

Homography-based Tracking

Theory and Applications

Selim Benhimane

selim.benhimane@sophia.inria.fr, benhiman@in.tum.de

Problem Statement

- A mobile camera and/or a mobile object.
- Follow in real-time the object in the image sequence.
- Estimate the object transformation in the image.
- Estimate the relative camera/object displacement.

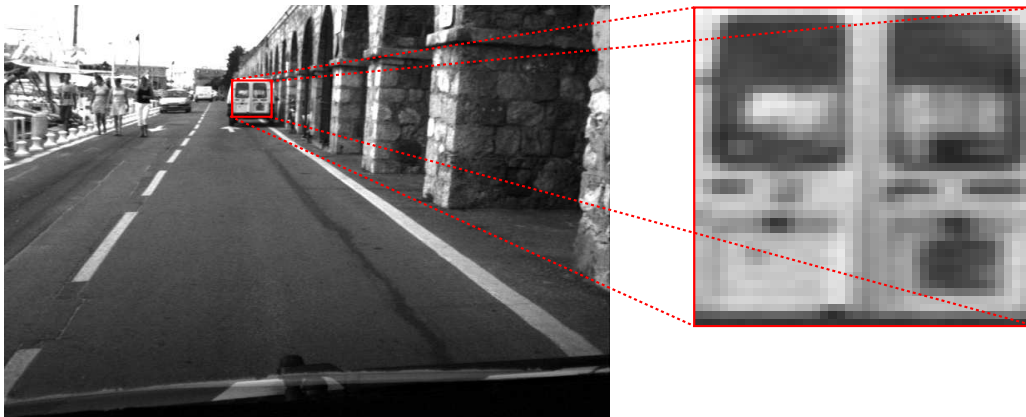
Problem Statement

- A mobile camera and/or a mobile object.
- Follow in real-time the object in the image sequence.
- Estimate the object transformation in the image.
- Estimate the relative camera/object displacement.



Problem Statement

- A mobile camera and/or a mobile object.
- Follow in real-time the object in the image sequence.
- Estimate the object transformation in the image.
- Estimate the relative camera/object displacement.



Problem Statement

- A mobile camera and/or a mobile object.
- Follow in real-time the object in the image sequence.
- Estimate the object transformation in the image.
- Estimate the relative camera/object displacement.



PART I : Theoretical Background

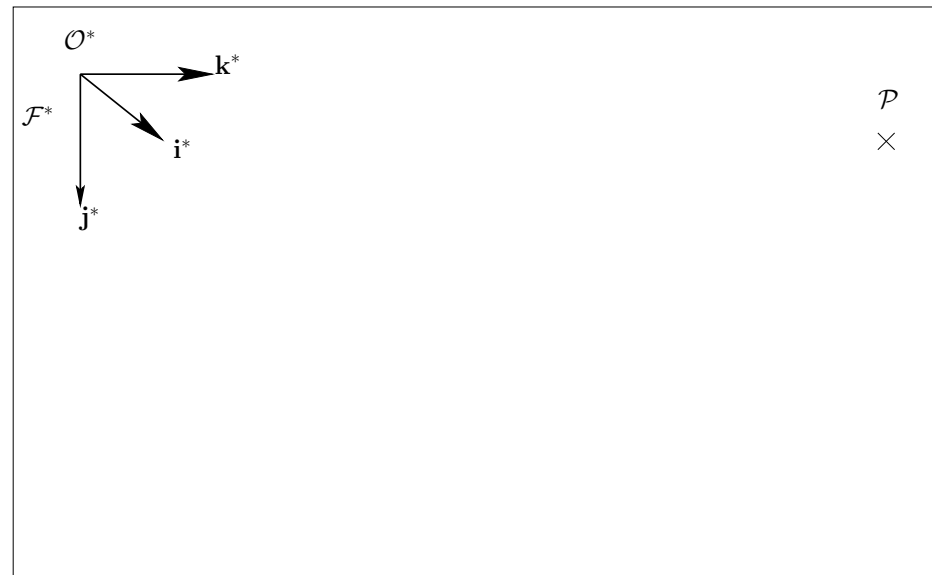
- 3D transformations
- Perspective projection
- Projective transformation
- Homography parameterization
- The camera model
- Image projective transformation

3D transformations

- The 3D space is considered Cartesian, i.e., an Euclidean space with an standard orthonormal frame.

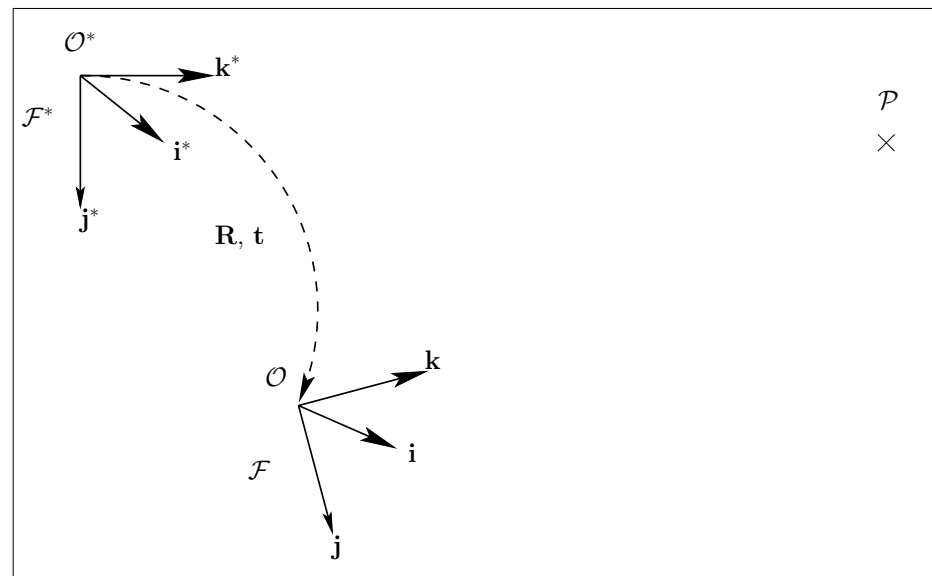
3D transformations

- Let \mathcal{P} be a 3D point and $\mathcal{F}^* = (\mathcal{O}^*, \mathbf{i}^*, \mathbf{j}^*, \mathbf{k}^*)$ be the “reference frame” of the Cartesian space.
- In the frame \mathcal{F}^* , \mathcal{P} has the coordinates $\mathcal{X}^* = [X^* \ Y^* \ Z^*]^\top \in \mathbb{R}^3$.



3D transformations

- Let \mathcal{P} be a 3D point and $\mathcal{F}^* = (\mathcal{O}^*, \mathbf{i}^*, \mathbf{j}^*, \mathbf{k}^*)$ be the “reference frame” of the Cartesian space.
- In the frame \mathcal{F}^* , \mathcal{P} has the coordinates $\mathcal{X}^* = [X^* \ Y^* \ Z^*]^\top \in \mathbb{R}^3$.
- Let \mathbf{R} be the rotation matrix and \mathbf{t} be the translation vector between \mathcal{F}^* and a second frame $\mathcal{F} = (\mathcal{O}, \mathbf{i}, \mathbf{j}, \mathbf{k})$ called the “current frame”.



3D transformations

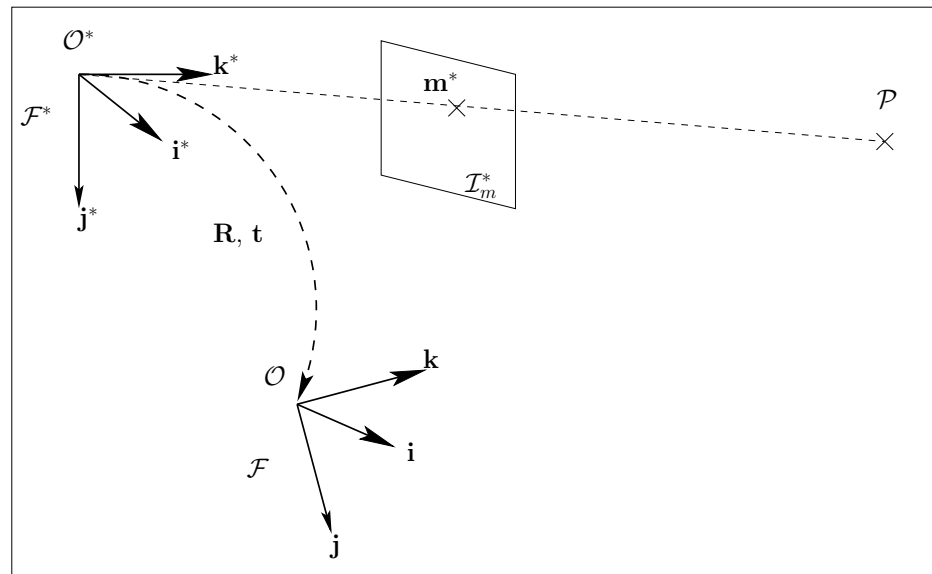
- Let \mathcal{P} be a 3D point and $\mathcal{F}^* = (\mathcal{O}^*, \mathbf{i}^*, \mathbf{j}^*, \mathbf{k}^*)$ be the “reference frame” of the Cartesian space.
- In the frame \mathcal{F}^* , \mathcal{P} has the coordinates $\boldsymbol{\mathcal{X}}^* = [X^* \ Y^* \ Z^*]^\top \in \mathbb{R}^3$.
- Let \mathbf{R} be the rotation matrix and \mathbf{t} be the translation vector between \mathcal{F}^* and a second frame $\mathcal{F} = (\mathcal{O}, \mathbf{i}, \mathbf{j}, \mathbf{k})$ called the “current frame”.
- \mathbf{R} is a (3×3) matrix: $\mathbf{R} \in \text{SO}(3)$ the Special Orthogonal group ($\mathbf{R}\mathbf{R}^\top = \mathbf{I}$).
- \mathbf{t} is a (3×1) vector: $\mathbf{t} \in \mathbb{R}^3$.
- In the frame \mathcal{F} , \mathcal{P} has the coordinates $\boldsymbol{\mathcal{X}} = [X \ Y \ Z]^\top \in \mathbb{R}^3$ verifying:

$$\boldsymbol{\mathcal{X}} = \mathbf{R}\boldsymbol{\mathcal{X}}^* + \mathbf{t}$$

Perspective projection

- We define the “reference metric image”, \mathcal{I}_m^* , the plane perpendicular to the axis \mathbf{k}^* situated at $1m$ from the projection center \mathcal{O}^* .
- The point \mathcal{P} is projected on the \mathcal{I}_m^* using the perspective model in a projective point $\mathbf{m}^* \in \mathbb{P}^2$ that has the homogenous metric coordinates $\mathbf{m}^* = [x^* \ y^* \ 1]^\top$ where:

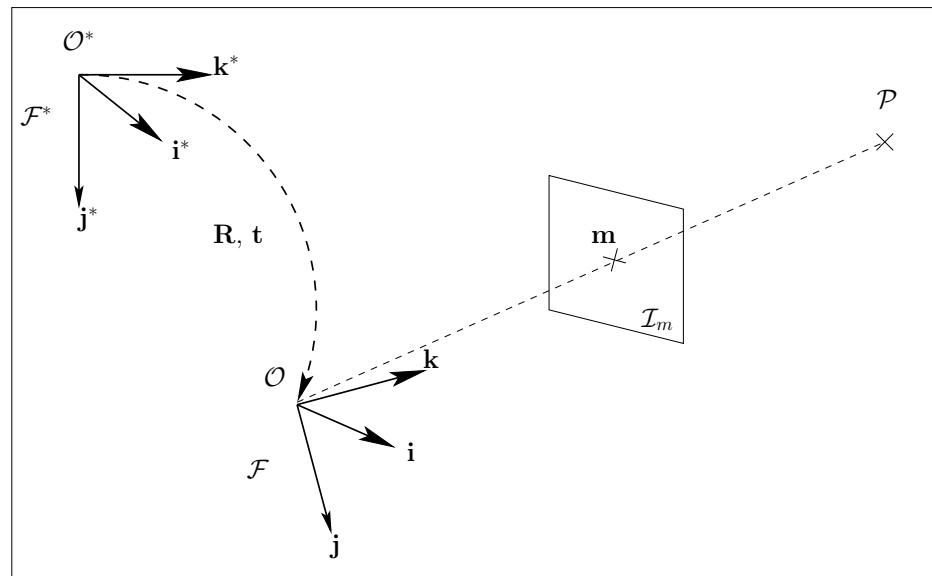
$$\mathbf{m}^* = Z^{*-1} \boldsymbol{\chi}^*$$



Perspective projection

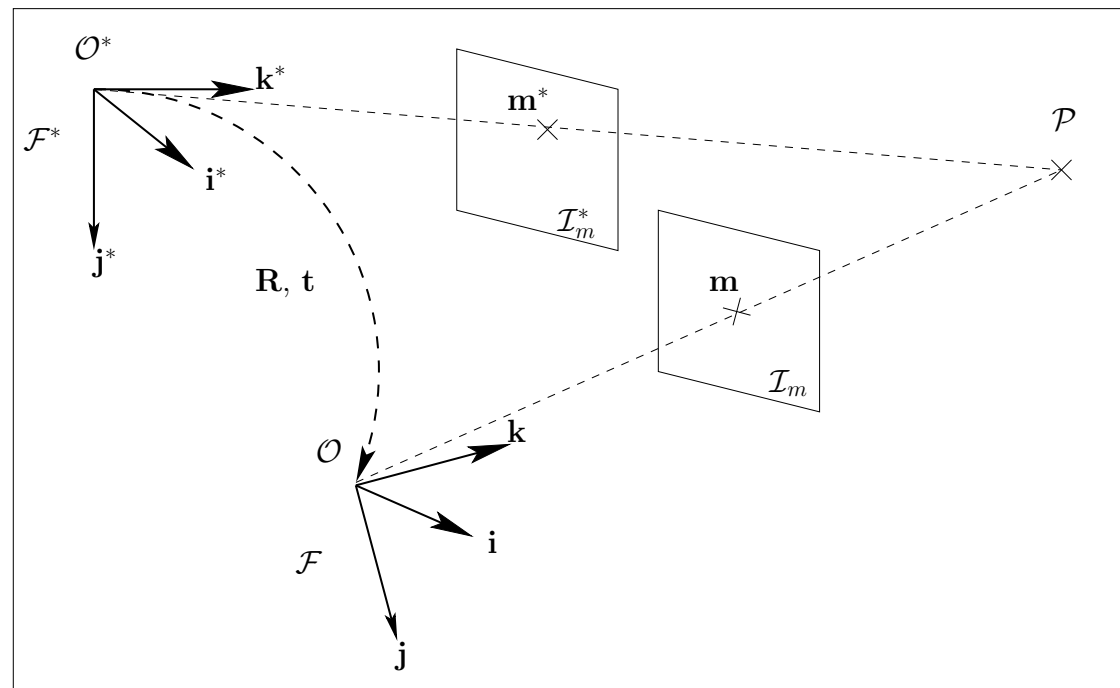
- Similarly, we define the “current metric image”, \mathcal{I}_m , the plane perpendicular to the axis k situated at 1_m from the projection center \mathcal{O} .
- The point \mathcal{P} is projected on the \mathcal{I}_m using the perspective model in a projective point $\mathbf{m} \in \mathbb{P}^2$ that has the homogenous metric coordinates $\mathbf{m} = [x \ y \ 1]^\top$ where:

$$\mathbf{m} = Z^{-1} \boldsymbol{\chi}$$



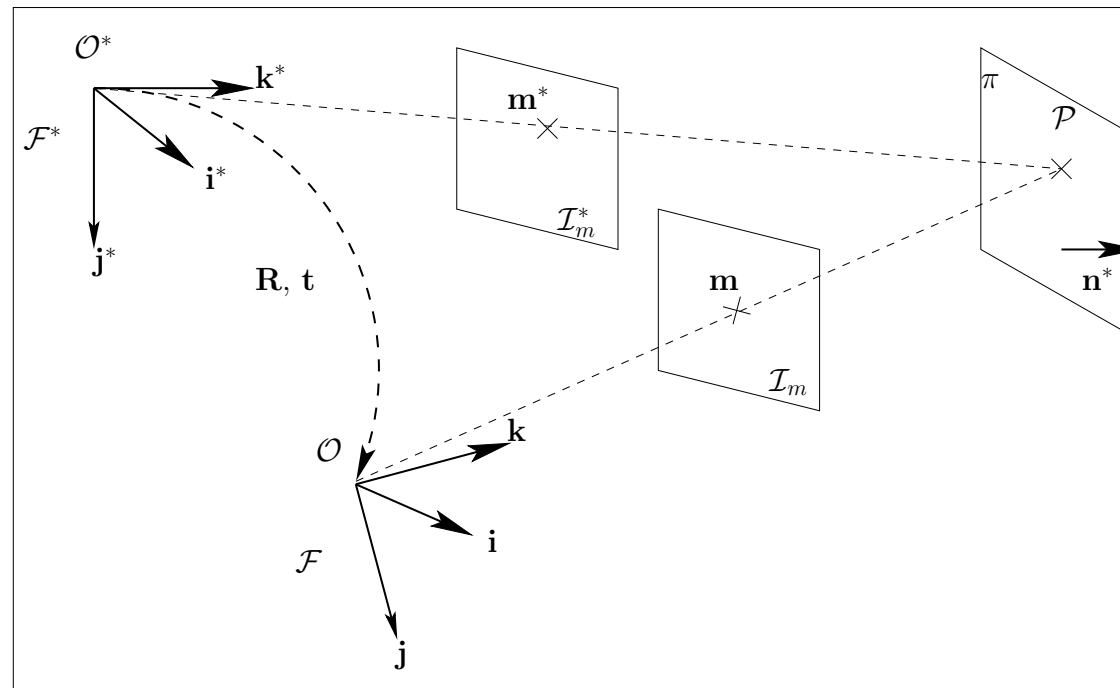
Perspective projection

- m^* and m are “2 corresponding points” since they are the projection of the same 3D point \mathcal{P} .



