3D VISION

3D Reconstruction

Systems Engineering and Automation

Miguel Hernández University

Table of Contents

3D Reconstruction

- **Objective**: to know the 3D coordinates of a point in space:

$$(x_w, y_w, z_w) \equiv (X, Y, Z)$$

- **DATA**:
  - We assume known the projection of the point (X, Y, Z) on the image plane of the two cameras
  - From the lateral coordinates of the pixels we get the coordinates without distortion.

Left camera Pixel $\left(x_f{}^l, y_f{}^l\right) \Rightarrow \left(x_n{}^l, y_n{}^l\right) \Rightarrow (x, y) \Rightarrow \tilde{m}_u$

Right camera Pixel $\left(x_f{}^r, y_f{}^r\right) \Rightarrow \left(x_n{}^r, y_n{}^r\right) \Rightarrow (x', y') \Rightarrow \tilde{m}'_u$

- **CALIBRATION DATA** : $\boxed{\tilde{m}_u \cong A\ [I \quad 0]\ \tilde{M}_c}$ $\boxed{\tilde{m}'_u \cong A'\ [R \quad t]\ \tilde{M}_c}$

Distortion:

$$\begin{bmatrix} x_n^d \\ y_n^d \\ 1 \end{bmatrix} \cong A^{-1} \cdot \tilde{m} \rightarrow \left.\begin{matrix} x_n^d = x_n(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) + 2p_1 x_n y_n + p_2(2x_n^2 + r^2) + s_1 r^2 + s_2 r^4 \\ y_n^d = y_n\ (1 + k_1 r^2 + k_2 r^4 + k_3 r^6) + 2p_2 x_n y_n + p_1(2y_n^2 + r^2) + s_1 r^2 + s_2 r^4 \\ r^2 = x_n^2 + y_n^2 \end{matrix}\right\} \rightarrow \tilde{m}_u \cong A \cdot \begin{bmatrix} x_n \\ y_n \\ 1 \end{bmatrix}$$

3D Reconstruction

Left image

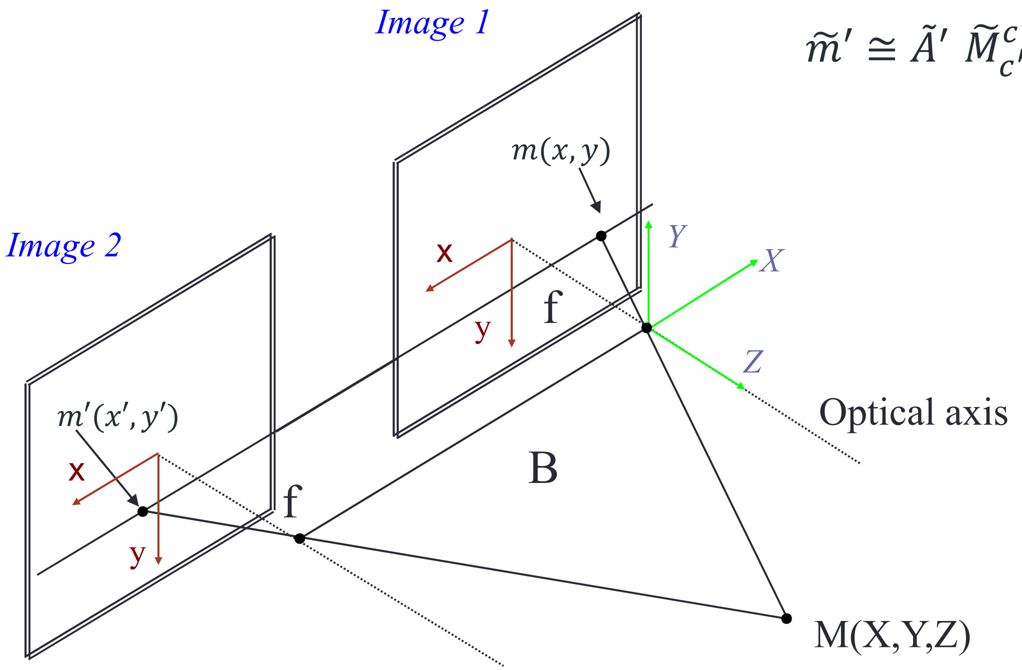
Right image

3D Reconstruction. Aligned Axes

- Aligned Axes
  - 3D coordinate system referred to the left camera
  - 2D coordinate systems: normalized central coordinates ($f = f'$)

$$\tilde{m} \cong \tilde{A} \cdot \tilde{M}_c = [A \quad 0] \begin{bmatrix} M \\ 1 \end{bmatrix} \qquad \tilde{M}_{c'}^{c} = \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix}$$

$$\tilde{m}' \cong \tilde{A}' \tilde{M}_{c'}^{c} \cdot \tilde{M}_c = A' \, [R \quad t] \begin{bmatrix} M \\ 1 \end{bmatrix}$$



$$\boxed{\tilde{m} \cong A \; M}$$

$$\boxed{\tilde{m}' \cong A' \; RM + A't}$$

$$R = I$$

$$A = A' = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad t = \begin{bmatrix} B \\ 0 \\ 0 \end{bmatrix}$$

3D Reconstruction. Aligned Axes

- Reconstruction equations:

$$\boxed{\tilde{m} \cong A\ M}$$

$$\boxed{\tilde{m}' \cong A'\ RM + A't}$$

$$R = I \qquad t = \begin{bmatrix} B \\ 0 \\ 0 \end{bmatrix} \qquad A = A' = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

- Left camera:

$$\begin{bmatrix} \lambda x \\ \lambda y \\ \lambda \end{bmatrix} = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \Rightarrow \left.\begin{matrix} \lambda x = f\ X \\ \lambda y = f\ Y \\ \lambda = Z \end{matrix}\right\} \Rightarrow \boxed{\begin{matrix} X = \dfrac{Z\ x}{f} \\ Y = \dfrac{Z\ y}{f} \end{matrix}}$$

$$\boxed{\begin{matrix} Z = \dfrac{f \cdot B}{(x' - x)} \\ y' = y \end{matrix}}$$

- Right Camera:

$$\begin{bmatrix} \mu x' \\ \mu y' \\ \mu \end{bmatrix} = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} B \\ 0 \\ 0 \end{bmatrix} \Rightarrow \left.\begin{matrix} \mu x' = f\ X + f\ B \\ \mu y' = f\ Y \\ \mu = Z \end{matrix}\right\}$$

$$\begin{matrix} Zx' = f\ \dfrac{Z\ x}{f} + f\ B \\ Zy' = f\ \dfrac{Z\ y}{f} \end{matrix}$$

3D Reconstruction. Aligned Axes

- Reconstruction Equations
  - Coordinates of a point in the Image 1 (Left)

$$\left({x_f}^l, {y_f}^l\right) \Rightarrow \left({x_u}^l, {y_u}^l\right) \equiv (x, y) = m$$

  - Coordinates of a point in the Image 2 (Right)

$$\left({x_f}^r, {y_f}^r\right) \Rightarrow \left({x_u}^r, {y_u}^r\right) \equiv (x', y') = m'$$

  - Disparity: $d = x' - x \quad \text{as} \quad y' = y$

$$X = \frac{B\ x}{d}$$

$$Y = \frac{B\ y}{d}$$

$$Z = \frac{f \cdot B}{d}$$

Disparity Map

- There is an inverse relationship between *Z* and *d*. For certain values of *B, f* : Points with equal disparity have the same *Z*
- For certain values of ***Z, f*** : The higher the baseline *B is,* the greater will be the disparity ***d***, and less influence will have the errors in the estimation of the 3D point coordinates
- For larger values of B it's harder to find the correspondence of a point in the two images.

