Two-view geometry: Computing fundamental matrix and Structure

Nov 28, 2017

EE, KAIST

김창익 (Kim, Changick)

Contents

- Review
- Computing fundamental matrix
- Automatic computation of F
- Structure computation (triangulation)
- 3D reconstruction

Some slides come from the following source.
Marc Pollefeys U. of North Carolina
And also refer to
H&Z's book Ch10 & 11.

Three questions:

(i) Correspondence geometry: Given an image point x in the first view, how does this constrain the position of the corresponding point x' in the second image?

A) A point in one view defines an epipolar line in the other view.

(ii) Camera geometry (motion): Given a set of corresponding image points $\{x_i \leftrightarrow x'_i\}$, $i=1,\ldots,n$, what are the cameras P and P' for the two views?

(iii) Scene geometry (structure): Given corresponding image points $x_i \leftrightarrow x'_i$ and cameras P, P', what is the position of (their pre-image) X in space?

Three questions:

(i) Correspondence geometry: Given an image point x in the first view, how does this constrain the position of the corresponding point x' in the second image?

$$\mathrm{l}' = \mathrm{F}\mathrm{x} \quad \longrightarrow \quad \mathrm{x}'^{\mathsf{T}}\mathrm{F}\mathrm{x} = 0$$

(ii) Camera geometry (motion): Given a set of corresponding image points $\{x_i \leftrightarrow x'_i\}$, i=1,…,n, what are the cameras P and P' for the two views?

$$\mathrm{P} = [\mathrm{I} \mid \mathbf{0}] \; \mathit{and} \; \mathrm{P}' = [[\mathrm{e}']_\times \mathrm{F} \mid \mathrm{e}'].$$

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

(See pp256, H&Z book)

$$\mathrm{F} \longrightarrow \mathrm{P}, \mathrm{P}' \qquad \mathrm{F} \longrightarrow \tilde{\mathrm{P}}, \tilde{\mathrm{P}}'$$

$$\left(\mathrm{F} = [\mathrm{e}']_\times \mathrm{P}'\mathrm{P}^{+}. \right)$$

(iii) Scene geometry (structure): Given corresponding image points $x_i \leftrightarrow x'_i$ and cameras P, P', what is the position of (their pre-image) X in space?

Basic equation: 8-point algorithm

We begin by describing the equations on F generated by point correspondences in two images.

$$\mathrm{x}'^{\mathrm{T}}\,\mathrm{F}\mathrm{x} = 0 \qquad \mathbf{x} = (x, y, 1)^{\mathsf{T}} \text{ and } \mathbf{x}' = (x', y', 1)^{\mathsf{T}}$$

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

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)

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

$$\mathrm{Af} = 0$$

the singularity constraint

An important property of the fundamental matrix is that it is singular, in fact of rank 2.

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

SVD from linearly computed F matrix (rank 3)

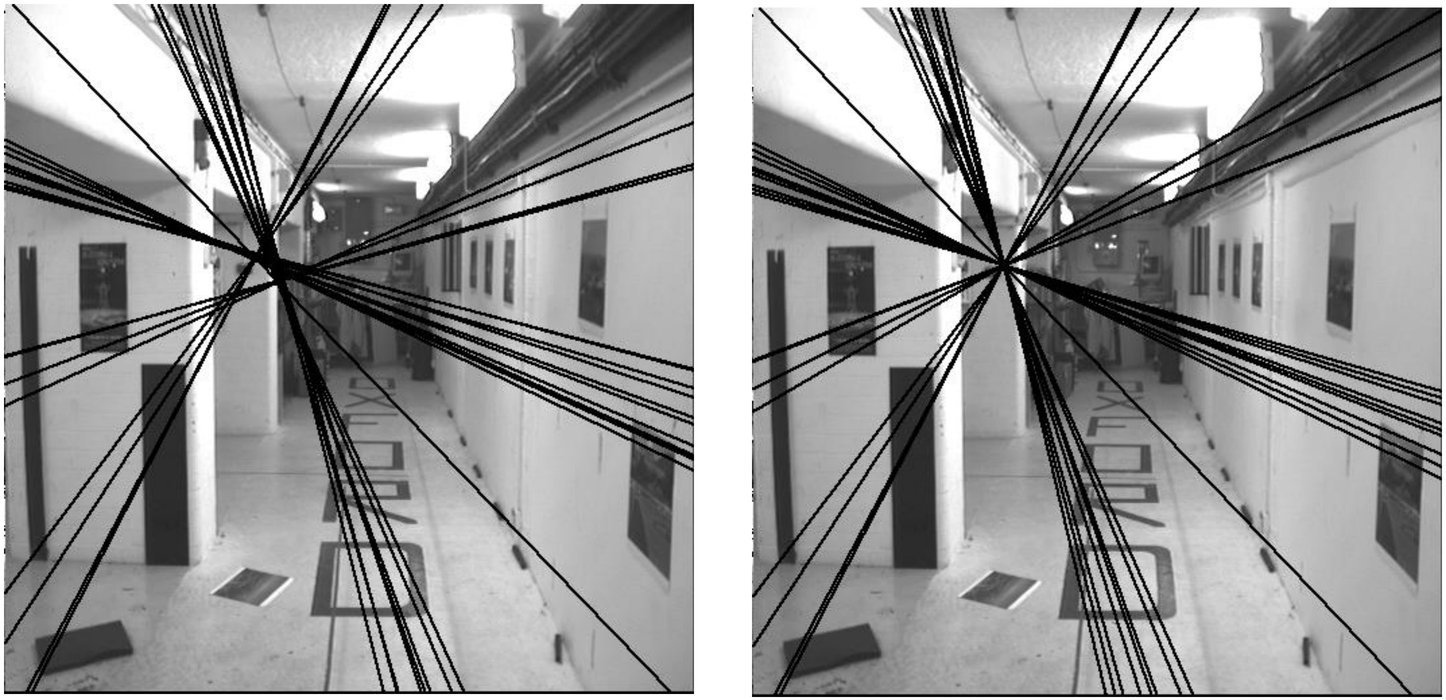
$$F = U \begin{bmatrix} \sigma_1 & & \\ & \sigma_2 & \\ & & \sigma_3 \end{bmatrix} V^T = U_1 \sigma_1 V_1^T + U_2 \sigma_2 V_2^T + U_3 \sigma_3 V_3^T$$

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

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

Computation of F-matrix

if the fundamental matrix is not singular then computed epipolar lines are not coincident,



The matrix F found by solving the set of linear equations will not in general have rank 2, and we should take steps to enforce this constraint.

Thus, the 8-point algorithm for computation of the fundamental matrix may be formulated as consisting of two steps, as follows.

(i) **Linear solution.** A solution $\mathtt{F}$ is obtained from the vector $\mathbf{f}$ corresponding to the smallest singular value of $\mathtt{A}$, where $\mathtt{A}$ is defined in (11.3).

(ii) **Constraint enforcement.** Replace $\mathtt{F}$ by $\mathtt{F}'$, the closest singular matrix to $\mathtt{F}$ under a Frobenius norm. This correction is done using the SVD.

The algorithm thus stated is extremely simple, and readily implemented, assuming that appropriate linear algebra routines are available. As usual normalization is required.

the minimum case – 7 point correspondences

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

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

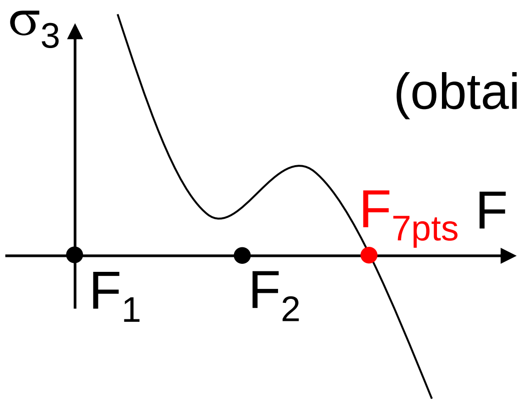
$$\Rightarrow \mathrm{A}[\mathrm{V}_8 \mathrm{V}_9] = 0_{9\mathrm{x}2} \qquad \left(\mathrm{e.g.} \mathrm{V}^{\mathrm{T}} \mathrm{V}_8 = [000000010]^{\mathrm{T}}\right)$$

$$\mathrm{x}_i{}^{\mathrm{T}} (\mathrm{F}_1 + \lambda \mathrm{F}_2) \mathrm{x}_i = 0, \forall i = 1 \ldots 7$$

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)

determinant

$$|A| = \begin{vmatrix} a & b & c \\ d & e & f \\ g & h & i \end{vmatrix} = a\begin{vmatrix} e & f \\ h & i \end{vmatrix} - b\begin{vmatrix} d & f \\ g & i \end{vmatrix} + c\begin{vmatrix} d & e \\ g & h \end{vmatrix}$$
$$= aei + bfg + cdh - ceg - bdi - afh.$$

