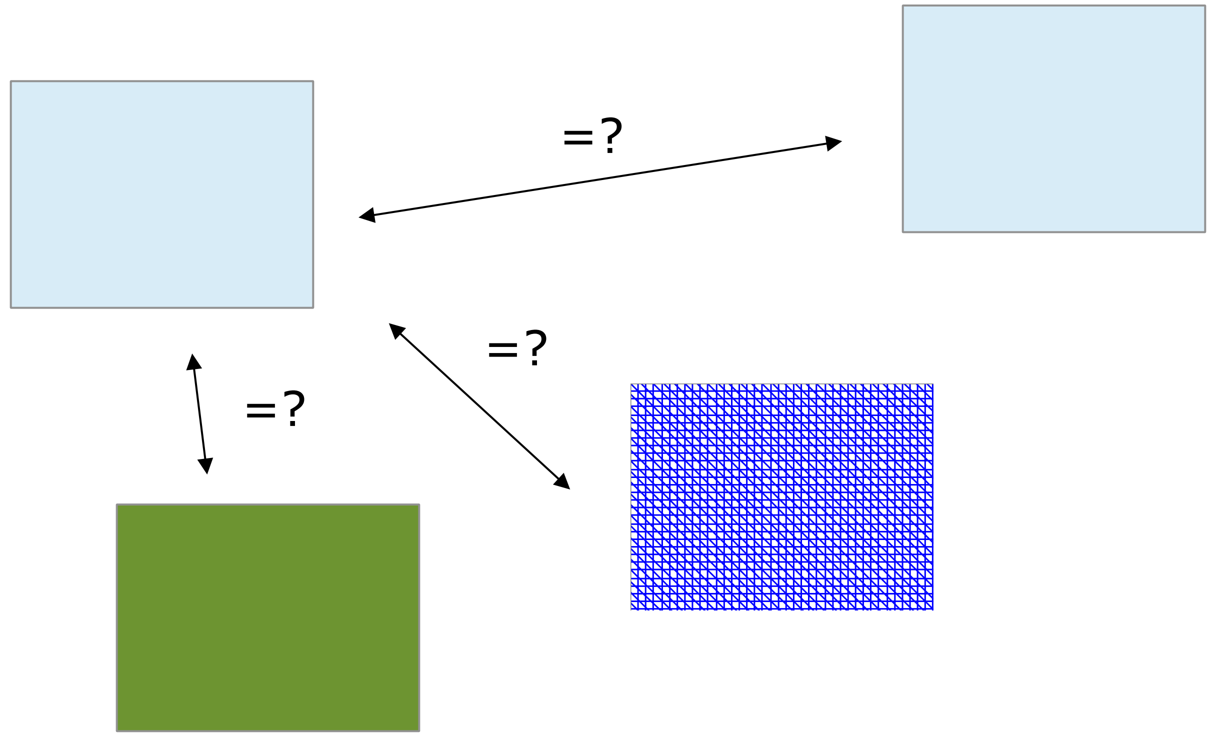
Chapitre 3

Description de régions d'intérêt dans les séquences vidéo:Descripteurs de base

Pourquoi décrire une région ?

- Pour trouver des correspondances entre les objets



Pourquoi décrire une région ?

- Pour trouver des correspondances entre les objets
  - En suivi, pour trouver la trajectoire d'un objet;
  - En reconnaissance d'objets, pour trouver un objet ayant une apparence donnée;
  - En reconnaissance d'actions/de poses pour identifier des gestes.

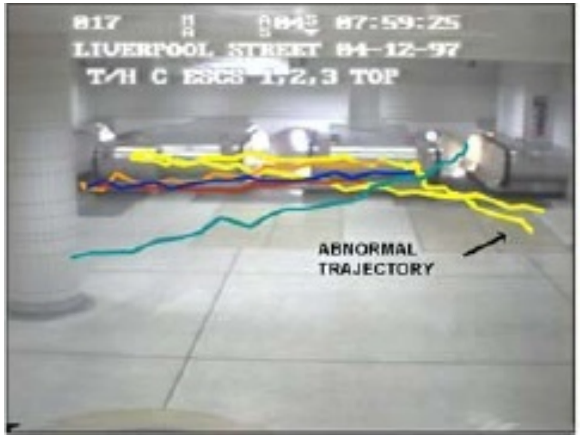
- Quel est le lien avec le chapitre précédent?
  - Les segments obtenus sont décrits pour réaliser un des trois objectifs ci-dessus.

Pourquoi décrire une région ?

- Correspondance en suivi d'objets
  - Objectif: Déterminer la trajectoire ou l'identité d'un objet.
  - Exemple:
    - Trouve les objets en mouvement par soustraction d'arrière-plan.
    - Utilise le recouvrement des objets d'une image à l'autre pour faire une correspondance temporelle.
    - Validation par l'apparence des objets.

Pourquoi décrire une région ?

- Suivi d'objets (suite)
  - Trouve le centroïde des objets.
  - La trajectoire est formée en reliant les centroïdes successifs d'un même objet.




Pourquoi décrire une région ?

- Suivi d'objets
  - Exemple:
    - Suivi des participants d'une réunion.
    - Recherche de la tête (Une ellipse de couleur peau déterminée par apprentissage).
    - Recherche exhaustive dans les régions ayant la bonne couleur pour trouver les paramètres de l'ellipse.
    - Couleurs modélisées par histogramme.

Pourquoi décrire une région ?