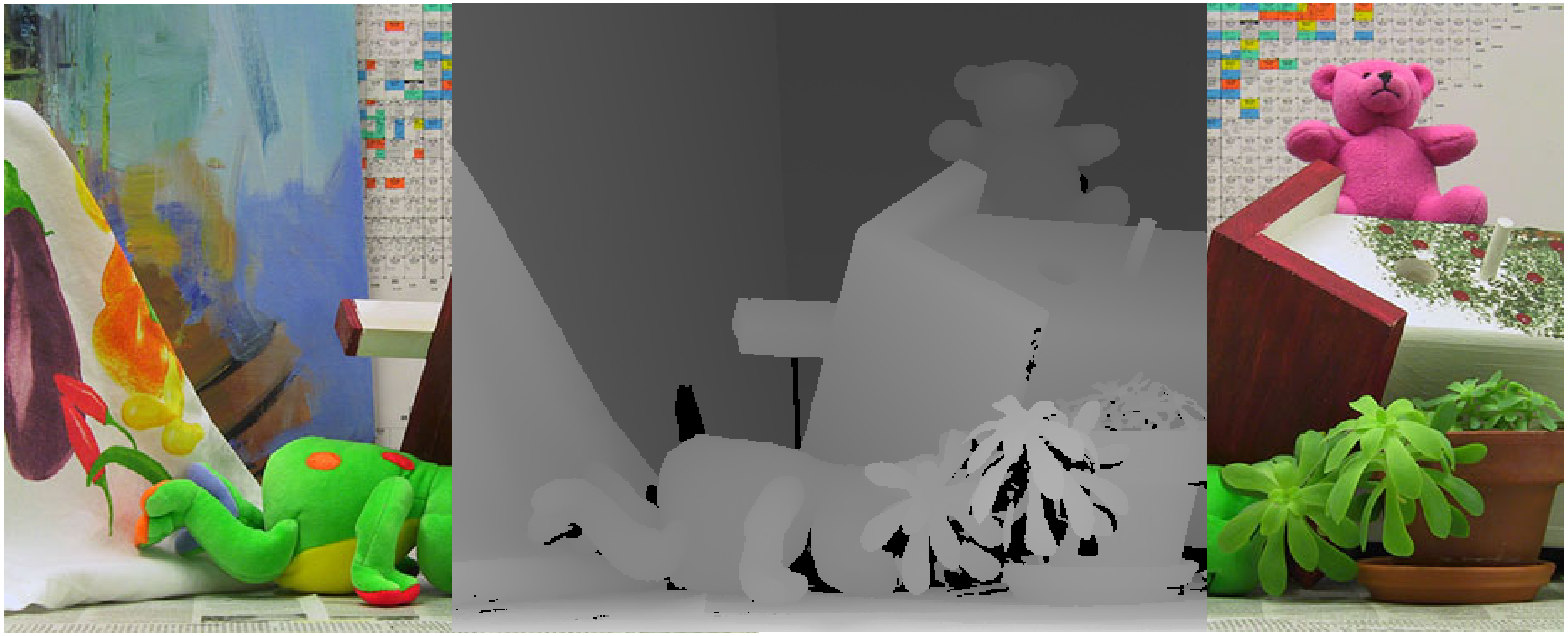
$$X = \frac{B\ x}{d}$$

$$Y = \frac{B\ y}{d}$$

$$Z = \frac{f \cdot B}{d}$$

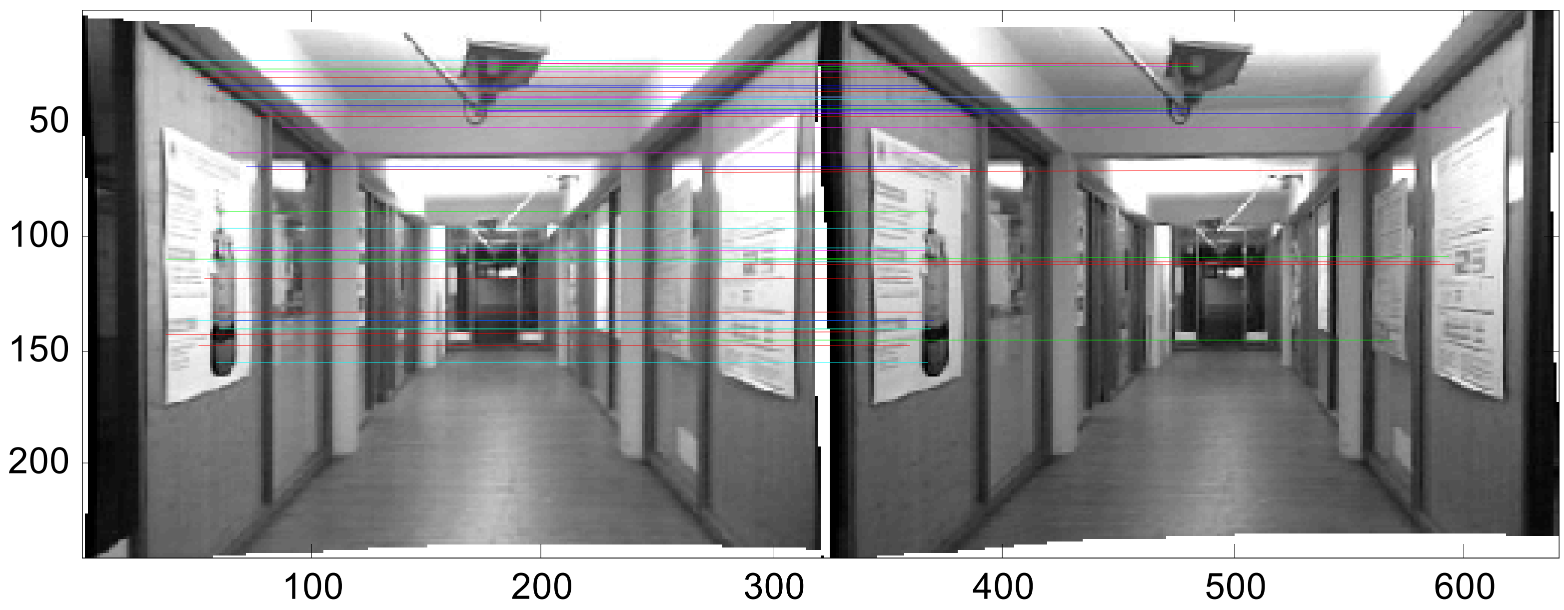
Disparity Map

- The disparity map is inversely proportional to the depth of the points in the scene



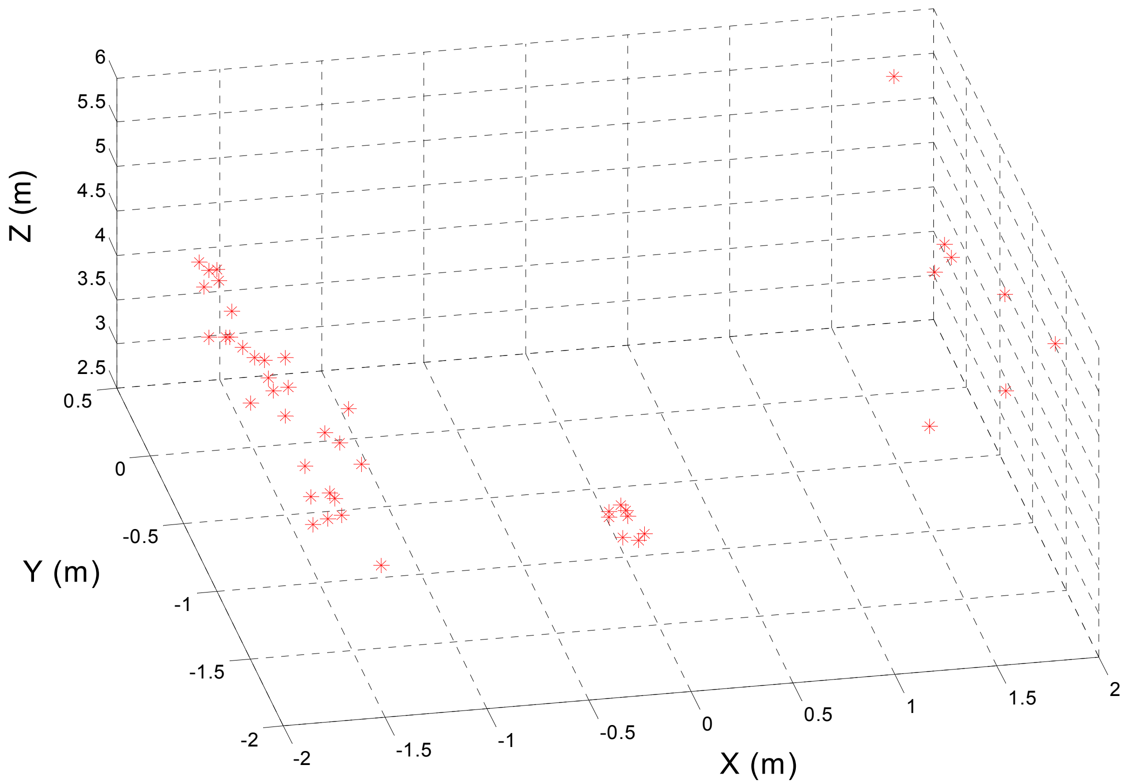
Example: Disperse Stereo Reconstruction

- Point Correspondences. Restrictions:
  - Epipolar: +- 1 pixel
  - SIFT descriptor of each point



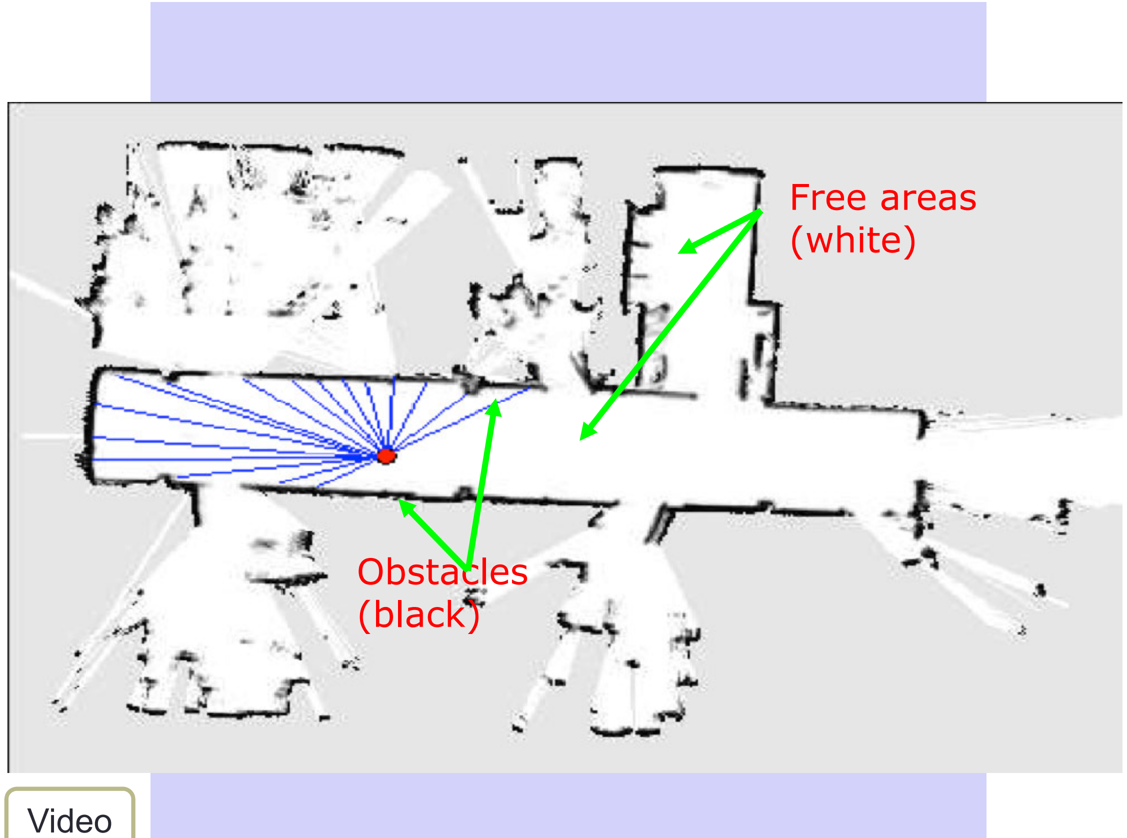
Example: SIFT

- 3D Reconstruction



3D Reconstruction

- Mobile Robotics: map generation



Video

Accuracy

- We are interested in knowing the accuracy of the 3D measurement depending on the parameters of the stereo pair.
- Let's look for an approximate expression of the error for the depth estimation:

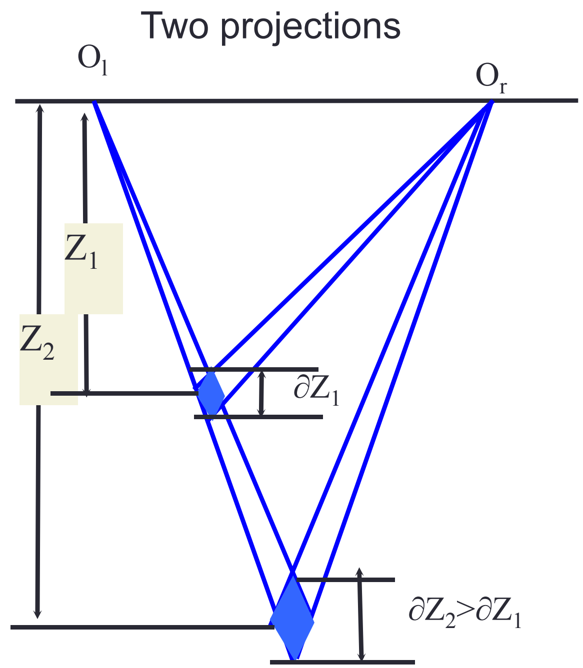
$$Z = f\frac{B}{d}$$

$$\varepsilon_Z = \left|\frac{\partial Z}{\partial B}\right| |\varepsilon_B| + \left|\frac{\partial Z}{\partial d}\right| |\varepsilon_d| + \left|\frac{\partial Z}{\partial f}\right| |\varepsilon_f| \;=\; \left|\frac{f}{d}\right| |\varepsilon_B| + \left|\frac{fB}{d^2}\right| |\varepsilon_d| + \left|\frac{B}{d}\right| |\varepsilon_f|$$

- We're interested in the error depending on the disparity.
  We can consider that:

$$\varepsilon_Z \approx \left|\frac{fB}{d^2}\right| |\varepsilon_d|$$

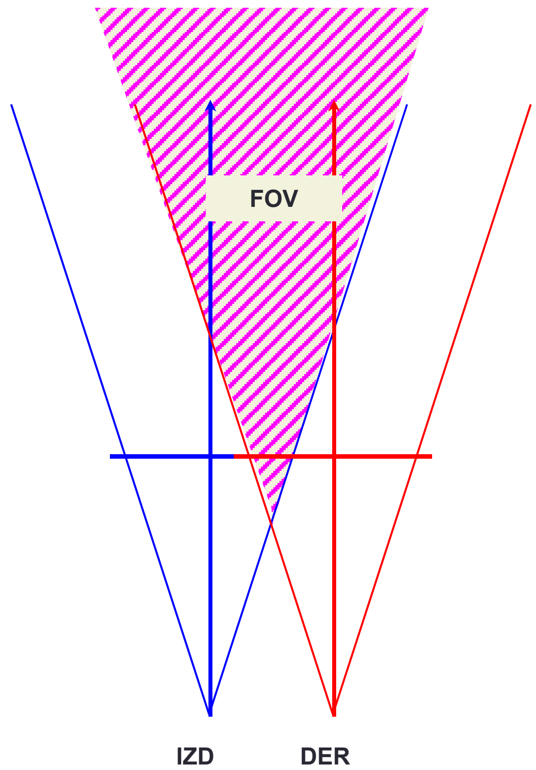
$$|\varepsilon_d| = 1 pixel$$



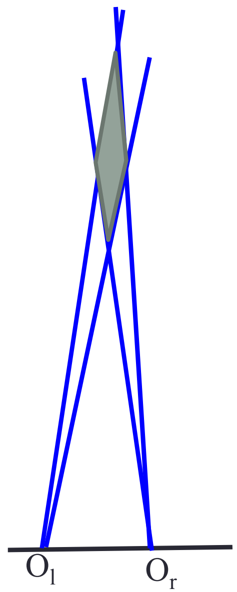
- If **Z** is **small → d** is **big,** we make a lower mistake in the measurement.
- By limiting the minimum disparity, we limit the maximum absolute error we make in 3D reconstruction.

Accuracy

- B small
  - Advantages:
    - Big common FOV
    - Increases the number of possible correspondences. Similar images



  - Disadvantages:
    - Increased error in estimating Z
    - Small disparity for corresponding points → low accuracy
    - Far points have disparity →0

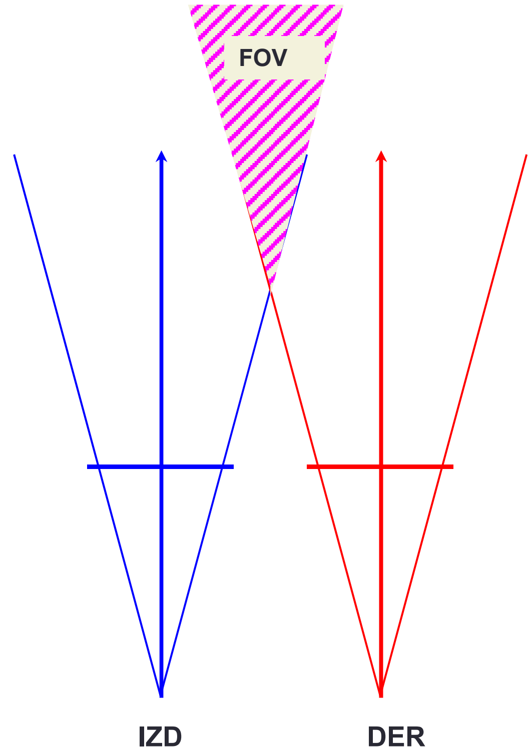


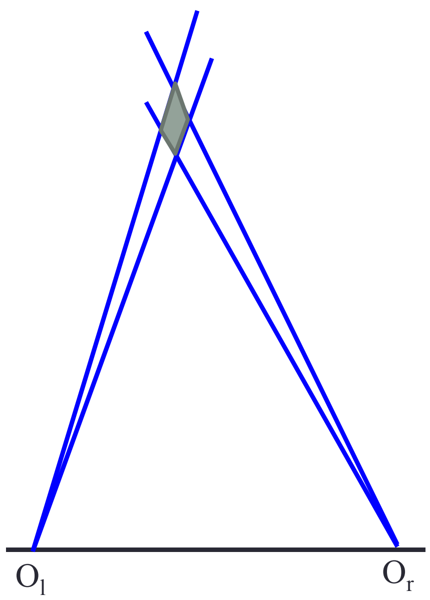
Accuracy

- B big
  - Advantages
    - Better accuracy in depth estimation.
  - Disadvantages
    - Small common FOV
    - Very different images

- Alternatives
  - If the environment is known, select B for the FOV to occupy the entire scene.
  - Convergent axes

- Watch out! FOV: it's not camera’s, we mean the common FOV.





Accuracy : Baseline (B)

- Reconstruction with two cameras with aligned axes

$$Z = \frac{f\ B}{x_2 - x_1} = \frac{f\ B}{d}$$

  - Data: K_x=30pixel/mm ; Z=2000mm f=20mm ; Detection error =±2pixel

$$\Rightarrow B = 100mm \qquad d = \frac{fB}{Z} = \frac{20\ \cdot\ 100}{2000} = 1.0\text{mm} \Rightarrow 30\text{pixeles}$$

$$\begin{cases} d = 28píxeles \quad \Rightarrow \quad d = 0.9333mm \ \ Z = \dfrac{fB}{d} = \dfrac{20 \cdot 100}{0.9333} = 2142.93 \\ d = 32píxeles \quad \Rightarrow \quad d = 1.0666mm \ \ Z = \dfrac{fB}{d} = \dfrac{20 \cdot 100}{1.0666} = 1875.11 \end{cases}$$

Detection Error Range: $2142.93 - 1875.11 = 267.82$

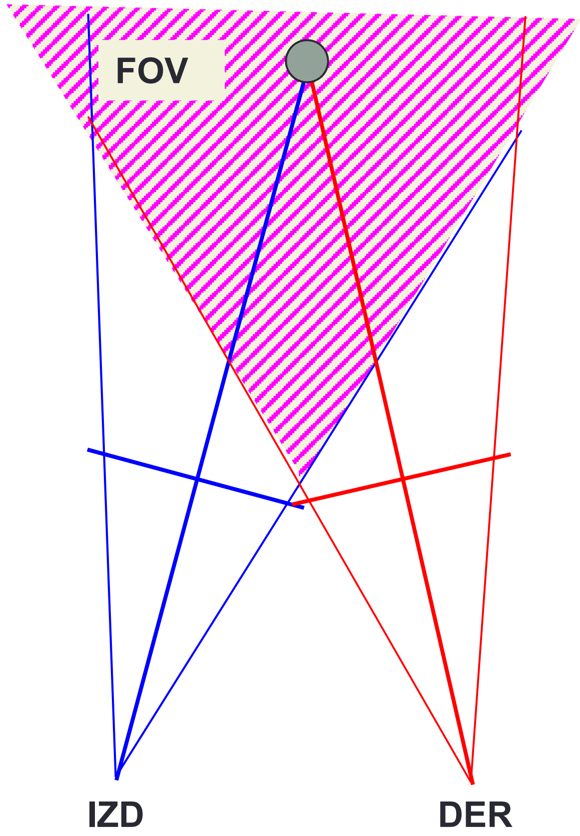
$$\Rightarrow B = 500mm \qquad d = \frac{fB}{Z} = \frac{20\ \cdot\ 500}{2000} = 5.0\text{mm} \Rightarrow 150\text{pixeles}$$

$$\begin{cases} d = 148píxeles \quad \Rightarrow \quad d = 4.9333mm \ \ Z = \dfrac{fB}{d} = \dfrac{20 \cdot 500}{4.9333} = 2027.04 \\ d = 152píxeles \quad \Rightarrow \quad d = 5.0666mm \ \ Z = \dfrac{fB}{d} = \dfrac{20 \cdot 500}{5.0666} = 1973.71 \end{cases}$$

Detection Error Range: $2027.04 - 1973.71 = 53.33$

Accuracy : Stereo with convergent axes

- Convergent axes:
  - FOV bigger.
  - Searching for correspondences on epipolar lines.
  - There is a finite Z distance area with zero disparity.



