3D VISION

Stereo Projective Calibration

Systems Engineering and Automation

Miguel Hernández University

Table of Contents

Stereo Projective Calibration

- *Projective Calibration*: The Fundamental Matrix ***F*** is known. We can obtain the epipoles, epipolar lines, and the homography between the epipolar lines

- *Affine Calibration*: The fundamental F matrix and the homography for points at infinity are known H_∞

- *Euclidean Calibration*: The intrinsic and extrinsic parameters involved in the image formation are known

Stereo Projective Calibration

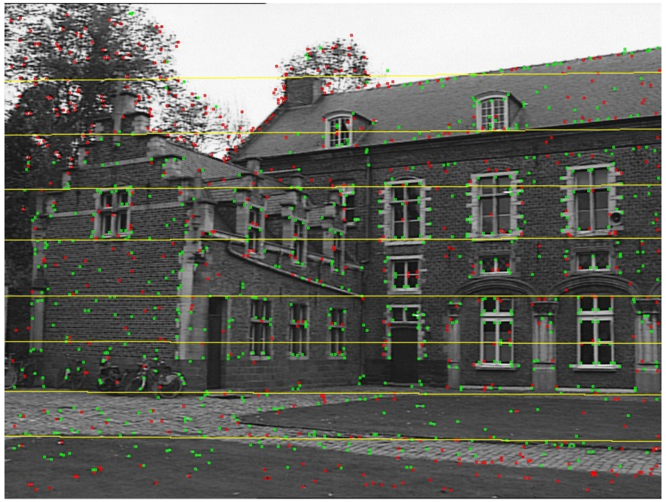
- Estimate of the Fundamental Matrix from a set of corresponding points in two images

$$\tilde{m}'^{T}{}_{i} F \ \tilde{m}_i = 0$$

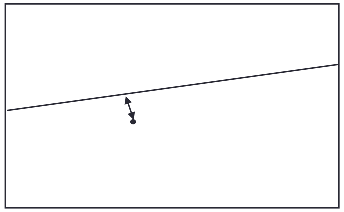
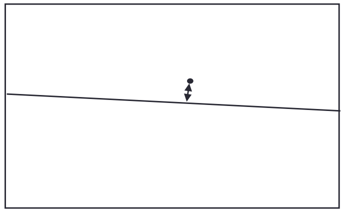
- Estimation of the Fundamental Matrix
  - 8-point algorithm
  - 7-point algorithm
  - Adjusting epipoles (rank of F)
  - Normalization

Fundamental Matrix Calibration

- Estimation error

$$d^2 = d(\mathrm{m}'_i, F\mathrm{m}_i)^2 + d\left(\mathrm{m}_i, \mathrm{F}^{\mathrm{T}}\mathrm{m}'_i\right)^2$$

"Visual 3D Modeling from Images" Marc Pollefeys
Tutorial Notes University of North Carolina – Chapel Hill, USA http://www.cs.unc.edu/~marc/tutorial.pdf

Fundamental Matrix Calibration

$$\tilde{m}'^T_i F\ \tilde{m}_i = 0 \qquad \tilde{m}' = [x' \quad y' \quad 1]^T, \quad \tilde{m} = [x \quad y \quad 1]^T$$

$$x'xf_{11} + x'yf_{12} + x'f_{13} + y'xf_{21} + y'yf_{22} + y'f_{23} + xf_{31} + yf_{32} + f_{33} = 0$$

$$\begin{bmatrix} x'_1x_1 & x'_1y_1 & x'_1 & y'_1x_1 & y'_1y_1 & y'_1 & x_1 & y_1 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x'_nx_n & x'_ny_n & x'_n & y'_nx_n & y'_ny_n & y'_n & x_n & y_n & 1 \end{bmatrix} \mathbf{f} = 0$$

$$\boxed{A_{xy}\ \mathbf{f} = 0}$$

$$\mathbf{f} = [f_{11} \quad f_{12} \quad f_{13} \quad f_{21} \quad f_{22} \quad f_{23} \quad f_{31} \quad f_{32} \quad f_{33}]^T$$

Solución mínimos cuadrados:

$$\min_{\mathbf{f}} \left[\left(\mathbf{A}_{xy}\mathbf{f}\right)^2\right]$$

$$SVD(A_{xy}) \rightarrow A_{xy} = UDV^T \Rightarrow \mathbf{f} = v_9$$
$$\|\mathbf{f}\| = 1$$

