IUP M2-ICM

Université Paul Sabatier
Toulouse

Métrologie et reconnaissance des formes

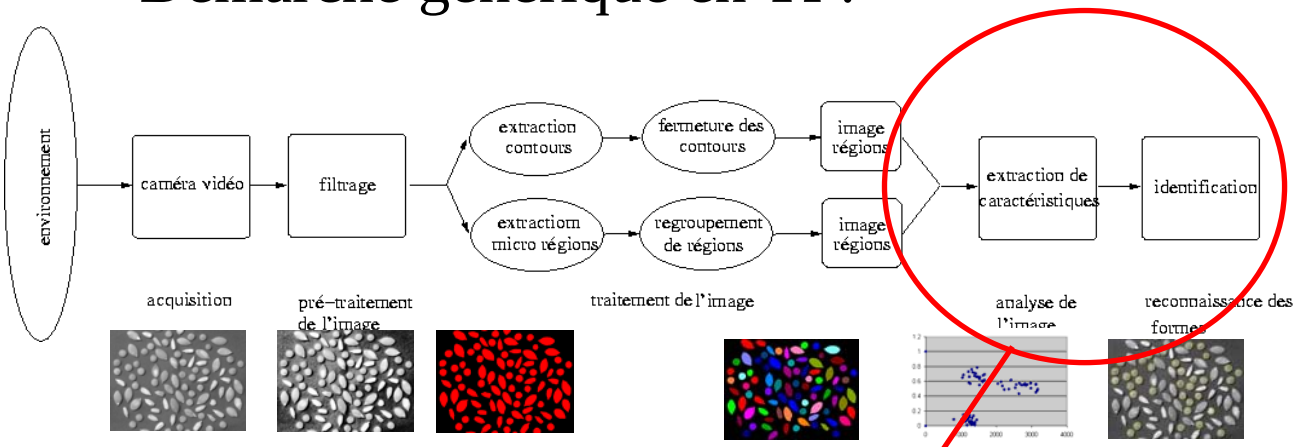
Frédéric Lerasle

Année 2008-09

Sommaire

- Généralités -

- Démarche générique en TI :



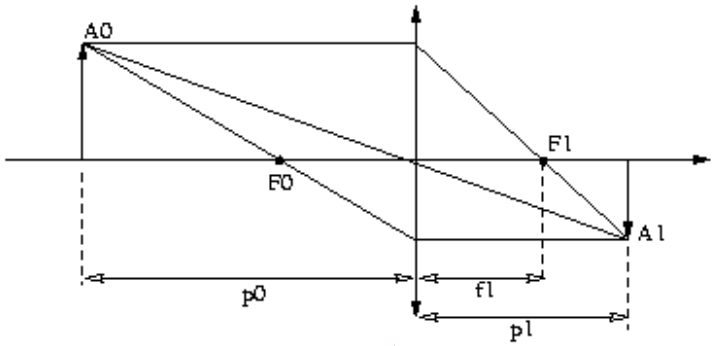
- mise en œuvre et acquisition des images
- prétraitement les images
- segmentation des régions pertinentes dans l'image
- **extraction des caractéristiques ou descripteurs de ces régions – éventuellement prétraiter ces données**
- **interprétation : (1) classer pour associer des symboles, (2) vérifier la conformité (mesures, présence,…)**

Sommaire

- Métrologie 3D (3h)
- Métrologie 2D (1h)
- Reconnaissance des formes (3h)
- Exercices (1h)

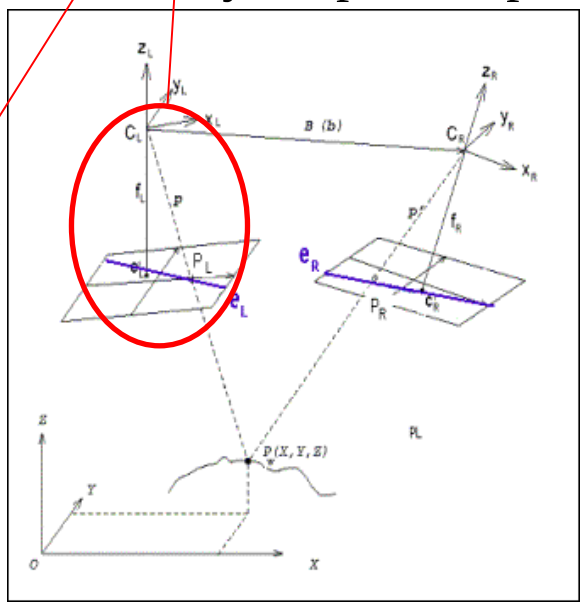
- Rappels -

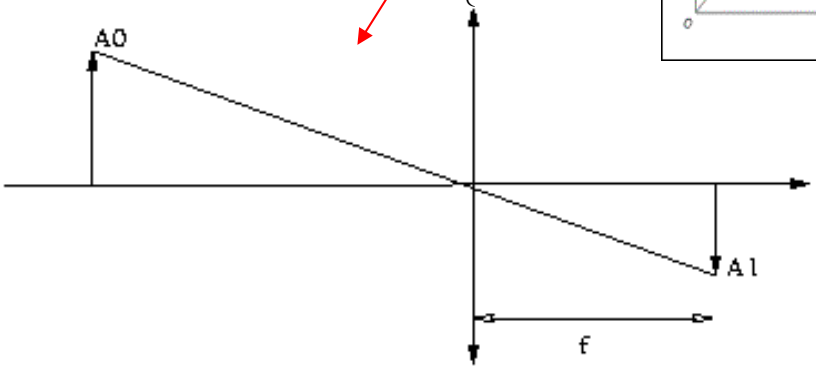
- Modèle mince
  - approximation du modèle épais i.e plans principaux confondus



- Modèle sténopé
  - appelé « *pin-hole* » car tous les rayons passent par le centre optique
  - un seul rayon conservé
  - projection perspective :

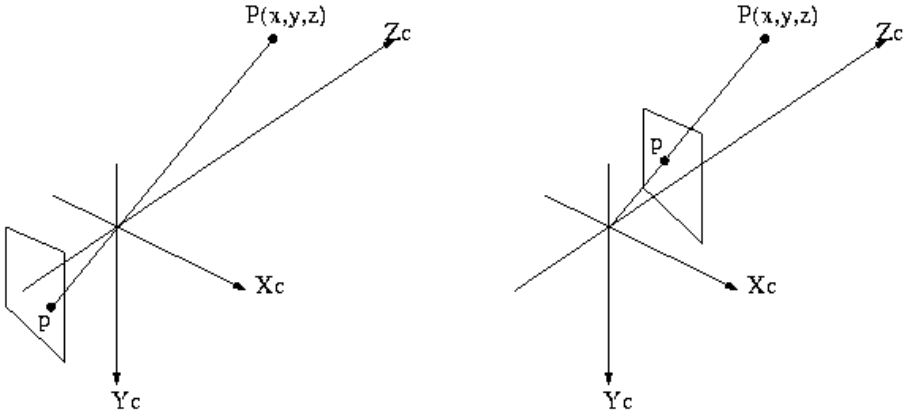
$$\begin{cases} x = f.\dfrac{X}{Z} \\ y = f.\dfrac{Y}{Z} \\ z = f \end{cases}$$



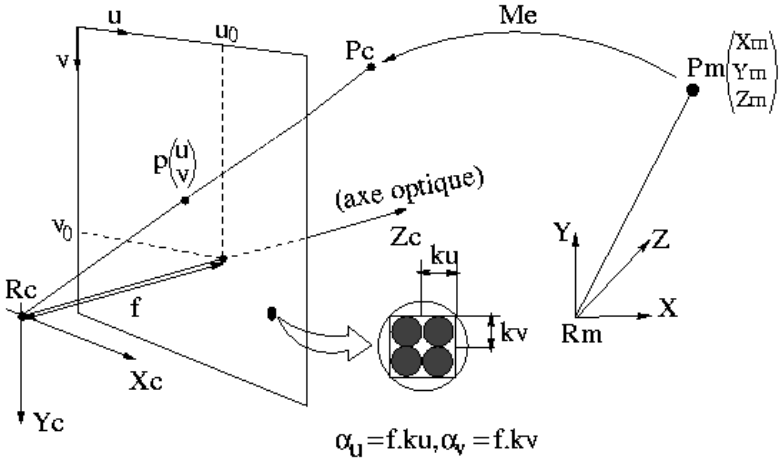


- Modélisation de la caméra -

- **<u>But</u>** : relation mathématique entre coordonnées 3D (monde) / 2D (image)
- Modèle sténopé



- **Deux transformations :**
  - la projection perspective $\Rightarrow$ relation point espace 3D / point image 2D
  - la transformation repère caméra / repère image - unité : le pixel



- Modélisation de la caméra -

- **Projection perspective -** en coordonnées homogènes* :

$$\begin{pmatrix} sx' \\ s.y' \\ s \end{pmatrix} = \begin{pmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} . \begin{pmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{pmatrix}$$

- **Transformation caméra/image -** exprimée en pixel :

$$\begin{cases} u = k_u . x' + u_0 \\ v = \mathrm{k_v} . y' + v_0 \end{cases}$$

- **Paramètres intrinsèques** - estimés par calibrage

$$Avec\ \alpha_\mathrm{u} = k_u . f , \alpha_\mathrm{v} = k_v . f :$$

$$\begin{pmatrix} s.u \\ s.v \\ s \end{pmatrix} = \begin{pmatrix} \alpha_u & 0 & u_0 & 0 \\ 0 & \alpha_v & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} . \begin{pmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{pmatrix}$$

- **Paramètres extrinsèques** - transformation mire/caméra -estimés par calibrage

$$\begin{pmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{pmatrix} = \begin{pmatrix} R & T \\ 0 & 1 \end{pmatrix} . \begin{pmatrix} X_m \\ Y_m \\ Z_m \\ 1 \end{pmatrix}$$

- Modélisation de la caméra -

- **Transformation mire/image** - matrice M de projection perspective (taille 3x4)

$$M = \begin{pmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \end{pmatrix} \; tel\ que : \begin{cases} u_i = \dfrac{m_{11}.X_i + m_{12}.Y_i + m_{13}.Z_i + m_{14}}{m_{31}.X_i + m_{32}.Y_i + m_{33}.Z_i + m_{34}} \\ v_i = \dfrac{m_{21}.X_i + m_{22}.Y_i + m_{23}.Z_i + m_{24}}{m_{31}.X_i + m_{32}.Y_i + m_{33}.Z_i + m_{34}} \end{cases}$$

$\Rightarrow$ 12 paramètres à estimer

- **Principe du calibrage**
  - utilisation d 'une **mire parfaitement modélisée**
  - appariements 2D/3D : $(u_i, v_i, X_i, Y_i, Z_i)_{i=1,\dots,n}$
  - résolution du système :