Projective transformation

- Suppose that we have $\mathcal{P} \in \pi$, where π is a plane with a normal vector \mathbf{n}^* (expressed in \mathcal{F}^*) and its distance to \mathcal{O}^* is d^* .
- If we suppose that \mathbf{n}^* verifies: $\|\mathbf{n}^*\| = \sqrt{\mathbf{n}^{*\top} \mathbf{n}^*} = d^{*-1}$, the equation of the plane π becomes: $\mathbf{n}^{*\top} \mathcal{X}^* = 1$.



Projective transformation

- Using the previous equations, we can establish a relationship between two corresponding points \mathbf{m} and \mathbf{m}^* :

$$\frac{Z}{Z^*} \mathbf{m} = \mathbf{H}_0 \mathbf{m}^* \quad \text{where} \quad \mathbf{H}_0 = \mathbf{R} + \mathbf{t} \mathbf{n}^{*\top}$$

- ◆ **Demonstration:**

Projective transformation

- Using the previous equations, we can establish a relationship between two corresponding points \mathbf{m} and \mathbf{m}^* :

$$\frac{Z}{Z^*} \mathbf{m} = \mathbf{H}_0 \mathbf{m}^* \quad \text{where} \quad \mathbf{H}_0 = \mathbf{R} + \mathbf{t} \mathbf{n}^{*\top}$$

- ◆ **Demonstration:**
- ◆ **3D transformation: $\mathcal{X} = \mathbf{R} \mathcal{X}^* + \mathbf{t}$.**

Projective transformation

- Using the previous equations, we can establish a relationship between two corresponding points \mathbf{m} and \mathbf{m}^* :

$$\frac{Z}{Z^*} \mathbf{m} = \mathbf{H}_0 \mathbf{m}^* \quad \text{where} \quad \mathbf{H}_0 = \mathbf{R} + t \mathbf{n}^{*\top}$$

- ◆ **Demonstration:**
- ◆ **3D transformation: $\mathcal{X} = \mathbf{R} \mathcal{X}^* + \mathbf{t}$.**
- ◆ **The plane equation: $\mathbf{n}^{*\top} \mathcal{X}^* = 1$.**

Projective transformation

- Using the previous equations, we can establish a relationship between two corresponding points \mathbf{m} and \mathbf{m}^* :

$$\frac{Z}{Z^*} \mathbf{m} = \mathbf{H}_0 \mathbf{m}^* \quad \text{where} \quad \mathbf{H}_0 = \mathbf{R} + \mathbf{t} \mathbf{n}^{*\top}$$

- ◆ Demonstration:
- ◆ 3D transformation: $\mathcal{X} = \mathbf{R} \mathcal{X}^* + \mathbf{t}$.
- ◆ The plane equation: $\mathbf{n}^{*\top} \mathcal{X}^* = 1$.
- ◆ Perspective projection: $\mathbf{m}^* = Z^{*-1} \mathcal{X}^*$ and $\mathbf{m} = Z^{-1} \mathcal{X}$.

Projective transformation

- Using the previous equations, we can establish a relationship between two corresponding points \mathbf{m} and \mathbf{m}^* :

$$\frac{Z}{Z^*} \mathbf{m} = \mathbf{H}_0 \mathbf{m}^* \quad \text{where} \quad \mathbf{H}_0 = \mathbf{R} + t \mathbf{n}^{*\top}$$

- ◆ **Demonstration:**
- ◆ **3D transformation:** $\mathcal{X} = \mathbf{R} \mathcal{X}^* + \mathbf{t}$.
- ◆ **The plane equation:** $\mathbf{n}^{*\top} \mathcal{X}^* = 1$.
- ◆ **Perspective projection:** $\mathbf{m}^* = Z^{*-1} \mathcal{X}^*$ and $\mathbf{m} = Z^{-1} \mathcal{X}$.
- ◆ $\implies Z \mathbf{m} = Z^* \mathbf{R} \mathbf{m}^* + \mathbf{t}$ and $Z^* \mathbf{n}^{*\top} \mathbf{m}^* = 1$.

Projective transformation

- Using the previous equations, we can establish a relationship between two corresponding points \mathbf{m} and \mathbf{m}^* :

$$\frac{Z}{Z^*} \mathbf{m} = \mathbf{H}_0 \mathbf{m}^* \quad \text{where} \quad \mathbf{H}_0 = \mathbf{R} + \mathbf{t} \mathbf{n}^{*\top}$$

- ◆ **Demonstration:**
- ◆ **3D transformation:** $\mathcal{X} = \mathbf{R} \mathcal{X}^* + \mathbf{t}$.
- ◆ **The plane equation:** $\mathbf{n}^{*\top} \mathcal{X}^* = 1$.
- ◆ **Perspective projection:** $\mathbf{m}^* = Z^{*-1} \mathcal{X}^*$ and $\mathbf{m} = Z^{-1} \mathcal{X}$.
- ◆ $\implies Z \mathbf{m} = Z^* \mathbf{R} \mathbf{m}^* + \mathbf{t}$ and $Z^* \mathbf{n}^{*\top} \mathbf{m}^* = 1$.
- ◆ $Z \mathbf{m} = Z^* \mathbf{R} \mathbf{m}^* + Z^* \mathbf{t} \mathbf{n}^{*\top} \mathbf{m}^*$.

Projective transformation

- Using the previous equations, we can establish a relationship between two corresponding points \mathbf{m} and \mathbf{m}^* :

$$\frac{Z}{Z^*} \mathbf{m} = \mathbf{H}_0 \mathbf{m}^* \quad \text{where} \quad \mathbf{H}_0 = \mathbf{R} + \mathbf{t} \mathbf{n}^{*\top}$$

- ◆ **Demonstration:**
- ◆ **3D transformation:** $\mathcal{X} = \mathbf{R} \mathcal{X}^* + \mathbf{t}$.
- ◆ **The plane equation:** $\mathbf{n}^{*\top} \mathcal{X}^* = 1$.
- ◆ **Perspective projection:** $\mathbf{m}^* = Z^{*-1} \mathcal{X}^*$ and $\mathbf{m} = Z^{-1} \mathcal{X}$.
- ◆ $\implies Z \mathbf{m} = Z^* \mathbf{R} \mathbf{m}^* + \mathbf{t}$ and $Z^* \mathbf{n}^{*\top} \mathbf{m}^* = 1$.
- ◆ $Z \mathbf{m} = Z^* \mathbf{R} \mathbf{m}^* + Z^* \mathbf{t} \mathbf{n}^{*\top} \mathbf{m}^*$.
- ◆ $Z \mathbf{m} = Z^* (\mathbf{R} + \mathbf{t} \mathbf{n}^{*\top}) \mathbf{m}^*$.

Projective transformation

- Using the previous equations, we can establish a relationship between two corresponding points \mathbf{m} and \mathbf{m}^* :

$$\frac{Z}{Z^*} \mathbf{m} = \mathbf{H}_0 \mathbf{m}^* \quad \text{where} \quad \mathbf{H}_0 = \mathbf{R} + \mathbf{t} \mathbf{n}^{*\top}$$

- ◆ **Demonstration:**
- ◆ **3D transformation:** $\mathcal{X} = \mathbf{R} \mathcal{X}^* + \mathbf{t}$.
- ◆ **The plane equation:** $\mathbf{n}^{*\top} \mathcal{X}^* = 1$.
- ◆ **Perspective projection:** $\mathbf{m}^* = Z^{*-1} \mathcal{X}^*$ and $\mathbf{m} = Z^{-1} \mathcal{X}$.
- ◆ $\implies Z \mathbf{m} = Z^* \mathbf{R} \mathbf{m}^* + \mathbf{t}$ and $Z^* \mathbf{n}^{*\top} \mathbf{m}^* = 1$.
- ◆ $Z \mathbf{m} = Z^* \mathbf{R} \mathbf{m}^* + Z^* \mathbf{t} \mathbf{n}^{*\top} \mathbf{m}^*$.
- ◆ $Z \mathbf{m} = Z^* (\mathbf{R} + \mathbf{t} \mathbf{n}^{*\top}) \mathbf{m}^*$.
- ◆ $\frac{Z}{Z^*} \mathbf{m} = \mathbf{H}_0 \mathbf{m}^* \quad \text{where} \quad \mathbf{H}_0 = \mathbf{R} + \mathbf{t} \mathbf{n}^{*\top}$.

Projective transformation

- Using the previous equations, we can establish a relationship between two corresponding points \mathbf{m} and \mathbf{m}^* :

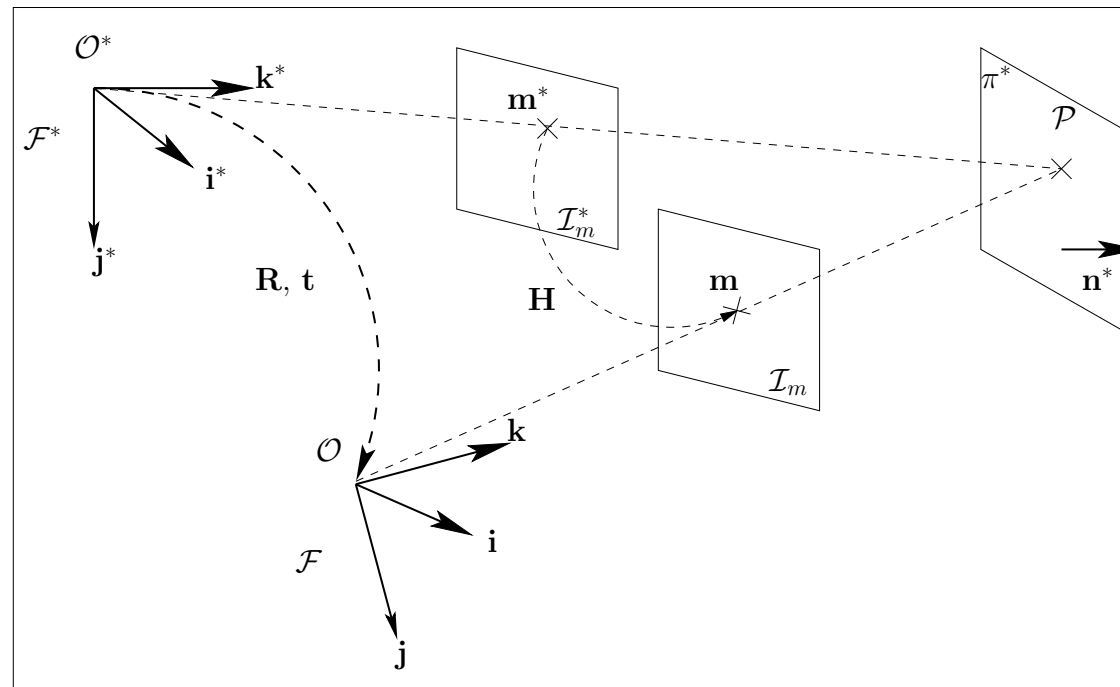
$$\frac{Z}{Z^*} \mathbf{m} = \mathbf{H}_0 \mathbf{m}^* \quad \text{where} \quad \mathbf{H}_0 = \mathbf{R} + \mathbf{t} \mathbf{n}^{*\top}$$

- Knowing \mathbf{m}^* and a matrix \mathbf{H} proportional to \mathbf{H}_0 : $\mathbf{H} = \alpha \mathbf{H}_0$ ($\alpha \in \mathbb{R}^*$), it is possible to compute \mathbf{m} without knowing the ratio $\frac{Z}{Z^*}$.
- It suffices to multiply \mathbf{m}^* by \mathbf{H} and then to normalize the third entry of \mathbf{m} to 1. If, we write the matrix \mathbf{H} as: $\mathbf{H} = \begin{bmatrix} \mathbf{h}_1 & \mathbf{h}_2 & \mathbf{h}_3 \end{bmatrix}^\top$, the normalization can be written:

$$\mathbf{m} = \begin{bmatrix} \frac{\mathbf{h}_1^\top \mathbf{m}^*}{\mathbf{h}_3^\top \mathbf{m}^*} & \frac{\mathbf{h}_2^\top \mathbf{m}^*}{\mathbf{h}_3^\top \mathbf{m}^*} & 1 \end{bmatrix}^\top \propto \mathbf{H} \mathbf{m}^*$$

Projective transformation

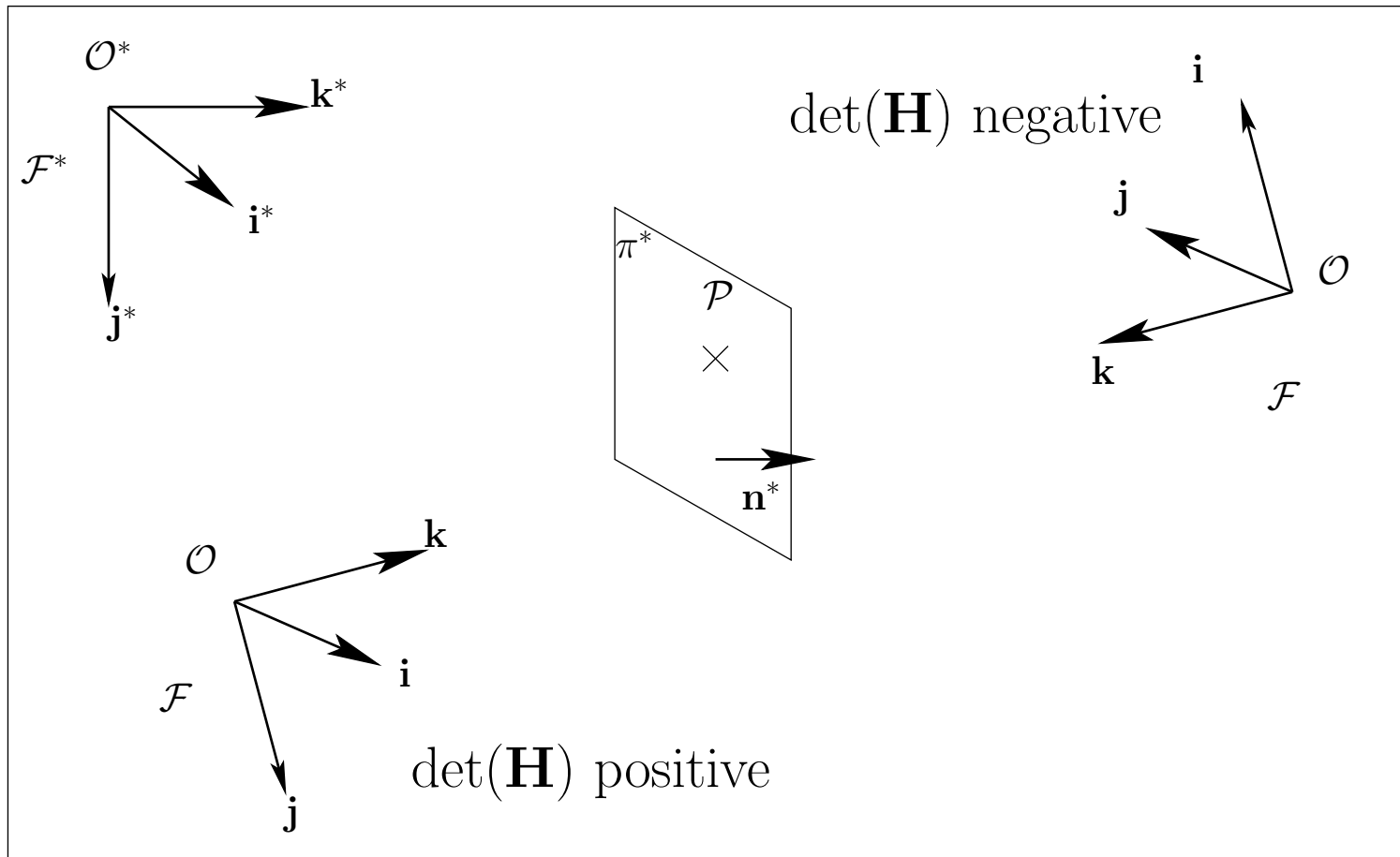
- The matrix \mathbf{H} is called “homography”. It matches the points of the plane π between the two metric images \mathcal{I}_m^* and \mathcal{I}_m .



Homography parameterization

- Given N non-collinear corresponding points $(\mathbf{m}_i^*, \mathbf{m}_i)$ where $i \in \{1, 2, \dots, N\}$ and $N \geq 4$, it is possible to compute the homography matrix \mathbf{H} .
- \mathbf{H} is defined up to a scale factor. It has 9 entries (a (3×3) matrix) and only 8 d.o.f.: 3 for the rotation, 3 for the translation and 2 for the normal vector to the plane.
- The scale factor can be chosen with different manners.
- We choose the scale factor of \mathbf{H} such that its determinant is equal to 1. Then, we have $\mathbf{H} \in \mathbb{SL}(3)$ the Special Linear group.
- This choice is well justified since $\det(\mathbf{H}) \leq 0$ happens when the projection center \mathcal{O} crosses the plane π , and we suppose that \mathcal{O}^* and \mathcal{O} are always in the same side of the plane π .

Homography parameterization



The camera model

- In the “pinhole model”, the camera performs a perspective projection.
- In the image \mathcal{I}^* , the 3D point \mathcal{P} is projected in $\mathbf{p}^* \in \mathbb{P}^2$:
 $\mathbf{p}^* = [u^* \ v^* \ 1]^\top = \mathbf{K}\mathbf{m}^*$ and in the image \mathcal{I} , it is projected in $\mathbf{p} \in \mathbb{P}^2$:
 $\mathbf{p} = [u \ v \ 1]^\top = \mathbf{K}\mathbf{m}$.

