8th Workshop on Planning, Perception and Navigation for Intelligent Vehicles

A bridge between Robotics and ITS technologies

Sensor based navigation

Philippe Martinet

Professor at Ecole Centrale of Nantes

Co-authors: G. Garcia, S. Dominguez Quijada, O. Kermorgant

November 1st, 2016

Rio de Janeiro, Brazil

Centrale Nantes

IRCCyN

cnrs advancing the frontiers

Content

- **Introduction**
- Generic models for cameras
- Visual geometry for localization
- Topological navigation system
- Illustrations
- Conclusion

From LASMEA, IRCCyN to LS2N

Prof. ECN
IRCCyN (11-16)
Robotics

Prof. ECN
LS2N (1/1/17)
ARMEN

Ass. Prof. CUST
LASMEA (90-00)
GRAVIR/CSR

Prof. IFMA
LASMEA (00-11)
GRAVIR/ROSACE

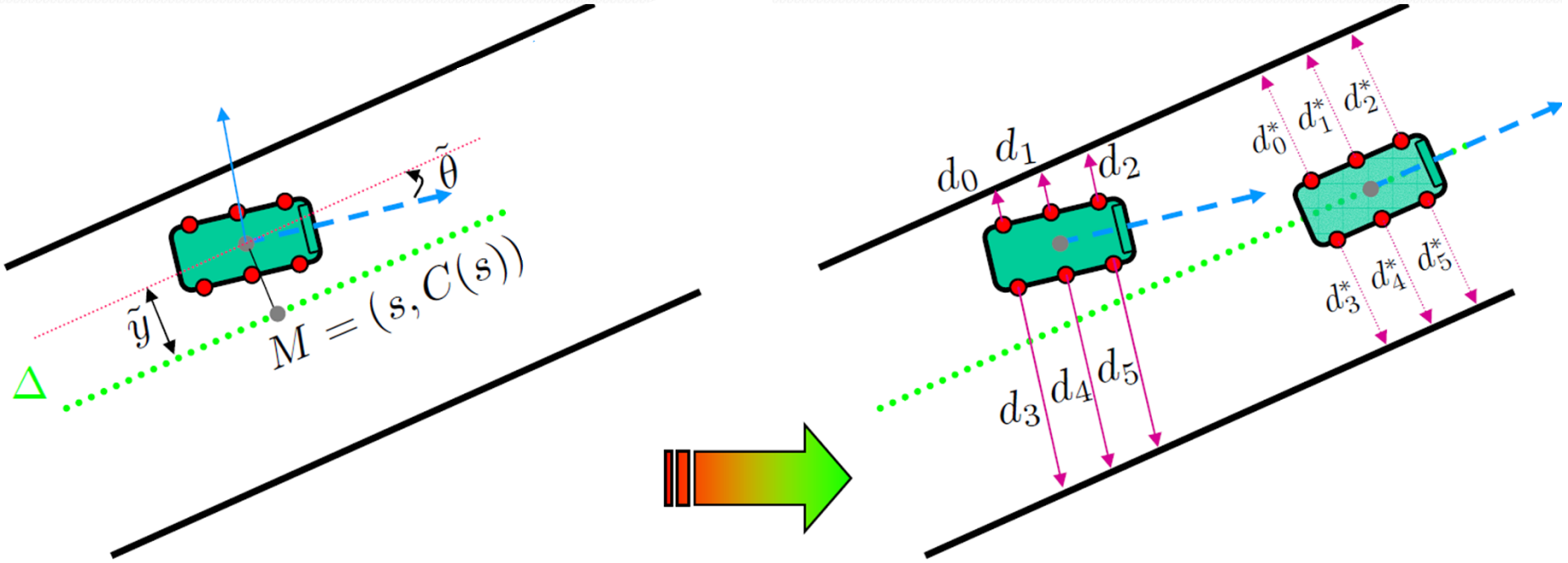
ARMEN: Research Areas

Leaders: S. Briot (CNRS Researcher), P. Martinet (Pr)
V. Begoc (A.P.), S. Briot (Researcher), A. Chriette (A.P.), G. Garcia (A.P.), O. Kermorgant (A.P.), P. Martinet (Pr)
S. Dominguez (RE), X. Koreki (RE), 9 Phds, 9 masters

- **Robot Design and Environments**
  - *Conceptual Design*
  - *Robust Design*
  - *Control-based Design*
  - *Design using new materials*
  - *Reduction of energy consumption*
- **Robot Interactions with the environment**
  - *Multi-Sensor Based Control*
  - *Control of reconfigurable Systems*
  - *Autonomous Robots*
- **Robot Perception of the environment**
  - *Multi robots and multisensor perception*
  - *Optimal Perception*
  - *Observers and identification*

Sensor based control: History

[ESPIAU87]



$$d = (d_0, d_1, d_2, d_3, d_4, d_5)^T$$

Lateral control in cartesian space:

Bringing $(\tilde{y}, \tilde{\theta})$ to $(0,0)$

Explicite declaration of Δ and required cartesian localization

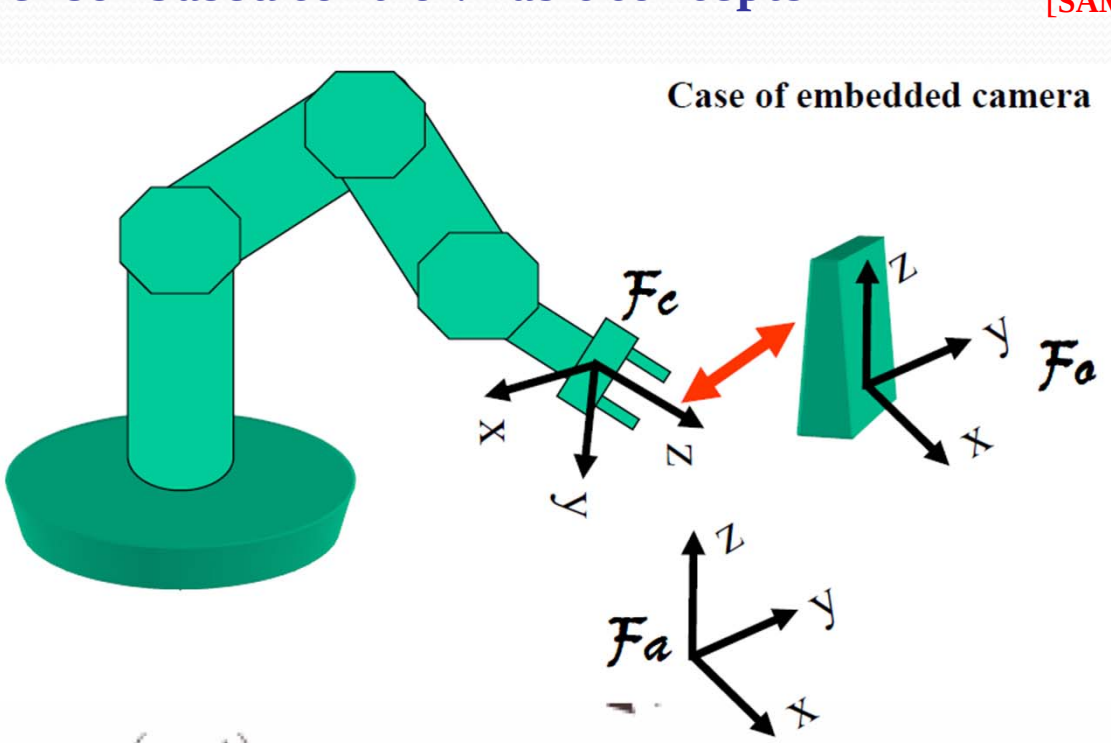
Lateral control in sensor space:

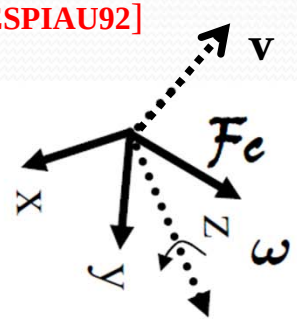
Bringing d to d^*

No explicite declaration of Δ and not required cartesian localization

Sensor based control: Basic concepts

[SAMSON90,ESPIAU92]





$\mathbf{x}$: 3D pose

$$\mathbf{s} = \mathbf{s}(\mathbf{x}, t)$$

Interaction matrix

If (fixed object) $\frac{\partial \mathbf{s}}{\partial t} = 0$

$$\dot{\mathbf{s}} = \frac{\partial \mathbf{s}}{\partial \mathbf{x}} \frac{d\mathbf{x}}{dt} + \frac{\partial \mathbf{s}}{\partial t} \Rightarrow \dot{\mathbf{s}} = \mathbf{L}_s \mathbf{v} + \frac{\partial \mathbf{s}}{\partial t}$$

$$\Downarrow$$

$$\dot{\mathbf{s}} = \mathbf{L}_s \mathbf{v}$$

Sensor/Vision based control: *Basic bibliography*

Book, Book Chapter

Robot Control: Task function approach [C. Samson, M. Le Borgne, B. Espiau, Clarendon Press, 1991]
Visual servoing [K. Hashimoto 1993]
Robotics Handbook : Chapter 24 Visual Servoing and Visual Tracking [Chaumette, Hutchinson 2008]
Robotics: Modelling, Planning and Control chapter 10 Visual servoing [Siciliano, B., Sciavicco, L., Villani, L., Oriolo, G. 2009]
Visual Servoing via Advanced Numerical Methods [G. Chesi, K. Hashimoto 2010]
Robotics, Vision and Control : Fundamental Algorithms in MATLAB [Corke, Peter, 2011]

Tutorials, workshop

Tutorial on visual servoing, S. Hutchinson, G. Hager, P.I. Corke, 1996, Tutorial at ICRA, Minneapolis, Minnessota, USA
IROS 2004 Tutorial on Advanced Visual Servoing , P. Martinet, E. Cervera, F. Chaumette, E. Malis, K. Hashimoto, 2004, Tutorial at IROS, Sendai, Japan
IROS 2002 Workshop on Visual Servoing , P. Martinet, E. Cervera, N. Andreff, F. Chaumette, 2002, Workshop at IROS, Lausanne, Switzerland
Visual servo control : Part I Classical Approaches, S. Hutchinson, F. Chaumette, 2006, RAS Magazine
Visual servo control : Part II Classical Approaches, F. Chaumette, S. Hutchinson, 2007, RAS Magazine

Fundamental article *(Citations : 1681 (october 2016))*

A new approach to visual servoing in robotics. B. Espiau, F. Chaumette, P. Rives. IEEE Trans. on Robotics and Automation, 8(3):313-326, 1992

Toolboxes

Visual Servoing Toolbox, E. Cervera and P. Martinet, 2003, Matlab Toolbox, Spain and France
VISP, E. Marchand, 2008, France
Robotics Toolbox, P.I. Corke, 2011, Matlab Toolbox, Australia
Machine Vision Toolbox, P.I. Corke, 2011, Matlab Toolbox, Australia

Many authors in robotics, Few authors in ITS

Sensor/Vision based control: *Basic approaches*

Image Based Visual Servoing *IBVS*

- 2D Features are extracted from images (Points, lines, shape, moments, …)
- Photometric based features (Histogram, Entropy, …)
- Information based (Mutual information, …)

Position Based Visual servoing *PBVS*

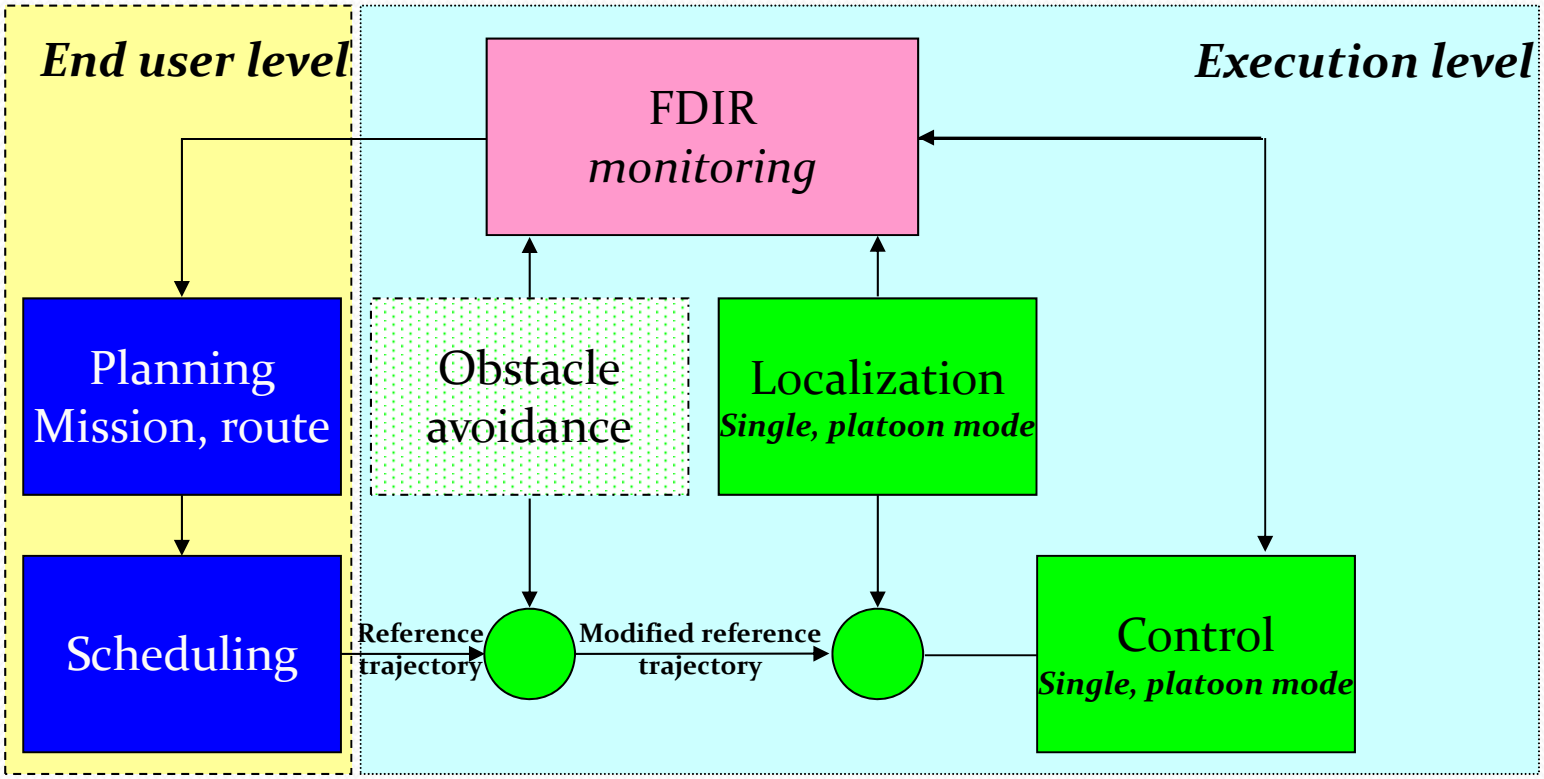
- 3D features are estimated from 2D features (Pose, Point, Line, plane, shape…)
- 3D features are estimated from pixels (Pose, Point, Line, plane, shape…)
- 3D feaures are directly given by the sensor system

Hybrid Based Visual Servoing *HBVS*

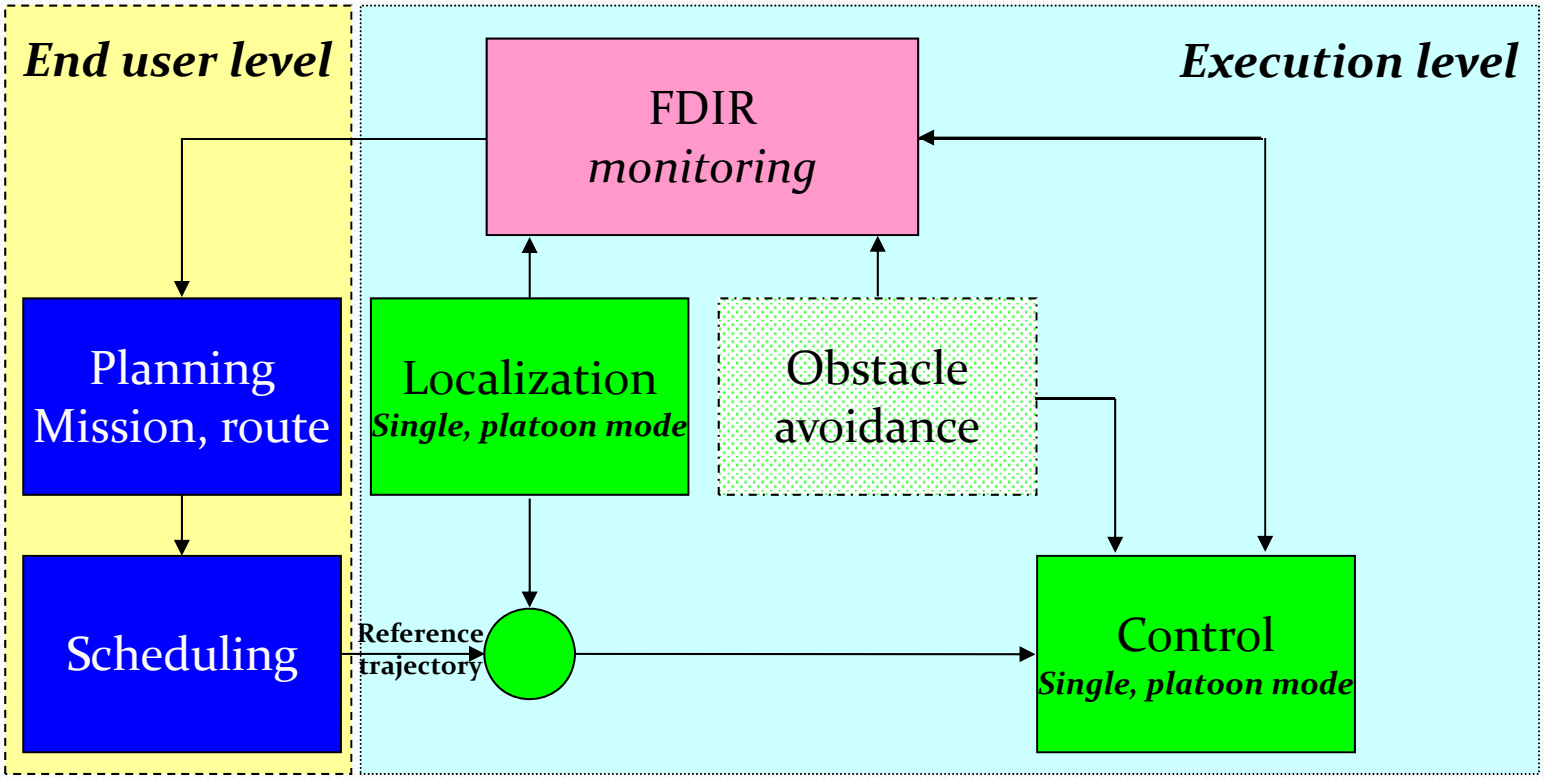
- Combination of 3D features with 2D features (2.5D, Stereo, …)
- Multi-sensors

Basically, a teaching by showing is often used in order to limit the impact of modeling errors

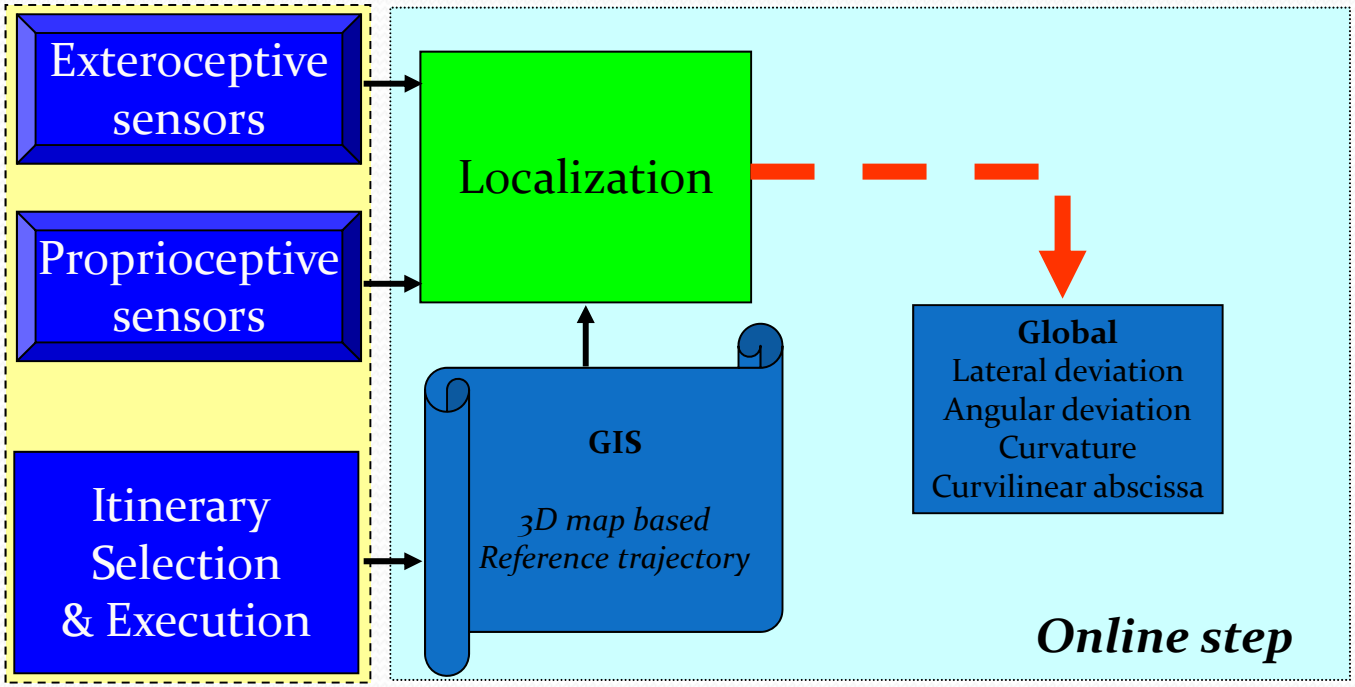
Navigation scheme



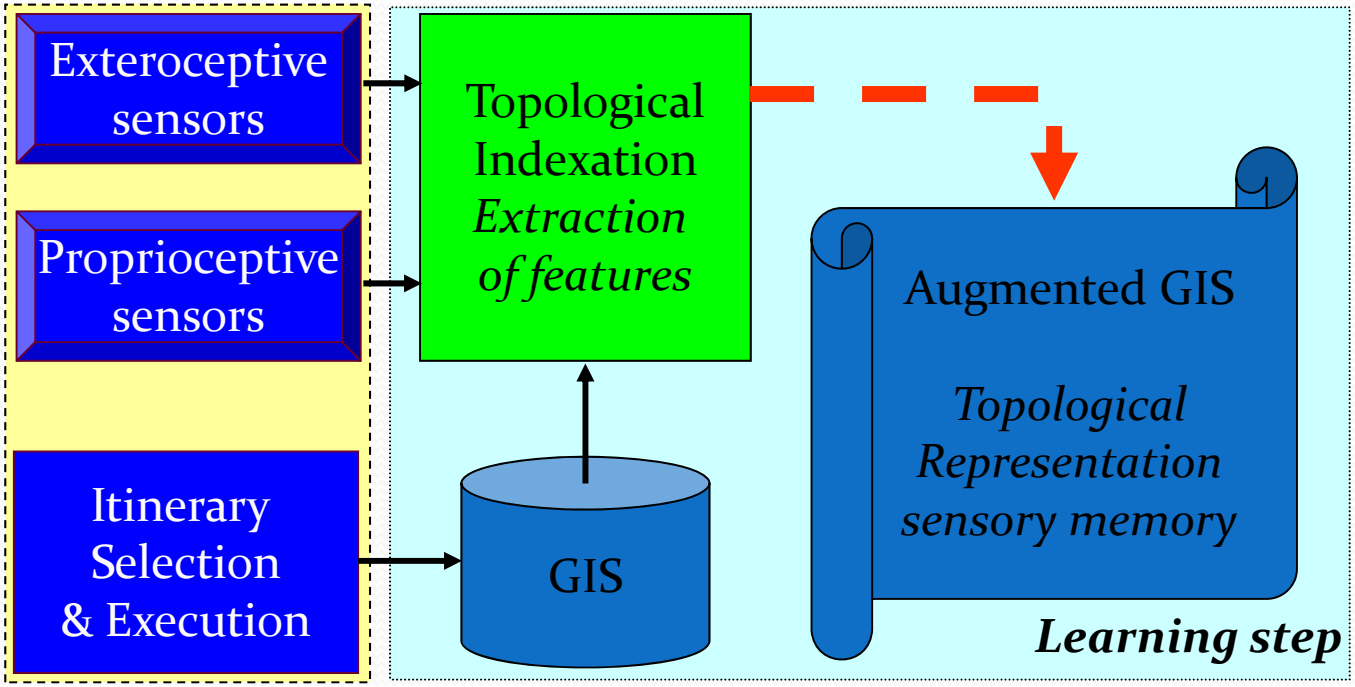
Navigation scheme



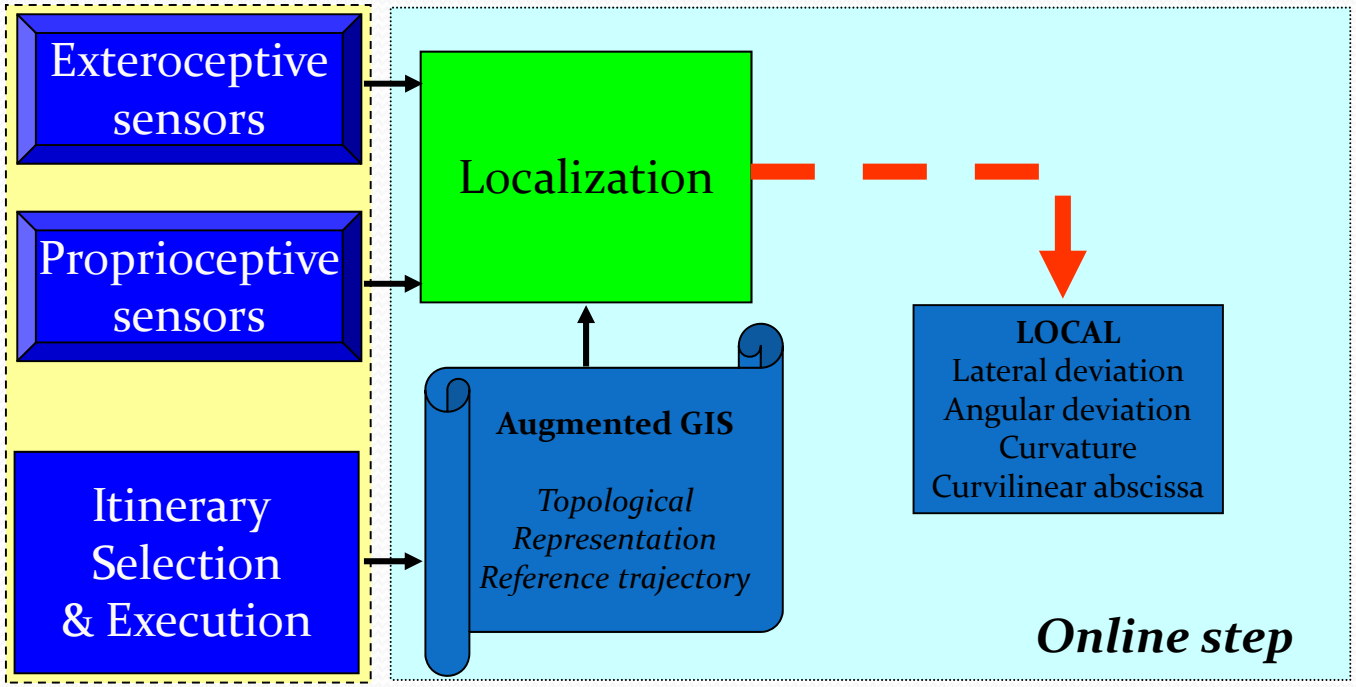
3D Map based navigation (absolute navigation)



Memory based navigation (relative navigation)



Memory based navigation (relative navigation)



Content

- Introduction
- **Generic models for cameras**
- Visual geometry for localization
- Topological navigation system
- Illustrations
- Conclusion

Different kind of cameras

Pinhole Camera

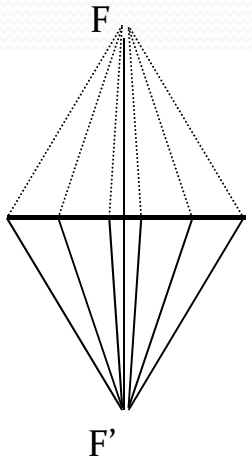
Catadioptric systems

Camera with fisheye lenses

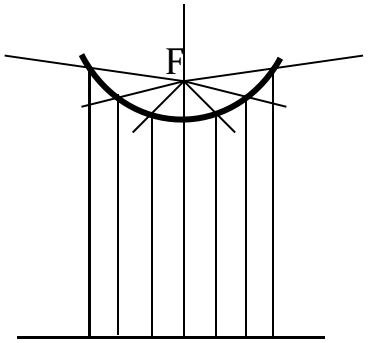
A unified model for camera

Single view point system : Classification

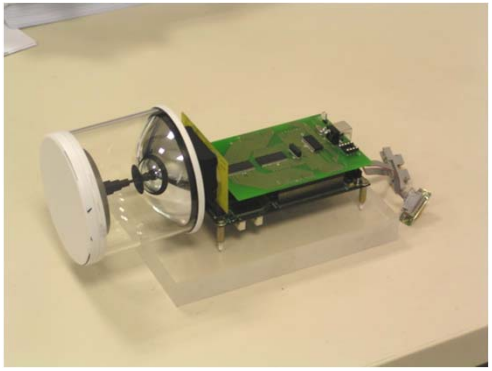
[Geyer00]

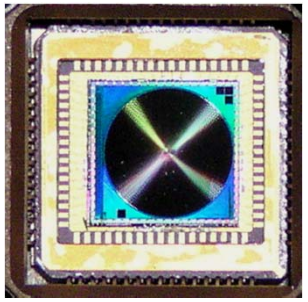


Planar

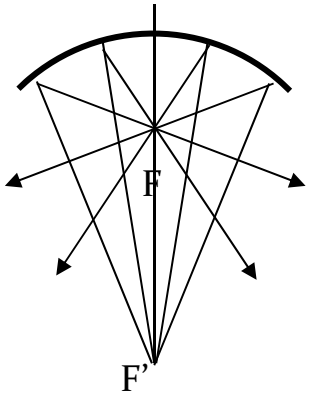


Parabolic

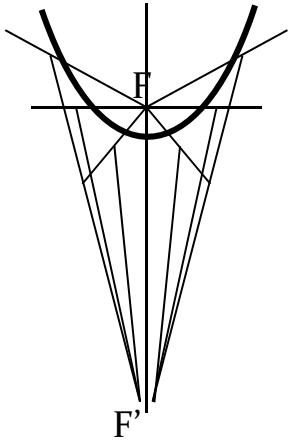




Log/polar retina



Elliptical

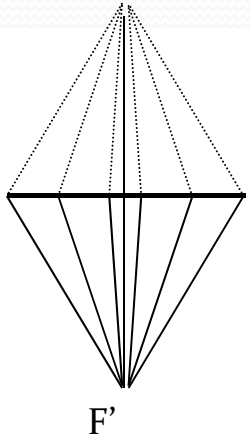


Hyperbolic

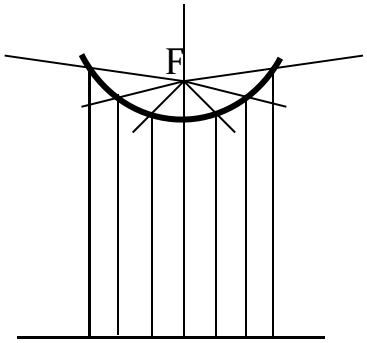
A unified model for camera

Single view point system :
Classification

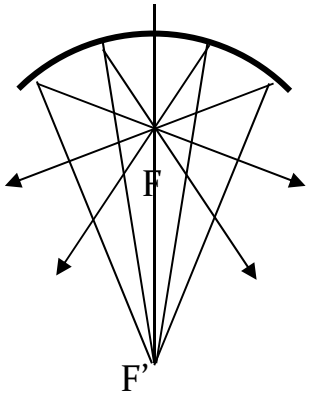
[Geyer00]