eigenvalues

$$|A - \lambda I| = 0$$

the NOT normalized 8-point algorithm

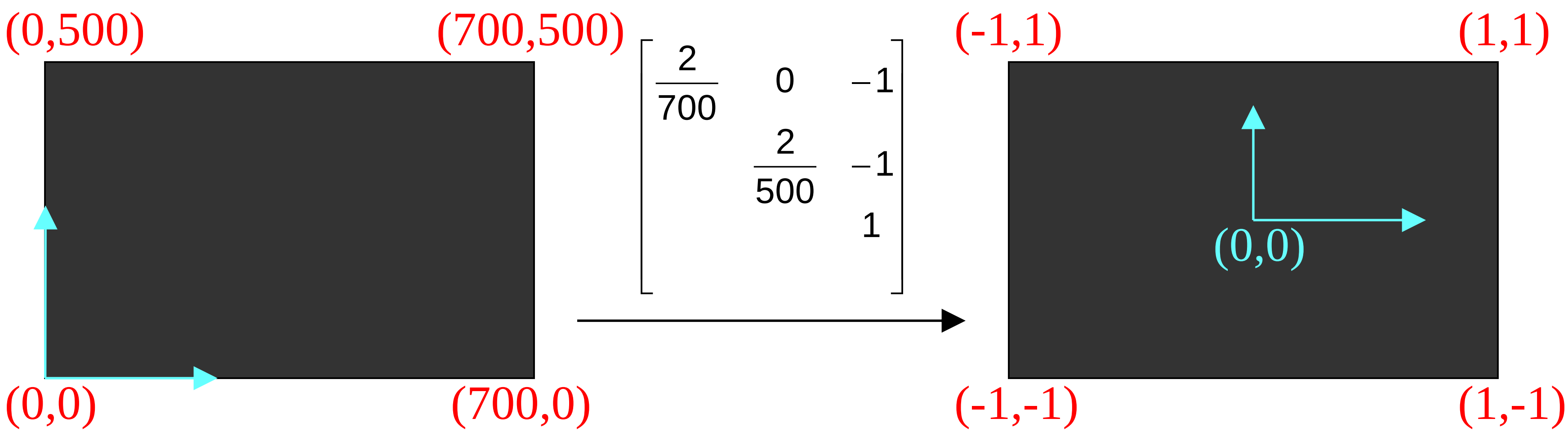
$$\begin{bmatrix} x_1 x_1' & y_1 x_1' & x_1' & x_1 y_1' & y_1 y_1' & y_1' & x_1 & y_1 & 1 \\ x_2 x_2' & y_2 x_2' & x_2' & x_2 y_2' & y_2 y_2' & y_2' & x_2 & y_2 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_n x_n' & y_n x_n' & x_n' & x_n y_n' & y_n y_n' & y_n' & x_n & y_n & 1 \end{bmatrix} \begin{bmatrix} f_{11} \\ f_{12} \\ f_{13} \\ f_{21} \\ f_{22} \\ f_{23} \\ f_{31} \\ f_{32} \\ f_{33} \end{bmatrix} = 0$$

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

Orders of magnitude difference
Between column of data matrix
$\rightarrow$ least-squares yields poor results

the normalized 8-point algorithm

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



$$\begin{bmatrix} \frac{2}{700} & 0 & -1 \\ & \frac{2}{500} & -1 \\ & & 1 \end{bmatrix}$$

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

Translate and scale each image so that the centroid of the reference points is at the origin of the coordinates and the RMS distance of the points from the origin is √2

Normalized 8-point algorithm for F

KAIST

Objective

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

Algorithm

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

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

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

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

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

Algorithm 11.1

algebraic minimization

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

Computation of F *with* det 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)

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

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)

Using the triangulation method

Gold standard

Objective

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

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

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

Algorithm

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

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

- (a) Choose camera matrices $\mathrm{P} = [\mathrm{I} \mid \mathbf{0}]$ and $\mathrm{P}' = [[\mathrm{e}']_\times \hat{\mathrm{F}} \mid \mathrm{e}']$, where e' is obtained from $\hat{\mathrm{F}}$.
- (b) From the correspondence $\mathrm{x}_i \leftrightarrow \mathrm{x}'_i$ and $\hat{\mathrm{F}}$ determine an estimate of $\widehat{\mathbf{X}}_i$ using the triangulation method of chapter 12.
- (c) The correspondence consistent with $\hat{\mathrm{F}}$ is obtained as $\hat{\mathrm{x}}_i = \mathrm{P}\widehat{\mathbf{X}}_i,\ \hat{\mathrm{x}}'_i = \mathrm{P}'\widehat{\mathbf{X}}_i$.

(iii) Minimize the cost

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

over $\hat{\mathrm{F}}$ 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 $\mathrm{P}' = [\mathtt{M} \mid \mathrm{t}]$, with $\hat{\mathrm{F}} = [\mathrm{t}]_\times \mathtt{M}$, and $\hat{\mathrm{x}}_i = \mathrm{P}\widehat{\mathbf{X}}_i,\ \hat{\mathrm{x}}'_i = \mathrm{P}'\widehat{\mathbf{X}}_i$.

First-order geometric error (Sampson error)

The Sampson approximation is used in the case of the variety defined by $\mathbf{x}^{\mathsf{T}}\mathbf{F}\mathbf{x} = 0$ to provide a first-order approximation to the geometric error.

$$\sum_i \frac{(\mathbf{x}_i'^{\mathsf{T}} \mathrm{F} \mathbf{x}_i)^2}{(\mathrm{F}\mathbf{x}_i)_1^2 + (\mathrm{F}\mathbf{x}_i)_2^2 + (\mathrm{F}^{\mathsf{T}}\mathbf{x}_i')_1^2 + (\mathrm{F}^{\mathsf{T}}\mathbf{x}_i')_2^2}.$$

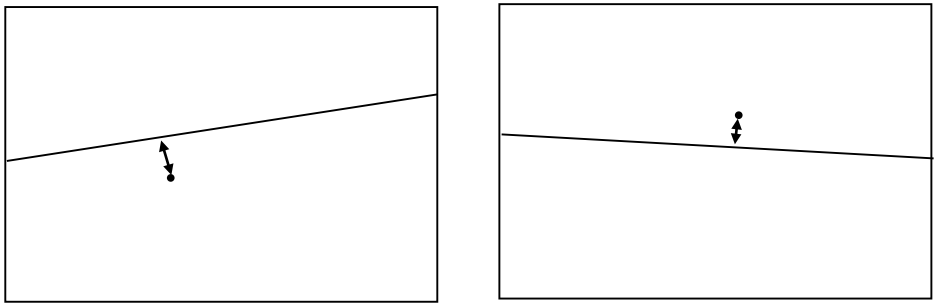
(problem if some x is located at epipole)

<u>advantage</u>: no subsidiary variables required

Symmetric epipolar error

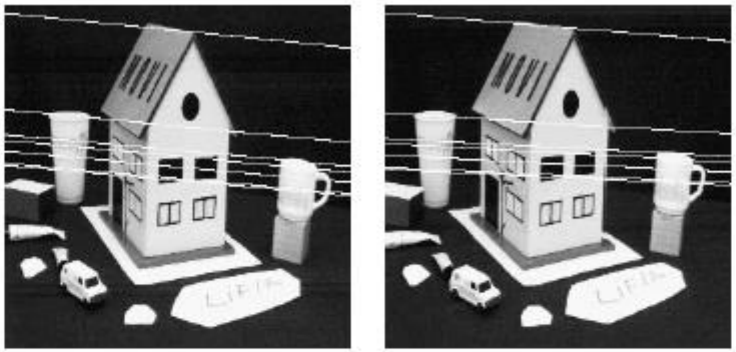
$$\sum_i d\left(\mathrm{x}'_i, F\mathrm{x}_i\right)^2 + d\left(\mathrm{x}_i, \mathrm{F}^{\mathrm{T}}\mathrm{x}'_i\right)^2$$

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



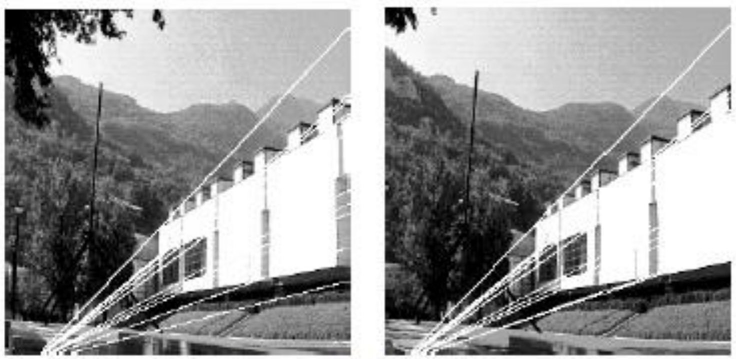
However, this cost function seems to give slightly inferior results to the Sampson cost function.