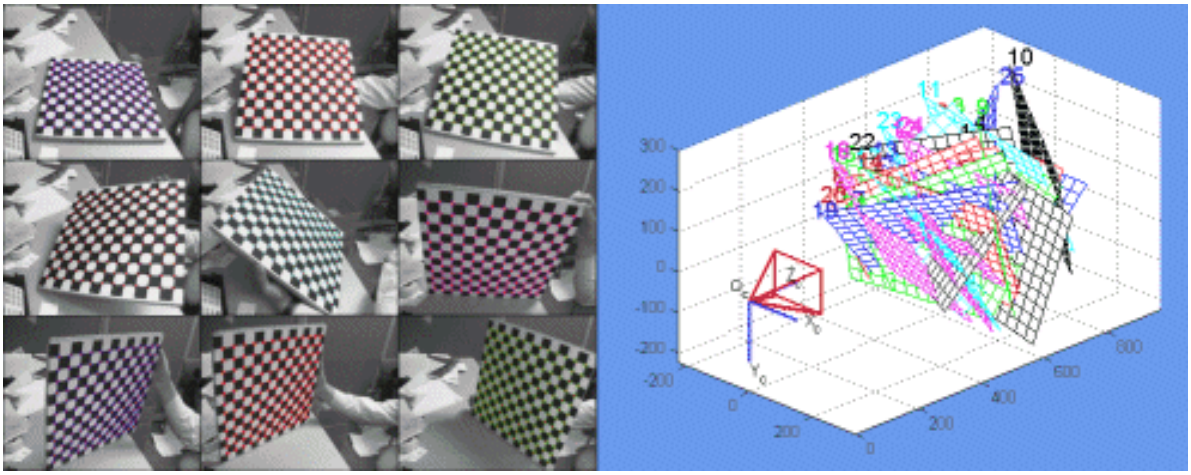
$$\begin{pmatrix} & & & & & & \vdots & & & & \\ X_i & Y_i & Z_i & 1 & 0 & 0 & 0 & 0 & -u_i.X_i & -u_i.Y_i & -u_i.Z_i \\ 0 & 0 & 0 & 0 & X_i & Y_i & Z_i & 1 & -v_i.X_i & -v_i.Y_i & -v_i.Z_i \\ & & & & & & \vdots & & & & \end{pmatrix} . \begin{pmatrix} m_{11} \\ m_{12} \\ m_{13} \\ m_{14} \\ m_{21} \\ m_{22} \\ m_{23} \\ m_{33} \\ m_{31} \\ m_{32} \\ m_{33} \end{pmatrix} = \begin{pmatrix} \vdots \\ u_i.m_{34} \\ v_i.m_{34} \\ \vdots \end{pmatrix}$$

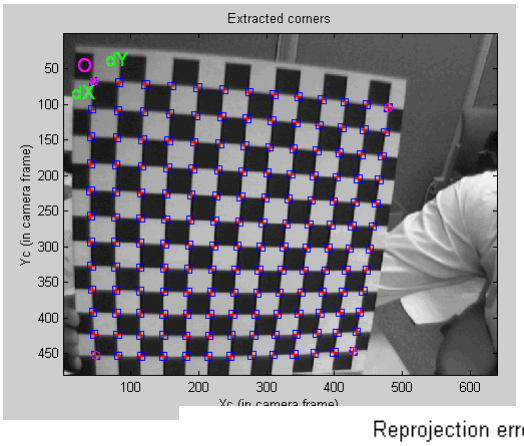
  - solution au sens des moindres carrés :

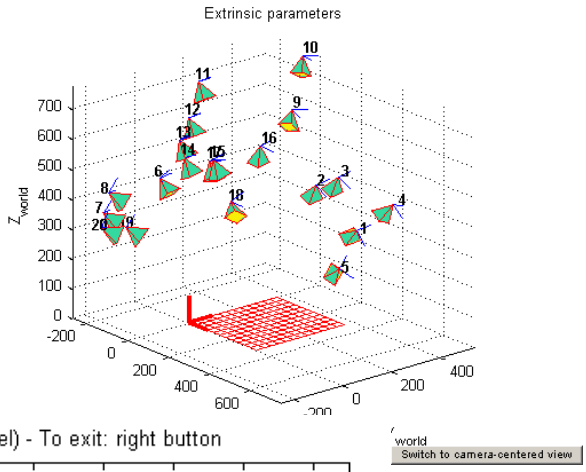
$$K_{2n\times 11}.X_{11} = u_{2n} \Rightarrow X_{11} = (K^t.K)^{-1}.K^t.u_{2n}$$

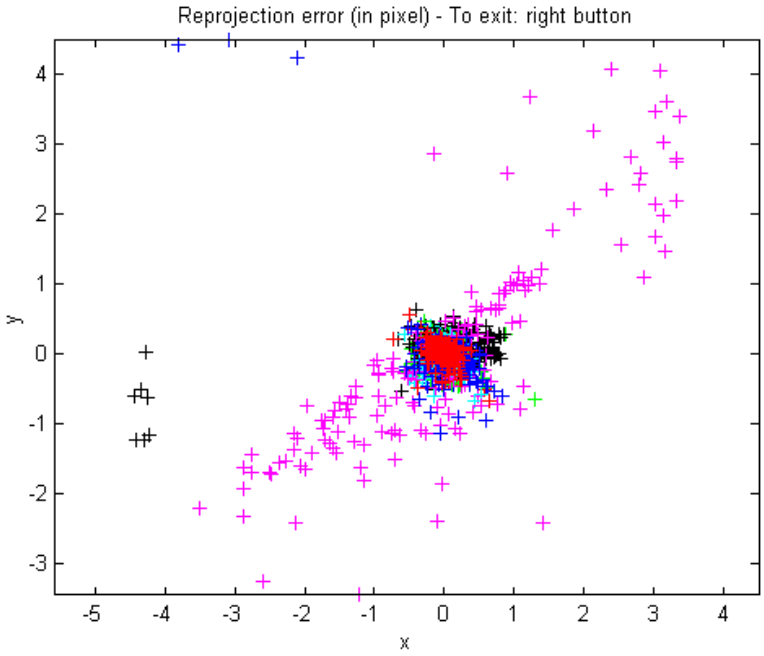
- Modélisation de la caméra -

- **« Camera calibration toolbox » de Matlab :**







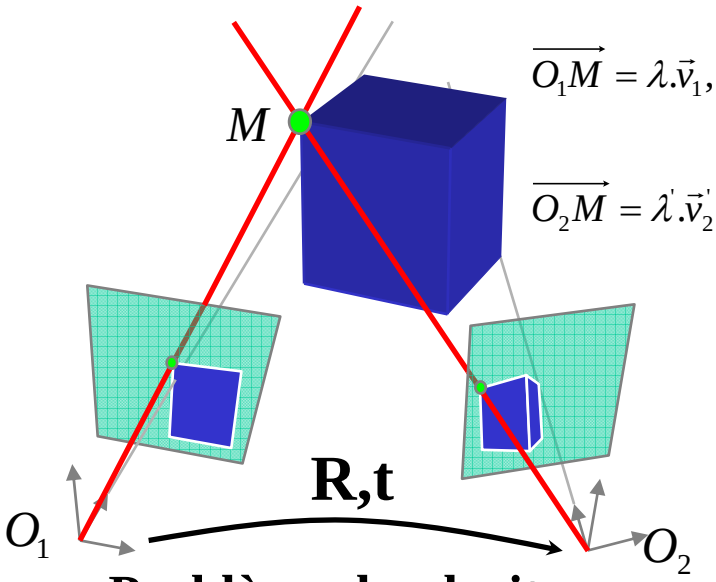


- Stéréovision -

- **Etalonnage :**
  - étalonnage indépendant de chacune des caméras
  - détermination du positionnement relatif entre caméras (choix arbitraire d 'une vue commune)

$$M_{gd} = M_d . M_g^{-1}$$

  - mise en correspondance et triangulation



$$\overrightarrow{O_1M} = \lambda . \vec{v}_1, \quad avec \quad \vec{v}_1 = \left(\frac{u_1 - u_0}{\alpha_u}, \quad \frac{v_1 - v_0}{\alpha_v}, \quad 1 \right)'$$

$$\overrightarrow{O_2M} = \lambda' . \vec{v}_2', avec \quad \vec{v}_2' = R.\left(\frac{u_2 - u_0'}{\alpha_u'}, \quad \frac{v_2 - v_0'}{\alpha_v'}, \quad 1 \right)'$$

- **Problème des droites non sécantes…**

- Stéréovision -

- **<u>Principe</u> :** analyse de plusieurs images prises simultanément sous des angles différents
- **<u>Etapes</u> :**

  1) étalonnage hors-ligne du système
  2) appariements de primitives images gauche/droite
  3) reconstruction 3D

- **Appariements - types de primitives**
  - régions - taille non conservée – segmentation
  - segments ⇒ reconstruction de segments 3D – scènes structurées – similarité de longueur, orientation,…
  - pixels - tous les pixels ou pixels spécifiques - information pauvre - occultations

image gauche image droite

- Stéréovision -

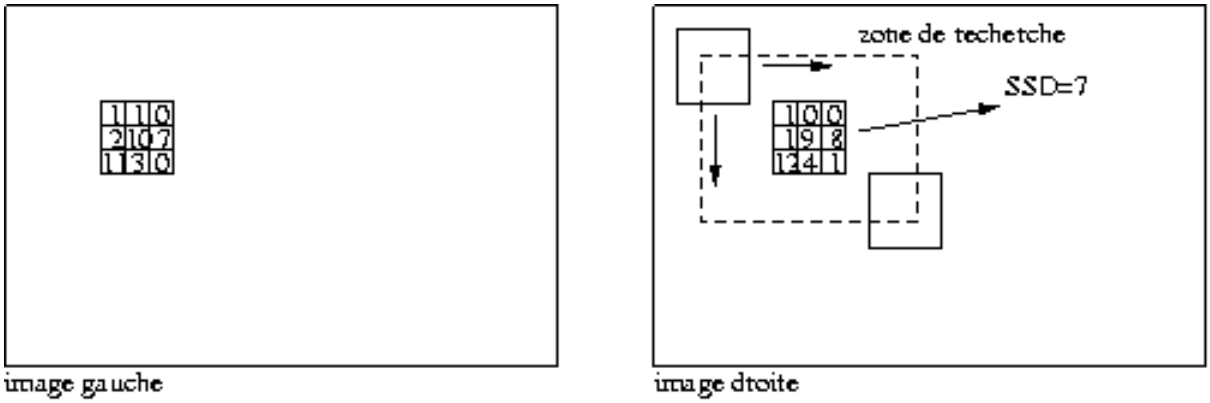
- **Appariements de pixels :**
  - scores paramétriques :

$$SSD(p_1, p_2) = \sum_{x=-M}^{+M} \sum_{y=-M}^{+M} [I(u_1 - x, v_1 - y) - I'(u_2 - x, v_2 - y)]$$

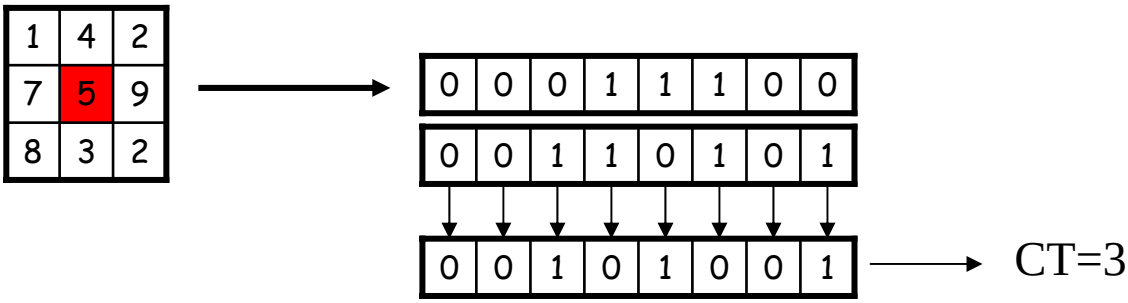
$$CC(p_1, p_2) = \sum_{x=-M}^{+M} \sum_{y=-M}^{+M} I(u_1 - x, v_1 - y).I'(u_2 - x, v_2 - y)$$

$$ZNCC(p_1, p_2) = \frac{\sum_{x,y} \left[I(u_1 - x, v_1 - y) - \bar{I}\right]\left[I'(u_2 - x, v_2 - y) - \bar{I}'\right]}{\sqrt{\sum_{x,y} \left[I(u_1 - x, v_1 - y) - \bar{I}\right]^2 \sum_{x,y} \left[I'(u_2 - x, v_2 - y) - \bar{I}'\right]^2}}$$

$p_1 = (u_1, v_1)^T$ apparié avec $p_2 = (u_2, v_2)^T$, I(u, v) niveau en (u, v)

