



Master 2 EEA & INFO

Parcours AURO & IAFA

UE Perception pour la Robotique



Introduction



Systèmes d'acquisition de données 3D

Systèmes passifs

Systèmes actifs

Techniques Shape from X :

- ombrage
- texture
- mouvement
- ...

Stéréovision

Capteurs à lumière structurée :
 projection <<lumière>>
 et
 capteur vidéo

Télémétrie :

impulsionnelle onde modulée



Particularités

Triangulation

Δt $\Delta\phi$ ou Δf

Méthodes indirectes

Méthodes directes

Accès à la distance



Sommaire (18h C/TD et 12h TP)

1. Acquisition 3D (9h)
 1. Généralités
 2. Modélisation de caméra
 3. Stéréovision passive
 4. Capteurs actifs
2. Modélisation 3D (3h)
3. Localisation et reconnaissance 3D (5h)
4. Applications par des exemples
5. Correction anale (1h)



Optique de caméra



éclairage

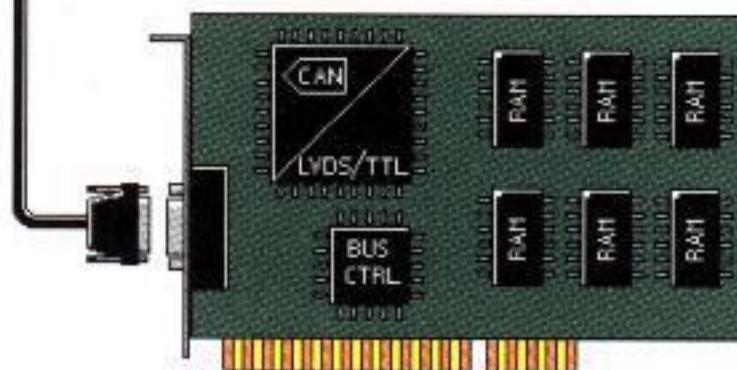
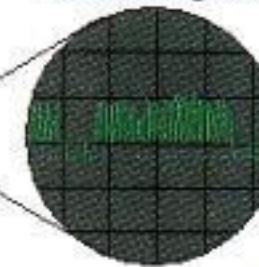
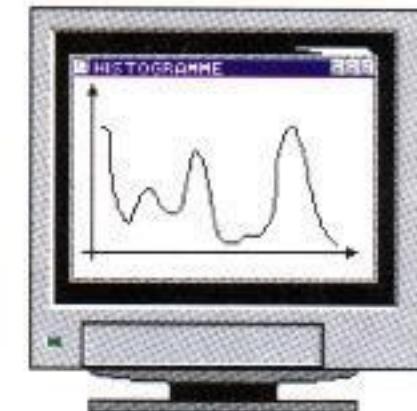


antique



transmission

normes-signaux

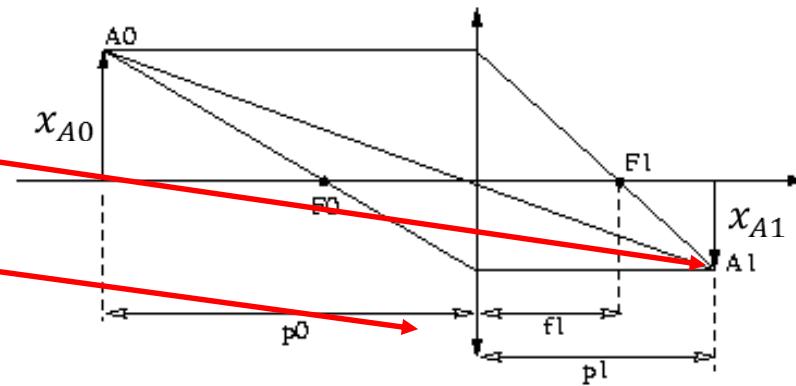
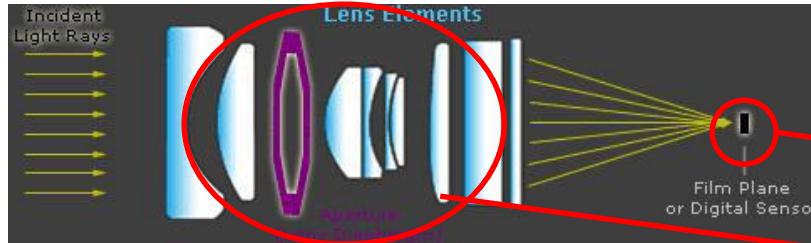
cartes d'acquisition
numériques ou analogiques

Traitement



Optique de caméra

■ Modèle d'optique mince



■ Définitions

- Foyer image *vs.* foyer objet
- Tirage optique
- Grandissement transversal : $G_t = \frac{x_{A1}}{x_{A0}} = \frac{p_1}{p_0}$
- Distance focale ou focale
- Relation de conjugaison : $\frac{1}{f_1} = \frac{1}{p_0} + \frac{1}{p_1}$

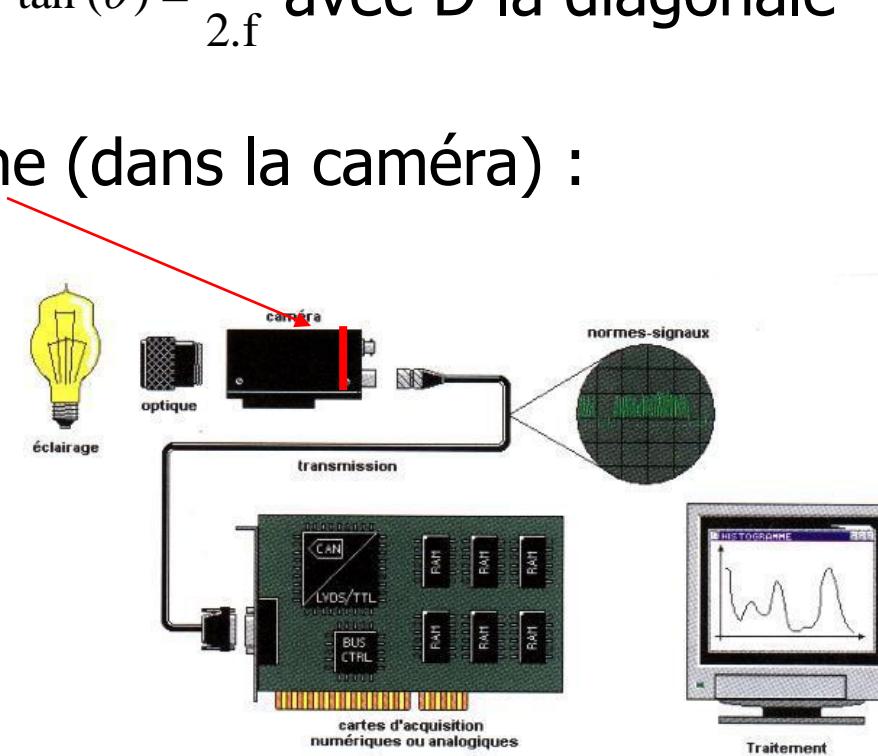


Paramètres d'une optique

■ Angle de champ

- Défini par 2θ tel que $\tan(\theta) = \frac{D}{2.f}$ avec D la diagonale de la rétine
- Format usuel de rétine (dans la caméra) :

TAILLE NOMINALE DU CAPTEUR	TAILLE STANDARDISÉE	DIAGONALE DE LA MATRICE CCD
1 pouce	12.8 mm x 9.6 mm	16 mm
2/3 pouce	8.8 mm x 6.6 mm	11 mm
1/2 pouce	6.4 mm x 4.8 mm	8 mm
1/3 pouce	4.8 mm x 3.6 mm	6 mm
1/4 pouce	3.2 mm x 2.4 mm	4 mm





Optique - Exercice

■ Exercices

- Calculer la focale pour le cas suivant : la taille de l'objet visualisé est assimilé à un segment de droite de 40 cm et le capteur est à 1m; On choisit une rétine 2/3 ".

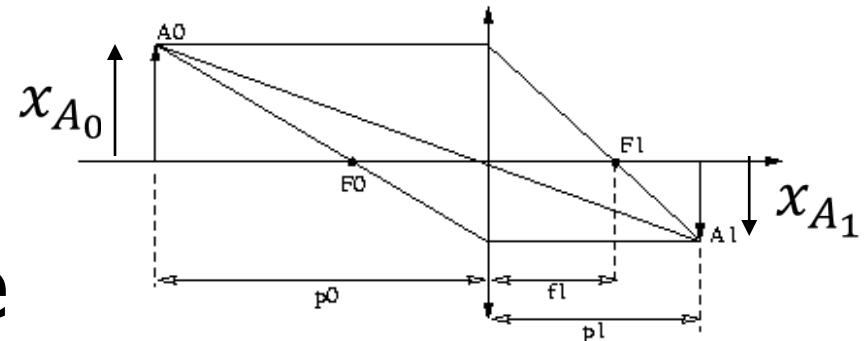
- Calculer la focale pour le cas suivant : angle de vue de 90° et un objet situé à 1m, rétine de 2/3"

Suite sur les capteurs image dans le module robotique industrielle avancée !!

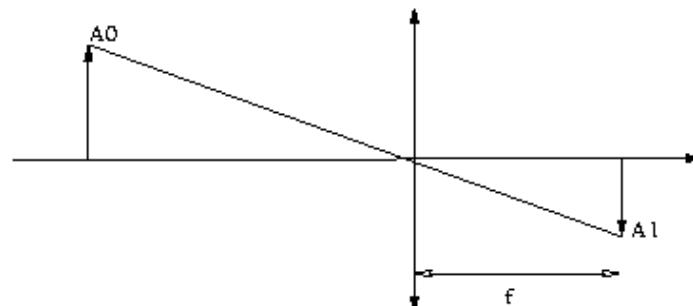


Modélisation de caméra perspective

- Modèle mince → optique = épaisseur nulle



- Modèle sténopé
 - Dit modèle « pinhole » → un seul rayon conservé

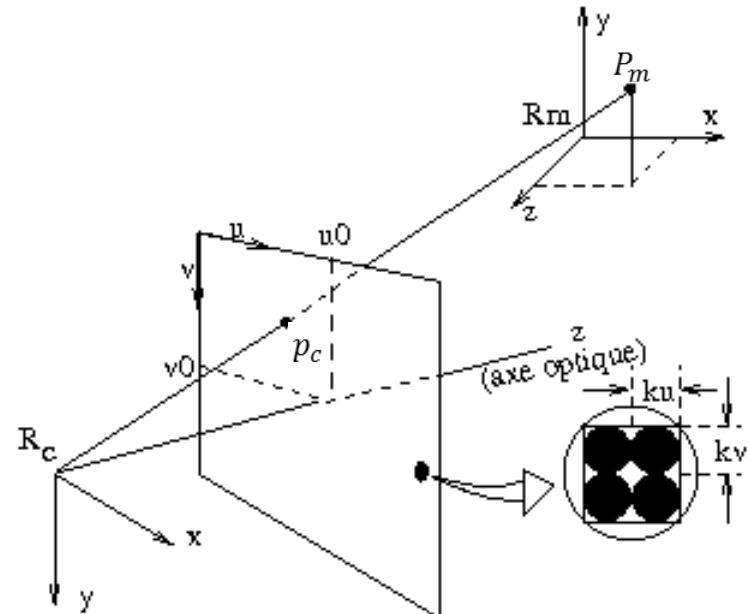
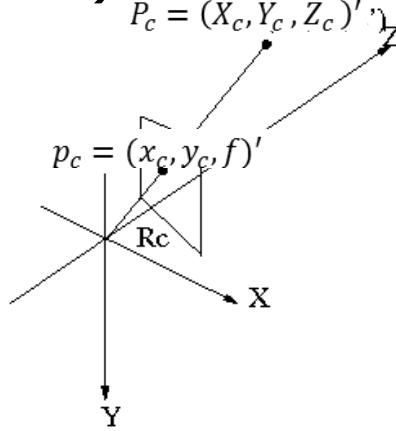
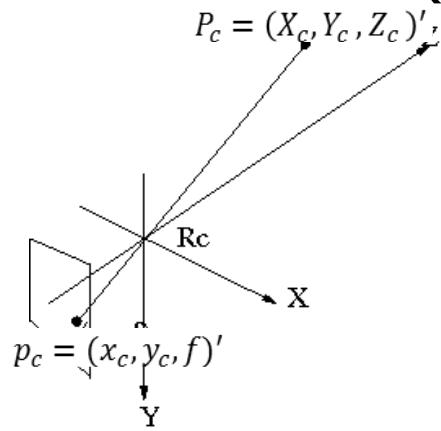


$$x_c = f \cdot \frac{X_c}{Z_c}, y_c = f \cdot \frac{Y_c}{Z_c}, z_c = f_1 = f$$



Modélisation de caméra perspective

- But : relation mathématique entre coordonnées 3D (monde) / 2D (image)



- Deux transformations :
 - Projection perspective
 - Transformation repères caméra / image (unité = pixel)



Modélisation de caméra perspective

- Projection perspective – en coordonnées homogènes

$$\begin{pmatrix} s \cdot x_c \\ s \cdot y_c \\ s \end{pmatrix} = \begin{pmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{pmatrix}$$

- Transformation repère caméra → repère image
- $$\begin{cases} u = k_u \cdot x_c + u_0 \\ v = k_v \cdot y_c + v_0 \end{cases}$$



Modélisation de caméra perspective

- Paramètres intrinsèques : repère caméra
→ repère image *Avec $\alpha_u = k_u \cdot f$, $\alpha_v = k_v \cdot f$:*

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

- Paramètres extrinsèques : repère monde
→ repère caméra

$$R_{\alpha\beta\gamma} = R_\gamma \cdot R_\beta \cdot R_\alpha = \begin{pmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{pmatrix}; T = \begin{pmatrix} T_x \\ T_y \\ T_z \end{pmatrix}; \begin{pmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{pmatrix} = \begin{pmatrix} R_{\alpha\beta\gamma} & T \\ 0 & 1 \end{pmatrix} \begin{pmatrix} X_m \\ Y_m \\ Z_m \\ 1 \end{pmatrix}$$



Modélisation de caméra perspective

■ Représentation en angles d'Euler

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

■ Transformation repère monde → repère image

$$\begin{pmatrix} s.u \\ s.v \\ s \end{pmatrix} = \begin{pmatrix} \alpha_u \cdot r_{11} + u_0 \cdot r_{31} & \alpha_u \cdot r_{12} + u_0 \cdot r_{32} & \alpha_u \cdot r_{13} + u_0 \cdot r_{33} & \alpha_u \cdot T_x + u_0 \cdot T_z \\ \alpha_v \cdot r_{21} + v_0 \cdot r_{31} & \alpha_v \cdot r_{22} + v_0 \cdot r_{32} & \alpha_v \cdot r_{23} + v_0 \cdot r_{33} & \alpha_v \cdot T_y + v_0 \cdot T_z \\ r_{31} & r_{32} & r_{33} & T_z \end{pmatrix} \begin{pmatrix} X_m \\ Y_m \\ Z_m \\ 1 \end{pmatrix}$$



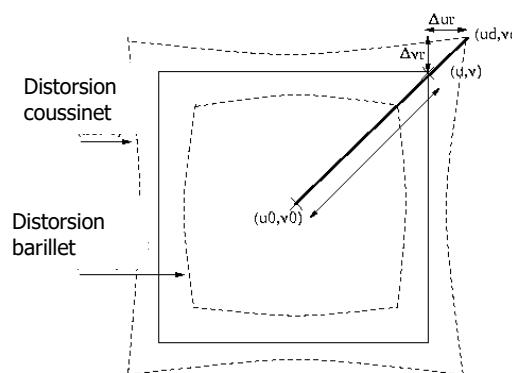
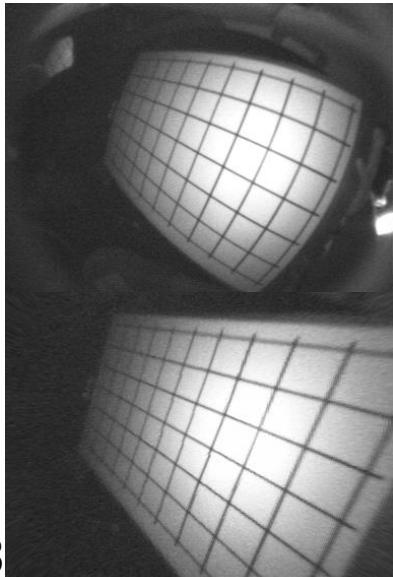
Modélisation de caméra perspective

■ Modèle de distorsion optique

■ Termes radial et tangentiel

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

■ Paramétrisation vs. focale $\rightarrow r_3 t_2, r_3 t_0, \color{red}{r_1 t_0}$



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

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

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

\Rightarrow A estimer: k_1, k_2, k_3, p_1, p_2



Etalonnage de caméra

- Optimisation non linéaire
 - Vecteur à estimer (ex : $r_3 t_0$) :

$$X = (u_0, v_0, \alpha_u, \alpha_v, \alpha, \beta, \gamma, T_x, T_y, T_z, k_1, k_2, k_3)^T$$

- Optimisation non linéaire :

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

avec :

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



Etalonnage de caméra

- Optimisation non linéaire
 - Linéarisation autour d'une valeur initiale X_0
 - ➔ système à résoudre: $V = A.\Delta X - L$

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

- Résolution par moindres carrés : $\text{Min}(V^T.V)$

$$V = \begin{pmatrix} v_{u_1} \\ v_{v_1} \\ \vdots \\ v_{u_n} \\ v_{v_n} \end{pmatrix}, L = \begin{pmatrix} u_1 - P^1(X_0) \\ v_1 - Q^1(X_0) \\ \vdots \\ u_n - P^n(X_0) \\ v_n - Q^n(X_0) \end{pmatrix}, \Delta X = \begin{pmatrix} u_0 - (u_0)_0 \\ v_0 - (v_0)_0 \\ \alpha_u - (\alpha_u)_0 \\ \alpha_v - (\alpha_v)_0 \\ \vdots \end{pmatrix}, J = \begin{pmatrix} \frac{\partial P^1}{\partial u_0} & \frac{\partial P^1}{\partial v_0} & \cdots & \frac{\partial P^1}{\partial \alpha^1} & \cdots \\ \frac{\partial Q^1}{\partial u_0} & \frac{\partial Q^1}{\partial v_0} & \cdots & \frac{\partial Q^1}{\partial \alpha^1} & \cdots \\ \vdots & \vdots & \ddots & \vdots & \cdots \\ \frac{\partial P^n}{\partial u_0} & \frac{\partial P^n}{\partial v_0} & \cdots & \frac{\partial P^n}{\partial \alpha^1} & \cdots \\ \frac{\partial Q^n}{\partial u_0} & \frac{\partial Q^n}{\partial v_0} & \cdots & \frac{\partial Q^n}{\partial \alpha^1} & \cdots \end{pmatrix}$$

