Computing F
class 13

Multiple View Geometry

Comp 290-089

Marc Pollefeys

Multiple View Geometry course schedule

(subject to change)

Jan. 7, 9	Intro & motivation	Projective 2D Geometry
Jan. 14, 16	(no class)	Projective 2D Geometry
Jan. 21, 23	Projective 3D Geometry	(no class)
Jan. 28, 30	Parameter Estimation	Parameter Estimation
Feb. 4, 6	Algorithm Evaluation	Camera Models
Feb. 11, 13	Camera Calibration	Single View Geometry
Feb. 18, 20	Epipolar Geometry	3D reconstruction
Feb. 25, 27	Fund. Matrix Comp.	Structure Comp.
Mar. 4, 6	Planes & Homographies	Trifocal Tensor
Mar. 18, 20	Three View Reconstruction	Multiple View Geometry
Mar. 25, 27	MultipleView Reconstruction	Bundle adjustment
Apr. 1, 3	Auto-Calibration	Papers
Apr. 8, 10	Dynamic SfM	Papers
Apr. 15, 17	Cheirality	Papers
Apr. 22, 24	Duality	Project Demos

Two-view geometry

Epipolar geometry

3D reconstruction

F-matrix comp.

Structure comp.

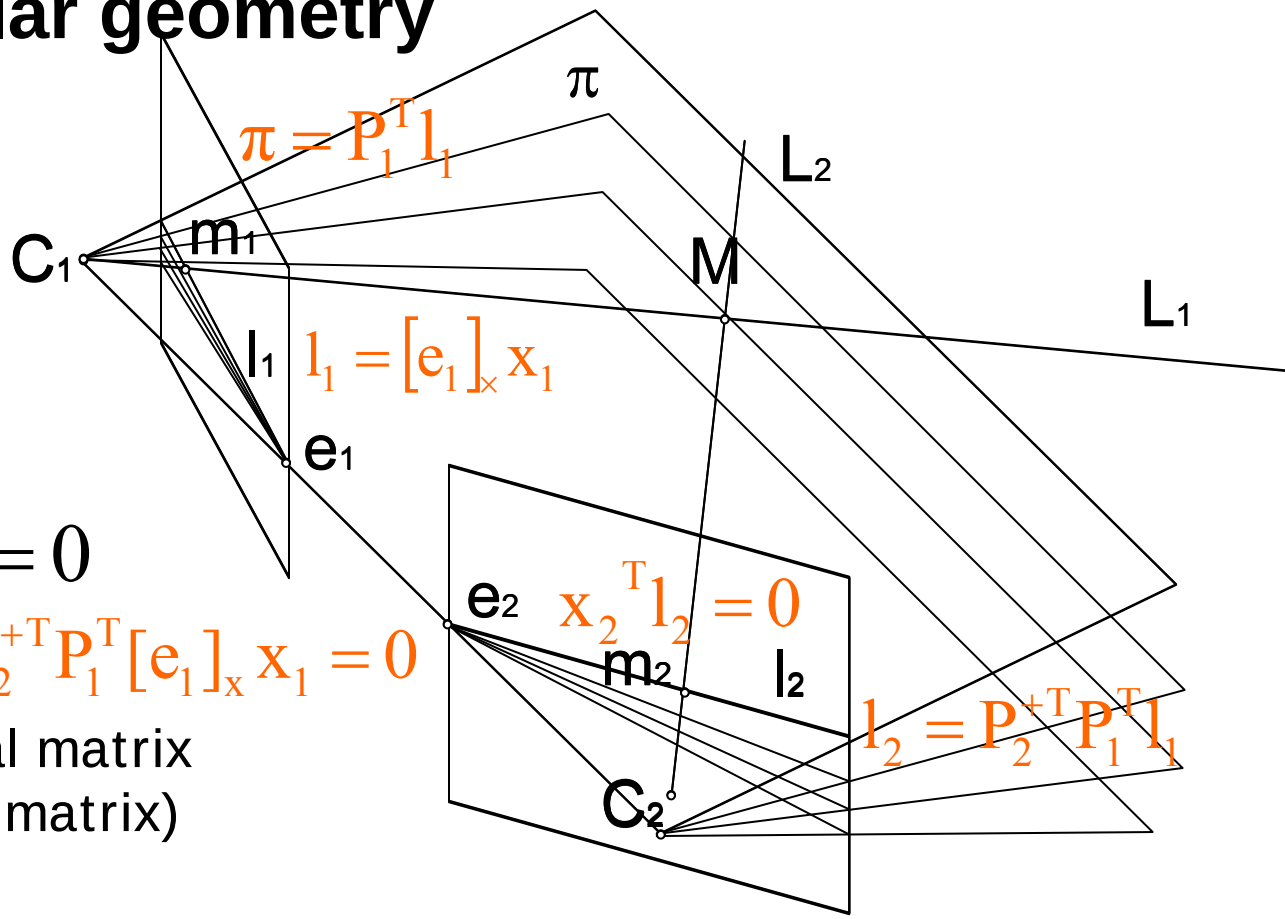
Epipolar geometry

Underlying structure in set of matches for rigid scenes

$$\overbrace{m_2^T \overbrace{\mathbf{F}}^{\mathbf{I}^T_1} m_1}^{\mathbf{I}_2} = 0$$

$$x_2^T P_2^{+T} P_1^T [e_1]_x x_1 = 0$$

Fundamental matrix
(3x3 rank 2 matrix)



Canonical representation:

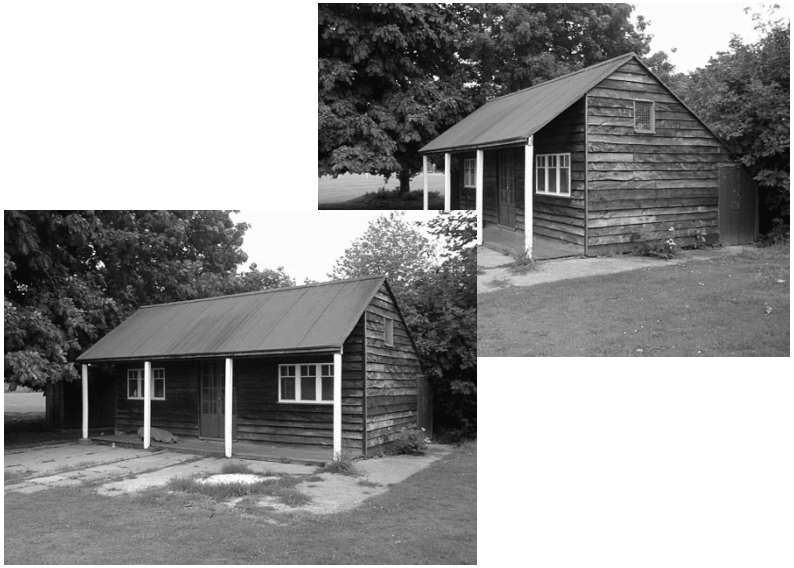
$$P = [I \mid 0] \quad P' = [[e']_\times F + e' v^T \mid \lambda e']$$

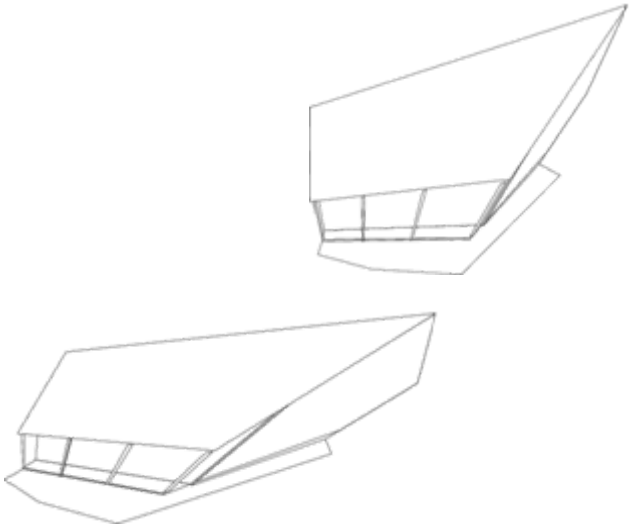
1. Computable from corresponding points
2. Simplifies matching
3. Allows to detect wrong matches
4. Related to calibration

The projective reconstruction theorem

If a set of point correspondences in two views determine the fundamental matrix uniquely, then the scene and cameras may be reconstructed from these correspondences alone, and any two such reconstructions from these correspondences are projectively equivalent

allows reconstruction from pair of uncalibrated images!





Objective

Given two uncalibrated images compute (P_M,P'_M,$\{X_{Mi}\}$)
(i.e. within similarity of original scene and cameras)

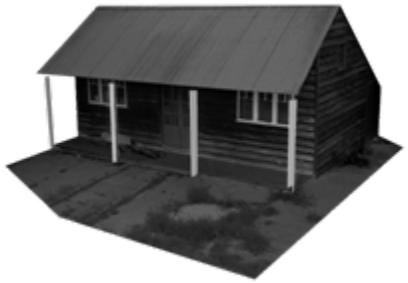
Algorithm

(i) Compute projective reconstruction (P,P‘,$\{X_i\}$)
  (a) Compute F from $x_i \leftrightarrow x'_i$
  (b) Compute P,P‘ from F
  (c) Triangulate X_i from $x_i \leftrightarrow x'_i$

(ii) Rectify reconstruction from projective to metric

Direct method: compute H from control points $X_{Ei} = HX_i$

$$P_M = PH^{-1} \quad P'_M = P'H^{-1} \quad X_{Mi} = HX_i$$



Stratified method:

(a) Affine reconstruction: compute π_∞

$$H = \begin{bmatrix} I & 0 \\ \pi_\infty & \end{bmatrix}$$

(b) Metric reconstruction: compute IAC ω

$$H = \begin{bmatrix} A^{-1} & 0 \\ 0 & 1 \end{bmatrix} \qquad AA^T = \left(M^T \omega M\right)^{-1}$$

Image information provided	View relations and projective objects	3-space objects	reconstruction ambiguity
point correspondences	F		projective
point correspondences including vanishing points	F, H_∞	π_∞	affine
Points correspondences and internal camera calibration	F, H_∞ ω, ω'	π_∞ Ω_∞	metric

Epipolar geometry: basic equation

$$\mathbf{x}'^{\mathrm{T}} \mathrm{F}\mathbf{x} = 0$$

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

separate known from unknown

$$[x'x, x'y, x', y'x, y'y, y', x, y, 1][f_{11}, f_{12}, f_{13}, f_{21}, f_{22}, f_{23}, f_{31}, f_{32}, f_{33}]^{\mathrm{T}} = 0$$

(data) (unknowns)

