Machine Vision for Robotics and HCI
Two-View Geometry

Matthew Dailey

Computer Science and Information Management
Asian Institute of Technology

Readings

Readings for these lecture notes:

- Hartley, R., and Zisserman, A. *Multiple View Geometry in Computer Vision*, Cambridge University Press, 2004, Chapter 9–12, 18.
- Lowe, D.G. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision* 2(60): 91–110, 2004.

Outline

In this part we consider three problems:

- Given a point $\mathbf{x}$ in one view of a scene, how is the position of the corresponding point $\mathbf{x}_i$ in a second view constrained?
- Given a set of corresponding points in two images, what are the cameras $\mathtt{P}$ and $\mathtt{P}'$?
- Given corresponding points in two images and the cameras $\mathtt{P}$ and $\mathtt{P}'$, what are the points' positions in 3 space?

Outline

The epipolar geometry is the projective geometry between two views. It only depends on the cameras' internal parameters and the relative pose of the two cameras.

The fundamental matrix is a 3×3 rank 2 matrix $\mathtt{F}$ capturing this geometry.

We will see that for any point $\mathbf{X} \in \mathbb{R}^3$, if $\mathbf{X}$ is imaged as $\mathbf{x}$ in view 1 and $\mathbf{x}'$ in view 2, the image points will satisfy $\mathbf{x}'^T \mathtt{F} \mathbf{x} = 0$.

Why do we care about $\mathtt{F}$?

- Given point correspondences, we can compute $\mathtt{F}$ in a manner similar to computing $\mathtt{H}$.
- Given $\mathtt{F}$, we can recover the cameras $\mathtt{P}$ and $\mathtt{P}'$, up to a projectivity.
- If we know $\mathtt{K}$ and $\mathtt{K}'$, we can also retrieve the Euclidean motion of the camera between the views, with ambiguity about the scale and absolute reference frame.

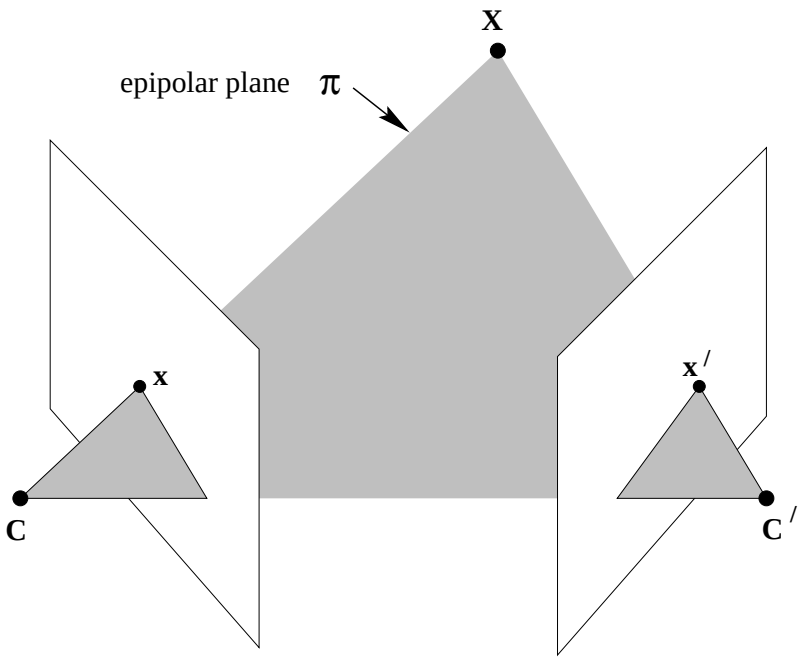
Epipolar geometry and the fundamental matrix

Epipolar planes

The baseline is the line joining two camera centers.

The baseline is the axis for a pencil of planes called epipolar planes.

When we have a point $\mathbf{X}$ in $\mathbb{R}^3$ imaged as $\mathbf{x}$ and $\mathbf{x}'$ in two views, we can see that $\mathbf{X}$, $\mathbf{x}$, $\mathbf{x}'$, and the camera centers all lie on the same plane.

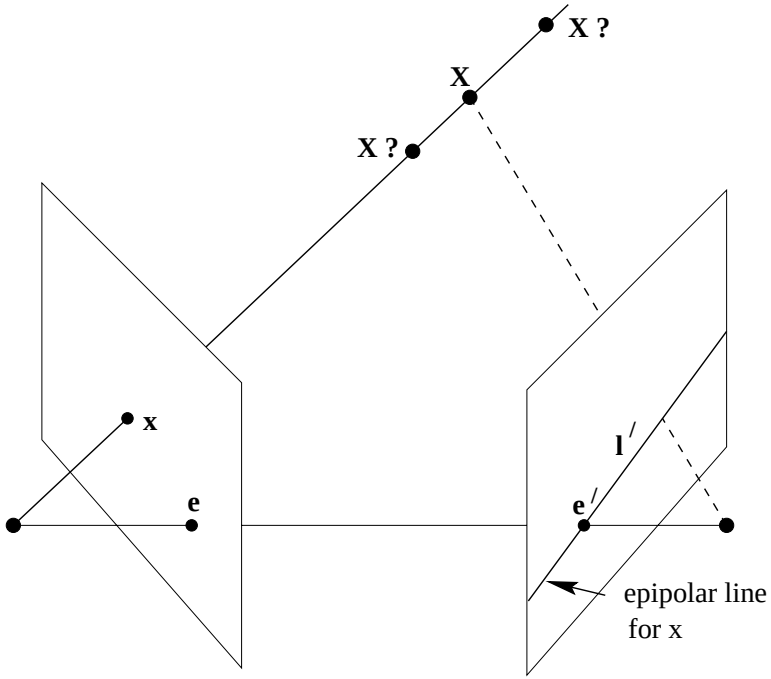


Hartley and Zisserman (2004), Fig. 9.1a

Epipolar geometry and the fundamental matrix

Epipolar lines

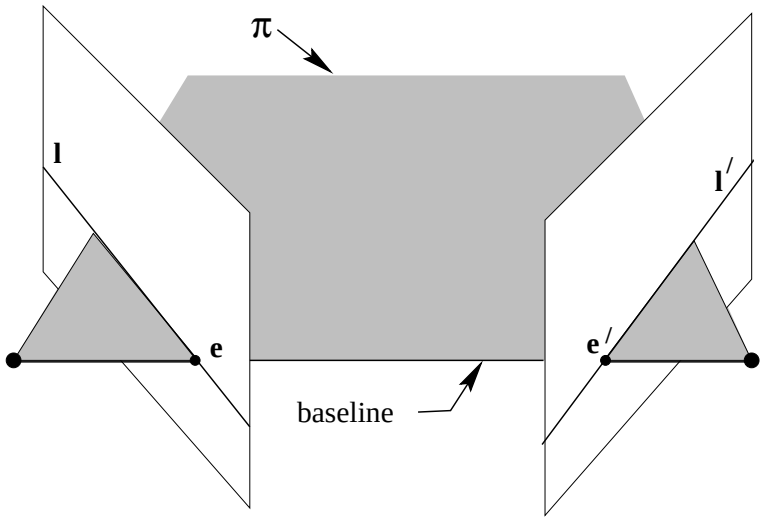
We see that the image of the backprojection of $\mathbf{x}$ in view 2 forms a line $\mathbf{l}'$. This line is called the epipolar line corresponding to $\mathbf{x}$.



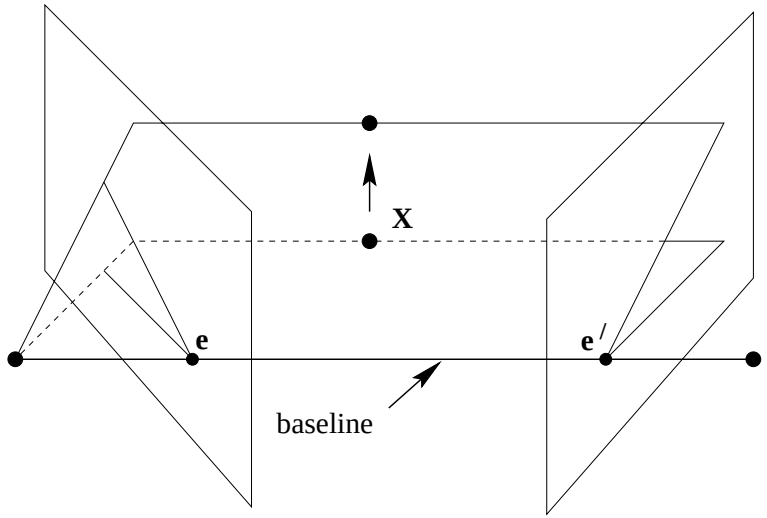
Hartley and Zisserman (2004), Fig. 9.1b

When attempting stereo correspondence, we see immediately that if the epipolar geometry is known, we don't have to search the entire image for a corresponding point — we only have to search along the epipolar line.

The epipoles are the intersections of the baseline with the image planes:



The baseline intersects the image planes at epipoles $\mathbf{e}$ and $\mathbf{e}'$



As we move a point $\mathbf{X}$ in $\mathbb{R}^3$, the epipolar planes rotate around the baseline and always intersect the epipoles.

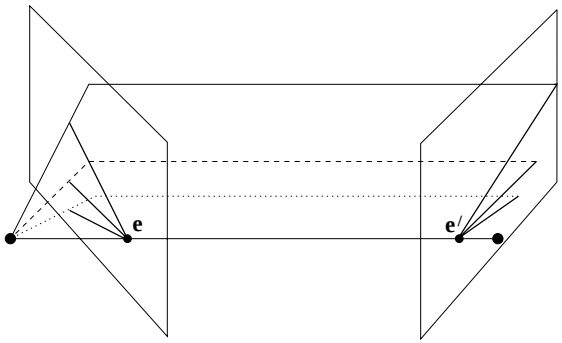
Hartley and Zisserman, Fig. 9.2

To summarize the main terms in epipolar geometry:

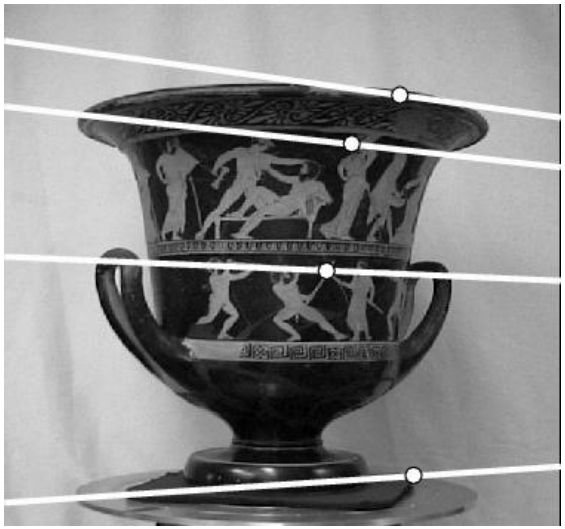
- The epipole is the point of intersection of the baseline with the image plane.
- An epipolar plane is a plane containing the baseline.
- An epipolar line is the intersection of an epipolar plane with the image plane. All epipolar lines intersect at the epipole.

Epipolar geometry and the fundamental matrix

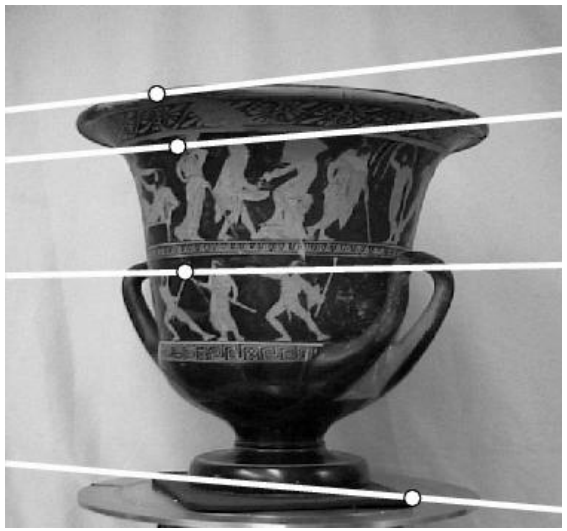
Example 1



Example camera setup with converging cameras



Left image, with points and epipolar lines



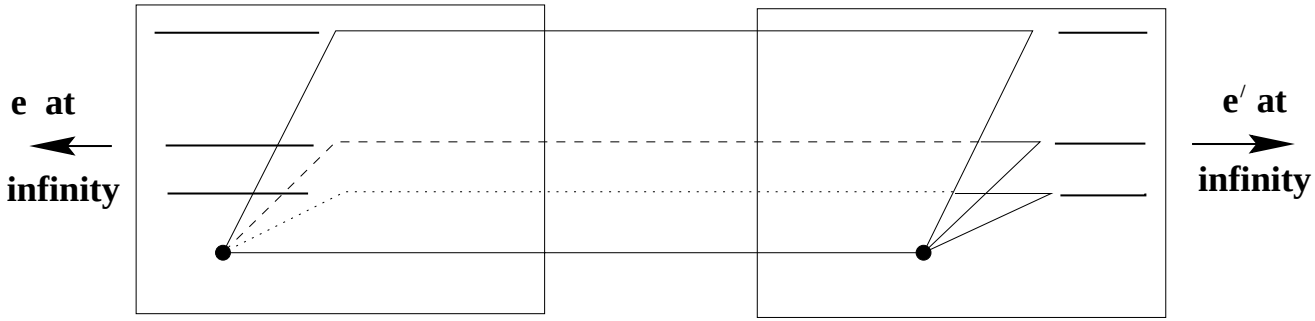
Right image, with corresponding points/lines

Hartley and Zisserman (2004), Fig. 9.3

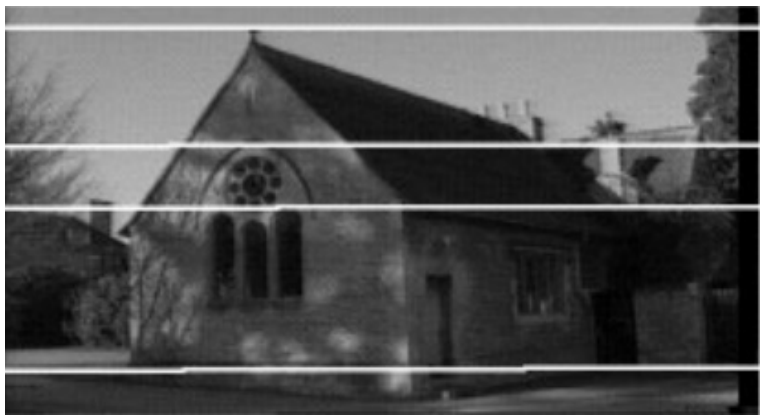
The two cameras are related by translation plus rotation. The epipoles are finite (but outside the image).

Epipolar geometry and the fundamental matrix

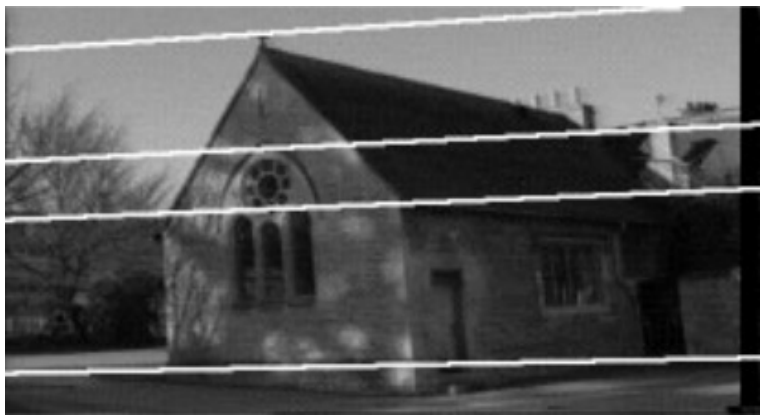
Example 2



Example camera setup with parallel cameras



Left image, with some sample epipolar lines



Right image, with corresponding epipolar lines

Hartley and Zisserman (2004), Fig. 9.4

The two image planes are nearly parallel. The motion is a translation with a small rotation around the z axis. The epipoles are points at infinity.

The fundamental matrix is the "algebraic representation of epipolar geometry" (Hartley and Zisserman, 2004, p. 241).

As we know now, a point $\mathbf{x}'$ in image 2 correspoding to point $\mathbf{x}$ in image 1 must lie on the epipolar line $\mathbf{l}'$. We need a map from points in one image to epipolar lines in the other image, i.e., $\mathbf{x} \mapsto \mathbf{l}'$.

Recall that a homography is a 3×3 rank 3 matrix mapping a point in one image to a point in the other image.

The fundamental matrix $\mathtt{F}$ is a 3×3 rank 2 matrix mapping a point in image 1 to the corresponding line $\mathbf{l}'$ in image 2, i.e., $\mathbf{l}' = \mathtt{F}\mathbf{x}$.

Since the point $\mathbf{x}'$ corresponding to $\mathbf{x}$ necessarily lies on $\mathbf{l}'$, we know that $\mathbf{x}'^T\mathbf{l}' = 0$ and therefore, $\mathbf{x}'^T\mathtt{F}\mathbf{x} = 0$.

This allows us to compute $\mathtt{F}$ from a set of point correspondences $\{\mathbf{x}_i \leftrightarrow \mathbf{x}'_i\}$.

Definition

If we have two images acquired by (linear) cameras with non-coincident centers, the fundamental matrix $\mathbf{F}$ is the unique 3×3 rank 2 homogeneous matrix satisfying

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

for all corresponding points $\mathbf{x} \leftrightarrow \mathbf{x}'$.

Properties of $\mathtt{F}$

(i) **Transpose:** If $\mathtt{F}$ is the fundamental matrix for cameras $(\mathtt{P}, \mathtt{P}')$, then $\mathtt{F}^T$ is the fundamental matrix for $(\mathtt{P}', \mathtt{P})$.

