Computer and robot vision

Tracking
for robotics and augmented reality

Éric Marchand

UMR IRISA

Eric Marchand

Professor, Université de Rennes 1

Member of the INRIA Lagadic team

http://www.irisa.fr/lagadic/team/marchand

A first application: augmented reality

Definition : [Azuma 97]

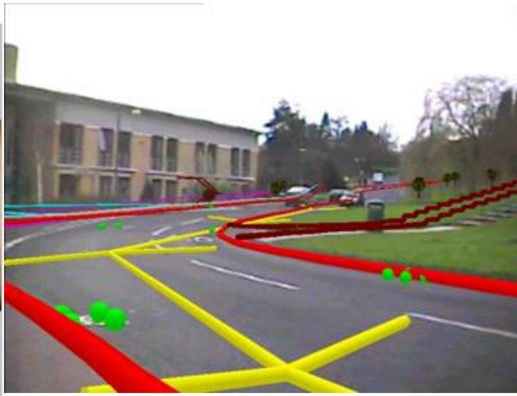
- Add virtual object in the video stream
- In real-time

Theoretical problem to be solved

- Find the camera position

Extensions

- post-production



Applications

Diffusion d'événements sportifs

Effets spéciaux

Étude d'impact

Tourisme interactif

Applications militaires

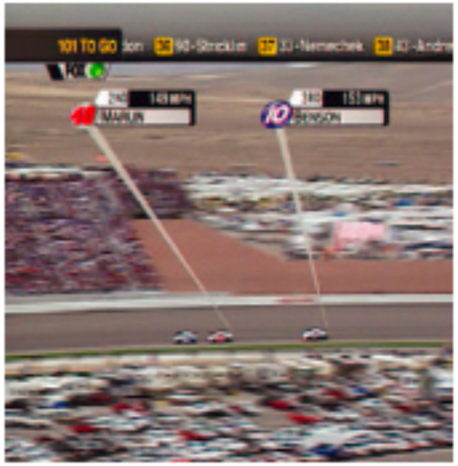
Design intérieur

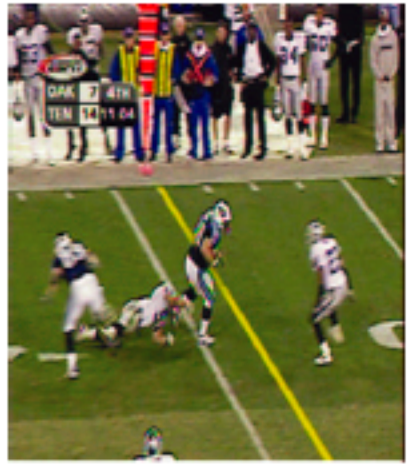
Aide à la maintenance, assemblage

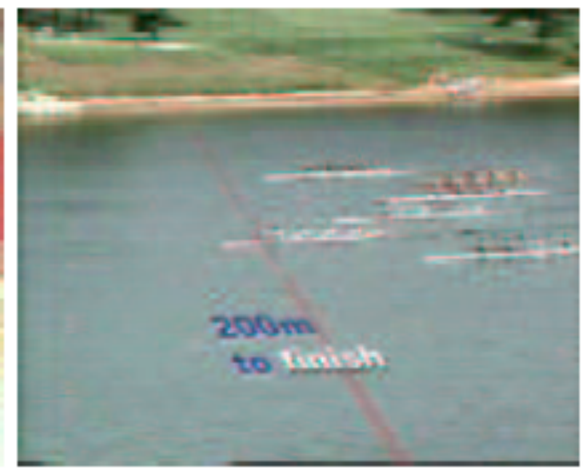
Médecine

Jeux

Sport







Sports


alles in OBI
FUTURSHOW
Ebly


TF1 9:04
LEE 0-0 VAL
PlayStation 2
EUROCARD
AMSTEL

FX


DIVX
VIDEO

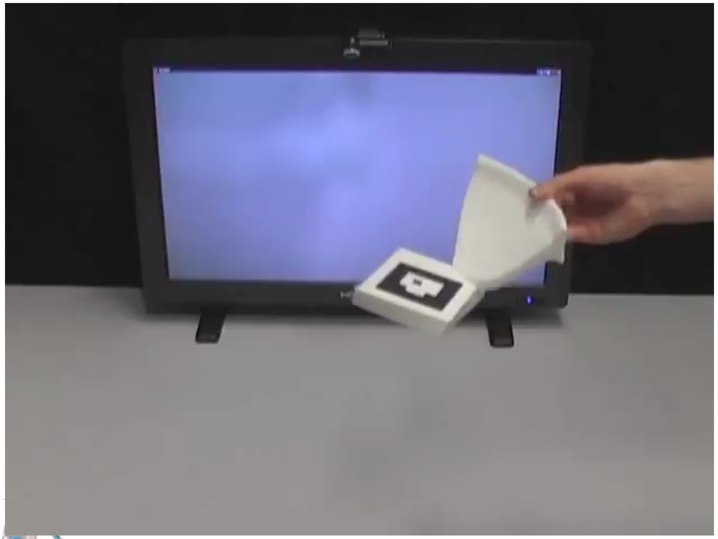

Terminator 3: Rise of the Machines™
© 2003 IMF Internationale Medien und Film GmbH & Co. 3 Produktions KG


Hello

FX

Impact study / architecture / archeology

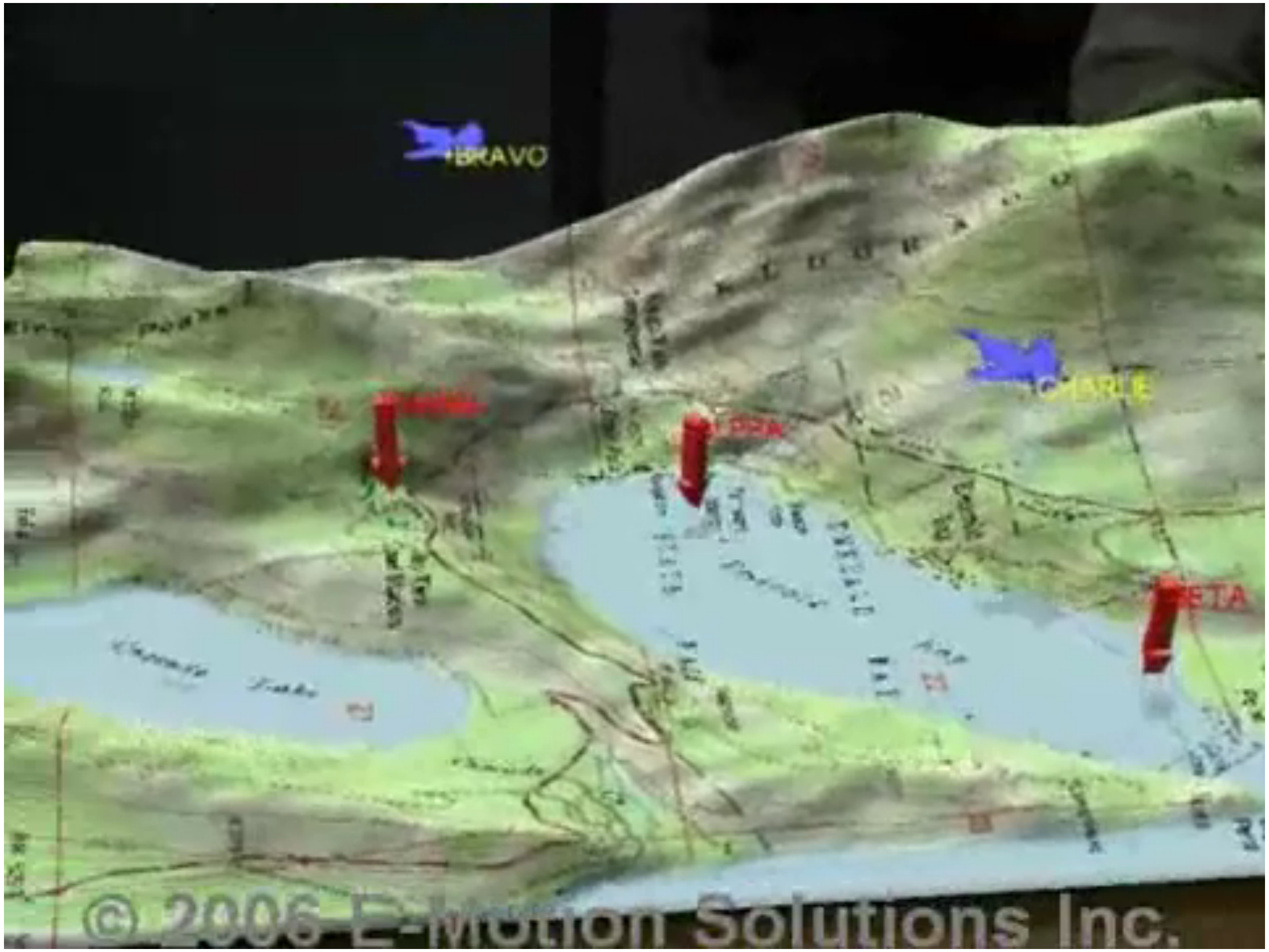
Cultural heritage



IRISA

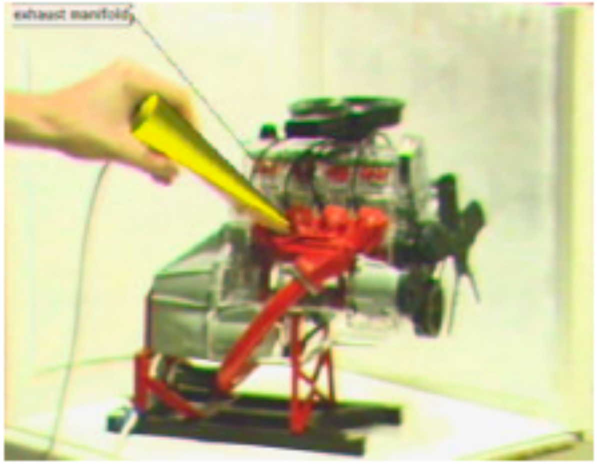
Military applications

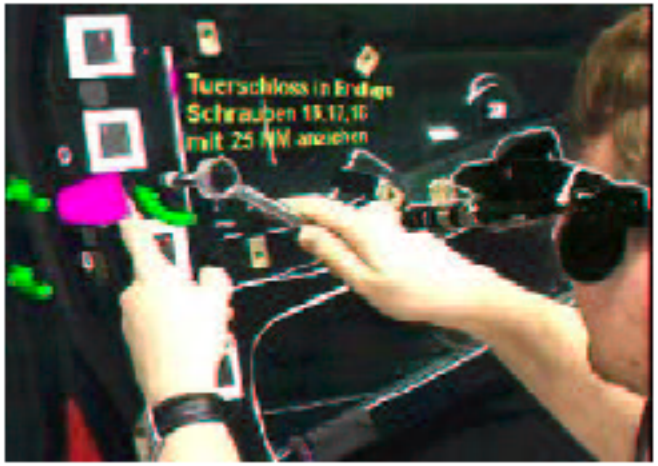
Military applications



Design intérieur

Industrial application, assembly

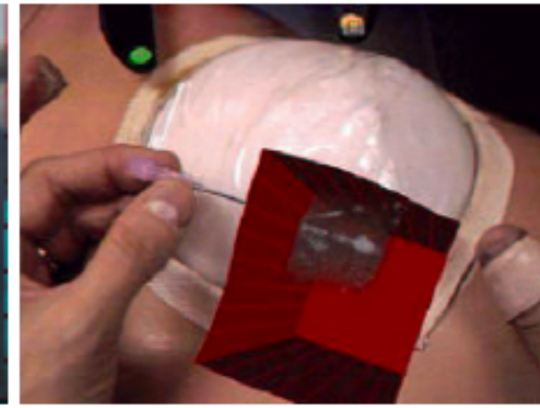


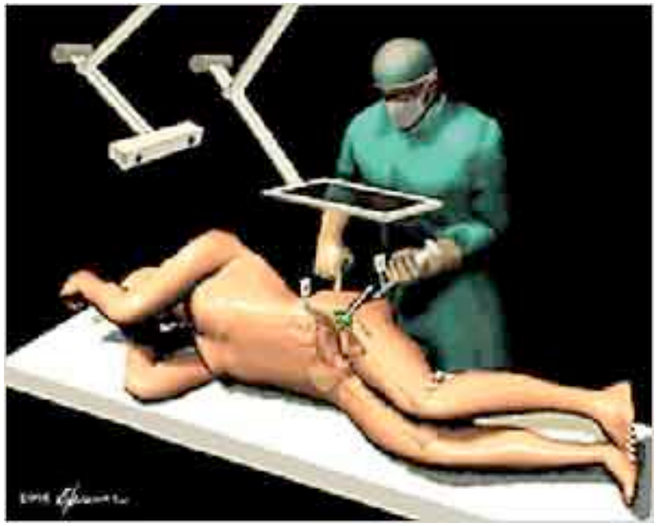


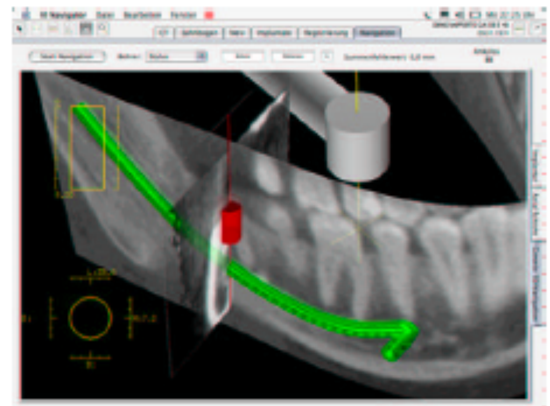
IRISA

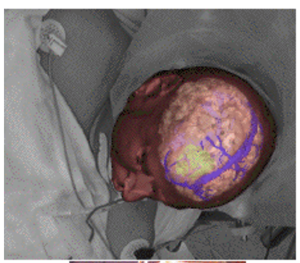
UMR IRISA

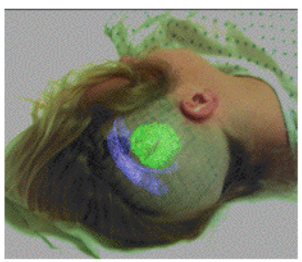
AR in medecine

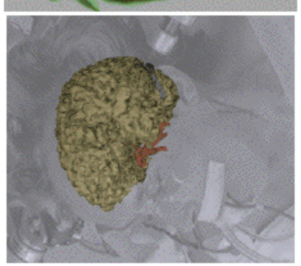








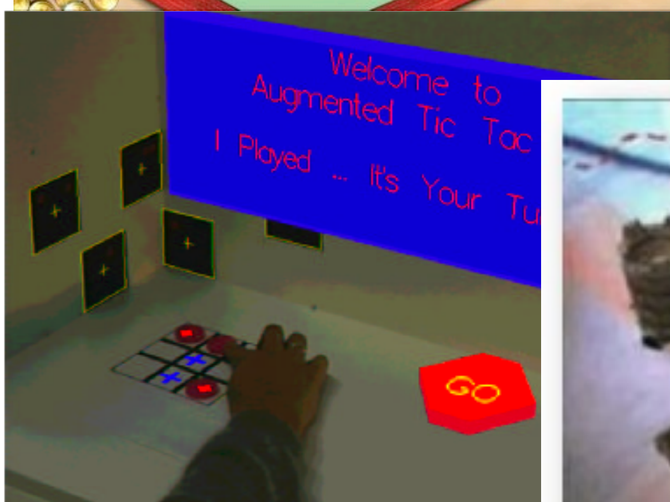


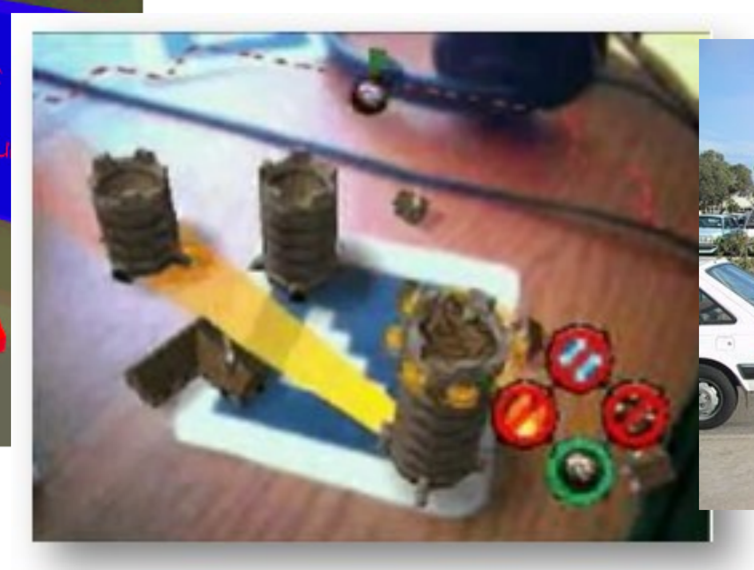


Game

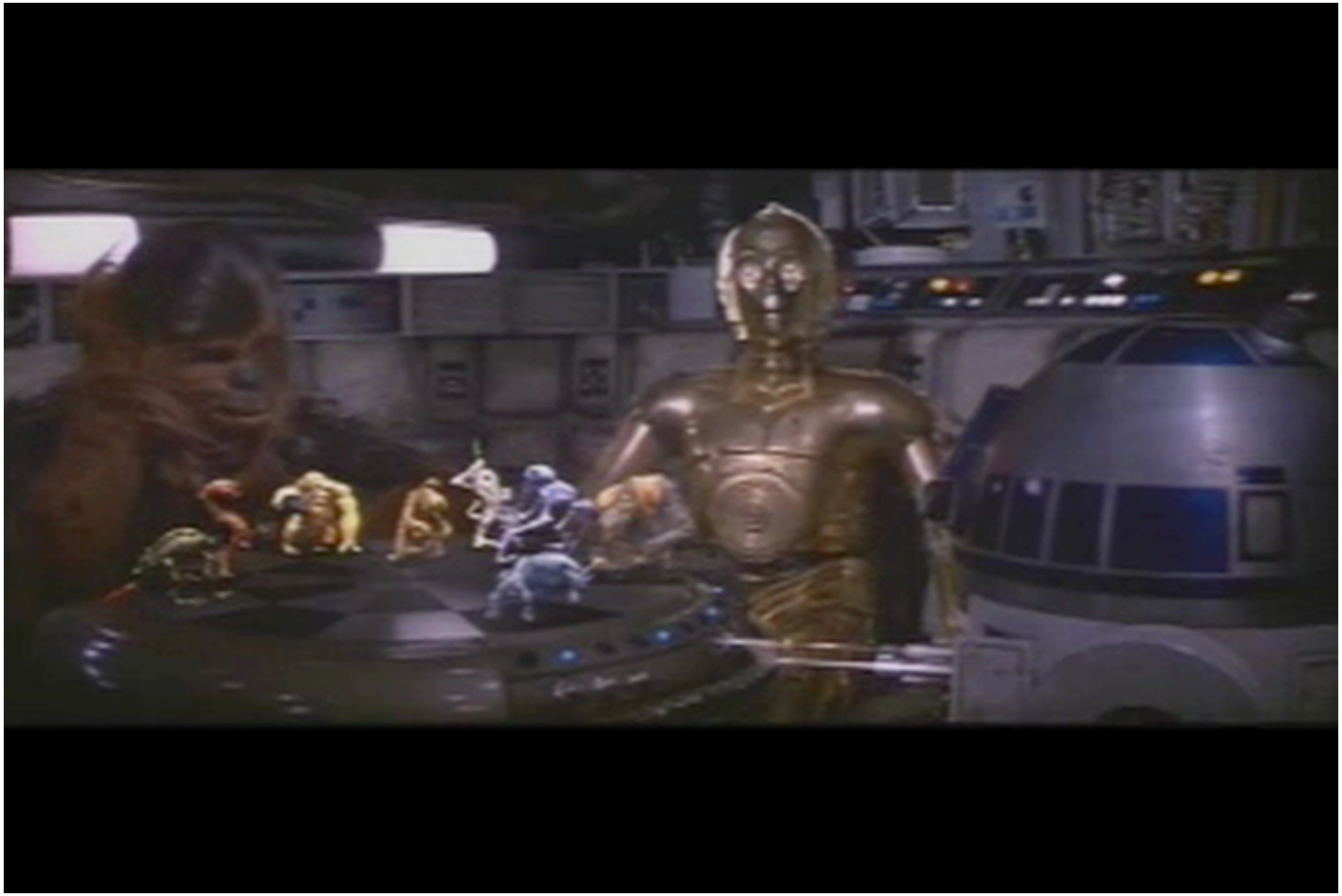

Will
$1,280
Ella
$943
$1,535
Ryan
PARKING
$992
PLAY3-LIVE


ARQuake
Outdoor Augmented Reality Gaming
Wearable Computer Lab
University of South Australia
http://wearables.unisa.edu.au
August 2002

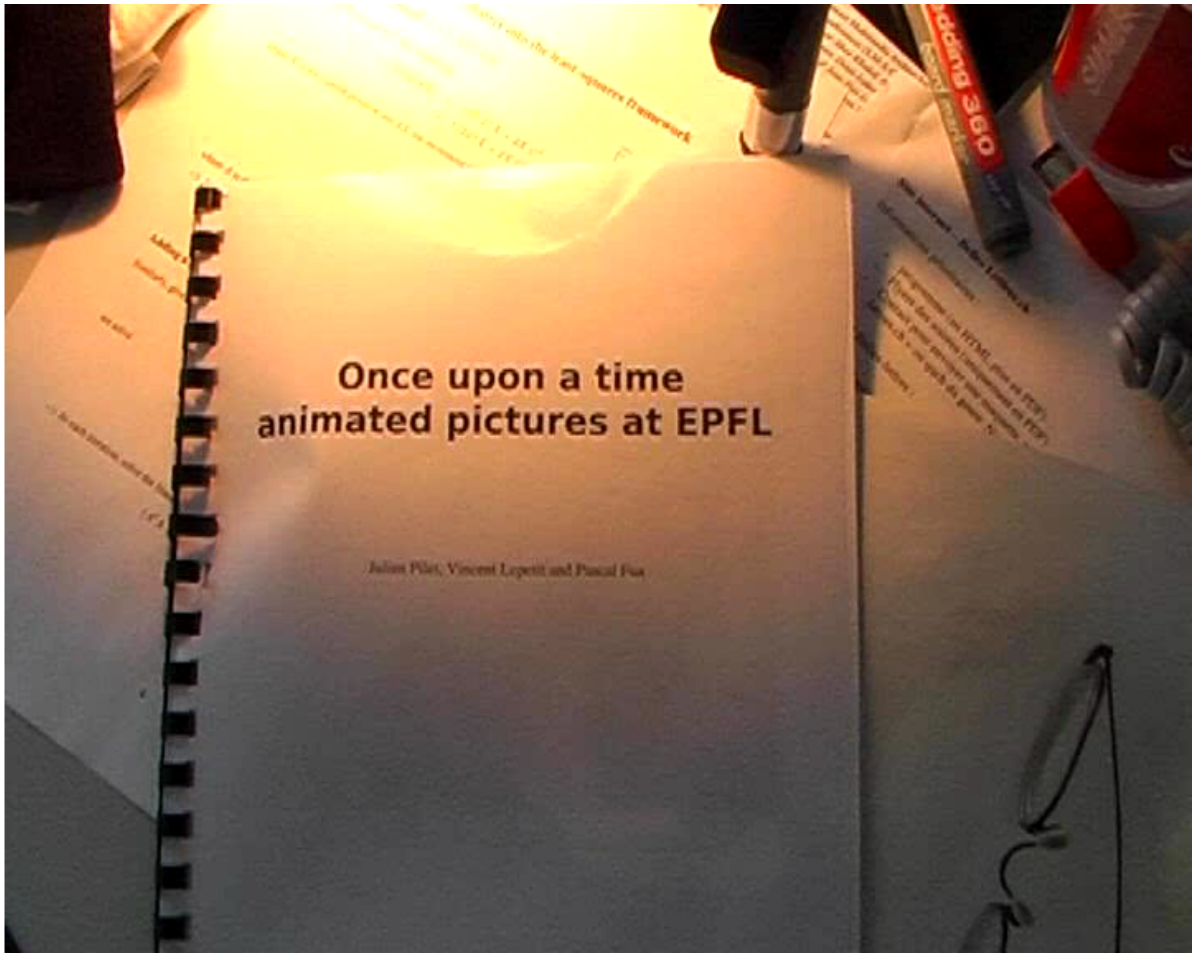

Welcome to
Augmented Tic Tac
I Played ... It's Your Tu
GO



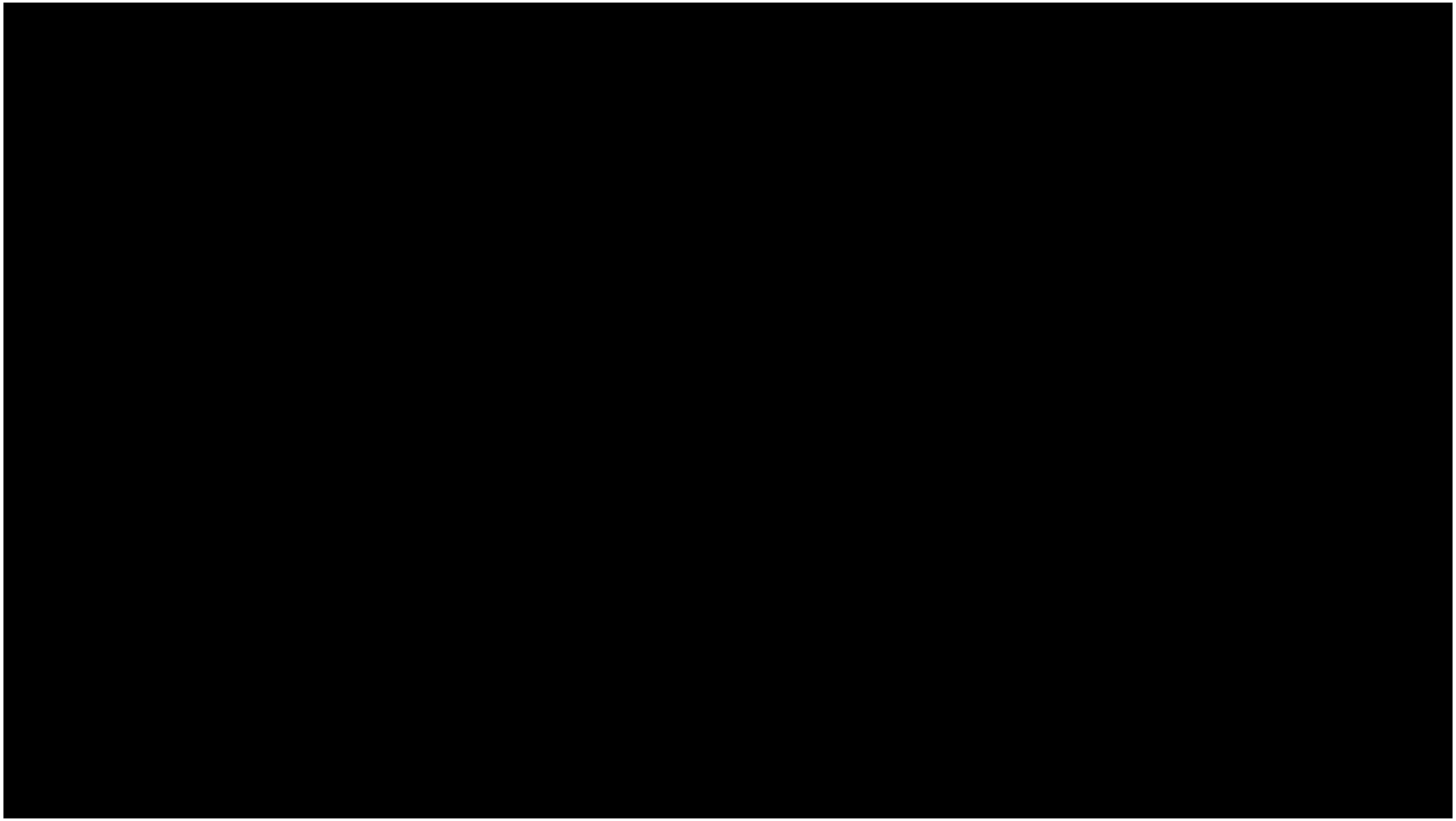
Game from outer space…



Augmented book



No comments…



Application to augmented reality

Augmented reality

- Coherent insertion of virtual objects within real images stream

Augmented reality is handled as a 2D-3D registration issue

- Post production
  - Full knowledge of the video sequence
  - Localization of the camera and structure of the scene
  - Bundle adjustment techniques (Realviz, 2D3)

- On-line augmented reality
  - Real-time requirements
  - No knowledge on the future
  - [Navab][Lepetit-Fua][Berger][Kutulakos],…

So…

Augmented reality is… viewpoint computation

Goal :

- Tracking the camera
  - in a sequential way (video streams, real time)
  - for getting stable and accurate augmentation results

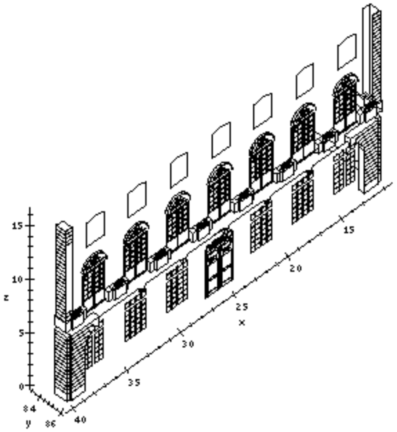
Model-based tracking

Pro

- fast and sequential (real-time)
- no drift

Cons

- the scene has to be partially known (markers, natural features)
  - tedious task of reconstructing or measuring scene features
- difficult to define a lot of features
  - jittering effect



Loria

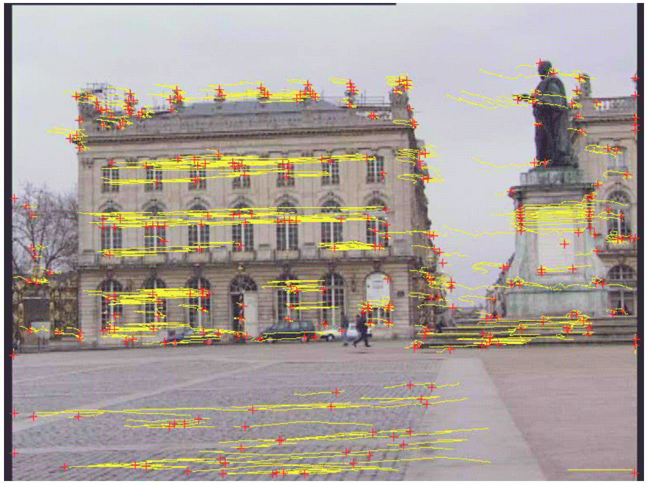
Motion computation

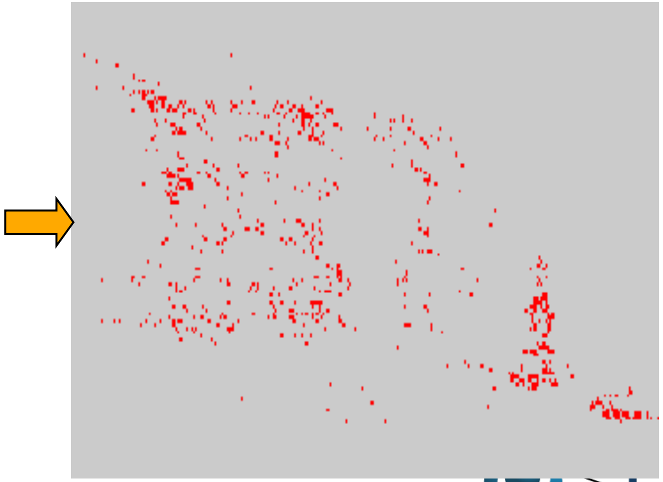
Pro

- do not require any model of the scene
- easy to track a lot of features + bundle adjustment
  - very accurate registration, negligible jitter

Cons

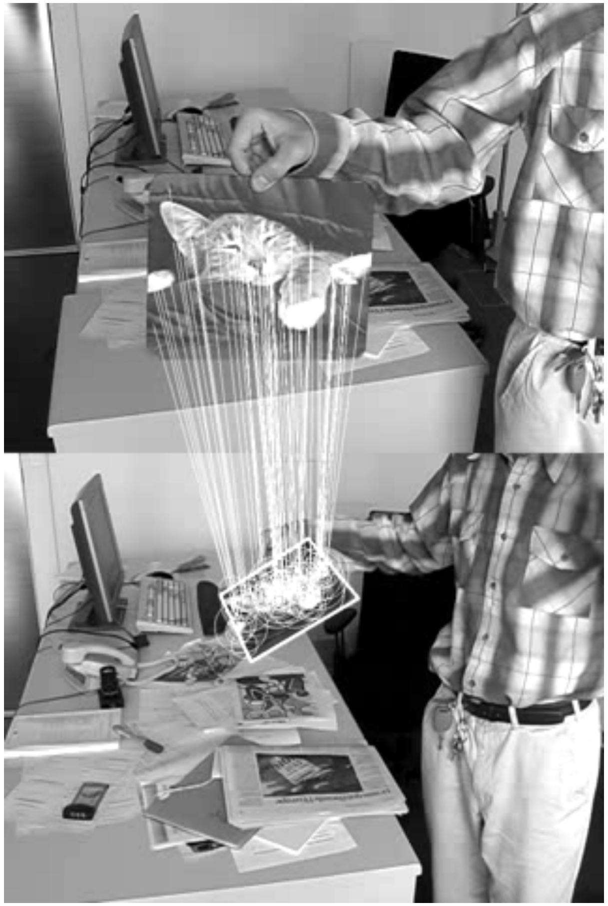
- slow and not sequential
- The world and the virtual coordinate system must be aligned manually





Tracking by matching

Establish the correspondance between primitives extracted from 2 images



Pro

- No tracking and then no real (re)initialization

Cons

- Viewpoint change (3D change)
- Image transformation (translation, rotation, scale change)
- Illumination change, Occlusion
- Low frame rate ?