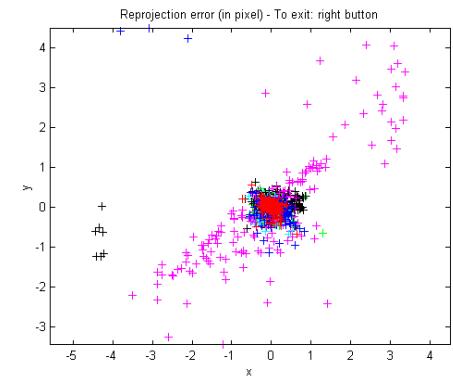
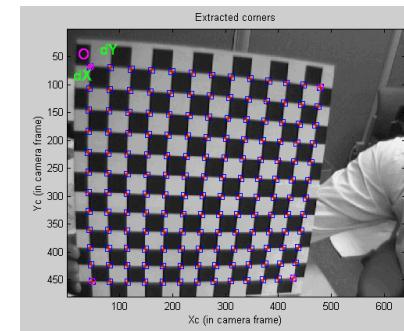
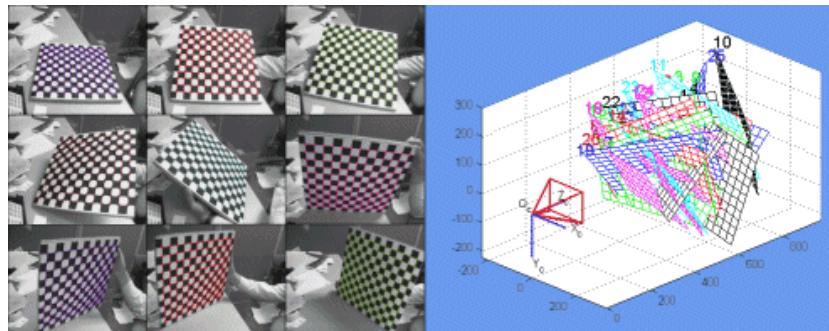
- Calcul des dérivées, estimée initiale



Etalonnage de caméra – Mise en oeuvre

- « Boîte à outils » → OpenCV, Matlab, etc.)
 - Mire 3D de dimensions connues
 - Réglage de la caméra : mise au point, ouverture
 - Appariements 2D/3D, extraction subpixel des amers mire
 - Multi-vues, pour m vues :

$$X_{7+6*m} = (u_0, v_0, \alpha_u, \alpha_v, k_1, k_2, k_3, \alpha^1, \beta^1, \gamma^1, t_x^1, t_y^1, t_z^1, \dots, \alpha^m, \dots, t_z^m)^T$$



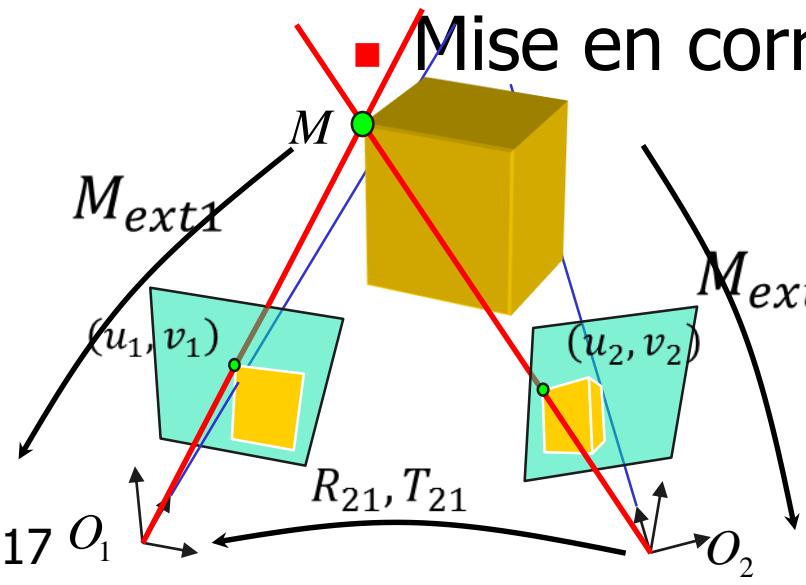


Stéréovision passive - Principe

■ Principe :

- Étalonnage indépendant de chaque caméra
- Détermination du positionnement relatif inter caméra : $M_{21} = M_{ext1} \cdot M_{ext2}^{-1}$

■ Mise en correspondance et triangulation



$$\overrightarrow{O_1 M} = \lambda \cdot \vec{v}_1 \text{ avec } \vec{v}_1 = \left(\frac{u_1 - u_0}{\alpha_u}, \frac{v_1 - v_0}{\alpha_v}, 1 \right)^T$$
$$\overrightarrow{O_2 M} = \lambda' \cdot \vec{v}_2 \text{ avec } \vec{v}_2 = \left(\frac{u_2 - u'_0}{\alpha'_u}, \frac{v_2 - v'_0}{\alpha'_v}, 1 \right)^T$$



Stéréovision passive - Principe

- Appariements – types de primitives :
 - Denses ou éparses ?
 - Régions – tailles non conservées – segmentation
 - Segments → reconstruction de segments 3D – scènes structurées – similarité de longueur, orientation, etc.
 - **Pixels** – tous les pixels ou pixels spécifiques

image gauche



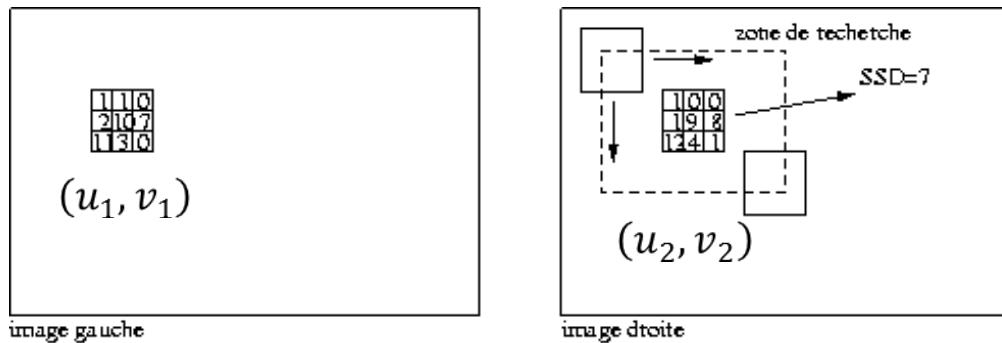
image droite





Stéréovision passive – Appariement de pixels

- Scores paramétriques :



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

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

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

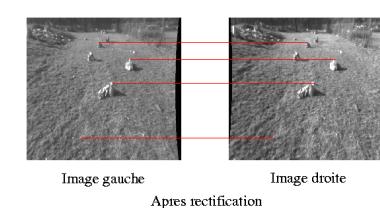
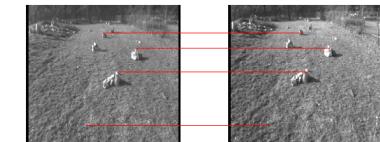
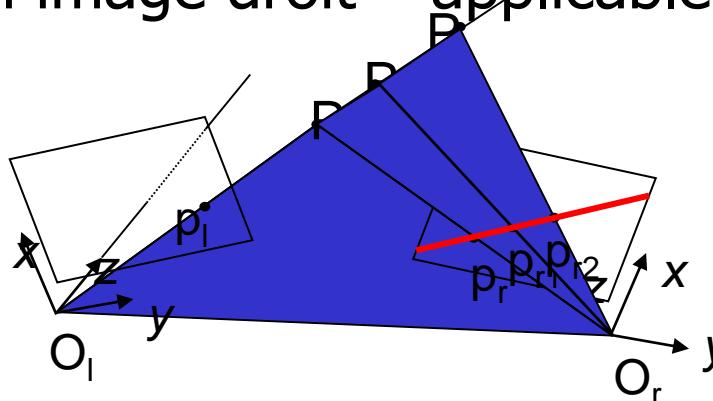
$p_1 = (u_1, v_1)^T$ apparié avec $p_2 = (u_2, v_2)^T$, $I(u, v)$ niveau en (u, v)

- Scores non paramétriques : Census Transform, etc.

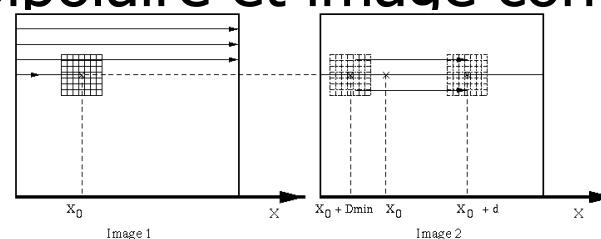


Stéréovision passive – Appariement de pixels

- Contrainte épipolaire
 - Droite issue de l'intersection de (o_1, o_2, p_1) avec le plan image droit – applicable aux points/segments



- Rectification des images pour stéréo dense – lignes épipolaire et image confondues

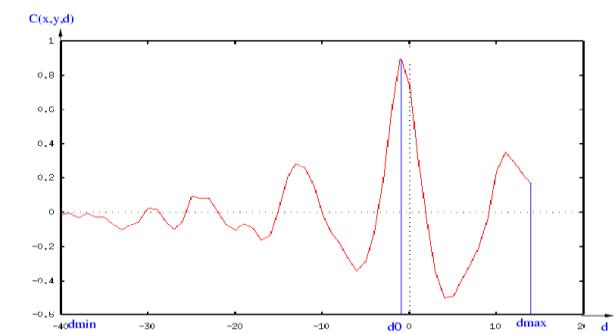
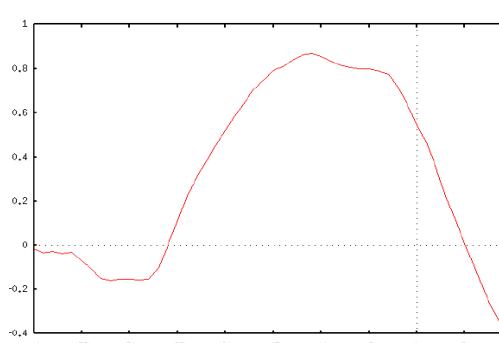
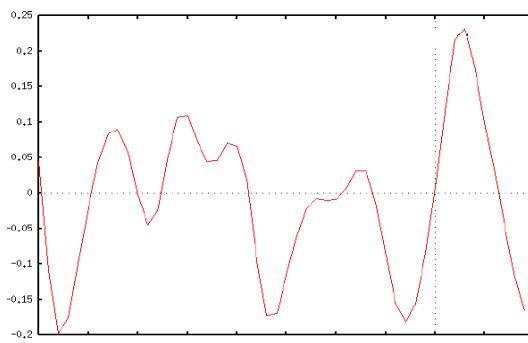
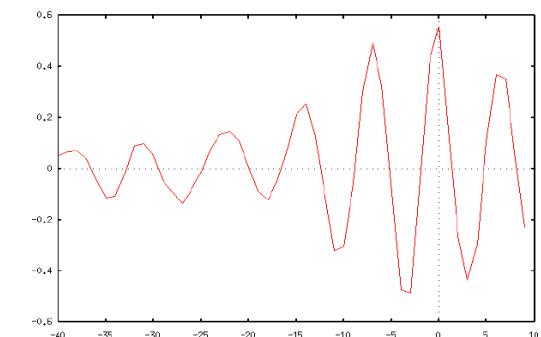
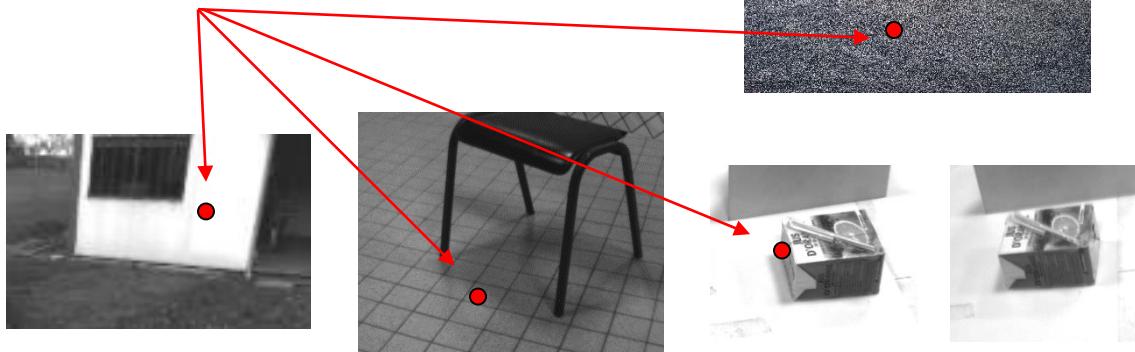




Stéréovision passive – Appariement de pixels

- Images rectifiées - Filtrage par étude de la corrélation

Pixel à appairer – 4 contextes

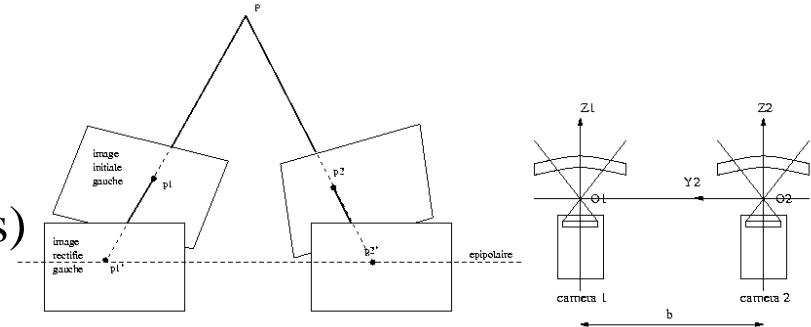




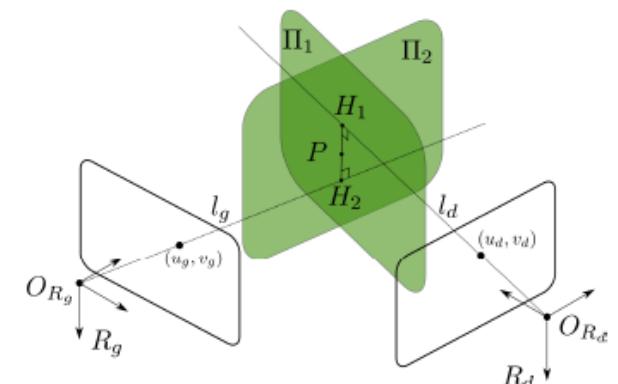
Stéréovision passive - Reconstruction

- Sur images rectifiées → triviale si disparité (d) calculée

$$\begin{aligned} X_{c_1} &= X_{c_2}, Y_{c_1} = Y_{c_2} + b, Z_{c_1} = Z_{c_2} \\ \Rightarrow Z_c &= \frac{b \cdot \alpha_v}{d}; d = |v_1 - v_2| \text{ (caméras identiques)} \end{aligned}$$

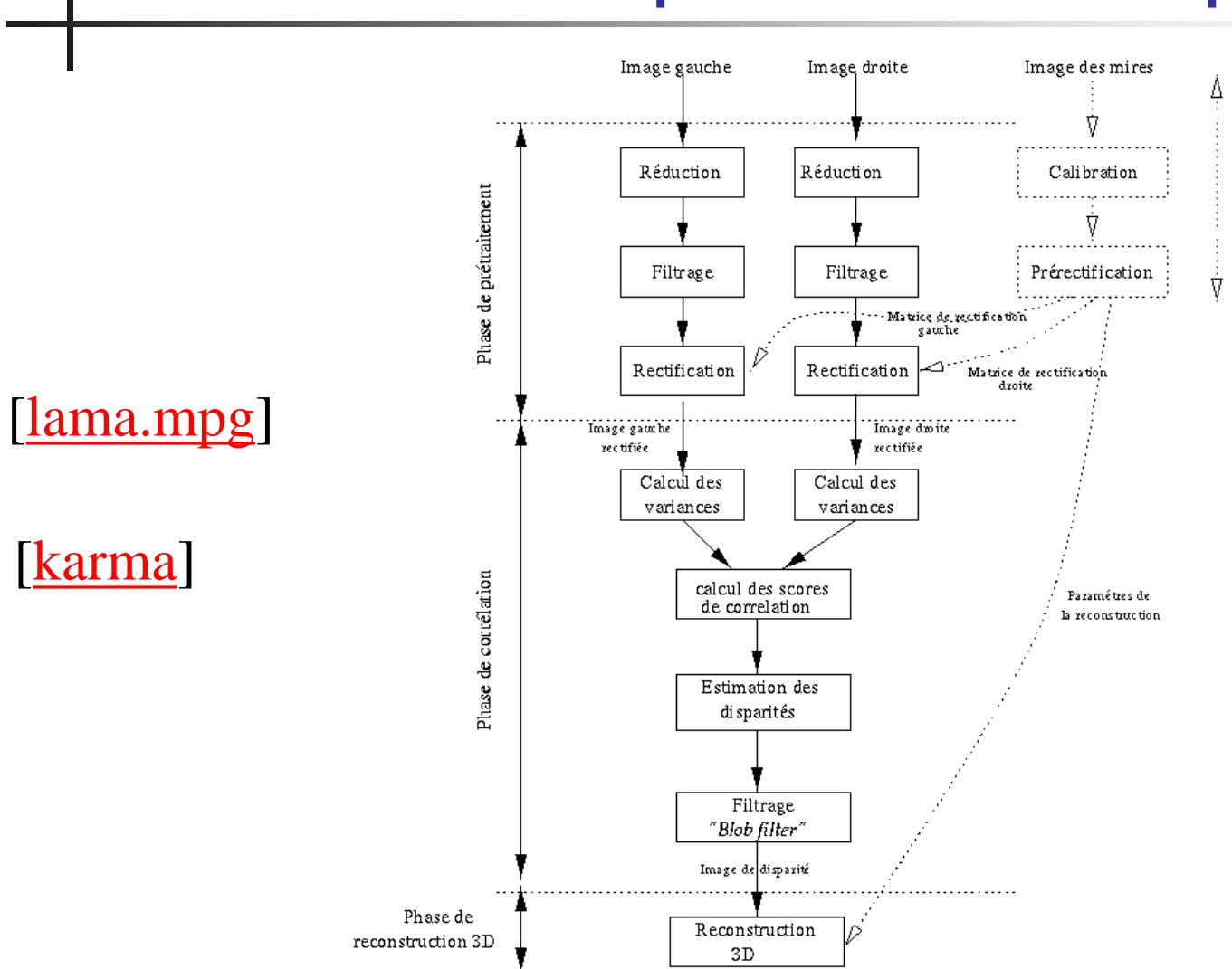


- Sur images non rectifiées
 - Calcul de H_1 et H_2 resp. intersection de (Π_2, l_d) et (Π_1, l_g)
 - Calcul du point milieu du segment $[H_1, H_2]$





