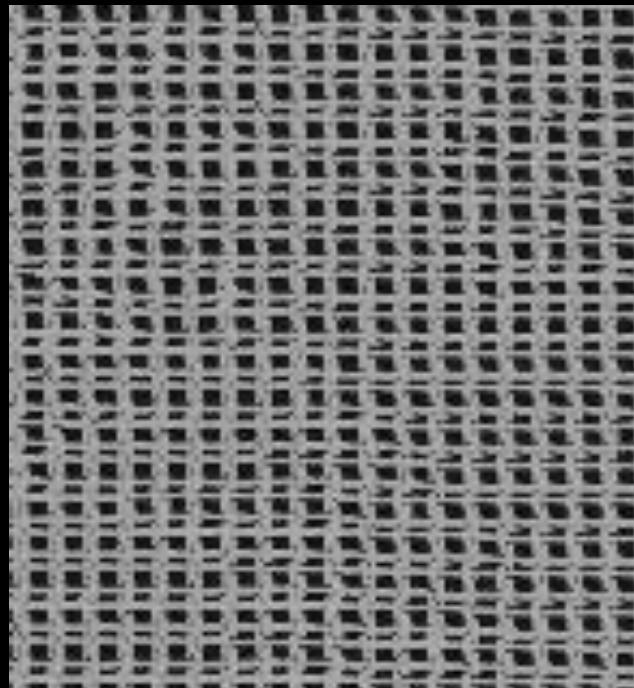
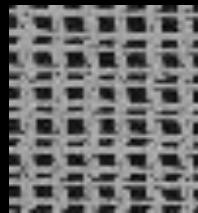


aluminum wire

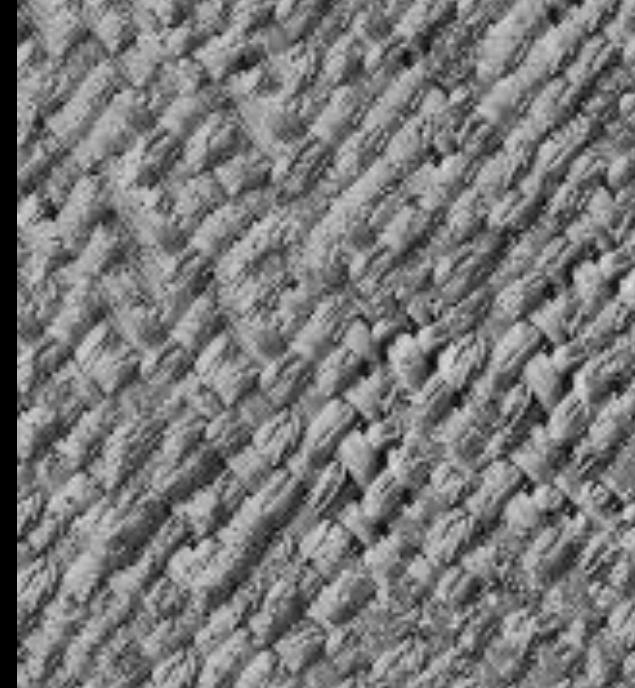
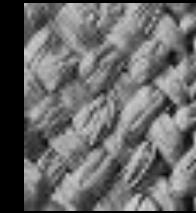