(ii) **Epipolar lines:** For any point $\mathbf{x}$ in image 1, the epipolar line is $\mathbf{l}' = \mathtt{F}\mathbf{x}$, and for any point $\mathbf{x}'$ in image 2, the epipolar line is $\mathbf{l} = \mathtt{F}^T\mathbf{x}'$.

(iii) The **epipole**: for any point $\mathbf{x}$ except $\mathbf{e}$, the epipolar line $\mathbf{l}' = \mathtt{F}\mathbf{x}$ contains the epipole $\mathbf{e}'$. Therefore $\mathbf{e}'^T(\mathtt{F}\mathbf{x}) = (\mathbf{e}'^T\mathtt{F})\mathbf{x} = 0$, so $\mathbf{e}'^T\mathtt{F} = \mathbf{0}$, i.e., $\mathbf{e}'$ is the left null-vector of $\mathtt{F}$ and $\mathbf{e}$ is the right null-vector of $\mathtt{F}$.

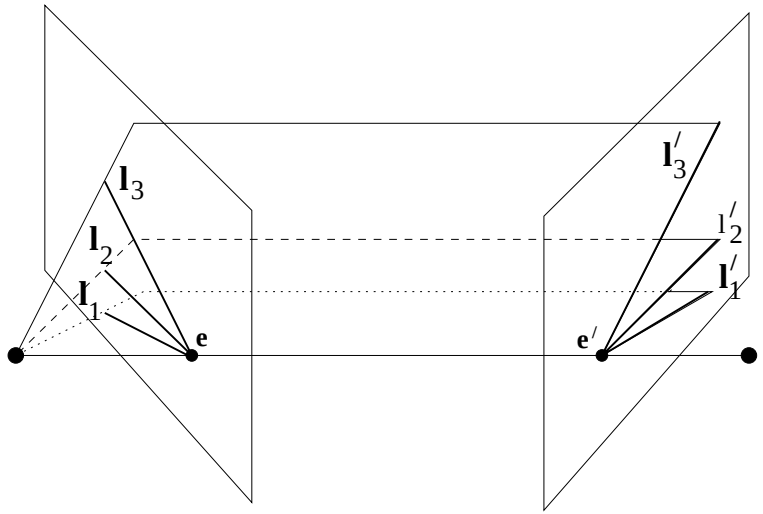
(iv) $\mathtt{F}$ has 7 degrees of freedom: 8 for a 3×3 homogeneous matrix minus one for the constraint that $\det \mathtt{F} = 0$.

(v) $\mathtt{F}$ is known as a correlation, which is a projective map taking a point to a line. If $\mathbf{l}$ and $\mathbf{l}'$ are corresponding epipolar lines, any point $\mathbf{x}$ on $\mathbf{l}$ is mapped to the same epipolar line $\mathbf{l}'$.

Epipolar geometry and the fundamental matrix

Epipolar lines correspond

All points on an epipolar line $\mathbf{l}$ in image 1 correspond to the same epipolar line $\mathbf{l}'$ in image 2.



Hartley and Zisserman (2004), Fig. 9.6a

Suppose we know know $\mathtt{P} = \mathtt{K}[\mathtt{I} \mid \mathbf{0}]$ and $\mathtt{P}' = \mathtt{K}'[\mathtt{R} \mid \mathbf{t}]$ (we have calibrated cameras).

This occurs, for example, in the case of a fixed stereo head.

$\mathtt{F}$ can be computed directly in this case as $\mathtt{F} = \left[\mathbf{e}'\right]_\times \mathtt{K}'\mathtt{R}\mathtt{K}^{-1}$.[1]

[Verify this is true by seeing how $\mathtt{F}$ constructed this way transforms point $\mathbf{x}$ in camera 1 ($\mathtt{P}$), and recall that in $\mathbb{P}^2$, the cross product of two points gives the line containing them.]

[1] Remember that

$$\left[\mathbf{a}\right]_\times = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix}$$

is a skew-symmetric matrix having $\mathbf{a}$ as its null space and can be used to convert the cross product operation into a matrix multiplication: $\mathbf{a} \times \mathbf{b} = \left[\mathbf{a}\right]_\times \mathbf{b}$.

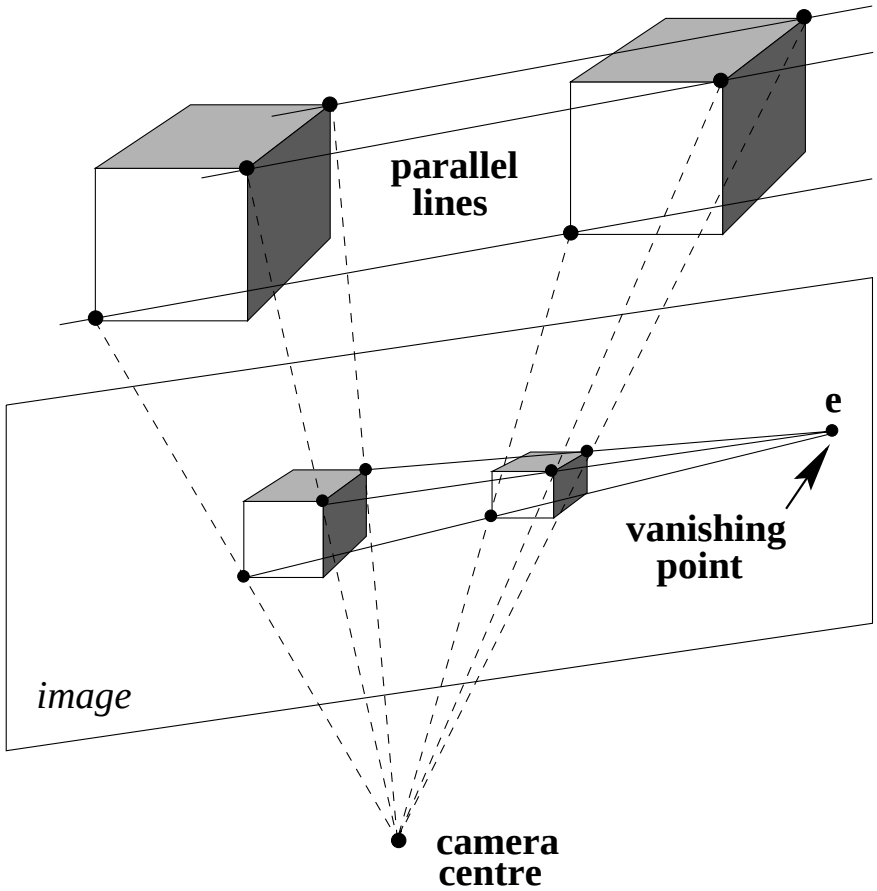
Epipolar geometry and the fundamental matrix

Special motion: translation

Special motions occur when particular relationships between the rotation and translation in a rigid motion hold.

Under the special motion of pure translation of the world $-\mathbf{t}$, the points move on straight lines parallel to $\mathbf{t}$.

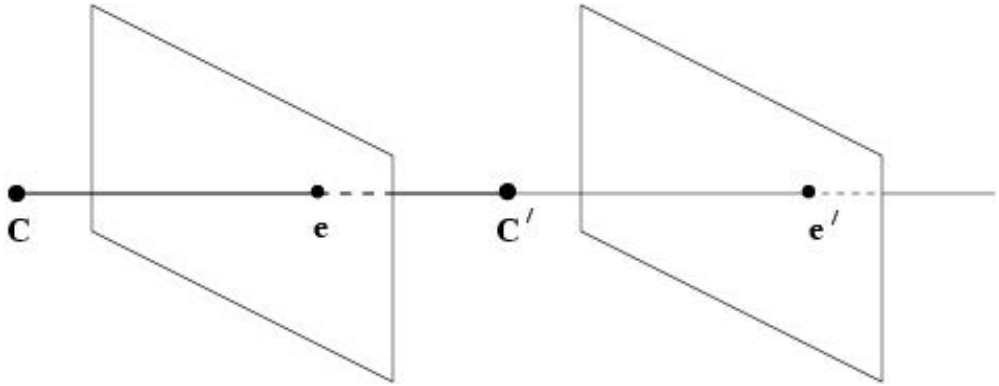
The 3D points slide along parallel rails under pure translation.



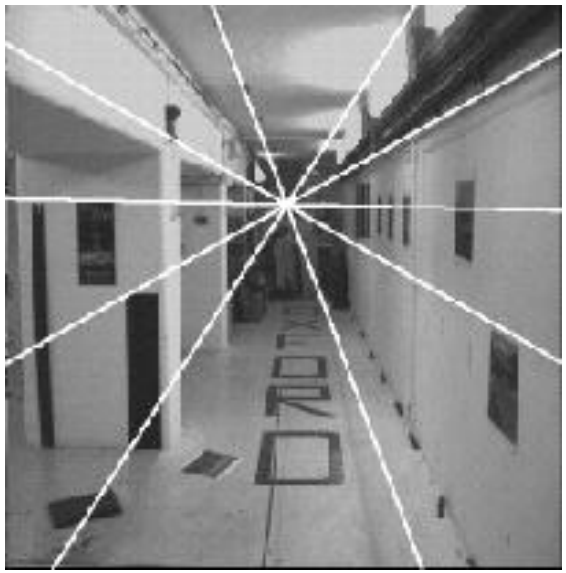
Hartley and Zisserman (2004), Fig. 9.7

Epipolar geometry and the fundamental matrix

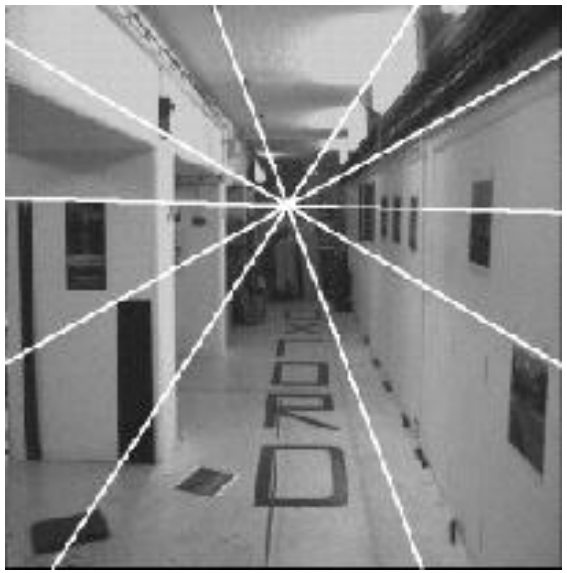
Special motion: translation



(a)



(b)



(c)

Hartley and Zisserman (2004), Fig. 9.8

Under pure translation, the epipole, called the focus of expansion, is the same in both images (we say the images are auto-epipolar).

Since we have no rotation and the cameras are the same, we can write $\mathtt{F} = \left[\mathbf{e}'\right]_\times \mathtt{K}\mathtt{K}^{-1} = \left[\mathbf{e}'\right]_\times$.

As an example, consider translation in the direction of the x axis. The epipole will be $(1, 0, 0)^T$ so we want

$$\mathtt{F} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix},$$

then $\mathbf{x} = (x_1, y_1, 1)^T$ gets mapped to $\mathbf{l}' = (0, -1, y_1)^T$ (i.e. $y_2 = y_1$).[2]

[2] Transforming images so that this is true is called rectification.

Epipolar geometry and the fundamental matrix

General motion

A general motion can always be decomposed as follows:

- Rotate the camera
- Correct the calibration matrix
- Translate the camera.

We know the rotation and calibration change is governed by a homography $\mathtt{H}$.

We now know that the pure translation is governed by $\hat{\mathtt{F}} = \left[\mathbf{e}'\right]_\times$.

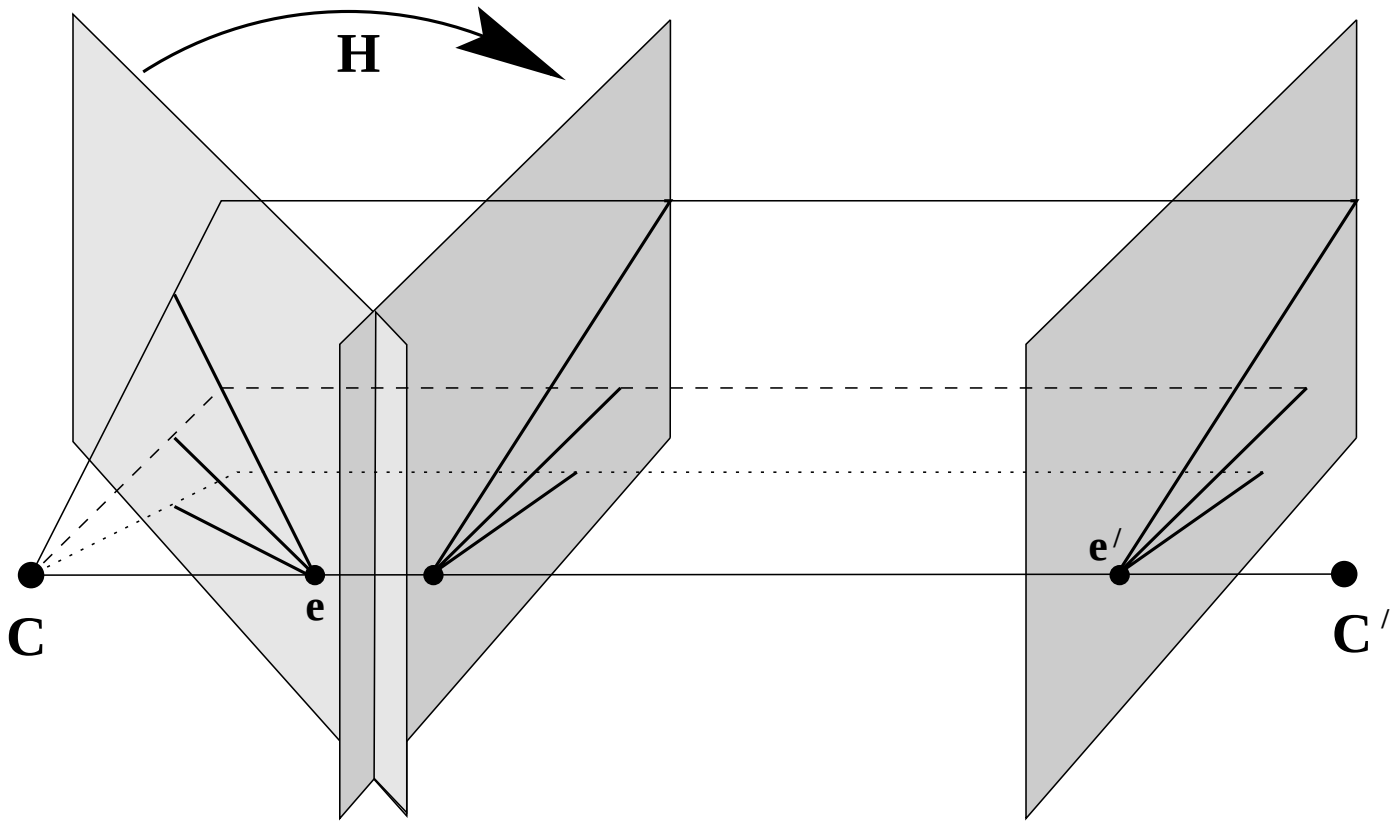
Putting these together, we obtain $\mathbf{x}'\hat{\mathtt{F}}\hat{\mathbf{x}} = 0$ where $\hat{\mathbf{x}} = \mathtt{H}\mathbf{x}$, so $\mathbf{x}'^T \left[\mathbf{e}'\right]_\times \mathtt{H}\mathbf{x} = 0$, and the final fundamental matrix is

$$\mathtt{F} = \left[\mathbf{e}'\right]_\times \mathtt{H}.$$

The effects of rotation and translation compose.

Epipolar geometry and the fundamental matrix

General motion



Hartley and Zisserman (2004), Fig. 9.9

Epipolar geometry and the fundamental matrix

Retrieving $\mathtt{P}, \mathtt{P}'$ from $\mathtt{F}$

Epipolar geometry does not depend on Euclidean measurements such as angles between rays.

$\mathtt{F}$ only depends on relationships between image points.

This means that $\mathtt{F}$ does not change under projective transformations on the world.

Put another way, the fundamental matrix for $(\mathtt{P}, \mathtt{P}')$ is the same as the fundamental matrix for $(\mathtt{PH}, \mathtt{P}'\mathtt{H})$.

A pair of camera matrices uniquely determines a fundamental matrix, but the converse is not true.

Epipolar geometry and the fundamental matrix

Retrieving $\mathtt{P}, \mathtt{P}'$ from $\mathtt{F}$

Since $\mathtt{P}$ and $\mathtt{P}'$ are not uniquely determined by $\mathtt{F}$, we will typically assume a canonical form where $\mathtt{P} = [\mathtt{I} \mid \mathbf{0}]$.

Then we will try to find $\mathtt{P}' = [\mathtt{M} \mid \mathbf{m}]$ consistent with $\mathtt{P}$ and $\mathtt{F}$.

It turns out that for this definition of $\mathtt{P}$, $\mathtt{P}'$, we can write $\mathtt{F} = \left[\mathbf{m}\right]_\times \mathtt{M}$.

(See text for proof, and see the definition of $\mathtt{F}$ for known $\mathtt{P}, \mathtt{P}'$.)

Let's consider the ambiguous relationship between $\mathtt{F}$ and $\mathtt{P}'$.

We just saw that $\mathtt{F}$ is invariant under projective transformations of 3-space.