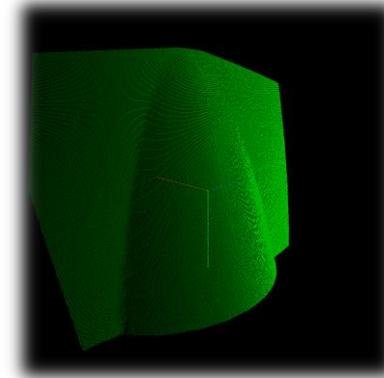
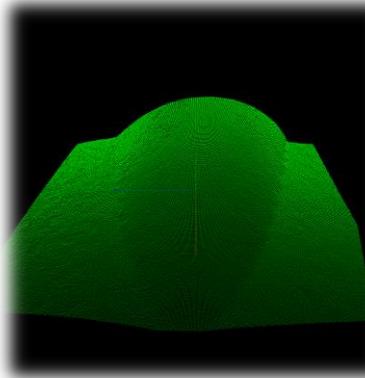
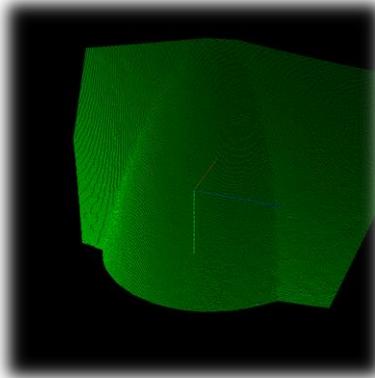
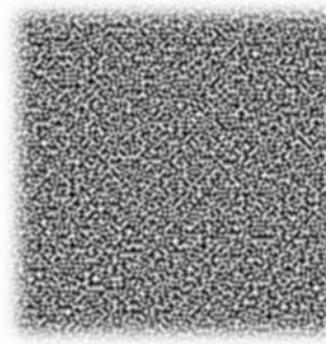
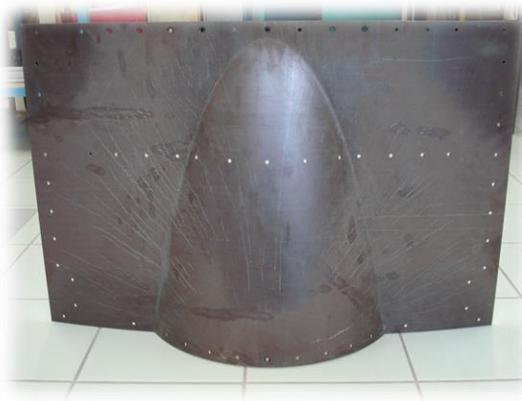
Stéréovision passive - Récapitulatif





Reconstruction par stéréovision passive - Exemple

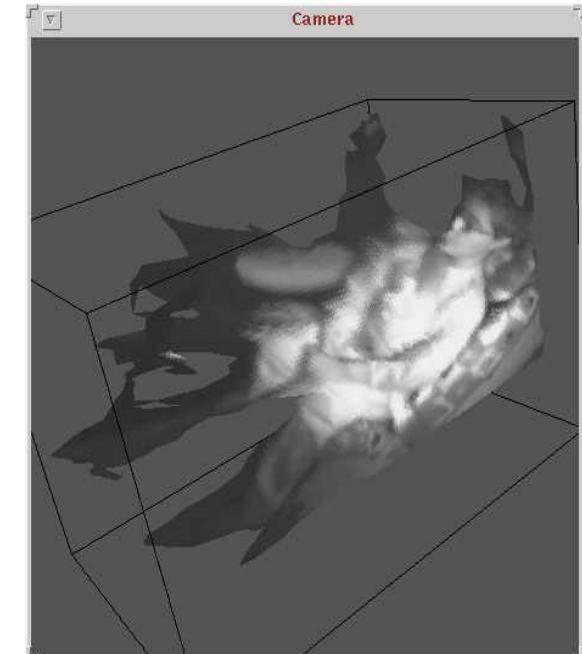
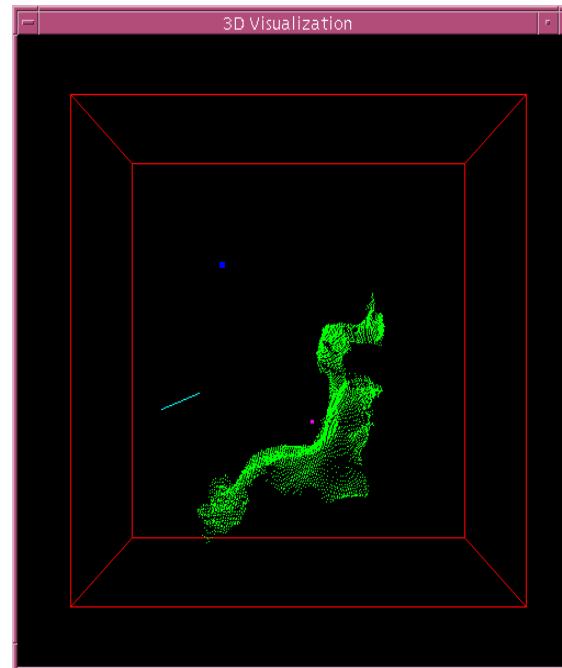
- Inspection optique de pièces aéronautiques chez AIRBUS : panneaux de fuselage ou de voilure, métalliques ou composites



Reconstruction par stéréovision passive - Exemple

- Occupation d'un cockpit chez CONTINENTAL : capteur CMOS, focale 2.1mm, illuminateur NIR

[[siemens.avi](#)]





Stéréovision passive - Exercice

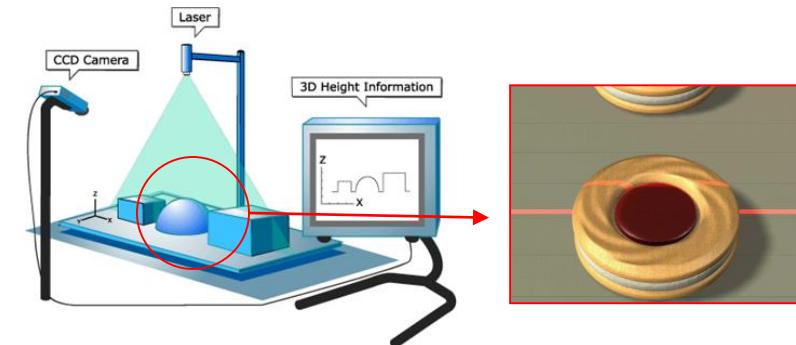
- **Exercice : stéréovision embarquée sur véhicule routier – Le système est monté à une hauteur de 80cm. Le sol est supposé plat. Les axes optiques sont orientés vers le bas d'un angle site de 20°. On donne : résolution de 320x240 pixels, focale de 2.1mm, base inter-caméras de 12cm, caméras en configuration 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 $\alpha_v \approx 105^\circ$. En déduire la disparité à la distance minimum de perception sur le sol. Quelle disparité pour un point situé sur le sol à 3m puis 5m du véhicule ?
 - Conclure sur son utilisation en détection d'obstacles.

[[videre.mpg](#)]



Capteurs actifs

- Principe : émetteur et récepteur de rayonnement
- Exemples :
 - RADAR, peu utilisé
 - Ultrason, télémétrie laser
 - Lumière structurée : profilométrie, capteur RGB-D
 - Caméra active





Télémétrie laser

- Mesure de distance

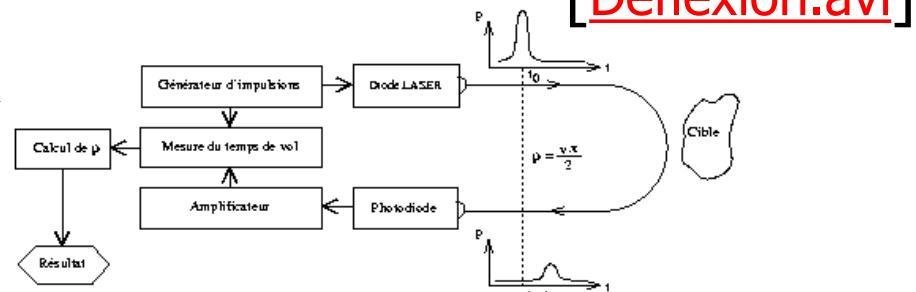
- Exemple du temps de vol – impulsion de courte durée – mesure de temps entre émission / réception

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

v vitesse de propagation
du signal

ρ distance à l'objet
 τ temps de vol

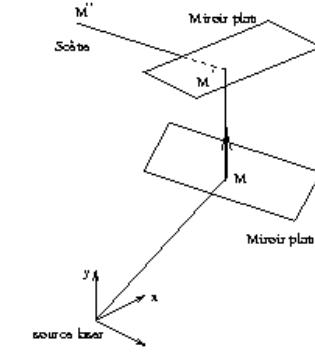
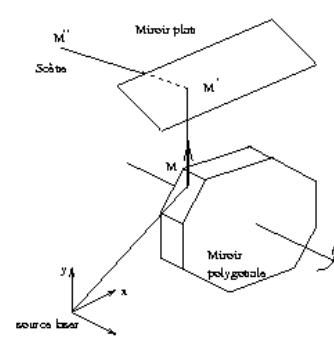
[[Deflexion.avi](#)]



[[Nav-SICK.avi](#)]

[[Nav-Velod.mp4](#)]

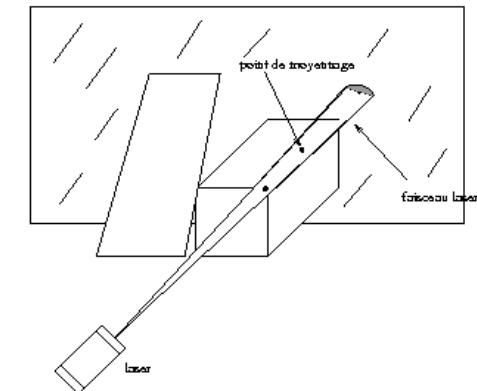
- Télémétrie 3D ou 2D → système de défexion (miroirs)





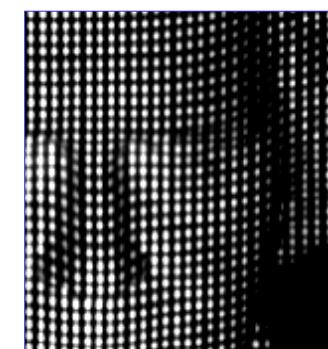
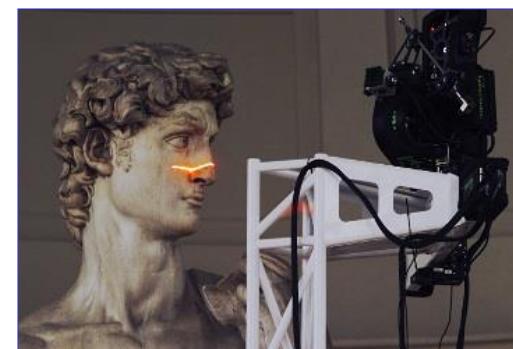
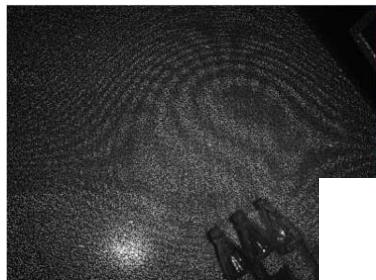
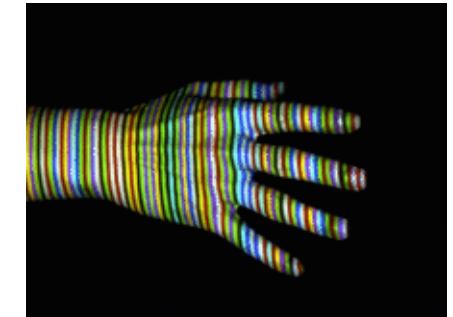
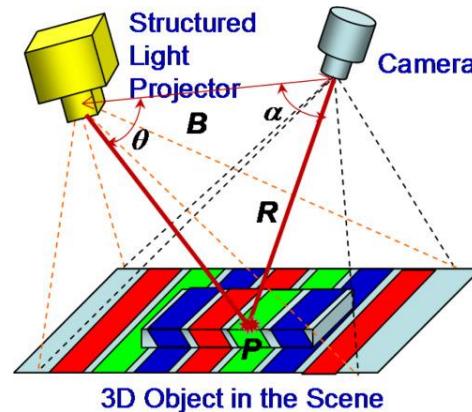
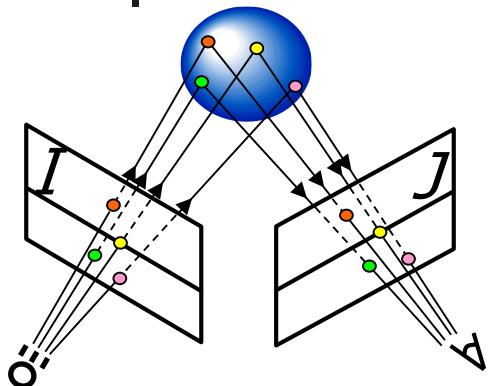
Télémétrie laser

- Avantages
 - Mesure directe de distance
 - Portée
 - Signal peu divergent
- Inconvénients
 - Surfaces spéculaires et réflexions multiples
 - Problème des points de moyennage
 - Résolution liée au système de déflexion
- Précision du laser SICK : *3cm pour [1-5m], supérieur ensuite*





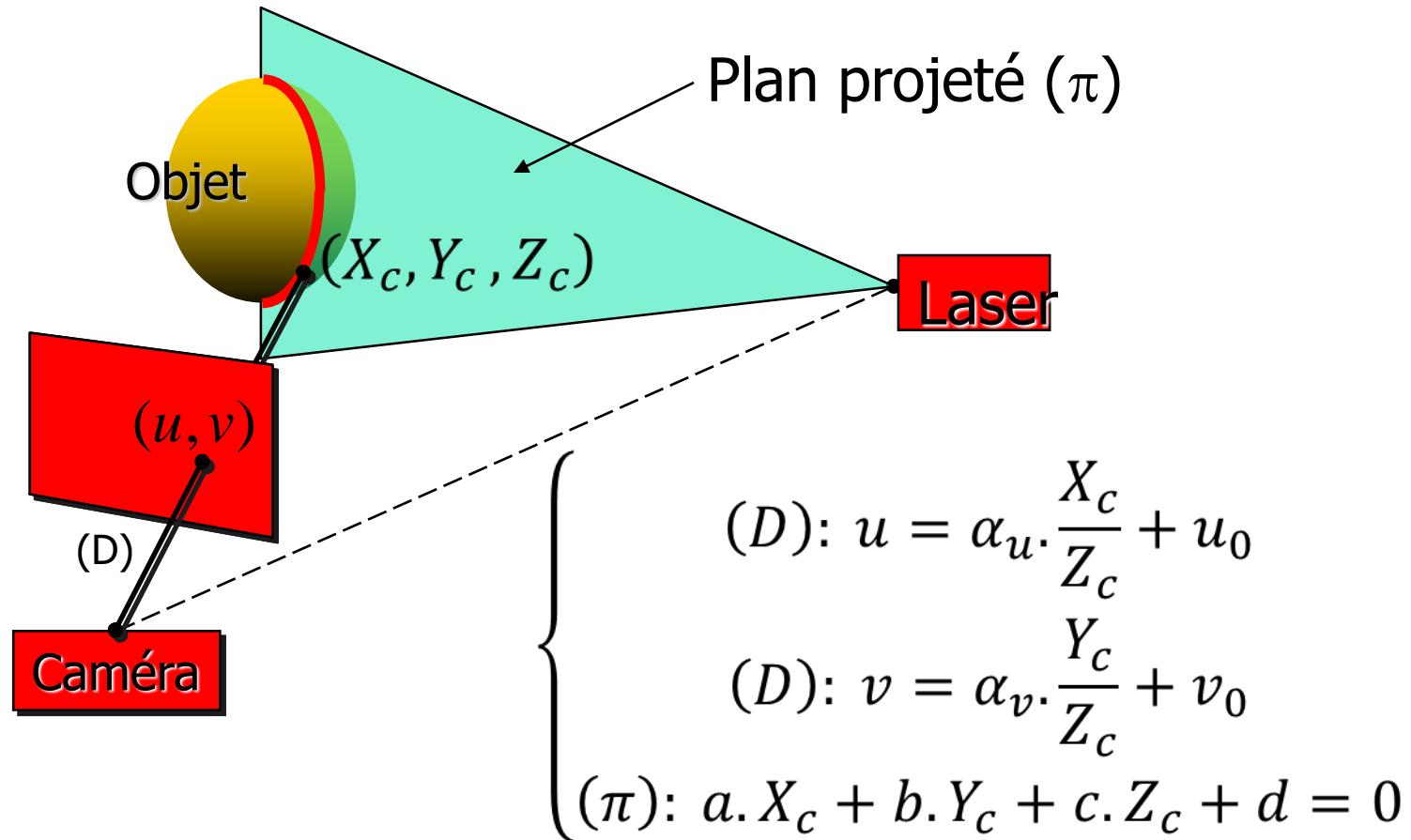
Lumière structurée



[[Kinect-SDK.avi](#)]



Lumière structurée

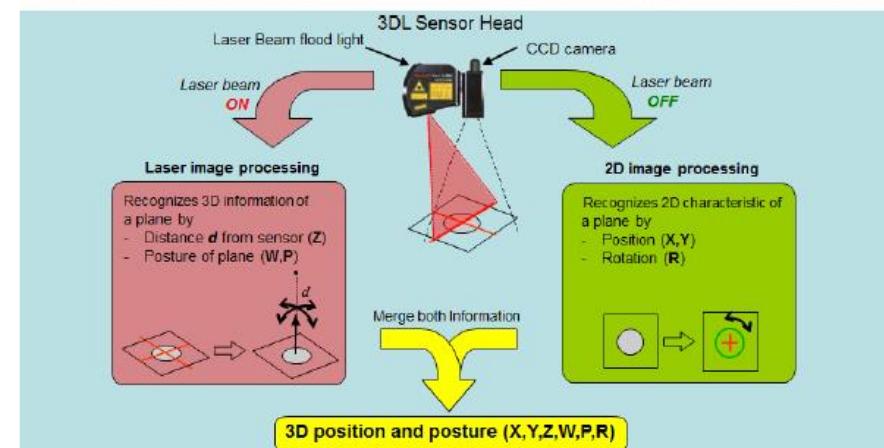
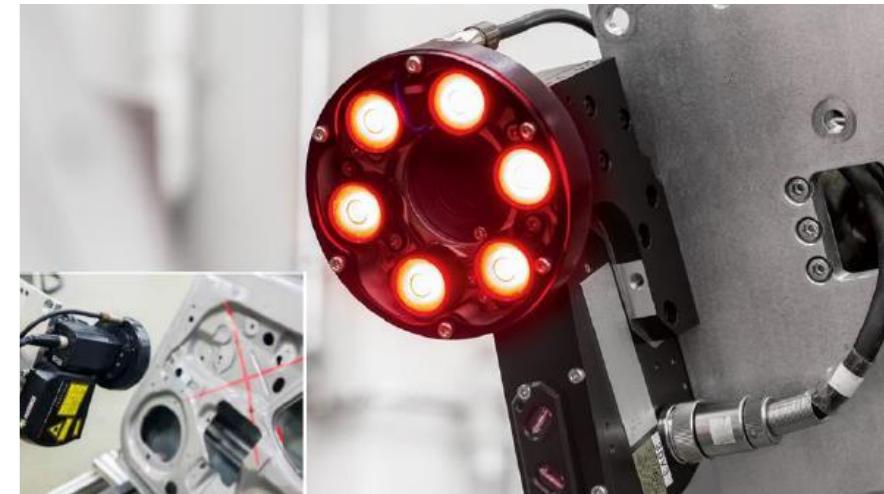


→ Requiert un étalonnage de la caméra et du plan laser !!



