



---

M1 IUP SI

Université Paul Sabatier

Toulouse

PERCEPTION 3D

Frédéric Lerasle

Année 2008-09

# Sommaire

- chapitre I : acquisition

- généralités
- modélisation de la caméra
- stéréovision
- capteurs actifs

- chapitre II : modélisation

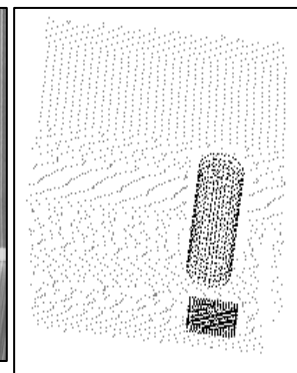
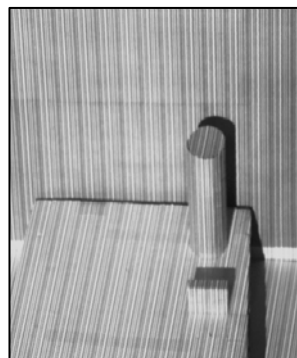
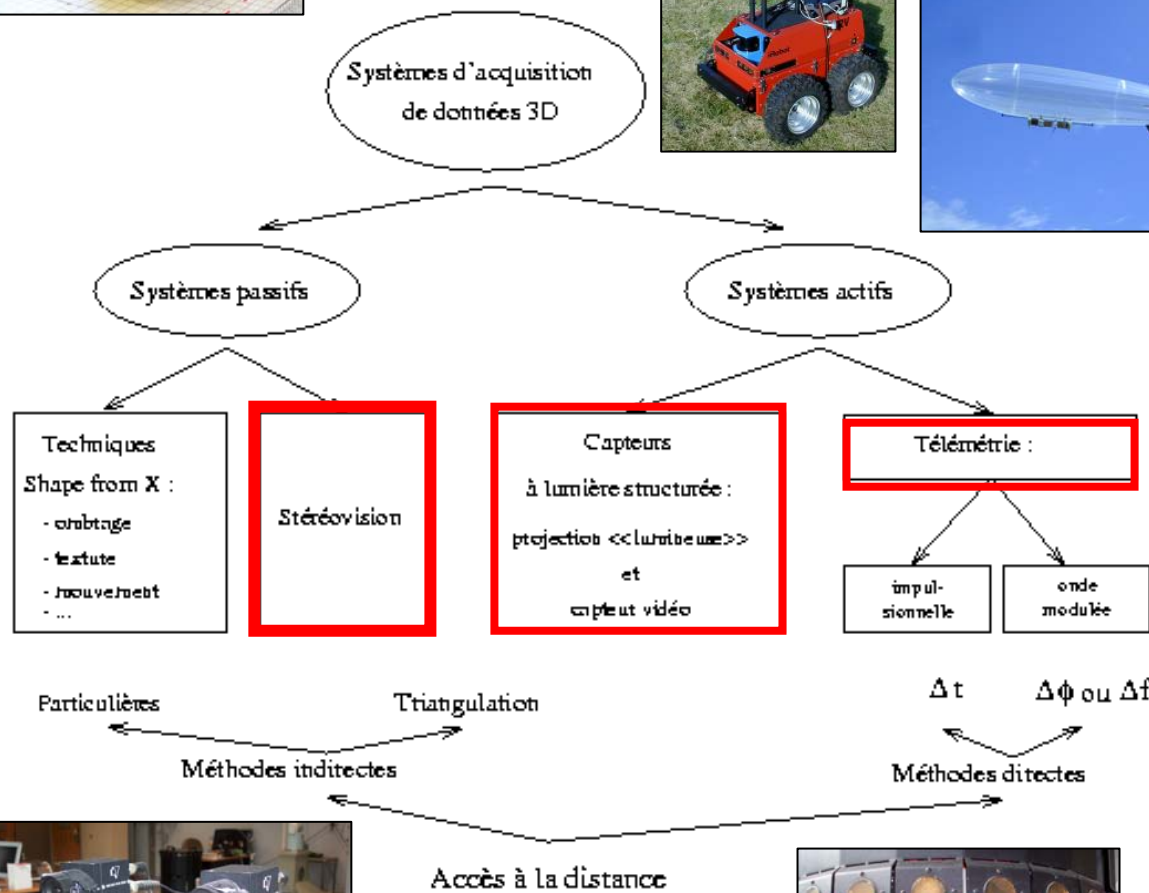
- généralités
- techniques
- stratégie incrémentale

- chapitre III : localisation/reconnaissance

- outils pour la localisation
- application à la saisie d'objets
- outils pour la reconnaissance

# Chapitre I

## - Généralités -

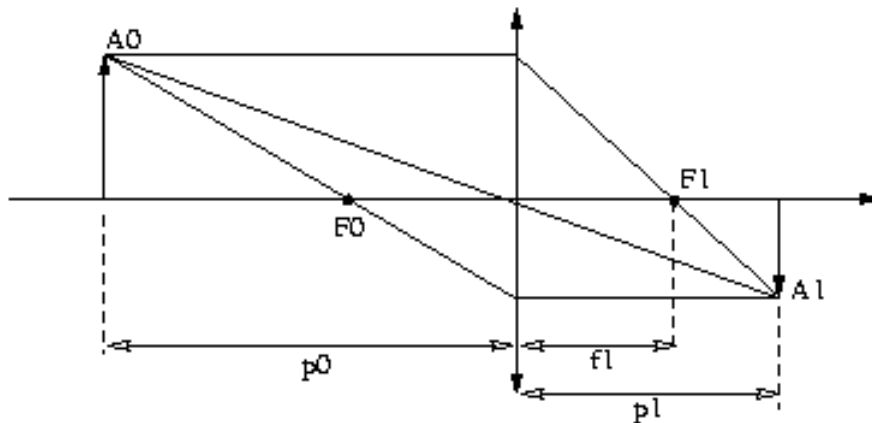


# Chapitre 1

## - Rappels -

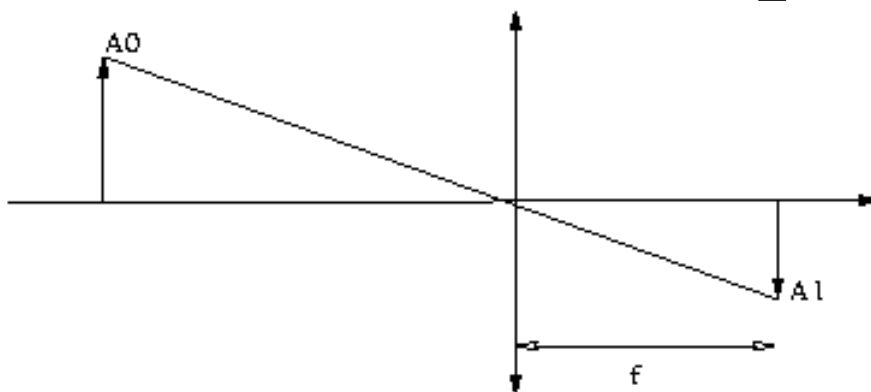
### ● Modèle mince

- approximation du modèle épais i.e plans principaux confondus
- non réaliste pour des grandes valeurs de focales



### ● Modèle sténopé

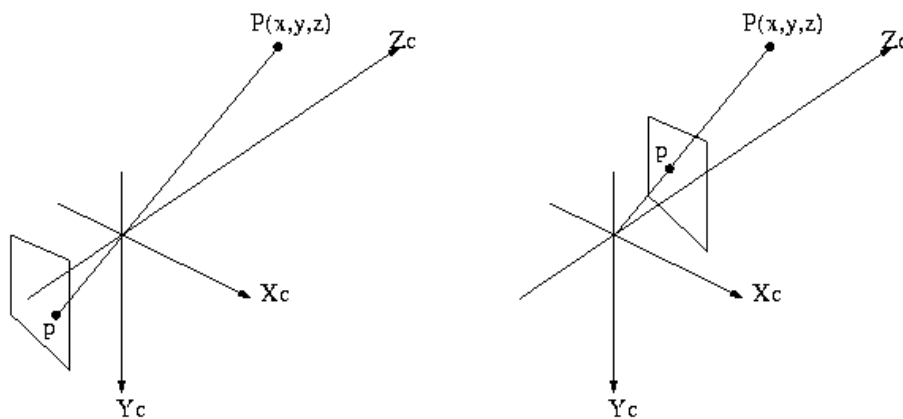
- appelé *pin-hole* car tous les rayons passent par le centre optique
- un seul rayon conservé...
- projection perspective :  $x = f \cdot \frac{X}{Z}$ ,  $y = f \cdot \frac{Y}{Z}$ ,  $z = f$



## Chapitre 3

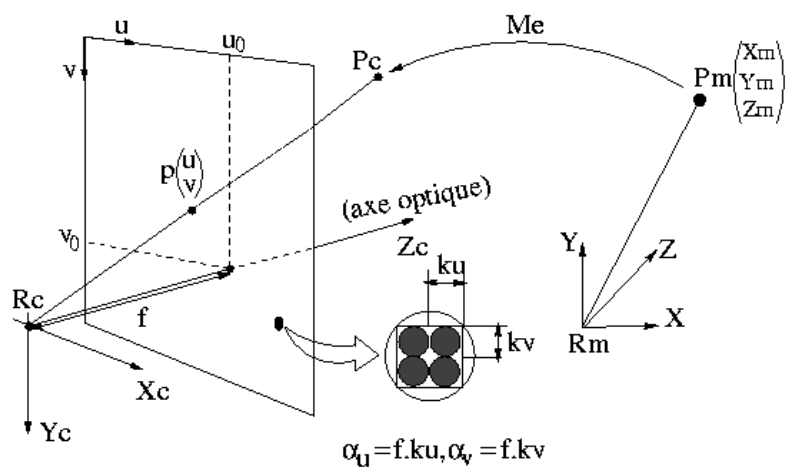
### - Modélisation de la caméra -

- **But** : 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



## Chapitre I

### - Modélisation de la caméra -

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

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

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

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

- **Paramètres intrinsèques** - estimés par calibrage

Avec  $\alpha_u = k_u \cdot f$ ,  $\alpha_v = k_v \cdot f$  :

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

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

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

## Chapitre I

### - Modélisation de la caméra -

- **Transformation mire/image** - matrice M de projection perspective

$$M = \begin{pmatrix} \alpha_u \cdot r_{11} + u_0 \cdot r_{31} & \alpha_u \cdot r_{12} + u_0 \cdot r_{32} & \alpha_u \cdot r_{13} + u_0 \cdot r_{33} & \alpha_u \cdot T_x + u_0 \cdot T_z \\ \alpha_v \cdot r_{21} + v_0 \cdot r_{31} & \alpha_v \cdot r_{22} + v_0 \cdot r_{32} & \alpha_v \cdot r_{23} + v_0 \cdot r_{33} & \alpha_v \cdot T_y + v_0 \cdot T_z \\ r_{31} & r_{32} & r_{33} & T_z \end{pmatrix}$$

$\Rightarrow$  12 paramètres à estimer

- **Principe du calibrage linéaire (1)**

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

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

$$A_{2n \times 11} \cdot X_{11} = L_{2n} \Rightarrow X = (A^t \cdot A)^{-1} \cdot A^t \cdot L$$

- décomposition de la matrice M
- mauvais conditionnement du système
- non prise en compte de la distorsion

## Chapitre I

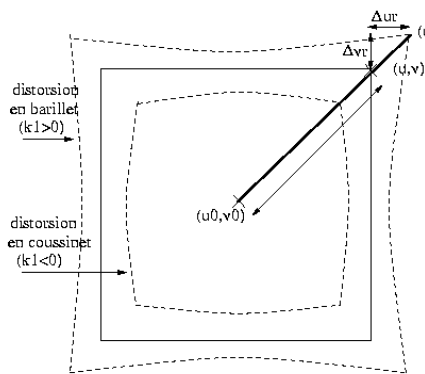
### - Modélisation de la caméra -

- **Distorsion et calibrage non linéaire**

- Modèle de distorsion **radiale** et tangentielle
- rajout d'un terme correctif dans la transformation caméra/image :

$$\begin{cases} (u_d)_i = u_i + \Delta u_i \\ (v_d)_i = v_i + \Delta v_i \end{cases} \text{ avec } \begin{cases} \Delta u_i = \Delta u_r + \Delta u_t \\ \Delta v_i = \Delta v_r + \Delta v_t \end{cases}$$

- modèle de distorsion : ordre 1 à 3... selon la focale



$$\begin{cases} \Delta u_r = (u - u_0) \cdot (k_1 \cdot r^2 + k_2 \cdot r^4 + k_3 \cdot r^6) \\ \Delta v_r = (v - v_0) \cdot (k_1 \cdot r^2 + k_2 \cdot r^4 + k_3 \cdot r^6) \end{cases}$$

$$\begin{cases} \Delta u_t = p_1 \cdot [r^2 + 2 \cdot (u - u_0)^2] + 2 \cdot p_2 \cdot (u - u_0) \cdot (v - v_0) \\ \Delta v_t = p_2 \cdot [r^2 + 2 \cdot (v - v_0)^2] + 2 \cdot p_1 \cdot (u - u_0) \cdot (v - v_0) \end{cases}$$

$$\text{avec } r = \sqrt{[(u - u_0) / \alpha_u]^2 + [(v - v_0) / \alpha_v]^2}$$

$$\Rightarrow \text{A estimer : } k_1, k_2, k_3, p_1, p_2$$

- appellation : r0t0, r1t0 (k1), r3t0 (k1, k2, k3)

- vecteur à estimer (r1t0) :

$$X = (u_0, v_0, k_1, \alpha_u, \alpha_v, \alpha^1, \beta^1, \gamma^1, t_x^1, t_y^1, t_z^1)^T$$

- optimisation non linéaire :

$$\text{Min } F(X) = \sum_{i=1}^n (v_{u_i}^2 + v_{v_i}^2)$$

avec :

$$\begin{cases} u_i + v_{u_i} = u_0 + \Delta u_i + \alpha_u \cdot \frac{r_{11} \cdot X_i + r_{12} \cdot Y_i + r_{13} \cdot Z_i + t_x}{r_{31} \cdot X_i + r_{32} \cdot Y_i + r_{33} \cdot Z_i + t_z} = P(X) \\ v_i + v_{v_i} = v_0 + \Delta v_i + \alpha_v \cdot \frac{r_{21} \cdot X_i + r_{22} \cdot Y_i + r_{23} \cdot Z_i + t_y}{r_{31} \cdot X_i + r_{32} \cdot Y_i + r_{33} \cdot Z_i + t_z} = Q(X) \end{cases}$$



## Chapitre I

### - Modélisation de la caméra -

- **Principe du calibrage : optimisation non linéaire**

⇒ linéarisation autour d'une valeur initiale  $X_0$

$$P(X) = P(X_0) + \frac{\partial P}{\partial u_0} \Delta u_0 + \frac{\partial P}{\partial v_0} \Delta v_0 + \dots + \frac{\partial P}{\partial \alpha} \Delta \alpha + \frac{\partial P}{\partial \beta} \Delta \beta + \dots + \frac{\partial P}{\partial k_1} \Delta k_1$$

⇒ système à résoudre :  $V = A \cdot \Delta X - L$

- **Résolution au sens des moindres carrés :  $\text{Min}(V^T \cdot V)$**

$$\Rightarrow \Delta X = (A^T \cdot A)^{-1} \cdot A^T \cdot L$$

$$V = \begin{pmatrix} v_{u_i} \\ v_{v_i} \\ \vdots \\ v_{u_n} \\ v_{v_n} \end{pmatrix}, L = \begin{pmatrix} u_i - P^1(X_0) \\ v_i - Q^1(X_0) \\ \vdots \\ u_n - P^n(X_0) \\ v_n - Q^n(X_0) \end{pmatrix}, \Delta X = \begin{pmatrix} u_0 - (u_0)_0 \\ v_0 - (v_0)_0 \\ \alpha_u - (\alpha_u)_0 \\ \alpha_v - (\alpha_v)_0 \\ \vdots \end{pmatrix}$$

$$A = \begin{pmatrix} \frac{\partial P^1}{\partial u_0} & \frac{\partial P^1}{\partial v_0} & \dots & \frac{\partial P^1}{\partial \alpha^1} & \dots & \frac{\partial P^1}{\partial t_z^1} & \frac{\partial P^1}{\partial \alpha^2} & \dots & \frac{\partial P^1}{\partial k_1} & \dots \\ \frac{\partial Q^1}{\partial u_0} & \frac{\partial Q^1}{\partial v_0} & \dots & \frac{\partial Q^1}{\partial \alpha^1} & \dots & \frac{\partial Q^1}{\partial t_z^1} & \frac{\partial Q^1}{\partial \alpha^2} & \dots & \frac{\partial Q^1}{\partial k_1} & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \frac{\partial Q^n}{\partial u_0} & \frac{\partial Q^n}{\partial v_0} & \dots & \frac{\partial Q^n}{\partial \alpha^1} & \dots & \frac{\partial Q^n}{\partial t_z^1} & \frac{\partial Q^n}{\partial \alpha^2} & \dots & \frac{\partial Q^n}{\partial k_1} & \dots \end{pmatrix}$$

- **Calcul des dérivées (analytiques ou numérique) :**

$$\frac{\partial P}{\partial k_1} = (u - u_0) \cdot r^2, \quad \frac{\partial P}{\partial k_2} = (u - u_0) \cdot r^4, \dots$$

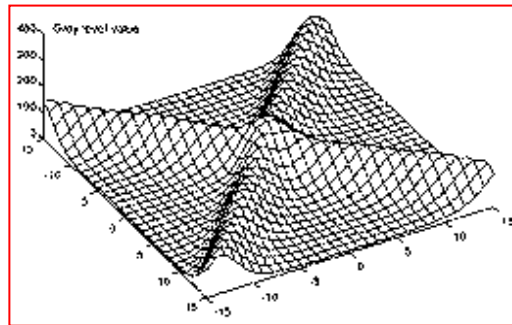
- **Attention à l'estimée initiale de  $X_0$  !!!**

# Chapitre 1

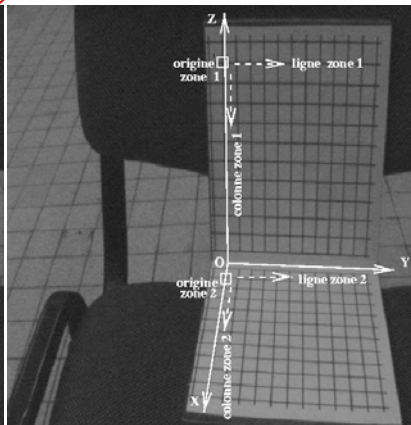
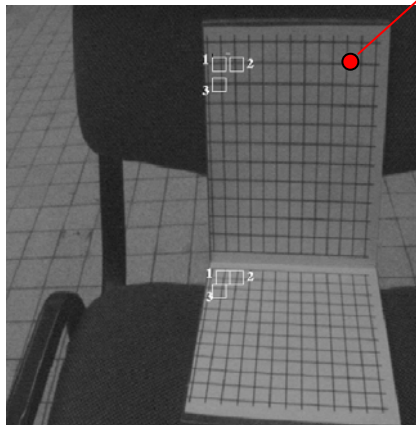
## - Modélisation de la caméra -

- **Mise en œuvre du calibrage :**

- extraction précise des amers (précision sub-pixel)  
dans les images **brutes**  $\Rightarrow$  modèle photogrammétrique



- appariements points image / points mire



292.87, 90.57, 0.0, 0.0, 0.0  
292.42, 101.59, 0.0, 0.03, 0.0  
292.33, 112.54, 0.0, 0.06, 0.0  
...