It is also true that this is the only amgbiguity, i.e., $\mathtt{F}$ determines $\mathtt{P}$ and $\mathtt{P}'$ up to a projective transformation:

Projective ambiguity

If $\mathtt{F}$ is the fundamental matrix for both $(\mathtt{P}, \mathtt{P}')$ and $(\tilde{\mathtt{P}}, \tilde{\mathtt{P}}')$, then there exists a homography $\mathtt{H}$ such that $\tilde{\mathtt{P}} = \mathtt{P}\mathtt{H}$ and $\tilde{\mathtt{P}}' = \mathtt{P}'\mathtt{H}$.

The ambiguity in $\mathtt{P}$ and $\mathtt{P}'$ explained by $\mathtt{F}$ can also be understood through a degrees-of-freedom argument:

- The two cameras are 12-element homogeneous matrices with 11 degrees of freedom, for a total of 22 degrees of freedom.
- A homography has 15 degrees of freedom
- The fundamental matrix has 7 degrees of freedom.
- Once we know $\mathtt{F}$ and the homography $\mathtt{H}$ transforming the scene, all 22 degrees of freedom in $\mathtt{P}, \mathtt{P}'$ are fixed.

Epipolar geometry and the fundamental matrix

Retrieving $\mathtt{P}, \mathtt{P}'$ from $\mathtt{F}$

Now we are ready to calculate the canonical camera matrices from $\mathtt{F}$.

Given $\mathtt{F}$, we define $\mathtt{P} = [\mathtt{I} \mid \mathbf{0}]$ and $\mathtt{P}' = [\mathtt{SF} \mid \mathbf{e}']$, where $\mathbf{e}'$ is the epipole of image 2 (i.e. $\mathbf{e}'^T \mathtt{F} = \mathbf{0}$) and $\mathtt{S} = \left[\mathbf{s}\right]_\times$ is any skew-symmetric matrix.

$\mathtt{F}$ will be the fundamental matrix for $(\mathtt{P}, \mathtt{P}')$. A good choice for $\mathtt{S}$ is $\left[\mathbf{e}'\right]_\times$.

The most general solution making the ambiguity of $\mathtt{P}'$ explicit is

$$\mathtt{P} = [\mathtt{I} \mid \mathbf{0}] \qquad \mathtt{P}' = [\left[\mathbf{e}'\right]_\times \mathtt{F} + \mathbf{e}'\mathbf{v}^T \mid \lambda \mathbf{e}']$$

where $\mathbf{v}$ is any 3-vector and λ is a non-zero scalar.

So if we know $\mathtt{F}$, the cameras $\mathtt{P} = [\mathtt{I} \mid \mathbf{0}]$ and $\mathtt{P}' = [\left[\mathbf{e}'\right]_\times \mathtt{F} \mid \mathbf{e}']$ are fine choices for obtaining a projective reconstruction of the scene points.

Now we'll see that a metric reconstruction can be had if we know $\mathtt{K}$ and $\mathtt{K}'$.

The essential matrix is a specialization of the fundamental matrix to the case of normalized image coordinates.

If we know $\mathtt{K}$ and $\mathtt{K}'$, let $\hat{\mathbf{x}} = \mathtt{K}^{-1}\mathbf{x}$ and let $\hat{\mathbf{x}}' = \mathtt{K}'^{-1}\mathbf{x}'$.

Then the essential matrix is the matrix $\mathtt{E}$ satisfying

$$\hat{\mathbf{x}}'^{T}\mathtt{E}\hat{\mathbf{x}} = 0$$

This leads to the relationship between the fundamental and essential matrices:

$$\mathtt{E} = \mathtt{K}'^{T}\mathtt{F}\mathtt{K}.$$

Epipolar geometry and the fundamental matrix

Recovering $\mathtt{P}$, $\mathtt{P}'$ from $\mathtt{E}$

The camera matrices $\mathtt{P}$ and $\mathtt{P}'$ can be extracted from $\mathtt{E}$ up to scale and a four-fold ambiguity.

First, assume $\mathtt{P} = [\mathtt{I} \mid \mathbf{0}]$ and take the SVD $\mathtt{U} \cdot \mathrm{diag}(1, 1, 0) \cdot \mathtt{V}^T$ of normalized $\mathtt{E}$.

There will be four consistent choices for $\mathtt{P}'$:

$$\mathtt{P}' = [\mathtt{U}\mathtt{W}\mathtt{V}^T \mid +\mathbf{u}_3] \text{ or } [\mathtt{U}\mathtt{W}\mathtt{V}^T \mid -\mathbf{u}_3] \text{ or } [\mathtt{U}\mathtt{W}^T\mathtt{V}^T \mid +\mathbf{u}_3] \text{ or } [\mathtt{U}\mathtt{W}^T\mathtt{V}^T \mid -\mathbf{u}_3].$$

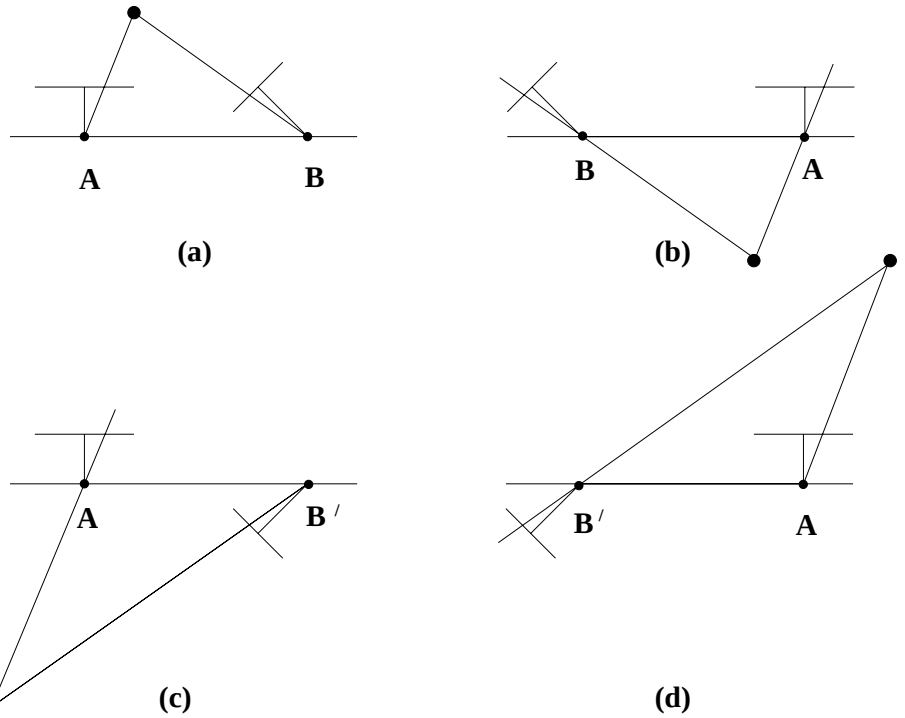
where

$$\mathtt{W} = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Epipolar geometry and the fundamental matrix

Recovering $\mathtt{P}$, $\mathtt{P}'$ from $\mathtt{E}$

Once we have the four possible solutions, we choose the one in which the reconstruction of some point $\mathbf{X}$ is visible in both images:



Hartley and Zisserman (2004), Fig. 9.12

The scale of the scene cannot be recovered, unless some metric property such as the baseline or the distance between two world points is known.

Outline

How can we reconstruct the 3D scene from a pair of 2D images?

Here is a general framework:

- Compute $\mathtt{F}$ from point correspondences, as described in Chapter 11.
- Compute $\mathtt{P}, \mathtt{P}'$ from $\mathtt{F}$, using the formula from the previous section.
- For each point correspondence $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$, triangulate to compute the point in space projecting to these two image points, as described in Chapter 12.
- The result will be a projective reconstruction of the 3D scene points $\mathbf{X}_i$.

There are many variations. For example, if $\mathtt{K}$ and $\mathtt{K}'$ are known we can compute $\mathtt{E}$ instead of $\mathtt{F}$ and obtain a metric reconstruction.

3D reconstruction overview

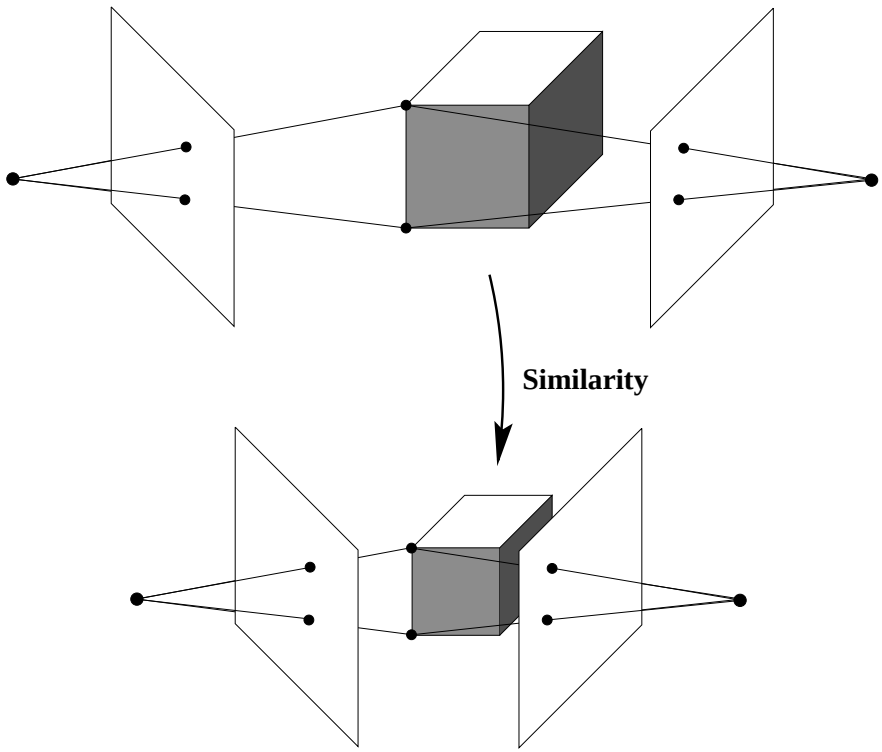
Reconstruction ambiguity

Normally we will assume that the scene is determined at best up to a Euclidean transformation with respect to the world frame.

I.e., we won't try to get latitude and longitude out of images!

The second ambiguity that we can't resolve from images alone is the size or scale of the scene.

In this case we say our reconstruction is up to a similarity transformation.



Hartley and Zisserman (2004), Fig. 10.2(a)

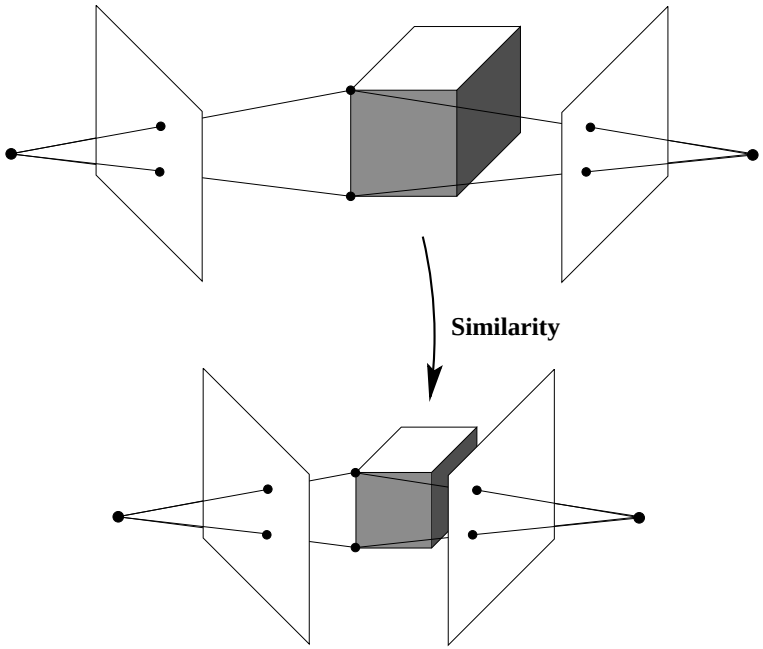
3D reconstruction overview

Reconstruction ambiguity

If we don't know the internal calibration of our cameras, we also have a projective ambiguity.

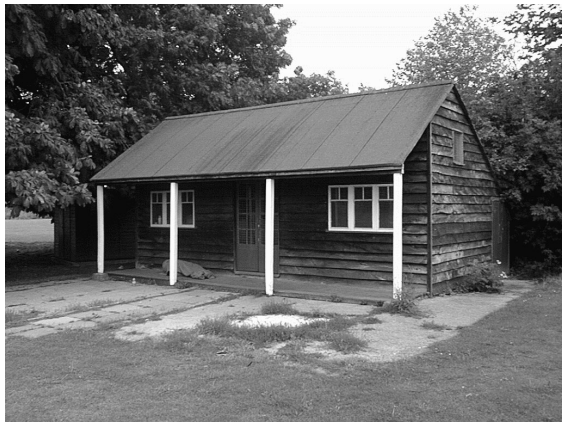
The projective reconstruction theorem states that this is the only ambiguity: if a set of point correspondences in two views determines $\mathbf{F}$ uniquely, the scene can be reconstructed, and any two such reconstructions are projectively equivalent.

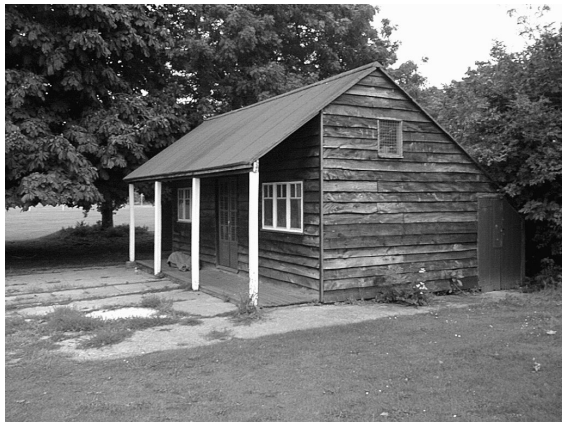
See text for proof. This is an important fact.

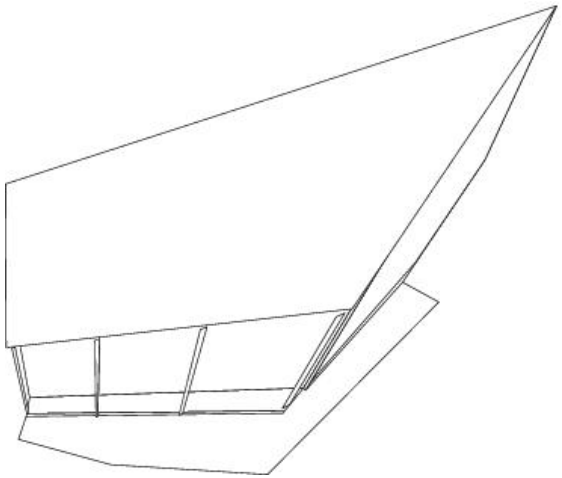


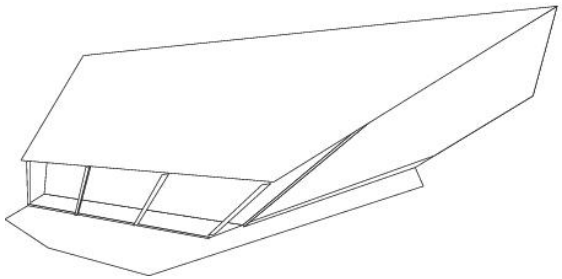
Hartley and Zisserman (2004), Fig. 10.2(a)

Example of possible projective reconstructions from two views:









Hartley and Zisserman (2004), Fig. 10.3

3D reconstruction overview

Various ways to upgrade a projective reconstruction

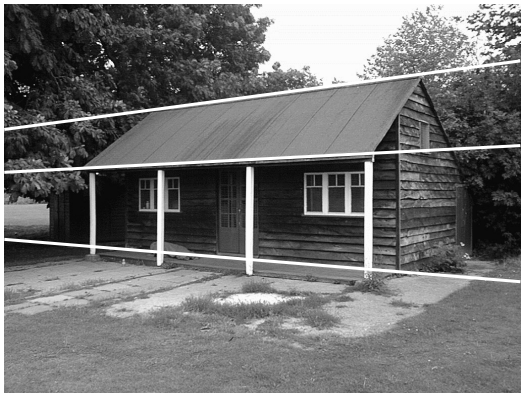
What can we do to turn a 2-view projective reconstruction into a more accurate reconstruction? We can look for scene constraints or apply some prior information.

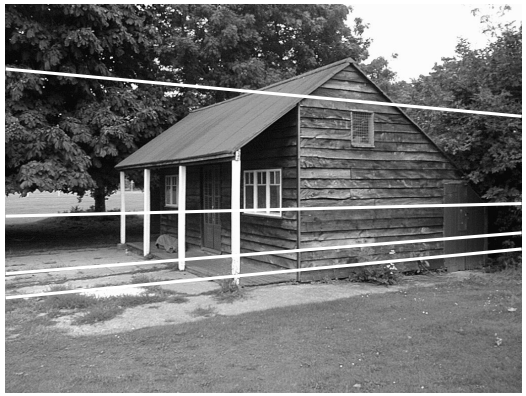
If the motion is pure translation, we can find the epipole (the convergence of the motion of the points) and obtain the cameras directly as $\mathtt{P} = [\mathtt{I} \mid \mathbf{0}]$ and $\mathtt{P}' = [\mathtt{I} \mid \mathbf{e}']$.

