Stereo Camera Calibration

Overview

- Introduction
- Epipolar Geometry
- Calibration Methods
- Summary / Further Readings

Motivation

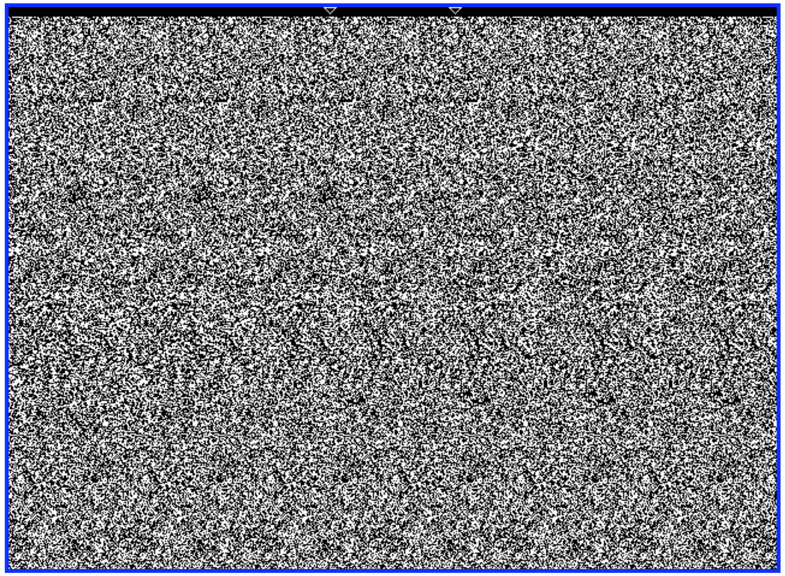
- 3D-Vision Problems
  - Correspondence problem
  - Reconstruction problem

Motivation

- 3D-Vision Problems
  - Correspondence problem
  - Reconstruction problem

How does depth perception work?

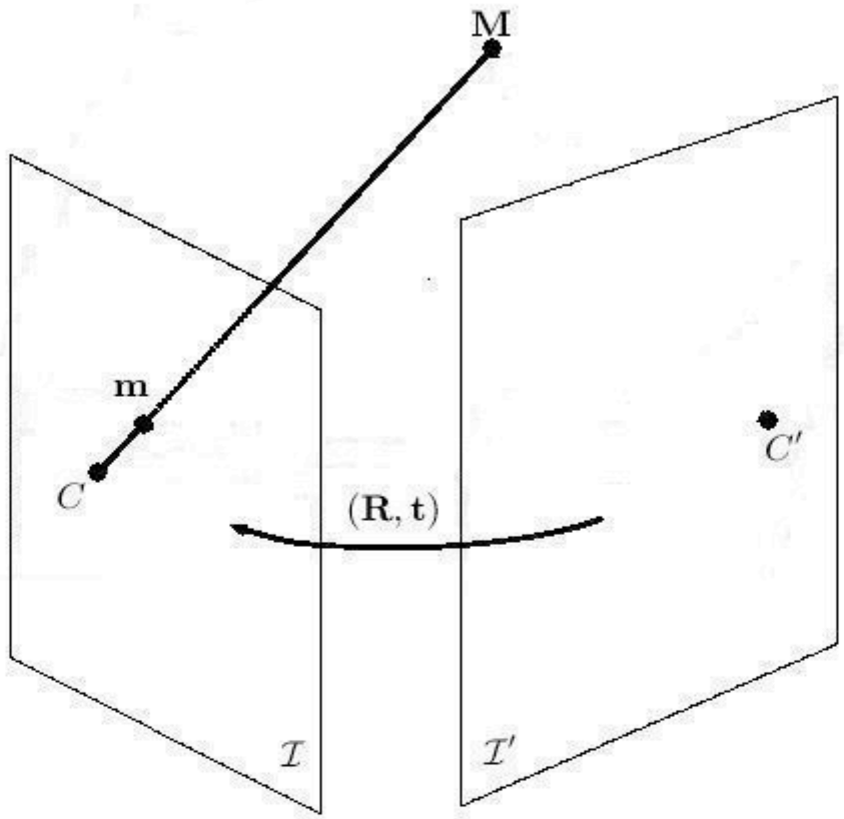
- High level process?
  - Objects are matched...
- Low level process!
  - Proof: Random Dot Images
  - Problem: Matching the Dots