The camera model

- In the “pinhole model”, the camera performs a perspective projection.
- In the image \mathcal{I}^* , the 3D point \mathcal{P} is projected in $\mathbf{p}^* \in \mathbb{P}^2$:
 $\mathbf{p}^* = [u^* \ v^* \ 1]^\top = \mathbf{K}\mathbf{m}^*$ and in the image \mathcal{I} , it is projected in $\mathbf{p} \in \mathbb{P}^2$:
 $\mathbf{p} = [u \ v \ 1]^\top = \mathbf{K}\mathbf{m}$.
- The matrix \mathbf{K} is a (3×3) upper triangular matrix that depends on the camera intrinsics parameters:

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

where f is the focal in pixels, s is the skew (describing the orthogonality of the sensor axes), r describes the ratio of the pixels dimensions, $[u_0 \ v_0 \ 1]$ are the coordinates of the image of the principal point $\mathbf{m}_0 = [0 \ 0 \ 1]^\top$ in pixels.

The camera model

- In the “pinhole model”, the camera performs a perspective projection.
- In the image \mathcal{I}^* , the 3D point \mathcal{P} is projected in $\mathbf{p}^* \in \mathbb{P}^2$:
 $\mathbf{p}^* = [u^* \ v^* \ 1]^\top = \mathbf{K}\mathbf{m}^*$ and in the image \mathcal{I} , it is projected in $\mathbf{p} \in \mathbb{P}^2$:
 $\mathbf{p} = [u \ v \ 1]^\top = \mathbf{K}\mathbf{m}$.

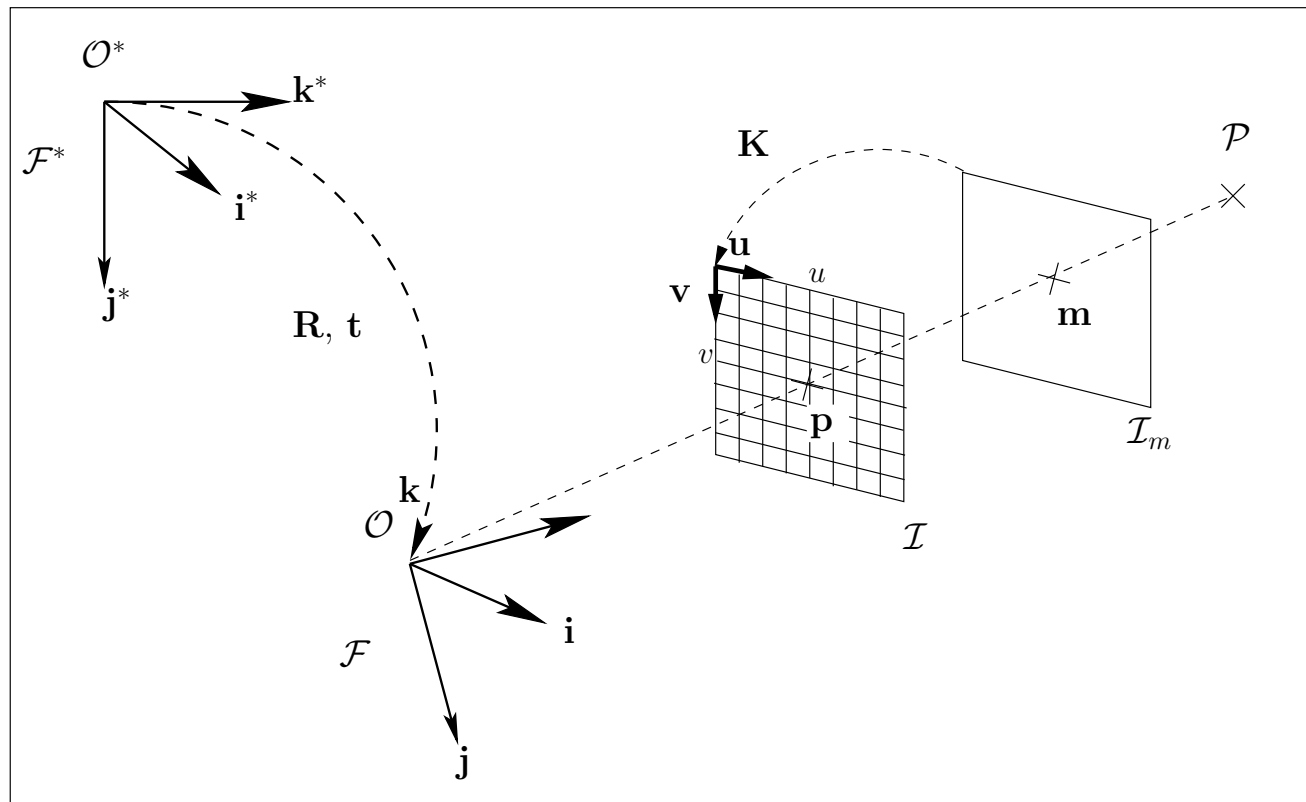


Image projective transformation

- The two corresponding image points p and p^* verifies: $p \propto Gp^*$. where the image homography matrix G verifies:

$$G = KHK^{-1}$$

- ◆ **Demonstration:**

Image projective transformation

- The two corresponding image points \mathbf{p} and \mathbf{p}^* verifies: $\mathbf{p} \propto \mathbf{G}\mathbf{p}^*$. where the image homography matrix \mathbf{G} verifies:

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

- ◆ **Demonstration:**
- ◆ **Camera model projection: $\mathbf{p}^* = \mathbf{K}\mathbf{m}^*$ and $\mathbf{p} = \mathbf{K}\mathbf{m}$.**

Image projective transformation

- The two corresponding image points \mathbf{p} and \mathbf{p}^* verifies: $\mathbf{p} \propto \mathbf{G}\mathbf{p}^*$. where the image homography matrix \mathbf{G} verifies:

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

- ◆ **Demonstration:**
- ◆ **Camera model projection: $\mathbf{p}^* = \mathbf{K}\mathbf{m}^*$ and $\mathbf{p} = \mathbf{K}\mathbf{m}$.**
- ◆ **Metric image homography: $\mathbf{m} \propto \mathbf{H}\mathbf{m}^*$.**

Image projective transformation

- The two corresponding image points \mathbf{p} and \mathbf{p}^* verifies: $\mathbf{p} \propto \mathbf{G}\mathbf{p}^*$. where the image homography matrix \mathbf{G} verifies:

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

- ◆ **Demonstration:**
- ◆ **Camera model projection: $\mathbf{p}^* = \mathbf{K}\mathbf{m}^*$ and $\mathbf{p} = \mathbf{K}\mathbf{m}$.**
- ◆ **Metric image homography: $\mathbf{m} \propto \mathbf{H}\mathbf{m}^*$.**
- ◆ **$\implies \mathbf{K}^{-1}\mathbf{p} \propto \mathbf{H}\mathbf{K}^{-1}\mathbf{p}^*$.**

Image projective transformation

- The two corresponding image points \mathbf{p} and \mathbf{p}^* verifies: $\mathbf{p} \propto \mathbf{G}\mathbf{p}^*$. where the image homography matrix \mathbf{G} verifies:

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

- ◆ Demonstration:
- ◆ Camera model projection: $\mathbf{p}^* = \mathbf{K}\mathbf{m}^*$ and $\mathbf{p} = \mathbf{K}\mathbf{m}$.
- ◆ Metric image homography: $\mathbf{m} \propto \mathbf{H}\mathbf{m}^*$.
- ◆ $\implies \mathbf{K}^{-1}\mathbf{p} \propto \mathbf{H}\mathbf{K}^{-1}\mathbf{p}^*$.
- ◆ $\mathbf{p} \propto \mathbf{K}\mathbf{H}\mathbf{K}^{-1}\mathbf{p}^*$.

Image projective transformation

- The two corresponding image points p and p^* verifies: $p \propto Gp^*$. where the image homography matrix G verifies: $G = KHK^{-1}$.

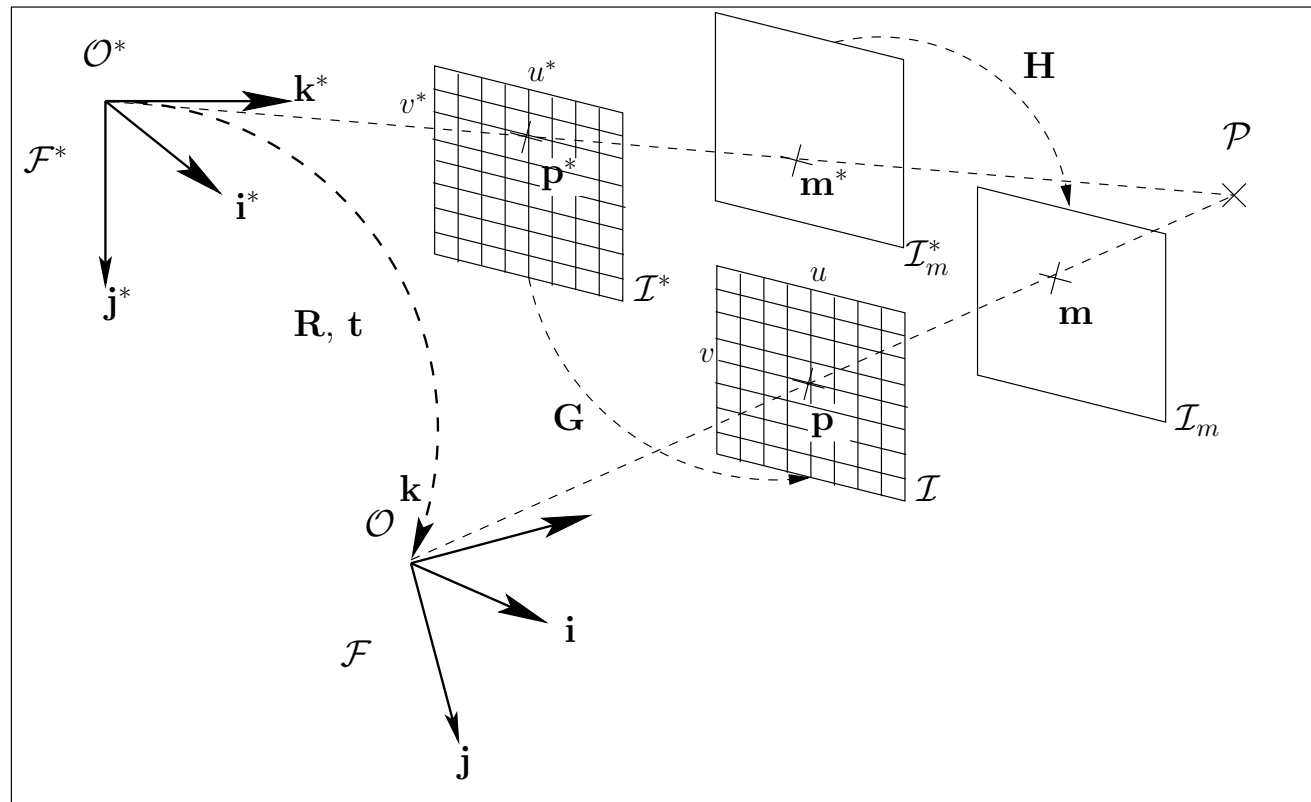


Image projective transformation

- The two corresponding image points \mathbf{p} and \mathbf{p}^* verifies: $\mathbf{p} \propto \mathbf{G}\mathbf{p}^*$. where the image homography matrix \mathbf{G} verifies: $\mathbf{G} = \mathbf{K}\mathbf{H}\mathbf{K}^{-1}$
- Given N non-collinear corresponding points $(\mathbf{p}_i^*, \mathbf{p}_i)$ where $i \in \{1, 2, \dots, N\}$ and $N \geq 4$, it is possible to compute the image homography matrix \mathbf{G} up to a scale factor.
- It is clear that $\mathbf{G} \in \text{SL}(3)$ iff $\mathbf{H} \in \text{SL}(3)$.
 - ◆ **Demonstration:**

Image projective transformation

- The two corresponding image points \mathbf{p} and \mathbf{p}^* verifies: $\mathbf{p} \propto \mathbf{G}\mathbf{p}^*$. where the image homography matrix \mathbf{G} verifies: $\mathbf{G} = \mathbf{K}\mathbf{H}\mathbf{K}^{-1}$
- Given N non-collinear corresponding points $(\mathbf{p}_i^*, \mathbf{p}_i)$ where $i \in \{1, 2, \dots, N\}$ and $N \geq 4$, it is possible to compute the image homography matrix \mathbf{G} up to a scale factor.
- It is clear that $\mathbf{G} \in \text{SL}(3)$ iff $\mathbf{H} \in \text{SL}(3)$.
 - ◆ **Demonstration:**
 - ◆ $\det(\mathbf{G}) = \det(\mathbf{K}\mathbf{H}\mathbf{K}^{-1})$

Image projective transformation

- The two corresponding image points \mathbf{p} and \mathbf{p}^* verifies: $\mathbf{p} \propto \mathbf{G}\mathbf{p}^*$. where the image homography matrix \mathbf{G} verifies: $\mathbf{G} = \mathbf{K}\mathbf{H}\mathbf{K}^{-1}$
- Given N non-collinear corresponding points $(\mathbf{p}_i^*, \mathbf{p}_i)$ where $i \in \{1, 2, \dots, N\}$ and $N \geq 4$, it is possible to compute the image homography matrix \mathbf{G} up to a scale factor.
- It is clear that $\mathbf{G} \in \text{SL}(3)$ iff $\mathbf{H} \in \text{SL}(3)$.
 - ◆ **Demonstration:**
 - ◆ $\det(\mathbf{G}) = \det(\mathbf{K}\mathbf{H}\mathbf{K}^{-1})$
 - ◆ $\det(\mathbf{G}) = \det(\mathbf{K})\det(\mathbf{H})\det(\mathbf{K}^{-1})$

Image projective transformation

- The two corresponding image points \mathbf{p} and \mathbf{p}^* verifies: $\mathbf{p} \propto \mathbf{G}\mathbf{p}^*$. where the image homography matrix \mathbf{G} verifies: $\mathbf{G} = \mathbf{K}\mathbf{H}\mathbf{K}^{-1}$
- Given N non-collinear corresponding points $(\mathbf{p}_i^*, \mathbf{p}_i)$ where $i \in \{1, 2, \dots, N\}$ and $N \geq 4$, it is possible to compute the image homography matrix \mathbf{G} up to a scale factor.
- It is clear that $\mathbf{G} \in \text{SL}(3)$ iff $\mathbf{H} \in \text{SL}(3)$.
 - ◆ **Demonstration:**
 - ◆ $\det(\mathbf{G}) = \det(\mathbf{K}\mathbf{H}\mathbf{K}^{-1})$
 - ◆ $\det(\mathbf{G}) = \det(\mathbf{K})\det(\mathbf{H})\det(\mathbf{K}^{-1})$
 - ◆ $\det(\mathbf{G}) = \det(\mathbf{K})\det(\mathbf{H})\frac{1}{\det(\mathbf{K})}$

Image projective transformation

- The two corresponding image points \mathbf{p} and \mathbf{p}^* verifies: $\mathbf{p} \propto \mathbf{G}\mathbf{p}^*$. where the image homography matrix \mathbf{G} verifies: $\mathbf{G} = \mathbf{K}\mathbf{H}\mathbf{K}^{-1}$
- Given N non-collinear corresponding points $(\mathbf{p}_i^*, \mathbf{p}_i)$ where $i \in \{1, 2, \dots, N\}$ and $N \geq 4$, it is possible to compute the image homography matrix \mathbf{G} up to a scale factor.
- It is clear that $\mathbf{G} \in \text{SL}(3)$ iff $\mathbf{H} \in \text{SL}(3)$.
 - ◆ **Demonstration:**
 - ◆ $\det(\mathbf{G}) = \det(\mathbf{K}\mathbf{H}\mathbf{K}^{-1})$
 - ◆ $\det(\mathbf{G}) = \det(\mathbf{K})\det(\mathbf{H})\det(\mathbf{K}^{-1})$
 - ◆ $\det(\mathbf{G}) = \det(\mathbf{K})\det(\mathbf{H})\frac{1}{\det(\mathbf{K})}$
 - ◆ $\det(\mathbf{G}) = \det(\mathbf{H})$

Image projective transformation

- The two corresponding image points \mathbf{p} and \mathbf{p}^* verifies: $\mathbf{p} \propto \mathbf{G}\mathbf{p}^*$. where the image homography matrix \mathbf{G} verifies: $\mathbf{G} = \mathbf{K}\mathbf{H}\mathbf{K}^{-1}$
- Given N non-collinear corresponding points $(\mathbf{p}_i^*, \mathbf{p}_i)$ where $i \in \{1, 2, \dots, N\}$ and $N \geq 4$, it is possible to compute the image homography matrix \mathbf{G} up to a scale factor.
- It is clear that $\mathbf{G} \in \text{SL}(3)$ iff $\mathbf{H} \in \text{SL}(3)$.
- The matrix \mathbf{G} has 8 d.o.f. and can be described by a (8×1) vector parameter \mathbf{x} .
- We define a group action from $\text{SL}(3)$ on \mathbb{P}^2 :

$$\mathbf{w} : \text{SL}(3) \times \mathbb{P}^2 \rightarrow \mathbb{P}^2$$

- We define the warping $\mathbf{w}(\mathbf{G})$ (a \mathbb{P}^2 automorphism):