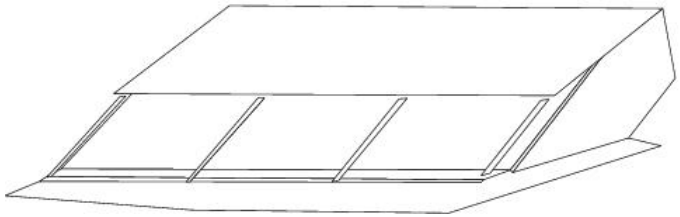
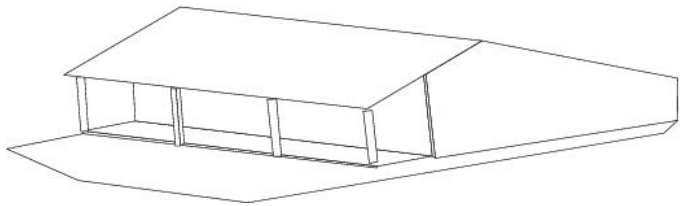
If three sets of parallel lines can be identified, the plane defined by their three points of intersection is the plane at infinity π_∞.

If we can find π_∞, a homography mapping the observed plane at infinity to its canonical position $(0, 0, 0, 1)$ in the affine frame will give us an affine reconstruction, correct up to an affine transformation.

Example affine reconstructions after mapping the plane at infinity:





Hartley and Zisserman (2004), Fig. 10.4

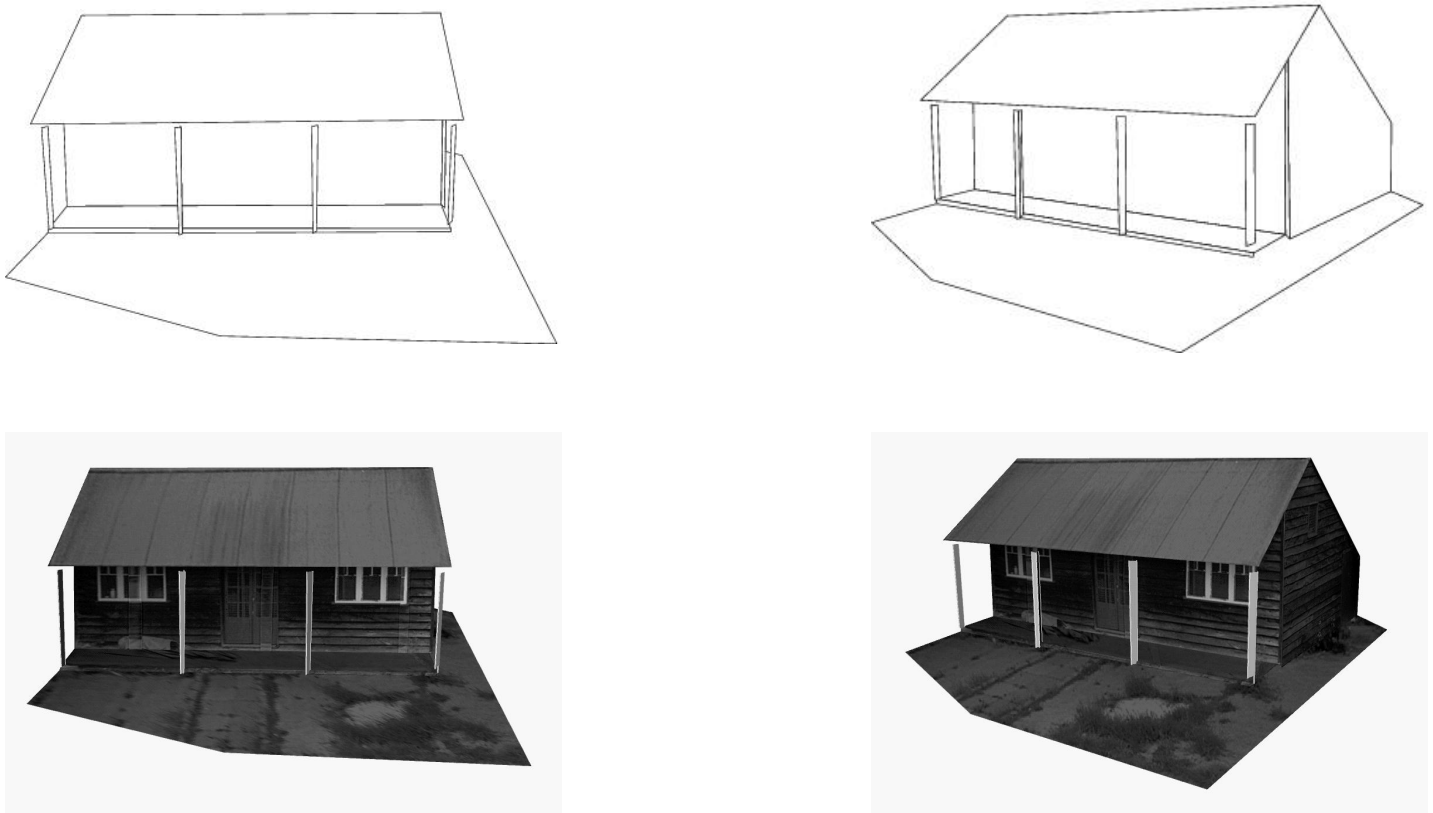
Once an affine reconstruction is had, if the absolute conic can be identified, it can be mapped to its canonical position on the plane at infinity.

This gives a metric reconstruction, correct up to a similarity.

The absolute conic can be determined, for example, by finding orthogonal lines and planes in the scene.

3D reconstruction overview

Metric reconstruction ambiguity



Hartley and Zisserman (2004), Fig. 10.5

Other methods include using known camera parameters or some knowledge of some ground truth relationships.

Among the most robust approaches to obtaining a metric reconstruction involves estimating the absolute dual quadric. We'll see this method in some detail later.

But all of these techniques require, to begin with,

- Finding good correspondences which is closely related to robust estimation of $\mathtt{F}$, and
- Triangulation.

These are the next two topics.

Outline

Computation of $\mathtt{F}$

Introduction

As we saw before, if we know $\mathtt{F}$ we can perform projective reconstruction from two views.

Another important use of $\mathtt{F}$ is in image rectification, in which we want to align the epipolar lines in two images in order to make the search for dense stereo correspondences simple.

We will obtain a set of point correspondences and use our now-familiar estimation tools to get $\mathtt{F}$.

As before, we'll start with correspondences $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$ and define a linear minimization of algebraic error.

Then we'll perform nonlinear estimation to obtain a ML (Gold Standard) estimate minimizing geometric error.

We'll finally look at robust estimation towards eliminating the bad correspondences (outliers) from the set of putative correspondences.

Computation of F

Estimating F by linear minimization of algebraic error

From the relationship

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

we write $\mathbf{x} = (x, y, 1)^T$ and $\mathbf{x}' = (x', y', 1)^T$ and expand the matrix equation in terms of the scalar elements.

We obtain

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

Writing F as a vector $\mathbf{f}$ in row-major order, we obtain the inner product

$$(x'x, x'y, x', y'x, y'y, y', x, y, 1)\mathbf{f} = 0.$$

and for n points we obtain the linear system

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

Computation of F

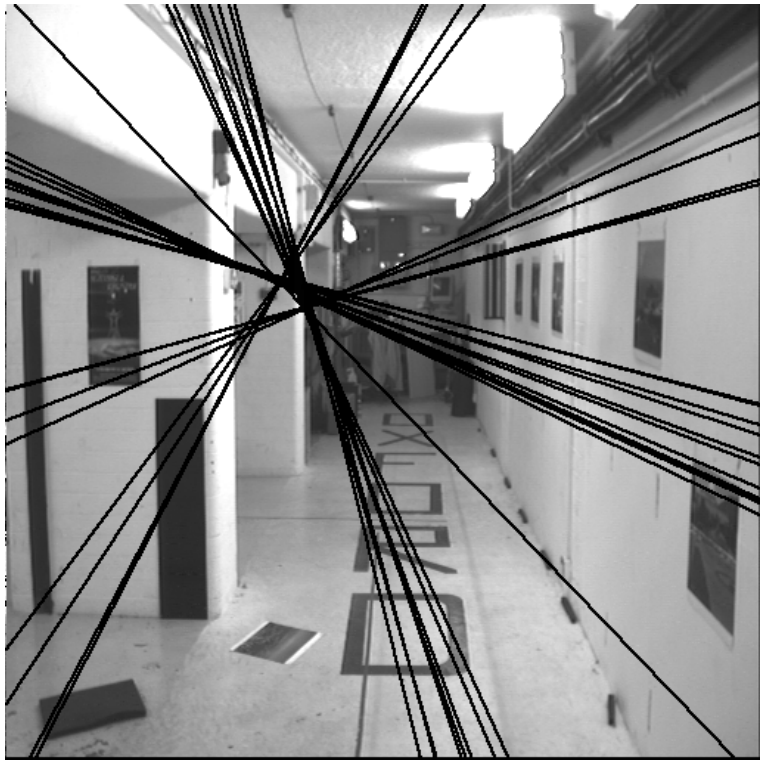
Estimating F by linear minimization of algebraic error: singularity constraint

We can obtain an exact solution with 8 points in a non-degenerate configuration as the right null space of A.

In the overdetermined case we choose the last right singular vector of A.

One problem is that there is a singularity constraint on F — it should be rank 2.

The linear estimate will not necessarily meet that constraint, and this could have consequences (picture on right).



Hartley and Zisserman (2004), Fig. 11.1(a)

Computation of F

Estimating F by linear minimization of algebraic error: enforcing the singularity constraint

We saw how SVD can be used to enforce orthonormal constraints on $\mathtt{R}$.

We use the SVD to force the linear estimate of $\mathtt{F}$ to the nearest rank 2 matrix.

Obtain $\mathtt{U}\mathtt{D}\mathtt{V}^T = \mathtt{F}$, then replace $\mathtt{F}$ with $\mathtt{F}' = \mathtt{U}\ \mathrm{diag}(\sigma_1, \sigma_2, 0)\mathtt{V}^T$ where σ_1 and σ_2 are the first two singular values in $\mathtt{D}$.

The result is guaranteed to be the rank 2 matrix closest to $\mathtt{F}$ in terms of Frobenius norm.

Hartley and Zisserman (2004), Fig. 11.1(b)

Computation of $\mathbf{F}$

8 points vs. 7 points

This algorithm, invented in 1981 by Longuet-Higgins, is called the 8-point algorithm for $\mathbf{F}$, as it requires 8 correspondences.

In fact, more correspondences can be used for an overdetermined solution.

Since $\mathbf{F}$ has 7 degrees of freedom (one is lost to homogeneity and one is lost to singularity), it is also possible to estimate $\mathbf{F}$ from 7 correspondences by using the singularity constraint and the 2-dimensional null space of the design matrix. This is called the 7-point algorithm for $\mathbf{F}$.

Computation of F

Normalized 8-point algorithm

As with previous DLT algorithms, the 8-point algorithm can perform poorly with arbitrary data, so prenormalization is required. We use the same isotropic scaling as for estimation of H.

Normalized 8-point algorithm for F: Objective

Given $n \geq 8$ image point correspondences $\{\mathbf{x}_i \leftrightarrow \mathbf{x}'_i\}$, determine the fundamental matrix F such that $\mathbf{x}'^T_i \mathbf{F} \mathbf{x}_i = 0$.

Normalized 8-point algorithm for $\mathtt{F}$: Algorithm

(i) **Normalization**: Transform $\hat{\mathbf{x}}_i = \mathtt{T}\mathbf{x}_i$ and $\hat{\mathbf{x}}'_i = \mathtt{T}'\mathbf{x}'_i$ with isotropic scaling matrices $\mathtt{T}$ and $\mathtt{T}'$.

(ii) Find $\hat{\mathtt{F}}'$ for $\hat{\mathbf{x}}_i \leftrightarrow \hat{\mathbf{x}}'_i$:

- (a) **Linear solution**: Calculate $\hat{\mathtt{A}}$ from the correspondences and obtain $\hat{\mathtt{F}}$ from its SVD.
- (b) **Singularity enforcement**: Replace $\hat{\mathtt{F}}$ by $\hat{\mathtt{F}}'$ such that $\det \hat{\mathtt{F}}' = 0$ using the SVD.

(iii) **Denormalization**: Return $\mathtt{F} = \mathtt{T}'^{T}\hat{\mathtt{F}}'\mathtt{T}$.

Computation of $\mathtt{F}$

Gold Standard method

The Gold Standard method tries to obtain the ML estimate of $\mathtt{F}$ under the assumption of Gaussian error. It minimizes the reprojection error

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

where $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$ are the measured correspondences and $\hat{\mathbf{x}}_i \leftrightarrow \hat{\mathbf{x}}'_i$ are estimated true correspondences exactly satisfying $\hat{\mathbf{x}}_i'^T \mathtt{F} \hat{\mathbf{x}}_i = 0$.

To perform the minimization, we posit camera matrices $\mathtt{P} = [\mathtt{I} \mid \mathbf{0}]$ and $\mathtt{P}' = [\mathtt{M} \mid \mathbf{t}]$ as well as 3D points $\mathbf{X}_i$.

Then we let $\hat{\mathbf{x}}_i = \mathtt{P}\mathbf{X}_i$ and $\hat{\mathbf{x}}'_i = \mathtt{P}'\mathbf{X}_i$ then we vary $\mathtt{P}'$ and $\mathbf{X}_i$ to minimize the error expression using Levenberg-Marquardt.

Finally we obtain $\mathtt{F} = \left[\mathbf{t}\right]_\times \mathtt{M}$, which will satisfy $\hat{\mathbf{x}}_i'^T \mathtt{F} \hat{\mathbf{x}}_i = 0$.

Computation of F

Sampson approximation to reprojection error

The Gold Standard method is a bit complex to implement, so we normally use simpler cost functions such as the Sampson distance

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

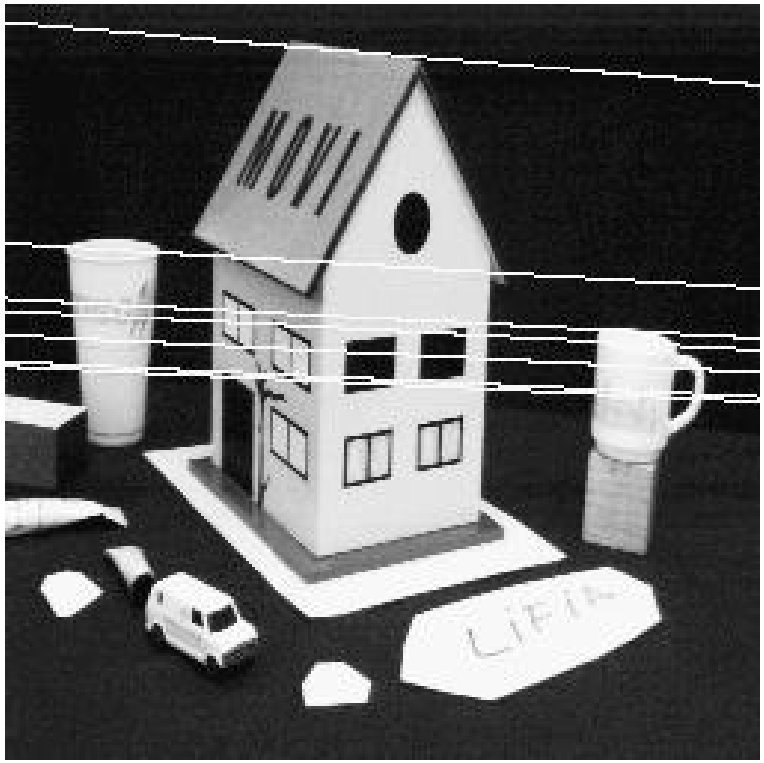
that do not involve subsidiary variables like $\mathbf{X}_i$ and $\mathbf{P}'$ in the Gold Standard method.

In practice Sampson performs about as well as MLE but the parameter vector is much smaller.

Computation of F

Experimental evaluation

See Section 11.5 of the text for an experimental evaluation of three algorithms (normalized 8-point, Gold Standard, and an iterative linear algorithm).