The Ambiguity of Correspondence

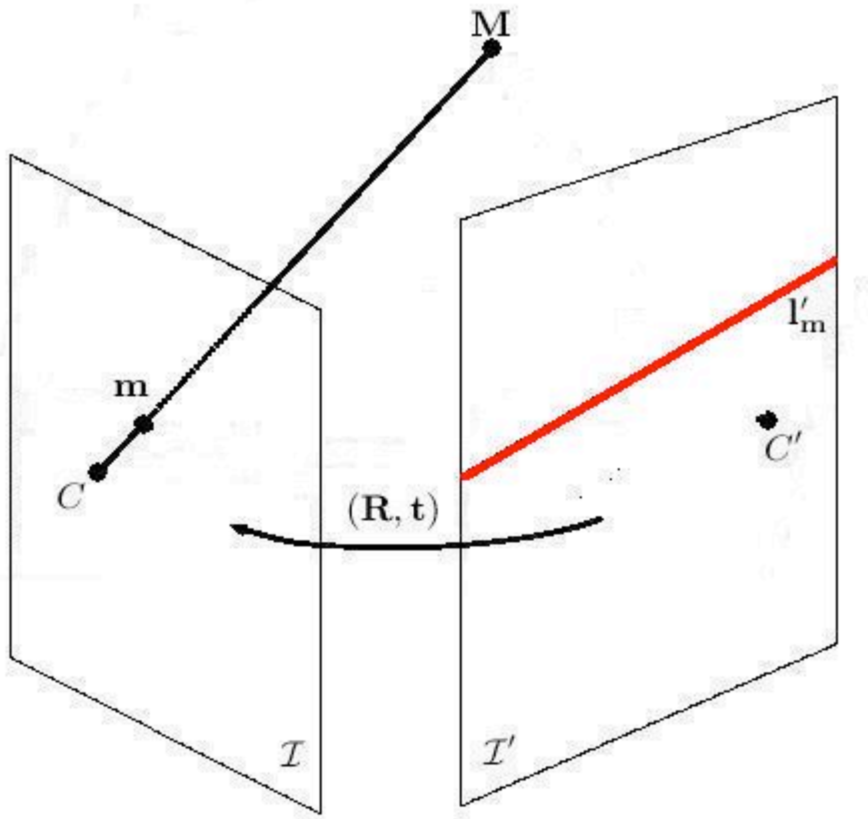
- Real Correspondence
- Just looking similar

The Epipolar Constraint



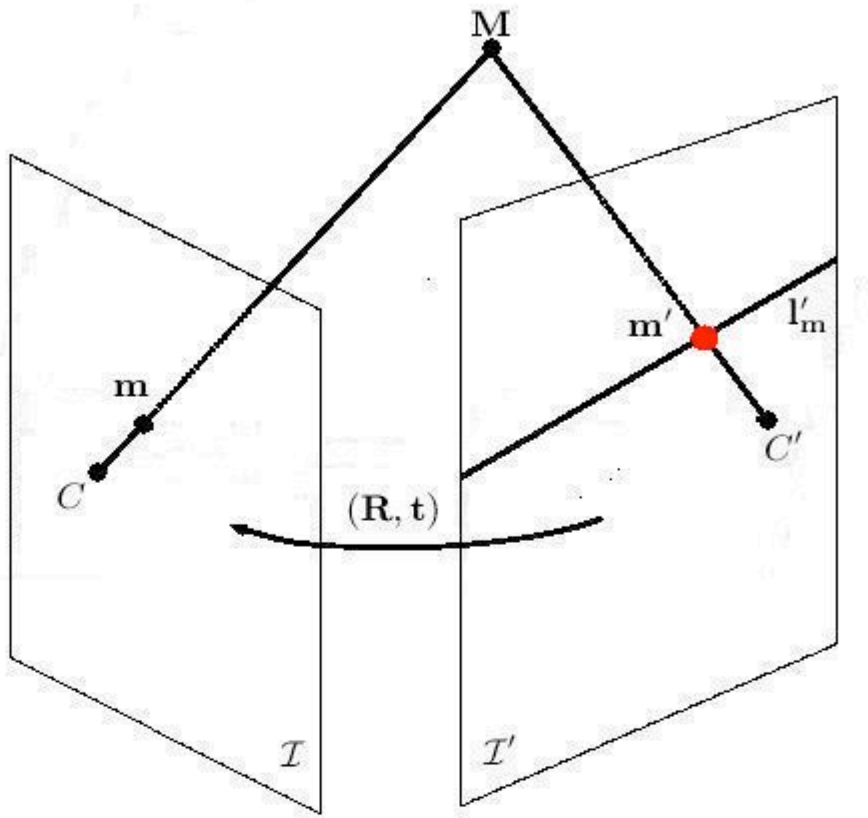
World Point M corresponds to point m in Image Plane I