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

UMR IRISA

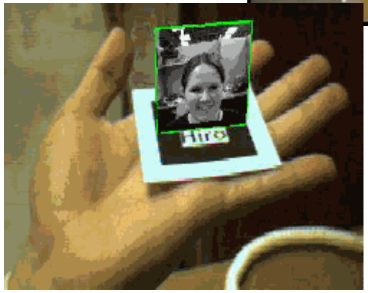
What is not in the scope of this talk

Visualization devices

- HMD
- Video see through, optical see through

Augmented reality for post production

AR toolkit



Rendering

- Image synthesis
- Lighting considerations

UMR IRISA

First,… camera model

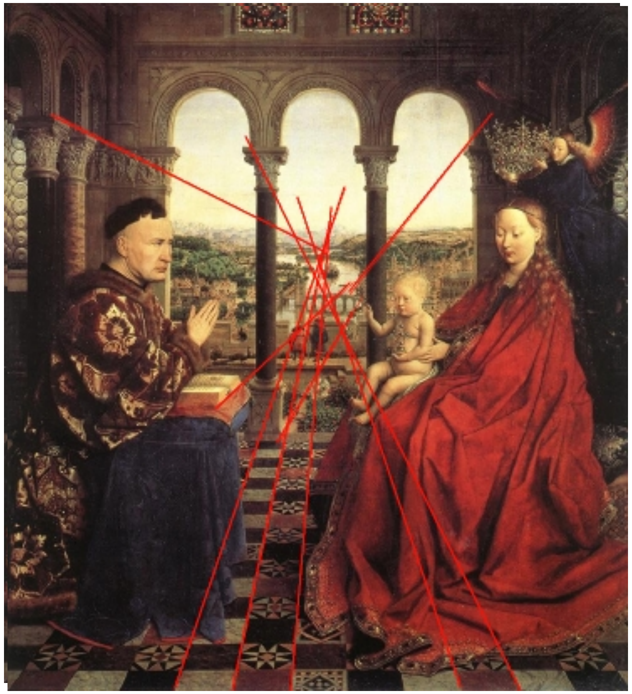
Perspective discovery

Ancient egypt, Bayeux tapestry (11e), Lorenzetti, les effets du bon gouvernement, 1340

« Jesus Before the Caïf », by Giotto (1305). The ceiling rafters show the Giotto's introduction of convergent perspective.

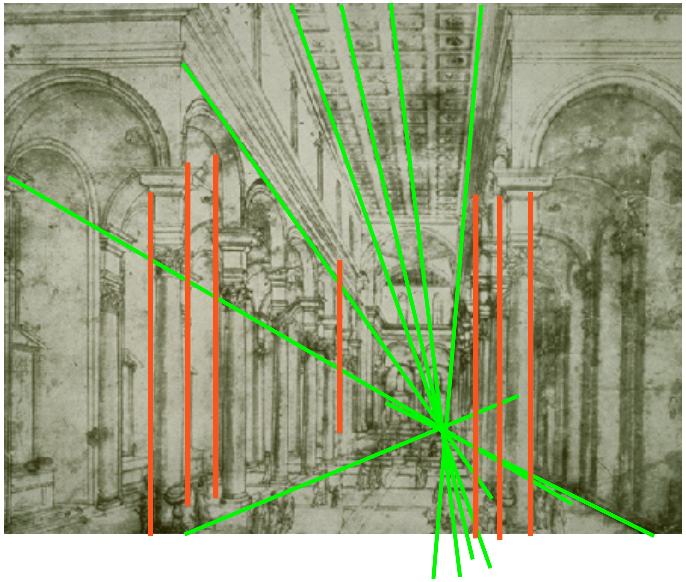
Detailed analysis, however, reveals that the ceiling has an inconsistent vanishing point and that the Caïf's dais is in parallel perspective, with no vanishing point.

Perspective discovery



Van Eyck, 1435 (flandre)

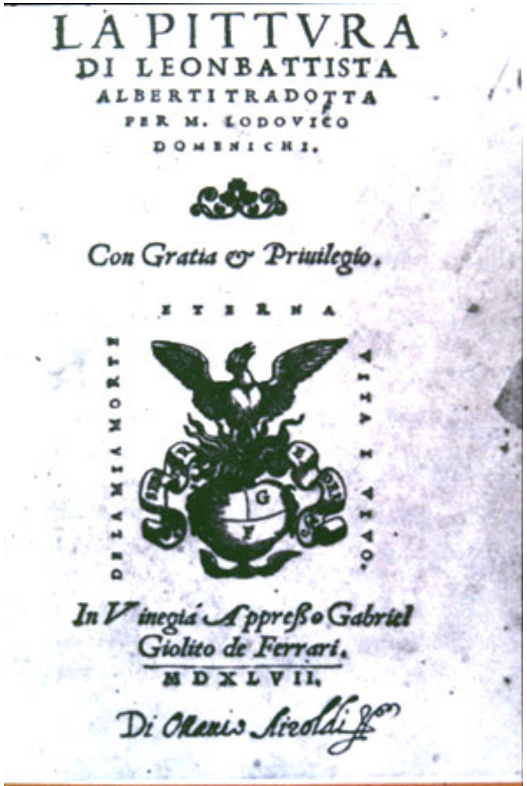
Brunelleschi: Santa Maria della Fiore, 1435



Brunelleschi, début 15ème

Alberti, *della pittura* 1435

As Brunelleschi made no written record of his perspective findings, it remained for Alberti to be the first to put the theory into writing, in his treatise on painting, *Della pittura (1435).* There, Alberti gave practical information for painters and advice on how to paint istoria or history paintings.



LA PITTVRA
DI LEONBATTISTA
ALBERTI TRADOTTA
PER M. LODOVICO
DOMENICHI.

Con Gratia & Priuilegio.

ETERNA

In Vinegia Appresso Gabriel
Giolito de Ferrari.
MDXLVII.

« Herod’s Feast » by Masolino (1435), where many receding horizontal lines project to a single central vanishing point

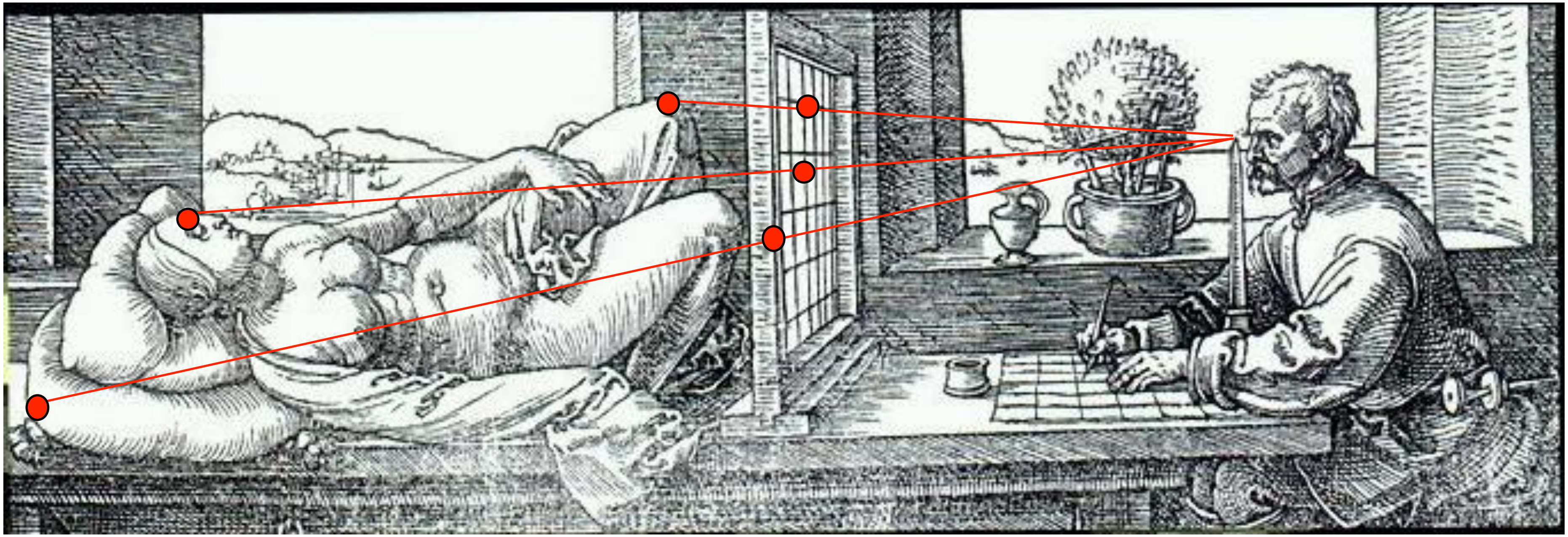
Three examples of paintings in one-point perspective with laterally shifted vanishing points.

- Left panel. « The Annunciation » by Fra Angelico (1436-1443)
- Central panel: « Presentation of the Virgin » by Fra Carnevale (1467)
- right panel: « The Vision of St. Catherine » by Titian (1503?)

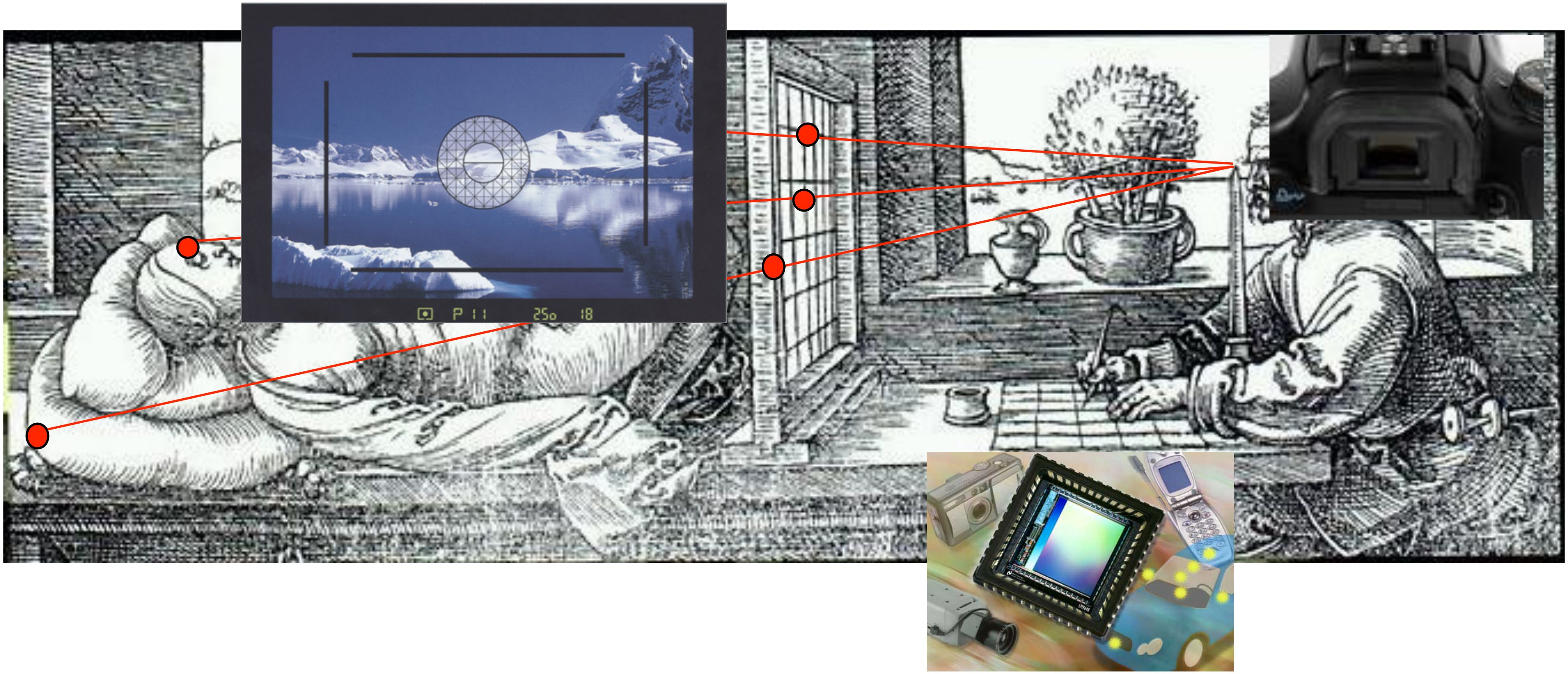
Dürer perspective device

Albrecht Dürer: Artist Drawing a Nude with Perspective Device 1525

Dürer perspective device



Dürer perspective device



Dürer perspective device

Camera obscura

Camera obscura (Cam"e*ra ob*scu"ra) [LL. camera chamber + L. obscurus, obscura, dark.] (Opt.)

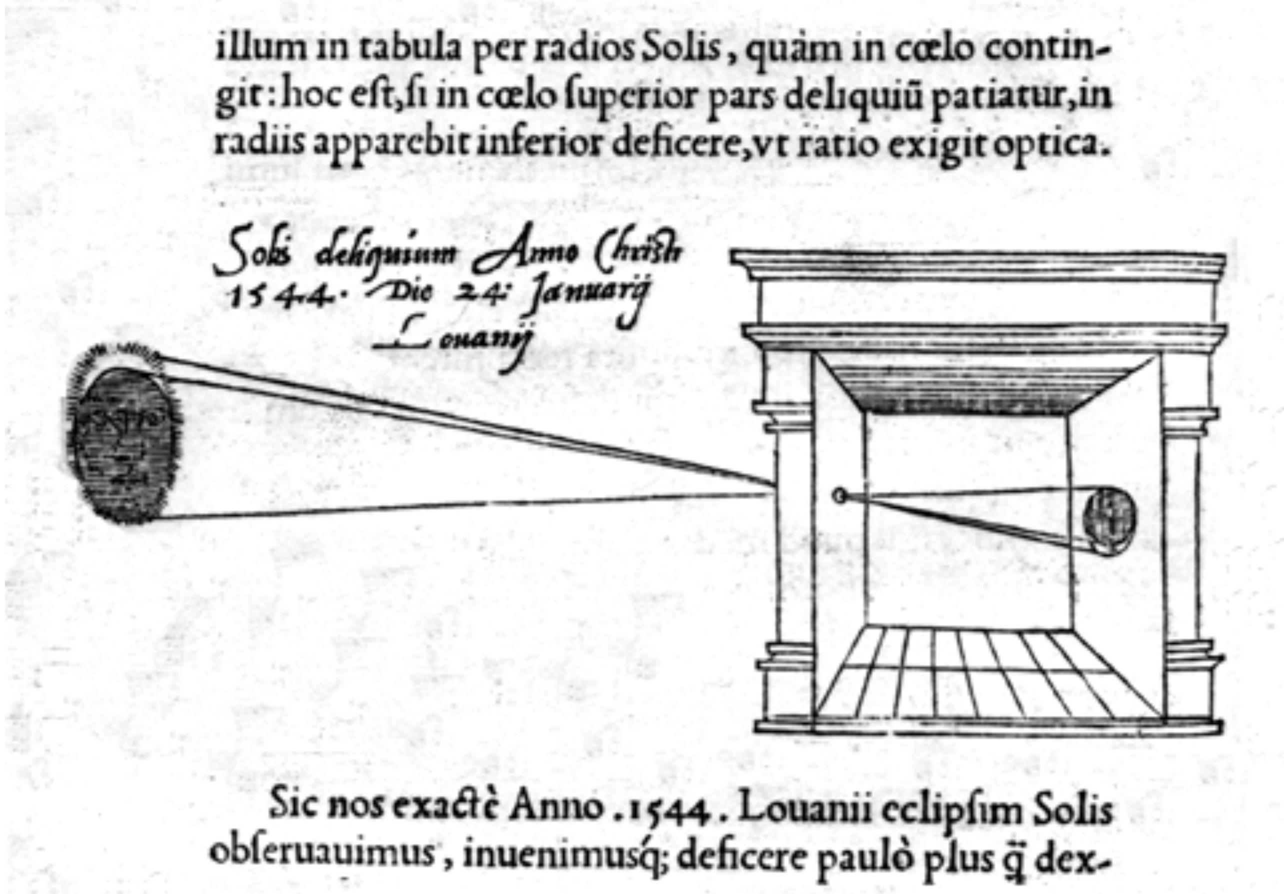
- An apparatus in which the images of external objects, formed by a convex lens or a concave mirror, are thrown on a paper or other white surface placed in the focus of the lens or mirror within a darkened chamber, or box, so that the outlines may be traced.

- (Photog.) An apparatus in which the image of an external object or objects is, by means of lenses, thrown upon a sensitized plate or surface placed at the back of an extensible darkened box or chamber variously modified; - commonly called simply the camera.

Websters Dictionary, 1913

Camera Obscura

De radio Astronomica et Geometrica, Gemma Frisius 1544



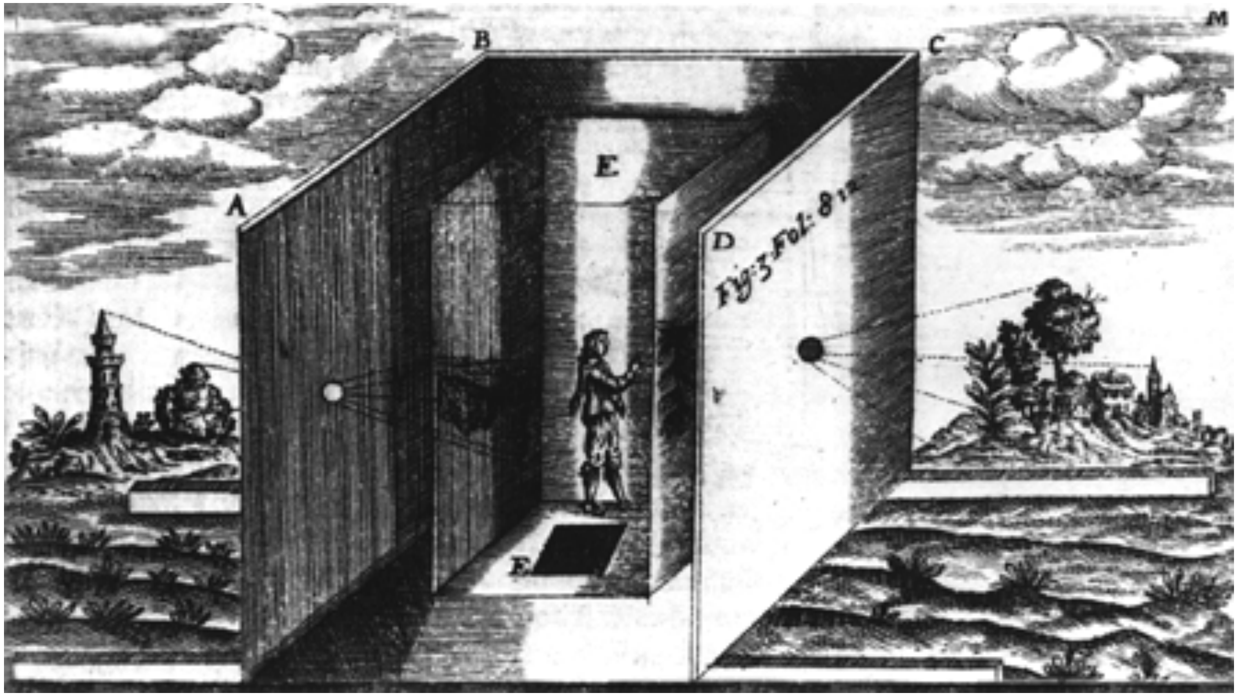
illum in tabula per radios Solis, quàm in cœlo contin-
git: hoc eſt, ſi in cœlo ſuperior pars deliquiũ patiatur, in
radiis apparebit inferior deficere, vt ratio exigit optica.

Sic nos exactè Anno .1544. Louanii eclipſim Solis
obſeruauimus, inuenimusq; deficere paulò plus q̃ dex-

Camera obscura

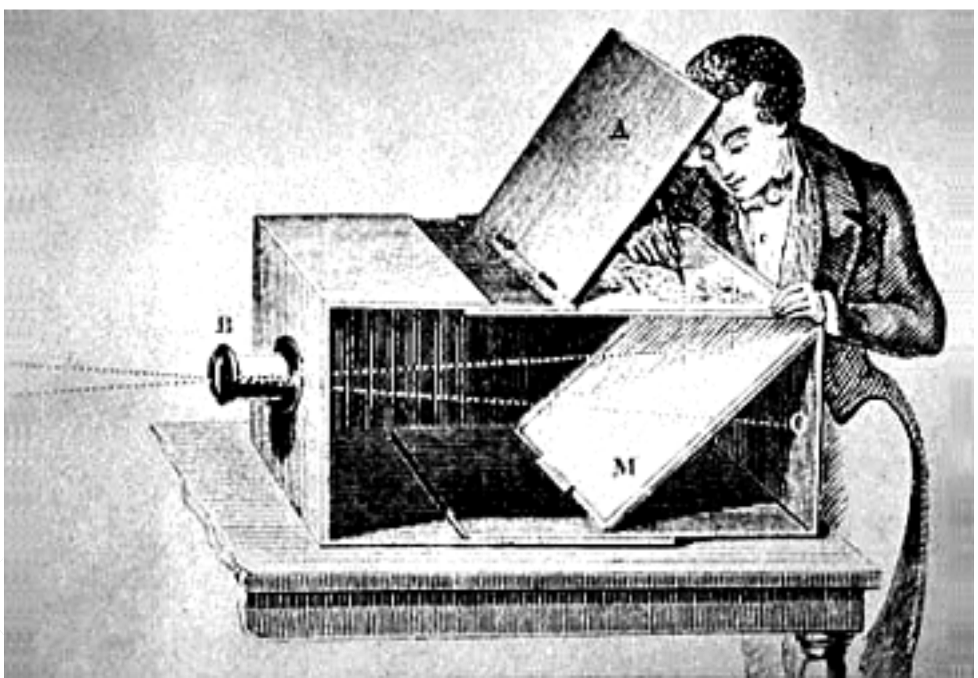
Camera Obscura, Athanasius Kircher, 1646

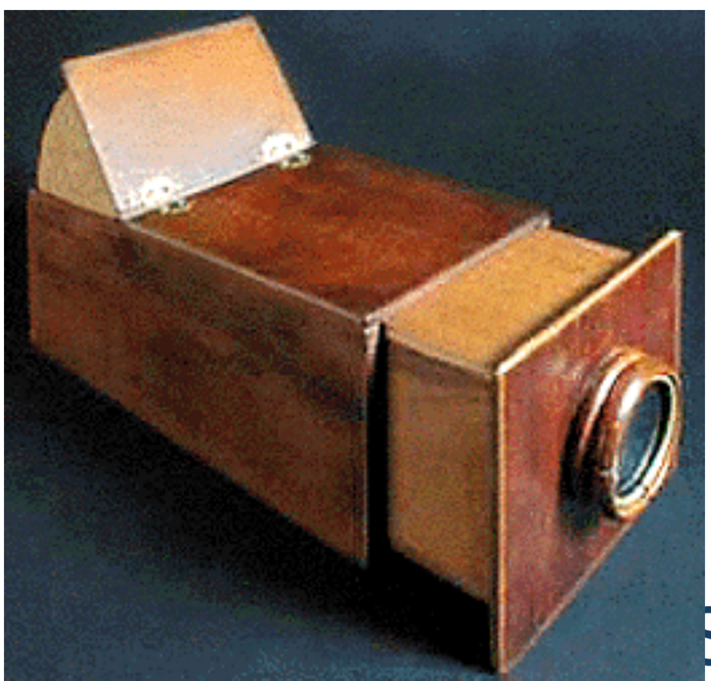
- In Gernsheim, H., The Origins of Photography



Camera obscura

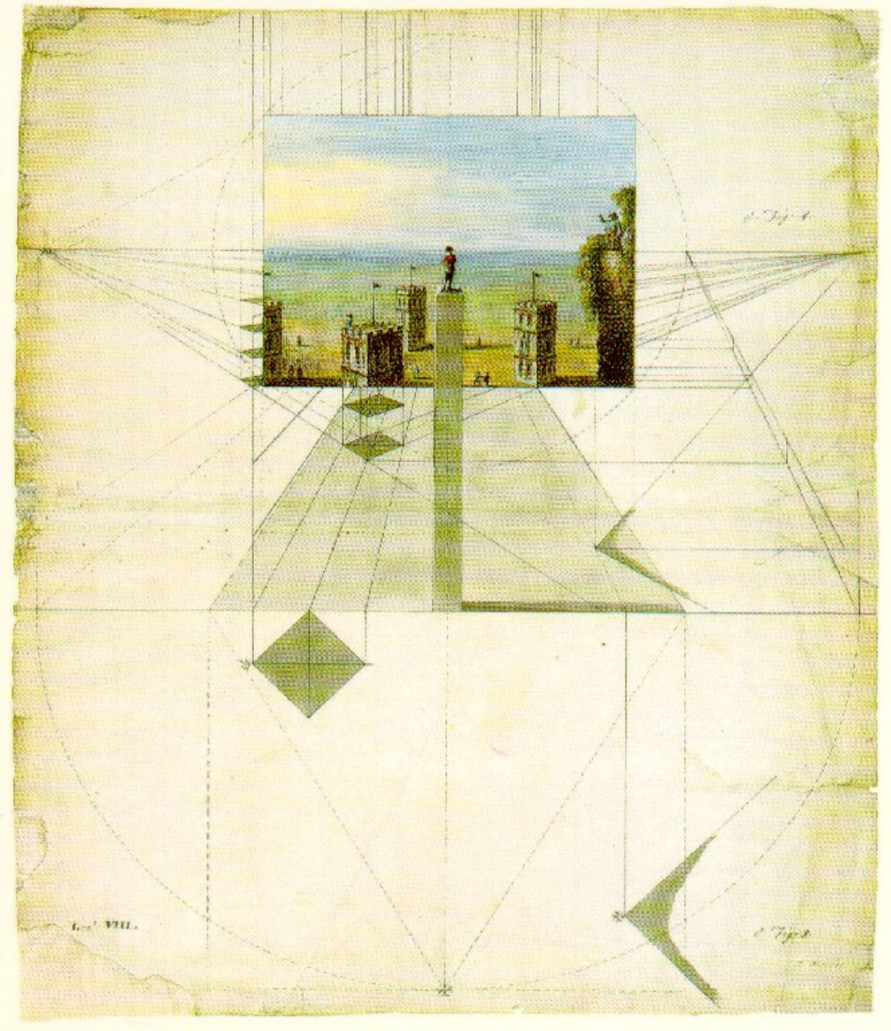
- Invented in the sixteenth century, the camera obscura is made out of an arrangement of lenses and mirrors in a box that is darkened, The machine permits accuracy in a drawing, often of topographical detail. When looking through the lens of a camera obscura, the view presented is actually reflected through the mirrors onto the paper or cloth and allows the artist to draw by tracing the outline.





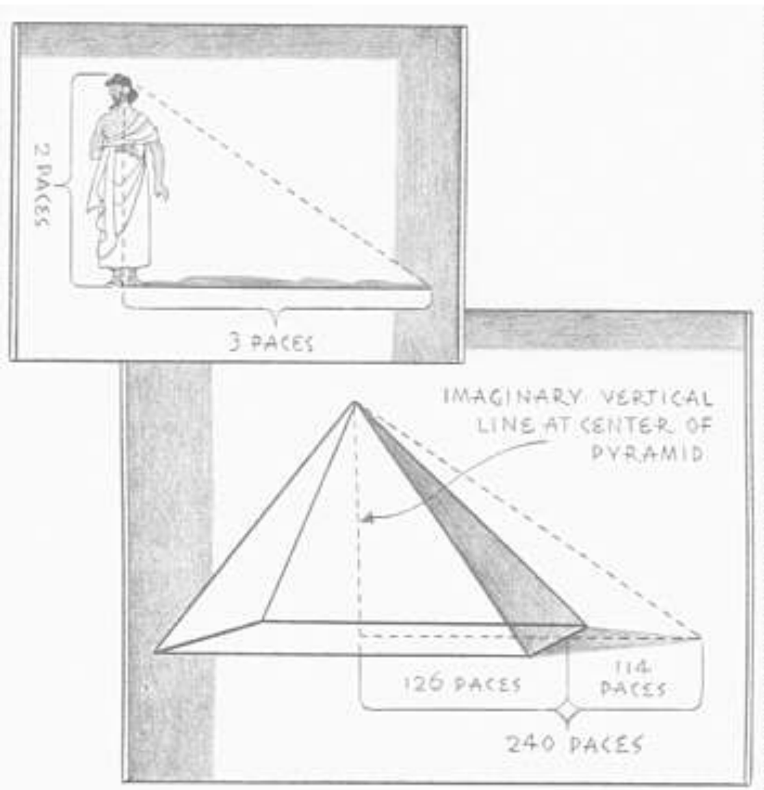
UMR IRISA

Perspective projection



Perspective projection: back to basic

Thales Theorem (625-546 Av JC)



Height of Pyramid (Imaginary Post) =

$$\text{SHADOW OF IMAGINARY POST} \times \frac{\text{Thales' Height}}{\text{Thales' Shadow}}$$

HEIGHT OF PYRAMID =

$$(\tfrac{1}{2}\text{ its Base} + \text{its Shadow}) \times \frac{\text{Thales' Height}}{\text{Thales' Shadow}}$$

$$Hp = (126\text{ paces} + 114\text{ paces}) \times \frac{2\text{ paces}}{3\text{ paces}}$$

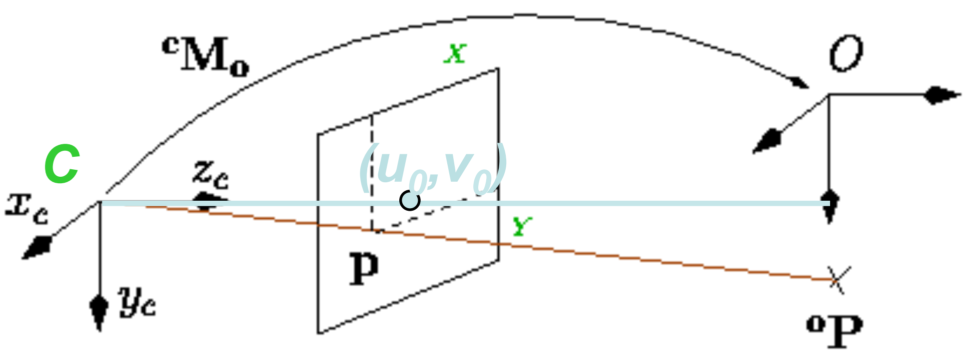
$$Hp = 240 \times 2/3 = 160\text{ paces!}$$

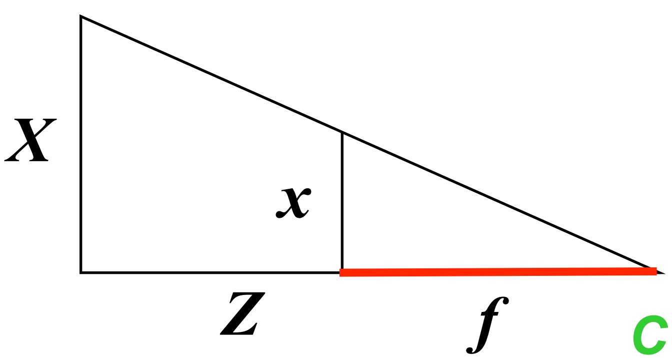
Perspective projection

Definitions

- The origin of the camera frame *Rc* is the center of projection *C*
- x axes is parallel to image lines and y axes is parallel to image columns

- Intersection of z axes with image plane is the principal point u_0, v_0
- focal distance $f = d(C,\pi)$





Perspective projection

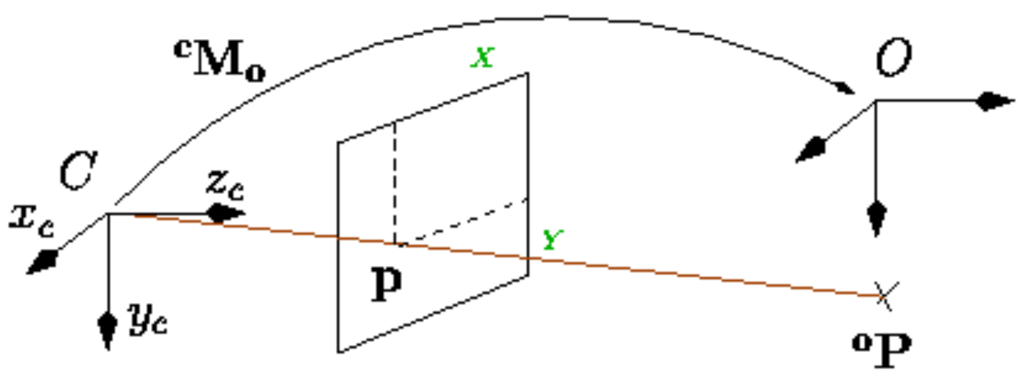
In *Rc*, the perspective projection of a point $M(X,Y,Z)$ on the image point $m(x,y)$ with:

$$x = f\,X \,/\, Z \;,\; Y = f\,Y \,/\, Z$$

All the points on the CM straight line given by:

$$\begin{cases} fX - Zx = 0 \\ fY - Zy = 0 \end{cases}$$

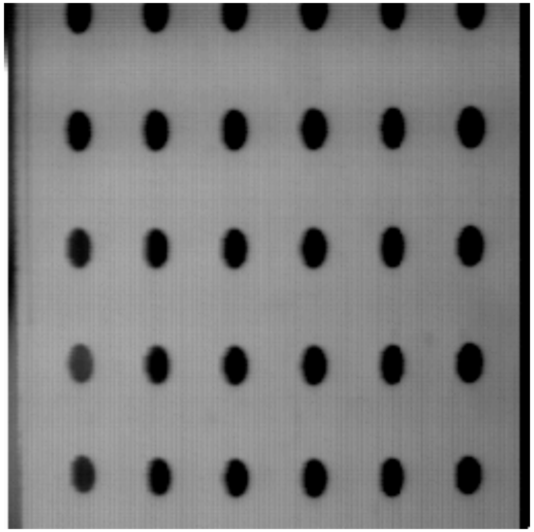
are projected on the same point m. It is impossible to determine the position of M without *a priori* knowledge with one camera.