3D Reconstruction . Convergent Axes

- **Objective**: to know the 3D coordinates of a point in space:

$$(x_w, y_w, z_w) \equiv (X, Y, Z)$$

- **DATA**:
  - We assume known the projection of the point (X, Y, Z) on the image plane of the two cameras
  - From the lateral coordinates of the pixels we get the coordinates without distortion.

Left camera Pixel $\quad (x_f{}^l, y_f{}^l) \Rightarrow (x_n{}^l, y_n{}^l) \Rightarrow (x, y) \Rightarrow \tilde{m}_u$

Right camera Pixel $\quad (x_f{}^r, y_f{}^r) \Rightarrow (x_n{}^r, y_n{}^r) \Rightarrow (x', y') \Rightarrow \tilde{m}'_u$

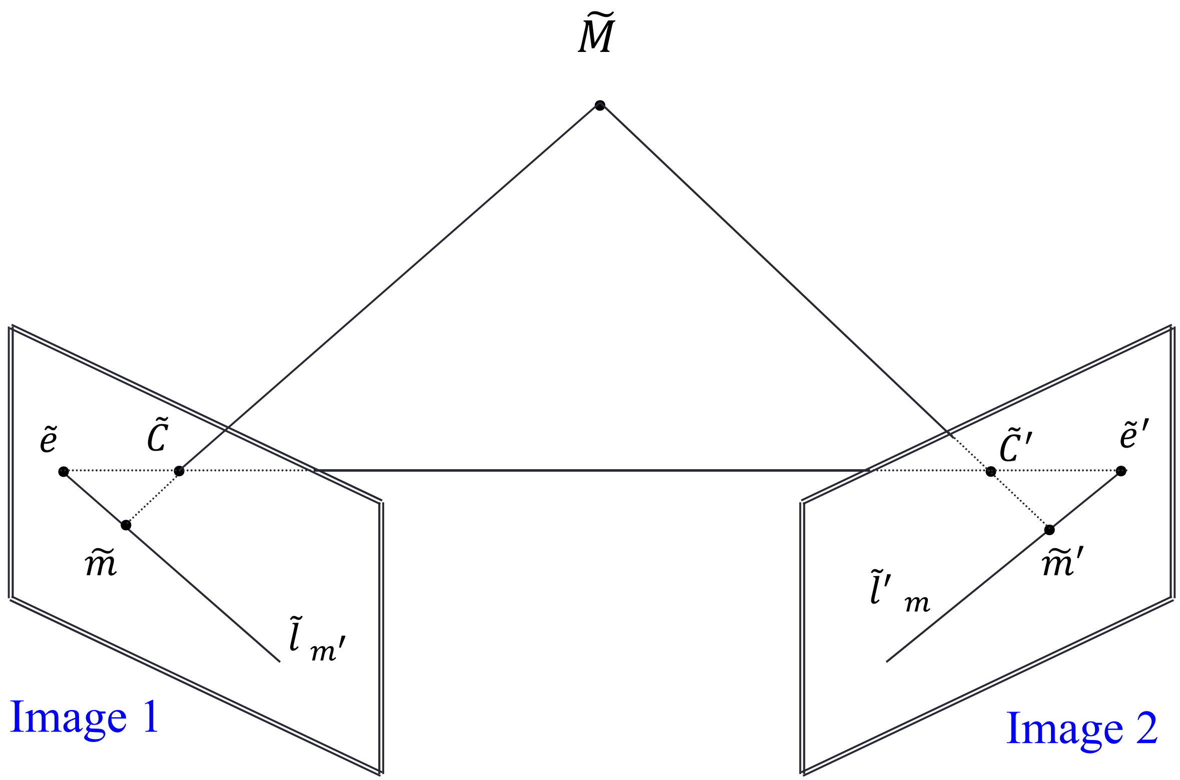
- **CALIBRATION DATA** : $\boxed{\tilde{m}_u \cong A\ [I \quad 0]\ \tilde{M}_c}$ $\quad \boxed{\tilde{m}'_u \cong A'\ [R \quad t]\ \tilde{M}_c}$

Distortion:

$$\begin{bmatrix} x_n^d \\ y_n^d \\ 1 \end{bmatrix} \cong A^{-1} \cdot \tilde{m} \rightarrow \left.\begin{matrix} x_n^d = x_n(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) + 2p_1 x_n y_n + p_2(2x_n^2 + r^2) + s_1 r^2 + s_2 r^4 \\ y_n^d = y_n\ (1 + k_1 r^2 + k_2 r^4 + k_3 r^6) + 2p_2 x_n y_n + p_1(2y_n^2 + r^2) + s_1 r^2 + s_2 r^4 \\ r^2 = x_n^2 + y_n^2 \end{matrix}\right\} \rightarrow \tilde{m}_u \cong A \cdot \begin{bmatrix} x_n \\ y_n \\ 1 \end{bmatrix}$$

3D Reconstruction. Convergent Axes

- Triangulación



Triangulation (Convergent Axes)

- Ray projection:

$$\tilde{P} \cong A\,[I \quad 0]$$

$$\tilde{P}' \cong A'[R \quad t]$$

$$\tilde{m}_i \cong \tilde{P}\ \tilde{M}_i \Rightarrow \begin{bmatrix} \lambda\ x \\ \lambda\ y \\ \lambda \end{bmatrix} = \begin{bmatrix} P_1^T \\ P_2^T \\ P_3^T \end{bmatrix} \tilde{M}_i$$

$$\tilde{m}'_i \cong \tilde{P}'\ \tilde{M}_i \Rightarrow \begin{bmatrix} \mu x' \\ \mu\ y' \\ \mu \end{bmatrix} = \begin{bmatrix} P'^T_1 \\ P'^T_2 \\ P'^T_3 \end{bmatrix} \tilde{M}_i$$

$$\begin{matrix} P_3^T \tilde{M}_i x = P_1^T\ \tilde{M}_i \\ P_3^T \tilde{M}_i y = P_2^T\ \tilde{M}_i \end{matrix} \Rightarrow \begin{bmatrix} P_3^T x - P_1^T \\ P_3^T y - P_2^T \end{bmatrix} \tilde{M}_i = 0$$

$$\begin{matrix} P'^T_3 \tilde{M}_i x' = P'^T_1\ \tilde{M}_i \\ P'^T_3 \tilde{M}_i y' = P'^T_2\ \tilde{M}_i \end{matrix} \Rightarrow \begin{bmatrix} P'^T_3 x - P'^T_1 \\ P'^T_3 y - P'^T_2 \end{bmatrix} \tilde{M}_i = 0$$

- Triangulation:

$$\begin{bmatrix} P_3^T x - P_1^T \\ P_3^T y - P_2^T \\ P'^T_3 x' - P'^T_1 \\ P'^T_3 y' - P'^T_2 \end{bmatrix} \tilde{M}_i = 0$$

$$A_{xy}\ \tilde{M}_i = 0$$

- Least squares estimation:

$$\min_{\tilde{M}} \left[\left(\left| A_{xy} \tilde{M} \right| \right)^2 \right]$$

$$SVD(A_{xy}) \rightarrow A_{xy} = UDV^T$$
$$\tilde{M} = v_n \qquad \left\| \tilde{M} \right\| = 1$$

- Nonlinear estimation: re-projection error

$$\min_{\tilde{M}_i} \left[\left(\tilde{m} - \lambda^{-1} P \tilde{M}_i \right)^2 + \left(\tilde{m}' - \lambda'^{-1} P' \tilde{M}_i \right)^2 \right]$$

Triangulation

- Triangulation (non-homogeneous solution - Euclidean):

$$\begin{cases}(p_{31}x - p_{11})X_w + (p_{32}x - p_{12})Y_w + (p_{33}x - p_{13})Z_w = p_{14} - p_{34}x \\ (p_{31}y - p_{21})X_w + (p_{32}y - p_{22})Y_w + (p_{33}y - p_{23})Z_w = p_{24} - p_{34}y\end{cases}$$

$$\begin{cases}(p'_{31}x' - p'_{11})X_w + (p'_{32}x' - p'_{12})Y_w + (p'_{33}x' - p'_{13})Z_w = p'_{14} - p'_{34}x' \\ (p'_{31}y' - p'_{21})X_w + (p'_{32}y' - p'_{22})Y_w + (p'_{33}y' - p'_{23})Z_w = p'_{24} - p'_{34}y'\end{cases}$$

$$A_{xy}M = b$$

$$A_{xy} = \begin{bmatrix}(p_{31}x - p_{11}) & (p_{32}x - p_{12}) & (p_{33}x - p_{13}) \\ (p_{31}y - p_{21}) & (p_{32}y - p_{22}) & (p_{33}y - p_{23}) \\ (p'_{31}x' - p'_{11}) & (p'_{32}x' - p'_{12}) & (p'_{33}x' - p'_{13}) \\ (p'_{31}y' - p'_{21}) & (p'_{32}y' - p'_{22}) & (p'_{33}y' - p'_{23})\end{bmatrix} \quad b = \begin{bmatrix}p_{14} - p_{34} \\ p_{24} - p_{34} \\ p'_{14} - p'_{34} \\ p'_{24} - p'_{34}\end{bmatrix}$$

- Least squares estimation (pseudo-inverse):

$$\boxed{M = \left(A_{xy}^T A_{xy}\right)^{-1} A_{xy}^T \cdot b}$$

Comparison

- **ALIGNED AXES**
  - Advantage: Simplicity of calculation
  - Drawbacks:
    - Construction difficulty
    - If B is high, both images are very different, with possible bad correspondences
    - If B is low, poor accuracy

- **CONVERGENT AXES**
  - Advantages:
    - Ease of construction
    - Simple correspondence if the disparity is low
  - Drawbacks :
    - Calculation complexity
    - If B is high, correspondence is difficult, occlusions
    - If B is low, poor accuracy

Table of Contents

Objectives

- To what extent we can get information about the location of objects in the scene and the cameras from two (not calibrated) views
- Data: correspondences between points projected into two image planes

$$\tilde{m}_i \cong \tilde{P}\ \tilde{M}_i \cong [P \quad p] \begin{bmatrix} M_i \\ 1 \end{bmatrix} \qquad \tilde{m}'_i \cong \tilde{P}'\ \tilde{M}_i \cong [P' \quad p'] \begin{bmatrix} M_i \\ 1 \end{bmatrix}$$

- Fundamental Matrix: $\boxed{\tilde{m}'^T F \tilde{m} = 0}$

- Essential Matrix: $\tilde{m} \cong A\tilde{q} \qquad \tilde{m}' \cong A'\tilde{q}'$

$$\boxed{\tilde{q}'^T\ E\ \tilde{q} = 0}$$

General scheme

- Calculate the Fundamental/Essential matrix from the point correspondneces (Projective Calibration)

$$\tilde{m}_{\mathrm{i}} \leftrightarrow \tilde{m}_{\mathrm{i}}'$$

- Calculate the projective matrices of each camera from the matrix Fundamental or Essential matrix (in this later case we know the intrinsic parameters)

- Triangulation: for each correspondence $\tilde{m}_{\mathrm{i}} \leftrightarrow \tilde{m}_{\mathrm{i}}'$ calculate the point in the projective space that is projected at these two points.