The Epipolar Constraint



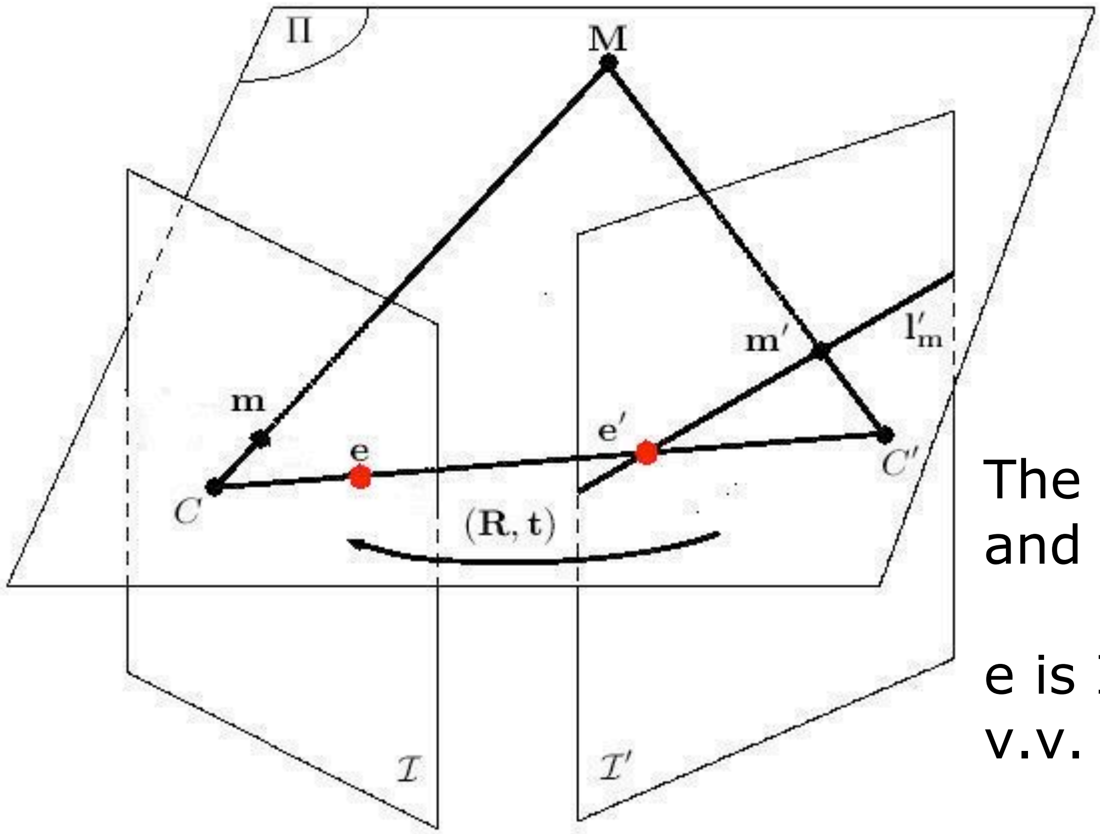
Ray CM has Image in I'

The Epipolar Constraint



Corresponding Point in I' can be found...

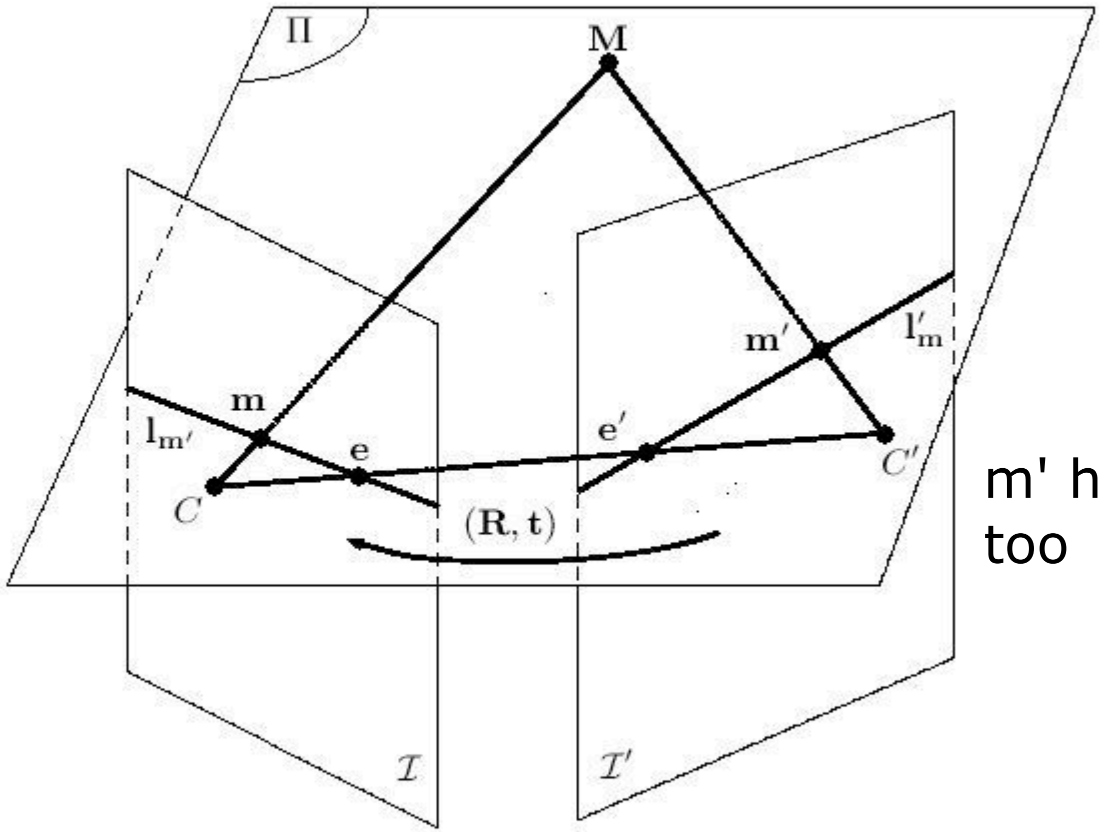
The Epipolar Constraint



The Points C, C' and M build a Plane.

e is Image of C' and v.v.