More Brodatz Results

french canvas



rafia weave

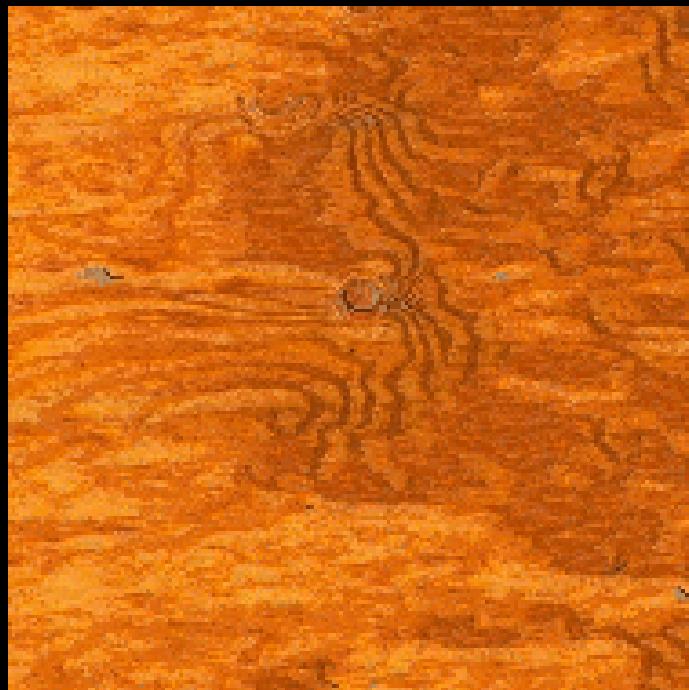


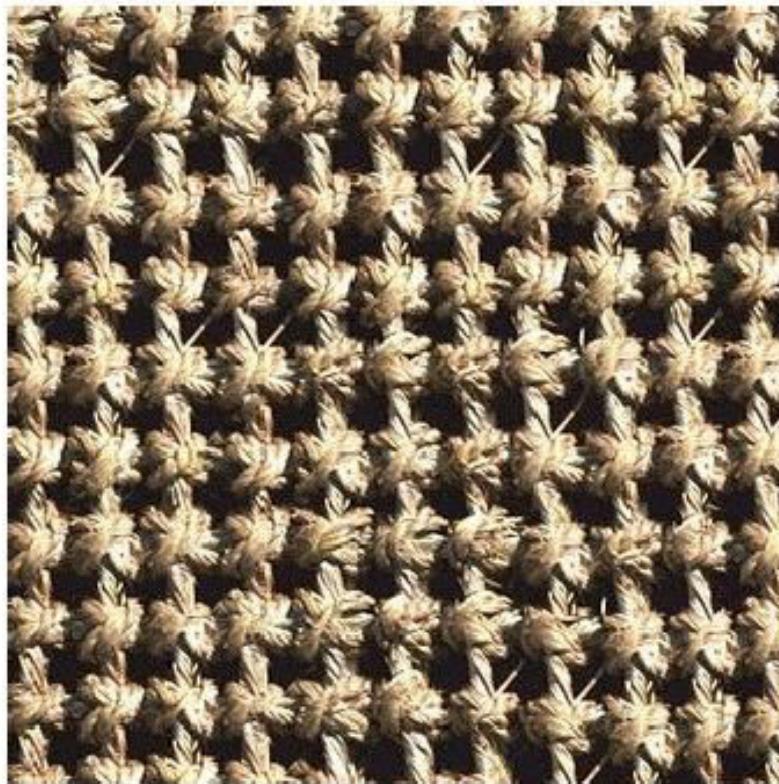
More Results

wood



granite





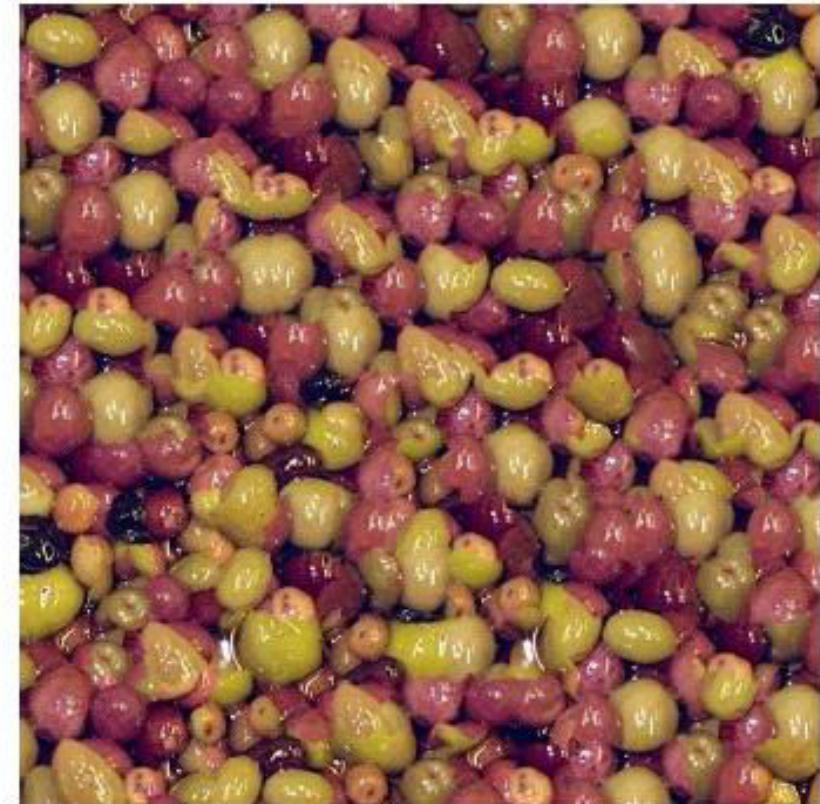
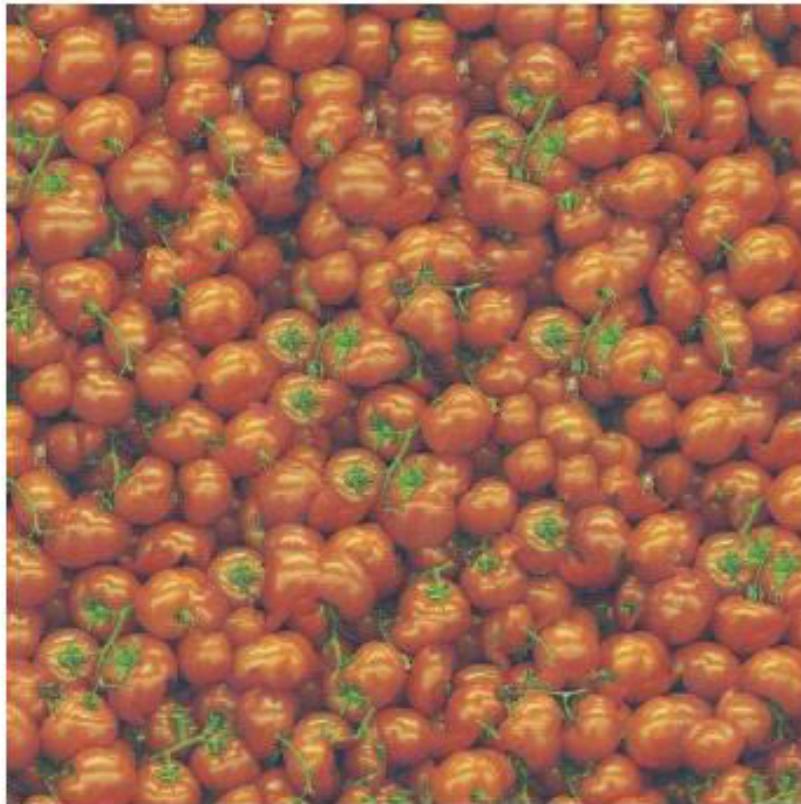
ut it becomes harder to lau
ound itself, at "this daily
ving rooms," as House De
scribed it last fall. He fail
ut he left a ringing question
more years of Monica Lewin
inda Tripp?" That now seen
Political comedian Al Fras
ext phase of the story will