Matrix of each camera P, P'

- From the Fundamental matrix (***F***):

$$F \cong A'^{-T}[t]_\wedge RA^{-1}$$

$$F \cong [e']_\wedge A' RA^{-1} \cong A'^{-T} RA^{T}[e]_\wedge$$

  - Both camera matrices are determined except for a projective transformation of 3D space
  - One solution is (canonical camera):

$$\tilde{P} \cong [I \quad | \, 0] \qquad \tilde{P}' \cong [[\tilde{e}'] \wedge F \quad | \, \tilde{e}']$$

$$\tilde{F}^T \cdot \tilde{e}' = 0$$

  - P' is singular => optical center in the plane at infinity

Stereo projective model:

$$\left.\begin{array}{r} \tilde{P} \cong A\,[\,I \quad |\,0] \\ \tilde{P}' \cong A'[\,R \quad |\,t] \\ \tilde{e}' \cong A' \quad t \end{array}\right\} \Rightarrow \left.\begin{array}{r} \tilde{P} \cong [\,I \quad |\,0] \\ \tilde{P}' \cong [\,A'RA^{-1} \quad |\,A't] \end{array}\right\} \Rightarrow \left.\begin{array}{r} \tilde{P} \cong [\,I \quad |\,0] \\ \tilde{P}' \cong [\,H_\infty \quad |\,\lambda\tilde{e}'] \end{array}\right\}$$

"Multiple View Geometry in Computer Vision" R. Hartley and A. Zisserman Cambridge University Press, March 2004.

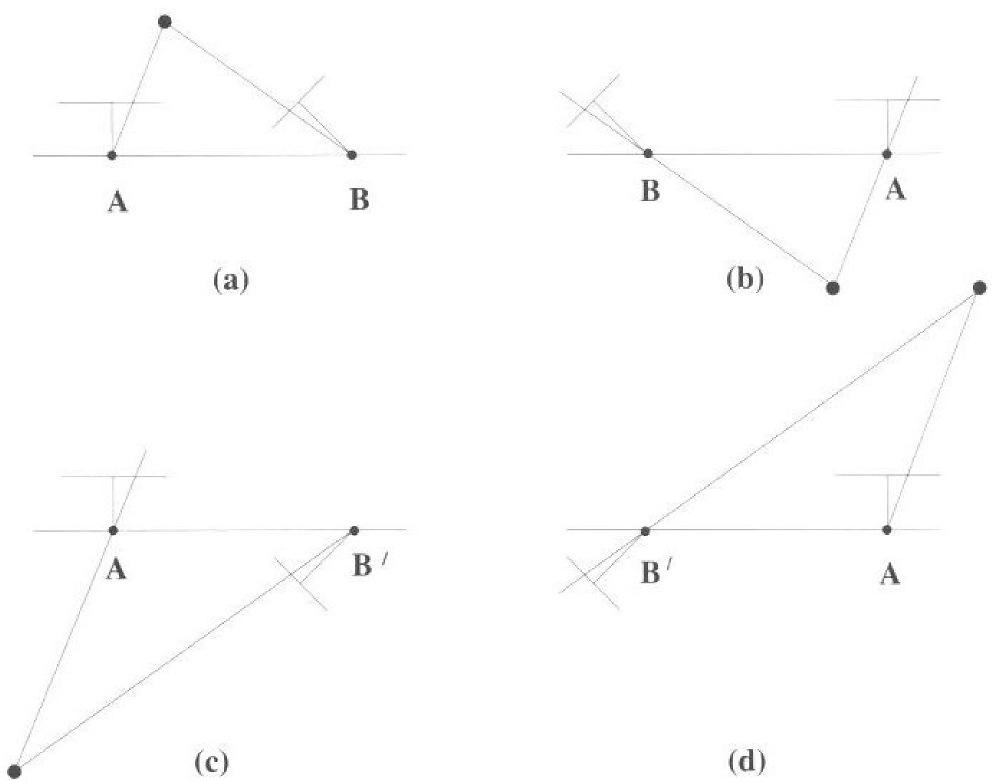
Matrix of each camera P, P'

- From the Essential Matrix (***E***): $E = [t]_\wedge R$
  - Both camera matrices are determined except for a scale factor and an ambiguity of four possible configurations:

$$\tilde{P} = [\, I \quad | \; 0]$$
$$\tilde{P}' = [\, R \quad | \; t]$$

$$\tilde{P}' = \left[\, UWV^T \quad \middle| \lambda \cdot u_3 \right]$$
$$\tilde{P}' = \left[\, UW^TV^T \quad \middle| \lambda \cdot u_3 \right]$$
$$\tilde{P}' = \left[\, UWV^T \quad \middle| -\lambda \cdot u_3 \right]$$
$$\tilde{P}' = \left[\, UW^TV^T \quad \middle| -\lambda \cdot u_3 \right]$$

$$SVD\,(\,E) \rightarrow U \cdot D \cdot V^T \qquad W = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$



"A computer algorithm for reconstructing a scene from two projections"
H. Christopher Longuet-Higgins Nature. 293 (1981)

Triangulation

- Disambiguation from the Essential Matrix:
  - 3D points are reconstructed for the 4 possible configurations
  - We keep the configuration where the points are in front of the camera (Z>0)

- Triangulation:

$$\begin{bmatrix} P_3^T x - P_1^T \\ P_3^T y - P_2^T \\ P'^T_3 x' - P'^T_1 \\ P'^T_3 y' - P'^T_2 \end{bmatrix} \tilde{M}_i = 0$$

- Least squares estimation:

$$\min_{\tilde{M}} \left[\left(\left| A_{xy} \tilde{M} \right| \right)^2 \right]$$

$$SVD(A_{xy}) \rightarrow A_{xy} = UDV^T$$

$$\tilde{M} = v_n \quad \left\| \tilde{M} \right\| = 1$$

"A computer algorithm for reconstructing a scene from two projections"
H. Christopher Longuet-Higgins Nature. 293 (1981)

Ambiguity of the Reconstruction

- In general, it is not possible to reconstruct the absolute position and orientation referred to a Coordinates System, without additional information about the scene.

- For non-calibrated cameras:
  - The scene is determined except for a projective transformation $\widetilde{H}$

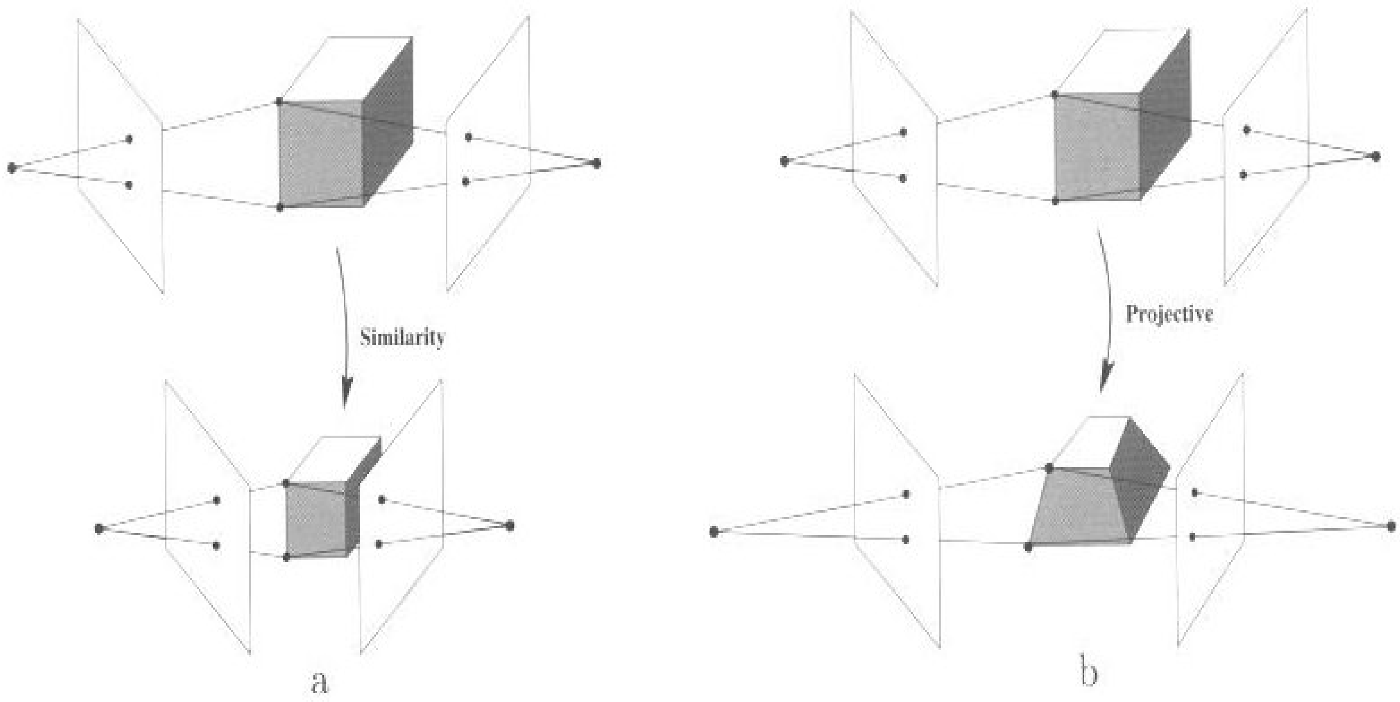
- For a pair of calibrated cameras (*A, A'*):
  - Scene is determined except for a similarity transformation (Rotation, Translation, and Scale)

$$\widetilde{H}_S = \begin{bmatrix} R & t \\ 0 & \lambda \end{bmatrix}$$

$$\text{Interchanging} \quad \widetilde{M}_i \to \widetilde{H}_S \widetilde{M}_i \ , \quad \tilde{P} \to \tilde{P}\widetilde{H}_S^{-1}, \quad \tilde{P}' \to \tilde{P}'\widetilde{H}_S^{-1}$$

$$\widetilde{m}_i \cong \tilde{P}\widetilde{M}_i \cong (\tilde{P}\widetilde{H}_S^{-1})(\widetilde{H}_S\widetilde{M}_i), \quad \widetilde{m}_i{}' \cong \tilde{P}\widetilde{M}_i{}' \cong (\tilde{P}'\widetilde{H}_S^{-1})(\widetilde{H}_S\widetilde{M}_i{}')$$