Lumière structurée et profilométrie 3D

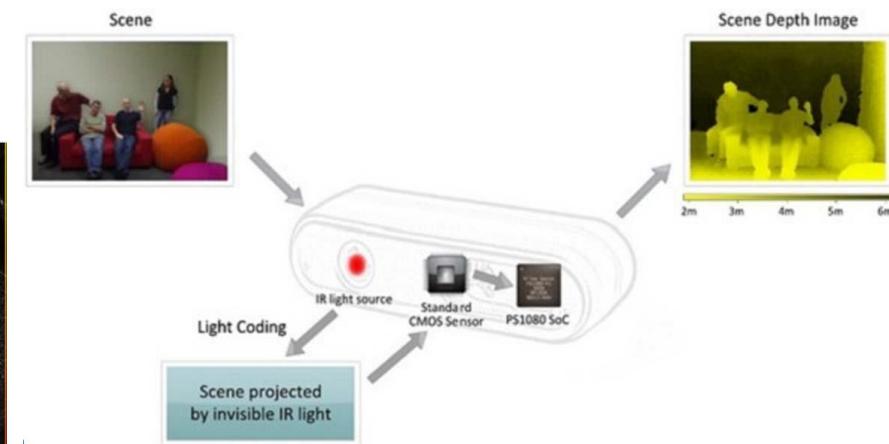
Contrôle sur objets quelconques
Couplage caméra 2D + laser
→ Profilométrie 3D : estimation
du plan support
→ Segmentation 2D
→ Métrologie 2D ou 3D ponctuelle
par triangulation avec le plan
support





Lumière structurée – Codage binaire (rayons)

- Capteur RGB-D : Kinect, Asus, Creative senz3D, Intel F200
 - type lumière structurée (IR)
 - matrice dense de points codés → *texture pseudo aléatoire* (« speckle »)
 - 3 tailles de « speckles » [0.8-1.2m], [1.2-2m], [2-3.5m]

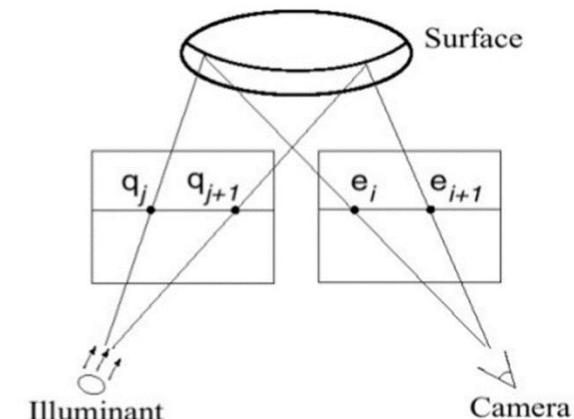




Lumière structurée – Codage binaire (rayons)

■ Capteur RGB-D

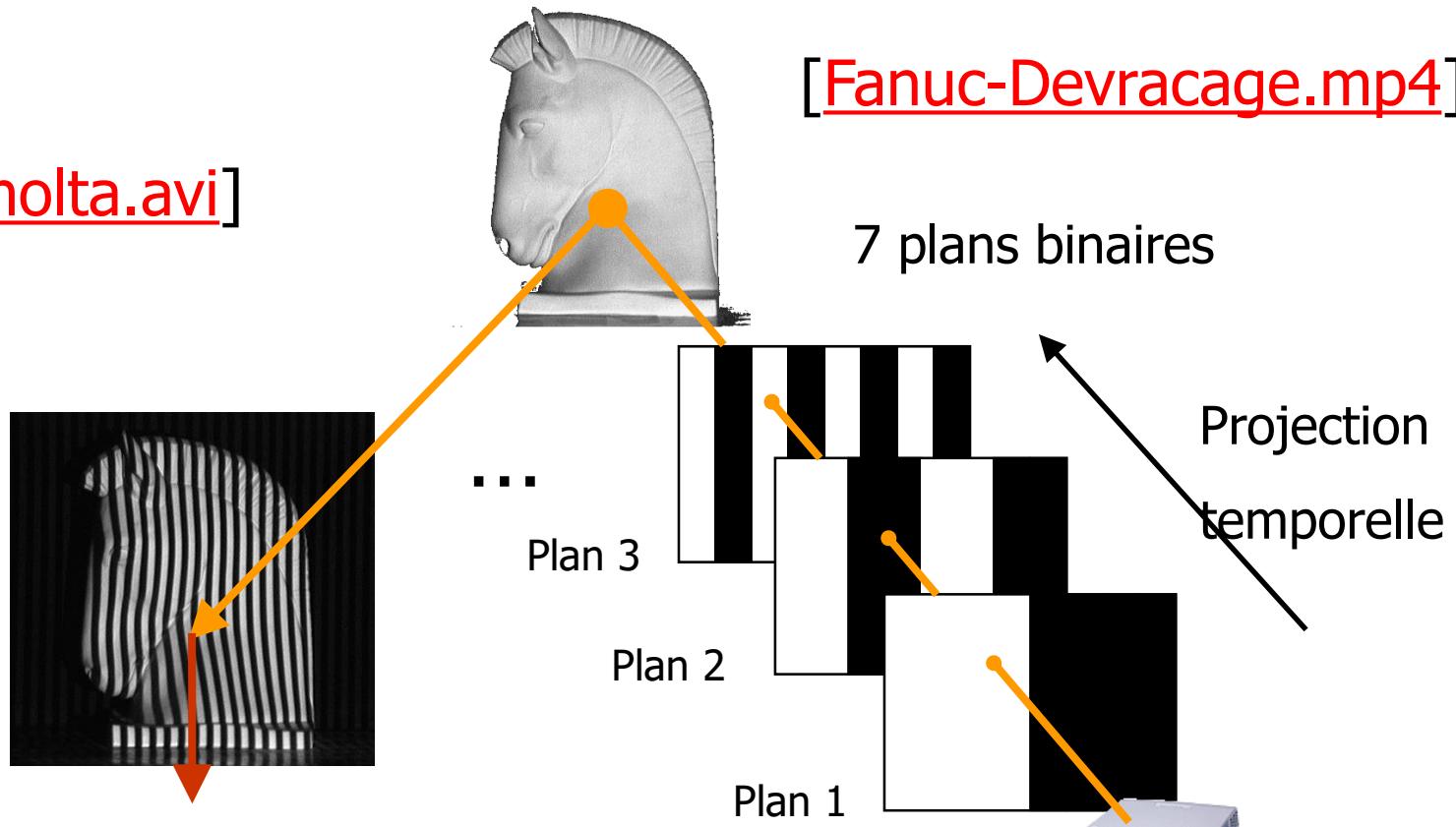
- Calibrage entre faisceaux projetés et caméra (préalablement calibrée)
- Projection des faisceaux /plan focal → *image virtuelle*
- Rectification des images
- Appariements (score de similarité) et calcul disparité
- Résolution : *320x240 pixels*
- Précision : *1cm à [0.8-3.5m]*
- Résolution 3D : *3mm à 2m*





Lumière structurée – Codage binaire (plans)

[[Minolta.avi](#)]

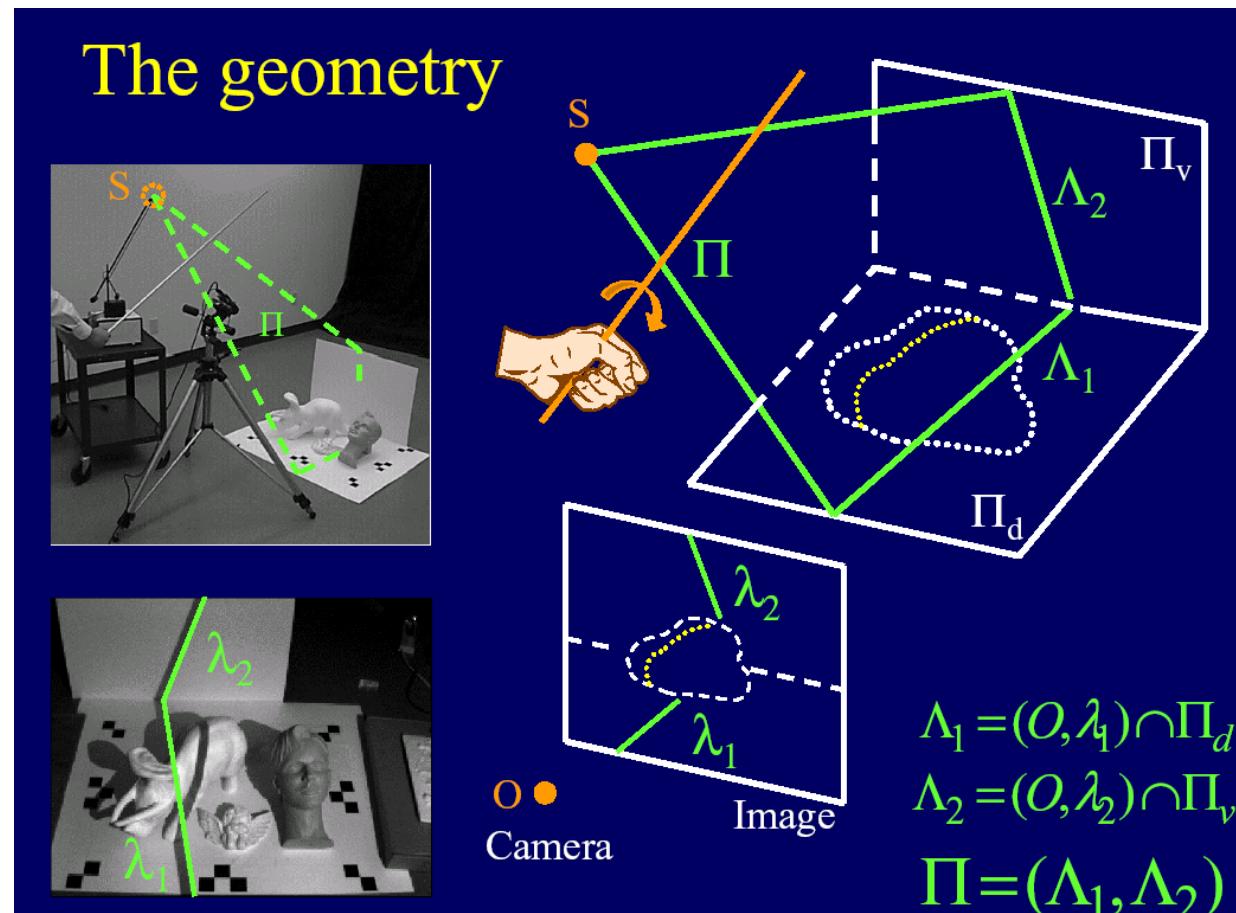


**Ex. de code pixel 1010010
→ identification du plan**



Lumière structurée par plans et ombrage - Exercice

Explicit the methodology
to reconstruct the scene





Caméra active

- Concept du télémètre laser – Fonctionnel en extérieur
- Exemples :
 - « Swiss ranger » SR4000 : temps de vol, résolution 176x144 pixels, 20 fps.
 - Kinect v2 : temps de vol, résolution 512x424 pixels, 30 fps
 - PMDVision CamCube : modulation de phase, résolution 204x204 pixels, 20 fps
 - Intel RealSense : stéréovision passive avec éclairage (IR), 640x480 pixels, 30/60 fps

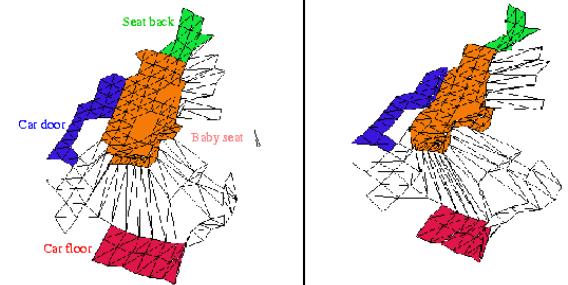
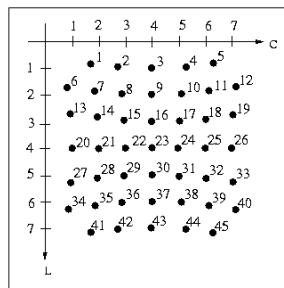
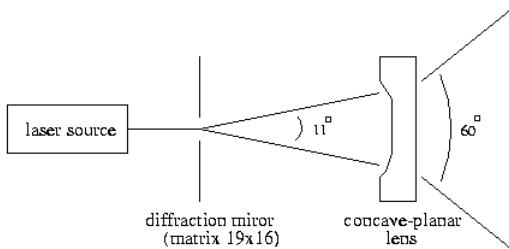


REALSENSE



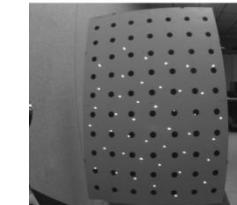
Lumière structurée - Exercice

- Exercice : reconstruction par triangulation active basée sur diode laser, grille de diffraction et caméra CCD



- Montrer que les équations des faisceaux sont donnés par l'expression suivante et proposer une procédure de calibrage

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

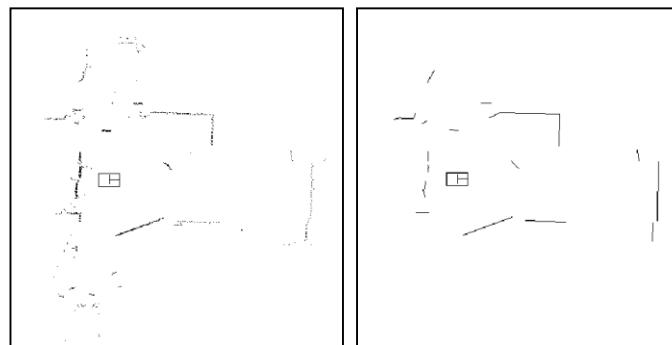


- Quelles contraintes pour réaliser les appariements entre faisceaux laser et spots image pour la reconstruction ?



Télémétrie laser - Exercice

- Exercice : un capteur multi-sensoriel pour la navigation d'un robot mobile en intérieur embarquant un banc 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
 - La segmentation laser consiste à regrouper les points successifs alignés. Proposer une démarche
 - Calibrer le système laser/banc stéréo. Les capteurs sont supposés dans un plan.
 - Comment mettre en correspondance les informations issues des deux capteurs ?





Références

- Point Cloud Library, url <http://pointclouds.org/>
- OpenCV library, url <http://opencv.org/>
- Perception visuelle par imagerie vidéo. Dhome *et al.*, Hermès&Lavoisier, 2003
- Vision par Ordinateur. Horaud & Monga, Hermès, 1993
- Three Dimensional Computer Vision. A Geometric Viewpoint. Faugeras, MIT Press, 1993.