It doesn't impose that F is singular

Fundamental Matrix Calibration

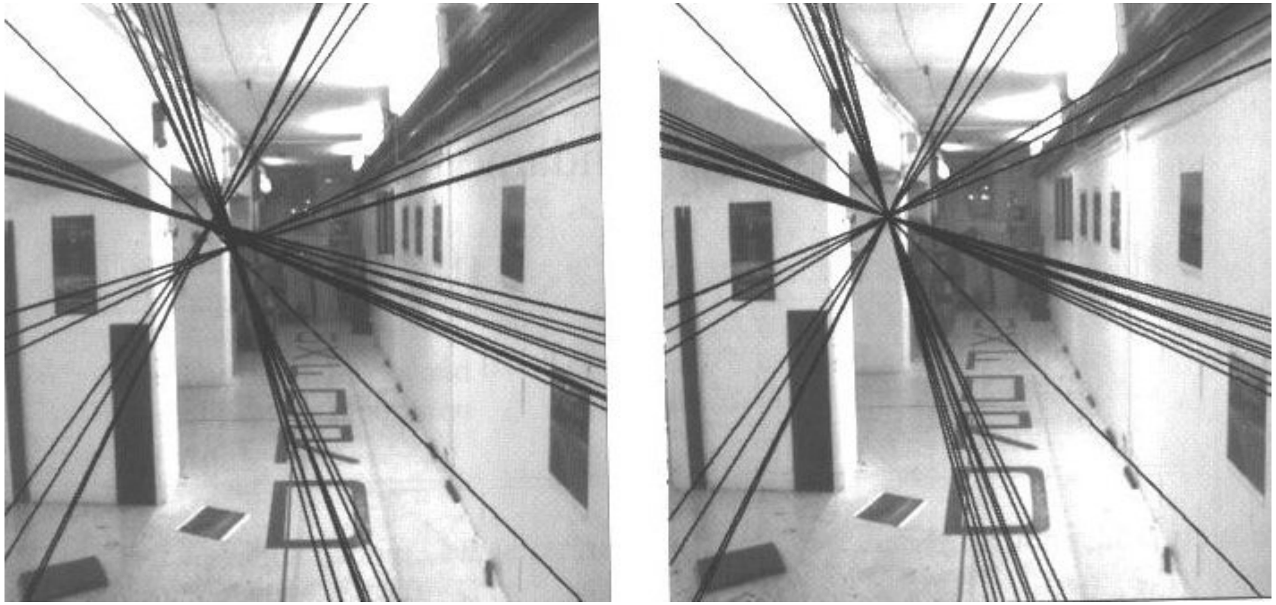
- Imposing restrictions on F:
  - Matrix F must be singular but the least squares algorithm does not impose this restriction
  - Solution:
    - Calculate **SVD** decomposition of **F** and impose that the lowest singular value is zero

$$F = UDV^T \longrightarrow \boxed{F^* = UD^*V^T, \qquad |F^*| = 0}$$

$$D = \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{bmatrix} \rightarrow D^* = \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Fundamental Matrix Calibration

- Rank adjustment of matrix F



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

Fundamental Matrix Calibration

- Normalization:
  - The condition of the matrix A_{xy} depends on the absolute values of x,y:
    - There might be rows with very high values and others with very small values

$$\begin{bmatrix} x'_1x_1 & x'_1y_1 & x'_1 & y'_1x_1 & y'_1y_1 & y'_1 & x_1 & y_1 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x'_nx_n & x'_ny_n & x'_n & y'_nx_n & y'_ny_n & y'_n & x_n & y_n & 1 \end{bmatrix} \mathbf{f} = 0$$

  - This makes the estimation very sensitive to noise
  - Solution: coordinates normalization

Normalization: Transform image coordinates: $\hat{m}_i = Tm_i, \ \hat{m}'_i = T'm'_i$

where T and T' are linear transformations (translation and scaling)

- Translation: the centroids of the points $\hat{m}_i, \ \hat{m}'_i$ are in *(0,0)*