Ambiguity of the Reconstruction



Calibrated Cameras

Non-calibrated Cameras

"Multiple View Geometry in Computer Vision" R. Hartley and A. Zisserman Cambridge University Press, March 2004.

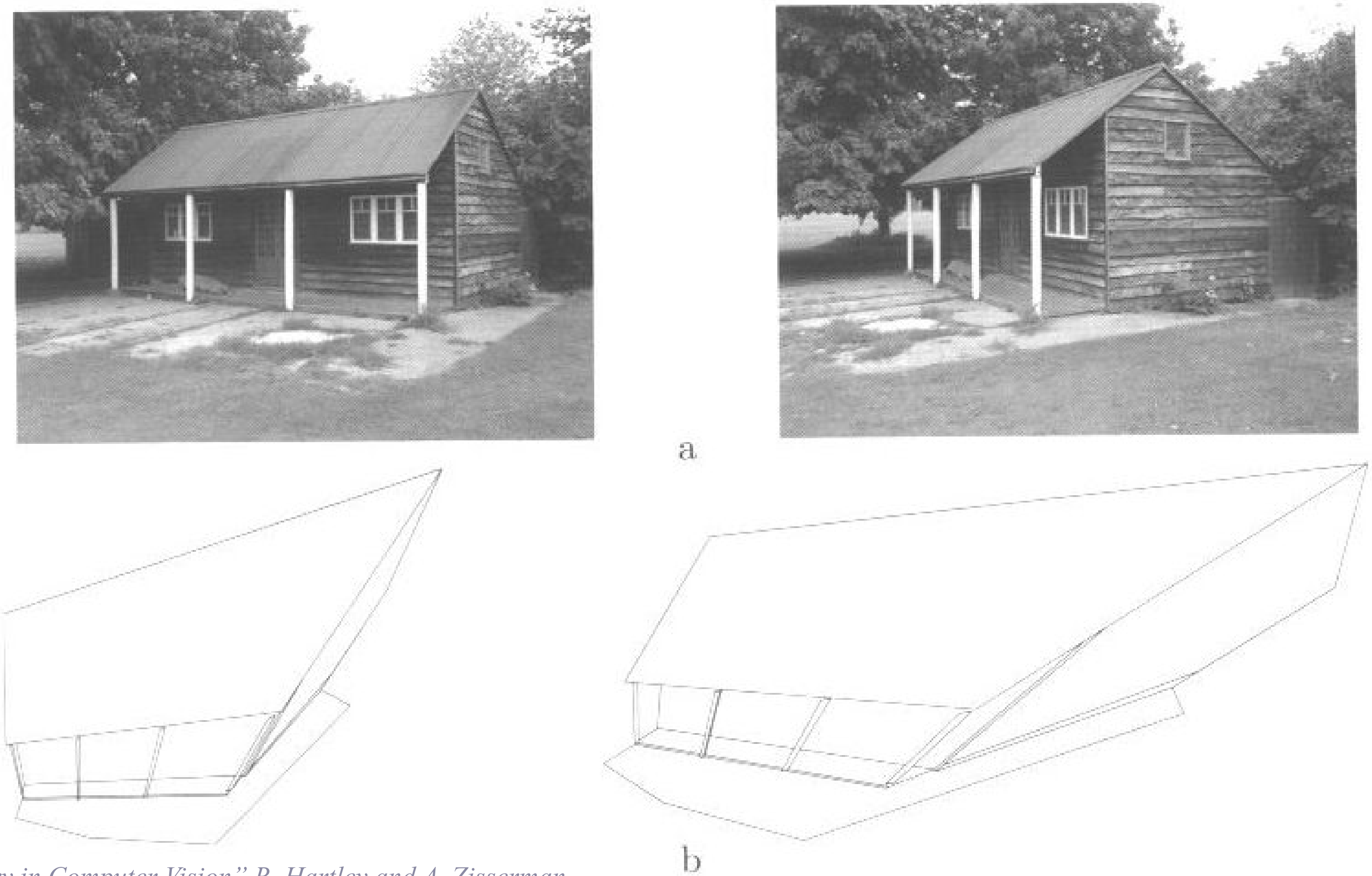
Projective Reconstruction

- Advantages:
  - Allows to project a scene without information about calibration or position of the two cameras.
  - The True reconstruction (Euclidea) is related with some projective transformation from the previous one.

$$\left(\tilde{P}, \tilde{P}', \{\tilde{M}_i\}\right) \quad ; \quad \left(\tilde{P}_E, \tilde{P}_E', \{\tilde{M}_{Ei}\}\right)$$

$$\tilde{P}_E = \tilde{P}\tilde{H}^{-1}, \quad \tilde{P}_E' = \tilde{P}'\tilde{H}^{-1}, \quad \tilde{M}_{Ei} = \tilde{H}\tilde{M}_i$$

Projective Reconstruction



"Multiple View Geometry in Computer Vision" R. Hartley and A. Zisserman Cambridge University Press, March 2004.

Projective Reconstruction

- Applications:
  - **Parallax**: Robot navigation, collision distance
  - **Homography between views**: 3D modeling from views

Projective Parallax

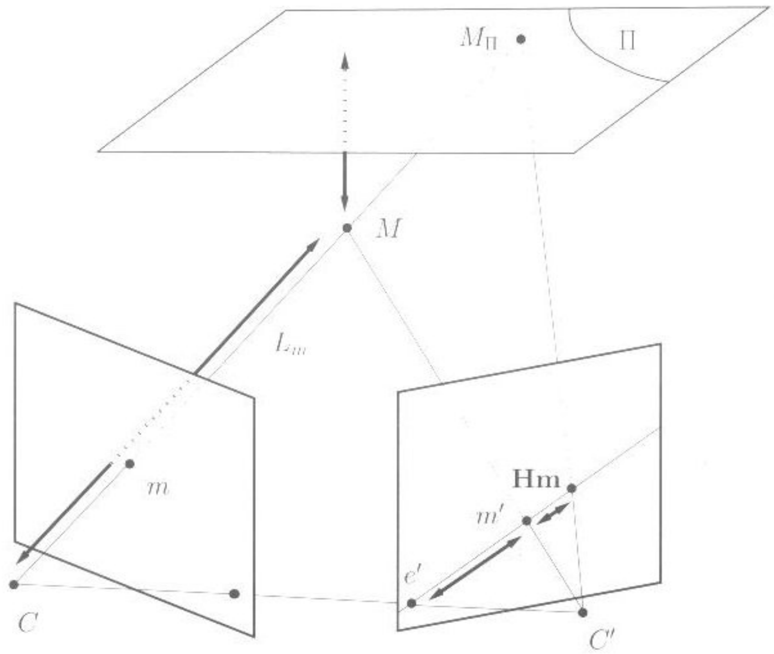
- **Parallax**: distance to a plane (obstacle)

$$\tilde{m} \cong [I_3 \quad | 0_3]\tilde{M} \quad \Rightarrow \quad \tilde{M} \cong \begin{bmatrix} \tilde{m} \\ \rho \end{bmatrix}$$

$$\tilde{m}' \cong [H \quad | \mu \ e']\tilde{M} \quad \Rightarrow \tilde{m}' \cong H\tilde{m} + \mu\rho \ e'$$

$$\tilde{m}' \cong H\tilde{m} + \kappa \ e'$$

$$\boxed{[\tilde{m}']_{\wedge} H\tilde{m} + \kappa \ [\tilde{m}']_{\wedge} e' = 0}$$



- ***k***: ratio of distances from ***m'*** a ***H$_m$*** y ***e'***

"Multiple View Geometry in Computer Vision" R. Hartley and A. Zisserman
Cambridge University Press, March 2004.