the remanential counoress, at "this daily
ut tuda brears counre Tring rooms," as Heft he fast nd it
ers dat noears ortseas ribed it last nt hest bedian Al. R
economical Houtd ith Al. Heft ars of, as da Lewindailf
dian Al Ths," as Lewing questies last aticarsticall. He
is dian Al last fal counda Lew, at "this dailyears d ily
nd itical counoestscribed it last fall. He fall. Hefft
rs orobheoned it nd it he left a ringing questica Lewin.
licars coecoms," astore years of Monica Lewinow see
a Thas Fring roomne stooniscat nowea re left a roouse
bouestof Mie lefft a Lest fast ringe lāuesticars Hef
nd it rip?" Trihouself, a ring ind it'sonestidit a ring que
astical cois ore years of Moung fall. He ribof Mouse
ore years of tuda Tripp?" That bedian Al Lest fassee yeas
nda Tripp? Political comedian Al et he f w se ring que
political coua re years of the storears ofa s 1 Frat nica L
res Lew se lest a rime 1 He fas quest ringe of, at beou

Figure from Texture Synthesis by Non-parametric Sampling, A. Efros and T.K. Leung,
Proc. Int. Conf. Computer Vision, 1999 copyright 1999, IEEE

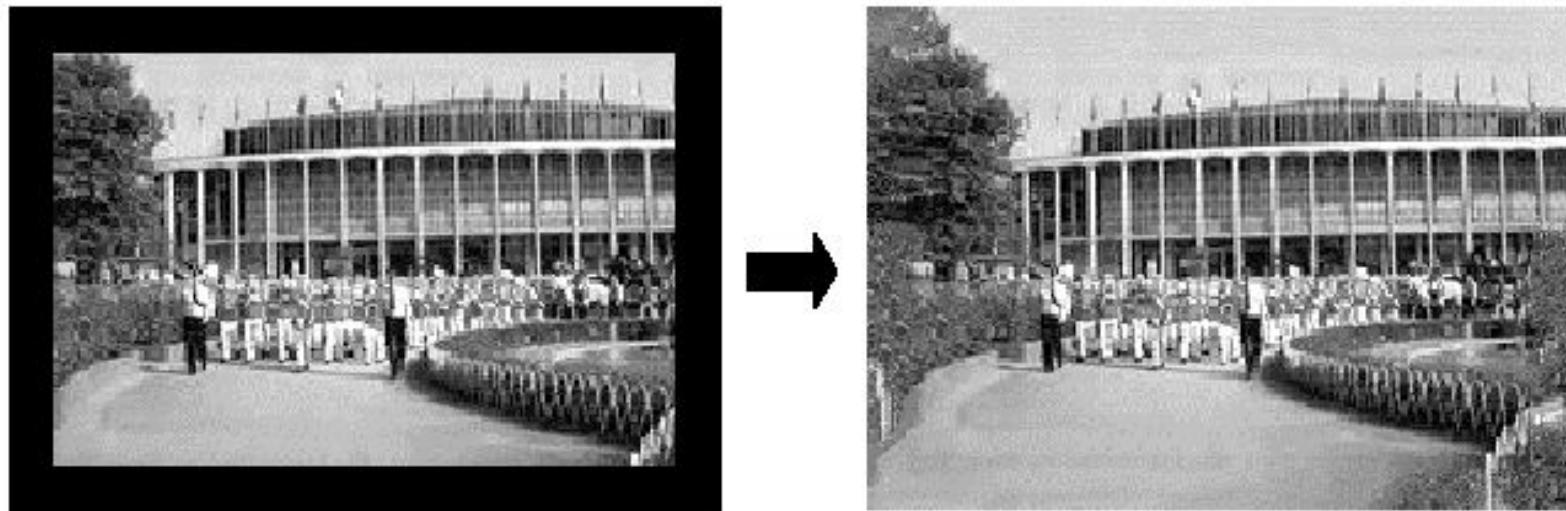


Failures



Exemple d'application de la synthèse de texture

Image Extrapolation

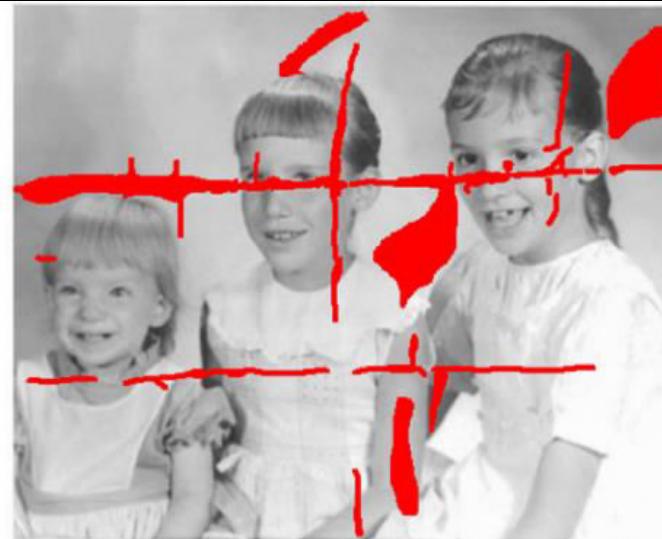


Exemple d'application de la synthèse de texture



- Starting from the initial configuration, we “grow” the texture one pixel at a time

Image inpainting



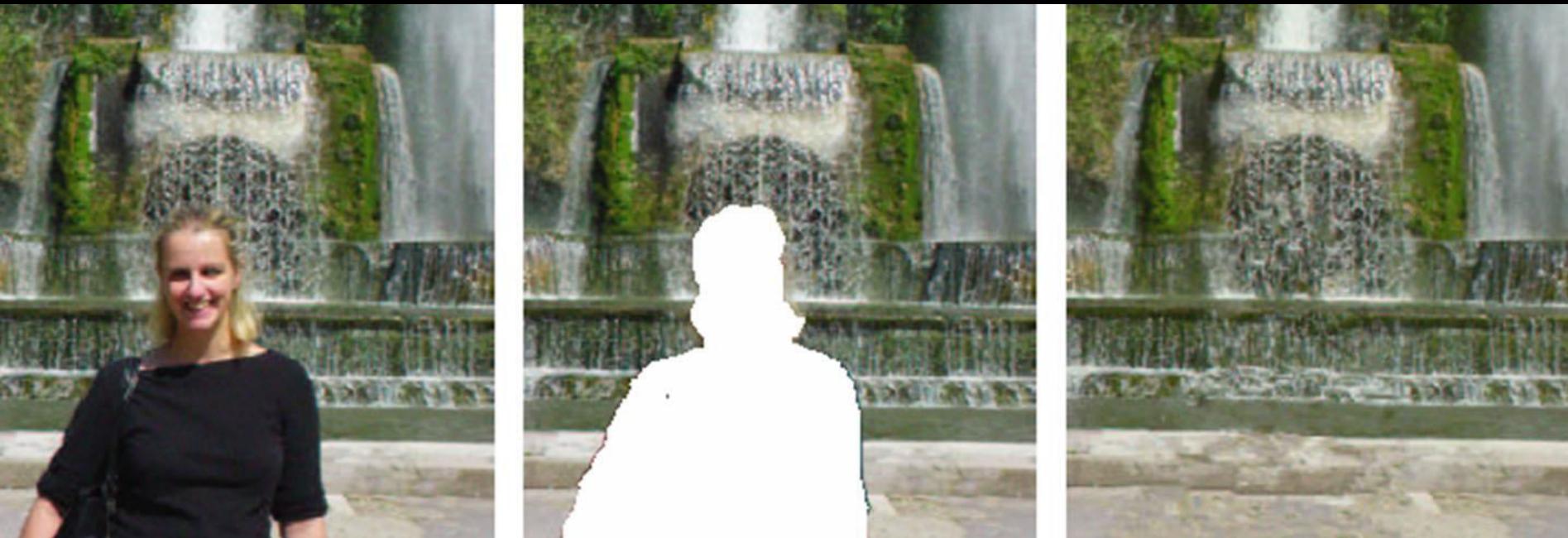
"Image Inpainting", Bertalmío,
Sapiro, Caselles & Ballester,
SIGGRAPH 2000