- Scaling: the average distance from the points to the origin is $\sqrt{2}$

$$T = \begin{bmatrix} 1/\sigma_x & 0 & -\mu_x/\sigma_x \\ 0 & 1/\sigma_y & -\mu_y/\sigma_y \\ 0 & 0 & 1 \end{bmatrix} \quad T' = \begin{bmatrix} 1/\sigma'_x & 0 & -\mu'_x/\sigma'_x \\ 0 & 1/\sigma'_y & -\mu'_y/\sigma'_y \\ 0 & 0 & 1 \end{bmatrix}$$

Alternative: $$T = T' = \begin{bmatrix} 2/n_c & 0 & -1 \\ 0 & 2/n_f & -1 \\ 0 & 0 & 1 \end{bmatrix}$$

8 points Normalized Algorithm

- Given n≥8 point correspondences on the image:

$$m_i \leftrightarrow m'_i \qquad \tilde{m}'^T_i F \; \tilde{m}_i = 0$$

(i) **Normalization**: calculate the transformations ***T***, ***T'*** (centroid in (0,0) and average distance $\sqrt{2}$), and apply the transformations to each pairs of points $(m_i, m'_i) \rightarrow (\hat{m}_i, \hat{m}'_i) \quad \hat{m}_i = Tm_i, \; \hat{m}'_i = T'm'_i$

(ii) **Calculate the Fundamental matrix** $\hat{F}'$ with the correspondences $(\hat{m}_i, \hat{m}'_i)$ from the smallest singular value of the matrix A_{xy} (last singular vector $\hat{v}_n$) *SVD(A_{xy})*

(iii) **Imposing matrix F singularity**: replace $\hat{F}'$ by $\hat{F} = UD^*V^T$ so that $|\hat{F}| = 0$ by means .

(iv) **Denormalization**: calculate $F = T'^T \hat{F} \; T$ corresponding to the original coordinates.

7 points Normalized Algorithm

- Given n=7 corresponding points on the image:
  - The Null space of A_{xy} will have 2 dimensions (f_1, f_2)->(F_1, F_2)

$$\boxed{A_{xy}\ \mathbf{f} = 0}$$

  - The solution is obtained by imposing the uniqueness of F

$$\det(\alpha F_1 + (1-\alpha)F_2) = 0$$

  - It's a cubic equation in $\alpha \Rightarrow$ (1 or 3 real solutions)

$$\boxed{F = \alpha F_1 + (1-\alpha)F_2}$$

  - The normalization and denormalization process indicated previously would also need to be applied

Other methods

- Algebraic minimization to impose matrix F uniqueness:
  - Uses Mahalanobis distance instead of quadratic distance

- Minimize Geometric Distance

$$d = \sum_i \left(d(m_i, \hat{m}_i) + d(m_i{}', \hat{m}_i{}')\right)$$

$$m_i \leftrightarrow m'_i \quad \text{correspondences}$$

$$\hat{m}_i \leftrightarrow \hat{m}'_i \quad \text{“}\mathit{true}\text{” correspondences (estimated)}$$

  - Nonlinear algorithm, is based on the estimate of least squares
  - Requires estimating P, P' from F, to calculate “*true*” coordinates (Projective 3D Reconstruction)
  - From the projective reconstruction (3D coordinates) projections are estimated:

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

Essential Matrix Calibration

- When the cameras are calibrated (*A*, *A'*) (6 dof)

$$\tilde{q}'_i \, E \, \tilde{q}_i = 0$$

$$\boxed{\begin{aligned}\tilde{q}_i &= A^{-1}\tilde{m}\\ \tilde{q}'_i &= A'^{-1}\tilde{m}'\end{aligned}}$$

  - The procedure is similar to the calculation of F, only varies the adjustment of the E restrictions
  - Calculate decomposition **SVD**(E) and impose that the least singular value is zero, and the other two are equal.

$$E = UDV^T \longrightarrow \boxed{E^* = UD^*V^T, \qquad |E^*| = 0}$$

$$D = \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{bmatrix} \rightarrow D^* = \begin{bmatrix} (\lambda_1 + \lambda_2)/2 & 0 & 0 \\ 0 & (\lambda_1+\lambda_2)/2 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Table of Contents

RANSAC: Objectives

- Estimate an Homography/Correlation between images:
  - Data (measurements) contain errors (Gaussian distribution)
  - The correspondence problem is complex $\Rightarrow$ false correspondences (they don't follow Gaussian distribution)
  - No pre-filtering of correspondences is required
  - Integrates into a single process the estimation of ***H / F*** and the filtering of incorrect correspondences (“*outliers*”)