- processus multi-images

$\Rightarrow$  estimation directe de  $6.m + 4$  paramètres avec  $m$  nombre de vues totales

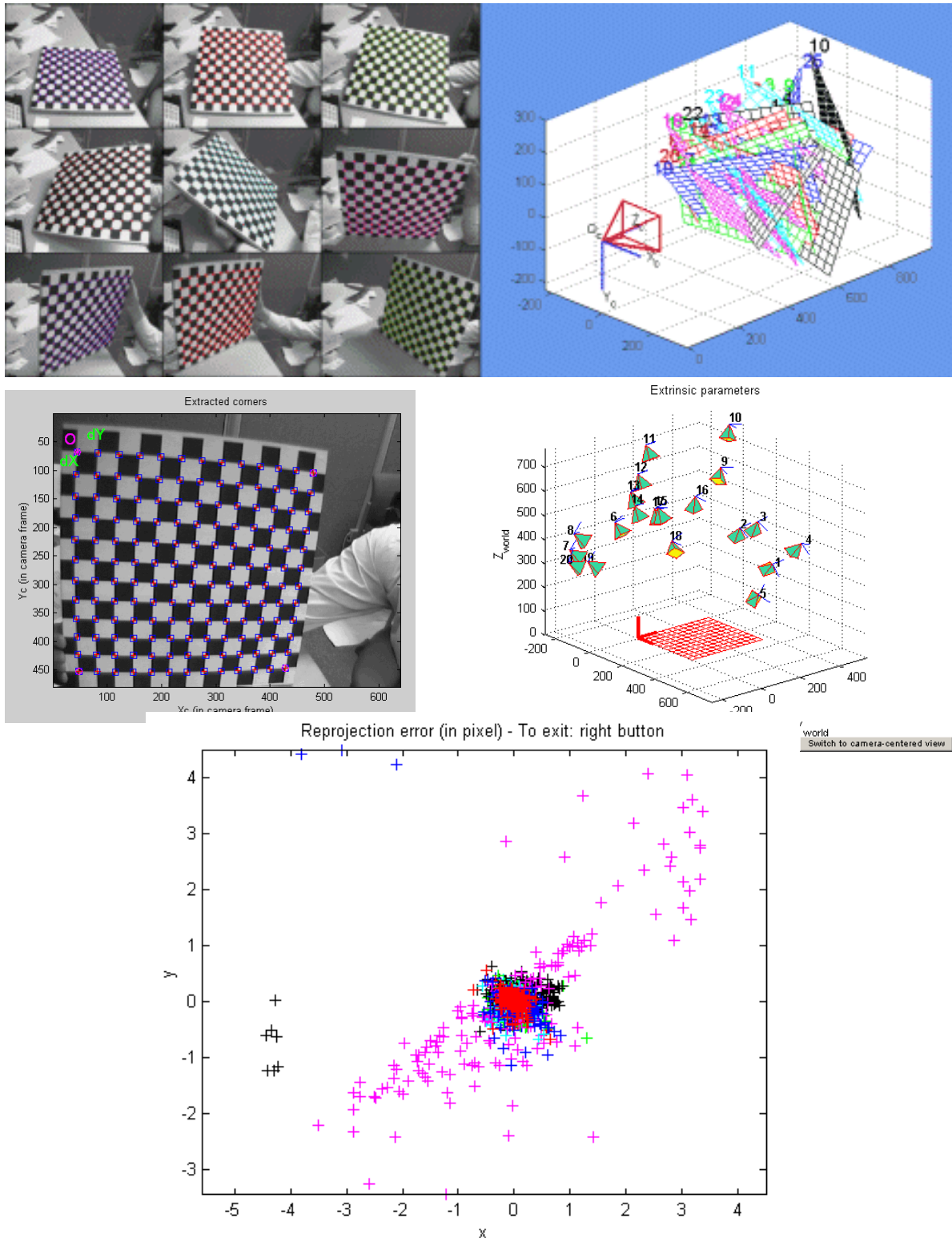
$$X = (u_0, v_0, k_1, \alpha_u, \alpha_v, \alpha^1, \beta^1, \gamma^1, t_x^1, t_y^1, t_z^1, \dots, \alpha^m, \dots, t_z^m)^T$$

- attention au positionnement de la mire - en gros parallèle au plan image
- résidu moyen de re-projection :  $\frac{1}{10} \text{ pixel}$

# Chapitre I

## - Modélisation de la caméra -

### ● *camera calibration toolbox for Matlab* :



## Chapitre I

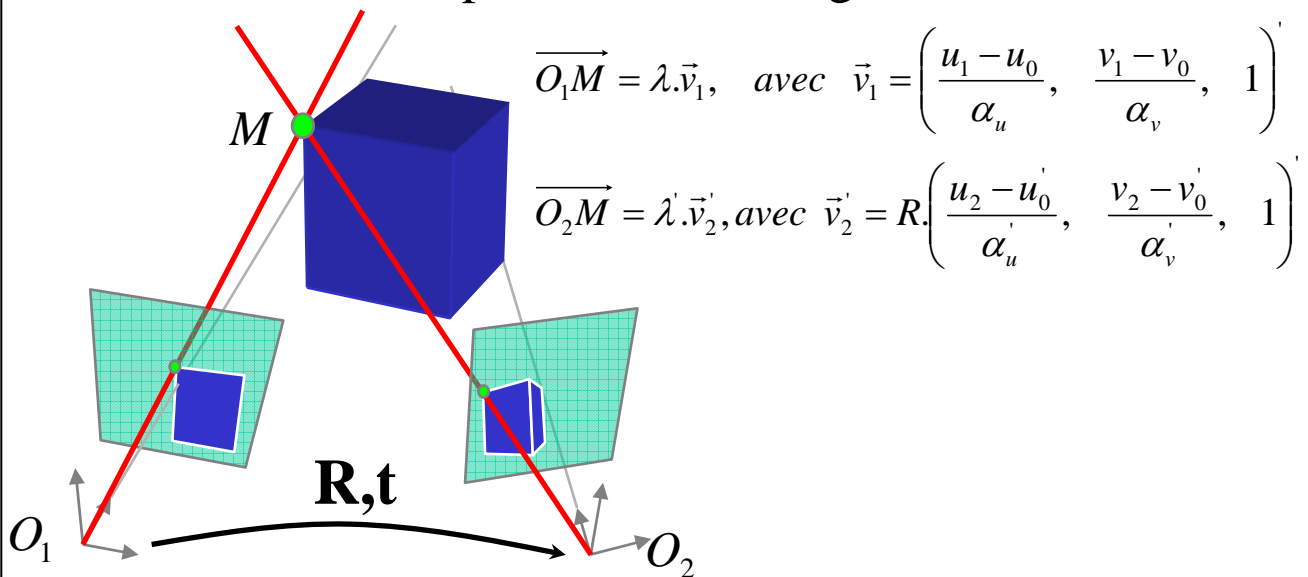
### - Stéréovision -

#### ● 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



## Chapitre I

### - Stéréovision -

- **Principe** : analyse de plusieurs images prises simultanément ou non sous des angles différents
- **Etapes** :
  - 1) étalonnage hors-ligne du système
  - 2) appariements de primitives images gauche/droite
  - 3) reconstruction 3D
- **Appariements - types de primitives**
  - denses ou éparses ?
  - **régions** - taille non conservée – segmentation
  - **segments**  $\Rightarrow$  reconstruction de segments 3D – scènes structurées – similarité de longueur, orientation,...
  - **pixels** - tous les pixels ou pixels spécifiques - information pauvre - occultations



image gauche



image droite

# Chapitre I

## - Stéréovision -

### ● Appariements de pixels :

#### ● scores paramétriques :

$$SSD(p_1, p_2) = \sum_{x=-M}^{+M} \sum_{y=-M}^{+M} [I(u_1 - x, v_1 - y) - I'(u_2 - x, v_2 - y)]^2$$

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

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

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

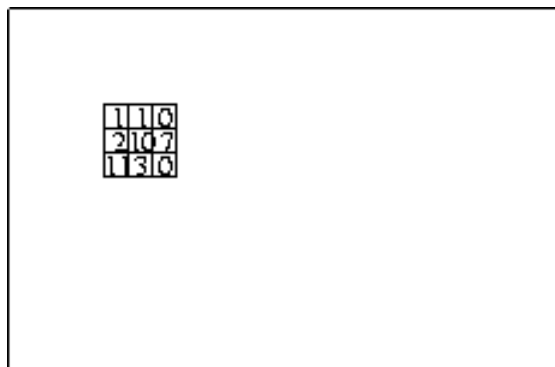


image gauche

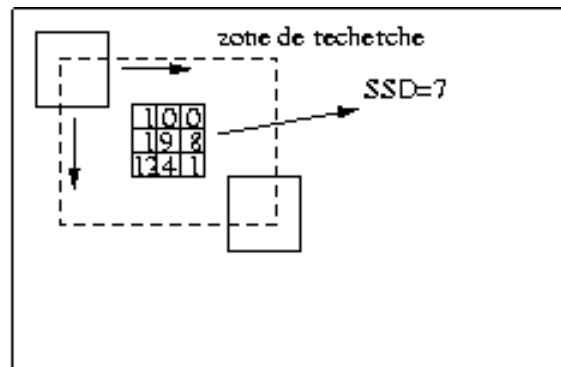
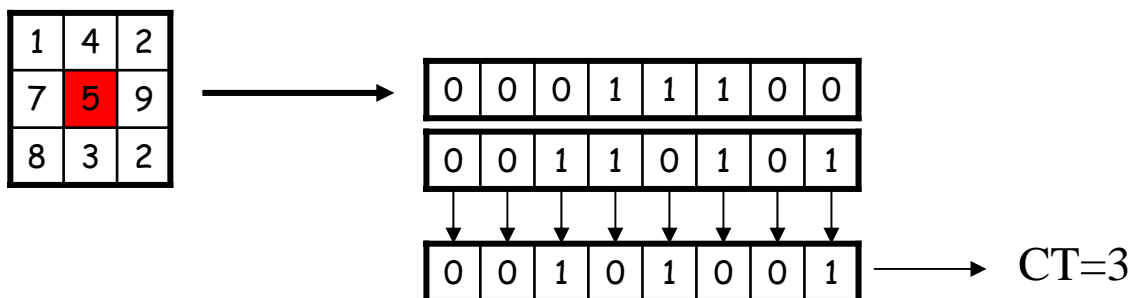


image droite

#### ● scores non paramétriques – exemple : Census Transform

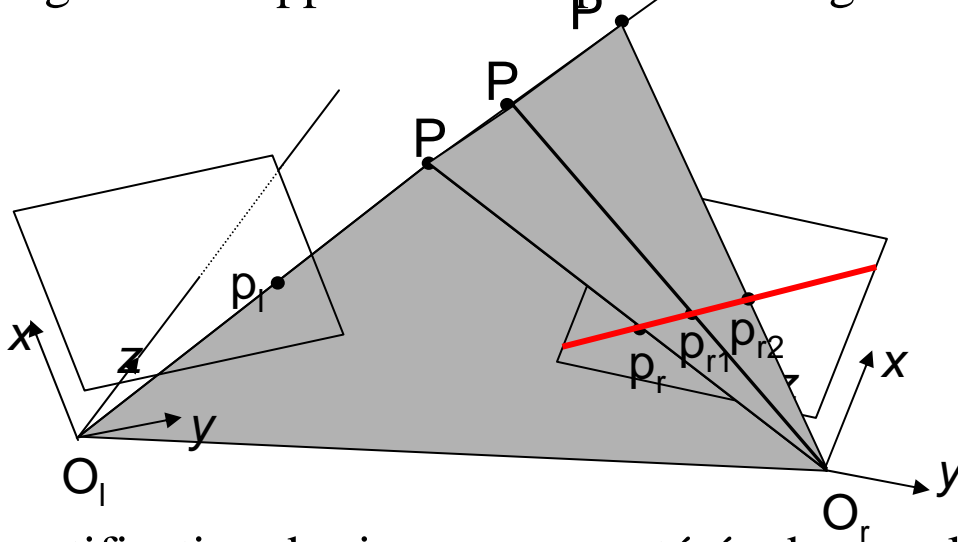


# Chapitre I

## - 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 et image confondues

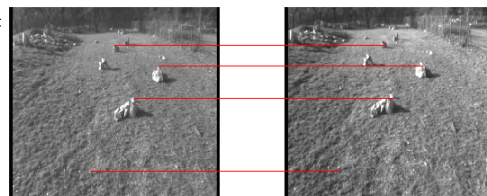
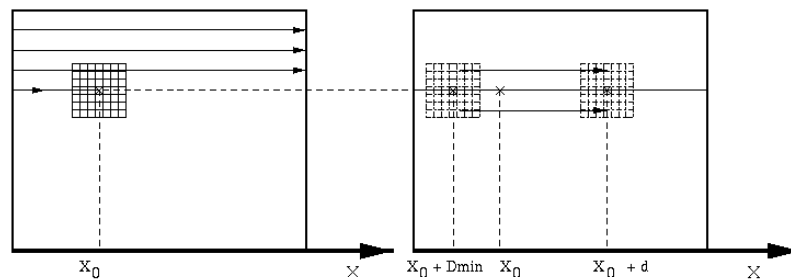


Image gauche Image droite  
Avant rectification

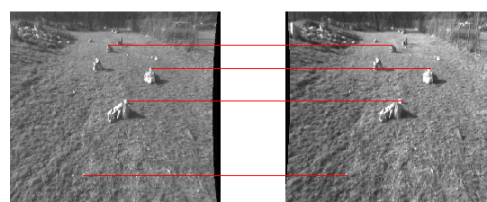


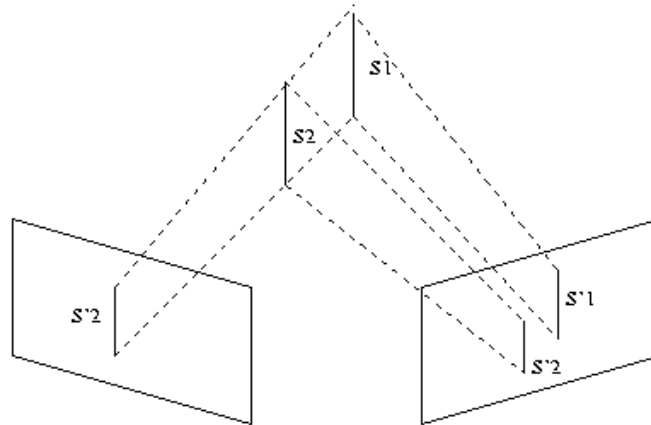
Image gauche Image droite  
Après rectification

# Chapitre I

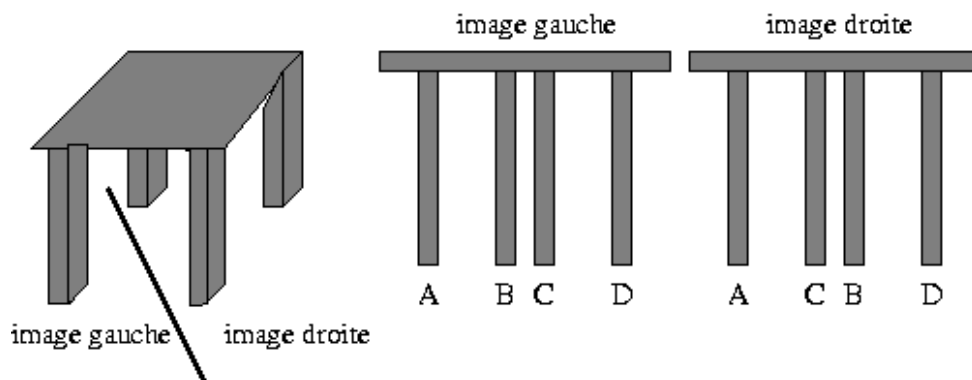
## - Stéréovision -

- **Contraintes heuristiques... parfois fausses**

- contrainte « unicité »



- contrainte « ordre »