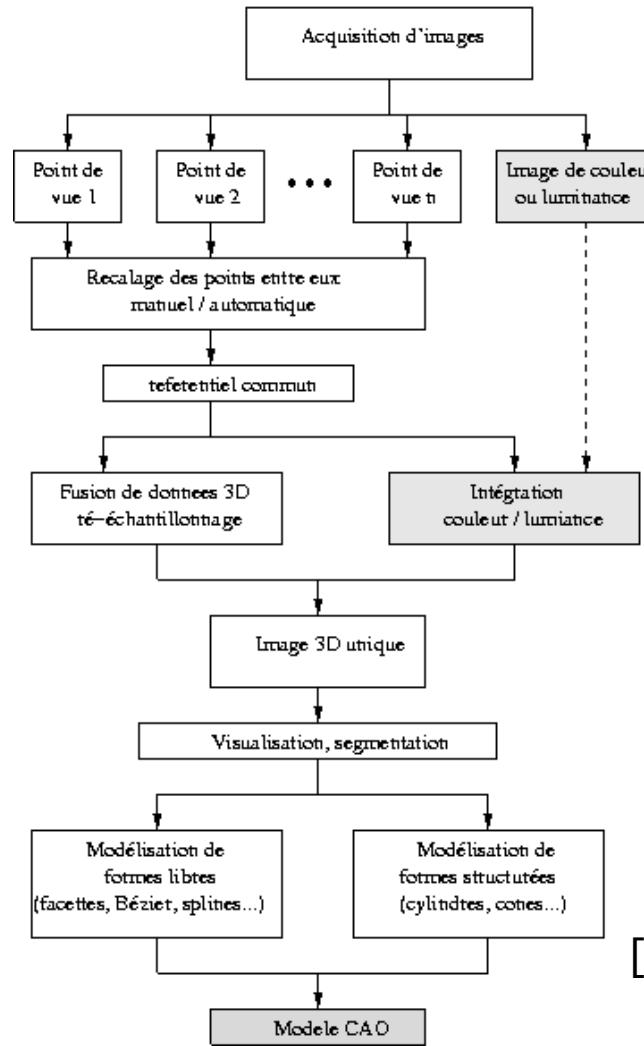


Sommaire (18h C/TD et 12h TP)

1. Acquisition 3D
2. Modélisation 3D (3h)
 1. Généralités
 2. Modélisation de formes libres / structurées
 3. Segmentation 3D
 4. Modélisation incrémentale par perception active
3. Localisation et reconnaissance 3D
4. Applications par des exemples
5. Correction annale



Algorithme générique

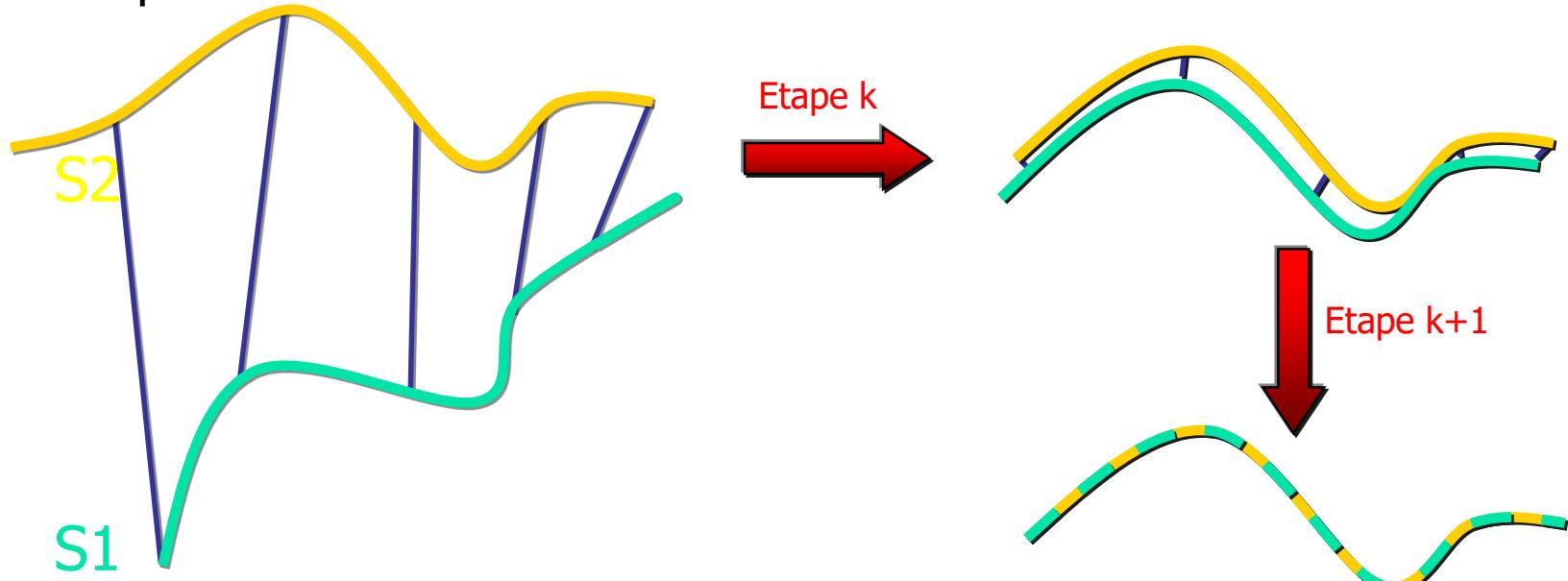


[vidéos de Noomeo et Kinect fusion]



Recalage automatique

- Estimation (R, T) à partir d'appariements entre ensemble de points S_1 et S_2 **en recouvrement**



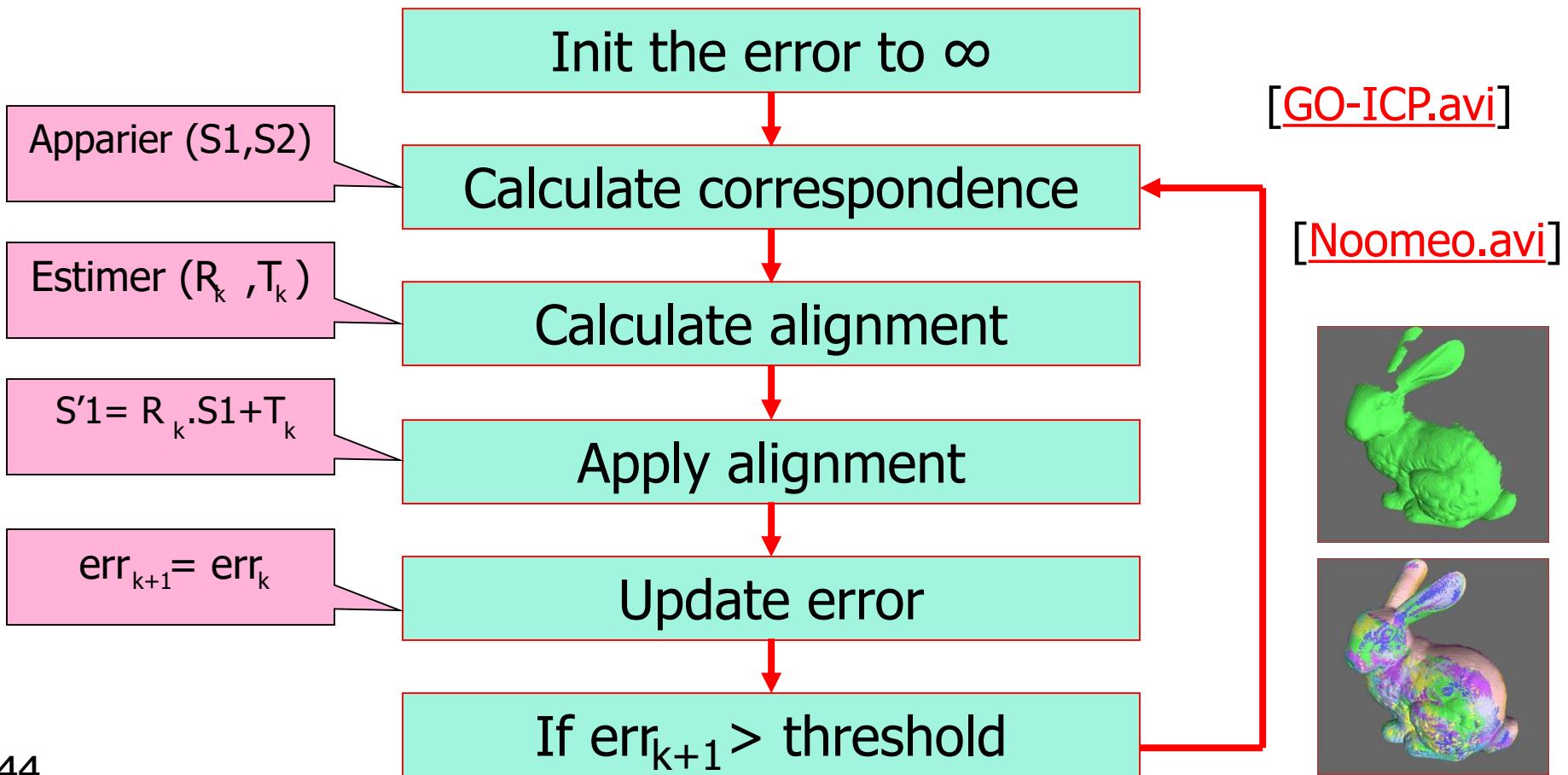
- Applications à l'ensemble de points S_1 et évaluation de la qualité du recalage :

$$err = \text{mediane} \left[\min_{X_1 \in S_1} \min_{X_2 \in S_2} \|R_k X_1 + T_k - X_2\| \right]$$

[\[Reconstruction-Kinect.avi\]](#)

Recalage automatique

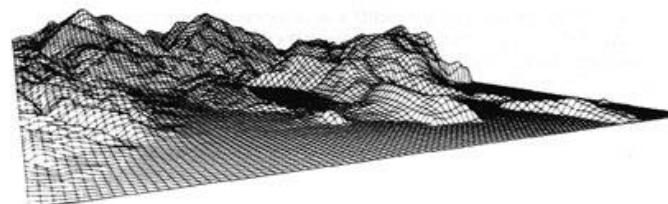
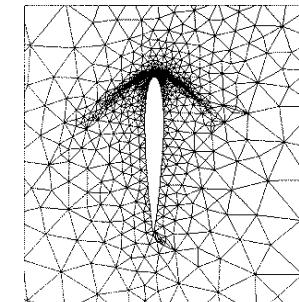
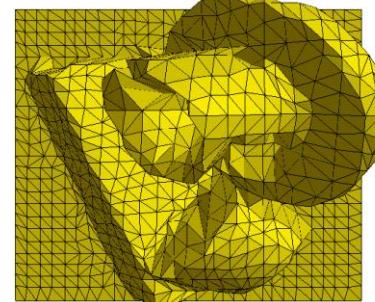
- Algorithme générique « Iterative Closest Point » (ICP)





Modélisation de formes libres

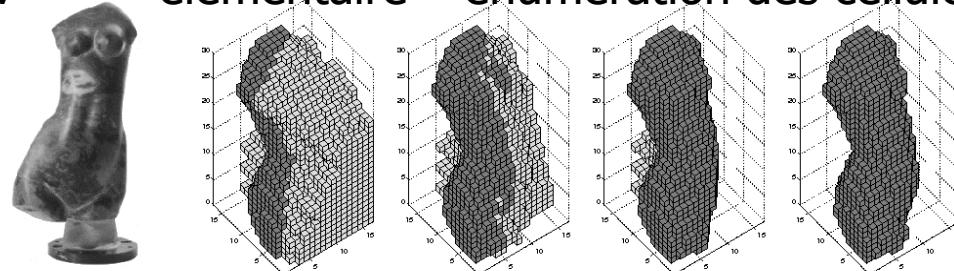
- Absence de modèles mathématiques...
- Modèles par facettes
 - Maillage par facettisation – rectangulaire, triangulaire, etc. – régulier ou non, hiérarchique ou non – adapté à tout type de scène
 - Exemple : le modèle numérique de terrain (MNT) – élévation z sur une grille de mesure (x,y) – modèle dit $2D^{1/2}$





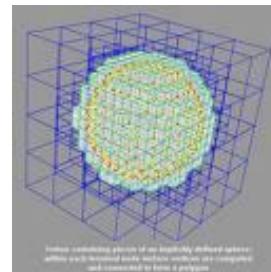
Modélisation de formes libres

- Représentation par éléments volumiques
 - « Voxels » *i.e.* volume élémentaire – énumération des cellules occupées



Vue dessus coté droit de la statuette - triangulation^(b) active - espace de travail : 17x17x30 voxels - arête : 5mm

- « Octrees » *i.e.* cubes de tailles variables

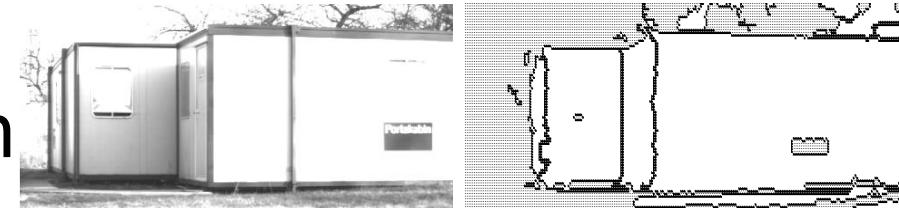


[[TableTop.avi](#)]

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

Modélisation de formes structurées

- Modèles mathématiques simples avec peu de paramètres (<10)
- Exemple #1 : le plan



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

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

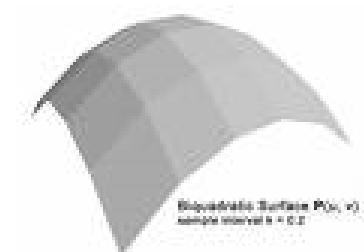
- Exemple #2 : la surface biquadratique

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

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

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

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



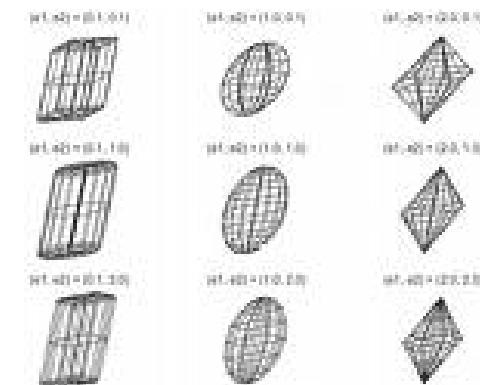


Modélisation de formes structurées

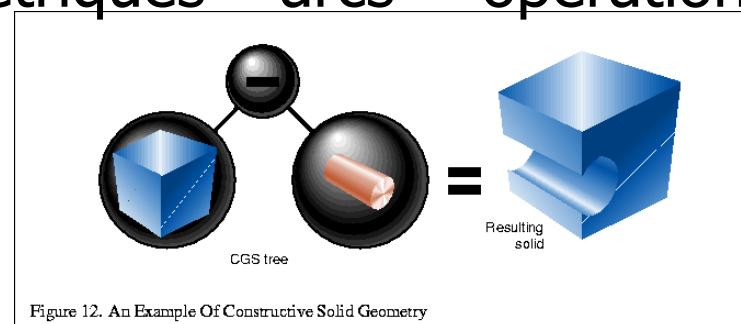
■ Primitives géométriques

- Sphères, cylindres, cônes, tores, ...
- Superquadriques

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



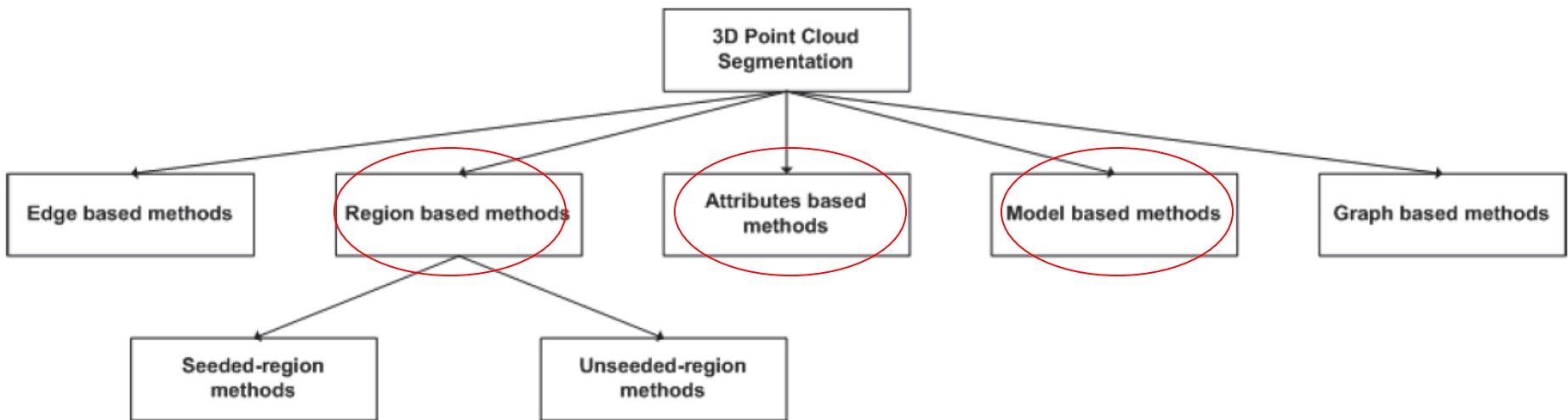
- Structuration via des arbres de « Constructive Solid Geometry » - noeuds = éléments géométriques – arcs = opérations ensemblistes





Segmentation 3D

- Taxonomie des approches [Nguyen *et al.*, 2013]

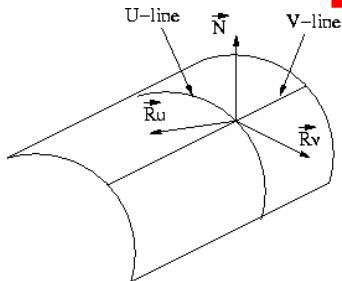


- Quid des attributs ?



Segmentation 3D

- Attribut normale
 - Adaptée aux surfaces planes
 - Calculée après facettisation sinon par tangentes locales
 - Calcul par masques de Sobel, Kirsch



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

$$x_u = \frac{\partial x}{\partial u}, y_u = \frac{\partial y}{\partial u}, z_u = \frac{\partial z}{\partial u}, \text{ etc.}$$

- Attribut courbure
 - Adaptée aux surfaces de degré supérieur
 - Calcul des rayons de courbures principaux R1 et R2

$$(LN - M^2)R^2 + (E - 2FM + GL)R + (EG - F^2) = 0, \begin{bmatrix} E & F \\ F & G \end{bmatrix} = \begin{bmatrix} \vec{R}_u \cdot \vec{R}_u & \vec{R}_u \cdot \vec{R}_v \\ \vec{R}_u \cdot \vec{R}_v & \vec{R}_v \cdot \vec{R}_v \end{bmatrix}, \begin{bmatrix} L & M \\ M & N \end{bmatrix} = \begin{bmatrix} \vec{R}_{uu} \cdot \vec{N} & \vec{R}_{uv} \cdot \vec{N} \\ \vec{R}_{uv} \cdot \vec{N} & \vec{R}_{vv} \cdot \vec{N} \end{bmatrix}$$

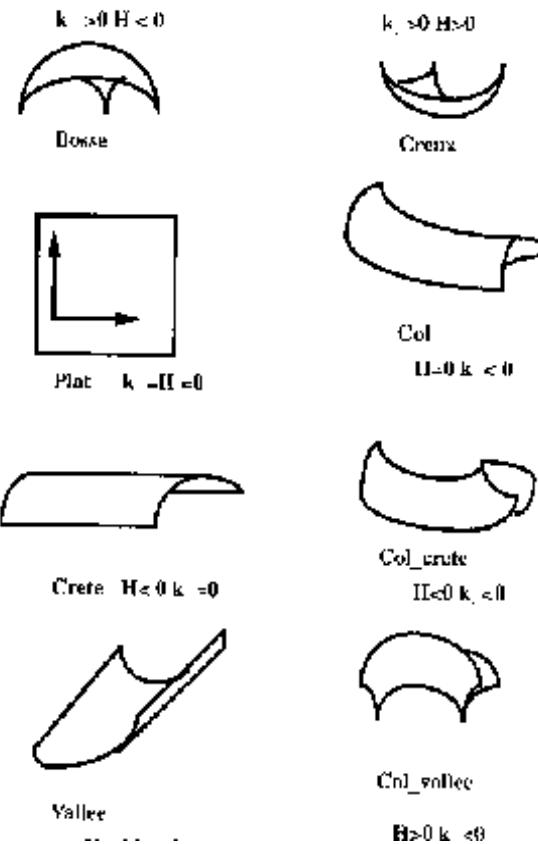
$$\vec{R}_{uu} = (x_{uu}, y_{uu}, z_{uu}), \vec{R}_{vv} = \dots, x_{uu} = \frac{\partial^2 x}{\partial^2 u}, y_{uu} = \frac{\partial^2 y}{\partial^2 u}, \text{ etc.}$$



Segmentation 3D

■ Attribut courbure

- Courbure de gauss K et moyenne H : $K = \frac{1}{R_1 \cdot R_2}, H = \frac{\frac{1}{R_1} + \frac{1}{R_2}}{2}$





Segmentation 3D

■ Approche basée **régions**

- But : segmenter en région (dis-similarité des attributs)
- Détermination des sauts/marches de profondeur (ordre 0), discontinuité de normale (ordre 1), discontinuité de courbure (ordre 2)
- Trois démarches : ascendante, descendante ou hybride

Début

Init.