(linear)

$$\begin{bmatrix} x'_1 x_1 & x'_1 y_1 & x'_1 & y'_1 x_1 & y'_1 y_1 & y'_1 & x_1 & y_1 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x'_n x_n & x'_n y_n & x'_n & y'_n x_n & y'_n y_n & y'_n & x_n & y_n & 1 \end{bmatrix} \mathrm{f} = 0$$

$$\mathrm{A}\mathrm{f} = 0$$

the singularity constraint

$$e'^{T} F = 0 \qquad Fe = 0 \qquad \det F = 0 \qquad \operatorname{rank} F = 2$$

SVD from linearly computed F matrix (rank 3)

$$F = U \begin{bmatrix} \sigma_1 & & \\ & \sigma_2 & \\ & & \sigma_3 \end{bmatrix} V^T = U_1 \sigma_1 V_1^T + U_2 \sigma_2 V_2^T + U_3 \sigma_3 V_3^T$$

Compute closest rank-2 approximation $\min \|F - F'\|_F$

$$F' = U \begin{bmatrix} \sigma_1 & & \\ & \sigma_2 & \\ & & 0 \end{bmatrix} V^T = U_1 \sigma_1 V_1^T + U_2 \sigma_2 V_2^T$$

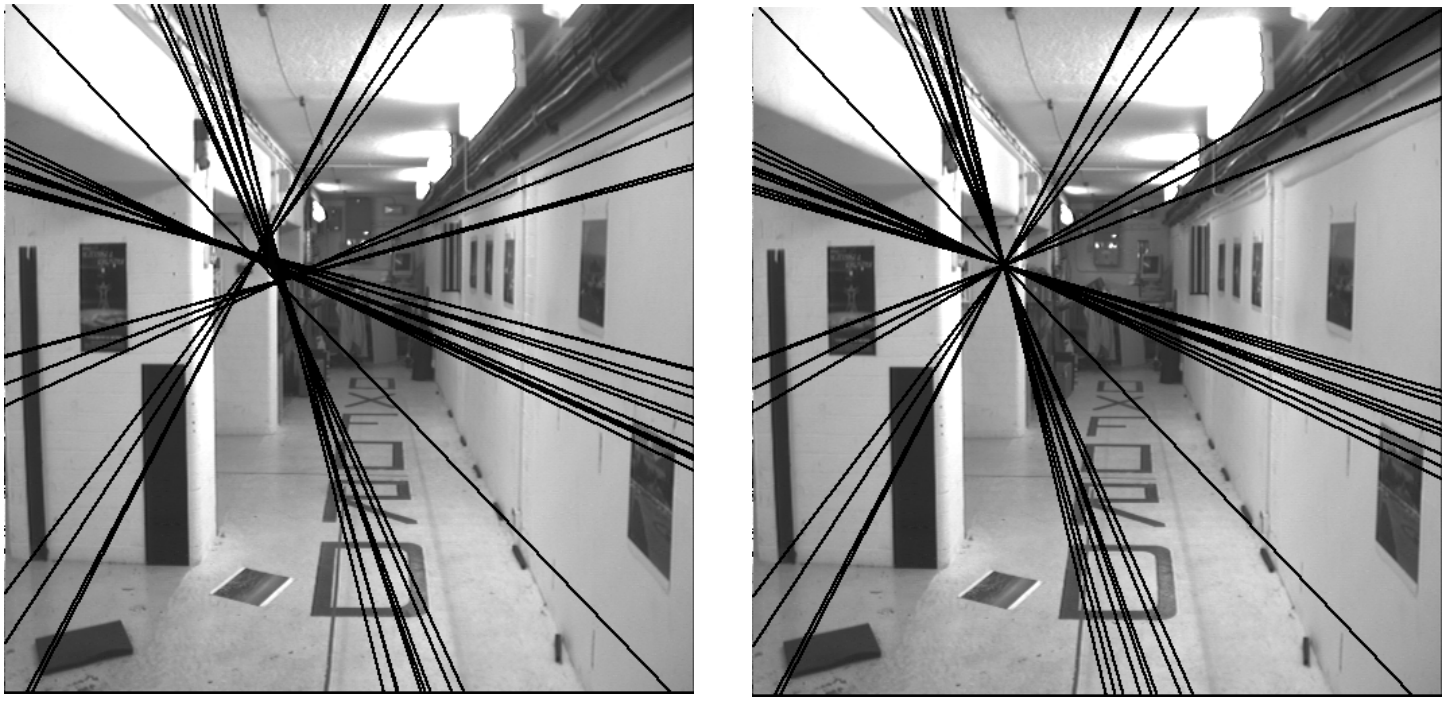


Fig. 11.1. **Epipolar lines.** *(a) the effect of a non-singular fundamental matrix. Epipolar lines computed as* $\mathbf{l}' = \mathbf{F}\mathbf{x}$ *for varying* $\mathbf{x}$ *do not meet in a common epipole. (b) the effect of enforcing singularity using the SVD method described here.*

the minimum case – 7 point correspondences

$$\begin{bmatrix} x'_1 x_1 & x'_1 y_1 & x'_1 & y'_1 x_1 & y'_1 y_1 & y'_1 & x_1 & y_1 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x'_7 x_7 & x'_7 y_7 & x'_7 & y'_7 x_7 & y'_7 y_7 & y'_7 & x_7 & y_7 & 1 \end{bmatrix} \mathrm{f} = 0$$

$$\mathrm{A} = \mathrm{U}_{7\mathrm{x}7}\mathrm{diag}(\sigma_1,\ldots,\sigma_7,0,0)\mathrm{V}_{9\mathrm{x}9}{}^{\mathrm{T}}$$

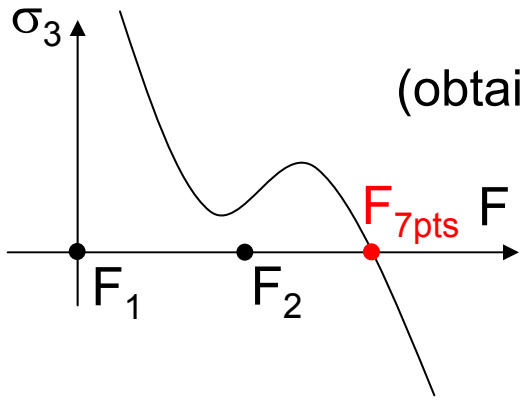
$$\Rightarrow \mathrm{A}[\mathrm{V}_8\,\mathrm{V}_9] = 0_{9\mathrm{x}2} \qquad (\mathrm{e.g.}\,\mathrm{V}^{\mathrm{T}}\mathrm{V}_8 = [000000010]^{\mathrm{T}})$$

$$\mathrm{x}_i{}^{\mathrm{T}}(\mathrm{F}_1 + \lambda\mathrm{F}_2)\mathrm{x}_i = 0, \forall i = 1\ldots7$$

one parameter family of solutions

but $F_1+\lambda F_2$ not automatically rank 2

the minimum case – impose rank 2



(obtain 1 or 3 solutions)

$$\det(F_1 + \lambda F_2) = a_3\lambda^3 + a_2\lambda^2 + a_1\lambda + a_0 = 0$$ (cubic equation)

$$\det(F_1 + \lambda F_2) = \det F_2 \det(F_2^{-1}F_1 + \lambda I) = 0$$

Compute possible λ as eigenvalues of $F_2^{-1}F_1$
(only real solutions are potential solutions)

the NOT normalized 8-point algorithm

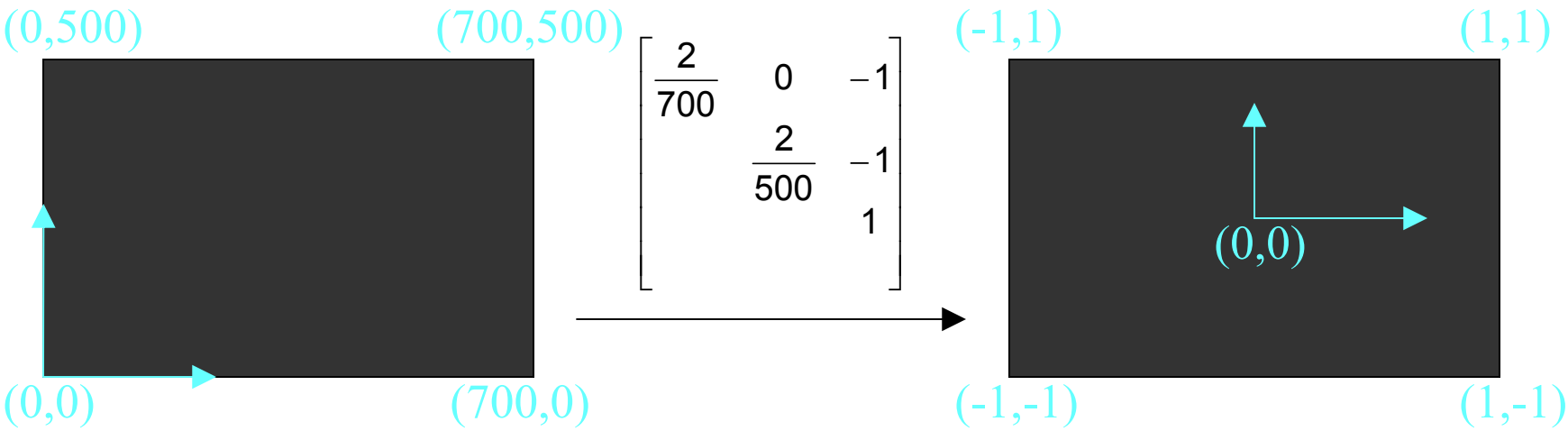
$$\begin{bmatrix} x_1x_1' & y_1x_1' & x_1' & x_1y_1' & y_1y_1' & y_1' & x_1 & y_1 & 1 \\ x_2x_2' & y_2x_2' & x_2' & x_2y_2' & y_2y_2' & y_2' & x_2 & y_2 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_nx_n' & y_nx_n' & x_n' & x_ny_n' & y_ny_n' & y_n' & x_n & y_n & 1 \end{bmatrix} \begin{bmatrix} f_{11} \\ f_{12} \\ f_{13} \\ f_{21} \\ f_{22} \\ f_{23} \\ f_{31} \\ f_{32} \\ f_{33} \end{bmatrix} = 0$$

~10000 ~10000 ~100 ~10000 ~10000 ~100 ~100 ~100 1

Orders of magnitude difference
Between column of data matrix
→ least-squares yields poor results

the normalized 8-point algorithm

Transform image to ~[-1,1]x[-1,1]



Least squares yields good results (Hartley, PAMI´97)

Objective

Given $n \geq 8$ image point correspondences $\{\mathbf{x}_i \leftrightarrow \mathbf{x}'_i\}$, determine the fundamental matrix $\mathtt{F}$ such that $\mathbf{x}'^{\mathsf{T}}_i \mathtt{F} \mathbf{x}_i = 0$.