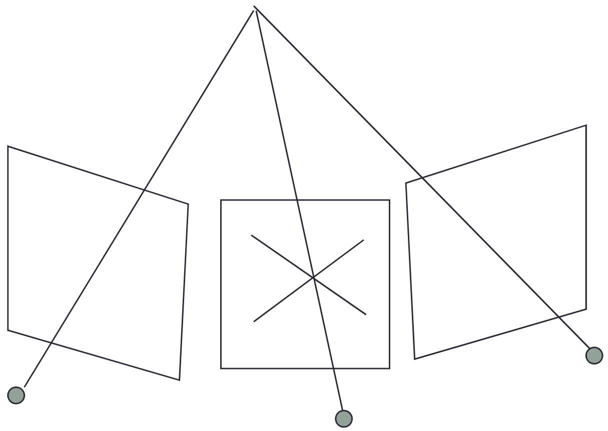
Projective Parallax

- Parallax:
  - Directly proportional to the distance to the plane
  - Inversely proportional to depth
  -
- Data: F, H
- Change of sign (zero-crossing) of ***k*** indicates whether the point is in front of or behind the plane
  - Projective transformations do not preserve order in depth

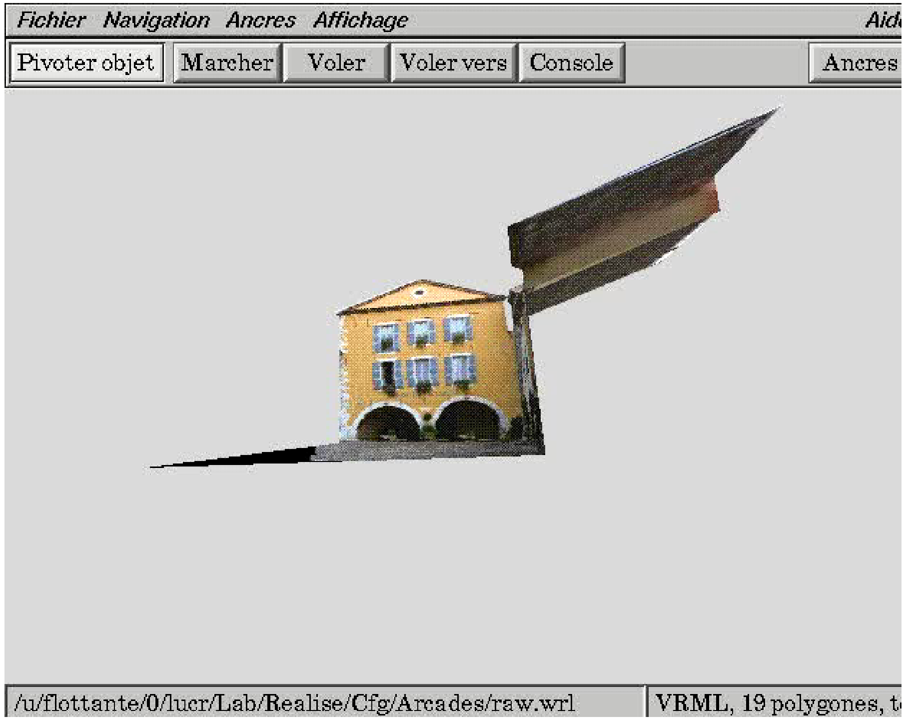
Homography between views

- Allows to get the view from another point without a 3D model
  - Just define a new camera matrix ***P''*** and calculate the two fundamental matrices (***F_{12}***, ***F_{13}***) -> transfer matrices
  - The new image is calculated as an intersection of the epipolar lines

$$\tilde{m}'' \cong F_{12}\tilde{m} \wedge F_{23}\tilde{m}'$$



Homography between views



Video

INRIA - Shopia Antipolis

Euclidean Reconstruction

- Direct Reconstruction: Projective -> Euclidean
  - If we have at least 5 control points whose 3D position is known in the Euclidean space, we can calculate the projective transformation H_m directly from any projective reconstruction

$$\tilde{m}_i \leftrightarrow \tilde{m}'_i$$

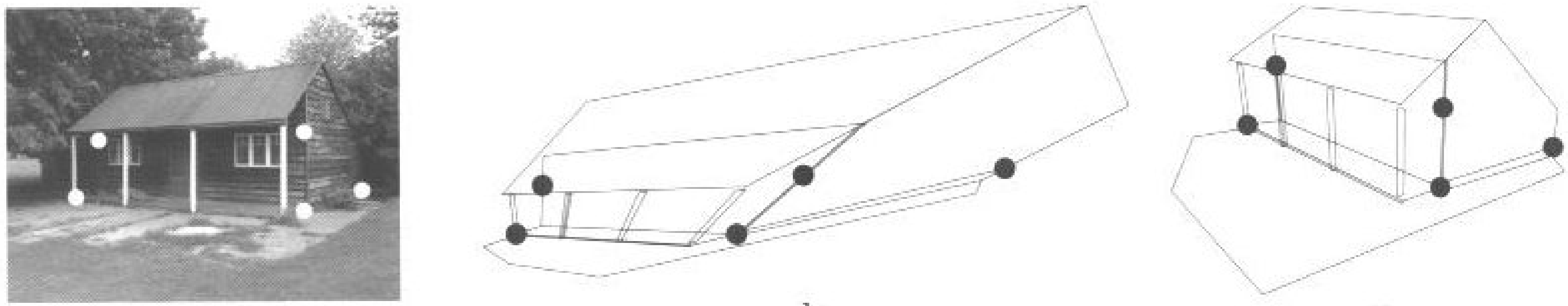
$$\left(\tilde{P}, \tilde{P}', \{\tilde{M}_i\}\right)$$

$$\{\tilde{M}_{Ei}\}$$

1) $\tilde{M}_{Ei} = H_m \, \tilde{M}_i$

2) $\tilde{m}_{Ei} = \tilde{P} {H_m}^{-1} \, \tilde{M}_{E\,i}$

$\tilde{m}'_{Ei} = \tilde{P}' {H_m}^{-1} \, \tilde{M}_{E\,i}$



a b c

"Multiple View Geometry in Computer Vision" R. Hartley and A. Zisserman Cambridge University Press, March 2004.

Table of Contents

Visual Odometry

- Visual Odometry: incremental location using a camera (calibrated) using multiple points of view:
  - Mobile robot, drone, ….
- SLAM: Simultaneous Location and Mapping
  - Incorporates the estimation of the map
- Data: correspondences between points projected on multiple image planes:

$$\tilde{m}_i^{(1)} \leftrightarrow \tilde{m}_i^{(k)}$$

$$\tilde{m}_i^{(1)} \cong \tilde{P}_{(1)}\, \tilde{M}_i \cong A\, [I \quad 0] \begin{bmatrix} M_i \\ 1 \end{bmatrix}$$

$$\tilde{m}_i^{(k)} \cong \tilde{P}_{(k)}\, \tilde{M}_i \cong A\, [R_{(k)} \quad t_{(k)}] \begin{bmatrix} M_i \\ 1 \end{bmatrix}$$

$$\boxed{\tilde{m}^{(k)^T} F\ \tilde{m}^{(1)} = 0}$$

$$\boxed{\tilde{q}^{(k)^T}\ E\ \ \tilde{q}^{(1)} = 0}$$

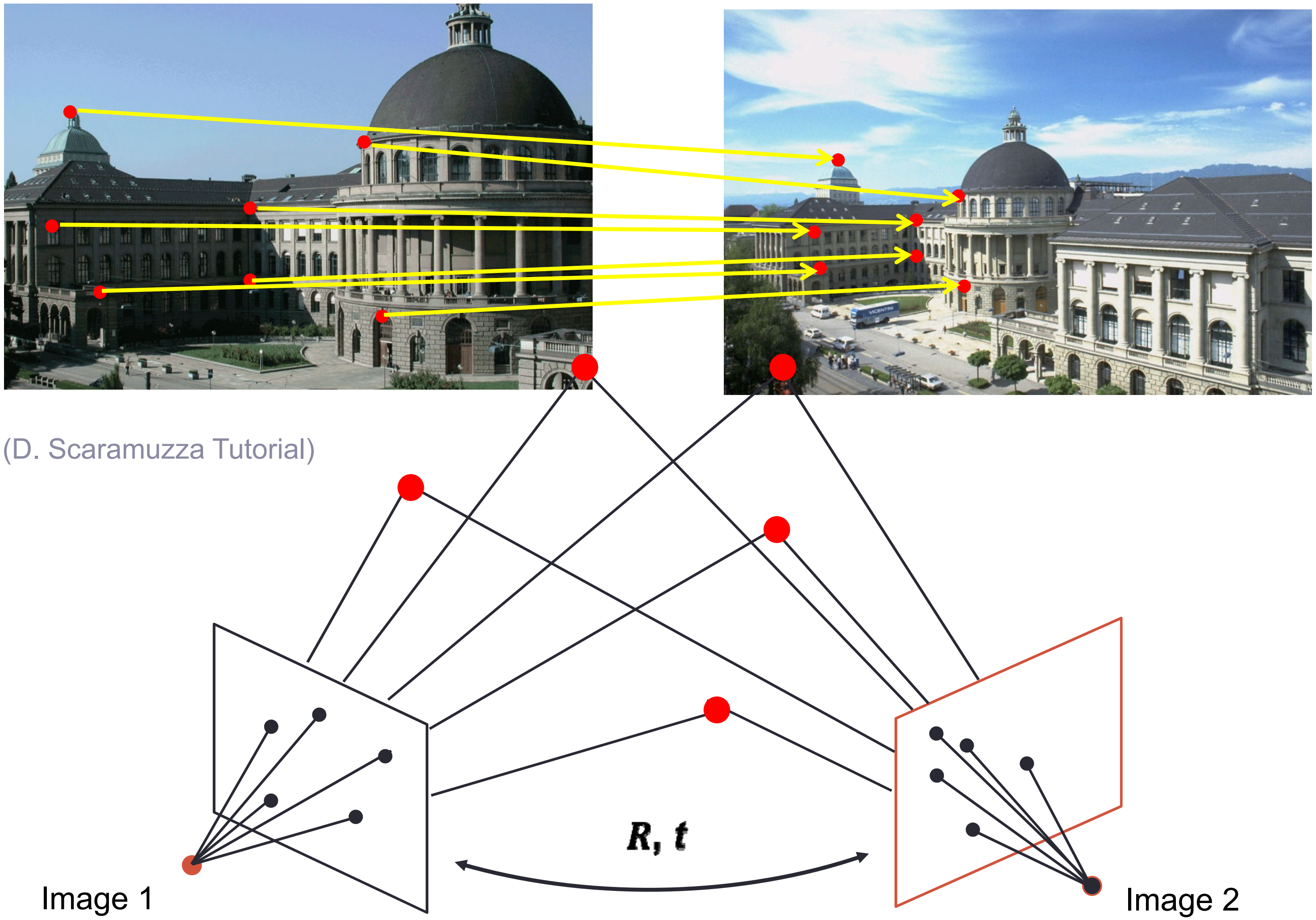
$$\boxed{E \cong A^T F\ A \cong \left[t_{(k)}\right]_{[\wedge]} \cdot R_{(k)}}$$

$R_{(k)}$ $t_{(k)}$ Rotation and translation (direction) between views (1) and (k)

"Visual Odometry: Part I - The First 30 Years and Fundamentals" Davide Scaramuzza
IEEE Robotics and Automation Magazine, Volume 18, issue 4, 2011 http://rpg.ifi.uzh.ch/docs/VO_Part_I_Scaramuzza.pdf

"Visual odometry: Part II - Matching, robustness, optimization, and applications" Davide Scaramuzza
IEEE Robotics and Automation Magazine, Volume 19, issue 2, 2012 http://rpg.ifi.uzh.ch/docs/VO_Part_II_Scaramuzza.pdf

Visual Odometry



(D. Scaramuzza Tutorial)

Visual Odometry

- Visual Odometry: incremental location using a camera (calibrated) using multiple points of view:
  - Mobile robot, drone, ….

