UPS
Université Paul Sabatier
Toulouse 3

« 3D Perception » module

3A « Robotics and Interactive Systems » (SRI/UPSSITECH)

F.LERASLE

2023-2024

Summary (18h C/TD and 9h TP)

1. Camera self-calibration (2h)
2. Structure from motion (2h)
3. 3D modeling (4h)
4. 3D localisation and recognition (5h)
5. Applications by examples (3h)
6. Exercice correction (2h)
7. Practices :
    1. Incremental structure from motion (3h)
    2. 3D localisation (3h)
    3. Structure from motion (3h)

Camera calibration - Basics

- Intrinsic parameters: camera frame → image frame

$$\begin{cases} x = f.\frac{X_c}{Z_c} \\ y = f.\frac{Y_c}{Z_c} \end{cases} \qquad \begin{cases} u = k_u.x + u_0 \\ v = k_v.y + v_0 \end{cases}$$

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

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

- Extrinsic parameters: world frame → camera frame

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

- Parameters for distortion modeling *i.e.* (k_1, k_2, k_3) *for* $r_3 t_0, (k_1, k_2, k_3, p_1, p_2)$ *for* $r_3 t_2, \ldots$

Camera calibration - Basics

- Camera calibration
  - (+) Control, accurate
  - (-) A priori camera settings, not modifiable online
  - (-) Offline process, non automatic, requires a checkboard

→ self-calibration (online, automatic)

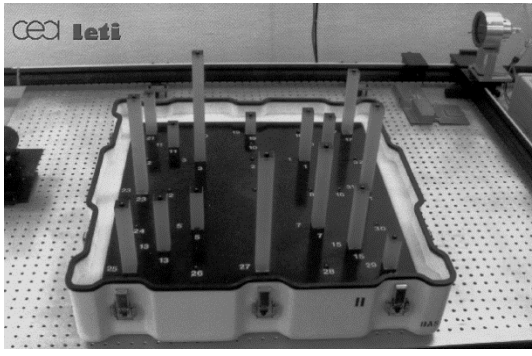
Towards the « self-calibration »

- Hypothesis: Connaissance approximatif de la scène
- Goal: re-estimate the 3D structure of the checkboard, jointly with the estimation of the intrinsic/extrinsic parameters [Dhome *et al.*, 2003]

$$X_{7+6*m+3*n} = (u_0, v_0, \alpha_u, \alpha_v, k_1, k_2, k_3, X_1, Y_1, Z_1, \ldots, X_n, Y_n, Z_n, \alpha^1, \beta^1, \gamma^1, t_x^1, t_y^1, t_z^1, \ldots, \alpha^m, \ldots, t_z^m)^T$$

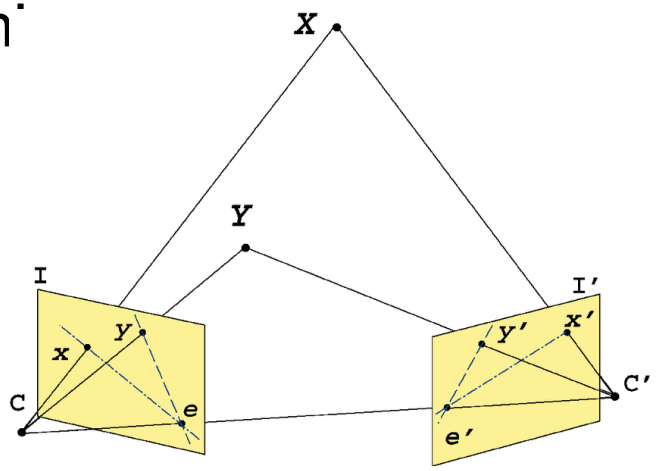
  - System redundancy: $r = 2*m*n - (7 + 3*n + 6*m)$
  - Reconstruction at a scale factor ➔ Fix two points in the pattern
  - Importance of the initialisation… ➔ estimate the parameters sequentially
  - Convergence criteria: $\overline{C(X)} = \frac{1}{10}\ pixel$

$$\min_X C(X) \ \textit{with}\ C(X) = \sum_{i=1}^{n} \left(V_{u_i}^2 + V_{v_i}^2\right)^2$$



Self-calibration – State-of-the-art

- Approaches' categorization [Skordas, 1995]
  - Moving camera, epipolar geometry characterization then intrinsic parameter identification
  - Sequence of images, inter-image matching, relative reconstruction method of scenes
  - Active vision *i.e.* known/accurate camera motions, inter-image match



Self-calibration by epipolar geometry

- Fundamental matrix
  - Applied to a moving camera (3) or a stereo syst.
  - Definition: $\begin{pmatrix} u_2 \\ v_2 \\ 1 \end{pmatrix} . F . \begin{pmatrix} u_1 \\ v_1 \\ 1 \end{pmatrix} = 0$
  - Estimation:
    - n inter-image matchings
    - Estimation MSE : $U^t . X = 0$

$$U = (u_1.u_2, v_1.u_2, u_2, u_1.v_2, v_1.v_2, v_2, u_1, v_1, 1)^t$$

$$\mathrm{X} = (F_{11}, F_{12}, F_{13}, F_{21}, F_{22}, F_{23}, F_{31}, F_{32}, F_{33})^t$$

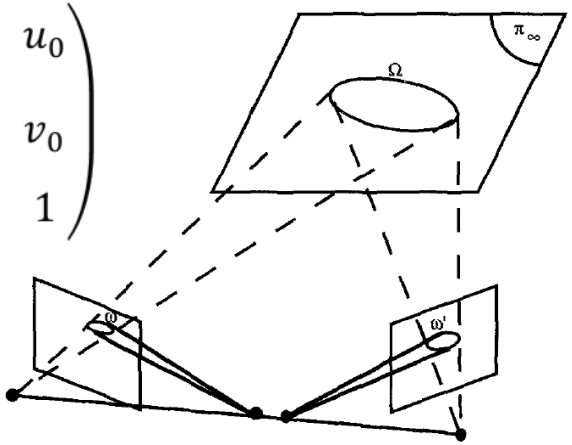
$$\min_X \left\| \widetilde{U}.X \right\|^2 \text{ with } \|X\|^2 = 1, \widetilde{U} = (U_1^t, \ldots, U_n^t)^t$$

Self-calibration by epipolar geometry

- Intrinsic identification
  - Concept of absolute conic and Kuppa's matrix B
  - Inference of B from F

→ 3 motions

$$\Delta = \det(B)\,(B^{-1})^t = \begin{pmatrix} \alpha_u^2 + u_0^2 + \alpha_v^2.\cot(\theta)^2 & u_0.v_0 - \alpha_v^2.\cot(\theta)^2 & u_0 \\ u_0.v_0 - \alpha_v^2.\cot(\theta)^2 & \alpha_v^2.\frac{1}{\sin(\theta)^2} + v_0^{\,2} & v_0 \\ u_0 & v_0 & 1 \end{pmatrix}$$



References

- OpenCV library, url http://opencv.org/
- **[Dhome *et al.*, 2003]** Perception visuelle par imagerie vidéo. Dhome *et al.*, Hermès&Lavoisier, 2003.
- **[Skordas, 1995]** Une revue des derniers progrès en autocalibration de caméras CCD. GRETSI, 1995. http://hdl.handle.net/2042/1893
- **[Hartley, 2000]** Multiple View Geometry in Computer Vision, Hartley and Zisserman

Summary (17h C/TD and 9h TP)

1. Camera self-calibration
2. Structure from motion (SfM)
3. 3D modeling
4. 3D localisation and recognition
5. Applications by examples

[Principle.mp4]

SfM – State-of-the-art

- Goal: 3D reconstruction (3D points X) and/or camera motion estimation (nbview-1 extrinsic parameters)

$$R_1 = I, T_1 = 0, R_2' = R_2.R_1^{-1}, T_2' = T_2 - R_2.R_1^{-1}.T_1$$

- Approaches' categorization [Hartley, 2000]
  - Global = several variants

$$\varepsilon(X,R,T) = \sum_{i=1}^{m}\sum_{j=1}^{n} w_{ij}.\left\| P(X_i,R_j,T_j) - \begin{pmatrix} u_{ij} \\ v_{ij} \end{pmatrix} \right\|^2$$

=1 if point i is visible in view j

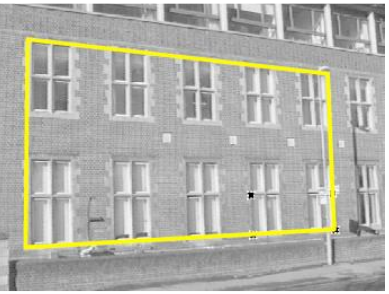
Predicted image location

Observed image location

  - Incremental (TP n°3): growing reconstruction process, initial two-view reconstruction then add new views/3D points

SfM – Homography model

- A 3 by 3 matrix H between image points



$$s.\begin{pmatrix} u_2 \\ v_2 \\ 1 \end{pmatrix} = \begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{pmatrix} . \begin{pmatrix} u_1 \\ v_1 \\ 1 \end{pmatrix}$$

2Nx8 2Nx1

$$A.H = b \rightarrow H = (A^T A)^{-1}(A^T B)$$

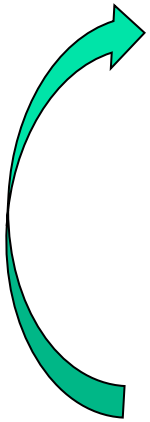
8x1

- Two images related by H if and only if
  - Both images are viewing the same plane in the scene
  - Both images are taken by cameras from a different angle (pure rotation, R)

$$x = KX, x' = KRX = KRK^{-1}x$$

Incremental SfM – Framework (OpenMVG), TP n°2

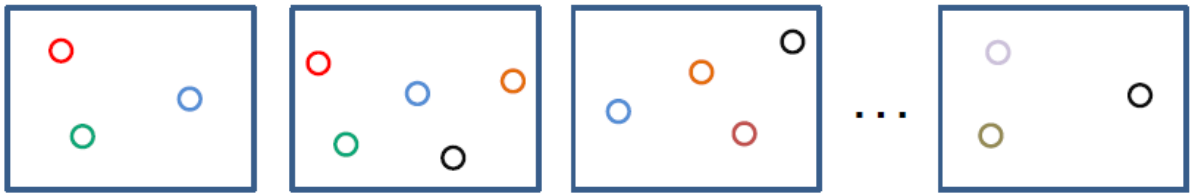
- Key point matchings for image pairs
- Selection of an image pair: initialization
- Reconstruction growing
  - Solve for pose of additional cameras which observe alreading reconstructed 3D points
  - Solve for new 3D points which are viewed in at least two cameras
  - Bundle adjustment to minimize reprojection error



Source: slides adapted from Derek Hoiem, Univ. of Illinois (Computer Vision, lecture #15)

Incremental SfM – Key point matchings

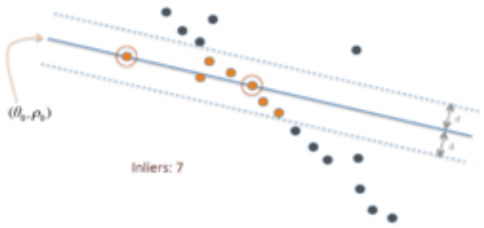
- 2D key point extraction in each view: SURF, BRIEF, ORB, etc.
- Match key points between image pairs given their associated descriptors
- Estimation of F matrix and find inlier key point matchings



Points of same color have been matched to each other

RANSAC principle

- RANSAC = RANdom SAmple Consensus
  - Outliers *vs.* inliers concept - Example for a line estimation



Algorithm 1 RANSAC

1: Select randomly the minimum number of points required to determine the model parameters.
2: Solve for the parameters of the model.
3: Determine how many points from the set of all points fit with a predefined tolerance ϵ.
4: If the fraction of the number of inliers over the total number points in the set exceeds a predefined threshold τ, re-estimate the model parameters using all the identified inliers and terminate.
5: Otherwise, repeat steps 1 through 4 (maximum of N times).

RANSAC – Application to SfM

- Required for H and F model

Model	F	F [Long, 81]	H
#Sample	7	8	4
#Model	3	1	1

- Sensitive to the error tolerance for model fitting (ε)

Incremental SfM – Initialization

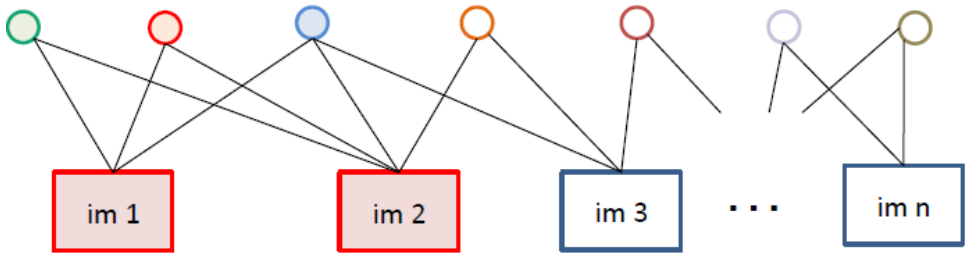
- Choose two images that provide a stable estimate of relative pose *e.g.*

$$\frac{\#\ inliers\ for\ H}{\#\ inliers\ for\ F} < 0.7\ and\ many\ inliers\ for\ F$$

- Estimate essential matrix E and extract pose:

$$R_2', T_2'\ with\ R_1 = I, T_1 = 0 \qquad \text{E}= K^T.F.K$$

- Triangulate associated 3D points X
- Perform bundle adjustment to refine pose and 3D points