Algorithm

(i) **Normalization:** Transform the image coordinates according to $\hat{\mathbf{x}}_i = \mathtt{T}\mathbf{x}_i$ and $\hat{\mathbf{x}}'_i = \mathtt{T}'\mathbf{x}'_i$, where $\mathtt{T}$ and $\mathtt{T}'$ are normalizing transformations consisting of a translation and scaling.

(ii) Find the fundamental matrix $\hat{\mathtt{F}}'$ corresponding to the matches $\hat{\mathbf{x}}_i \leftrightarrow \hat{\mathbf{x}}'_i$ by

  (a) **Linear solution:** Determine $\hat{\mathtt{F}}$ from the singular vector corresponding to the smallest singular value of $\hat{\mathtt{A}}$, where $\hat{\mathtt{A}}$ is composed from the matches $\hat{\mathbf{x}}_i \leftrightarrow \hat{\mathbf{x}}'_i$ as defined in (11.3).

  (b) **Constraint enforcement:** Replace $\hat{\mathtt{F}}$ by $\hat{\mathtt{F}}'$ such that $\det \hat{\mathtt{F}}' = 0$ using the SVD (see section 11.1.1).

(iii) **Denormalization:** Set $\mathtt{F} = \mathtt{T}'^{\mathsf{T}} \hat{\mathtt{F}}' \mathtt{T}$. Matrix $\mathtt{F}$ is the fundamental matrix corresponding to the original data $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$.

Algorithm 11.1. *The normalized 8-point algorithm for* $\mathtt{F}$.

algebraic minimization

possible to iteratively minimize algebraic distance subject to det F=0 (see book if interested)

Objective

Find the fundamental matrix $\mathtt{F}$ that minimizes the algebraic error $\|\mathtt{A}\mathbf{f}\|$ subject to $\|\mathbf{f}\| = 1$ and $\det \mathtt{F} = 0$.

Algorithm

(i) Find a first approximation $\mathtt{F}_0$ for the fundamental matrix using the normalized 8-point algorithm 11.1. Then find the right null-vector $\mathbf{e}_0$ of $\mathtt{F}_0$.

(ii) Starting with the estimate $\mathbf{e}_i = \mathbf{e}_0$ for the epipole, compute the matrix $\mathtt{E}_i$ according to (11.4), then find the vector $\mathbf{f}_i = \mathtt{E}_i\mathbf{m}_i$ that minimizes $\|\mathtt{A}\mathbf{f}_i\|$ subject to $\|\mathbf{f}_i\| = 1$. This is done using algorithm A5.6(*p*595).

(iii) Compute the algebraic error $\boldsymbol{\epsilon}_i = \mathtt{A}\mathbf{f}_i$. Since $\mathbf{f}_i$ and hence $\boldsymbol{\epsilon}_i$ is defined only up to sign, correct the sign of $\boldsymbol{\epsilon}_i$ (multiplying by minus 1 if necessary) so that $\mathbf{e}_i^{\mathsf{T}}\mathbf{e}_{i-1} > 0$ for $i > 0$. This is done to ensure that $\boldsymbol{\epsilon}_i$ varies smoothly as a function of $\mathbf{e}_i$.

(iv) The previous two steps define a mapping $\mathbb{R}^3 \rightarrow \mathbb{R}^9$ mapping $\mathbf{e}_i \mapsto \boldsymbol{\epsilon}_i$. Now use the Levenberg–Marquardt algorithm (section A6.2(*p*600)) to vary $\mathbf{e}_i$ iteratively so as to minimize $\|\boldsymbol{\epsilon}_i\|$.

(v) Upon convergence, $\mathbf{f}_i$ represents the desired fundamental matrix.

Algorithm 11.2. *Computation of* $\mathtt{F}$ *with* $\det \mathtt{F} = 0$ *by iteratively minimizing algebraic error.*

Geometric distance

Gold standard

Sampson error

Symmetric epipolar distance

Gold standard

Maximum Likelihood Estimation (= least-squares for Gaussian noise)

$$\sum_i d\left(\mathrm{x}_i, \hat{\mathrm{x}}_i\right)^2 + d\left(\mathrm{x}'_i, \hat{\mathrm{x}}'_i\right)^2 \quad \text{subject to } \hat{\mathrm{x}}'^{\mathrm{T}} \mathrm{F}\hat{\mathrm{x}} = 0$$

Initialize: normalized 8-point, (P,P‘) from F, reconstruct X_i

Parameterize:

$$\mathrm{P} = [\mathrm{I} \mid 0], \mathrm{P}' = [\mathrm{M} \mid \mathrm{t}], \mathrm{X}_i$$

(overparametrized)

$$\hat{\mathrm{x}}_i = \mathrm{P}\mathrm{X}_i, \hat{\mathrm{x}}_i = \mathrm{P}'\mathrm{X}_i$$

Minimize cost using Levenberg-Marquardt
(preferably sparse LM, see book)

Objective

Given $n \geq 8$ image point correspondences $\{\mathbf{x}_i \leftrightarrow \mathbf{x}'_i\}$, determine the Maximum Likelihood estimate $\hat{\mathtt{F}}$ of the fundamental matrix.

The MLE involves also solving for a set of subsidiary point correspondences $\{\hat{\mathbf{x}}_i \leftrightarrow \hat{\mathbf{x}}'_i\}$, such that $\hat{\mathbf{x}}_i'^{\mathsf{T}}\hat{\mathtt{F}}\hat{\mathbf{x}}_i = 0$, and which minimizes

$$\sum_i d(\mathbf{x}_i, \hat{\mathbf{x}}_i)^2 + d(\mathbf{x}'_i, \hat{\mathbf{x}}'_i)^2.$$

Algorithm

(i) Compute an initial rank 2 estimate of $\hat{\mathtt{F}}$ using a linear algorithm such as algorithm 11.1.

(ii) Compute an initial estimate of the subsidiary variables $\{\hat{\mathbf{x}}_i, \hat{\mathbf{x}}'_i\}$ as follows:

  (a) Choose camera matrices $\mathtt{P} = [\mathtt{I} \mid \mathbf{0}]$ and $\mathtt{P}' = [[\mathbf{e}']_\times \hat{\mathtt{F}} \mid \mathbf{e}']$, where $\mathbf{e}'$ is obtained from $\hat{\mathtt{F}}$.

  (b) From the correspondence $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$ and $\hat{\mathtt{F}}$ determine an estimate of $\widehat{\mathbf{X}}_i$ using the triangulation method of chapter 12.

  (c) The correspondence consistent with $\hat{\mathtt{F}}$ is obtained as $\hat{\mathbf{x}}_i = \mathtt{P}\widehat{\mathbf{X}}_i$, $\hat{\mathbf{x}}'_i = \mathtt{P}'\widehat{\mathbf{X}}_i$.

(iii) Minimize the cost

$$\sum_i d(\mathbf{x}_i, \hat{\mathbf{x}}_i)^2 + d(\mathbf{x}'_i, \hat{\mathbf{x}}'_i)^2$$

over $\hat{\mathtt{F}}$ and $\widehat{\mathbf{X}}_i$, $i = 1, \ldots, n$. The cost is minimized using the Levenberg–Marquardt algorithm over $3n + 12$ variables: $3n$ for the n 3D points $\widehat{\mathbf{X}}_i$, and 12 for the camera matrix $\mathtt{P}' = [\mathtt{M} \mid \mathbf{t}]$, with $\hat{\mathtt{F}} = [\mathbf{t}]_\times \mathtt{M}$, and $\hat{\mathbf{x}}_i = \mathtt{P}\widehat{\mathbf{X}}_i$, $\hat{\mathbf{x}}'_i = \mathtt{P}'\widehat{\mathbf{X}}_i$.

Algorithm 11.3. *The Gold Standard algorithm for estimating* $\mathtt{F}$ *from image correspondences.*

Gold standard

Alternative, minimal parametrization (with a=1)

$$\mathbf{F} = \begin{bmatrix} a & b & -ax - by \\ c & d & -cx - dy \\ -ax' - cy' & -bx' - dy' & F_{33} \end{bmatrix}$$

$$\text{with} \quad F_{33} = (ax + by)x' + (cx + dy)y'$$

(note (x,y,1) and (x‘,y‘,1) are epipoles)

problems:

- a=0 $\rightarrow$ pick largest of a,b,c,d to fix
- epipole at infinity

  $\rightarrow$ pick largest of x,y,w and of x’,y’,w’

4x3x3=36 parametrizations!

reparametrize at every iteration, to be sure

Zhang&Loop's approach CVIU'01

$$\tilde{\tilde{\mathbf{m}}}_i = \mathbf{P}\tilde{\mathbf{m}}_i \quad \text{and} \quad \tilde{\tilde{\mathbf{m}}}'_i = \mathbf{P}'\tilde{\mathbf{m}}'_i \,, \tag{5}$$

then the fundamental matrix in the transformed space is given by

$$\hat{\tilde{\mathbf{F}}} = \mathbf{P}'^{-T}\mathbf{F}\mathbf{P}^{-1} \,. \tag{6}$$

Given an initial estimate of matrix $\mathbf{F_0}$ obtained for example with Hartley's normalized 8-point algorithm [3], we compute the epipoles $\tilde{\mathbf{e}}_0$ and $\tilde{\mathbf{e}}'_0$. The matrices $\mathbf{P}$ and $\mathbf{P}'$ are 3×3 permutation matrices determined as follows:

1. Initialize $\mathbf{P}$ and $\mathbf{P}'$ to be identity matrices.
2. Find the position of the largest element of $\mathbf{F}_0$, denoted by (i_0, j_0). (Index of a vector or a matrix starts with 0 as in C++)
3. If $j_0 \neq 0$, permute rows 0 and j_0 of matrix $\mathbf{P}$ and permute elements 0 and j_0 of epipole $\tilde{\mathbf{e}}_0$.
4. If $i_0 \neq 0$, permute rows 0 and i_0 of matrix $\mathbf{P}'$ and permute elements 0 and i_0 of epipole $\tilde{\mathbf{e}}'_0$.
5. If $|\tilde{\mathbf{e}}_0[1]| > |\tilde{\mathbf{e}}_0[2]|$, permute elements 1 and 2 of epipole $\tilde{\mathbf{e}}_0$ and permute rows 1 and 2 of matrix $\mathbf{P}$.
6. If $|\tilde{\mathbf{e}}'_0[1]| > |\tilde{\mathbf{e}}'_0[2]|$, permute elements 1 and 2 of epipole $\tilde{\mathbf{e}}'_0$ and permute rows 1 and 2 of matrix $\mathbf{P}'$.

