Overview

- **Lecture 1: Stereo Reconstruction I**: epipolar geometry, fundamental matrix.
- **Lecture 2: Stereo Reconstruction II**: correspondence algorithms, triangulation.
- **Lecture 3: Structure and Motion**: ambiguities, computing the fundamental matrix, recovering ego-motion, applications.
- **Lecture 4: Object detection**: the adaBoost algorithm for face detection.

Further reading (www addresses) and the lecture notes are on [http://www.robots.ox.ac.uk/~az/lectures](http://www.robots.ox.ac.uk/~az/lectures)
Stereo Reconstruction

Shape (3D) from two (or more) images

Example

images

shape

surface reflectance
Scenarios

The two images can arise from

- A stereo rig consisting of two cameras
  - the two images are acquired simultaneously

or

- A single moving camera (static scene)
  - the two images are acquired sequentially

The two scenarios are geometrically equivalent
The objective

**Given** two images of a scene acquired by known cameras compute the 3D position of the scene (structure recovery)

**Basic principle:** triangulate from corresponding image points

- Determine 3D point at intersection of two back-projected rays

**Corresponding points** are images of the same scene point

**Triangulation**

The back-projected points generate rays which intersect at the 3D scene point
An algorithm for stereo reconstruction

1. For each point in the first image determine the corresponding point in the second image
   (this is a search problem)

2. For each pair of matched points determine the 3D point by triangulation
   (this is an estimation problem)

The correspondence problem

Given a point \( x \) in one image find the corresponding point in the other image

This appears to be a 2D search problem, but it is reduced to a 1D search by the epipolar constraint
Outline

1. Epipolar geometry
   - the geometry of two cameras
   - reduces the correspondence problem to a line search

2. Stereo correspondence algorithms

3. Triangulation

Notation

The two cameras are $P$ and $P'$, and a 3D point $X$ is imaged as

$$x = PX \quad x' = P'X$$

$P$ : $3 \times 4$ matrix
$x$ : 4-vector
$x'$ : 3-vector

Warning

for equations involving homogeneous quantities ‘$=$’ means ‘equal up to scale’
Epipolar geometry

Given an image point in one view, where is the corresponding point in the other view?

- A point in one view “generates” an epipolar line in the other view
- The corresponding point lies on this line
Epipolar line

Epipolar constraint
- Reduces correspondence problem to 1D search along an epipolar line

Epipolar geometry continued

Epipolar geometry is a consequence of the coplanarity of the camera centres and scene point

The camera centres, corresponding points and scene point lie in a single plane, known as the epipolar plane
Nomenclature

- The epipolar line $l/T$ is the image of the ray through $x$.
- The epipole $e$ is the point of intersection of the line joining the camera centres with the image plane:
  - this line is the baseline for a stereo rig, and
  - the translation vector for a moving camera.
- The epipole is the image of the centre of the other camera: $e = PC/T$, $e/T = P'/C$.

The epipolar pencil

As the position of the 3D point $X$ varies, the epipolar planes “rotate” about the baseline. This family of planes is known as an epipolar pencil. All epipolar lines intersect at the epipole.

(a pencil is a one parameter family)
Epipolar geometry depends only on the relative pose (position and orientation) and internal parameters of the two cameras, i.e. the position of the camera centres and image planes. It does not depend on the scene structure (3D points external to the camera).

Note, epipolar lines are in general not parallel.
Homogeneous notation for lines

Recall that a point \((x, y)\) in 2D is represented by the homogeneous 3-vector \(\mathbf{x} = (x_1, x_2, x_3)^\top\), where \(x = x_1/x_3, y = x_2/x_3\).

A line in 2D is represented by the homogeneous 3-vector

\[
\mathbf{l} = \begin{pmatrix} l_1 \\ l_2 \\ l_3 \end{pmatrix}
\]

which is the line \(l_1x + l_2y + l_3 = 0\).

**Example** represent the line \(y = 1\) as a homogeneous vector.

Write the line as \(-y + 1 = 0\) then \(l_1 = 0, l_2 = -1, l_3 = 1\), and \(\mathbf{l} = (0, -1, 1)^\top\).

Note that \(\mu(l_1x + l_2y + l_3) = 0\) represents the same line (only the ratio of the homogeneous line coordinates is significant).