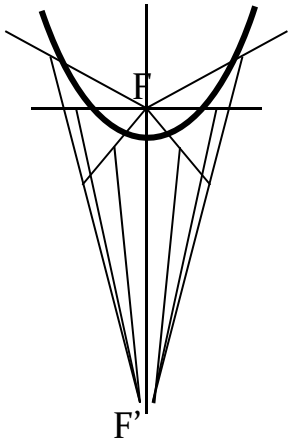
Planar



Parabolic



Elliptical



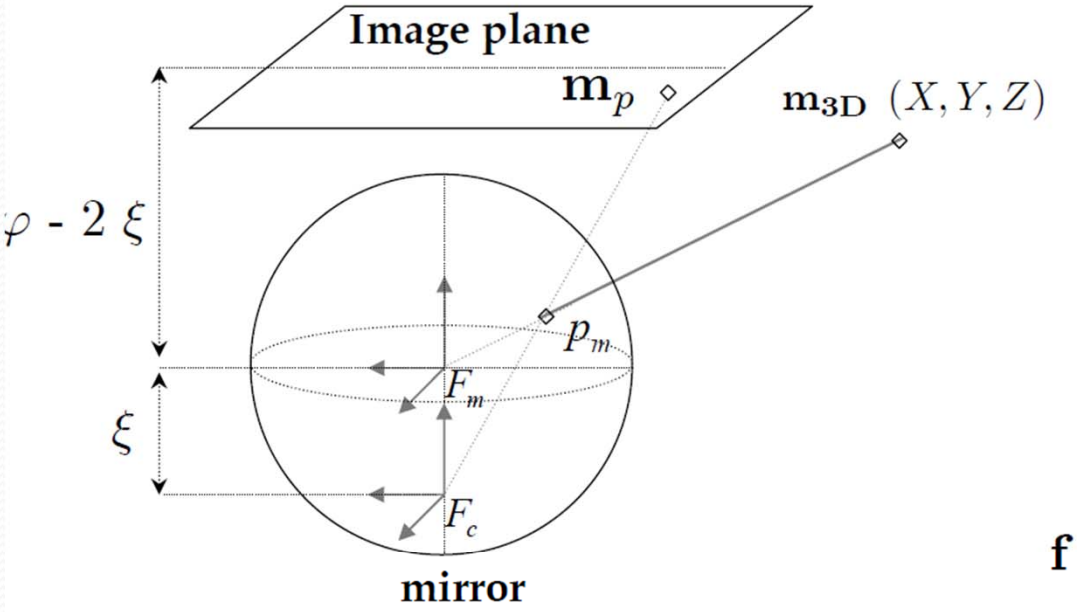
Hyperbolic

φ and ξ : Mirror parameters

Mirror type	ξ	φ
Parabolic	1	1+2p
Hyperbolic	$\frac{d}{\sqrt{d^2+4p^2}}$	$\frac{d+2p}{\sqrt{d^2+4p^2}}$
Elliptical	$\frac{d}{\sqrt{d^2+4p^2}}$	$\frac{d-2p}{\sqrt{d^2+4p^2}}$
Planar	0	1
Conventiona l	0	1

A unified model for camera

Single view point system : case of point



φ and ξ : Mirror parameters

$$\mathbf{M} = \begin{bmatrix} \varphi - \xi & 0 & 0 \\ 0 & \varphi - \xi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{K} = \begin{bmatrix} f & fs & u_0 \\ 0 & fr & v_0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{f}(\mathbf{m_{3D}}) = \begin{bmatrix} \frac{X}{\epsilon_1 Z + \xi\sqrt{X^2+Y^2+Z^2}} \\ \frac{Y}{\epsilon_1 Z + \xi\sqrt{X^2+Y^2+Z^2}} \\ 1 \end{bmatrix}$$

Special cases :

$\epsilon_1 = 1$ and $\xi = 0$ ➔ perspective projection

$\epsilon_1 = 0$ and $\xi = 1$ ➔ spherical projection

Generic Projection function

$$\mathbf{m}_p = \mathbf{K}\ \mathbf{M}\ \mathbf{f}(\mathbf{m_{3D}})$$

Case of fisheye camera

Fisheye camera model : definition

★ r = distance between X and the image point
★ θ = angle between the incoming ray and the principal axis

★ $\mathcal{F}_m$: camera frame
★ $\mathcal{F}_i$: image frame

Pinhole camera

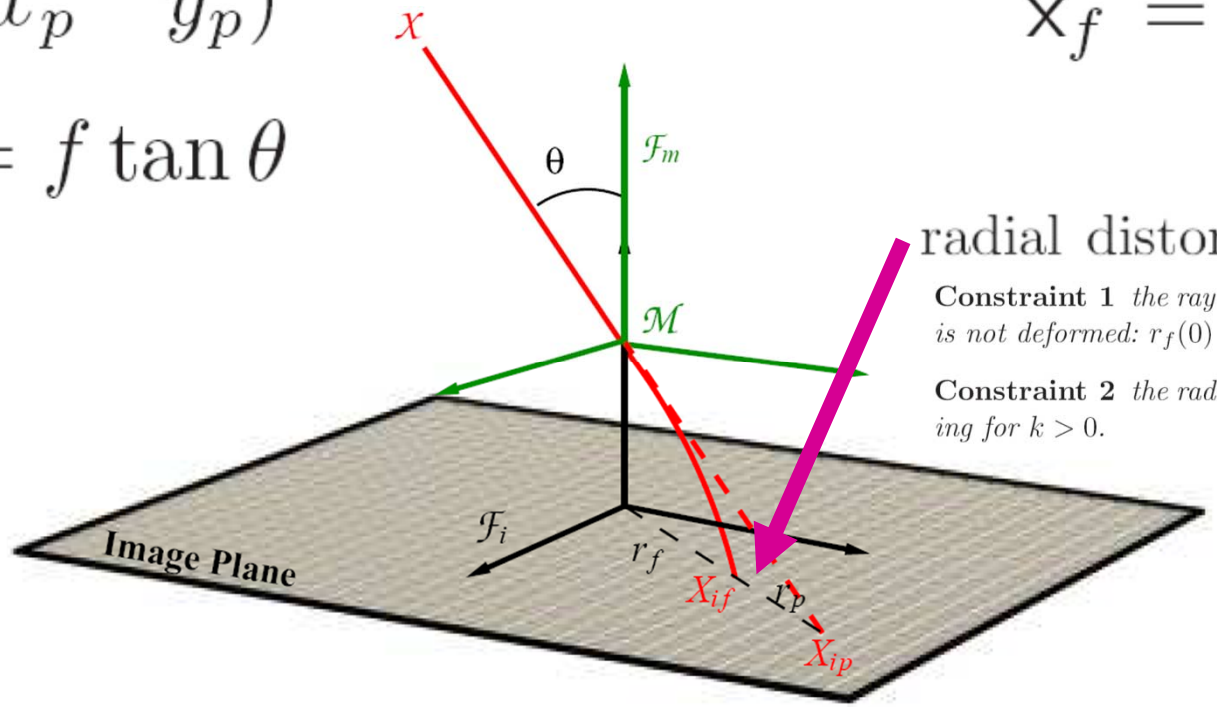
$$\mathsf{x}_p = (x_p \quad y_p)^T$$

$$\mathsf{X} = (X\,Y\,Z)^T$$

$$r_p(\theta) = f \tan\theta$$

Fisheye camera

$$\mathsf{x}_f = (x_f \quad y_f)^T$$



radial distortions

Constraint 1 *the ray arriving along the distortion axis is not deformed:* $r_f(0) = 0$.

Constraint 2 *the radius* $r_f(k)$ *is monotonically increasing for* $k > 0$.

Case of fisheye camera

Fisheye camera model

Pinhole based models

Mapping : $r_p \xrightarrow[T_1]{} r_f$

[T. Pajdla97,Hartley00,Zhang98,Ma2006…]

[Fitzgibbon01] [Li05] [Basu95] [Devernay01] [Hartley07]

Captured rays based models

Mapping : $\theta \xrightarrow[T_2]{} r_f$

[Kingslake89] [Fleck94,Stevenson95] [Ray94] [Smith92]

[xiong97,kannala04,schwalbe05,Scaramuzza06] [Kumler00] [Bakstein02]

Case of fisheye camera

Generic camera model : Unified Spherical Model

For fisheye camera

[Courbon07]

$$\mathbf{r}_f = r_f(r_p) = \frac{\frac{f_f}{f_p} r_p}{1+\xi\sqrt{\frac{r_p^2}{f_p^2}+1}}$$

is a T_1-mapping linking the perspective Radius r_p and the fisheye radius r_f

Constraint 1 : easily verified $(r_f(0) = 0)$
Constraint 2 : $r_f(k)$ is monotonically increasing for $k > 0$

$$\mathbf{r}_f = r_f(\theta) = \frac{f_f \tan\theta}{1+\xi\sqrt{\tan^2\theta+1}}$$

is a T_2-mapping linking the radius r_f and the incidence angle θ

The generic camera model is candidate for fisheye camera modeling

Case of fisheye camera an Enhanced Unified Model

Unified Model

$$\boldsymbol{m} = \begin{pmatrix} x_m \\ y_m \\ 1 \end{pmatrix} = \begin{pmatrix} \dfrac{x}{z+\xi\rho} \\ \dfrac{y}{z+\xi\rho} \\ 1 \end{pmatrix}$$

$$\rho = \sqrt{x^2 + y^2 + z^2}$$

ξ is a projection parameter

[MEI07, Courbon07-12]

Enhanced Unified Model

$$\boldsymbol{m} = \begin{pmatrix} \dfrac{x}{\alpha\rho + (1-\alpha)z} \\ \dfrac{y}{\alpha\rho + (1-\alpha)z} \\ 1 \end{pmatrix}$$

$$\rho = \sqrt{\beta(x^2 + y^2) + z^2}$$

$$K = \begin{pmatrix} f_u & 0 & u_0 \\ 0 & f_v & v_0 \end{pmatrix}$$

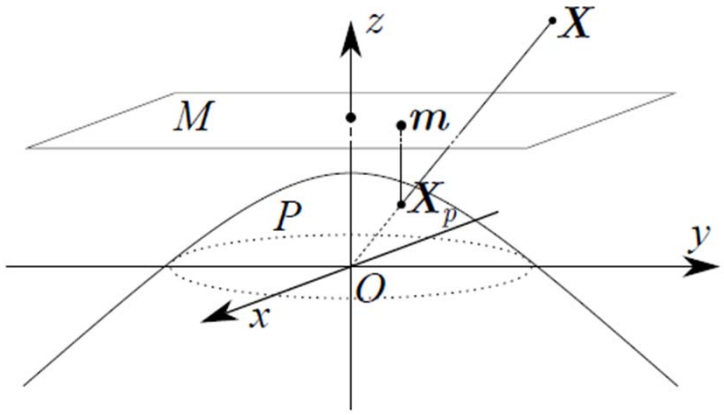
$$\boldsymbol{p} = K\boldsymbol{m}$$

[Khomutenko RA-L16]

Case of fisheye camera an Enhanced Unified Model

[Khomutenko RA-L16]

Projection surface



If every projected point can be written as:

$$\boldsymbol{m} = \left(\frac{x}{\eta(\boldsymbol{X})} \quad \frac{y}{\eta(\boldsymbol{X})} \quad 1 \right)^T$$

$\eta : \mathbb{R}^3 \to \mathbb{R}_+$ is homogeneous of degree 1

The projection surface P is defined by:

$$\eta(\boldsymbol{X}) = 1$$

All the points like $\boldsymbol{X}_p = \dfrac{\boldsymbol{X}}{\eta(\boldsymbol{X})}$ belong to the projection surface:

$$\eta(\boldsymbol{X}_p) = \eta\left(\frac{\boldsymbol{X}}{\eta(\boldsymbol{X})}\right) = \frac{\eta(\boldsymbol{X})}{\eta(\boldsymbol{X})} = 1$$

For the Enhanced Unified Model:

$$\underset{\gamma}{\eta} = (1-\alpha)z + \alpha\sqrt{\beta(x^2+y^2)+z^2}$$

Let us define the inversion g of the projection f so that: $f(g(\boldsymbol{m})) = \boldsymbol{m}$

The following mapping would transform $\boldsymbol{m}$ into $\boldsymbol{X}_p$:

$$g : \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \mapsto \begin{pmatrix} x \\ y \\ z(x,y) \end{pmatrix}$$

The inverse mapping is: $z = \dfrac{1-\alpha^2\beta r^2}{\alpha\sqrt{1-(\alpha-\gamma)\beta r^2}+\gamma}$

$r^2 = x^2 + y^2$

Content

- Introduction
- Generic models for cameras
- **Visual geometry for localization**
- Topological navigation system
- Illustrations
- Conclusion

Partial euclidian reconstruction

Unified model - Case of points

Introducing $\eta = s\frac{\rho}{|Z|} = \sqrt{1 + X^2/Z^2 + Y^2/Z^2} > 0$ s=sign(Z)

$$\begin{cases} x = \frac{X/Z}{1+\xi\eta} \\ y = \frac{Y/Z}{1+\xi\eta} \end{cases} \quad \begin{cases} (1+\xi\eta)\, x = \frac{X}{Z} \\ (1+\xi\eta)\, y = \frac{Y}{Z} \end{cases} \quad \begin{cases} \eta^2 - (x^2+y^2)(1+\xi\eta)^2 - 1 = \end{cases}$$

$$\eta = \frac{\pm\gamma_x + \xi(x^2+y^2)}{1-\xi^2(x^2+y^2)}$$

η computed without ambiguity

where $\gamma_x = \sqrt{1 + (1-\xi^2)(x^2+y^2)}$

$$\mathbf{m_m} = (\eta^{-1} + \xi)\overline{\mathbf{m}} \quad \text{with} \quad \overline{\mathbf{m}} = [x \; y \; \frac{1}{1+\xi\eta}]^T$$

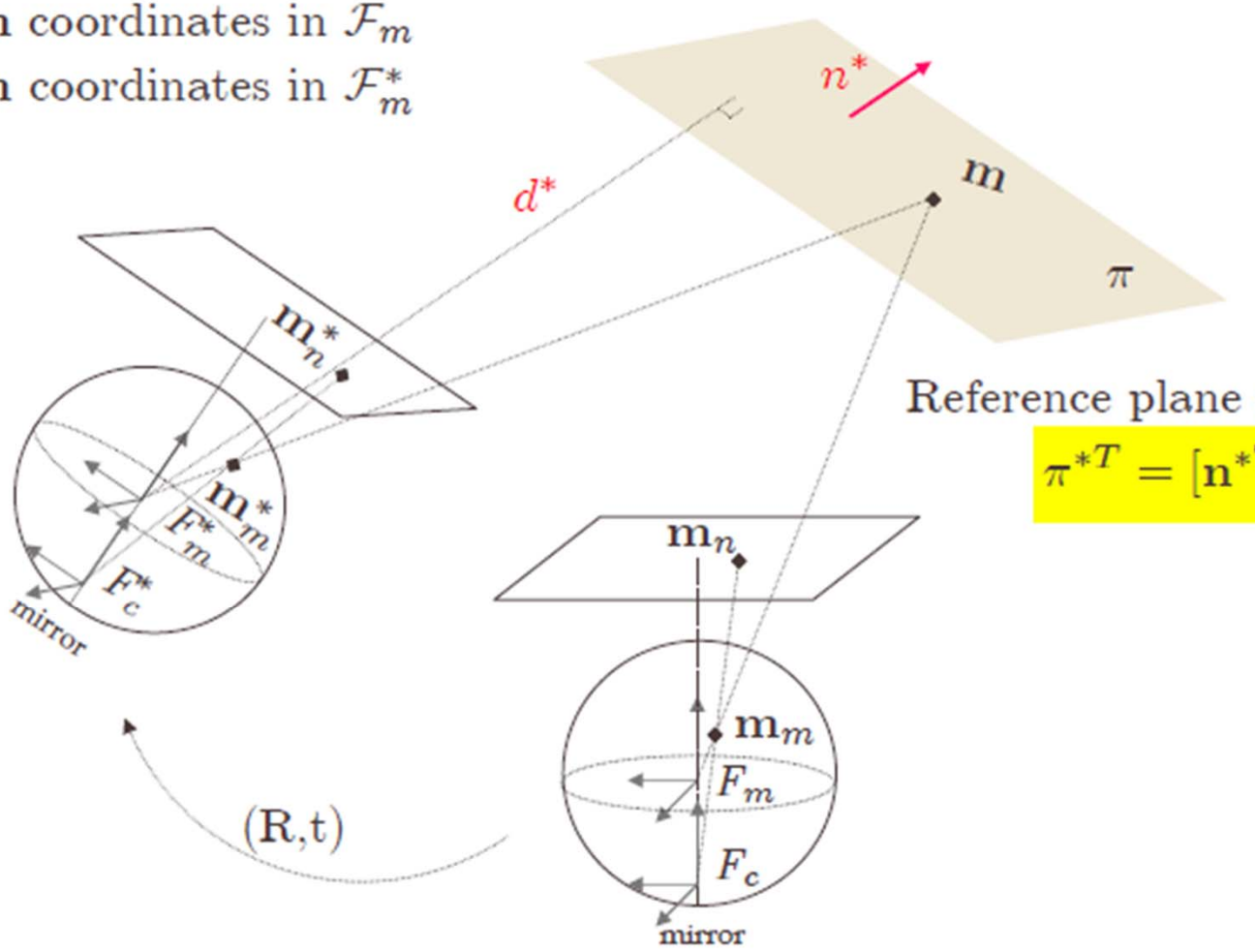
$\mathbf{m}_n$

Partial euclidian reconstruction

Unified model - Case of points

$[X\ Y\ Z]^T$ **m** coordinates in $\mathcal{F}_m$

$[X^*\ Y^*\ Z^*]^T$ **m** coordinates in $\mathcal{F}_m^*$



Reference plane π in F_m^*

$$\pi^{*T} = [\mathbf{n}^{*\mathbf{T}} \quad -d^*]$$

Partial euclidian reconstruction Unified model - Case of points

$$\mathrm{m}_m = (\eta^{-1} + \xi)\overline{\mathbf{m}} = \frac{1}{\rho}\begin{bmatrix} X & Y & Z \end{bmatrix}^T$$

$$\mathbf{m}^*_{\mathbf{m}} = (\eta^{*-1} + \xi)\overline{\mathbf{m}}^* = \frac{1}{\rho^*}\begin{bmatrix} X^* & Y^* & Z^* \end{bmatrix}^T$$

$$\overline{\mathbf{m}} = [x\ y\ \frac{1}{1+\xi\eta}]^T$$

$$\overline{\mathbf{m}^*} = [x^*\ y^*\ \frac{1}{1+\xi\eta^*}]^T$$

Homogeneous coordinates:

$$\underline{\mathbf{m}} = [X\ Y\ Z\ H]^T \text{ and } \underline{\mathbf{m}}^* = [X^*\ Y^*\ Z^*\ H^*]^T$$

$$\rho(\eta^{-1} + \xi)\overline{\mathbf{m}} = \begin{bmatrix} \mathbf{I_3} & 0 \end{bmatrix}\underline{\mathbf{m}} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \end{bmatrix}\underline{\mathbf{m}}^*$$

The distance from the world point m to the plane (π)

$$d(m,\pi) = \pi^{*T} \cdot \underline{\mathbf{m}}^* = [\mathbf{n}^{*\mathbf{T}} \ -d^*][X^*\ Y^*\ Z^*\ H^*]^T = \rho^*(\eta^{*-1} + \xi)\mathbf{n}^{*T}\overline{\mathbf{m}}^* - d^*H^*$$

Partial euclidian reconstruction Unified model - Case of points

$$H^* = \frac{\rho^*(\eta^{*-1}+\xi)}{d^*}\mathbf{n}^{*T}\overline{\mathbf{m}}^* - \frac{d(m,\pi)}{d^*}$$

$$\rho^*(\eta^{*-1}+\xi)\overline{\mathbf{m}}^* = \begin{bmatrix} X^* & Y^* & Z^* \end{bmatrix}^T$$

$$\underline{\mathbf{m}}^* = \rho^*(\eta^{*-1}+\xi)\mathbf{A}^*_\pi\overline{\mathbf{m}}^* + \mathbf{b}^*_\pi$$

where

$$\begin{cases} \mathbf{A}^*_\pi = \begin{bmatrix} \mathbf{I_3} & \frac{\mathbf{n}^*}{d^*} \end{bmatrix}^T \\ \mathbf{b}^*_\pi = \begin{bmatrix} \mathbf{0_{1\times 3}} & -\frac{d(m,\pi)}{d^*} \end{bmatrix} \end{cases}$$

$$\rho(\eta^{-1}+\xi)\overline{\mathbf{m}} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \end{bmatrix}\underline{\mathbf{m}}^*$$

$$\rho(\eta^{-1}+\xi)\overline{\mathbf{m}} = \rho^*(\eta^{*-1}+\xi)\mathbf{H}_\pi\overline{\mathbf{m}}^* + \alpha\mathbf{t}$$

with $\mathbf{H}_\pi = \mathbf{R} + \frac{\mathbf{t}}{d^*}\mathbf{n}^{*T}$ and $\alpha = -\frac{d(m,\pi)}{d^*}$

Euclidian Homography

Partial euclidian reconstruction Unified model - Case of points

$m \in \pi$

$$\overline{\mathbf{m}} = \beta_{x,x^*} \mathbf{H}_\pi \overline{\mathbf{m}}^*$$

where $\beta_{x,x^*} = \frac{\eta^{*-1}+\xi}{\eta^{-1}+\xi} \frac{\rho^*}{\rho}$

Linear form

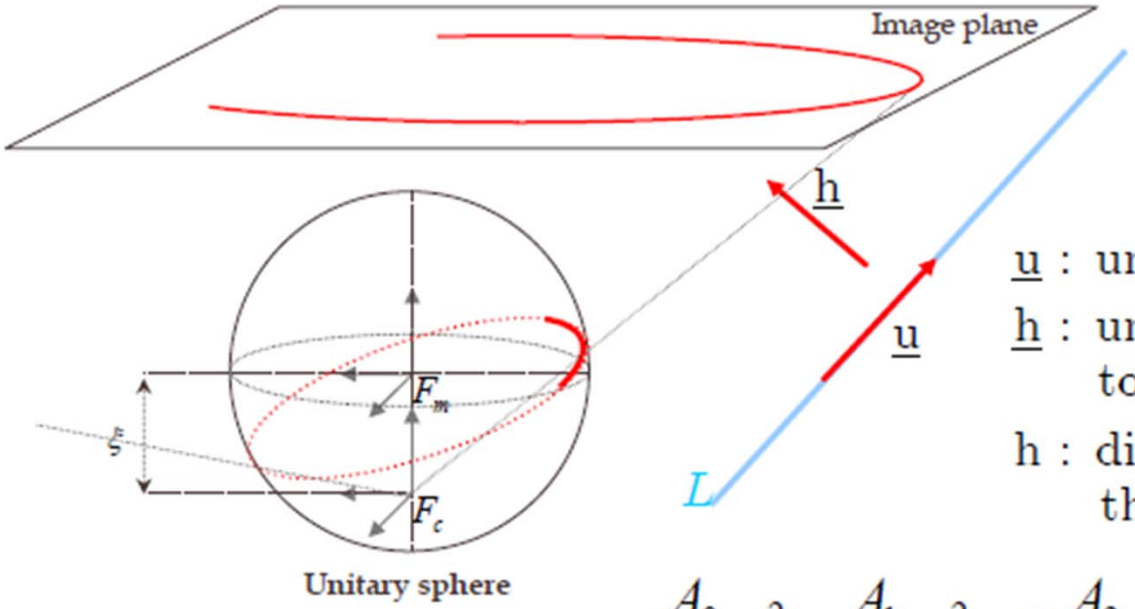
$$\overline{\mathbf{m}} \times \mathbf{H}_\pi \overline{\mathbf{m}}^* = \mathbf{0} \rightarrow \mathbf{H}_\pi \rightarrow \mathbf{R} \text{ and } \mathbf{t}_{d^*} = \frac{\mathbf{t}}{d^*}$$

$$\sigma = \frac{\rho}{\rho^*} = (1 + \mathbf{n}^{*T} \mathbf{R}^T \mathbf{t}_{d^*}) \frac{(\eta^{*-1}+\xi) n^{*T} \overline{\mathbf{m}}^*}{(\eta^{-1}+\xi) n^{*T} \mathbf{R}^T \overline{\mathbf{m}}}$$

Partial euclidian reconstruction

Unified model - Case of lines

Similar results are obtained from lines *(using polar lines)*



Plücker coordinates

$$L = (\underline{u}, \underline{h}, h)^T$$

$\underline{u}$: unitary vector along the line **L**

$\underline{h}$: unitary vector orthogonal to the interpretation plane

h : distance between the origin of the frame and the line

$$\frac{A_0}{A_5}x^2 + \frac{A_1}{A_5}y^2 + 2\frac{A_2}{A_5}xy + 2\frac{A_3}{A_5}x + 2\frac{A_4}{A_5}y + 1 = 0$$

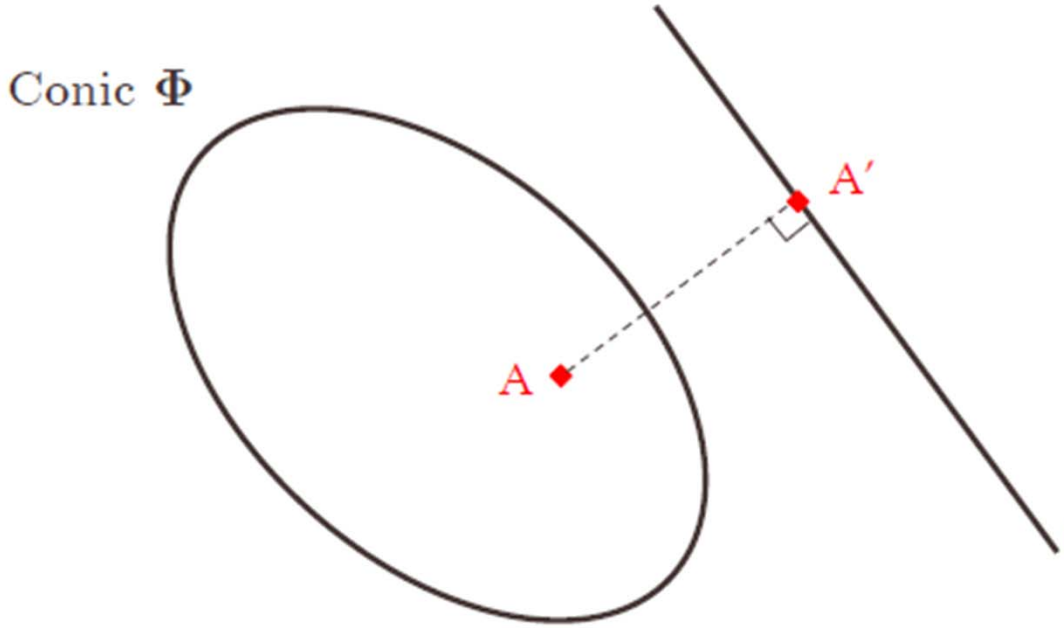
$$B_i(\underline{h}, \xi, K_M) = \frac{A_i}{A_5}$$

Partial euclidian reconstruction

Unified model - Case of lines

Similar results are obtained from lines *(using polar lines)*

Polar lines



ϕ a 2D conic curve

A a point in the definition plane of ϕ

l_i : the corresponding polar line

$l_i \propto \Phi A$

Partial euclidian reconstruction

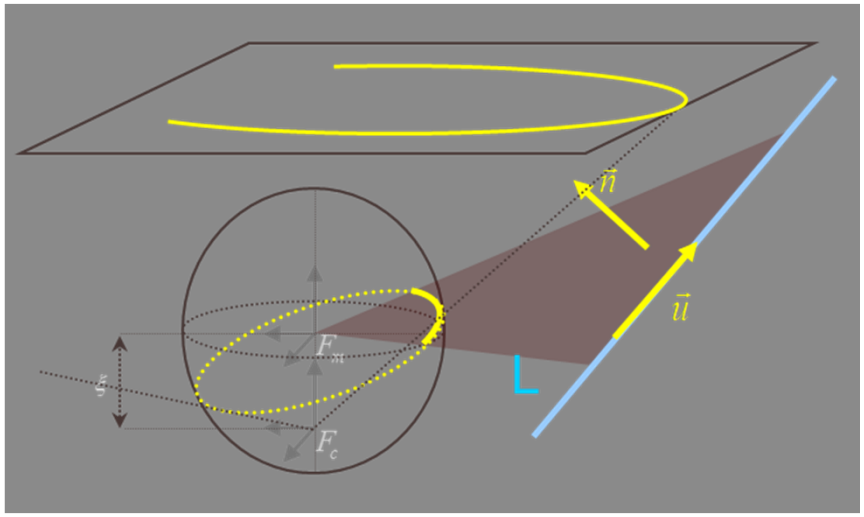
Unified model - Case of lines

Similar results are obtained from lines *(using polar lines)*

Polar lines

$$\Omega = \begin{pmatrix} h_x^2 - \xi^2\left(1-h_y^2\right) & h_x h_y\left(1-\xi^2\right) & h_x h_z \\ h_x h_y\left(1-\xi^2\right) & h_y^2 - \xi^2\left(1-h_x^2\right) & h_y h_z \\ h_x h_z & h_y h_z & h_z^2 \end{pmatrix}$$

$$\underline{\mathbf{x}}_\mathbf{i}{}^T \underbrace{\mathbf{K}^{-T}\Omega\mathbf{K}^{-1}}_{\mathbf{\Omega_i}} \underline{\mathbf{x}}_\mathbf{i} = 0$$