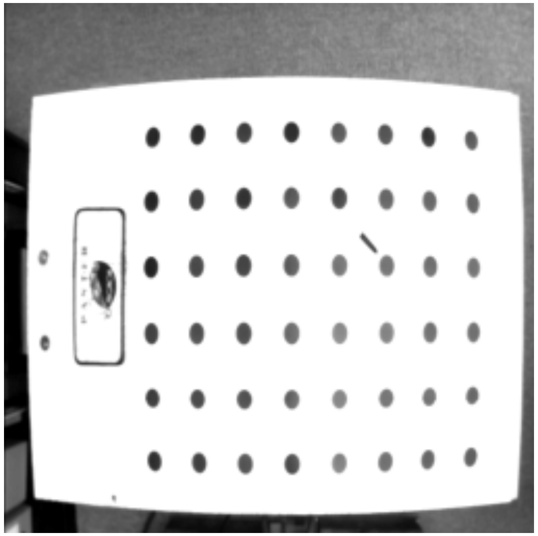
Radial distortion

Let K be the radial distortion coefficient. Then, position of point m that is observed int the image is given by:

$$x_d = x + Kx(x^2 + y^2)\ ,\ y_d = y + Ky(x^2 + y^2)$$



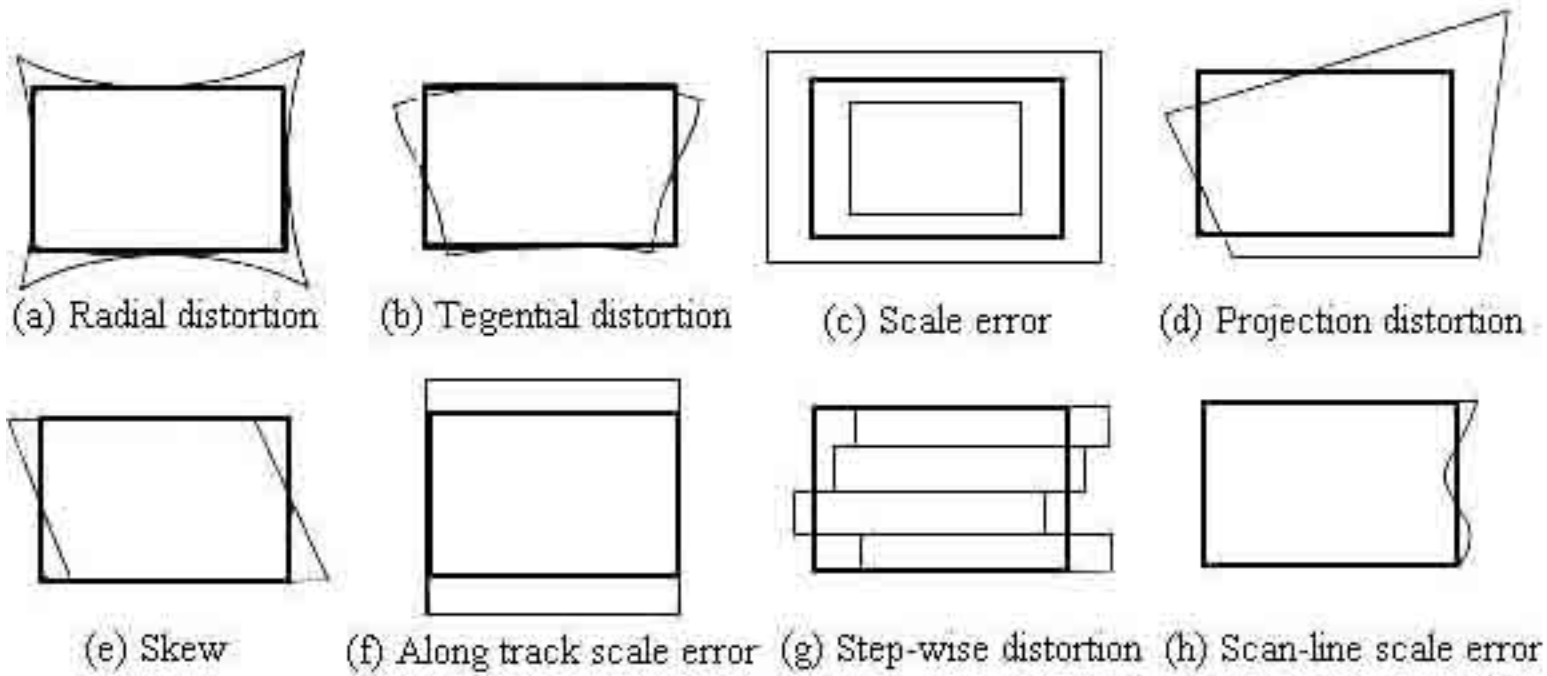
f=12mm



f=4.8mm

UMR IRISA

Other potential distortions

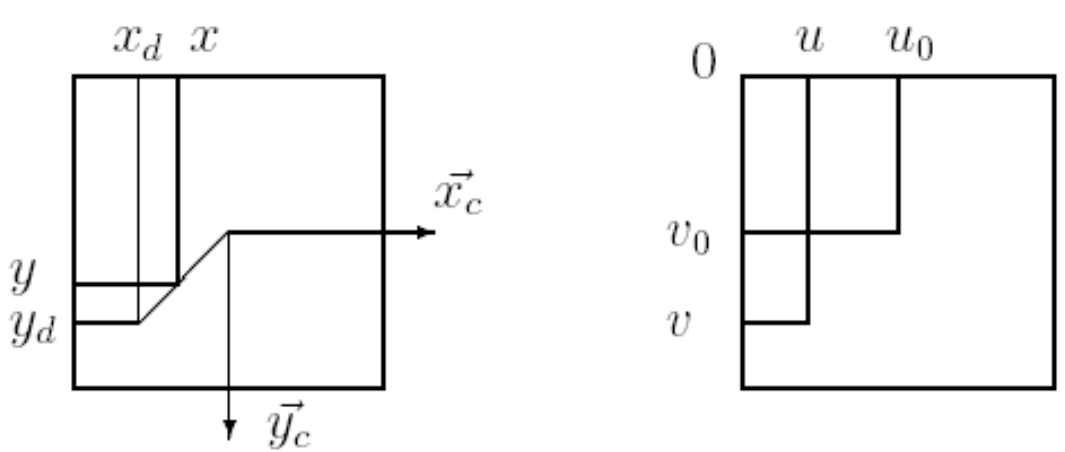


Sampling

Let us define m_d by it position m_p (u,v) in the digitized image d (expressed in pixel):

$$u = u_0 + \frac{x_d}{l_x} \; , \; v = v_0 + \frac{y_d}{l_y}$$

where l_x and l_y are the pixel size and u_0, v_0 are the translation of the center of the coordinate system (principal point coordinates)



Complete camera model

The model is given by

$$\begin{cases} u = u_0 + p_x \frac{X}{Z} + K_d p_x \frac{X}{Z}\left(\frac{X^2}{Z^2} + \frac{Y^2}{Z^2}\right) \\ v = v_0 + p_y \frac{Y}{Z} + K_d p_y \frac{Y}{Z}\left(\frac{X^2}{Z^2} + \frac{Y^2}{Z^2}\right) \end{cases}$$

where

$$p_x = f/l_x, \;\; p_y = f/l_y \;\; \text{et} \;\; K_d = K f^2$$

The unknown parameters ξ, called intrinsic parameters are:

$$\xi = (u_0, v_0, p_x, p_y, K_d)$$

Camera model

When camera distortion can be neglected, the camera model is simply given by:

$$\begin{cases} u = u_0 + p_x \frac{X}{Z} \\ v = v_0 + p_y \frac{Y}{Z} \end{cases}$$

UMR IRISA

Camera model

After calibration, we can get normalized coordinated from digitized ones using:

$$\begin{cases} x = (u - u_0)/p_x \\ y = (v - v_0)/p_y \end{cases}$$

projection equations are then given by

$$x = \frac{X}{Z}\ ,\ y = \frac{Y}{Z}$$

If K_d cannot be neglected, we can undistort the image

Undistortion

Perspective model: linear notation

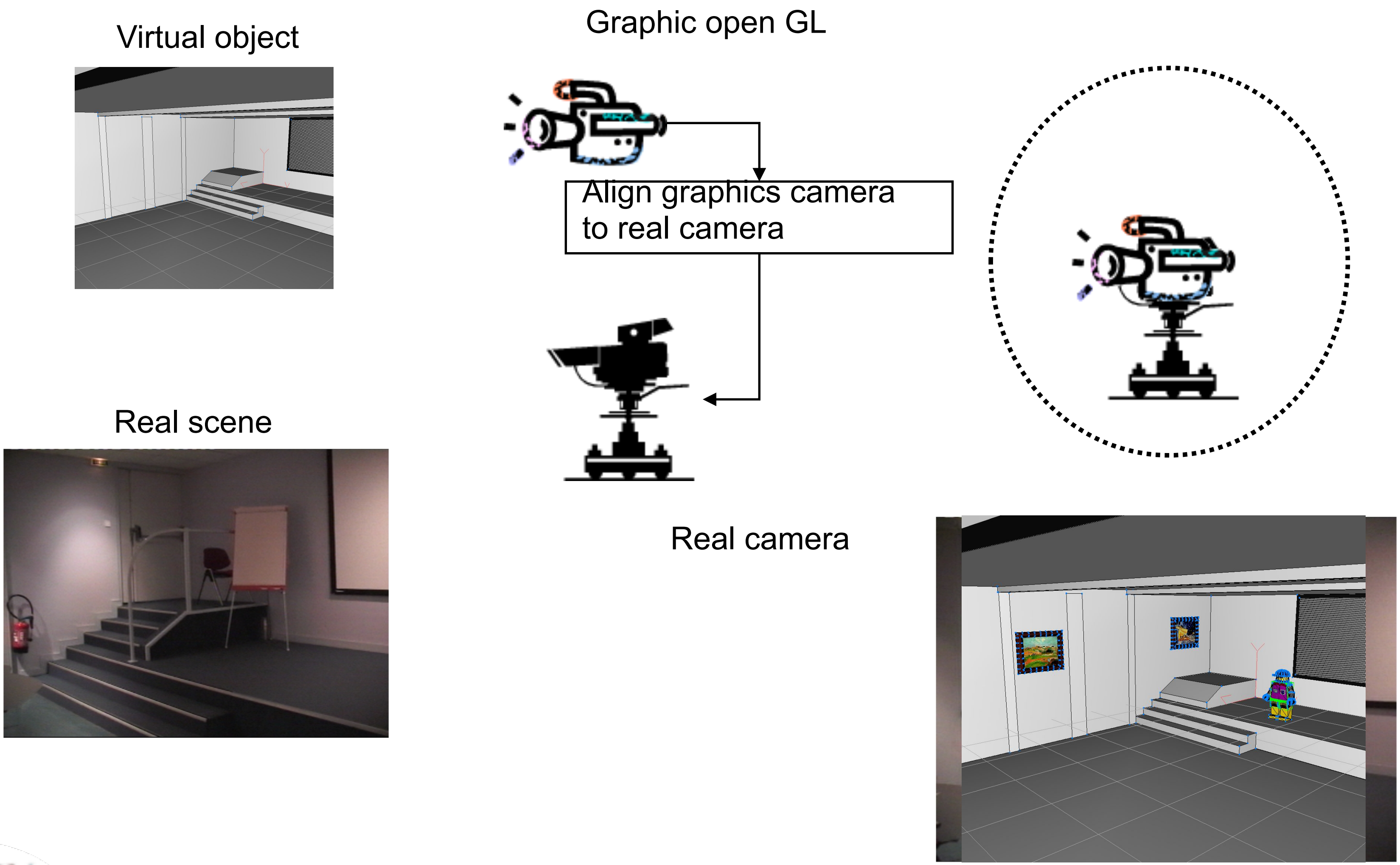
$$\begin{cases} u = u_0 + p_x x \\ v = v_0 + p_y y \end{cases}$$

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \begin{pmatrix} p_x & 0 & u_0 \\ 0 & p_y & v_0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

$$= \underbrace{\begin{pmatrix} p_x & 0 & u_0 \\ 0 & p_y & v_0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}}_{\mathbf{K}} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}$$

Second,… Model-based tracking

Camera alignment for augmented reality



3D localization, pose estimation

Goal

- Determine 3D camera camera location wrt. an object using only one image of this object

Then

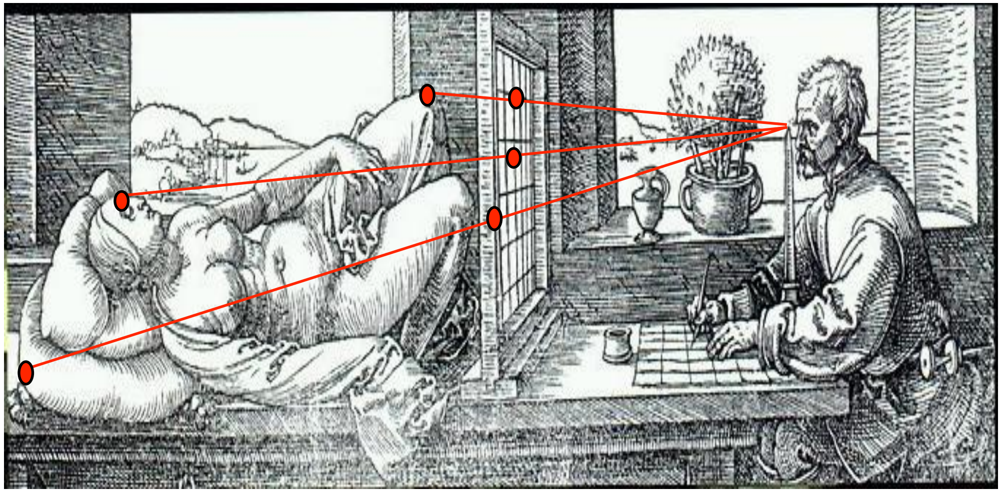
- With no a priori knowledge, localization in impossible
- Position of specific features have to be known in an object related frame

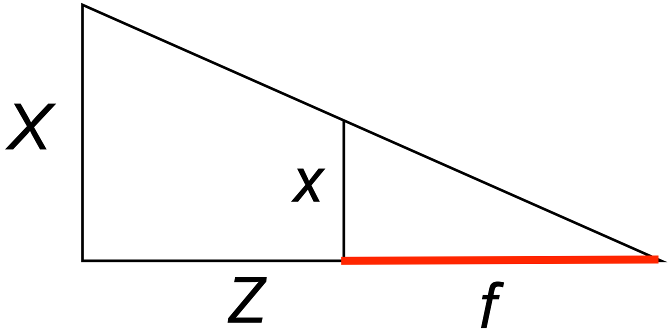
Approach: similar to calibration

- Simplification of the Toscanis-Faugeras method
- Dementhon-Davis method
- Non-linear minimization

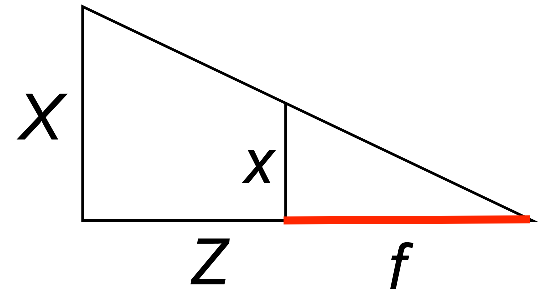
Middle school geometry

Théorème de Thalès





Generalization



We know (x,y) and the object model ${}^o\mathbf{P}$

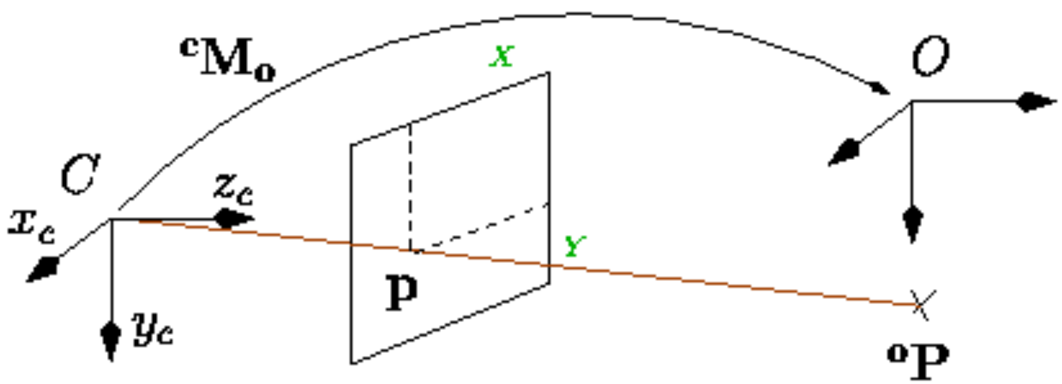
We seek the pose ${}^c\mathbf{M}_o$

Solution is quite simple : change frame first

$$
{}^c\mathbf{P} = {}^c\mathbf{M}_o {}^o\mathbf{P} \Leftrightarrow \begin{cases} {}^c\mathbf{X} = (\mathbf{r}_1\ \ 0)\,{}^o\mathbf{P} + t_x \\ {}^c\mathbf{Y} = (\mathbf{r}_2\ \ 0)\,{}^o\mathbf{P} + t_y \\ {}^c\mathbf{Z} = (\mathbf{r}_3\ \ 0)\,{}^o\mathbf{P} + t_z \end{cases}
$$

Then project

$$
\begin{cases} x = \dfrac{(\mathbf{r}_1\ \ 0)\,{}^o\mathbf{P} + t_x}{(\mathbf{r}_3\ \ 0)\,{}^o\mathbf{P} + t_z} \\[2ex] y = \dfrac{(\mathbf{r}_2\ \ 0)\,{}^o\mathbf{P} + t_y}{(\mathbf{r}_3\ \ 0)\,{}^o\mathbf{P} + t_z} \end{cases}
$$



UMR IRISA

"Linear" approach

Let's note the pose ${}^c\mathbf{M}_o$

$$
{}^c\mathbf{P} = {}^c\mathbf{M}_o {}^o\mathbf{P} \Leftrightarrow \begin{cases} {}^c\mathbf{X} = (\mathbf{r_1} \;\; 0){}^o\mathbf{P} + t_x \\ {}^c\mathbf{Y} = (\mathbf{r_2} \;\; 0){}^o\mathbf{P} + t_y \\ {}^c\mathbf{Z} = (\mathbf{r_3} \;\; 0){}^o\mathbf{P} + t_z \end{cases}
$$

The perspective projection equation gives:

$$
\begin{cases} x = \frac{(\mathbf{r_1} \;\; 0){}^o\mathbf{P} + t_x}{(\mathbf{r_3} \;\; 0){}^o\mathbf{P} + t_z} \\ y = \frac{(\mathbf{r_2} \;\; 0){}^o\mathbf{P} + t_y}{(\mathbf{r_3} \;\; 0){}^o\mathbf{P} + t_z} \end{cases}
$$

Which with simple developments leads to:

$$
\begin{cases} r_{31}{}^oXx + r_{32}{}^oYx + r_{33}{}^oZx + xt_x - r_{11}{}^oX - r_{12}{}^oY - r_{13}{}^oZ - t_x = 0 \\ r_{31}{}^oXy + r_{32}{}^oYy + r_{33}{}^oZy + xt_y - r_{11}{}^oX - r_{12}{}^oY - r_{13}{}^oZ - t_x = 0 \end{cases}
$$

UMR IRISA

“Linear” solution

We obtain an homogeneous system with 12 unknowns parameters:

$$\mathbf{A}\mathbf{I} = \begin{pmatrix} \vdots \\ A_i \\ \vdots \end{pmatrix} \mathbf{I} = \mathbf{0} \qquad \text{avec}$$

Where

- **A** depends on the data extracted from the image
- **I** is function of the parameters to be estimated

Each point of the object gives 2 equations $\mathbf{A}_i\, \mathbf{I} = \begin{bmatrix} 0 & 0 \end{bmatrix}^T$

$$\mathbf{A}_i = \begin{pmatrix} -{}^oX_i & -{}^oY_i & -{}^oZ_i & 0 & 0 & 0 & x_i{}^oX_i & x_i{}^oY_i & x_i{}^oZ_i & -1 & 0 & x_i \\ 0 & 0 & 0 & -{}^oX_i & -{}^oY_i & -{}^oZ_i & y_i{}^oX_i & y_i{}^oY_i & y_i{}^oZ_i & 0 & -1 & y_i \end{pmatrix}$$

UMR IRISA

Solution of the linear system

System to be solved **AI = 0**

- Where **I** is a non null vector of size *2n*

Solution 1

- Compute **I** a vector of the null space of **A** (SVD)

Solution 2

- Considering that r_{ij} is a rotation matrix
- Solved the system under the constraint that $[r_{31}, r_{32}, r_{33}]$ is a unitary vector

Constrained minimization

$$\mathbf{AX_1} + \mathbf{BX_2} = 0 \quad \text{under the constraint that} \quad \|\mathbf{X_1}\| = 1$$

With

$$\mathbf{X_1} = (r_{31}, r_{32}, r_{33})^T$$
$$\mathbf{X_2} = (r_{11}, r_{12}, r_{13}, r_{21}, r_{22}, r_{23}, t_x, t_y, t_z)^T$$
$$\mathbf{A} = \begin{pmatrix} xX_o & xY_o & xZ_o \\ yY_o & yY_o & yZ_o \end{pmatrix}$$
$$\mathbf{B} = \begin{pmatrix} -^oX & -^oY & -^oZ & 0 & 0 & 0 & -1 & 0 & x \\ 0 & 0 & 0 & -^oX & -^oY & -^oZ & 0 & -1 & y \end{pmatrix}$$

UMR IRISA

Constrained minimization

A direct solution is impossible $\mathbf{I} = \mathbf{0}$

We consider a minimization with Lagrangian

System can be rewritten as

$$\mathbf{A}.\mathbf{X_1} + \mathbf{B}.\mathbf{X_2} = 0 \text{ avec } \mathbf{X_1} = (r_{31}, r_{32}, r_{33})^T$$

We minimize the following criterion :

$$C = \|\mathbf{A}.\mathbf{X_1} + \mathbf{B}.\mathbf{X_2}\|^2 + \lambda\left(1 - \|\mathbf{X_1}\|^2\right)$$

where

- $\mathbf{X}_1$ is a line of ${}^c\mathbf{R}_o$
- $\mathbf{X}_2$ is a function of the pose
- $\mathbf{A}$ and $\mathbf{B}$ are function of the *N* measures (x_i, y_i and ${}^oX_i, {}^oY_i, {}^oZ_i$)

Constrained minimization

Let the partial derivatives of *C* be null :

$$\begin{cases} \frac{1}{2}\frac{\partial C}{\partial \mathbf{X_1}} = \mathbf{A}^T\mathbf{A}.\mathbf{X_1} + \mathbf{A}^T\mathbf{B}.\mathbf{X_2} - \lambda\mathbf{X_1} = 0 \\ \frac{1}{2}\frac{\partial C}{\partial \mathbf{X_2}} = \mathbf{B}^T\mathbf{A}.\mathbf{X_1} + \mathbf{B}^T\mathbf{B}.\mathbf{X_2} = 0 \end{cases}$$

We obtained:

$$\begin{cases} \mathbf{X_2} = -(\mathbf{B}^T\mathbf{B})^{-1}\mathbf{B}^T\mathbf{A}.\mathbf{X_1} \\ E.\mathbf{X_1} = \lambda\mathbf{X_1} \quad \text{avec} \quad E = \mathbf{A}^T\mathbf{A} - \mathbf{A}^T\mathbf{B}(\mathbf{B}^T\mathbf{B})^{-1}\mathbf{B}^T\mathbf{A} \end{cases}$$

With

- We have $C = \lambda$ if $\mathbf{X}_1$ is a unit eigen vector of $\mathbf{E}$ corresponding to the eigen value λ.
- $\mathbf{X}_1$ is the eigen vector corresponding to the smallest eigen value of $\mathbf{E}$.

Dementhon-Davis method

Pose computation in 25 lines of code

And… that is true…

Dementhon-Davis

Considering the projection equations, for each point (x_i, y_i)

$$\Leftrightarrow \begin{cases} x_i = \dfrac{{}^cX_i}{{}^cZ_i} = \dfrac{r_{11}{}^oX_i + r_{12}{}^oY_i + r_{13}{}^oZ_i + T_x}{r_{31}{}^oX_i + r_{32}{}^oY_i + r_{33}{}^oZ_i + T_z} \\[2em] y_i = \dfrac{{}^cY_i}{{}^cZ_i} = \dfrac{r_{21}{}^oX_i + r_{22}{}^oY_i + r_{23}{}^oZ_i + T_y}{r_{31}{}^oX_i + r_{32}{}^oY_i + r_{33}{}^oZ_i + T_z} \end{cases}$$

Let us pose
$$\begin{cases} \epsilon_i = (r_{31}{}^oX_i + r_{32}{}^oY_i + r_{33}{}^oZ_i)/T_z & (1) \\ \mathbf{I}^T = \frac{1}{t_z}(r_{11}, r_{12}, r_{13}, T_x) & (2) \\ \mathbf{J}^T = \frac{1}{t_z}(r_{21}, r_{22}, r_{23}, T_y) & (3) \end{cases}$$

Dementhon-Davis

We obtain

$$\left\{\begin{array}{l}\underbrace{\begin{pmatrix} {}^oX_i & {}^oY_i & {}^oZ_i & 1\end{pmatrix}}_{\mathbf{A_1}}\mathbf{I} = \underbrace{x_i(1+\epsilon_i)}_{\mathbf{B_1}} \\ \underbrace{\begin{pmatrix} {}^oX_i & {}^oY_i & {}^oZ_i & 1\end{pmatrix}}_{\mathbf{A_2}}\mathbf{J} = \underbrace{y_i(1+\epsilon_i)}_{\mathbf{B_2}}\end{array}\right.$$

That is two linear systems with N equations and 4 unknowns

- If ε_i in known

4 non coplanar points are necessary to solve such system

Just one more thing…

- ε_i in unknown

Dementhon-Davis

Linear iterative method

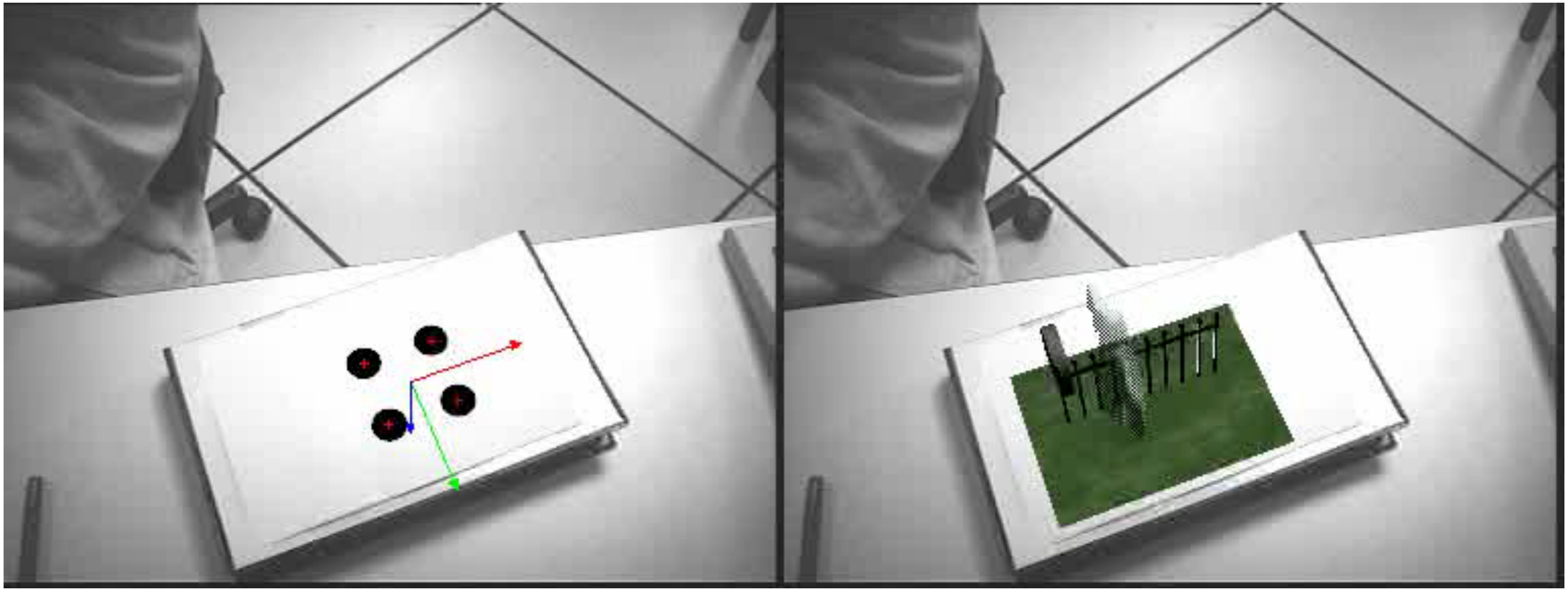
1. Initialization
2. Solve the linear systems $\mathbf{A_1 I = B_1}$ and $\mathbf{A_2 J = B_2}$
   - **A**$^+$ has to be computed only once
3. Equation (2) and (3)

$$T_z = \frac{i}{\mathbf{I}} \text{ and } i \text{ is a unitary vector } (i = \frac{\mathbf{I}}{||\,\mathbf{I}\,||})$$

$$T_z = 1/\,||\,\mathbf{I}\,|| \text{ where } T_z = 2/(||\,\mathbf{I}\,|| + ||\,\mathbf{J}\,||)$$

4. $z = i \times j \Rightarrow r_{ij}, T_x \text{ and } T_y$
5. $\epsilon_i = (r_{31}{}^oX_i + r_{32}{}^oY_i + r_{33}{}^oZ_i)/T_z$

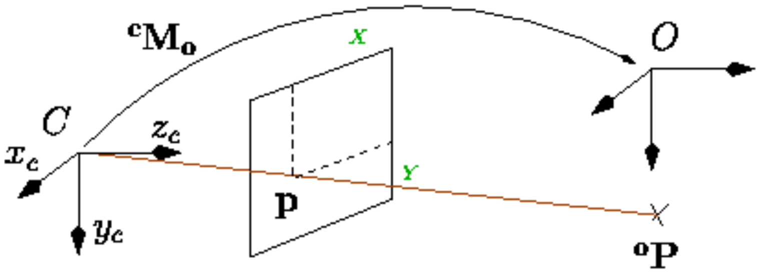
Dementhon-Davis



Pose estimation: non-linear minimization

Goal

- Estimate the pose ${}^{c}\mathbf{M}_{o}$ of an object with respect to the camera frame



Example for point features

- Minimizing the error between the observation $\mathbf{p}^*$ and the projection of the model in the image

$$\widehat{{}^{c}\mathbf{M}_{o}} = \arg \min_{{}^{c}\mathbf{R}_{o},{}^{c}\mathbf{t}_{o}} \sum_{i} \left(pr_{\xi}\left({}^{c}\mathbf{M}_{o}, {}^{o}\mathbf{P}_{i}\right) - \mathbf{p}_{i}^{*} \right)^{2}$$

where ${}^{o}\mathbf{P}$ are the coordinates of the same points in the object frame

Pose: non linear minimization

We have to estimate the pose that minimize

$$\widehat{{}^c\mathbf{M}_o} = \arg\min_{{}^c\mathbf{R}_o, {}^c\mathbf{t}_o} \sum_i \left(pr_\xi\left({}^c\mathbf{M}_o, {}^o\mathbf{P}_i\right) - \mathbf{p}_i^*\right)^2$$

$\left(pr_\xi\left({}^c\mathbf{M}_o, {}^o\mathbf{P}_i\right) - \mathbf{p}_i^*\right)^2$ is the distance between the observation and the projection of the object model

Rotation ${}^c\mathbf{R}_o$ is parameterized using the $\theta\mathbf{u}$ vector where θ is the rotation angle along the **u** vector

This leads to 6 independent vector to estimate

Minimization using a Gauss-Newton method

Linearization of the non-linear system

Problem: no general method to solve *f(**r**)=0*

There exists iterative method that linearize the problem in order to find an adequate solution

$$\begin{aligned} f_i(\mathbf{r}+\delta\mathbf{r}) &= f_i(\mathbf{r}) + \delta r_1 \frac{\partial f_1}{\partial r_1}(\mathbf{r}) + \ldots + \delta r_n \frac{\partial f_1}{\partial r_n}(\mathbf{r}) + O(\mid \delta\mathbf{r} \mid^2) \\ &\approx f_i(\mathbf{r}) + \nabla f_i(\mathbf{r})\delta\mathbf{r} \end{aligned} \tag{1}$$

where $\nabla f_i(\mathbf{r}) = (\frac{\partial f_1}{\partial r_1}, \ldots, \frac{\partial f_1}{\partial r_n})^T$ is the gradient of f_i in **r** and where second order terms are neglected.