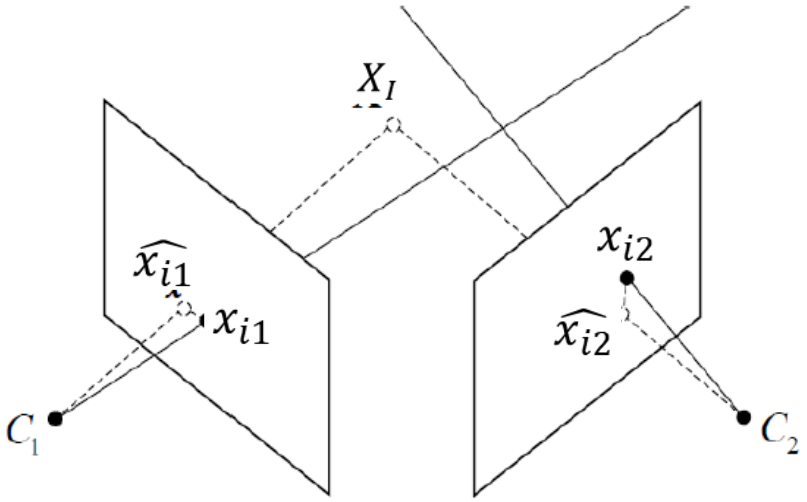


Incremental SfM – 3D point triangulation

- Non linear solution
- Minimize projected error ε(.) while satisfying $\widehat{x_{ij}}^{t}.F_{jj'}.\widehat{x_{ij'}} = 0$

$$\varepsilon(X_i) = dist\left(\widehat{x_{ij}}, x_{ij}\right) + dist\left(\widehat{x_{ij'}}, x_{ij'}\right) \forall\, j \neq j'$$

$$\widehat{x_{ij}} = M_{int}.M_{ext}\left(R_j, T_j\right).X_i = P_j.X_i$$



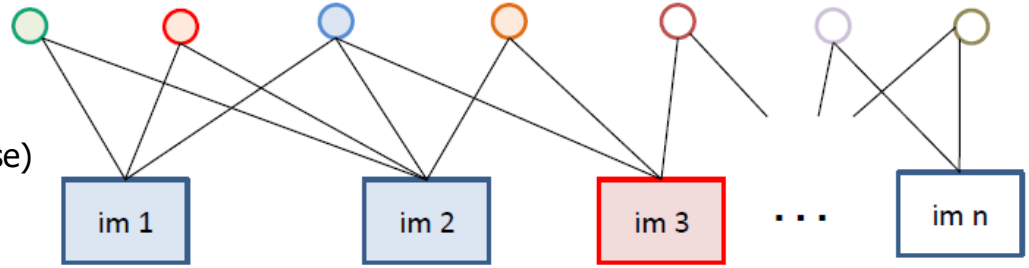
Incremental SfM – Bundle adjustment

- Non linear method for refining structure and motion/pose resp. $X = \{X_i\}_{i=1,\dots,m}$ and $\{R_j, T_j\}_{j=1,\dots,n}$
- Reprojection error minimization
- Require precise initial conditions for pose and 3D points

$$\varepsilon(X, R, T) = \sum_{i=1}^{m} \sum_{j=1}^{n} w_{ij} . \left\| P(X_i, R_j, T_j) - \begin{pmatrix} u_{ij} \\ v_{ij} \end{pmatrix} \right\|^2$$

Incremental SfM – Growing reconstruction

- Resection: solve pose for image(s) which have the most triangulated points
- Triangulate: solve for any new points that have at least two cameras
- Remove 3D points that are outliers
- Bundle adjustment → Stop OpenMVG
- Possibly: dense 3D reconstruction → OpenMVS



Filled points = point triangulation
Filled rectangles = « resectioned » images (solved pose)

SfM – Exercise/synthesis

- Where does SfM fail ?
- Differences between SfM and self-calibration ?
- Differences between SfM and SLAM ?

TP n°1 : Incremental SfM

- Python + OpenCV
- Two view geometry
  - KP matchings (SIFT)
  - Fundamental matrix F estimation – 8 unknows
  - Essential matrix then baseline pose estimation
  - Triangulation
- Integration of new views
  - PnP + RANSAC

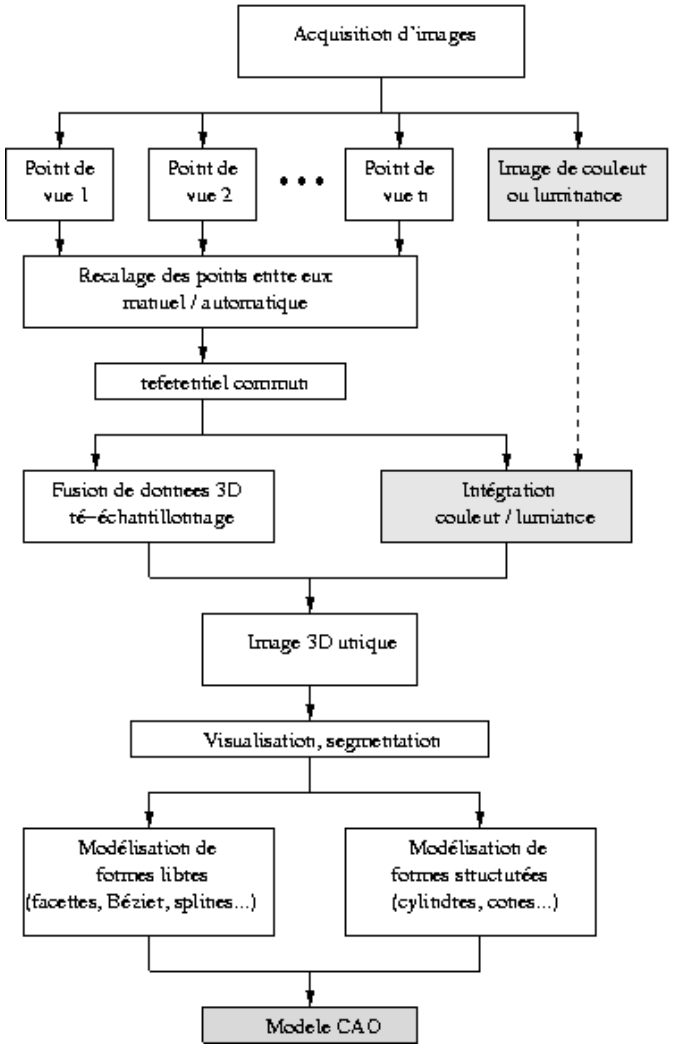
References

- Opensource SfM library, url https://openmvg.readthedocs.io/en/latest/software/SfM/SfM/
- Open3D library, url https://open3D.org
- **[Moulon *et al.*, 2012]** Adaptative structure from motion with a contrario model estimation. Moulon, Monasse, and Marlet. Asian Conf. on Computer Vision, 2012.
- **[Dhome *et al.*, 2003]** Perception visuelle par imagerie vidéo. Dhome *et al.*, Hermès&Lavoisier, 2003.
- **[Hartley, 2000]** Multiple View Geometry in Computer Vision, Hartley and Zisserman.
- **[Long, 81]** A computer algorithm for reconstructing a scene from two projections. Longuet-Higgins, Nature, 1981.

Summary (17h C/TD and 9h TP)

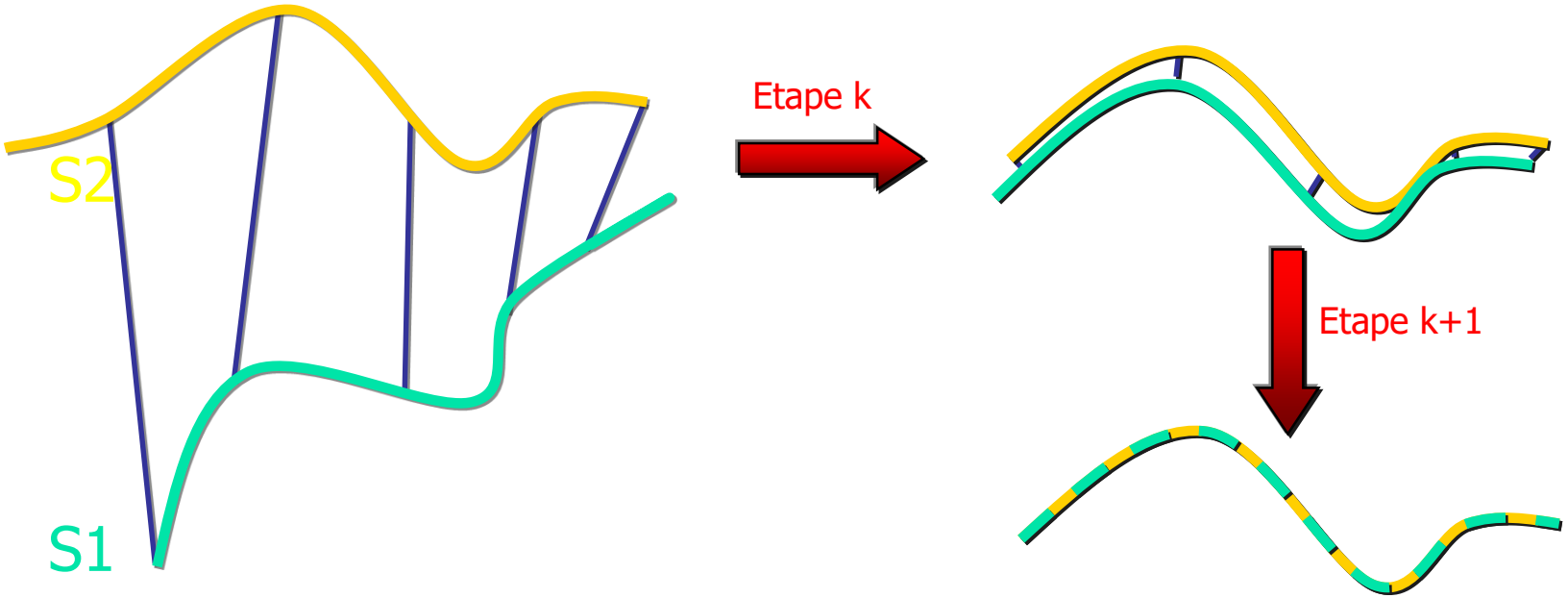
1. Camera self-calibration
2. Structure from motion
3. 3D modeling
4. 3D localisation and recognition
5. Applications by examples
6. Exercice correction

Generic algorithm



Automatic fitting/aligning

- Estimation of (R,T) between two sets of overlapped points S1 and S2



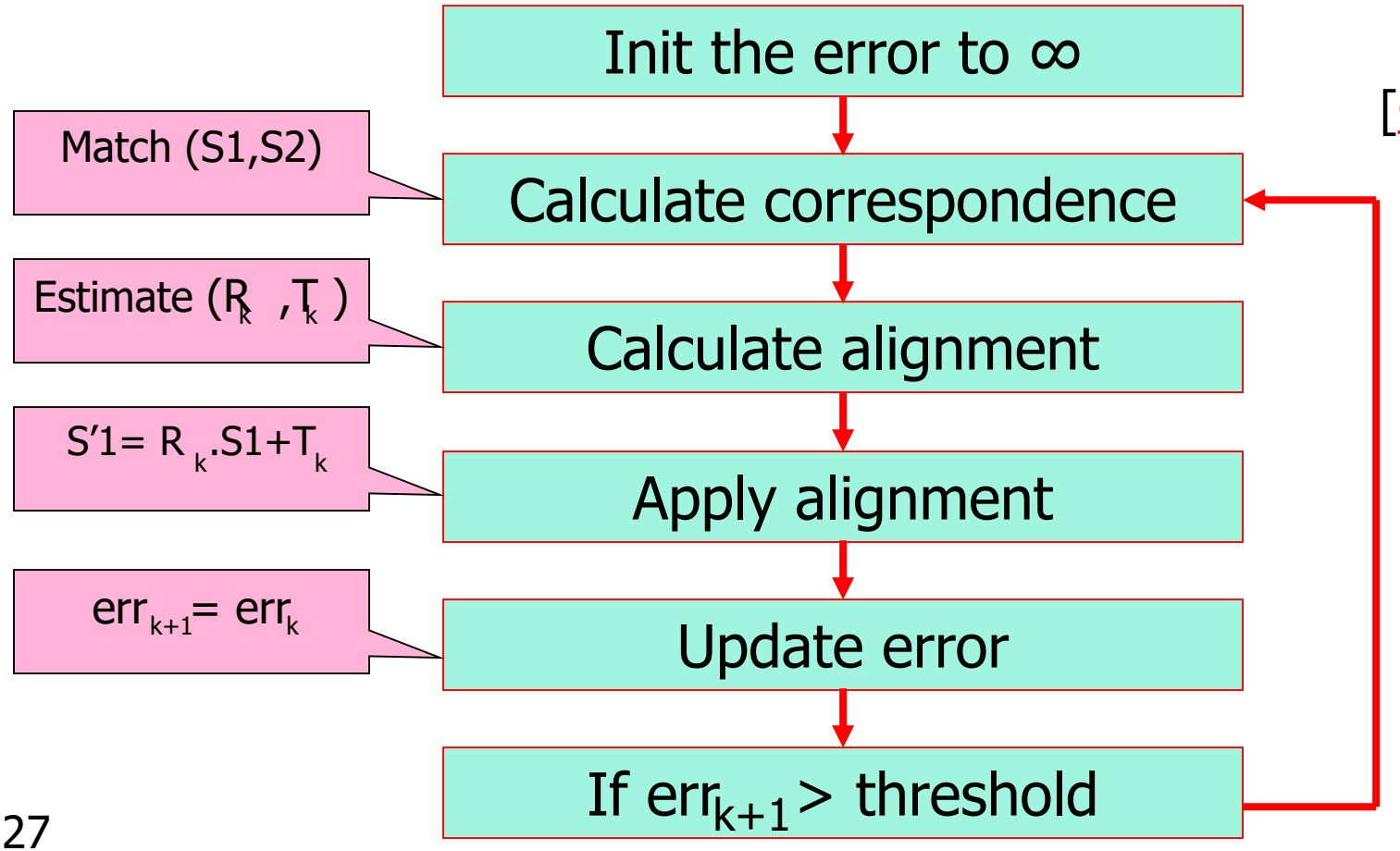
- Transformation of the whole points S1 and evaluation of the fitting quality :

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

[Reconstruction-Kinect.avi]

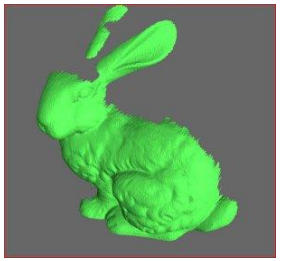
Automatic aligning (TP n°2)

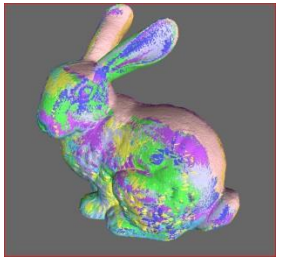
- Generic algorithm « Iterative Closest Point » (ICP)



[GO-ICP.avi]

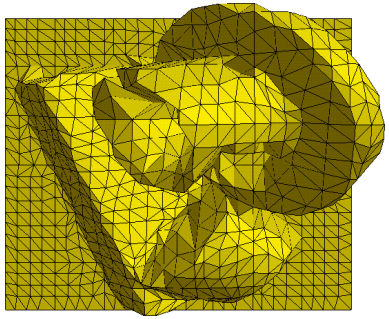
[Noomeo.avi]

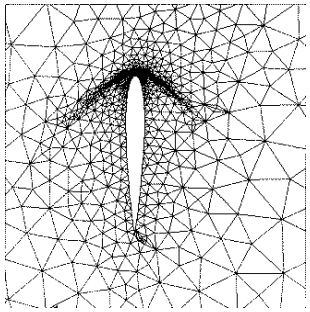




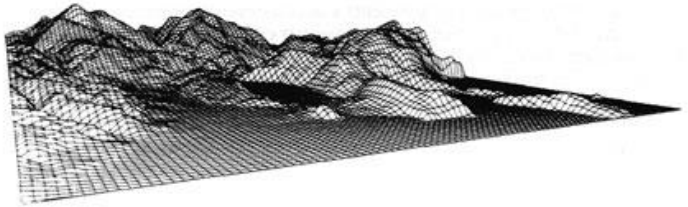
Freeform modeling

- Absence of mathematical models…
- Facet-based models
  - Mesh by faceting – rectangular, triangular,... – regular or not, hierarchical or not – suitable to any type of scene