House images, Hartley and Zisserman (2004), Fig. 11.2

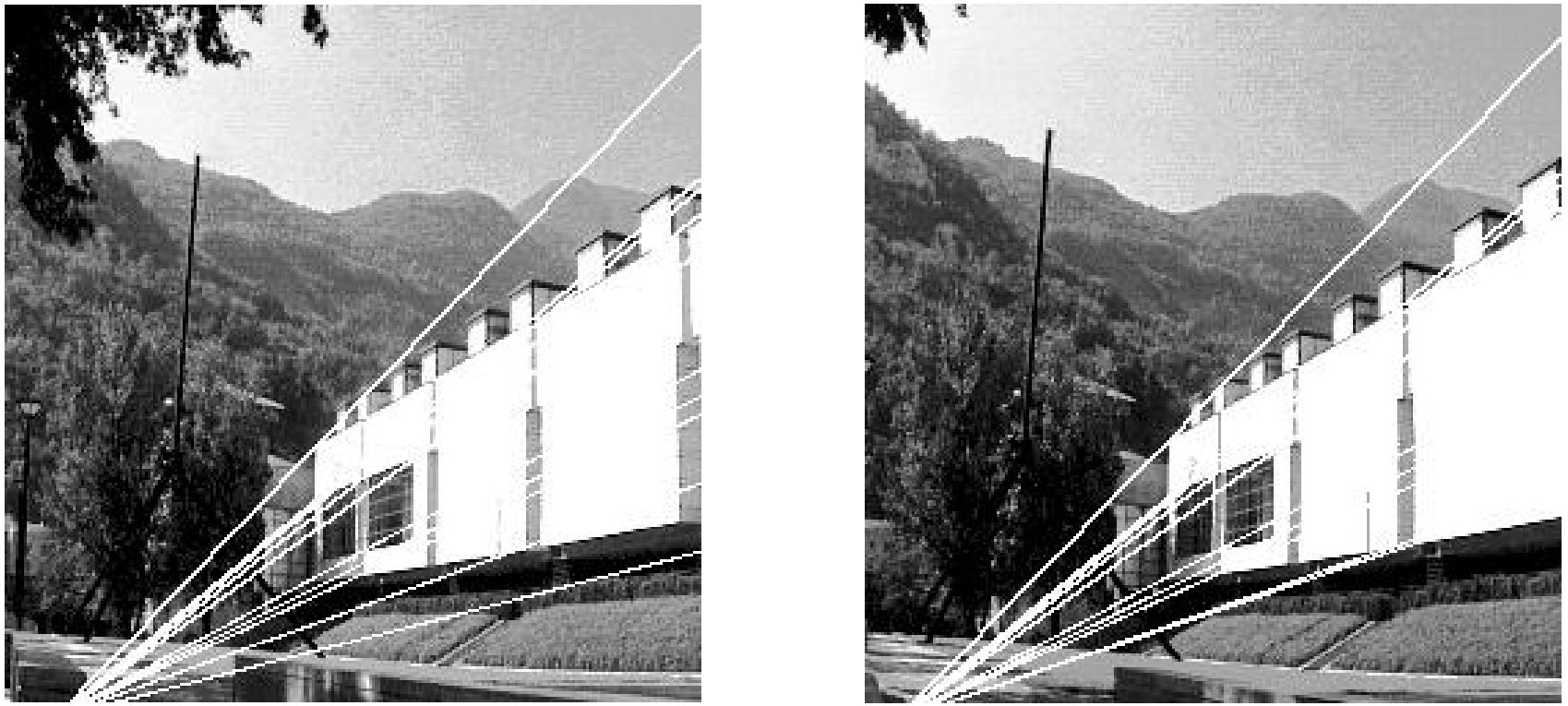
Computation of F

Experimental evaluation

Statue images, Hartley and Zisserman (2004), Fig. 11.2

Computation of F

Experimental evaluation



Grenoble Museum, Hartley and Zisserman (2004), Fig. 11.2

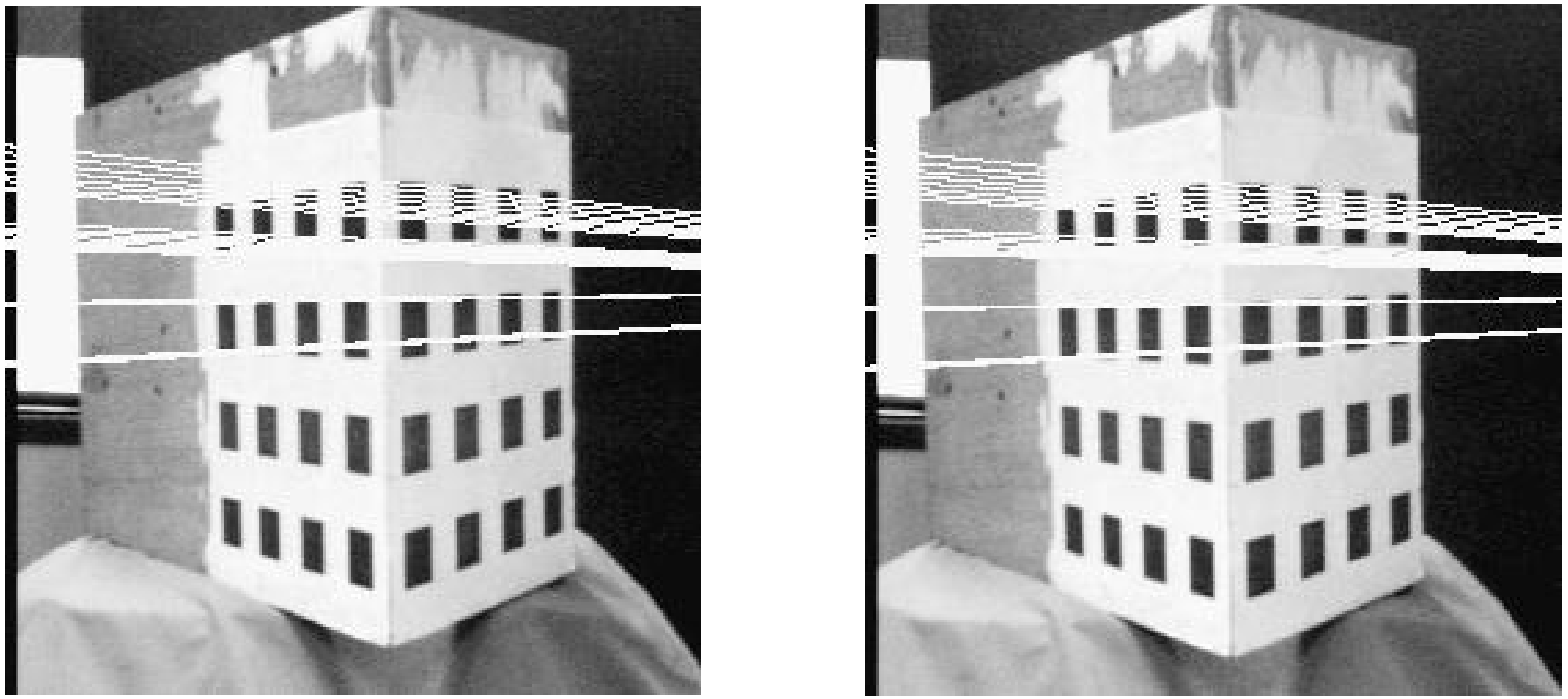
Computation of F

Experimental evaluation

Corridor scene, Hartley and Zisserman (2004), Fig. 11.2

Computation of F

Experimental evaluation

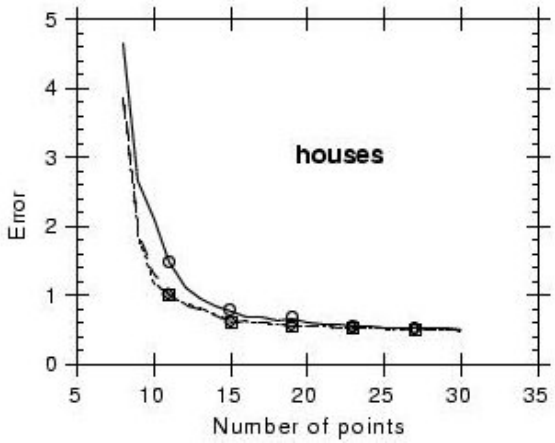


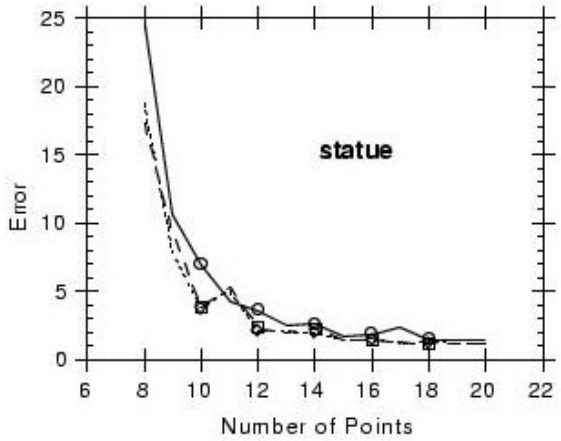
Calibration rig, Hartley and Zisserman (2004), Fig. 11.2

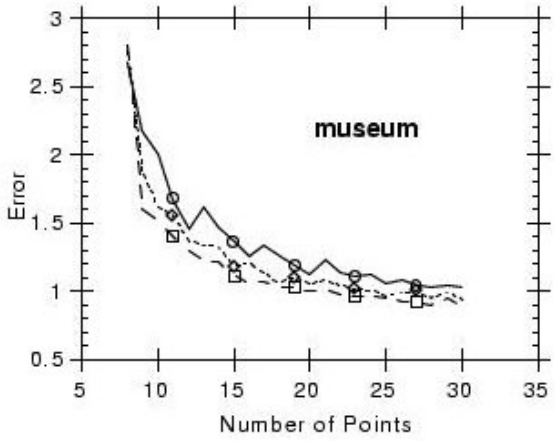
Computation of F

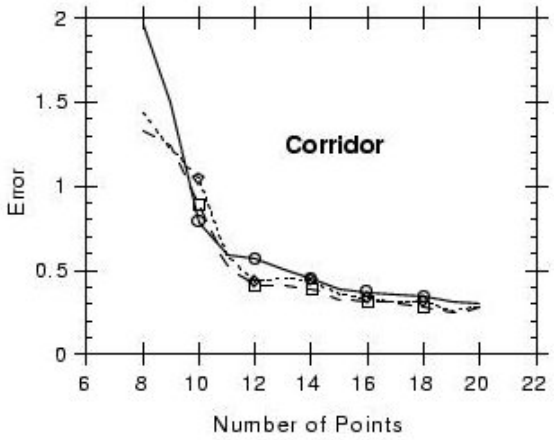
Experimental evaluation

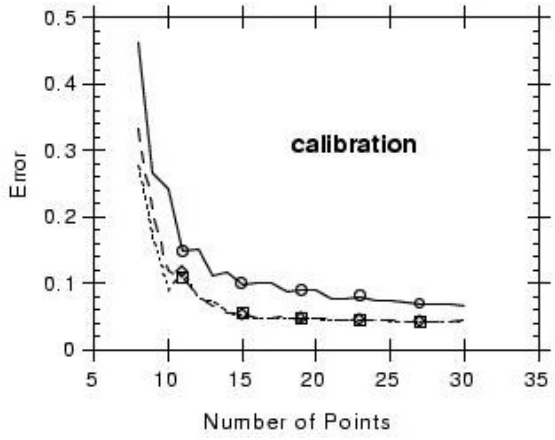
Results: solid = normalized 8-point, long dashed = Gold Standard, dotted = iterative. Note low error for "calibration."











Hartley and Zisserman (2004), Fig. 11.3

Computation of F

Automatic computation of F

As in the case of homography estimation, if we're extracting correspondences automatically, we will have many outliers.

The idea is to use RANSAC to eliminate outliers.

In this case we use the 7-point algorithm because it takes fewer random samples to ensure an inlier set.

The disadvantage is that the 7-point algorithm has 3 solutions, all of which need to be tried for support.

For the distance measure $d_\perp$ any geometric error measure can be used but it should be used consistently.

The Sampson approximation to geometric error previously described is a good candidate.

Automatic F estimation: Objective

Compute the fundamental matrix between two images.

Automatic F estimation: Algorithm

(i) **Interest points**: Get a set of interest points in each image

(ii) **Putative correspondences**: Compute an initial set of correspondence using proximity and similarity

(iii) **RANSAC robust estimation**: Repeat for N samples

- (a) Select 7 correspondences randomly and compute F
- (b) Calculate the distance $d_\perp$ for each correspondence
- (c) Compute the number of inliers for F ($d_\perp < t$)
- (d) Repeat for each of the possible solutions from the 7-point algorithm

Choose the F highest support.

(iv) **Nonlinear estimation**: Reestimate F from all inliers minimizing geometric error with Levenberg-Marquardt

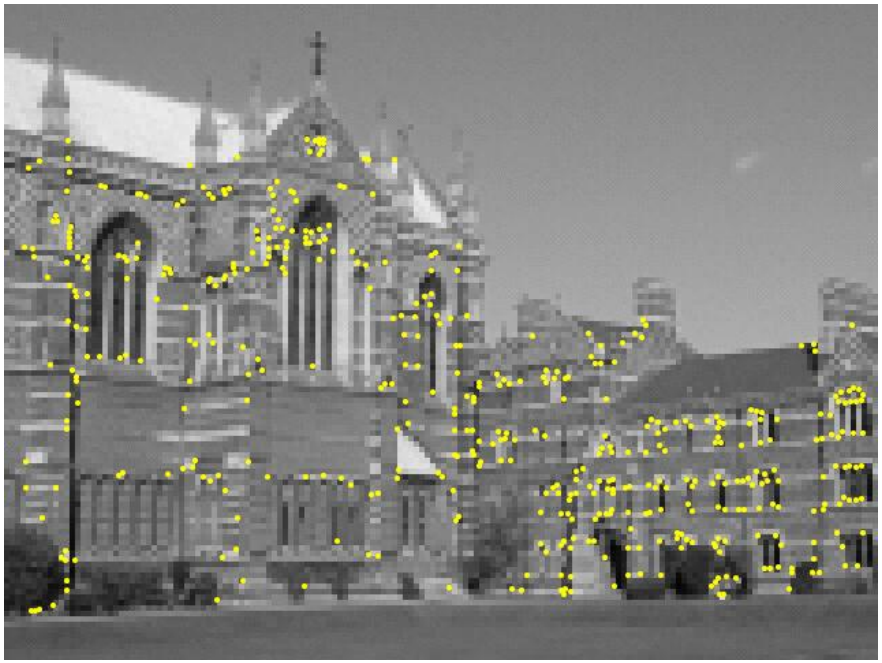
(v) **Guided matching**: Determine additional correspondences within search strips around epipolar lines.

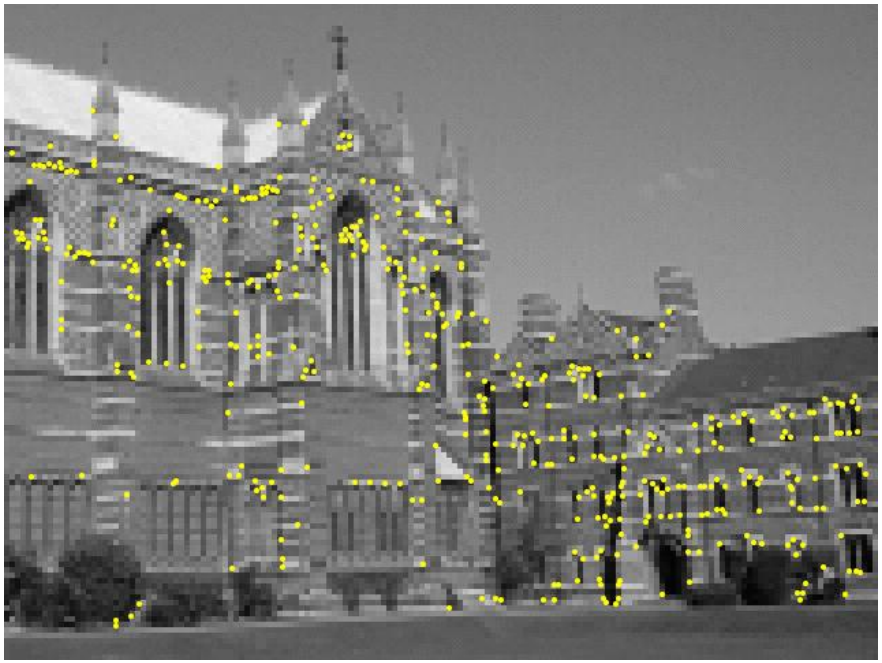
Here are results of Hartley and Zisserman's experimental evaluation of the automatic F estimation algorithm for a translation motion:

Original image pair, Hartley and Zisserman (2004), Fig. 11.4

Computation of F

Automatic computation of F: Experiment

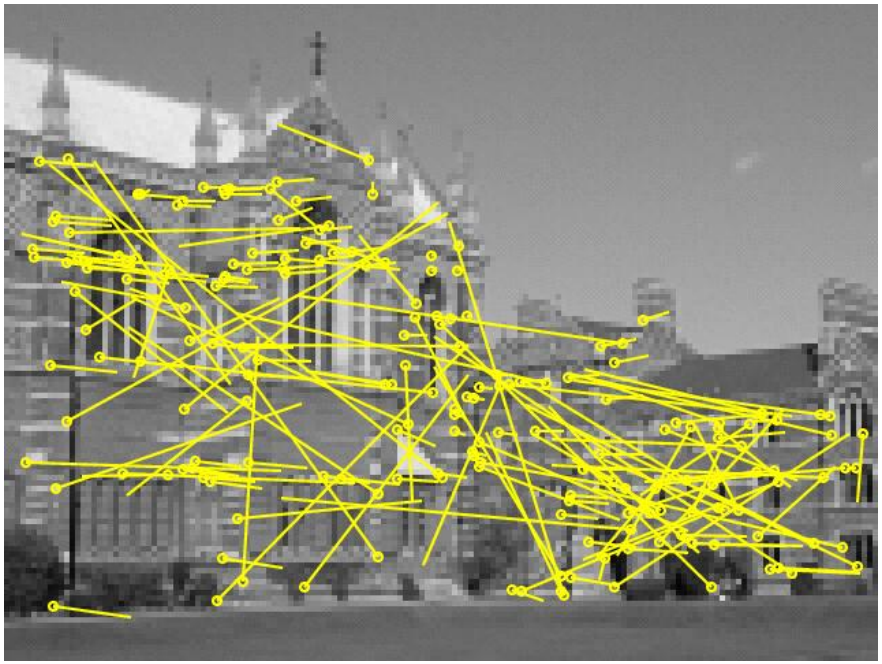




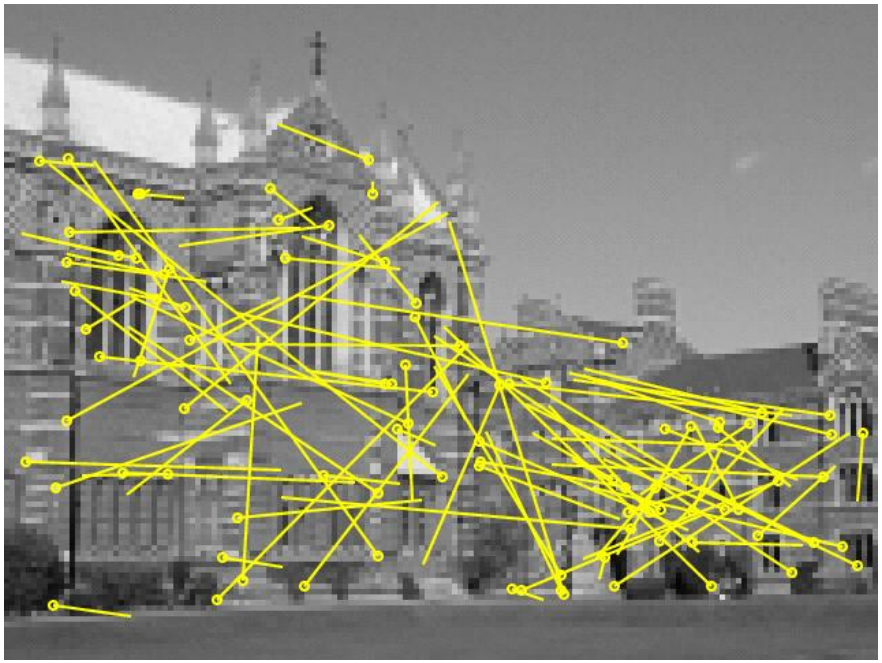
Detected corners, Hartley and Zisserman (2004), Fig. 11.4