Faire

Nb_points=Nb_points+1

Mise à jour paramètres modèle

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

Fin

Début

Sélection de régions élémentaires R (Type, Nb_points)

Estimation des paramètres

Tri des régions R

Construction liste des régions/points connexes L_R

Faire

Examiner résidu pour R fusionnée avec L_R

Tant que fusion=TRUE

Fin



Références

- Point Cloud Library, url <http://pointclouds.org/>
- OpenCV library, url <http://opencv.org/>
- **[Golovinsky *et al.*, 2009]** Min-cut based segmentation of point clouds. Golovinsky & Funkhouser. IEEE Int. Conf. on Computer Vision, 2009.
- **[Goulette, 1999]** Modélisation 3D automatique, outils de géométrie différentielle. Goulette. Les Presses de l'Ecole des Mines, 1999.
- **[Nguyen *et al.*, 2013]** 3D Point Cloud Segmentation: a Survey. Nguyen, and Le. IEEE Int. Conf. on Robotics, Automation, and Mechatronics, 2013.



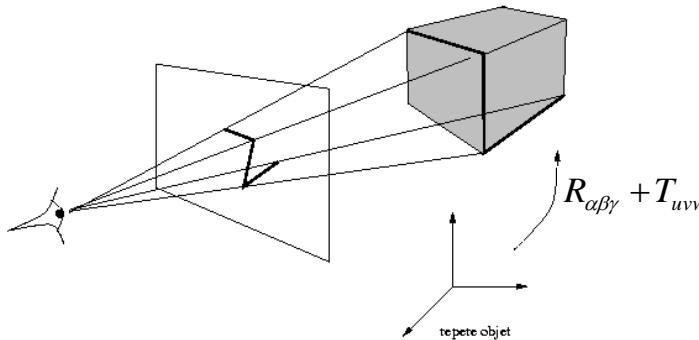
Sommaire (18h C/TD et 12h TP)

1. Acquisition 3D
2. Modélisation 3D
3. Localisation et reconnaissance 3D (5h)
 1. Localisation
 2. Reconnaissance
 3. Application à la manipulation
4. Applications par des exemples
5. Correction annale



Localisation 2D/3D

- But : détermination des paramètres extrinsèques connaissant
 - Les paramètres intrinsèques
 - Un modèle 3D de la scène
 - Des appariements entre primitives modèle et visuelles



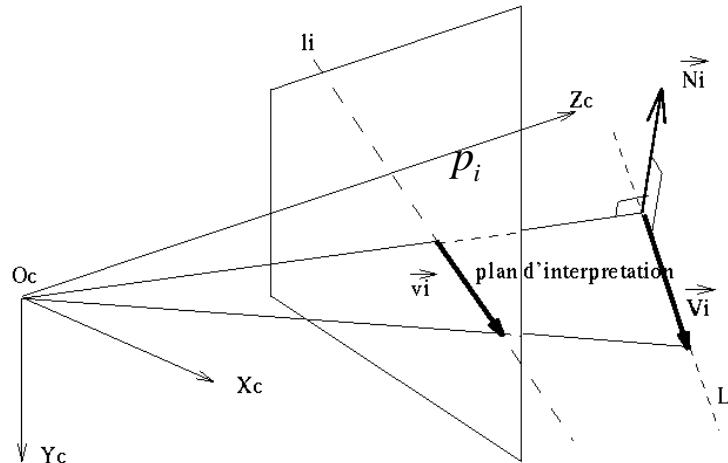
- Classification des approches selon :
 - Le modèle de projection : orthographique, **perspective**
 - La nature des primitives appariés : points, **droites**, cercles
 - La méthode de résolution : **analytique** e.g. P3P, P3L, Pose-IT, **numérique** e.g. P-n-L



Localisation 2D/3D : « PnL »

- Localisation « Perspective-3-Lignes » (P3L) :
 - Principe [Dhome *et al.*, 2003] :

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



- Résolution :

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

8 solutions théoriques \Rightarrow 3 solutions plausibles !



Localisation 2D/3D : « PnL »

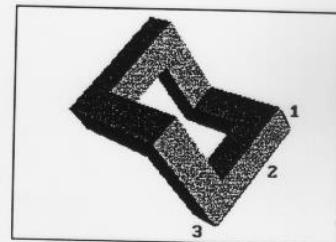


Figure 3a : Segments sélectionnés.

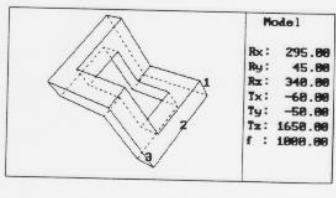


Figure 3b : Arêtes correspondantes.

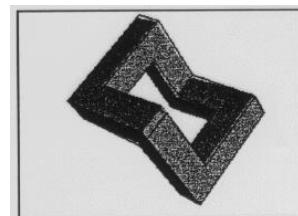
arêtes non visibles

	Sol.81 Rx: 296.34 Ry: 18.62 Rz: 149.88 Tx: -124.11 Ty: 185.56 Tz: -2480.94 f : 1898.88
	Sol.83 Rx: 69.14 Ry: 317.78 Rz: 156.48 Tx: 295.25 Ty: 532.29 Tz: 2286.38 f : 1898.88
	Sol.84 Rx: 296.34 Ry: 18.62 Rz: 149.88 Tx: -124.11 Ty: -185.56 Tz: 2480.94 f : 1898.88
	Sol.86 Rx: 296.86 Ry: 42.22 Rz: 326.48 Tx: 51.56 Ty: -78.66 Tz: 1646.53 f : 1898.88
	Sol.87 Rx: 118.06 Ry: 42.22 Rz: 336.48 Tx: 51.66 Ty: 78.66 Tz: -1646.53 f : 1898.88
	Sol.88 Rx: 73.66 Ry: 349.38 Rz: 329.88 Tx: -133.17 Ty: 451.27 Tz: -1654.32 f : 1898.88

Tz < 0

recouvrement

solution après
filtrage



Localisation 2D/3D : « PnL »

- Perspective-n-Lignes : résolution numérique
 - Critère à minimiser : $\varepsilon = \sum_{i=1}^n F(X, \overrightarrow{N_i}, P_i)^2 = V^T \cdot V$
 - Résolution : $\min_X (V^T V), V = A \cdot \Delta X - L, \Delta X = (A^T A)^{-1} \cdot A^T \cdot L$
 - Calcul des dérivées

$$\frac{\partial R_\alpha}{\partial \alpha} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{pmatrix}, \quad \frac{\partial R_\beta}{\partial \beta} = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{pmatrix}, \quad \frac{\partial R_\gamma}{\partial \gamma} = \begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \quad A = \begin{bmatrix} \frac{\partial F_1}{\partial \alpha} & \cdots & \frac{\partial F_1}{\partial T_z} \\ \vdots & \ddots & \vdots \\ \frac{\partial F_n}{\partial \alpha} & \cdots & \frac{\partial F_n}{\partial T_z} \end{bmatrix}, \quad L = \begin{bmatrix} -F_1(X_0) \\ \vdots \\ -F_n(X_0) \end{bmatrix}$$

- Critère d'arrêt ? Condition initiale ?

[\[Diotasoft-PnL\]](#)

[\[Diotasoft-PnL+ICP\]](#)



Localisation 2D/3D : « PnL »

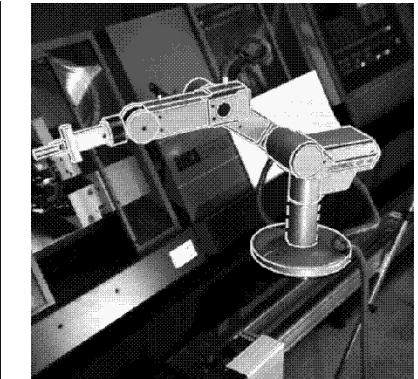
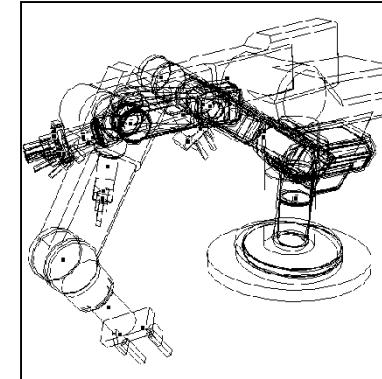
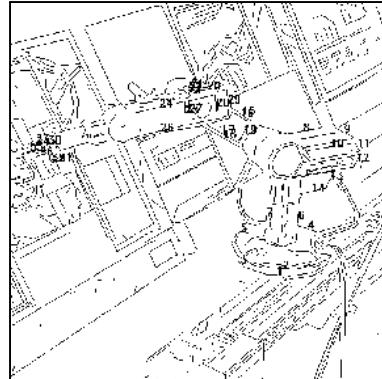
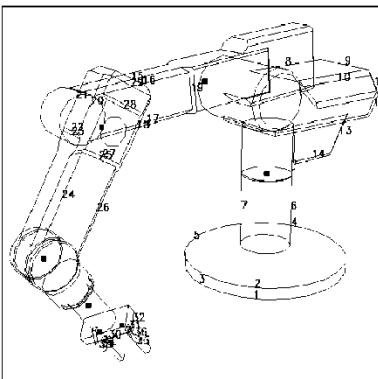
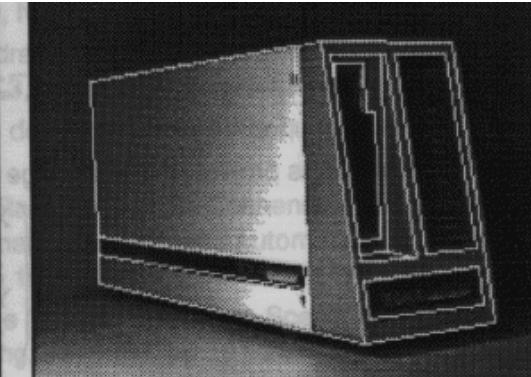
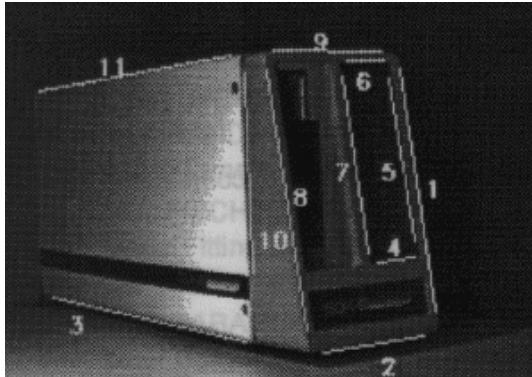
- Précision *vs.* nombre de caméras

	α	β	γ	Tx	Ty	Tz
1 caméra	0.8°	1.1°	0.1°	4.7mm	3.2mm	29.9m m
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



Localisation 2D/3D : « PnL »

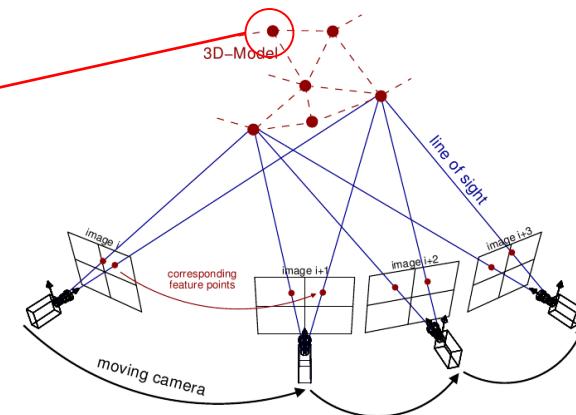
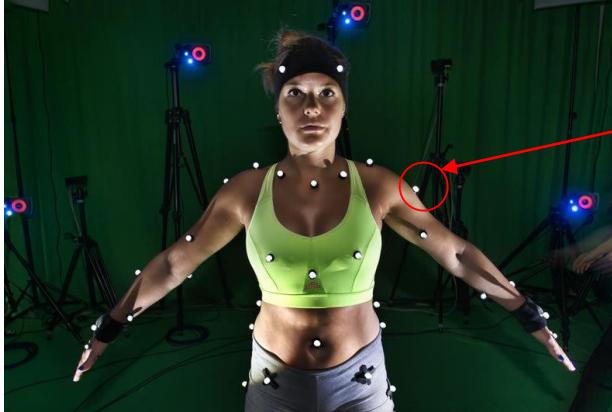
- Exemples sur objet rigide et articulé





Capture optique de mouvement humain

- Localisation PnP, n caméras étalonnées
 - Modèle 3D (segments corporels)
 - Appariements de marqueurs passifs/actifs
 - Localisation 3D des marqueurs à qq mm



[[MoCap-OptiTrack](#)]



Localisation 2D/3D : « PnL » - Exercice

■ 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 d'étalonnage de caméra vue précédemment pour la localisation monoculaire à partir d'appariements de points.



Localisation 2D/3D : « Pose-IT »

- Solution approchée [Dhome *et al.*, 2003]
 - Même hypothèse que PnP mais :
 - Pas de jacobienne à inverser
 - Pas de conditions initiales
 - Passage progressif modèle de projection orthographique à l'échelle vers perspectif
 - **But :** Estimer les extrinsèques en posant :

$$I = (r_{11}, r_{12}, r_{13}), J = (r_{21}, r_{22}, r_{23}), K = (r_{31}, r_{32}, r_{33})$$



Localisation 2D/3D : « Pose-IT »

■ Algorithme :

❶ $t=0$ et $\varepsilon_i(t)=0$ pour $i \in [1, n]$, en posant $I^* = \frac{f}{T_z}(I, u)$, $J^* = \frac{f}{T_z}(J, v)$

❷ calcul des vecteurs I^* et J^* par résolution de syst. linéaire

$$A = \begin{bmatrix} X_m^1 & Y_m^1 & Z_m^1 & 1 \\ \vdots & \vdots & \vdots & \vdots \\ X_m^n & Y_m^n & Z_m^n & 1 \end{bmatrix}, S_x = \begin{bmatrix} x_1(1 + \varepsilon_1(t)) \\ \vdots \\ x_n(1 + \varepsilon_n(t)) \end{bmatrix}, S_y = \begin{bmatrix} y_1(1 + \varepsilon_1(t)) \\ \vdots \\ y_n(1 + \varepsilon_n(t)) \end{bmatrix}$$

❸ $N_I = \sqrt{I_1^2 + I_2^2 + I_3^2}$, $N_J = \sqrt{J_1^2 + J_2^2 + J_3^2}$

$$I = (I_1 \quad I_2 \quad I_3) / N_I, J = (J_1 \quad J_2 \quad J_3) / N_J, K = I \wedge J, J = K \wedge I$$

$$T_z = \frac{f}{N_I}, T_x = I_4 \cdot \frac{T_z}{f}, T_w = J_4 \cdot \frac{T_z}{f}$$

❹ $t = t+1$; calcul des nouveaux $\varepsilon_i(t) = \frac{K \cdot P_m^i}{T_z}$

❺ $|(\varepsilon_i(t) - \varepsilon_i(t-1)) / n| < \text{seuil}$ alors stop, sinon retour en ❷

[[Pose-IT.avi](#)]



Localisation 3D/3D

- Appariements de N points 3D scène *vs.* modèle, resp.
 $\{p_i^s, p_i^m\}_{i=1,\dots,N}$
- Estimation par MC non linéaire

$$\text{Arg} \min_{R,T} \sum_{i=1}^N \|p_i^s - R.p_i^m - T\|$$

[\[Rob-spatiale.mp4\]](#)

[\[Head-Pose-Li.mp4\]](#) [\[Fanuc-3DL.mp4\]](#)

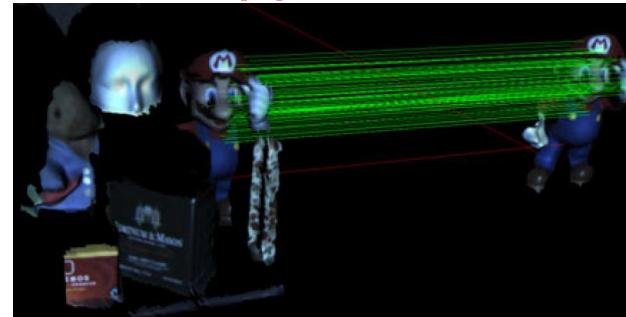
Sommaire (18h C/TD et 6h TP)

1. Auto-étalonnage de caméra
2. Modélisation 3D
3. Localisation et reconnaissance 3D
 1. Localisation : PnP, PnL, Pose-IT
 2. Reconnaissance
4. Applications par des exemples
5. Correction annale



Reconnaissance 3D/3D

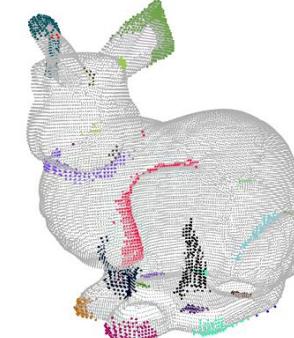
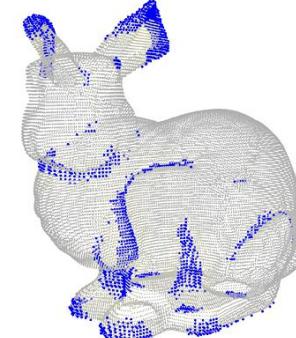
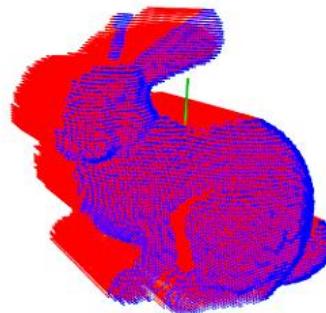
- But : (1) appariement automatiques de primitives scène/modèle, (2) localisation
- Complexité pour apparter : si O ensemble des primitives 3D modèle (de taille n) et D ensemble des primitives 3D scène (de taille m) tel que $m \geq n$.
Nombre d'appariements possibles : $C_m^n \cdot n! = \frac{m!}{(m-n)!}$
→ Problème combinatoire !!
- Extraction de primitives clés, calcul de descripteurs locaux ou globaux associés, appariements, prédiction / vérification de pose