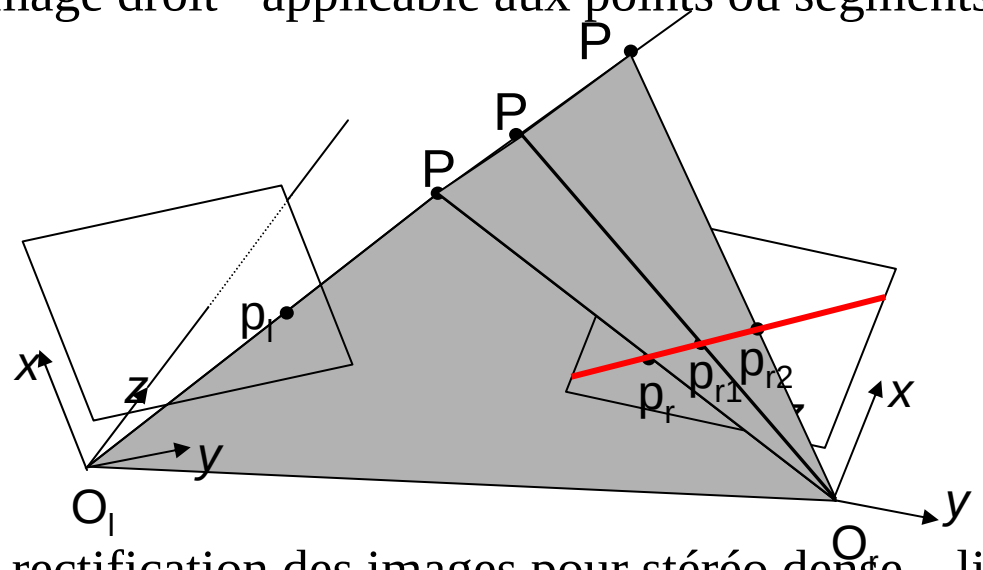


  - scores non paramétriques – exemple : Census Transform

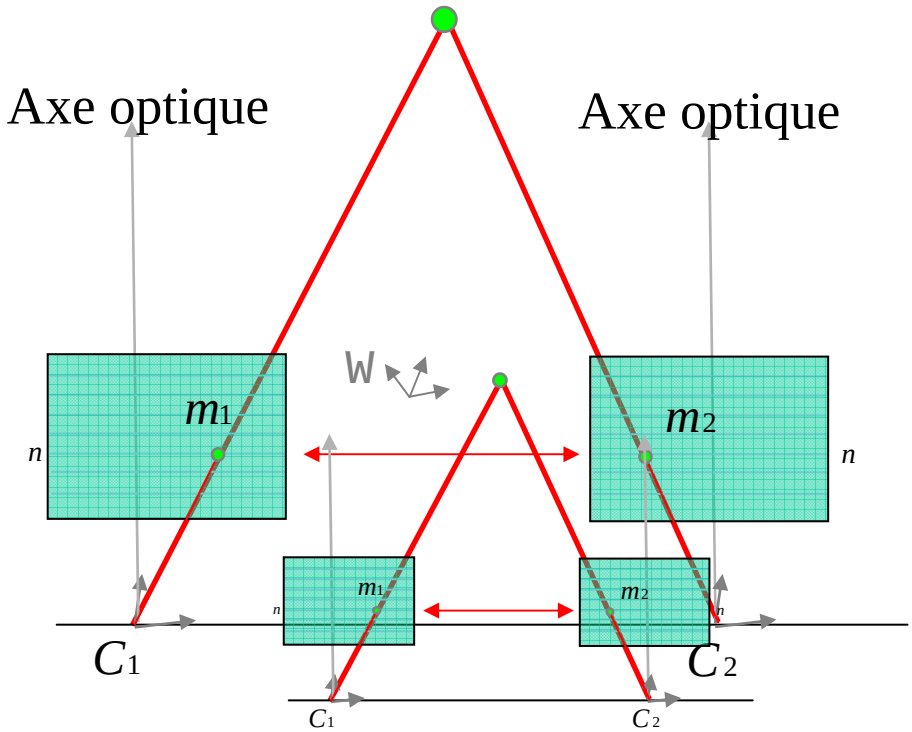


- Stéréovision -

- **Contrainte épipolaire :**
  - droite issue de l'intersection de (O_1, O_2, p_1) avec le plan image droit - applicable aux points ou segments

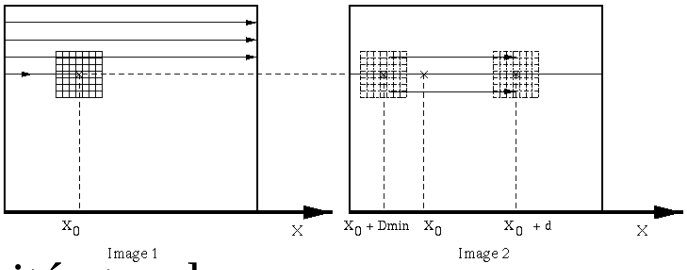


  - rectification des images pour stéréo dense – lignes épipolaires confondues avec les lignes image - matérielle ou logicielle

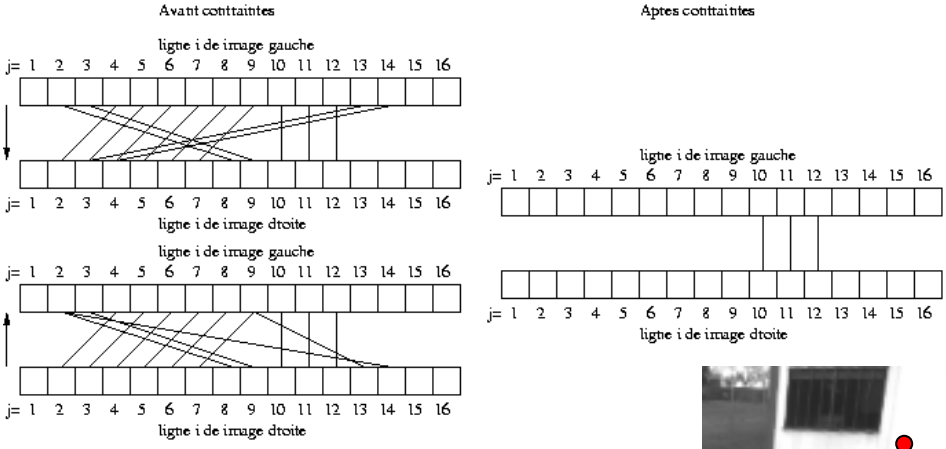


- Stéréovision -

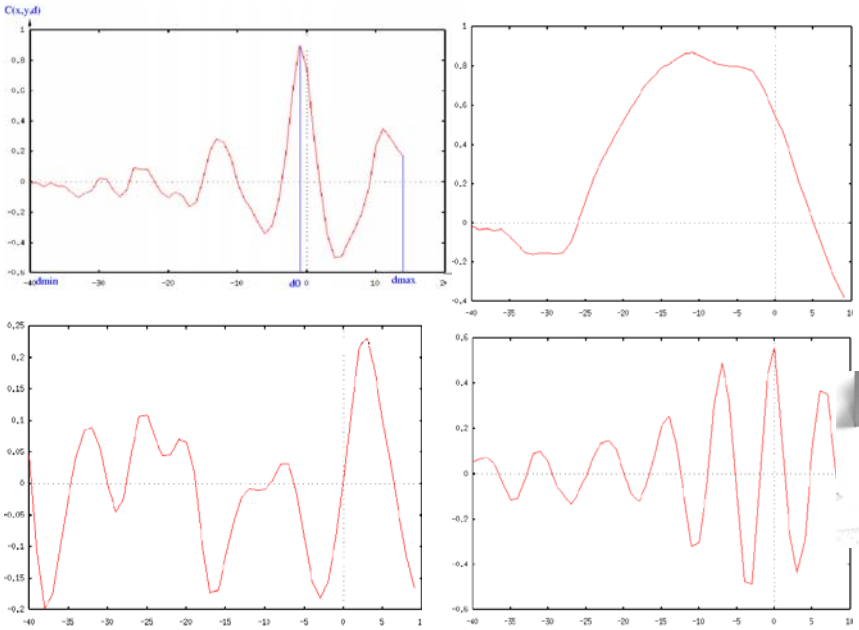
- **Stratégie en stéréovision dense**
  - application sur les lignes épipolaires – appariements bidirectionnels

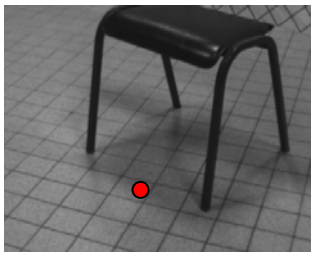


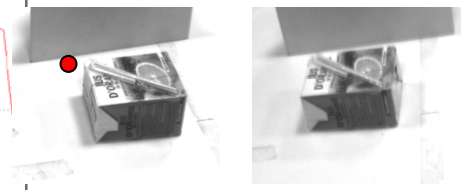
  - contraintes unicité et ordre



  - étude de la courbe de corrélation

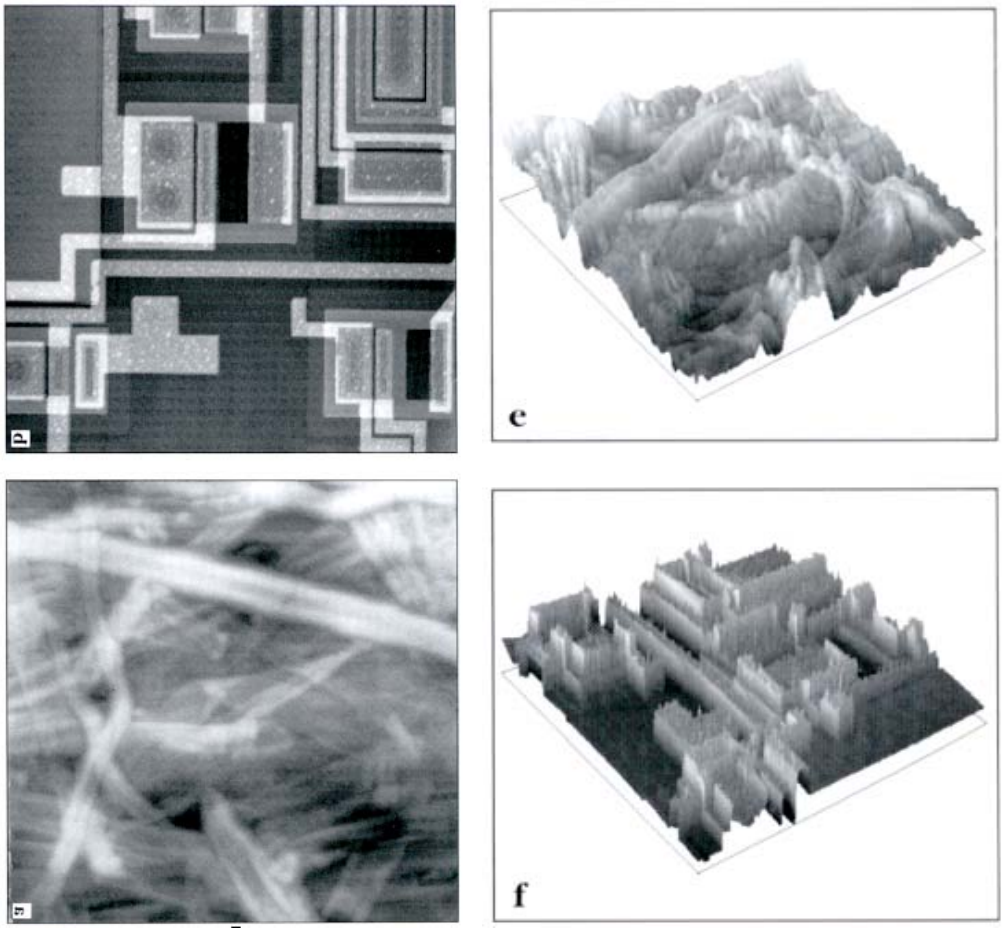






- Stéréovision -

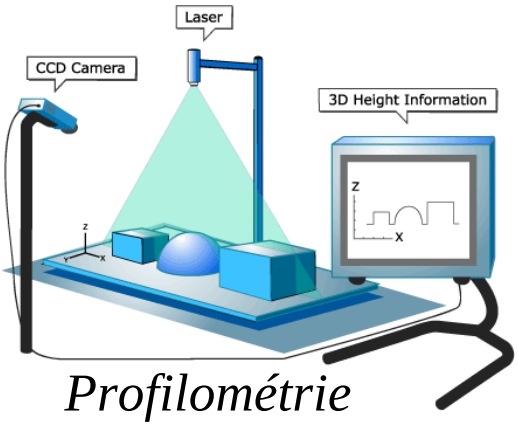
- **Quelques illustrations**

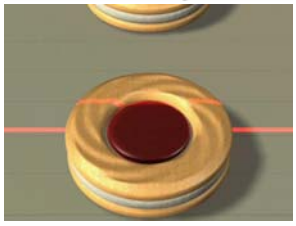


Images de profondeur et reconstruction 3D : papier - haut - et circuit imprimé – bas -