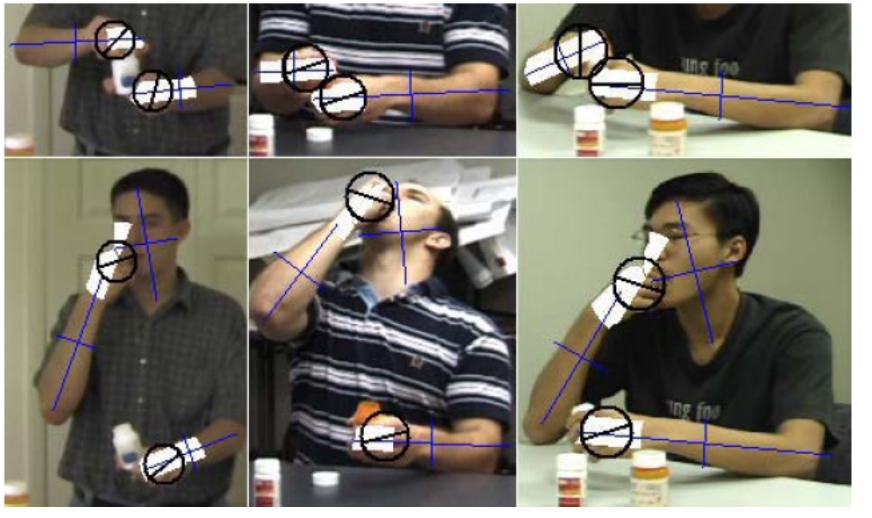
- Suivi d'objets (suite)
  - Trouve la prochaine position de l'ellipse de la tête par méthode statistique (filtre de particules). *Pas de soustraction d'arrière-plan.*

Pourquoi décrire une région ?

- Détection de prise de médicaments
  - Objectif: Déterminer si une personne a pris ses médicaments.
  - Exemple:
    - Détection des régions de peau.
    - Modélisation des régions par des ellipses. La dimension permet de différencier les mains du visage.
    - Détection des yeux et de la bouche.

Pourquoi décrire une région ?

- Détection de prise de médicaments (suite)
  - Détection des bouteilles de médicaments par extraction d'arêtes et par leur forme.
  - Détection de la trajectoire des mains et des bouteilles.



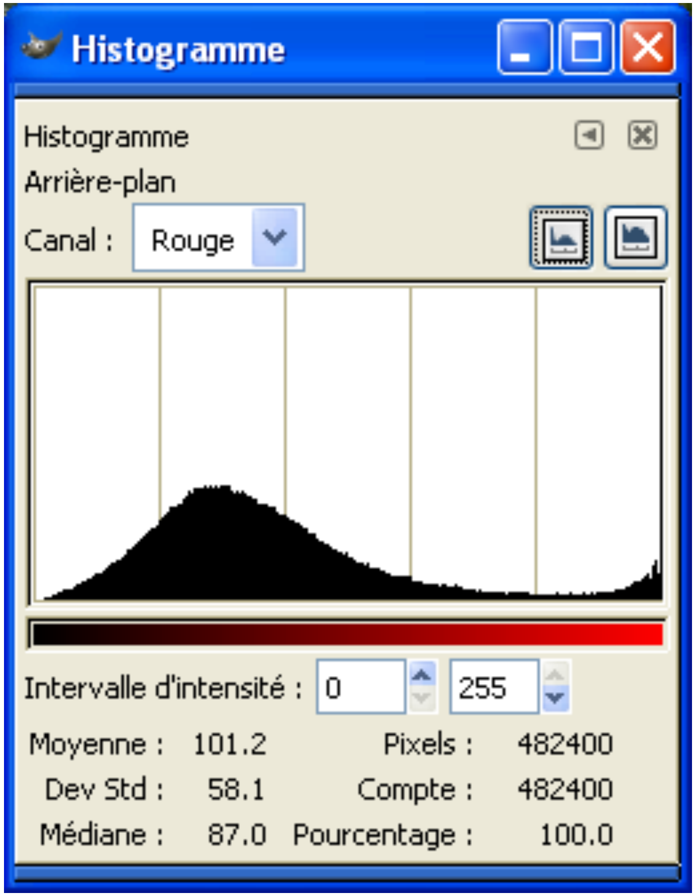
Plan du chapitre

Modèles d'apparence

- Couleur
  - Histogramme
  - RGB vs HSV
  - Comparaison d'histogrammes
- Texture
  - Arêtes
  - Banque de filtres
  - Matrice de co-occurrences
- Forme
  - Moments

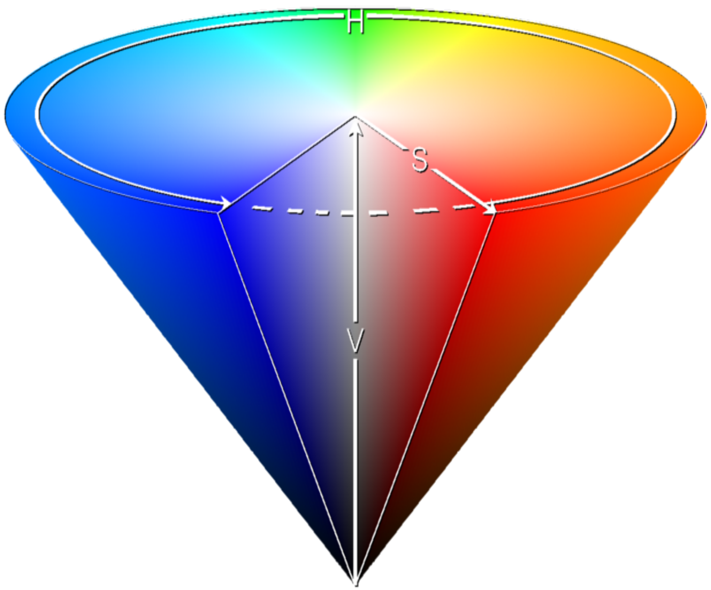
Modèle d'apparence: couleur

- Modélisation de la couleur:
  - Typiquement par un histogramme.
  - Calcule le nombre d'occurrences de chaque couleur (ou classe) dans l'image.
  - Trois histogrammes pour chaque canal de couleur (RGB, HSV, etc.), peut être concaténés en un histogramme.
  - *Exemple MATLAB: CalcHisto.m*



Modèle d'apparence: couleur

- HSV vs RGB



Source: Wikipedia

$$H \in [0, 360)$$
$$S, V, R, G, B \in [0, 1]$$

From RGB to HSV:

Let *MAX* equal the maximum of the (*R*, *G*, *B*) values, and *MIN* equal the minimum of those values.