Reconnaissance 3D/3D

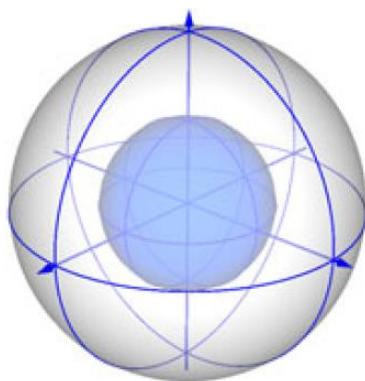
- Primitives 3D = ici nuages de points
- Extraction de points clés *i.e.* robuste au bruit, variation de point de vue, occultations, discriminant
- Détection : **LSP**, **ISS**, KPQ, etc. [Salti *et al.*, 2011]
 - LSP = « Local Surface Patch »
 - Extremum local de courbure SI dans un voisinage
 - ISS = « Intrinsic Shape Signature »
 - Calcul de matrice de dispersion et comparaison des valeurs propres





Reconnaissance 3D/3D

- Descripteurs locaux : **SHOT** [Tomari *et al.*, 2010], spin image, PFH, 3D shape context [Kortgen *et al.*, 2003]
 - SHOT = « Signature of Histograms Orientation »
 - Définition d'un repère local invariant


$$M_I = \frac{1}{k} \sum_{i=0}^k (p_i - \bar{p})(p_i - \bar{p})^T, \bar{p} = \frac{1}{k} \sum_{i=0}^k p_i$$

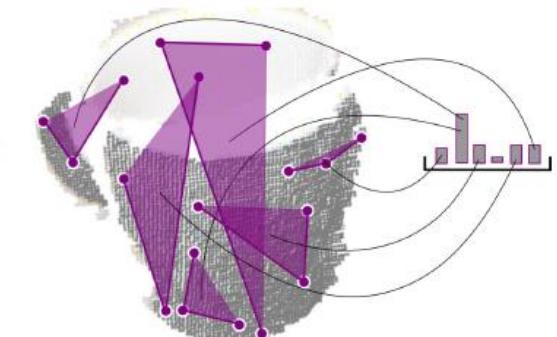
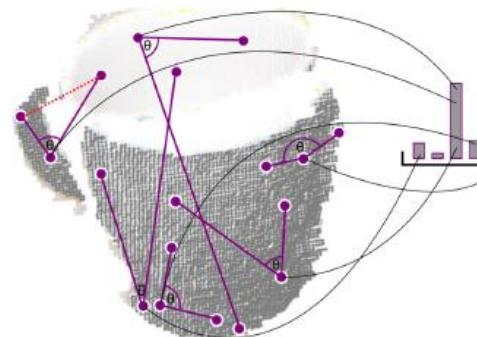
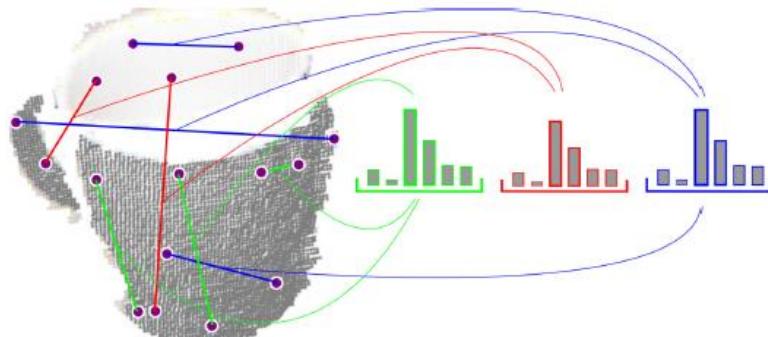
- Définition d'un voisinage (grille sphérique, 32 divisions)
- Histogramme (352 cellules) de différence d'orientation θ entre normales

$$\cos \theta_i = n_p \cdot n_{p_i} \quad \underline{h} = (h_{c_1}, h_{c_2}, \dots, h_{c_n}), h_{c_i} = \sum_{u \in R} \delta_{c_i}(b_u), b_u \in \{c_1, \dots, c_n\}, \sum_{i=1}^n h_{c_i} = 1$$



Reconnaissance 3D/3D

- Descripteurs globaux : PFH, VPFH, CVFH, **ESF**
[Alexandre, 2012][Brkié et al., 2014]
 - ESF = « Ensemble of Shape Function »
 - Pas de prétraitement (calcul normales)
 - Voxellisation
 - Calcul de histogrammes sur angle, distance entre points, aire, etc. avec distinguo dans/hors surface objet ou les deux.
 - $64 \times 3 \times 3 = 576$ cellules au total





Reconnaissance 3D/3D

- Processus d'appariements de points clés
 - (1) Détection de points clé robustes et calcul de descripteurs saillants
→ Définition préalable d'un voisinage
 - (2) Appariement primitives modèle/scène
 - Par mesures de similarités entre vecteurs de descripteurs $\underline{h}^1, \underline{h}^2$: distance (normes L₁, L₂, L_∞), produit scalaire

$$d_{L_1}(h^1, h^2) = \sum_{k=1}^M |h^1(k) - h^2(k)|$$

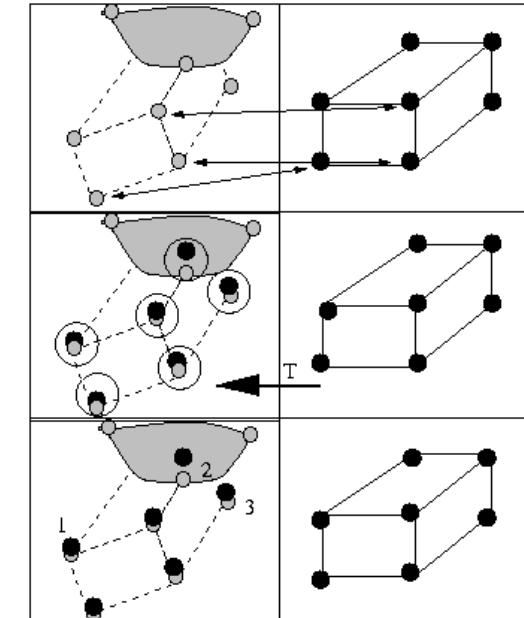
$$d_{L_2}(h^1, h^2) = \sqrt{\sum_{k=1}^M (h^1(k) - h^2(k))^2}$$

$$d_{L_\infty}(h^1, h^2) = \max_{1 \leq k \leq M} |h^1(k) - h^2(k)|$$



Reconnaissance 3D/3D

- Prédiction / vérification :
 - Hypothèse de localisation
 - Vérification
 - ⇒ appariements et localisation résolus simultanément
- Étape 1 : prédiction
 - Établir quelques appariements
 - Estimer la localisation
 - Appliquer à l'ensemble des primitives
- Etape 2 : vérification
 - Former de nouveaux appariements
 - Fusionner nouveaux et précédents appariements
 - Mise à jour de la localisation

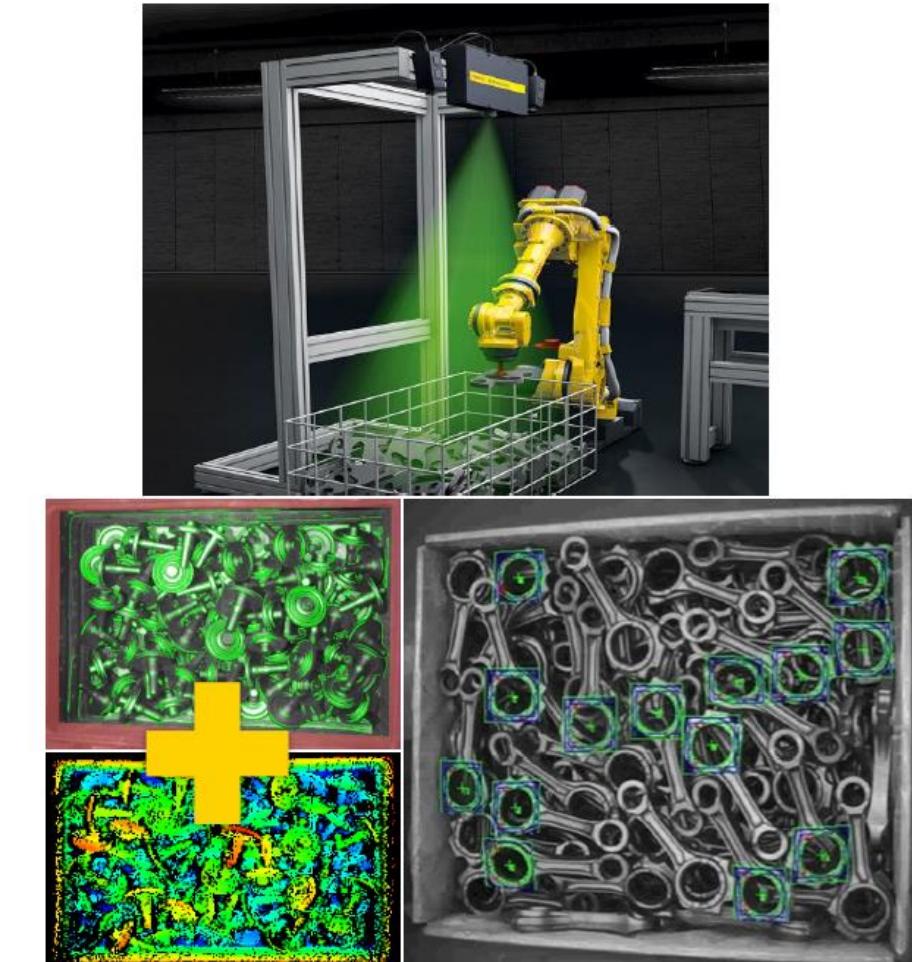




Robotique industrielle et dévracage - 1

- Stéréovision active (+ éclairage annulaire) déportée : nuages 3D, gestion des occultations
- Détection contours + template matching → segmentation 2D d'objets + sélection
- Extraction de P.I → localisation 3D/3D puis ICP
- Planification de prise sur objet priorisé

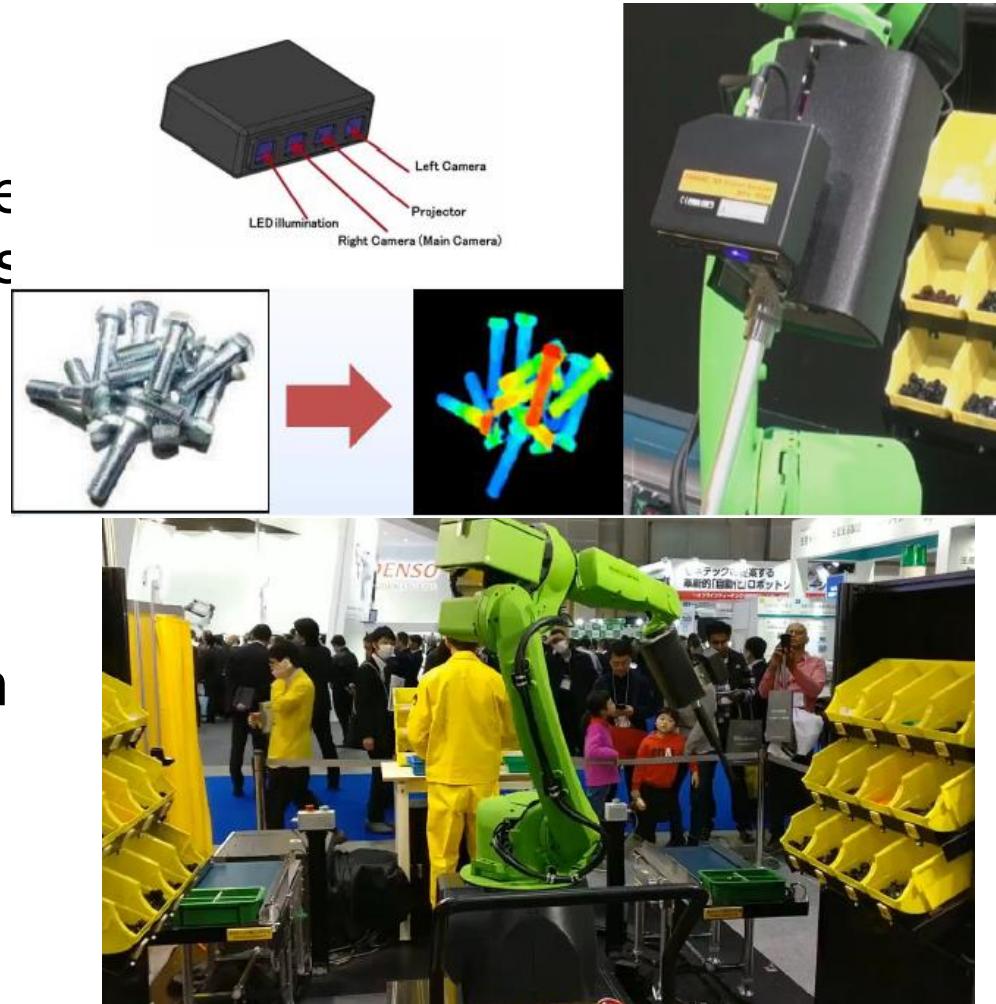
[\[Fanuc-devracement.mp4\]](#)





Robotique industrielle et dévracage - 2

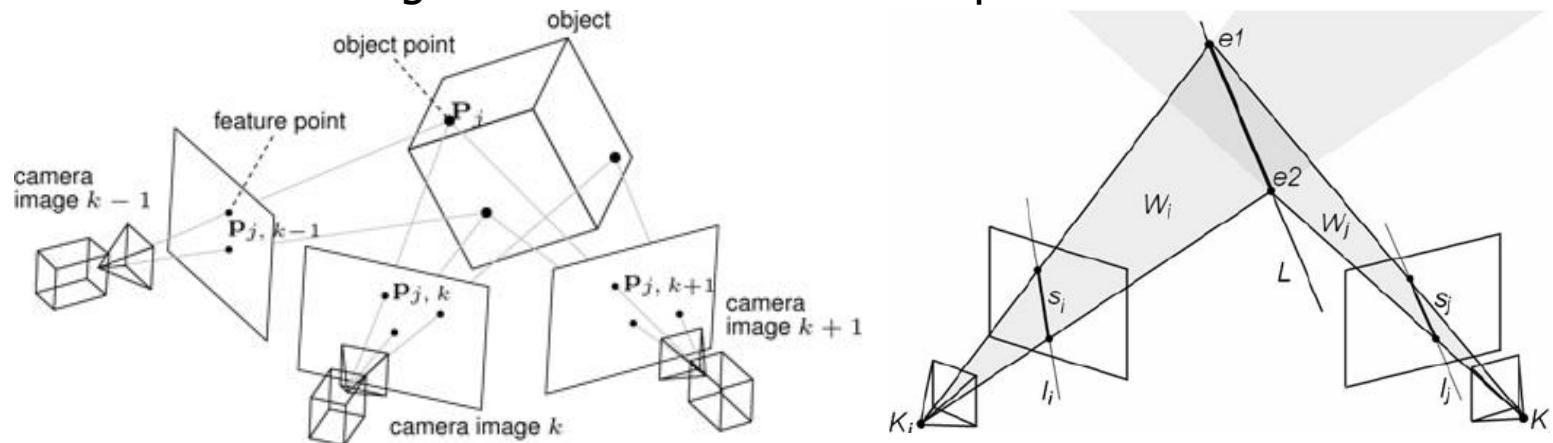
- Stéréovision active (+ éclairage annulaire) embarquée : nuages 3D, gestion des occultations
- Détection de contours + template matching → segmentation 2D d'objets et sélection
- Extraction de P.I → localisation 3D/3D puis ICP
- Planification de prise sur objet priorisé





Reconnaissance 2D/3D

- Démarche générique :
 - Requiert un modèle 3D, étalonnage caméra, hyp. localisation
 - Appariements 2D/3D
 - Par critères géométriques : proximité entre primitives 2D et primitives 3D après projection, longueur/orientation/recouvrement si primitives = arête modèle et segment image
 - Par critères apparence si modèle = géométrie + apparence
 - Calcul localisation 2D/3D → prédiction
 - Vérification globale sur l'ensemble des primitives modèle





Références

- Point Cloud Library, url <http://pointclouds.org/>
- OpenCV library, url <http://opencv.org/>
- **[Alexandre, 2012]** 3D Descriptors for Object and Category Recognition: a Comparative Evaluation. Alexandre. Int. Conf. on Intelligent Robotic Systems, 2012.
- **[Bay et al., 2006]** SURF: Speeded Up Robust Features. Bay, Tuytelaars, and Van Gool. European Conf. on Computer Vision, 2006.
- **[Brkié et al., 2014]** Temporal Ensemble of Shape Functions. Brkié, Aldoma, Vincze, Segvié, and Kalafatié. Eurographics Workshop on 3D Object Retrieval, 2014.
- **[Dhome et al., 2003]** Perception visuelle par imagerie vidéo. Dhome *et al.*, Hermès&Lavoisier, 2003.
- **[Johnson et al., 1998]** Surface Matching for Object Recognition in Complex 3-D Scenes. Johnson, and Hebert. Journal Image and Vision Computing, 1998.
- **[Knopp et al., 2011]** Hough Transforms and 3D SURF for robust 3D classification. Knopp, Prasad, Willems, Timofle, Van Gool. European Conf. on Computer Vision, 2011.
- **[Kortgen et al., 2003]** 3D Shape Matching with 3D Shape Contexts. Kortgen, Park, Novotni, and Klein. European Seminar on Computer Graphics, 2003.