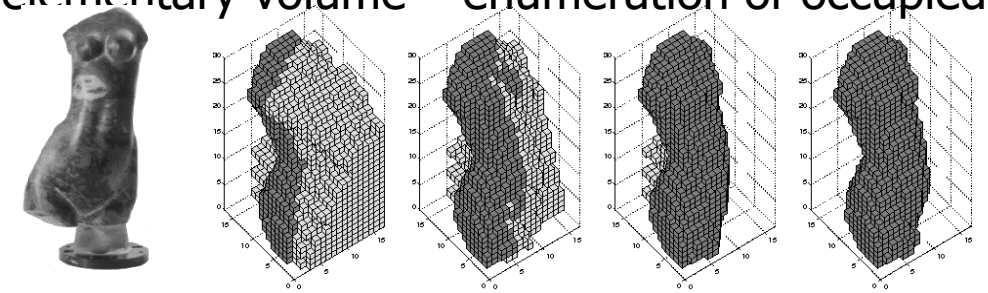
  - Example: digital terrain model (MNT) – elevation z on a measuring grid (x,y) – model called $2D^{1/2}$



Freeform modeling

- Representation based on volumetric elements
  - « Voxels » *i.e.* elementary volume – enumeration of occupied cells

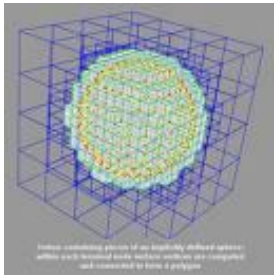
[Octomap.mp3]



Right side view of the statue – active triangulation – work space : 17x17x30 voxels - ridge: 5mm

  - « Octrees » *i.e.* cubes of varying sizes

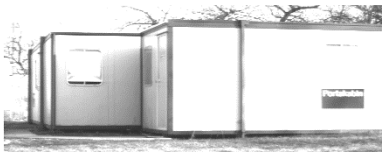
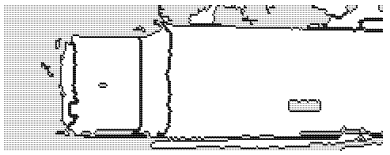
[TableTop.avi]



  - Deformable surfaces: splines, Bezier's curves, B-splines

Modeling of solid forms

- Simple mathematical models *i.e.* few parameters (<10)
- Example #1: the plane

$$\mathrm{Res} = \sum_{i=1}^{m} \left(\mathrm{n}^{t}.\mathrm{P}_{i} - d\right)^{2}, \mathrm{P}_{G} = \frac{1}{m}\sum_{i=1}^{m} \mathrm{P}_{i}, \mathrm{M} = \sum_{i=1}^{m} \left(\mathrm{P}_{i} - \mathrm{P}_{G}\right)\left(\mathrm{P}_{i} - \mathrm{P}_{G}\right)^{t}$$

$$\underset{\mathrm{n,d}}{\mathrm{Min}}\,\mathrm{Res} \Rightarrow \mathrm{Res} = \mathrm{n}^{t}.\mathrm{M}.\mathrm{n}, \mathrm{d} = \mathrm{n}^{t}.\mathrm{P}_{G}$$

- Example #2: the biquadratic surface

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

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

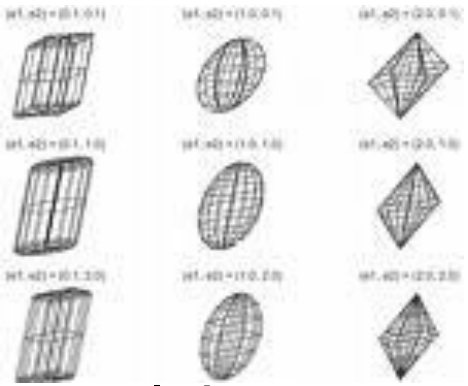
$$\underset{a}{Min}\,\mathrm{Res} \Rightarrow a = M^{-1}.D$$

$$\textit{with}\ \ \mathrm{D} = \sum_{i=1}^{m} \mathrm{z}_{i}.Q(x_i, y_i), M = \sum_{i=1}^{m} Q(x_i, y_i).Q(x_i, y_i)^{t}$$

Modeling of structured forms

- Geometrical primitives
  - Spheres, cylinders, cones, torres, …
  - Superquadrics

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



  - Structuration based on « Constructive Solid Geometry » tree - nodes = geometrical elements – arcs = set operations

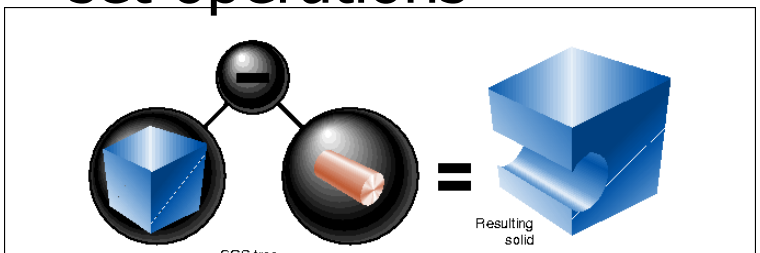
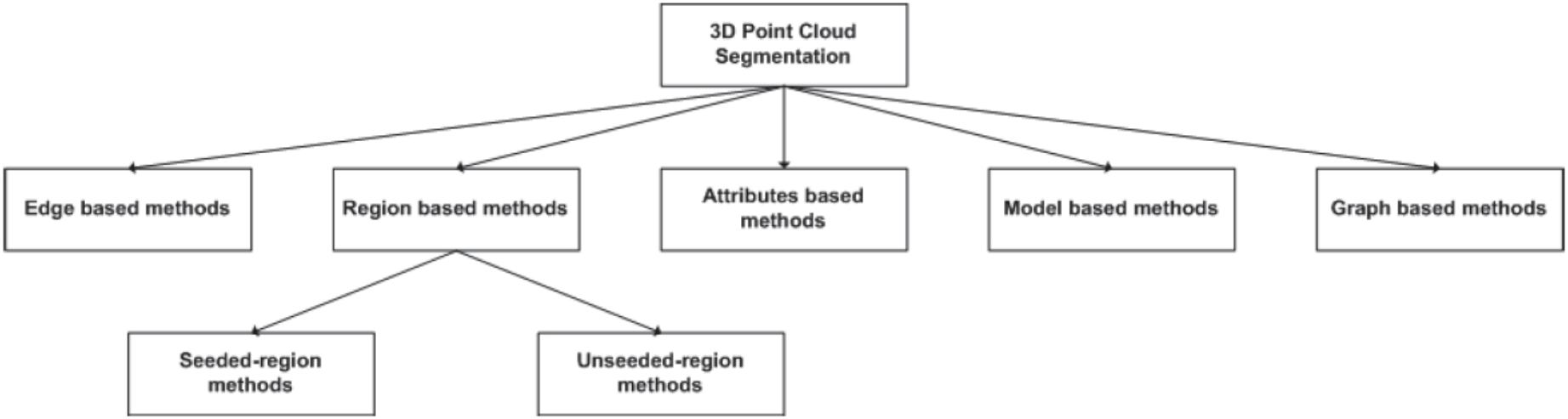


Figure 12. An Example Of Constructive Solid Geometry

3D segmentation

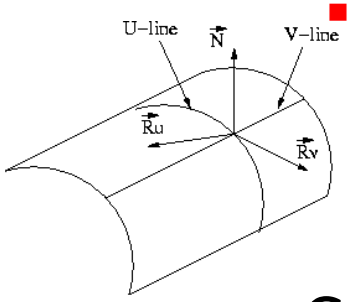
- Approaches' taxonomy [Nguyen *et al.,* 2013]



- What about the attributs ?

3D segmentation

- Normal Attribute
  - Suitable to plane surfaces
  - Estimated after faceting otherwise by local tangents…
  - Calculation by masks of Sobel, Kirsch



$$\vec{N} = \overrightarrow{R_u} \wedge \overrightarrow{R_v}, \overrightarrow{R_u} = (x_u, y_u, z_u) \; et \; \overrightarrow{R_v} = (x_v, y_v, z_v),$$
$$x_u = \frac{\partial x}{\partial u}, y_u = \frac{\partial y}{\partial u}, z_u = \frac{\partial z}{\partial u}, etc.$$

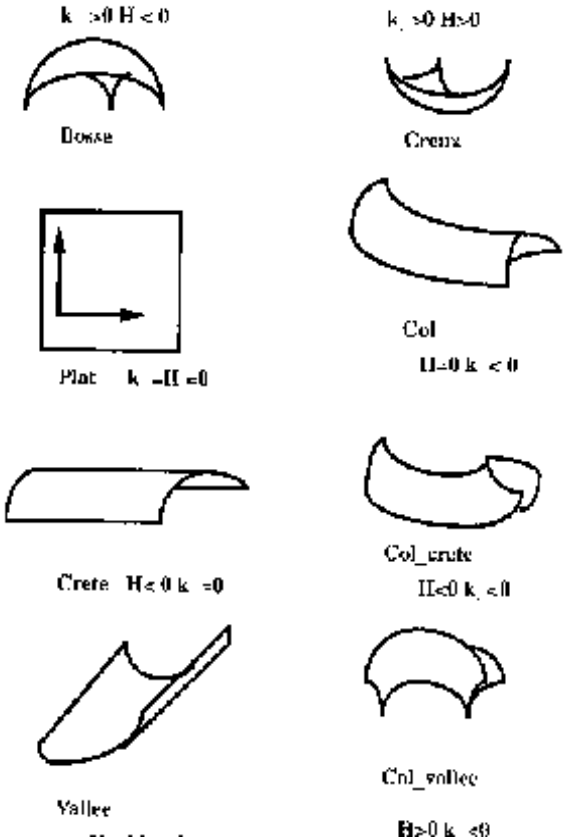
- Curvature (curve) Attribute
  - Suitable to surfaces with high degrees
  - Calculation of the principal curvature radius R1 and R2

$$(LN - M^2)R^2 + (E - 2FM + GL)R + (EG - F^2) = 0, \begin{bmatrix} E & F \\ F & G \end{bmatrix} = \begin{bmatrix} \overrightarrow{R_u}.\overrightarrow{R_u} & \overrightarrow{R_u}.\overrightarrow{R_v} \\ \overrightarrow{R_u}.\overrightarrow{R_v} & \overrightarrow{R_v}.\overrightarrow{R_v} \end{bmatrix}, \begin{bmatrix} L & M \\ M & N \end{bmatrix} = \begin{bmatrix} \overrightarrow{R_{uu}}.\vec{N} & \overrightarrow{R_{uv}}.\vec{N} \\ \overrightarrow{R_{uv}}.\vec{N} & \overrightarrow{R_{vv}}.\vec{N} \end{bmatrix}$$

$$\overrightarrow{R_{uu}} = (x_{uu}, y_{uu}, z_{uu}), \overrightarrow{R_{vv}} = \cdots, x_{uu} = \frac{\partial^2 x}{\partial^2 u}, y_{uu} = \frac{\partial^2 y}{\partial^2 u}, etc.$$

3D segmentation

- Attribute curvature
  - Gauss curvature K and mean curvature H: $K = \frac{1}{R_1.R_2}, H = \frac{\frac{1}{R_1} + \frac{1}{R_2}}{2}$



3D segmentation

- Edge-based approach
  - Applied on depth image
    - Calculation of the gradient and local maxima search
    - Chaining ⊥ gradient direction
  - Applied on 3D point cloud
    - Définition of a 3D neighborhood
    - Calculation of vector directions of the central point to its neighbors
    - A edge = direction discontinuity
- Attribute-based approach
  - Based on a clustering phase
  - Example : Hough transformation applied on curvatures (see slide 33)

3D segmentation

- Region-based approaches
  - Goal: segment in region (dissimilarity attribute)
  - Detection of depth jumps/steps (order 0), discontinuity of normals (order 1), discontinuity of curvatures (order 2)
  - Three techniques: bottom up, top down or hybrid

Seed = point

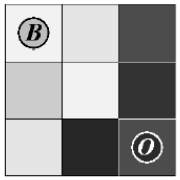
Begin
Init.
Do
  Nb_points=Nb_points+1
  Update of the model parameters
While (discontinuity=FALSE) or (residue<threshold)
End

Seed = region

Begin
  Selection of elementary regions R (Type, Nb_points)
  Parameter estimation
  Sorting regions R
  Building lists of connected regions/points L_R
  Do
    Consider residue when R fusing with L_R
  While fusion=TRUE
End

3D segmentation

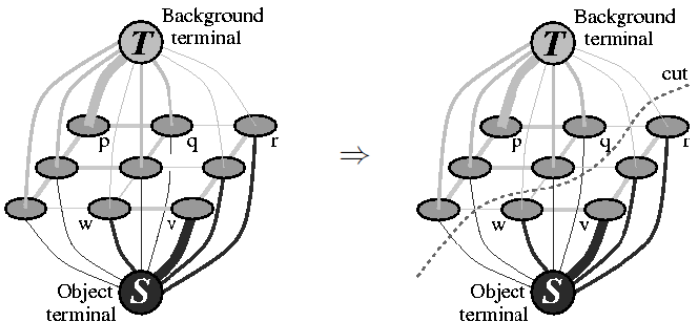
- Graph-based approaches
  - Image seen like a graph: vertex (V), arc (A)
  - Segmentation = vertices labelling when minimizing a criteria E → graph cut



(a) Image with seeds.

(d) Segmentation results.

$$E(\lambda) = \sum_{p \in V} U_p(\lambda_p) + \sum_{\{p,q\} \in A} U_{p,q}(\lambda_p, \lambda_q)$$



(b) Graph. (c) Cut.