The polar line of the optical center with respect to the conic $\mathbf{\Omega_i}$ is given by:

$$\mathbf{l_i} \propto \mathbf{\Omega_i O_i}$$

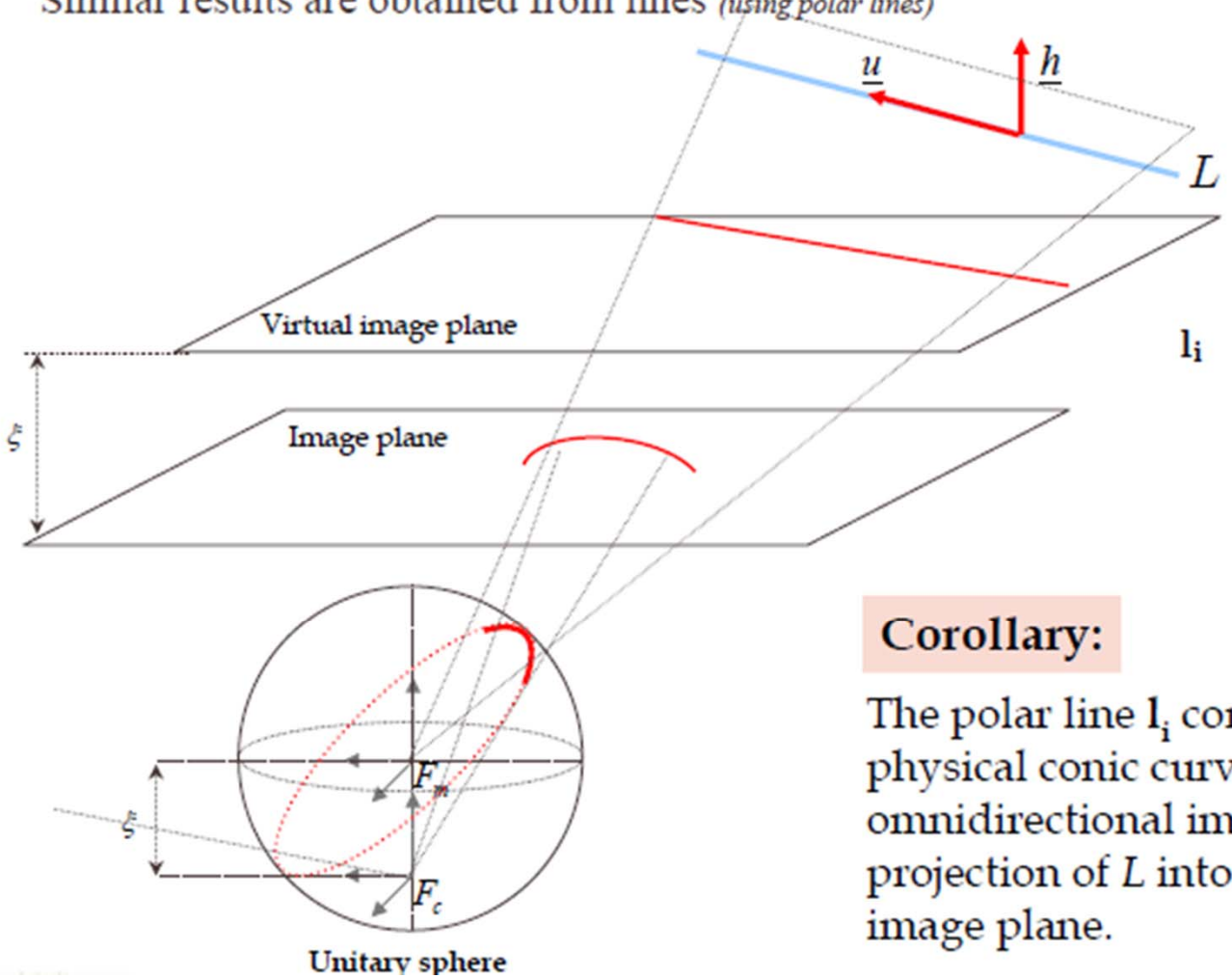
principal point

$$\mathbf{O_i} = [u_0 \;\; v_0 \;\; 1]^T = \mathbf{K}[0\;0\;1]^T$$

Partial euclidian reconstruction

Unified model - Case of lines

Similar results are obtained from lines *(using polar lines)*



$$\mathbf{l_i} \propto \mathbf{K}^{-T}\boldsymbol{\Omega}\mathbf{K}^{-1}\mathbf{O_i}$$

$$\propto \mathbf{K}^{-T}\boldsymbol{\Omega}\mathbf{K}^{-1}\mathbf{K}\begin{pmatrix}0\\0\\1\end{pmatrix}$$

$$\propto \mathbf{K}^{-T}\begin{pmatrix}h_x\\h_y\\h_z\end{pmatrix}$$

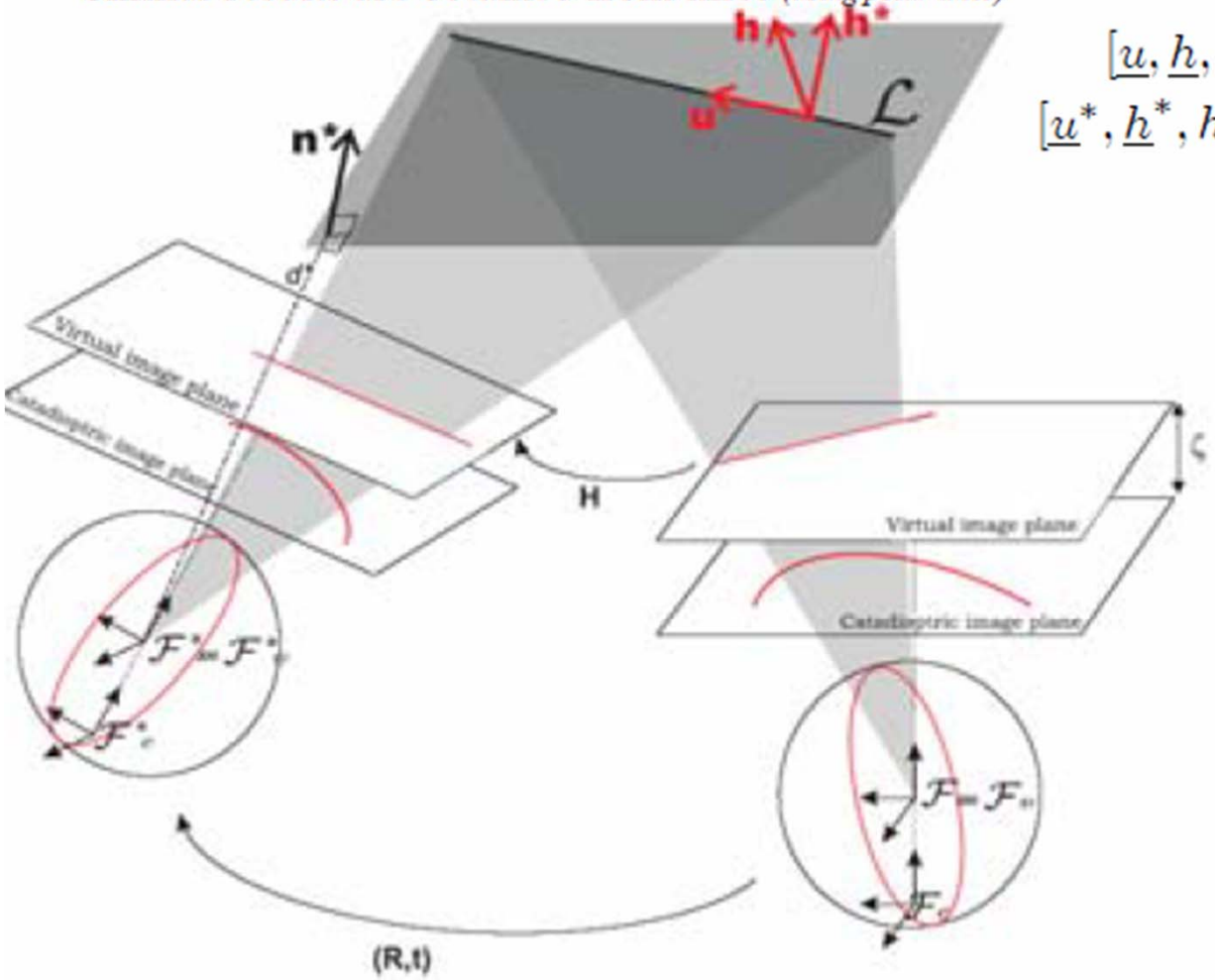
Corollary:

The polar line $\mathbf{l_i}$ computed from the physical conic curve projection of L in the omnidirectional image is the perspective projection of L into the virtual camera image plane.

Partial euclidian reconstruction

Unified model - Case of lines

Similar results are obtained from lines *(using polar lines)*



$[\underline{u}, \underline{h}, h]$ **L** coordinates in $\mathcal{F}_m$

$[\underline{u}^*, \underline{h}^*, h^*]$ **L** coordinates in $\mathcal{F}_m^*$

Reference plane π in F_m^*

$$\pi^{*T} = [\mathbf{n}^{*\mathbf{T}} \quad -d^*]$$

Partial euclidian reconstruction

Unified model - Case of lines

Similar results are obtained from lines *(using polar lines)*

$\mathcal{X}_1$ and $\mathcal{X}_2$ are two points in the 3D space lying on the line $\mathcal{L}$

$$\underline{\mathbf{h}} \propto \mathbf{H}^{-\top}\underline{\mathbf{h}}^*$$

$$\mathbf{l_i} \propto \mathbf{K}^{-T}\underline{\mathbf{h}} \Rightarrow \mathbf{l_i} \propto \mathbf{K}^{-T}\mathbf{H}^{-T}\underline{\mathbf{h}}^*$$

$$\mathbf{l_i}^* \propto \mathbf{K}^{-T}\underline{\mathbf{h}}^* \Rightarrow \mathbf{l_i} \propto \mathbf{K}^{-T}\mathbf{H}^{-T}\mathbf{K}^T\mathbf{l_i}^*$$

$$\mathbf{l_i} \propto \mathbf{G}^{-\top}\mathbf{l_i^*}$$

$$\mathbf{G} = \mathbf{K}\mathbf{H}\mathbf{K}^{-1}$$

Linear form

$$\mathbf{l_i} \times \mathbf{G}^{-\top}\mathbf{l_i^*} = 0$$

with 4 couples (l_i, l_i^*) $\Rightarrow$ $\mathbf{H}$ $\Rightarrow$ $\mathbf{R}$ and $\mathbf{t}_{d^*} = \frac{\mathbf{t}}{d^*}$

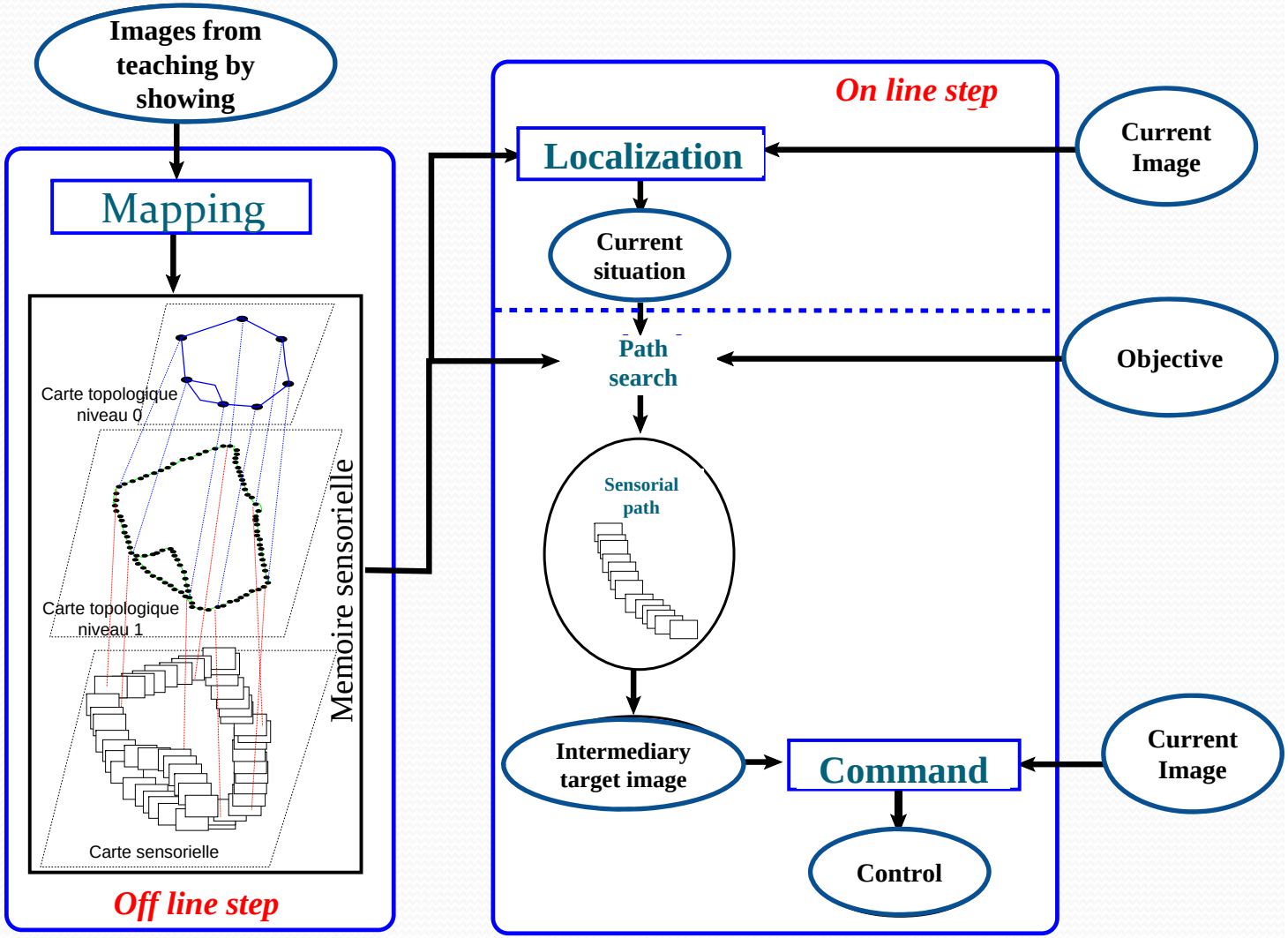
$$\mathbf{H} = \mathbf{R} + \frac{\mathbf{t}}{d^*}\mathbf{n}^{*T}$$

$$r = \frac{h}{h^*} = (1 + \mathbf{t}_d^{*\top}\mathbf{R}^\top\mathbf{n}^*)\frac{\|\mathbf{n}^* \times \mathbf{K}^\top\mathbf{l_i^*}\|}{\|\mathbf{R}\mathbf{n}^* \times \mathbf{K}^\top\mathbf{l_i}\|}$$

Content

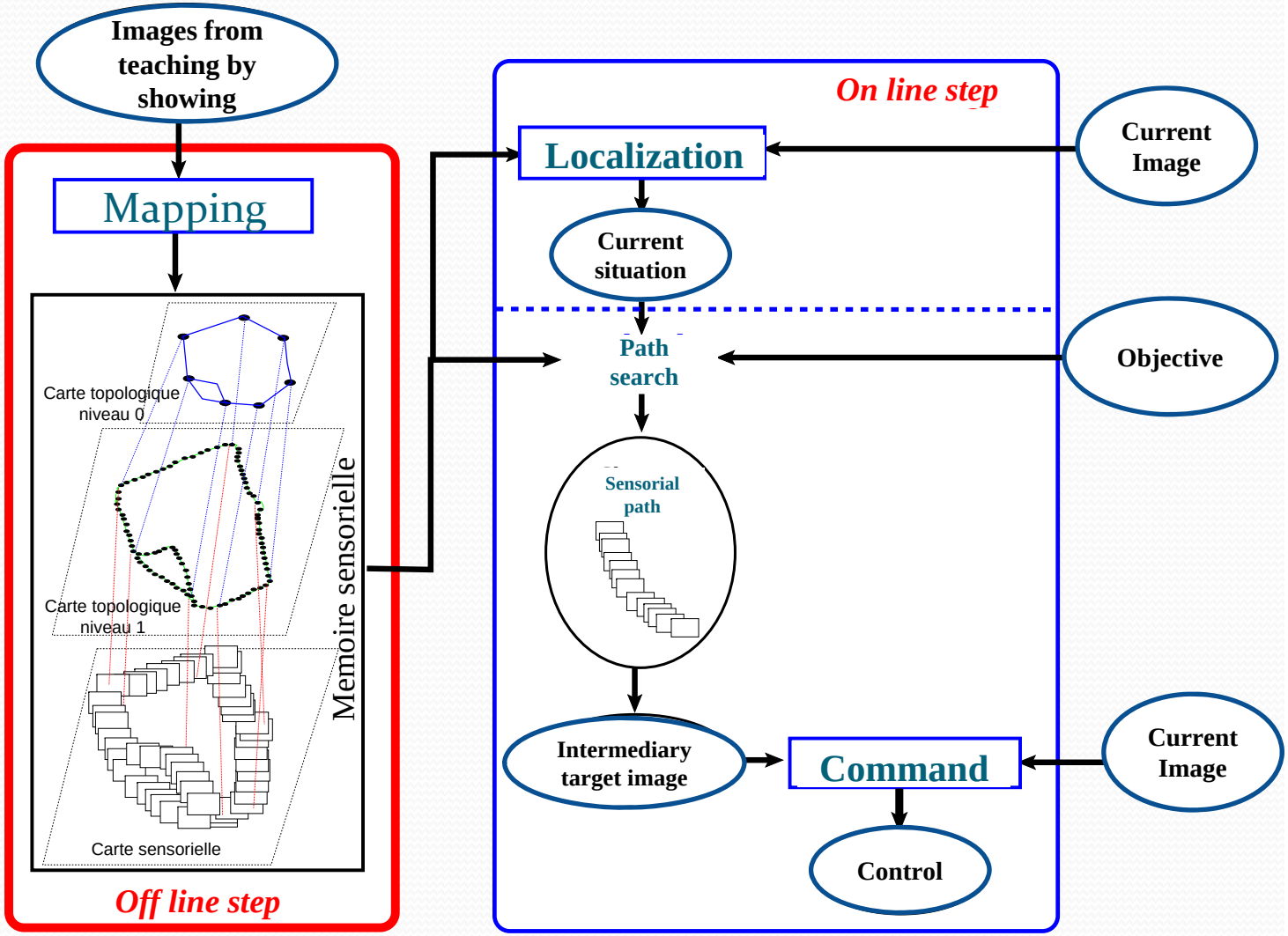
- Introduction
- Generic models for cameras
- Visual geometry for localization
- **Topological navigation system**
- Illustrations
- Conclusion

Global system



Global system

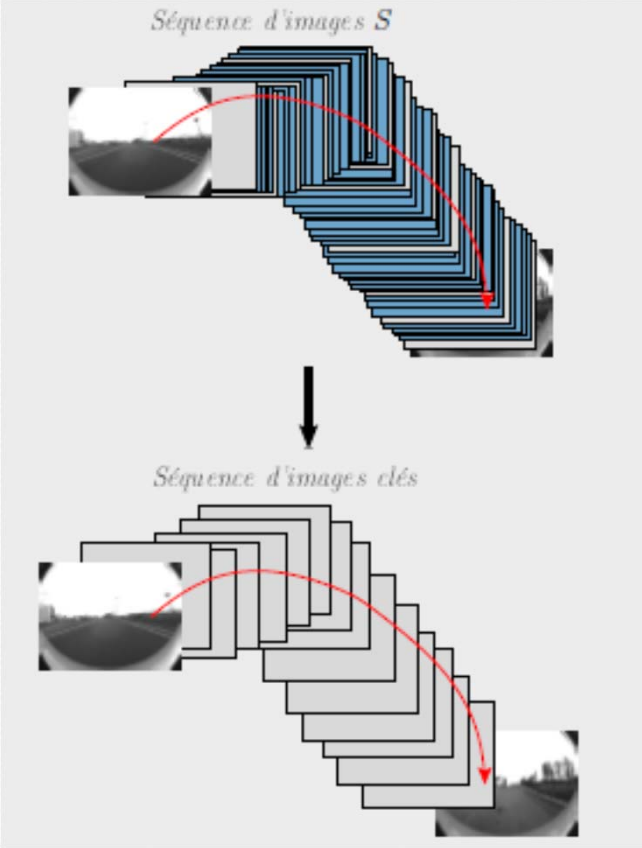
Supervised learning step: building of the sensory memory



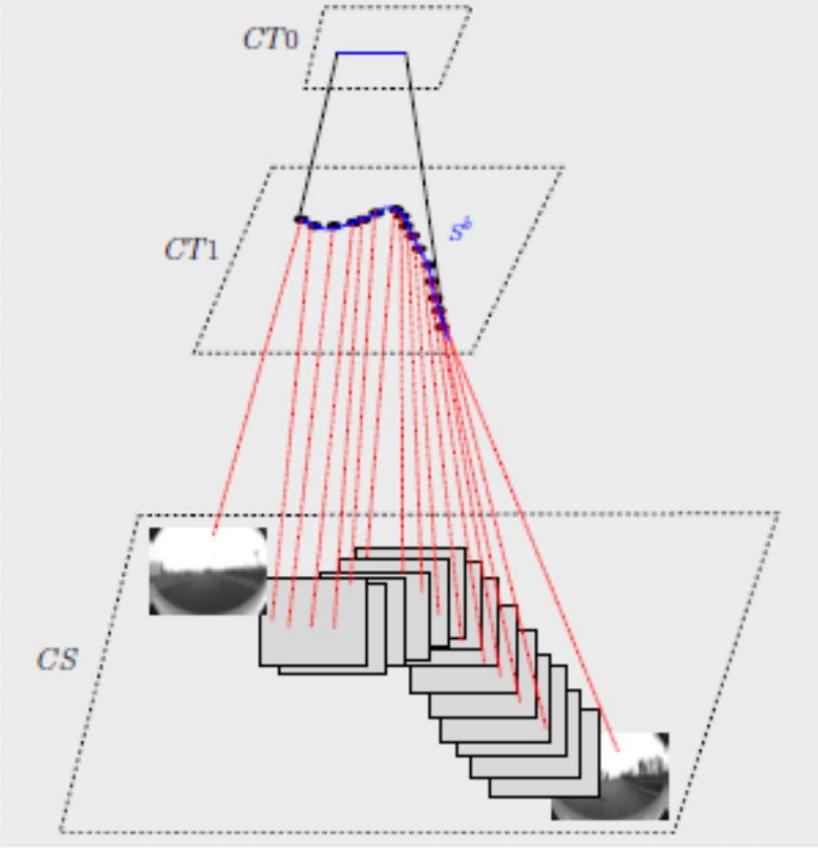
Different modules

Learning step (first arc, first two nodes)

Key images selection

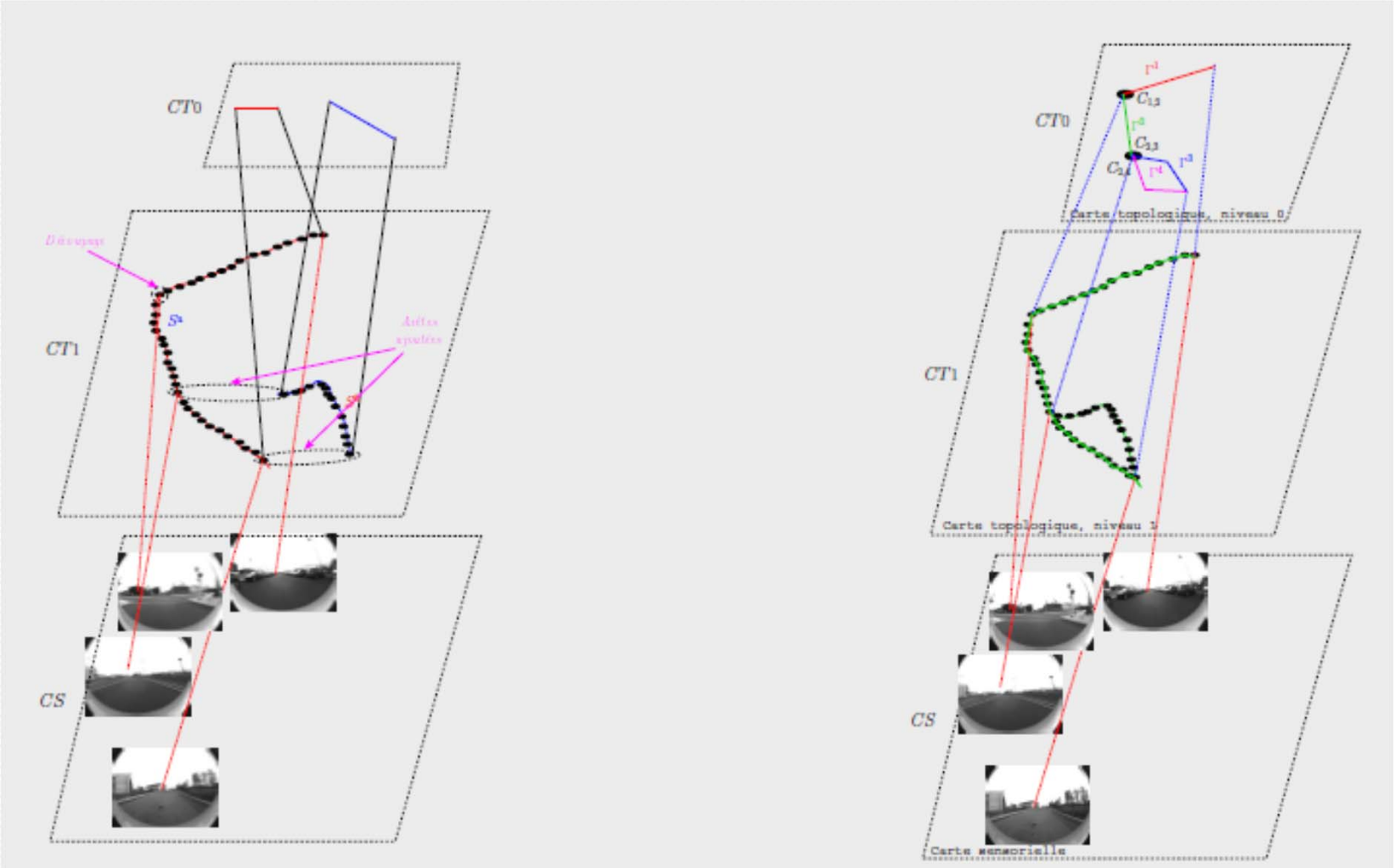


Building the sensory memory



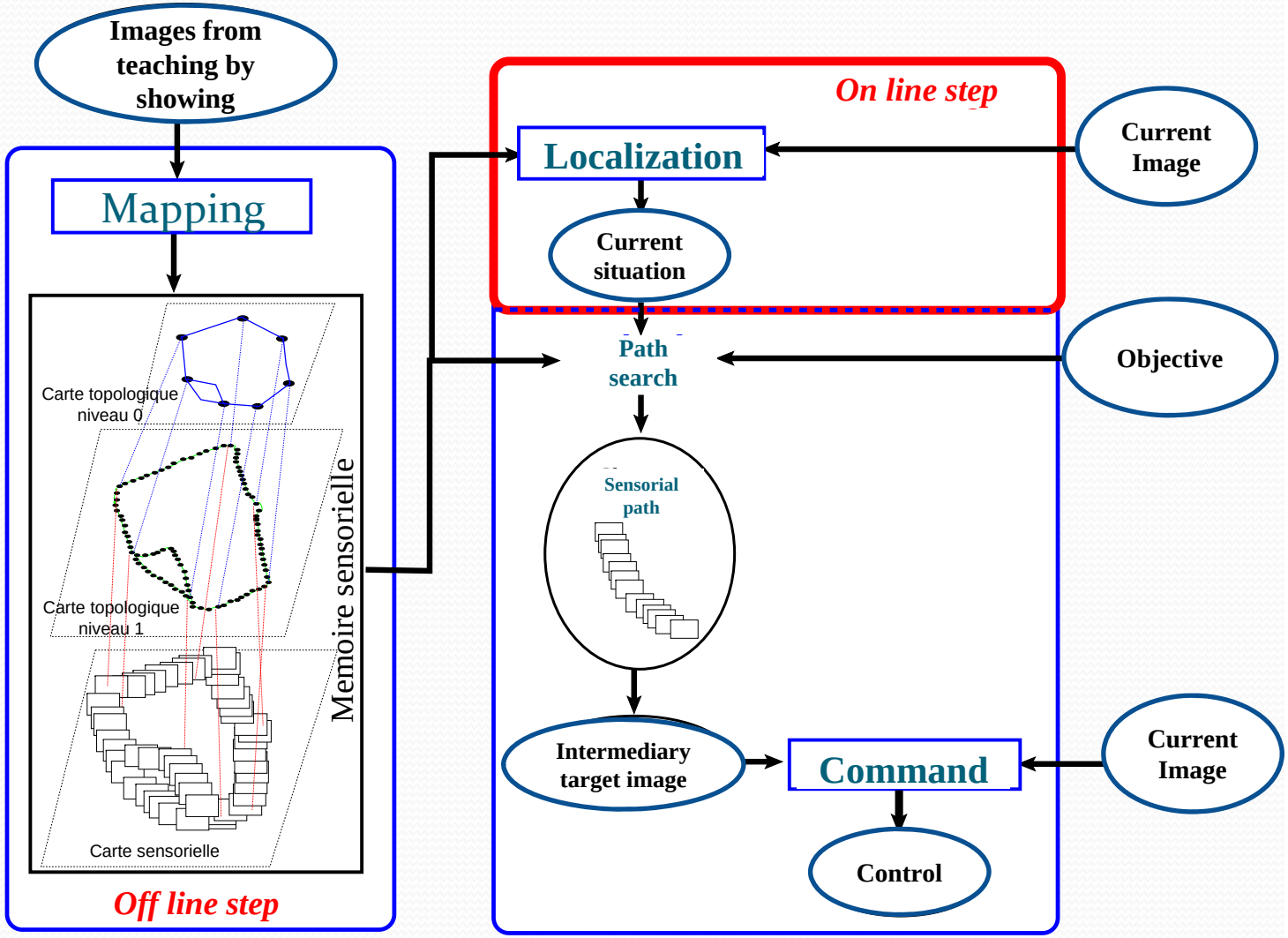
Different modules

Learning step (expanding the processus…)



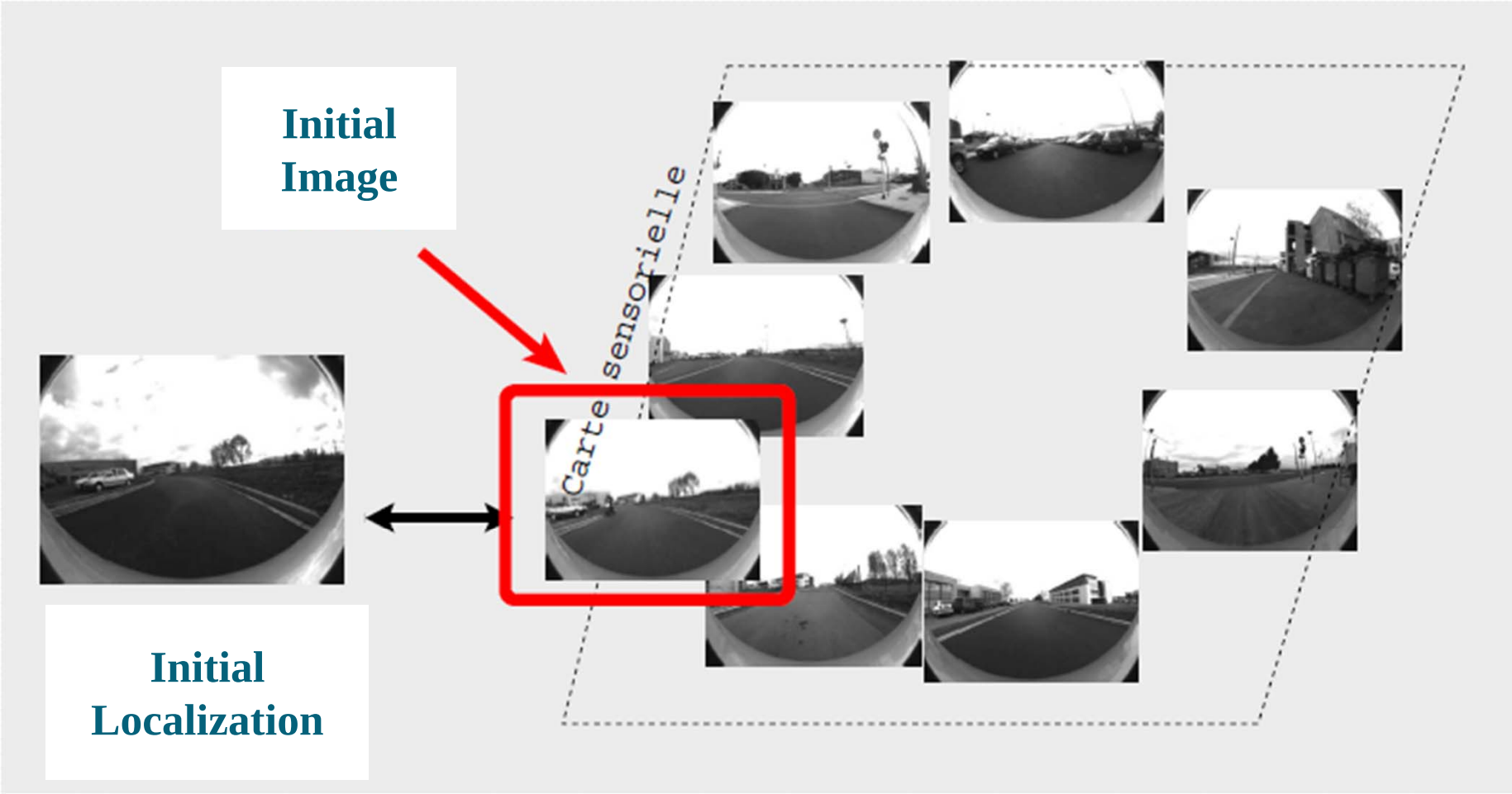
Global system

Initialization: primary localization



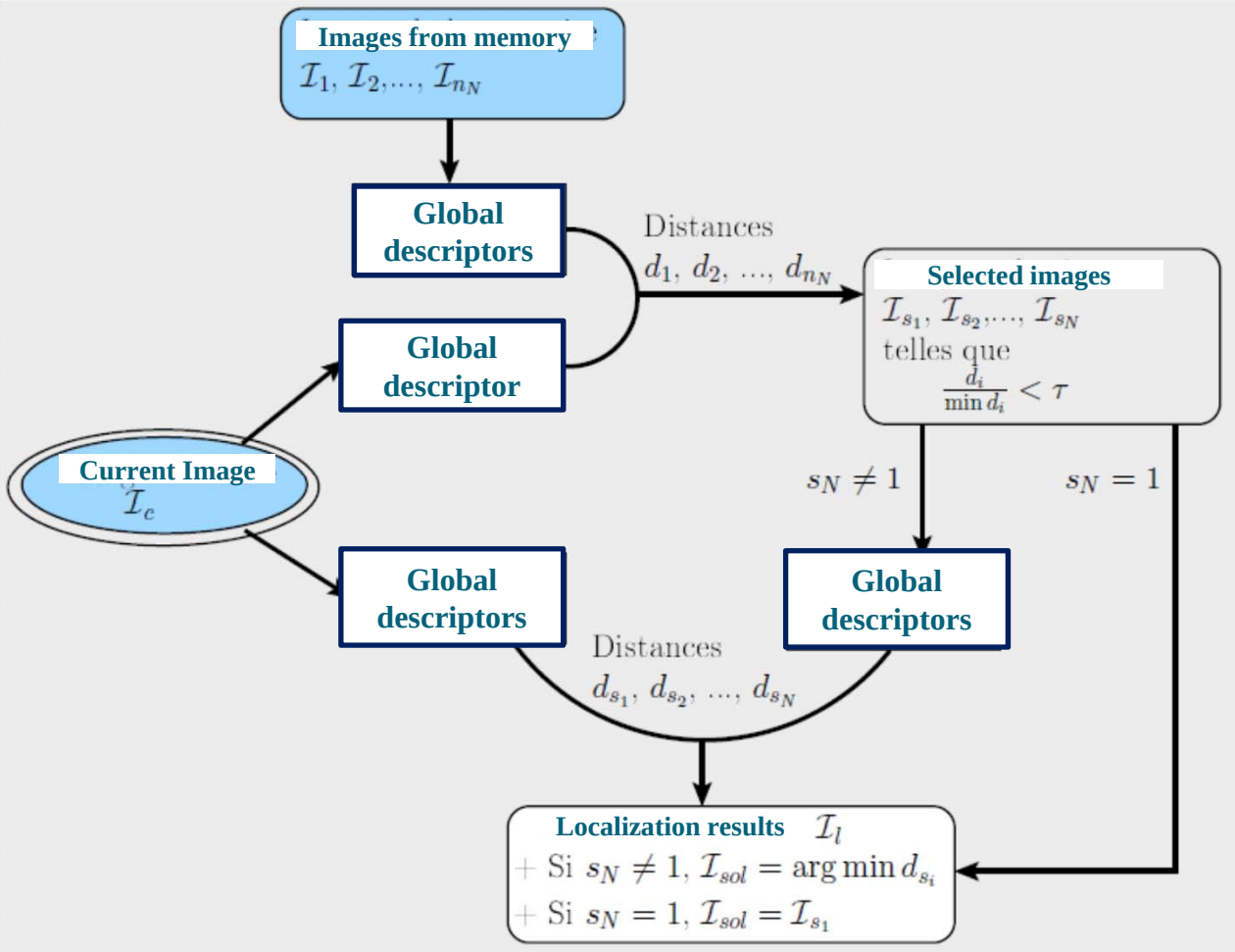
Differents modules — Initial localization

How to find the closest image in the database ?