Experimental evaluation of the algorithms



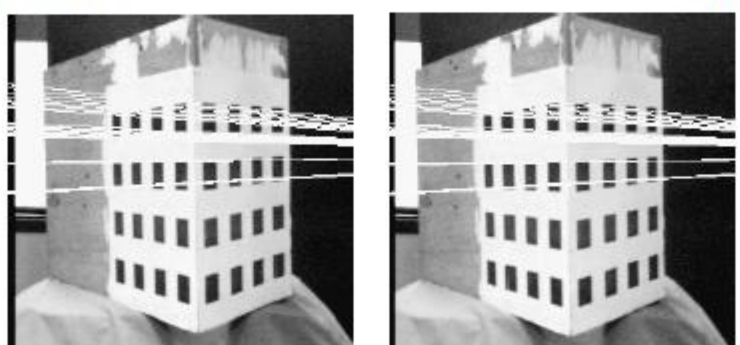
Houses Images

Statue image



Grenoble Museum

Corridor scene



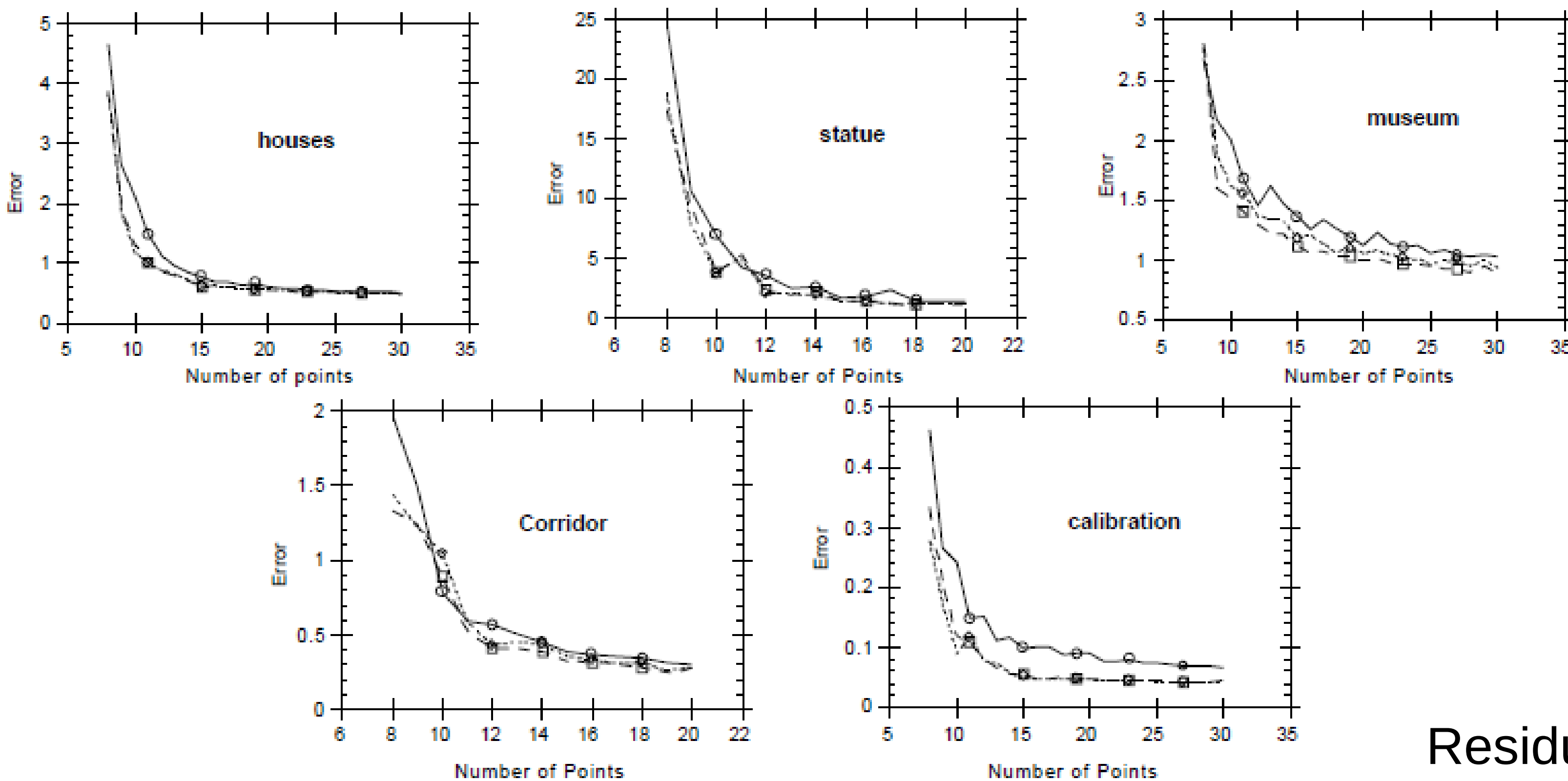
Calibration rig

Experimental evaluation of the algorithms

(i) The normalized 8-point algorithm (algorithm 11.1). Solid line

(ii) Minimization of algebraic error whilst imposing the singularity constraint (algorithm 11.2). Short dashed line

(iii) The Gold Standard geometric algorithm (algorithm 11.3). Long dashed line



Residual error:

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

(for all points!)

Recommendations:

- Do not use the unnormalized 8-point algorithm.
- For a quick method, easy to implement, use the normalized 8-point algorithm 11.1. This often gives adequate results, and is ideal as a first step in other algorithms.
- If more accuracy is desired, use the algebraic minimization method, either with or without iteration on the position of the epipole.
- As an alternative that gives excellent results, use an iterative-minimization method that minimizes the Sampson cost function (11.9). This and the iterative algebraic method give similar results.
- To be certain of getting the best results, if Gaussian noise is a viable assumption, implement the Gold Standard algorithm.

Automatic computation of F

(I) **The RANSAC sample**: Only 7 point correspondences are used to estimate F. This has the advantage that a rank 2 matrix is produced, and it is not necessary to coerce the matrix to rank 2 as in the linear algorithms. A second reason for using 7 correspondences, rather than 8 say with a linear algorithm, is that the number of samples that must be tried in order to ensure a high probability of no outliers is exponential in the size of the sample set. The slight disadvantage in using 7 correspondences is that it may result in 3 real solutions for F, and all 3 must be tested for support.

Sample size	Proportion of outliers ϵ						
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

Table 4.3. *The number N of samples required to ensure, with a probability $p = 0.99$, that at least one sample has no outliers for a given size of sample, s, and proportion of outliers, ϵ.*

Automatic computation of F

(II) **The distance measure**: Given a current estimate of F (from the RANSAC sample) the distance $d_\perp$ measures how closely a matched pair of points satisfies the epipolar geometry.

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

$$\sum_i \frac{(\mathbf{x}_i'^{\mathsf{T}} \mathrm{F} \mathbf{x}_i)^2}{(\mathrm{F}\mathbf{x}_i)_1^2 + (\mathrm{F}\mathbf{x}_i)_2^2 + (\mathrm{F}^{\mathsf{T}}\mathbf{x}'_i)_1^2 + (\mathrm{F}^{\mathsf{T}}\mathbf{x}'_i)_2^2}.$$

$d_\perp^2$

Automatic computation of F

(III) **Guided matching**: Further interest point correspondences are now determined using the estimated F to define a search strip about the epipolar line.

restrict search range to neighborhood of epipolar line
(±1.5 pixels)

relax disparity restriction (along epipolar line)

Automatic computation of F

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.

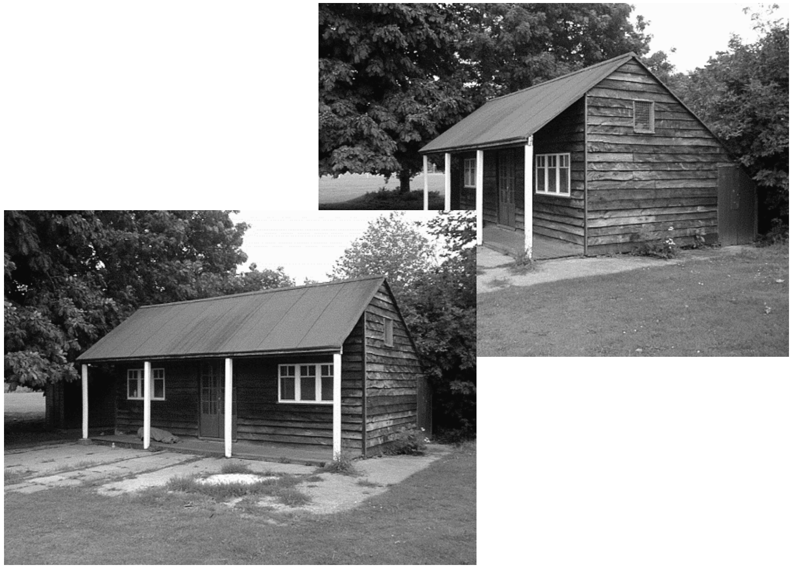
   Choose the F with the largest number of inliers. In the case of ties choose the solution that has the lowest standard deviation of inliers.
(iv) **Non-linear estimation:** re-estimate F from all correspondences classified as inliers by minimizing a cost function, e.g. (11.6), using the Levenberg–Marquardt 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.

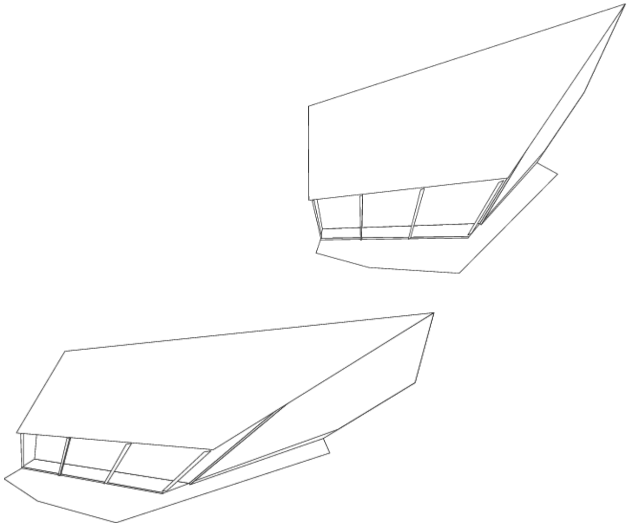
The last two steps can be iterated until the number of correspondences is stable.

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!





Rectification ambiguity

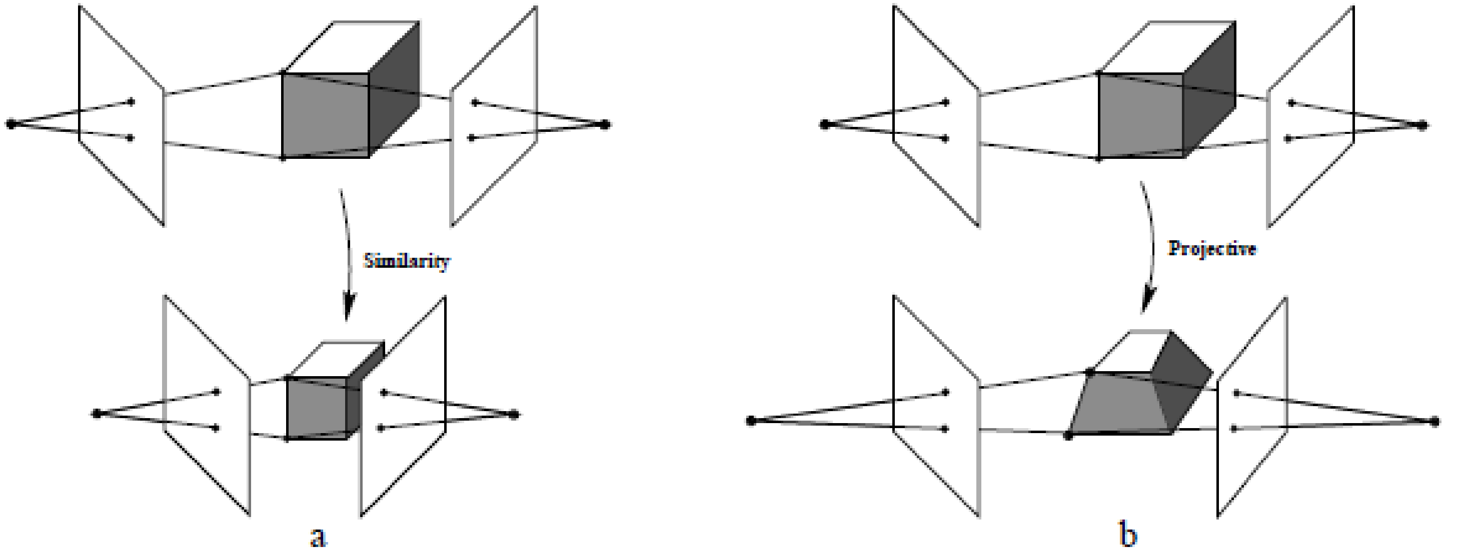


Fig. 10.2. **Reconstruction ambiguity.** *(a) If the cameras are calibrated then any reconstruction must respect the angle between rays measured in the image. A similarity transformation of the structure and camera positions does not change the measured angle. The angle between rays and the baseline (epipoles) is also unchanged. (b) If the cameras are uncalibrated then reconstructions must only respect the image points (the intersection of the rays with the image plane). A projective transformation of the structure and camera positions does not change the measured points, although the angle between rays is altered. The epipoles are also unchanged (intersection with baseline).*

Stratified reconstruction

The “stratified” approach to reconstruction is to begin with a projective reconstruction and then to refine it progressively to an affine and finally a metric reconstruction, if possible.