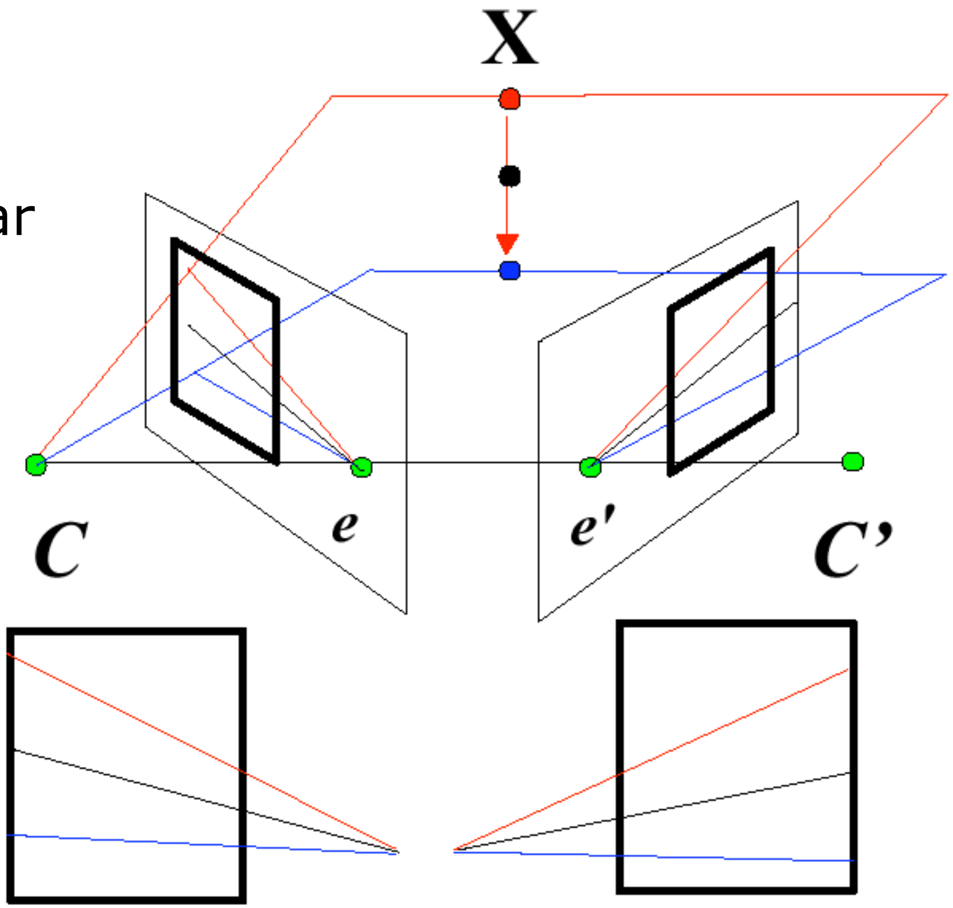
The Epipolar Constraint



m' has epipolar line too

Pencils of Epipolar Lines

Different World Points can have different epipolar lines.



Perspective Projection Matrix

- There are normalized and unnormalized cameras

$$\mathbf{P} = \begin{bmatrix} fk_u & fk_u \cot\theta & u_0 & 0 \\ 0 & fk_v / \sin\theta & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

Decomposition possible: $\mathbf{P} = \mathbf{A}\mathbf{P}_N$

Camera Intrinsic Matrix

- Result of decomposition $\mathbf{P} = \mathbf{A}\mathbf{P}_N$

$$\mathbf{A} = \begin{bmatrix} \alpha_u & \alpha_u \cot\theta & u_0 & 0 \\ 0 & \alpha_v / \sin\theta & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

$$\mathbf{P}_N = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

$$\tilde{m}_n = \mathbf{A}^{-1}\tilde{m}$$

Skew Symmetric Matrix

- Mapping from 3D-Vector to 3x3 antisymmetric Matrix

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}_x = \begin{bmatrix} 0 & -x_3 & x_2 \\ x_3 & 0 & -x_1 \\ -x_2 & x_1 & 0 \end{bmatrix}$$

$$\det([\cdot]_x) = 0 \qquad [t]_x = -[t]_x^T$$

$$\boxed{\vec{a} \times \vec{b} = [\vec{a}]_x \vec{b}}$$

The Essential Matrix

- Mapping from Point to Line

$$l_{m'} = \mathbf{E}\tilde{m}'$$

$\mathbf{E}$ is Essential Matrix

$l_{m'}$ is epipolar Line for $\tilde{m}'$

Defining EG with Normalized Images

- Relation between camera coordinate systems

$$\vec{M} = \mathbf{R}\vec{M}' + \vec{t}$$

- Normalized Coordinates

$$\tilde{m} = \frac{\vec{M}}{Z} \qquad \tilde{m}' = \frac{\vec{M}'}{Z'}$$

Defining EG with Normalized Images

$$
\begin{aligned}
\tilde{m} &= \frac{1}{Z}\left(Z'\mathbf{R}\tilde{m}' + \vec{t}\right) \\
\vec{t} \times \tilde{m} &= \frac{Z'}{Z}\vec{t} \times \mathbf{R}\tilde{m}' \\
\tilde{m} \bullet \vec{t} \times \tilde{m} &= \tilde{m} \bullet \vec{t} \times \left(\mathbf{R}\tilde{m}'\right) \\
0 &= \tilde{m}^T \vec{t} \times \left(\mathbf{R}\tilde{m}'\right)
\end{aligned}
$$