Differents modules — Initial localization

How to find the closest image in the database ?

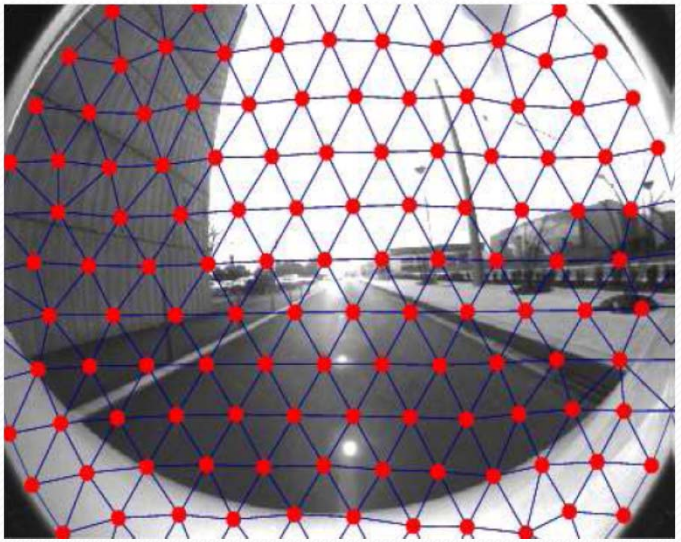


Differents modules — Initial localization

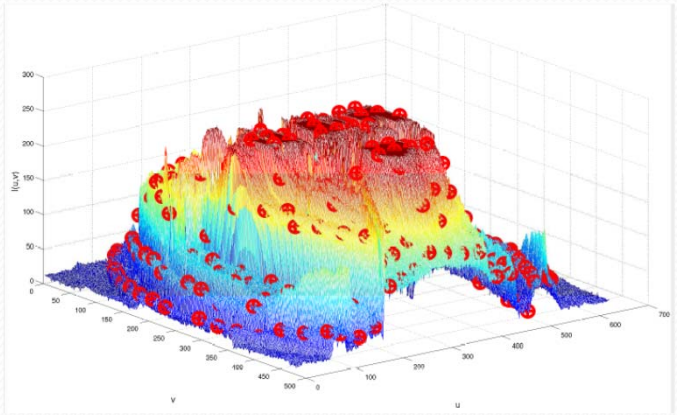
How to find the closest image in the database ?

Use of a regular mesh [PERSSON04]
Cubic interpolation of the image surface

Global descriptor : vector of the interpolated grey level pixels at the position of the control point



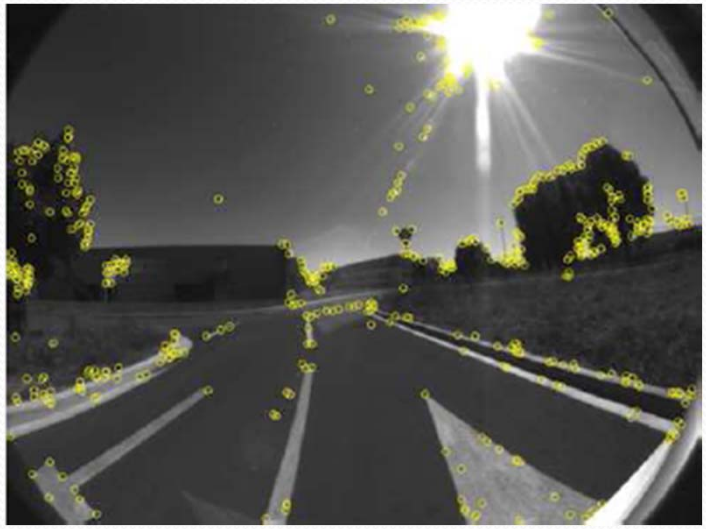
Control point distribution (one example)



How to find the closest image in the database ?

Local descriptor : interest points detected by Harris Stephen detector

Use of patches 11x11

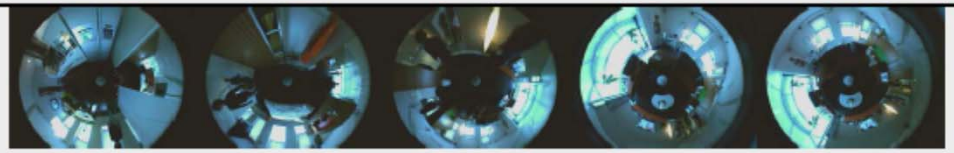


Matching done by using the centered cross correlation score ZNCC

Use of the number of matched point to compute the similarity between 2 images

How to find the closest image in the database ?

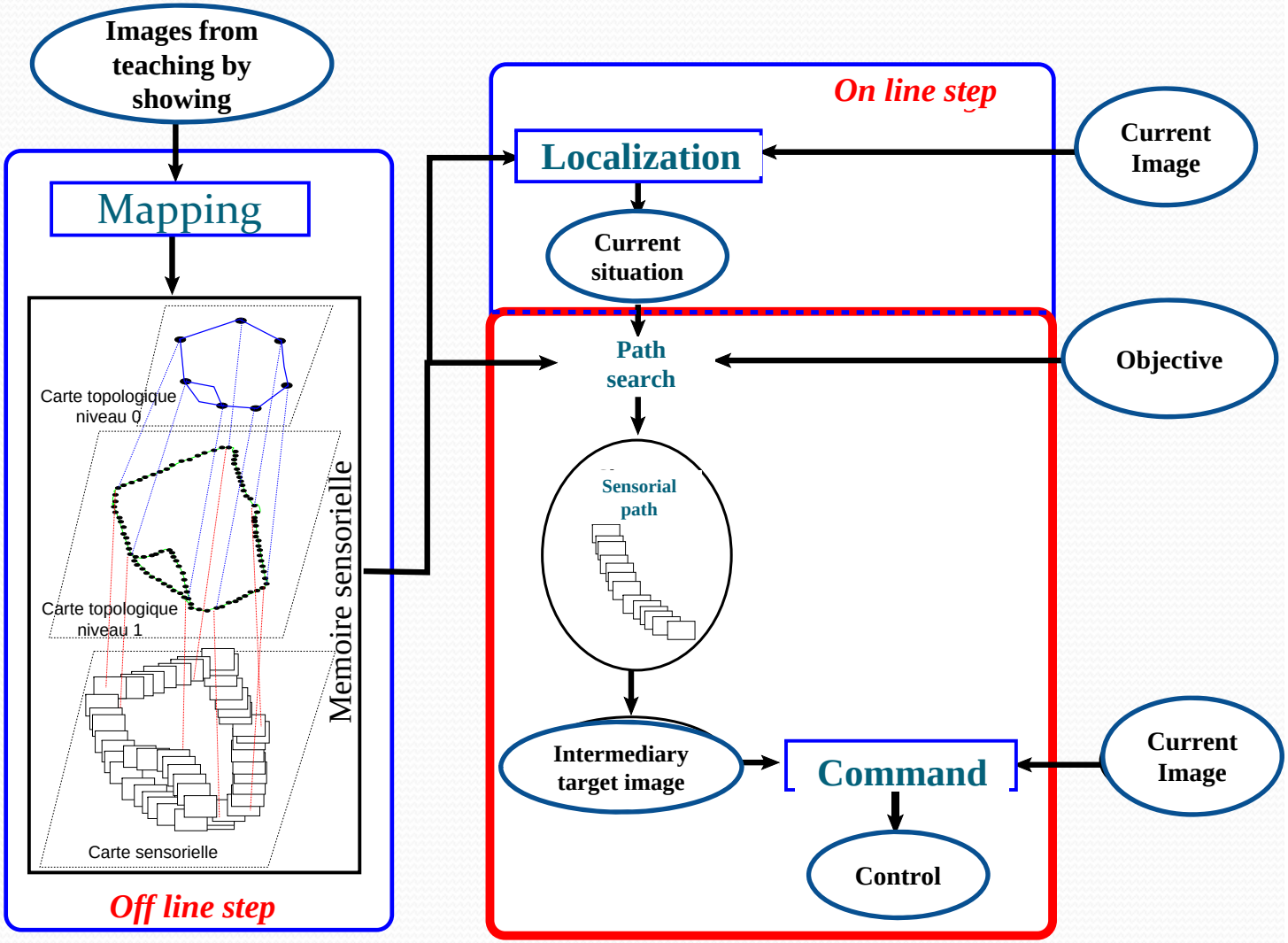
Evaluation of the approach [ICRA08] using three image databases regarding 3 main criteria: Used memory size, percentage of success, computation time for localization

Caméra	Taille image	Nb images	Images
Catadioptrique	1024 × 768	978	
Fisheye 110 deg	384 × 288	188	
Fisheye 190 deg	640 × 480	445	

Several approaches were compared : GONZ [GONZALES02], PHLAC [LINAKER04], CUB, SURF, SIFT, HARRIS, CUBHAR

Global system

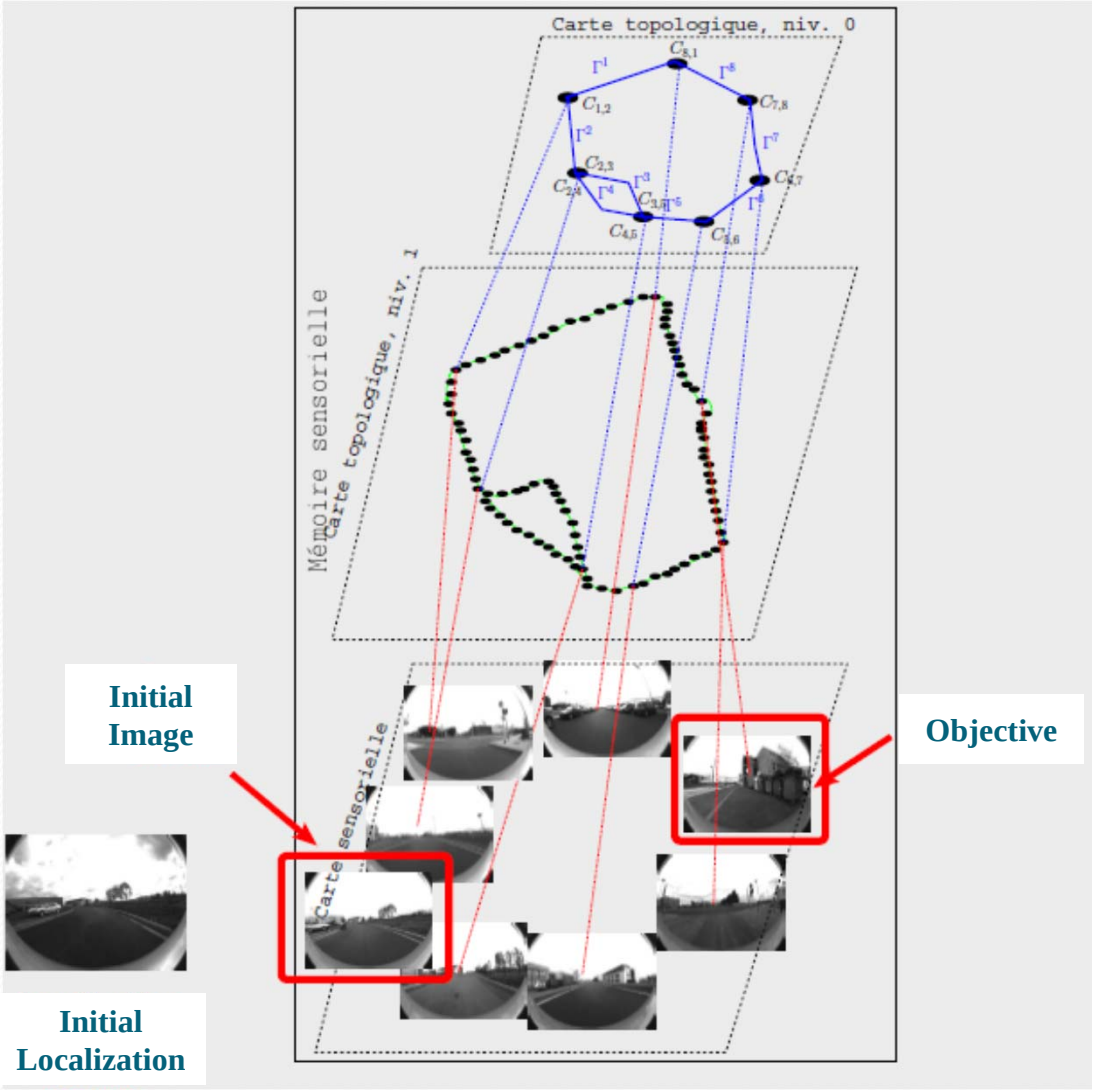
One line navigation and control



Differents modules

Planifying the visual route

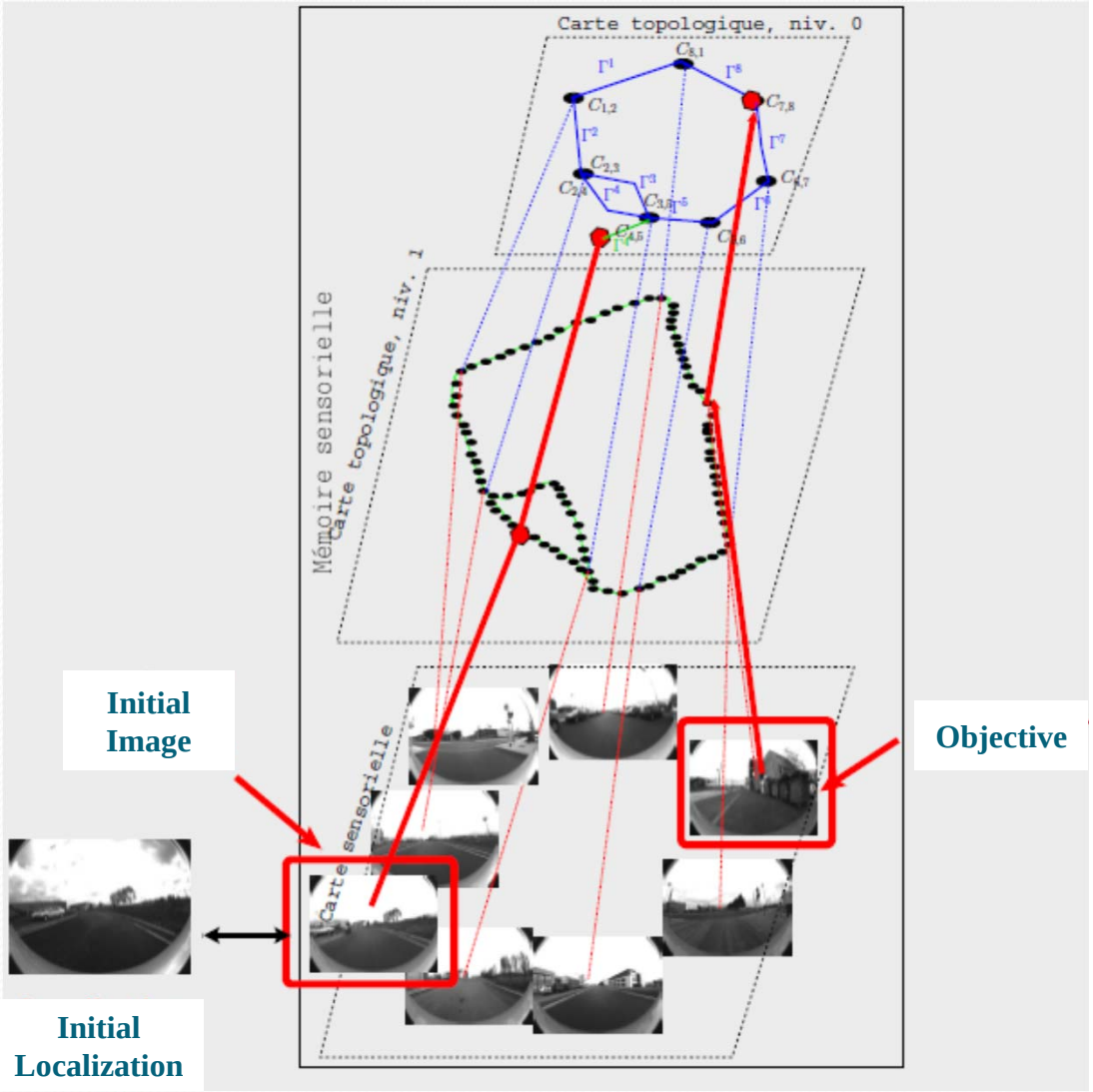
How to find the optimal path ?



Differents modules

Planifying the visual route

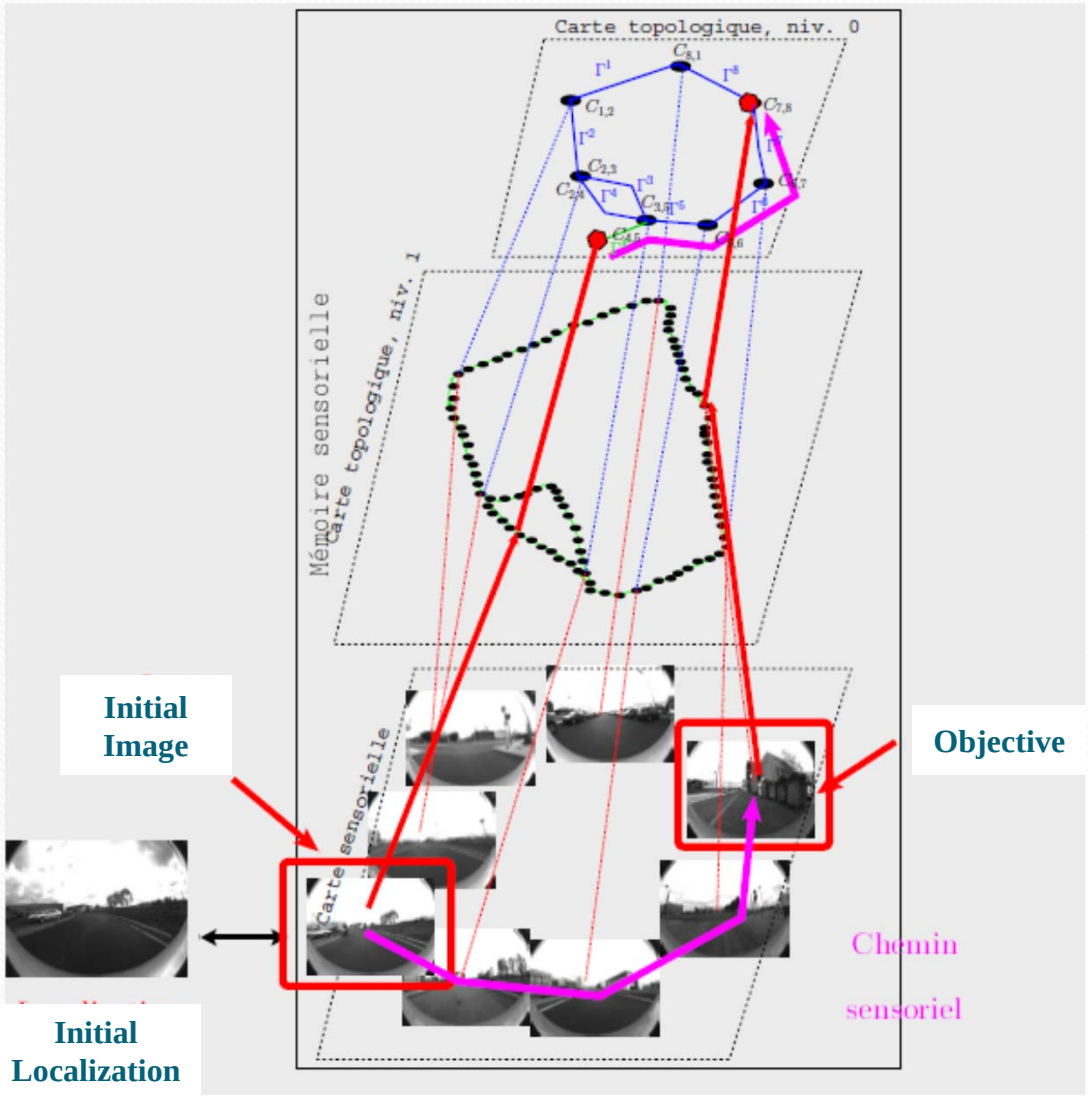
How to find the optimal path ?



Differents modules

Planifying the visual route

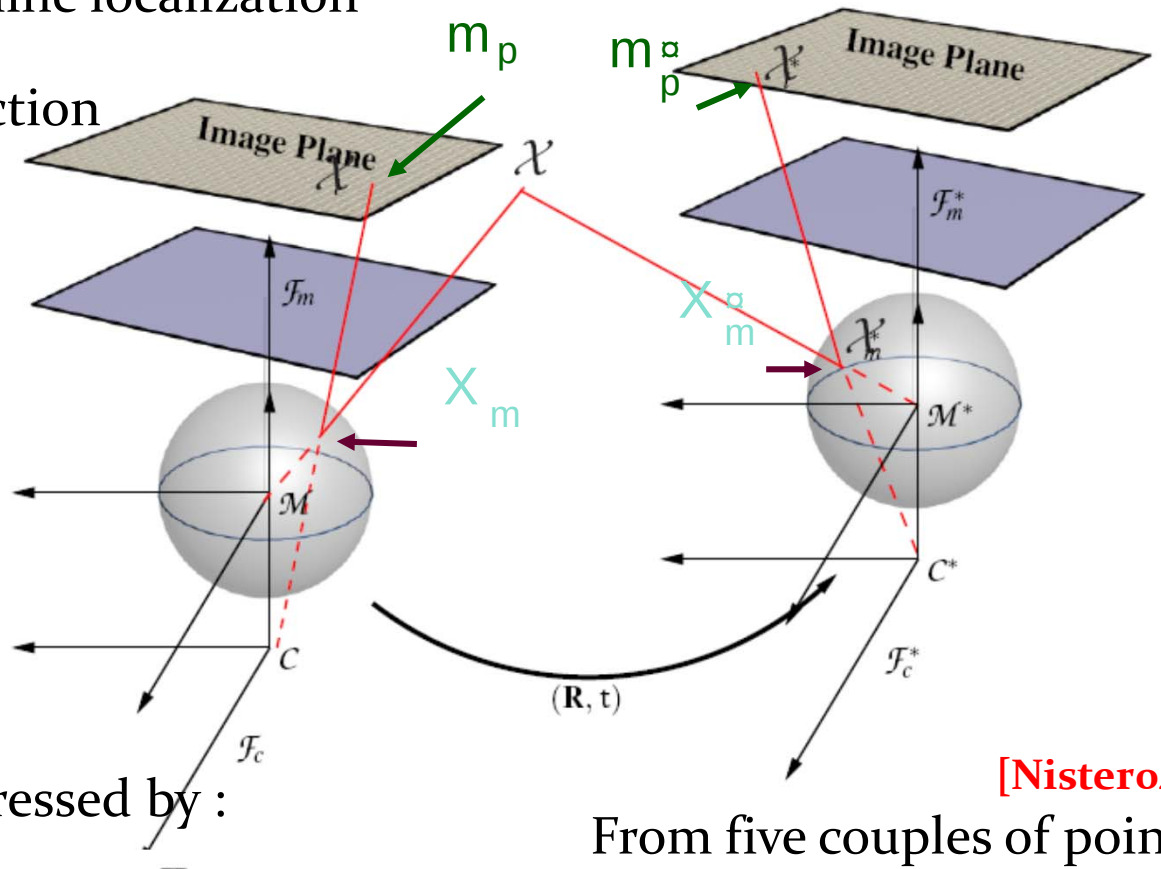
How to find the optimal path ?



Differents modules

One line localization

Scaled euclidian reconstruction



[Nistero4]

Epipolar constraint can be expressed by :

$$X_m{}^T \mathbf{R}(t \times X_m{}^{*T}) = X_m{}^T \mathbf{R}\,[t]_\times\, X_m{}^{*T} = 0$$

From five couples of points E can be estimated

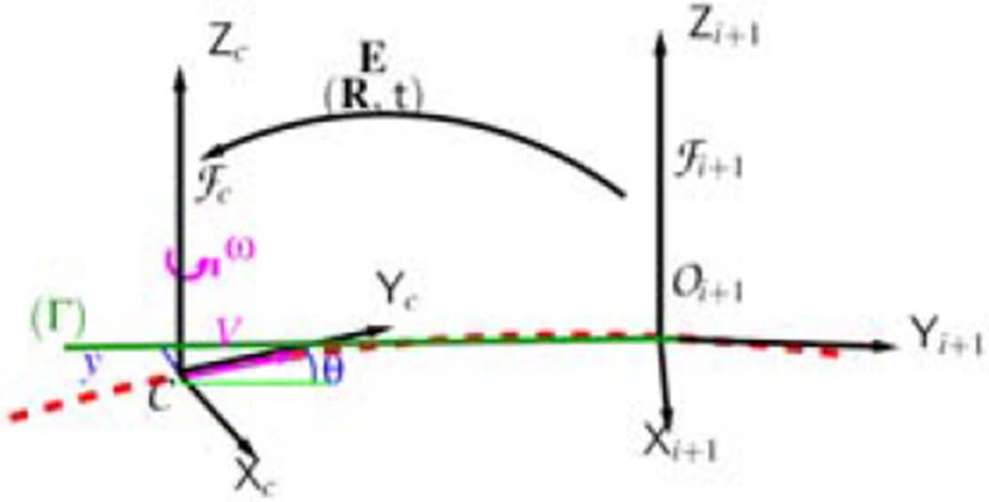
For pinhole model $X_m{}^T \mathbf{E} X_m{}^{*T} = 0$

Outliers are rejected by using RANSAC

Differents modules — State estimation and control (one example)

State model of the mobile robot

$$(s, y, \theta)$$



★ $\mathcal{F}_i = (O_i, \mathbf{X_i}, \mathbf{Y_i}, \mathbf{Z_i})$

★ $\mathcal{F}_{i+1} = (O_{i+1}, \mathbf{X_{i+1}}, \mathbf{Y_{i+1}}, \mathbf{Z_{i+1}})$ the frames attached to the robot when $\mathcal{I}_i$

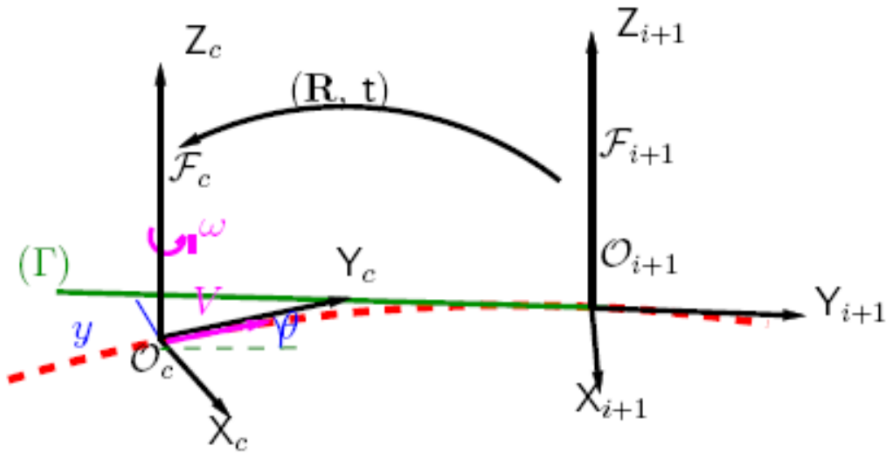
Kinematic model of the mobile robot

$$\begin{cases} \dot{s} = \dfrac{V \cos\theta}{1 - yc(s)} \\ \dot{y} = V \sin\theta \\ \dot{\theta} = \omega - \dfrac{Vc(s)\cos\theta}{1 - yc(s)} \end{cases}$$

⇒ Chained System theory [IAV04]

Differents modules State estimation and control (one example)

Path following approach

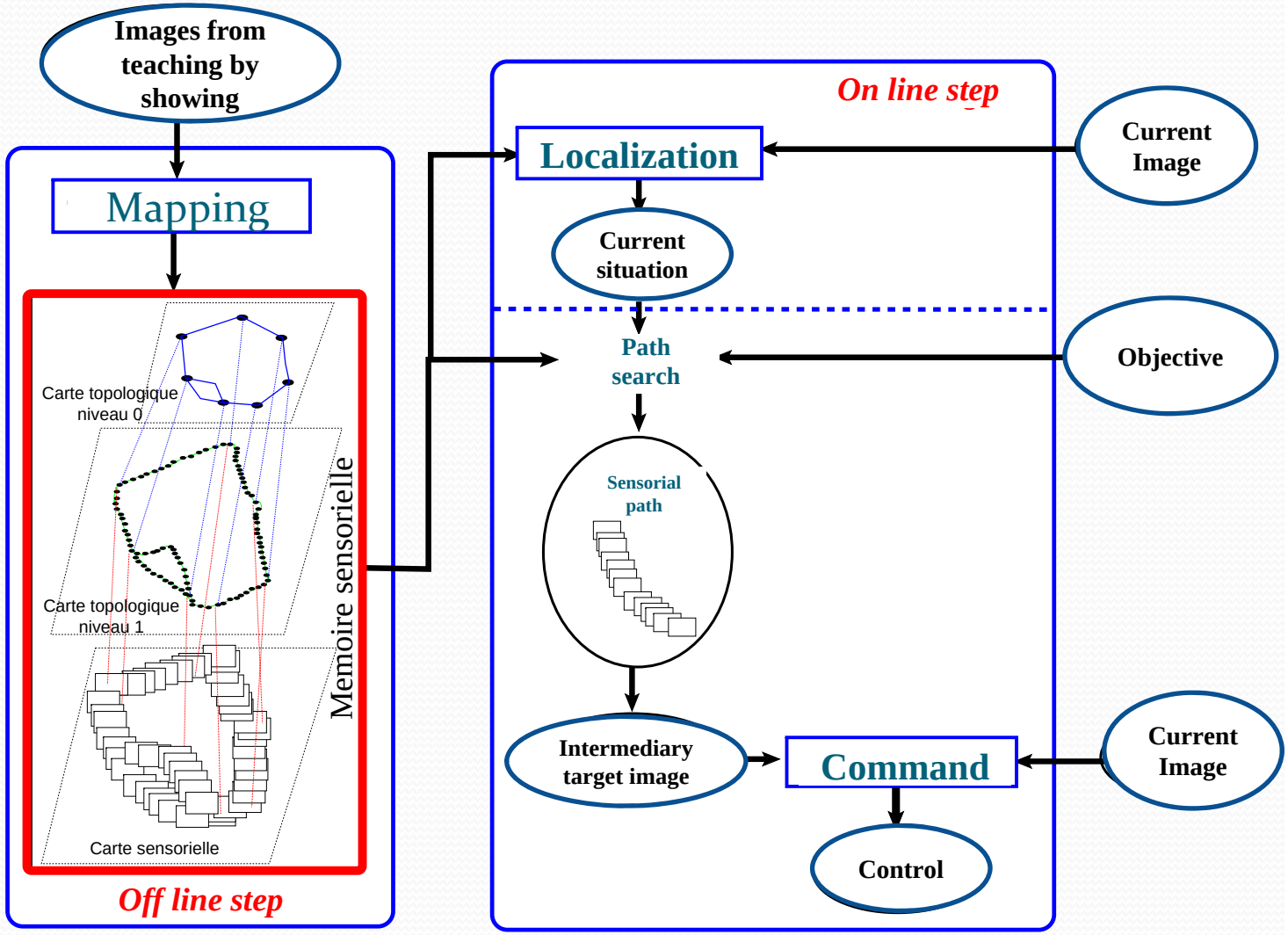


Goal: $y \longrightarrow 0$, $\theta \longrightarrow 0$ before $\mathcal{F}_c$ reaches $\mathcal{F}_{i+1}$

Asymptotically stable guidance control law based on the chain system approach:

$$\delta(y,\theta) = \arctan\left(-l\left[\cos^3\theta\left(-K_d\tan\theta - K_p y\right)\right]\right)$$

Case of vision



Case of vision

Topological map level 0 (CT0)

- ✓High level representation : corridor, routes, ..