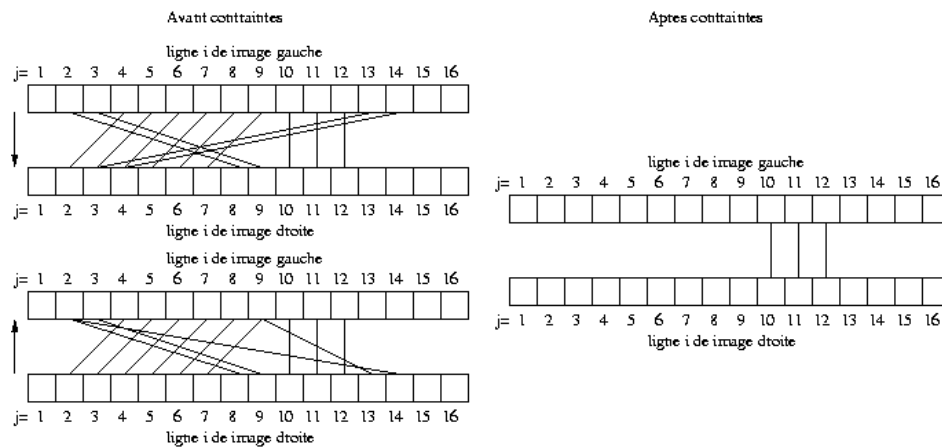




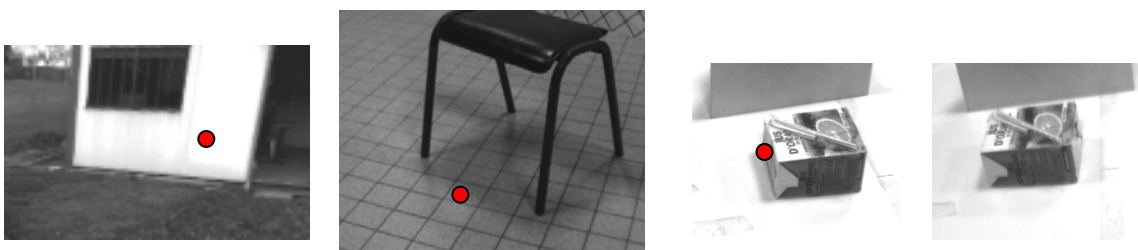
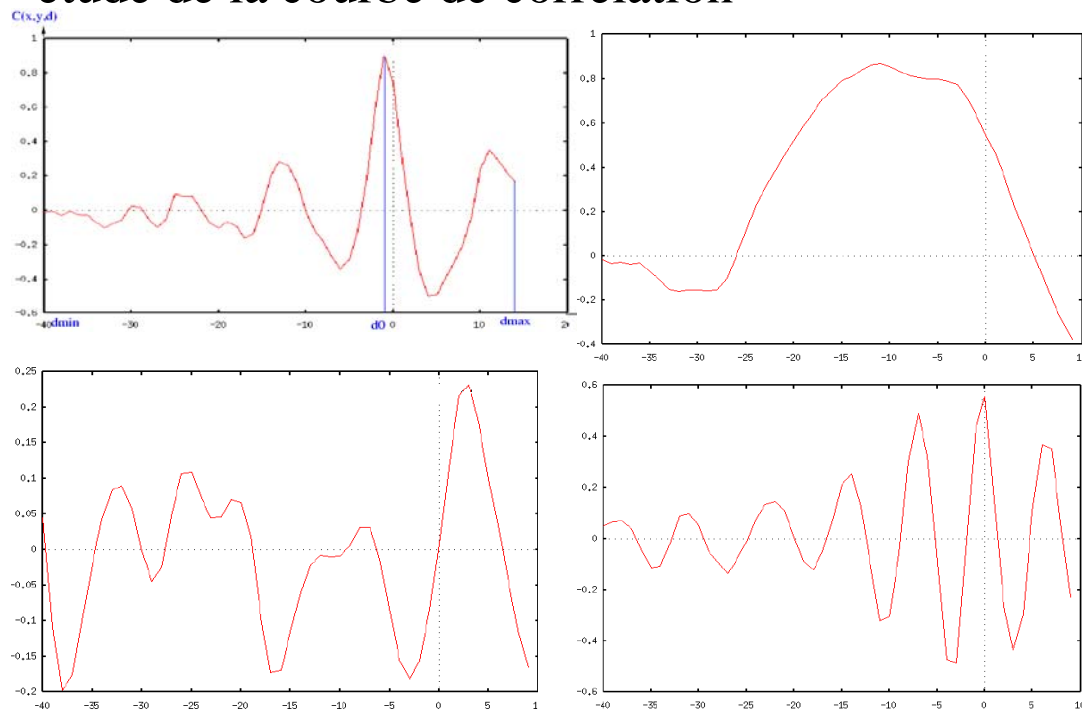
# Chapitre I

## - Stéréovision -

- **Stratégie en stéréovision dense**
- appariements bidirectionnels, contraintes heuristiques



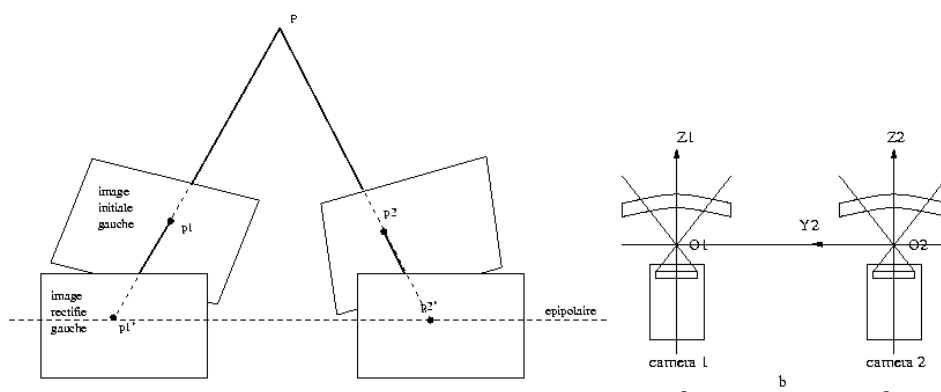
- **étude de la courbe de corrélation**



# Chapitre I

## - Stéréovision -

- **Reconstruction pour images rectifiées - trivial si disparité calculée...**



$$X_{c_1} = X_{c_2}, Y_{c_1} = Y_{c_2} + b, Z_{c_1} = Z_{c_2} \Rightarrow Z_c = \frac{b \cdot \alpha_v}{|v_1 - v_2|}$$

- **Reconstruction pour images non rectifiées – « intersection » de deux droites de vue**

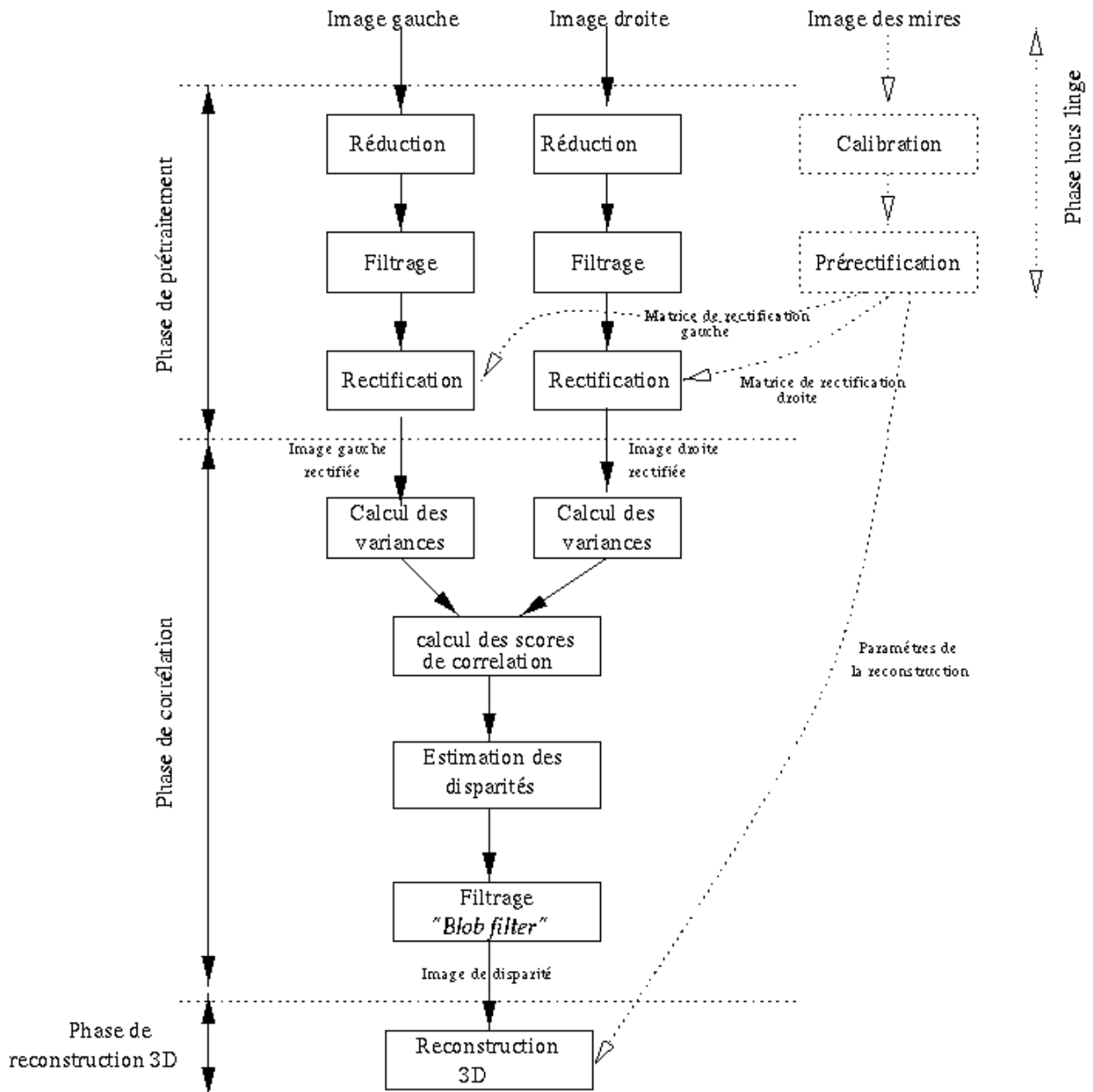
$$\left\{ \begin{array}{l} u' = \alpha_u' \cdot \frac{r_{11} \cdot X + r_{12} \cdot Y + r_{13} \cdot Z + t_x}{r_{31} \cdot X + r_{32} \cdot Y + r_{33} \cdot Z + t_z} + u_0' \\ v' = \alpha_v' \cdot \frac{r_{21} \cdot X + r_{22} \cdot Y + r_{23} \cdot Z + t_y}{r_{31} \cdot X + r_{32} \cdot Y + r_{33} \cdot Z + t_z} + v_0' \\ u = \alpha_u \cdot \frac{X}{Z} + u_0 \\ v = \alpha_v \cdot \frac{Y}{Z} + v_0 \end{array} \right.$$

- **Une 3ème caméra. Pour quoi faire ?**

# Chapitre I

## - Stéréovision -

### • Algorithme de stéréovision dense



## Chapitre I

### - Stéréovision -

- **Performances :**

- précision fonction de la distance au carré : à 10m pour une base 40cm, précision de ~30 cm
- champ de perception lié à la vergence entre caméras
- dépendance aux conditions d'éclairement...
- temps de calcul fonction de taille image, plage de disparité
- temps de calcul pour stéréovision dense

Facteur de réduction	1	2	3	4	5	6
Taille des images	576x768	266x384	192x256	144x192	114x152	96x128
Pentium1 1.5GHz	0.05Hz	0.5Hz	1.5Hz	4Hz	9Hz	12Hz

- variantes : approche pyramidale, stéréovision éparsée, stéréovision axiale

- **Exemple : aide au parking d'un véhicule routier**

[[videre.mpg](#)]

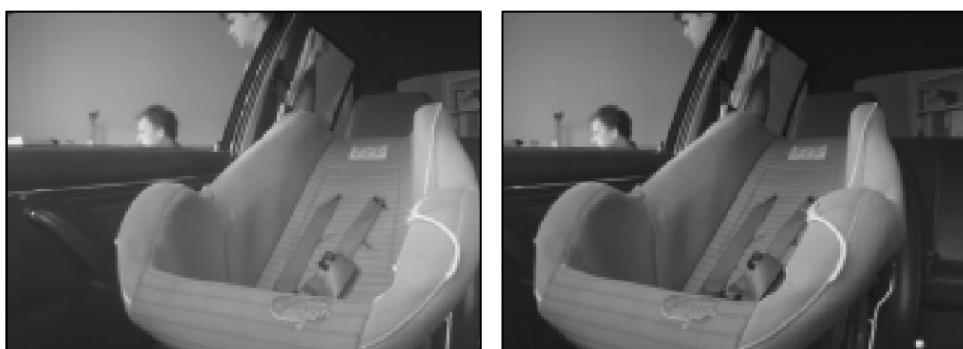
- **Exemple : navigation d'un robot mobile en milieu extérieur**

[[lama.mpg](#)] [[karma](#)]

## Chapitre I

### - Stéréovision -

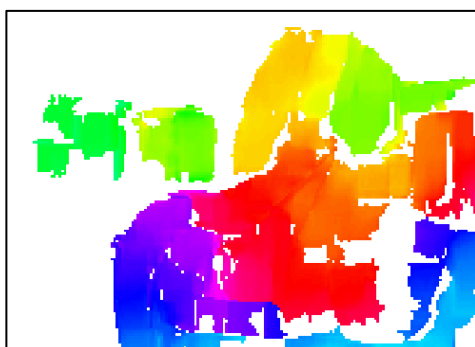
- **Exemple : stéréo-corrélation pour la supervision du cockpit d'un véhicule**



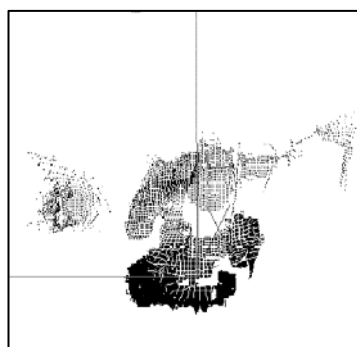
*Images originales*



*Images rectifiées*



*Image de disparité*

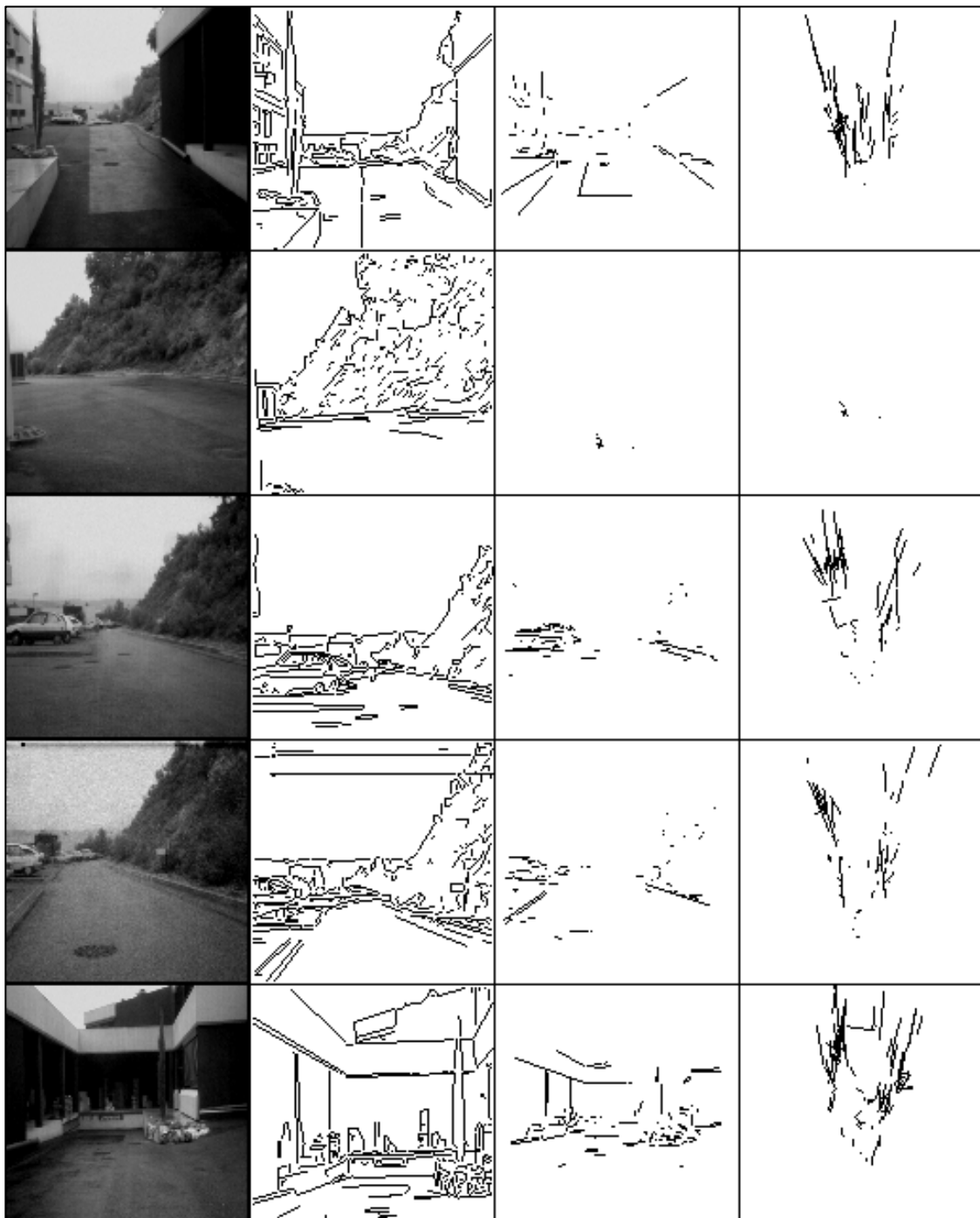


*Reconstruction 3D*

## Chapitre I

### - Stéréovision -

- **Exemple : stéréo-segment pour la reconstruction de sites semi-structurés**



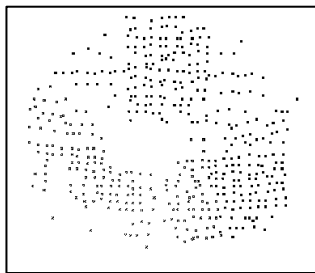
## Chapitre I

### - Stéréovision -

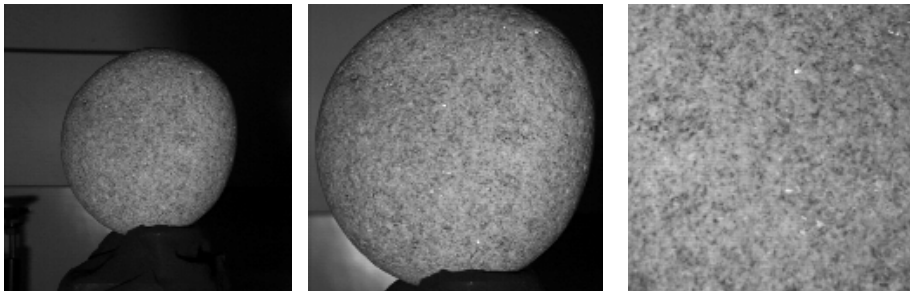
- **Exemple : stéréovision axiale pour la reconstruction d'objets**



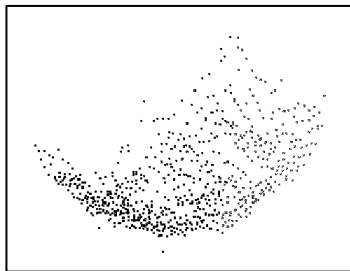
*Images originales*



*Reconstruction du cube*



*Images originales*



*Reconstruction du galet*

## Chapitre I

### - Stéréovision -

● **Exercice : stéréovision embarquée sur véhicule routier – Le système est monté à une hauteur de 80cm. Le sol est supposé plat. Les axes optiques sont orientés vers le bas d'un angle site de  $20^\circ$ . On donne : résolution de 320x240 pixels, focale de 2.1mm, base inter-caméras de 12cm, caméras en configuration épipolaire, rétine CCD de 6.4x4.8mm**

- lister les avantages/inconvénients de cette focale.
- calculer le champ de vue vertical. En déduire la largeur de la zone aveugle sur le sol ou tout obstacle est indétectable.
- montrer que  $\alpha_v \approx 105$ . En déduire la disparité à la distance minimum de perception sur le sol. Quelle disparité pour un point situé sur le sol à 3m du véhicule ? Quelle distance au sol pour une disparité minimale de 2 pixels.
- conclure sur son utilisation en détection d'obstacles.