Computation of F

Automatic computation of F: Experiment



188 putative matches

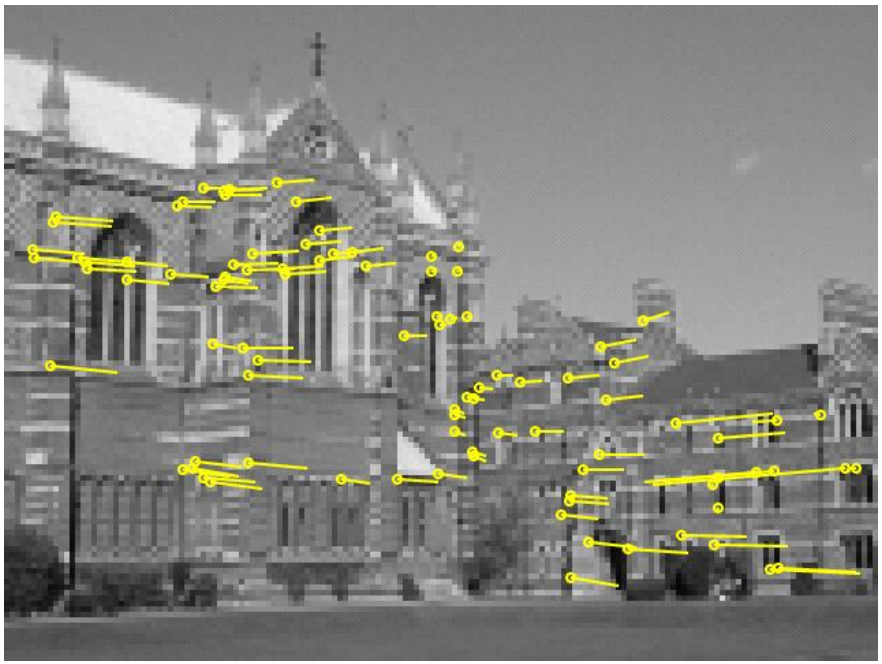


Outliers according to RANSAC (89/188)

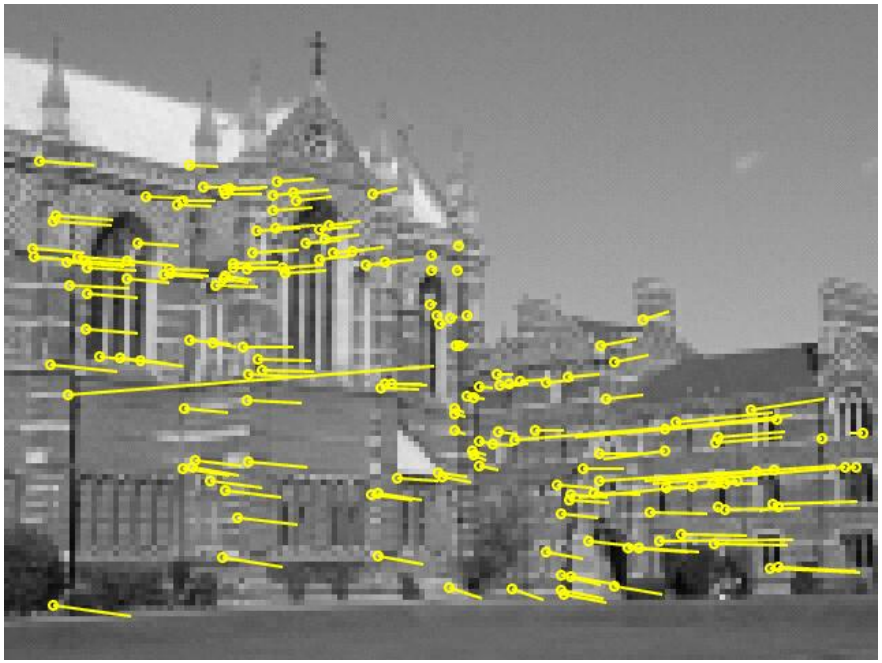
Hartley and Zisserman (2004), Fig. 11.4

Computation of F

Automatic computation of F: Experiment



Inliers according to RANSAC (99/188)



Final set of 157 matches from guided matching

Hartley and Zisserman (2004), Fig. 11.4

Notice there is at least one error in the guided match set.

Outline

Sparse correspondence with SIFT

Introduction

Notice that many of the techniques we've looked at thus far require a set of point correspondences $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$.

See the Torr toolbox for a practical implementation using Harris corners with the robust RANSAC-based method for $\mathtt{F}$.

The methods generally rely on minimal camera motion in which case local image similarity techniques work well.

However, when there is significant motion between two images, a rich feature detector with invariance to rotation and scale is desirable.

Lowe's (2004) Scale Invariant Feature Transform (SIFT) extracts feature points with a vector of attributes that is invariant to scale and rotation in the plane and robust to moderate amounts of various kinds of affine distortion and noise.

Sparse correspondence with SIFT

SIFT steps

SIFT (Lowe, 2004) performs four basic steps:

- **Scale-space extrema detection:** A difference-of-Gaussian filter is run at several scales to find points that are local maxima or minima in space and scale.
- **Keypoint localization:** For each candidate location, the scale and location is determined, and unstable locations are discarded.
- **Orientation assignment:** Orientations are assigned to keypoint locations.
- **Keypoint descriptor:** A descriptor of the image intensities around the keypoint location is computed. The descriptors are obtained relative to the keypoint's location, orientation, and scale, so that the descriptor is invariant to scales, rotations (in the image plane), and translations.

The scale space is obtained by convolving Gaussians of various sizes with the input image:

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y)$$

where $*$ is convolution and the 2D Gaussian is defined by

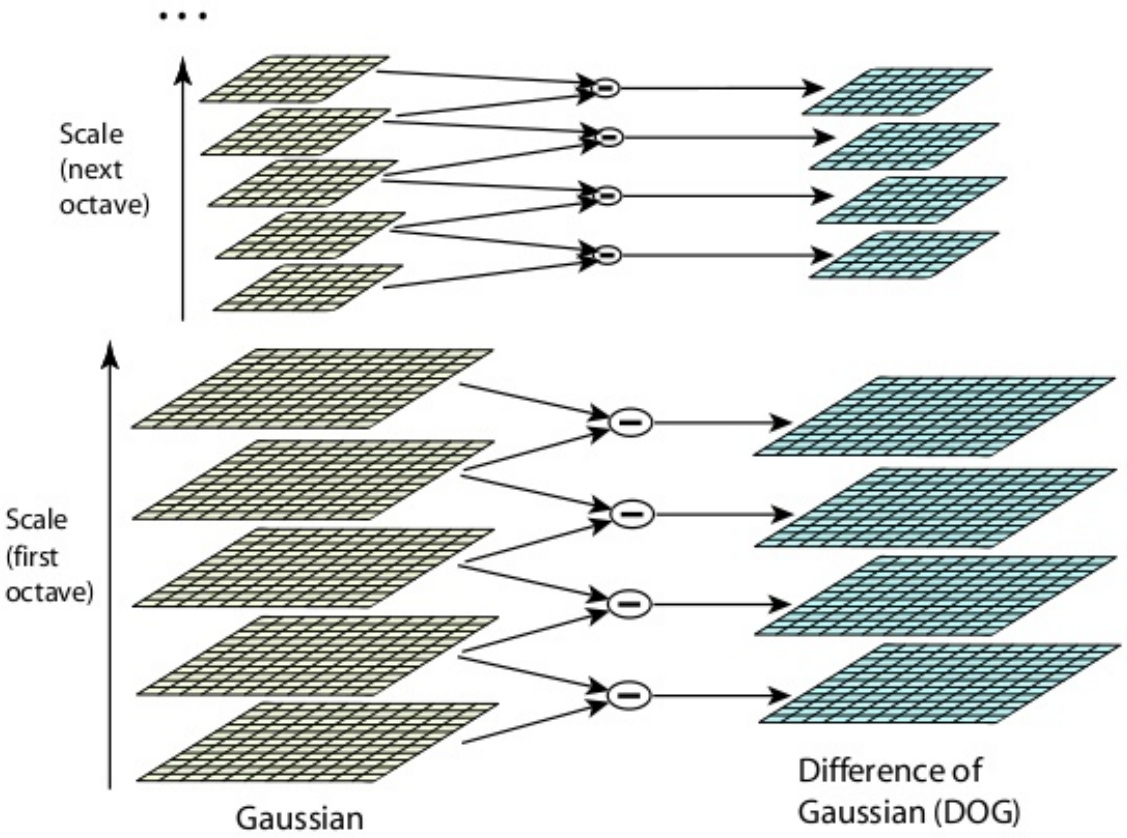
$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/(2\sigma^2)}.$$

Stable keypoints are identified using the difference of Gaussian function:

$$\begin{aligned} D(x, y, \sigma) &= (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y) \\ &= L(x, y, k\sigma) - L(x, y, \sigma) \end{aligned}$$

Sparse correspondence with SIFT

SIFT scale space

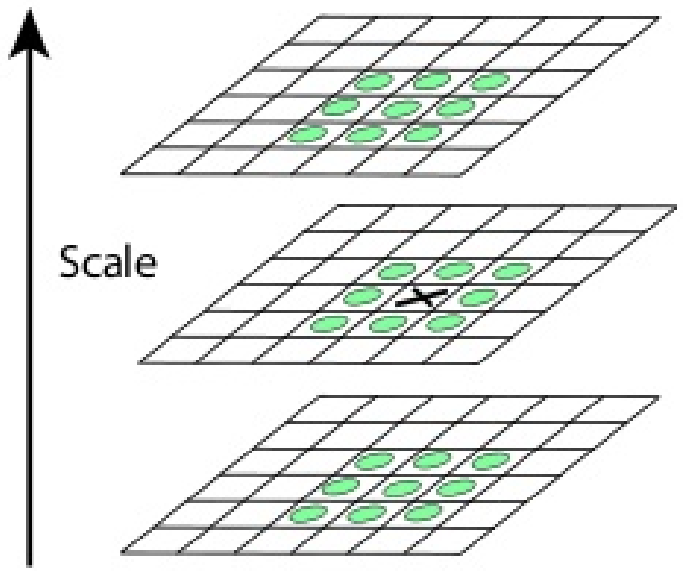


Lowe (2004), Fig. 1

Sparse correspondence with SIFT

SIFT scale space

Candidate keypoint locations are maxima and minima of the difference-of-Gaussian scale space in x, y, and scale:



Lowe (2004), Fig. 2

Sparse correspondence with SIFT

SIFT keypoint localization

For the localization step, we could just take the location and scale at which the point was detected.

Instead, Lowe fits a quadratic function to the local values of $D(x, y, \sigma)$ then obtains a sub-pixel estimate of the location of the extremum, $\hat{\mathbf{x}}$.

Low-contrast extrema are immediately discarded.

Then, the eigenvalues of the 2×2 Hessian matrix around $\hat{\mathbf{x}}$ are examined to determine whether the extremum reflects a simple edge or a more complex corner-like region.

Simple edge-like candidate keypoints are discarded.

Sparse correspondence with SIFT

SIFT orientation assignment

Using the Gaussian smoothed image at the scale closest to the detected difference of Gaussian extremum, we collect a histogram of image gradients in 36 directions in the region around the point.

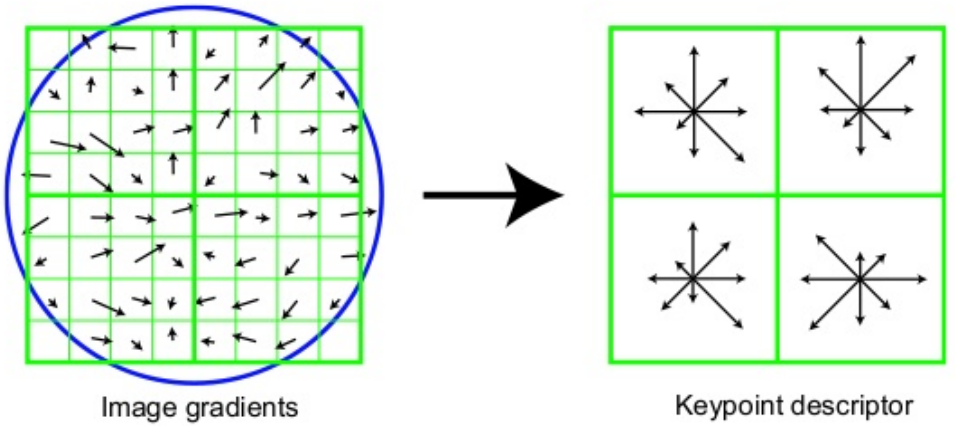
For the highest peak in the orientation histogram and any other strong peaks in the orientation histogram, SIFT creates a keypoint descriptor with that orientation.

This means we can get multiple keypoints for the same location (x, y, σ), but this only happens approximately 15% of the time.

Sparse correspondence with SIFT

SIFT keypoint descriptor

Finally, we compute a set of local histograms of gradient directions, relative to the dominant orientation.



Lowe (2004), Fig. 7

This is a 2x2 grid from a 16×16 sample array, but the standard implementations use 4×4 descriptor grid from a 16×16 sample array, to obtain a 128-element descriptor.

The blue circle is a Gaussian weighting window.

Sparse correspondence with SIFT

SIFT

Lowe's main goal was to perform object recognition in cluttered environments, but several computer vision and robotics groups have found SIFT useful for wide baseline matching to get correspondences for other algorithms.

Next we look at an example from a possible application in robotics.

Sparse correspondence with SIFT

Experiment

Akash Dev Nakarmi collected a series of 21 pictures at the AIT golf course:



- The camera, a Sony DSC-200, was approximately 50cm from the ground.
- Between each image, the camera was moved forward about 25cm.
- Small random fluctuations in rotation were allowed.

Sparse correspondence with SIFT

Experiment

I ran the images through Rob Hess' SIFT implementation `siftfeat`, which is based on OpenCV.

Go to `http://web.engr.oregonstate.edu/~hess/` to download this excellent open source software.

The output is an array of `structs` or a text file containing the number of detected features, the number of attributes per feature, and the data:

```
3062 128
228.141322 455.408286 43.929582 1.980753
 22 0 0 0 0 0 0 26 42 2 0 1 2 4 8 68 35 6 0 0
 10 28 64 81 92 84 1 0 7 11 41 94 87 2 0 4 2 0 0 55
 89 31 10 66 41 1 4 47 145 49 5 8 32 87 23 43 26 116 3 1
 57 130 10 4 61 2 1 18 25 2 3 64 123 8 4 92 73 2 1 35
 145 27 1 4 24 32 6 43 21 6 0 0 84 107 5 3 1 0 0 3
 10 0 1 6 52 0 0 6 18 0 0 31 145 7 0 0 0 3 4 82
 9 3 0 0 1 9 4 7
...
```

Sparse correspondence with SIFT

Experiment

AIT golf course image 0 with 3062 SIFT features detected.

Sparse correspondence with SIFT

Experiment

AIT golf course image 1 with 2993 SIFT features detected.

Sparse correspondence with SIFT

Experiment

AIT golf course image 2 with 3042 SIFT features detected.

Sparse correspondence with SIFT

Experiment

To obtain initial sets of putative correspondences I ran the SIFT features through Hess' implementation of the k-D tree-based best bin first (BBF) approximate nearest neighbor search (Beis and Lowe, 1997).

Using a distance ratio threshold of 0.49 I obtained 555 matches for image pair $(0, 1)$ and 467 matches for pair $(1, 2)$.

The nearest neighbor search is one-way and does not remove duplicate matches. I removed duplicate matches as a post-process.

After de-duping I had 547 and 449 unique putative matches for image pair $(0, 1)$ and $(1, 2)$, respectively.

I ran the resulting putative matches through OpenCV's implementation of RANSAC-based fundamental matrix estimation with a $d_\perp$ threshold of 0.5 pixels and a 99% confidence interval for the number of samples to attempt.

The result was 284 inliers for pair $(0, 1)$ and 256 inliers for pair $(1, 2)$.

Sparse correspondence with SIFT

Experiment

Inliers for pair $(0, 1)$:

Image 0 matches to image 1

Image 1 matches from image 0

Sparse correspondence with SIFT

Experiment

Inliers for pair $(1, 2)$:

Image 1 matches to image 2

Image 2 matches from image 1

Outline

Rectification

Introduction

In dense correspondence estimation, we like to have our image pairs aligned so that the epipolar lines are aligned with the rows of the images.

We will use the fundamental matrix to obtain a pair of 2D homographies matching the epipolar lines with the image rows and arranging things so that corresponding points have similar x coordinates.

The resulting images can be used for dense stereo matching.

The homography technique only works when the epipoles are not inside the image.

When the epipoles are inside the image we require more sophisticated methods such as polar rectification (Pollefeys, ICCV 1999).