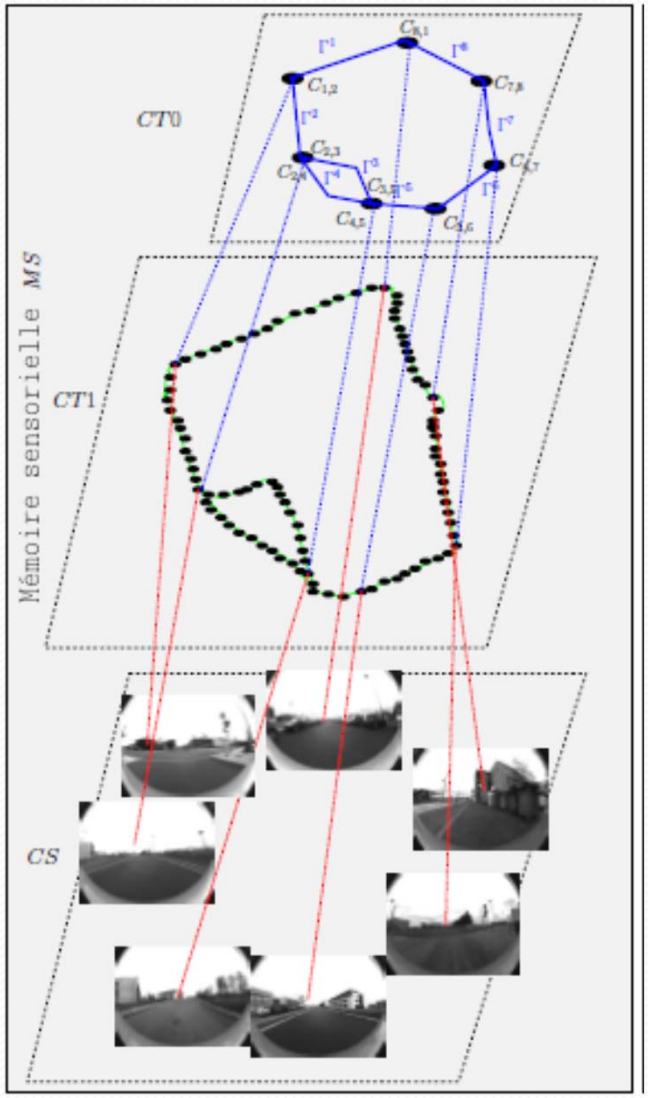
Topological map level 1 (CT1)

- ✓Image network representation
- ✓Navigability
- ✓Commandability
- ✓Plannification

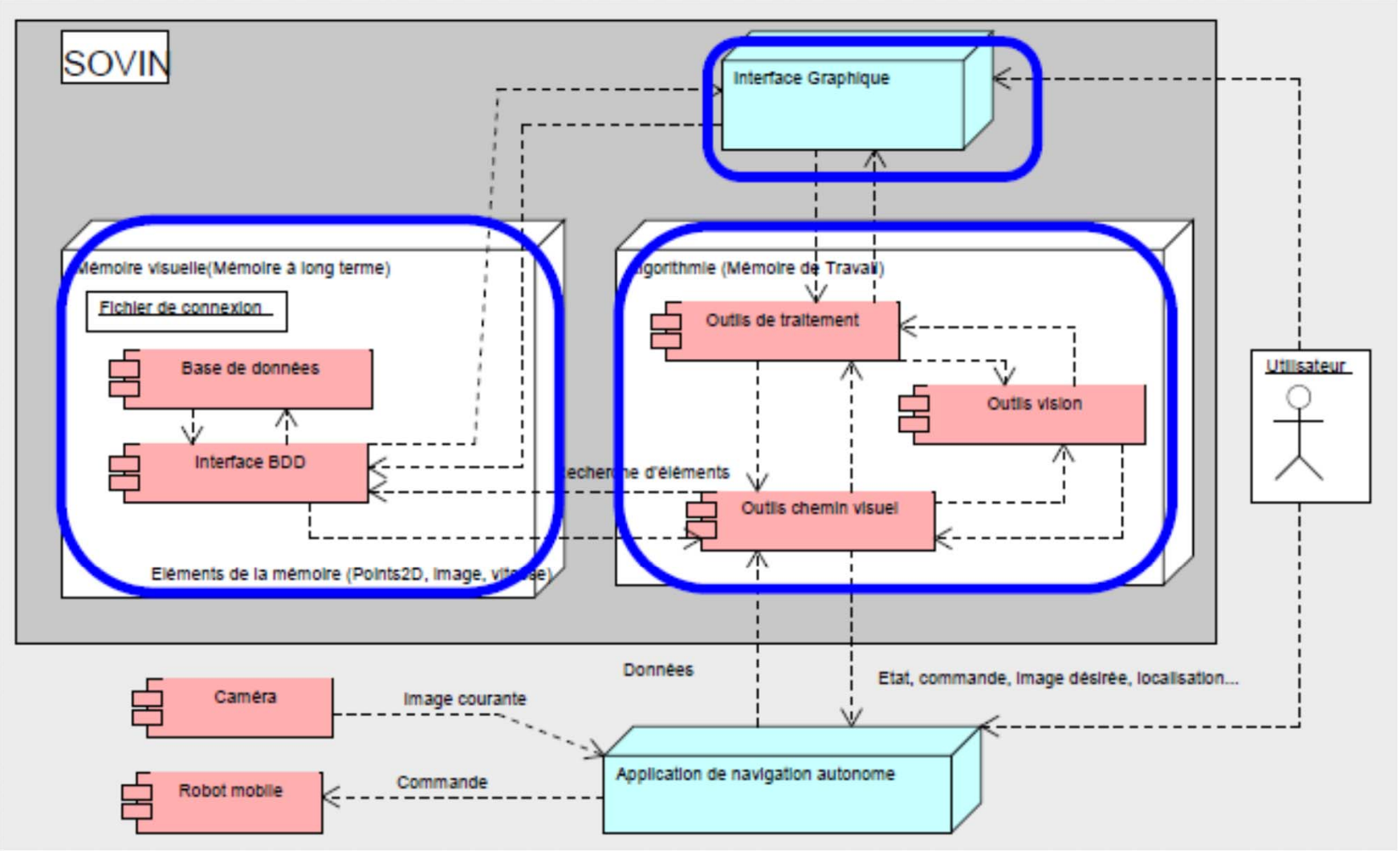
Sensory map (CS)

- ✓Initial localization
- ✓Sensor based control
- ✓Vizualization of the environment
- ✓Human interface

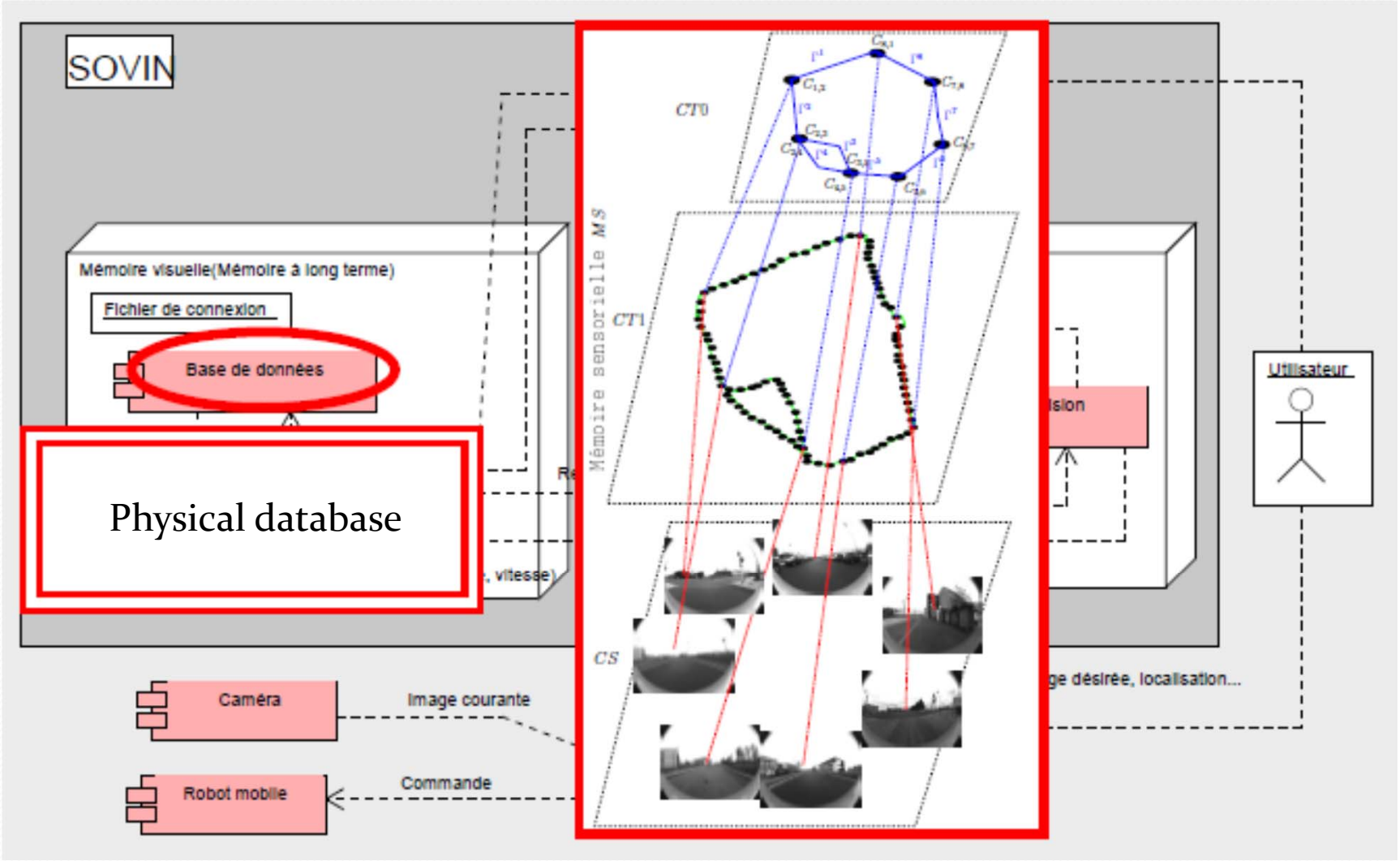
Sensory memory



SOVIN A global software environment for topological navigation in wide spaces

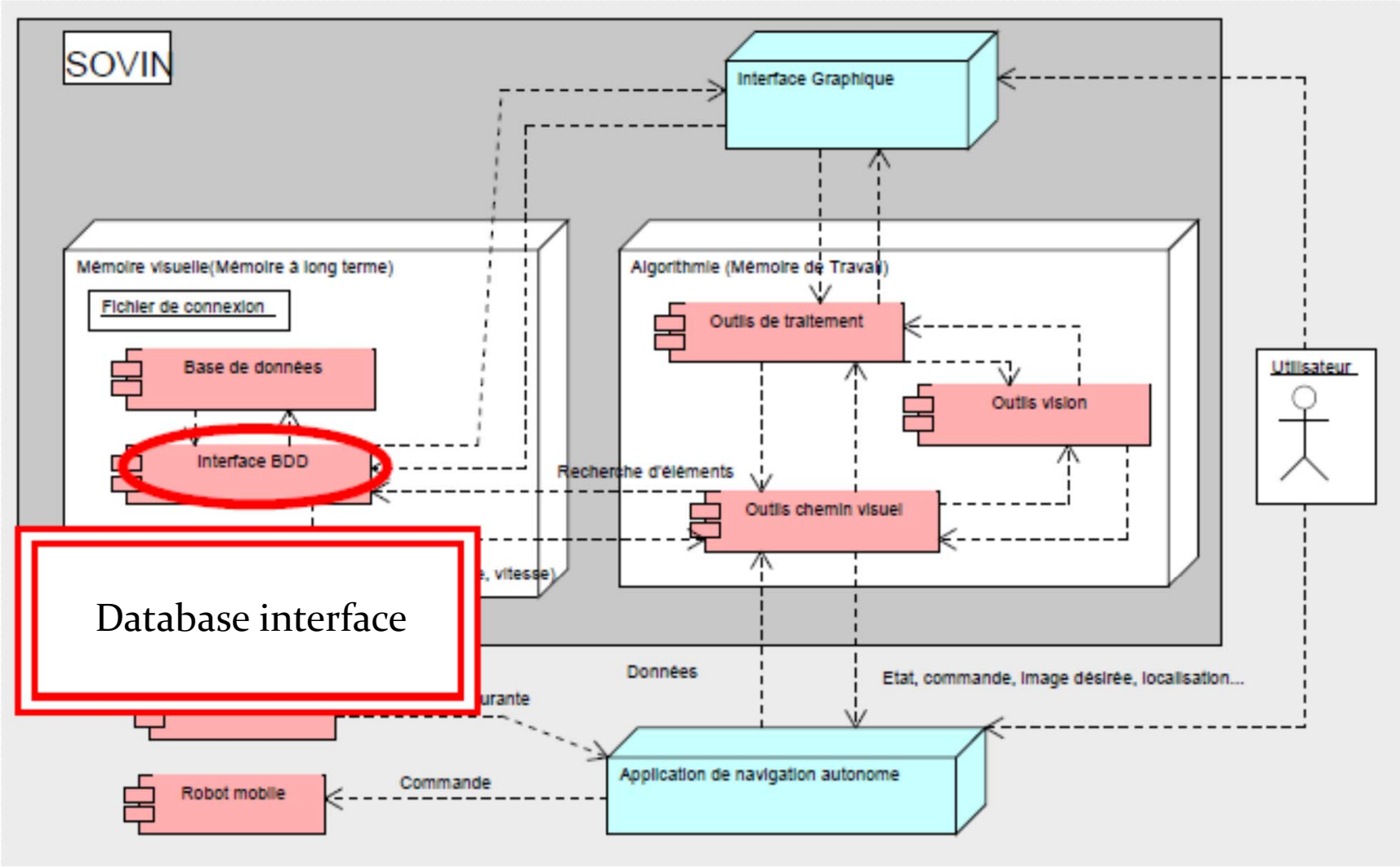


SOVIN A physical database organized in three levels



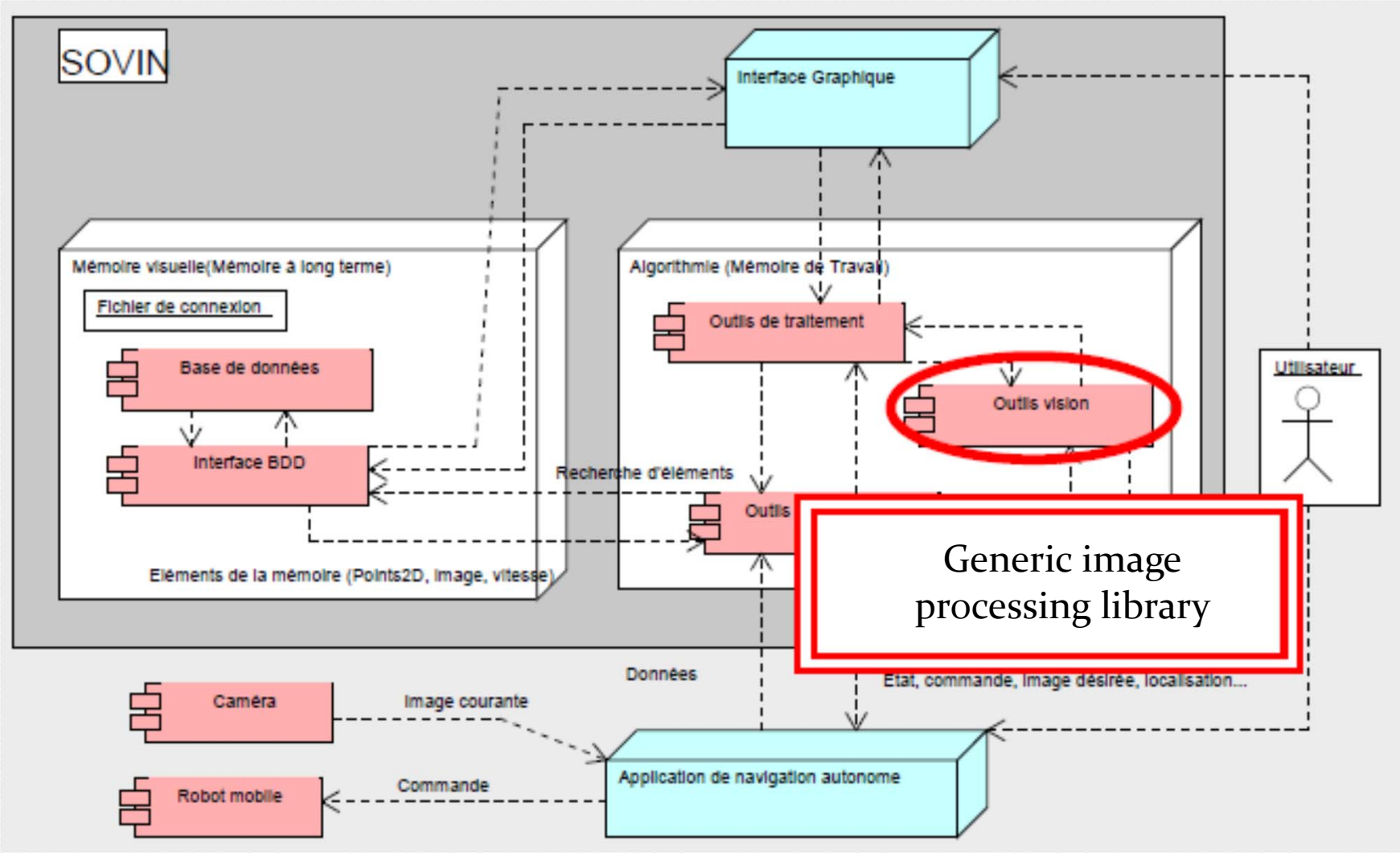
SOVIN An efficient database interface for on line and real time applications

[Courbon 2008]



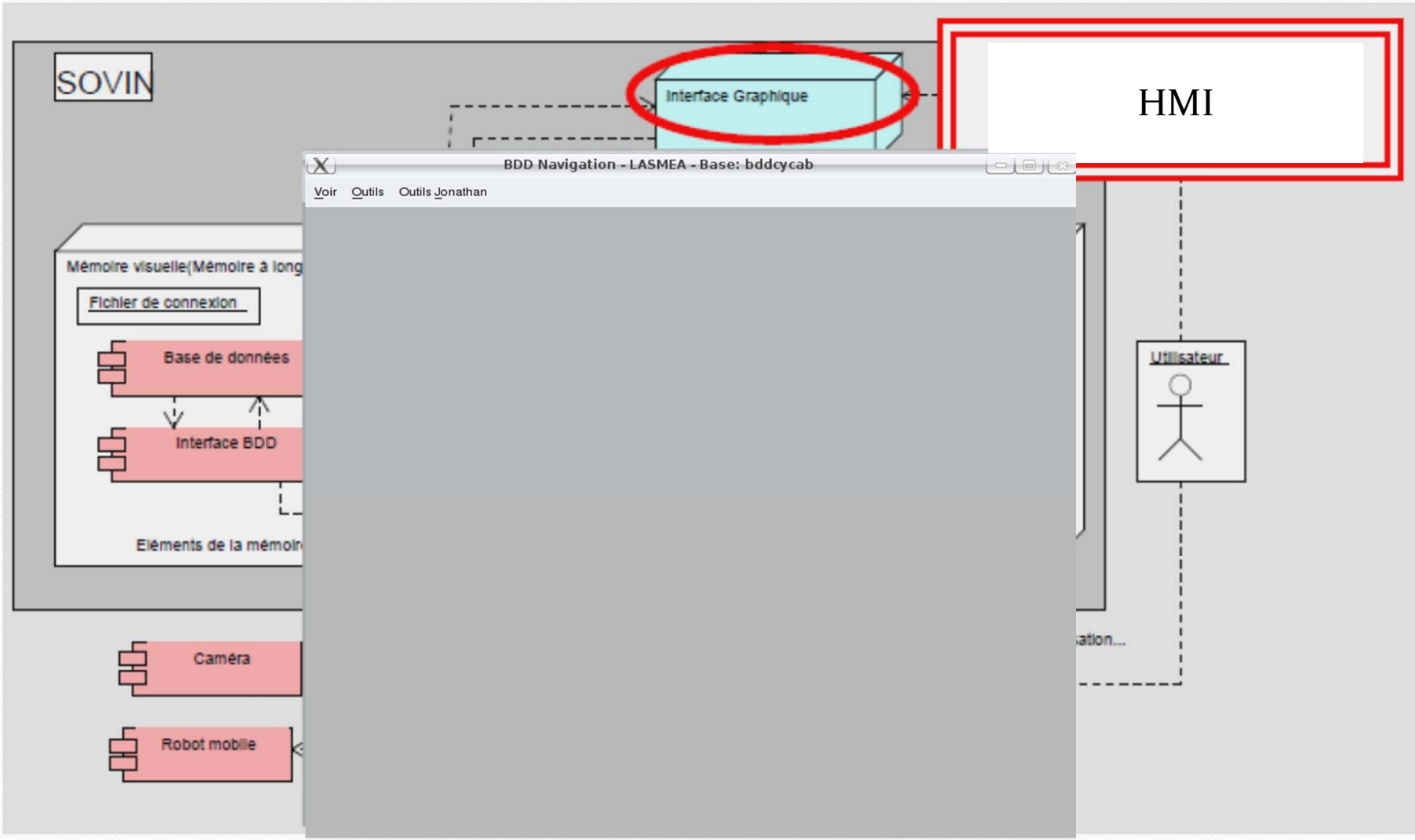
SOVIN A generic library (generic visual sensor) for visual algorithm processing

[Courbon 2008]



SOVIN An efficient Human Machine Interface for managing applications

[Courbon 2008]

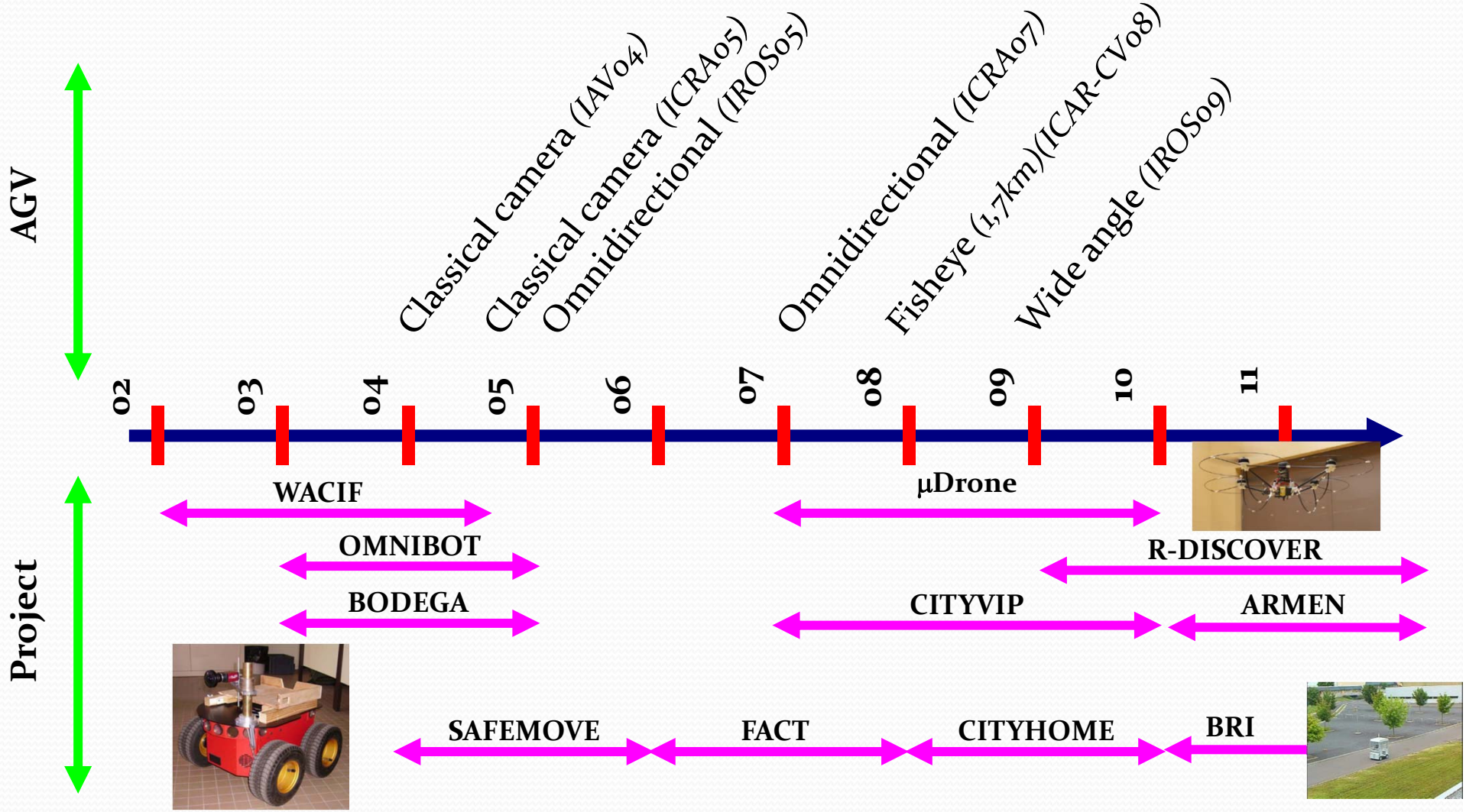


Content

- Introduction
- Generic models for cameras
- Visual geometry for localization
- Topological navigation system
- **Illustrations**
- Conclusion

Applications and projects in LASMEA

AGV using visual memory



Indoor applications : WACIF project (2002-2005)

Autonomous Robot for Telepresence navigation, localization, learning, warning and communication capabilities

- **Study and development of a demonstrator** like a personal robot fully integrated in the context of « wireless home »
- **An innovant domotic HMI** dedicated for software useful for human, family and home applications

→ **Wireless communication and services**

- **Telepresence and telesurveillance tasks**
  home supervision through any wireless point, abnormal event detection in family environment (intrusion, noise, …)

→ **Autonomous navigation strategy**

Indoor applications : WACIF project (2002-2005)

Global navigation strategy

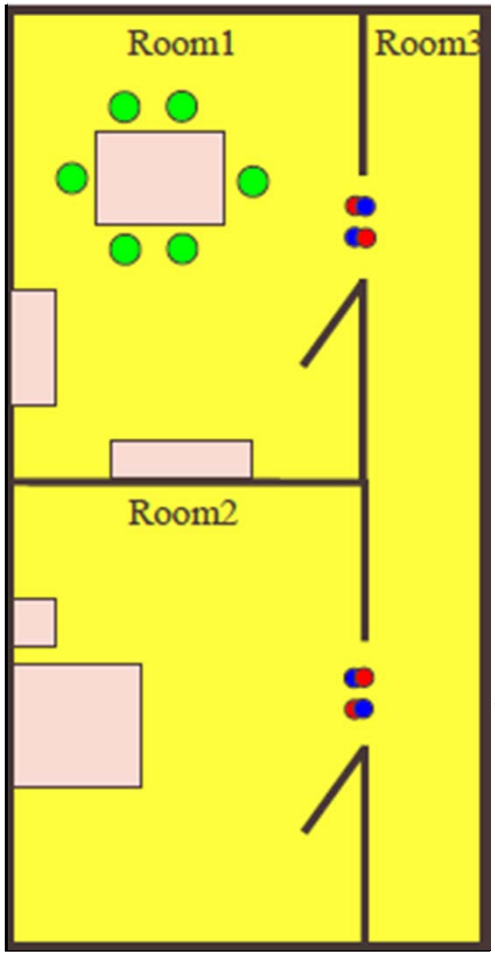
Topological representation
Predefined trajectory
Graph representation
Visual memory concept

Hypothesis : knowledge

$\mathrm{HS} = \mathrm{HOME\ Space}$

$\mathrm{RS}_i = \mathrm{Room\ Space}\ i$

$\mathrm{HS} = \sum_i RS_i$



- **Starting point**
- **Ending point**

Topological representation step is a part of the knowledge we get from the HOME environment

Indoor applications : WACIF project (2002-2005)

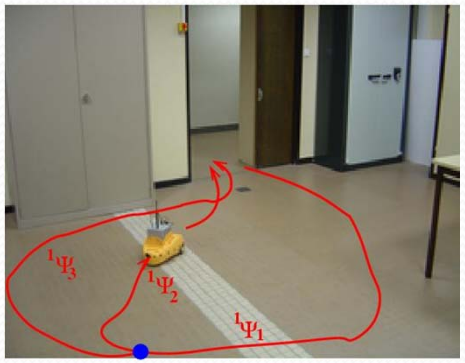
Global navigation strategy

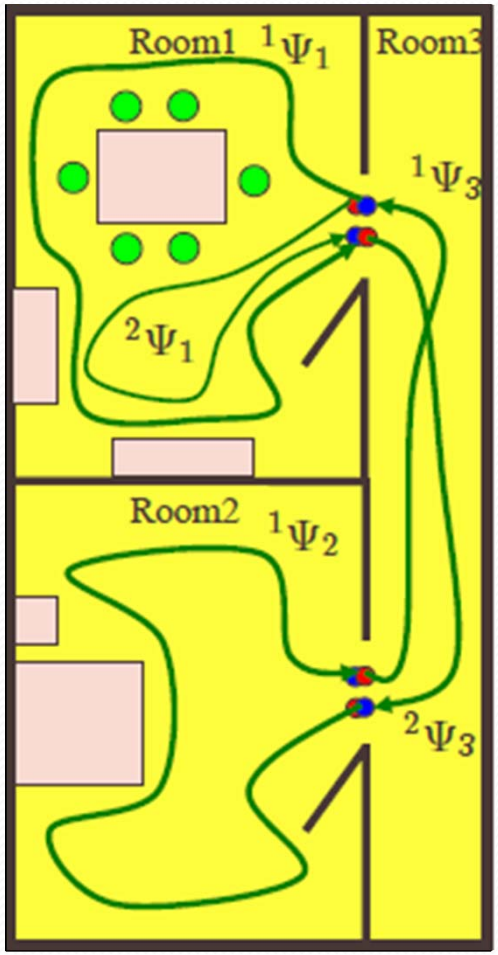
Topological representation

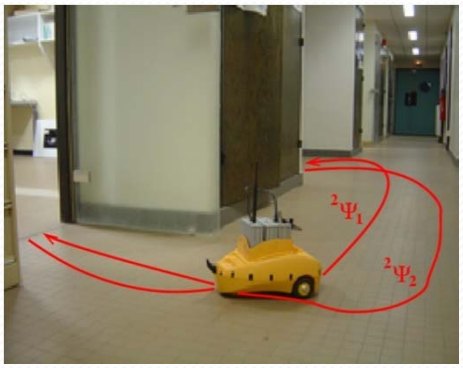
Predefined trajectory

Graph representation

Visual memory concept







Predefined trajectory step is totally supervised using robot teleoperation

Indoor applications : WACIF project (2002-2005)