Linearization of the non-linear system

With the Gauss-Newton method, we do no want to determine the value of $\delta\mathbf{r}$ that ensures $f(\mathbf{r})=0$ but the value that minimizes the cost function:

$$E(\mathbf{r}+\delta\mathbf{r}) = \|\mathbf{f}(\mathbf{r}+\delta\mathbf{r})\|^2 \approx \|\mathbf{f}(\mathbf{r}) + \mathbf{J}_\mathbf{f}(\mathbf{r})\delta\mathbf{r}\|^2$$

This is a linear minimization problem (solved by a least-square approach) and we have:

$$\delta\mathbf{r} = -\mathbf{J}_\mathbf{f}^+(\mathbf{r})\mathbf{f}(\mathbf{r}) = \left(\mathbf{J}_\mathbf{f}^\mathbf{T}(\mathbf{r})\mathbf{J}_\mathbf{f}(\mathbf{r})\right)^{-1}\mathbf{J}_\mathbf{f}^\mathbf{T}(\mathbf{r})\mathbf{f}(\mathbf{r})$$

Computing the Jacobian

We have to compute the Jacobian that links the variation of the measurements **x** = *(x,y)* to the variation of the pose.

That is :

$$\frac{\partial \mathbf{x}}{\partial t} = \frac{\partial \mathbf{x}}{\partial \mathbf{r}} \frac{\partial \mathbf{r}}{\partial t}$$

or

$$\dot{\mathbf{x}} = \mathbf{J}\mathbf{v}$$

UMR IRISA

Jacobian: case of the point

Some definitions

- Let *(O,x,y,z)* be the camera frame
- Let ***x****(X,Y,Z)* be the 3D position the point
- Let the camera velocity be $\mathbf{v} = (V, \Omega) = (V_x, V_y, V_z, \Omega_x, \Omega_y, \Omega_z)$

The relation that link the point velocity $\dot{\mathbf{X}}$ to the camera velocity is given by:

$$\dot{\mathbf{X}} = -V - \Omega \times \mathbf{X}$$

Jacobian: case of the point

$$\dot{\mathbf{X}} = -V - \Omega \times \mathbf{X}$$

Is equivalent to

$$\left\{\begin{array}{lcl} \dot{X} & = & -V_x - \Omega_y Z + \Omega_z Y \\ \dot{Y} & = & -V_y - \Omega_z X + \Omega_x Z \\ \dot{Z} & = & -V_z - \Omega_x Y + \Omega_y X \end{array}\right.$$

On the other hand, the perspective equation gives $\left\{\begin{array}{lcl} x & = & \frac{X}{Z} \\ y & = & \frac{Y}{Z} \end{array}\right.$

Which can be derived

$$\left\{\begin{array}{lcl} \frac{\partial x}{\partial t} & = & \frac{\partial x}{\partial X}\frac{\partial X}{\partial t} + \frac{\partial x}{\partial Z}\frac{\partial Z}{\partial t} \\ \\ \frac{\partial y}{\partial t} & = & \frac{\partial y}{\partial Y}\frac{\partial Y}{\partial t} + \frac{\partial y}{\partial Z}\frac{\partial Z}{\partial t} \end{array}\right. \Leftrightarrow \left\{\begin{array}{lcl} \dot{x} & = & \frac{\dot{X}}{Z} - \frac{X}{Z^2}\dot{Z} \\ \\ \dot{y} & = & \frac{\dot{Y}}{Z} - \frac{Y}{Z^2}\dot{Z} \end{array}\right.$$

Jacobian: case of the point

Considering $(\dot{X}, \dot{Y}, \dot{Z})$ obtained in (1)

$$\left\{ \begin{array}{lcl} \dot{x} & = & \frac{\dot{X}}{Z} - \frac{X}{Z^2}\dot{Z} \;=\; \frac{-V_x - \Omega_y Z + \Omega_z Y}{Z} - \frac{X}{Z^2}(-V_z - \Omega_x Y + \Omega_y X) \\ \dot{y} & = & \frac{\dot{Y}}{Z} - \frac{Y}{Z^2}\dot{Z} \;=\; \frac{-V_y - \Omega_z X + \Omega_x Z}{Z} - \frac{Y}{Z^2}(-V_z - \Omega_x Y + \Omega_y X) \end{array} \right.$$

or

$$\left\{ \begin{array}{lcllllll} \dot{x} & = & -\frac{1}{Z}V_x & & +\frac{X}{Z^2}V_z & +\frac{XY}{Z^2}\Omega_x & -(1+\frac{X^2}{Z^2})\Omega_y & +\frac{Y}{Z}\Omega_z \\ \dot{y} & = & & -\frac{1}{Z}V_y & +\frac{Y}{Z^2}V_z & +(1+\frac{Y^2}{Z^2})\Omega_x & -\frac{XY}{Z^2}\Omega_y & -\frac{X}{Z}\Omega_z \end{array} \right.$$

Jacobian: case of the point

We finally have

$$\left\{\begin{array}{cclllll} \dot{x} & = & -\frac{1}{Z}V_x & & +\frac{x}{Z}V_z & +xy\Omega_x & -(1+x^2)\Omega_y & +y\Omega_z \\ \dot{y} & = & & -\frac{1}{Z}V_y & +\frac{y}{Z}V_z & +(1+y^2)\Omega_x & -xy\Omega_y & -x\Omega_z \end{array}\right.$$

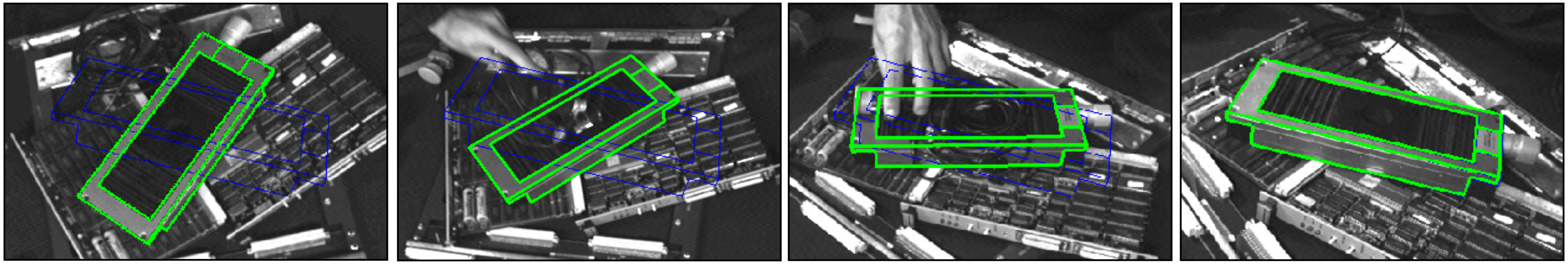
or

$$\begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} = \begin{pmatrix} -\frac{1}{Z} & 0 & \frac{x}{Z} & xy & -(1+x^2) & y \\ 0 & -\frac{1}{Z} & \frac{y}{Z} & (1+y^2) & -xy & -x \end{pmatrix} \mathbf{v}$$

Pose: an obvious link with visual servoing

Visual servoing

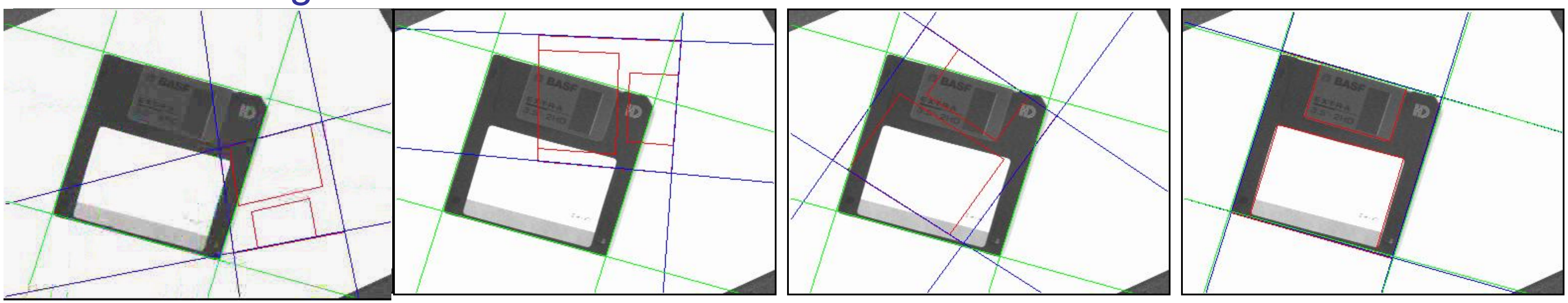
- Move a camera in order to observe an object at a given position in the image.



Pose: an obvious link with visual servoing

Visual servoing

- Move a camera in order to observe an object at a given position in the image.



Pose calculation via non-linear methods is similar to visual servoing

Virtual Visual Servoing [Sundareswaran 98]

- Virtually moves a camera so that the projection of a model of the object corresponds to the observed image
- The end position of the virtual camera is the expected pose

UMR IRISA

Virtual visual servoing

We want to minimize the following error

$$\mathbf{\Delta} = \mathbf{s}(\mathbf{r}) - \mathbf{s}^*$$

where

- **s*** is the position of the features in the image
- **s(r)** is the current position of the projected features for a pose **r**

The displacement of the projected features due to a variation of the pose is given by

$$\dot{\mathbf{s}} = \frac{\partial \mathbf{s}}{\partial \mathbf{r}} \frac{d\mathbf{r}}{dt} = \mathbf{L_s} \mathbf{v}$$

If we specify an exponential decrease of the error $\dot{\mathbf{e}} = -\lambda \mathbf{e}$

The control law that ensure the minimization is

$$\mathbf{v} = -\lambda \mathbf{L}_{\mathbf{s}}^{+} (\mathbf{s}(\mathbf{r}) - \mathbf{s}^*)$$

UMR IRISA

Virtual visual servoing: robustness to outliers

The residue $\Delta_{\mathcal{R}} = \rho\big(\mathbf{s}(\mathbf{r}) - \mathbf{s}^*\big)$

Tukey's M-estimator $w_i = \dfrac{\psi(\delta_i/\sigma)}{\delta_i/\sigma}$

$$\psi(u) = \begin{cases} u(C^2 - u^2)^2 & |u| \leq C \\ 0 & \text{else} \end{cases}$$

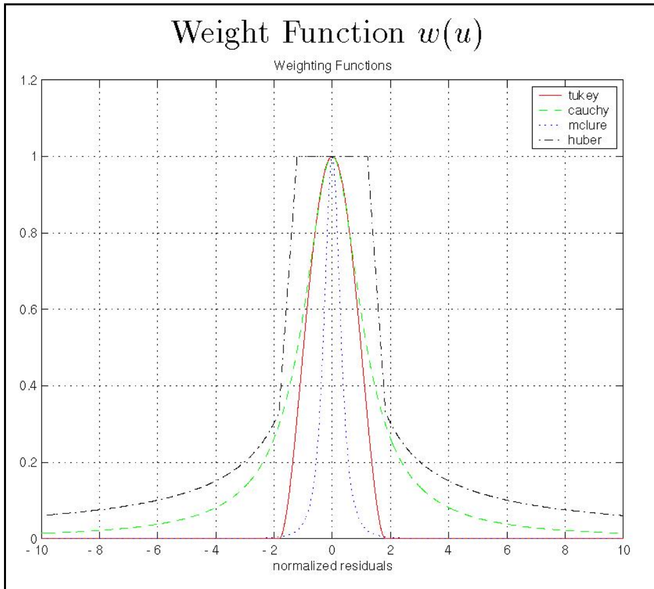
- where ρ is a robust function (M-estimation)

The control law, similar to an IRLS, which minimizes s-s* is given by

$$\mathbf{v} = -\lambda(\mathbf{D}\mathbf{L_s})^+\mathbf{D}(\mathbf{s}(\mathbf{r}) - \mathbf{s}^*)$$

where

$$\mathbf{D} = \begin{bmatrix} w_1 & & 0 \\ & \ddots & \\ 0 & & w_n \end{bmatrix}$$



Visual features

Can use any kind of visual feature

- Constraint: compute $\mathbf{L_s}$

Mix various visual features within the same process

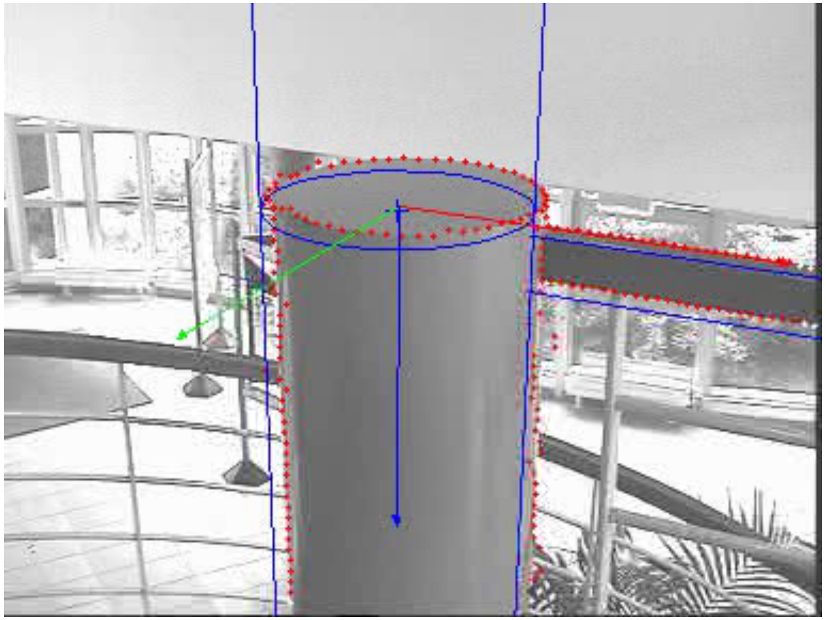
$$\dot{\mathbf{s}} = \begin{bmatrix} \dot{\mathbf{s}}_1 \\ \vdots \\ \dot{\mathbf{s}}_n \end{bmatrix} = \begin{bmatrix} \mathbf{L}_{s_1} \\ \vdots \\ \mathbf{L}_{s_n} \end{bmatrix} \mathbf{v} = \mathbf{L}_s \mathbf{v}$$

- Constraint : $\mathbf{L_s}$ must be full rank

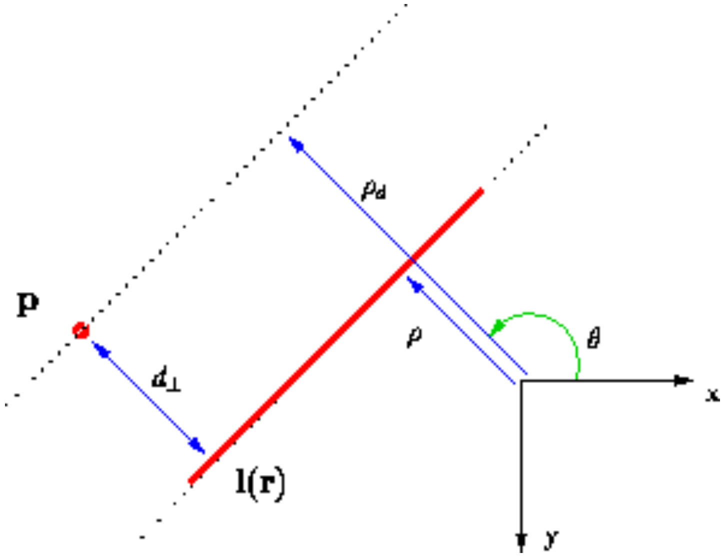
Visual features

Distance to a moving line

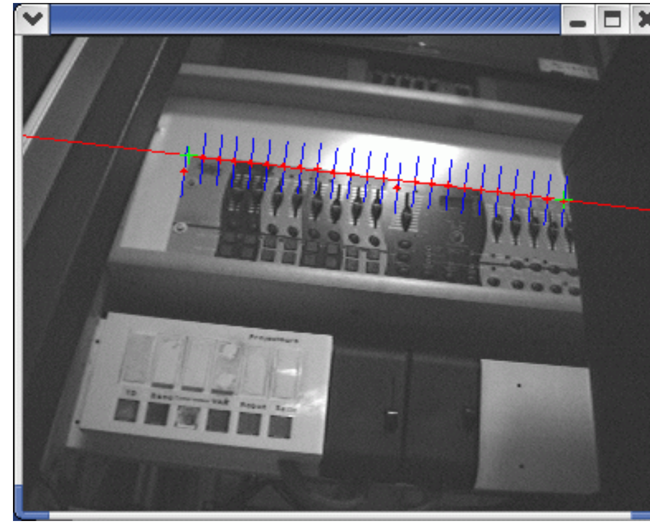
- **p** : point extracted in the image using the ECM algorithm
- **l(r)** : projection of the object model for pose **r**



$$d_l = d_\perp(\mathbf{p}, \mathbf{l}(\mathbf{r})) = \rho(\mathbf{l}(\mathbf{r})) - \rho_d,$$

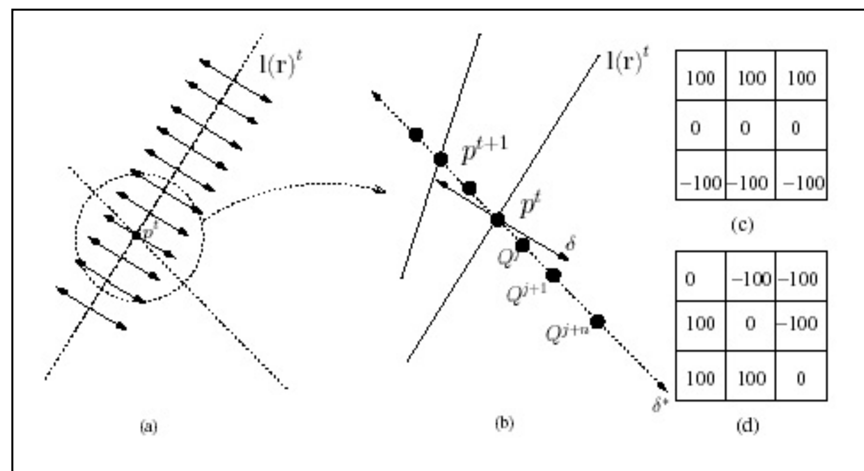


Low-level image processing: ME



Local tracking of edge points

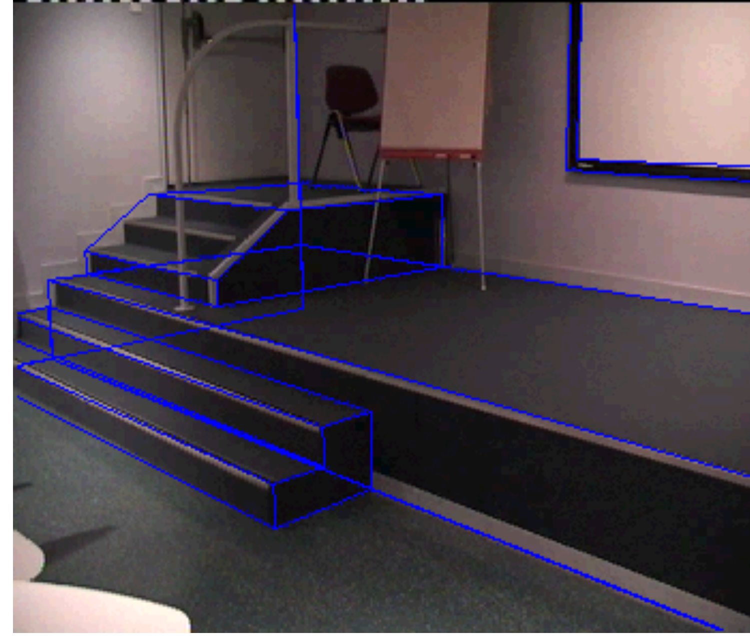
- ME algorithm [Bouthemy PAMI 89]
- 1D search algorithm
- Convolution with oriented mask
- Comparison with the convolution at time t-1, same orientation, same contrast

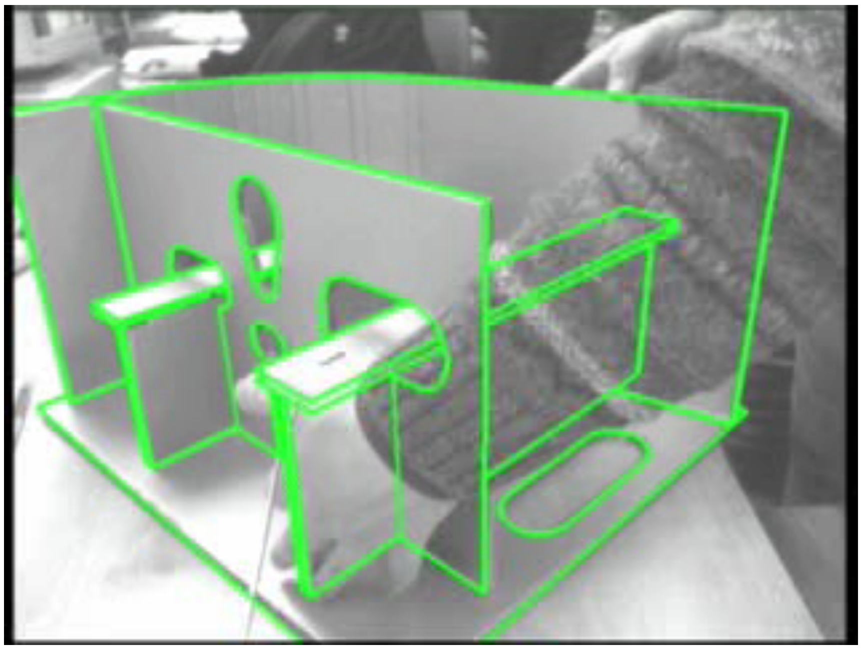


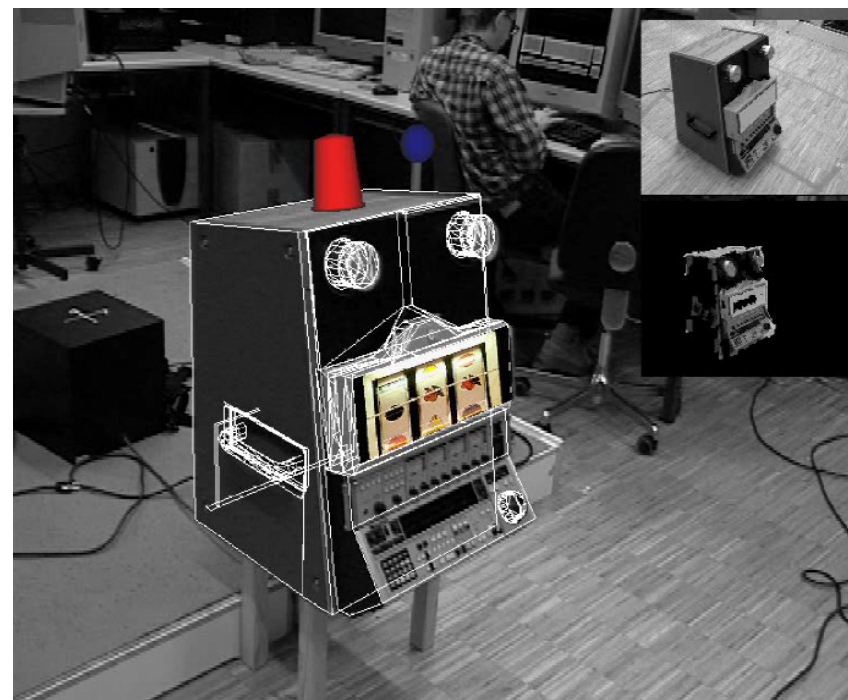
Frame rate performance

Not always efficient in textured area

3D model-based tracking

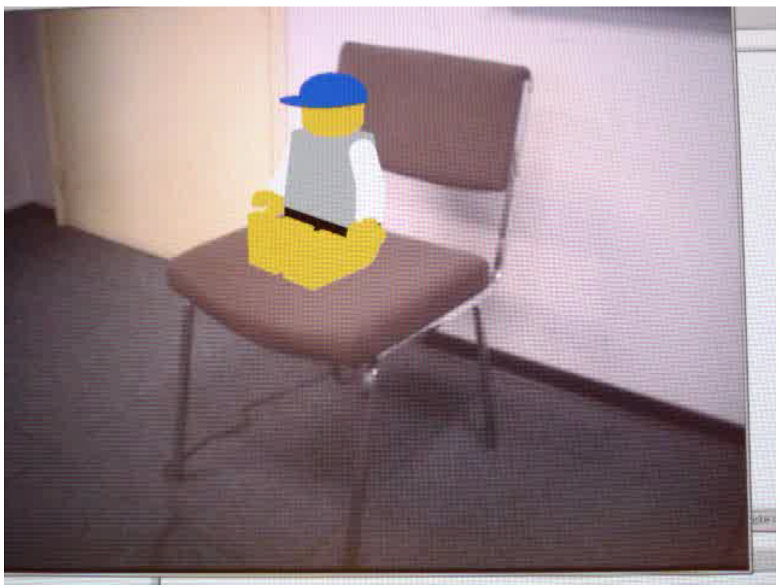


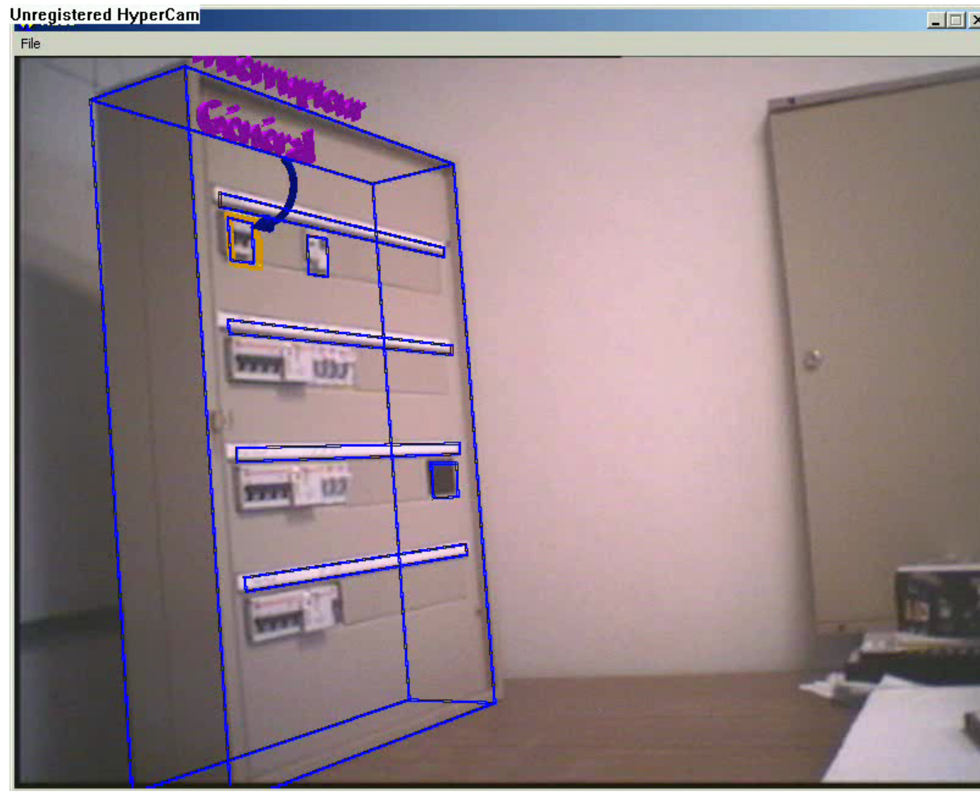




3D model-based tracking: augmented reality

Application to augmented reality



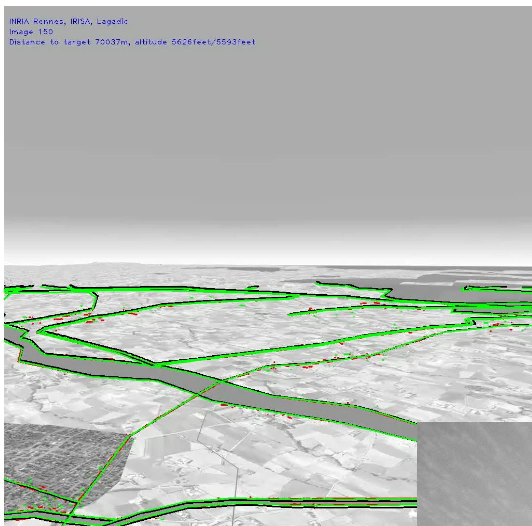


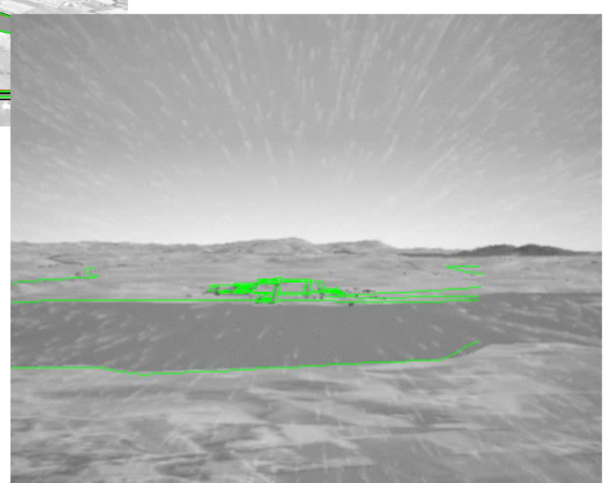
3D model-based tracking: augmented reality

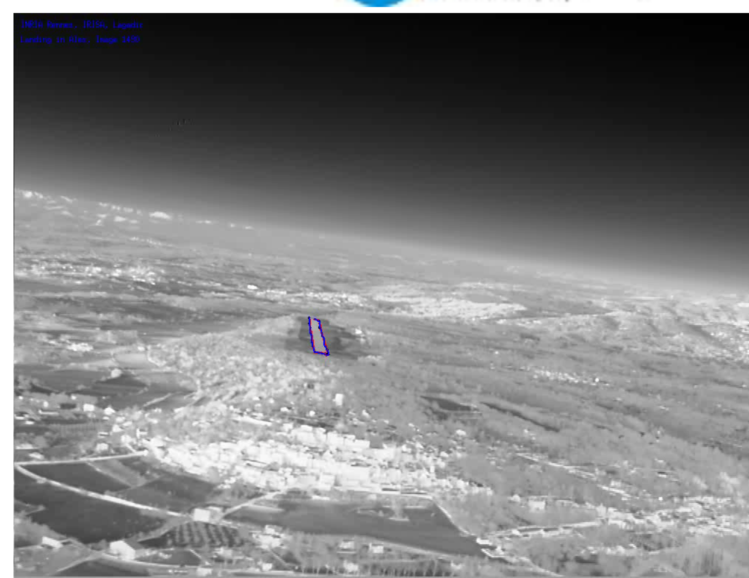
Application to augmented reality

Aircraft localization and landing

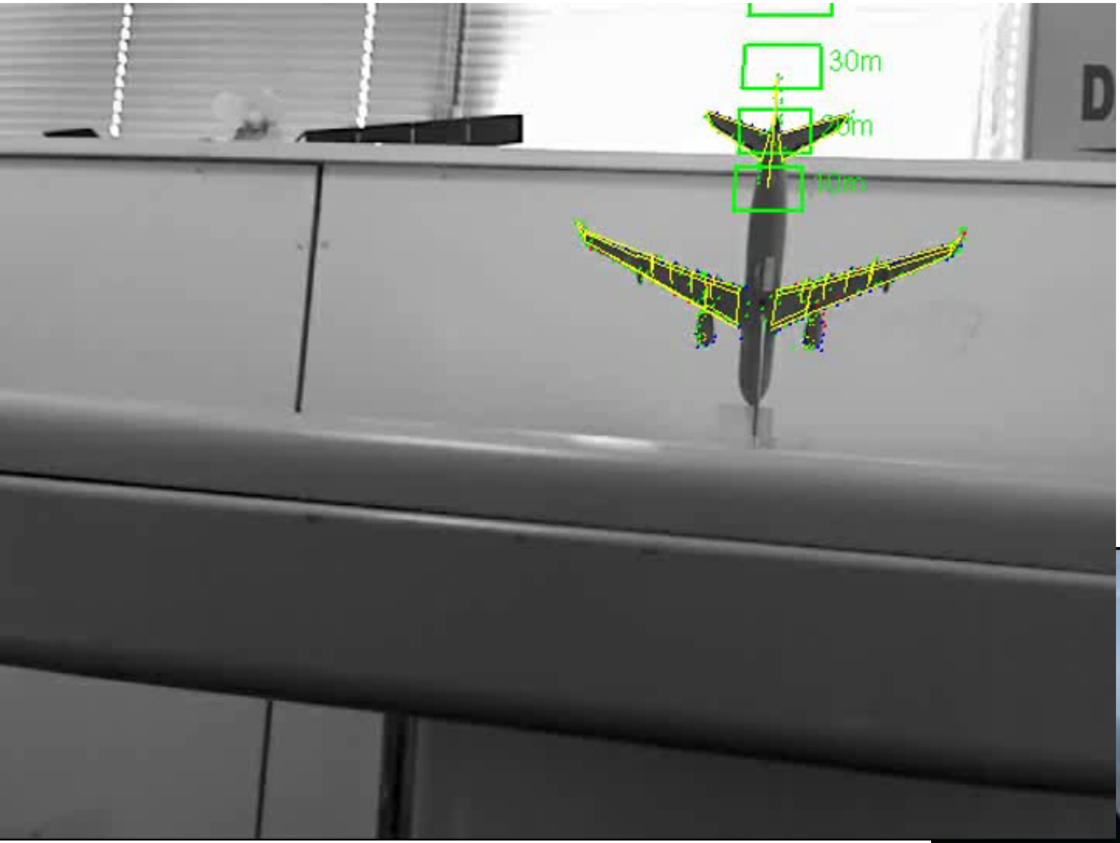
- FP6 Aerospace Pegase project (with Dassault aviation)
- 3D model : a large set of vectorial data (roads, rivers, coasts) managed hierarchically