$$H = \begin{cases} \text{undefined}, & \text{if } MAX = MIN \\ 60 \times \frac{G-B}{MAX-MIN} + 0, & \text{if } MAX = R \\ & \text{and } G \geq B \\ 60 \times \frac{G-B}{MAX-MIN} + 360, & \text{if } MAX = R \\ & \text{and } G < B \\ 60 \times \frac{B-R}{MAX-MIN} + 120, & \text{if } MAX = G \\ 60 \times \frac{R-G}{MAX-MIN} + 240, & \text{if } MAX = B \end{cases}$$

$$S = \begin{cases} 0, & \text{if } MAX = 0 \\ 1 - \frac{MIN}{MAX}, & \text{otherwise} \end{cases}$$

$$V = MAX$$

Modèle d'apparence: couleur

- Caractéristiques des histogrammes:
  - Relativement invariant:
    - Translations;
    - Rotations dans l'axe de l'image;
    - Changements d'échelle;
    - Occlusions partielles.
- Comment les comparer ?
  - Intersection
  - Distance

Modèle d'apparence: couleur

- Intersection d'histogrammes
  - Vérifie la qualité de l'inclusion de l'histogramme modèle h_M dans l'histogramme image h_i.
  - Il y a correspondance si tous (ou presque) les pixels des *K* classes de l'histogramme h_M sont inclus dans les *K* classes de l'histogramme h_i.

$$\text{Intersection}(h_i, h_m) = \sum_{j=1}^{K} \min(h_i[j], h_m[j])$$

$$\text{Correspondance}(h_i, h_M) = \frac{\sum_{j=1}^{K} \min(h_i[j], h_m[j])}{\sum_{j=1}^{K} h_M[j]}$$

Modèle d'apparence: couleur

- Distance entre deux histogrammes
  - Vérifie la similitude de deux histogrammes.
  - Considère un histogramme comme un vecteur.
  - Attention, calcule le nombre de différences, et non pas leurs importances ou leurs distributions…

  - City block (Norme L1)

$$D_1(h_i, h_M) = \sum_{j=1}^{K} \left| h_i[j] - h_M[j] \right|$$

  - Euclidienne (Norme L2)

$$D_2(h_i, h_M) = \sqrt{\sum_{j=1}^{K} (h_i[j] - h_M[j])^2}$$

Modèle d'apparence: couleur

- Distance entre deux histogrammes
  - Considère un histogramme comme une distribution discrète de probabilités.
  - Attention, calcule le nombre de différences, et non pas leurs importances ou leur distributions…
  - Bhattacharyya

$$D(h_i, h_M) = -\log \sum_{j=1}^{K} \sqrt{P(h_i[j])P(h_M[j])}$$

  - *Exemple MATLAB: dbat.m*

Modèle d'apparence: couleur

- Distance entre deux histogrammes
  - Considère un histogramme comme un vecteur.
  - Prix minimum pour passer d'une distribution à une autre.
  - Minimum difference of pair assignments (MDPA)

$$D(h_i, h_M) = \sum_{j=0}^{K-1} \left| \sum_{k=0}^{j} (h_i[k] - h_M[k]) \right|$$

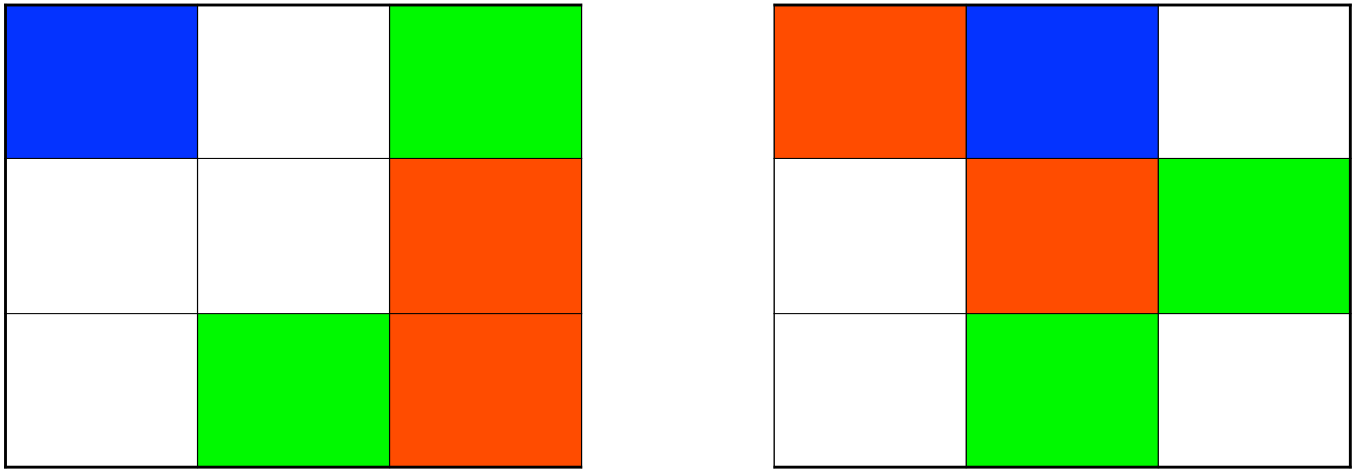
  - *Exemple MATLAB: MDPA.m*

Modèle d'apparence: texture

- Histogramme d’orientation des arêtes
- Densité des arêtes
- Banque de filtres
  - Filtres de Gabor
- Matrice de co-occurrences

Modèle d'apparence: texture

- Texture: Pourquoi ?
  - La modélisation par histogramme n'est pas précise, car la position des couleurs n'est pas prise en compte.



Modèle d'apparence: texture

- Texture: Pourquoi ?
  - Modélisation par texture est plus précise, car elle prend en compte la position relative des couleurs directement ou indirectement.
  - Par contre, plus long à calculer.