- **RANSAC**: RANdom SAmple Consensous
  - Try to separate the false correspondences (‘***outliers*** ’) of the true correspondences (‘***inliers*** ’) that allows you to calculate optimally *H / F*
  - The speed of convergence depends on the number of measures.
    - Combinatorial explosion. Limit as much as possible the number of data in the image (the most 'relevant')
  - *Applications:* curve fitting, Hough transform, Homography, Fundamental Matrix, POSE

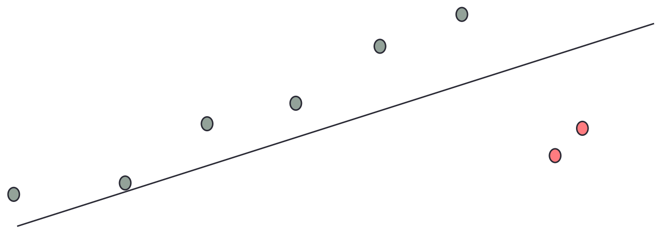
“Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography” M.A. Fischler, R.C. Bolles Communications of the ACM 24 (6) 381-395, 1981

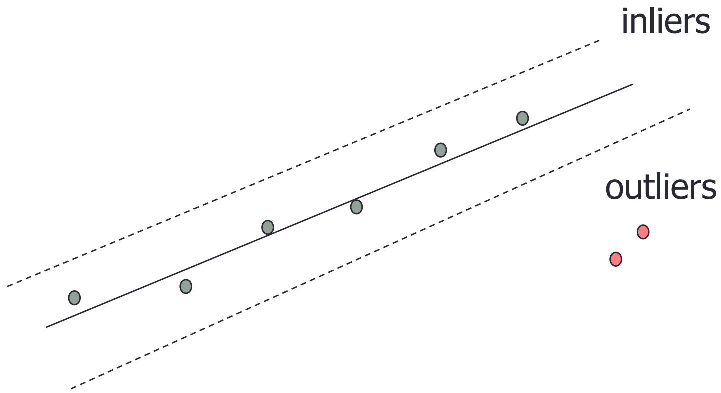
Example: adjusting a line

- Estimating a line fitting:
  - Least squares

$$\begin{bmatrix} x_i & y_i & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = 0 \Rightarrow \begin{bmatrix} x_1 & y_1 & 1 \\ \vdots & \vdots & \vdots \\ x_n & y_n & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \bar{0}$$

$$SVD\left(\begin{bmatrix} x_1 & y_1 & 1 \\ \vdots & \vdots & \vdots \\ x_n & y_n & 1 \end{bmatrix} \right) = UDV^T \Rightarrow \begin{bmatrix} a & b & c \end{bmatrix} = {v_n}^T$$

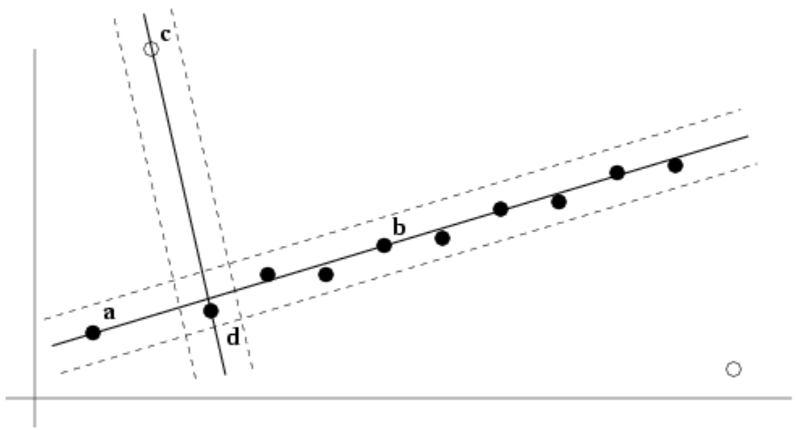




Estimation considering only “*inliers*”

Example: adjusting a line

- RANSAC: Estimate with minimum data set
  - Example: two points (Subset ***s***)
  - Random selection of points
  - Check the *consensus* (Likelihood) selection with the other points of ***S***
  - It is assumed that most of the data is correct '***inlier***'



RANSAC: Algorithm

- Objective
  - Robust model fitting for a dataset ***S*** containing errors '*outliers*'

- Algorithm:

(i) Randomly select a point sample ***s*** of the total set ***S*** and calculate the model with this subset

(ii) Determine the subset of points S_i that are within a distan threshold ***t*** to the model. The subset S_i is the ***consensus*** *set and defines the* '***inliers***' of ***S***.