First-order geometric error (Sampson error)

$$\sum e^{\mathrm{T}}\left(\mathrm{J}\mathrm{J}^{\mathrm{T}}\right)^{-1}e \qquad \sum\frac{e^{\mathrm{T}}e}{\mathrm{J}\mathrm{J}^{\mathrm{T}}}$$

(one eq./point
$\Rightarrow \mathrm{J}\mathrm{J}^{\mathrm{T}}$ scalar)

$$e=\sum \mathrm{x}'^{\mathrm{T}}\,\mathrm{F}\mathrm{x}=0 \qquad \frac{\partial e}{\partial x_i}=\mathrm{x}'^{\mathrm{T}}\,\mathrm{F}\begin{bmatrix}1\\0\\0\end{bmatrix}$$

$$\mathrm{J}\mathrm{J}^{\mathrm{T}}=\left(\mathrm{x}'^{\mathrm{T}}\,\mathrm{F}\right)_1^2+\left(\mathrm{x}'^{\mathrm{T}}\,\mathrm{F}\right)_2^2+\left(\mathrm{F}\mathrm{x}\right)_1^2+\left(\mathrm{F}\mathrm{x}\right)_2^2$$

$$\sum\frac{e^{\mathrm{T}}e}{\mathrm{J}\mathrm{J}^{\mathrm{T}}} \qquad \sum\frac{\mathrm{x}'^{\mathrm{T}}\,\mathrm{F}\mathrm{x}}{\left(\mathrm{x}'^{\mathrm{T}}\,\mathrm{F}\right)_1^2+\left(\mathrm{x}'^{\mathrm{T}}\,\mathrm{F}\right)_2^2+\left(\mathrm{F}\mathrm{x}\right)_1^2+\left(\mathrm{F}\mathrm{x}\right)_2^2}$$

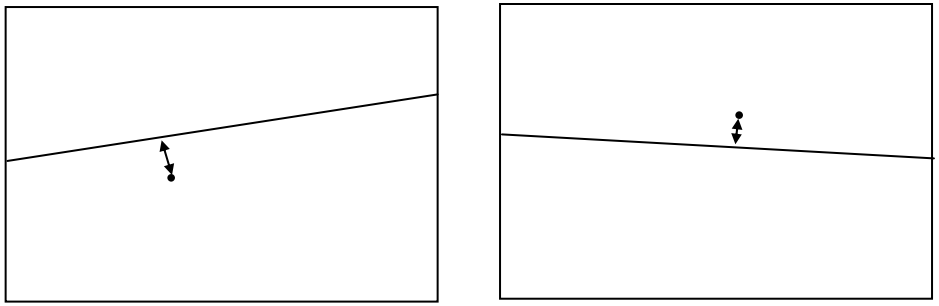
(problem if some x is located at epipole)

advantage: no subsidiary variables required

Symmetric epipolar error

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

$$= \sum \mathrm{x}'^{\mathrm{T}}\,\mathrm{Fx}\left(\frac{1}{\left(\mathrm{x}'^{\mathrm{T}}\,\mathrm{F}\right)_1^2 + \left(\mathrm{x}'^{\mathrm{T}}\,\mathrm{F}\right)_2^2} + \frac{1}{\left(\mathrm{Fx}\right)_1^2 + \left(\mathrm{Fx}\right)_2^2}\right)$$



Some experiments:

Fig. 11.2. *Image pairs used for the algorithm comparison. In the top two the epipoles are far from the image centres. In the middle two the epipoles are close (Grenoble) and in the image (Corridor). For the calibration images the matched points are known extremely accurately.*

Some experiments:

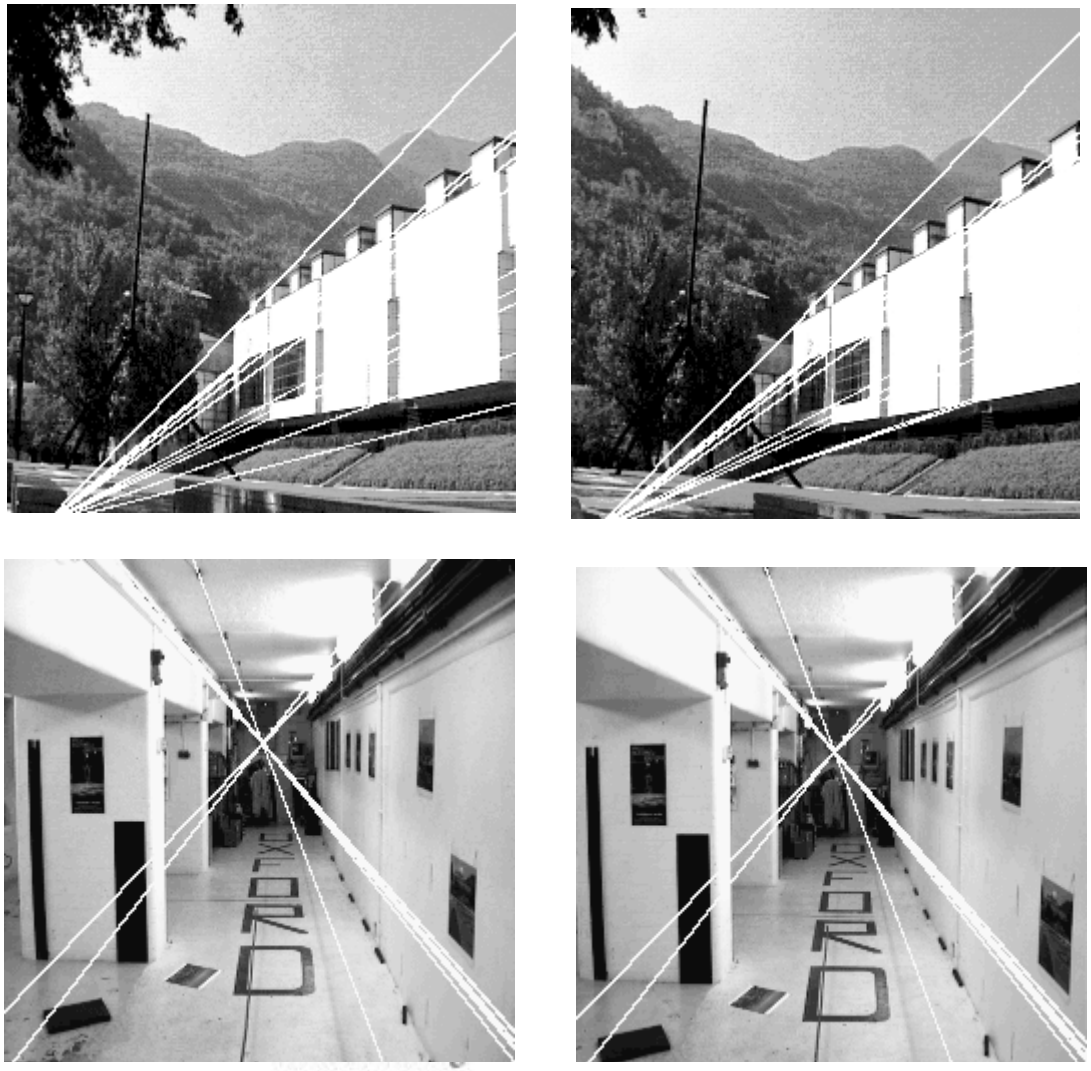


Fig. 11.2. *Image pairs used for the algorithm comparison. In the top two the epipoles are far from the image centres. In the middle two the epipoles are close (Grenoble) and in the image (Corridor). For the calibration images the matched points are known extremely accurately.*

Some experiments:

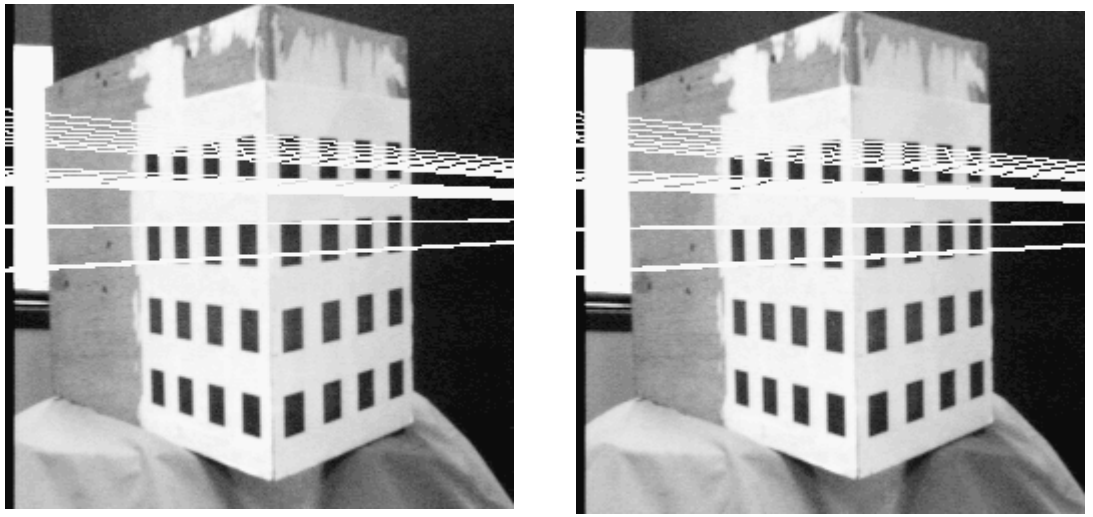
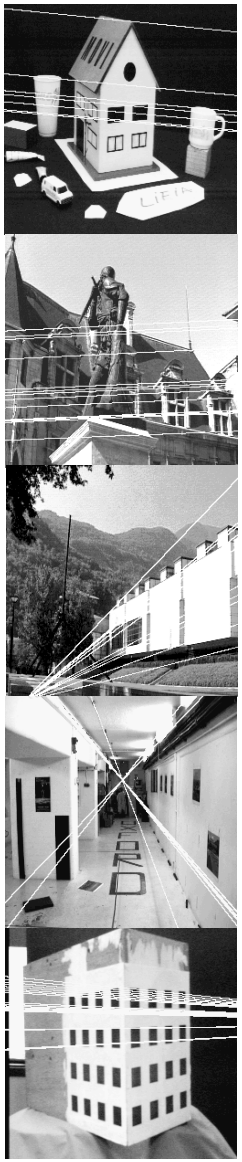
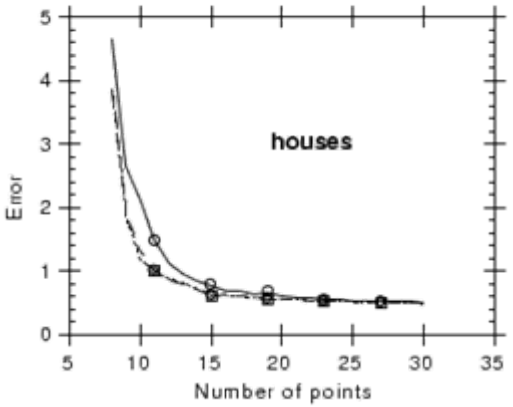
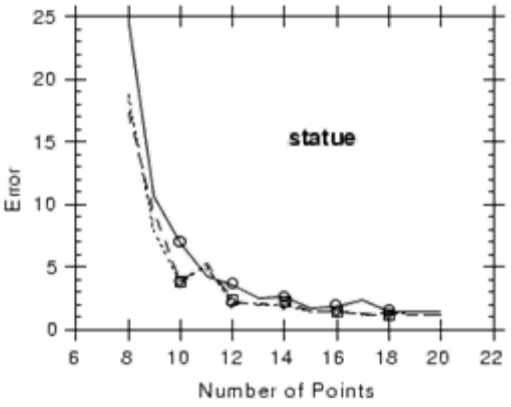