Modèle d'apparence: texture

- Texture basé sur les arêtes
  - Le gradient est calculé par une convolution entre un filtre et l'image.
  - Par exemple: Filtre de Sobel
  - Ensuite, on seuille pour obtenir les arêtes.
  - Des noyaux utilisés couramment sont ceux de Sobel:

$$G_x = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} * I \qquad G_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} * I$$

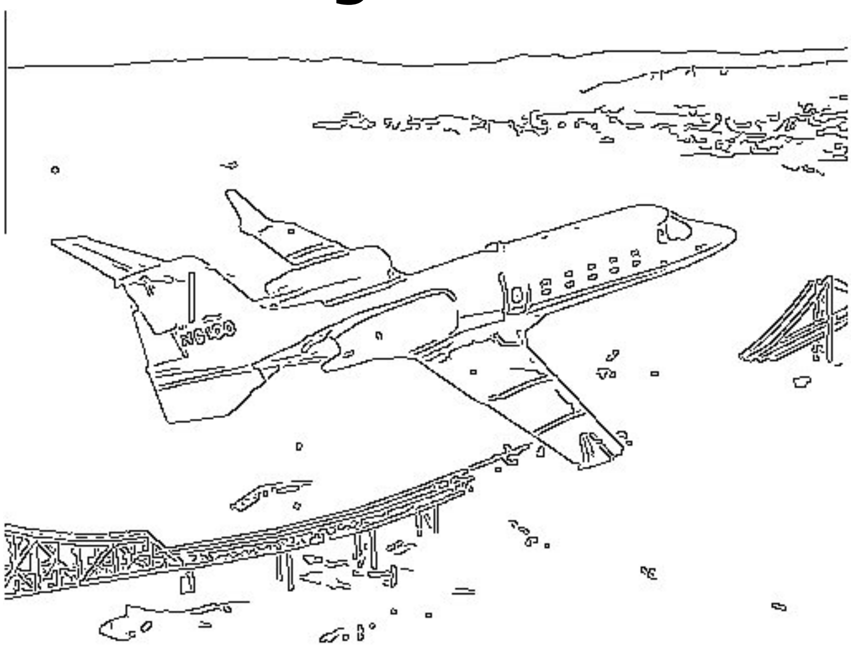
Modèle d'apparence: texture

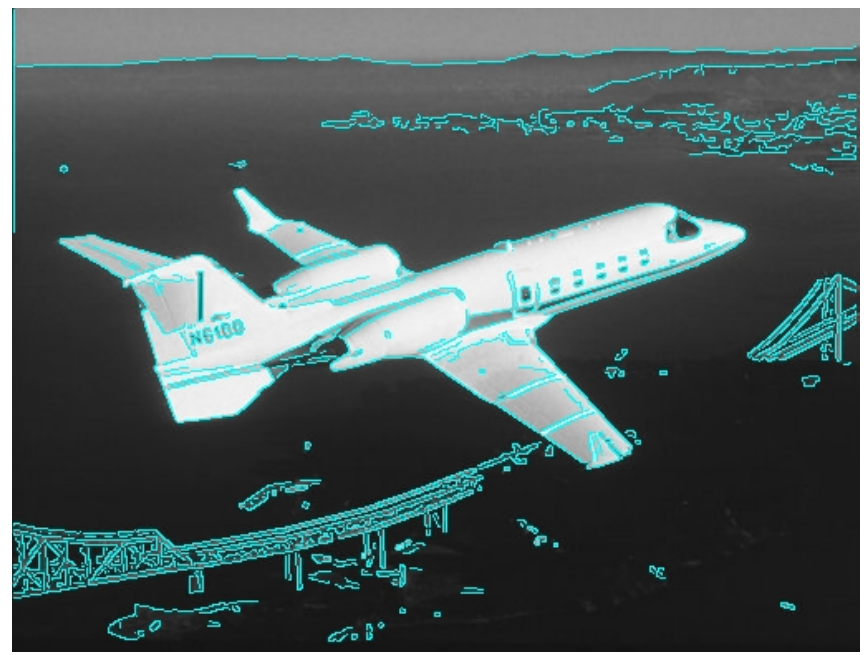
- La convolution en 2D est définie pour G_x comme:

$$G_x(x,y) = \sum_{i=-1}^{1} \sum_{j=-1}^{1} Sobel_x(i,j) I(x-i, y-j)$$

Modèle d'apparence: texture

- Exemple d'arêtes extraites après un seuillage.





  - *Exemple MATLAB: Edge2.m*

Modèle d'apparence: texture

- Une première méthode de modélisation de texture utilisant les arêtes est l'histogramme de l'orientations des gradients.

- Cet histogramme est construit par la fréquence de chaque angle quantifié, l'angle étant calculé par

$$\theta = \arctan\left(\frac{G_y}{G_x}\right)$$

Modèle d'apparence: texture

- La force (norme) des gradients est utilisée pour pondérer les fréquences:

$$F_G = \sqrt{G_x^2 + G_y^2}$$

- Donc, un angle qui correspond à un gradient ayant une force plus élevée aura plus d'importance dans le calcul du nombre d'occurrences.
- Exemple: Si F_G=20, on compte 2 occurrences, si F_G=50, on compte 5 occurrences, etc. Donc, pas 1 angle, 1 vote.

Modèle d'apparence: texture

- Deuxième façon: Densité des arêtes
  - Nombre d'arêtes *p* dans une région *R* de *N* pixels (occupation):

$$F_{occupation} = \left| \left\{ \frac{p \,|\, Mag(p) \geq T}{N} \right\} \right|$$

  - Densité + orientation: histogrammes normalisés des magnitudes et des orientations des arêtes de la région R de N pixels.

$$F_{densité/orientation} = (H_{mag}(R), H_{ori}(R))$$

Modèle d'apparence: texture

- Banque de filtres
  - Barres et taches:
    - Si une région d'image a une distribution spatiale semblable au filtre, la réponse de la convolution avec le filtre sera grande.
    - Typiquement, on utilise plusieurs filtres dont certains détectent des barres à différentes échelles et orientations et certains détectent des taches.
    - On peut obtenir ces filtres en combinant des filtres gaussiens.
      - Tache: Somme de 3 gaussiennes symétriques avec des écart-types différents.