(iii) Id the subset S_i is greater than a threshold ***T***, re-estimate the model using all the points of S_i and finish.

(iv) If the size of S_i is lower than ***T***, select a new subset and repeat the previous step.

(v) After ***N*** attempts, select the subset S_i with greater consensus, re-estimating the model using all the points in the subset S_i

RANSAC: Algorithm

- Parameters:
  - ***s***: number of data used to estimate the model (known)
  - ***n***: total number of data (known)
  - ***d***: points-model distance function (chosen)
  - ***e***: probability of 'outliers' (estimated)
  - ***p***: Probability that the algorithm will select only "inliers". Is the probability that it will produce a useful result (chosen)

  - ***t***: point-consensus distance threshold (calculated/chosen)
  - ***T***: number of points that must meet the consensus to finish the algorithm (estimated)
  - ***N***: number of iterations required (estimated)

RANSAC: '*inliers*' threshold (distance ***t***)

- Choose ***t*** so that the likelihood of 'inliers' (*1*-***e***) is for example 0.95
  - It is often chosen empirically
  - If the noise of the measurements can be adjusted to a Gaussian of mean zero and standard deviation σ then $d_\perp^2$ follows a distribution χ_m^2

m: model *co-dimension* (distance)

Co-dimension	Model	T^2
1	Line, F	$3.84\sigma^2$
2	H, P	$5.99\sigma^2$
3	T	$7.81\sigma^2$

- Using of the median of the distances (does not require predefined thresholds> LMEDS (*Least median of Squares*)

RANSAC: consensus threshold (*T*)

- Choose the acceptable size of the set as the number of 'inliers' estimated in the total set

- Being
  - ***e:*** the probability of errors (*outliers*)
  - ***n***: total set size ***S***

$$T = (1 - e) \cdot n$$

- The value of ***e*** (probability of outliers) can be calculated adaptively in each iteration of the algorithm.

$$e^{(i)} = 1 - \frac{size(S_i)}{n}$$

RANSAC: max. number of samples

- Choose ***N*** so that, with probability *p*, at least one random sample is free of '*outliers*'. (e.g. *p*=0.99)
- Being **e** the probability of errors (outliers)
  - Prob. that there is some '*outlier*' in ***s***: $1-(1-e)^s$
  - Prob. that there is some '*outlier*' in ***s*** for ***N*** Iterations: $1-p$

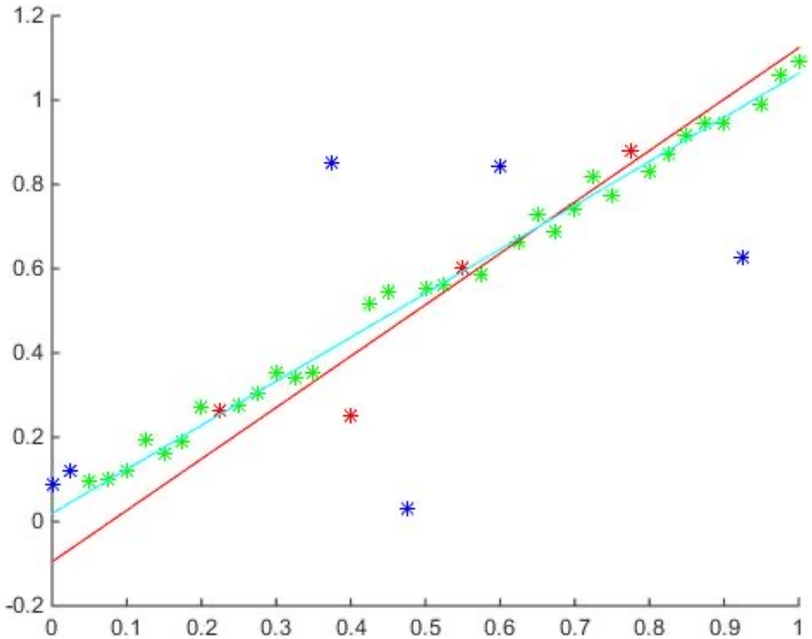
$$(1-(1-e)^s)^N = 1-p$$

$$\boxed{N = \log(1-p)\,/\,\log(1-(1-e)^s)}$$

	Proportion of outliers e						
s	5%	10%	20%	25%	30%	40%	50%
2	2	3	5	6	7	11	17
3	3	4	7	9	11	19	35
4	3	5	9	13	17	34	72
5	4	6	12	17	26	57	146
6	4	7	16	24	37	97	293
7	4	8	20	33	54	163	588
8	5	9	26	44	78	272	1177