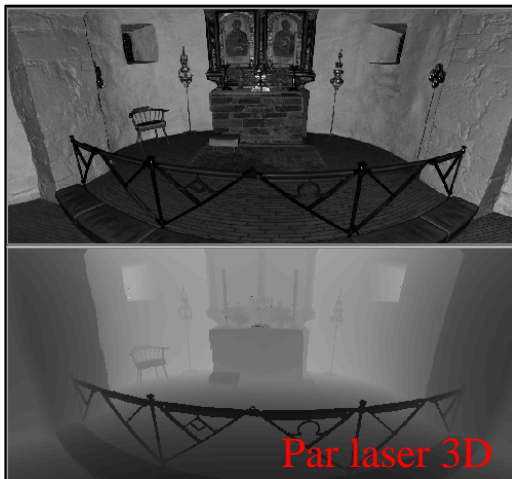
# Chapitre I

## - Capteurs actifs -

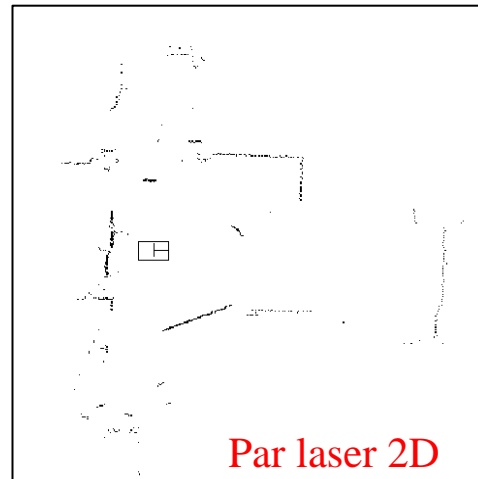
- **Principe** : à la fois émetteur et récepteur de rayonnement -mesure de distance instantanée

- **Quelques capteurs actifs :**

- capteurs RADAR
- capteurs ultrasons
- capteurs laser :

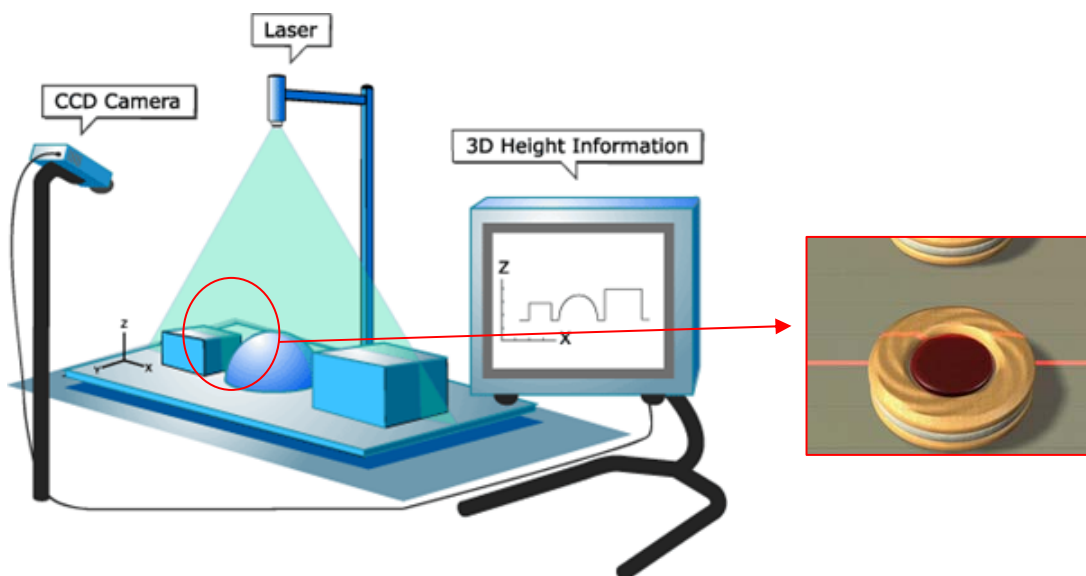


Par laser 3D



Par laser 2D

- capteurs à triangulation :



# Chapitre I

## - Capteurs actifs -

### ● Méthodes de mesure en télémétrie laser :

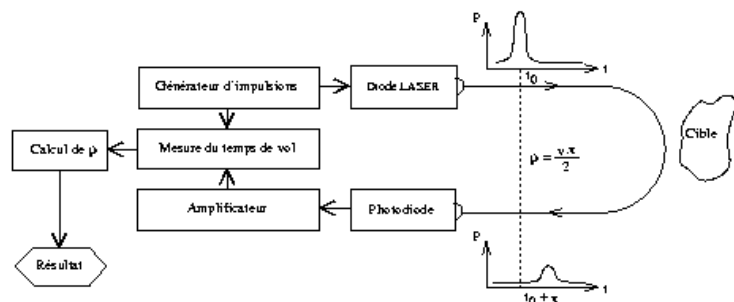
- mesure par **temps de vol** - impulsion de courte durée  
- mesure de l'intervalle de temps entre émission et réception - exemple : laser SICK

$$v \cdot \tau = 2 \cdot \rho$$

$v$  vitesse de propagation  
du signal

$\rho$  distance à l'objet

$\tau$  temps de vol

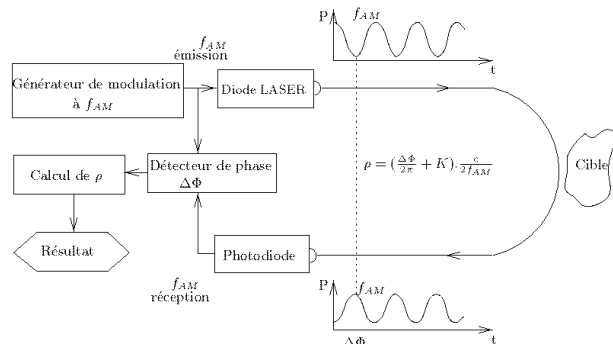


- mesure par **comparaison de phase** - faisceau laser modulé en amplitude - mesure de la différence de phase -

$$\rho(\Delta\Phi) = \frac{c \cdot \Delta\Phi}{4 \cdot \pi \cdot f_{AM}} = \frac{\lambda_{AM} \cdot \Delta\Phi}{4 \cdot \pi}$$

$\lambda_{AM}$  longueur d'onde

$c$  vitesse de propagation de l'onde



- mesure par **modulation de fréquence** - variation linéaire de la fréquence dans l'intervalle  $v \pm \Delta v / 2$

$$\rho = \frac{c \cdot f_b}{2 \cdot f_m \cdot \Delta v}$$

$f_m$  fréquence de modulation

$f_b$  signal de battement

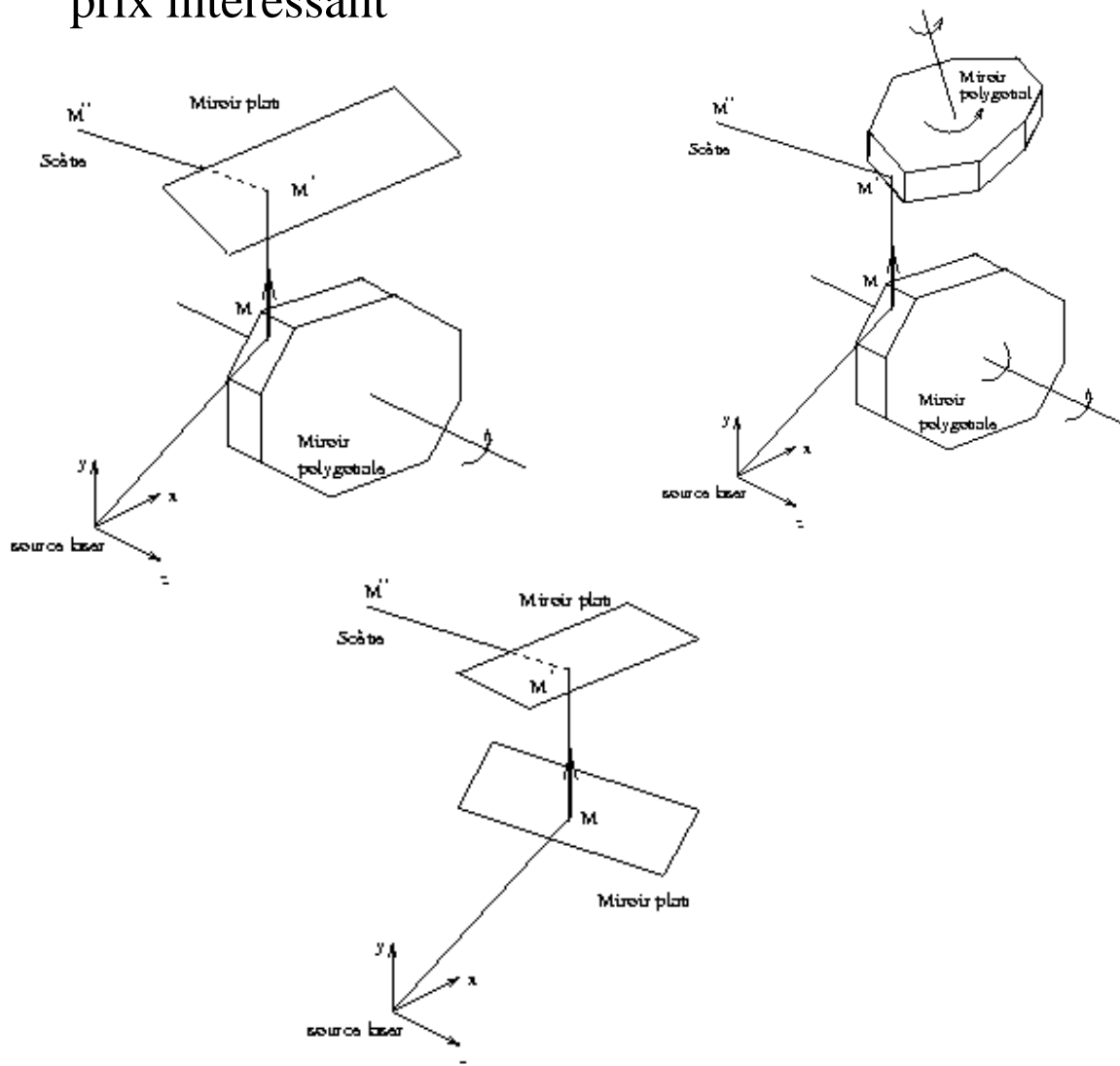
**Nécessité d'un système de déflection pour dévier le faisceau !**

## Chapitre I

### - Capteurs actifs -

#### ● **Systèmes de déflections des télémètres laser :**

- **miroir plan + miroir tournant** - le plus fréquent - miroir polygonal rotatif (balayage azimut) et miroir plan (balayage en site) - temps de latence
- **deux miroirs tournants** - temps de latence minimisés - coûteux
- **deux miroirs plans** - temps de latence importants - prix intéressant

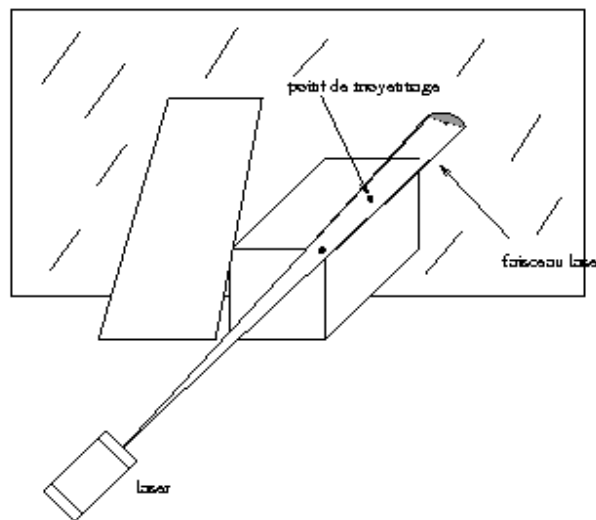


# Chapitre I

## - Capteurs actifs -

### ● Performances de la télémétrie laser :

- (+) **mesure directe de la distance** (pas de triangulation)
- (+) volume de travail important (150m max)
- (+) émetteur et récepteur proches
- (-) complexité mécanique (système de déflection)
- (-) forte imprécision de mesures (bruits, offsets électroniques, jeux mécaniques,...)
- (-) problème des points de moyennage
- (-) durée de balayage importante
- (-) sécurité oculaire
- (-) résolution liée au système de déflection



## Chapitre I

### - Capteurs actifs -

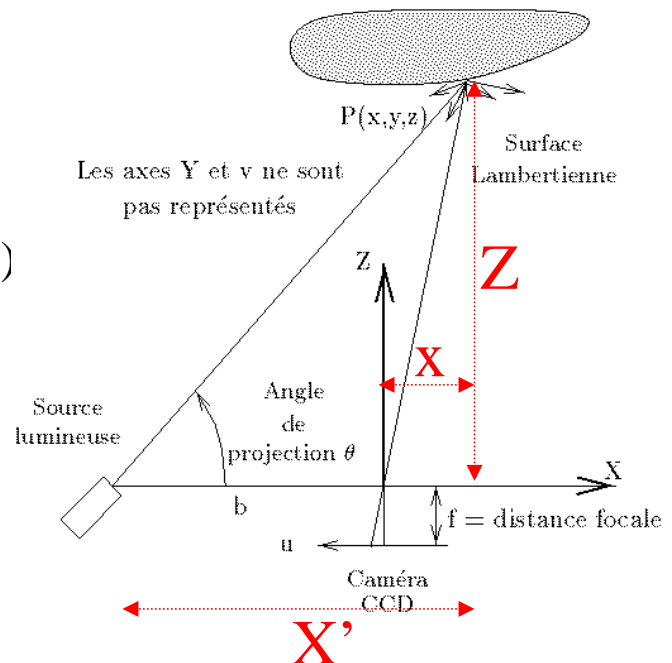
#### ● Capteurs à triangulation active :

- sur-éclairer la cible avec un faisceau de lumière et observer avec une caméra
- position relative source lumineuse/caméra connue (via un calibrage)  $\Rightarrow$  triangulation
- différente forme de lumière structurée : à points, plans, codée ou non,...

#### ● Lumière structurée en triangulation active (1) :

- projection d'un point lumineux - balayage de la scène - calibrage du faisceau - précision excellente ( $\sim 1\text{mm}$ )

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

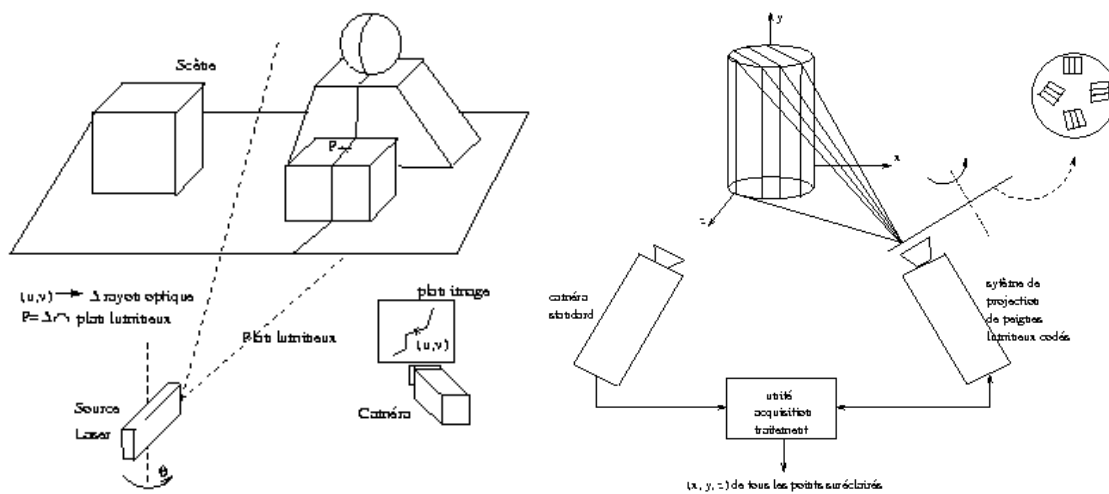


# Chapitre I

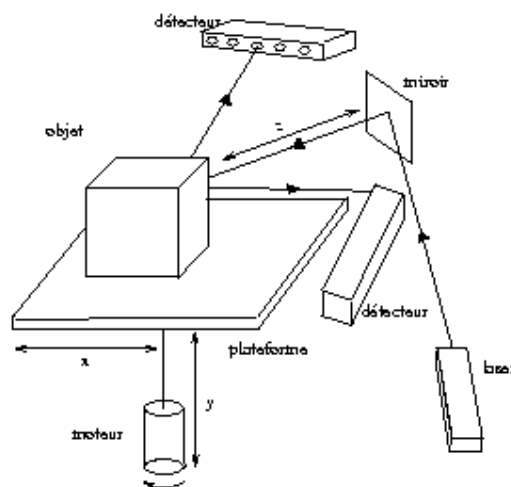
## - Capteurs actifs -

- **Lumière structurée en triangulation active (2) :**

- projection d'un plan lumineux - balayage de la scène
- calibrage du plan - reconstruction de segments
- projection de plusieurs plans (peigne) - vitesse d'acquisition ↗ - identifier les raies dans l'image



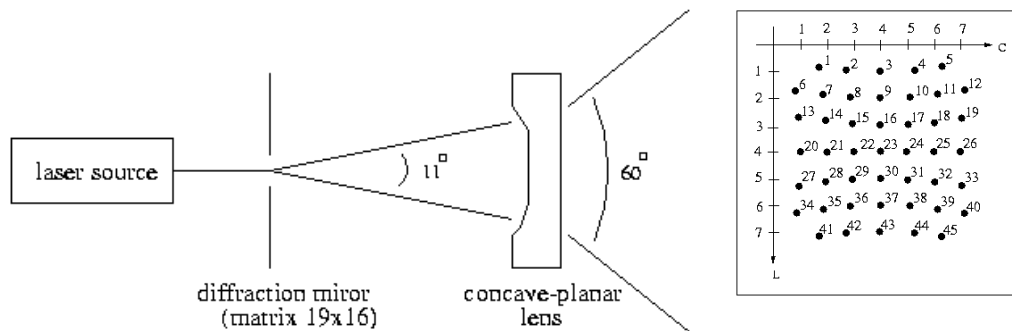
- stéréovision active - sur éclairage par spot laser - occultations - balayage de la scène



# Chapitre I

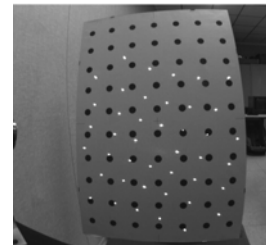
## - Capteurs actifs -

### ● Exercice : reconstruction par lumière structurée basée sur une diode laser, grille de diffraction et caméra CCD

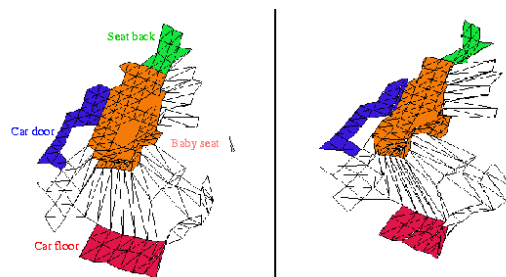


- pour calibrer, on cherche les equations des faisceaux laser dans le repère caméra à partir d'images d'une mire plane. Montrer que les équations des faisceaux sont donnés par l'expression suivante et proposer une procédure de calibrage

$$\begin{cases} x = a_k \cdot z + p_k \\ y = b_k \cdot z + q_k \end{cases}$$



- proposer une méthode pour extraire les spots avec une précision sub-pixelique
- quels contraintes pour réaliser les appariements entre faisceaux laser et spots image pour la reconstruction

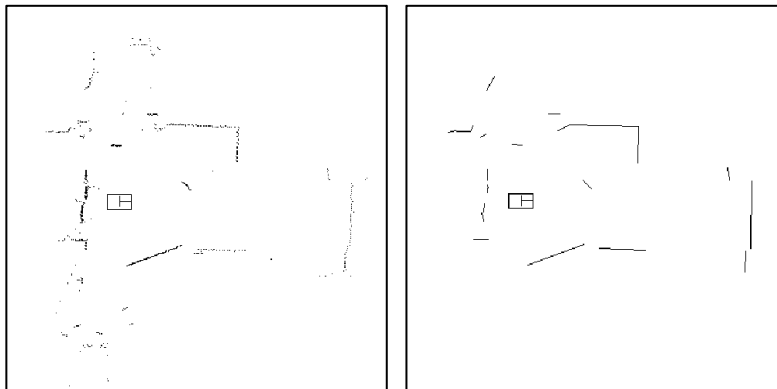


## Chapitre I

### - Capteurs actifs -

#### ● Exercice : un capteur multi-sensoriel pour la navigation d'un robot mobile en intérieur embarquant une tête stéréo et un laser SICK

- on privilégie la stéréo-segment. Comment appliquer la contrainte épipolaire ? Quelles contraintes autres ?
- Caractériser le vecteur directeur de chaque segment 3D reconstruit. On se focalise par la suite sur les segments verticaux
- Calibrer le système laser / banc stéréo
- La segmentation laser consiste à regrouper les points successifs alignés. Sachant que  $d$  est la distance orthogonale du point au segment et  $\varepsilon$  un seuil, montrer que :  $\frac{1}{n} \sum d_i^2 \leq \varepsilon$



- Comment mettre en correspondance les informations issues des deux capteurs ?