Video

(Dyson Robotics 360, Andrew Davison, Imperial College, London)

Visual Odometry

- Visual Odometry: incremental location using a camera (calibrated) using multiple points of view:
  - Incremental calculation: $\tilde{m}_i^{(k)} \leftrightarrow \tilde{m}_i^{(k+1)}$

$$\tilde{m}_i^{(k)} \cong A\ [R_{(k)} \quad t_{(k)}] \begin{bmatrix} M_i \\ 1 \end{bmatrix} \qquad \tilde{m}_i^{(k+1)} \cong A\ [R_{(k+1)} \quad t_{(k+1)}] \begin{bmatrix} M_i \\ 1 \end{bmatrix}$$

$${}^{(k)}R_{(k+1)} \quad {}^{(k)}t_{(k+1)} \quad \text{Rotation and translation (direction) between views(k) y (k + 1)}$$

$$\tilde{m}^{(k+1)^T} F\ \tilde{m}^{(k)} = 0 \qquad \boxed{E \cong A^T F\ A \cong \left[{}^{(k)}t_{(k+1)}\right]_{[\wedge]} \cdot {}^{(k)}R_{(k+1)}}$$

  - Odometry update:

$$\boxed{\begin{aligned} R_{(k+1)} &= R_{(k)} \cdot {}^{(k)}R_{(k+1)} \\ t_{(k+1)} &= t_{(k)} + R_{(k)} \cdot {}^{(k)}t_{(k+1)} \end{aligned}}$$

    - Translation (scale) is calculated from the reconstructed 3D points distances (map); $\{M_i\}^k \leftrightarrow \{M_i\}^{k+1} \Rightarrow {}^{(k)}t_{(k+1)}$

Visual Odometry: General scheme

- Algorithm:
  - Calculate the Fundamental/Essential matrix from the point correspondences (Projective Calibration) (If we move on a plane you can use the 3-point algorithm):

$$\tilde{m}_i^{(k)} \leftrightarrow \tilde{m}_i^{(k+1)} \qquad \tilde{m}^{(k+1)^T} F \; \tilde{m}^{(k)} = 0 \qquad \tilde{q}^{(k+1)^T} \; E \;\; \tilde{q}^{(k)} = 0 \qquad \tilde{q} = A^{-1}\tilde{m} \qquad E = A^T F A$$

  - Decomposition of the Essential matrix into an anti-symmetrical matrix and a rotation matrix (4 configurations): $E \cong [t]_{[\wedge]} \cdot R$
  - 3D reconstruction of the points (map) for the four configurations : $\boxed{\{\; M_i \;\}}$
  - We keep the one where the 3D points are in front of the camera (Z>0). Rotation and translation direction: $\boxed{{}^{(k)}R_{(k+1)} \;; \quad \lambda \cdot {}^{(k)}t_{(k+1)}}$
  - Odometry update: $\boxed{R_{(k+1)} = R_{(k)} \cdot {}^{(k)}R_{(k+1)}}$
  - Translation is calculated from the reconstructed 3D points (PnP) $\{M_i\}^k \leftrightarrow \{M_i\}^{k+i} \;\; \Rightarrow \;\; {}^{(k)}t_{(k+i)}$
    - It is calculated between two positions where the translation (baseline) is large to reduce the estimation error

$$\boxed{t_{(k+i)} = t_{(k)} + R_{(k)} \cdot {}^{(k)}t_{(k+i)}}$$

Visual Odometry

- Projective uncertainty (scale factor)
  - From two views we cannot extract the translation distance, only the direction
  - The map and location are defined up to a scale factor (Similarity Transformation))
    - A scene with a large object at a great distance and a large camera translation, produces the same projected points as a scene with a small object nearby and a small camera translation
  - For successive views we can calculate the scale factor in the next step, such as the ratio of distances between points on the 3D map generated from the previous one.
  - The initial scale factor is estimated:
    - From metric image information: a known size marker
    - An additional sensor: IMU, accelerometer

Visual Odometry

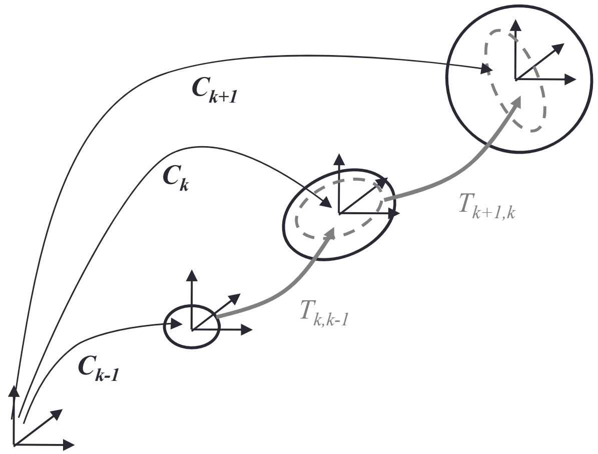
- Projective uncertainty (scale factor)

Video

(Habitación de Ames)

Visual Odometry

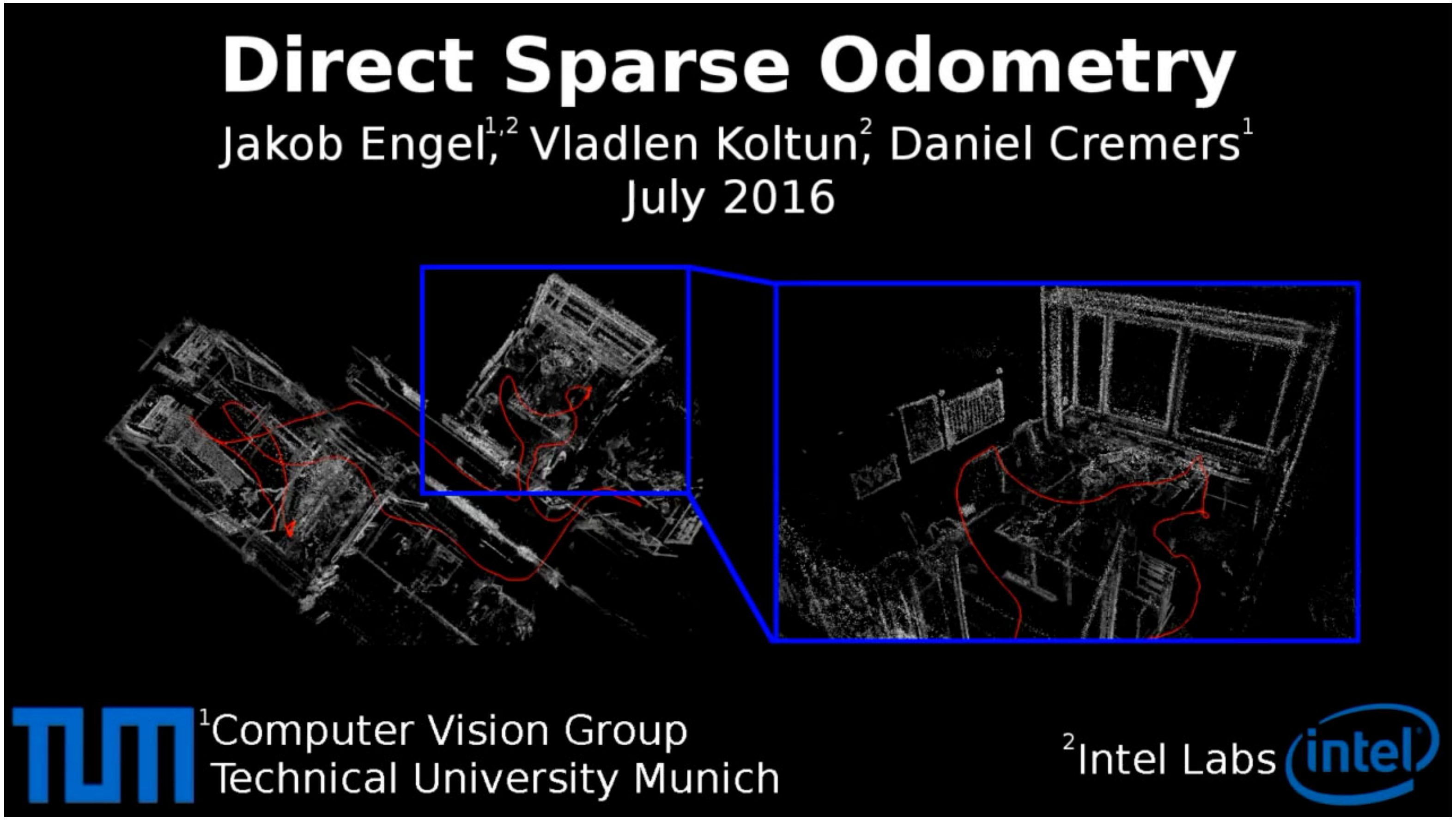
- Implementation issues:
  - In order to obtain a robust 3D reconstruction (map), the distance between views must be high, but this makes point correspondence difficult
  - SLAM: Key-frames are used:
    - Tracking and odometric calculation is performed for each captured image
    - The map is only updated when the distance between views is high enough
  - Error Propagation:
    - The error estimation accumulates which will cause the odometry to be incorrect over time
    - Kalman Filter (EKF), Particle Filter



(D. Scaramuzza Tutorial)

Visual Odometry

- Example:



Video

Jakob Engel, Vladlen Koltun, Daniel Cremers
July 2016 Univ. Munich

Table of Contents