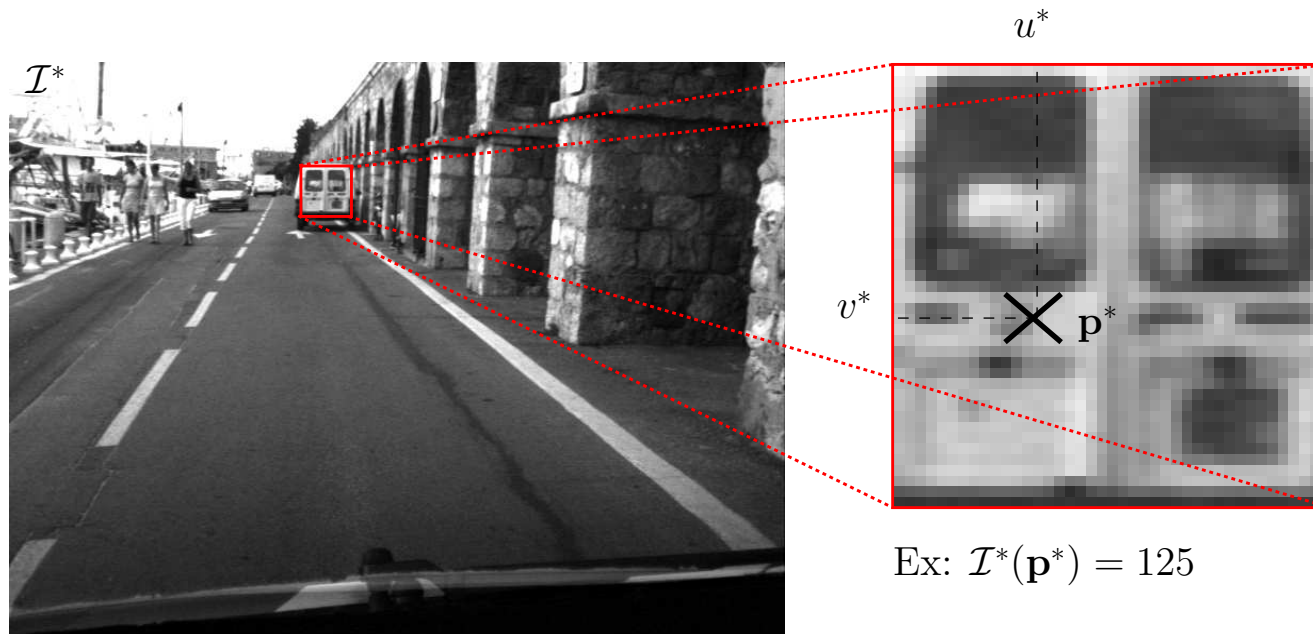
$$\mathbf{w}(\mathbf{G}) : \mathbf{p}^* \mapsto \mathbf{p} = \mathbf{w}(\mathbf{G})(\mathbf{p}^*) \propto \mathbf{G}\mathbf{p}^*$$

PART II : Visual tracking

- The image model
- Template-based tracking
- System linearization
- The ESM algorithm
- The ESM features

The image model

- A $(n \times m)$ image \mathcal{I}^* can be considered as $(n \times m)$ matrix containing the pixels intensities.



The image model

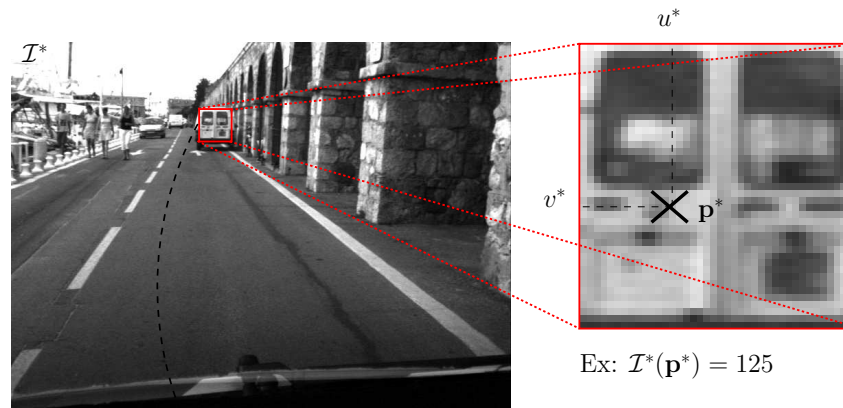
- A $(n \times m)$ image \mathcal{I}^* can be considered as $(n \times m)$ matrix containing the pixels intensities.
- We suppose that it exists a regular function I^* :

$$I^* : \mathbb{P}^2 \rightarrow \mathbb{R}$$
$$\mathbf{p}^* = [u^* \ v^* \ 1]^\top \mapsto I^*(u^*, v^*)$$

where $\forall \mathbf{p}^* = [u^* \ v^* \ 1]^\top$, $(u^*, v^*) \in \{1, \dots, n\} \times \{1, \dots, m\}$, we have:
 $I^*(\mathbf{p}^*) = \mathcal{I}^*(u^*, v^*)$.

The image model

- A $(n \times m)$ image \mathcal{I}^* can be considered as $(n \times m)$ matrix containing the pixels intensities.
- We use the “image constancy assumption”: $\mathcal{I}^*(\mathbf{p}^*) = \mathcal{I}(\mathbf{w}(\mathbf{G})(\mathbf{p}^*))$.



Template-based tracking

- To track a template in the image \mathcal{I} is to find the (8×1) vector \bar{x} containing the parameters of the projective transformation $\bar{\mathbf{G}} \in \text{SL}(3)$ i.e. to find $\bar{\mathbf{G}}$ such that $\forall i \in \{1, 2, \dots, q\}$, we have:

$$\mathcal{I}(\mathbf{w}(\bar{\mathbf{G}})(\mathbf{p}_i^*)) = \mathcal{I}^*(\mathbf{p}_i^*)$$

Template-based tracking

- To track a template in the image \mathcal{I} is to find the (8×1) vector $\bar{\mathbf{x}}$ containing the parameters of the projective transformation $\bar{\mathbf{G}} \in \mathbb{S}\mathbb{L}(3)$ i.e. to find $\bar{\mathbf{G}}$ such that $\forall i \in \{1, 2, \dots, q\}$, we have:

$$\mathcal{I} \left(\mathbf{w} \left(\bar{\mathbf{G}} \right) \left(\mathbf{p}_i^* \right) \right) = \mathcal{I}^* \left(\mathbf{p}_i^* \right)$$

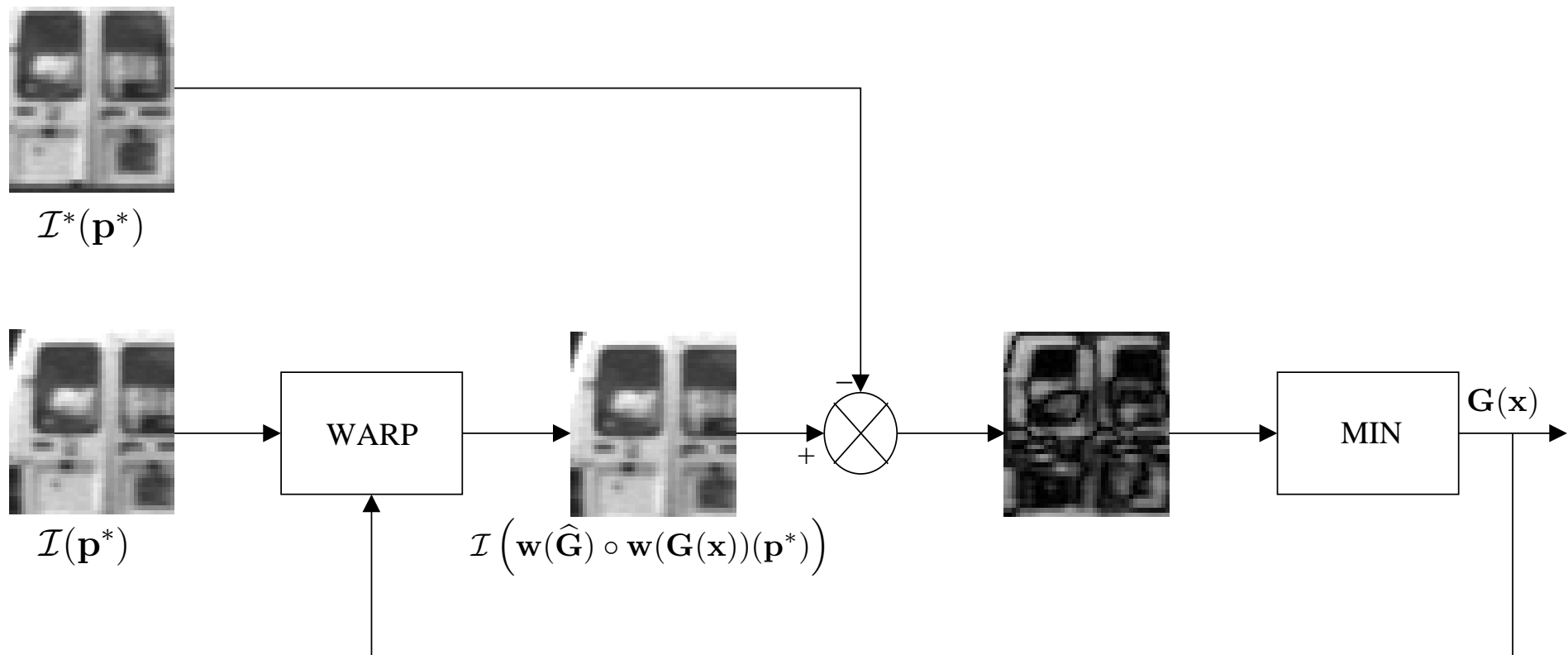
- Suppose that we have an approximation $\hat{\mathbf{G}}$ of $\bar{\mathbf{G}}$, the problem consists in finding the incremental homography $\mathbf{G}(\mathbf{x})$ such that the difference between the region of the image \mathcal{I} warped with the composition $\mathbf{w}(\hat{\mathbf{G}}) \circ \mathbf{w}(\mathbf{G}(\mathbf{x}))$ and the corresponding region in the image \mathcal{I}^* is equal to zero i.e finding \mathbf{x} such that $\forall i \in \{1, 2, \dots, q\}$, we have:

$$y_i(\mathbf{x}) = \mathcal{I} \left(\mathbf{w}(\hat{\mathbf{G}}) \circ \mathbf{w}(\mathbf{G}(\mathbf{x})) \left(\mathbf{p}_i^* \right) \right) - \mathcal{I}^* \left(\mathbf{p}_i^* \right) = 0$$

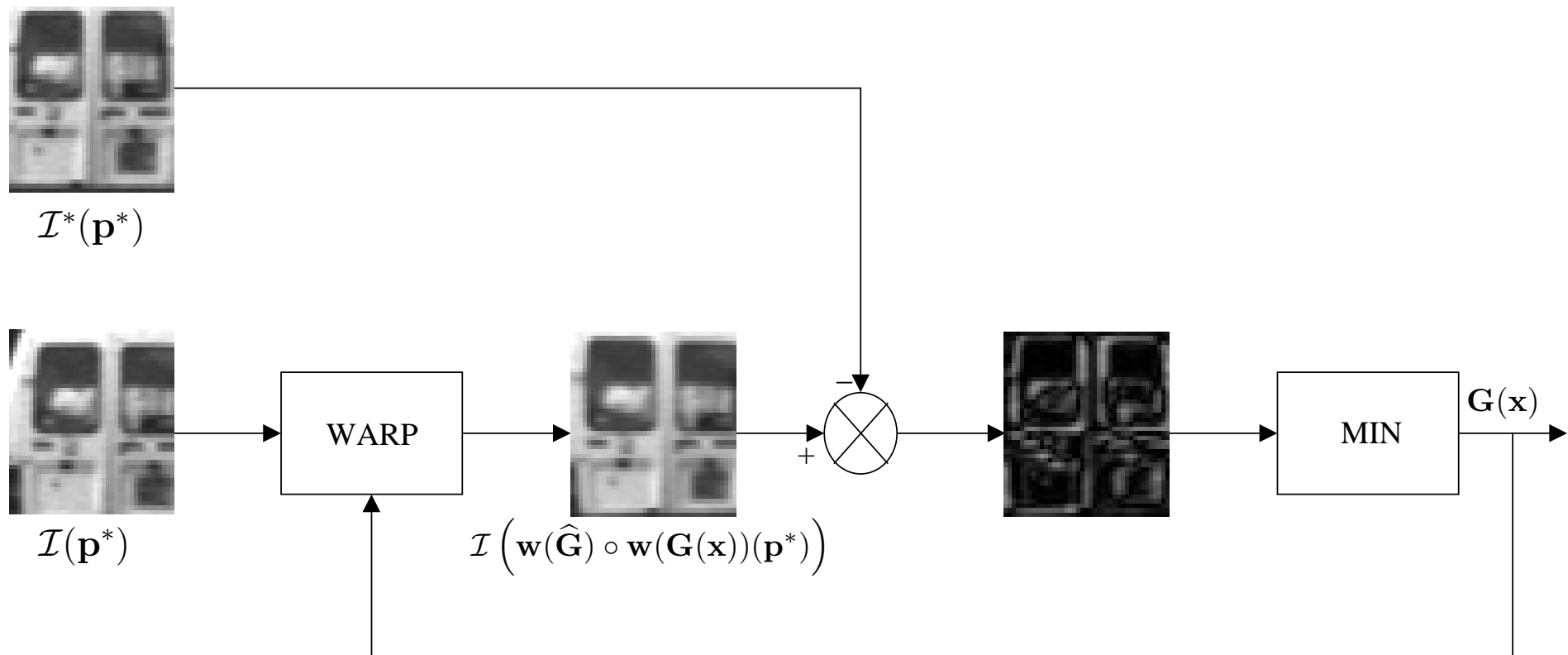
Template-based tracking

- Let $\mathbf{y}(\mathbf{x})$ be the $(q \times 1)$ vector containing image differences:
 $\mathbf{y}(\mathbf{x}) = [y_1(\mathbf{x}) \ y_2(\mathbf{x}) \ \dots \ y_q(\mathbf{x})]^\top$, the problem consists in finding $\mathbf{x} = \mathbf{x}_0$ such that: $\mathbf{y}(\mathbf{x}_0) = \mathbf{0}$.
- The solution to the non linear system verifies $\mathbf{G}(\mathbf{x}_0) = \hat{\mathbf{G}}^{-1} \overline{\mathbf{G}}$
- Many minimization methods could be used.
- The real-time constraint (less than 40 ms for 25 images/s acquisition).
- Generally, the problem is solved iteratively, after linearizing the system.