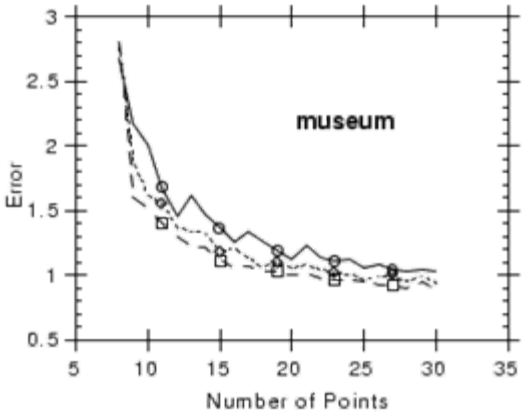
Fig. 11.2. *Image pairs used for the algorithm comparison. In the top two the epipoles are far from the image centres. In the middle two the epipoles are close (Grenoble) and in the image (Corridor). For the calibration images the matched points are known extremely accurately.*

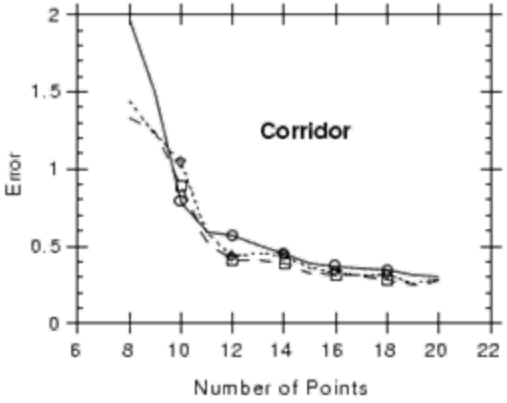
Some experiments:

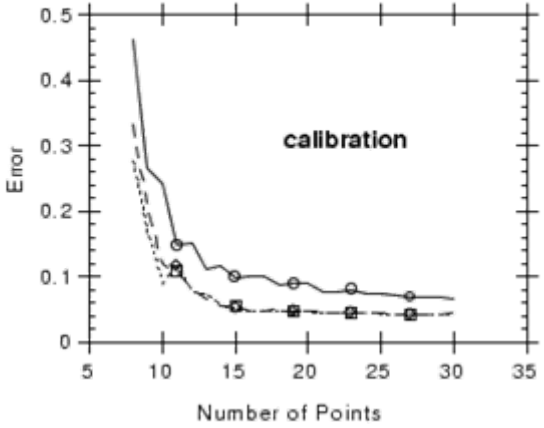












Residual error: $\sum d(\mathrm{x}'_i, F\mathrm{x}_i)^2 + d(\mathrm{x}_i, \mathrm{F}^{\mathrm{T}}\mathrm{x}'_i)^2$ (for all points!)

Fig. 11.3. **Results of the experimental evaluation of the algorithms.** *In each case, three methods of computing* F *are compared. Residual error is plotted against the number of points used to compute* F. *In each graph, the top (solid line) shows the results of the normalized 8-point algorithm. Also shown are the results of minimizing geometric error (long dashed line) and iteratively minimizing algebraic error subject to the determinant constraint (short dashed line). In most cases, the result of iteratively minimizing algebraic error is almost indistinguishable from minimizing geometric error. Both are noticeably better than the non-iterative normalized 8-point algorithm, though that algorithm also gives good results.*

Recommendations:

1. Do not use unnormalized algorithms

2. Quick and easy to implement: 8-point normalized

3. Better: enforce rank-2 constraint during minimization

4. Best: Maximum Likelihood Estimation
   (minimal parameterization, sparse implementation)

Automatic computation of F

(i) Interest points
(ii) Putative correspondences
(iii) RANSAC
(iv) Non-linear re-estimation of F
(v) Guided matching
(repeat (iv) and (v) until stable)

Feature points

- Extract feature points to relate images
- Required properties:
  - Well-defined

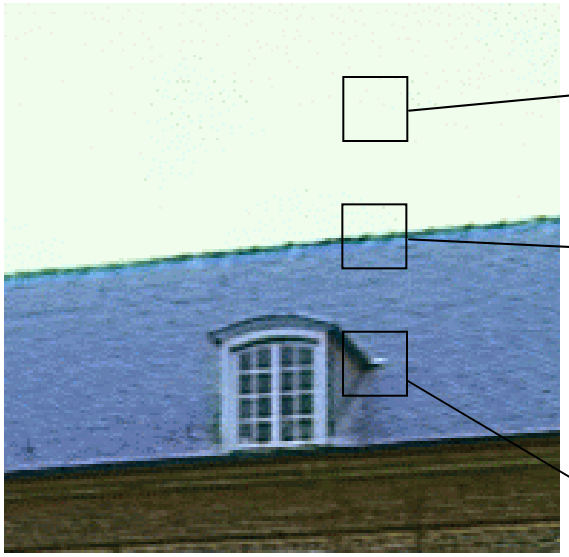
    (i.e. neigboring points should all be different)
  - Stable across views

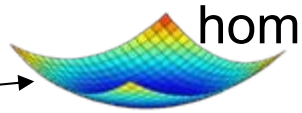
    (i.e. same 3D point should be extracted as feature for neighboring viewpoints)

Feature points

(e.g.Harris&Stephens´88; Shi&Tomasi´94)

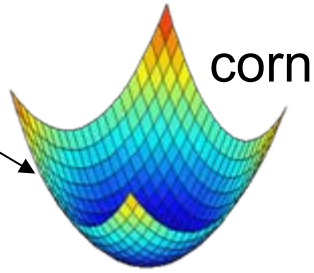
Find points that differ as much as possible from all neighboring points





homogeneous

edge



corner

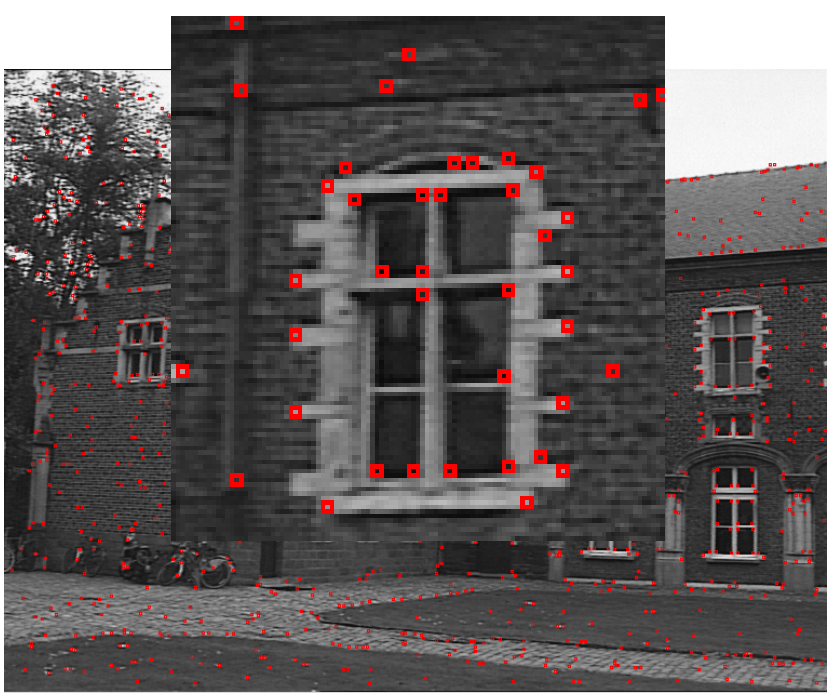
$$SSD \approx \Delta^{\mathsf{T}} \mathbf{M} \Delta$$

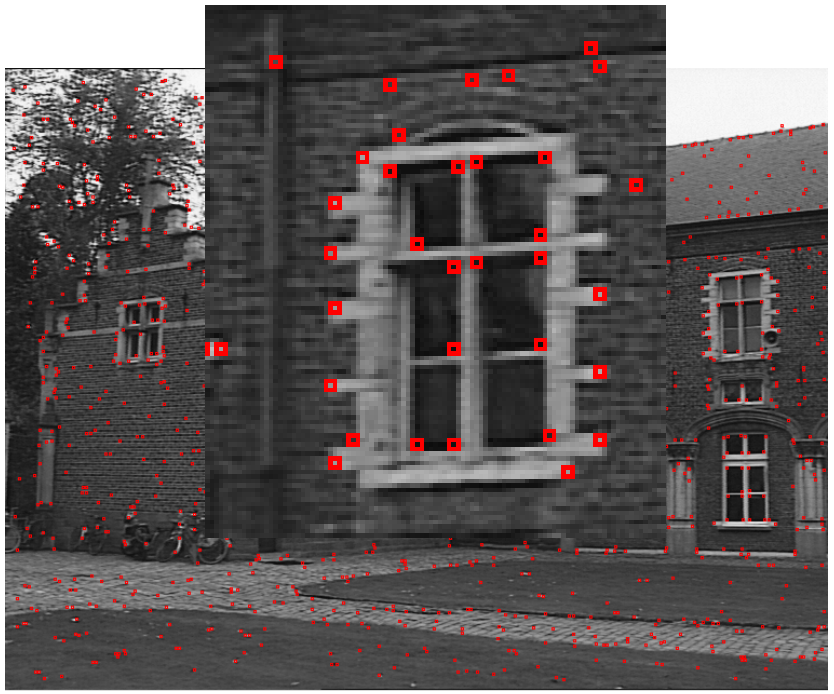
$$\mathbf{M} = \iint_W \begin{bmatrix} \frac{\partial I}{\partial x} \\ \frac{\partial I}{\partial y} \end{bmatrix} \begin{bmatrix} \frac{\partial I}{\partial x} & \frac{\partial I}{\partial y} \end{bmatrix} w(x, y)\, dx dy$$