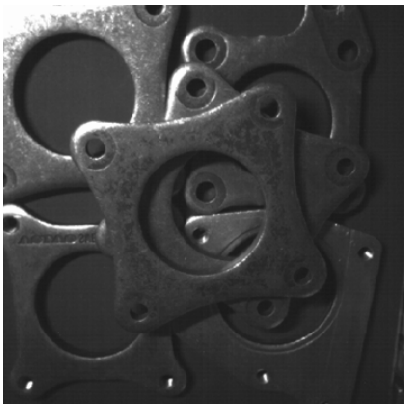
- Lumière structurée -

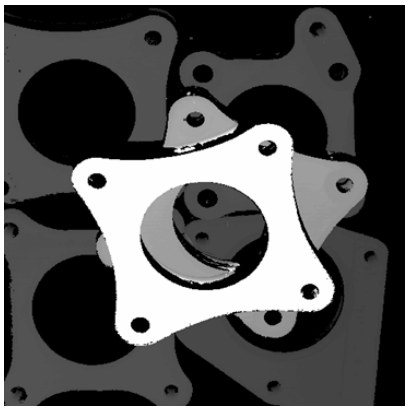
- **Couplage caméra+laser :**
  - laser source peu divergente et structurée
  - attention à la sécurité oculaire
  - différents motifs projetés
  - **<u>applications</u>**: profilométrie, métrologie 3D, contrôle de surfaces peu contrastées





Profilométrie



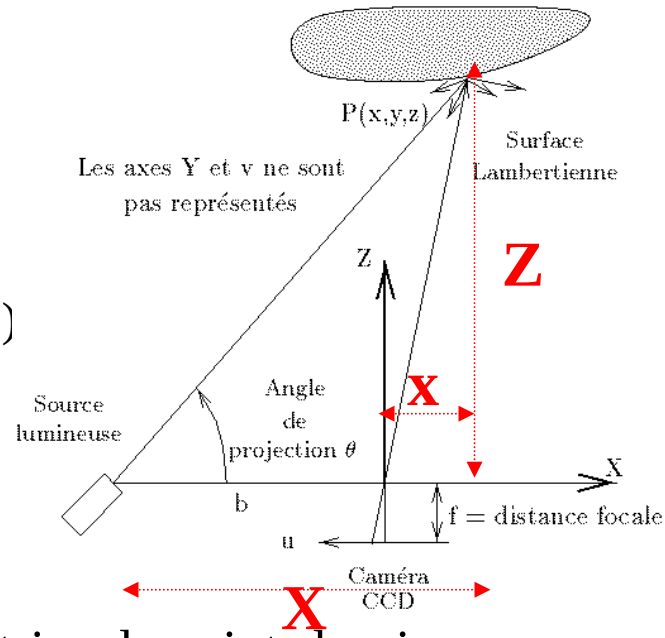


Mesures de profondeur

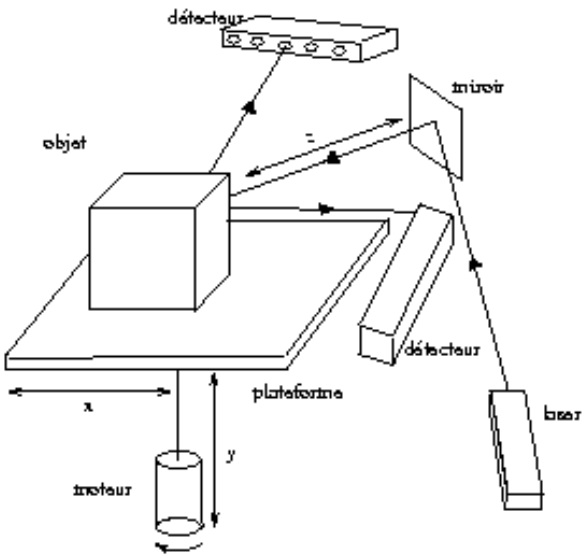
- Lumière structurée -

- **Principe pour un point lumineux :**
  - projection d ’un point lumineux - balayage de la scène - calibrage du faisceau - précision excellente

$$(x\ y\ z) = \frac{b}{f.\cot an\theta - u}.(u\ v\ f)$$



  - projection d ’une matrice de points lumineux - calibrage - identifier les points dans l’image - vitesse d ’acquisition ↗
- **Stéréovision active :**

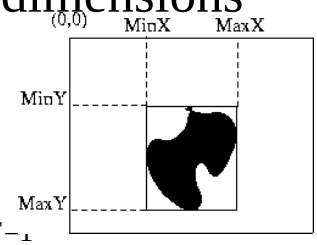


- Descripteurs de régions -

- Descripteurs de contours, de <u>régions</u> (spatiales, forme, texture, topologiques,…)

- Descripteurs spatiaux :

  - boîte englobante (« bounding box ») - dimensions



  - aire : $A(R) = \sum_{y=1}^{L-1} \sum_{x=1}^{C-1} f(x,y)$
  - centre de gravité :

$$\bar{x} = \frac{1}{A(R)} \sum_{y=0}^{L-1} \sum_{x=0}^{C-1} x.f(x,y), \quad \bar{y} = \frac{1}{A(R)} \sum_{y=0}^{L-1} \sum_{x=0}^{C-1} y.f(x,y)$$

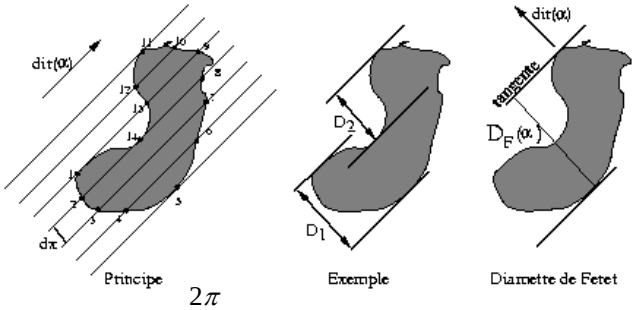
  - périmètre P(R) : nombre de pixels de la région ayant un voisin en dehors de celle-ci

  - diamètre, diamètre géodésique, …

  - direction par intercepts : 0°,45°,90°,135°

$$D(X,\alpha) = \int N_\alpha(X).dx$$

$N_\alpha(X)$ *nombre de points frontières interceptant les lignes* $d(\alpha)$



  - périmètre de Crofton : $P(R) = \int_{\alpha=0}^{2\pi} D(R,\alpha).d\alpha$

  - diamètre de Feret $D_F(\alpha)$: plus grande distance des points de contours de la région à T

- Descripteurs de régions -

Caractéristiques de forme (1) :

- compacité : $F_c = P(R)^2 / (4\pi . A(R))$
- élongation : $E(R) = 1 - (4\pi . A(R)) / P(R)^2$
- circularité de Heywood : $F_{cH} = P(R) / (2.\sqrt{\pi . A(R)})$
- moments : $m_{pq} = \sum_{y=0}^{L-1} \sum_{x=0}^{C-1} x^p . y^q . f(x, y)$
- moments d'inertie :

$$\mu_{pq} = \sum_{y=0}^{L-1} \sum_{x=0}^{C-1} (x - \overline{x})^p . (y - \overline{y})^q . f(x, y)$$

- dispersion autour du CdG (μ_{20}, μ_{02}), dissymétrie de la distribution (μ_{30}, μ_{03}), son caractère pointu (μ_{40}, μ_{04})
- moments de Hu invariants aux rotations, échelles, translations !

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

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

...

- orientation : valeur propre max de la matrice d'inertie

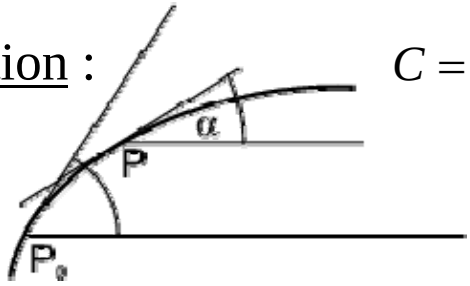
$$M = \begin{bmatrix} \mu_{20} \mu_{11} \\ \mu_{11} \mu_{02} \end{bmatrix}, \varphi = \frac{1}{2} Arc \tan \left[\frac{2\mu_{11}}{\mu_{20} - \mu_{02}} \right], -\frac{\pi}{4} \leq \varphi \leq \frac{\pi}{4}$$

- Descripteurs de régions -

Descripteurs de forme (2) :

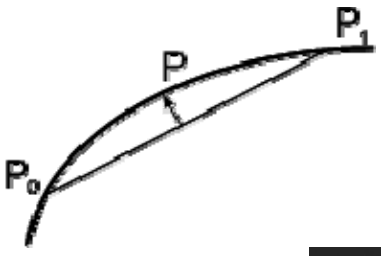
- courbure

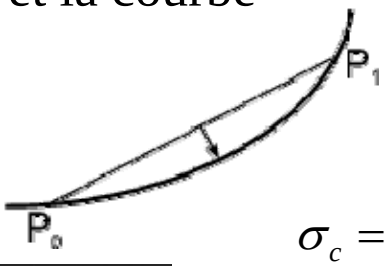
☞ <u>définition</u> :



$$C = \lim_{P_0 \to P} \frac{\alpha - \alpha_0}{PP_0}$$

☞ <u>mise en œuvre</u> : calcul de la distance entre le centre de la corde et la courbe

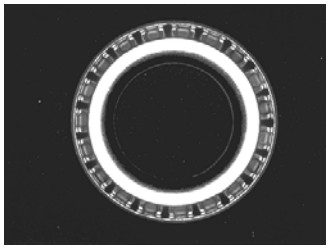


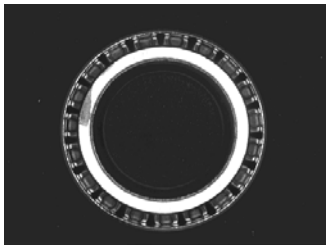


$\sigma_c = 1.206$

☞ <u>exemple</u> :

$\sigma_c = 0.547$





- descripteurs de Fourier :

☞ <u>principe</u> : un contour de coordonnées (x,y) noté sous forme complexe – les contours d'une région assimilés à une séquence de nombres complexes

$$u(n) = x(n) + j.y(n), n = 0,\ldots, N-1$$

$$a(k) = \sum_{n=0}^{N-1} u(n).\exp\left(\frac{-j2\pi k.n}{N}\right), 0 \le k \le N-1$$

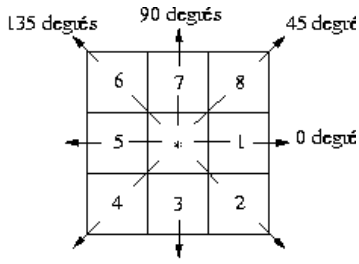
- Descripteurs de régions -

• Caractéristiques de texture (1) :

• matrices de **co-occurrence** :

☞ <u>principe</u> :

Soit I de dimension $L_x \times L_y$. *Soient deux pixels resp.* (x_1, y_1) *et* (x_2, y_2) *reliés par une relation géométrique* (distance d, direction θ) *et de ndg* resp. *i et j* :



$$p(i,j,d,0°) = Card[((k,l),(m,n)) \in (L_x \times L_y) \times (L_x \times L_y) / \mathrm{k-m=0} / |\mathrm{l-n}| = d, I(k,l) = i, I(m,n) = j]$$

$$p(i,j,d,45°) = Card[((k,l),(m,n)) \in (L_x \times L_y) \times (L_x \times L_y) / (\mathrm{k-m=d, l-n=-d}) \text{ or } (\mathrm{k-m=-d, l-n=d}), \mathrm{I(k,l)=i, I(m,n)=j}]$$

p coefficients de la matrice de co – occurence de dimension $L_x \times L_y$

☞ <u>exemple</u> :

0	0	1	1
0	0	1	1
0	2	2	2
2	2	3	3

niveaux de gris	0	1	2	3
0	#(0,0)	#(0,1)	#(0,2)	#(0,3)
1	#(1,0)	#(1,1)	#(1,2)	#(1,3)
2	#(2,0)	#(2,1)	#(2,2)	#(2,3)
3	#(3,0)	#(3,1)	#(3,2)	#(3,3)

$$P_{0°} = \begin{pmatrix} 4 & 2 & 1 & 0 \\ 2 & 4 & 0 & 0 \\ 1 & 0 & 6 & 1 \\ 0 & 0 & 1 & 2 \end{pmatrix}, P_{90°} = \begin{pmatrix} 6 & 0 & 2 & 0 \\ 0 & 4 & 2 & 0 \\ 2 & 2 & 2 & 2 \\ 0 & 0 & 2 & 0 \end{pmatrix}$$

$$P_{135°} = \begin{pmatrix} 2 & 1 & 3 & 0 \\ 1 & 2 & 1 & 0 \\ 3 & 1 & 0 & 2 \\ 0 & 0 & 2 & 0 \end{pmatrix}, P_{45°} = \begin{pmatrix} 4 & 1 & 0 & 0 \\ 1 & 2 & 2 & 0 \\ 0 & 2 & 4 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

Matrices symétriques !