← Illustration on 2D segmentation

3D segmentation – example on image restoration

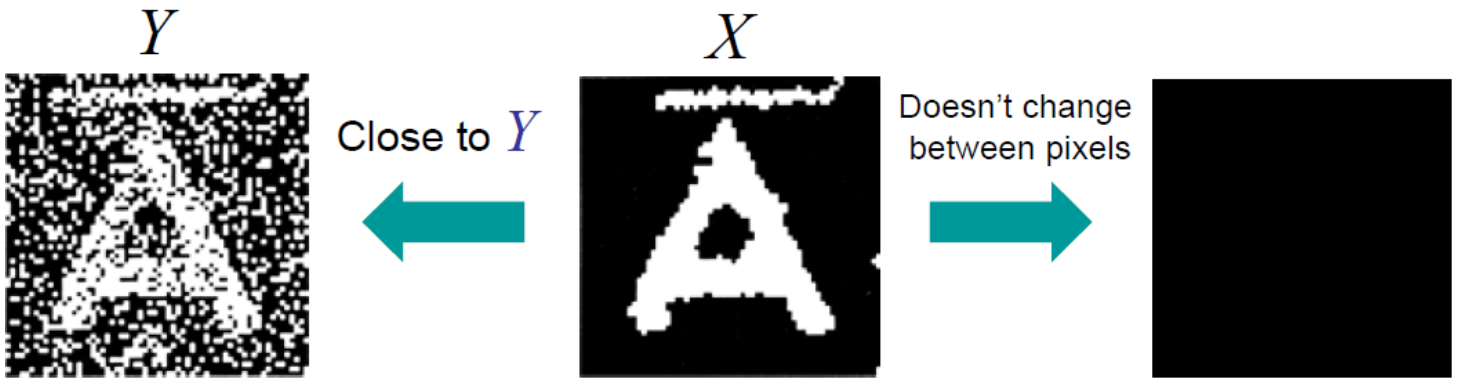
- (simple) illustration on binary image restoration

Assigns X_v ($= 0$ or 1) to each pixel v

$$\min_X E(X) \quad \text{avec} \quad E(X) = \sum_{v \in V} \lambda \mid Y_v - X_v \mid + \sum_{(u,v) \in E} \kappa \mid X_u - X_v \mid$$

All pixels

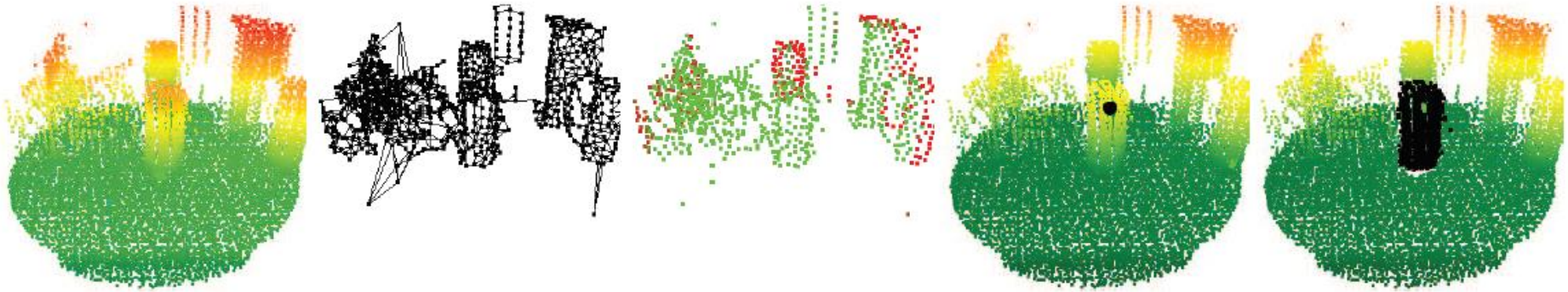
Neighboring pixels



3D segmentation

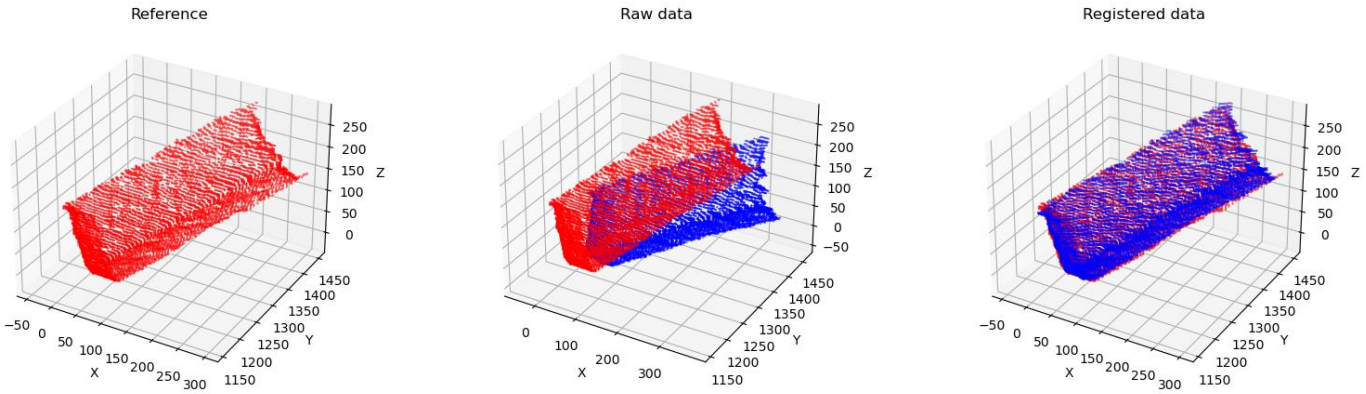
- Graph-based approach, extension to 3D [Golovinsky *et al.,* 2009]: foreground segmentation (*a priori* known seeds) vs. background
  - Characterize Up: distance to the seeds
  - Up,q: k-PPV (k=4), arcs weights ↘ distance
  - « Min cut » algorithm

  → minimize the cut cost within the graph k-PPV



(a) Input (b) Graph (c) Background Penalty (d) Foreground Constraint (e) Result

TP n°2 : 3D segmentation and conformity control



- Python + CloudCompare software
- 3D model registration - reference
- 3D (test) object dimension checking
  - Fitting based on ICP
  - Checking based on ICP errors

References

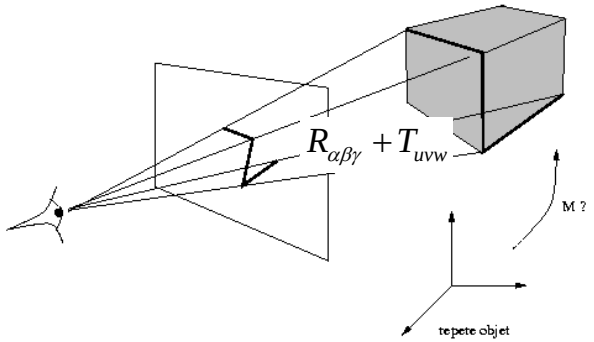
- Point Cloud Library, url http://pointclouds.org/
- OpenCV library, url http://opencv.org/
- **[Golovinsky *et al.,* 2009]** Min-cut based segmentation of point clouds. Golovinsky & Funkhouser. IEEE Int. Conf. on Computer Vision, 2009.
- **[Goulette, 1999]** Modélisation 3D automatique, outils de géométrie 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.

Summary (17h C/TD and 9h TP)

1. Camera self-calibration
2. Structure from motion
3. 3D modeling
4. 3D localisation and recognition
    1. Localisation : PnP, PnL, Pose-IT
    2. Recognition
5. Applications by examples
6. Exercice correction

2D/3D localisation (TP n° 3)

- Goal: estimate the extrinsic parameters knowing
  - The intrinsic parameters
  - A CAD 3D model of the scene
  - Some matchings between model primitives and visual primitives

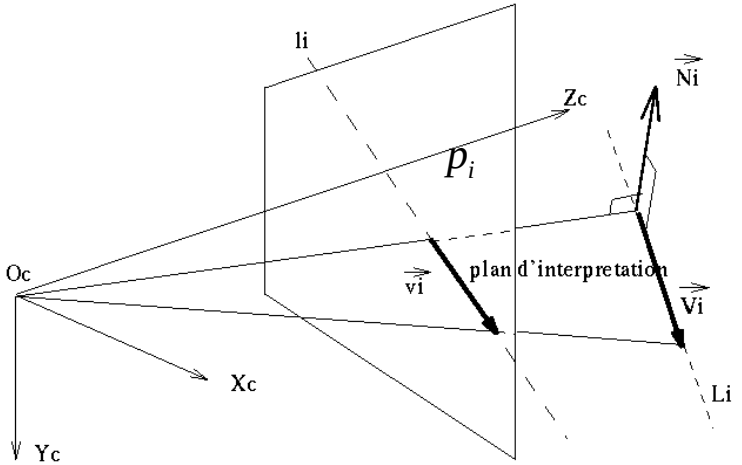


- Approaches' taxonomy according to:
  - The model projection : orthographic, perspective
  - The matched primitives type: points, lines, circles
  - The method resolution: analytical *e.g.* P3P, P3L, Pose-IT, numerical *e.g.* P-n-L

2D/3D localisation: « PnL »

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

$$F(X, \overrightarrow{N_i}, P_i) = \overrightarrow{N_i}.[R_{\alpha\beta\gamma}.P_i + T_{xyz}], \overrightarrow{N_i} = \frac{\overrightarrow{O_c p_i} \times \overrightarrow{v_i}}{\left\| \overrightarrow{O_c p_i} \times \overrightarrow{v_i} \right\|}$$

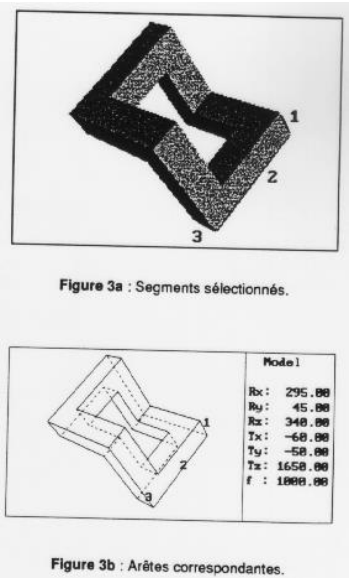


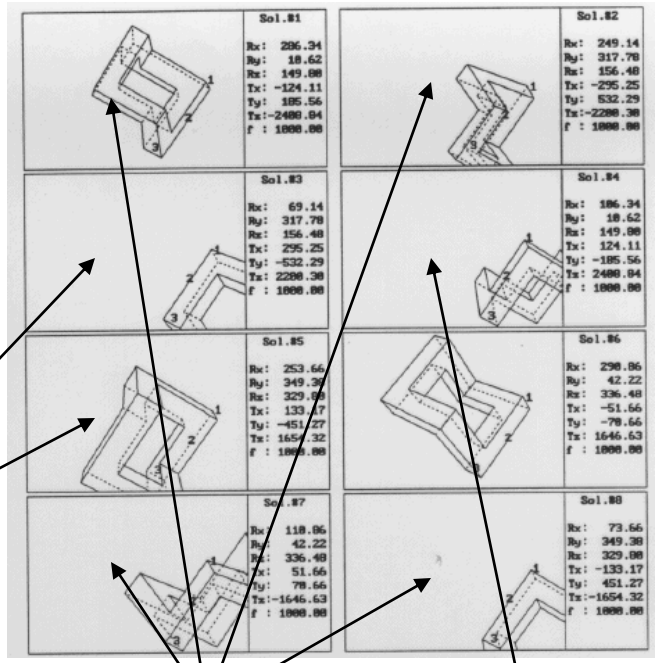
  - Resolution:

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

8 theoretical solutions ⇒ 3 possibles solutions !

2D/3D localisation: « PnL »



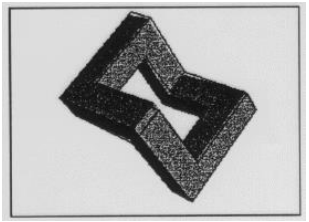


non visibles edges

Tz < 0

overlapping

solution after filtering



2D/3D localisation: « PnL »

- Perspective-n-Lignes: numerical resolution
  - Criteria to minimize: $\varepsilon = \sum_{i=1}^{n} F\left(X, \overrightarrow{N_i}, P_i\right)^2 = V^T.V$
  - Resolution: $\min_X (V^T V), V = A.\Delta X - L, \Delta X = (A^t A)^{-1}.A^t.L$
  - Calculation of the derivatives

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

$$A = \begin{bmatrix} \frac{\partial F_1}{\partial \alpha} & \cdots & \frac{\partial F_1}{\partial T_z} \\ \vdots & \ddots & \vdots \\ \frac{\partial F_n}{\partial \alpha} & \cdots & \frac{\partial F_n}{\partial T_z} \end{bmatrix}, L = \begin{bmatrix} -F_1(X_0) \\ \vdots \\ -F_n(X_0) \end{bmatrix}$$

  - Termination criteria ? Initial condition ?

[Diotasoft-PnL] [Diotasoft-PnL+ICP]

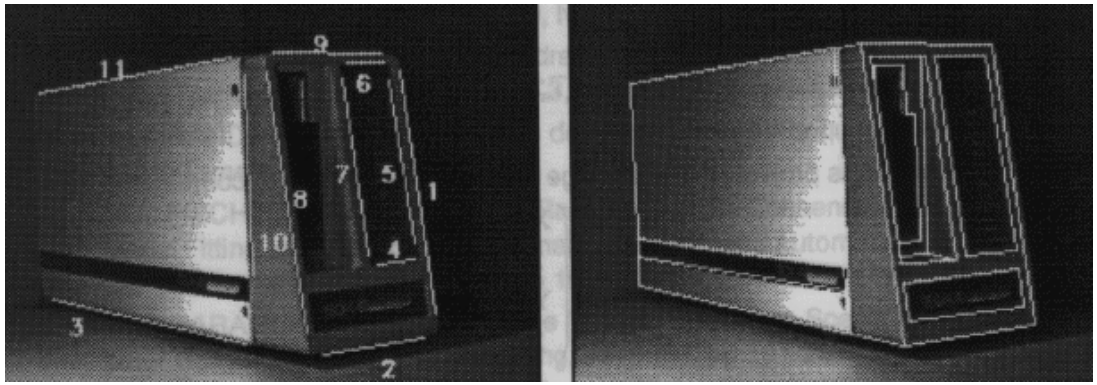
2D/3D localisation: « PnL »

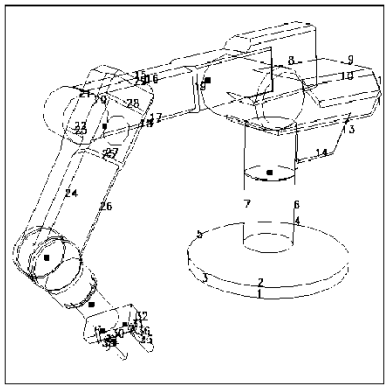
- Precision *vs.* number of cameras

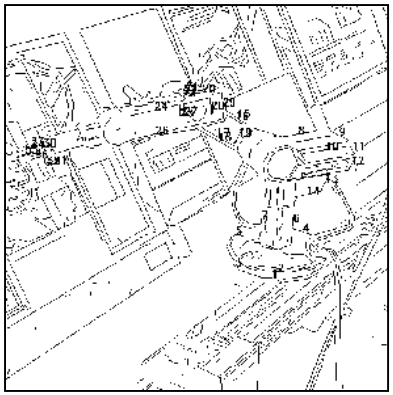
	α	β	γ	Tx	Ty	Tz
1 camera	0.8°	1.1°	0.1°	4.7mm	3.2mm	29.9mm
2 cameras	0.9°	1.0°	0.3°	5.1mm	2.8mm	9.3mm
3 cameras	0.7°	0.9°	0.3°	4.7mm	2.6mm	7.0mm