# Sommaire

- chapitre I : acquisition

- généralités
- modélisation de la caméra
- stéréovision
- capteurs actifs

- chapitre II : modélisation

- généralités
- modélisation de formes libres / structurées
- algorithmes de segmentation 3D
- modélisation incrémentale en robotique mobile

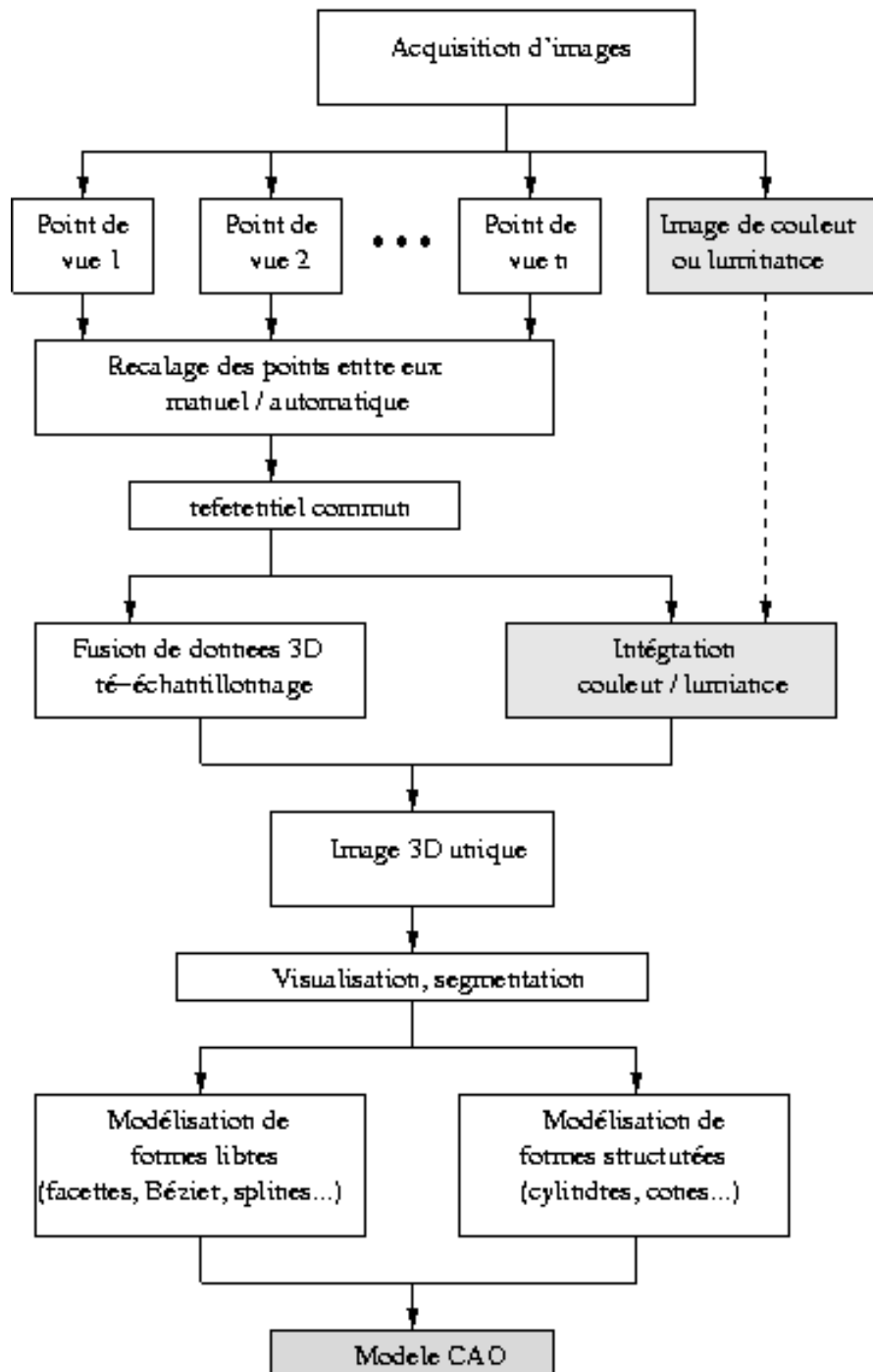
- chapitre III : localisation/reconnaissance

- outils pour la localisation
- application à la saisie d'objets
- outils pour la reconnaissance

## Chapitre II

### - Généralités -

- Démarche générique :



## Chapitre II

### - Généralités -

- **Recalage automatique**

- But : pour deux ensembles de points  $S_1$  et  $S_2$  en recouvrement, recherche de la transformation 3D  $(R,T)$  minimisant le résidu

$$QR = \underset{X_1 \in S_1}{\text{mediane}} \left[ \min_{X_2 \in S_2} \|RX_1 + T - X_2\| \right]$$

- algorithme de recalage « Iterative Closest Point »

**Début**

**Faire**

Liste\_points=Extrait\_points\_3D( $S_1$ )

Appariements=Calcule\_Appariement(Liste\_points,  $S_2$ )

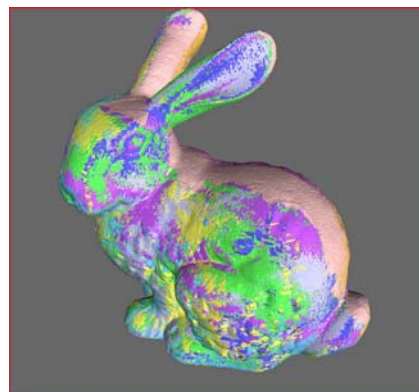
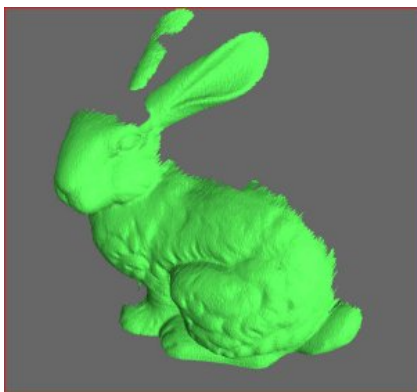
(Transfo\_3D,résidu)=Evalue\_Transf(Appariements)

Applique\_Transfo( $S_1$ ,Transfo\_3D)

Evalue\_Seuls\_Appariement(résidu)

**Tant que** (résidu < seuil) ou (Nbre\_Iteration < Iter\_Max)

**Fin**



## Chapitre II

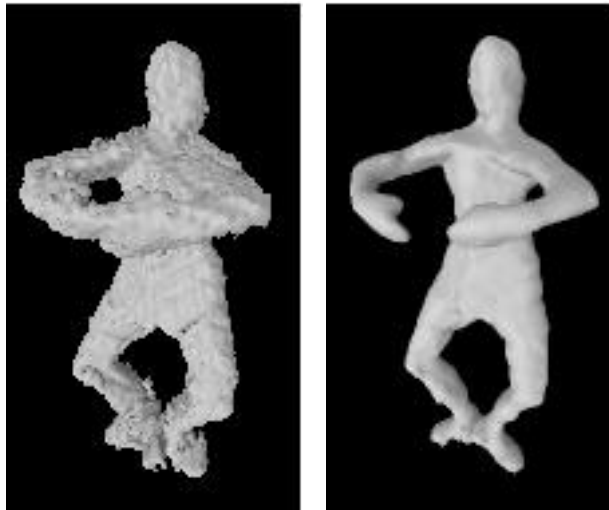
### - Généralités -

- **Intégration et ré-échantillonnage**

- But : diminuer le nombre de points

- ☞ choix de points aléatoires (échantillonnage randomisé)

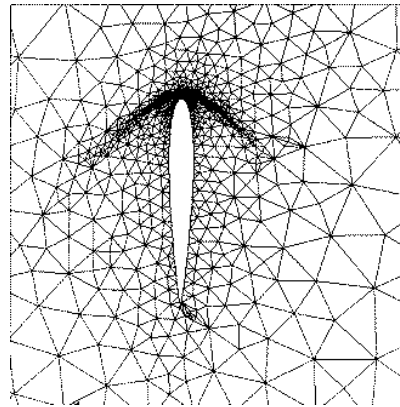
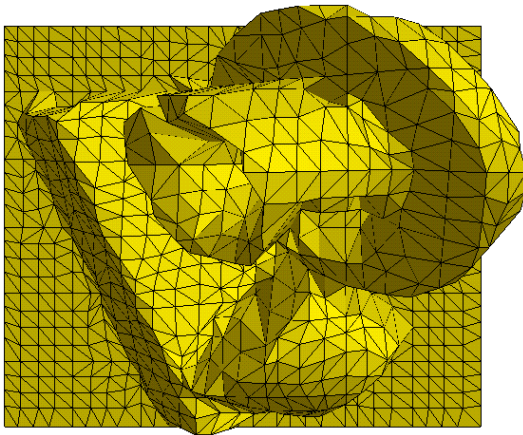
- ☞ conserver l'information utile *i.e.* supprimer les points de courbure nulle



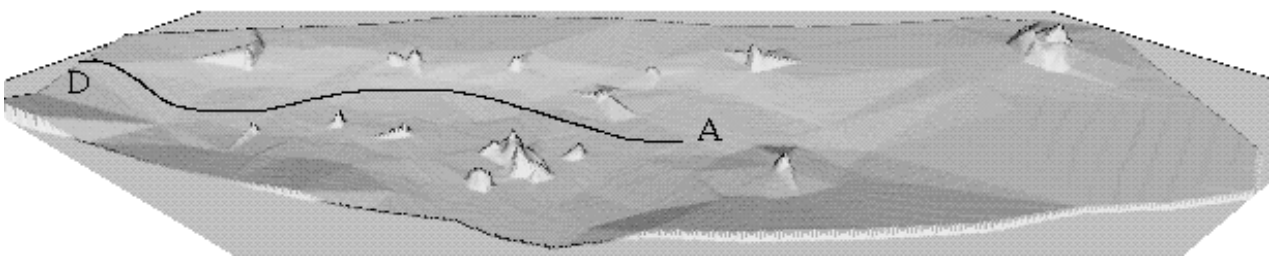
## Chapitre II

### - Modélisation de formes libres -

- **Absence de modèles mathématiques... ou intégrant un grand nombre de paramètres**
- **Modèles par facettes**
  - **maillage et facettisation** - rectangulaire, triangulaire,... - régulier ou non - hiérarchique ou non
  - adapté à tout type de scène



- un cas particulier : **le modèle numérique de terrain** (MNT) – élévation  $z$  sur une grille de mesure  $(x,y)$  - modèle dit  $2D^{1/2}$

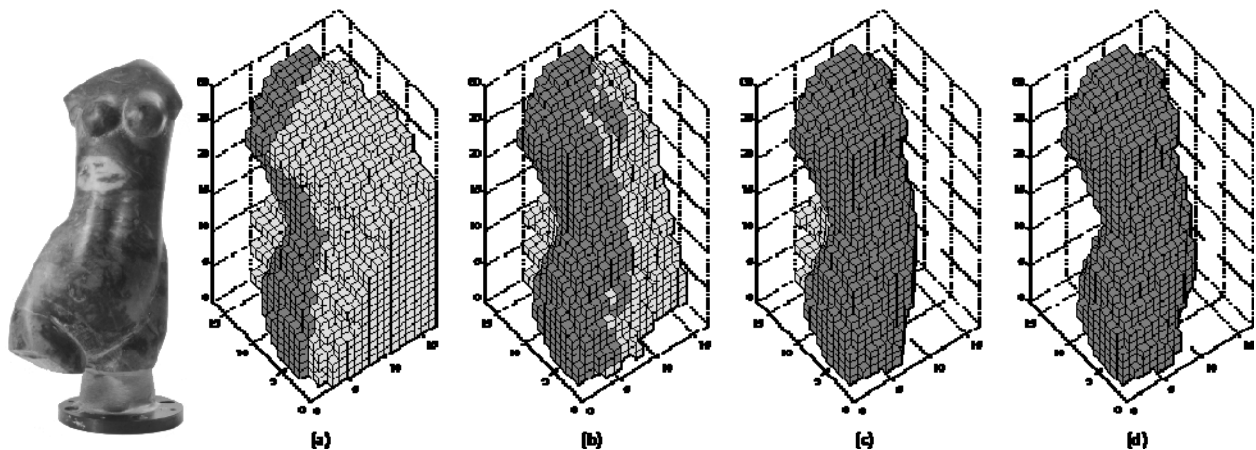


## Chapitre II

### - Modélisation de formes libres -

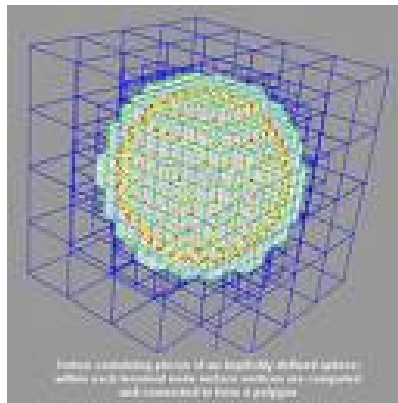
#### ● Représentation par éléments volumiques

- « **voxels** » ou représentation de l'espace par volume élémentaire - énumération des cellules occupées



*Vue dessus coté droit de la statuette -  
triangulation active - espace de travail :  
17x17x30 voxels - arête : 5mm*

- « **octrees** » ou représentation par des cubes de taille variable



- Surfaces déformables : splines, courbes de Bézier, B-spline...

## Chapitre II

### - Modélisation de formes structurées -

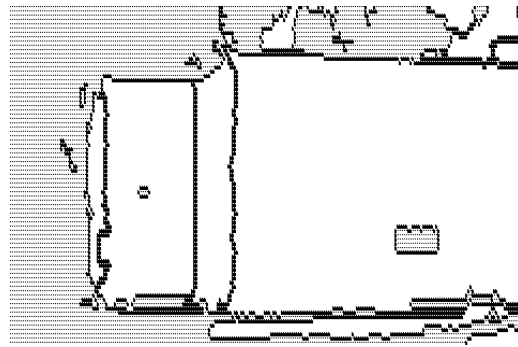
- **Modèles mathématiques simples avec peu de paramètres (< 10)**

- **Deux exemples...**

- pour un plan :

$$\text{Res} = \sum_{i=1}^m (n^t \cdot P_i - d)^2, P_G = \frac{1}{m} \sum_{i=1}^m P_i, M = \sum_{i=1}^m (P_i - P_G)(P_i - P_G)^t$$

$$\text{Min}_{n,d} \text{Res} \Rightarrow \text{Res} = n^t \cdot M \cdot n, d = n^t \cdot P_G$$



- pour une surface bi-quadratique :

$$z = a_1 + a_2 \cdot x + a_3 \cdot y + a_4 \cdot x^2 + a_5 \cdot xy + a_6 \cdot y^2$$

$$\text{Res} = \sum_{i=1}^m [z_i - a^t \cdot Q(x_i, y_i)]^2 \text{ avec } Q(x, y) = [1 \ x \ y \ x^2 \ xy \ y^2]^t$$

$$\text{Min}_a \text{Res} \Rightarrow a = M^{-1} \cdot D$$

$$\text{avec } D = \sum_{i=1}^m z_i \cdot Q(x_i, y_i), M = \sum_{i=1}^m Q(x_i, y_i) \cdot Q(x_i, y_i)^t$$



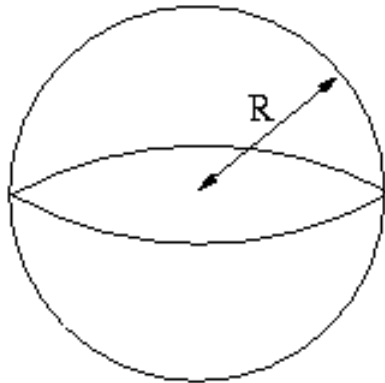
Bi-quadratic Surface  $P(x, y)$   
sample interval  $h = 0.2$

## Chapitre II

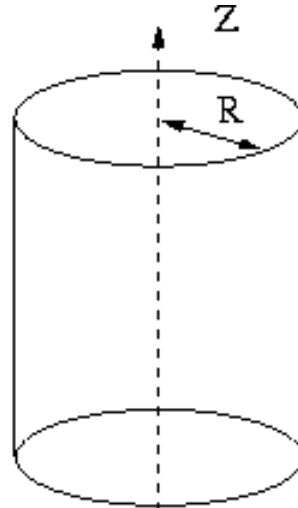
### - Modélisation de formes structurées -

#### ● Primitives géométriques simples (1) :

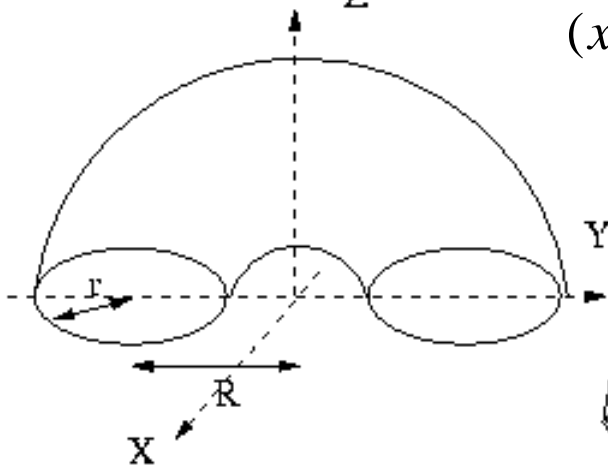
- sphères, cylindres, cônes, tores...