- Descripteurs de régions -

Caractéristiques de texture (2) :

- matrices de **co-occurrence** :
  - ☞ pour invariance aux rotations : $P = \sum_{\theta=0°,\ldots,135°} P_\theta$
  - ☞ 14 attributs déduits de ces matrices, par exemple pour le contraste : $f_2 = \sum_{n=0}^{N_g-1} n^2 \left\{ \sum_{|i-j|=n} \frac{p(i,j)}{R} \right\}$
  - ☞ classifier une texture = calculer la distance de ses descripteurs à ceux de textures apprises

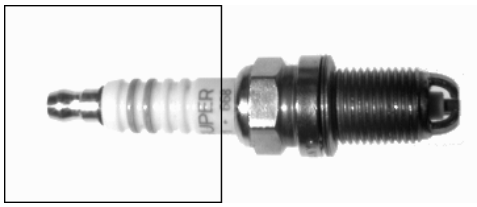
Caractéristiques topologiques

- nombre de trous (H)
- nombre de composantes connexes (C)
- nombre d'Euler : E=C-H
- nombre de branches du squelette

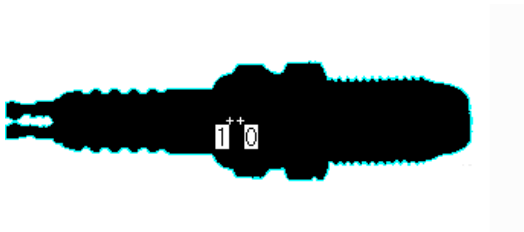
- Positionnement -

- <u>Un constat</u> : positionnement souvent imprécis ⇒ problème pour vérification de présence et métrologie

- Positionnement par l'objet entier
  - <u>principe</u> : recalage par le CdG de l'objet segmenté



  - (-) : CdG souvent instable… de part les conditions de capture

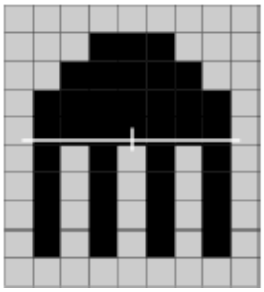


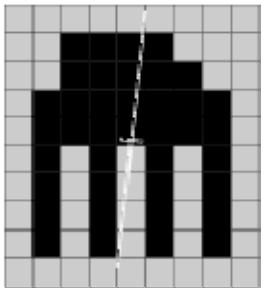
- Positionnement par contour(s)
  - <u>principe</u> : recalage par un contour spécifique

  - (-) : sensibilité, qualité du contour

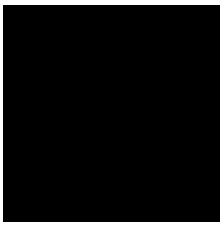
- Positionnement -

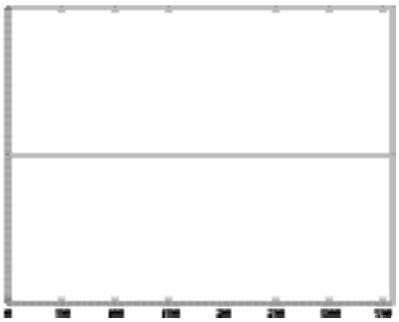
- Orientation
  - calcul de l'axe principal
    - ☞ principe basée sur la matrice d'inertie
    - ☞ (-) : cas des objets symétriques ou circulaires, temps de calcul

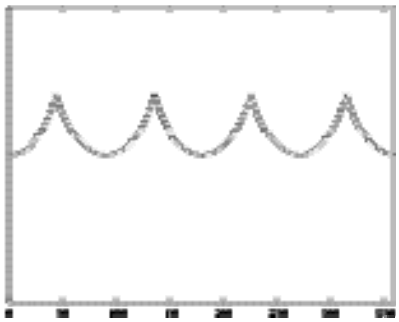




  - signature distance versus angle
    - ☞ <u>définition</u> : distance contour-CdG pour chaque direction







    - ☞ <u>mise en œuvre</u> : calcul du décalage maximisant le recouvrement des signatures des objets référence et courant

- Notions de bases -

- Notions d'individu et de variables :

$$\underline{x}_i = [x_{i1} \ldots x_{iP}]' \text{ et } i = 1, \ldots, N_j \textit{ de la classe } C_j$$

- Distance euclidienne entre individus
  - **<u>définition</u>** :

$$d^2(\underline{x}_i, \underline{x}_j) = \sum_{k=1}^{P} [x_{ik} - x_{jk}]^2 = (\underline{x}_i - \underline{x}_j)'.(\underline{x}_i - \underline{x}_j)$$

  - (-) : échelle des composantes, dispersion des composantes

- Distance de Mahalanobis
  - **<u>définition</u>** :

$$d^2(\underline{x}_i, \underline{x}_j) = [\underline{x}_i - \underline{x}_j]' W_j^{-1} [\underline{x}_i - \underline{x}_j]$$

$$avec W_j = [w_{kl}]_{1 \le k \le P, 1 \le l \le P}$$

$$w_{kl} = \frac{1}{N_j} \sum_{\underline{x}_i \in C_j} [x_i(k) - \bar{x}_j(k)][x_i(l) - \bar{x}_j(l)]$$

  - (+) : distance assurant une grande cohésion du nuage
  - (-) : connaissance a priori des classes, besoin de beaucoup d'individus

- Distance et matrice de covariance totale T

$$t_{kl} = \frac{1}{N_T} \sum_{i=1}^{N_T} [x_i(k) - \bar{x}(k)].[x_i(l) - \bar{x}(l)]$$

- Prétraitement des données -

- Analyse en composante principale (ACP) :
  - **<u>but</u> :** réduction de la dimension de l'espace
  - basé sur la matrice de covariance totale T
  - **<u>principe</u>** : calcul de l'axe u séparant au mieux les données dans l'espace i.e. donnant la variance totale maximale – base orthonormée

$$[T-\lambda I]\underline{u} = \underline{0}$$

  - **<u>démarche pratique</u>** : extraire les n vecteurs propres de la matrice T correspondant aux n premières valeurs propres car proportionnelles au pouvoir discriminant de chaque axe
  - choix du nombre d'axes ?

$$\frac{\sum_{i=1}^{n} \lambda_i}{\sum_{i=1}^{P} \lambda_i} \approx 1, P \text{ dim}ension\ de\ l'espace\ initial$$

- Prétraitement des données -

Analyse Factorielle discriminante (AFD) :

- **<u>but</u> :** réduction de la dimension de l'espace des données mais en mode supervisé
- basé sur la matrice de covariance intra-classes :

$$W = \lfloor w_{k,l} \rfloor \, avec \; \bar{x}_j(k) \, valeur \;\; moyenne \;\; du$$

$$paramètre k \, dans \;\; la \;\; classe \, C_j \; (1 \le j \le Q)$$

$$w_{k,l} = \frac{1}{N_T} \sum_{j=1}^{Q} \sum_{x_i \in C_j} \left[x_i(k) - \bar{x}_j(k) \right] . \left[x_i(l) - \bar{x}_j(l) \right]$$

- basé sur matrice interclasses B :

$$B = \left[b_{k,l} \right] \; avec \;\; b_{k,l} = \frac{1}{N_T} \sum_{j=1}^{Q} N_j . \left[\bar{x}_j(k) - \bar{x}(k) \right] . \left[\bar{x}_j(l) - \bar{x}(l) \right]$$

- **<u>principe</u>** : calcul de l'axe qui regroupe au mieux chaque classe et sépare au mieux les différentes classes $\quad W^{-1}.B.\underline{u} = \lambda.\underline{u}$

- Mode supervisé -

- <u>But :</u> classifier une forme X parmi une classe $\{C_j\}_{j=1,\dots,Q}$ représentées par leurs individus $\{\underline{x}_i\}_{\underline{x}_i \in C_j}$

  définir $Q-1$ *fonctions de décision* $d_1(\underline{X}),\dots d_{Q-1}(\underline{X})$

  telles que si $d_k(\underline{X}) > d_l(\underline{X})$ *alors* $X \in C_k$

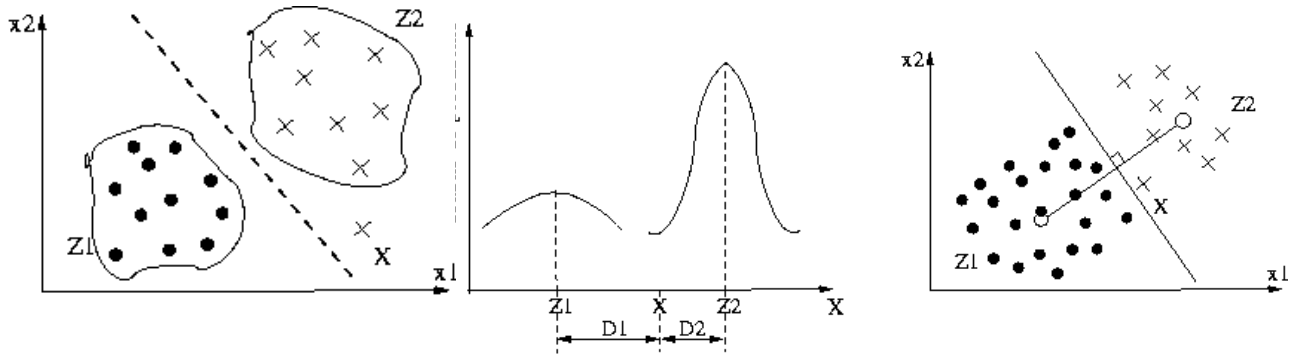
  i.e. séparer les classes C_k *et* C_l *par*

  $d_{kl}(\underline{X}) = d_k(\underline{X}) - d_l(\underline{X}) = 0$

- Classes représentées par des prototypes
  - classement de X à partir du prototype le plus proche en distance euclidienne :

$$d_j(\underline{X}) = d(\underline{X}, \underline{\bar{x}}_j)_{1 \le j \le Q}$$

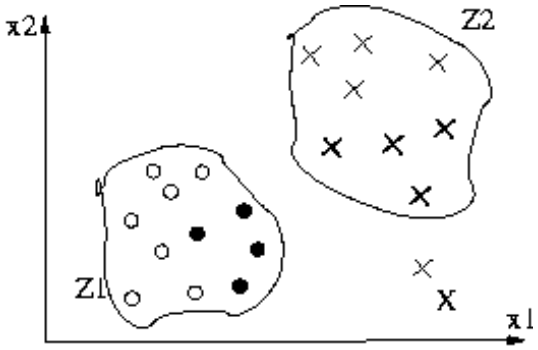
  - (+) : compacité de la modélisation
  - (-) : pondération des caractéristiques…
  - (-) : dispersion des formes d'apprentissage