Writing both the point and line in homogeneous coordinates gives

\[l_1x_1 + l_2x_2 + l_3x_3 = 0\]

- point on line \(l.x = 0\) or \(l^\top x = 0\) or \(x^\top l = 0\)

- The line \(\mathbf{l}\) through the two points \(\mathbf{p}\) and \(\mathbf{q}\) is \(\mathbf{l} = \mathbf{p} \times \mathbf{q}\)

**Proof**

\[
\mathbf{l.p} = (\mathbf{p} \times \mathbf{q}).\mathbf{p} = 0 \quad \mathbf{l.q} = (\mathbf{p} \times \mathbf{q}).\mathbf{q} = 0
\]

- The intersection of two lines \(\mathbf{l}\) and \(\mathbf{m}\) is the point \(\mathbf{x} = \mathbf{l} \times \mathbf{m}\)

**Example**: compute the point of intersection of the two lines \(\mathbf{l}\) and \(\mathbf{m}\) in the figure below

\[
\mathbf{l} = \begin{pmatrix} 0 \\ -1 \\ 1 \end{pmatrix} \quad \mathbf{m} = \begin{pmatrix} -1 \\ 0 \\ 2 \end{pmatrix}
\]

\[
\mathbf{x} = \mathbf{l} \times \mathbf{m} = \begin{vmatrix} i & j & k \\ 0 & -1 & 1 \\ -1 & 0 & 2 \end{vmatrix} = \begin{pmatrix} -2 \\ -1 \\ -1 \end{pmatrix}
\]

which is the point \((2,1)\)
Matrix representation of the vector cross product

The vector product $\mathbf{v} \times \mathbf{x}$ can be represented as a matrix multiplication

$$\mathbf{v} \times \mathbf{x} = \begin{pmatrix} v_2x_3 - v_3x_2 \\ v_3x_1 - v_1x_3 \\ v_1x_2 - v_2x_1 \end{pmatrix} = [\mathbf{v}]_\times \mathbf{x}$$

where

$$[\mathbf{v}]_\times = \begin{bmatrix} 0 & -v_3 & v_2 \\ v_3 & 0 & -v_1 \\ -v_2 & v_1 & 0 \end{bmatrix}$$

• $[\mathbf{v}]_\times$ is a $3 \times 3$ skew-symmetric matrix of rank 2.
• $\mathbf{v}$ is the null-vector of $[\mathbf{v}]_\times$, i.e. $[\mathbf{v}]_\times \mathbf{v} = \mathbf{0}$, since $\mathbf{v} \times \mathbf{v} = [\mathbf{v}]_\times \mathbf{v} = \mathbf{0}$

Example: compute the cross product of $\mathbf{l}$ and $\mathbf{m}$

$$\mathbf{l} = \begin{pmatrix} 0 \\ -1 \\ 1 \end{pmatrix} \quad \mathbf{m} = \begin{pmatrix} -1 \\ 0 \\ 2 \end{pmatrix} \quad [\mathbf{v}]_\times = \begin{bmatrix} 0 & -v_3 & v_2 \\ v_3 & 0 & -v_1 \\ -v_2 & v_1 & 0 \end{bmatrix}$$

$$\mathbf{x} = \mathbf{l} \times \mathbf{m} = [\mathbf{l}]_\times \mathbf{m} = \begin{bmatrix} 0 & -1 & -1 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix} \begin{pmatrix} -1 \\ 0 \\ 2 \end{pmatrix} = \begin{pmatrix} -2 \\ -1 \\ -1 \end{pmatrix}$$

Note

$$\begin{bmatrix} 0 & -1 & -1 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix} \begin{pmatrix} 0 \\ -1 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \quad ([\mathbf{l}]_\times \mathbf{1} = \mathbf{0})$$
Algebraic representation of epipolar geometry

We know that the epipolar geometry defines a mapping

$$x \rightarrow l'$$

point in first image \hspace{1cm} epipolar line in second image

- the map only depends on the cameras $P, P'$ (not on structure)
- it will be shown that the map is linear and can be written as $l' = Fx$, where $F$ is a $3 \times 3$ matrix called the fundamental matrix

Derivation of the algebraic expression $l' = Fx$

Outline

Step 1: for a point $x$ in the first image
back project a ray with camera $P$

Step 2: choose two points on the ray and
project into the second image with camera $P'$

Step 3: compute the line through the two
image points using the relation $l' = p \times q$
• choose camera matrices

\[
P = K [ R | t ]
\]

- **internal calibration**
- **rotation** from world to camera coordinate frame
- **translation**