$\mathbf{M}$ should have large eigenvalues

Feature = local maxima (subpixel) of $F(\lambda_1, \lambda_2)$

Feature points



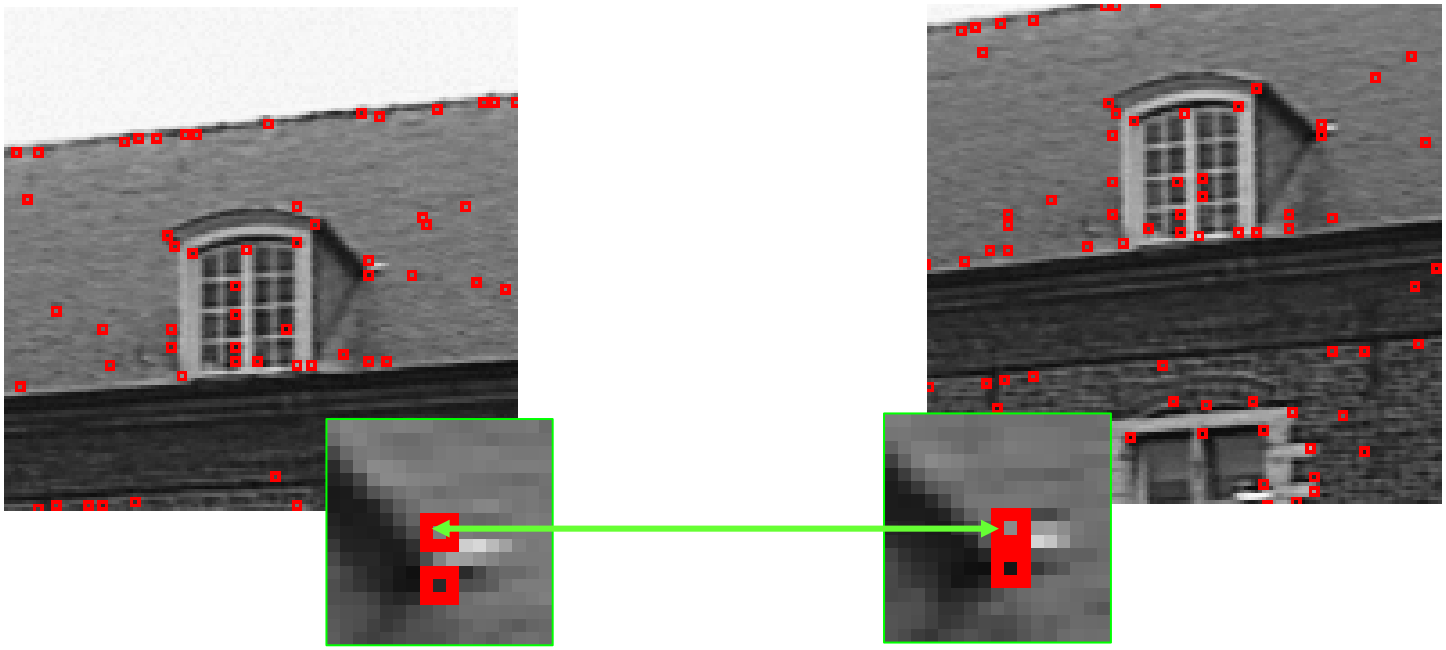


Select strongest features **(e.g. 1000/image)**

Feature matching

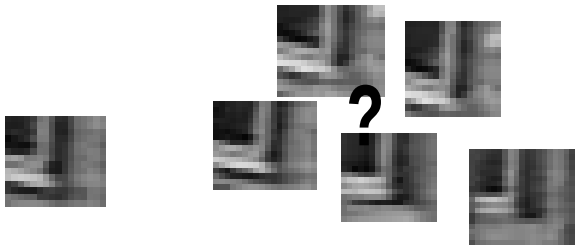
Evaluate NCC for all features with similar coordinates

$$\text{e.g.}\,(x', y') \in \left[x - \tfrac{w}{10}, x + \tfrac{w}{10}\right] \times \left[y - \tfrac{h}{10}, y + \tfrac{h}{10}\right]$$

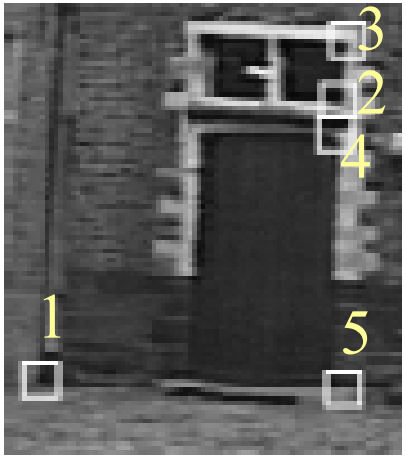


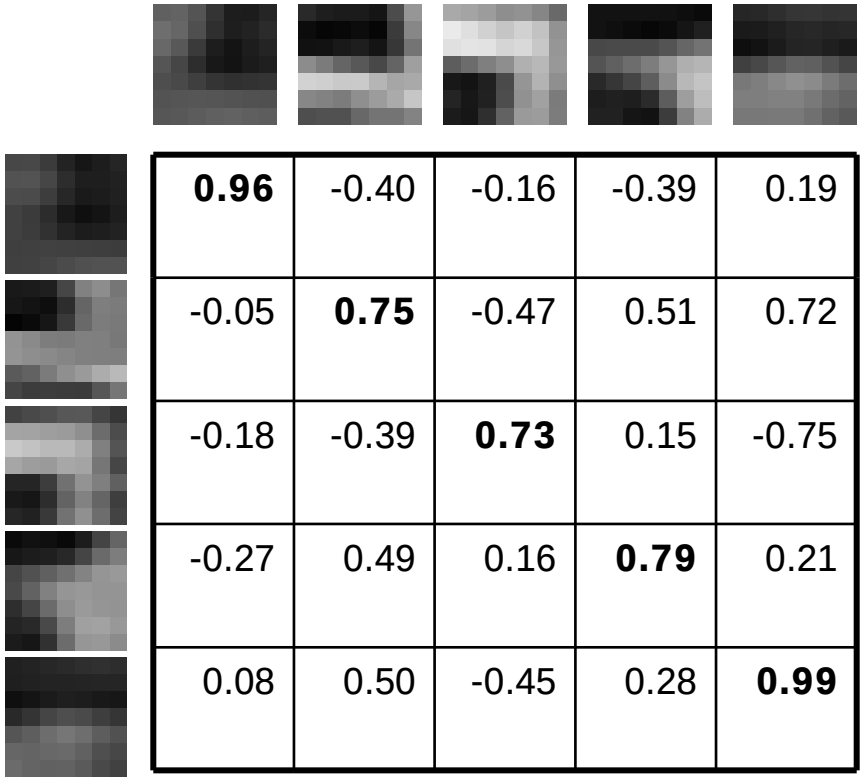
Keep mutual best matches

Still many wrong matches!

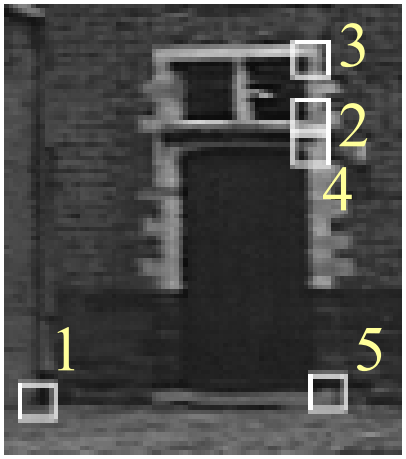


Feature example





0.96	-0.40	-0.16	-0.39	0.19
-0.05	**0.75**	-0.47	0.51	0.72
-0.18	-0.39	**0.73**	0.15	-0.75
-0.27	0.49	0.16	**0.79**	0.21
0.08	0.50	-0.45	0.28	**0.99**



Gives satisfying results
for small image motions

Wide-baseline matching…

- Requirement to cope with larger variations between images
  - Translation, rotation, scaling
  - Foreshortening

  } geometric transformations

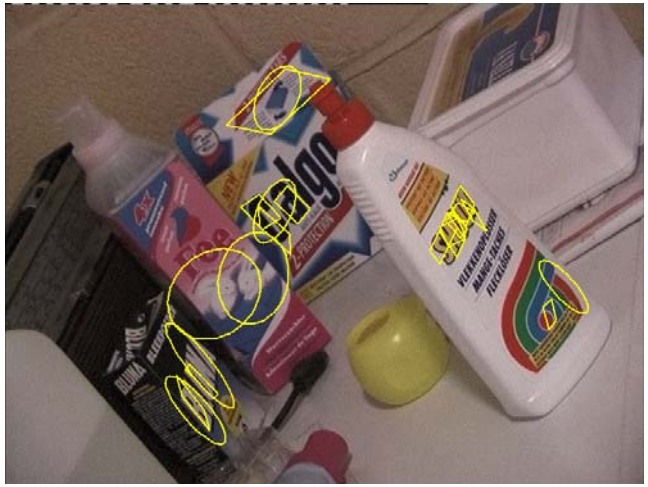
  - Non-diffuse reflections
  - Illumination

  } photometric changes

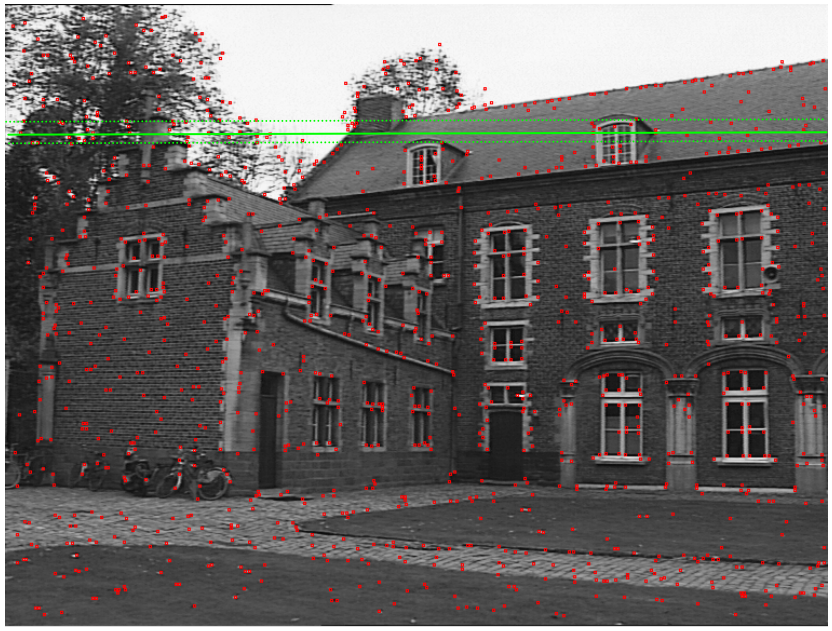
Wide-baseline matching…

(Tuytelaars and Van Gool BMVC 2000)

Wide baseline matching for two different region types



Finding more matches



restrict search range to neighborhood of epipolar line
(±1.5 pixels)
relax disparity restriction (along epipolar line)

RANSAC

Step 1. Extract features

Step 2. Compute a set of potential matches

Step 3. do

- Step 3.1 select minimal sample (i.e. 7 matches)
- Step 3.2 compute solution(s) for F
- Step 3.3 determine inliers (verify hypothesis)

} (generate hypothesis)

until Γ(*#inliers*,*#samples*)<95%

Step 4. Compute F based on all inliers

Step 5. Look for additional matches

Step 6. Refine F based on all correct matches

$$\Gamma = 1 - (1 - \left(\frac{\#inliers}{\#matches}\right)^7)^{\#samples}$$

#inliers	90%	80%	70%	60%	50%
#samples	5	13	35	106	382

How many samples?