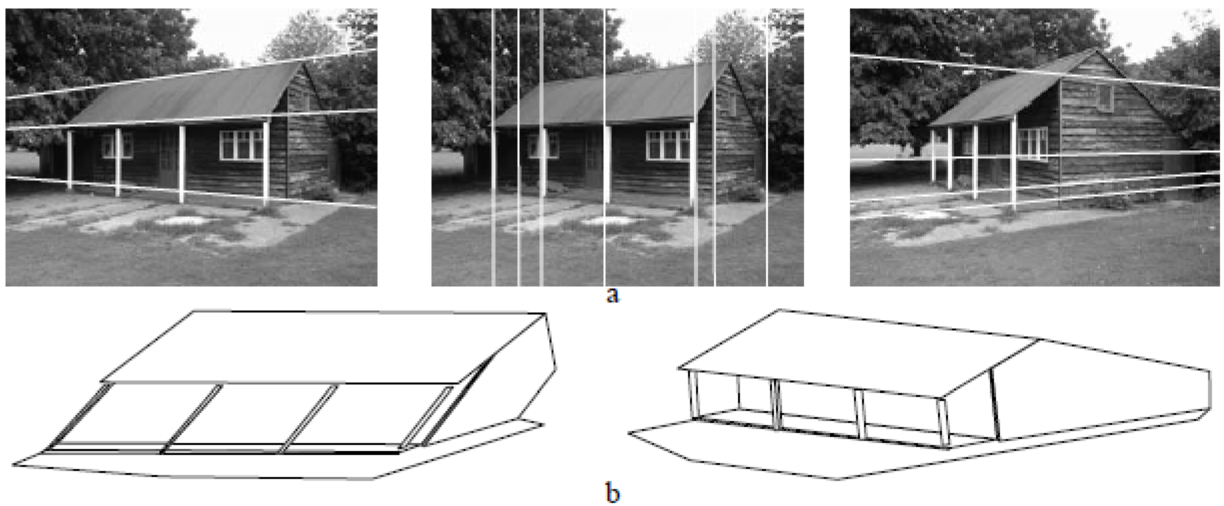


Fig. 10.4. **Affine reconstruction.** *The projective reconstruction of figure 10.3 may be upgraded to affine using parallel scene lines. (a) There are 3 sets of parallel lines in the scene, each set with a different direction. These 3 sets enable the position of the plane at infinity, π_∞, to be computed in the projective reconstruction. The wireframe projective reconstruction of figure 10.3 is then affinely rectified using the homography (10.2). (b) Shows two orthographic views of the wireframe affine reconstruction. Note that parallel scene lines are parallel in the reconstruction, but lines that are perpendicular in the scene are not perpendicular in the reconstruction.*

Stratified reconstruction

The “stratified” approach to reconstruction is to begin with a projective reconstruction and then to refine it progressively to an affine and finally a metric reconstruction, if possible.

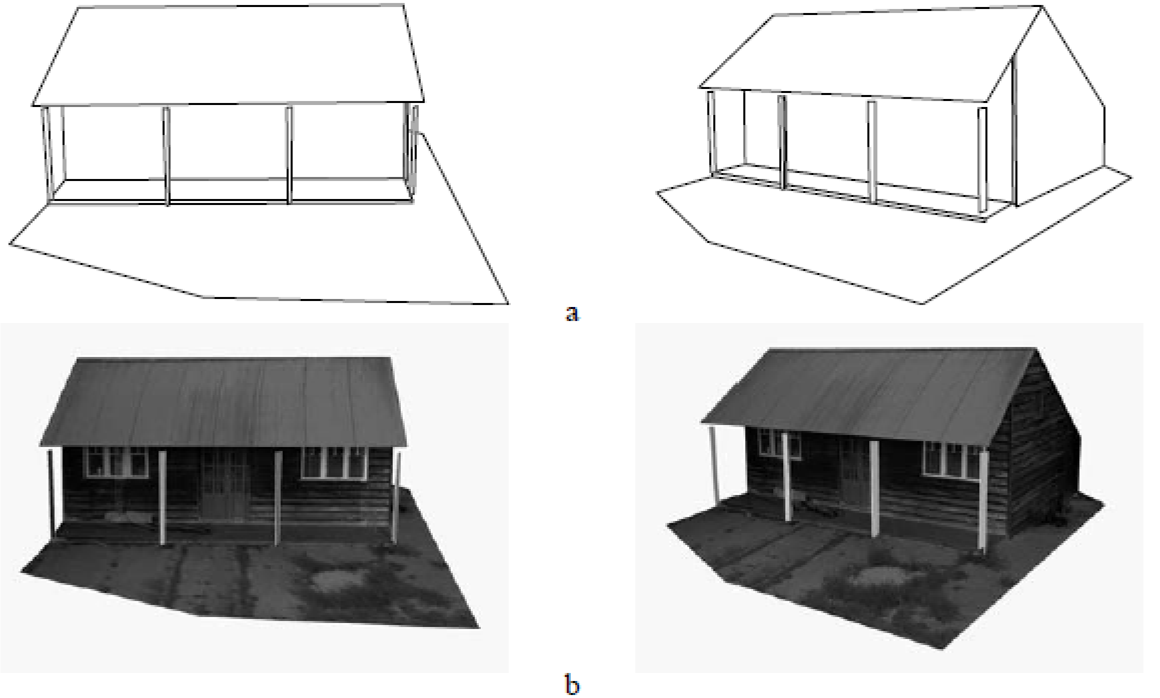


Fig. 10.5. **Metric reconstruction.** *The affine reconstruction of figure 10.4 is upgraded to metric by computing the image of the absolute conic. The information used is the orthogonality of the directions of the parallel line sets shown in figure 10.4, together with the constraint that both images have square pixels. The square pixel constraint is transferred from one image to the other using* H_∞. *(a) Two views of the metric reconstruction. Lines which are perpendicular in the scene are perpendicular in the reconstruction and also the aspect ratio of the sides of the house is veridical. (b) Two views of a texture mapped piecewise planar model built from the wireframes.*

Uncalibrated case

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

$$H = \begin{bmatrix} I \mid 0 \\ \pi \end{bmatrix}$$

(b) Metric reconstruction: compute IAC ω

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

Direct reconstruction

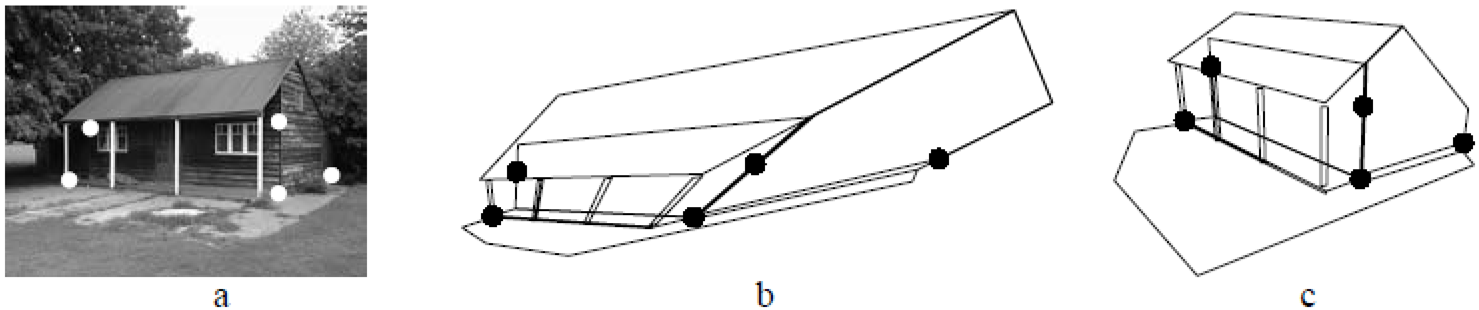


Fig. 10.6. **Direct reconstruction.** *The projective reconstruction of figure 10.3 may be upgraded to metric by specifying the position of five (or more) world points: (a) the five points used; (b) the corresponding points on the projective reconstruction of figure 10.3; (c) the reconstruction after the five points are mapped to their world positions.*