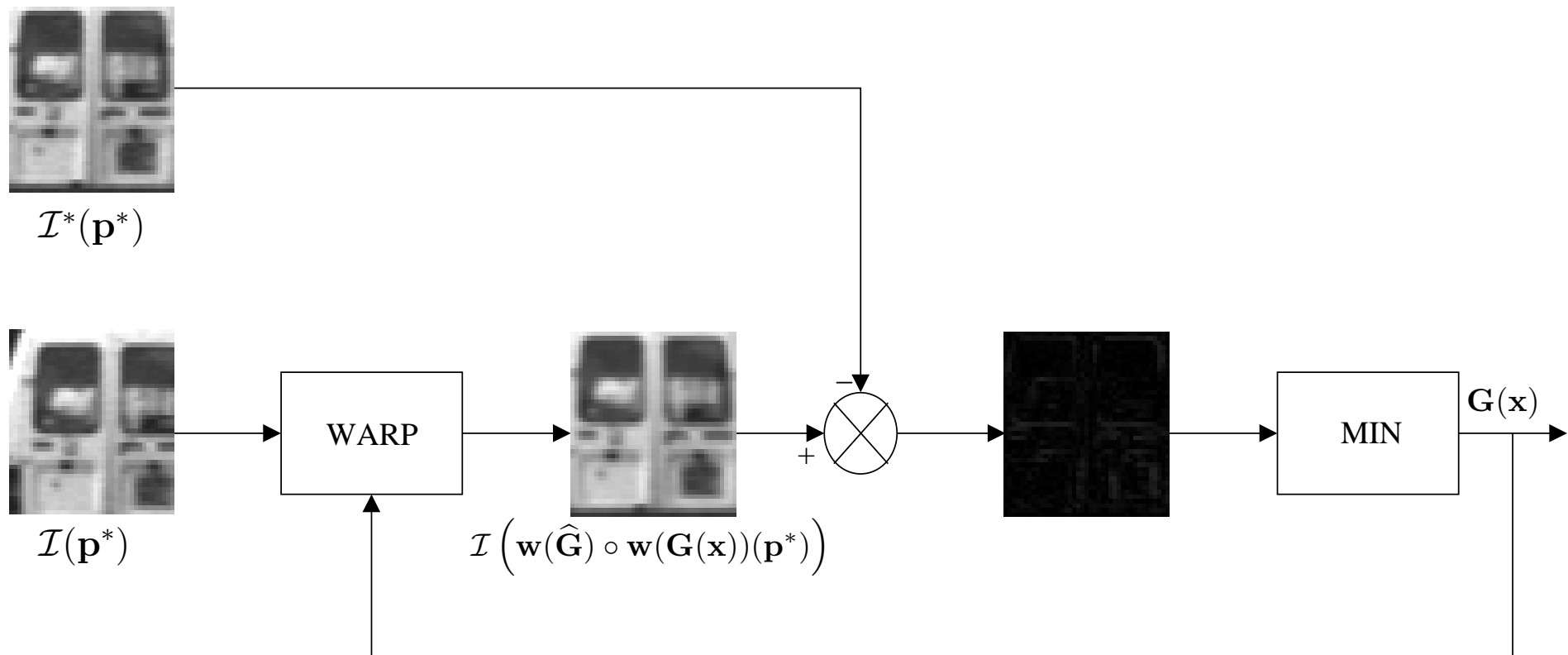
Template-based tracking



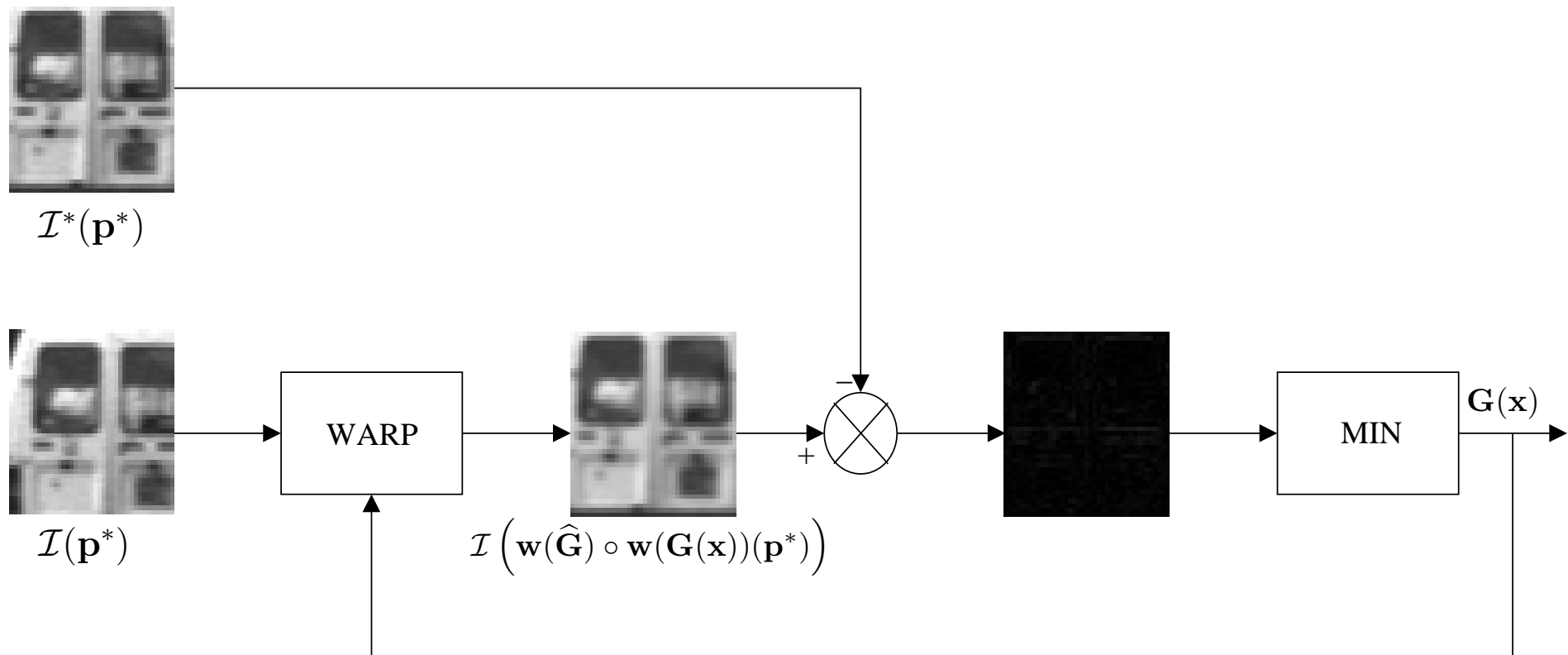
Template-based tracking



Template-based tracking



Template-based tracking



Linearizing the system (1)

- Let the $(q \times 8)$ matrix $\mathbf{J}(\mathbf{x})$ be the Jacobian matrix defined as:

$$\mathbf{J}(\mathbf{x}) = \nabla_{\mathbf{x}} \mathbf{y}(\mathbf{x}) = \begin{bmatrix} \frac{\partial y_1(\mathbf{x})}{\partial x_1} & \frac{\partial y_1(\mathbf{x})}{\partial x_2} & \cdots & \frac{\partial y_1(\mathbf{x})}{\partial x_8} \\ \frac{\partial y_2(\mathbf{x})}{\partial x_1} & \frac{\partial y_2(\mathbf{x})}{\partial x_2} & \cdots & \frac{\partial y_2(\mathbf{x})}{\partial x_8} \\ \cdots & \cdots & \cdots & \cdots \\ \frac{\partial y_q(\mathbf{x})}{\partial x_1} & \frac{\partial y_q(\mathbf{x})}{\partial x_2} & \cdots & \frac{\partial y_q(\mathbf{x})}{\partial x_8} \end{bmatrix}$$

- Let the $(q \times 8)$ matrix $\mathbf{M}(\mathbf{x}_1, \mathbf{x}_2)$ defined as $\forall (\mathbf{x}_1, \mathbf{x}_2) \in \mathbb{R}^8 \times \mathbb{R}^8$:

$$\mathbf{M}(\mathbf{x}_1, \mathbf{x}_2) = \nabla_{\mathbf{x}_1} (\mathbf{J}(\mathbf{x}_1) \mathbf{x}_2) = \begin{bmatrix} \frac{\partial^2 y_1(\mathbf{x}_1)}{\partial \mathbf{x}_1^2} \mathbf{x}_2 & \cdots & \frac{\partial^2 y_q(\mathbf{x}_1)}{\partial \mathbf{x}_1^2} \mathbf{x}_2 \end{bmatrix}^\top$$

- Each (8×8) Hessian matrix can be written as :

$$\frac{\partial^2 y_i(\mathbf{x})}{\partial \mathbf{x}^2} = \begin{bmatrix} \frac{\partial^2 y_i(\mathbf{x})}{\partial x_1 \partial x_1} & \cdots & \frac{\partial^2 y_i(\mathbf{x})}{\partial x_1 \partial x_8} \\ \cdots & \cdots & \cdots \\ \frac{\partial^2 y_i(\mathbf{x})}{\partial x_1 \partial x_8} & \cdots & \frac{\partial^2 y_i(\mathbf{x})}{\partial x_8 \partial x_8} \end{bmatrix}$$

Linearizing the system (2)

- To linearize the vector $\mathbf{y}(\mathbf{x})$, we use a 2nd order Taylor series approximation about $\mathbf{x} = \mathbf{0}$:

$$\mathbf{y}(\mathbf{x}) = \mathbf{y}(\mathbf{0}) + \mathbf{J}(\mathbf{0}) \mathbf{x} + \frac{1}{2} \mathbf{M}(\mathbf{0}, \mathbf{x}) \mathbf{x} + \mathbf{O}(\|\mathbf{x}\|^3)$$

- For $\mathbf{x} = \mathbf{x}_0$, the system can be written as follows:

$$\mathbf{y}(\mathbf{x}_0) \approx \mathbf{y}(\mathbf{0}) + \left(\mathbf{J}(\mathbf{0}) + \frac{1}{2} \mathbf{M}(\mathbf{0}, \mathbf{x}_0) \right) \mathbf{x}_0 = \mathbf{0}$$

- We define the SSD cost function:

$$f(\mathbf{x}) = \frac{1}{2} \|\mathbf{y}(\mathbf{0}) + \mathbf{J}(\mathbf{0})\mathbf{x} + \frac{1}{2} \mathbf{M}(\mathbf{0}, \mathbf{x})\mathbf{x}\|^2$$

Iterative minimization

- A necessary condition to have a local or a global minimum at $\mathbf{x} = \mathbf{x}_0$ is that:

$$\nabla_{\mathbf{x}} f(\mathbf{x})|_{\mathbf{x}=\mathbf{x}_0} = \mathbf{0}$$

- $\nabla_{\mathbf{x}} f(\mathbf{x}) = (\mathbf{J}(\mathbf{0})^\top + \mathbf{M}(\mathbf{0}, \mathbf{x}) + \mathbf{O}(\|\mathbf{x}\|^2))^\top (\mathbf{y}(\mathbf{0}) + \mathbf{J}(\mathbf{0})\mathbf{x} + \mathbf{O}(\|\mathbf{x}\|^2)).$
- Newton minimization solves the problem iteratively.
- At each iteration, an incremental \mathbf{x}_0 is estimated:

$$\mathbf{x}_0 = -\mathbf{S}^{-1} \mathbf{J}(\mathbf{0})^\top \mathbf{y}(\mathbf{0})$$

where the (8×8) matrix \mathbf{S} :

$$\mathbf{S} = \mathbf{J}(\mathbf{0})^\top \mathbf{J}(\mathbf{0}) + \sum_{i=0}^q \frac{\partial^2 y_i(\mathbf{x})}{\partial \mathbf{x}^2} \Big|_{\mathbf{x}=\mathbf{0}} y_i(\mathbf{0})$$

- Once \mathbf{x}_0 is estimated, the approximation $\hat{\mathbf{G}}$ is updated:

$$\hat{\mathbf{G}} \leftarrow \hat{\mathbf{G}} \mathbf{G}(\mathbf{x}_0)$$

Iterative minimization

- Different first order approximations:
 - ◆ Gradient descent: $\mathbf{S} \approx \hat{\mathbf{S}} = \alpha \mathbf{I}$ where $\alpha > 0$
 - ◆ Varying Gauss-Newton: $\mathbf{S} \approx \hat{\mathbf{S}} = \mathbf{J}(\mathbf{0})^\top \mathbf{J}(\mathbf{0})$
 - ◆ Constant Gauss-Newton: $\mathbf{S} \approx \hat{\mathbf{S}} = \hat{\mathbf{J}}(\mathbf{0})^\top \hat{\mathbf{J}}(\mathbf{0})$
 - ◆ Levenberg-Marquardt: $\mathbf{S} \approx \hat{\mathbf{S}} = \mathbf{J}(\mathbf{0})^\top \mathbf{J}(\mathbf{0}) + \alpha \mathbf{I}$ where $\alpha > 0$

Is there a second order approximation without the Hessians computation ?

The ESM algorithm

- A second order approximation without Hessians computation:
$$\mathbf{y}(\mathbf{x}) = \mathbf{y}(\mathbf{0}) + \frac{1}{2} (\mathbf{J}(\mathbf{0}) + \mathbf{J}(\mathbf{x})) \mathbf{x} + \mathbf{O}(\|\mathbf{x}\|^3).$$
 - ◆ **Demonstration:**

The ESM algorithm

- A second order approximation without Hessians computation:
$$\mathbf{y}(\mathbf{x}) = \mathbf{y}(\mathbf{0}) + \frac{1}{2} (\mathbf{J}(\mathbf{0}) + \mathbf{J}(\mathbf{x})) \mathbf{x} + \mathbf{O}(\|\mathbf{x}\|^3).$$
 - ◆ **Demonstration:**
 - ◆
$$\mathbf{y}(\mathbf{x}) = \mathbf{y}(\mathbf{0}) + \mathbf{J}(\mathbf{0})\mathbf{x} + \frac{1}{2}\mathbf{M}(\mathbf{0}, \mathbf{x})\mathbf{x} + \mathbf{O}(\|\mathbf{x}\|^3).$$

The ESM algorithm

- A second order approximation without Hessians computation:

$$\mathbf{y}(\mathbf{x}) = \mathbf{y}(\mathbf{0}) + \frac{1}{2} (\mathbf{J}(\mathbf{0}) + \mathbf{J}(\mathbf{x})) \mathbf{x} + \mathbf{O}(\|\mathbf{x}\|^3).$$

- ◆ **Demonstration:**

- ◆ $\mathbf{y}(\mathbf{x}) = \mathbf{y}(\mathbf{0}) + \mathbf{J}(\mathbf{0})\mathbf{x} + \frac{1}{2}\mathbf{M}(\mathbf{0}, \mathbf{x})\mathbf{x} + \mathbf{O}(\|\mathbf{x}\|^3).$

- ◆ $\mathbf{J}(\mathbf{x}) = \mathbf{J}(\mathbf{0}) + \mathbf{M}(\mathbf{0}, \mathbf{x}) + \mathbf{O}(\|\mathbf{x}\|^2).$

The ESM algorithm

- A second order approximation without Hessians computation:

$$\mathbf{y}(\mathbf{x}) = \mathbf{y}(\mathbf{0}) + \frac{1}{2} (\mathbf{J}(\mathbf{0}) + \mathbf{J}(\mathbf{x})) \mathbf{x} + \mathbf{O}(\|\mathbf{x}\|^3).$$

- ◆ **Demonstration:**

- ◆ $\mathbf{y}(\mathbf{x}) = \mathbf{y}(\mathbf{0}) + \mathbf{J}(\mathbf{0})\mathbf{x} + \frac{1}{2}\mathbf{M}(\mathbf{0}, \mathbf{x})\mathbf{x} + \mathbf{O}(\|\mathbf{x}\|^3).$

- ◆ $\mathbf{J}(\mathbf{x}) = \mathbf{J}(\mathbf{0}) + \mathbf{M}(\mathbf{0}, \mathbf{x}) + \mathbf{O}(\|\mathbf{x}\|^2).$

- ◆ $\mathbf{M}(\mathbf{0}, \mathbf{x}) = \mathbf{J}(\mathbf{x}) - \mathbf{J}(\mathbf{0}) + \mathbf{O}(\|\mathbf{x}\|^2).$

The ESM algorithm

- A second order approximation without Hessians computation:
$$\mathbf{y}(\mathbf{x}) = \mathbf{y}(\mathbf{0}) + \frac{1}{2} (\mathbf{J}(\mathbf{0}) + \mathbf{J}(\mathbf{x})) \mathbf{x} + \mathbf{O}(\|\mathbf{x}\|^3).$$
 - ◆ **Demonstration:**
 - ◆ $\mathbf{y}(\mathbf{x}) = \mathbf{y}(\mathbf{0}) + \mathbf{J}(\mathbf{0})\mathbf{x} + \frac{1}{2}\mathbf{M}(\mathbf{0}, \mathbf{x})\mathbf{x} + \mathbf{O}(\|\mathbf{x}\|^3).$
 - ◆ $\mathbf{J}(\mathbf{x}) = \mathbf{J}(\mathbf{0}) + \mathbf{M}(\mathbf{0}, \mathbf{x}) + \mathbf{O}(\|\mathbf{x}\|^2).$
 - ◆ $\mathbf{M}(\mathbf{0}, \mathbf{x}) = \mathbf{J}(\mathbf{x}) - \mathbf{J}(\mathbf{0}) + \mathbf{O}(\|\mathbf{x}\|^2).$
 - ◆ $\mathbf{y}(\mathbf{x}) = \mathbf{y}(\mathbf{0}) + \mathbf{J}(\mathbf{0})\mathbf{x} + \frac{1}{2} (\mathbf{J}(\mathbf{x}) - \mathbf{J}(\mathbf{0})) \mathbf{x} + \mathbf{O}(\|\mathbf{x}\|^3).$

The ESM algorithm

- A second order approximation without Hessians computation:

$$\mathbf{y}(\mathbf{x}) = \mathbf{y}(\mathbf{0}) + \frac{1}{2} (\mathbf{J}(\mathbf{0}) + \mathbf{J}(\mathbf{x})) \mathbf{x} + \mathbf{O}(\|\mathbf{x}\|^3).$$

- ◆ **Demonstration:**

- ◆ $\mathbf{y}(\mathbf{x}) = \mathbf{y}(\mathbf{0}) + \mathbf{J}(\mathbf{0})\mathbf{x} + \frac{1}{2}\mathbf{M}(\mathbf{0}, \mathbf{x})\mathbf{x} + \mathbf{O}(\|\mathbf{x}\|^3).$

- ◆ $\mathbf{J}(\mathbf{x}) = \mathbf{J}(\mathbf{0}) + \mathbf{M}(\mathbf{0}, \mathbf{x}) + \mathbf{O}(\|\mathbf{x}\|^2).$

- ◆ $\mathbf{M}(\mathbf{0}, \mathbf{x}) = \mathbf{J}(\mathbf{x}) - \mathbf{J}(\mathbf{0}) + \mathbf{O}(\|\mathbf{x}\|^2).$

- ◆ $\mathbf{y}(\mathbf{x}) = \mathbf{y}(\mathbf{0}) + \mathbf{J}(\mathbf{0})\mathbf{x} + \frac{1}{2} (\mathbf{J}(\mathbf{x}) - \mathbf{J}(\mathbf{0})) \mathbf{x} + \mathbf{O}(\|\mathbf{x}\|^3).$

- ◆ $\mathbf{y}(\mathbf{x}) = \mathbf{y}(\mathbf{0}) + \frac{1}{2} (\mathbf{J}(\mathbf{0}) + \mathbf{J}(\mathbf{x})) \mathbf{x} + \mathbf{O}(\|\mathbf{x}\|^3).$

The ESM algorithm

- A second order approximation without Hessians computation:
$$\mathbf{y}(\mathbf{x}) = \mathbf{y}(\mathbf{0}) + \frac{1}{2} (\mathbf{J}(\mathbf{0}) + \mathbf{J}(\mathbf{x})) \mathbf{x} + \mathbf{O}(\|\mathbf{x}\|^3).$$

The ESM algorithm

- A second order approximation without Hessians computation:
$$\mathbf{y}(\mathbf{x}) = \mathbf{y}(\mathbf{0}) + \frac{1}{2} (\mathbf{J}(\mathbf{0}) + \mathbf{J}(\mathbf{x})) \mathbf{x} + \mathbf{O}(\|\mathbf{x}\|^3).$$
- $\mathbf{y}(\mathbf{x}_0) \approx \mathbf{y}(\mathbf{0}) + \frac{1}{2} (\mathbf{J}(\mathbf{0}) + \mathbf{J}(\mathbf{x}_0)) \mathbf{x}_0$

The ESM algorithm

- A second order approximation without Hessians computation:

$$\mathbf{y}(\mathbf{x}) = \mathbf{y}(\mathbf{0}) + \frac{1}{2} (\mathbf{J}(\mathbf{0}) + \mathbf{J}(\mathbf{x})) \mathbf{x} + \mathbf{O}(\|\mathbf{x}\|^3).$$

- $\mathbf{y}(\mathbf{x}_0) \approx \mathbf{y}(\mathbf{0}) + \frac{1}{2} (\mathbf{J}(\mathbf{0}) + \mathbf{J}(\mathbf{x}_0)) \mathbf{x}_0$