• first camera

\[
P = K [ I | 0 ]
\]

world coordinate frame aligned with first camera

• second camera

\[
P' = K' [ R | t ]
\]

---

**Step 1:** for a point \(x\) in the first image

back project a ray with camera \(P = K [ I | 0 ]\)

A point \(x\) back projects to a ray \(X(Z)\) that satisfies

\[
PX(Z) = K[I | 0]X(Z) = x
\]

where \(Z\) is the point’s depth, since

\[
x = \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = K[I | 0] \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = K \begin{pmatrix} x \\ y \\ z \end{pmatrix}
\]

\[
X(Z) = \begin{pmatrix} zK^{-1}x \\ 1 \end{pmatrix}
\]
**Step 2**: choose two points on the ray and project into the second image with camera $P'$

Consider two points on the ray $X(Z) = \begin{pmatrix} zK^{-1}x \\ 1 \end{pmatrix}$

- $Z = 0$ is the camera centre $\begin{pmatrix} 0 \\ 1 \end{pmatrix}$
- $Z = \infty$ is the point at infinity $\begin{pmatrix} K^{-1}x \\ 0 \end{pmatrix}$

Project these two points into the second view

$$P' \begin{pmatrix} 0 \\ 1 \end{pmatrix} = K'[R \mid t] \begin{pmatrix} 0 \\ 1 \end{pmatrix} = K't$$

$$P' \begin{pmatrix} K^{-1}x \\ 0 \end{pmatrix} = K'[R \mid t] \begin{pmatrix} K^{-1}x \\ 0 \end{pmatrix} = K'RK^{-1}x$$

**Step 3**: compute the line through the two image points using the relation $l' = p \times q$

Compute the line through the points $l' = (K't) \times (K'RK^{-1}x)$

Using the identity $(Ma) \times (Mb) = M^{-T}(a \times b)$ where $M^{-T} = (M^{-1})^T = (M^T)^{-1}$

$$l' = K'^{-T}(t \times (RK^{-1}x)) = K'^{-T}[t] \times RK^{-1}x$$

$F$ is the fundamental matrix

$$l' = Fx \quad F = K'^{-T}[t] \times RK^{-1}$$

Points $x$ and $x'$ correspond ($x \leftrightarrow x'$) then $x'^Tl' = 0$

$$x'^T Fx = 0$$
Example I: compute the fundamental matrix for a parallel camera stereo rig

\[ P = K[I \mid 0] \quad P' = K'[R \mid t] \]

\[ K = K' = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad R = I \quad t = \begin{pmatrix} t_x \\ 0 \\ 0 \end{pmatrix} \]

\[ F = K'^{-T} [t] \times R K^{-1} \]

\[ = \begin{bmatrix} 1/f & 0 & 0 \\ 0 & 1/f & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -t_x \\ 0 & t_x & 0 \end{bmatrix} \begin{bmatrix} 1/f & 0 & 0 \\ 0 & 1/f & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \]

\[ x'^{T} F x = \begin{pmatrix} x' & y' & 1 \end{pmatrix} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = 0 \]

- reduces to \( y = y' \), i.e. raster correspondence (horizontal scan-lines)

\[ F \] is a rank 2 matrix

The epipole \( e \) is the null-space vector (kernel) of \( F \) (exercise), i.e. \( Fe = 0 \)

In this case

\[ \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix} \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} = 0 \]

so that

\[ e = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \]

Geometric interpretation?
**Example II:** compute $F$ for a forward translating camera

$$P = K[I | 0] \quad P' = K'[R | t]$$

$$K = K' = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad R = I \quad t = \begin{bmatrix} 0 \\ 0 \\ t_z \end{bmatrix}$$

$$F = K'^{-T} [t] \times RK^{-1}$$

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

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

From $l' = Fx$ the epipolar line for the point $x = (x, y, 1)^T$ is

$$l' = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} -y \\ x \\ 0 \end{bmatrix}$$

The points $(x, y, 1)^T$ and $(0, 0, 1)^T$ lie on this line
Summary: Properties of the Fundamental matrix

- **F** is a rank 2 homogeneous matrix with 7 degrees of freedom.

- **Point correspondence:**
  if \( \mathbf{x} \) and \( \mathbf{x}' \) are corresponding image points, then \( \mathbf{x}'^\top \mathbf{F} \mathbf{x} = 0 \).