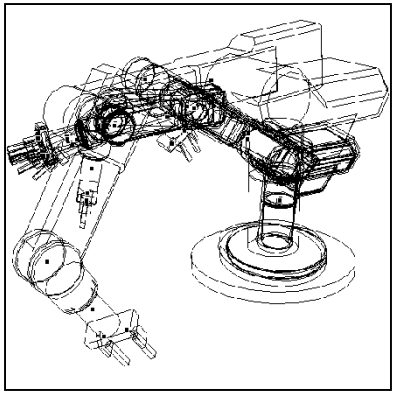
2D/3D localisation: « PnL »

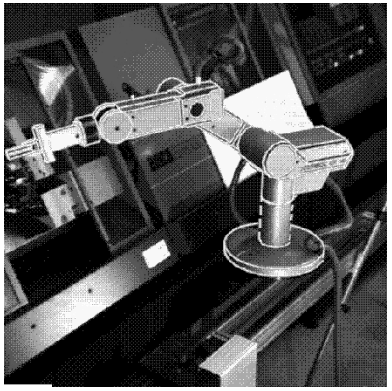
- Examples for rigid/articulated objects





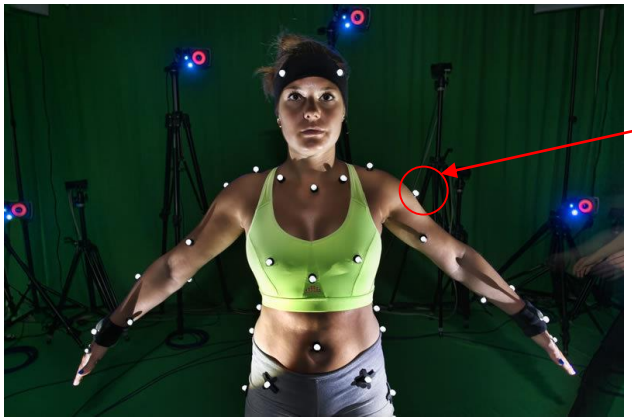


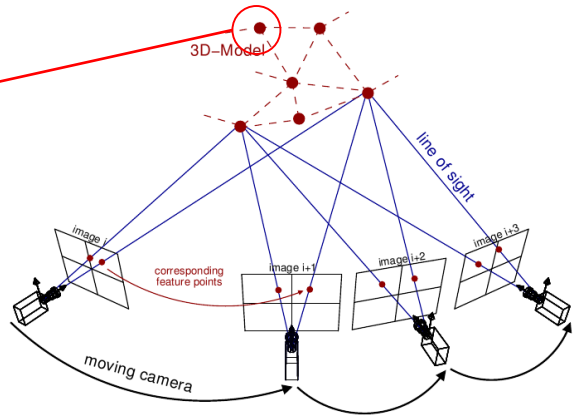




Human motion capture

- PnP localisation, m calibrated cameras
  - 3D models (body segments)
  - Passive/active marker matchings
  - 3D localisation of the markers (a few mm)





[MoCap-OptiTrack]

2D/3D localisation: « PnL » - Exercise

- Exercise: algorithm variant
  - What would be the precision gain when extending the localization to several cameras ?
  - Adapt then the criteria to minimize
  - Propose a suited criteria in order to extend to point matchings for monocular vision
  - Adapt the calibration methodology dedicated to monocular vision and point matchings

2D/3D localisation: « Pose-IT »

- Initialization method [Dhome *et al.*, 2003]
  - Same hypothesis as PnP
  - Progressif transition from the scale orthographic projection model to the perspective one
  - **Goal :** Estimate the extrinsic parameters. With:

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

2D/3D localisation: « Pose-IT »

- Algorithm:

❶ t=0 and $\varepsilon_i(t) = 0 \; for \; i \in [1,n], with: I^* = \frac{f}{T_z}(I,u), J^* = \frac{f}{T_z}(J,v)$

❷ Computation of the vectors I* and J* after resolution of the linear syst.

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

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

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

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

❹ t = t+1; computation of the new $\varepsilon_i(t) = \frac{K.P_m^i}{T_z}$

❺ $|(\epsilon_i(t) - \epsilon_i(t-1))/n| < \text{threshold then stop, otherwise}$ go back to ❷

[Pose-IT.avi]

2D/3D localisation - Exercise

- Compare PnL/PnP and « Pose-IT » in terms of advantages and drawbacks

3D/3D localisation

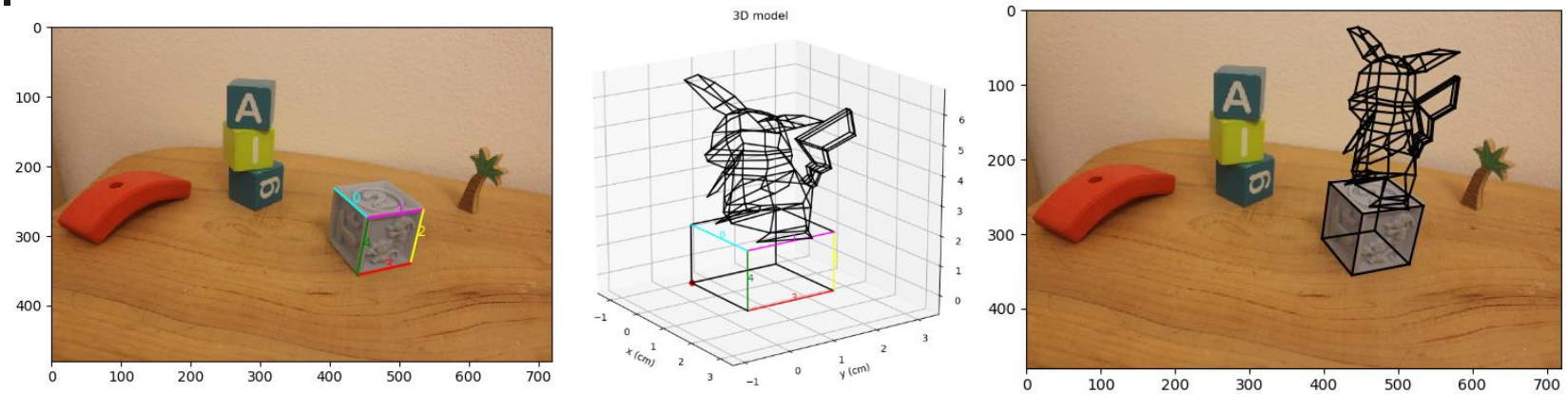
- Matchings of N points related to scene *vs.* model, resp. $\{p_i^s, p_i^m\}_{i=1,\dots,N}$
- Non linear MSE:

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

[Rob-spatiale.mp4] [Head-Pose-Li.mp4] [Fanuc-3DL.mp4]

[Lourakis et al., 2013] *Model-based pose estimation for rigid objects. Lourakis & Zabulis. Int. Conf. on Computer Vision Systems, 2013*

TP n°3 : 2D/3D localisation, PnL



- Python + Blender software
- 3D object modelling
- 2D/3D lines matching
- PnL implementation
- Projection of the estimated pose
- Augmented reality application

Summary (17h C/TD and 9h TP)

1. Camera self-calibration
2. Structure from motion
3. 3D modeling
4. 3D localisation and recognition
    1. Localisation : PnP, PnL, Pose-IT
    2. Recognition
5. Applications by examples
6. Exercice correction

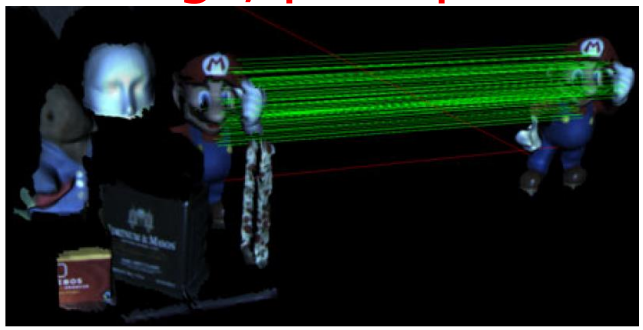
3D/3D recognition

- Goal: (1) automatic matchings of scene/model primitives, (2) localisation
- Matching complexity: given O the set of 3D model primitives (size n) and D the set of 3D scene primitives (size m) such as $m \geq n$. Number of possible matchings:

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

→ Combinatorial problem !!

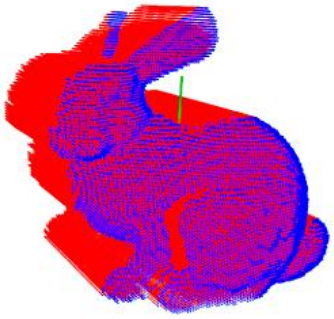
→ Key primitives extraction, computation of associated local and global descriptors, matchings, pose prediction / vérification



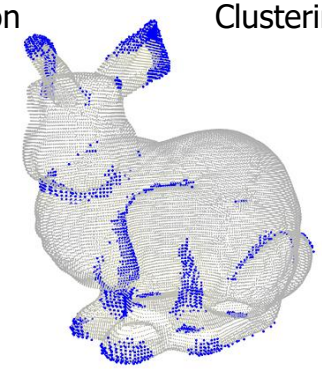
3D/3D recognition

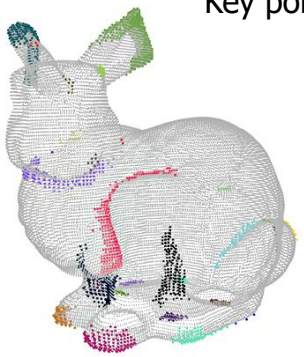
- 3D primitives = here points cloud
- Key points extraction *i.e.* robust to noise, view point variations, occlusions, discriminant
- Detection at fixed scale: LSP, ISS, KPQ,... [Salti *et al.*, 2011]
  - LSP = « Local Surface Patch »
    - Curvature local extremum SI in a certain neighborhood
  - ISS = « Intrinsic Shape Signature »
    - Computation of the dispersion matrix and eigenvalues comparaison

LSP extraction

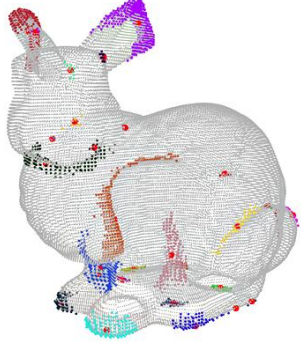


Clustering





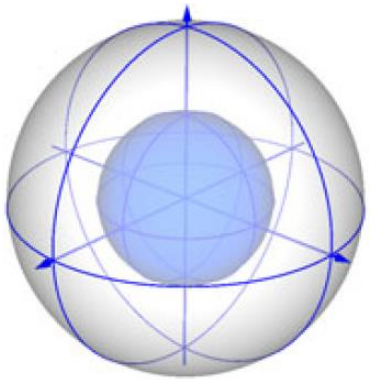
Key points to match afterwards



3D/3D recognition

- Local descriptors: SHOT [Tomari *et al.*, 2010], spin image, PFH, 3D shape context [Kortgen *et al.*, 2003]
  - SHOT = « Signature of Histograms Orientation »
    - Definition of a local invariant frame

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

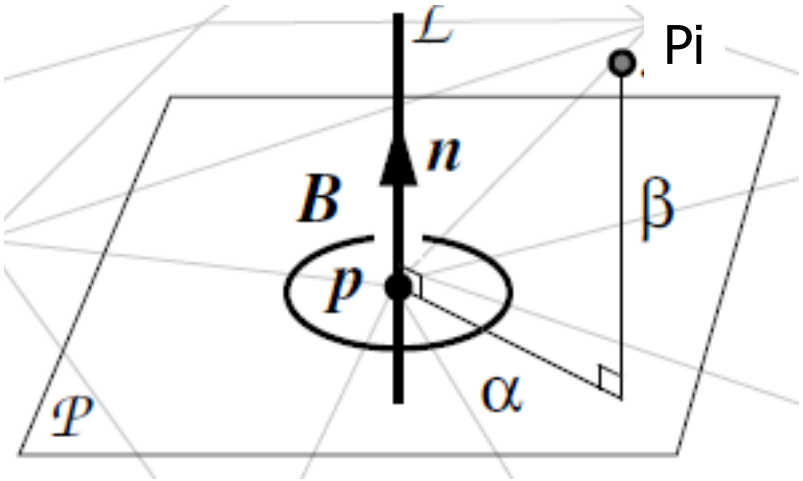


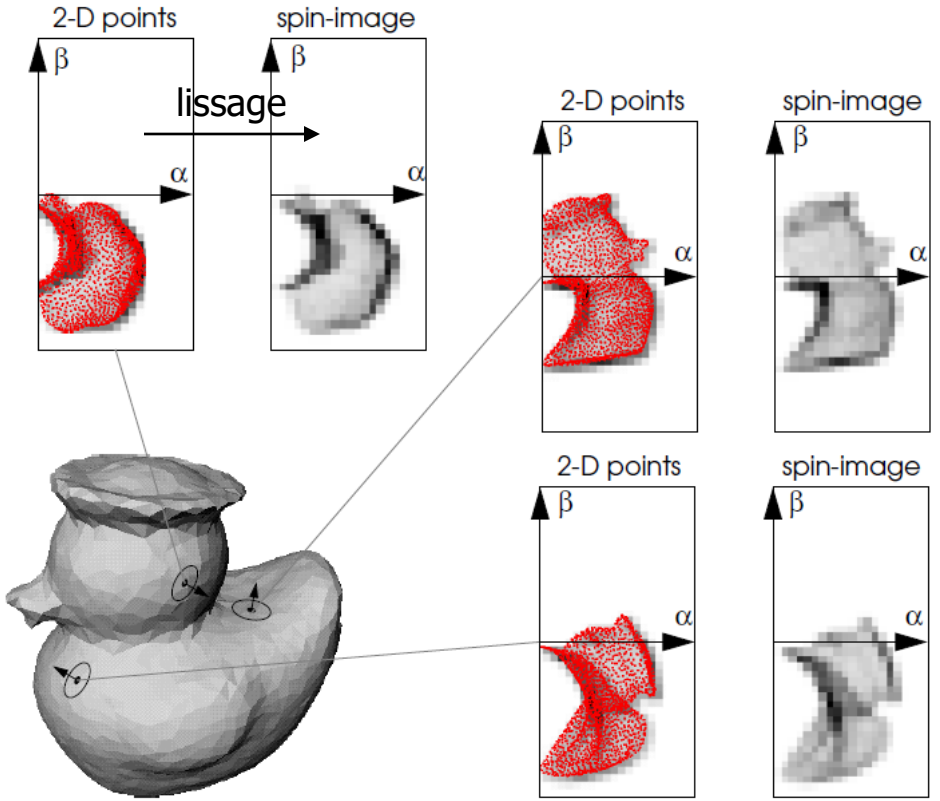
- Definition of a neighborhood (spherical grid, 32 partitions)
- Histogram (352 cells) of orientation difference θ between normals

$$\cos\theta_i = n_p . n_{p_i} \qquad \underline{h} = \left(h_{c_1}, h_{c_2}, \ldots, h_{c_n}\right), h_{c_i} = \sum_{u \in R} \delta_{c_i}(b_u), b_u \in \{c_1, \ldots, c_n\}, \sum_{i=1}^{n} h_{c_i} = 1$$