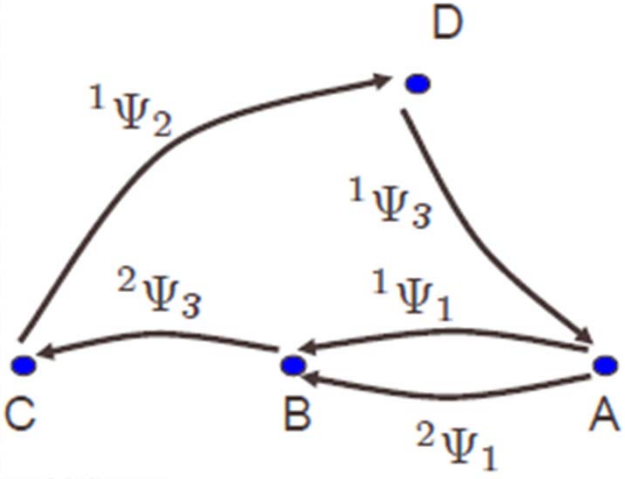
Global navigation strategy

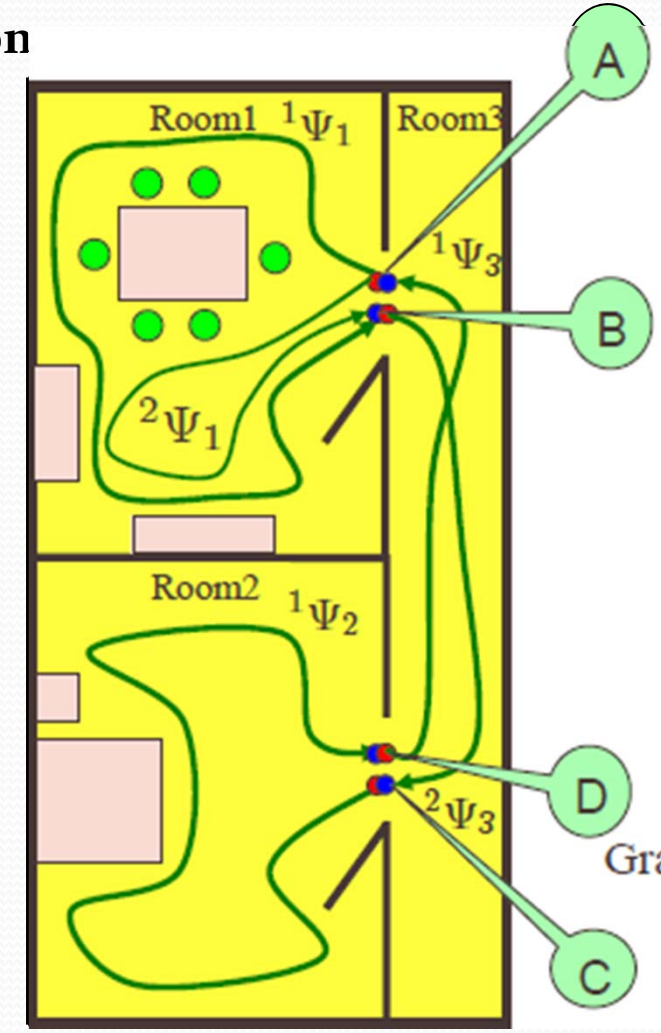
Topological representation

Predefined trajectory

Graph representation

Visual memory concept





Graph representation step is deduced from the previous step

Indoor applications : WACIF project (2002-2005)

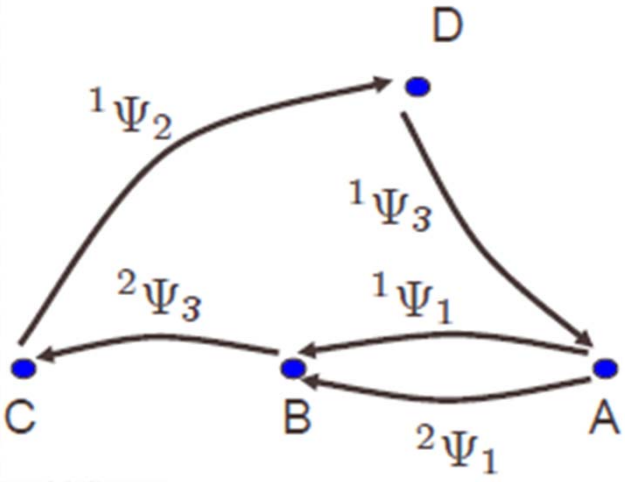
Global navigation strategy

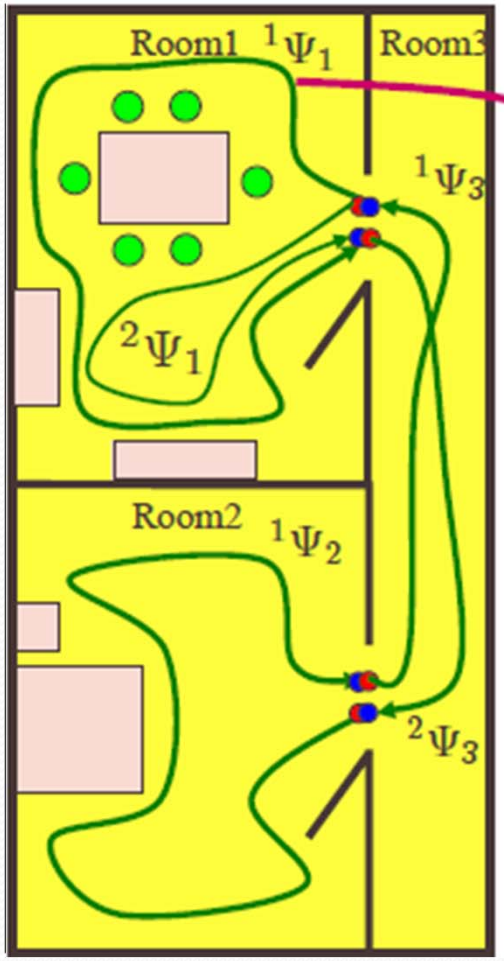
Topological representation

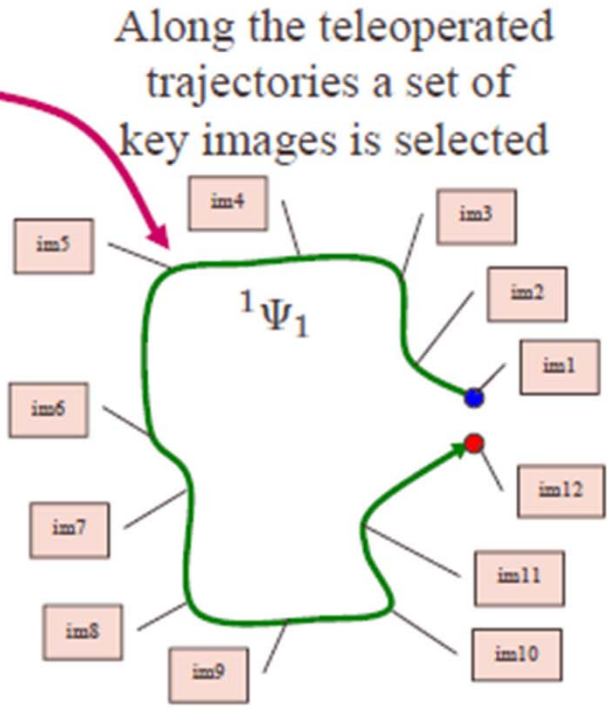
Predefined trajectory

Graph representation

Visual memory concept





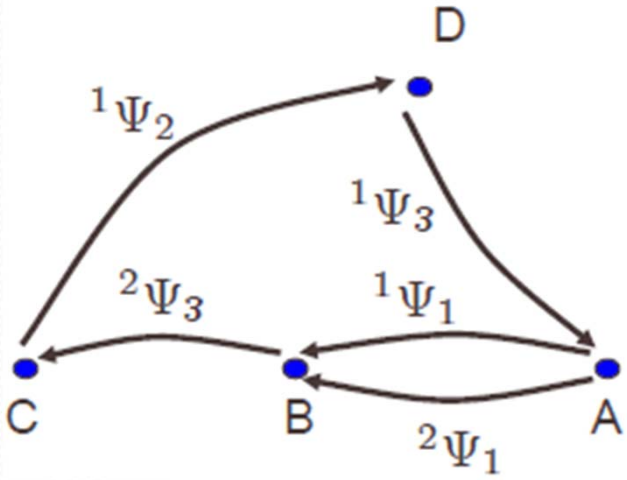


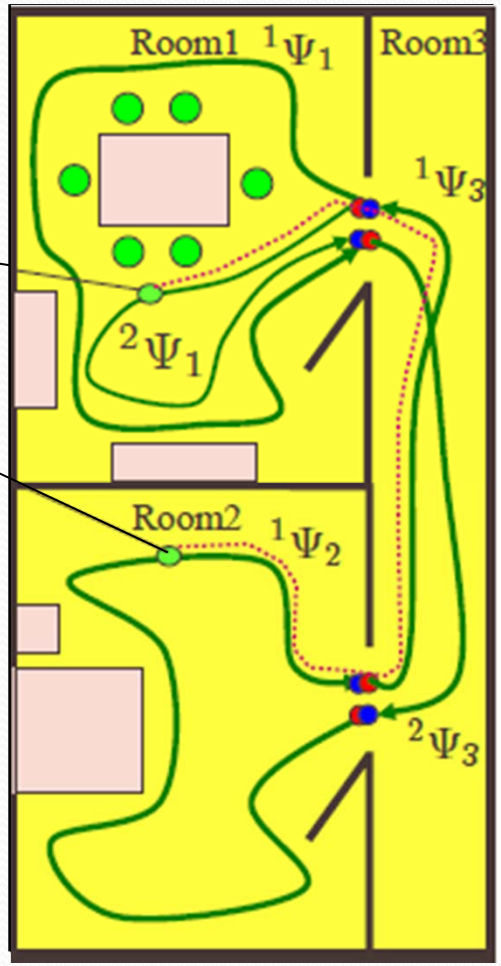
Indoor applications : WACIF project (2002-2005)

Global navigation strategy
Topological representation
Predefined trajectory
Graph representation
Visual memory concept

E : Ending point

S : Starting point

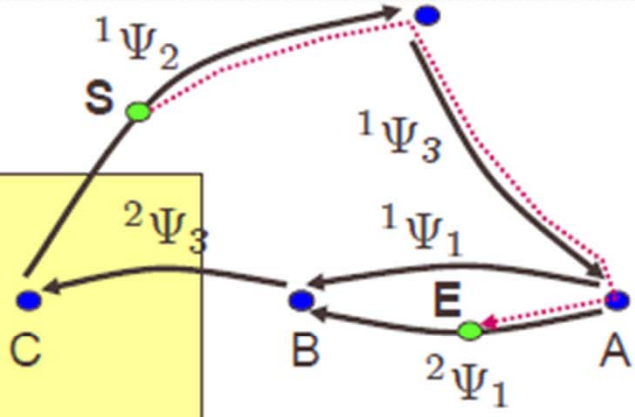




Indoor applications : WACIF project (2002-2005)

Global navigation strategy

Functional decomposition



Learning Off line

Define each $^i\Psi_j$: learn the trajectory

- ★ Robot teleoperation
- ★ Models features extraction and selection
- ★ Mosaic construction

Ait Ader [04]
Blanc [04]
Blanc [05]

Re-run of trajectory : On line

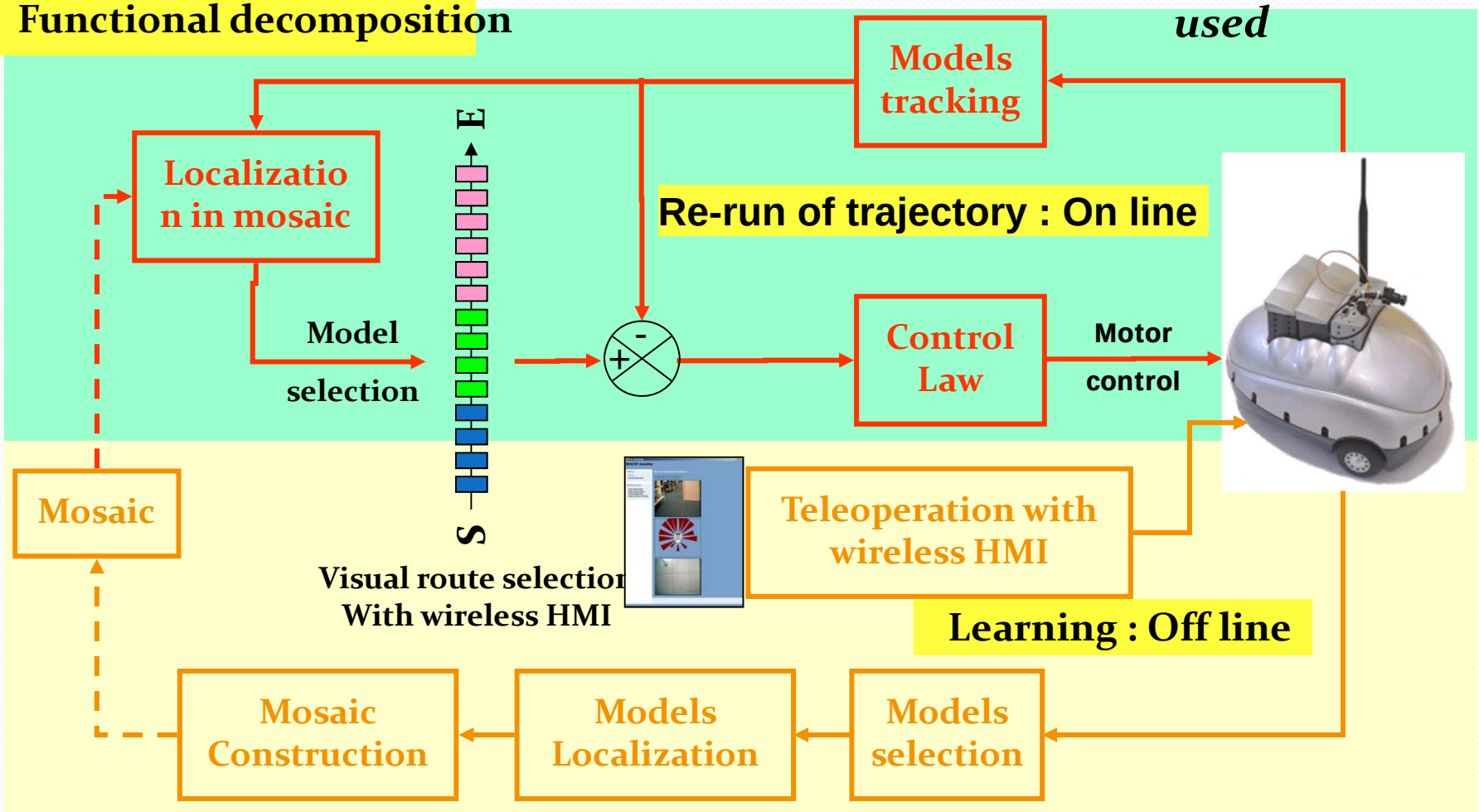
- ★ Extract the reference model from the visual route
- ★ Select and track models in current images
- ★ Localization in the mosaic
- ★ Lateral control of the non holonomous robot

Indoor applications : WACIF project (2002-2005)

Global navigation strategy

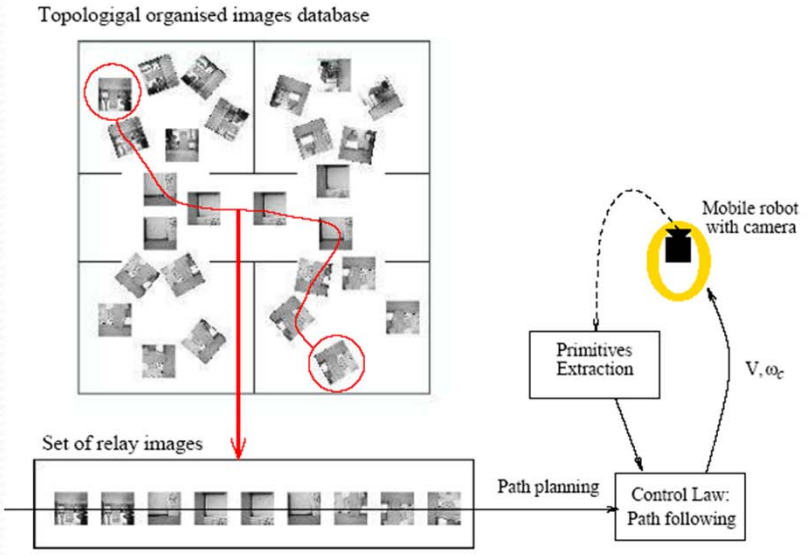
Ceiling images are used

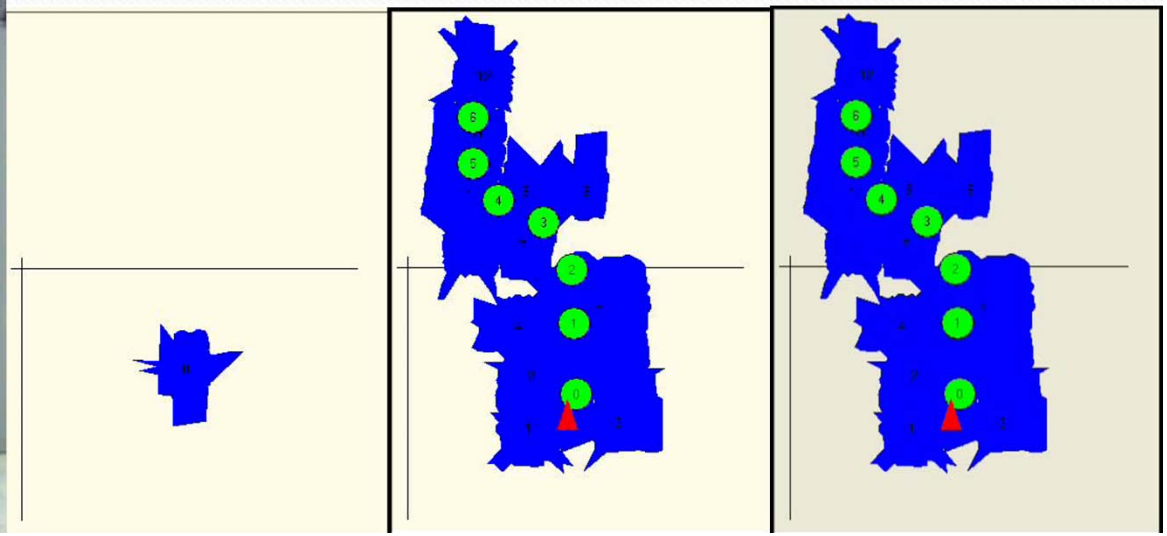
Functional decomposition



Indoor applications : WACIF project (2002-2005)

Navigation using visual memory





Indoor applications : OMNIBOT project

ROBEA

omnibot

From omnidirectional vision to mobile robot contr

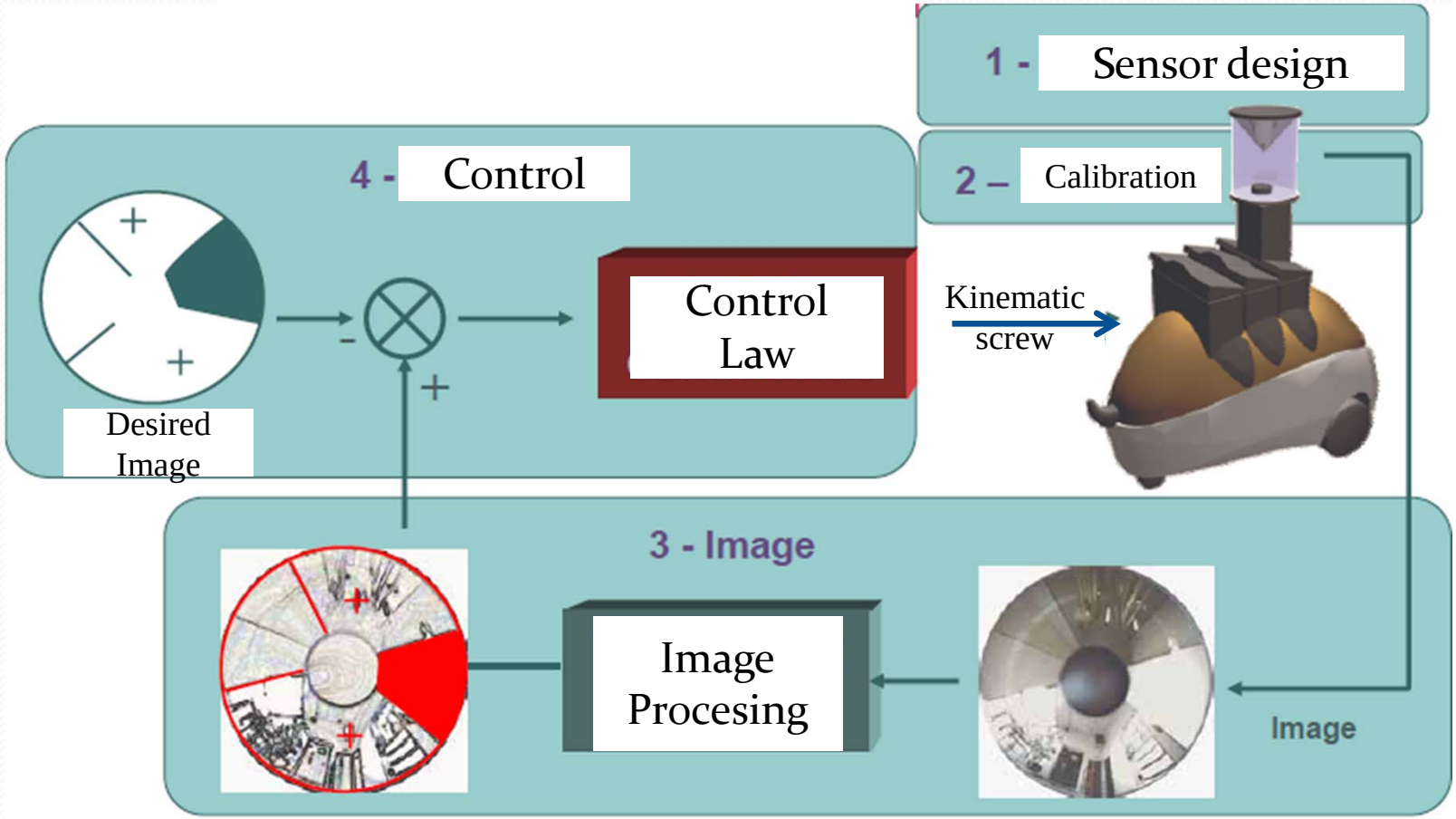
Development of a complete perception and control approach for mobile robot using omnidirectional vision

LASMEA

CREA EA 3299

INRIA

LIRMM

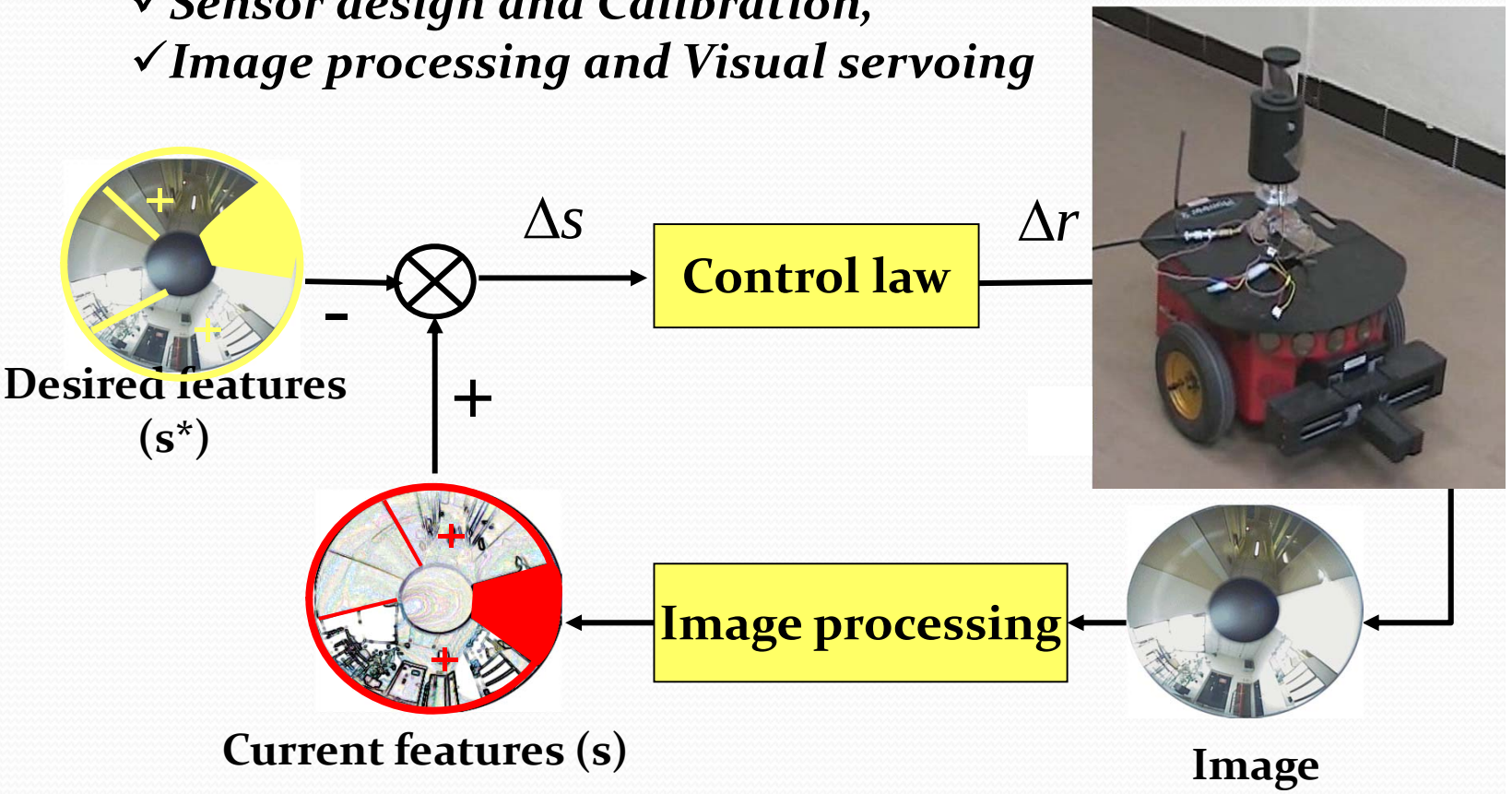


Indoor applications : OMNIBOT project

From omnidirectional vision to mobile robot contr

Development of a complete perception and control approach for mobile robot using omnidirectional vision :

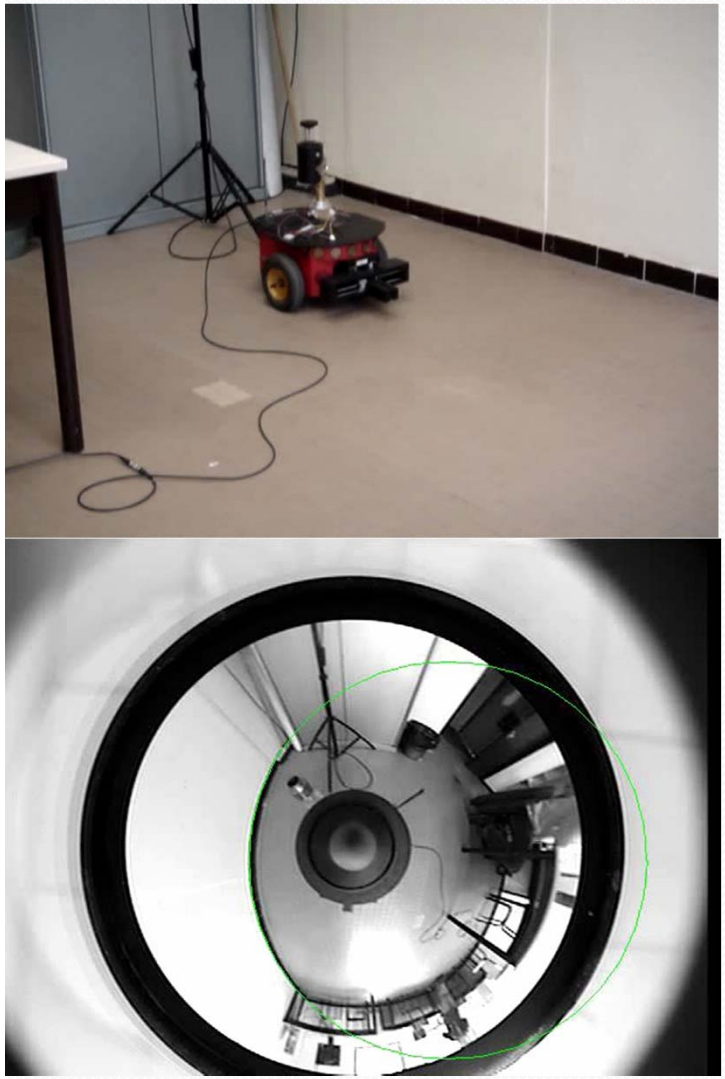
- ✓ ***Sensor design and Calibration,***
- ✓ ***Image processing and Visual servoing***



Indoor applications : OMNIBOT project

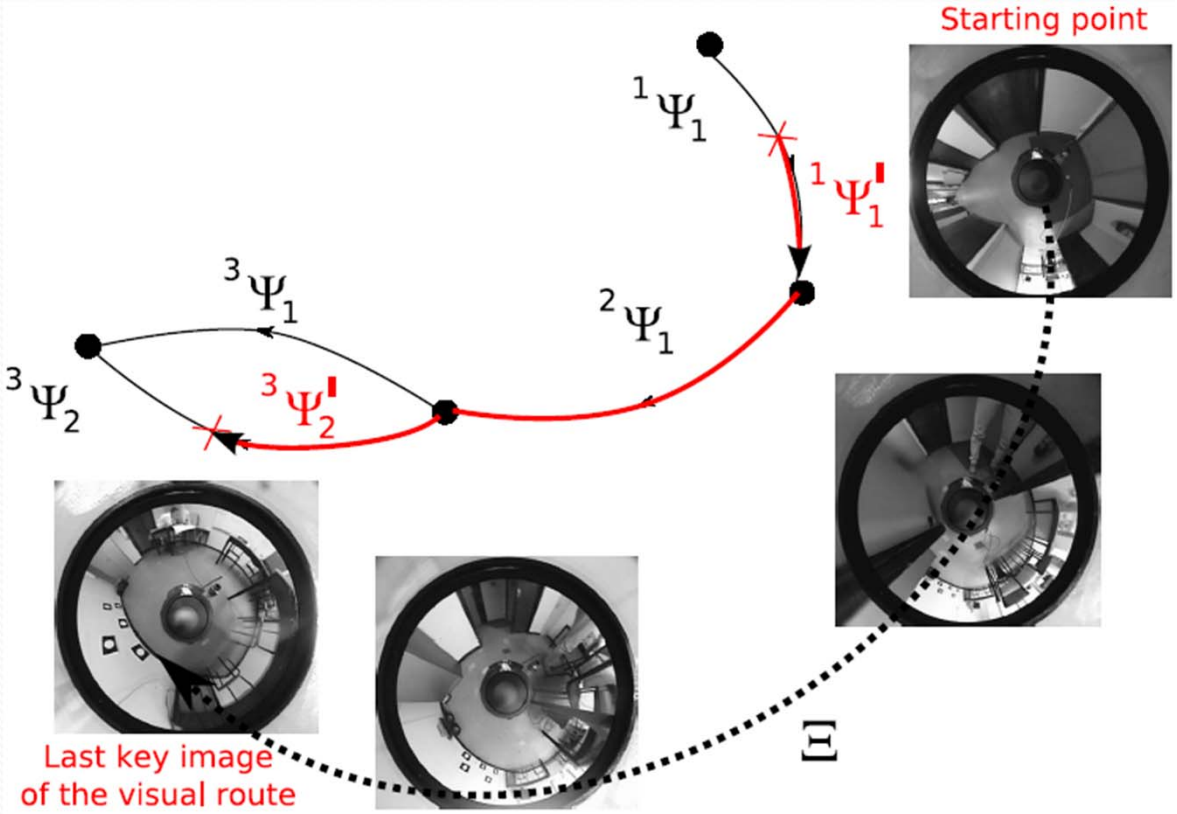
ROBEA

omnibot



H. Hadj Abdelkader [05]

Navigation with omnidirectional visual memory : [ICRA07]