- **Epipolar lines:**
  - \( \mathbf{l}' = \mathbf{F} \mathbf{x} \) is the epipolar line corresponding to \( \mathbf{x} \).
  - \( \mathbf{l} = \mathbf{F}^\top \mathbf{x}' \) is the epipolar line corresponding to \( \mathbf{x}' \).

- **Epipoles:**
  - \( \mathbf{Fe} = 0 \).
  - \( \mathbf{F}^\top \mathbf{e}' = 0 \).

- **Computation from camera matrices \( \mathbf{P}, \mathbf{P}' \):**
  \( \mathbf{P} = \mathbf{K} [I \mid 0], \quad \mathbf{P}' = \mathbf{K}' [R \mid t], \quad \mathbf{F} = \mathbf{K}'^{-\top} [t] \times \mathbf{K}^{-1} \).
Stereo correspondence algorithms

Problem statement

**Given**: two images and their associated cameras compute corresponding image points.

Algorithms may be classified into two types:

1. Dense: compute a correspondence at every pixel
2. Sparse: compute correspondences only for features

The methods may be top down or bottom up
Top down matching

1. Group model (house, windows, etc) independently in each image
2. Match points (vertices) between images

Bottom up matching

- epipolar geometry reduces the correspondence search from 2D to a 1D search on corresponding epipolar lines

- 1D correspondence problem
cross-eye viewing random dot stereogram

Correspondence algorithms

Algorithms may be top down or bottom up – random dot stereograms are an existence proof that bottom up algorithms are possible

From here on only consider bottom up algorithms

*Algorithms may be classified into two types:*
1. Dense: compute a correspondence at every pixel
2. Sparse: compute correspondences only for features
Dense correspondence algorithm

Parallel camera example – epipolar lines are corresponding rasters

Search problem (geometric constraint): for each point in the left image, the corresponding point in the right image lies on the epipolar line (1D ambiguity)

Disambiguating assumption (photometric constraint): the intensity neighbourhood of corresponding points are similar across images

Measure similarity of neighbourhood intensity by cross-correlation

Intensity profiles

• Clear correspondence between intensities, but also noise and ambiguity
Normalized Cross Correlation

subtract mean: \( A \leftarrow A - \langle A \rangle, B \leftarrow B - \langle B \rangle \)

\[
\text{NCC} = \frac{\sum_i \sum_j A(i, j)B(i, j)}{\sqrt{\sum_i \sum_j A(i, j)^2} \sqrt{\sum_i \sum_j B(i, j)^2}}
\]

Write regions as vectors

\( A \rightarrow \mathbf{a}, \ B \rightarrow \mathbf{b} \)

\[
\text{NCC} = \frac{\mathbf{a} \cdot \mathbf{b}}{||\mathbf{a}|| ||\mathbf{b}||}
\]

\(-1 < \text{NCC} < 1\)

Cross-correlation of neighbourhood regions

Invariant to \( I \rightarrow \alpha I + \beta \)  
(exercise)
cross correlation

disparity = x' - x
Why is cross-correlation such a poor measure in the second case?

1. The neighbourhood region does not have a “distinctive” spatial intensity distribution
2. Foreshortening effects

Sketch of a dense correspondence algorithm

For each pixel in the left image

- compute the neighbourhood cross correlation along the corresponding epipolar line in the right image
- the corresponding pixel is the one with the highest cross correlation

Parameters

- size (scale) of neighbourhood
- search disparity

Other constraints

- uniqueness
- ordering
- smoothness of disparity field

Applicability

- textured scene, largely fronto-parallel
**Example** dense correspondence algorithm

![left image](image1)

![right image](image2)

**3D reconstruction**

![right image](image3)

![depth map](image4)

intensity = depth
Views of a texture mapped 3D triangulation

Pentagon example
**Example:** depth and disparity for a parallel camera stereo rig

\[
K = K' = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad R = I \quad t = \begin{pmatrix} t_x \\ 0 \\ 0 \end{pmatrix}
\]

Then, \( y' = y \), and the disparity \( d = x' - x = \frac{f t_x}{Z} \).