3D/3D recognition

- Local descriptors: SHOT [Tomari *et al.*, 2010], spin image, PFH, 3D shape context [Kortgen *et al.*, 2003]
  - Spin image =





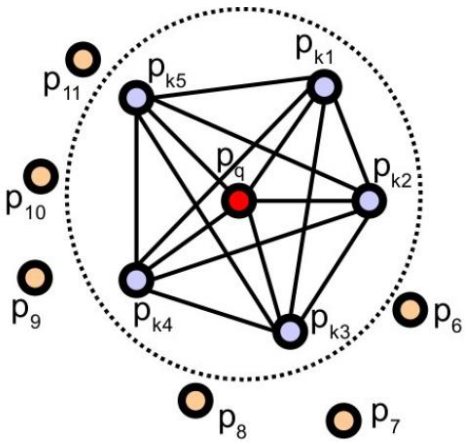
3D/3D recognition

- Local descriptors : SHOT, spin image, PFH [Rusu *et al.*, 2009], 3D shape context, etc.
  - PFH = « Point Feature Histogram »
    - Definition of a local frame (Darboux) invariant and sphere-type neighborhood

$$u = n_i, v = u \wedge \frac{p_i - p_j}{\|p_i - p_j\|_2}, \mathrm{w} = \mathrm{u} \wedge v$$

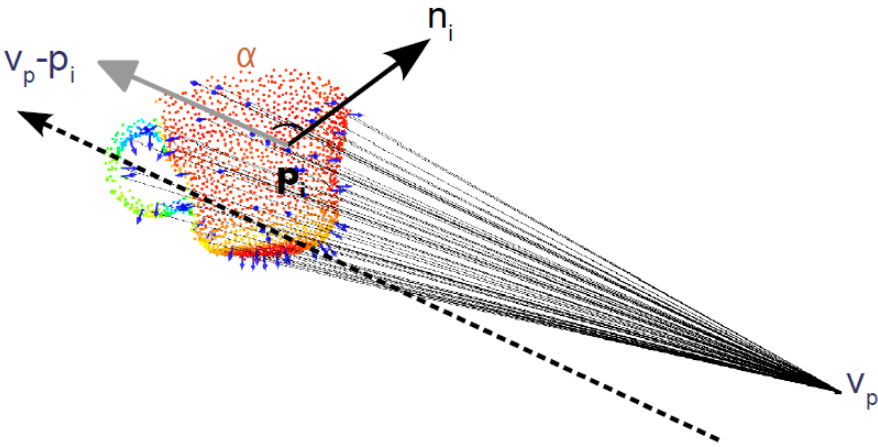
    - Histogram (33 cells) between relative orientation of the normals and distance between point pairs $p_i \; et \; p_j$

$$\alpha = v^T.n_j, \phi = u^T.\frac{p_i - p_j}{\|p_i - p_j\|_2}, \theta = \tan^{-1}\left(w^T.n_j, u^T.n_j\right)$$



3D/3D recognition

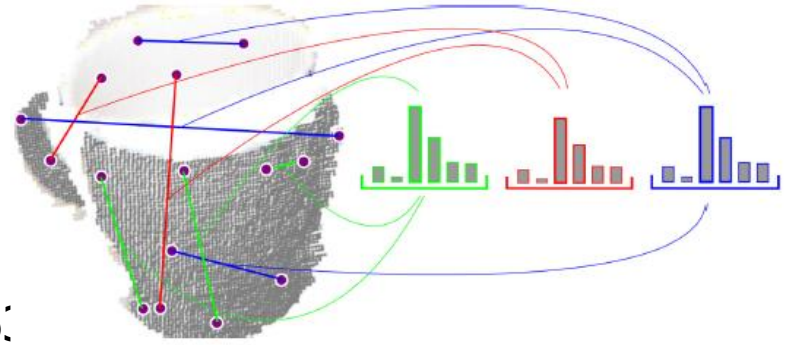
- Global descriptors : PFH, VPFH [Rusu *et al.*, 2010], CVFH, ESF
  - VPFH = « Viewpoint Feature Histogram »
    - Inspired by FPH (Darboux frame, histogram)
    - Histogram with 263 cells
      - 3 histos (45 cells) on orientations α, ϕ, Θ
      - 1 histo (128 cells) on angle between normals and view points
    - CVFH = extension to N regions of the model (N histograms VFH)

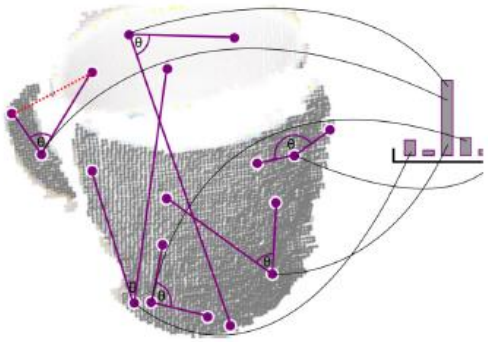


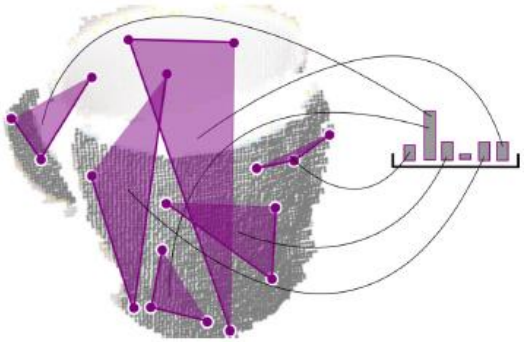
3D/3D recognition

- Global descriptors: PFH, VPFH, CVFH, ESF [Alexandre, 2012][Brkié *et al.*, 2014]
  - ESF = « Ensemble of Shape Function »
    - No pre-processing (no normal computation)
    - Voxelization
    - Calculation of histograms on angle, distance between points, area,... With distinction between in/our objet surface or both

  → 64 x 3x 3 = 576 cells in total

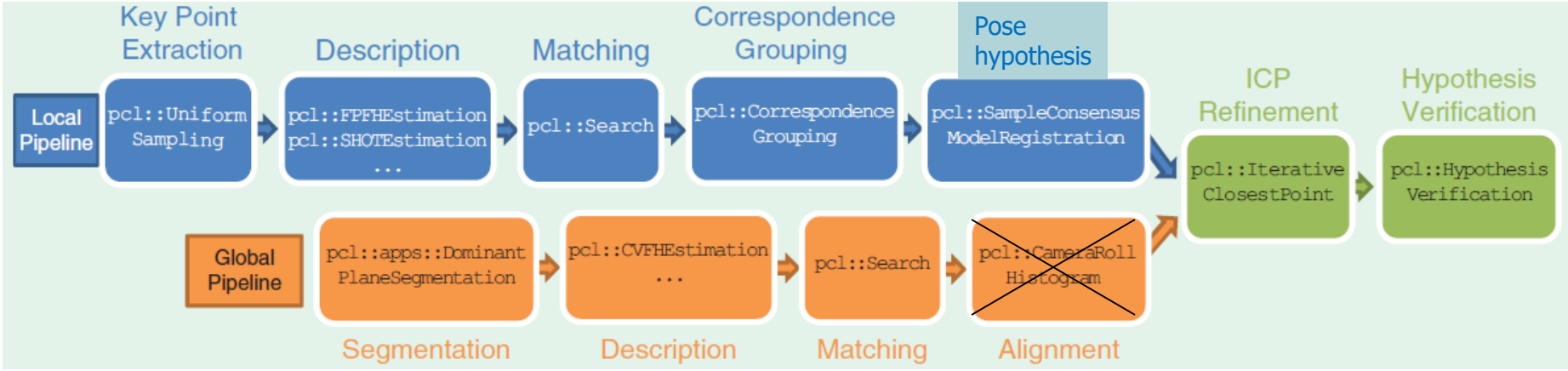






3D/3D recognition

- Generic methodology
  - Strategy without/with previous segmentation, resp. local/global descriptors
  - Hypothesis: 3D model 3D of the objects (CAD, meshes), 3D multi-views



3D/3D recognition

- Strategy with local descriptors
  - (1) Detection of robust key points and computation of discriminative descriptors

  → Preliminary definition of a neighborhood

  - (2) Matchings between model/scene primitives
    - Through similarity measures between descriptor vectors $\underline{h}^1, \underline{h}^2$: distance (norm L1, norm L2, norm L∞), dot product

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

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

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

3D/3D recognition

- Strategy with local descriptors
  - (3) Correspondances clustering and hypotheses filtering (Nb match > 3)

$$\left\|p_i^m - p_j^m\right\|_2 - \left\|p_i^s - p_j^s\right\|_2 < \varepsilon$$

  - 3D pose estimation based on RANSAC

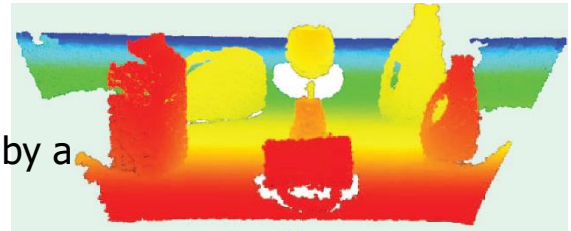
$$\text{Arg}\min_{R,T} \sum_{i=1}^{N} \|p_i^s - R.p_i^m - T\|_2^2$$

3D/3D recognition

- Strategy by global descriptors
  - Object segmentation
  - Descriptor computation for each object region
  - Matching with the model views

  → selection of the k best hypotheses

Point cloud issued by a RGB-D sensor



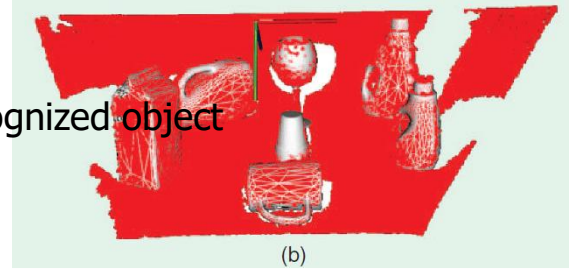
(a)

- Generic strategy
  - Pose refinement by ICP
  - Hypothesis verification

  → overlapping criteria between point clouds globally

  → sorting the k hypotheses

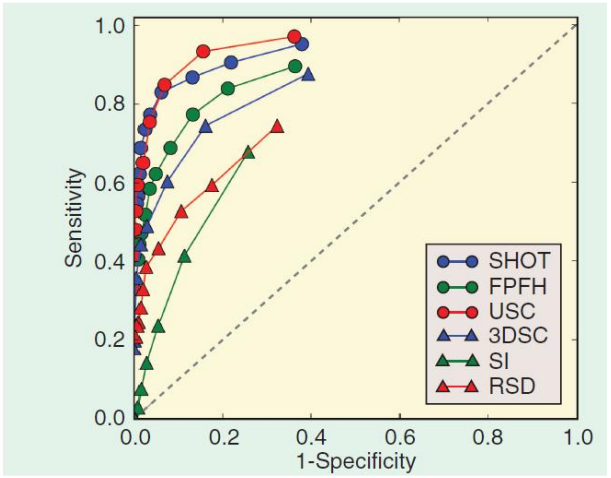
Mesh = recognized object

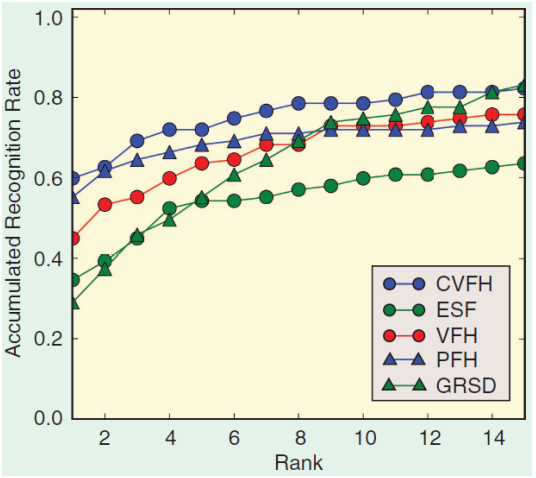


(b)

3D/3D recognition (Ex.)

- Generic strategy
  - Dataset = usual objects, 54 images of diverse scenes
  - Criteria based on statistics related to TP, FP, TN, FN

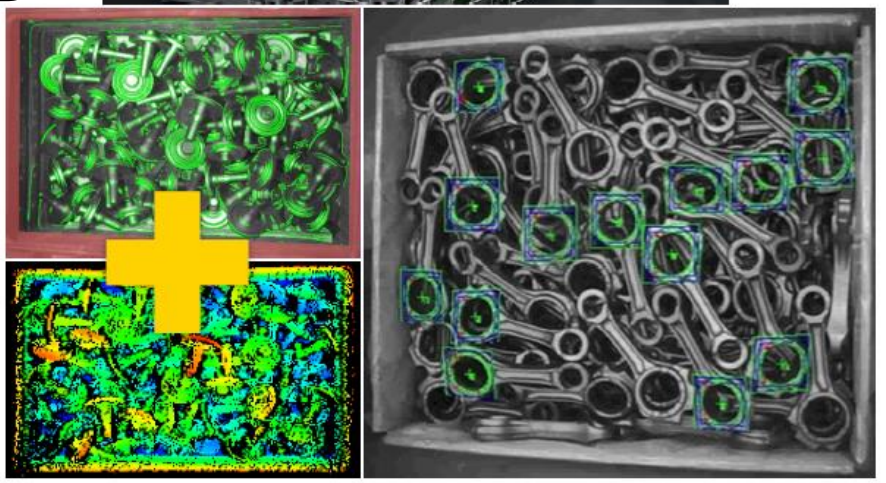




Industrial robotics and removal - 1

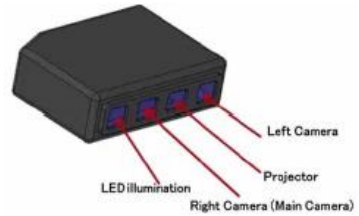
- Remote and active stereovision + right light: 3D point cloud, occlusion handling
- Edge detection + template matching → object 2D segmentation and selection
- Extraction of 3D keypoints → 3D/3D localization, then ICP
- Planning of object grasping

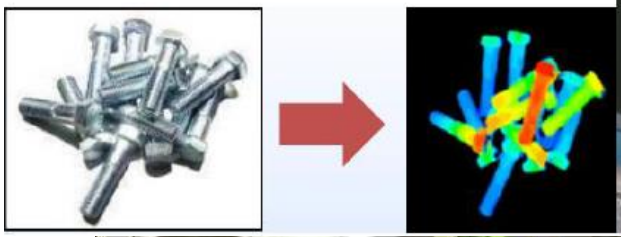
[Fanuc-devracage.mp4]