Références

- **[Rusu *et al.*, 2009]** Fast Point Feature Histograms (FPFH) for 3D registration. Rusu, Blodow, and Beetz. Int. Conf. on Robotics and Automation, 2009
- **[Rusu *et al.*, 2010]** Fast 3D Recognition and Pose Using the Viewpoint Feature Histogram (VFH). Rusu, Bradski, Thibaux, Hsu. Int. Conf. on Intelligent Robotic Systems, 2010
- **[Salti *et al.*, 2011]** A Performance Evaluation of 3D Keypoint Detectors. Salti, Tombari, Di Stefano. Int. Conf. on 3D Imaging, Modeling, Processing, Vizualisation and Transmission, 2011.
- **[Tombari *et al.*, 2010]** Unique Signatures of Histograms (SHOT) for Local Surface Description. Tombari, Salti, and Di Stefano. European Conf. on Computer Vision, 2010.



Sommaire (18h C/TD et 12h TP)

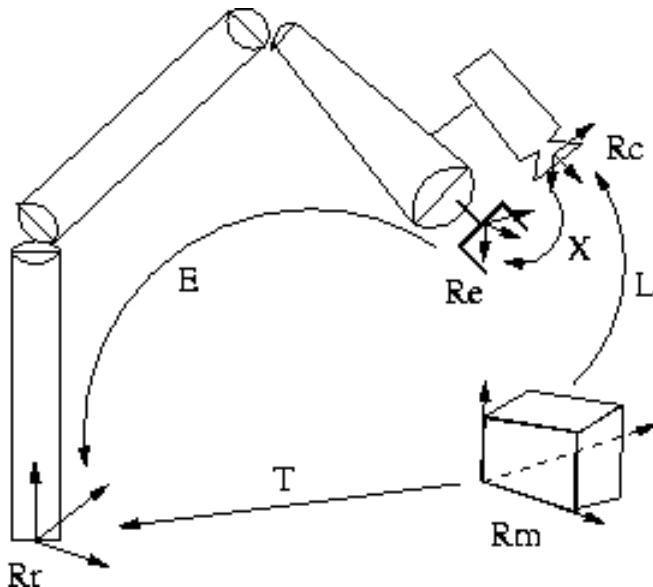
1. Acquisition 3D
2. Modélisation 3D
3. Localisation et reconnaissance 3D
4. Applications par des exemples
5. Correction annale



Etalonnage et manipulation d'objet

■ Calibration « bras / œil »

- But : guider le mouvement de la pince grâce à des infos visuelles → caractériser 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 ,

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



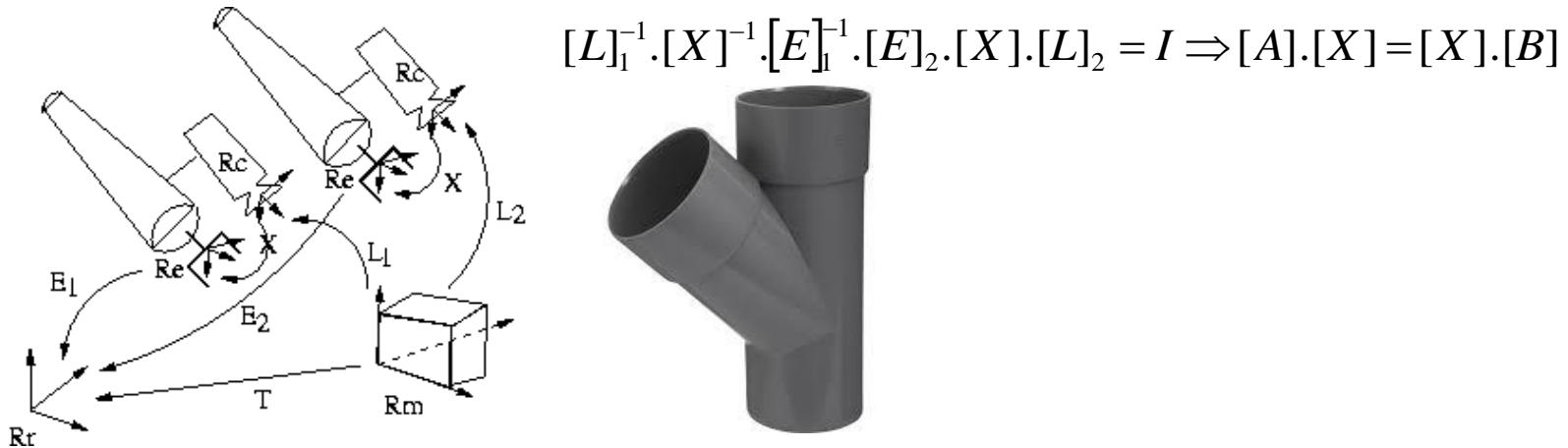
Etalonnage et manipulation d'objet

- Saisie de l'objet par une approche « look&move »

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

- Stratégie 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] :



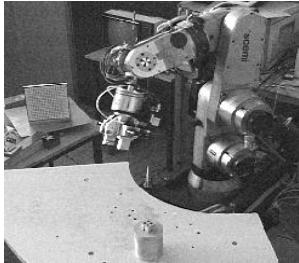
$$[L]_1^{-1} \cdot [X]^{-1} \cdot [E]_1^{-1} \cdot [E]_2 \cdot [X] \cdot [L]_2 = I \Rightarrow [A] \cdot [X] = [X] \cdot [B]$$



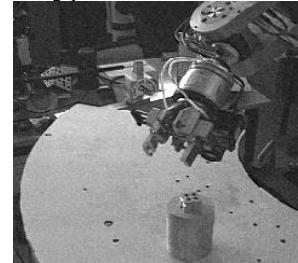
Reconnaissance et manipulation d'objet

■ Asservissement en position 3D :

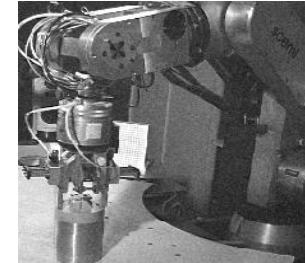
- se rapprocher le plus possible de la position de prise puis effectuer le dernier mouvement relatif de l'effecteur en aveugle
- une bonne localisation de l'objet
- un bon calibrage main/œil



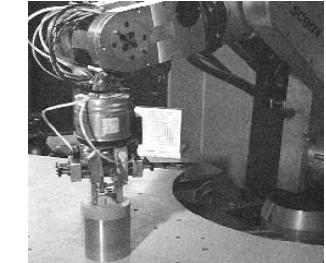
Site et position initiale



Dernière position d'approche



Position de saisie



Saisie de l'objet

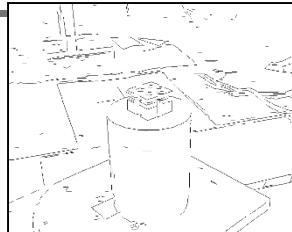
■ Étapes :

- ① acquisition+segmentation image
- ② prédiction de la localisation de l'objet (donnée par le modèle CAO du site)
- ③ mises en correspondance segments 2D/ arêtes 3D
- ④ déplacement du bras
- ⑤ retour en ①

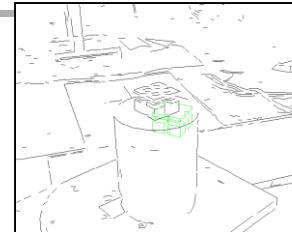
Reconnaissance et manipulation d'objet



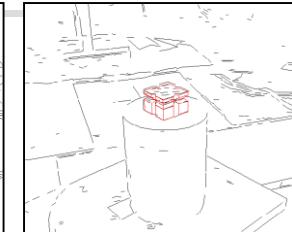
Image I



*Image I
segmentée*



*Prédiction
initiale*



Reconnaissance

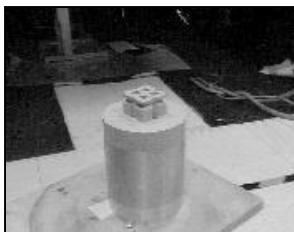
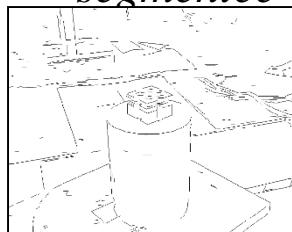
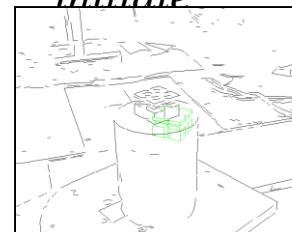


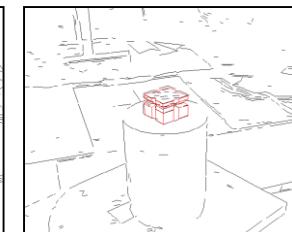
Image I



*Image I
segmentée*



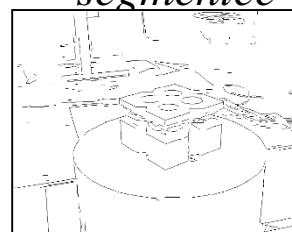
*Prédiction
initiale*



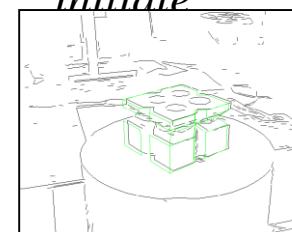
Reconnaissance



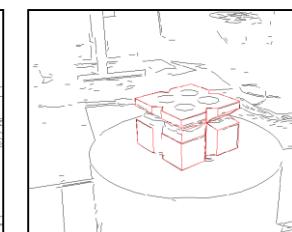
Image III



*Image III
segmentée*



*Prédiction
initiale*

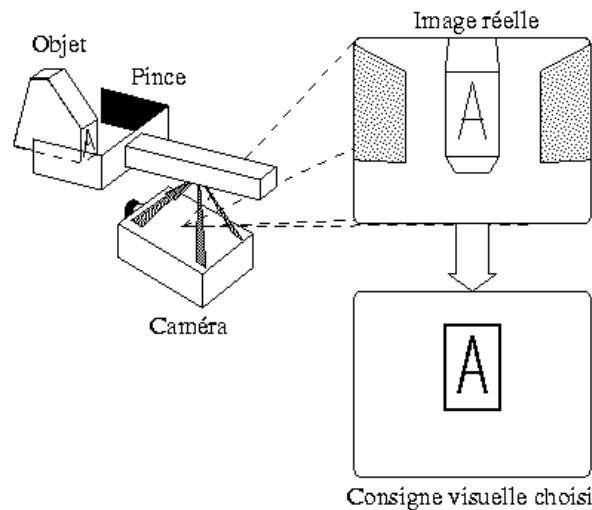


Reconnaissance



Reconnaissance et manipulation d'objet

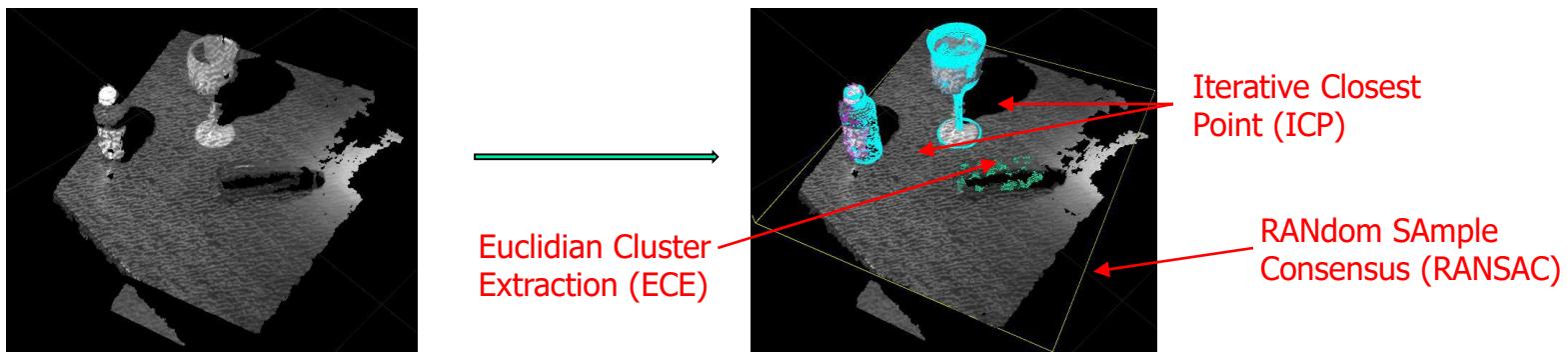
- Comment s'affranchir de la localisation 3D ?
→ l'asservissement visuel 2D *i.e.* asservissement sur une consigne image





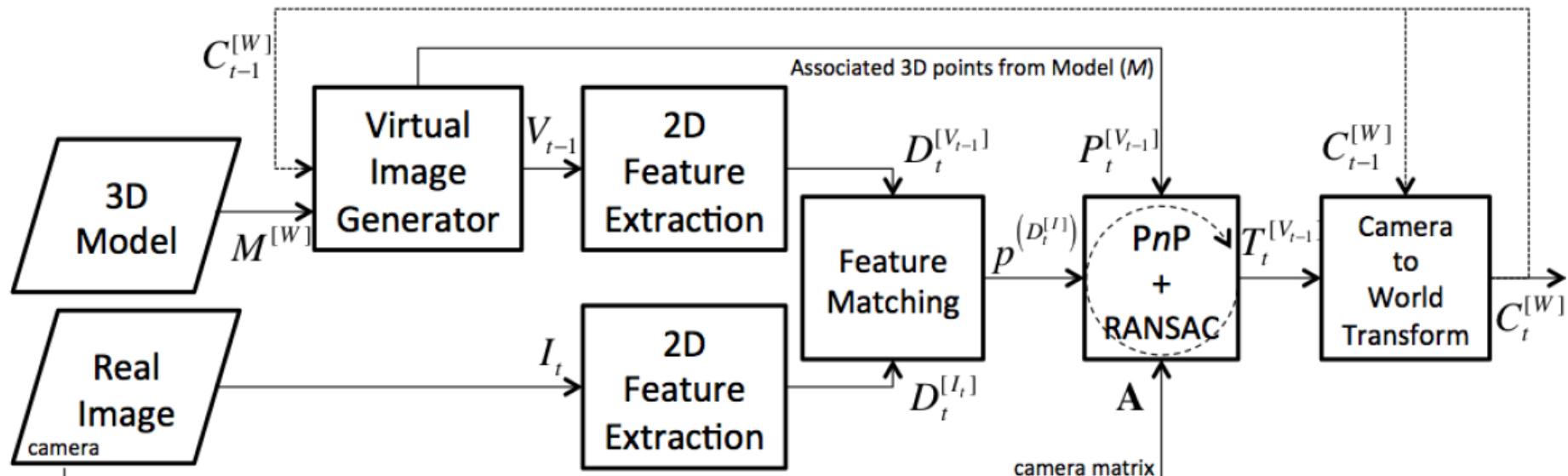
Algorithme « TableTop »

- « Inputs » : nuages de points 3D, modèle 3D des objets
- Hypothèses : objets sur une surface plane, objet symétrique par rotation, objets séparés
- Démarche, 3 étapes :
 - Random Sample Consensus (RANSAC) pour le plan
 - « Euclidian Cluster Extraction » (ECE) pour les objets
 - « Iterative Closest Point » (ICP)



Localisation monoculaire – « PnP »

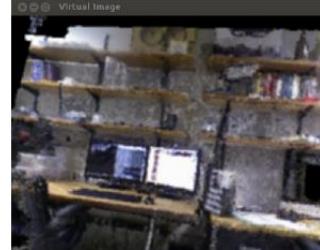
- Démarche [Jamarillo et al., 2013] :
 - (1) Création hors-ligne d'un modèle 3D RGB-D : ICP, etc.





Localisation monoculaire – « PnP »

- Démarche [Jamarillo et al., 2013] :
 - (2) Génération image virtuelle t-1 (loc. t-1)
 - (3) Appariements SURF entre t-1 et t
 - (4) Localisation PnP avec RANSAC $T_t^{[V_{t-1}]}$ et mise à jour localisation absolu $c_t^{[W]} = T_t^{[V_{t-1}]} \cdot c_{t-1}^{[W]}$



[\[Loca-SURF.avi\]](#)



Vision 3D et planification de trajectoire 3D

- Planification de trajectoire 3D avec incertitude
 - Estimée position 3D roue et coffre connue *a priori*
 - Reconnaissance 3D (roue, coffre) : appariements de P.I et localisation 3D/3D
 - Ré-estimation en ligne de la trajectoire eu égard à la localisation estimée

[[Fanuc-MotionPlanning.mp4](#)]



Références

- Tabletop, url http://www.ros.org/wiki/tabletop_object_detector
- **[Hinterstoisser et al., 2011]** Multimodal templates for real-time Detection of Texture-less Objects in Heavily Cluttered Scenes. Hinterstoisser, Holzer, Cagniart, Ilic, Konolidge, Navab, Lepetit. Int. Conf. on Computer Vision, 2011.
- **[Jamarillo et al., 2013]** 6-DoF pose localization in 3D point-cloud dense maps using a monocular camera. Jamarillo, Dryanovski, Valentin, Xiao. Int Conf. on Robotics and Biomimetics, 2013.



Références

- Point Cloud Library, url <http://pointclouds.org/>
- OpenCV library, url <http://opencv.org/>
- Perception visuelle par imagerie vidéo. Dhome *et al.*, Hermès&Lavoisier, 2003.
- Unique Signatures of Histograms (SHOT) for Local Surface Description. Tombari, Salti, and Di Stefano. European Conf. on Computer Vision, 2010.
- Fast Point Feature Histograms (FPFH) for 3D registration. Rusu, Blodow, and Beetz. Int. Conf. on Robotics and Automation, 2009
- 3D Shape Matching with 3D Shape Contexts. Kortgen, Park, Novotni, and Klein. European Seminar on Computer Graphics, 2003.
- Surface Matching for Object Recognition in Complex 3-D Scenes. Johnson, and Hebert. Journal Image and Vision Computing, 1998.
- Fast 3D Recognition and Pose Using the Viewpoint Feature Histogram (VFH). Rusu, Bradski, Thibaux, Hsu. Int. Conf. on Intelligent Robotic Systems, 2010
- Hough Transforms and 3D SURF for robust 3D classification. Knopp, Prasad, Willems, Timofte, Van Gool. European Conf. on Computer Vision, 2011.
- Temporal Ensemble of Shape Functions. Brkié, Aldoma, Vincze, Segvié, and Kalafatié. Eurographics Workshop on 3D Object Retrieval, 2014.
- 3D Descriptors for Object and Category Recognition: a Comparative Evaluation. Alexandre. Int. Conf. on Intelligent Robotic Systems, 2012.
- A Performance Evaluation of 3D Keypoint Detectors. Salti, Tombari, Di Stefano. Int. Conf. on 3D Imaging, Modeling, Processing, Vizualisation and Transmission, 2011.