**Derivation**

\[
\frac{x}{f} = \frac{X}{Z} \quad \frac{x'}{f} = \frac{X + t_x}{Z} \\
\frac{x'}{f} = \frac{x}{f} + \frac{t_x}{Z}
\]

**Note**

- Image movement (disparity) is inversely proportional to depth \( Z \)
  
  \( z \rightarrow \infty \), \( d \rightarrow 0 \)

- Depth is inversely proportional to disparity

**Error analysis**

\[
d = x' - x = \frac{f t_x}{Z} \quad \frac{z}{d} = \frac{f t_x}{d} \quad \frac{\delta z}{\delta d} = -\frac{f t_x}{d^2} = -\frac{Z^2}{f t_x}
\]

Measurement errors \( \delta x, \delta x' \rightarrow \delta d \)

\[
\delta z = -\frac{Z^2}{f t_x} \delta d \quad \text{depth error proportional to depth squared}
\]

Point position error ellipse

How can position uncertainty be reduced?
Rectification

For converging cameras
- epipolar lines are not parallel

Project images onto plane parallel to baseline

Rectification continued

Convert converging cameras to parallel camera geometry by an image mapping

Image mapping is a 2D homography (projective transformation)

\[ H = KRK^{-1} \] (exercise)

Example

original stereo pair

rectified stereo pair
Triangulation

Problem statement

**Given:** corresponding measured (i.e. noisy) points $x$ and $x'$, and cameras (exact) $P$ and $P'$, compute the 3D point $X$.

**Problem:** in the presence of noise, back projected rays do not intersect.

Measured points do not lie on corresponding epipolar lines.

Rays are skew in space.
1. Vector solution

Compute the mid-point of the shortest line between the two rays
2. Linear triangulation (algebraic solution)

Use the equations \( x = pX \) and \( x' = p'X \) to solve for \( X \)

For the first camera:

\[
P = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \end{bmatrix} = \begin{bmatrix} p^1_T \\ p^2_T \\ p^3_T \end{bmatrix}
\]

where \( p^i_T \) are the rows of \( P \)

- eliminate unknown scale in \( \lambda x = PX \) by forming a cross product \( x \times (PX) = 0 \)

\[
\begin{align*}
x(p^3_T X) - (p^1_T X) &= 0 \\
y(p^3_T X) - (p^2_T X) &= 0 \\
x(p^2_T X) - y(p^1_T X) &= 0
\end{align*}
\]

- rearrange as (first two equations only)

\[
\begin{bmatrix}
x(p^3_T) - p^1_T \\
y(p^3_T) - p^2_T
\end{bmatrix} X = 0
\]

Similarly for the second camera:

\[
\begin{bmatrix}
x'p'^3_T - p'^1_T \\
y'p'^3_T - p'^2_T
\end{bmatrix} X = 0
\]

Collecting together gives

\( AX = 0 \)

where \( A \) is the \( 4 \times 4 \) matrix

\[
A = \begin{bmatrix}
xp^3_T - p^1_T \\
yp^3_T - p^2_T \\
x'p'^3_T - p'^1_T \\
y'p'^3_T - p'^2_T
\end{bmatrix}
\]

from which \( X \) can be solved up to scale.

**Problem:** does not minimize anything meaningful

**Advantage:** extends to more than two views
3. Minimizing a geometric/statistical error

The idea is to estimate a 3D point \( \hat{X} \) which exactly satisfies the supplied camera geometry, so it projects as

\[
\hat{x} = P\hat{X} \quad \hat{x}' = P'\hat{X}
\]

and the aim is to estimate \( \hat{X} \) from the image measurements \( x \) and \( x' \).

\[
\min_{\hat{X}} \ C(x, x') = d(x, \hat{x})^2 + d(x', \hat{x}')^2
\]

where \( d(\ast, \ast) \) is the Euclidean distance between the points.

- It can be shown that if the measurement noise is Gaussian mean zero, \( \sim N(0, \sigma^2) \), then minimizing geometric error is the Maximum Likelihood Estimate of \( X \)

- The minimization appears to be over three parameters (the position \( X \)), but the problem can be reduced to a minimization over one parameter
Different formulation of the problem

The minimization problem may be formulated differently:

- Minimize
  \[ d(x, l)^2 + d(x', l')^2 \]

- \( l \) and \( l' \) range over all choices of corresponding epipolar lines.
- \( x \) is the closest point on the line \( l \) to \( x \).
- Same for \( x' \).

Minimization method

- Parametrize the pencil of epipolar lines in the first image by \( t \), such that the epipolar line is \( l(t) \)
- Using \( F \) compute the corresponding epipolar line in the second image \( l'(t) \)
- Express the distance function \( d(x, l)^2 + d(x', l')^2 \) explicitly as a function of \( t \)
- Find the value of \( t \) that minimizes the distance function
- Solution is a 6\(^{th}\) degree polynomial in \( t \)