- Mode supervisé -

- Classes représentées par tous les individus

  - analyse du voisinage de la forme à classer i.e. des k plus proches voisins (k-PPV)



  - **règle de décision** :

$$\{\underline{x}_i\}_{1\le i\le k} \in C_m \;\; et \;\; \{\underline{x}'_i\}_{1\le i\le k} \in C_n \;\; les \;\; k-PPV$$

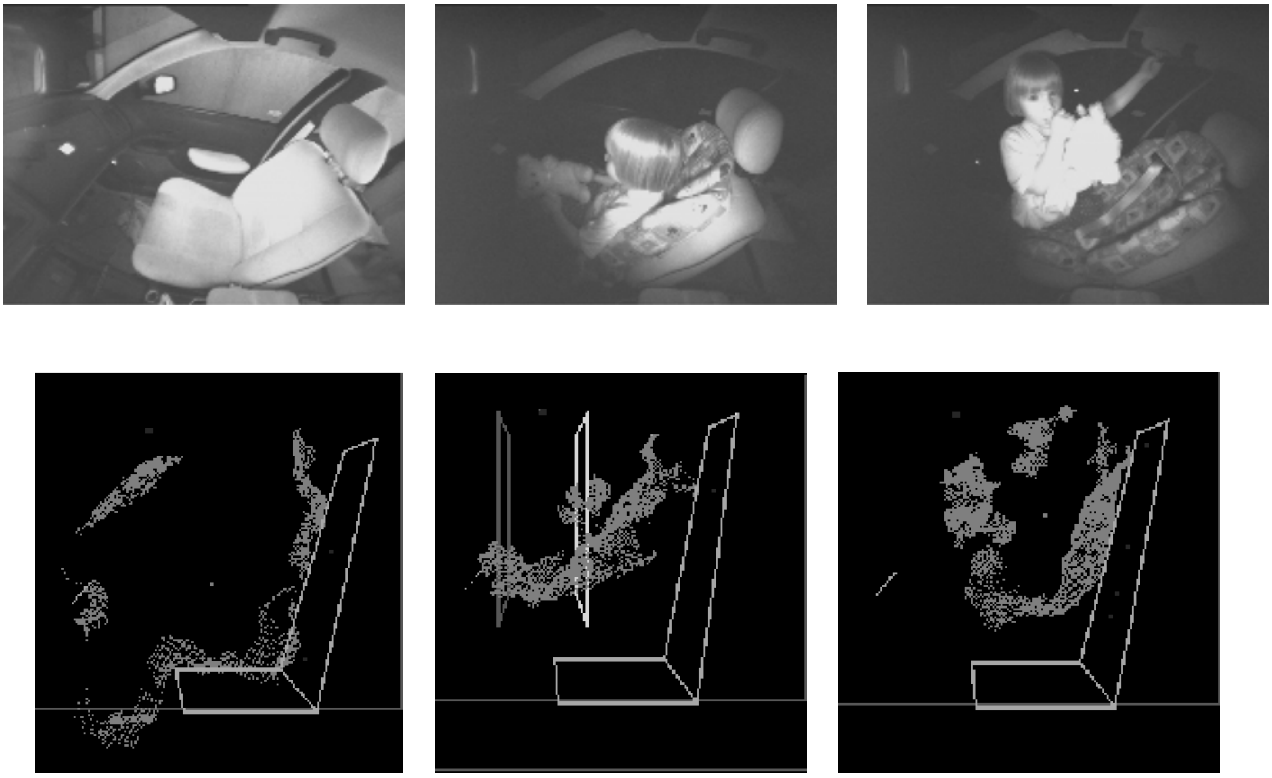
$$de \;\; \underline{X} \;\; pour \;\; la \;\; classe \;\; C_m \; et \; C_n :$$

$$si \sum_{i=1}^{k} d(\underline{x}_i, \underline{X}) < \sum_{i=1}^{k} d(\underline{x}'_i, \underline{X}) \; alors \; \underline{X} \in C_m$$

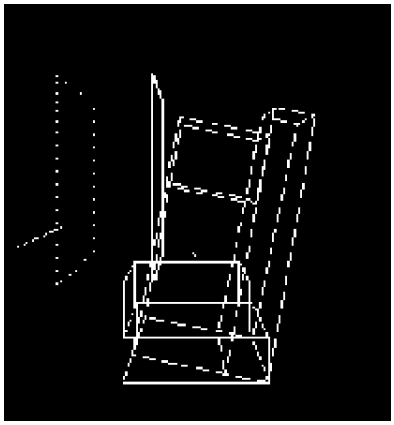
  - (+) : tient compte de la dispersion des classes
  - (-) : mémorisation de toutes les formes d'apprentissage
  - (-) : calcul d'un grand nombre de distance pour connaître les k-PPV
  - (-) : choix de k ? Problème des bords ?

- Mode supervisé -

- Application de classification par k-PPV : reconnaissance de situation



Images stéréo et reconstruction 3D

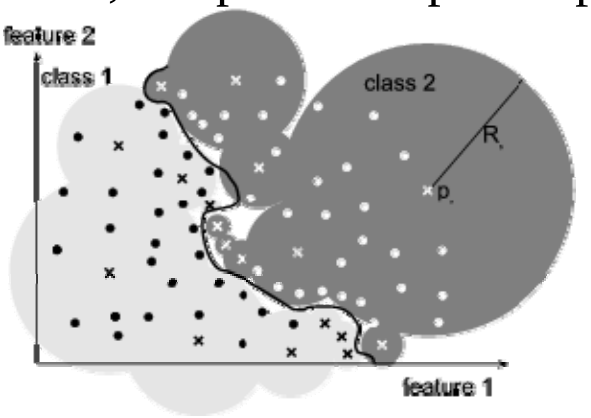


Boîtes pour le comptage des points 3D

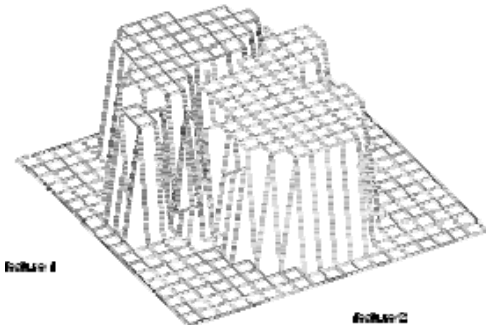
- Mode supervisé -

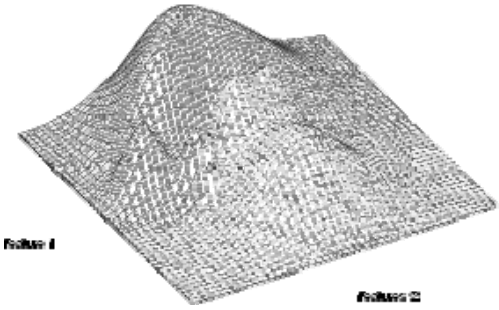
• Classes représentées par plusieurs prototypes

- approximation de la distribution de chaque classe par un ensemble de cercles, hypersphères,…représentés par un prototype



- (+) : limitation du nombre de formes à stocker…
- **<u>règle de décision/rejet</u>** : $d(\underline{x}_j, \underline{X}) \leq R_j$
- (-) : règle de rejet binaire ⇒ ajout de fonctions de probabilité décroissant suivant la distance au prototype
  - ☞ modèle « Restricted Coulomb Energy »
  - ☞ modèle « Radial Basis Function »





- Mode supervisé -

Classification bayésienne

- lois de probabilité de chaque classe connues
- **<u>décision bayésienne</u>** :

$p(\underline{X}/C_i)$ *probabilité conditionelle* $\underline{X}/\underline{X} \in C_i$

$p(C_i)$ *probabilité a priori de la classe* C_i

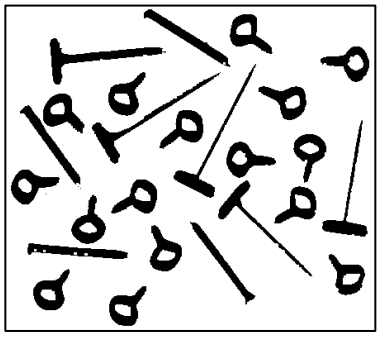
$d_i(\underline{X}) = p(\underline{X}/C_i).p(C_i),\ i = 1,\ldots,Q$

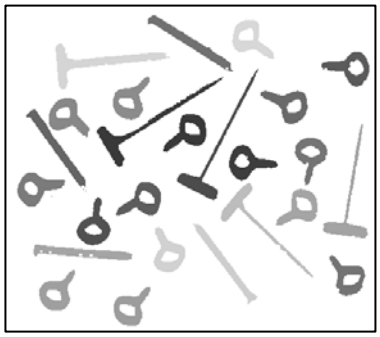
$\underline{X}$ *classée dans* C_i *si* $d_i(\underline{X}) > d_j(\underline{X})$

$$p(\underline{X}/C_i) = \frac{1}{(2\pi)^{\frac{P}{2}}|W_i|^{\frac{1}{2}}}.\exp\left[-\frac{1}{2}(\underline{X}-\underline{\bar{x}}_i)'.W_i^{-1}.(\underline{X}-\underline{\bar{x}}_i)\right]$$

- Mode supervisé -

• Exemples de classification bayésienne





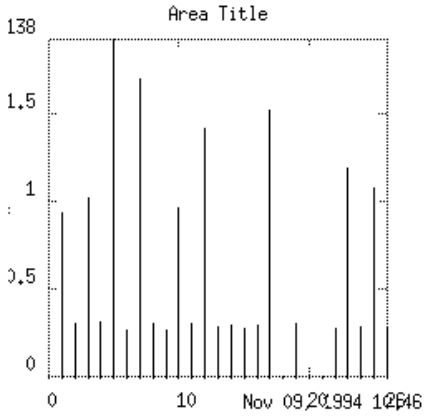
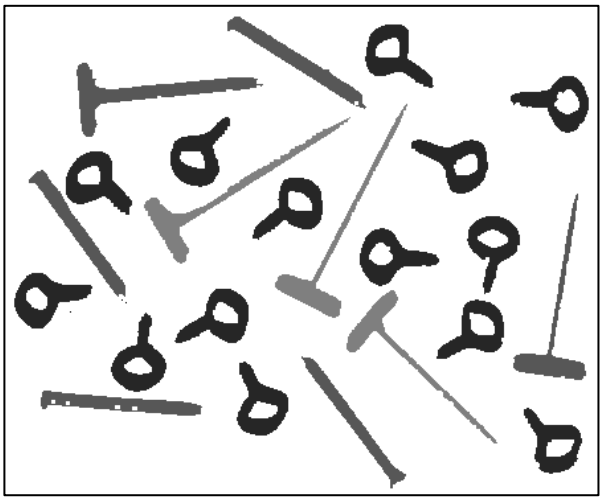


Image originale et image labellisée *Moment invariant d'ordre 2*

Classe	Objet	Numéro d'objet	Valeur de l'attribut
0	fond	0	-1
1	clou	1	0.93
2	Vis à tête	2	0.3
3	pointe	5	1.92
4	bruit	13	0.0