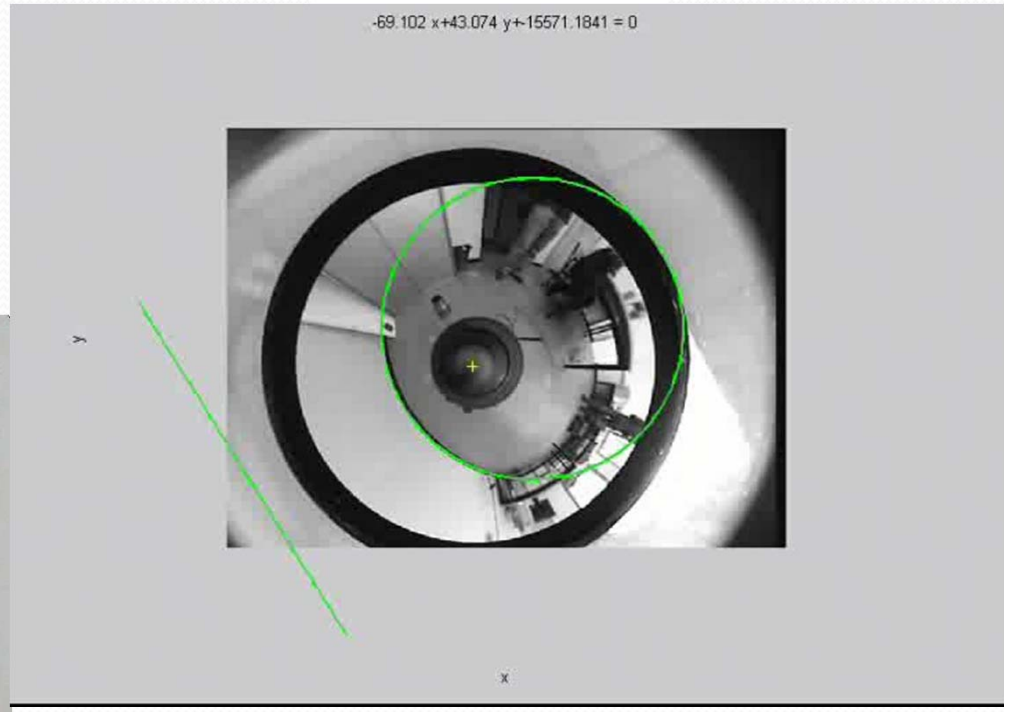


Indoor applications : OMNIBOT project

ROBEA

omnibot

Omnidirectional Visual Servoing from Polar Lines [Hadj 06]



Indoor applications : ROSACE project

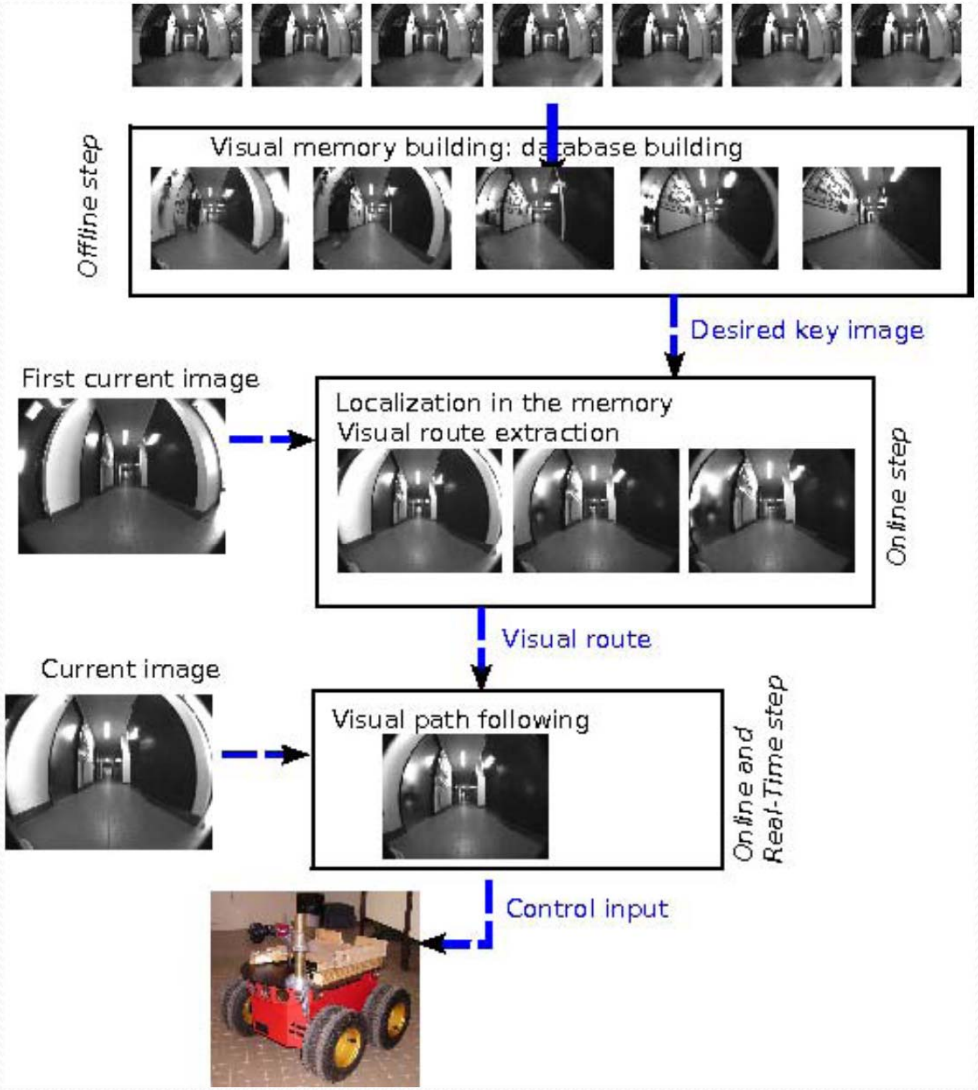
Using a fisheye camera

General framework :

Vision based memory navigation strategy :

3 steps :

1. ***Visual memory building***
2. ***Localization into the visual memory***
3. ***Navigation into the visual memory***



Indoor applications : ROSACE project

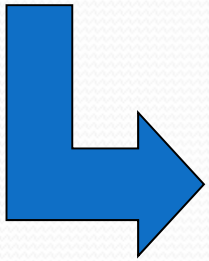
Using a fisheye camera

Localization

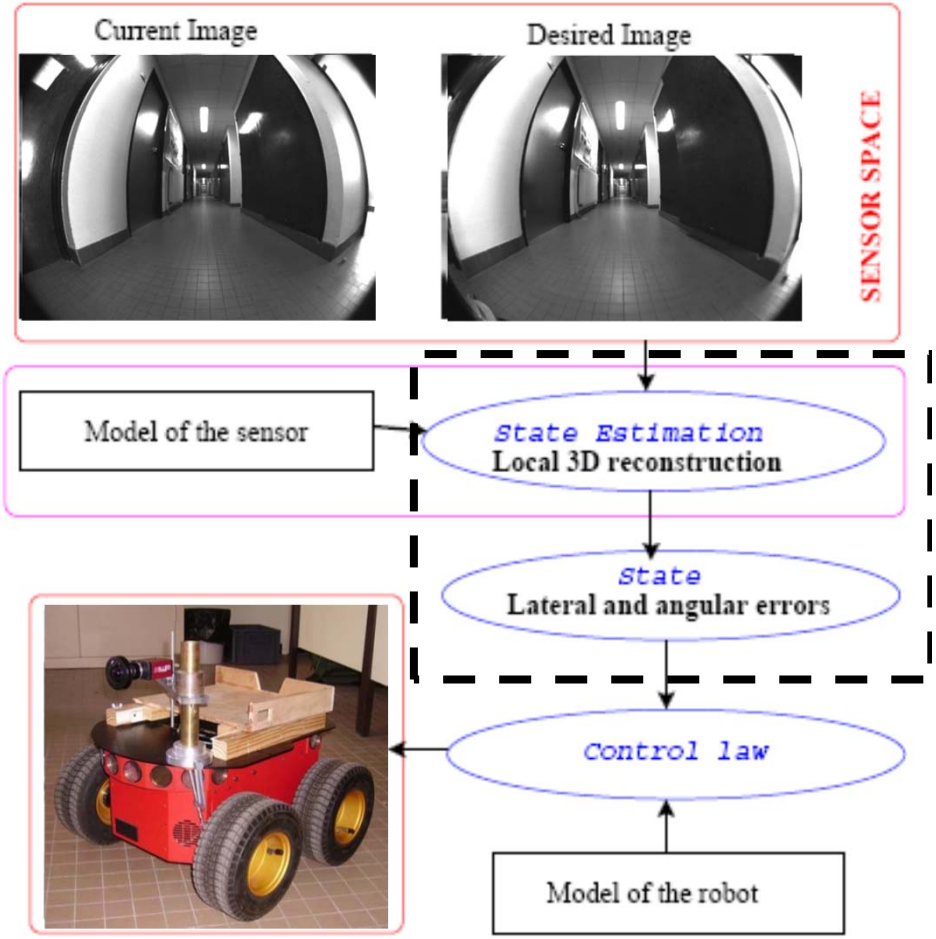
Vision based memory navigation strategy :

Localization into the visual memory

a- Global localization
b- Selection of key images
c- Relative pose computation



(y, θ) evaluation

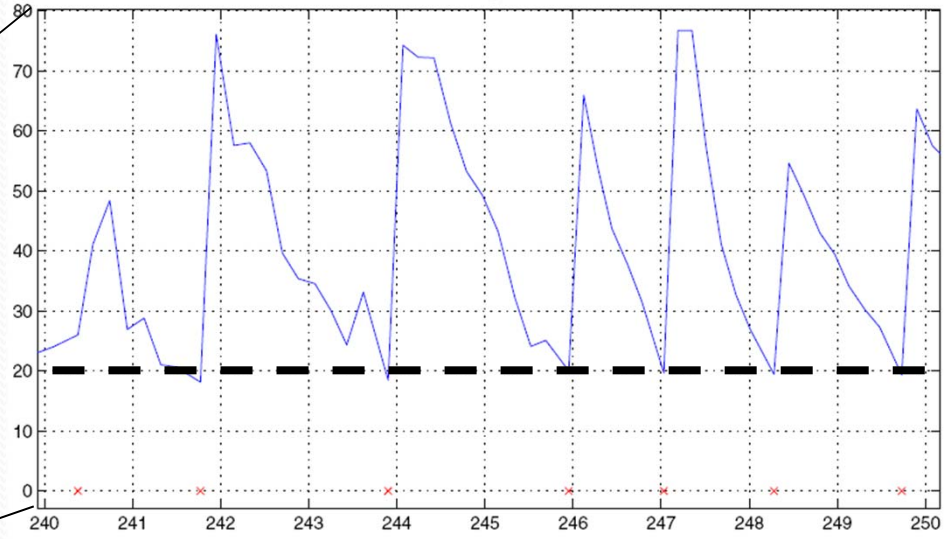


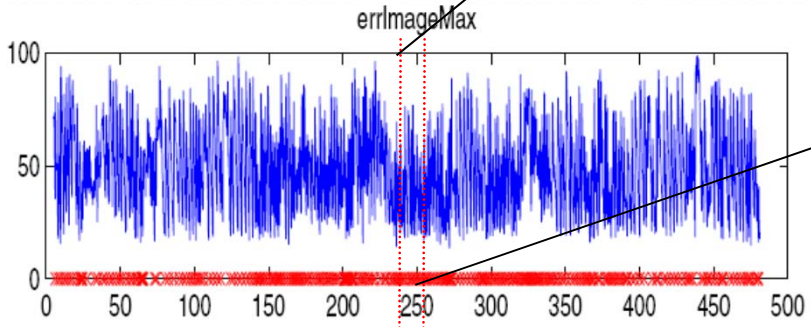
Indoor applications : ROSACE project

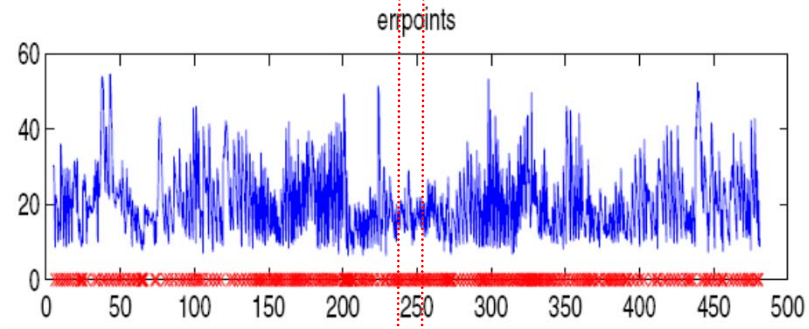
Using a fisheye camera

Current Image

Desired Image







Global localisation (6.5s) *306 matchings*
Visual route (280 key images) – Matching mean=108
- ErrImageMax:longest distance between an image point and its position in desired image
- Errpoints:Mean distance between those points

When errImageMax reaches 20
we decide to change key image

Indoor applications : ROSACE project

Using a fisheye camera

[Courbon08]

Indoor applications : ROSACE project

Using a fisheye camera

J. Courbon [07]

Automatic Guided Vehicles

robot control using a fisheye lens

Autonomous navigation using a visual memory

Outdoor applications : MOBIVIP project

urban context
[Royer04]

Visual memory

Reference Trajectory

and

Image features (3D points)

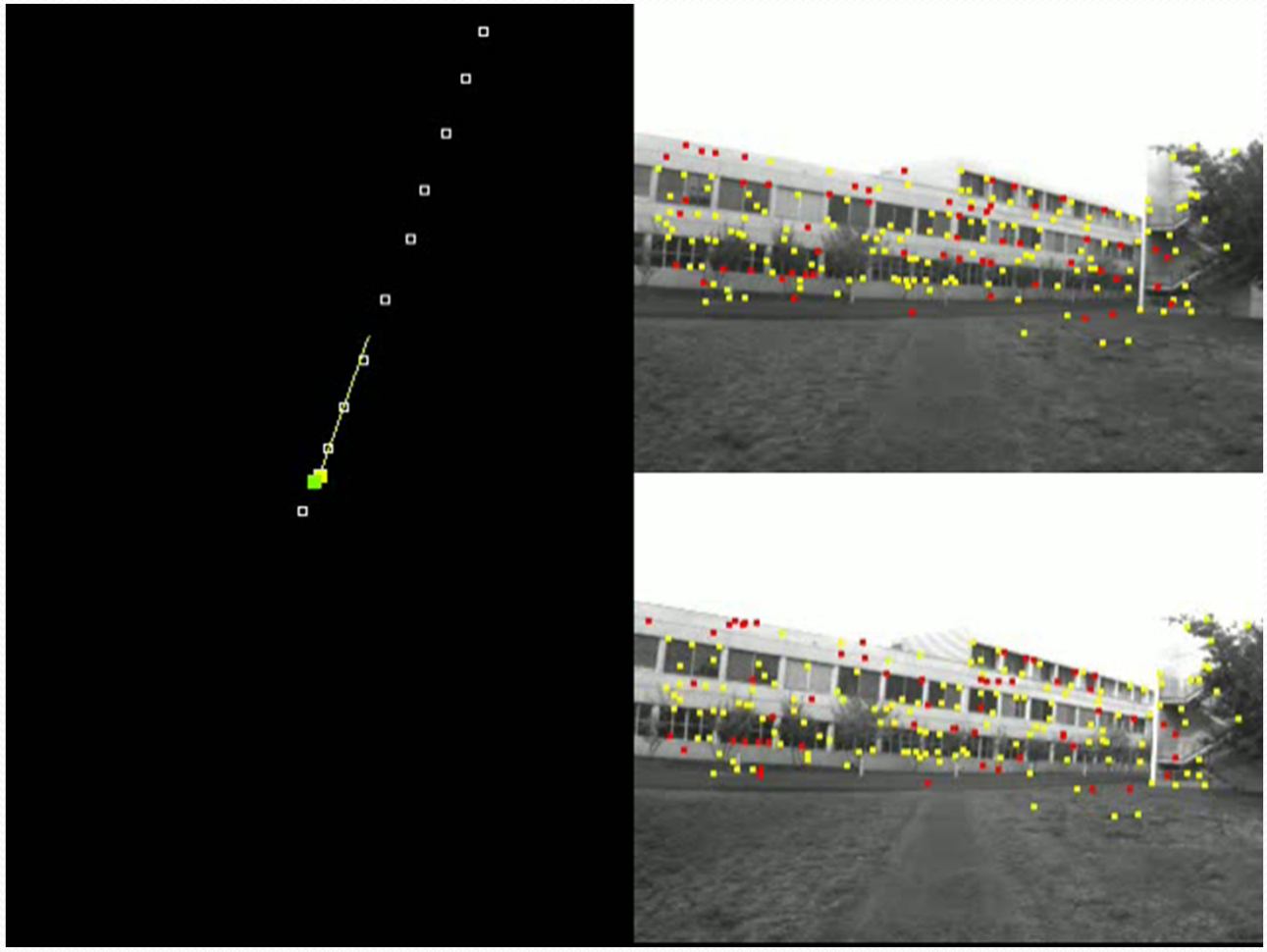
Experimental data Compiegne city center (BODEGA/MOBIVIP)

Outdoor applications : MOBIVIP project

urban context [Royer04]

Perception Localization using monocular camera

Reference Trajectory and localization



Key images And features

Current images and features

Outdoor applications : MOBIVIP project

urban context

[Royer06]

Perception and control

2006: Les Cezeaux Campus CFd

2006: Inner city of Clermont-Fd

Outdoor applications : CITYVIP project

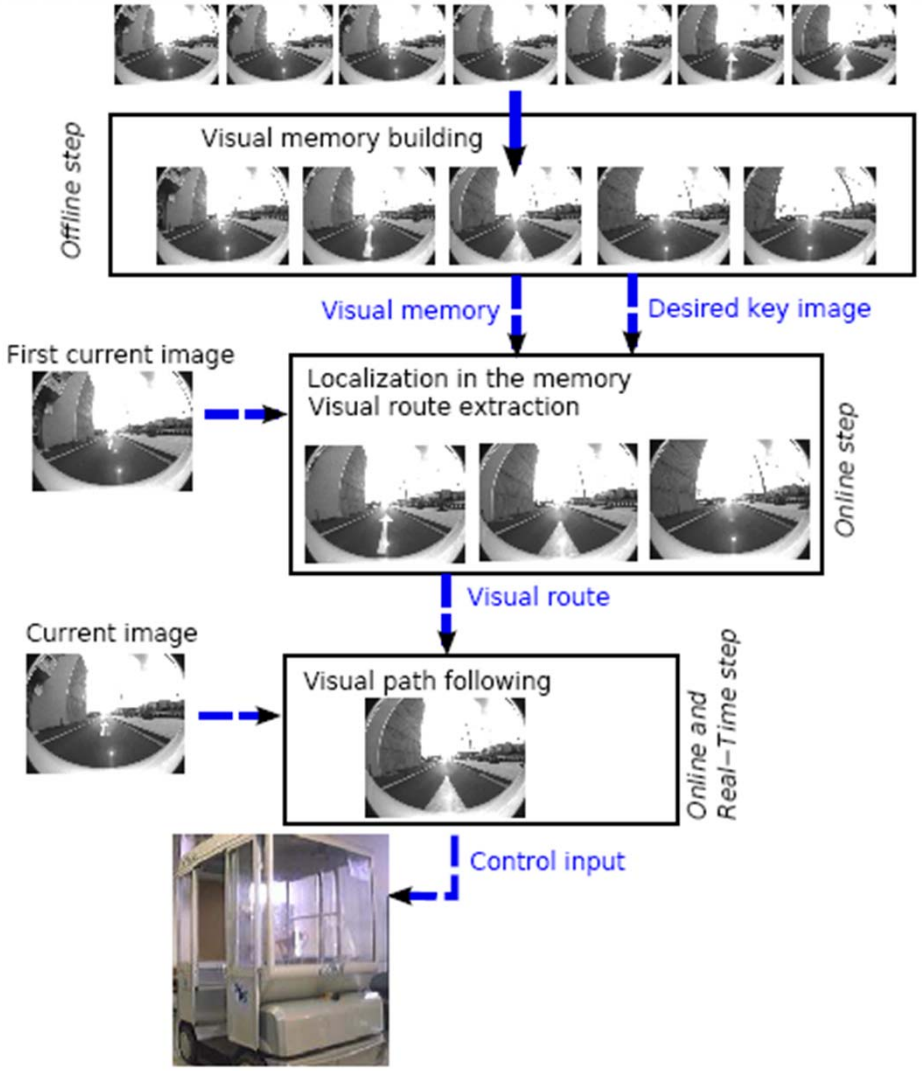
[ICRA09] urban context

General framework :

Vision based memory navigation strategy :

3 steps :

1. ***Visual memory building***
2. ***Localization into the visual memory***
3. ***Navigation into the visual memory***



Outdoor applications : CITYVIP project

[ICRA09] urban context

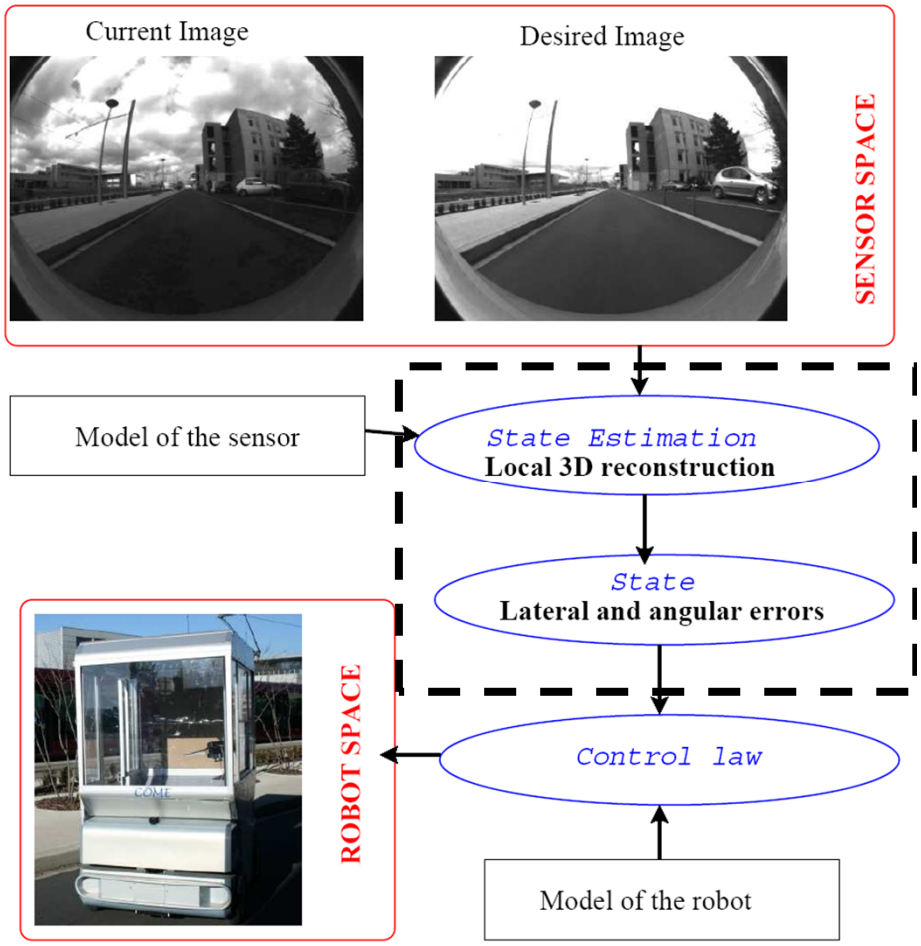
Localization

Vision based memory navigation strategy :

Localization into the visual memory

a- Global localization
b- Selection of key images
c- Relative pose computation

(y, θ) evaluation



Outdoor applications : CITYVIP project

[ICRA09] urban context

Testbed

Experimental robot is a Robucab from Robosoft

Algorithms are implemented in C^{++} language on a laptop using RTAI-Linux OS with a 2GHz Centrino processor

Fujinon fisheye lens, mounted onto a Marlin F131B camera
Field-of-view of 185 deg
Image resolution in the experiments was 800×600 pixels
Frame rate of 15fps

Longitudinal velocity V has been fixed to $1\,ms^{-1}$
K_p and K_d are tuned regarding a double pole located at value 0.3

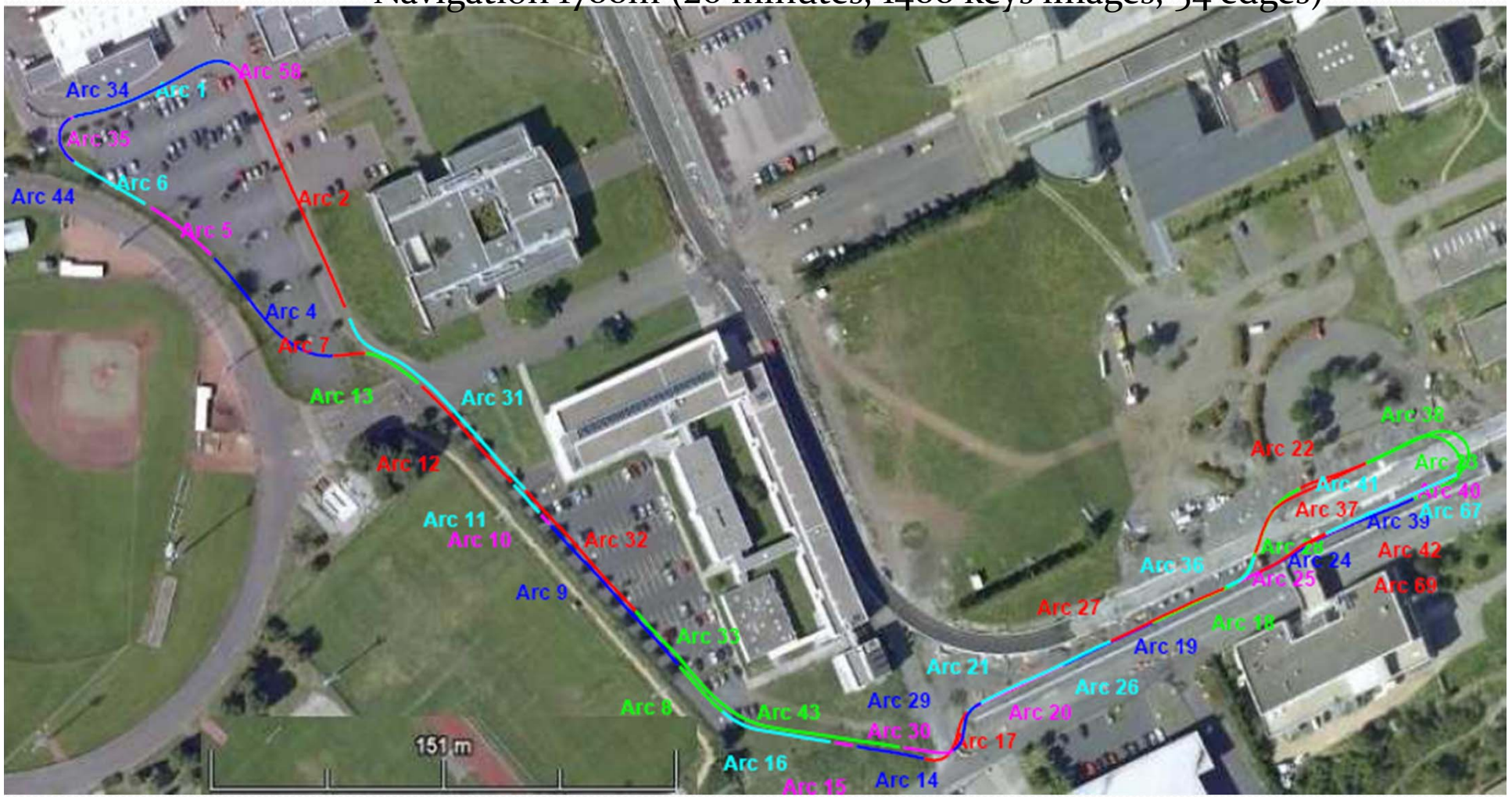
Outdoor applications : CITYVIP project

[ICRA09] **urban context**

Experiment

Loop 1200 m – 35 edges

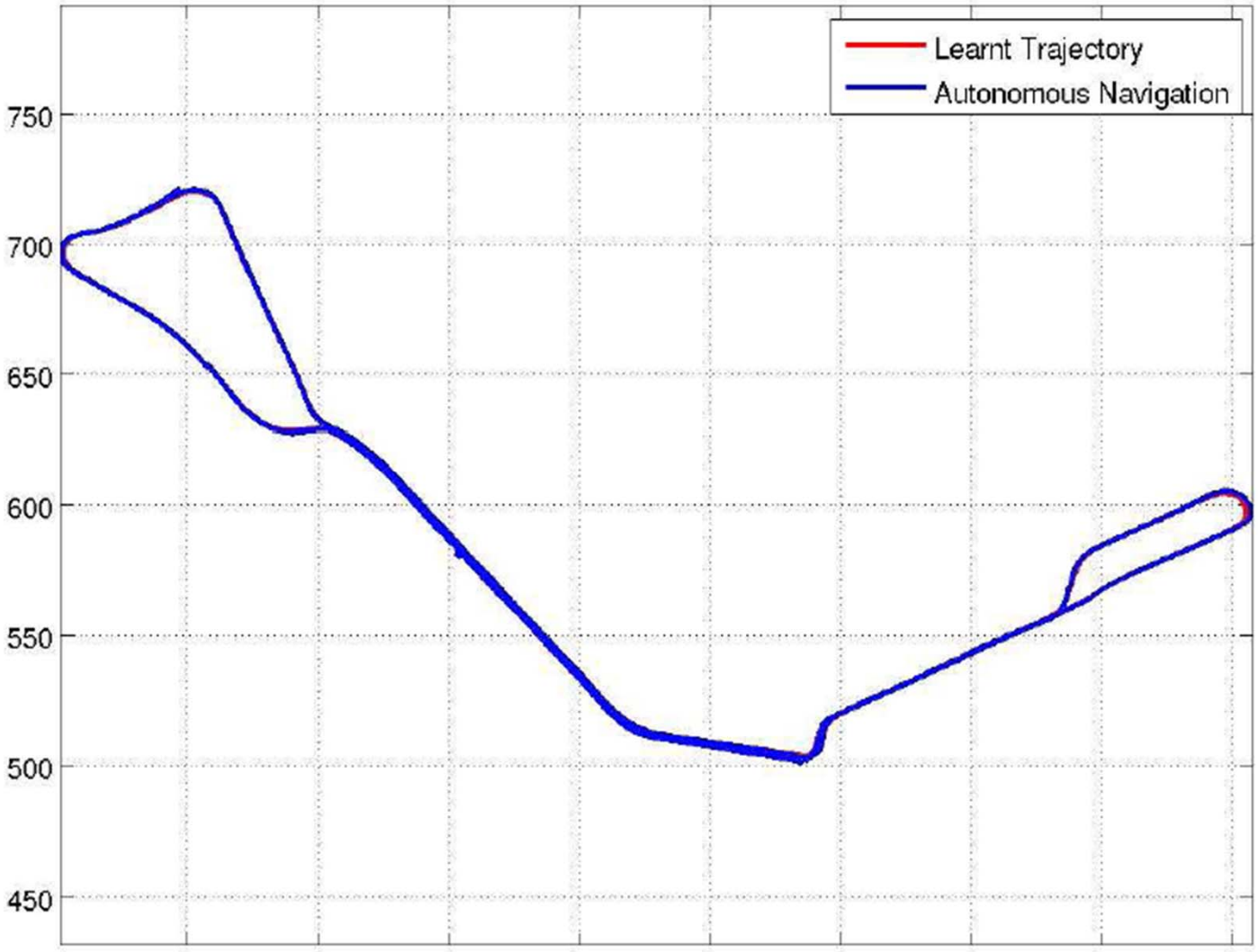
Navigation 1700m (26 minutes, 1400 keys images, 54 edges)



Outdoor applications : CITYVIP project

[ICRA09] urban context

Experiment



Lateral error:
- mean: 23 cm
-standard deviation: 30 cm.