Modèle d'apparence: texture

- Banque de filtres
  - Barres et taches:

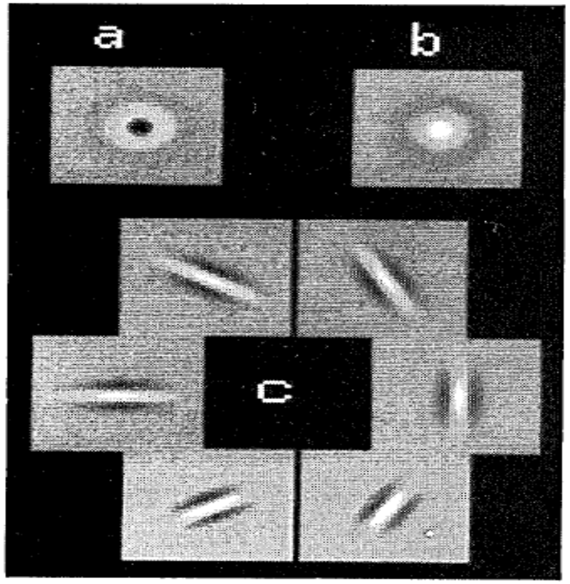
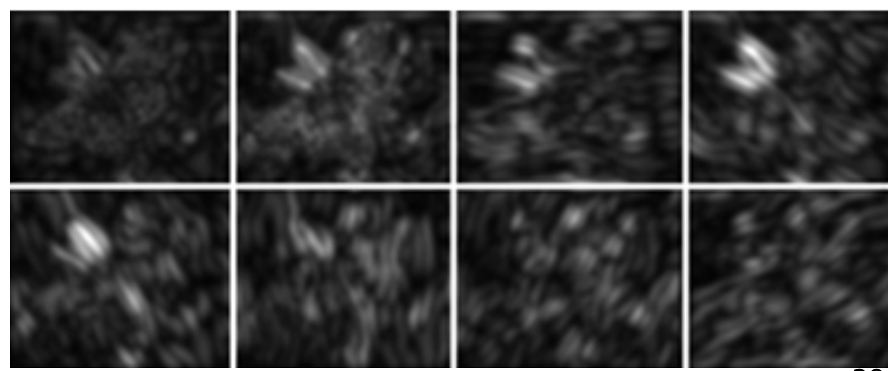


Fig. 2. Point-spread functions of some of the filters used in our simulation. The filters were designed after Young[25] by summing Gaussian functions $G(x_0, y_0, \sigma_x, \sigma_y) \doteq 1/2\pi\sigma_x\sigma_y \exp\{-(x - x_0/\sigma_x)^2 + (y - y_0/\sigma_y)^2]\}$ and have zero-mean value. a, Linear combination of three circular concentric Gaussian functions, $\mathrm{DOG2}(\sigma) \doteq a \cdot G(0, 0, \sigma_i, \sigma_i) + b \cdot G(0, 0, \sigma, \sigma) + c \cdot G(0, 0, \sigma_o, \sigma_o)$ with variance $\sigma_i{:}\sigma{:}\sigma_o$ in a ratio of 0.62:1:1.6 and $a{:}b{:}c$ in a ratio of 1:−2:1. b, Linear combination of two circular concentric Gaussian functions, $\mathrm{DOG1}(\sigma) \doteq a \cdot G(0, 0, \sigma_i, \sigma_i) + b \cdot G(0, 0, \sigma_o, \sigma_o)$, with variance $\sigma_i{:}\sigma{:}\sigma_o$ in a ratio of 0.71:1:1.14 and coefficients $a{:}b$ in a ratio of 1:−1. c, Linear combination of three offset identical Gaussian functions $\mathrm{DOOG2}(\sigma, r, \theta) \doteq a \cdot G(0, y_a, \sigma_x, \sigma_y) + b \cdot G(0, y_b, \sigma_x, \sigma_y) + c \cdot G(0, y_c, \sigma_x, \sigma_y)$. Variances are $\sigma_y = \sigma, \sigma_x = r \cdot \sigma$, offsets are $y_a = -y_c = \sigma, y_b = 0$, and coefficients are $a{:}b{:}c$ in a ratio of −1:2:−1 for the filter with an axis of symmetry along the x direction ($\theta = 0$). The other DOOG2( ) filters are obtained by rotation about the center of the middle Gaussian. The scaling coefficients $a_{\mathrm{DOG1}}{:}a_{\mathrm{DOG2}}{:}a_{\mathrm{DOOG2}}$ were in a ratio of 3:4.15:2, which was designed to equalize the dynamic range of the respective responses.