Rectification

Mapping **e** to infinity

A main part of the method is to find homography $\mathtt{H}$ mapping the epipole to a point at infinity.

In particular, for the epipolar lines to be aligned with the x axis, the epipole should be at $(1, 0, 0)$.

(Recall that for a pure translation the epipoles will be the same point in both images.)

The constraint that $\mathtt{H}\mathbf{e} = (1, 0, 0)$ will leave us 4 degrees of freedom for selecting $\mathtt{H}$.

We will use the remaining degrees of freedom to ensure that the original image looks similar to the original image.

Rectification

Mapping **e** to infinity

Suppose $\mathbf{x}_0$ be the origin and $\mathbf{e} = (f, 0, 1)^T$ is on the x axis already. Then the transformation

$$\mathtt{G} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -1/f & 0 & 1 \end{bmatrix}$$

will take $\mathbf{e}$ to $(f, 0, 0)^T$ as we would like, and an arbitrary point $(x, y, 1)^T$ is mapped to $(\hat{x}, \hat{y}, 1)^T = (x, y, 1 - x/f)^T$.

For the points "inside" the epipole, i.e., $|x/f| < 1$, we can write the approximation[3]

$$(\hat{x}, \hat{y}, 1)^T = (x, y, 1 - x/f)^T = (x(1 + x/f + \ldots), y(1 + x/f + \ldots), 1)^T \quad (1)$$

[3] The Taylor expansion of $f(x)$ about a point a is $f(a) + f'(a)(x - a) + \frac{f''(a)/2}{(x-a)^2} + \cdots$ so the second-order approximation of $\frac{x}{1-x/f}$ around $x = 0$ is $x + x^2/f$.

Rectification

Mapping **e** to infinity

To understand what $\mathtt{G}$ is doing, we find the Jacobian of the mapping:

$$\frac{\partial(\hat{x},\hat{y})}{\partial(x,y)} = \begin{bmatrix} 1+2x/f & 0 \\ y/f & 1+x/f \end{bmatrix}.$$

This means that near the origin, where $x = y = 0$, we have an identity map.

We can say, then, that $\mathtt{G}$ maps, to first order, points around the origin to the same points in the transformed image.

If we use $\mathtt{G}$, then, we want to adjust the origin to be the image point where we want minimal projective distortion (usually the center of the image).

For the desired origin $\mathbf{x}_0$ and epipole $\mathbf{e}$, then, we want a transformation $\mathbf{H} = \mathtt{GRT}$ where $\mathtt{T}$ takes $\mathbf{x}_0$ to the origin, $\mathtt{R}$ rotates $\mathbf{e}$ to $(f,0,1)^T$, and $\mathtt{G}$ takes $(f,0,1)^T$ to infinity.

Rectification

Matched transformation

To perform a corresponding rectifying transform on a second image, consider that we have two images J and J'.

We need to perform on J a transformation $\mathtt{H}$ and on J' a transformation $\mathtt{H}'$ so that

- the epipolar lines are matched,
- the center of the image is minimally distorted, and
- the disparity (motion along the x direction) is minimized.

To match the epipolar lines, recalling that $\mathtt{H}^{-T}$ is the line map corresponding to a point map $\mathtt{H}$, we want to enforce $\mathtt{H}^{-T}\mathbf{l} = \mathtt{H}'^{-T}\mathbf{l}'$.

To minimize disparity, we'll first pick $\mathtt{H}'$ then try to find a $\mathtt{H}$ minimizing

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

Hartley and Zisserman prove that any $\mathtt{H}$ and $\mathtt{H}'$ minimizing this cost function, for fundamental matrix $\mathtt{F} = \left[\mathbf{e}'\right]_\times \mathtt{M}$, must obey

$$\mathtt{H} = (\mathtt{I} + \mathtt{H}'\mathbf{e}'\mathbf{a}^T)\mathtt{H}'\mathtt{M}$$

for some vector $\mathbf{a}$.

In our case we further impose that $\mathtt{H}'$ should map $\mathbf{e}'$ to $(1,0,0)^T$, so $\mathtt{I} + \mathtt{H}'\mathbf{e}'\mathbf{a}^T = \mathtt{I} + (1,0,0)^T\mathbf{a}^T$ turns out to be

$$\mathtt{H}_\mathtt{A} = \begin{bmatrix} a & b & c \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

which is an affine transformation.

Now we know that given $\mathtt{H}'$ we must let $\mathtt{H} = \mathtt{H}_\mathtt{A}\mathtt{H}_0$ where $\mathtt{H}_0 = \mathtt{H}'\mathtt{M}$.

Rectification

Matched transformation

OK! So given H' mapping $\mathbf{e}'$ to infinity, we write $\hat{\mathbf{x}}'_i = \mathrm{H}'\mathbf{x}'_i$ and $\hat{\mathbf{x}}_i = \mathrm{H}_0\mathbf{x}_i$ then find H_A minimizing

$$\sum_i d(\mathrm{H}_\mathrm{A}\hat{\mathbf{x}}_i, \hat{\mathbf{x}}'_i)^2$$

To solve this problem, we write $\hat{\mathbf{x}}_i = (\hat{x}_i, \hat{y}_i, 1)^T$ and $\hat{\mathbf{x}}'_i = (\hat{x}'_i, \hat{y}'_i, 1)^T$ using matched points $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$. Then the cost function can be written

$$\sum_i (a\hat{x}_i + b\hat{y}_i + c - x'_i)^2 + (\hat{y}_i - \hat{y}'_i)^2$$

but since $\hat{y}_i - \hat{y}'_i$ is constant (the epipolar lines are parallel), we just minimize

$$\sum_i (a\hat{x}_i + b\hat{y}_i + c - x'_i)^2$$

using standard least-squares estimation.

Here's the algorithm summary:

Rectification: Objective

Given images J and J', return a resampled pair of images in which the epipolar lines are parallel to the x axis and matched, and the disparity between corresponding points in the resampled images is minimized.

Rectification: Algorithm

(i) Identify corresponding points $\mathbf{x}_i \leftarrow \mathbf{x}'_i$.

(ii) Compute $\mathtt{F}$ and extract $\mathbf{e}$, $\mathbf{e}'$.

(iii) Select $\mathtt{H}'$ mapping $\mathbf{e}'$ to $(1,0,0)^T$.

(iv) Find $\mathtt{H}$ minimizing

$$\sum_i d(\mathtt{H}\mathbf{x}_i, \mathtt{H}'\mathbf{x}'_i)^2$$

(v) Resample J according to $\mathtt{H}$ and J' according to $\mathtt{H}'$.

Rectification

Example rectification

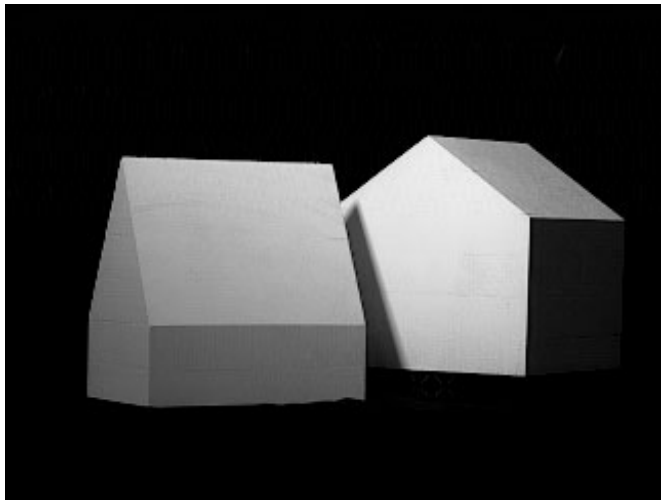


Image J

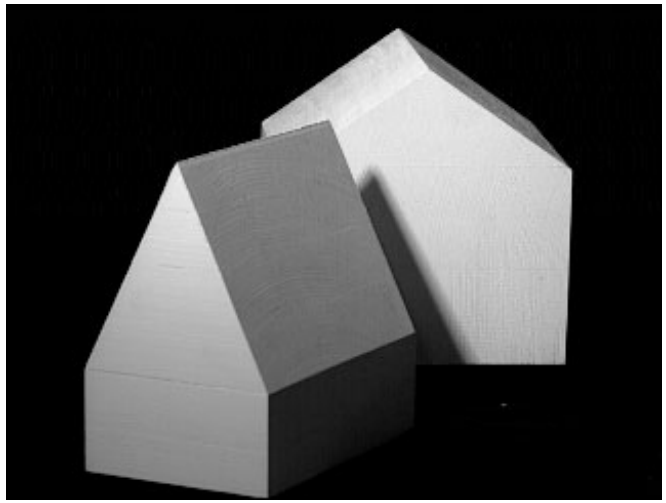
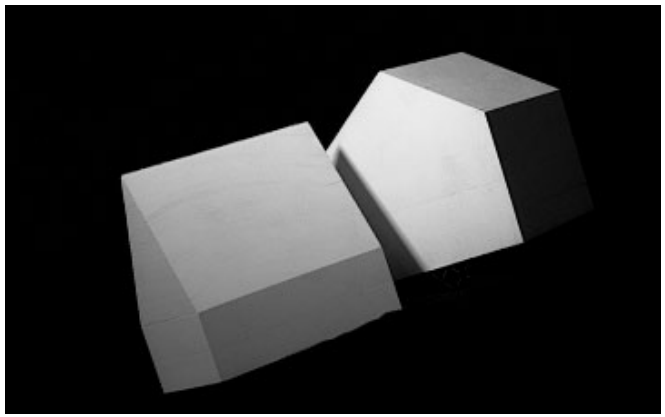
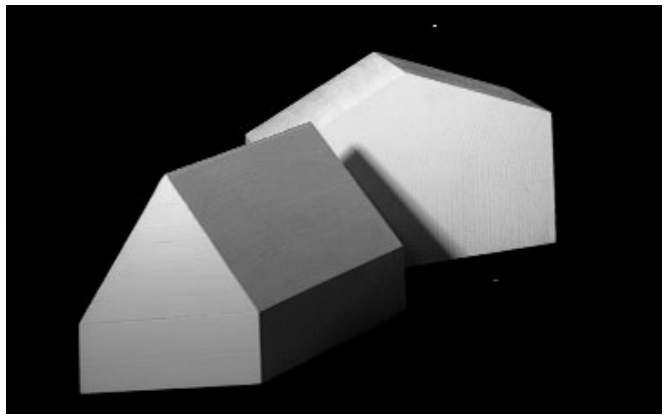


Image J'



J resampled by $\mathtt{H}$



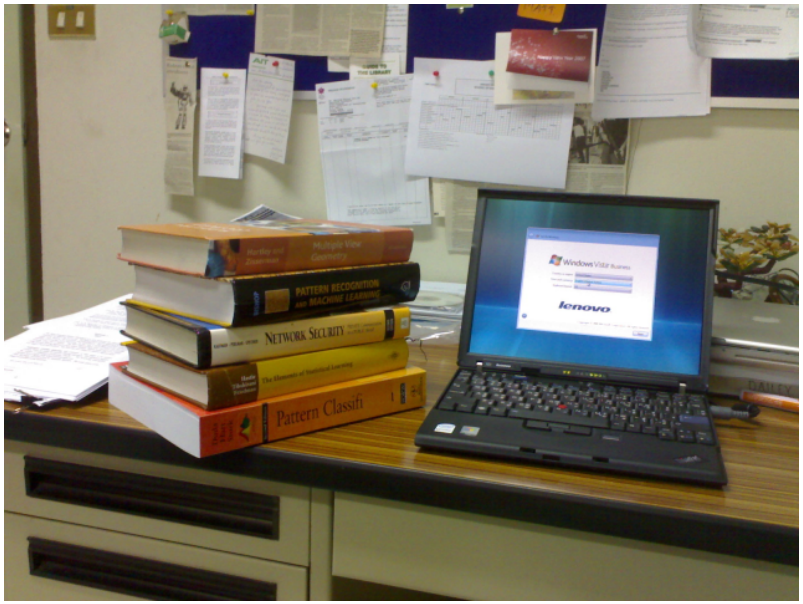
J' resampled by $\mathtt{H}'$

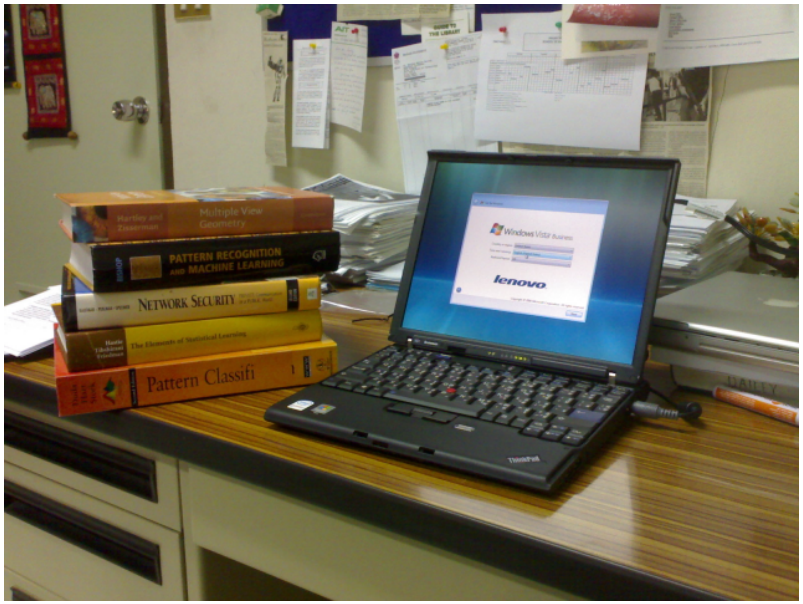
Hartley and Zisserman (2004), Fig. 11.11

Rectification

Example rectification

Here is another example from, my desk:

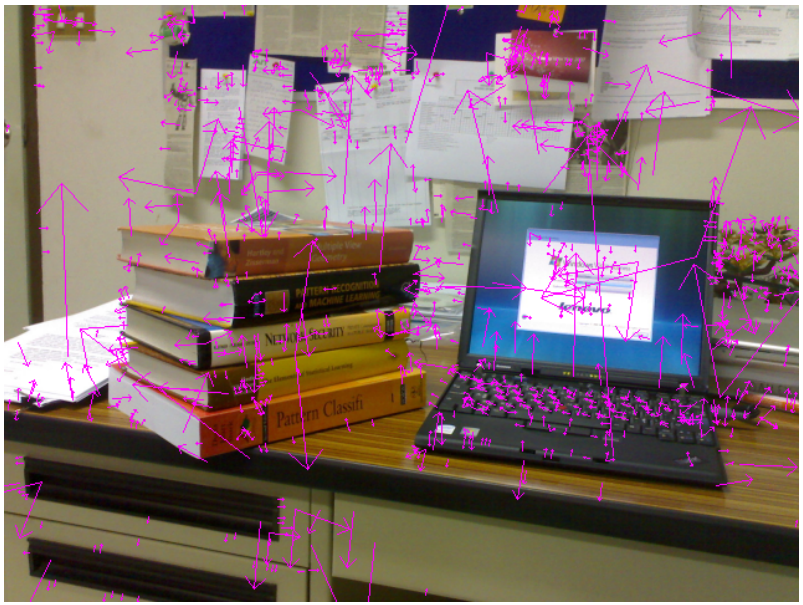


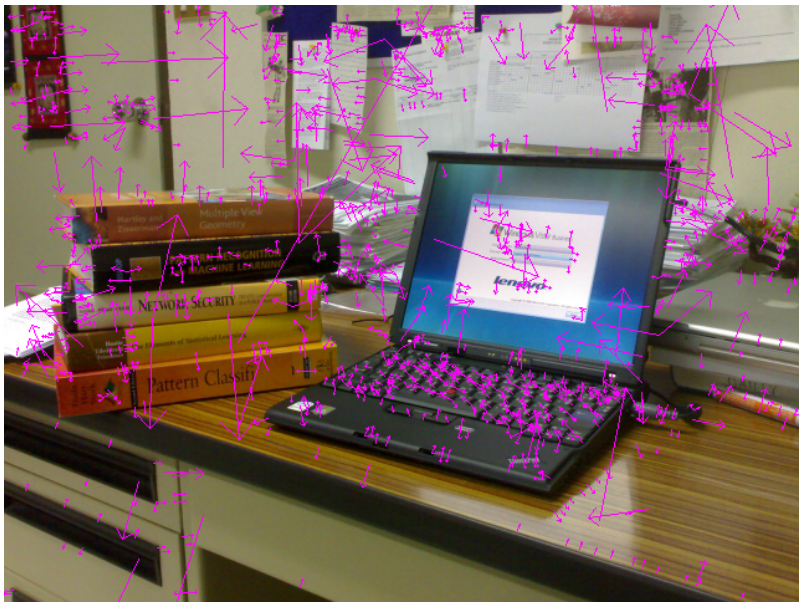


Rectification

Example rectification

First I ran Rob Hess' SIFT code to get 1106 and 999 features in image 1 and 2, respectively:





Rectification

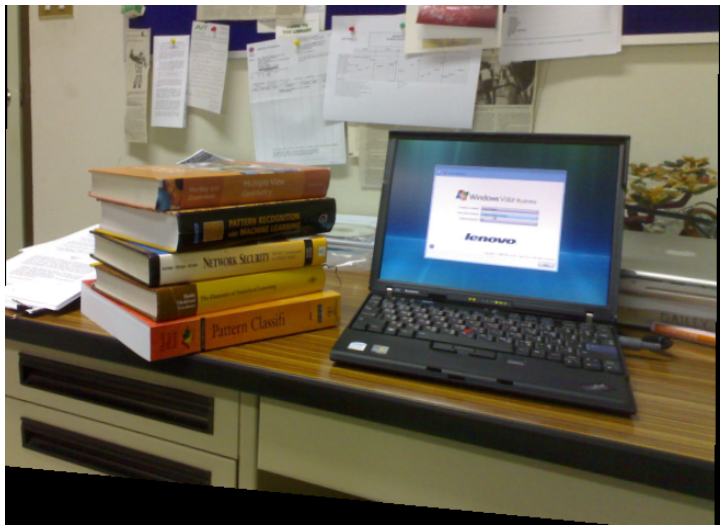
Example rectification

Then I used Rob Hess' implementation of Beis and Lowe's Best Bin First approximate nearest neighbor matching and OpenCV's `cvFindFundamentalMat()` with option `CV_FM_RANSAC` to estimate `F` and obtain correspondences:

Rectification

Example rectification

Then I used Hartley's minimum-distortion homography technique to obtain rectifying homographies and applied them to the images:



These images are now ready for dense stereo matching.