Industrial robotics and removal - 2

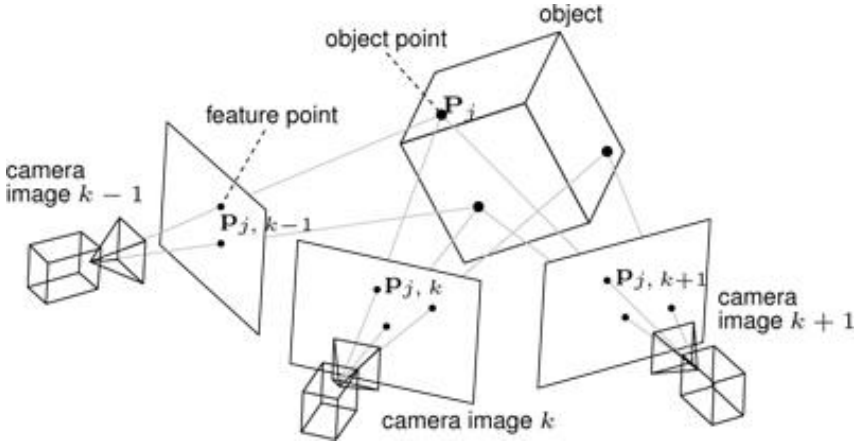
- Embedded active stereovision - ring light: 3D point cloud, occlusion handling
- Edge detection + template matching → object 2D segmentation and selection
- Detection of keypoints → 3D/3D localization, then ICP
- Planning of object grasping

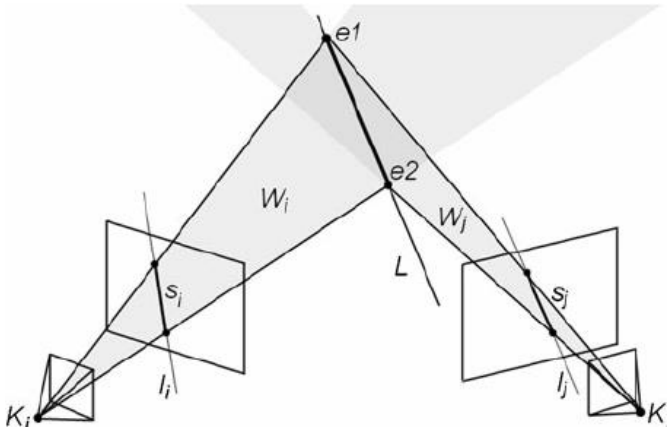




2D/3D recognition

- Generic strategy:
  - Requires a 3D model, the camera calibration, a pose hypothesis
  - 2D/3D matchings
    - Through geometrical criteria: spatial proximity between 2D primitives and 3D primitives after projection, length/orientation/overlapping whether primitives = model edges and image segments
    - Through appearance-based criteria whether model = geometry + appearance
  - Computation of the 2D/3D localisation → prediction
  - Global verification on the whole model primitive set





References

- 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 classication. 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.

References

- **[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.

Summary (17h C/TD and 9h TP)

1. Camera self-calibration
2. Structure from motion
3. 3D modeling
4. 3D localisation and recognition
5. Applications by examples
6. Exercice correction

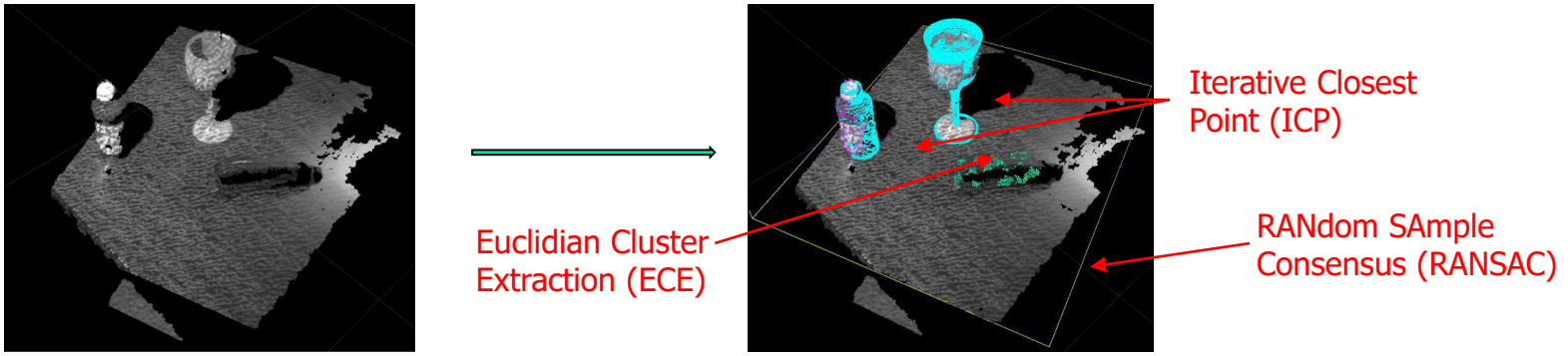
Coupling 3D vision and motion planning

- Trajectory planification in 3D with 3D uncertainties
  - Coarse position of the wheel and car cabin
  - 3D model based recognition (wheel, car cabin) : 3D point matching and 3D/3D localization
  - Online re-estimation of the trajectory according to the inferred pose.

[Fanuc-MotionPlanning.mp4]

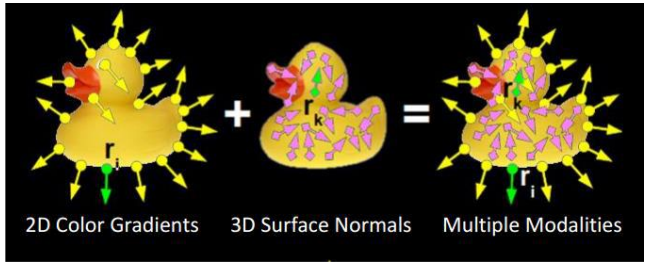
3D segmentation: « TableTop »

- « Inputs »: 3D points cloud, 3D models of the objects
- Hypotheses: objects placed on a plane surface, symmetrical objects by rotation, spatially separated objects
- Strategy, 3 steps:
  - Random Sample Consensus (RANSAC) for the plane
  - « Euclidian Cluster Extraction » (ECE) for the objects
  - « Iterative Closest Point » (ICP)

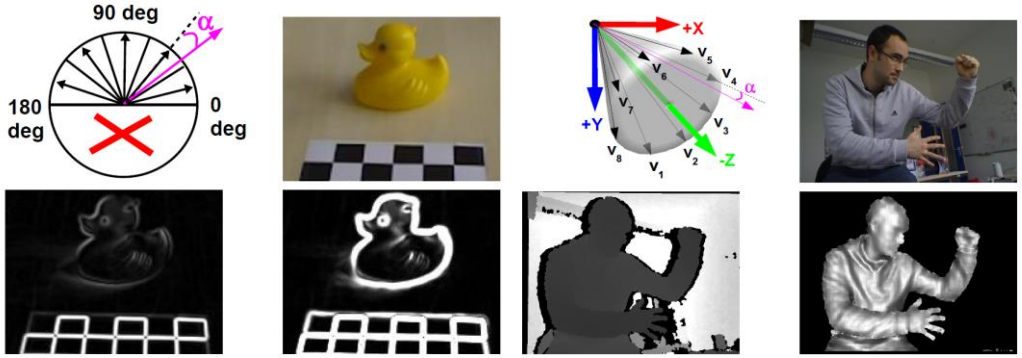


2D/3D template matching: « Linemod »

- « Inputs »: RGB-D image, knowledge of characteristic templates (color gradient, 3D normals)



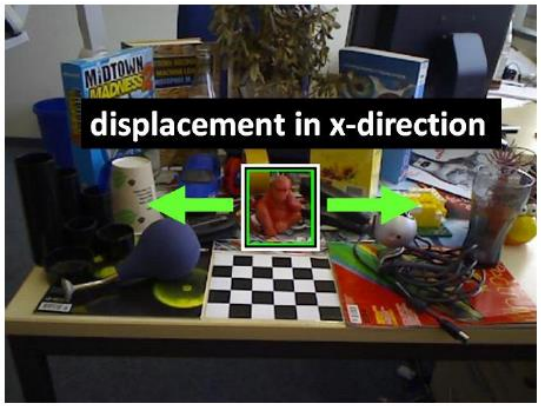
- Strategy *[Hinterstoisser et al., 2011]*
  - (1) Building of P templates/views with M modalities *i.e.* $\{O_m(r)\}_{m\in M, r\in P}$ (P~500, M=2)

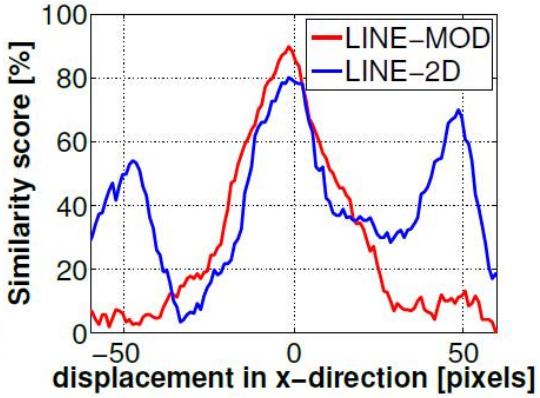


2D/3D template matching: « Linemod »

- Strategy [Hinterstoisser *et al.*, 2011]
  - (2) Sliding window
  - (3) Computation of similarity score between templates O_m *vs.* image $I_m, m \in \{g, D\}$

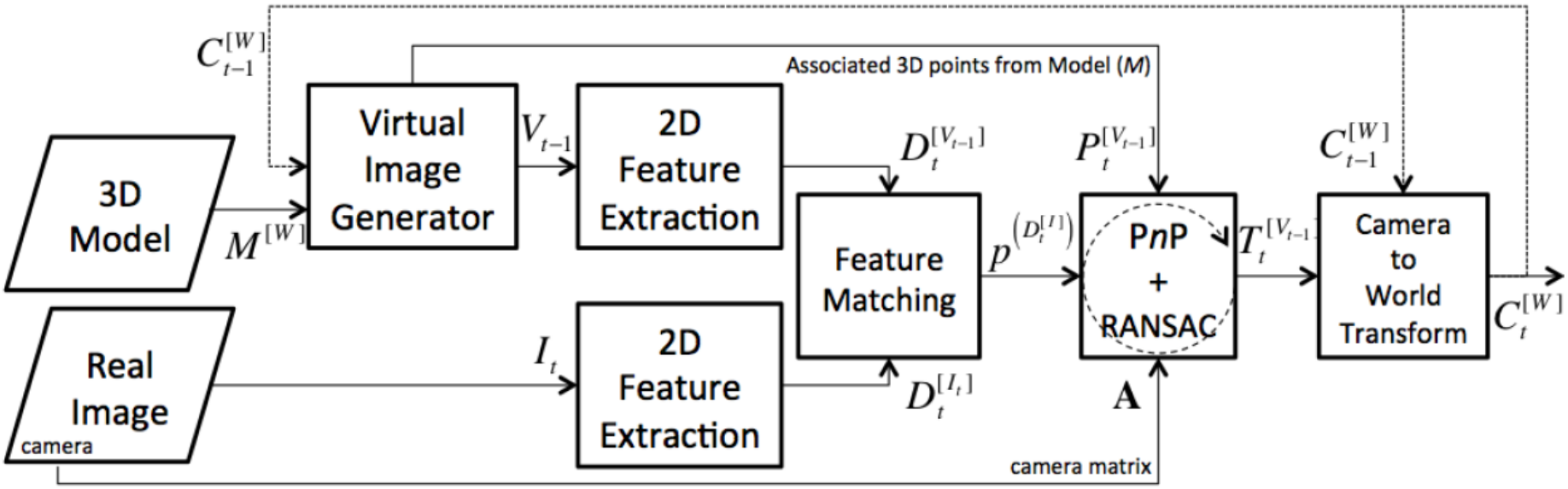
$$Arg \max_t \sum_{r \in P, m \in \{g,D\}} |O_m^T(r).I_m(t)|$$





Monocular localisation: « PnP »

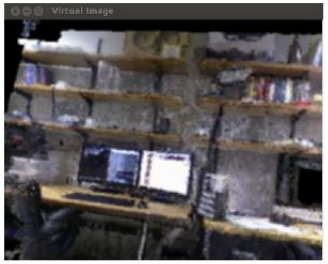
- Methodology [Jamarillo et al., 2013]:
  - (1) Offline creation of a RGB-D 3D scene model: ICP,...



Monocular localisation: « PnP »

- Methodology [Jamarillo *et al.*, 2013]:
  - (2) Virtual image generation t-1 (loc. t-1)
  - (3) SURF matchings between t-1 and t
  - (4) PnP localisation based on RANSAC $T_t^{[V_{t-1}]}$ and absolute localisation update

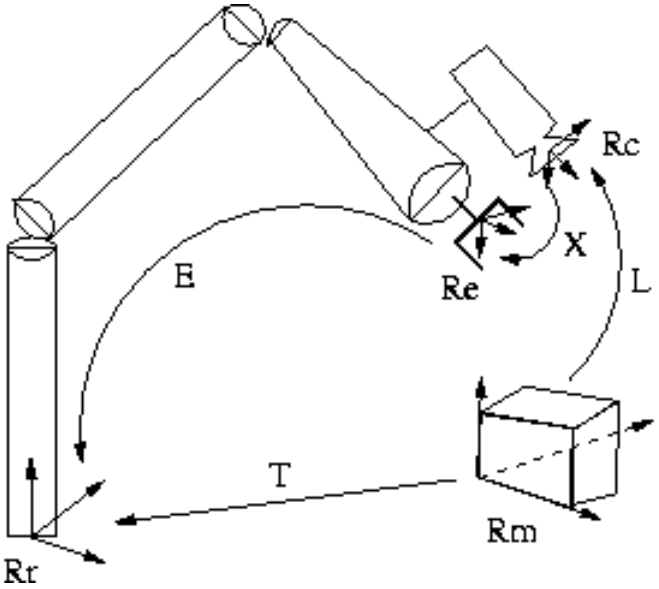
$$C_t^{[W]} = T_t^{[V_{t-1}]} . C_{t-1}^{[W]}$$



[Loca-SURF.avi]

Calibration and object manipulation

- Hand/eye calibration
  - Goal : guide the gripper movement thanks to 3D visual data
    ➔ characterize the relative position sensor *vs.* gripper
  - Frames and transformations :



R_r: reference frame of the manipulator
R_e: frame associated to the effector
R_c: camera frame
R_m: object model frame
[L]: object localisation in camera frame R_c
[X]: hand/eye matrix (to calibrate)
[E]: Pose of the effector frame *vs.* absolute frame (R_r)
[T]: object localization in the frame (R_r)
$[T_{prise}]$: pose gripper / R_m for object grasping