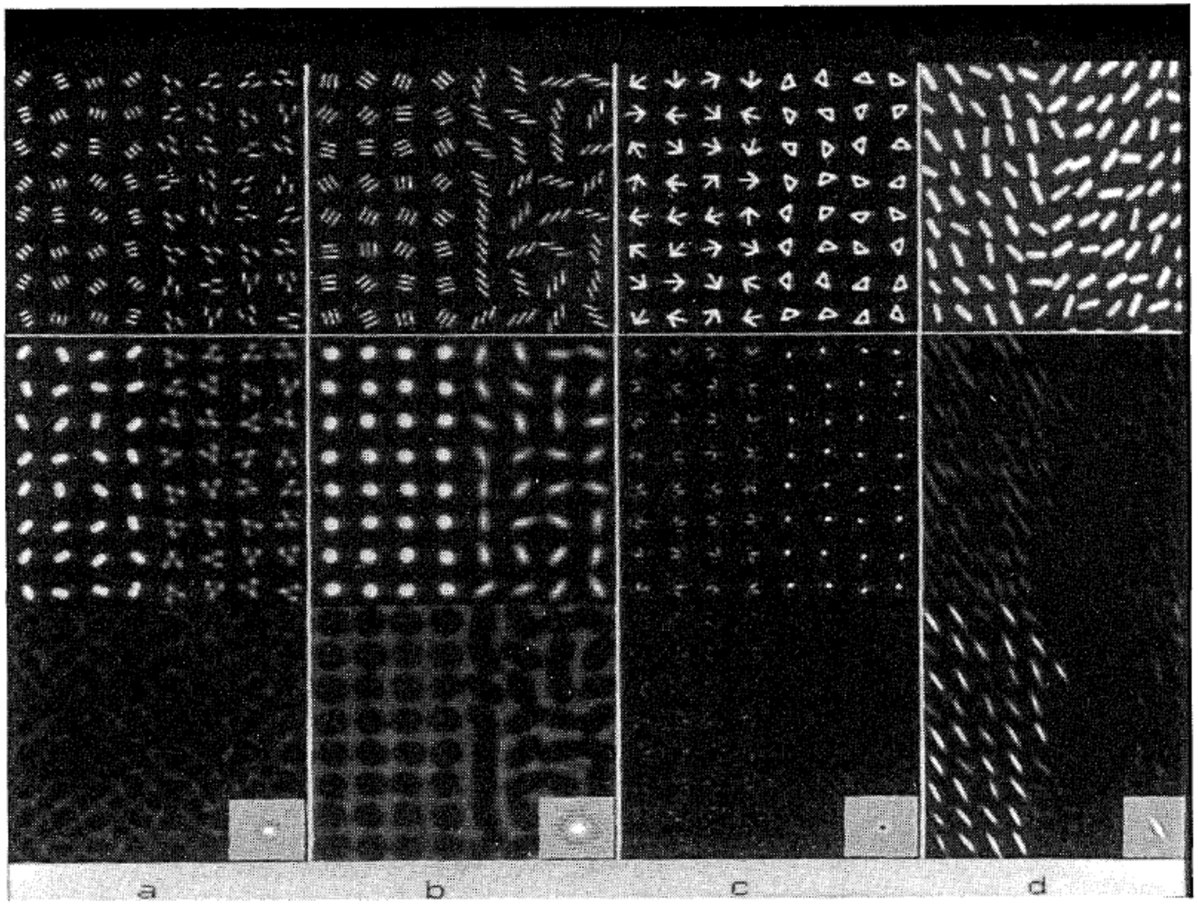
Modèle d'apparence: texture

- Banque de filtres
  - Barres et taches:
  - *Exemple MATLAB: tache.m*



Modèle d'apparence: texture

- Banque de filtres
  - Barres et taches:

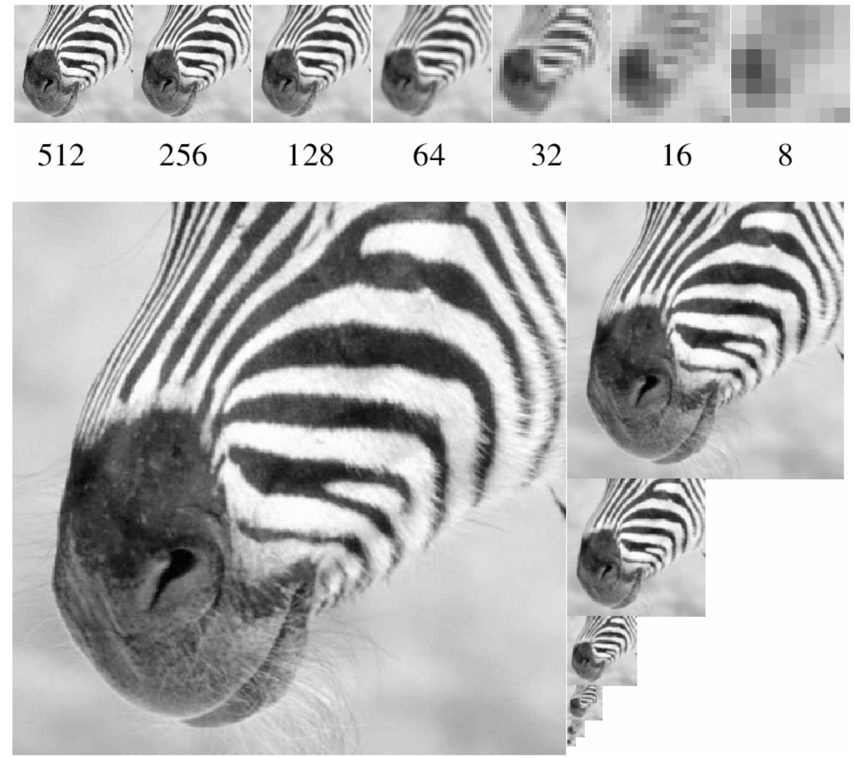


Modèle d'apparence: texture

- Banque de filtres
  - Pyramide gaussienne:
    - Analyse locale de la fréquence spatiale des intensités.
    - Si on filtre une image avec une gaussienne avec un écart-type petit, tous les fréquences spatiales élevées seront éliminés.
    - Si on filtre une image avec une gaussienne avec un écart-type grand, on obtient à la limite la valeur moyenne de l'image.

Modèle d'apparence: texture

- Banque de filtres
  - Pyramide gaussienne:
    - Filtrer l'image avec une gaussienne.
    - Échantillonner l'image résultante pour la réduire. Correspond à changer l'écart-type de la gaussienne.
    - Filtrer et échantillonner jusqu'à une taille prédéfinie.
    - *<u>Exemple MATLAB:</u>* *FiltrageGaussienne.m*