RANSAC (Example)

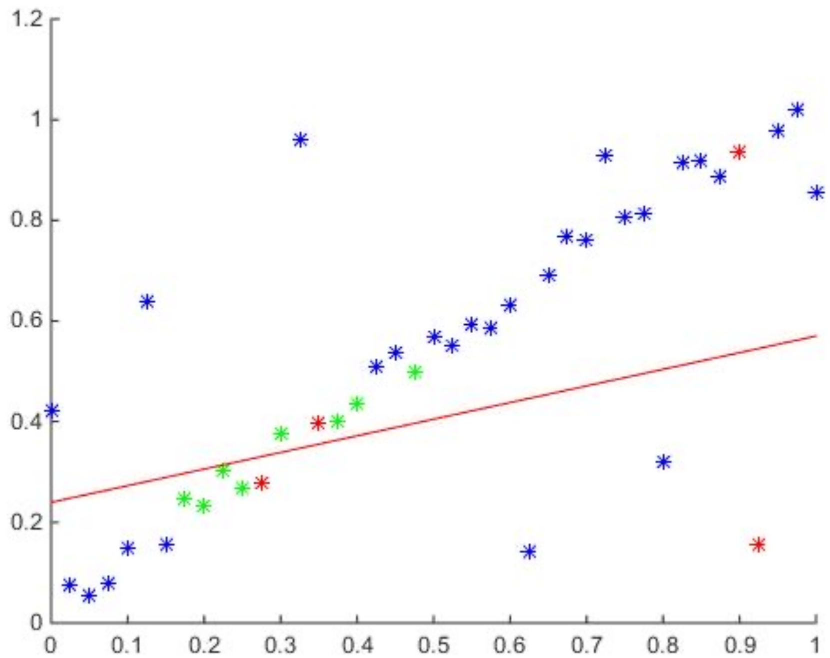
- Example line fitting: *e=0.2 ; s=4 ; n=50; T=40 (80%) ; N=9 ; t=0.1*
  - Red dots (chosen)
  - Green dots (complies with the consensus)
  - Blue dots (don't comply with the consensus)
  - Red line (first estimate)
  - Cyan line (second estimate, with all consensus points)



- Percentage of correct points: 85%
- First estimatation, average distance consensus points: 0.0372
- Second estimation, average distance consensus points : 0.0184

RANSAC (Example)

- Example line fitting: *e=0.2 ; s=4 ; n=50; T=40 (80%) ; N=9 ; t=0.1*
  - Red dots (chosen)
  - Green dots (complies with the consensus)
  - Blue dots (don't comply with the consensus)
  - Red line (first estimate)
  - Cyan line (second estimate, with all consensus points)



- Percentage of correct points: 24%
- First estimate, average distance consensus points : 0.0503

RANSAC: Applications

- Estimation of Homographs between projective planes:

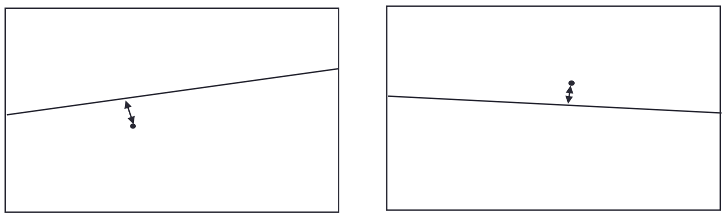
$$\tilde{m}'_j \cong H\ \tilde{m}_i \qquad \begin{bmatrix} \lambda\hat{x}' \\ \lambda\hat{y}' \\ \lambda \end{bmatrix} = H\ \tilde{m}_i$$

  - 4 points (***s***)
  - Distance: $d^2 = \left\| m'_j - \lambda^{-1}(H\tilde{m}_i) \right\|$