- We demonstrate that:

$$\mathbf{J}(\mathbf{0}) + \mathbf{J}(\mathbf{x}_0) = \begin{pmatrix} \mathbf{J}_{\mathcal{I}} + \mathbf{J}_{\mathcal{I}^*} \end{pmatrix} \mathbf{J}_{\mathbf{w}} (\mathbf{J}_{\mathbf{G}}(\mathbf{0}) + \mathbf{J}_{\mathbf{G}}(\mathbf{x}_0))$$

The ESM algorithm

- A second order approximation without Hessians computation:

$$\mathbf{y}(\mathbf{x}) = \mathbf{y}(\mathbf{0}) + \frac{1}{2} (\mathbf{J}(\mathbf{0}) + \mathbf{J}(\mathbf{x})) \mathbf{x} + \mathbf{O}(\|\mathbf{x}\|^3).$$

- $\mathbf{y}(\mathbf{x}_0) \approx \mathbf{y}(\mathbf{0}) + \frac{1}{2} (\mathbf{J}(\mathbf{0}) + \mathbf{J}(\mathbf{x}_0)) \mathbf{x}_0$

- We demonstrate that:

$$\mathbf{J}(\mathbf{0}) + \mathbf{J}(\mathbf{x}_0) = \left(\mathbf{J}_{\mathcal{I}} + \mathbf{J}_{\mathcal{I}^*} \right) \mathbf{J}_{\mathbf{w}} \left(\mathbf{J}_{\mathbf{G}}(\mathbf{0}) + \mathbf{J}_{\mathbf{G}}(\mathbf{x}_0) \right)$$

Computed using \mathcal{I} gradients

The ESM algorithm

- A second order approximation without Hessians computation:

$$\mathbf{y}(\mathbf{x}) = \mathbf{y}(\mathbf{0}) + \frac{1}{2} (\mathbf{J}(\mathbf{0}) + \mathbf{J}(\mathbf{x})) \mathbf{x} + \mathbf{O}(\|\mathbf{x}\|^3).$$
- $\mathbf{y}(\mathbf{x}_0) \approx \mathbf{y}(\mathbf{0}) + \frac{1}{2} (\mathbf{J}(\mathbf{0}) + \mathbf{J}(\mathbf{x}_0)) \mathbf{x}_0$
- We demonstrate that:

$$\mathbf{J}(\mathbf{0}) + \mathbf{J}(\mathbf{x}_0) = \left(\mathbf{J}_{\mathcal{I}} + \boxed{\mathbf{J}_{\mathcal{I}^*}} \right) \mathbf{J}_{\mathbf{w}} \left(\mathbf{J}_{\mathbf{G}}(\mathbf{0}) + \mathbf{J}_{\mathbf{G}}(\mathbf{x}_0) \right)$$

Computed using \mathcal{I}^* gradients

The ESM algorithm

- A second order approximation without Hessians computation:

$$\mathbf{y}(\mathbf{x}) = \mathbf{y}(\mathbf{0}) + \frac{1}{2} (\mathbf{J}(\mathbf{0}) + \mathbf{J}(\mathbf{x})) \mathbf{x} + \mathbf{O}(\|\mathbf{x}\|^3).$$
- $\mathbf{y}(\mathbf{x}_0) \approx \mathbf{y}(\mathbf{0}) + \frac{1}{2} (\mathbf{J}(\mathbf{0}) + \mathbf{J}(\mathbf{x}_0)) \mathbf{x}_0$
- We demonstrate that:

$$\mathbf{J}(\mathbf{0}) + \mathbf{J}(\mathbf{x}_0) = \left(\mathbf{J}_{\mathcal{I}} + \mathbf{J}_{\mathcal{I}^*} \right) \mathbf{J}_{\mathbf{w}} (\mathbf{J}_{\mathbf{G}}(\mathbf{0}) + \mathbf{J}_{\mathbf{G}}(\mathbf{x}_0))$$

Computed using \mathbf{p}_i^*

The ESM algorithm

- A second order approximation without Hessians computation:

$$\mathbf{y}(\mathbf{x}) = \mathbf{y}(\mathbf{0}) + \frac{1}{2} (\mathbf{J}(\mathbf{0}) + \mathbf{J}(\mathbf{x})) \mathbf{x} + \mathbf{O}(\|\mathbf{x}\|^3).$$
- $\mathbf{y}(\mathbf{x}_0) \approx \mathbf{y}(\mathbf{0}) + \frac{1}{2} (\mathbf{J}(\mathbf{0}) + \mathbf{J}(\mathbf{x}_0)) \mathbf{x}_0$
- We demonstrate that:

$$\mathbf{J}(\mathbf{0}) + \mathbf{J}(\mathbf{x}_0) = \begin{pmatrix} \mathbf{J}_{\mathcal{I}} + \mathbf{J}_{\mathcal{I}^*} \end{pmatrix} \mathbf{J}_{\mathbf{w}} \boxed{(\mathbf{J}_{\mathbf{G}}(\mathbf{0}) + \mathbf{J}_{\mathbf{G}}(\mathbf{x}_0))}$$

Depends on the homography parameterization

The ESM algorithm

- A second order approximation without Hessians computation:

$$\mathbf{y}(\mathbf{x}) = \mathbf{y}(\mathbf{0}) + \frac{1}{2} (\mathbf{J}(\mathbf{0}) + \mathbf{J}(\mathbf{x})) \mathbf{x} + \mathbf{O}(\|\mathbf{x}\|^3).$$

- $\mathbf{y}(\mathbf{x}_0) \approx \mathbf{y}(\mathbf{0}) + \frac{1}{2} (\mathbf{J}(\mathbf{0}) + \mathbf{J}(\mathbf{x}_0)) \mathbf{x}_0$

- We demonstrate that:

$$\mathbf{J}(\mathbf{0}) + \mathbf{J}(\mathbf{x}_0) = \begin{pmatrix} \mathbf{J}_{\mathcal{I}} + \mathbf{J}_{\mathcal{I}^*} \\ \mathbf{J}_{\mathbf{w}} \end{pmatrix} (\mathbf{J}_{\mathbf{G}}(\mathbf{0}) + \mathbf{J}_{\mathbf{G}}(\mathbf{x}_0))$$

- How to compute $\mathbf{J}_{\mathbf{G}}(\mathbf{x}_0)$ without knowing \mathbf{x}_0 ?
- The solution is to parameterize the projective transformation using the Lie Algebra $\mathfrak{sl}(3)$.

Lie Algebra parameterization

- Since $\mathbf{G}(\mathbf{x})$ is in the Lie group $\mathbb{S}\mathbb{L}(3)$ and in the neighborhood of the identity matrix, it can be written as $\mathbf{G}(\mathbf{x}) = \exp(\mathbf{A}(\mathbf{x}))$ where $\mathbf{A}(\mathbf{x})$ is in $\mathfrak{sl}(3)$ (the Lie algebra associated to $\mathbb{S}\mathbb{L}(3)$).
- The parameters \mathbf{x} can be chosen such that $\mathbf{A}(\mathbf{x}) = \sum_{i=1}^8 x_i \mathbf{A}_i$ where $\{\mathbf{A}_i, i \in \{1, \dots, 8\}\}$ is a basis of $\mathfrak{sl}(3)$:

$$\begin{aligned}
 \mathbf{A}_1 &= \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \mathbf{A}_3 = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \mathbf{A}_5 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \mathbf{A}_7 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix} \\
 \mathbf{A}_2 &= \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}, \mathbf{A}_4 = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \mathbf{A}_6 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \mathbf{A}_8 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}
 \end{aligned}$$

- With such parameterization, we demonstrate that: $\mathbf{J}_{\mathbf{G}}(\mathbf{x}_0)\mathbf{x}_0 = \mathbf{J}_{\mathbf{G}}(\mathbf{0})\mathbf{x}_0$
- Then, we obtain:

$$\mathbf{y}(\mathbf{x}_0) \approx \mathbf{y}(\mathbf{0}) + \frac{1}{2} ((\mathbf{J}_{\mathcal{I}} + \mathbf{J}_{\mathcal{I}^*}) \mathbf{J}_{\mathbf{w}} \mathbf{J}_{\mathbf{G}}(\mathbf{0})) \mathbf{x}_0$$

The ESM algorithm

- We define

$$\mathbf{J}_{esm} = \frac{1}{2} (\mathbf{J}_{\mathcal{I}} + \mathbf{J}_{\mathcal{I}^*}) \mathbf{J}_{\mathbf{w}} \mathbf{J}_{\mathbf{G}}(\mathbf{0})$$

- We minimize iteratively the following cost function:

$$f(\mathbf{x}) = \frac{1}{2} \|\mathbf{y}(\mathbf{0}) + \mathbf{J}_{esm} \mathbf{x}\|^2$$

- We demonstrate that the optimal solution can be written as:

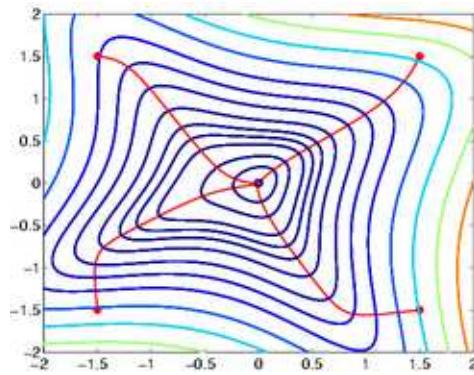
$$\mathbf{x}_0 = (\mathbf{J}_{esm}^{\top} \mathbf{J}_{esm})^{-1} \mathbf{J}_{esm}^{\top} \mathbf{y}(\mathbf{0})$$

- Once again, the approximation $\hat{\mathbf{G}}$ is updated:

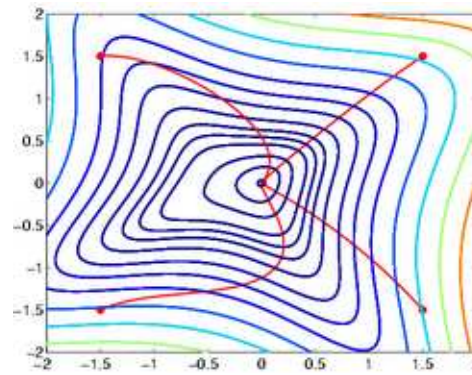
$$\hat{\mathbf{G}} \leftarrow \hat{\mathbf{G}} \mathbf{G}(\mathbf{x}_0)$$

Speed of convergence

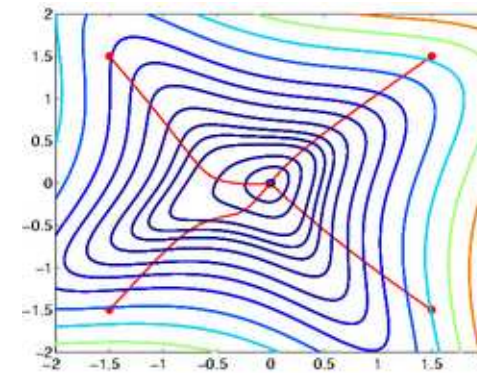
Quadratic signals: $y_i(\mathbf{x}) = c_i + \mathbf{b}_i^\top \mathbf{x} + \mathbf{x}^\top \mathbf{A}_i \mathbf{x}$.



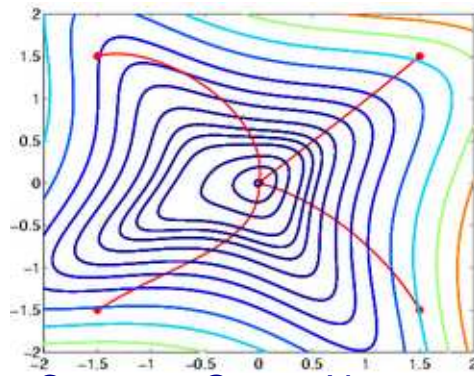
Varying Steepest Desc.



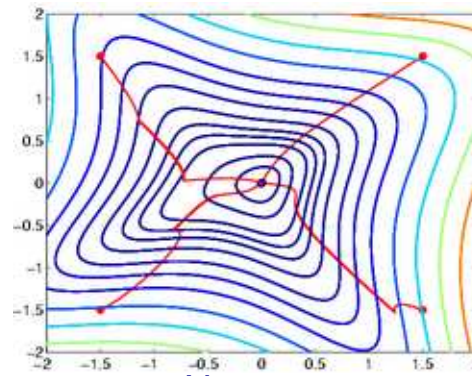
Constant Steepest Desc.



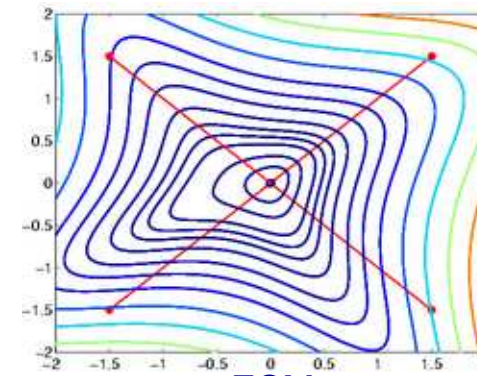
Varying Gauss-Newton



Constant Gauss-Newton



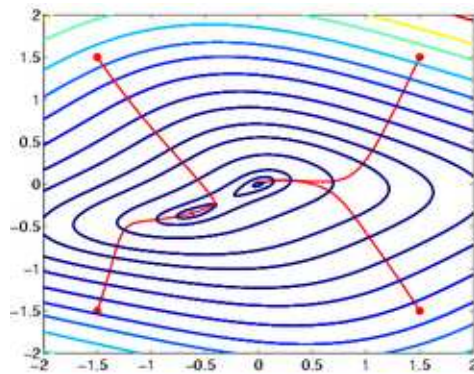
Newton



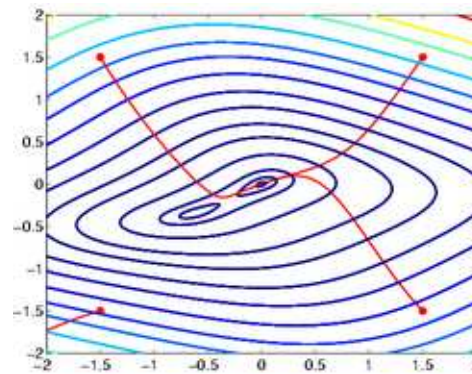
ESM

Avoiding local minima

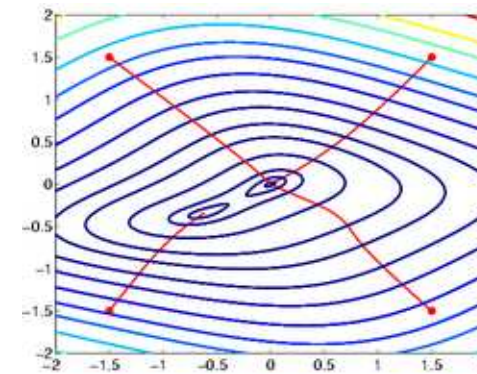
Quadratic signals: $y_i(\mathbf{x}) = c_i + \mathbf{b}_i^\top \mathbf{x} + \mathbf{x}^\top \mathbf{A}_i \mathbf{x}$.



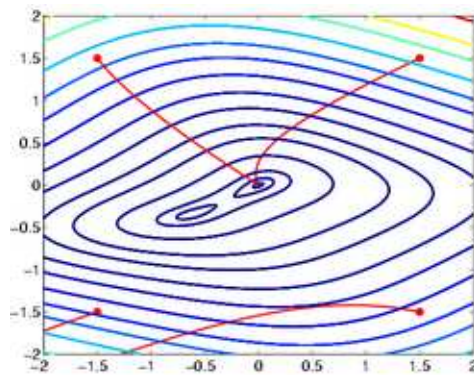
Varying Steepest Desc.



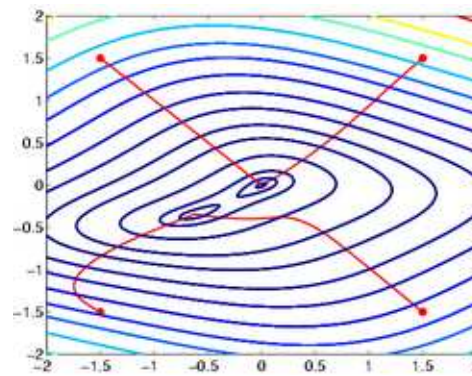
Constant Steepest Desc.