Image inpainting

“Fragment-based image completion”,
Drori, Cohen-Or, Yeshurun,
SIGGRAPH 2003



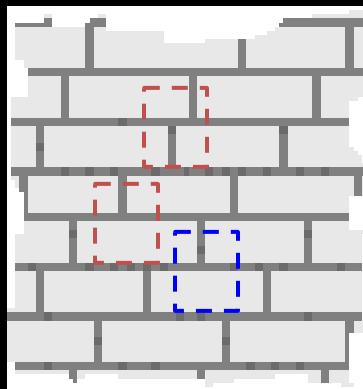
Image inpainting



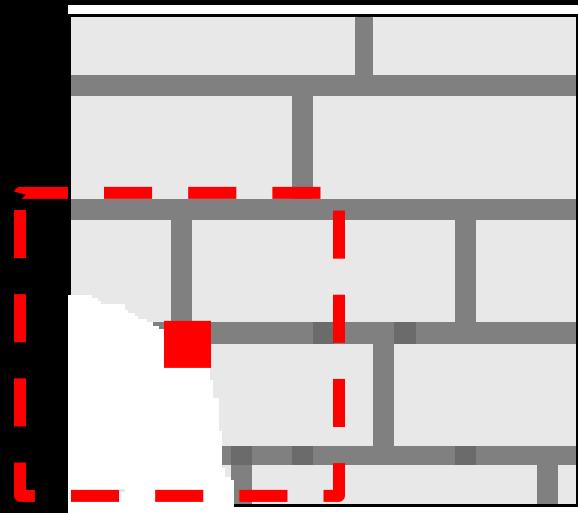
<http://www.cmap.polytechnique.fr/~peyre/cours/x2004signal/>

Synthèse de texture

La méthode par patch de Efros et Leung



*Infinite sample
image*



Generated image

Pour synthétiser chaque pixel : on cherche un voisinage (patch) similaire dans l'image.

On ne pourra pas trouver de voisinage/patch exact, mais on prendra le patch le plus proche au sens d'un critère

Un algorithme pour l'inpainting

- Lister les pixels qui font partie du bord du masque
- Calculer la priorité de chaque pixel du bord du masque
- A partir du pixel ayant la priorité la plus haute (on l'appelle le target pixel), extraire un patch (fenêtre carrée) centré en ce pixel. On appelle le target patch.
- Rechercher dans l'image le patch le plus ressemblant au target patch.
- Le copier à la place du target pixel.
- Mettre à jour le masque en prenant en compte le patch collé.
- Répéter jusqu'à ce que le masque contienne 0 pixel.

Texture : résumé

- Définition difficile
- De nombreux outils pour les caractériser :
 - Stat du 1^{er} ordre
 - Matrice de cooccurrence (Haralick), LBP...
 - Filtres de Gabor...
- Utilisation en segmentation, en reco de formes, en CBIR, et dans bien d'autres domaines...

Références

- *Master ASE & Master Informatique – RdF: Reconnaissance de Formes*, IEEA
- http://www.cs.rpi.edu/~cutler/classes/advancedgraphics/S12/lectures/22_texture_synthesis.pdf
- Texture, Selected Topics in Computational Vision, Moran Hirsh CS 202-1-5091, Computer Science Department, BGU
- Texture Synthesis by Non-parametric Sampling, A. Efros and T.K. Leung, Proc. Int. Conf. Computer Vision, 1999 copyright 1999, IEEE

Ayé, le cours est fini...