Air refuelling, carrier landing

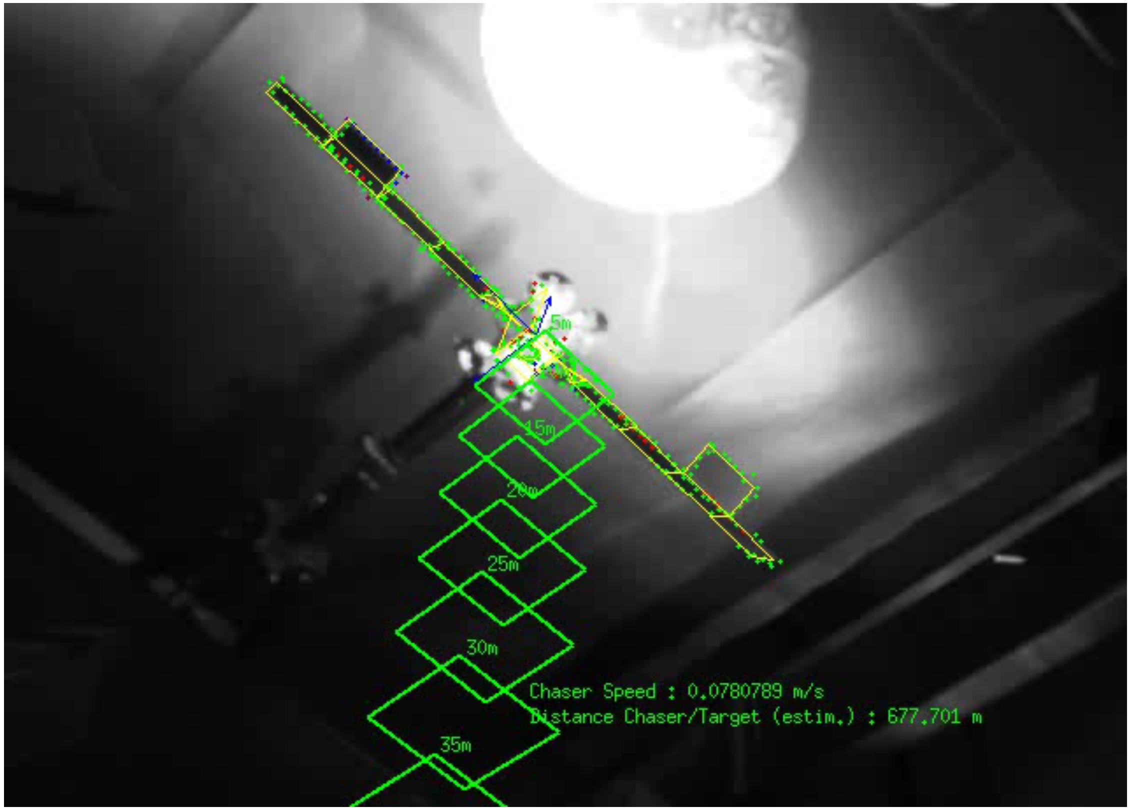


Visual tracking and servoing during the alignment and descent phases

Internal view

External view

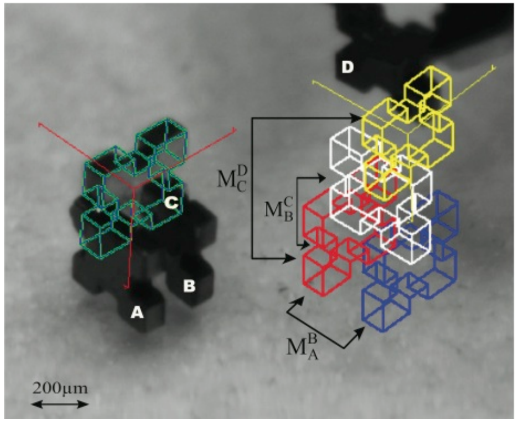
Satellite tracking (with EADS Atrium)

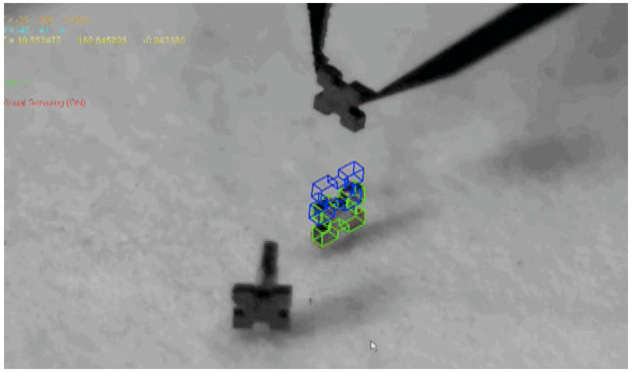


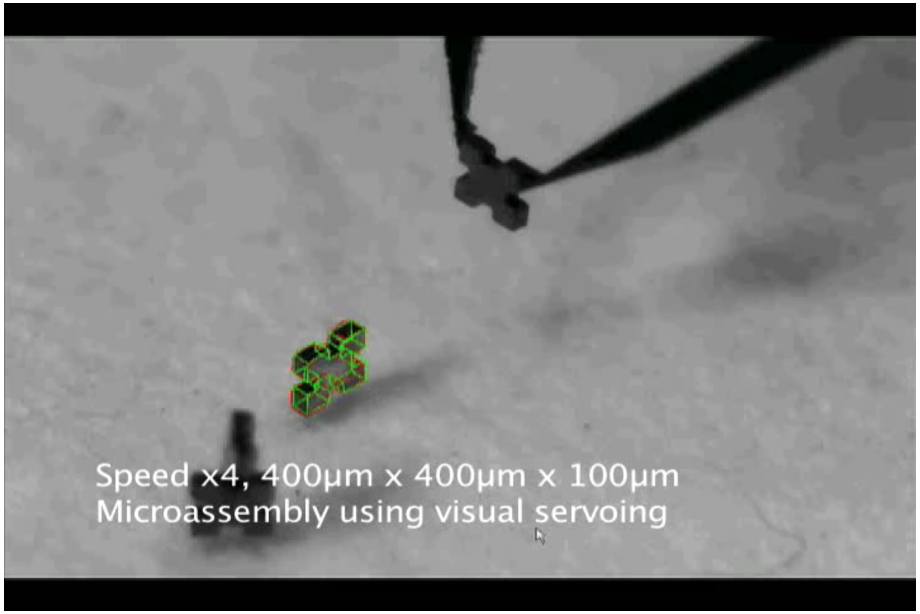
3D model-based tracker
Application to micro manipulation

Application to MEMS micro manipulation (collaboration with FEMTO-ST, Besançon)

Assembly of complex MEMS compounds

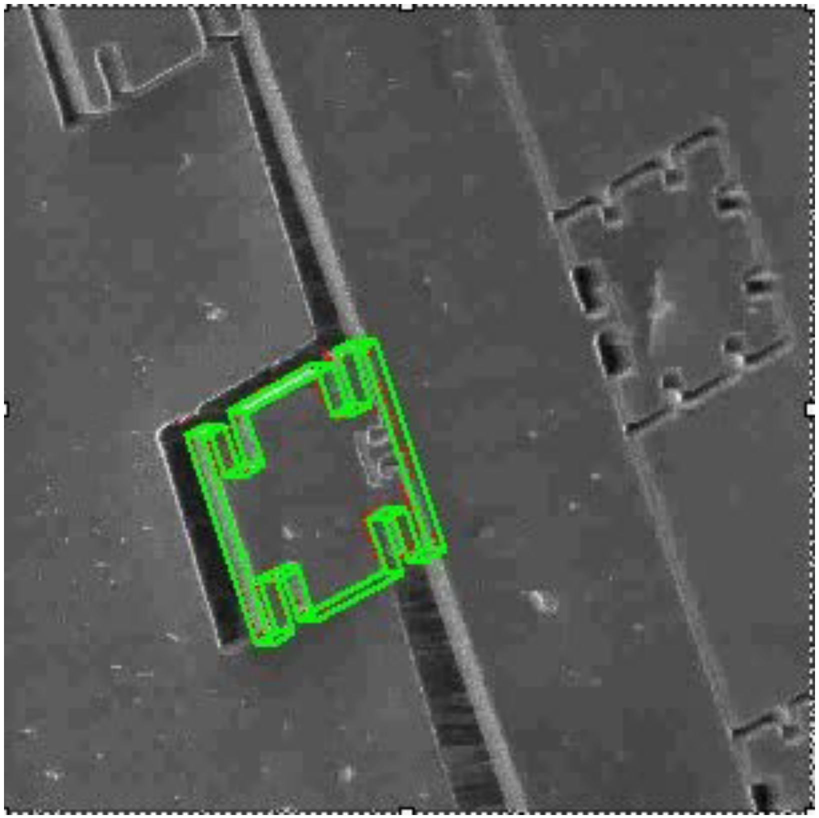






Size 400 µm x 400µm

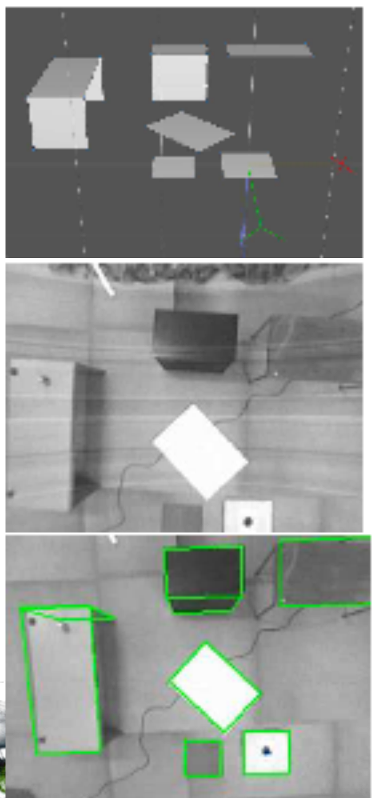
MEMS Tracking

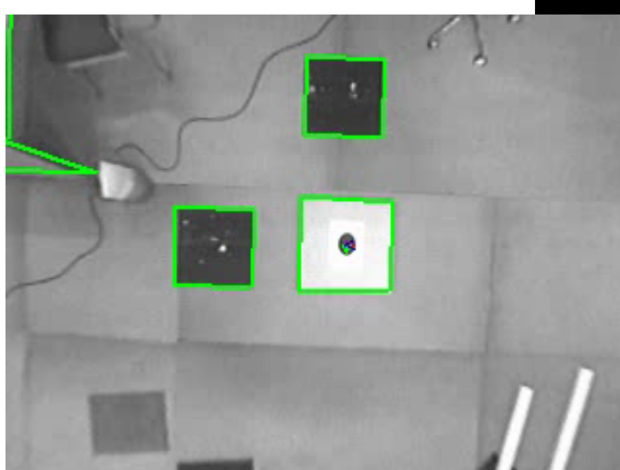


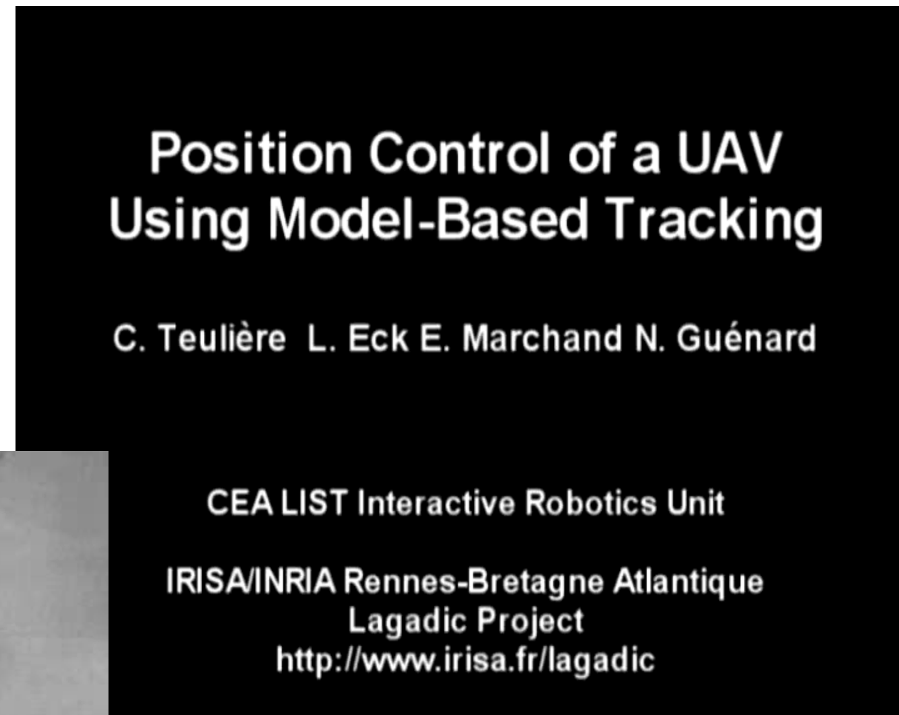
3D model-based tracking for UAV position control

Quadrirotor localization and position-based control

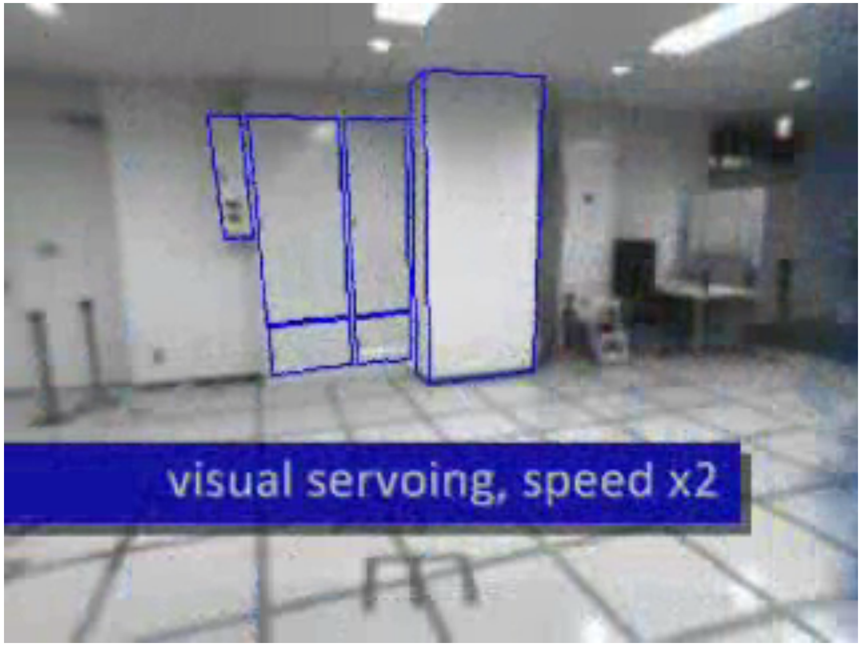
3D tracking, position an velocity estimation required for the control scheme

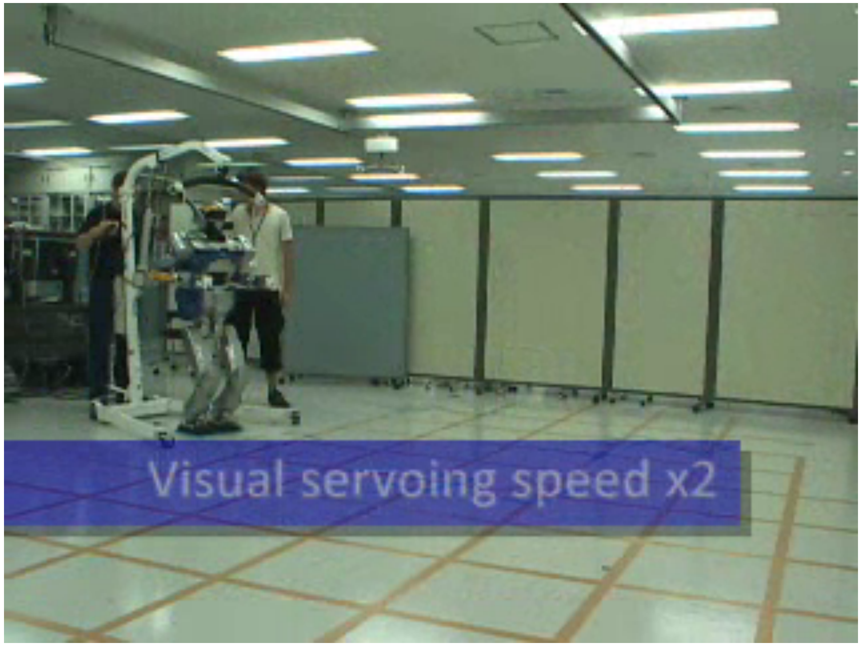






Humanoïd robotics





AIST/JRL/LAAS/IRISA

UMR IRISA

Overview: model-based tracking

Visual tracking

- Various tracking algorithms
- Contour-based tracking
  - Monocular 3D model-based tracking
  - Multi-cameras 3D model-based tracking
  - Tracking in central catadioptric cameras
- Hybrid tracking
  - Introducing a spatio-temporal constraints
  - Optic flow

Illustration with applications in

- Augmented reality
- Visual servoing

UMR IRISA

Extension to multiple cameras

New objective function

$$\Delta = \sum_{i=1}^{k_1} \left(pr_{\xi_1}\left({}^{c_1}\mathbf{M}_o, {}^{o}\mathbf{P}_i\right) - {}^{c_1}s_i^*\right)^2 + \sum_{j=1}^{k_2} \left(pr_{\xi_2}\left({}^{c_2}\mathbf{M}_o, {}^{o}\mathbf{P}_j\right) - {}^{c_2}s_j^*\right)^2$$

But if calibration of the stereo system is known
this is equivalent to

$$\Delta = \sum_{i=1}^{k_1} \left(pr_{\xi_1}\left({}^{c_1}\mathbf{M}_o, {}^{o}\mathbf{P}_i\right) - {}^{c_1}s_i^*\right)^2 + \sum_{j=1}^{k_2} \left(pr_{\xi_2}\left({}^{c_2}\mathbf{M}_{c_1}\,{}^{c_1}\mathbf{M}_o, {}^{o}\mathbf{P}_j\right) - {}^{c_2}s_j^*\right)^2$$

Extension to multiple cameras

The velocity of virtual camera 1 is linked to the velocity of virtual camera 2 by:

$$ {}^{2}\mathbf{v} = {}^{c_1}\mathbf{V}_{c_2}\, {}^{1}\mathbf{v} \quad \text{with} \quad {}^{c_1}\mathbf{V}_{c_2} = \begin{bmatrix} {}^{c_1}\mathbf{R}_{c_2} & [{}^{c_1}\mathbf{t}_{c_2}]_\times \\ \mathbf{0} & {}^{c_1}\mathbf{R}_{c_2} \end{bmatrix} $$

Which leads to the following interaction matrix:

$$ \begin{bmatrix} \dot{\mathbf{s}}_1 \\ \dot{\mathbf{s}}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{L}_1 \\ \mathbf{L}_2\, {}^{\mathbf{c}_2}\mathbf{V}_{\mathbf{c}_1} \end{bmatrix} {}^{1}\mathbf{v} $$

And the corresponding control law

$$ {}^{1}\mathbf{v} = -\lambda \begin{bmatrix} \widehat{\mathbf{D}}_1 \widehat{\mathbf{L}_{\mathbf{s}1}} \\ \widehat{\mathbf{D}}_2 \widehat{\mathbf{L}_{\mathbf{s}2}}\, {}^{c_2}\mathbf{V}_{c_1} \end{bmatrix}^{+} \begin{bmatrix} \mathbf{D}_1 \\ \mathbf{D}_2 \end{bmatrix} \begin{bmatrix} \mathbf{s}_1(\mathbf{r}_1) - \mathbf{s}_1^* \\ \mathbf{s}_2(\mathbf{r}_2) - \mathbf{s}_2^* \end{bmatrix} $$

UMR IRISA

Tracking the APFR

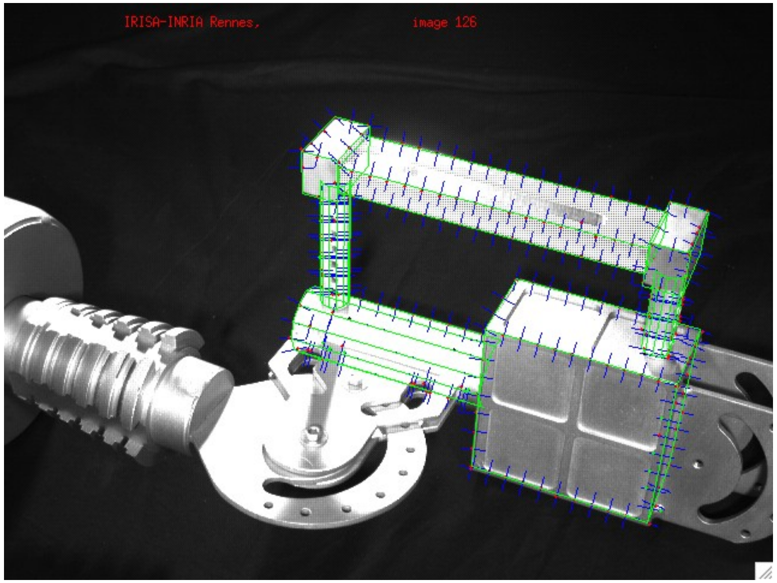
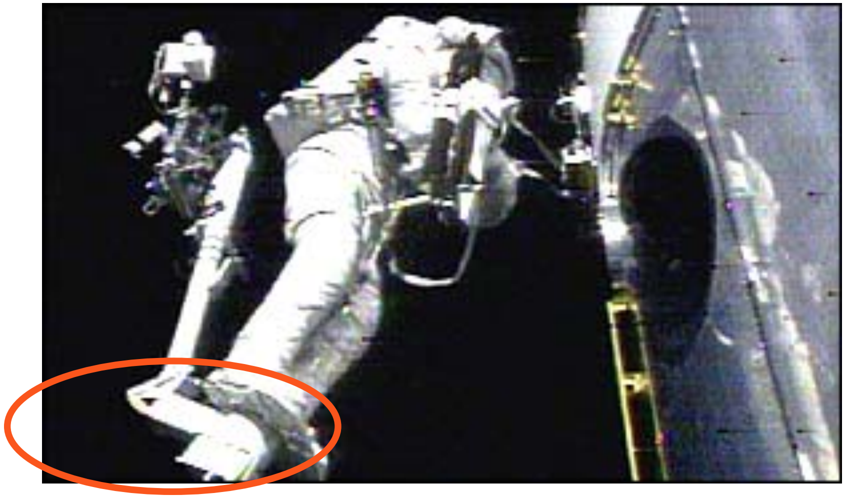


Image courtesy ESA

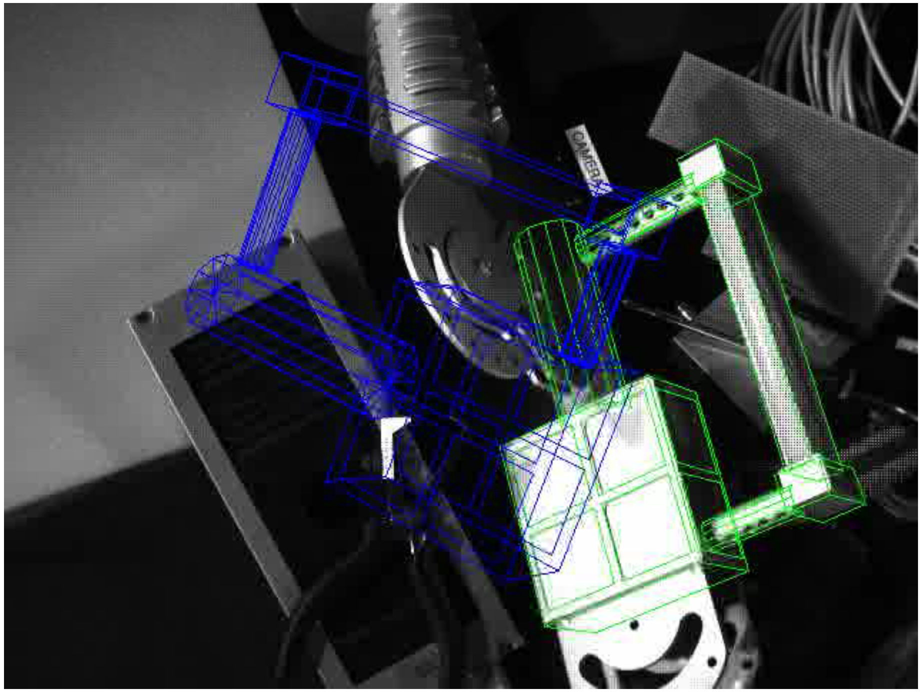


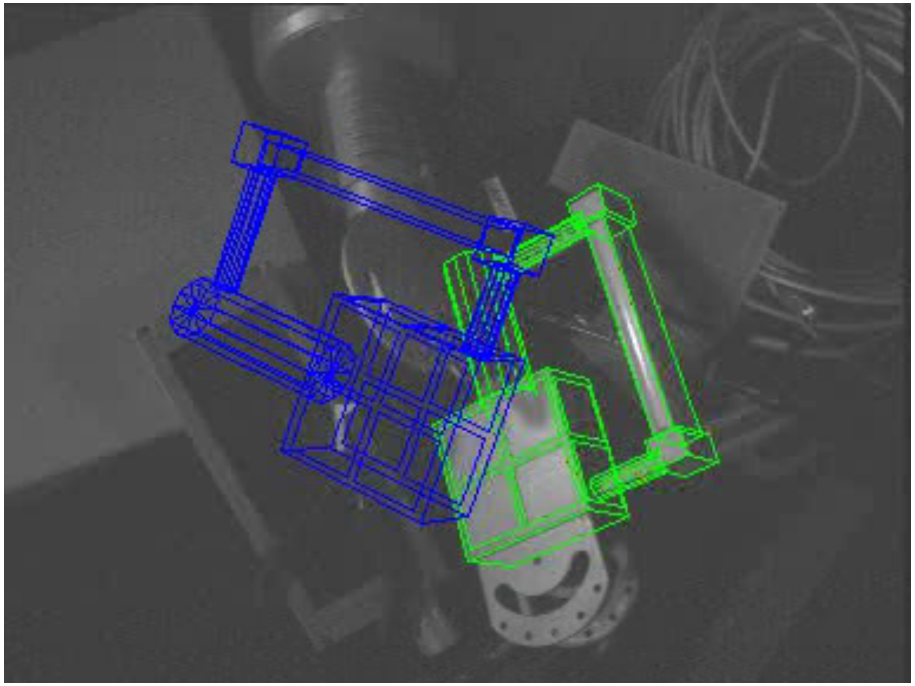
Articulated portable foot restraint

APFR CAD model

UMR IRISA

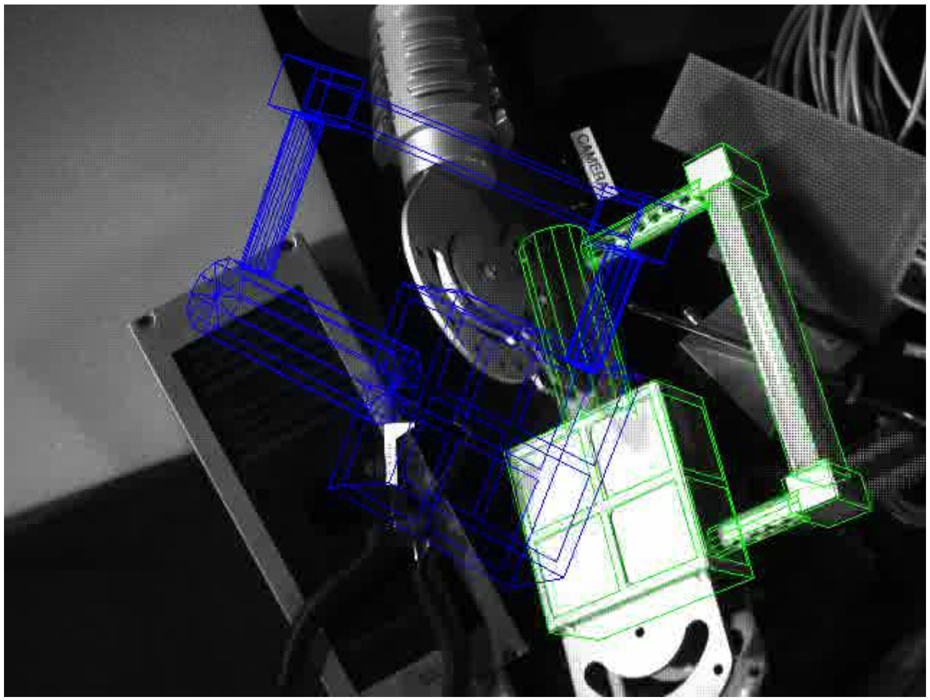
Visual servoing with stereo tracking

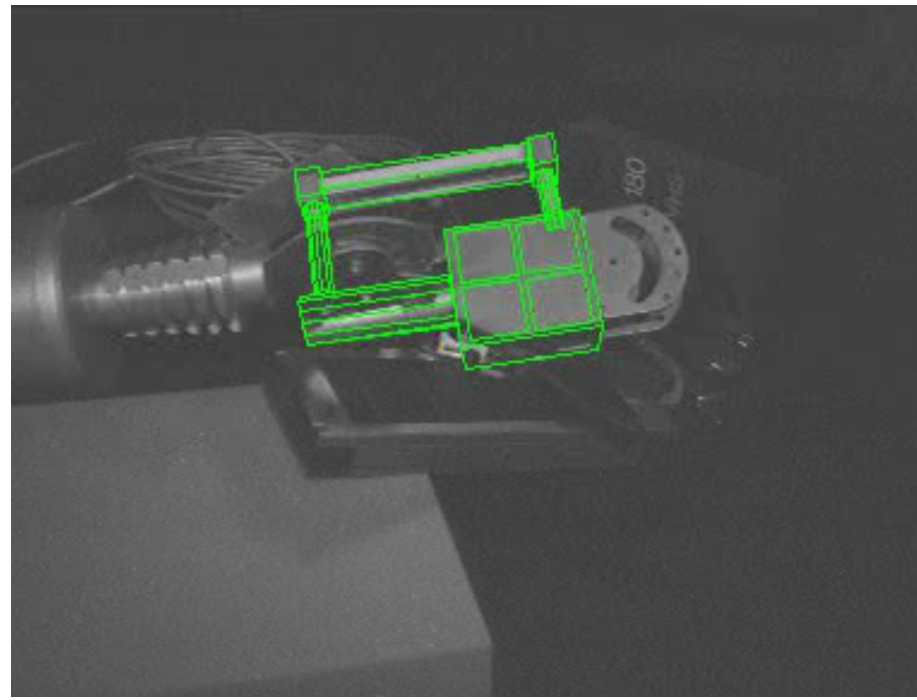




Small baseline

Visual servoing with stereo tracking

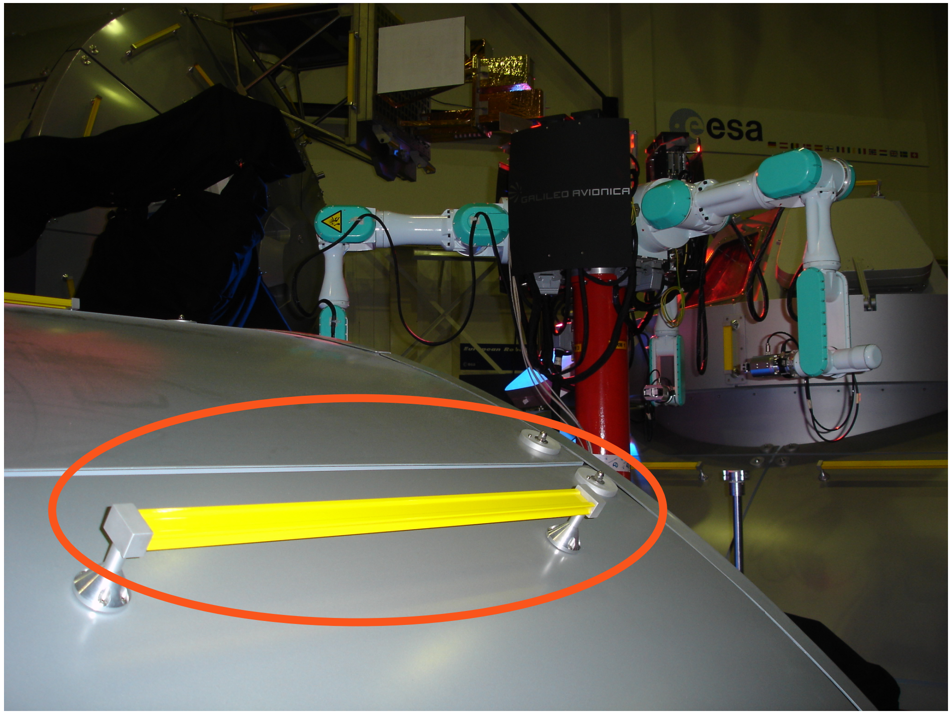




Wide baseline

Walking across the ISS, grasping the handrail

- Edge not really… sharp
- Cast shadows



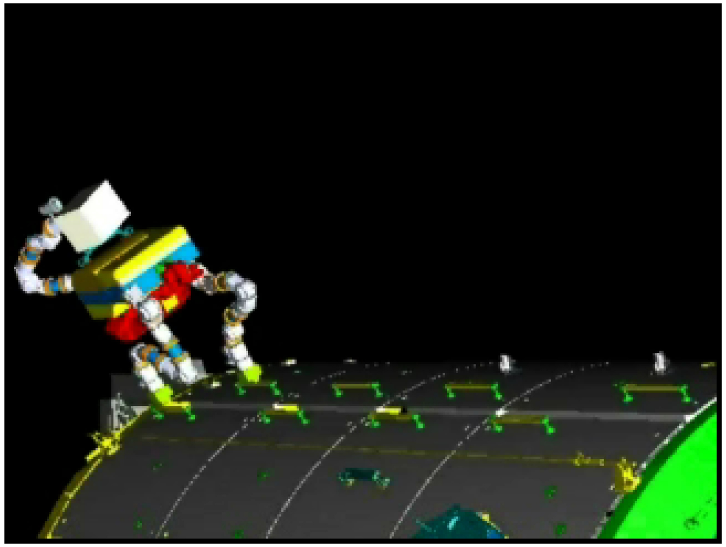
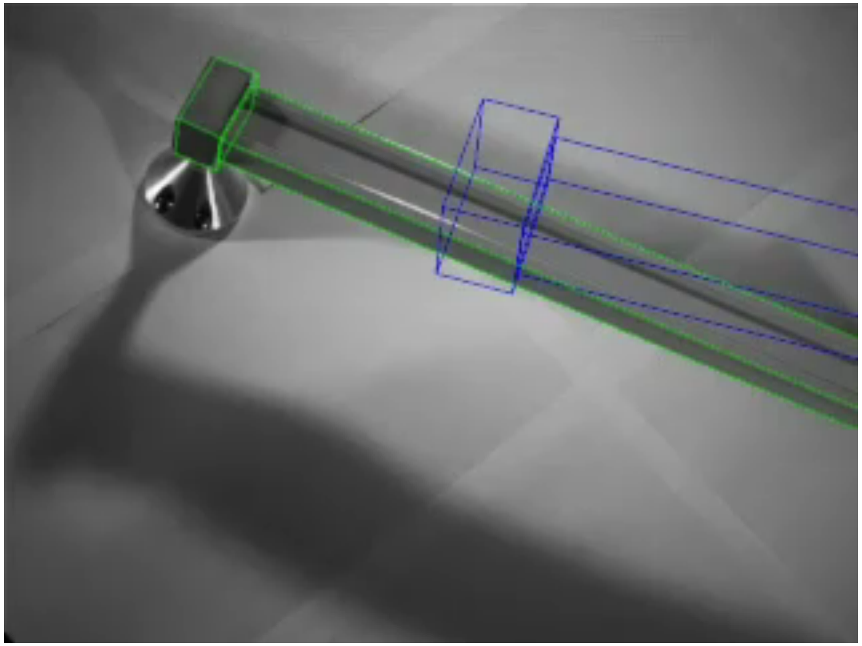
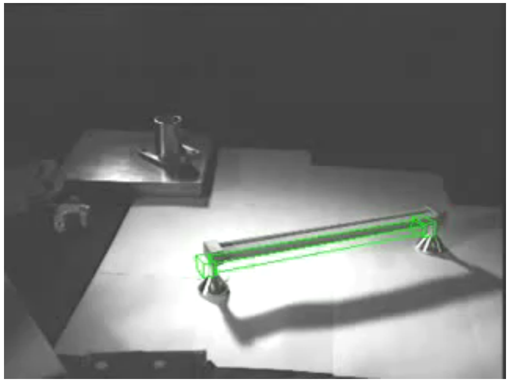