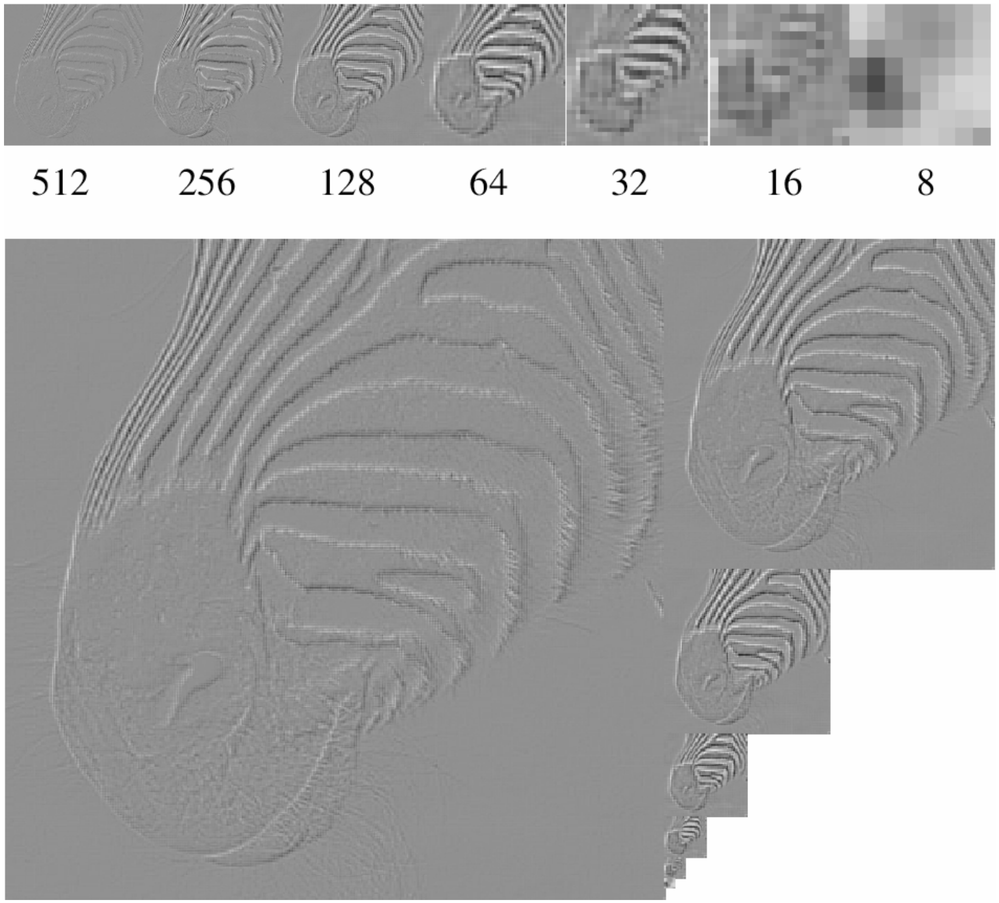


Modèle d'apparence: texture

- Banque de filtres
  - Pyramide laplacienne:
    - L'échelle grossière de la pyramide gaussienne est une prédiction d'une échelle plus fine.
    - La pyramide laplacienne limite les redondances en enregistrant seulement les erreurs de prédiction.
    - Si on agrandit l'échelle grossière (quadruple la grosseur des pixels) pour obtenir l'échelle plus fine, on peut enregistrer seulement l'erreur de reconstruction.

Modèle d'apparence: texture

- Banque de filtres
  - Pyramide laplacienne:
    - Chaque échelle de la pyramide laplacienne correspond à la réponse d'un filtre passe-bande.



Modèle d'apparence: texture

- Banque de filtres
  - Filtres de Gabor
    - Transformée de Fourier: Produit scalaire entre une sinusoïde (cosinus+sinus) et une fonction.
    - Filtres de Gabor: Transformée de Fourier locale. Chaque filtre de Gabor est une sinusoïde multipliée par une gaussienne.
    - Les filtres de Gabor sont en paire. Un filtre détecte les composantes symétriques, l'autre détecte les composantes antisymétriques.

Modèle d'apparence: texture

- Banque de filtres
  - Filtres de Gabor

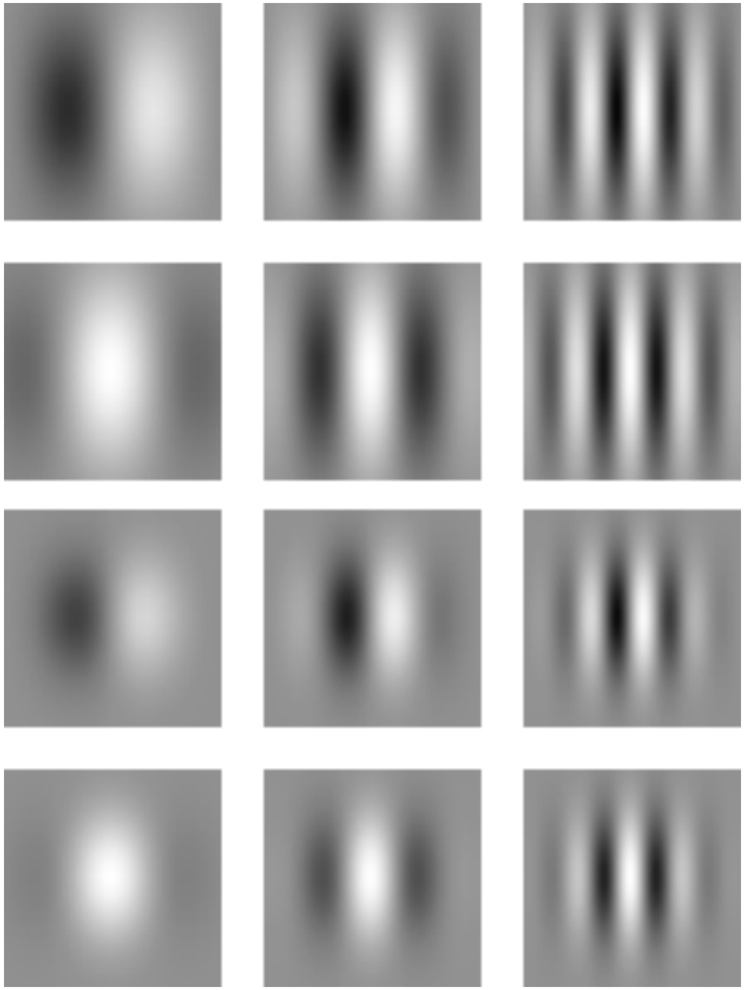
$$G_{symétrique}(x,y) = \cos(k_x x + k_y y)e^{-\left\{\frac{x^2+y^2}{2\sigma^2}\right\}}$$

$$G_{antisymétrique}(x,y) = \sin(k_x x + k_y y)e^{-\left\{\frac{x^2+y^2}{2\sigma^2}\right\}}$$

(k_x, k_y) donne la fréquence spatiale de réponse, et σ est l'échelle (étendu du filtre).

Modèle d'apparence: texture

- Banque de filtres
  - Filtres de Gabor
  - *Exemple MATLAB: FiltreGabor.m*



Modèle d'apparence: texture

- Banque de filtres
  - Comparaison:
    - Par des statistiques sur les réponses:
      - Moyenne;
      - Variance;
      - Écart-type;
      - Covariance;
      - etc.