Defining EG with Normalized Images

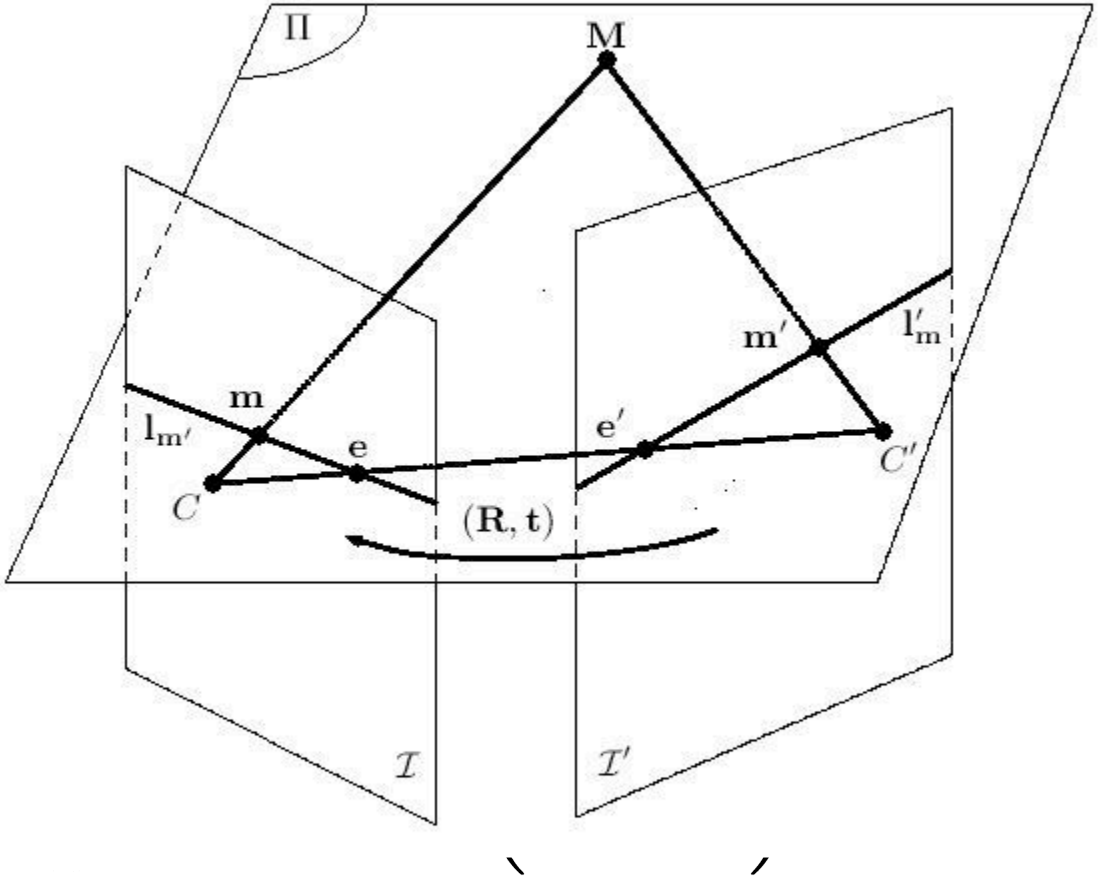
$$\tilde{m} =$$

Coplanar

$$\vec{t} \times \tilde{m} =$$

$$\tilde{m} \bullet \vec{t} \times \tilde{m} =$$

$$0 =$$



Defining EG with Normalized Images

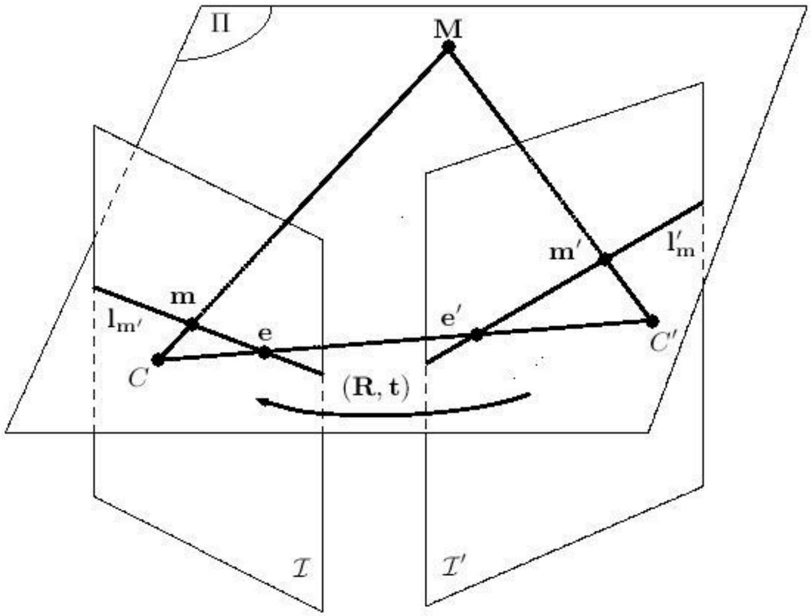
$$0 = \tilde{m}^T \vec{t} \times \left(\mathbf{R}\tilde{m}'\right)$$

$$0 = \tilde{m}^T \underbrace{[\vec{t}]_x \mathbf{R}}_{\mathbf{E}} \tilde{m}'$$

$$\Rightarrow \tilde{m}^T \mathbf{E} \tilde{m}' = 0$$

$$\tilde{m}^T l_{m'} = 0$$

$$\mathbf{E}\tilde{m}' = l_{m'}$$



Properties of **E**

- Determined completely by the transform between the cameras
- det(E)=0
- $l_{m'} = \mathbf{E}\tilde{m}' \Longleftrightarrow l'_m = \mathbf{E}^T\tilde{m}$
- $\tilde{m}^T\mathbf{E}\tilde{m}' = 0$