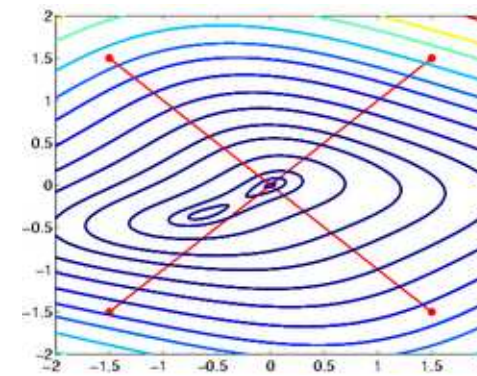
Varying Gauss-Newton



Constant Gauss-Newton



Newton

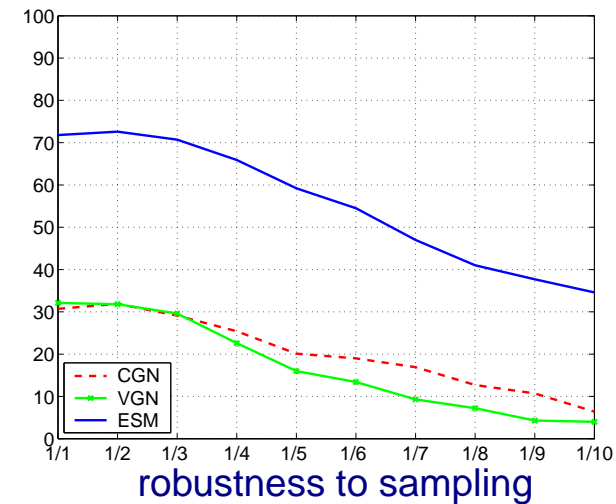
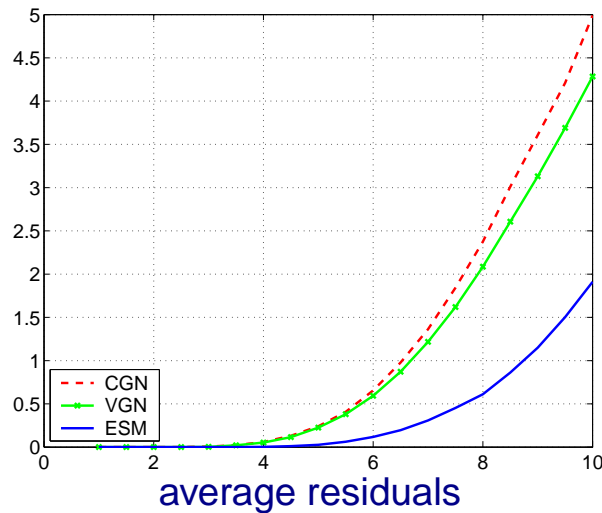
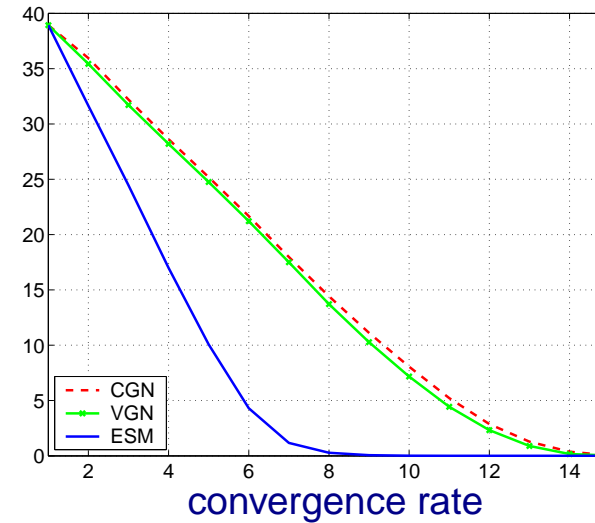
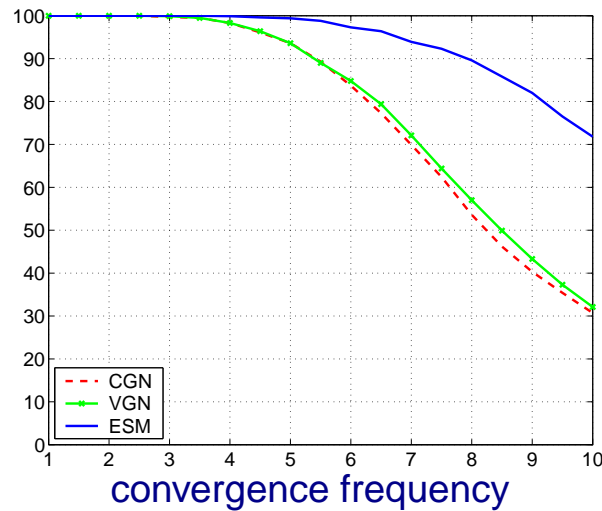


ESM

Comparison with standard methods



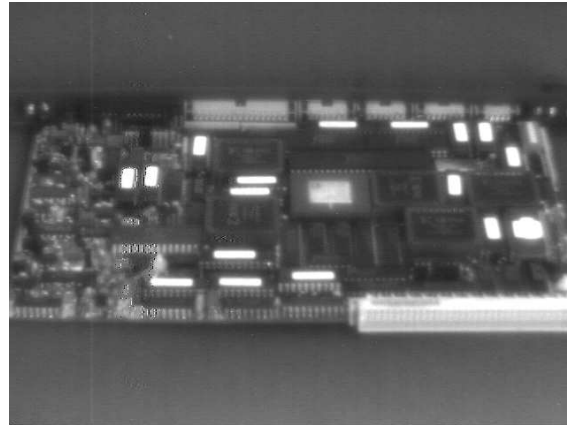
Comparison with standard methods



PART II : Experimental Results & applications

- Tracking sequences off-line
- Real-time Car-platooning task
- Real-time Visual Servoing