Modèle d'apparence: texture

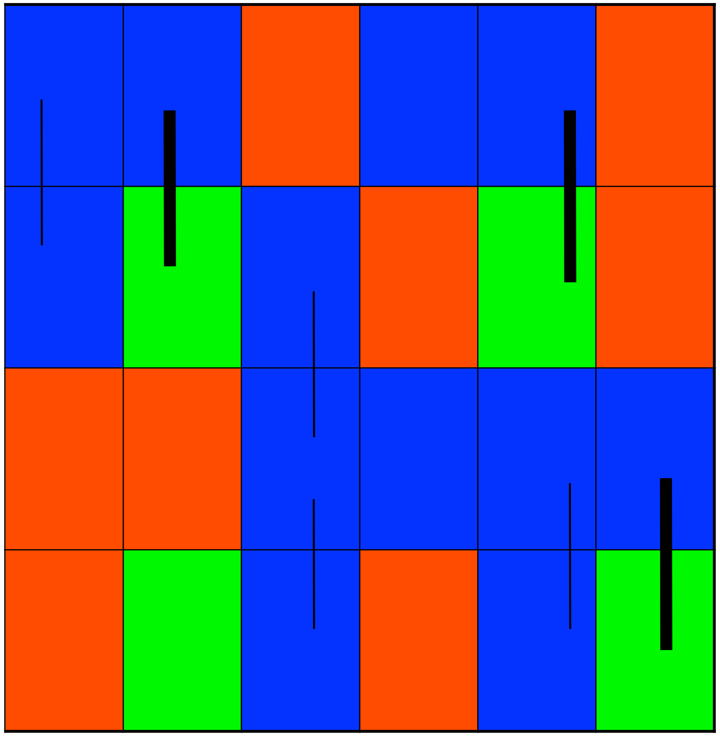
- Matrice de co-occurrences
  - Matrice qui enregistre le nombre fois que deux couleurs (valeurs d'intensité) similaires ont la même position relative dans l'image. Une matrice de co-occurrence $C(i,j)$ est définie pour la relation spatiale (dx,dy):

$$C(i,j) = \left| \left\{ (x,y) \in R \,\middle|\, I(x,y) = i \wedge I(x+dx, y+dy) = j \right\} \right|$$

Modèle d'apparence: texture

- Matrice de co-occurrences
  - Exemple (dx=0,dy=1)

	2	1	3
	1	0	1
	3	3	4



Modèle d'apparence: texture

- Matrice de co-occurrences
  - Comparaison:
    - On utilise les mêmes techniques que pour les histogrammes qu'on adapte pour deux dimensions.
    - Norme L1

$$D_1(C_i, C_M) = \sum_{j=1}^{K} \sum_{k=1}^{L} \left| C_i[j,k] - C_M[j,k] \right|$$

    - Norme L2

$$D_2(C_i, C_M) = \sqrt{\sum_{j=1}^{K} \sum_{k=1}^{L} \left(C_i[j,k] - C_M[j,k] \right)^2}$$

Modèle d'apparence: forme

- Modélisation par moments
  - Modélisation par la distribution des points composants l'objet.
  - Moment géométrique pour pixel x,y avec couleur $f(x,y)$:

$$m_{pq} = \sum_{x,y \in R} x^p y^q f(x,y)$$

Modèle d'apparence: forme

- Modélisation par moments
  - En pratique…
  - Aire:

$$m_{00} = \sum_{x,y \in R} x^0 y^0 = \sum_{x,y \in R} 1$$

  - Centroïde:

$$x_c = \frac{m_{10}}{m_{00}} = \frac{\sum_{x,y \in R} x^1 y^0}{m_{00}} = \frac{\sum_{x,y \in R} x}{m_{00}}$$

$$y_c = \frac{m_{01}}{m_{00}} = \frac{\sum_{x,y \in R} x^0 y^1}{m_{00}} = \frac{\sum_{x,y \in R} y}{m_{00}}$$

Modèle d'apparence: forme

- Pour rendre les moments invariants aux translations et aux rotations -> Moments centraux:

$$\mu_{pq} = \sum_{x,y \in R} (x - x_c)^p (y - y_c)^q f(x, y)$$

- Pour rendre les moments invariants à l’échelle:

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

Modèle d'apparence: forme

- Modélisation par moments
  - Descripteurs de formes pouvant être invariants à l'échelle, aux translations et aux rotations.
  - Exemple de moments de Hu d'ordre 2 (pas invariant à l'échelle):

$$\phi_1 = \eta_{20} + \eta_{02} = \sum_{x,y \in R} (x - x_c)^2 + \sum_{x,y \in R} (y - y_c)^2$$

$$\phi_2 = (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2$$

$$\phi_2 = \left(\sum_{x,y \in R} (x - x_c)^2 - \sum_{x,y \in R} (y - y_c)^2 \right)^2 + 4\left(\sum_{x,y \in R} (x - x_c)(y - y_c)^2 \right)^2$$

  - *Exemple MATLAB: Moments.m*

Bibliographie

- D. Batz et al. A computer vision sytem for monitoring medication intake, in Second Canadian Conference on Computer and Robot Vision (CRV'05)
- L.M. Fuentes, S.A. Velastin, People tracking in surveillance applications, in 2nd IEEE Int. Workshop on PETS, 2001
- H.N. Charif, S.J. McKenna, Tracking the activity of participants in a meeting, Machine Vision and Applications, Vol.17, No. 2, 2006,pp. 83-93
- S.H. Cha, S.N. Srihari, On measuring the distance between histograms, Pattern Recognition, Vol. 35, 2002, pp. 1355-1370
- J. Malik et al., Preattentive texture discrimination with early vision mechanisms, Journal of the Optical Society of America, Vol.7 No. 5 1990, pp.923-932
- D.A. Forsyth, J. Ponce, Computer Vision: A Modern Approach, Prentice Hall, 2002
- L.G. Shapiro, G.C.Stockman, Computer Vision, Prentice-Hall, 2001
- R.M. Haralik et al., Textural Features for Image Classification, IEEE Transactions on Systems, Man, and Cybernetics, Vol. SMC-3, No. 6, 1973, pp.610-621
- M.K.Hu, Visual Pattern Recognition by Moment Invariants, IRE Transactions on Information Theory, 1962, pp.179-187
- J. Flusser, Moment Invariants in Image Analysis, Transactions On Engineering, Computing and Technology, V. 11, 2006, pp. 196-201