Choose N so that, with probability p, at least one random sample is free from outliers. e.g. p=0.99

$$\left(1-\left(1-e\right)^{s}\right)^{N}=1-p$$

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

	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

Objective Compute the fundamental matrix between two images.

Algorithm

(i) **Interest points:** Compute interest points in each image.
(ii) **Putative correspondences:** Compute a set of interest point matches based on proximity and similarity of their intensity neighbourhood.
(iii) **RANSAC robust estimation:** Repeat for N samples, where N is determined adaptively as in algorithm 4.5(p121):
  (a) Select a random sample of 7 correspondences and compute the fundamental matrix F as described in section 11.1.2. There will be one or three real solutions.
  (b) Calculate the distance $d_\perp$ for each putative correspondence.
  (c) Compute the number of inliers consistent with F by the number of correspondences for which $d_\perp < t$ pixels.
  (d) If there are three real solutions for F the number of inliers is computed for each solution, and the solution with most inliers retained.

  Choose the F with the largest number of inliers. In the case of ties choose the solution that has the lowest standard deviation of inliers.
(iv) **Non-linear estimation:** re-estimate F from all correspondences classified as inliers by minimizing a cost function, e.g. (11.6), using the Levenberg–Marquardt algorithm of section A6.2(p600).
(v) **Guided matching:** Further interest point correspondences are now determined using the estimated F to define a search strip about the epipolar line.

The last two steps can be iterated until the number of correspondences is stable.

Algorithm 11.4. *Algorithm to automatically estimate the fundamental matrix between two images using RANSAC.*

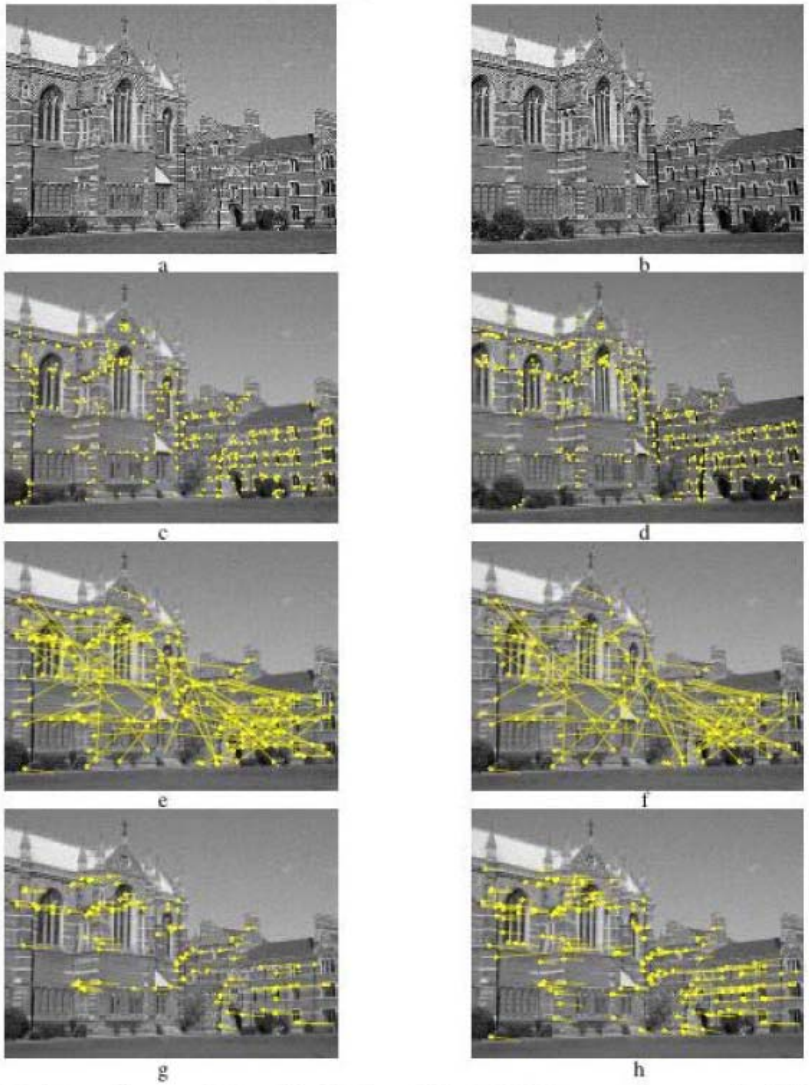
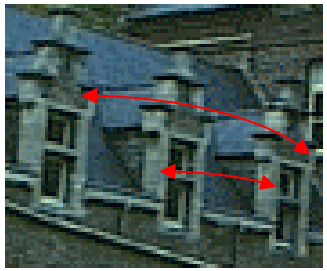
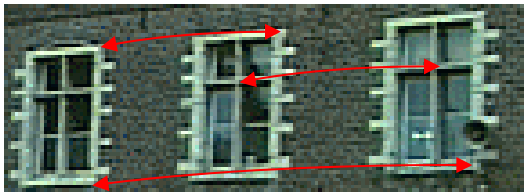


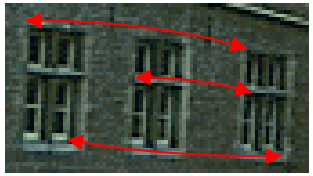
Fig. 11.4. **Automatic computation of the fundamental matrix between two images using RANSAC.** *(a) (b) left and right images of Keble College, Oxford. The motion between views is a translation and rotation. The images are* 640×480 *pixels. (c) (d) detected corners superimposed on the images. There are approximately 500 corners on each image. The following results are superimposed on the left image: (e) 188 putative matches shown by the line linking corners, note the clear mismatches; (f) outliers – 89 of the putative matches. (g) inliers – 99 correspondences consistent with the estimated* $\mathbf{F}$*; (h) final set of 157 correspondences after guided matching and MLE. There are still a few mismatches evident, e.g. the long line on the left.*

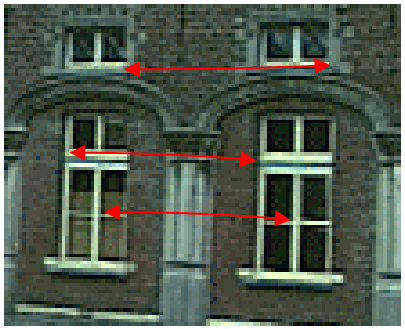
More problems:

- Absence of sufficient features (no texture)
- Repeated structure ambiguity





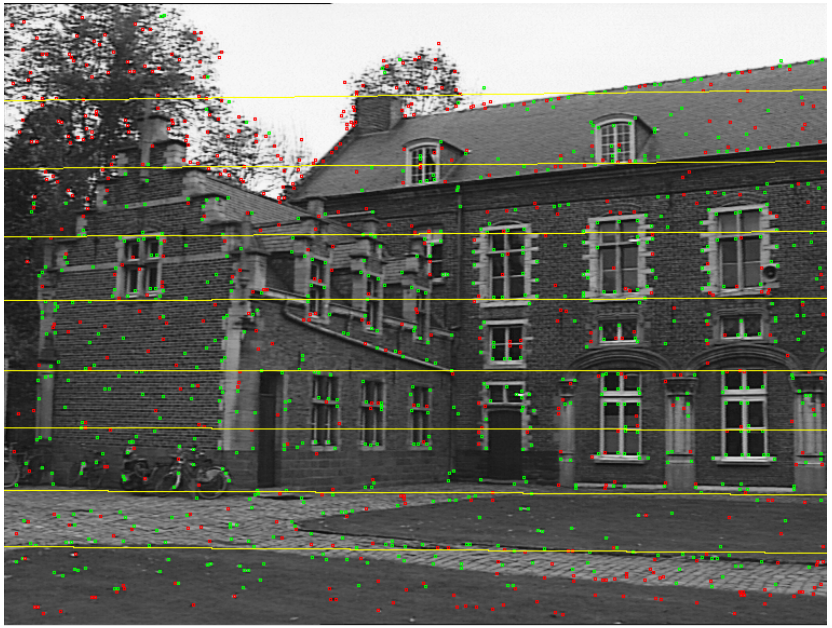




- Robust matcher also finds support for wrong hypothesis
- solution: detect repetition

(Schaffalitzky and Zisserman, BMVC'98)

two-view geometry



geometric relations between two views is fully described by recovered 3x3 matrix F

Special case:

Enforce constraints for optimal results:

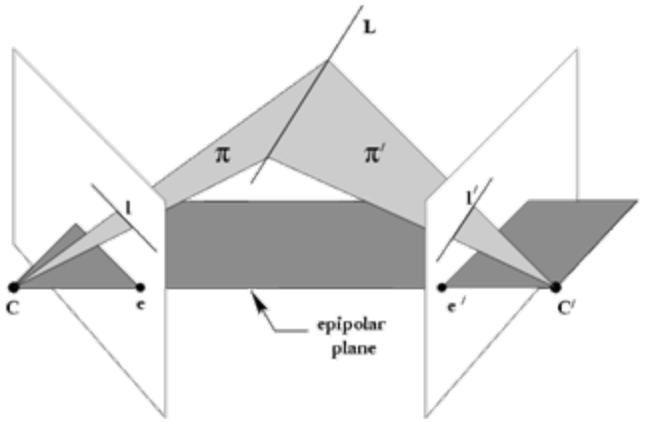
- Pure translation (2dof),
- Planar motion (6dof),
- Calibrated case (5dof) ➔ compute essential matrix; but for E, two singular values are equal.