Choix des prototypes



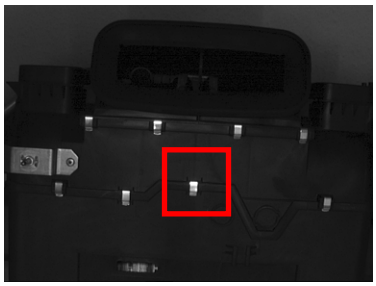
Résultat de la classification

Numéro d'objet	Valeur de l'attribut
3	1.01
12	1.41

Objet mal classé

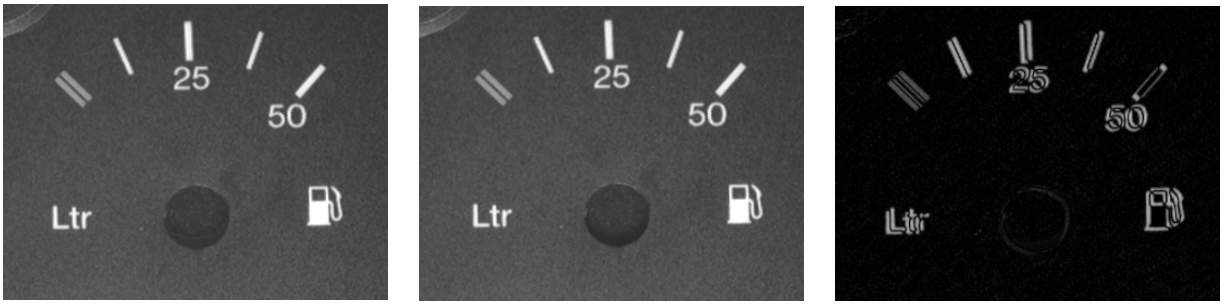
- Mode supervisé -

- « Template matching »
  - une approche non ascendante…
  - **<u>but</u>** : détecter la présence d'une forme a priori
  - utilisation de scores de similarité
  - (+) : pas de segmentation préalable de l'image
  - (-) : temps de calcul, robustesse au facteur d'échelle
  - **<u>optimisation des calculs</u>** : recherche hiérarchique, sous échantillonnage, troncature des calculs
  - **<u>exemple</u>** :

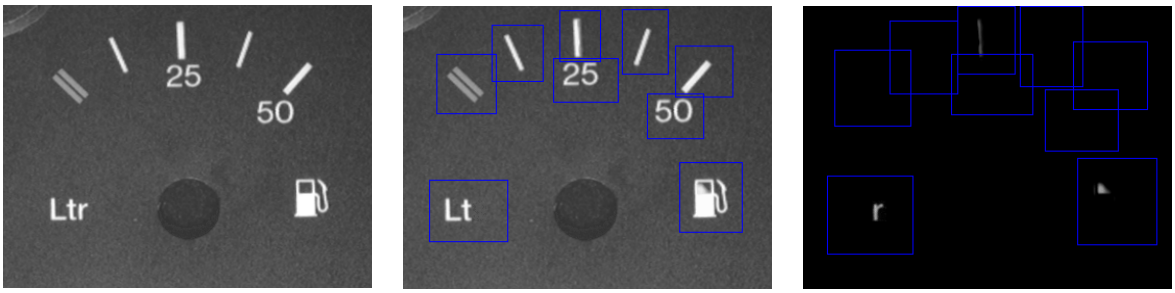


- Mode supervisé -

- Application : inspection de qualité d'impression
  - **but** : vérifier la lisibilité des caractères sur une jauge à carburant
  - **démarche classique :** différence entre image de référence puis tests, requiert un positionnement très précis…

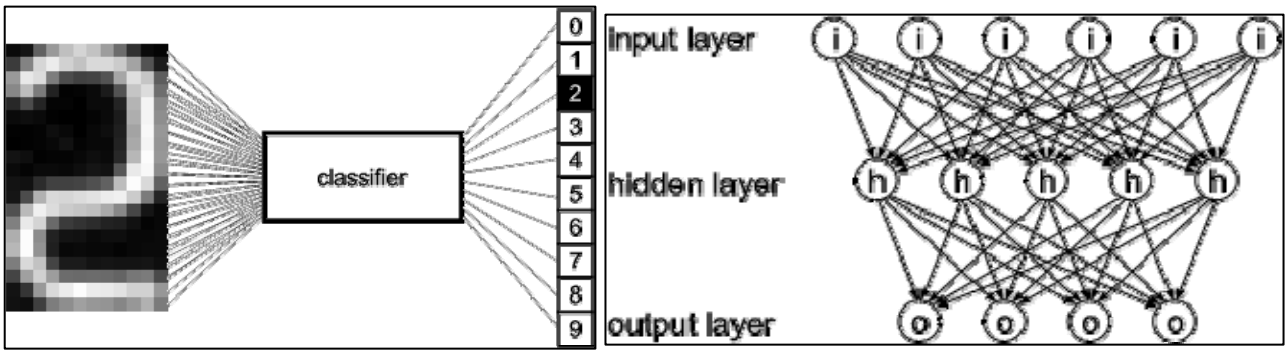


❶ recalage grossier à partir du trou et corrélation sur « template matching »
❷ recalage fin de chaque région par corrélation sur « template matching »
❸ différence d'images – segmentation régions
❹ filtrage du bruit à partir de caractéristiques telles que aire, ratio, contraste,…



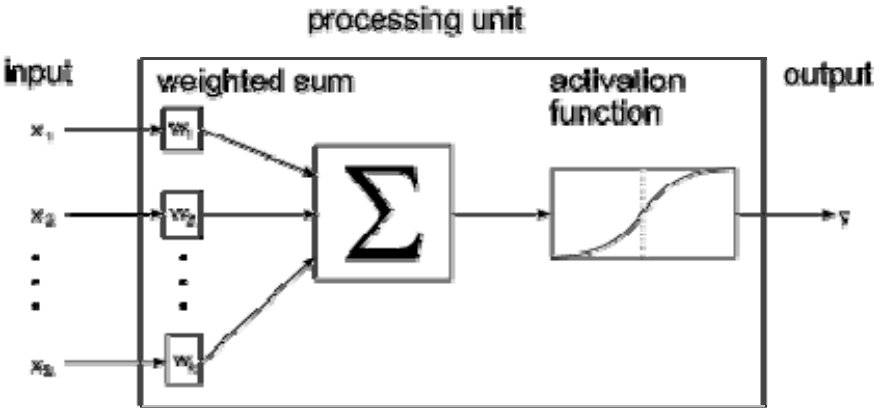
- Mode supervisé -

- Réseau de neurones (1)
  - **perceptron multicouches** : trois couches interconnectées de neurones



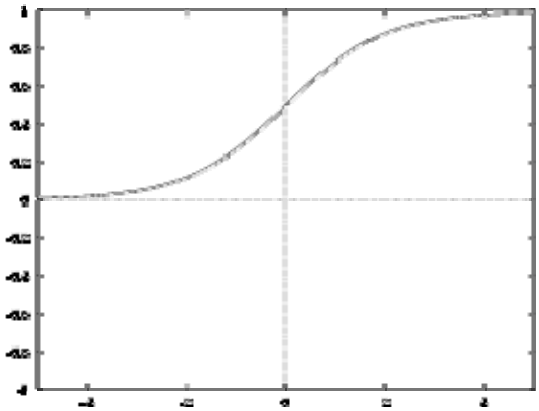
  - **neurone élémentaire** : somme pondérée des entrées et une fonction d'activation en sortie

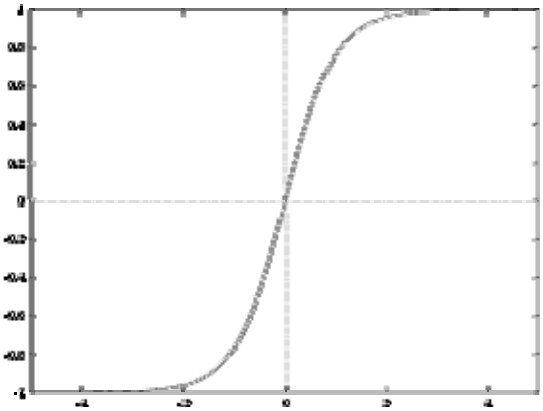
$$a_j = f_a(\sum_k w_{kj} o_k - \theta_j)$$



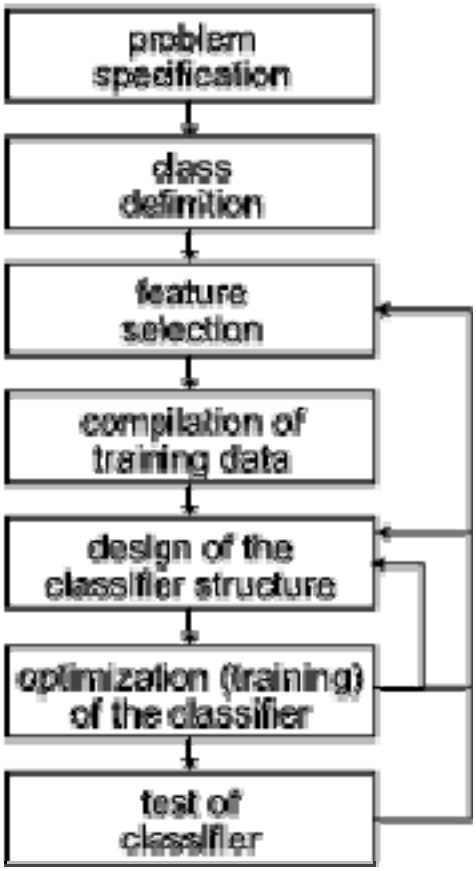
- Mode supervisé -

- Réseaux de neurones (2)
  - θ_j seuil d'activation du neurone
  - fonction d'activation dérivable – <u>exemples</u> :





  - apprentissage des poids w_{kj} par rétro-propagation de l'erreur



- Mode supervisé -

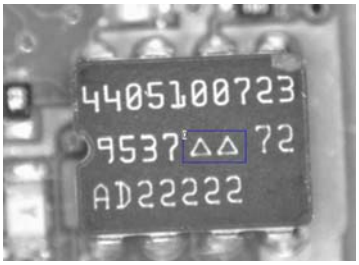
Application : OCR sur circuit imprimé (1)

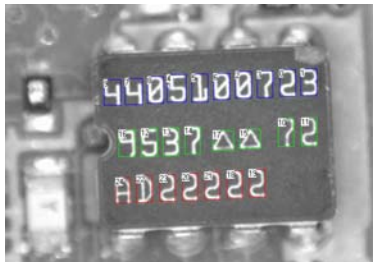
- **but :** vérification de présence
- utilisation de filtres polarisants
- **démarche classique (1) :**

☞ ajustement en position du circuit, 2 solutions :

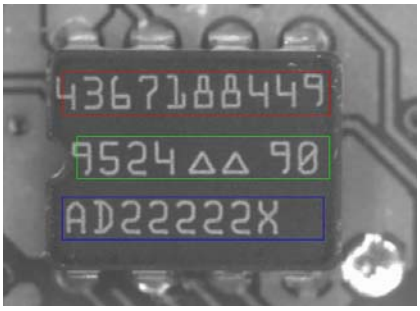
❶ par détection des arêtes verticales puis horizontales du circuit

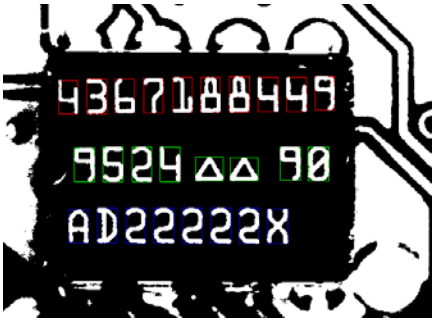
❷ par détection des deux triangles (« template matching ») puis recherche en relatif des autres caractères