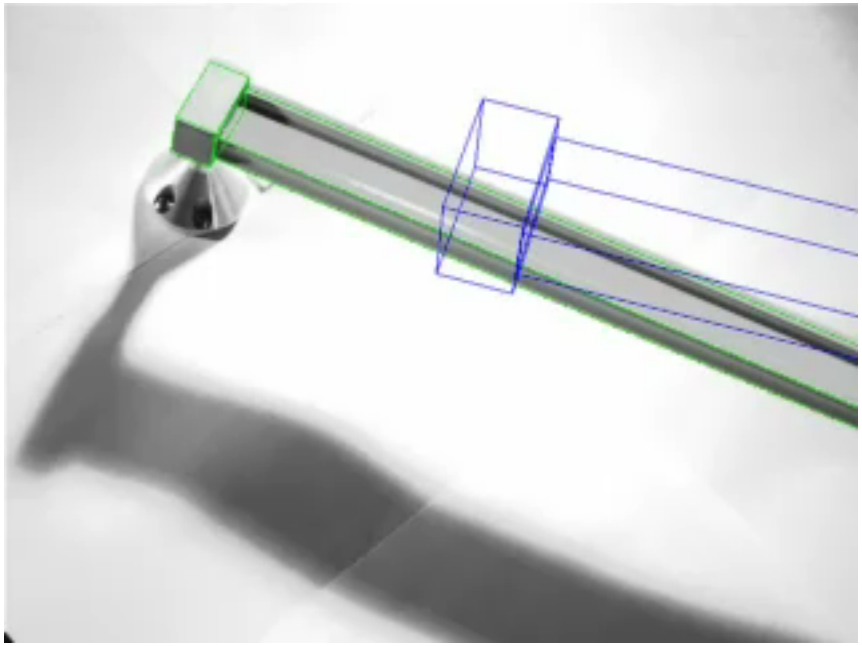
Image courtesy ESA

- Tracking at 20Hz

Servo a handrail under nominal conditions

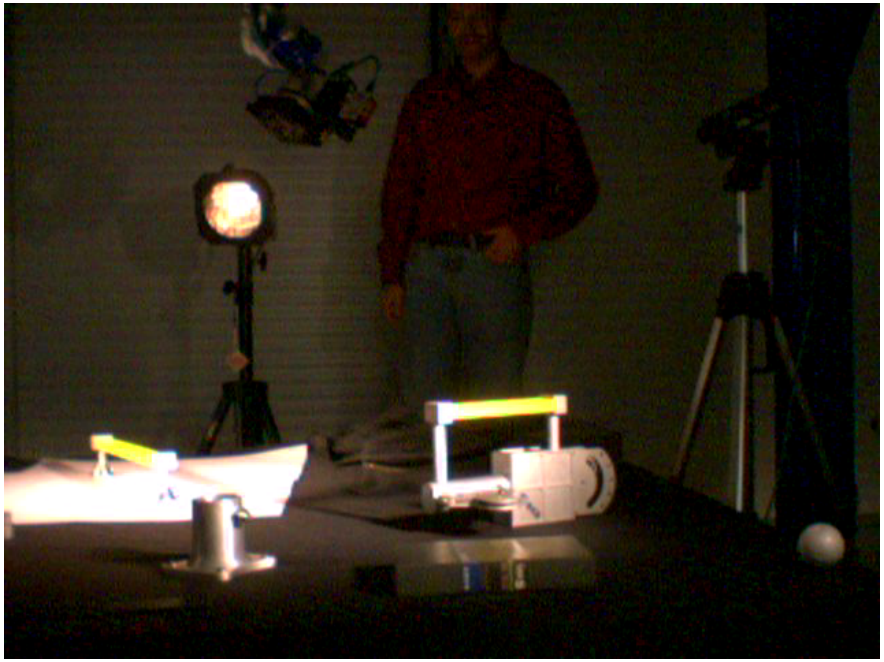






UMR IRISA

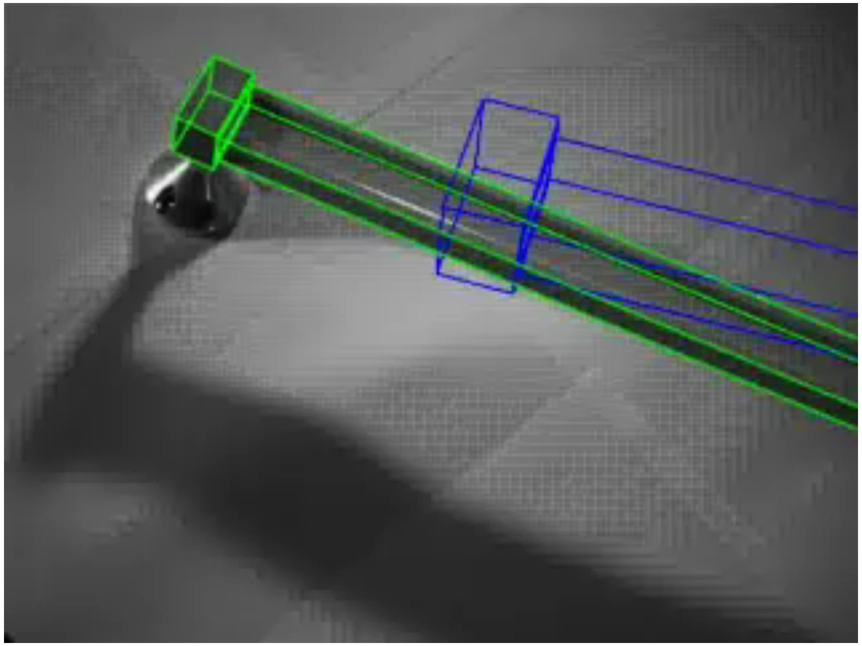
Servo a handrail with lighting variations



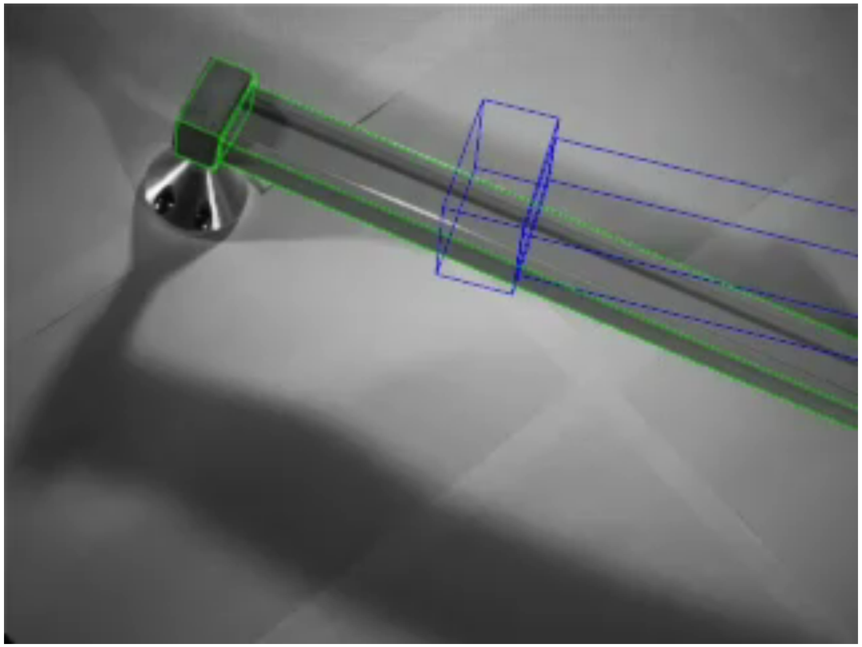
External view of the experiment

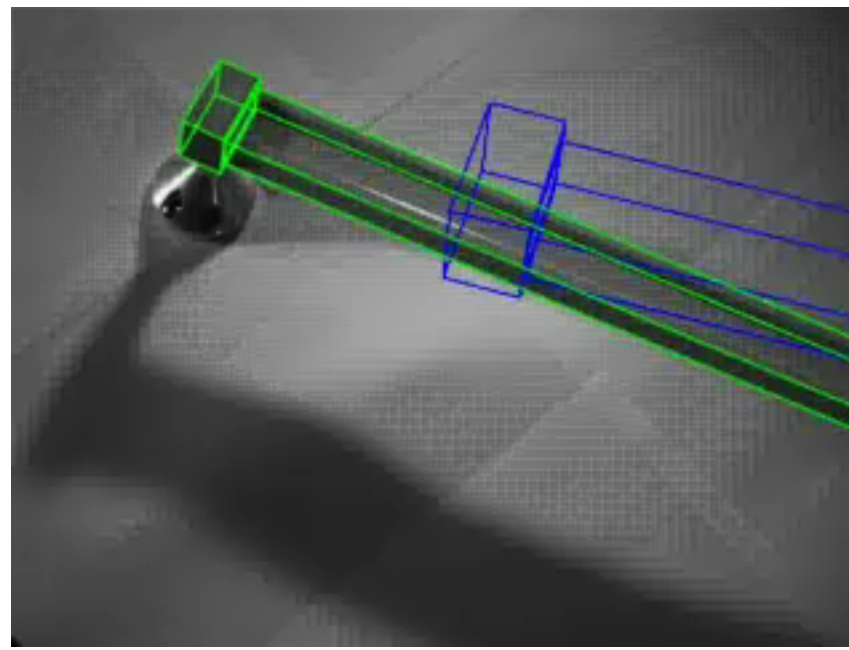
Camera view

- Occlusions
- Lighting variations



Servo a handrail with lighting variations





Overview: model-based tracking

Visual tracking

- Various tracking algorithms
- Contour-based tracking
  - Monocular 3D model-based tracking
  - Multi-cameras 3D model-based tracking
  - Tracking in central catadioptric cameras
- Hybrid tracking
  - Introducing a spatio-temporal constraints
  - Optic flow

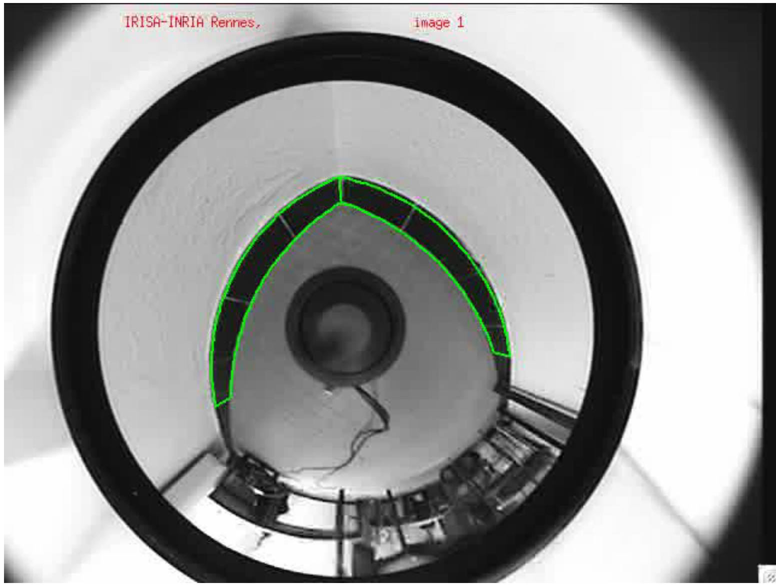
Illustration with applications in

- Augmented reality
- Visual servoing

Tracking in central catadioptric cameras

Same approach

$$\Delta = \sum_i \left(pr_\xi\left({}^c\mathbf{M}_o, {}^o\mathbf{S}_i\right) - s_i^*\right)^2$$



but with

- New projection model
  - A straight line projects as a conic
- New interaction matrix
  - Related to the distance of a point to the projection of a line

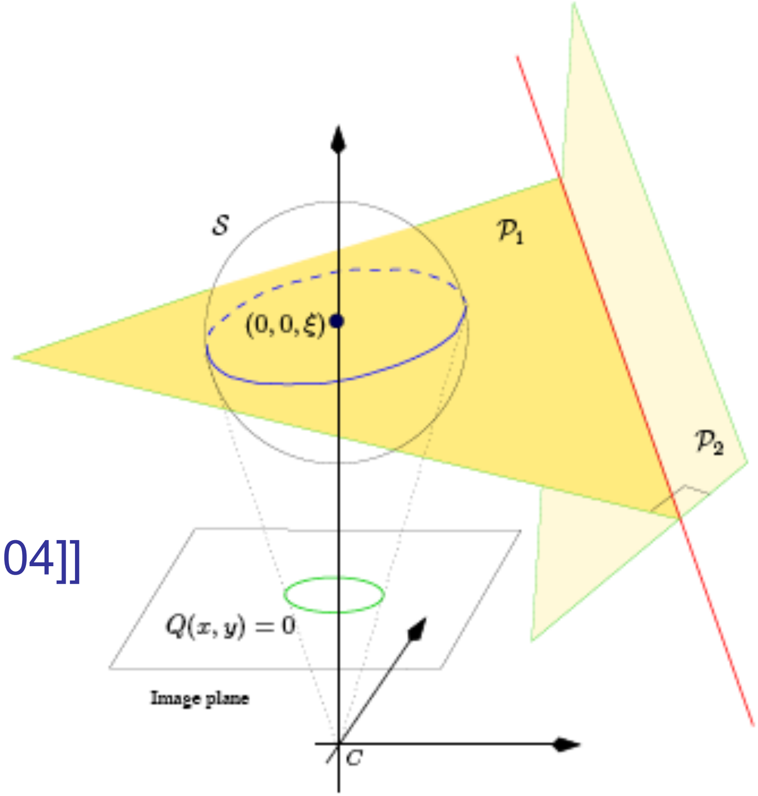
Tracking in central catadioptric cameras

Projection model

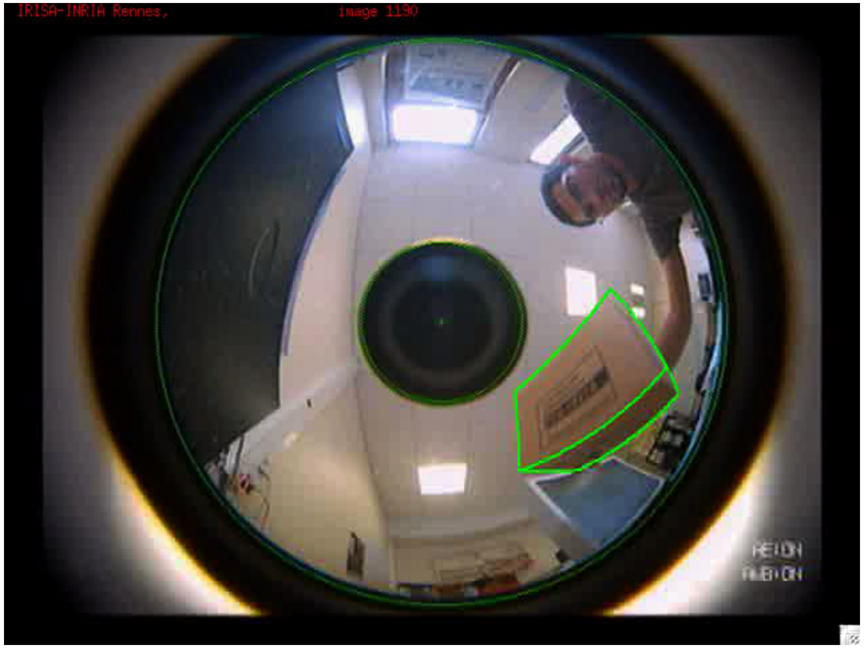
- Projection on a sphere, then on the
- image plane

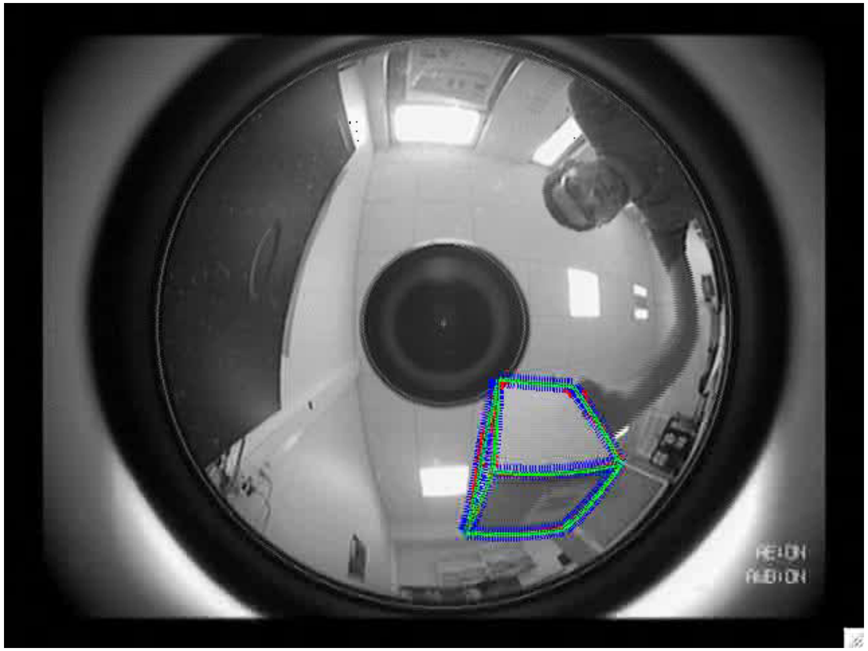
New interaction matrix

- Various parameterizations of the line
  - Plucker coordinates
    [Andreff 01, Adj Abdelkader et al 04]]
  - Intersection of two planes
    [Marchand et al 06]



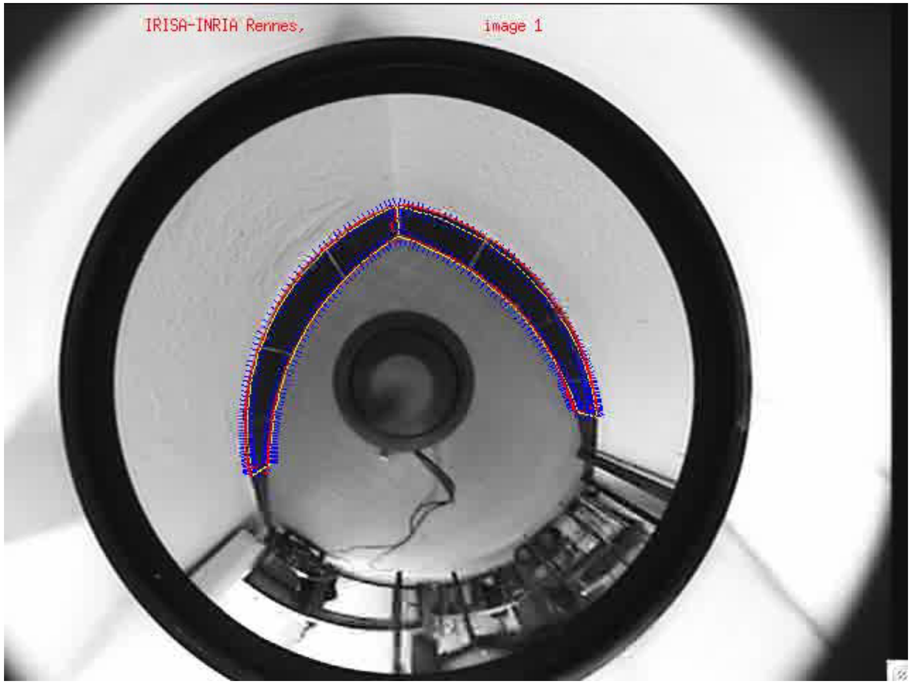
Tracking in central catadioptric cameras

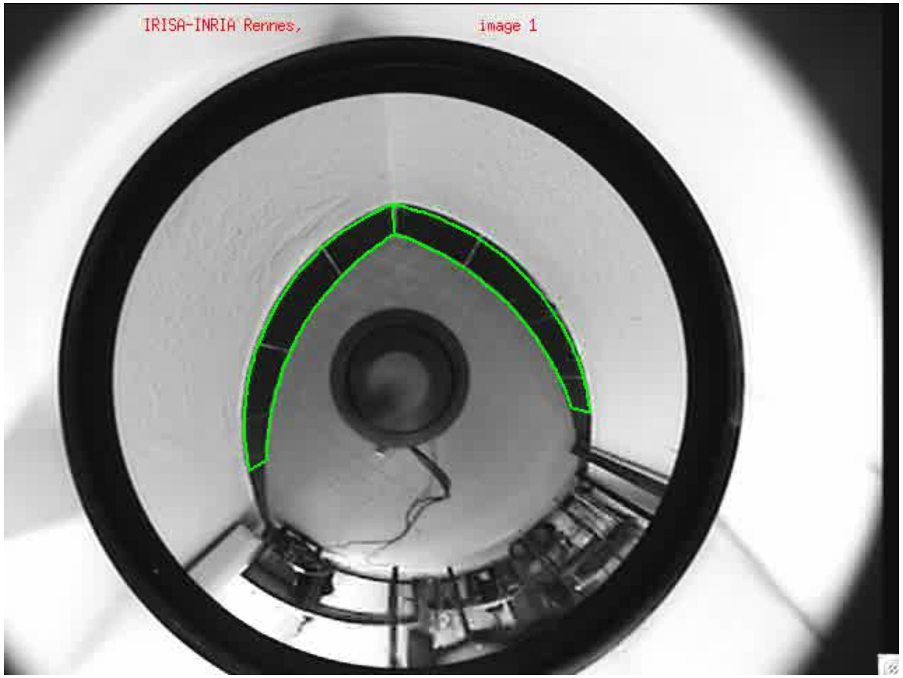




Tracking a box at video rate

Tracking plinths for mobile robotics

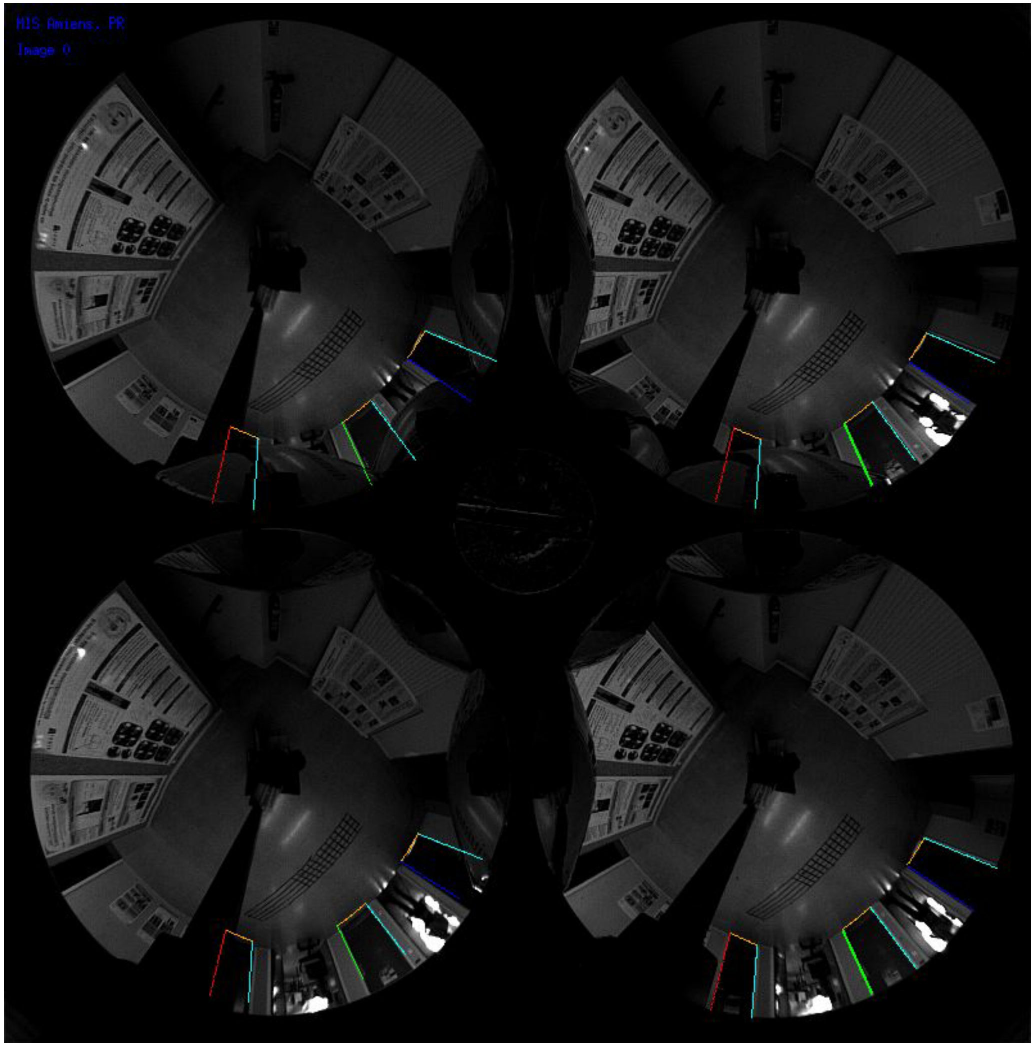




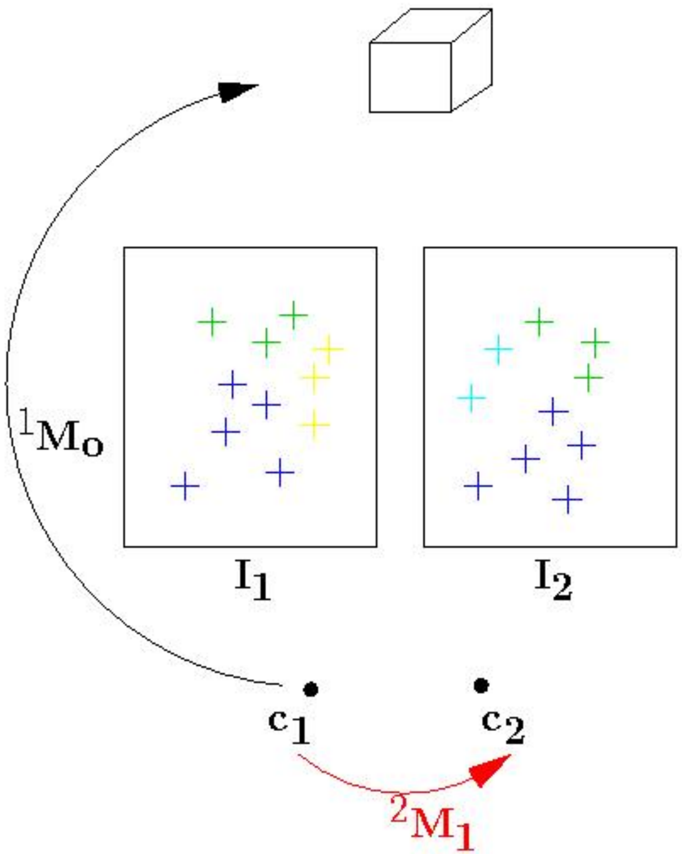
Original images thanks to Lasmea

UMR IRISA

Stereo omni



Second : displacement estimation

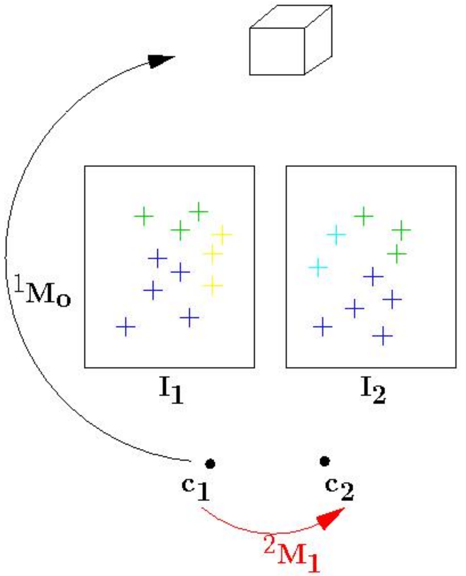


Motion estimation

tr(p2) is the coordinates of a point *transferred* in a reference image according to the camera displacement

$$\Delta = \sum_i (\mathbf{p}_1^i - {}^2tr_1(\mathbf{p}_2))^2$$

Allows to estimate the camera displacement



Advantages

- Implicit spatio-temporal constraints
- Less jitter

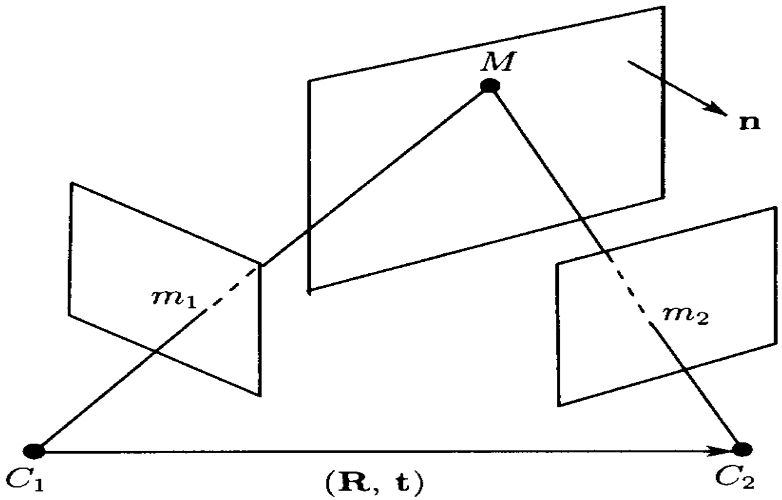
Issue

- Prone to drift if reference image is modified

Case of planar scenes

Planarity constraint

- if the surfaces of the objects are planar, there exists an analytic transformation from the left image coordinates to the right image coordinates.