- Estimación de la matriz fundamental
  - 7/8 points
  - Distance to Epipolar line :

$$\tilde{m}'^T_j F\ \tilde{m}_i = 0$$

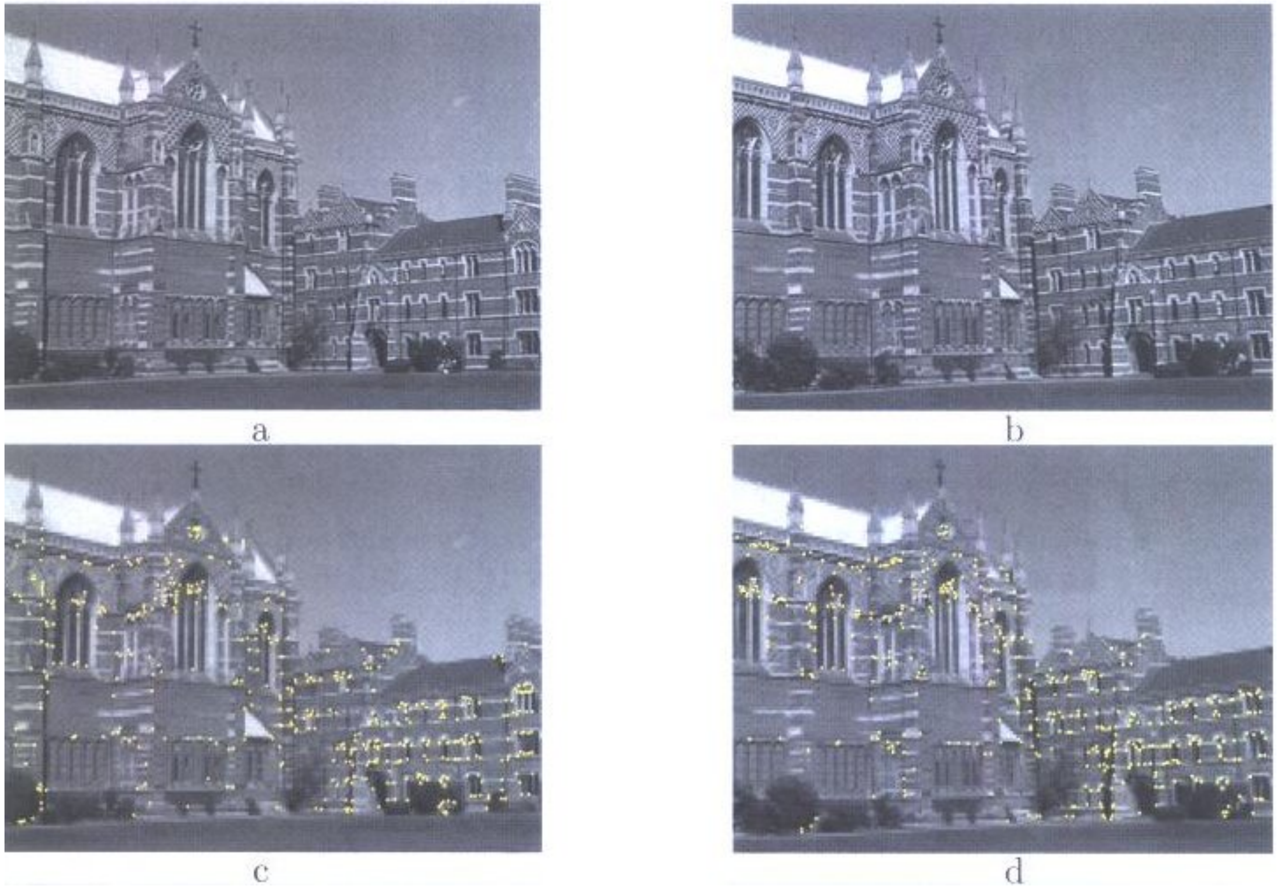
$$d^2 = d(\mathrm{m}'_i, F\mathrm{m}_i)^2 + d\left(\mathrm{m}_i, \mathrm{F}^{\mathrm{T}}\mathrm{m}'_i\right)^2$$



$$d(\tilde{m}_i, F^T\tilde{m}'_i) = \frac{\tilde{m}_i{}^T \cdot \tilde{l}}{\sqrt{l_1^2 + l_2^2}} \qquad \tilde{l} \cong F^T\tilde{m}'_i = [l_1, l_2, l_3]$$