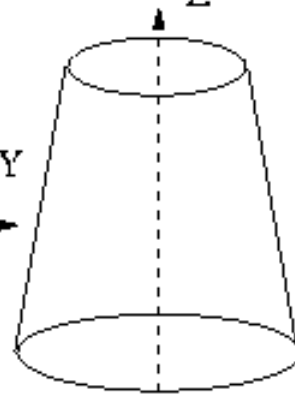
$$(x-a)^2 + (y-b)^2 + (z-c)^2 = R^2$$



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



$$(\sqrt{x^2 + y^2} - R)^2 + z^2 = r^2$$



$$x^2 + y^2 = k.z$$



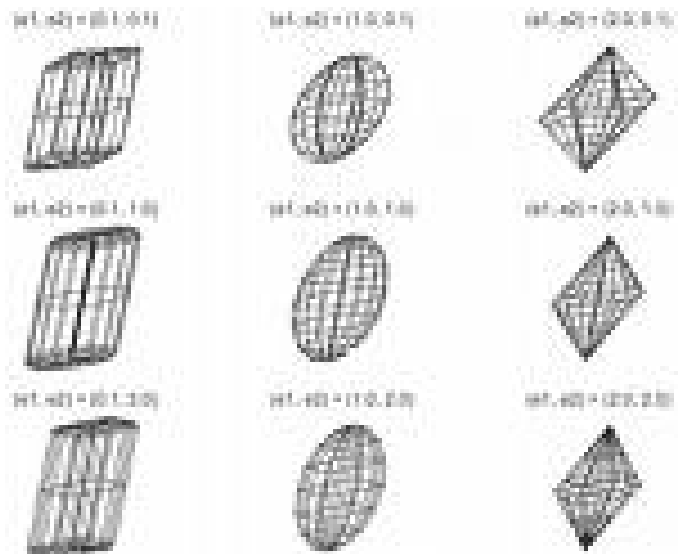
## Chapitre II

### - Modélisation de formes structurées -

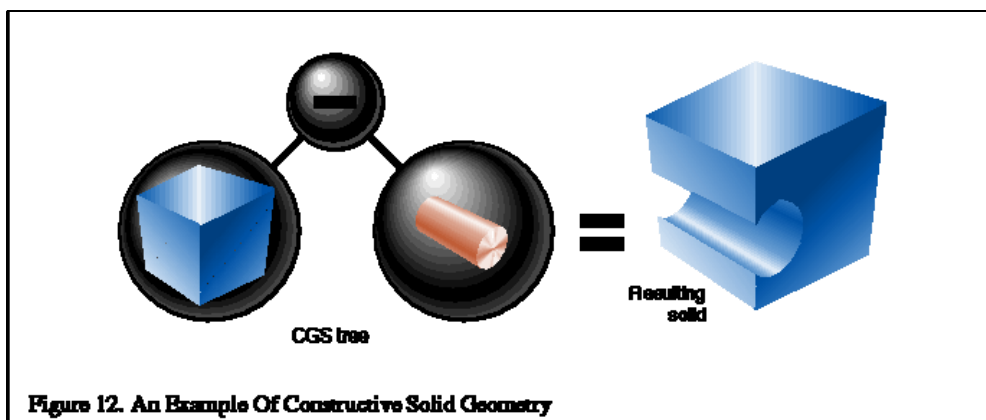
- **Primitives géométriques simples (2) :**

- superquadriques *i.e.* quadriques modifiées avec 11 paramètres indépendants

$$\left( \left| \frac{x}{a} \right|^{\frac{2}{\varepsilon_2}} + \left| \frac{y}{b} \right|^{\frac{2}{\varepsilon_2}} \right)^{\frac{\varepsilon_2}{\varepsilon_1}} + \left| \frac{z}{c} \right|^{\frac{2}{\varepsilon_1}} = 1$$



- structuration par le biais d'arbres de « Constructive Solid Geometry » - feuilles = éléments géométriques – noeuds = opérations ensemblistes



## Chapitre II

### - Segmentation -

#### ● **Par croissance de régions**

- But : extraire les primitives d'un type donné
- détermination des sauts/marches de profondeur (ordre 0), discontinuité de normale (ordre 1), discontinuité de courbure (ordre 2)
- principe n°1 ou niveau bas :

##### **Début**

Init.

##### **Faire**

Nb\_points=Nb\_points+1

Mise à jour paramètres modèle

**Tant que** (discontinuité=FALSE) ou (résidu<seuil)

##### **Fin**

- principe n°2 ou niveau intermédiaire :

##### **Début**

Sélection de régions élémentaires R (Type, Nb\_points)

Estimation des paramètres

Tri des régions R

Construction liste des régions connexes L\_R

##### **Faire**

Examiner résidu pour R fusionnée avec L\_R

**Tant que** fusion=TRUE

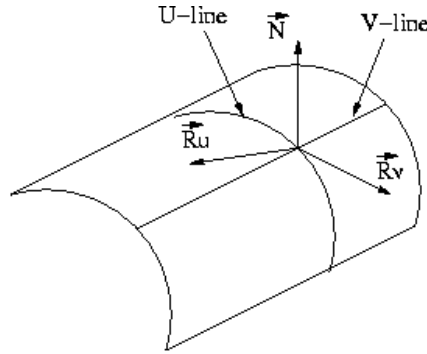
##### **Fin**

## Chapitre II

### - Segmentation -

#### ● La normale

- adaptée aux surfaces planes
- calculée après facetisation, sinon par tangentes locales



$$\vec{N} = \vec{R}_u \wedge \vec{R}_v, \text{ avec } \vec{R}_u = (x_u, y_u, z_u) \text{ et } \vec{R}_v = (x_v, y_v, z_v),$$

$$x_u = \frac{\partial x}{\partial u}, y_u = \frac{\partial y}{\partial u}, z_u = \frac{\partial z}{\partial u} \dots$$

- calcul à l'aide des masques de Sobel, Kirsch

#### ● La courbure (1)

- adaptée aux surfaces de degré supérieur
- calcul des rayons de courbure principaux R1 et R2

$$(LN - M^2)R^2 + (E - 2FM + GL)R + (EG - F^2) = 0$$

$$\begin{bmatrix} E & F \\ F & G \end{bmatrix} = \begin{bmatrix} \vec{R}_u \cdot \vec{R}_u & \vec{R}_u \cdot \vec{R}_v \\ \vec{R}_u \cdot \vec{R}_v & \vec{R}_v \cdot \vec{R}_v \end{bmatrix}, \begin{bmatrix} L & M \\ M & N \end{bmatrix} = \begin{bmatrix} \vec{R}_{uu} \cdot \vec{N} & \vec{R}_{uv} \cdot \vec{N} \\ \vec{R}_{uv} \cdot \vec{N} & \vec{R}_{vv} \cdot \vec{N} \end{bmatrix}$$

$$\vec{R}_{uu} = (x_{uu}, y_{uu}, z_{uu}), \vec{R}_{vv} = \dots, x_{uu} = \frac{\partial^2 x}{\partial u^2}, y_{uu} = \frac{\partial^2 y}{\partial u^2} \dots$$

## Chapitre II

### - Segmentation -

#### • La courbure (2)

- courbure de Gauss K et moyenne H :

$$K = \frac{1}{R_1 \cdot R_2}, H = \frac{\frac{1}{R_1} + \frac{1}{R_2}}{2}$$

$k > 0, H < 0$

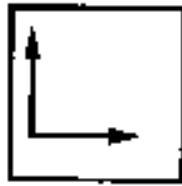


Dôme

$k > 0, H > 0$



Creux



Plat  $k = H = 0$



Col

$H = 0, k < 0$



Creste  $H < 0, k = 0$



Col\_croûte

$H < 0, k < 0$



Vallée



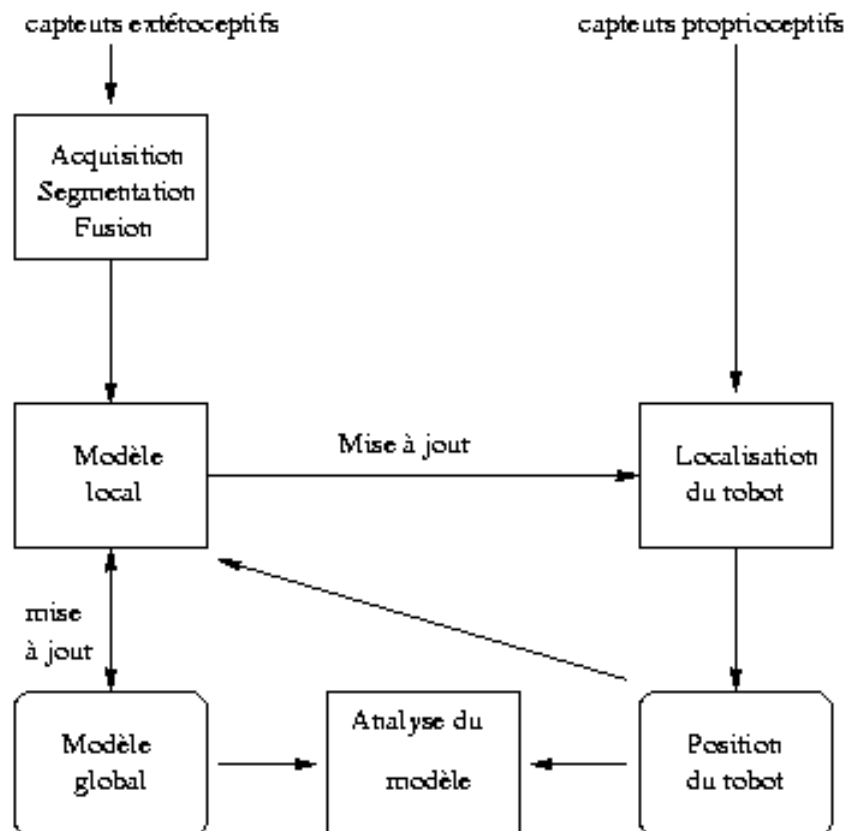
Col\_vallée

$H > 0, k < 0$

## Chapitre II

### - Modélisation incrémentale -

- **Problématique** : besoin de complétude - fusion de perceptions partielles depuis différents points de vue/capteurs

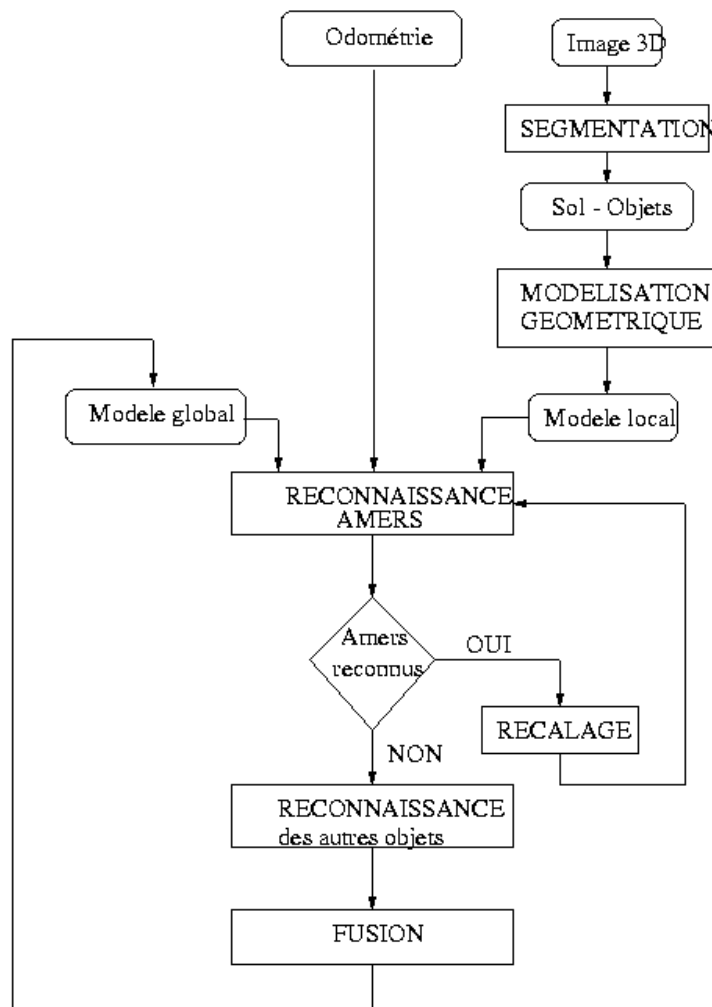


- modèle locale - extraction des primitives - segmentation - fusion multi-sensorielle (éventuellement)
- modèle global - intégration des modèles locaux - rajout/fusion de primitives
- localisation - mise à jour de la position du robot  $\Rightarrow$  correction de l'odométrie
- analyse du modèle - adapté le modèle à la tâche (modèle de navigation, de terrain)

## Chapitre II

### - Modélisation incrémentale -

- **Exemple** : modélisation d'un environnement d'extérieur pour la navigation (1)



- **un « bon » amer** - visible (non occulté) - non ambigu (en contact avec le sol) - précis (sommet)
- **le sol** - surface(s) majoritaire(s) dans la scène - normales équivalentes
- **4 fonctionnalités** - segmentation 3D, modélisation, reconnaissance/localisation, fusion

## Chapitre II

### - Modélisation incrémentale -

- **Exemple** : modélisation d'un environnement d'extérieur pour la navigation (2)

- **segmentation** sol/objets :

- ☞ extraction du sol - normale en chaque point - histogramme bi-variable des normales (tableau d'accumulation) -

- ☞ extraction des objets - discontinuité de profondeur - détection et fermeture des contours -

- ☞ identification des objets - grossissement de régions

- ☞ sélection des « vrais » amers

- ☞ modélisation des amers - calcul d'ellipsoïdes - matrice d'inertie - histogramme cumulé

⇒ *modèle locale*



- **reconnaissance** des amers - distance/intersection entre ellipsoïdes locale/globale ⇒ appariements, localisation (moindres carrés récurifs)

- **fusion** - mise à jour des sommets/matrices d'inertie des amers déjà existants - rajout des nouveaux amers

- problème des points de moyennage (laser)

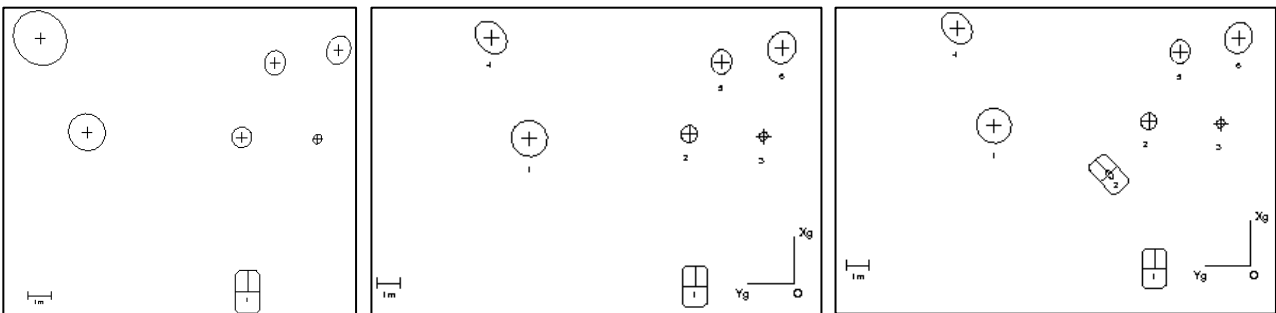
## Chapitre II

### - Modélisation incrémentale -

- **Exemple** : modélisation d'un environnement d'extérieur pour la navigation (3)



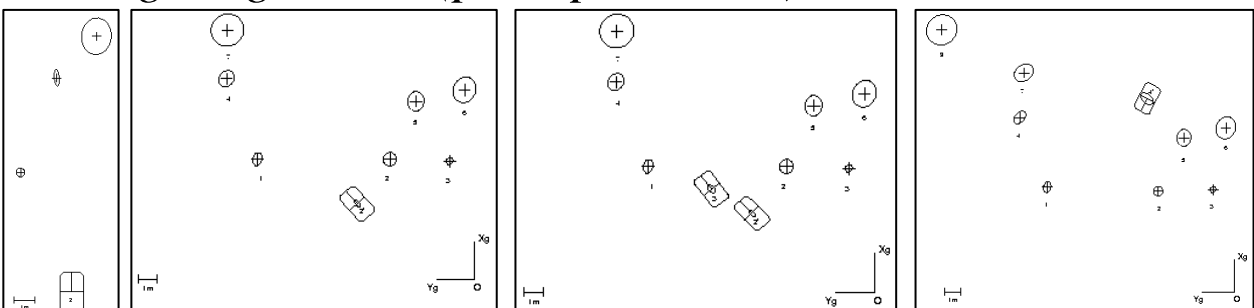
*Image segmentée (perception n° 1)*



*Amers identifiés - carte globale - déplacement du robot*



*Image segmentée (perception n° 2)*



*Amers identifiés - carte globale - déplacement du robot  
- carte globale (4 perceptions)*