Experimental Results



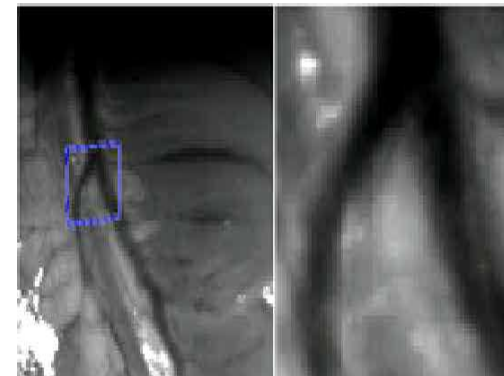
Elect. board



Blimp



CD



Beating Heart

Applications



Applications



Applications



Applications

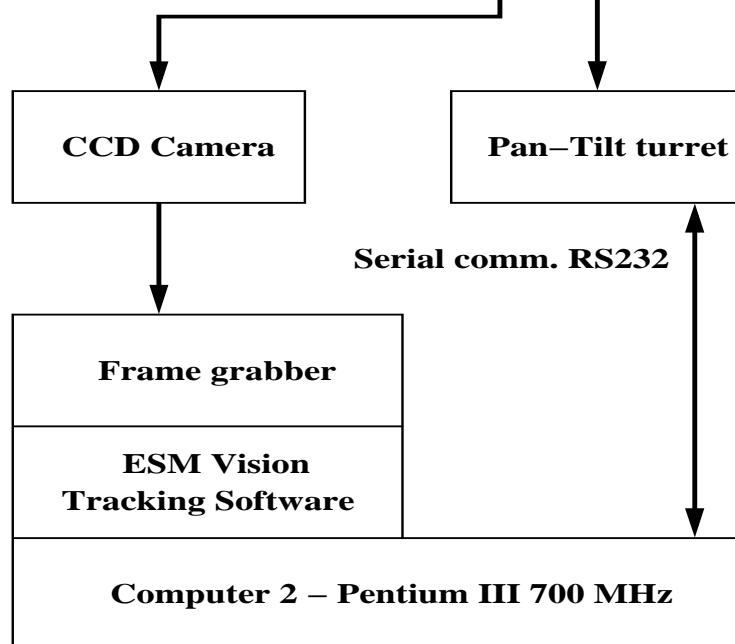


CCD Camera

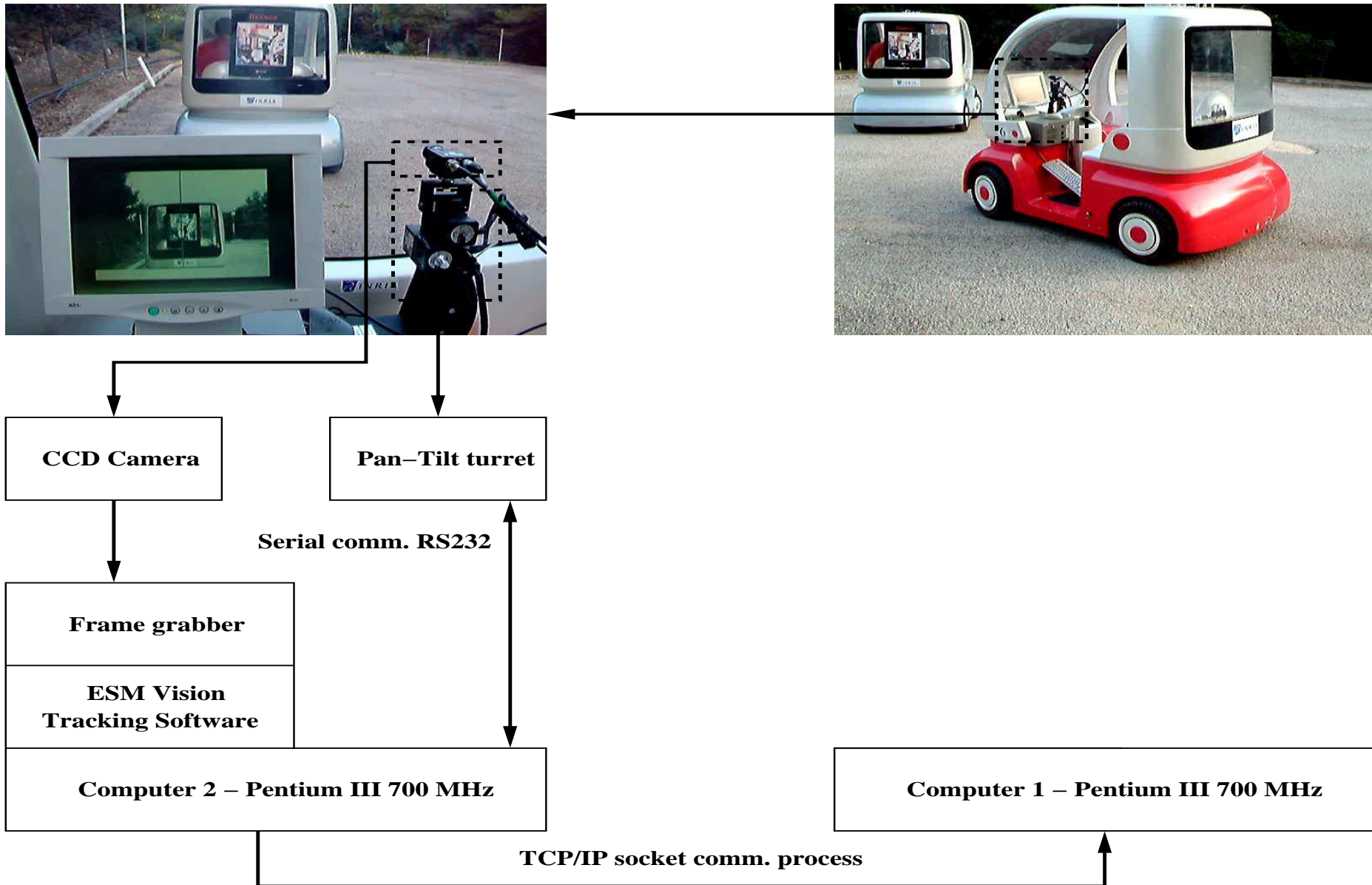
Pan-Tilt turret



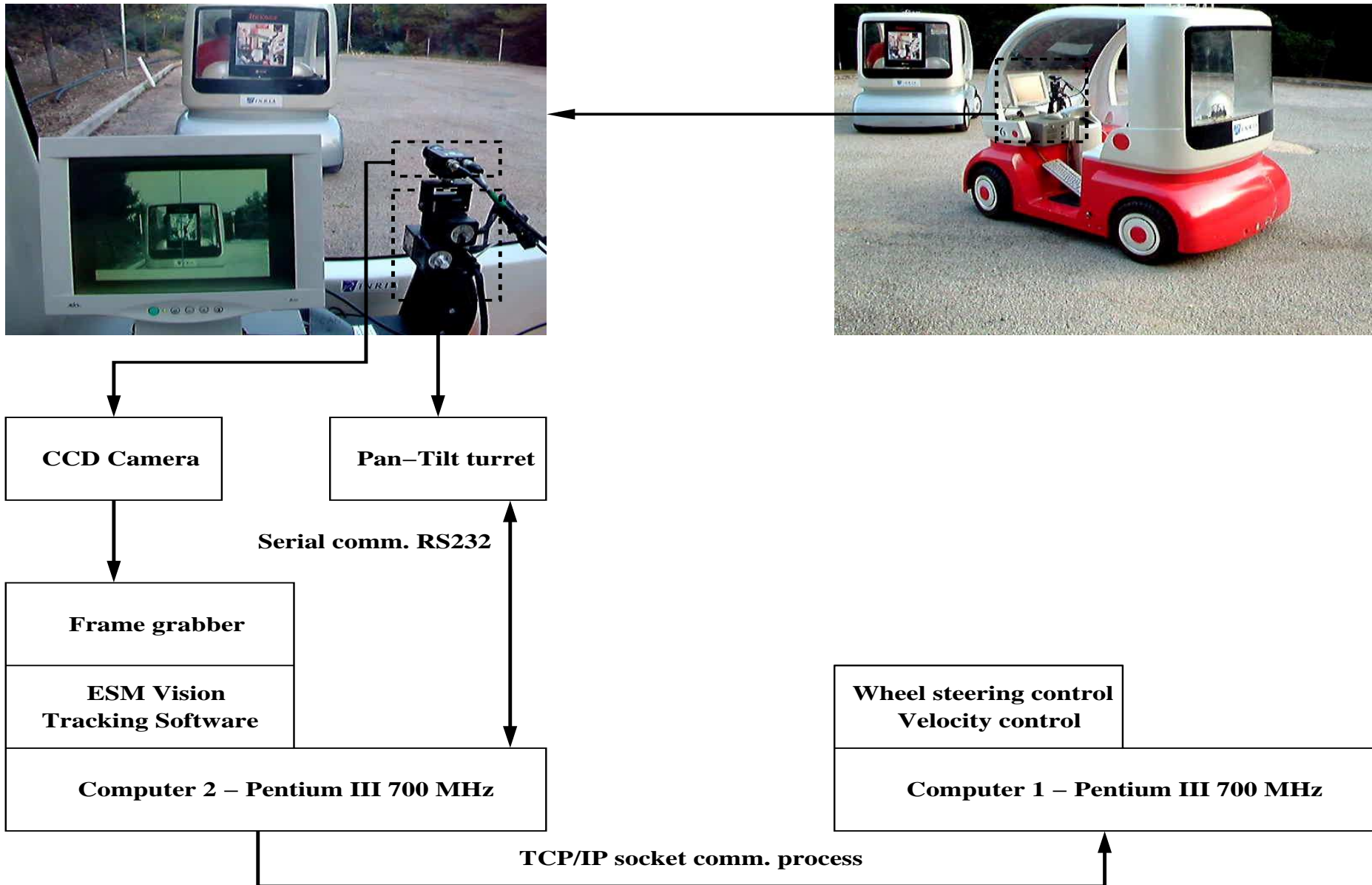
Applications



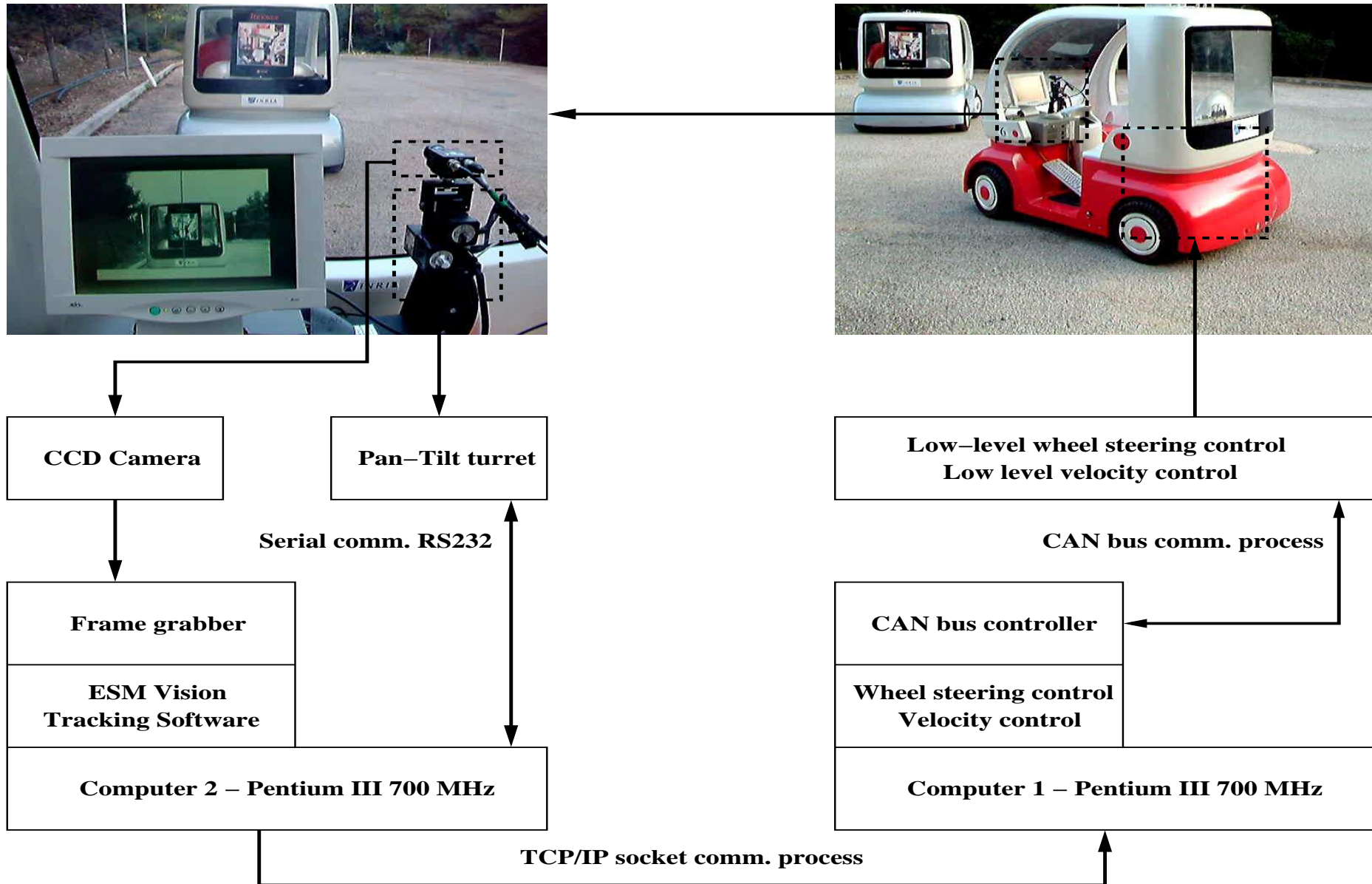
Applications



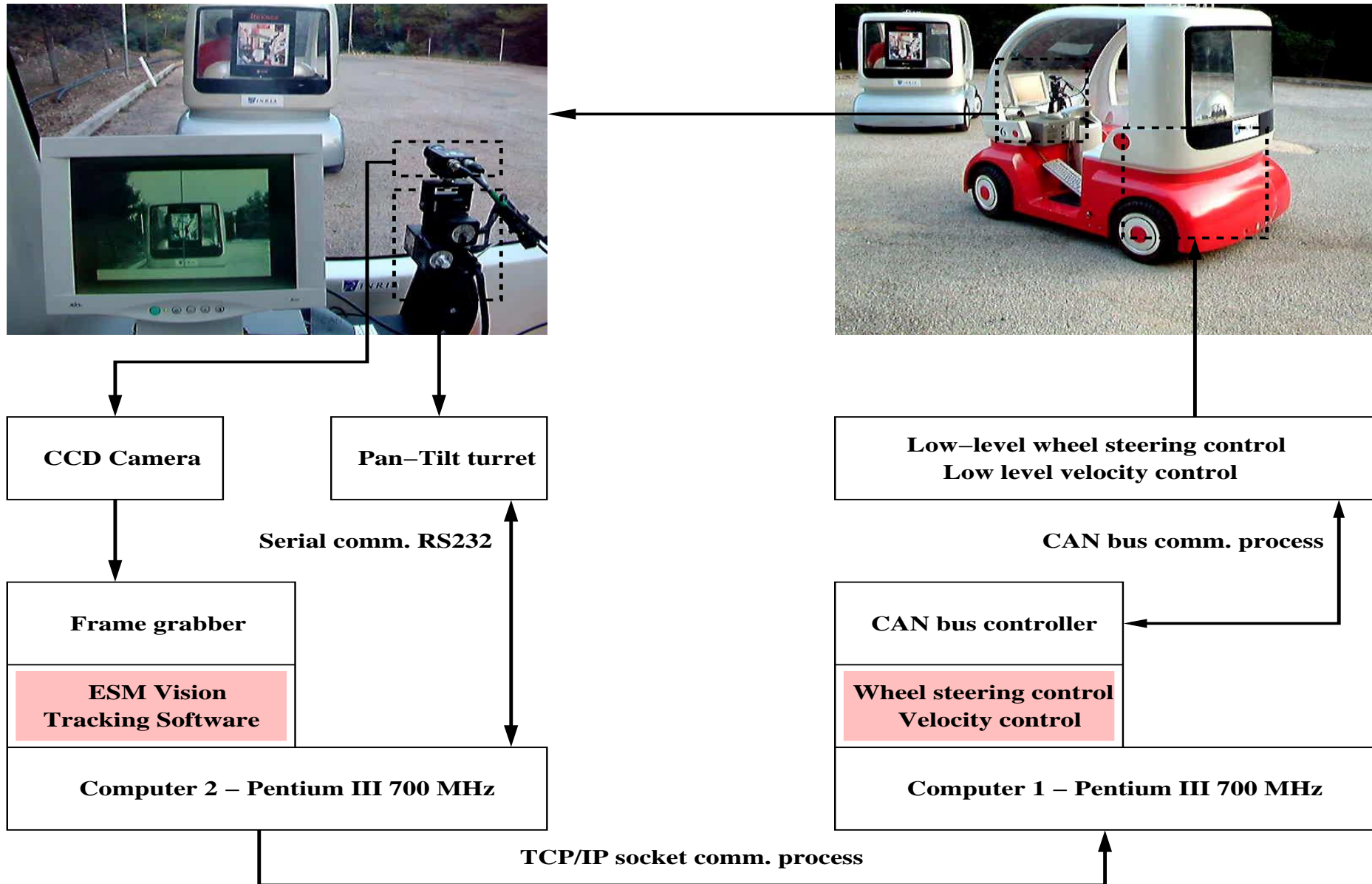
Applications



Applications



Applications



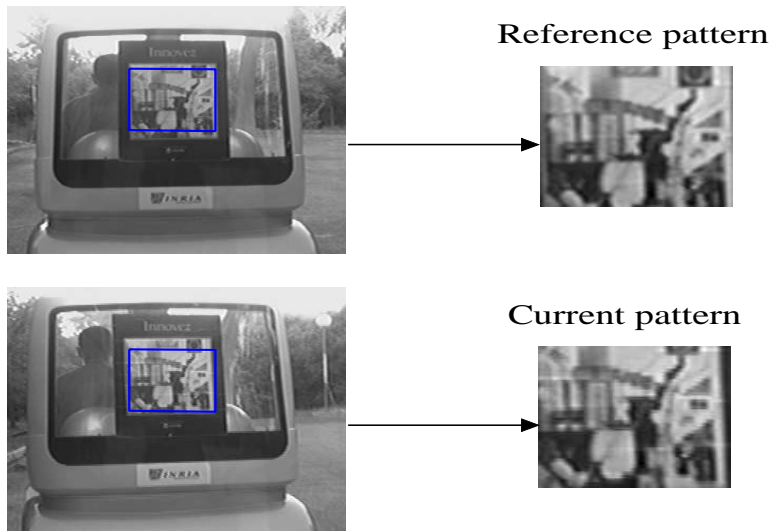
Applications

- At the desired position and orientation, a reference pattern is selected.



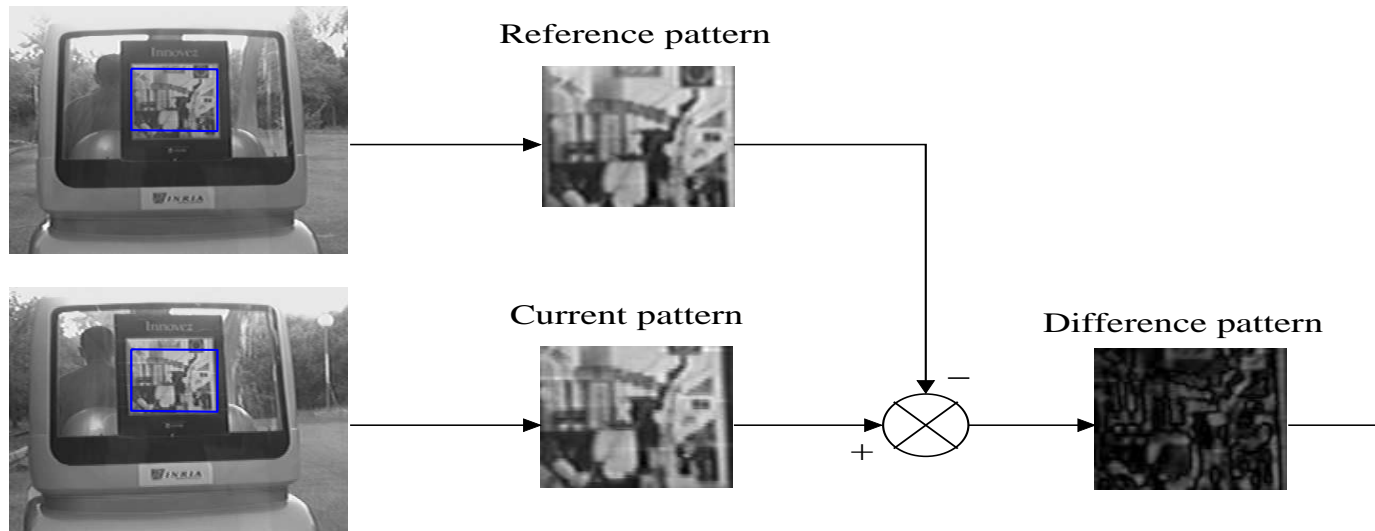
Applications

- At the desired position and orientation, a reference pattern is selected.



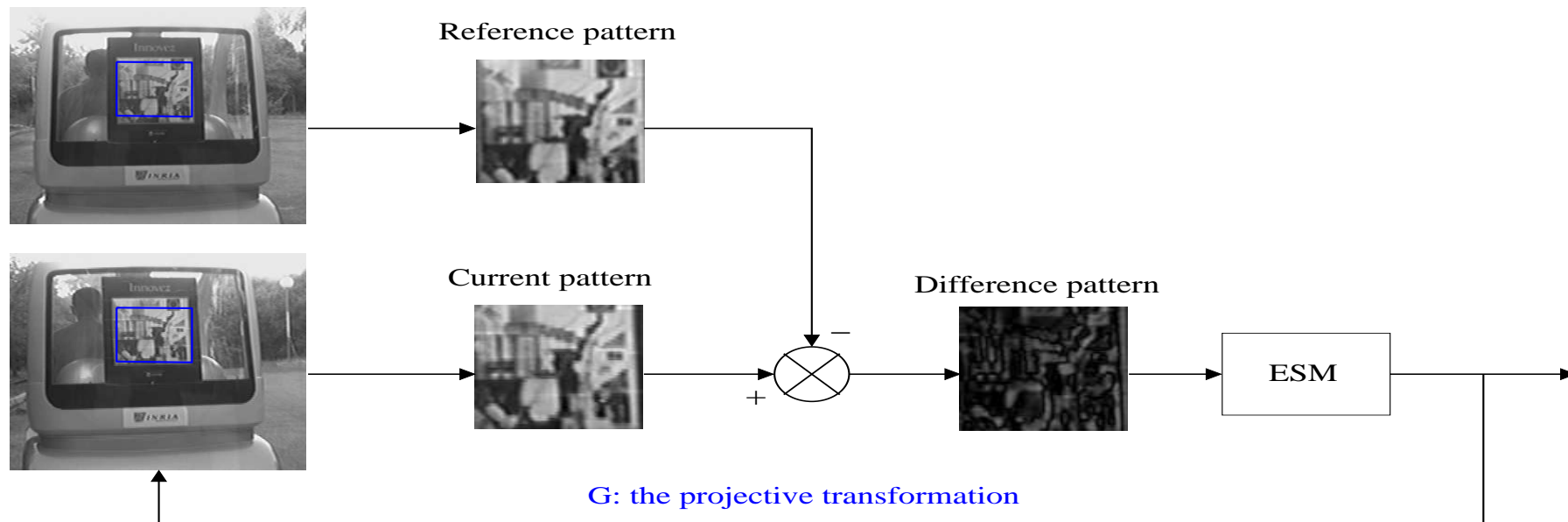
Applications

- At the desired position and orientation, a reference pattern is selected.



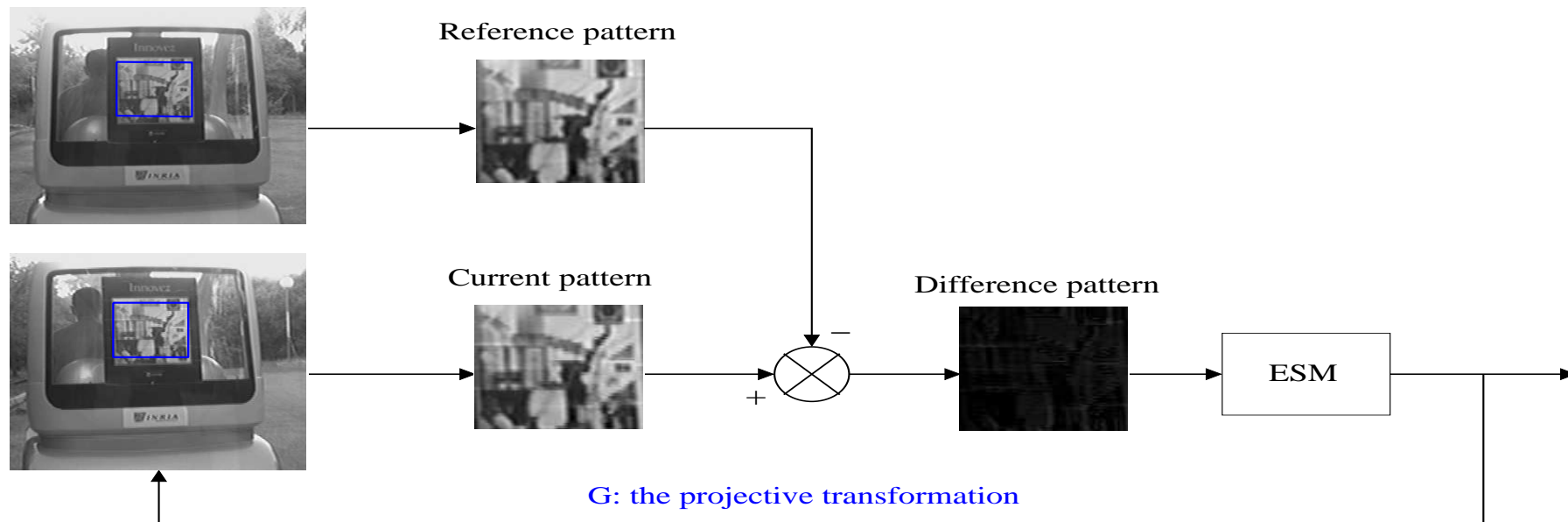
Applications

- At the desired position and orientation, a reference pattern is selected.
- The projective transformation of the pattern due to the relative motion between the two cars is estimated by the **ESM tracking algorithm** [Benhimane&Malis, IROS04].



Applications

- At the desired position and orientation, a reference pattern is selected.
- The projective transformation of the pattern due to the relative motion between the two cars is estimated by the **ESM tracking algorithm** [Benhimane&Malis, IROS04].



Applications

- At the desired position and orientation, a reference pattern is selected.
- The projective transformation of the pattern due to the relative motion between the two cars is estimated by the **ESM tracking algorithm** [Benhimane&Malis, IROS04].
- The relative rotation and translation are computed by decomposing the homography estimated.

$$\mathbf{G} = \mathbf{K}(\mathbf{R} + \mathbf{t}\mathbf{n}^{*\top})\mathbf{K}^{-1}$$

Applications

- At the desired position and orientation, a reference pattern is selected.
- The projective transformation of the pattern due to the relative motion between the two cars is estimated by the **ESM tracking algorithm** [Benhimane&Malis, IROS04].
- The relative rotation and translation are computed by decomposing the homography estimated.

$$\mathbf{G} = \mathbf{K}(\mathbf{R} + \mathbf{t}\mathbf{n}^{*\top})\mathbf{K}^{-1}$$

- The orientation error and the position error (up to a scale factor) are estimated.

Applications

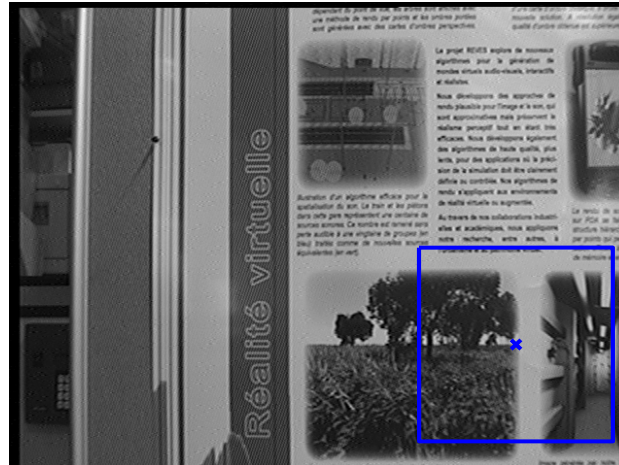


Car Platooning 1

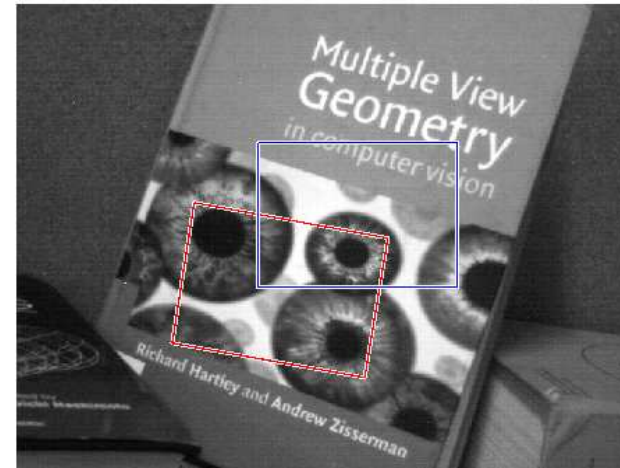


Car Platooning 2

Applications



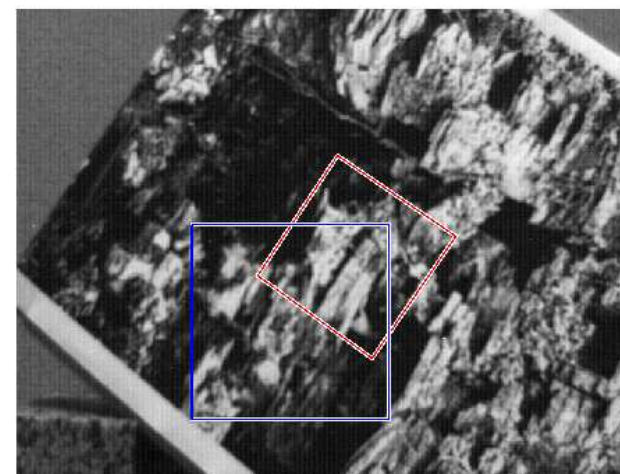
Visual servoing ??



Visual Servoing 1



Visual Servoing 2



Visual Servoing 3

Thank You For Your Attention !!!

Questions ???