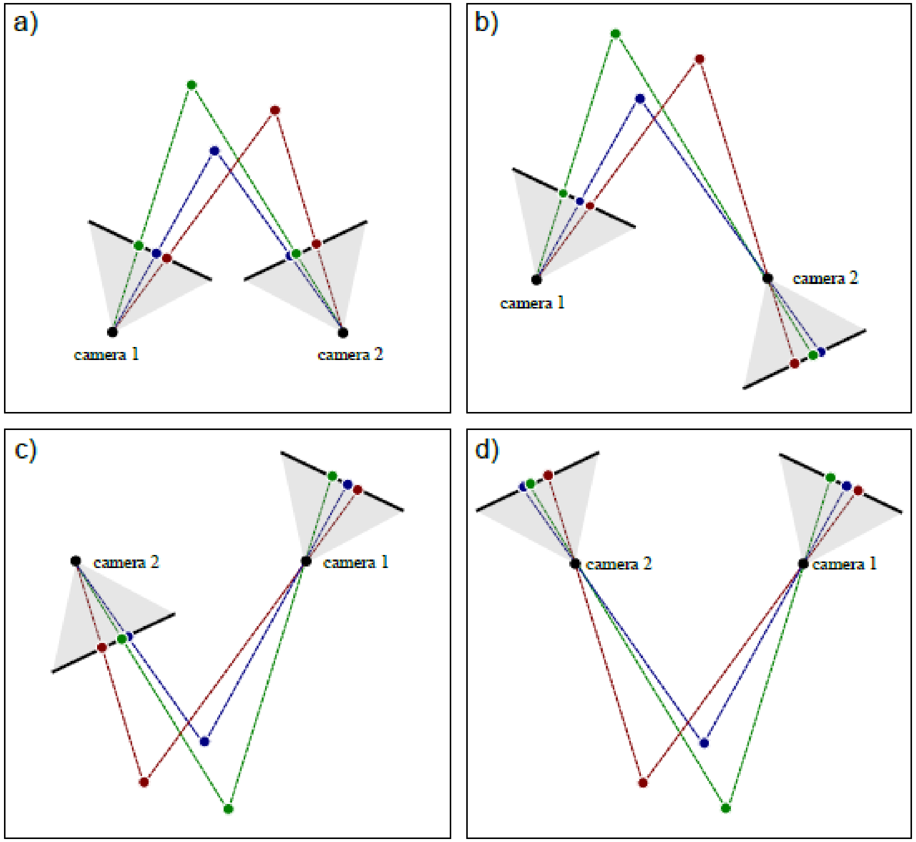
$$X_{Ei} = HX_i$$

Extraction of cameras from E-matrix

- The essential matrix may be computed directly from $\hat{x}'^{T} E \hat{x} = 0$ using normalized image coordinates, or else computed from the fundamental matrix using $E = K'^{T} F K$.
- Once the essential matrix is known, the camera matrices may be retrieved from E.
- In contrast with the fundamental matrix case, where there is a projective ambiguity, the camera matrices may be retrieved from the essential matrix up to scale and a four-fold ambiguity.

$$E = U\,\mathrm{diag}(1,1,0)V^{T},\ \textit{and}\quad P = [I \mid 0],$$

$$P' = [UWV^{T} \mid +u_3] \text{ or } [UWV^{T} \mid -u_3] \text{ or } [UW^{T}V^{T} \mid +u_3] \text{ or } [UW^{T}V^{T} \mid -u_3].$$



Only one of them (i.e., (a)) has all the points in front of both the cameras.

Refer to 9.6, HZ book for details

Recovering projection matrices

- A fundamental matrix can be decomposed to recover the camera motion, and, thereby, camera projection matrices.
  - If the camera calibration matrices **K** and **K'** are known, we can transform the recovered fundamental matrix into an essential matrix

$$\mathbf{E} \sim \mathbf{K}'^{\top}\mathbf{F}\mathbf{K}$$

and also $\mathbf{E} \sim [\mathbf{T}]_{\times}\mathbf{R}.$ (a)

  - This decomposition (a) can be achieved by computing the SVD of the essential matrix

$$\mathbf{E} = \mathbf{U}\boldsymbol{\Lambda}\mathbf{V}^{\top} = \mathbf{U}\,\mathrm{diag}(1, 1, 0)\mathbf{V}^{\top}$$

where $\boldsymbol{\Lambda} = \mathrm{diag}(\sigma_1, \sigma_2, \sigma_3)$ and the matrices $\mathbf{U}$ and $\mathbf{V}$ are orthogonal.

Recovering projection matrices

Result 9.17. *A 3×3 matrix is an essential matrix if and only if two of its singular values are equal, and the third is zero.*

Result 9.18. *Suppose that the SVD of* $\mathtt{E}$ *is* $\mathtt{U}\,\mathrm{diag}(1,1,0)\mathtt{V}^\mathsf{T}$. *Using the notation of (9.13), there are (ignoring signs) two possible factorizations* $\mathtt{E}=\mathtt{SR}$ *as follows:*

$$\mathtt{S}=\mathtt{UZU}^\mathsf{T} \quad \mathtt{R}=\mathtt{UWV}^\mathsf{T} \quad \textit{or} \quad \mathtt{UW}^\mathsf{T}\mathtt{V}^\mathsf{T}\ . \tag{9.14}$$

$$\mathbf{t}=\mathtt{U}\,(0,0,1)^\mathsf{T}=\mathbf{u}_3, \text{ the last column of } \mathtt{U}.$$

$$\mathtt{W}=\begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \text{and} \quad \mathtt{Z}=\begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}.$$

that $\mathtt{W}$ is orthogonal and $\mathtt{Z}$ is skew-symmetric.

Recovering projection matrices

- A fundamental matrix can be decomposed to recover the camera motion, and, thereby, camera projection matrices.

$$[\mathbf{T}]_{\times} = \mathbf{U} \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \mathbf{U}^{\top}$$

$$\mathbf{R} = \mathbf{U} \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \mathbf{V}^{\top}$$

The projection matrices follow directly from the recovered translation and rotation by aligning the reference coordinate system with the first camera to give:

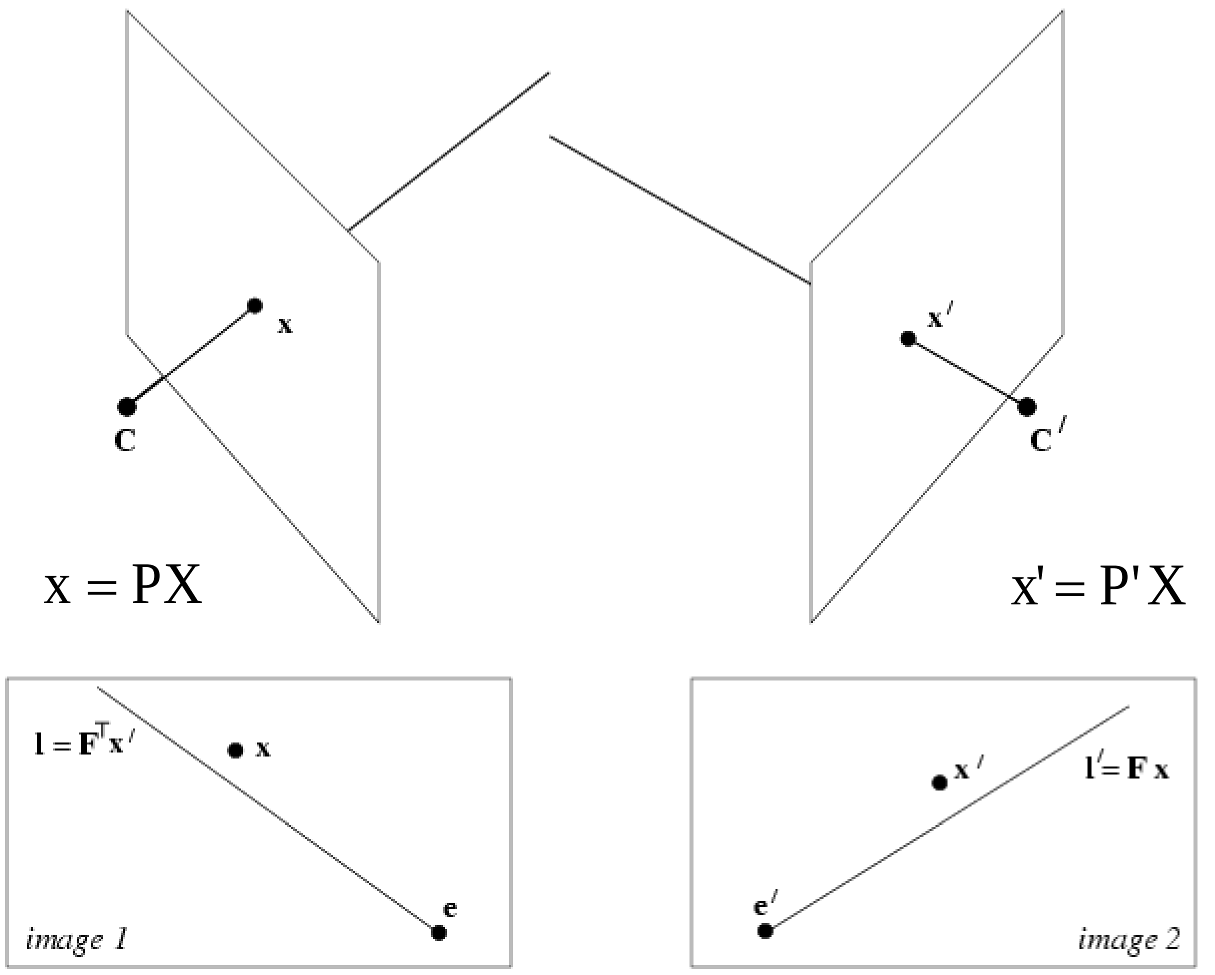
$$\begin{aligned} \mathbf{P} &= \mathbf{K}[\mathbf{I} \mid \mathbf{0}] \\ \mathbf{P}' &= \mathbf{K}'[\mathbf{R} \mid \mathbf{T}] \end{aligned}$$

where ***T*** is typically scaled such that $|\boldsymbol{T}| = 1$.

http://www.maths.lth.se/matematiklth/personal/calle/datorseende13/notes/forelas6.pdf

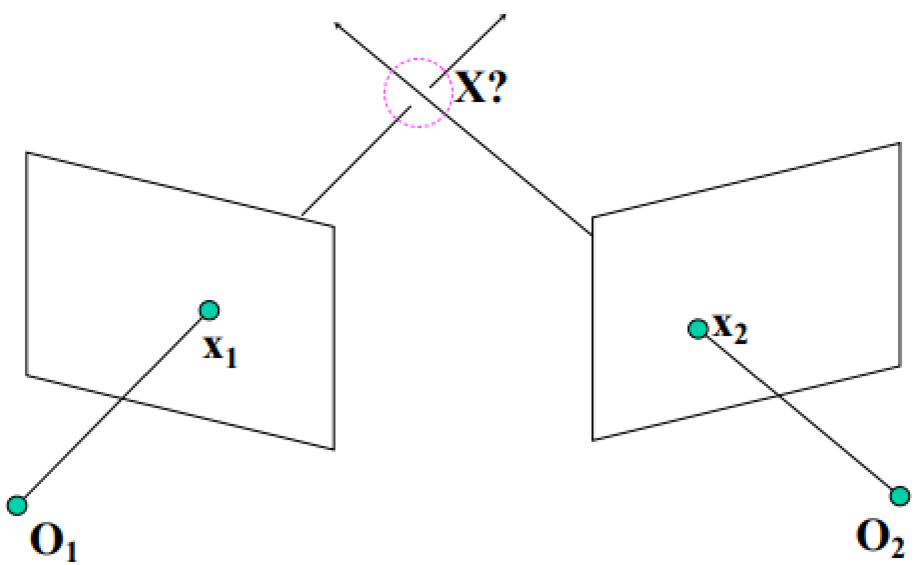
Point reconstruction

The two rays corresponding to a matching pair of points $\mathbf{x} \leftrightarrow \mathbf{x}'$ will meet in space if and only if the points satisfy the epipolar constraint. (below is the failure case)