Case of planar scenes

Let us assume that points belong to a plane $\mathcal{P}(\mathbf{n}, d)$

$$\left.\begin{array}{l} \mathbf{M_1} \in \mathcal{P}(\mathbf{n}, d) \Leftrightarrow \mathbf{n^T M_1} = d \\ \mathbf{M_2} = \mathbf{R M_1} + \mathbf{T} \end{array}\right\} \Rightarrow \mathbf{M_2} = \mathbf{R M_1} + \mathbf{T}\frac{\mathbf{n^T}}{d}\mathbf{M_1}$$

or $\quad Z_1 \mathbf{m_{p1}} = \mathbf{K_1 M_1} \qquad\qquad Z_2 \mathbf{K_2 m_{p2}} = \mathbf{K_2 M_2}$

leading to $\lambda \mathbf{m_{p2}} = \mathbf{H m_{p1}}$

with

$$\mathbf{H} = \mathbf{K_2 R K_1^{-1}} + \mathbf{K_2 t}\frac{\mathbf{n}^\top}{d}\mathbf{K_1^{-1}} \qquad \text{et} \quad \lambda = \frac{Z_2}{Z_1}$$

Motion estimation

Point transfert

- For planar object the transfert is function of the camera displacement **$^2M_1$** and of the homography **$^2H_1$** $$\mathbf{p}_2 = {}^2tr_1(\mathbf{p}_1) = {}^2\mathbf{H}_1\mathbf{p}_1$$

  with $${}^2\mathbf{H}_1 = \mathbf{K}^{-1}\left({}^2\mathbf{R}_1 + \frac{{}^2\mathbf{t}_1}{{}^1d}{}^1\mathbf{n}^\top\right)\mathbf{K}$$

- If **$^1M_0$** is known, estimating the displacement is equivalent to a pose estimation

Solutions

- Estimation of H [Simon, Berger IEEE CGA 02][Benhimane Malis IJRR 07]
- Estimation of R, t [Pressigout Marchand ICPR 04]

Illumination

- May also considered more complex illumination model in Δ

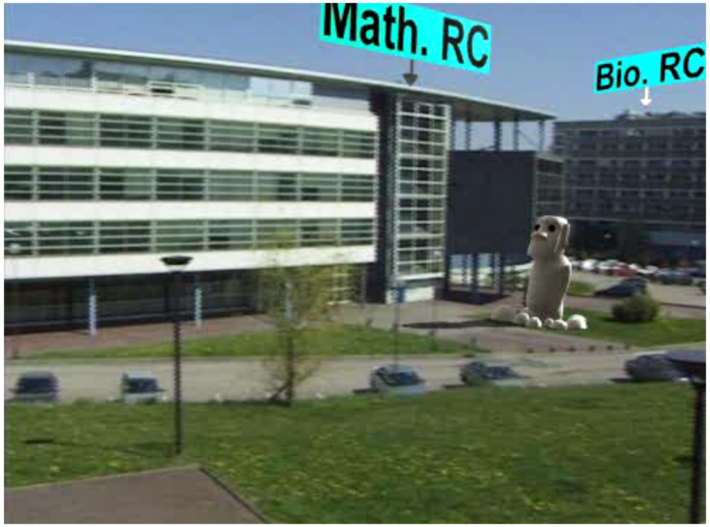
UMR IRISA

Multi-plane tracking

Pose from planar structures

Constraints in the homography estimation

[Simon, Berger IEEE CGA 02]



Virtual visual servoing

Estimation of R and t [Pressigout Marchand ICPR'04]

ESM: Efficient second order minimization

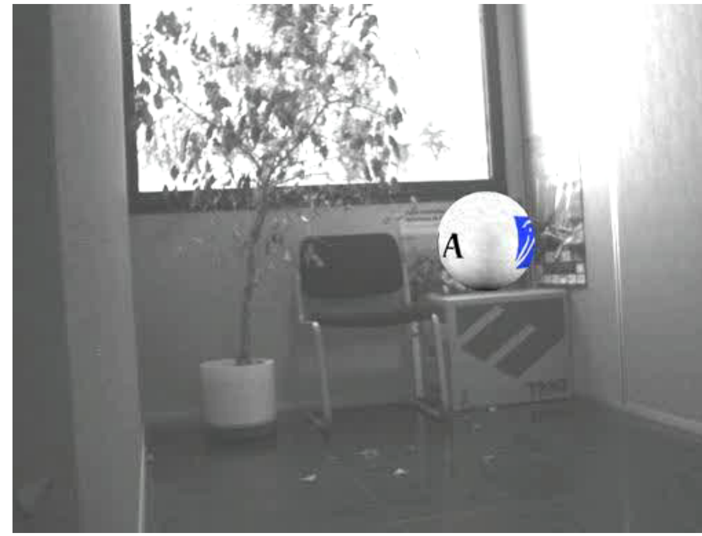
Use all the pixel in the tracked patch

$$\Delta = \sum_i \left(I_1(p_1^i) - I_2({}^2tr_1(p_1^i))\right)^2$$

$$\mathbf{p}_2 = {}^2tr_1(\mathbf{p}_1) = {}^2\mathbf{H}_1\mathbf{p}_1$$

Direct estimation of the homography

[Benhimane Malis IJRR 07]



INRIA SOPHIA ANTIPOLIS

UMR IRISA

Mathematical formulation

$I(x,y,t) =$ brightness at (x,y) at time *t*

Brighntess constancy assumption

$$I(x+\delta x, y+\delta y, t+\delta t) = I(x,y,t)$$

or

$$I(x+\frac{dx}{dt}\delta t, y+\frac{dy}{dt}\delta t, t+\delta t) = I(x,y,t)$$

Optical flow constraint equation (OFCE)

$$\boldsymbol{\nabla} I^\top \dot{\mathbf{x}} + I_t = \frac{dI}{dt}$$

Patch matching: SSD

How do we determine correspondences?

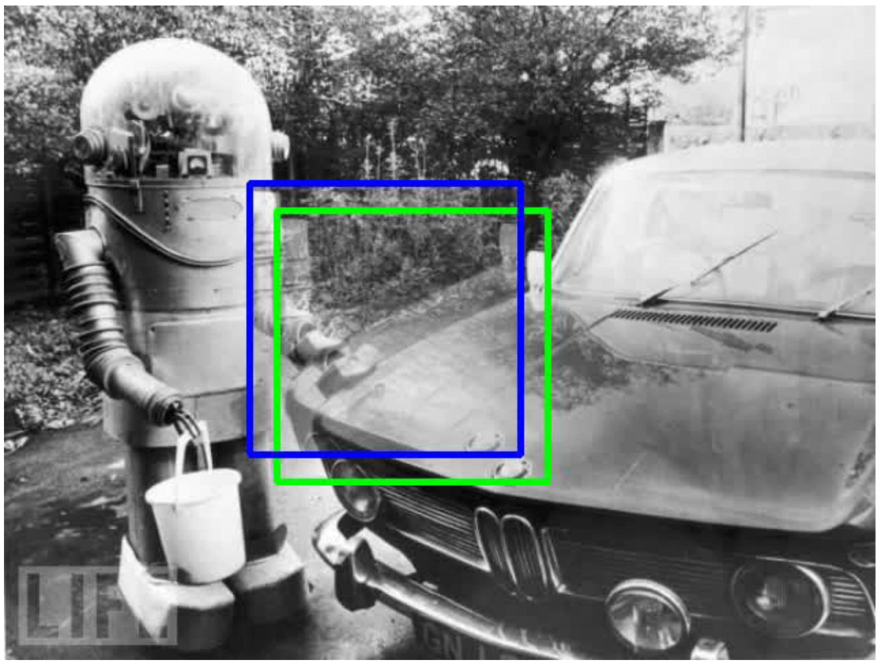
Simplest case

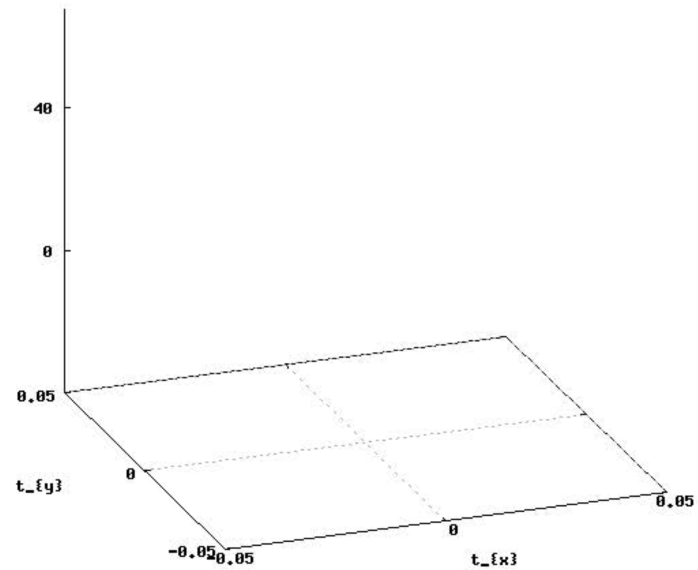
- block matching or SSD (sum squared differences)

$$E(\mathbf{h}) = \sum_{\mathbf{x}} [I(\mathbf{x}+\mathbf{h}) - T(\mathbf{x})]^2$$

Estimating the translational **h** motion between two images

SSD





Original Lucas-Kanade algorithm

Goal is to align a template ***T(x)*** to an input image ***I(x)***

x is a column vector containing image coordinates $(x,y)^T$

Could be also a small window in the image

Set of allowable warps **W(x,p),** where **p** is a vector of parameter, for example, for translation we have

$$\mathbf{W}(\mathbf{x},\mathbf{p}) = \begin{bmatrix} x + p_1 \\ y + p_2 \end{bmatrix}$$

W(x,p) can arbitrarily complex

The best alignment minimizes image dissimilarity

$$\sum_{\mathbf{x}}[I(\mathbf{W}(\mathbf{x},\mathbf{p})) - T(\mathbf{x})]^2$$

UMR IRISA

Original Lucas-Kanade algorithm

$$\sum_{\mathbf{x}} [I(\mathbf{W}(\mathbf{x}, \mathbf{p})) - T(\mathbf{x})]^2$$

is a non linear optimization. The warp **W(x,p)** may be linear but the pixel are, in general, non linear.

Assuming that **p** is known and best increment **Δp** is sought. The modified problem

$$\sum_{\mathbf{x}} [I(\mathbf{W}(\mathbf{x}, \mathbf{p} + \Delta\mathbf{p})) - T(\mathbf{x})]^2$$

Is solved with respect to. When found update

$$\mathbf{p} \longleftarrow \mathbf{p} + \Delta\mathbf{p}$$

Original Lucas-Kanade algorithm

$$\sum_{\mathbf{x}}[I(\mathbf{W}(\mathbf{x},\mathbf{p}+\Delta\mathbf{p}))-T(\mathbf{x})]^2$$

Linearized by performing first order Taylor expansion

$$\sum_{\mathbf{x}}[I(\mathbf{W}(\mathbf{x},\mathbf{p}))+\nabla I\frac{\partial\mathbf{W}}{\partial\mathbf{p}}\Delta\mathbf{p}-T(\mathbf{x})]^2$$

$\nabla I=[\frac{\partial I}{\partial x},\frac{\partial I}{\partial y}]$ is the image gradient computed at W(x,p),

The term $\frac{\partial\mathbf{W}}{\partial\mathbf{p}}$ is the Jacobian of the warp

Derive

$$\sum_{\mathbf{x}} [I(\mathbf{W}(\mathbf{x},\mathbf{p})) + \nabla I \frac{\partial \mathbf{W}}{\partial \mathbf{p}} \Delta\mathbf{p} - T(\mathbf{x})]^2$$

with respect to $\Delta\mathbf{p}$

$$2\sum_{x} \left[\nabla I \frac{\partial \mathbf{W}}{\partial \mathbf{p}}\right]^{\top} [I(\mathbf{W}(\mathbf{x},\mathbf{p})) + \nabla I \frac{\partial \mathbf{W}}{\partial \mathbf{p}} \Delta\mathbf{p} - T(\mathbf{x})]$$

setting equal to zero yelds

$$\Delta\mathbf{p} = \mathbf{H}^{-}1 \left[\nabla I \frac{\partial \mathbf{W}}{\partial \mathbf{p}}\right]^{\top} [T(\mathbf{x}) - I(\mathbf{W}(\mathbf{x},\mathbf{p}))]$$

where $\mathbf{H}$ is the Hessian matrix

$$\mathbf{H} = \sum_{\mathbf{x}} \left[\nabla I \frac{\partial \mathbf{W}}{\partial \mathbf{p}}\right]^{\top} \left[\nabla I \frac{\partial \mathbf{W}}{\partial \mathbf{p}}\right]^{\top}$$

KLT summary

iterate

1. Warp I with $\mathbf{W}(\mathbf{x}, \mathbf{p})$
2. Warp the gradient ∇I with $\mathbf{W}(\mathbf{x}, \mathbf{p})$
3. evaluate the Jacobian $\frac{\partial \mathbf{W}}{\partial \mathbf{p}}$ at $(\mathbf{x}, \mathbf{p})$ and compute the steepest descent image $\nabla I \frac{\partial \mathbf{W}}{\partial \mathbf{p}}$
4. Compute the Hessian $\mathbf{H}$
5. Compute $\Delta \mathbf{p} = \mathbf{H}^{-}1 \left[\nabla I \frac{\partial \mathbf{W}}{\partial \mathbf{p}}\right]^{\top} [T(\mathbf{x}) - I(\mathbf{W}(\mathbf{x}, \mathbf{p}))]$
6. update the parameters $\mathbf{p} \longleftarrow \mathbf{p} + \delta \mathbf{p}$

until $\|\Delta \mathbf{p}\| < \varepsilon$

Consider translation $\mathbf{W}(\mathbf{x};\mathbf{p}) = \begin{bmatrix} x + p_1 \\ y + p_2 \end{bmatrix}$. The Jacobian is then

$$\frac{\partial \mathbf{W}}{\partial \mathbf{p}} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$\begin{aligned}
\mathtt{H} &= \sum_{\mathbf{x}} \left[\nabla I \ \frac{\partial \mathbf{W}}{\partial \mathbf{p}}\right]^\top \left[\nabla I \ \frac{\partial \mathbf{W}}{\partial \mathbf{p}}\right] \\
&= \sum_{\mathbf{x}} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{\partial I}{\partial x} \\ \frac{\partial I}{\partial y} \end{bmatrix} [\frac{\partial I}{\partial x}, \frac{\partial I}{\partial x}] \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \\
&= \sum_{\mathbf{x}} \begin{bmatrix} \left(\frac{\partial I}{\partial x}\right)^2 & \frac{\partial I^2}{\partial x \partial y} \\ \frac{\partial I^2}{\partial x \partial y} & \left(\frac{\partial I}{\partial y}\right)^2 \end{bmatrix}
\end{aligned}$$

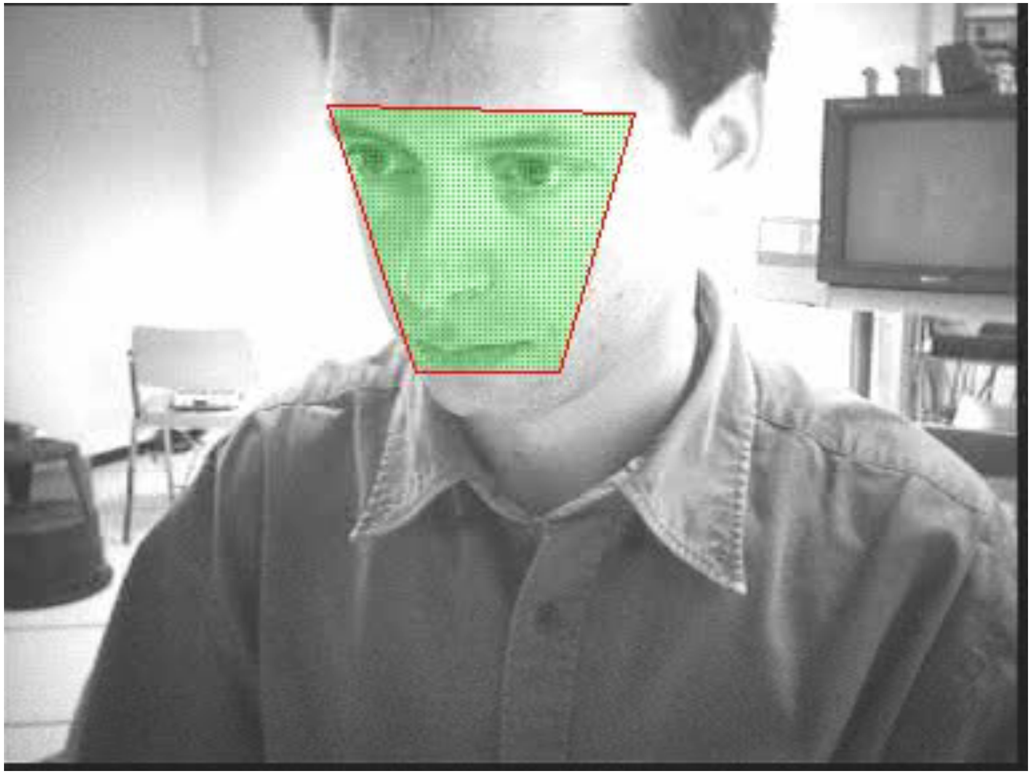
The image windows with varying derivatives in both directions. Homeogeneous areas are clearly not suitable. Texture oriented mostly in one direction only would cause instability for this translation.

$$\Delta \mathbf{p} = \mathtt{H}^{-1} \sum_{\mathbf{x}} \left[\nabla I \; \frac{\partial \mathbf{W}}{\partial \mathbf{p}} \right]^{\top} \left[T(\mathbf{x}) - I(\mathbf{W}(\mathbf{x}; \mathbf{p})) \right]$$

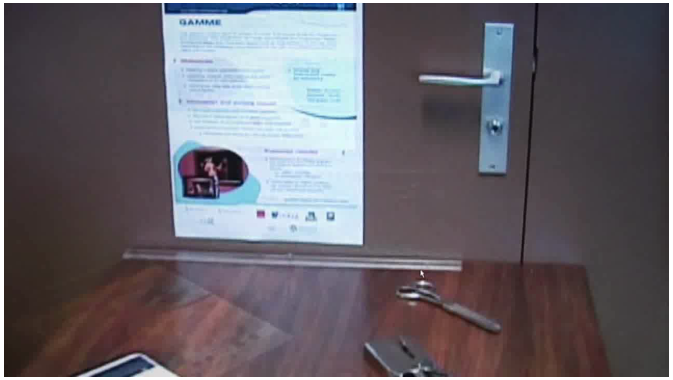
where $\mathtt{H}$ is the Hessian matrix

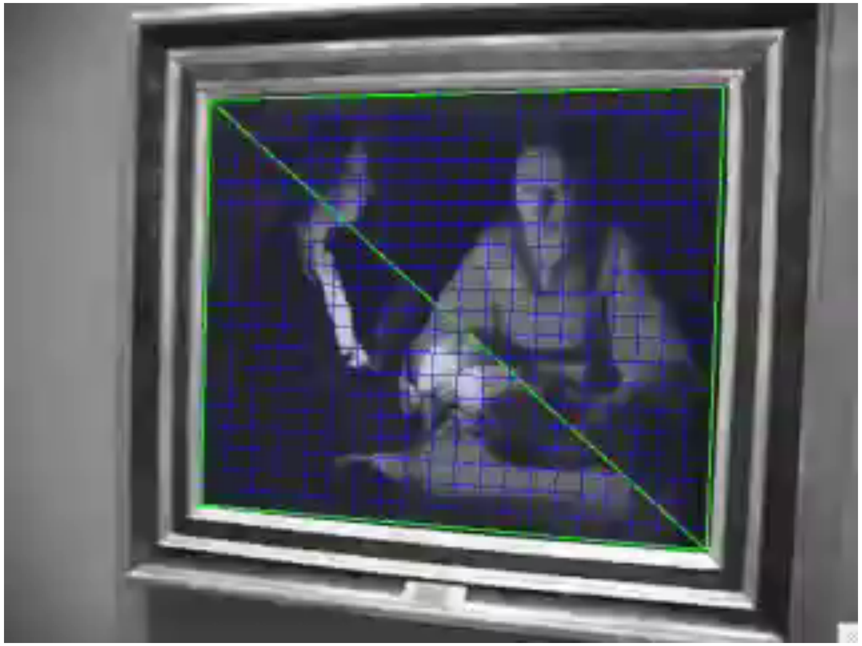
$$\mathtt{H} = \sum_{\mathbf{x}} \left[\nabla I \; \frac{\partial \mathbf{W}}{\partial \mathbf{p}} \right]^{\top} \left[\nabla I \; \frac{\partial \mathbf{W}}{\partial \mathbf{p}} \right]$$

UR1 tracking…




La Tour, ultraviolet

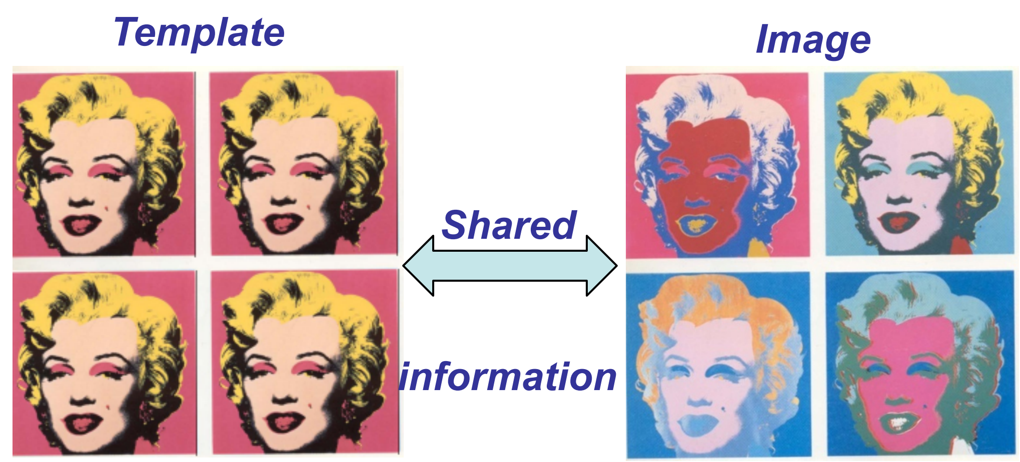




UMR IRISA

MI based tracking

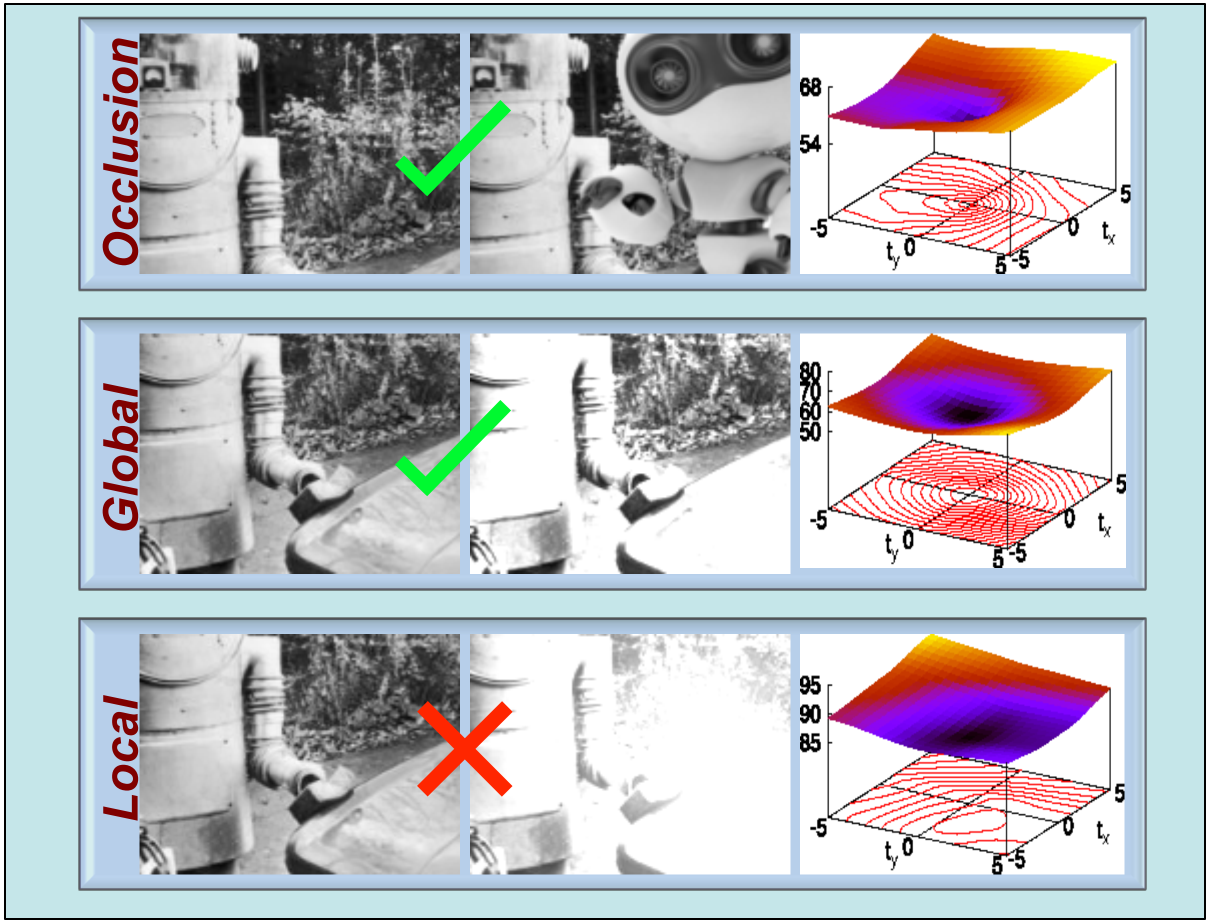
- Proposed template-based tracker :
  - Does not use directly the intensity
  - Uses the “information” of the template
- Properties:
  - Robust
  - Accurate
  - Efficient



Tracking approaches

Template-based

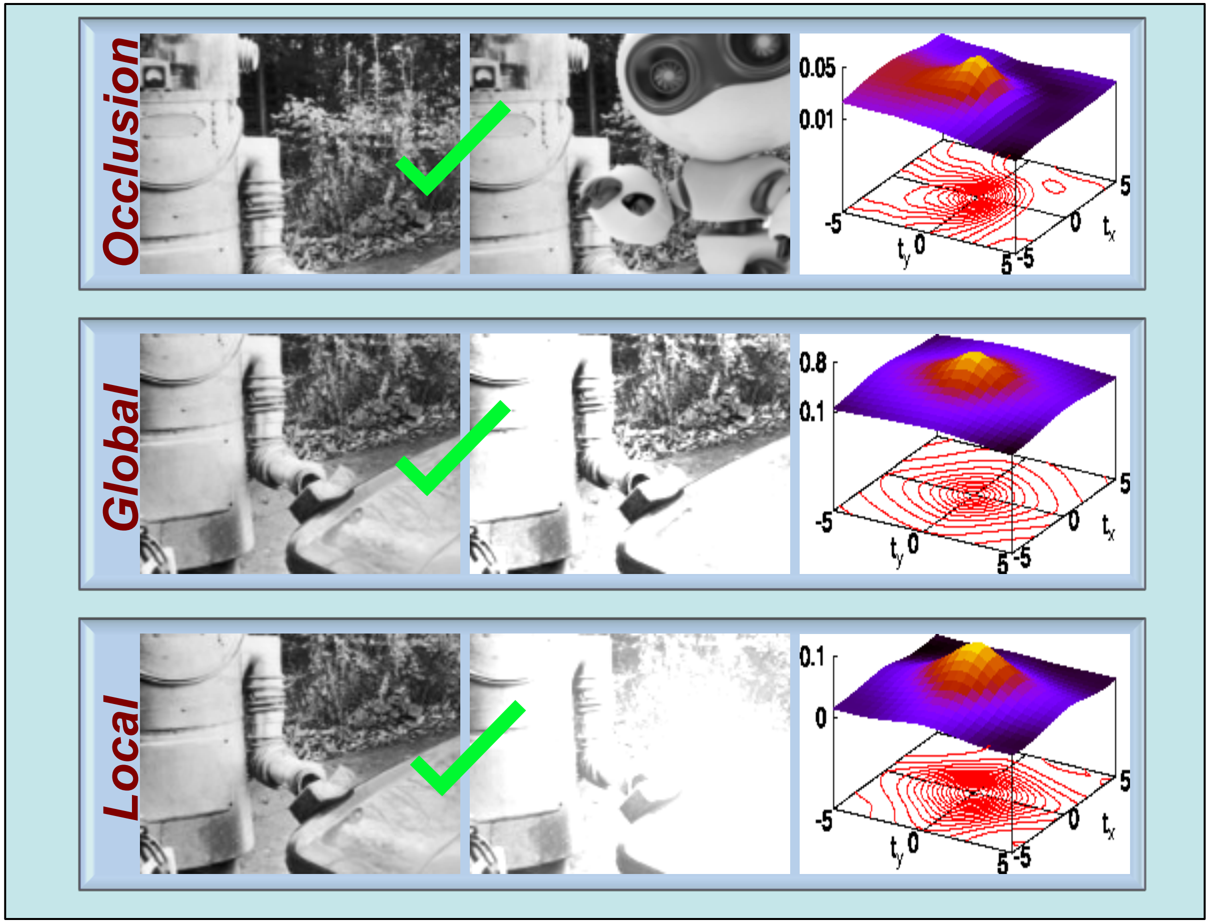
- SSD [Lucas 1981]
- MI [Viola 1995]



Tracking approaches

Template-based

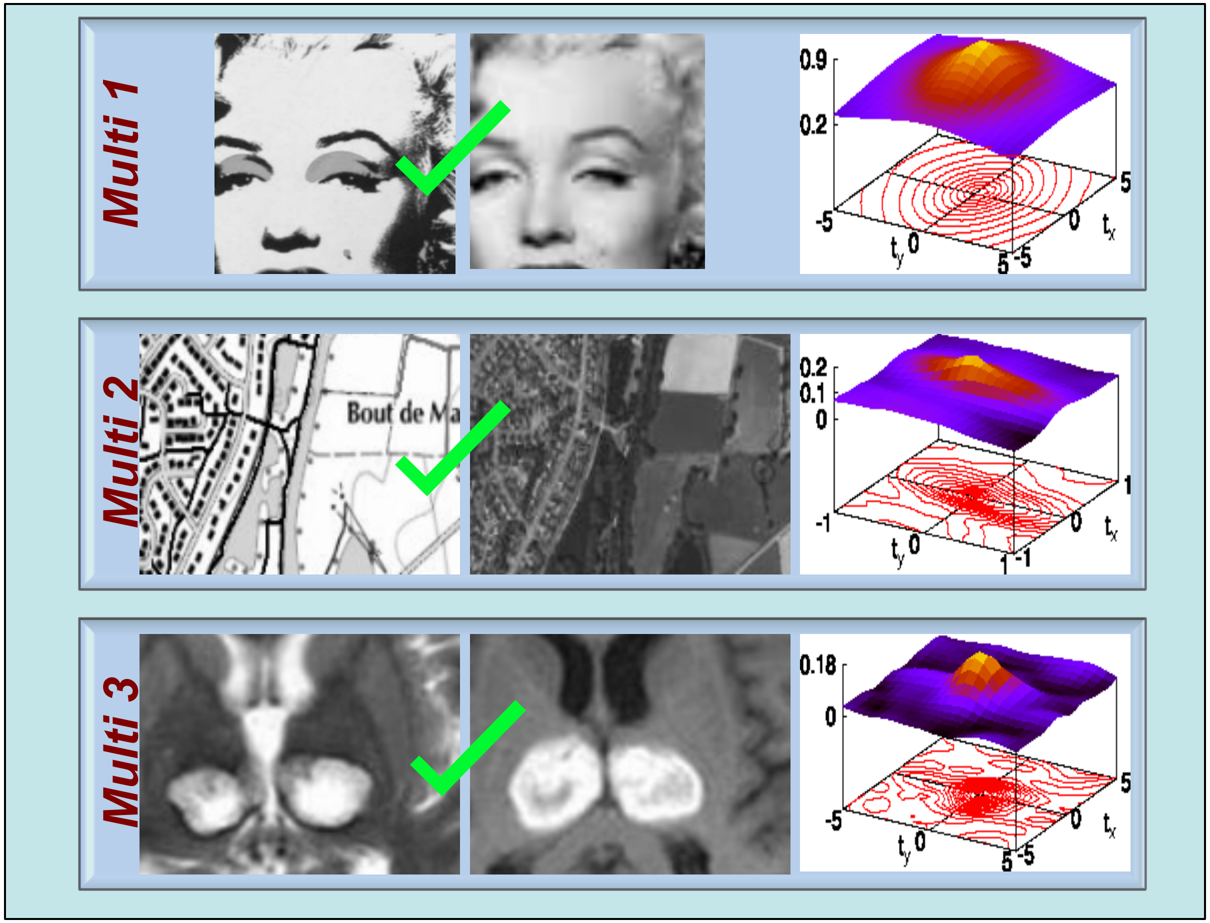
- SSD [Lucas 1981]
- MI [Viola 1995]