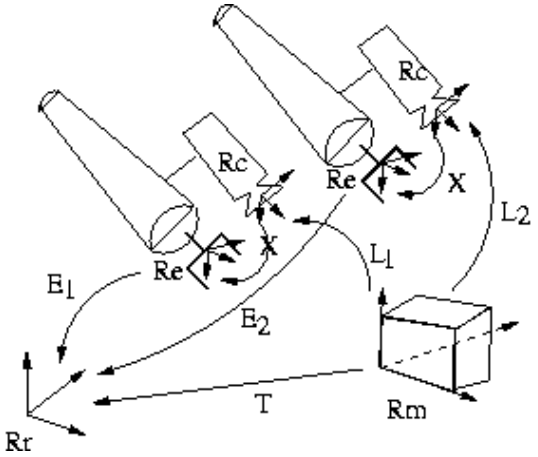
Calibration and object manipulation

- Object grasping based on a look&move approach

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

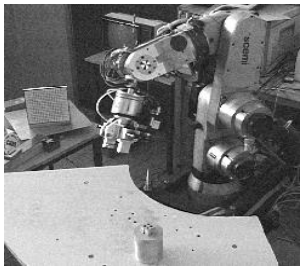
- Two strategies for hand/eye calibration
  - Localize the gripper thanks to features located in the gripper frame (pattern at the end of the gripper)
  - Perform two localizations on the same object from two arm positions (18 parameters to estimate) [Tsai87] :

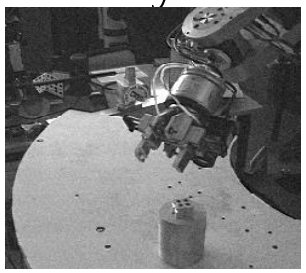
$$[L]_1^{-1} . [X]^{-1} . [E]_1^{-1} . [E]_2 . [X] . [L]_2 = I \Rightarrow [A] . [X] = [X] . [B]$$

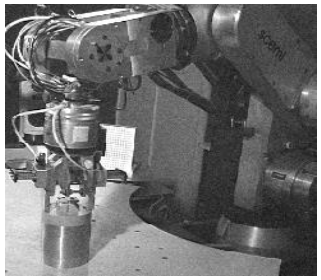


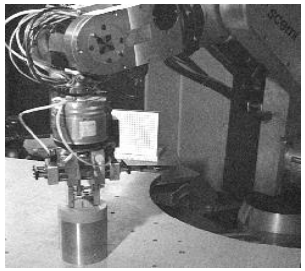
Recognition and object manipulation

- 3D position-based servoing :
  - Get as close as possible to the grasping pose then perform the last effector movement blindly
  - An accurate object localization
  - An accurate hand/eye calibration


Scene and init pose


Last pose during the approach


Grasping pose


Object grasping

- Steps :

❶ Image acquisition+segmentation
❷ prediction of the object localization (thanks to the CAD scene model)
❸ 2D segments / 3D edges matchings
❹ Arm motion
❺ Get back in ❶

Recognition and object manipulation

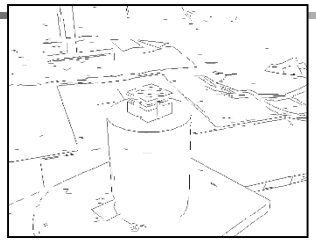
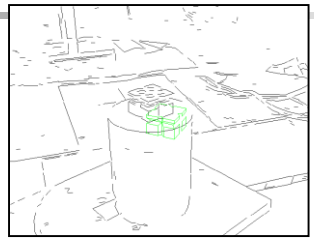
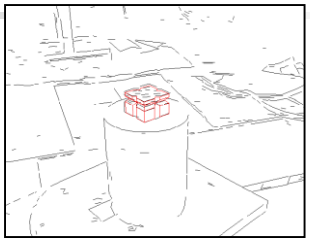





Image I | *Segmented image I* | *Initial prediction* | *Recognition*

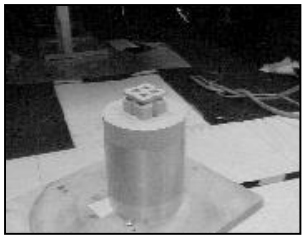
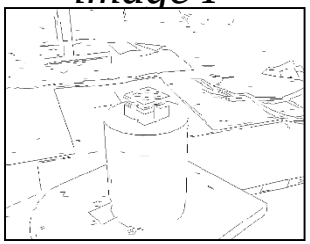
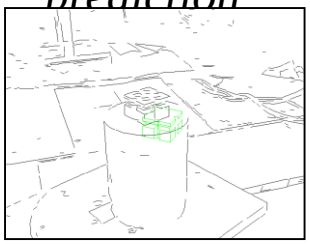
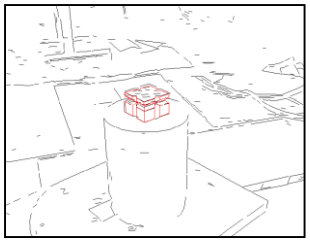





Image I | *Segmented image II* | *Initial prediction* | *Recognition*

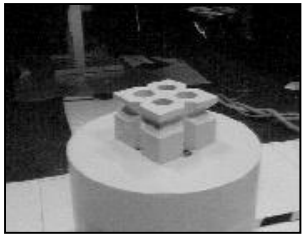
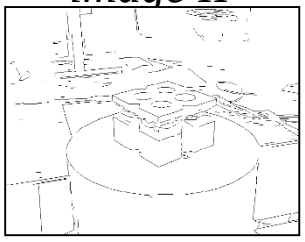
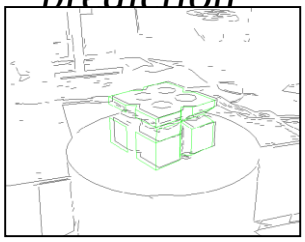
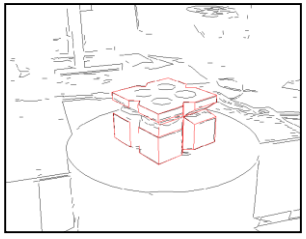
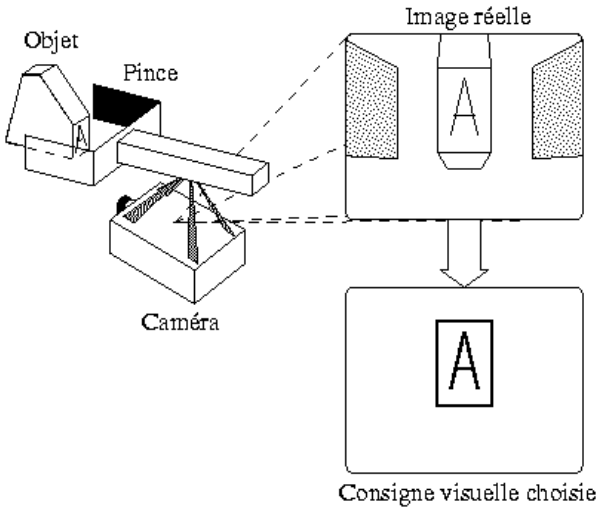





Image III | *Segmented image III* | *Prediction* | *Recognition*

Recognition and object manipulation

- How to get rid of the 3D localization ?

➔ 2D visual servoing *i.e.* servoing on reference image features (2A)



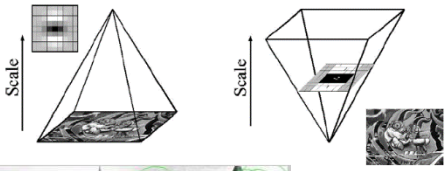
References

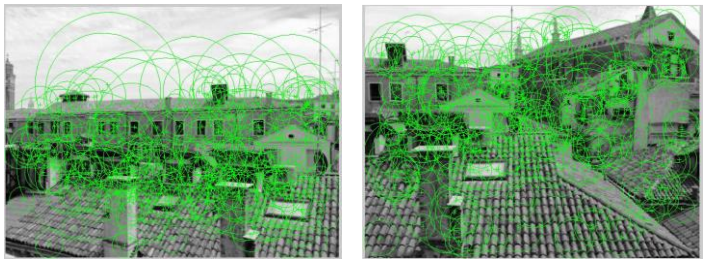
- Tabletop, url http://www.ros.org/wiki/tabletop_object_detector
- **[Hinterstoisser *et al.*, 2011]** Multimodal templates for real-time Detection of Texture-less Objects in Heavily Cluttered Scenes. Hinterstoisser, Holzer, Cagniart, Ilic, Konolidge, Navab, Lepetit. Int. Conf. on Computer Vision, 2011.
- **[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.

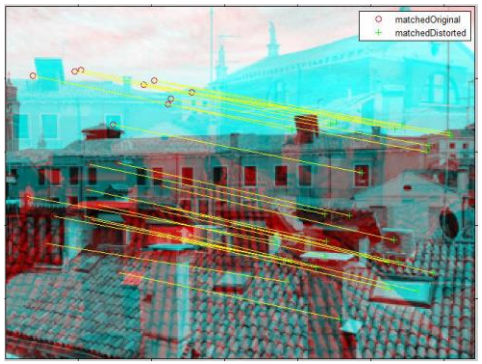
3D/3D recognition

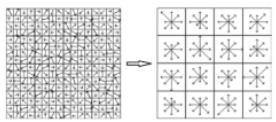
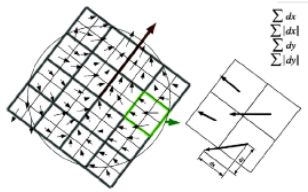
- Variable scale detection: KPQ-SI, ~~SURF 3D~~,...
  - SURF [Bay *et al.*, 2006] : SIFT vs. SURF... en 2D

$$H(x,\sigma = \begin{bmatrix} L_{xx}(x,\sigma) & L_{xy}(x,\sigma) \\ L_{xy}(x,\sigma) & L_{yy}(x,\sigma) \end{bmatrix}$$



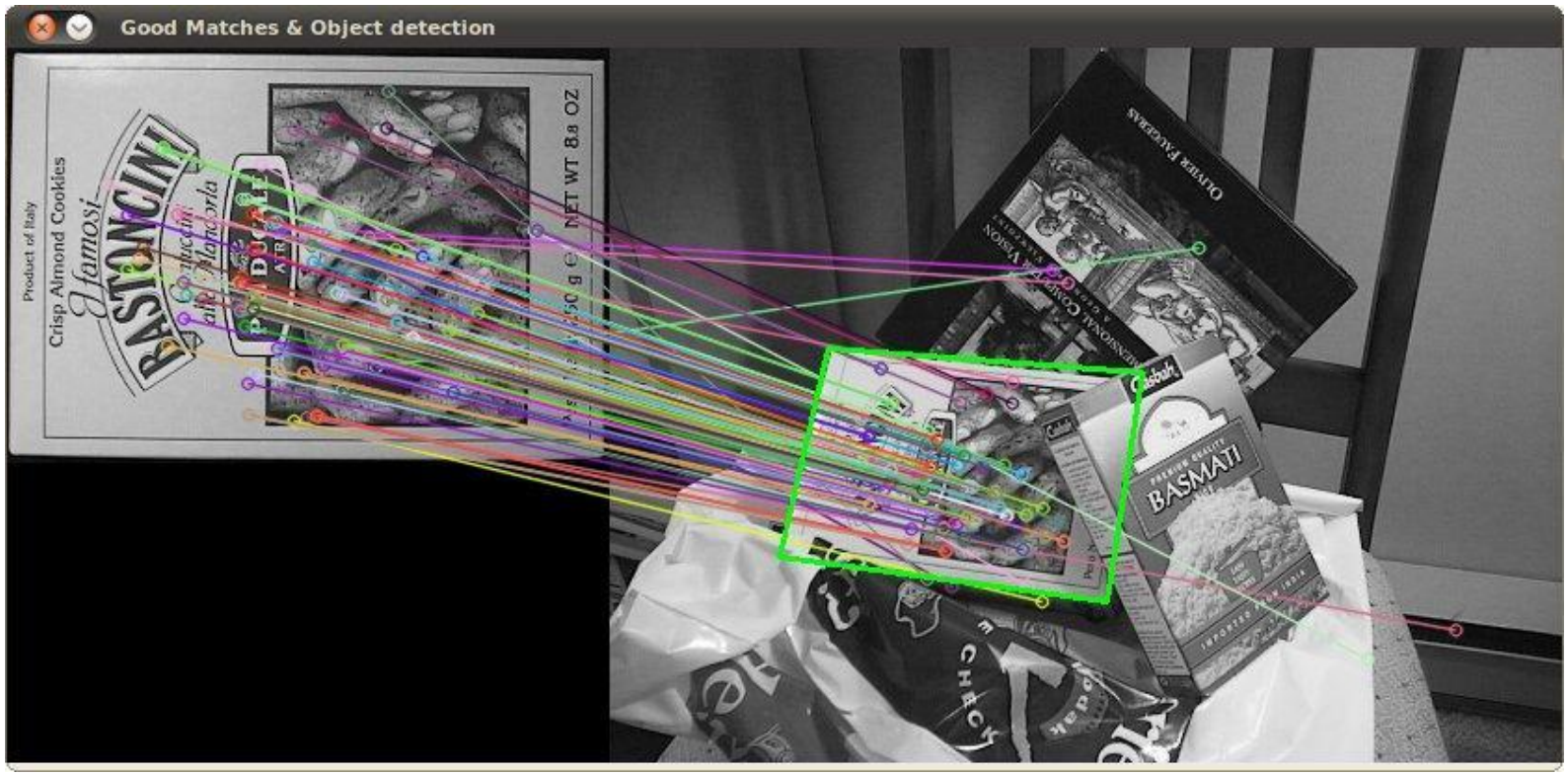




	SIFT	SURF
Scale Space	DOG convolved with difference size images	difference size box filter convolved with integral image
Key points detection	Local extrema detection->Non-maximum suppression->Eliminate edge responses with Hessian matirx.	Deter mint the potential key points with Hessian matrix and Non-maximum suppression
Orientation	image gradient magnitudes and orientations are sampled around the key point location, using the scale of the key point to select the level of Gaussian blur for the image	A sliding orientation window of size pi/3 detects the dominant orientation of the Gaussian weighted Haar wavelet responses at every sample point within a circular neighbourhood around the interest point.
Descriptor	The key point descriptor allows for significant shift in gradient positions by creating orientation histograms over 4x4 sample regions. The figure shows 8 directions for each orientation histogram , with the length of each arrow corresponding to the magnitude of that histogram entry.  Figure 1: SIFT Descriptor Generation	An oriented quadratic grid with 4×4 square sub-regions is laid over the interest point (left). For each square, the wavelet responses are computed from 5×5 samples. For each field, we collect the sums dx, |dx|; dy and |dy|, computed relatively to the orientation of the grid (right). 

3D/3D recognition

- Variable scale detection: KPQ-SI, SURF 3D, etc.
  - SURF [Bay *et al.*, 2006] : SIFT vs. SURF… en 2D



3D/3D recognition

- Detection at variable scale: KPQ-SI, SURF 3D [Knopp *et al.,* 2011],...
  - Space voxelization (cube 256^3)
  - 3D Hessian computation, maxima selection
  - 3D descriptor computation invariant to rotation and scale
    - Computation of the main directions
    - Definition of a neighborhood 3x3x3 around the point
    - Response Haar wavelet discretized in 6D

    → Final descriptor with 162 components