☞ positionnement de ROIs et segmentation par seuillage

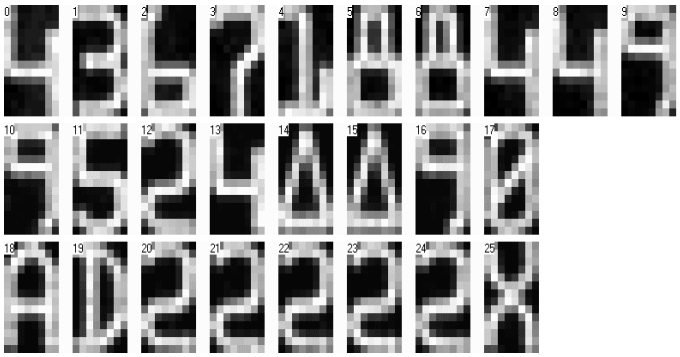




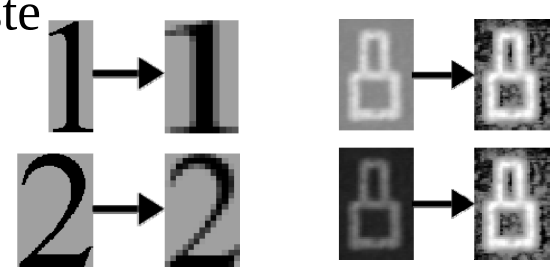
- Mode supervisé -

- Application : OCR sur circuit imprimé (2)
  - **démarche classique (2) :**

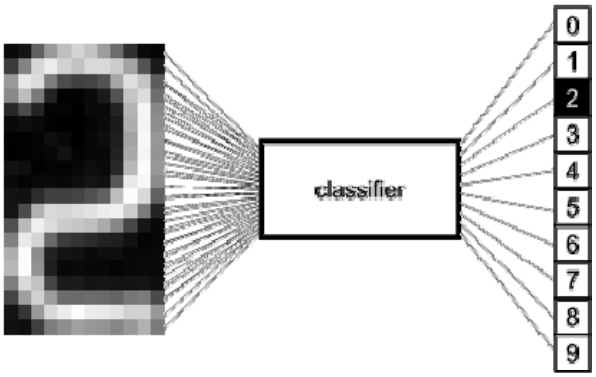
☞ extraction des rectangles englobant les régions segmentées et des imagettes associées



☞ ré-échantillonnage et normalisation de contraste



☞ classification par réseau de neurones

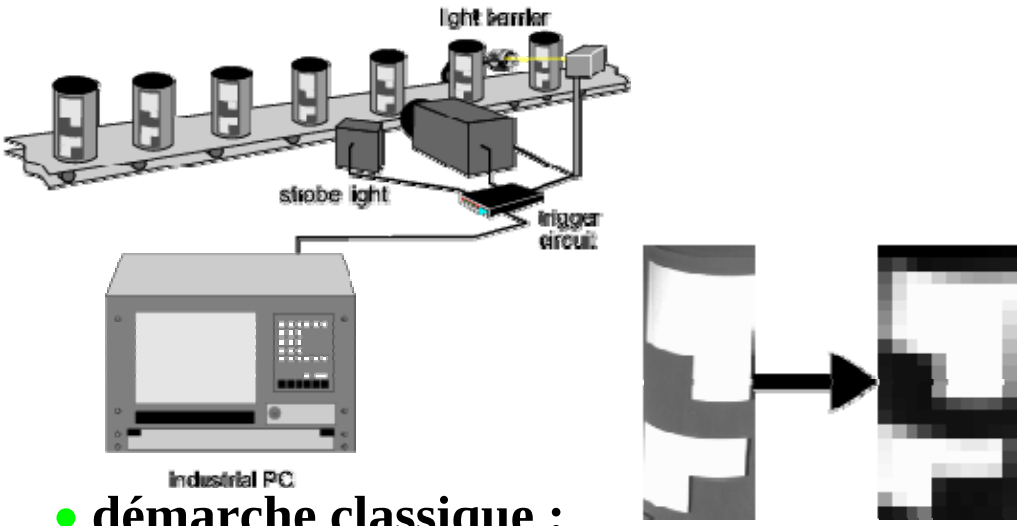


- Mode supervisé -

Application : Reconnaissance de code-barres

- **but** : identifier les codes de films 100, 200 et 400 ASA à 15Hz

- **dispositif** : caméra asynchrone, « shutter » très court (1ms), éclairage stroboscopique pendant 1µs



- **démarche classique :**

☞ recalage en position par une arête verticale – pas de variation horizontale

☞ positionnement d'une ROI

☞ ré-échantillonnage sur une résolution de 10x20 pixels pour limiter l'information traitée

☞ classification par réseau de neurones

- Mode non supervisé -

• Algorithme des centres mobiles ou « K-mean » (1)

- aucun *a priori* sur la répartition des individus en classes

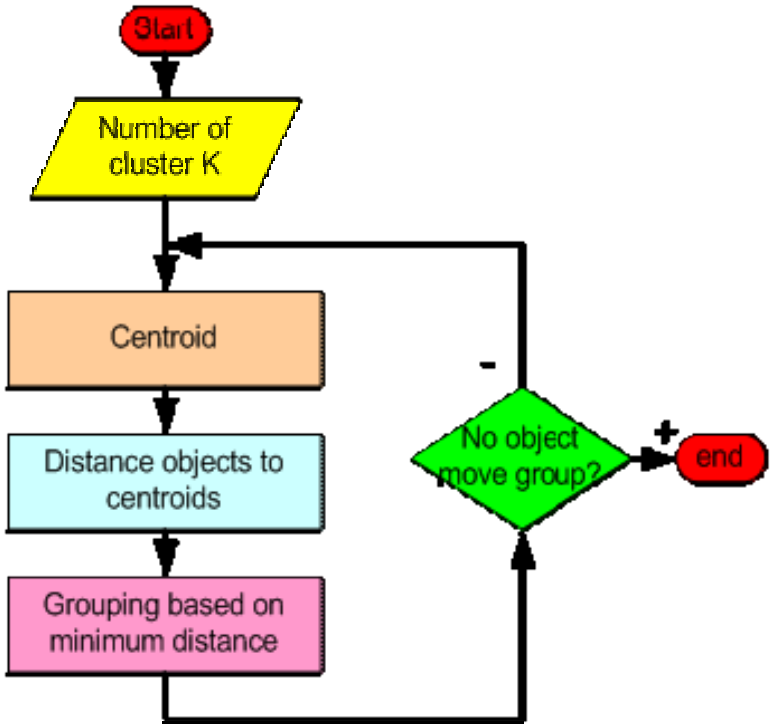
- **<u>algorithme</u>** :

❶ fixer le nombre de classes désirées Q

❷ choisir aléatoirement Q individus représentant les CdGs des classes

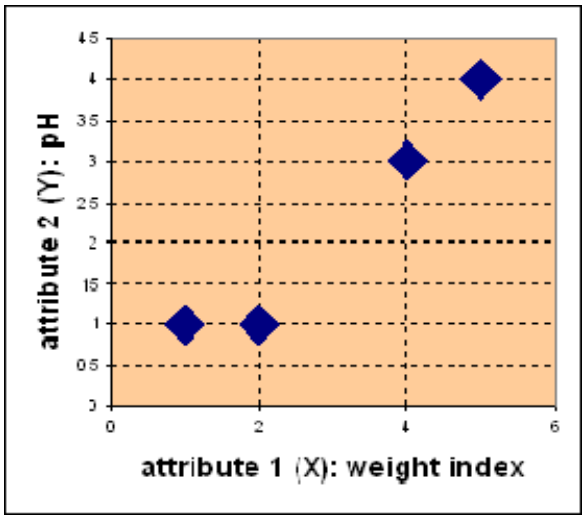
❸ réaffecter chaque individu $\underline{x}_i$ à la classe C_q dont le CdG est le plus proche

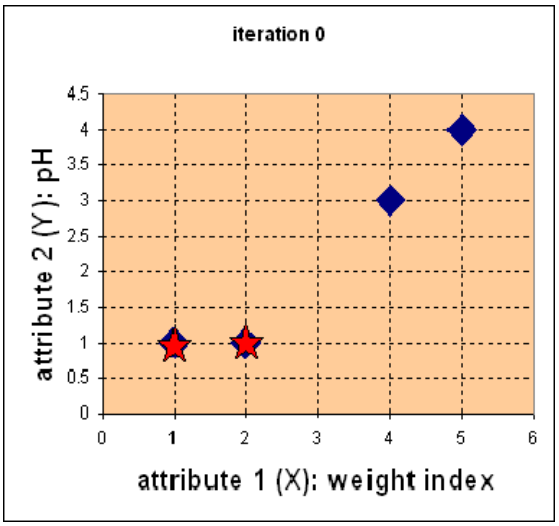
❹ calculer les CdGs des classes ainsi créées. Si changement dans répartition, revenir en ❷

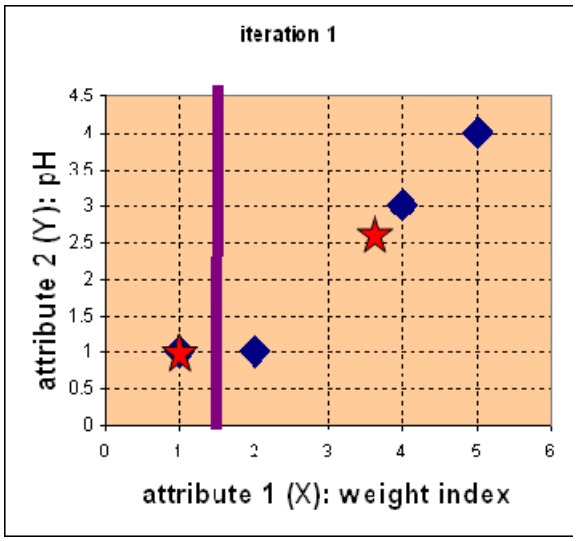


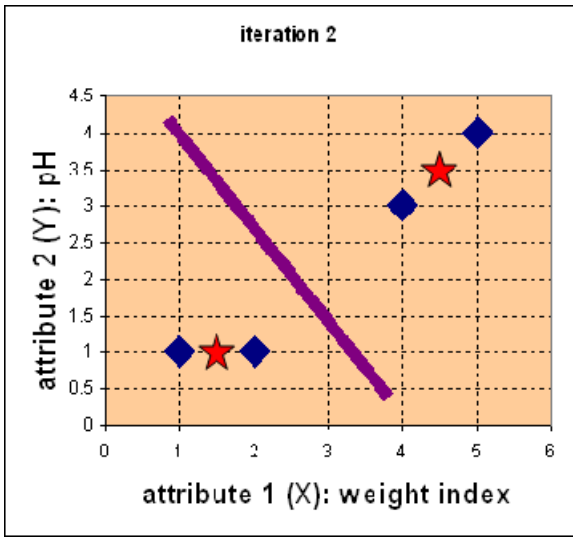
- Mode non supervisé -

- Algorithme des centres mobiles (2)
  - (-) : sensibilité aux conditions initiales et aux « outliers », Q fixé *a priori*, nombre important d'individus, « clusters » circulaires
  - exemple pour deux classes, 4 individus et 2 attributs :









- Mode non supervisé -

Classification hiérarchique ascendante

- **<u>but</u>** : minimiser la variance intra-classe
- basée classiquement sur la distance euclidienne
- **<u>algorithme</u>** :

❶ calculer les distances entre paires d'individus
❷ regrouper les deux individus les plus proches dans une même classe
❸ calculer les CdG des groupes

$$\underline{r}_3 = \frac{N_1.\underline{r}_1 + N_2.\underline{r}_2}{N_1 + N_2}$$

❹ calcul de la distance entre toutes les classes
❺ fusion des deux classes les plus proches si < seuil
❻ retour en ❷ sinon fin

- (-) : masse de calcul si ajout d'un individu, résultat lié à la distance utilisée

- Choix d'une méthode -

- Qualité
  - **précision :** erreur minimale, taux minimal de faux positifs et/ou faux négatifs
  - **lisibilité :** résultats interprétables
  - rapidité de la construction du modèle (« off-line », de la prise de décision (« on line »)
  - **autres** : besoins en apprentissage, ressources CPU et mémoire,…
- Ingénierie
  - « Usability » : facilité d'emploi
  - « embedability » : facilité d'introduction dans un système global
  - **flexibilité** : adaptation facile au changement de spécifications
  - **passage à l'échelle** : gros volume de données

Bibliographie

- Quelques livres… -

- **Industrial Image Processing**, C.Demant, B.Streicher-Abel et P.Waszkewitz, chez SPRINGER

- **Analyse d'images : filtrage et segmentation**, J.P.Cocquerez et S.Philipp

- **Traitement et analyse des images numériques**, S.Brès, J.M.Jolion, F.Lebourgeois, chez Hermès

- **Les systèmes de vision**, J.M.Jolion, chez Hermès