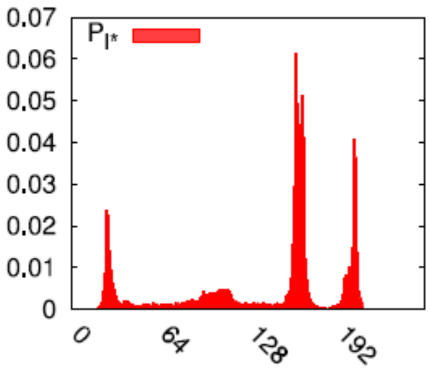
Tracking approaches

Template-based

- SSD [Lucas 1981]
- MI [Viola 1995]

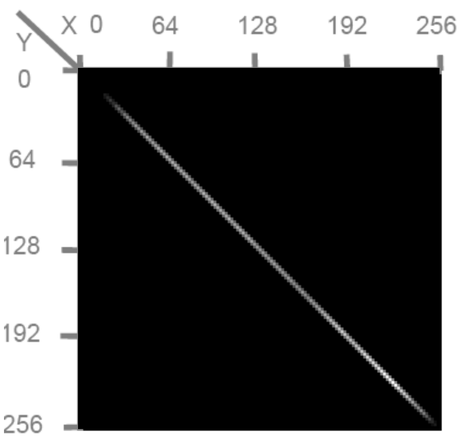


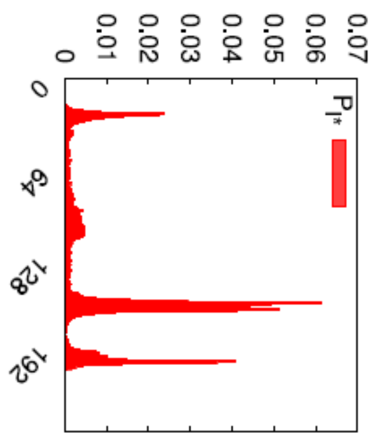
How to measure the mutual information



- Histogram $p_I(i)$:
  - Frequency of an intensity i in the image I
- Joint histogram $p_{II^*}(i,j)$:
  - Frequency of a couple of intensities (i,j) in the couple of images (I,I^*)

$\Rightarrow$ Provides the spatial information



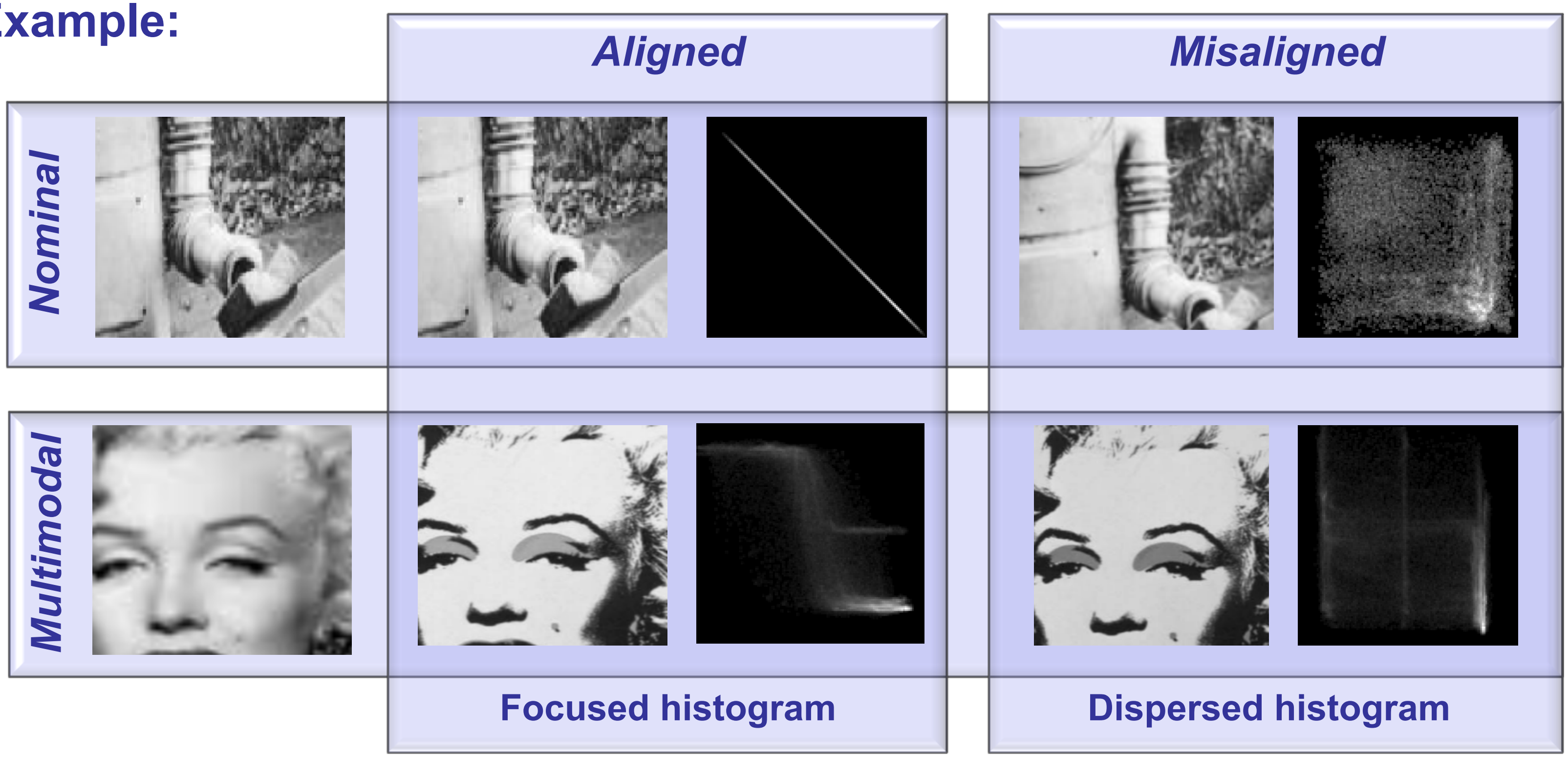


How to measure the mutual information

- MI depends on the dispersion of the joint histogram $p_{II^*}(i,j)$:

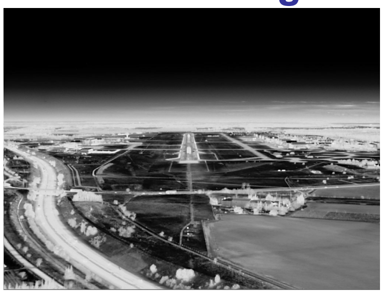
$$MI(I,I^*) = \sum_{i,j} p_{ij}(i,j) \log \left(\frac{p_{ij}(i,j)}{p_i(i)p_j(j)} \right)$$

- Example:

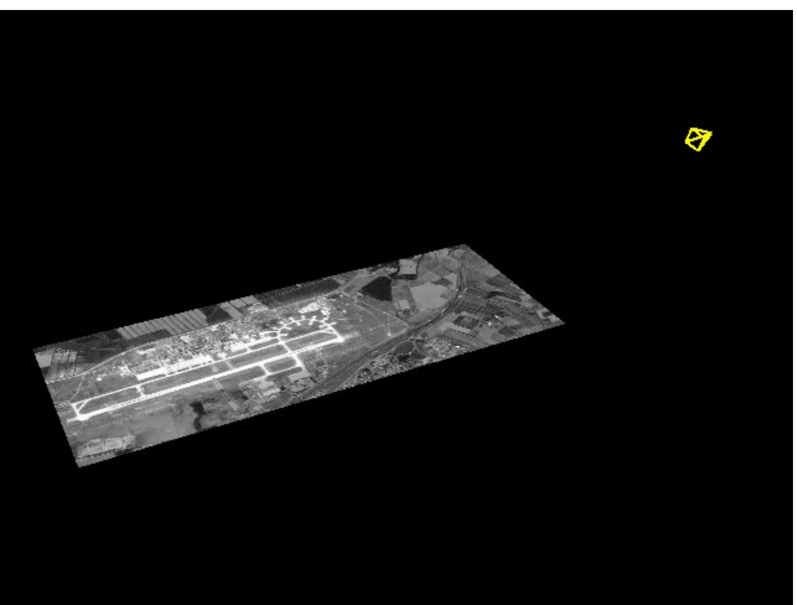


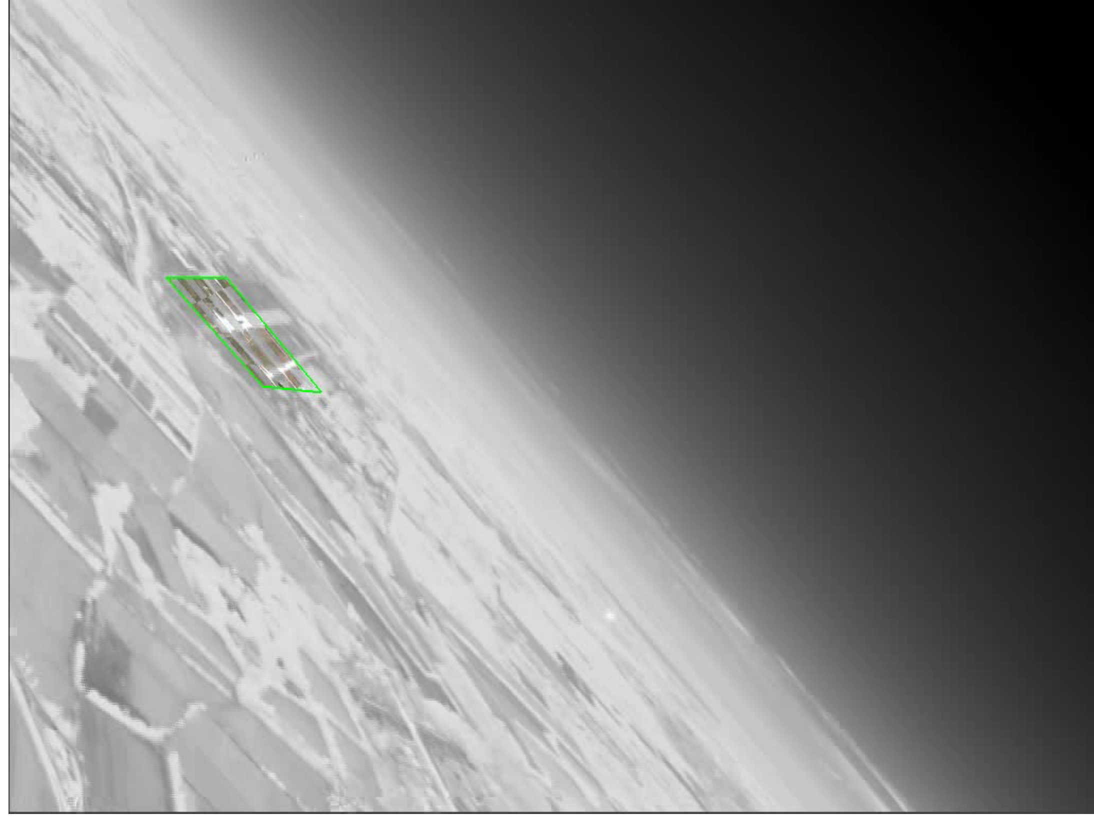
Reference image

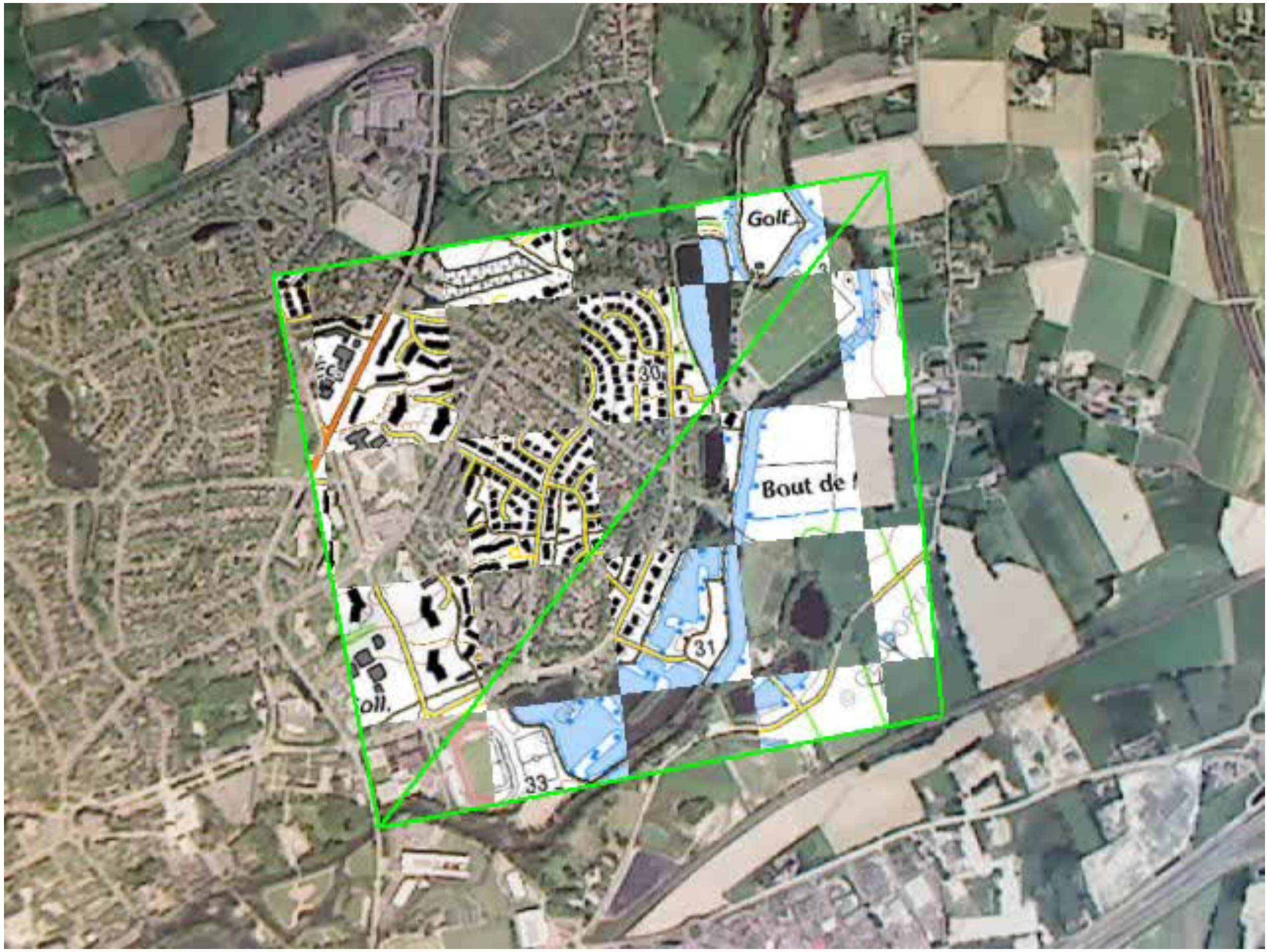
Current image

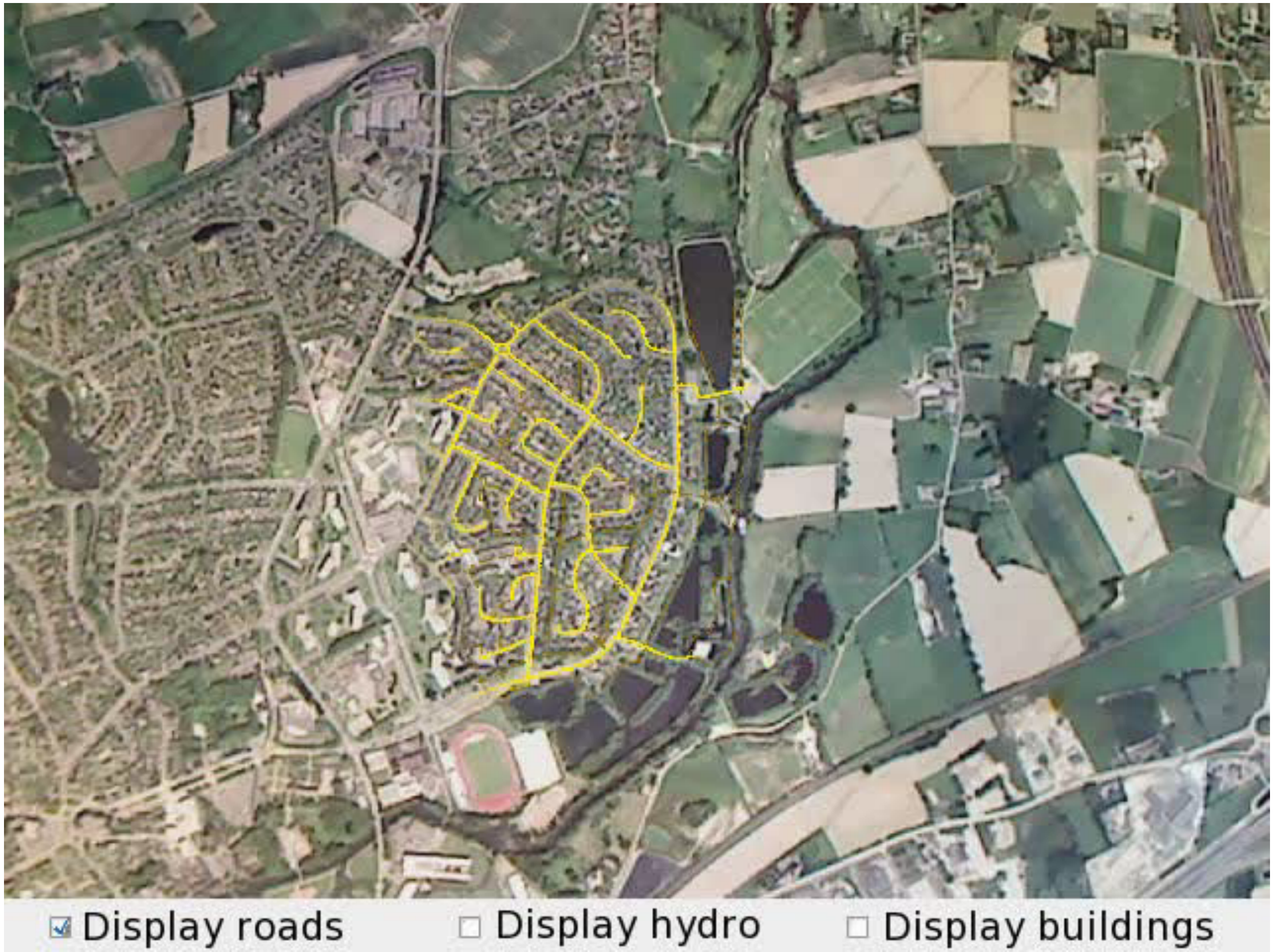


Localization






Golf
Bout de
30
31
33


Display roads
Display hydro
Display buildings

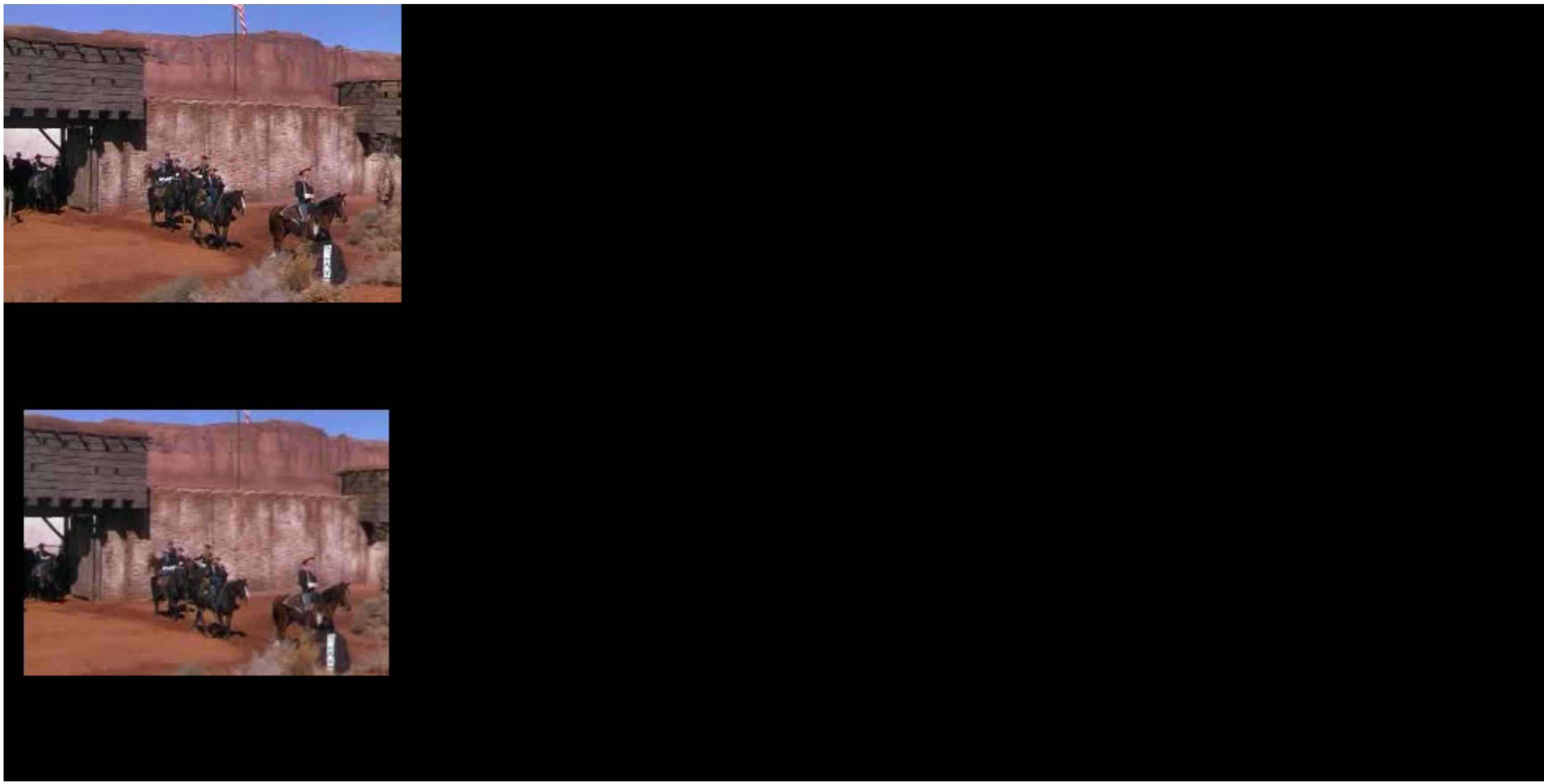

PLANET ROBOT


PLANET ROBOT
WIND-UP
MECHANISM


PLANET ROBOT
WIND-UP
MECHANISM

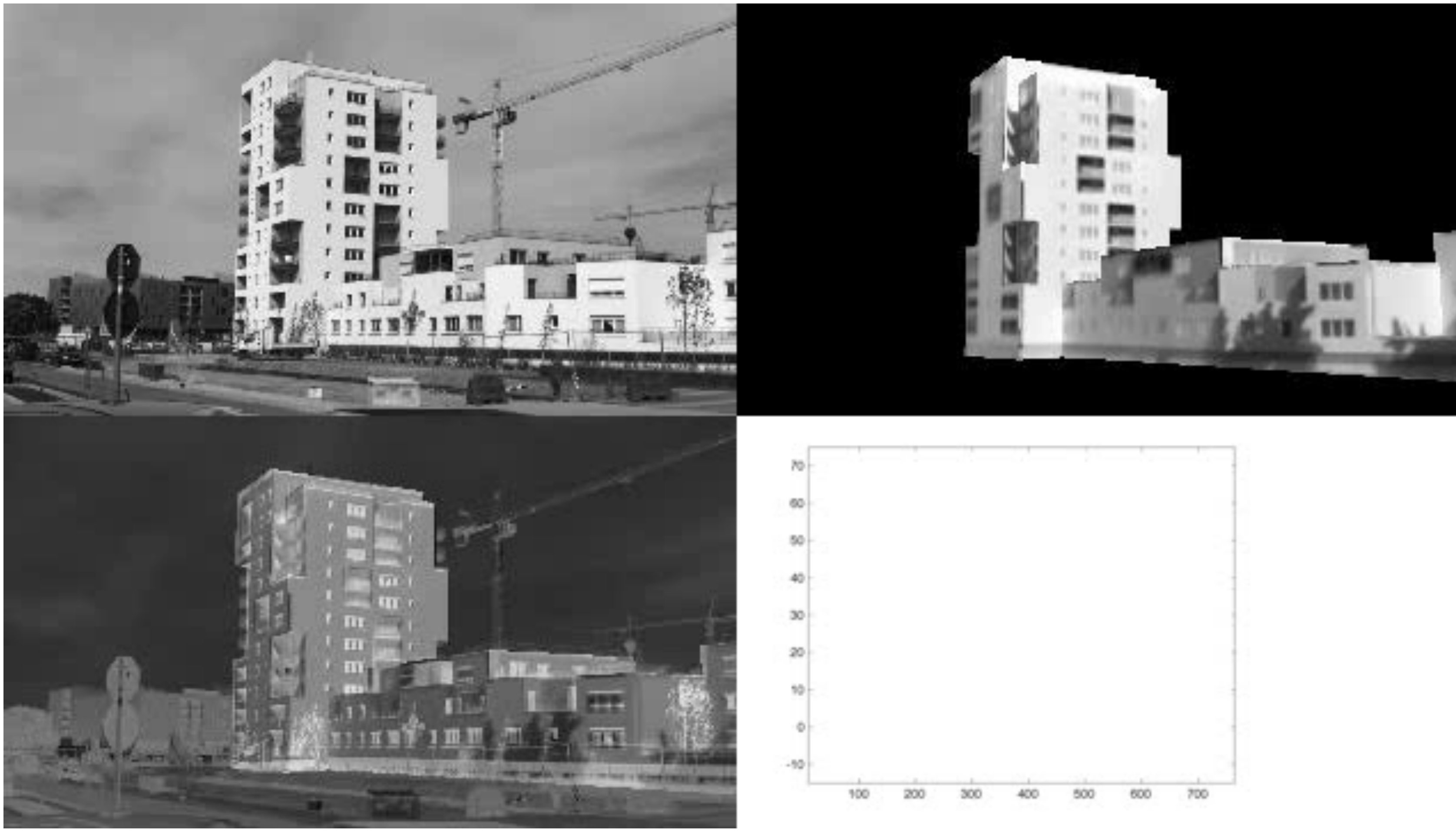
UMR IRISA

Application to mosaicing

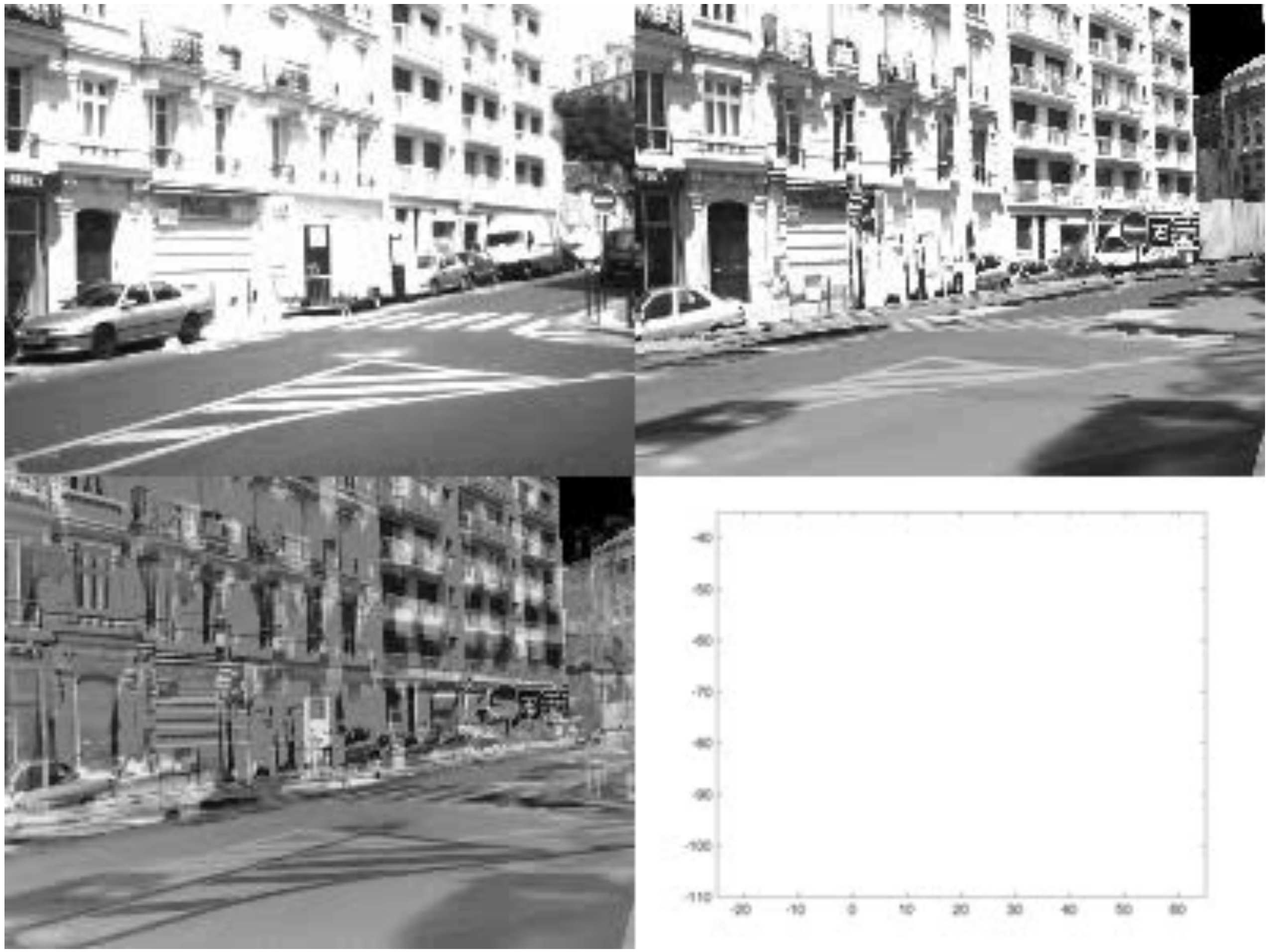


Dense localization

Dense localization



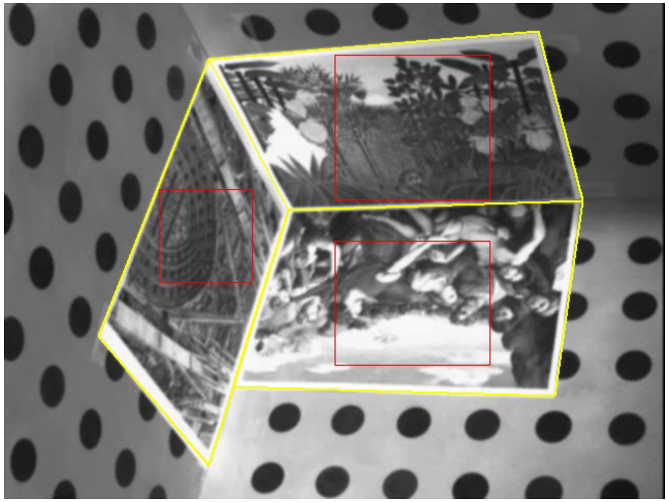
Dense localization



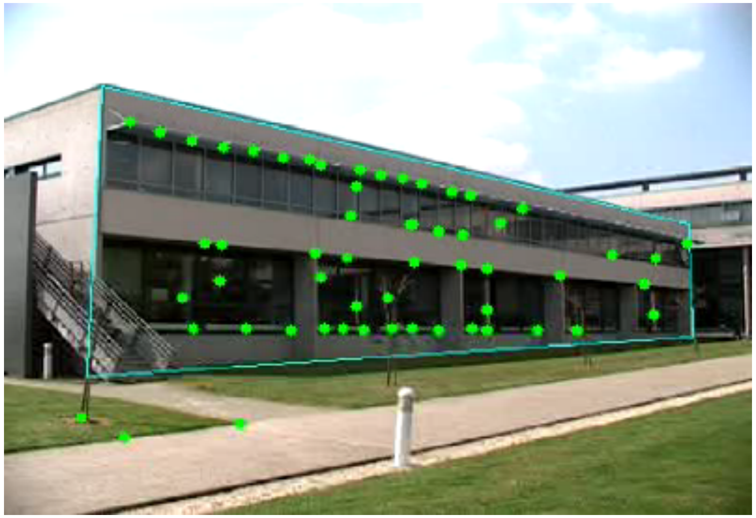
SLAM

Simultaneous Localization and Mapping

- On-line model construction
- 3D localization
- Recursive estimation process (EKF)



[Servant 07]



Parallel Tracking and Mapping for Small AR Workspaces

Extra video results made for ISMAR 2007 conference

Georg Klein and David Murray
Active Vision Laboratory
University of Oxford

http://www.irisa.fr/lagadic

UMR IRISA