# Sommaire

- chapitre I : acquisition

- généralités
- modélisation de la caméra
- stéréovision
- capteurs actifs

- chapitre II : modélisation

- généralités
- techniques
- stratégie incrémentale

- chapitre III : localisation/reconnaissance

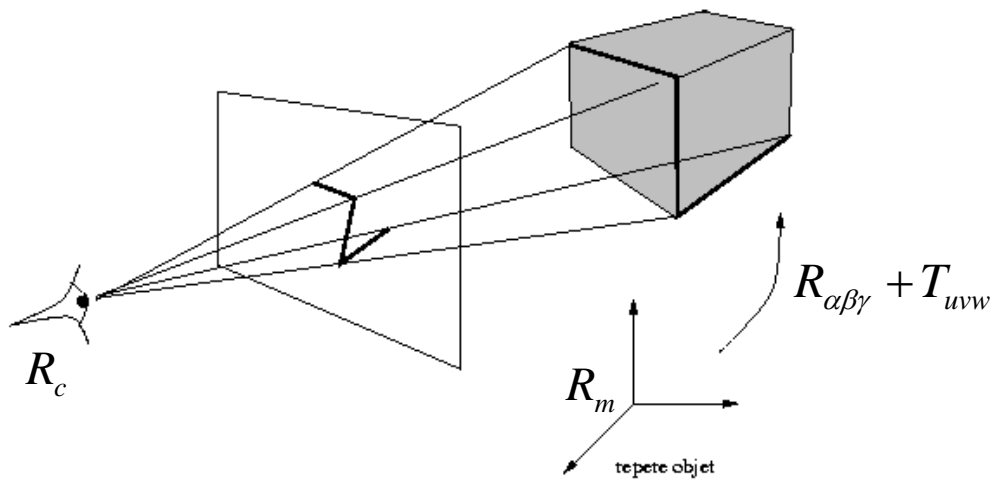
- Localisation
- Reconnaissance
- Application à la manipulation d'objets

## Chapitre III

### - Localisation -

● **But** : détermination des paramètres extrinsèques connaissant :

- les paramètres intrinsèques
- un ensemble de primitives 3D modélisant l'objet
- un ensemble d'appariements entre primitives modèle et visuelles



- Classification des approches en fonction :
  - du modèle de projection manipulé : orthographique, **perspectif**...
  - de la nature des primitives appariés : points, **droites**, cercles
  - de la méthode de résolution : analytique (P3P, P3L, Dementhon), **numérique**

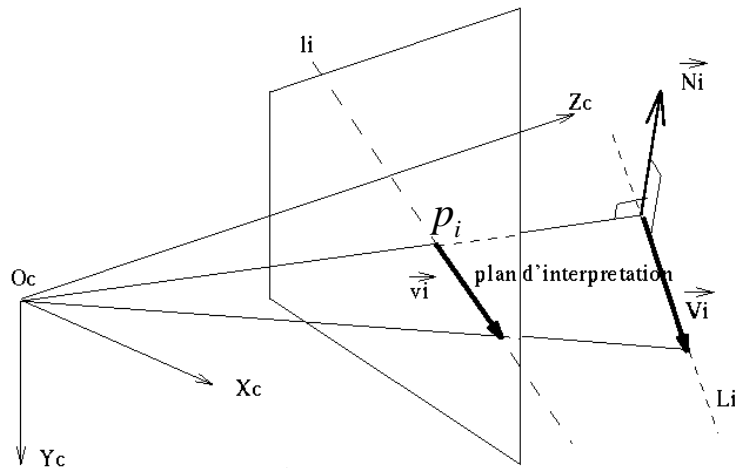
## Chapitre III

### - Localisation analytique -

#### ● Perspective-3-Lignes (P3L) :

- principe :

$$F(X, \vec{N}_i, P_i) = \vec{N}_i \cdot [R_{\alpha\beta\gamma} P_i + T_{uvw}], \vec{N}_i = \frac{O_c P_i \wedge \vec{v}_i}{\|O_c P_i \wedge \vec{v}_i\|}$$



- résolution : 
$$\begin{cases} R_{\alpha\beta\gamma} \cdot \vec{V}_i \cdot \vec{N}_i = 0 & (i = 1, \dots, 3) \\ \vec{N}_i \cdot (R_{\alpha\beta\gamma} \cdot P_i + T_{uvw}) = 0 \end{cases}$$

**8 solutions théoriques  $\Rightarrow$  3 solutions plausibles !**

# Localisation 3D

## - Localisation analytique -

### ● Perspective-3-Lignes : exemple

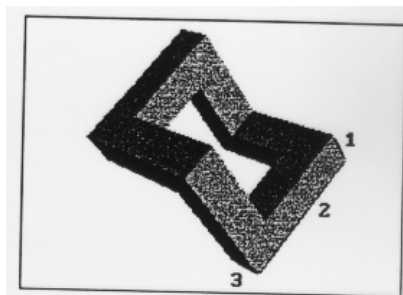


Figure 3a : Segments sélectionnés.

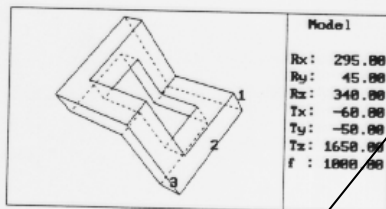
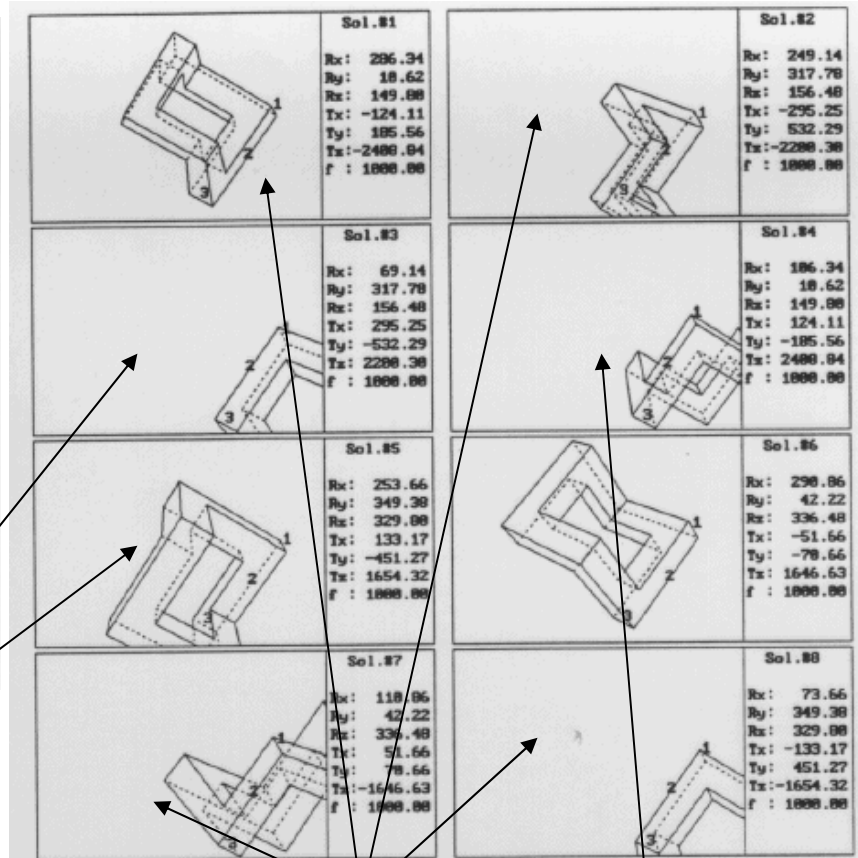


Figure 3b : Arêtes correspondantes.

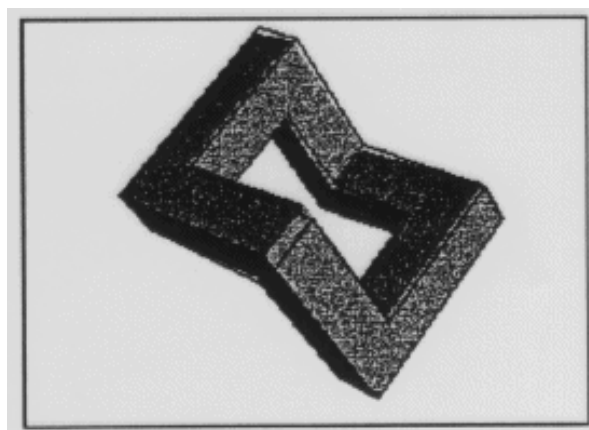


arêtes non visibles

$$Tz < 0$$

recouvrement

solution après  
filtrage



## Chapitre III

### - Localisation numérique -

#### ● **Perspective-n-Lignes:** résolution numérique

● critère à minimiser :  $\varepsilon = \sum_{i=1}^n F(X, \vec{N}_i, P_i)^2$

● résolution :  $\underset{X}{Min} (V^T V), V = A \cdot \Delta X - L, \Delta X = (A^t A)^{-1} \cdot A^t \cdot L$

$$A = \begin{pmatrix} \frac{\partial F_i}{\partial \alpha} & \dots & \frac{\partial F_i}{\partial T_w} \\ \vdots & \ddots & \vdots \\ \frac{\partial F_n}{\partial \alpha} & \dots & \frac{\partial F_n}{\partial T_w} \end{pmatrix}, L = \begin{pmatrix} -F_i(X_0) \\ \vdots \\ -F_n(X_0) \end{pmatrix}$$

● calcul des dérivées :  $\frac{\partial F_i}{\partial \varphi} = [N_i] \cdot \left[ R_\gamma \frac{\partial R_\beta}{\partial \varphi} R_\alpha \cdot P_i \right]$

$$\frac{\partial R_\alpha}{\partial \alpha} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{pmatrix}, \frac{\partial R_\beta}{\partial \beta} = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{pmatrix}, \frac{\partial R_\gamma}{\partial \gamma} = \begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

● critère d'arrêt ? Condition initiale  $X_0$  ?

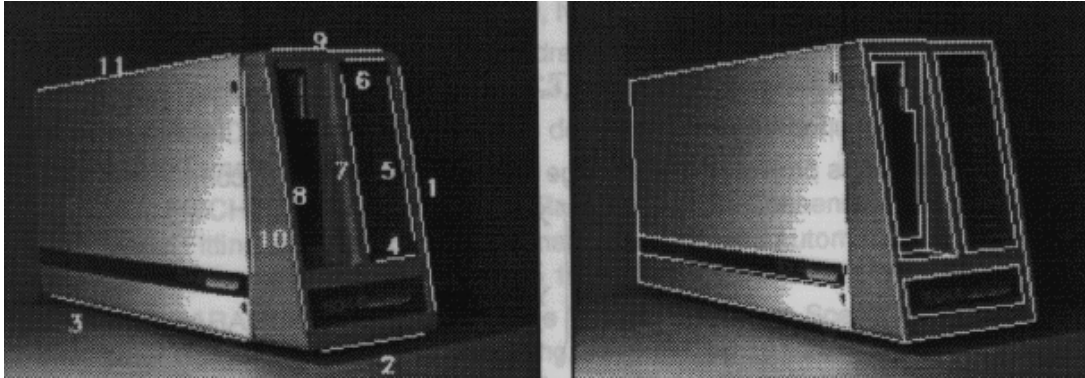
#### ● **Précision de la localisation / nombre de caméras**

	$\alpha$	$\beta$	$\gamma$	u	v	w
1 caméra	0.8°	1.1°	0.1°	4.7mm	3.2mm	29.9mm
2 caméras	0.9°	1.0°	0.3°	5.1mm	2.8mm	9.3mm
3 caméras	0.7°	0.9°	0.3°	4.7mm	2.6mm	7.0mm

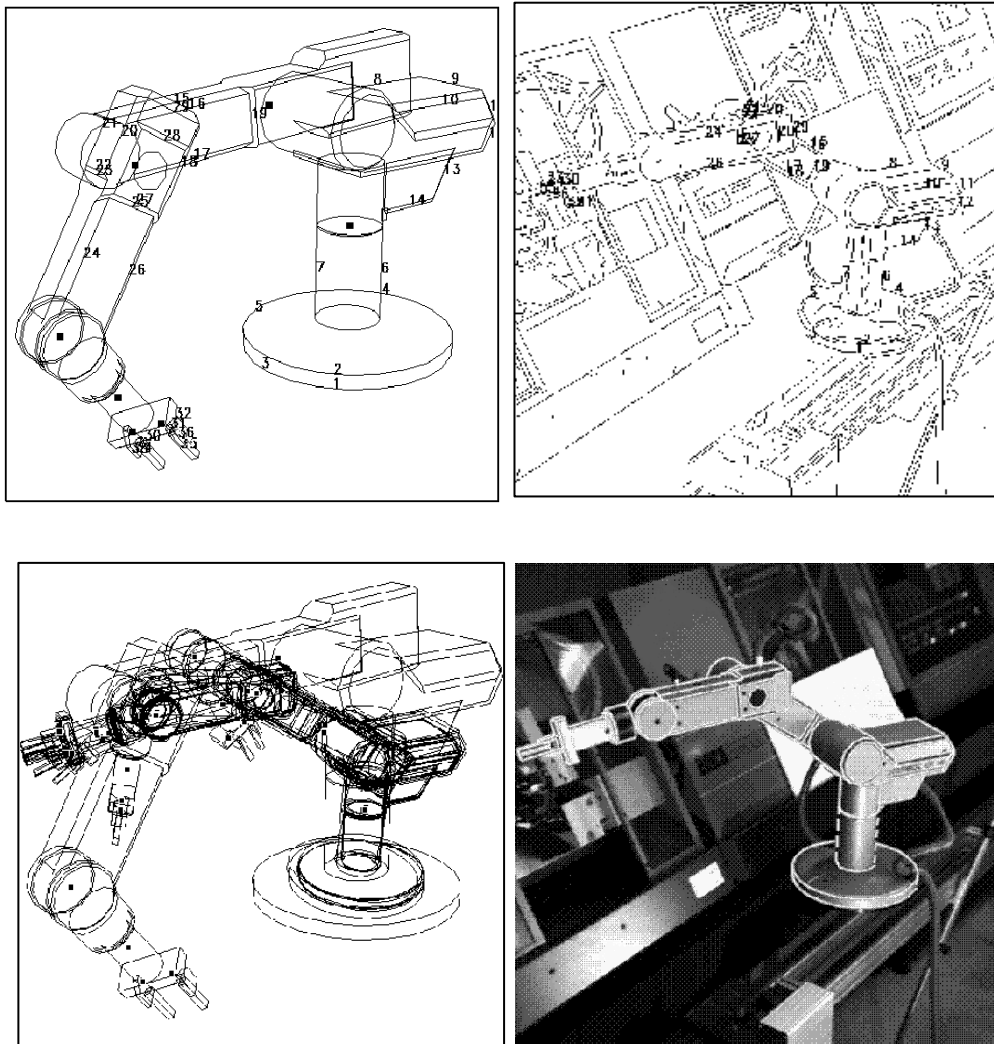
## Chapitre III

### - Localisation numérique -

- **Exemple : objet rigide**



- **Exemple : objet articulé**



## Chapitre III

### - Localisation numérique -

#### ● **Exercice : variantes de l'algorithme**

- quel serait l'apport en terme de précision en étendant la localisation à plusieurs caméras ?
- proposer alors le nouveau critère à minimiser.
- proposer un nouveau critère pour étendre à des appariements de points dans un cadre monoculaire.
- adapter la méthode de calibrage de caméra pour la localisation monoculaire à partir d'appariements de points

## Chapitre III

### - Reconnaissance -

- **But** : méthodologie pour établir des appariements objet/données puis localisation - reconnaissance i.e
  - appariements
  - localisation
- **Les étapes de la reconnaissance :**
  - apprentissage de(s) objet(s) à reconnaître (modèle)
  - localisation à partir d'appariements objet/données et d'un processus décisionnel
- **Complexité pour appairer** : si **O** ensemble des primitives objet (de taille  $n$ ) et **D** ensemble des primitives image (de taille  $m$ ) tel que  $m \geq n$ . Nombre d'appariements possibles :

$$C_m^n . n! = \frac{m!}{(m-n)!}$$

**⇒ Problème de nature combinatoire !**

- **Approches pour la décision :**  
exhaustives/heuristiques - isomorphisme de graphe -  
prédiction/vérification - recherche dans une structure  
d'arbre



## Chapitre III

### - Reconnaissance -

#### ● **Isomorphisme de graphes :**

- description des données et objet par un graphe
- construction d'un graphe de comptabilité - des nœuds - des arcs
- recherche d'une clique maximale dans ce graphe

⇒ **filtrer les nœuds dans le graphe de compatibilité**

#### ● **Exemple :**

modèle

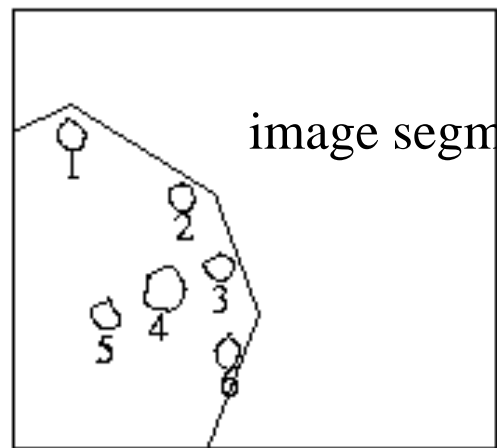
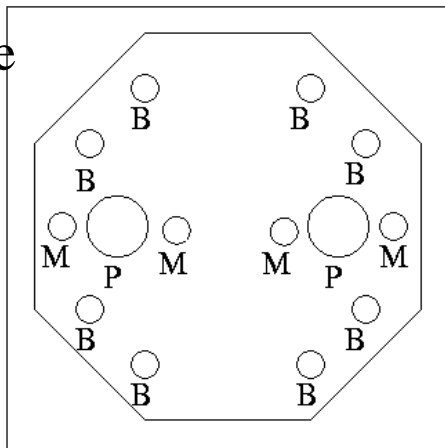
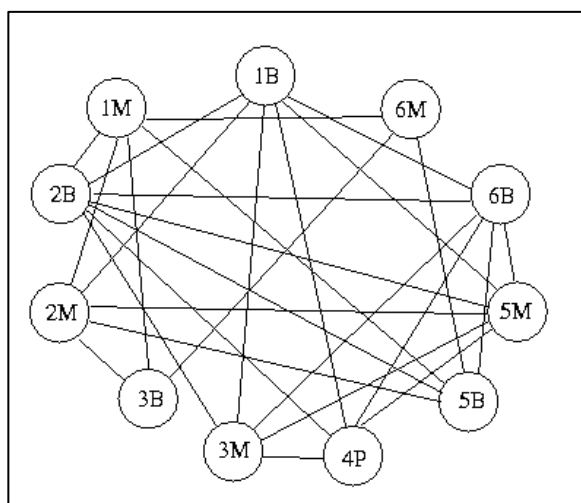
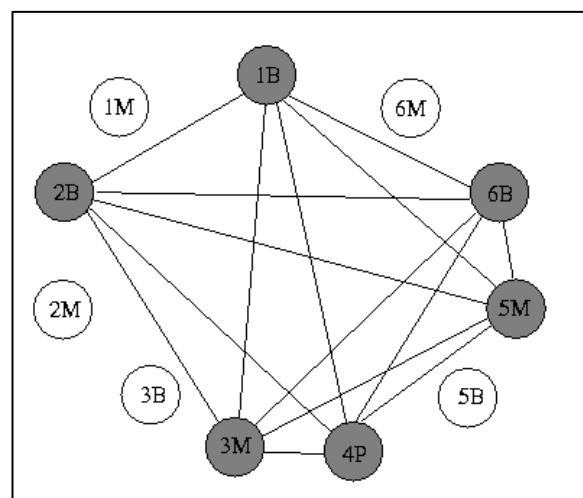


image segmentée



graphe de compatibilité



clique maximale

## Chapitre III

### - Reconnaissance -

- **Prédiction/vérification (2D/3D ou 3D/3D) :**

- hypothèse de localisation
- vérification

⇒ appariements et localisation résolus simultanément

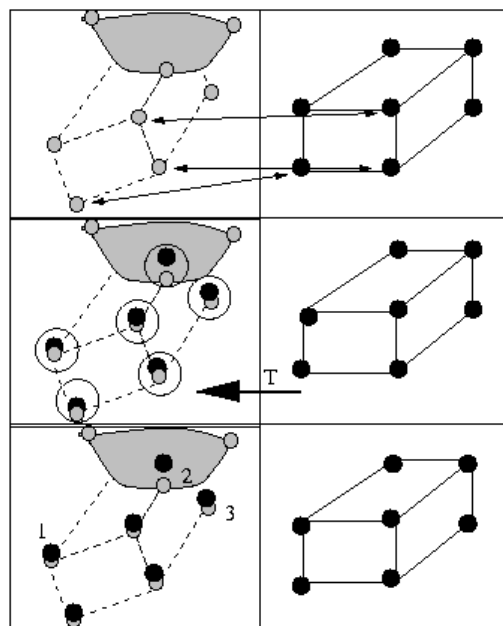
- **Etape 1 : la prédiction**

- établir quelques appariements
- estimer la localisation
- appliquer la loc. à l'ensemble des primitives