The Fundamental Matrix

- Fundamental Matrix F must satisfy

$$\tilde{m}^T \mathbf{F} \tilde{m}' = 0$$

where

$$\mathbf{F} = \mathbf{A}^{-T} \mathbf{E} \mathbf{A}'^{-1}$$

- A and A' are intrinsic Matrices of the two Cameras

$$0 = \left(\tilde{m}^T \mathbf{A}^{-T}\right) \mathbf{E} \left(\mathbf{A}'^{-1} \tilde{m}'\right)$$

Properties of **F**

- det(**F**)=0 because det(**E**)=0
- **F** is of rank 2
- **F** has 9 Parameters but 7DOF
- $l_{m'} = \mathbf{F}\tilde{m}' \Longleftrightarrow l'_m = \mathbf{F}^T\tilde{m}$
- $\tilde{m}^T\mathbf{F}\tilde{m}' = 0$

Summary

- Stereo Vision is about point matching
- **E** and **F** map points in one image to lines in the other.
- **F** satisfies $\tilde{m}^T \mathbf{F} \tilde{m}' = 0$ and has 7 DOF
  - Epipolar Geometry between two arbitrary images can be estimated by 7 point correspondences

Calibration Basics

- Assumptions
  - The full perspective projection model is used.
  - Intrinsic and Extrinsic parameters of the Cameras are unknown.
  - Rigid transformation between the two camera systems

Equivalent Problems

- 2 Images by a moving camera at two different moments (static).
- 2 Images by a fixed camera at two different moments (single object,dynamic)
- 2 Images by 2 Cameras at the same time
- 2 Images by 2 Cameras (static)

Definitions

- The Epipolar Equation

$$\tilde{m}_i^T \mathbf{F} \tilde{m}'_i = 0 \text{ with } \tilde{m}_i = (x_i, y_i, 1)^T$$

can be written as linear homogeneous equation

$$\vec{u_i}^T \vec{f} = 0$$

with
$$\begin{aligned} \vec{u}_i &= \left(x_i x'_i, x_i y'_i, x_i, y_i x'_i, y_i y'_i, y_i, x'_i, y'_i, 1\right)^T \\ \vec{f} &= (F_{11}, F_{12}, F_{13}, F_{21}, F_{22}, F_{23}, F_{31}, F_{32}, F_{33})^T \end{aligned}$$

Definitions

- For n point matches

$$\mathbf{U_n}^T \vec{f} = \vec{0}$$

$$\text{where } \mathbf{U_n} = \begin{pmatrix} \vec{u_1}^T \\ \vdots \\ \vec{u_n}^T \end{pmatrix}$$

Not that easy

- Noise
- Least-square methods are very sensitive to false matches
- Enforce rank 2 constraint or epipolar lines will "smear out"

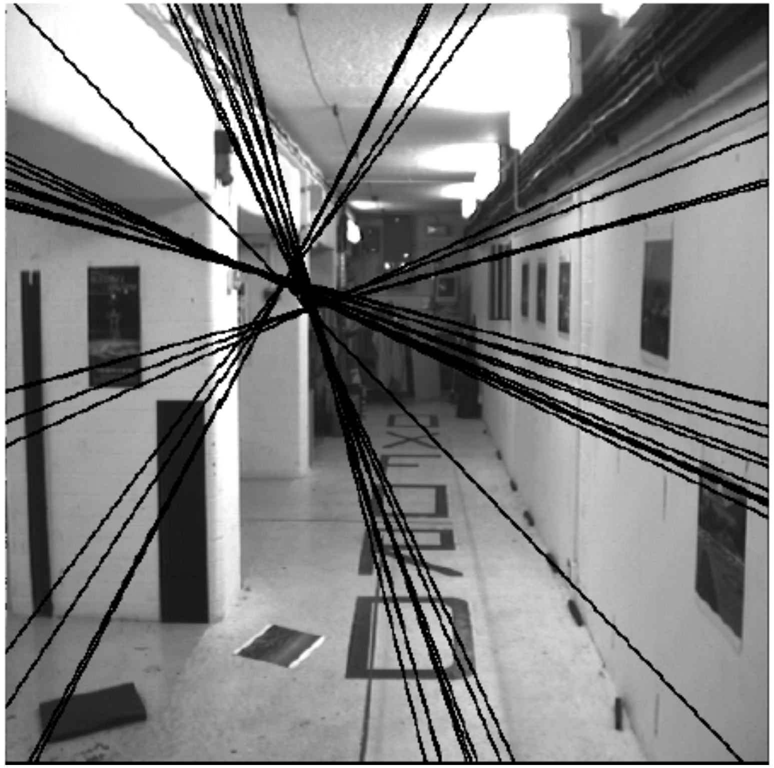
  find closest rank 2 approximation of F