Navigation stops after 1700m for security reasons (small number of matched points)

Outdoor applications : CITYVIP project

[ICRA09] urban context

Video

Topological navigation using visual memory

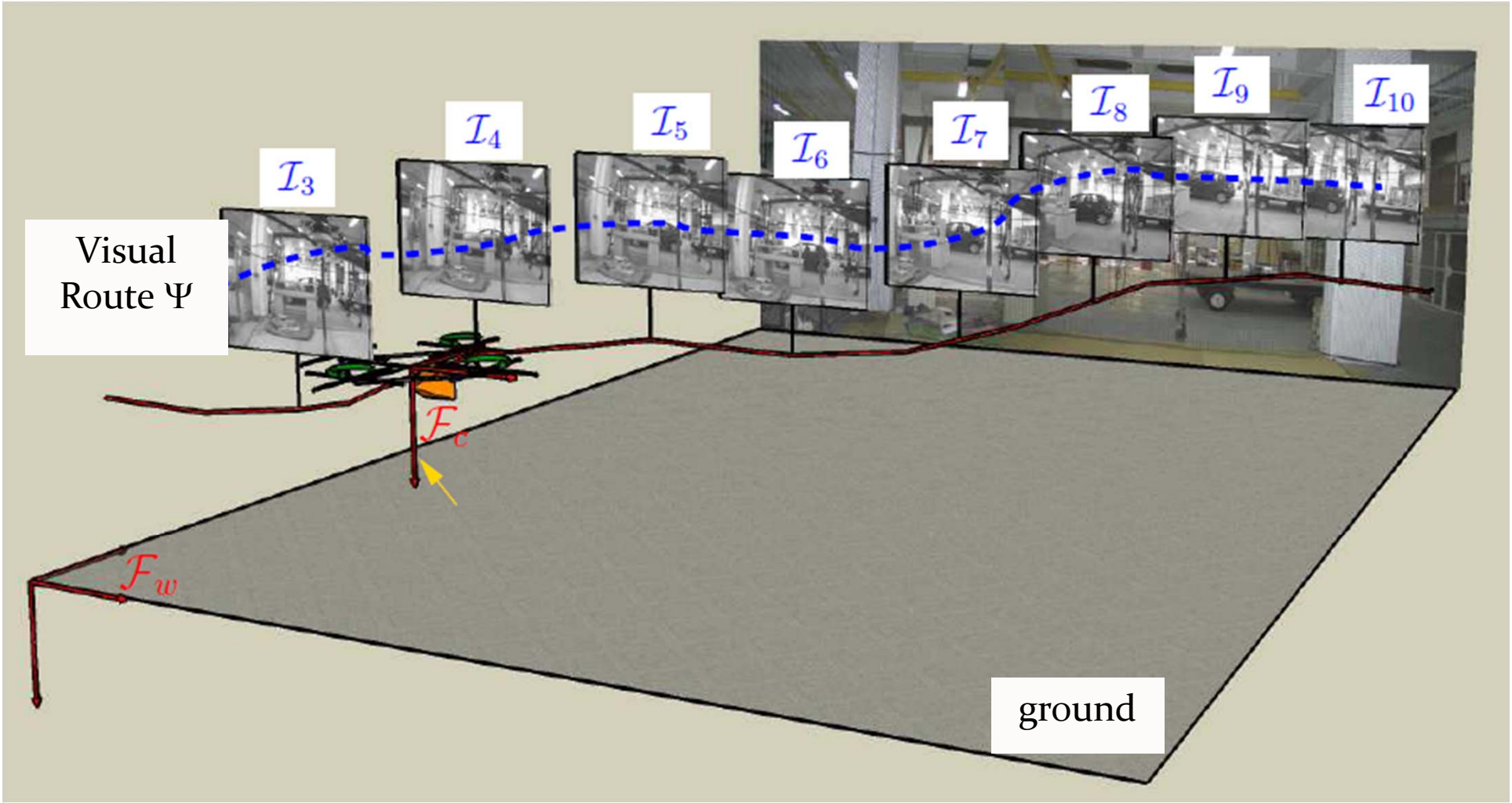
Cityvip
Clermont-Fd 2008

Using vision only
Clermont-Ferrand
LASMEA-GRAVIR

Indoor applications : µdrone project

Flying robot

[IROS09]



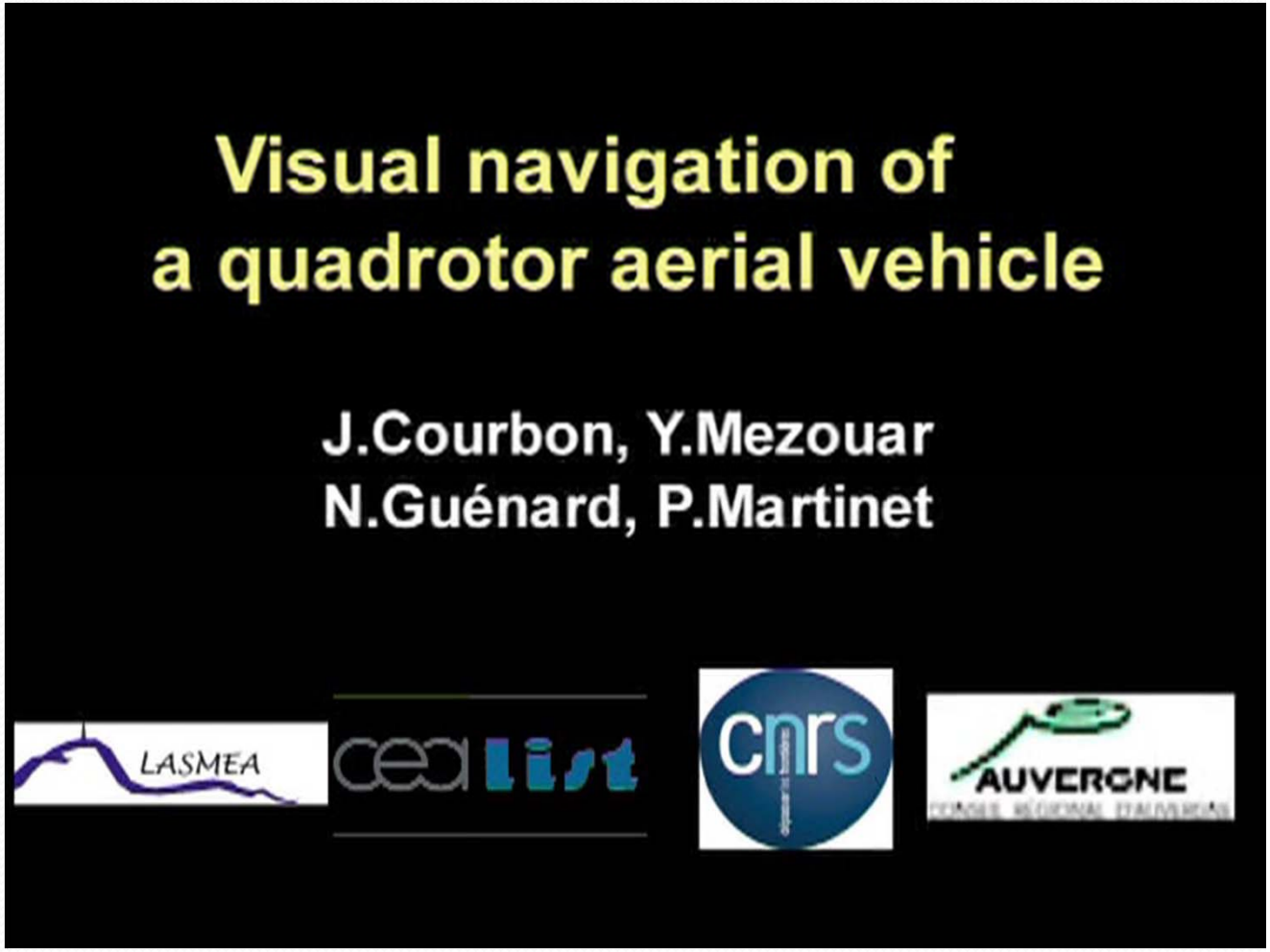
Indoor applications : µdrone project

Flying robot
[IROS09]

Topological navigation using visual memory

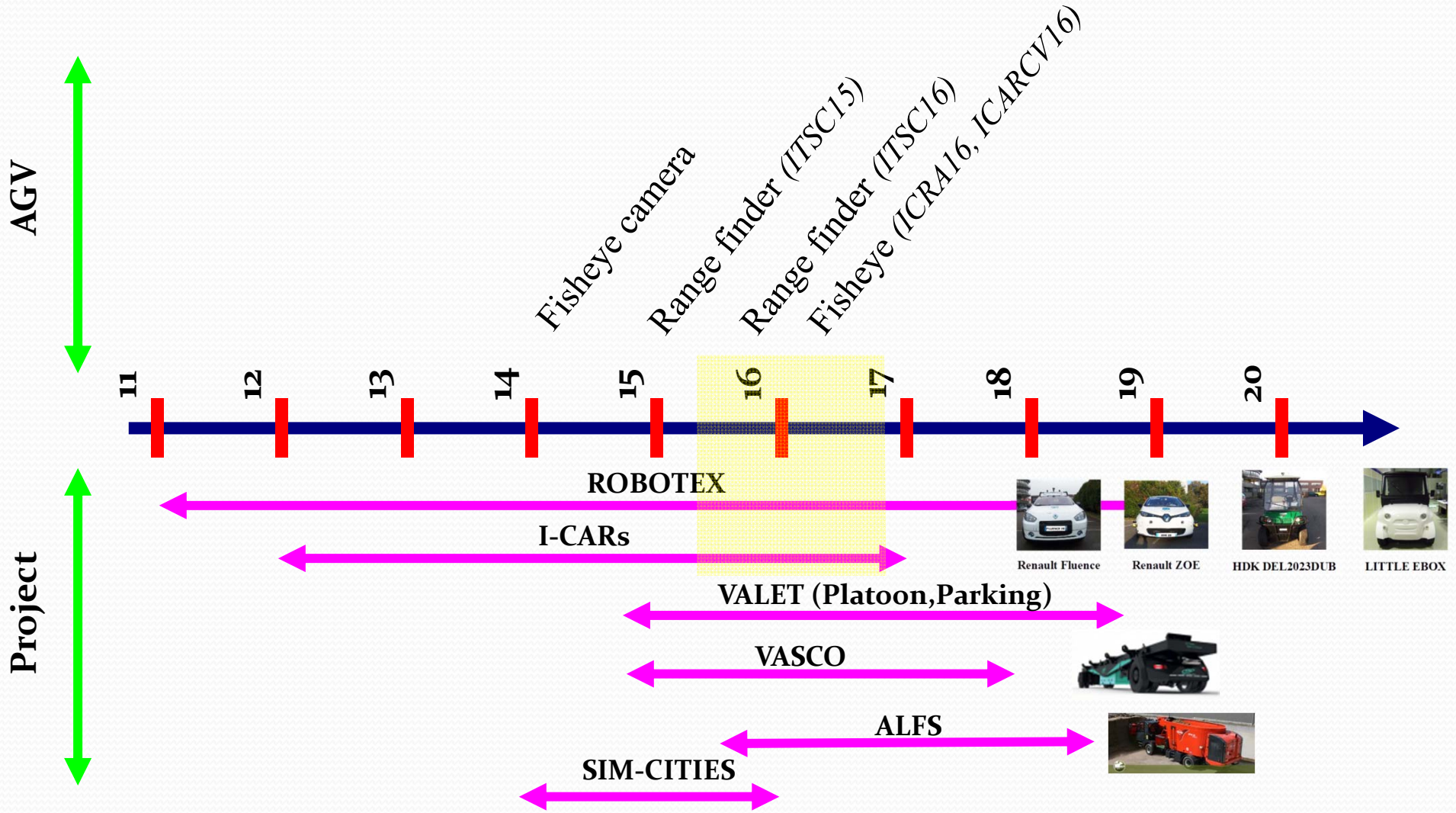
CEA 2009
Fontenay-aux-Roses

Using vision only
Clermont-Ferrand
LASMEA-GRAVIR



Applications and projects in IRCCyN/LS2N

AGV using sensory memory



I-CAR project

Renault Fluence

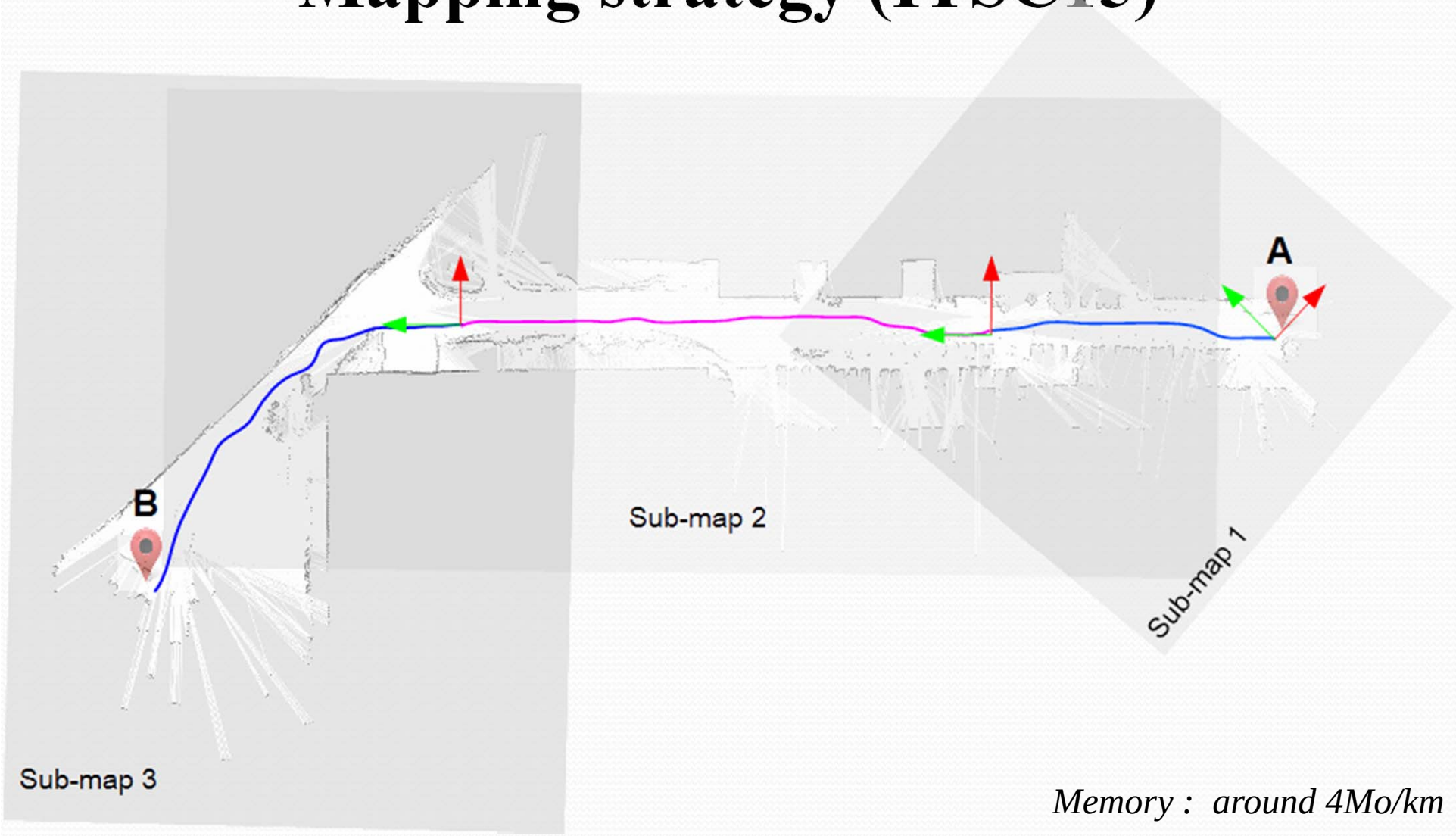
Renault ZOE

HDK DEL2023DUB

LITTLE EBOX

Robotization and instrumentation
Robust localization (Data fusion)
Mapping, SLAM
Automous navigation (topological)
Pedestrian Detection and tracking
Platooning
Parking and complex maneuvers

Mapping strategy (ITSC15)

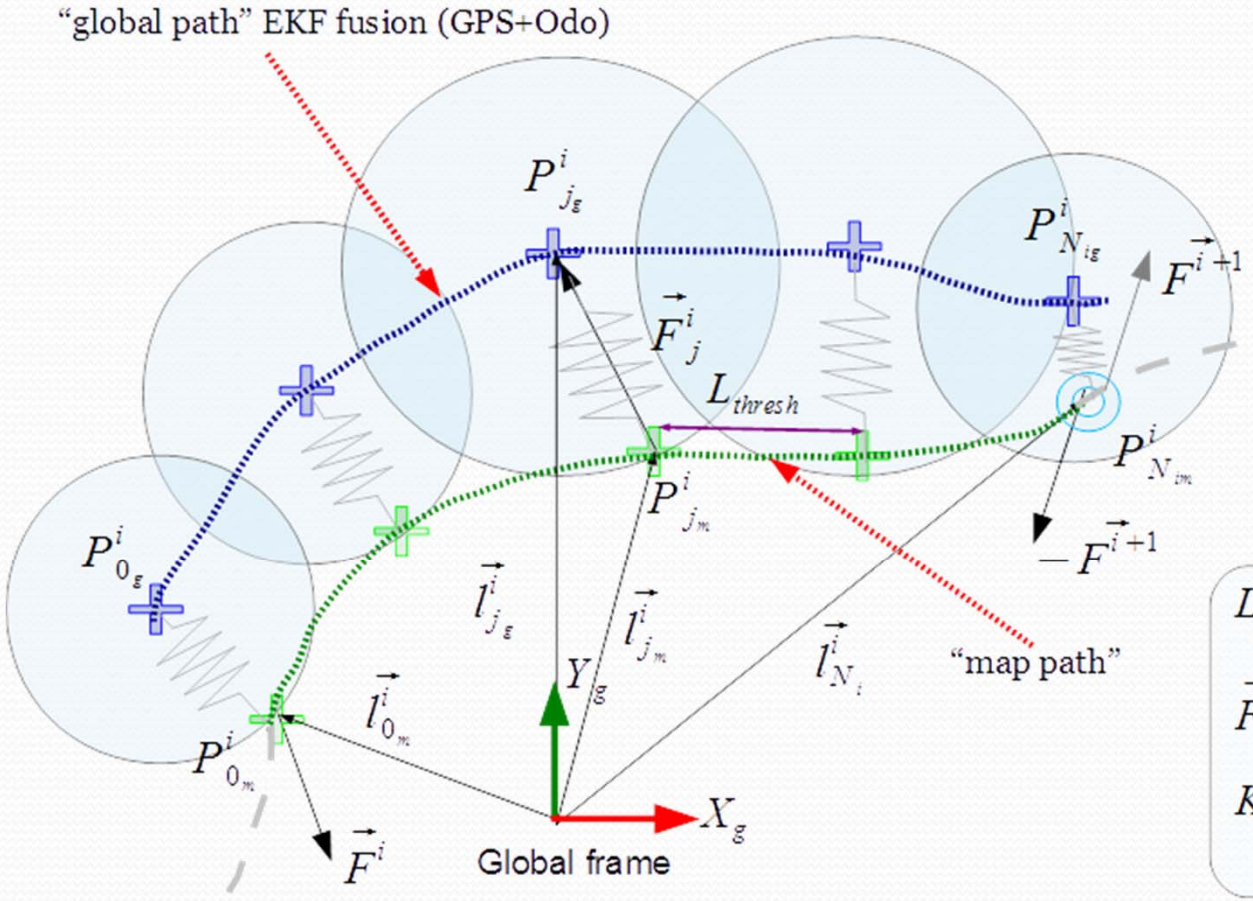


Memory : around 4Mo/km

Mapping strategy (ITSC15)

Superindex → sub-map → $i = 1 \cdots n$ **Sub-map index**

Subindex → path point → $j = 1 \cdots N_i$ **Path point index**



Global Optimization

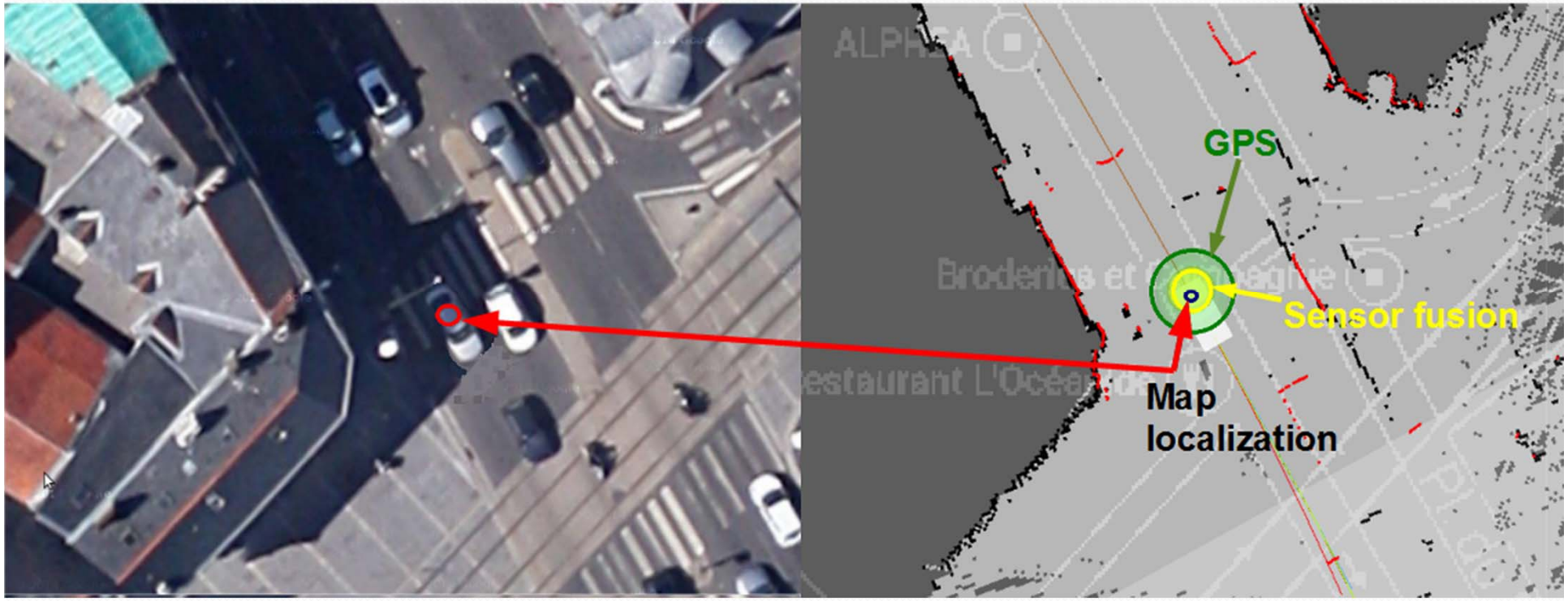
$$L_{j,j+1} \geqslant L_{threshold}$$

$$\vec{F}^i_j = K^i_j * \vec{l}^i_j$$

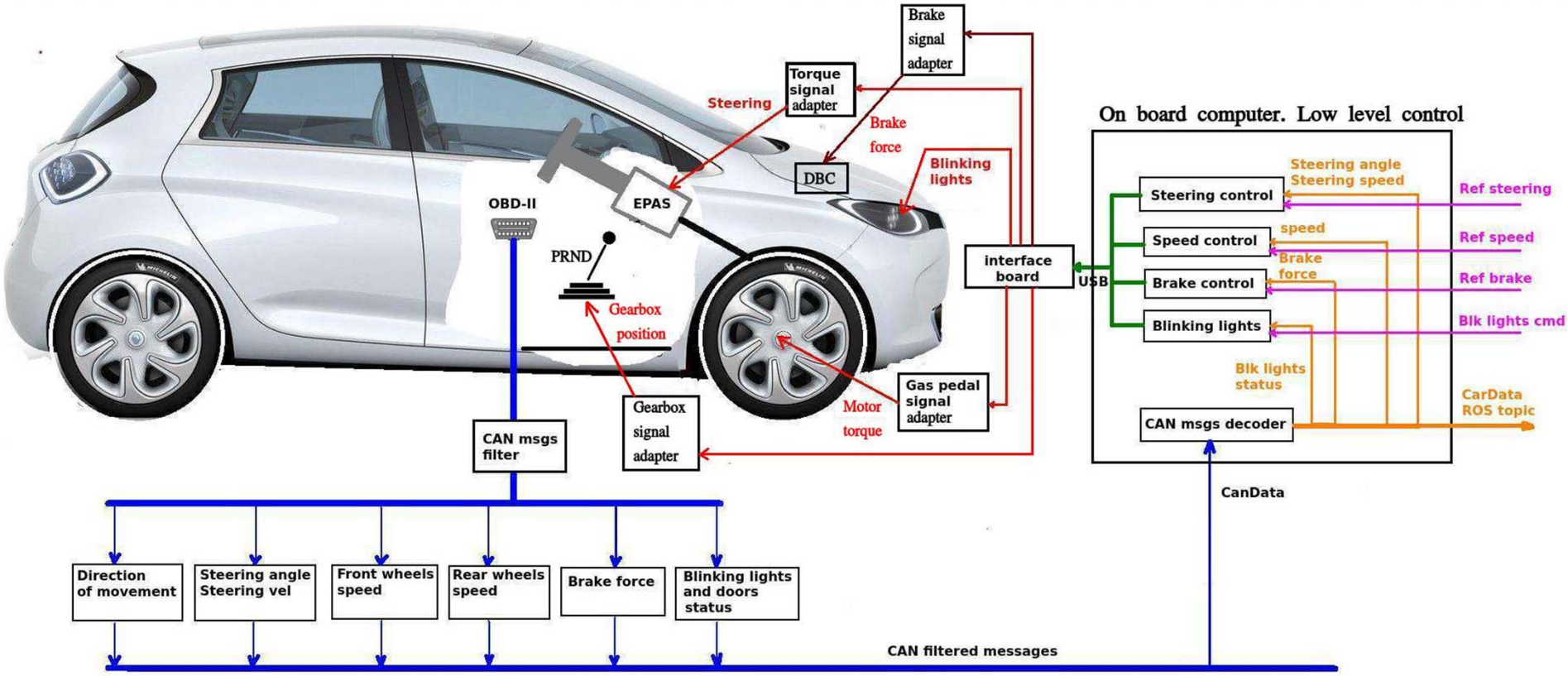
$$K^i_j = \frac{1}{cov^i_j}$$

Localization (ITSC15)

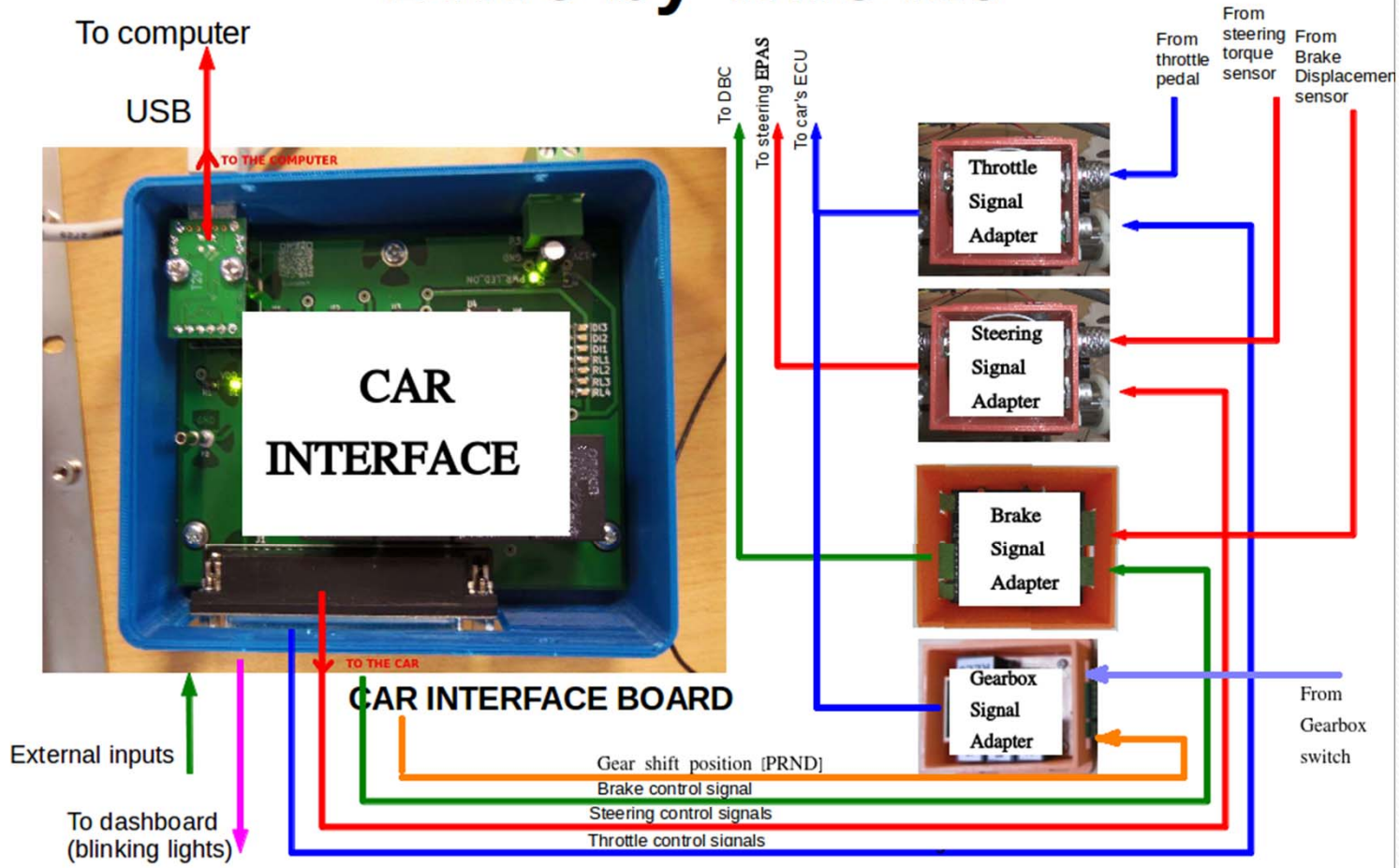
Enhanced localization using multi-maps localization



ROS based architecture



Drive-by-wire kit



Autonomous Navigation (ITSC16)

Autonomous Navigation

I-CAR project (2012-2016)

IRCCyN

Ecole Centrale de Nantes

S. Dominguez Quijada, G. Garcia, X. Koreki, A. Hamon, P. Martinet

Content

- Introduction
- Generic models for cameras
- Visual geometry for localization
- Topological navigation system
- Illustrations
- **Conclusion**

Conclusion

- Defining control in sensor space (with associated semantic level)
- Avoiding accumulation of modeling errors
- Sensor based control is effective in mobile robotic
- Sensor based control is a good alternative to path planning
- Sensor based control allows to be more reactive in dynamic environment

- We need to work hard to show results in real world with long life learning ability, try and see, and make decision…