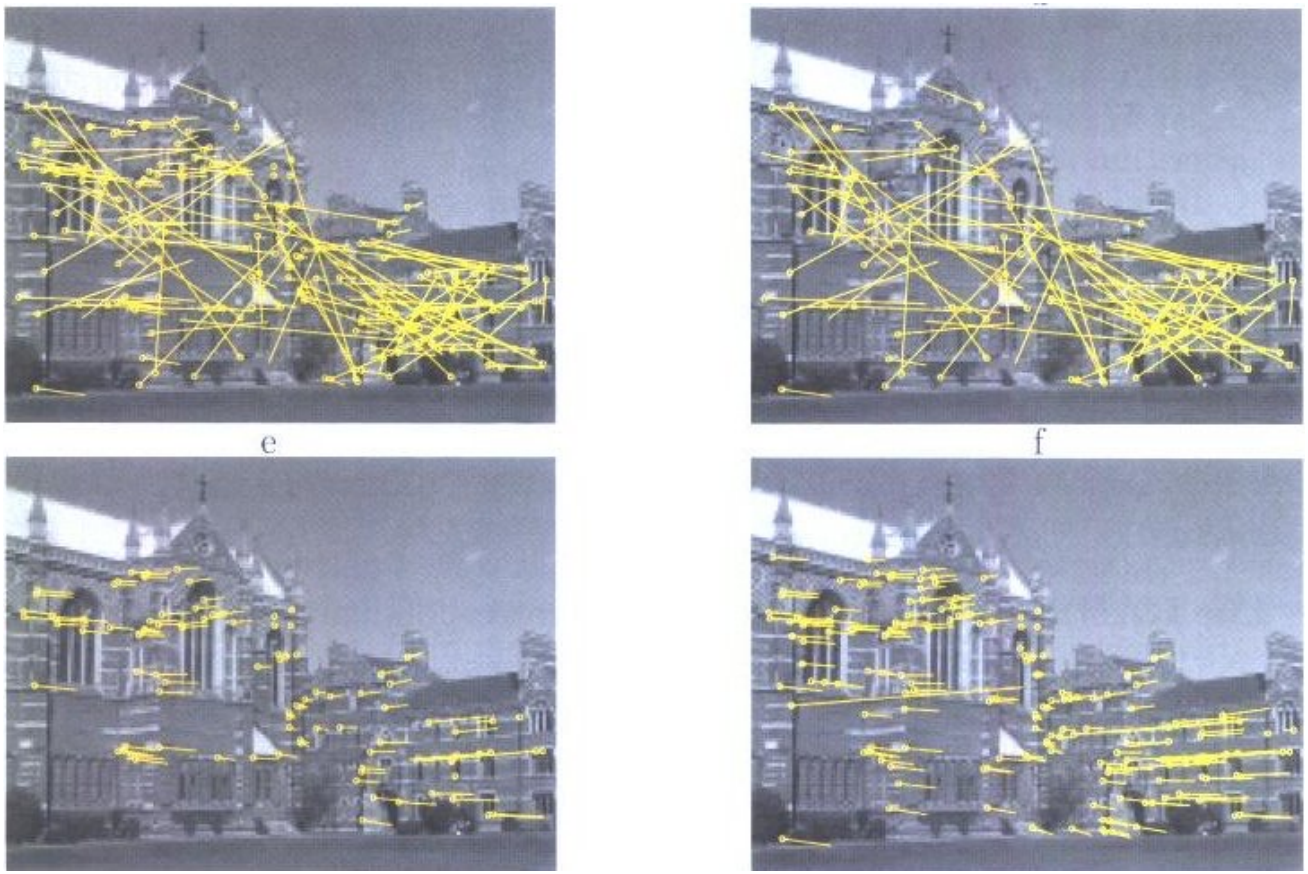
$$d(\tilde{m}'_i, F\tilde{m}_i) = \frac{\tilde{m}'_i{}^T \cdot \tilde{l}'}{\sqrt{l'^2_1 + l'^2_2}} \qquad \tilde{l}' \cong F\tilde{m}_i = [l'_1, l'_2, l'_3]$$

Example: calculation of F



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

Example: calculation of F



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

Summary

- Projection model of a stereo pair of cameras
- Homographies between different projective planes
- Projective Calibration → Fundamental Matrix/Essential Matrix
- Next steps :
  - Correspondence
  - 3D reconstruction