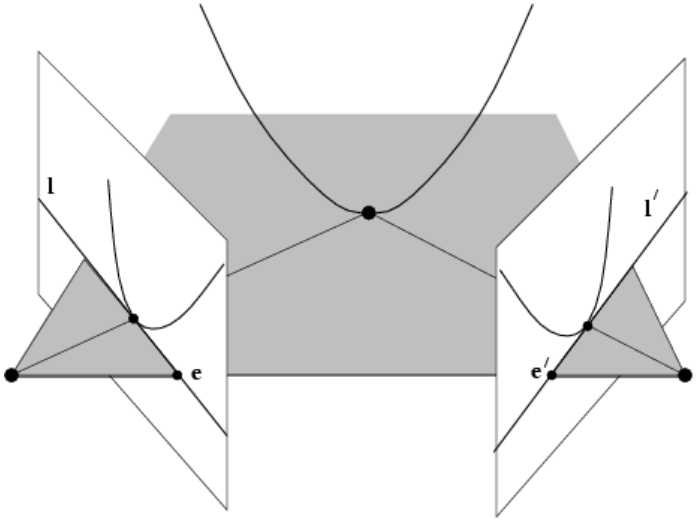
Result 11.1. *Let* $\mathtt{E}$ *be a* 3×3 *matrix with SVD given by* $\mathtt{E} = \mathtt{U}\mathtt{D}\mathtt{V}^{\mathsf{T}}$, *where* $\mathtt{D} = \mathrm{diag}(a, b, c)$ *with* $a \geq b \geq c$. *Then the closest essential matrix to* $\mathtt{E}$ *in Frobenius norm is given by* $\hat{\mathtt{E}} = \mathtt{U}\hat{\mathtt{D}}\mathtt{V}^{\mathsf{T}}$, *where* $\hat{\mathtt{D}} = \mathrm{diag}((a+b)/2, (a+b)/2, 0)$.

Other entities?

Lines give no constraint for two view geometry (but will for three and more views)

Curves and surfaces yield some constraints related to tangency





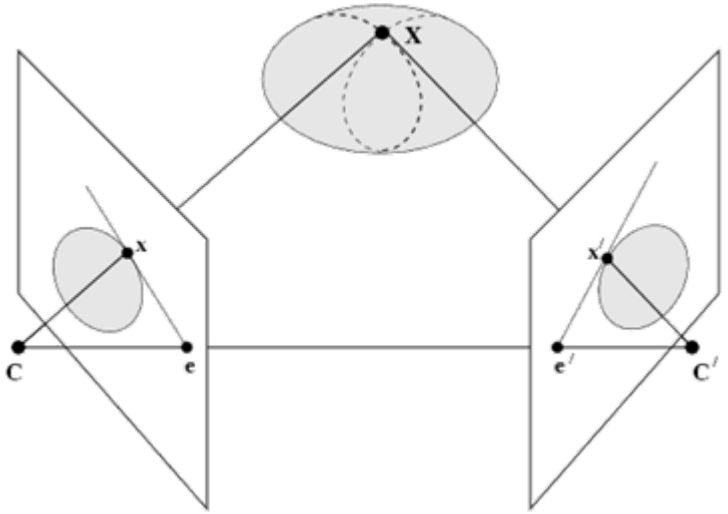


Fig. 11.6. **Epipolar tangency.** *(a) for a surface; (b) for a space curve – figure after Porrill and Pollard [Porrill-91]. In (a) the epipolar plane* $\mathbf{CC'X}$ *is tangent to the surface at* $\mathbf{X}$*. The imaged outline is tangent to the epipolar lines at* $\mathbf{x}$ *and* $\mathbf{x'}$ *in the two views. The dashed curves on the surface are the contour generators. In (b) the epipolar plane is tangent to the space curve. The corresponding epipolar lines* $\mathbf{l} \leftrightarrow \mathbf{l'}$ *are tangent to the imaged curve.*

Degenerate cases:

- Degenerate cases
  - Planar scene
  - Pure rotation
- No unique solution
  - Remaining DOF filled by noise
  - Use simpler model (e.g. homography)
- Model selection (Torr et al., ICCV´98, Kanatani, Akaike)
  - Compare H and F according to expected residual error (compensate for model complexity)

$\dim(N) = 1$: Unique solution – no degeneracy.

Arises from $n \geq 8$ point correspondences in general position. If $n > 8$ then the point correspondences must be perfect (i.e. noise-free).

$\dim(N) = 2$: 1 or 3 solutions.

Arises in the case of seven point correspondences, and also in the case of $n > 7$ perfect point correspondences where the 3D points and camera centres lie on a ruled quadric referred to as a critical surface. The quadric may be non-degenerate (a hyperboloid of one sheet) or degenerate.

$\dim(N) = 3$: Two-parameter family of solutions.

Arises if $n \geq 6$ perfect point correspondences are related by a homography, $\mathbf{x}'_i = \mathtt{H}\mathbf{x}_i$.

- Rotation about the camera centre (a degenerate motion).
- All world points on a plane (a degenerate structure).

Table 11.1. *Degeneracies in estimating* F *from point correspondences, classified by the dimension of th null-space* N *of* A *in (11.3–p279).*

The envelope of epipolar lines

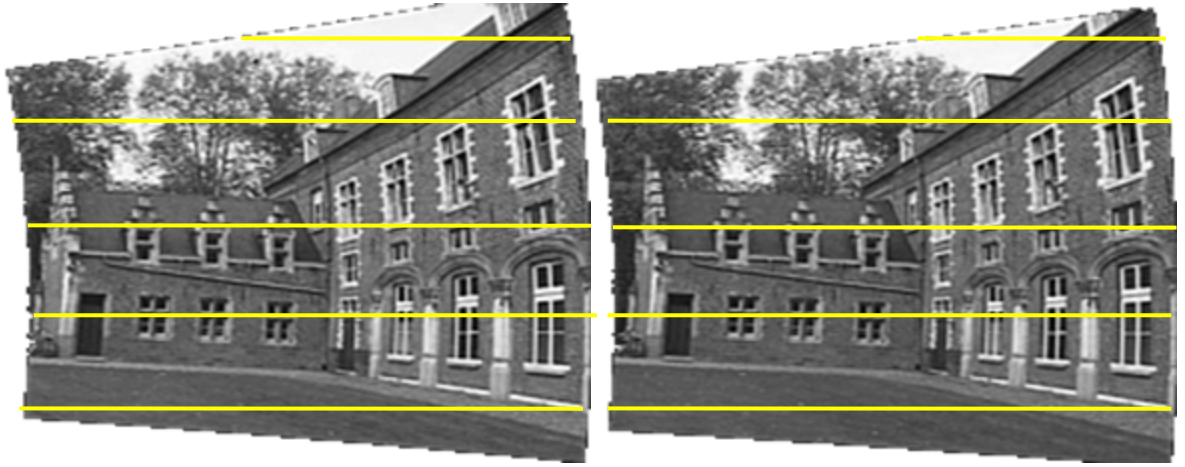
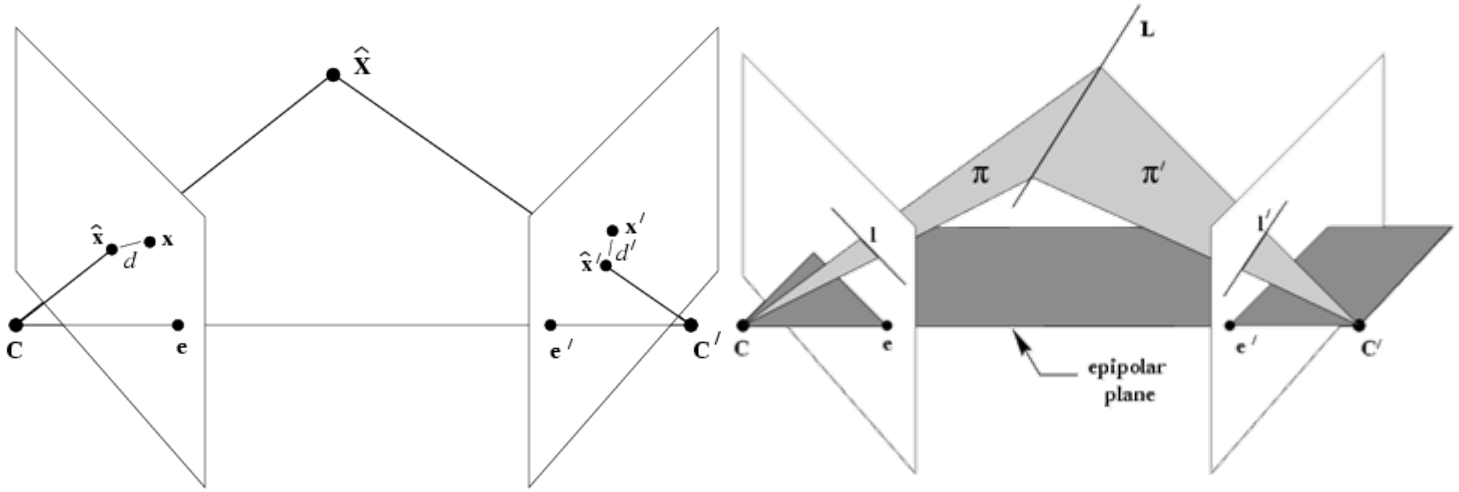
What happens to an epipolar line if there is noise?

Monte Carlo

Fig. 11.9. *(a) The point in the first image used to compute the epipolar envelopes in the second images. Note the ambiguity of which point is to be found in the second image. The marked point may represent the point on the statue's leg (foreground) or the point on the building behind the statue (background). In the second image, these two points are quite separate, and the epipolar line must pass through them both. (b) Computed corresponding epipolar lines computed from* $n = 15$ *point matches. The different lines correspond to different instances of injected noise in the matched points. Gaussian noise of 0.5 pixels in each coordinate was added to the ideal matched point positions before computing the epipolar line corresponding to the selected point. The ML estimator (Gold Standard algorithm) was used to compute* F*. This experiment shows the basic instability of the computation of the epipolar lines from small numbers of points. To find the point matching the selected point in the image at left, one needs to search over the regions covered by all these epipolar lines.*

$$C = \bar{l}\bar{l}^{T} - k^{2}\Sigma_{l}$$

n=10 *n*=15 *n*=25 *n*=50

Fig. 11.10. *The 95% envelopes of epipolar lines are shown for a noise level of* 0.5 *pixels, with* F *being computed from* $n = 10, 15, 25$ *and* 50 *points. In each case, Monte Carlo simulated results agreed closely with these results (though not shown here). For the case* $n = 15$*, compare with figure 11.9. Note that for* $n = 10$*, the epipolar envelope is very wide (> 90 degrees), showing that one can have very little confidence in an epipolar line computed from 10 points in this case. For* $n = 15$*, the envelope is still quite wide. For* $n = 25$ *and* $n = 50$*, the epipolar line is known with quite good precision. Of course, the precise shape of the envelope depends strongly on just what matched points are used to compute the epipolar structure.*

Next class:
image pair rectification
reconstructing points and lines