Outline

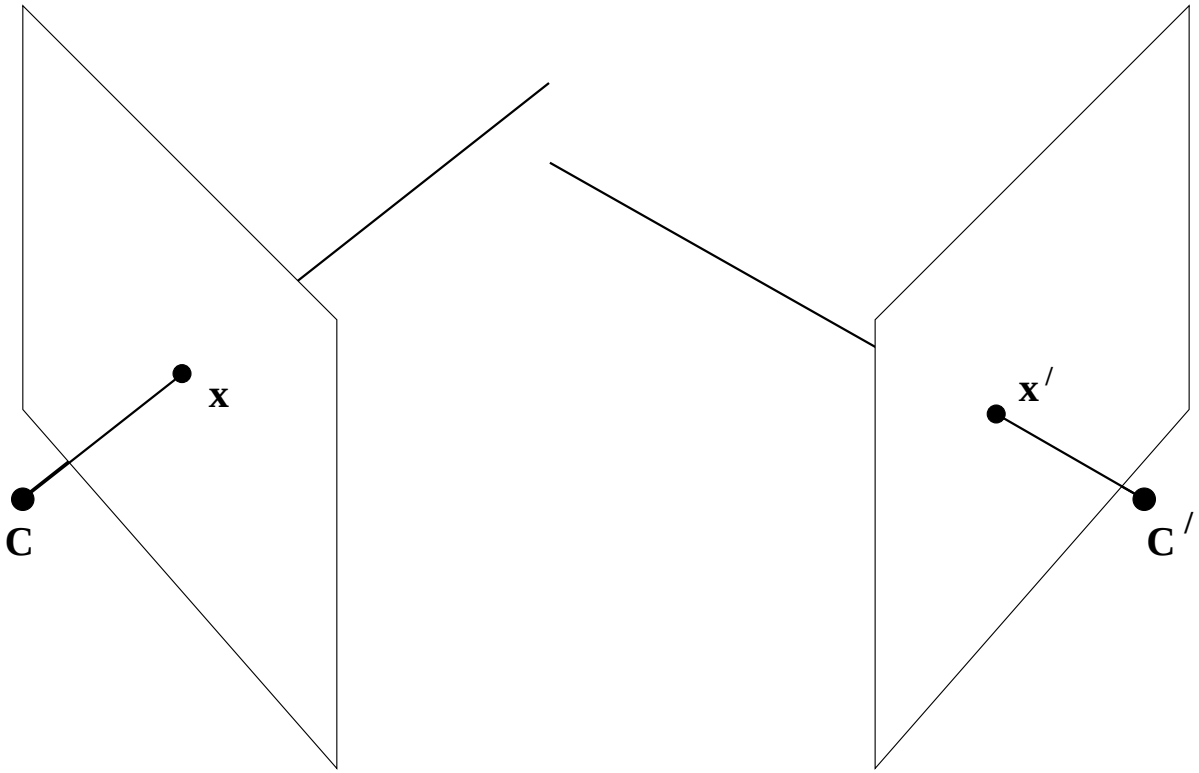
How can we compute the position of a point in 3 space given two views?

We'll assume we have a perfect estimate of $\mathtt{F}$ (and therefore $\mathtt{P}$ and $\mathtt{P}'$, up to some ambiguity).

We will

- Derive a simple linear solution
- Consider some limitations of the linear solution
- Define a cost function for an optimal reconstruction
- Discuss algorithms for minimizing that cost function

Backprojection for two corresponding points $\mathbf{x} \leftrightarrow \mathbf{x}'$ doesn't work because with image measurement error, the backprojected rays will be skew:



Hartley and Zisserman (2004), Fig. 12.1a

Skew is due to the fact that in general, corresponding points $\mathbf{x}$ and $\mathbf{x}'$ will not exactly satisfy the epipolar constraint $\mathbf{x}'^T \mathbf{F} \mathbf{x} = 0$:

Hartley and Zisserman (2004), Fig. 12.1b

Structure computation

Invariance of the solution

What properties would we like our reconstruction to have?

Let's call our triangulation method τ:

$$\mathbf{X} = \tau(\mathbf{x}, \mathbf{x}', \mathtt{P}, \mathtt{P}').$$

We would like τ to be invariant under projective transformations $\mathtt{H}$:

$$\tau(\mathbf{x}, \mathbf{x}', \mathtt{P}, \mathtt{P}') = \mathtt{H}^{-1}\tau(\mathbf{x}, \mathbf{x}', \mathtt{P}\mathtt{H}^{-1}, \mathtt{P}'\mathtt{H}^{-1})$$

If we adopt this goal, minimizing error in $\mathbb{P}^3$ will not work because distance and perpendicularity relationships are not invariant in $\mathbb{P}^3$.

Instead of minimizing error in $\mathbb{P}^3$, we would like instead to estimate a 3D point $\hat{\mathbf{X}}$ exactly satisfying

$$\hat{\mathbf{x}} = \mathrm{P}\hat{\mathbf{X}} \quad \hat{\mathbf{x}}' = \mathrm{P}'\hat{\mathbf{X}}$$

and maximizing the likelihood of the measurements under Gaussian error.

As usual, the maximum likelihood estimate under Gaussian errors minimizes reprojection error.

Since reprojection error only measures distance in the image, the ML estimate will be invariant under projective transformations of 3-space.

First we'll consider a simple linear estimate minimizing algebraic error similar to the DLT that is not optimal.

We use the cross product to eliminate the homogeneous scale factor. For each image we have $\mathbf{x} \times (\mathtt{P}\mathbf{X}) = \mathbf{0}$ giving

$$\mathbf{x} \times (\mathtt{P}\mathbf{X}) = \begin{pmatrix} x(\mathbf{p}^{3T}\mathbf{X}) - (\mathbf{p}^{1T}\mathbf{X}) \\ y(\mathbf{p}^{3T}\mathbf{X}) - (\mathbf{p}^{2T}\mathbf{X}) \\ x(\mathbf{p}^{2T}\mathbf{X}) - y(\mathbf{p}^{1T}\mathbf{X}) \end{pmatrix} = \mathbf{0}.$$

Taking two linearly independent equations from each camera we obtain the system $\mathtt{A}\mathbf{X} = \mathbf{0}$ with

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

We have 4 equations in 4 homogeneous unknowns.

Structure computation

DLT-style linear solution

As always, the solution to the homogeneous linear system is the last right singular vector of $\mathtt{A}$.

If $\mathbf{X}$ is known not to be close to the principal plane, we can force the last element of $\mathbf{X}$ to 1, obtaining an inhomogeneous linear system with 4 equations in 3 unknowns that can be solved using the pseudoinverse of $\mathtt{A}$.

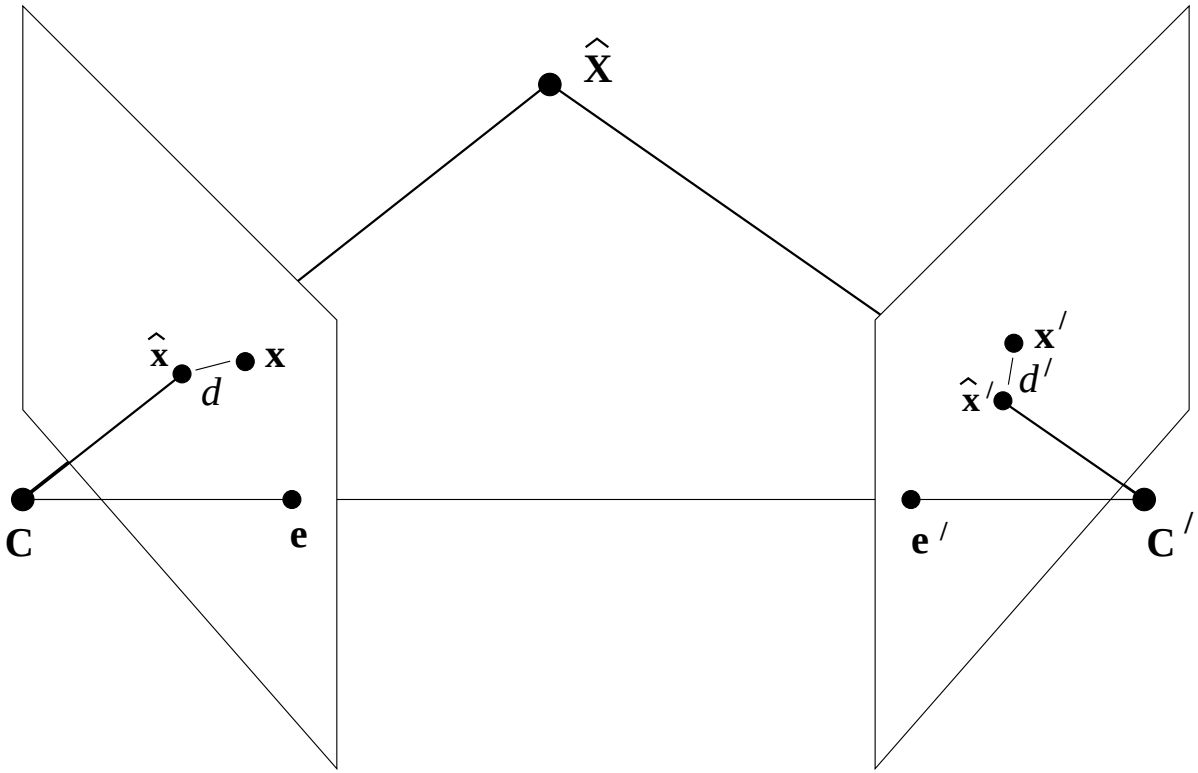
In terms of invariance, neither method is invariant under arbitrary projective transformations.

The inhomogeneous method but not the homogeneous method is invariant under affine transformations.

Nevertheless, in many cases the homogeneous method works just fine, and importantly, it generalizes to the situation where there are more than two views.

Under a Gaussian error assumption, the maximum likelihood estimate $\hat{\mathbf{X}}$ of $\mathbf{X}$ minimizes the reprojection error

$$\mathcal{C}(\mathbf{x}, \mathbf{x}') = d(\mathbf{x}, \hat{\mathbf{x}})^2 + d(\mathbf{x}', \hat{\mathbf{x}}')^2 \quad \text{subject to } \hat{\mathbf{x}}'^T \mathtt{F} \hat{\mathbf{x}} = 0.$$

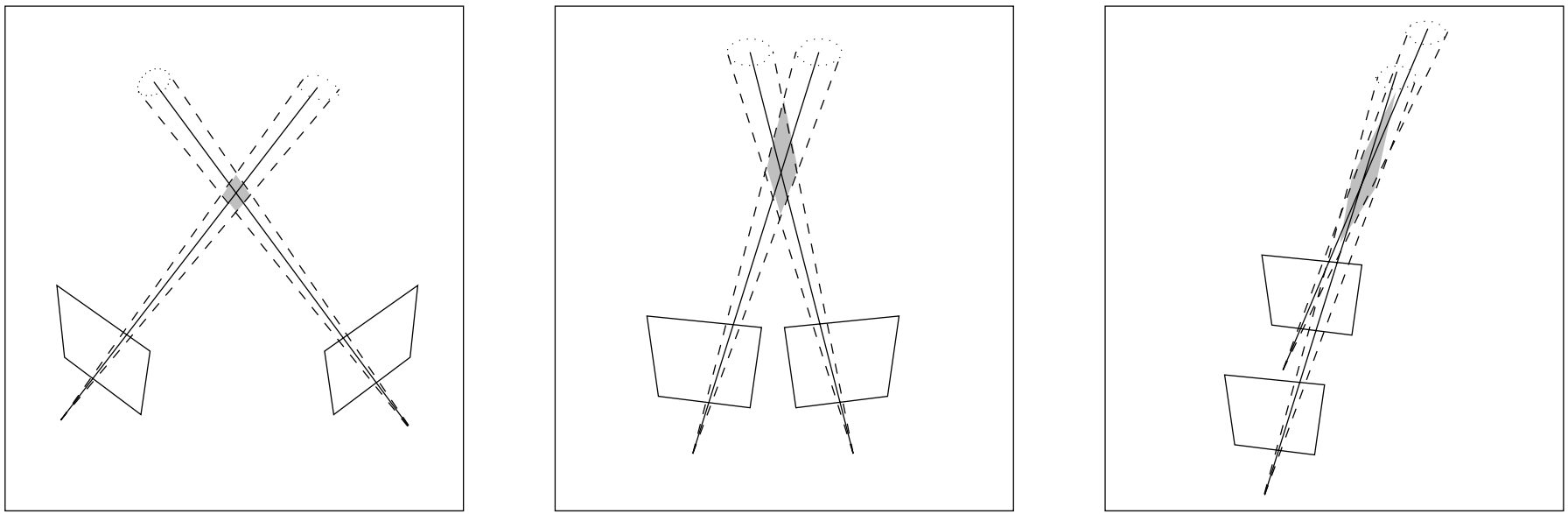


Hartley and Zisserman (2004), Fig. 11.2

To find $\hat{\mathbf{X}}$, we could use Levenberg-Marquardt beginning from the DLT estimate of $\mathbf{X}$.

We could also minimize Sampson error, which is a first-order approximation to reprojection error, or obtain an optimal estimate using a fairly complex but analytical algorithm. See the text.

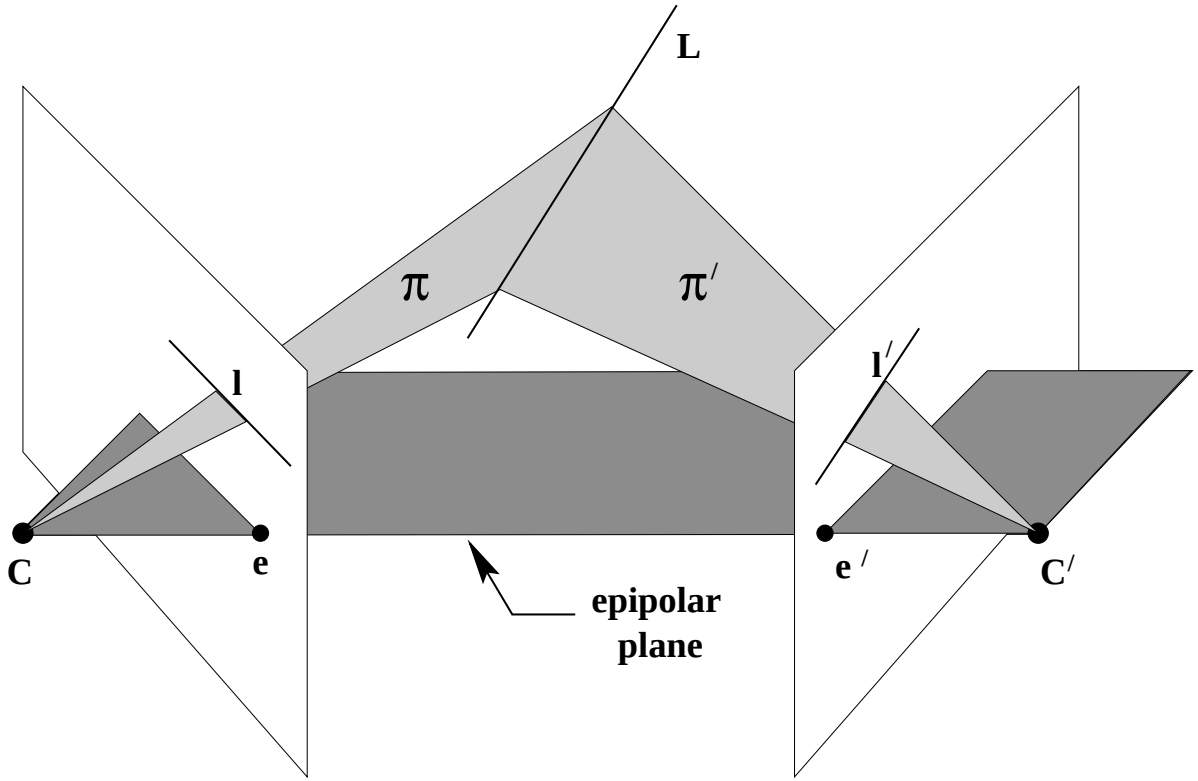
Intuitively, as noise in the image measurements increases, so does the expected error in the reconstruction:



Hartley and Zisserman, Fig. 12.6

An estimate of the expected error covariance for a reconstructed point can be obtained using the methods of Hartley and Zisserman, Chapter 5.

Lines can also be reconstructed from two views except for the degeneracy involved when they intersect or nearly intersect the epipoles, i.e., lie on an epipolar plane.



Hartley and Zisserman, Fig. 12.7

Outline

Conclusion

Three views

We looked at two-view reconstruction in some detail.

With three views, there are some benefits:

- Given point correspondences in two images, the image of the point in the third is determined exactly
- The same holds for lines. Additionally, three views of a line give us an overdetermined solution allowing us to minimize over measurement errors.
- There is a 3-view generalization of the fundamental matrix called the trifocoal tensor.

Next we'll cover 3-view geometry then N-view reconstruction.