Rank 2 Constraint

$$\mathbf{F} = \mathbf{U} \begin{bmatrix} \sigma_1 & 0 & 0 \\ 0 & \sigma_2 & 0 \\ 0 & 0 & \sigma_3 \end{bmatrix} \mathbf{V}^T$$

$$\mathbf{F}_{rank2} = \mathbf{U} \begin{bmatrix} \sigma_1 & 0 & 0 \\ 0 & \sigma_2 & 0 \\ 0 & 0 & 0 \end{bmatrix} \mathbf{V}^T$$

Rank 2 Constraint



Rank 3

Rank 2

7 Point Algorithm

- In this case n=7 and rank(U_7)=7
- SVD

  $$\mathbf{U}_7 = \mathbf{X}\mathbf{D}\mathbf{Y}^T$$

  $\mathbf{D}$ holds singular values

  $\mathbf{Y}$ holds right singular vectors

- 2 last columns of Y span the right null-space of U => F1 and F2

7 Point Algorithm

- $\mathbf{F} = \alpha\mathbf{F}_1 + (1-\alpha)\mathbf{F}_2$
  - 1 Parameter family of matrices because F is homogenous
- $\det(\mathbf{F}) = 0$ ($\mathbf{F}$ is of rank 2)

$$\Rightarrow \alpha$$

$$\Rightarrow \mathbf{F}$$

- Can have 1 or 3 Solutions!

8 Point Algorithm(s)

- Usually more than 7 Matches
- Many Methods (see [1])
  - Linear Methods
  - Nonlinear Methods
  - Robust Methods

Linear Methods

- Minimize epipolar Equation

$$\min_{\mathbf{F}} \sum_i \left(\tilde{m}_i^T \mathbf{F} \tilde{m}_i' \right)^2$$

or better

$$\min_{\vec{f}} \| \mathbf{U}_n \vec{f} \|^2$$

- There is a trivial solution: $\vec{f} = \vec{0}$

Eigen Analysis

- Impose a constraint on norm of $\vec{f}$

$$\| \vec{f} \| = 1$$

- Get unconstrained problem through Lagrange multipliers

$$\min_{\vec{f}} \underbrace{\| \mathrm{U}_n \vec{f} \|^2 + \lambda \left(1 - \| \vec{f} \|^2 \right)}_{X(f,\lambda)}$$

Eigen Analysis - Minimizing

- First derivative = 0

$$\| \mathbf{U}_n \vec{f} \|^2 + \lambda \left(1 - \| \vec{f} \|^2\right) =$$

$$\vec{f}^T \mathbf{U}_n^T \mathbf{U}_n \vec{f} + \cancel{\lambda} - \lambda \vec{f}^T \vec{f}$$

deriving...

$$\left(\mathbf{U}_n^T \mathbf{U}_n\right) \vec{f} = \lambda \vec{f}$$

λ is eigenvalue that corresponds to unit eigenvector $\vec{f}$

Eigen Analysis - Back substitution

- There are 9 eigenvalues
- Back substituting the solution

$$X\left(\vec{f},\lambda_i\right)=\lambda_i$$

- Solution is the Eigenvector of $\mathbf{U}_n^T\mathbf{U}_n$ associated to the smallest Eigenvalue (smallest Singular Vector of $\mathbf{U}_n$)

Problems with most linear methods

- The rank 2 constraint is not satisfied
- Minimized Quantity has no physical meaning
- High instability (in the numerical sense)
  => normalize Input Data!

Iterative Linear Method

- Minimize the distances between points and corresponding epipolar lines
- Symmetrical Distance

$$\min_{\mathbf{F}} \sum_i \left[d^2\left(\tilde{m}_i, \mathbf{F}\tilde{m}'_i\right) + d^2\left(\tilde{m}'_i, \mathbf{F}^T\tilde{m}_i\right) \right]$$

Iterative Linear Method - Weights

- Represent distances as weights

$$\min_{\mathbf{F}} \sum_i \left[w_i^2 \left(\tilde{m}_i^T \mathbf{F} \tilde{m}_i' \right)^2 \right]$$

- Problem: Weights depend on **F**
- Solution:
  - Set all weights to 1
  - Compute F
  - Calculate new weights from F
  - Repeat 'till satisfaction

Iterative Linear Method - Can't get no ....

- Pros:
  - Easy to implement
  - Minimizes Physical quantity
- Cons:
  - Rank-2 constraint isn't satisfied
  - No significant improvement[1]
- Better:
  - Nonlinear Minimization in Parameter Space [1]

Robust Methods - RANSAC

- Exploit heuristics
- Try to remove "outliers"
- RANdom SAmple Consensus

Robust Methods - RANSAC

Step 1. Extract features

Step 2. Compute a set of potential matches

Step 3. do

- Step 3.1 select minimal sample
- Step 3.2 compute solution(s) for F

(generate hypothesis)