Structure: Triangulation

- Given projections of a 3D point in two or more images (with known camera matrices), find the coordinates of the point



Linear triangulation method

for the first image, $\mathbf{x} \times (\mathtt{P}\mathbf{X}) = \mathbf{0}$ and writing this out gives

$$\begin{aligned} x(\mathbf{p}^{3\mathsf{T}}\mathbf{X}) - (\mathbf{p}^{1\mathsf{T}}\mathbf{X}) &= 0 \\ y(\mathbf{p}^{3\mathsf{T}}\mathbf{X}) - (\mathbf{p}^{2\mathsf{T}}\mathbf{X}) &= 0 \\ x(\mathbf{p}^{2\mathsf{T}}\mathbf{X}) - y(\mathbf{p}^{1\mathsf{T}}\mathbf{X}) &= 0 \end{aligned}$$

where $\mathbf{p}^{i\mathsf{T}}$ are the rows of $\mathtt{P}$. These equations are *linear* in the components of $\mathbf{X}$.
An equation of the form $\mathtt{A}\mathbf{X} = \mathbf{0}$ can then be composed, with

$$\mathtt{A} = \begin{bmatrix} x\mathbf{p}^{3\mathsf{T}} - \mathbf{p}^{1\mathsf{T}} \\ y\mathbf{p}^{3\mathsf{T}} - \mathbf{p}^{2\mathsf{T}} \\ x'\mathbf{p}'^{3\mathsf{T}} - \mathbf{p}'^{1\mathsf{T}} \\ y'\mathbf{p}'^{3\mathsf{T}} - \mathbf{p}'^{2\mathsf{T}} \end{bmatrix}$$

where two equations have been included from each image, giving a total of four equations in four homogeneous unknowns.

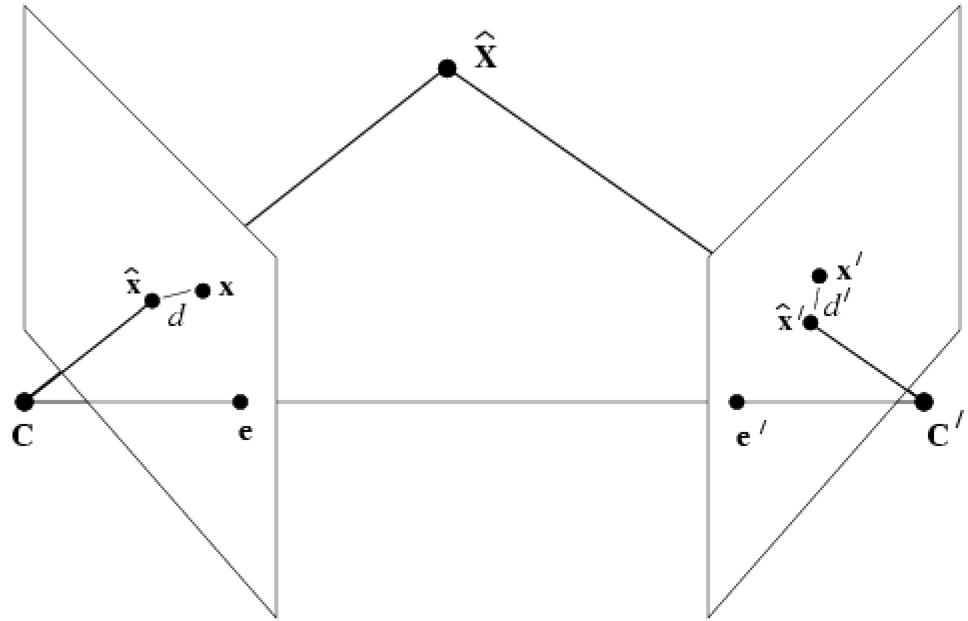
Finds the solution of **AX = 0**
as the unit singular vector corresponding to the smallest singular value of **A**.

geometric error

A typical observation consists of a noisy point correspondence $\mathbf{x} \leftrightarrow \mathbf{x}'$ which does not in general satisfy the epipolar constraint.

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

$$\text{or equivalently subject to } \hat{\mathrm{x}} = \mathrm{P}\hat{\mathrm{X}} \text{ and } \hat{\mathrm{x}}' = \mathrm{P}'\hat{\mathrm{X}}$$



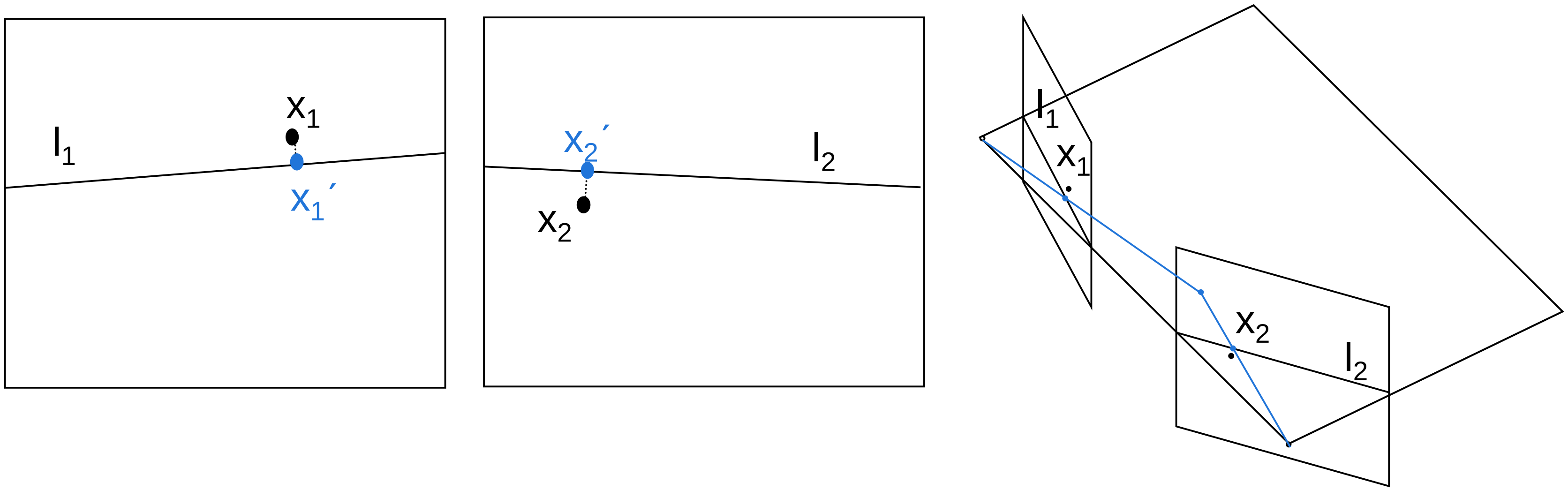
Once $\hat{\mathrm{x}}'$ and $\hat{\mathrm{x}}$ are found, the point $\hat{\mathrm{X}}$ may be found by any triangulation method, since the corresponding rays will meet precisely in space.

This cost function could, of course, be minimized using a numerical minimization method such as Levenberg–Marquardt. A close approximation to the minimum may also be found using a first-order approximation to the geometric cost function, namely the Sampson error.

Optimal 3D point in epipolar plane

A method of triangulation that finds the global minimum of the cost function (i.e., geometric error) using a non-iterative algorithm (refer to 12.5, HZ book)

Given an epipolar plane, find best 3D point for (x_1,x_2)



Select closest points (x_1´,x_2´) on epipolar lines

Obtain 3D point through exact triangulation

Guarantees minimal reprojection error

(given this epipolar plane)

Optimal epipolar plane

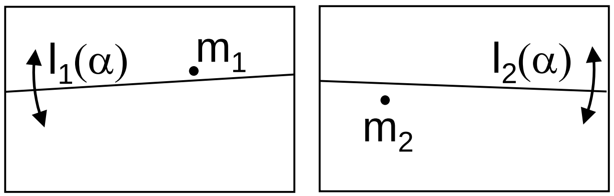
- **Reconstruct matches in projective frame by minimizing the reprojection error**

$$d\left(x_1, P_1X\right)^2 + d\left(x_2, P_2X\right)^2 \qquad \textbf{3DOF}$$

- **Non-iterative method** (Hartley and Sturm, CVIU´97)

Determine the epipolar plane for reconstruction

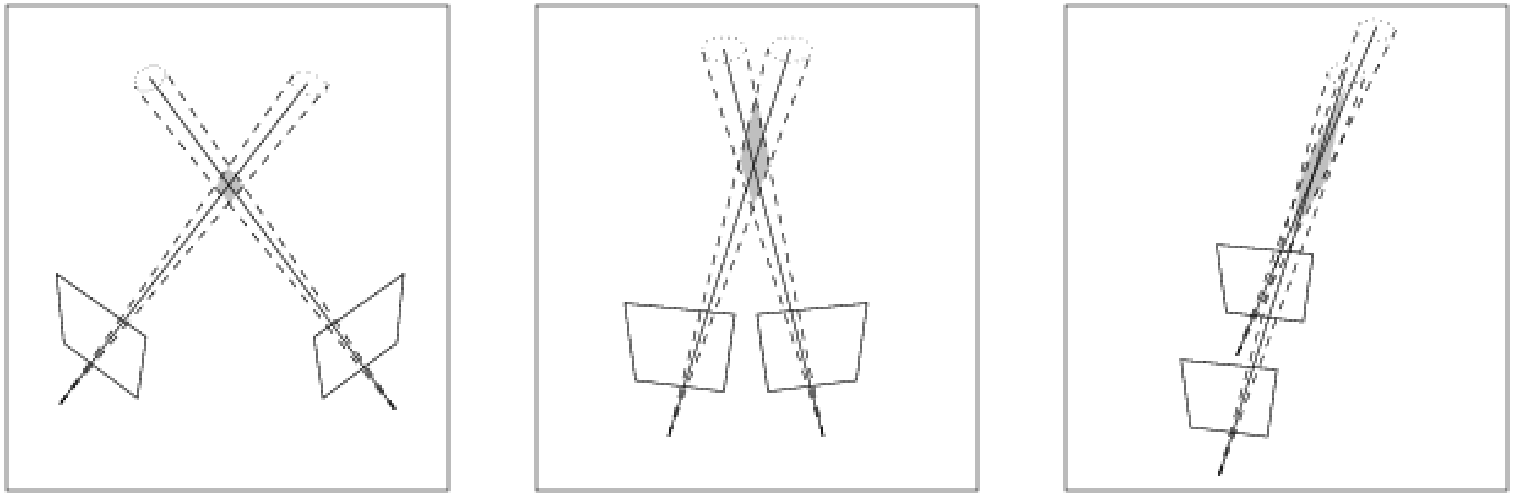
$$D\left(x_1, l_1(\alpha)\right)^2 + D\left(x_2, l_2(\alpha)\right)^2 \qquad \textbf{1DOF}$$



(polynomial of degree 6
check all minima, incl ∞)

Reconstruct optimal point from selected epipolar plane

Reconstruction uncertainty



consider angle between rays

The shaded region in each case illustrates the shape of the uncertainty region, which depends on the angle between the rays. Points are less precisely localized along the ray as the rays become more parallel. Forward motion in particular can give poor reconstructions since rays are almost parallel for much of the field of view.