- **Etape 2 : la vérification**

- former des nouveaux appariements
- fusionner nouveaux et précédents appariements
- mise à jour de la localisation

- **Illustration (3D/3D) :**

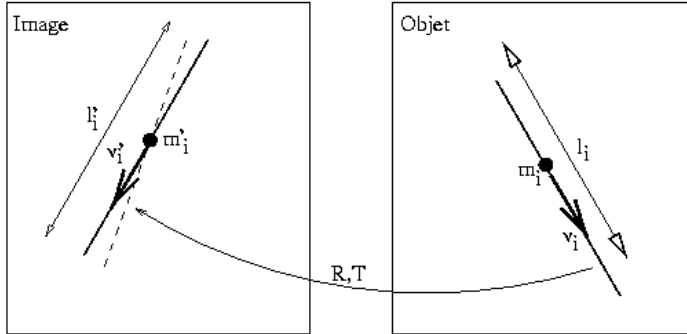


## Chapitre III

### - Reconnaissance -

#### ● Prédiction/vérification (3D/3D) :

##### ● vérification locale :



si  $v_i$  (resp.  $v'_i$ ) vecteur directeur

et  $m_i$  (resp.  $m'_i$ ) point milieu

de  $l_i$  (resp.  $l'_i$ ) :

$$\|v'_i - R.v_i\| \leq \varepsilon_1$$

$$\|m'_i - R.m_i - T\| \leq \varepsilon_2$$

$$(l'_i - l_i)^2 \leq \varepsilon_3$$

##### ● vérification globale :

$$Q_1 = \frac{1}{k} \sum_{i=1}^k \|v'_i - R^k.v_i\|^2$$

$$Q_2 = \frac{1}{k} \sum_{i=1}^k \|m'_i - R^k.m_i - T\|^2$$

$$Q_3 = \frac{1}{k} \sum_{i=1}^k (l'_i - l_i)^2$$

#### ● Prédiction/vérification (2D/3D) :

● vérification locale :  $(u_i - u'_i)^2 + (v_i - v'_i)^2 \leq \varepsilon_1$

● vérification globale :  $\sum_{i=1}^k ((u_i - u'_i)^2 + (v_i - v'_i)^2) \leq \varepsilon_2$

#### ● Que choisir comme seuils $\varepsilon_i$ ?

## Chapitre III

### - Reconnaissance -

#### ● Recherche d'une structure d'arbre :

- *premier niveau* : nœuds hypothèses i.e transformation objet/données
- *niveaux suivants* : nœud  $a_{ij}$  représentant un appariement  $m_i / c_i$

- recherche d'un chemin valide :

$$m_1 / c_1, \dots, nul, m_i / c_i, \dots$$

- critères précédents et nombre d'appariements le plus grand possible

#### ● Stratégie de recherche (1)

- abandon d'un chemin à l'instant courant si :

$$S_i + (n_H - i) \leq m_H$$

$$\text{ou } S_i + (n_H - i) \leq k_{\max} \text{ avec :}$$

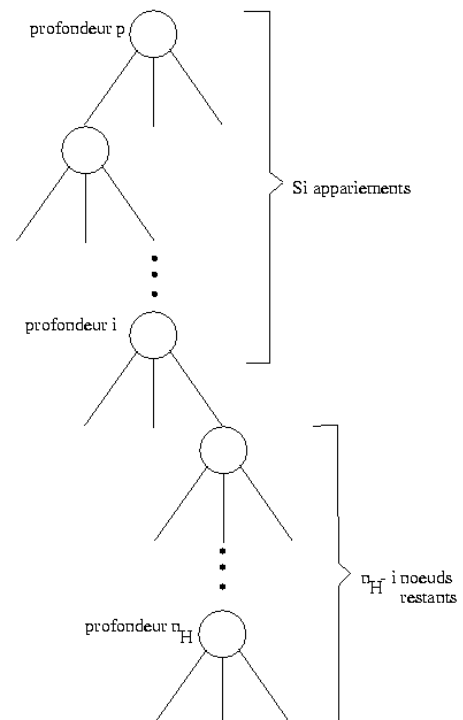
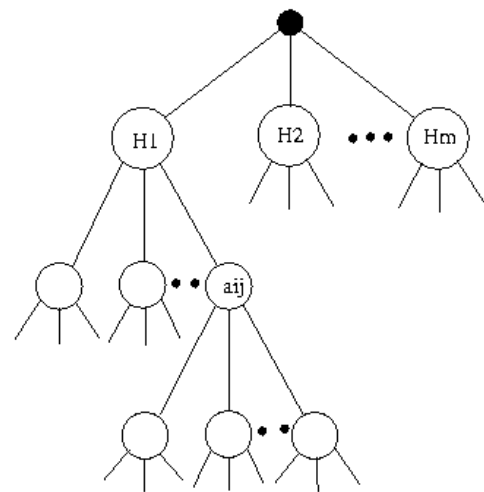
$n_H$  nbre de primitives

objet prédites visibles

$m_H$  nbre min d'appariements

$k_{\max}$  nbre max d'appariements

à l'instant courant

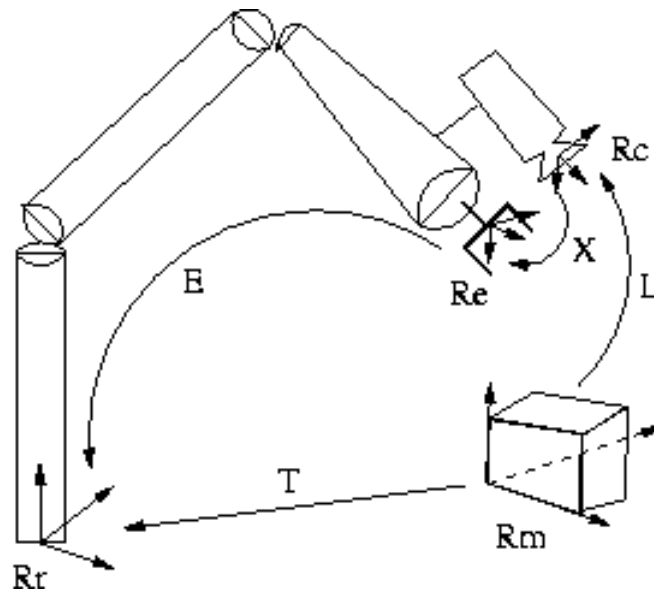


## Chapitre III

### - Application à la manipulation -

- **Calibration « bras / œil » :**

- But : guider le mouvement de la pince à l'aide d'infos issues de la caméra  $\Rightarrow$  déterminer la position relative du capteur / organe terminal



- Repères et transformations :

$R_r$  : repère de référence du bras articulé,

$R_e$  : repère lié à l'effecteur,

$R_c$  : repère caméra,

$R_m$  : repère du modèle de l'objet observé,

$[L]$  : localisation objet dans le repère caméra,

$[X]$  : matrice de passage bras / œil (à calibrer),

$[E]$  : positionnement du repère effecteur dans le repère absolu  $R_r$ ,

$[T]$  : localisation objet dans le repère  $R_r$ ,

$[T_{\text{prise}}]$  : position pince /  $R_m$  pour saisir objet

## Chapitre III

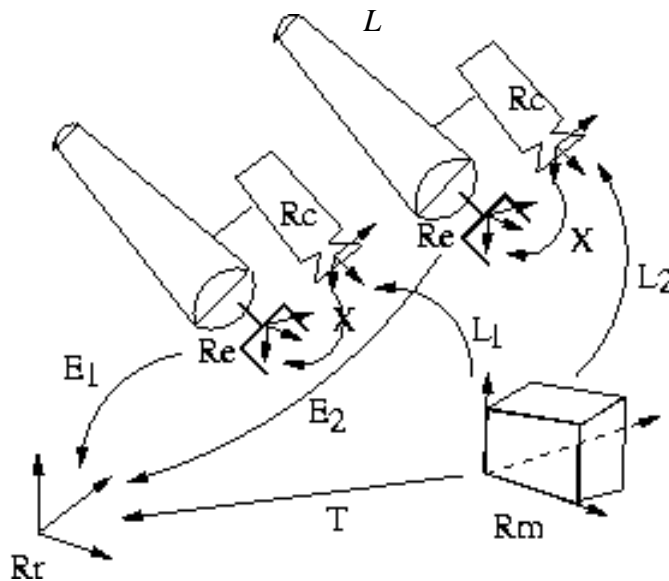
### - Application à la manipulation -

- Saisie de l'objet par une approche *look and move* :

$$[E]_1 = [E]_0 \cdot [X] \cdot [L]_0 \cdot [T_{prise}]$$

- Stratégies de calibration « bras / œil » :
    - localiser la pince à l'aide de primitives connues dans le repère pince (mire en bout de pince)
    - effectuer deux localisations sur le même objet depuis deux positions du bras (18 inconnues)
- [Tsai87] :

$$[L]_1^{-1} \cdot [X]^{-1} \cdot [E]_1^{-1} \cdot [E]_2 \cdot [X] \cdot [L]_2 = I \Rightarrow [A] \cdot [X] = [X] \cdot [B]$$



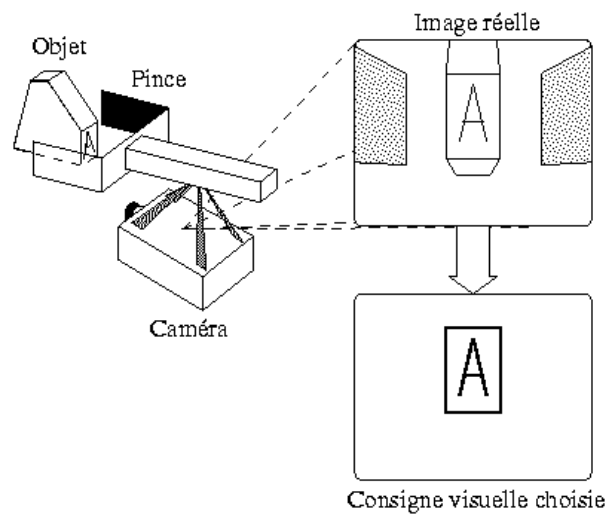
- minimiser des distances à des plans d'interprétation (12 inconnues) :

$$F(X, T, E_l, N_i^j, P_i^j) = R_{E_l} \cdot R_X \cdot N_i^l \cdot \left[ [R_T P_i^m + T_T] - [R_{E_l} T_X + T_{E_l}] \right]$$

## Chapitre III

### - Application à la manipulation -

- Une alternative à l'asservissement en position :  
**l'asservissement visuel** - pas de modèle géométrique inverse du robot - asservissement sur une consigne image (cours M2P SI)



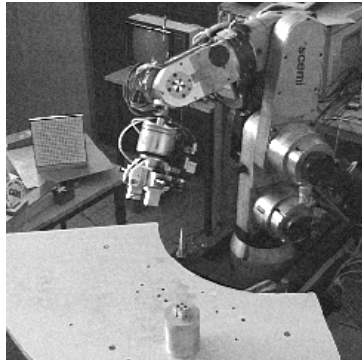
## Chapitre III

### - Application à la manipulation -

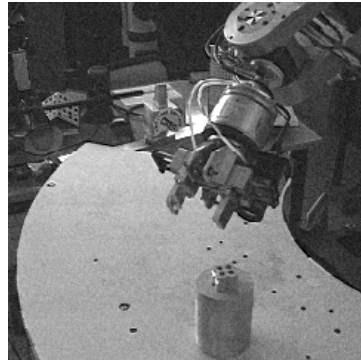
- **Principe d'un asservissement en position :**

- se rapprocher le plus possible de la position de prise puis effectuer le dernier mouvement relatif de l'effecteur en aveugle
- une bonne localisation de l'objet
- un bon calibrage main/œil

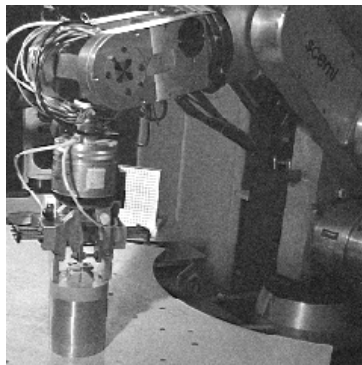
Site et  
position  
initiale



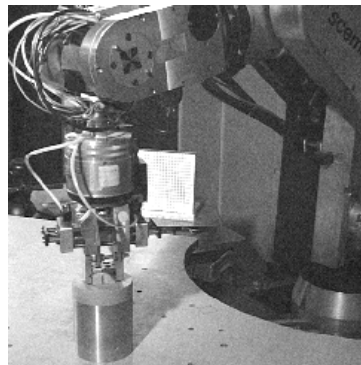
*Dernière  
position  
d'approche*



Position  
de saisie



Saisie de  
l'objet



- **Etapes (1) :**

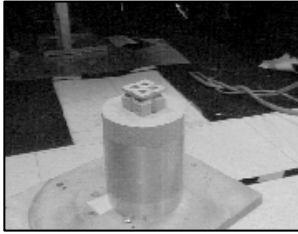
- ❶ acquisition+segmentation image
- ❷ prédiction de la localisation de l'objet (donnée par le modèle CAO du site)
- ❸ mises en correspondance (cliques maximales)
- ❹ déplacement du bras
- ❺ retour en ❶



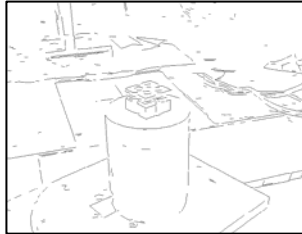
## Chapitre III

### - Application à la manipulation -

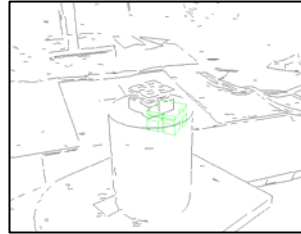
#### ● Etapes (2) :



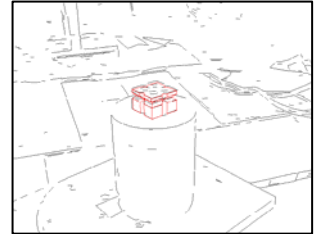
*Image I*



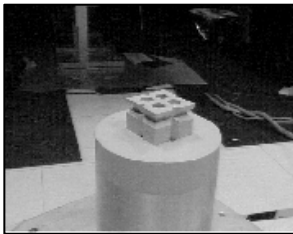
*Image I  
segmentée*



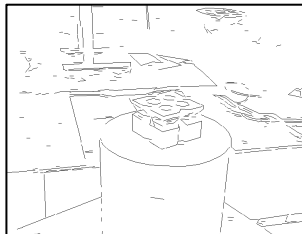
*Prédiction  
initiale*



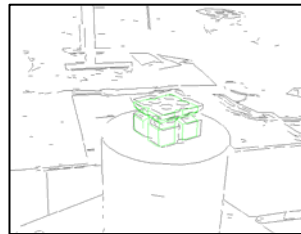
*Reconnaissance  
localisation*



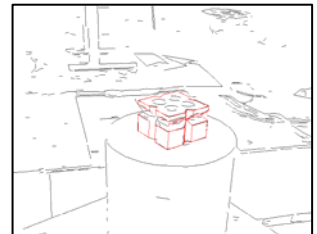
*Image II*



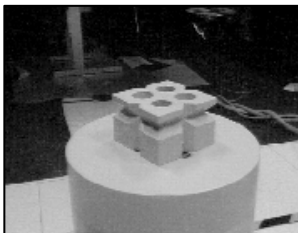
*Image II  
segmentée*



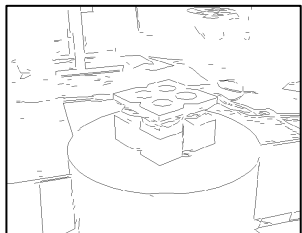
*Prédiction*



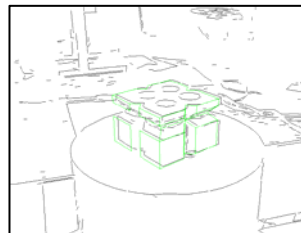
*Reconnaissance  
localisation*



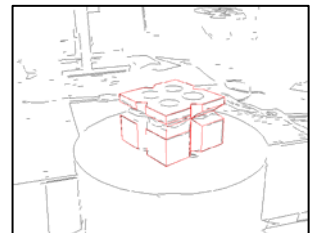
*Image III*



*Image III  
segmentée*



*Prédiction*



*Reconnaissance  
localisation*

## Références bibliographiques

- **Vision par Ordinateur.** R.Horaud et O.Monga, Hermès, 1993.
- **Modélisation 3D automatique – outils de géométrie différentielle.** F.Goulette, Les Presses, 1999.
- **Perception visuelle par imagerie vidéo.** M.Dhome, Hermès&Lavoisier, 2003.
- **Vision stéréoscopique et perception multi-sensorielle.** N.Ayache, Inter-Editions, 1989.
- **Three Dimensional Computer Vision. A Geometric Viewpoint.** O.Faugeras, MIT Press, 1993.