- Step 3.3 determine inliers (verify hypothesis)

until $\Gamma(\#inliers, \#samples) \geq 95\%$

Step 4. Compute F based on all inliers

Step 5. Look for additional matches

Step 6. Refine F based on all correct matches

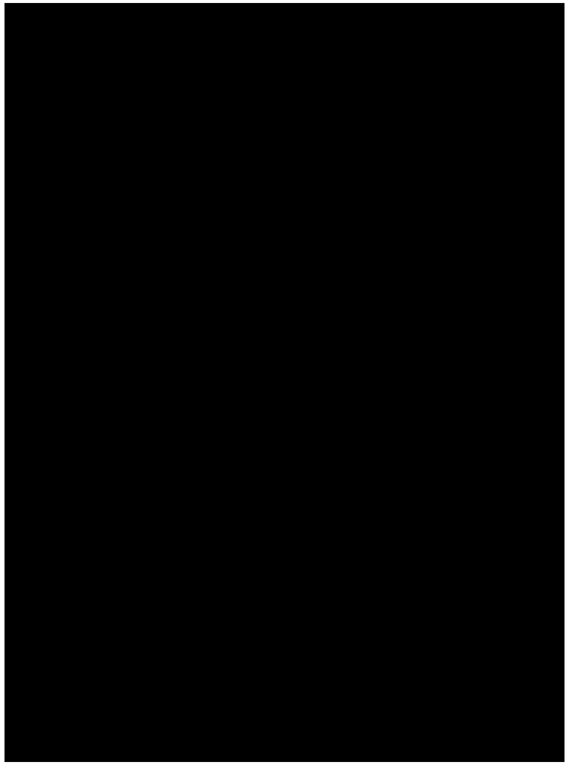
Robust Methods - RANSAC

- Propability that at least one random sample is free of outliers

$$\Gamma = \left(1 - \left(\frac{\#inliers}{\#matches}\right)^{samplesize}\right)^{\#samples}$$

Experiments / Tricks

- Image Rectification [2]



Experiments / Tricks

- Image Rectification [2]
- Demos online

Further Readings

- *"Epipolar Geometry in Stereo Motion and Object Recognition"*

Gang Xu and **Zhengyou Zhang**

Many (if not all) algorithms

You don't want to buy it...

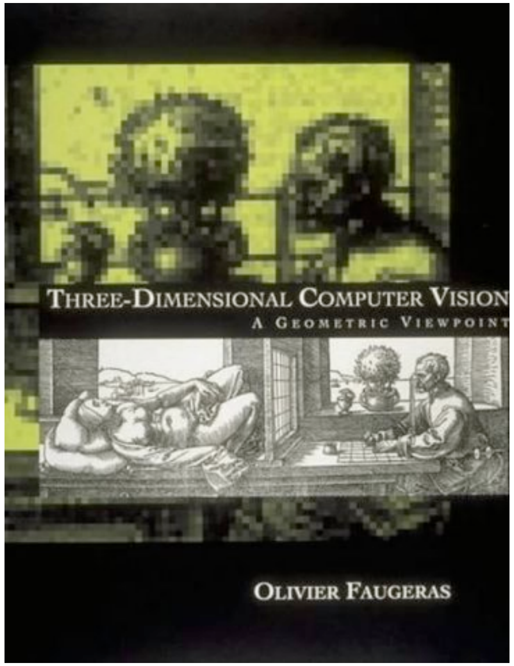
Get it from the library...

[1]

Further Readings

- *"Three-Dimensional Computer Vision - A Geometric Viewpoint"*

Olivier Faugeras

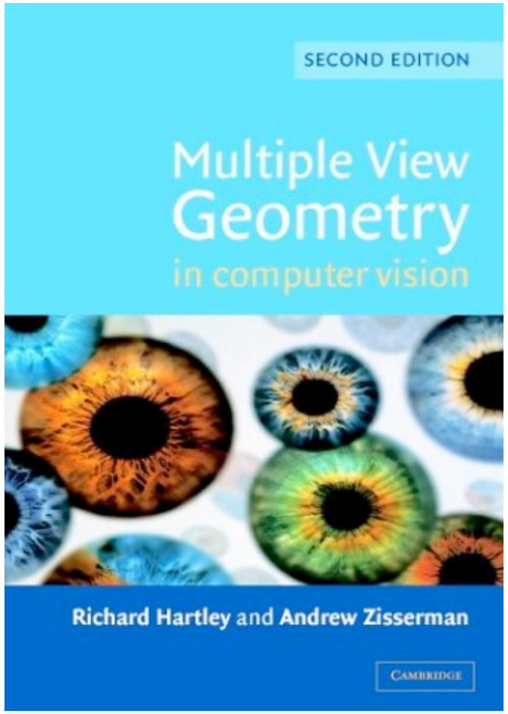


Good Basics, Interesting Chapter about Edge-Detection. Many Images.

[2]

Further Readings

- *"Multiple View Geometry in Computer Vision"*



Richard Hartley and **Andrew Zisserman**

[3]

Further Readings

- Web

http://www.esat.kuleuven.ac.be/~pollefey/tutorial/node3.html

http://www.cs.unc.edu/~blloyd/comp290-089/fmatrix/
(matlab functions)

http://www-sop.inria.fr/robotvis/personnel/zzhang/CalibEnv/CalibEnv.html
(demo)

http://www.vision.caltech.edu/bouguetj/calib_doc/
(matlab toolbox)

[3]

That's it

Thanks