3D reconstruction

- **Reconstruction ambiguity**

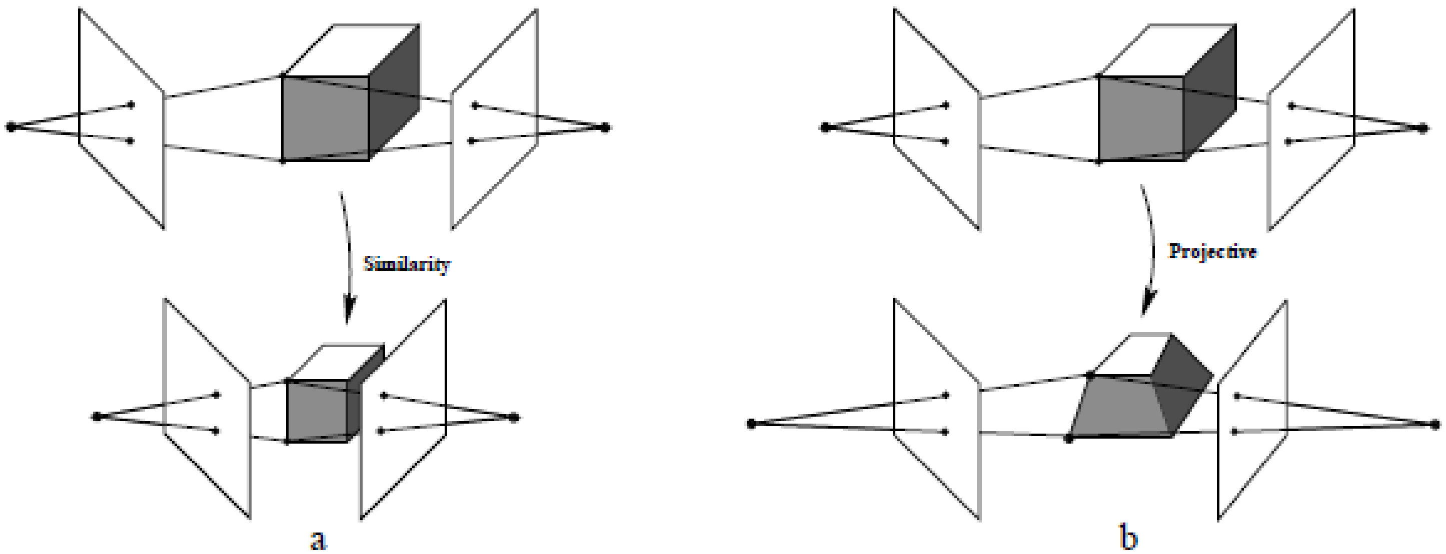
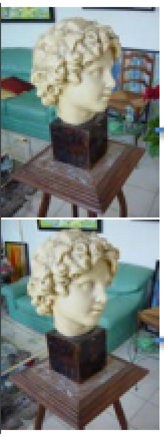


Fig. 10.2. **Reconstruction ambiguity.** *(a) If the cameras are calibrated then any reconstruction must respect the angle between rays measured in the image. A similarity transformation of the structure and camera positions does not change the measured angle. The angle between rays and the baseline (epipoles) is also unchanged. (b) If the cameras are uncalibrated then reconstructions must only respect the image points (the intersection of the rays with the image plane). A projective transformation of the structure and camera positions does not change the measured points, although the angle between rays is altered. The epipoles are also unchanged (intersection with baseline).*

3D reconstruction

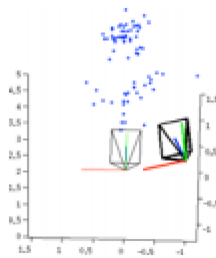
- **Calibrated case**



keypoints → match

keypoints → match

match → fundamental matrix → essential matrix → [R|t] → triangulation →



$$E = K'^{T}FK.$$

Projective to affine reconstruction

- **Uncalibrated case**

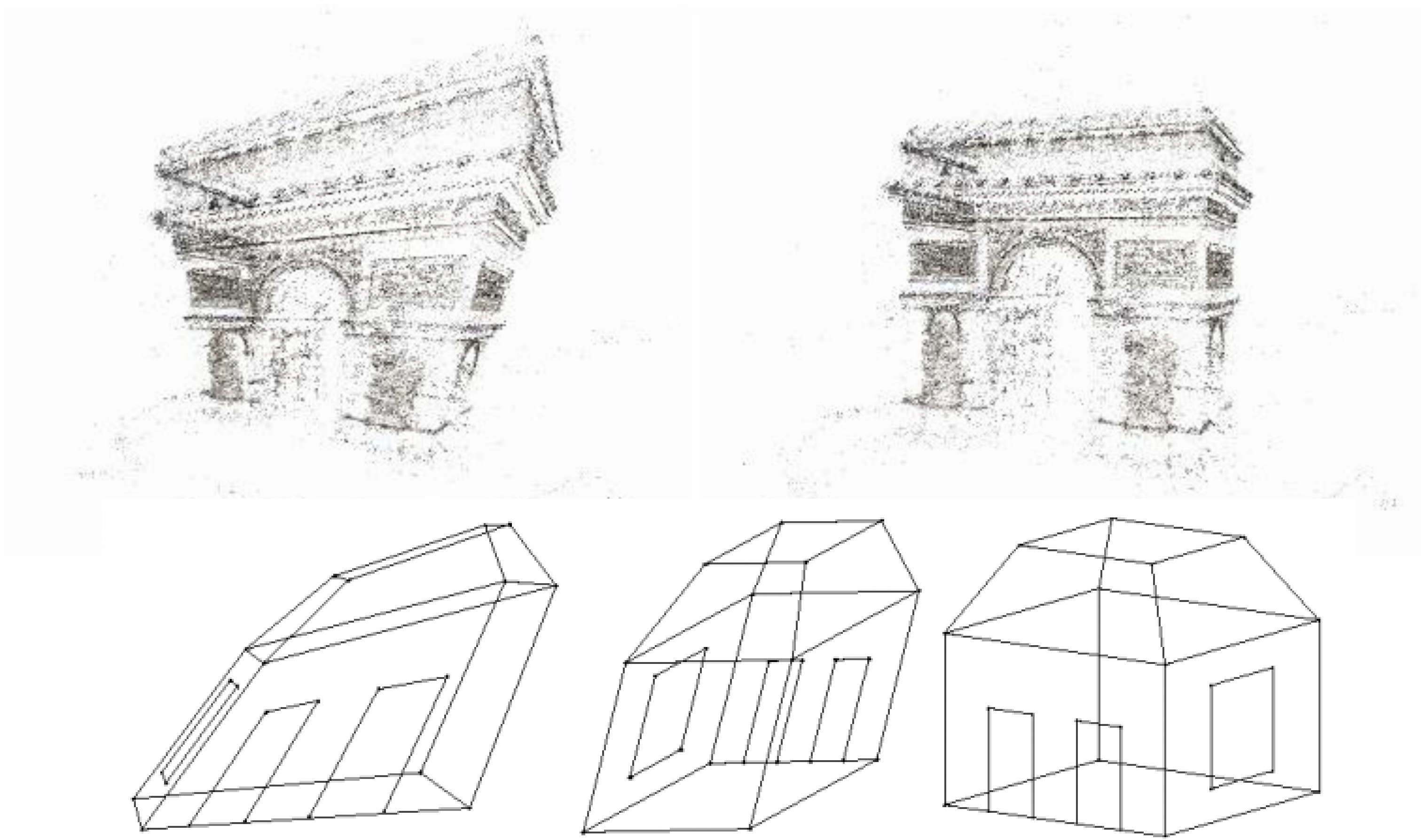


Figure 1. Projective structure $\boldsymbol{X}_p$, affine structure $\boldsymbol{X}_a$, and Euclidean structure $\boldsymbol{X}_e$ obtained in different stages of reconstruction (MaSKS fig6.10)

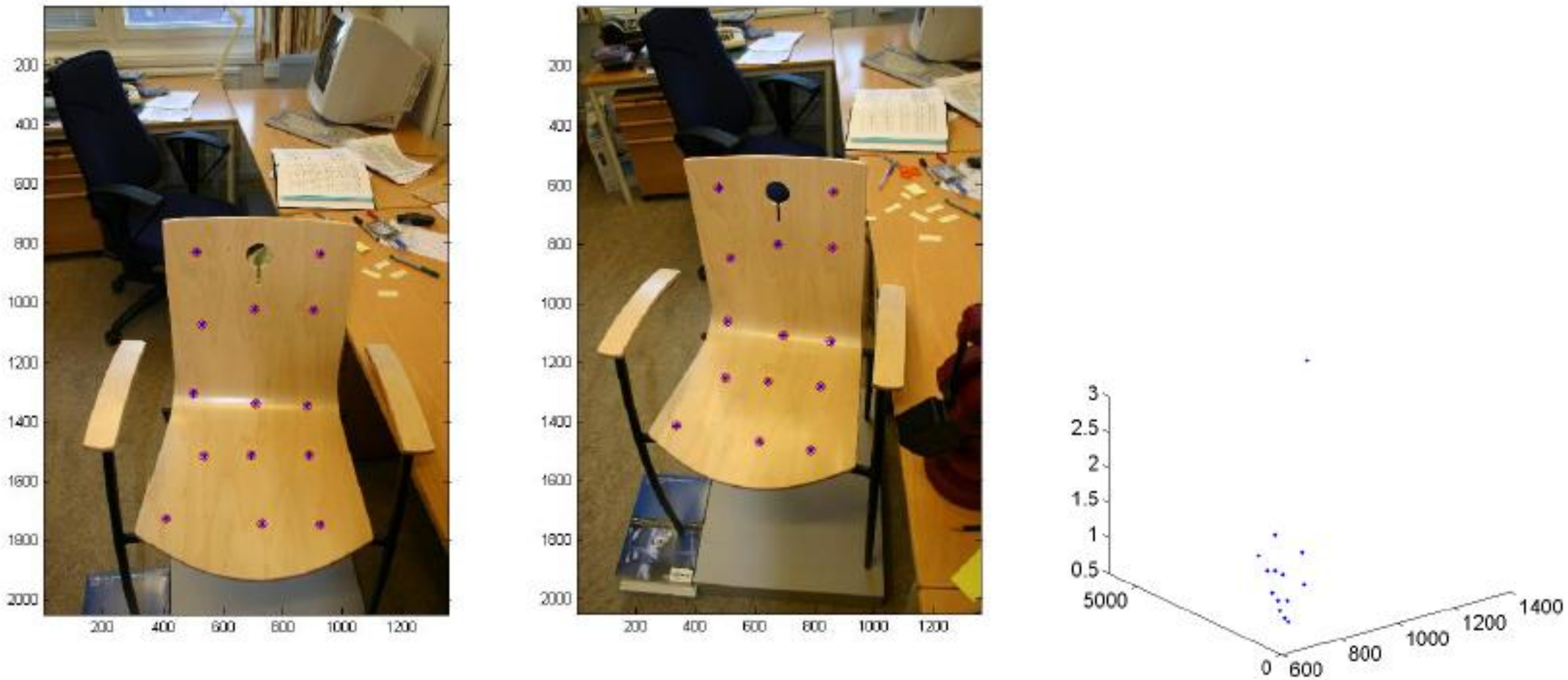
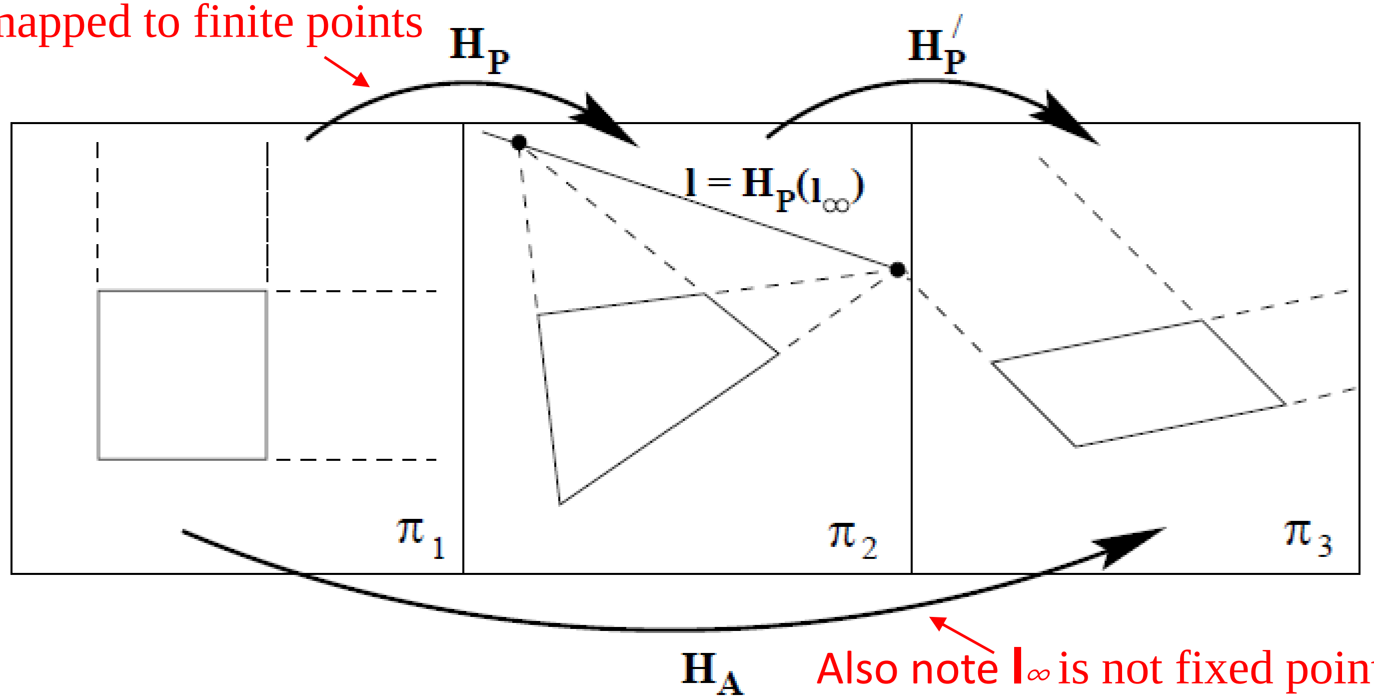


Figure 1: Two images of a chair with 14 known point correspondences. Blue $*$ are the image measurements and red o are the reprojections. The 3D points (to the right) look strange because of the projective ambiguity (note the difference in scale on the axes).

Affine Rectification

- **Recall Fig. 2.12, HZ book as starting point**

Under a projective transformation ideal points
may be mapped to finite points



Also note $\mathbf{l}_\infty$ is not fixed pointwise under
an affine transformation

	Affine upgrade
2D	line at infinity $\boldsymbol{\ell}_\infty = (0,0,1)^\top$
3D	plane at infinity $\boldsymbol{\pi}_\infty = (0,0,0,1)^\top$

Affine upgrade

The line at infinity $\boldsymbol{\ell}_\infty$ is used to do affine upgrade in the 2D reconstruction because it stays fixed under an affine transformation. In the 3D case, the plane at infinity $\boldsymbol{\pi}_\infty$ is the corresponding entity that allows us to achieve the same goal.

- Plane at infinity $\boldsymbol{\pi}_\infty$

Parallelism is preserved in the affine structure. The plane at infinity π_∞ enables us to identify parallelism. In particular,

  - Two planes are parallel iff the line of intersection is on π_∞.
  - Two lines are parallel iff the point of intersection is on π_∞ .

Points on $\boldsymbol{\pi}_\infty$ are ideal points of the form $\boldsymbol{X} = (X, Y, Z, 0)^\top$ satisfying the equation

$$(0, 0, 0, 1)^\top \boldsymbol{X} = 0$$

Considering the way a projective transformation acts on planes, we want to find H such that $H^{-T}\pi = (0,0,0,1)^T$. Such a transformation is given by

$$H = \begin{bmatrix} I \mid 0 \\ \pi^T \end{bmatrix}. \tag{10.2}$$

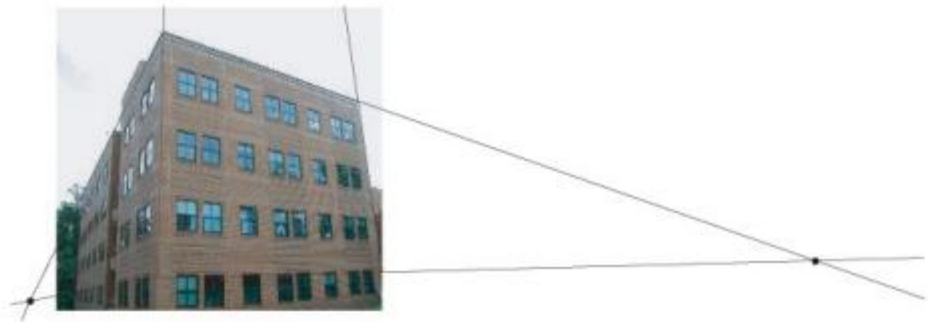
Indeed, it is immediately verified that $H^T(0,0,0,1)^T = \pi$, and thus $H^{-T}\pi = (0,0,0,1)^T$, as desired. The transformation H is now applied to all points and the two cameras.

Note that the plane π is a 4-vector in the coordinate frame of the projective reconstruction. In other words, π is the image of π_∞.

Then how to specify π?

π

- As with any plane, we need 3 points to specify **π**, which we can get from three vanishing points, assuming they are measurable.
- Vanishing points



Parallel lines and the associated vanishing points. Two vanishing points are shown; the third is off the page far above the picture.

- As an alternative, we can find the 3D coordinates of a vanishing point purely from 2D data in the two image planes. H&Z suggest the following algorithm:
  - (1) Find vanishing point v1 in image 1 from a pair of imaged parallel lines.
  - (2) Compute epipolar line in image 2: $\mathbf{l}' = \mathbf{F}\mathbf{v}_1$. (The point in the second image corresponding to $\mathbf{v}_1$ must lie on this line.)
  - (3) Intersect $\mathbf{l}'$ with a corresponding parallel line in image 2, call it $\mathbf{v}_2$.
  - (4) Find projective depth of ($\mathbf{v}_1$, $\mathbf{v}_2$).

  Note that step (2) ensures that the backprojected rays through $\mathbf{v}_1$ and $\mathbf{v}_2$ in 3D will intersect.

π for affine upgrade

- Let $\boldsymbol{\pi} = (\mathbf{v}^T, v_4^T) = (v_1, v_2, v_3, v_4)^T$

Now that we have three vanishing points $\boldsymbol{X}_p^j$, $j = 1, 2, 3$, we can then solve for $(\boldsymbol{v}^\top, v_4)^\top$ from the linear system

$$(v_1, v_2, v_3, v_4)\boldsymbol{X}_p^j = 0 \qquad j = 1, 2, 3$$