Overview

The main points covered in this lecture are:

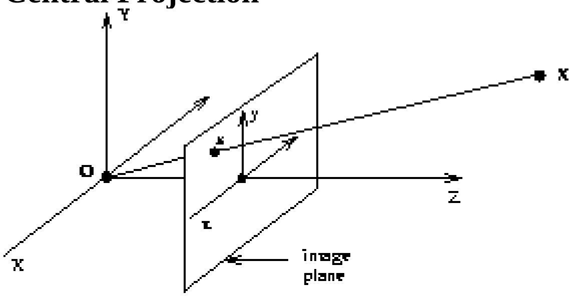
- A perspective (central) projection camera is represented by a 3×4 matrix.
- The most general perspective transformation transformation between two planes (a world plane and the image plane, or two image planes induced by a world plane) is a plane projective transformation. This can be computed from the correspondence of four (or more) points.
- The epipolar geometry between two views is represented by the fundamental matrix. This can be computed from the correspondence of seven (or more) points.

Both *geometric* and *algebraic* descriptions are given.

Bob Fisher
Wed Apr 16 00:58:54 BST 1997

Imaging Geometry

Central Projection



$$\lambda \begin{pmatrix} x \\ y \\ f \end{pmatrix} = \begin{pmatrix} X \\ Y \\ Z \end{pmatrix}$$

$$\frac{x}{f} = \frac{X}{Z} \qquad \frac{y}{f} = \frac{Y}{Z}$$

Vector notation:

$$\mathbf{x} = f\frac{\mathbf{X}}{Z}$$

where $\mathbf{x}$ and $\mathbf{X}$ are 3-vectors, with $\mathbf{x} = (x, y, f)^\top$.

Here central projection is represented in the coordinate frame attached to the camera. Generally, there is not direct access to this camera coordinate frame. Instead, we need to determine the mapping from a world coordinate frame to an image coordinate system (see next slide).

Bob Fisher
Wed Apr 16 00:58:54 BST 1997

Next Up Previous

Next: Summary and Properties **Up:** No Title **Previous:** Imaging Geometry

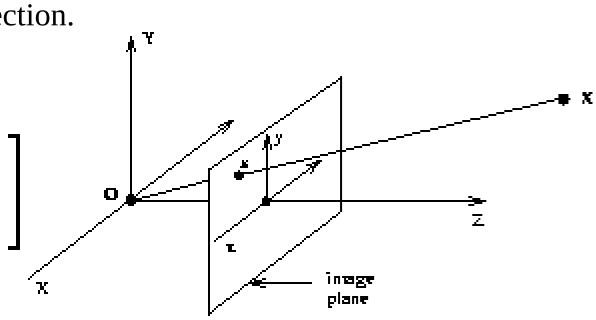
3x4 Projection Matrix

There are three coordinate systems involved --- camera, image and world.

1. **Camera**: perspective projection.

$$\begin{bmatrix} x_c \\ y_c \\ f \end{bmatrix} = \lambda \begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix}$$

where $\lambda = f/Z_c$.



This can be written as a linear mapping between homogeneous coordinates (the equation is only up to a scale factor):

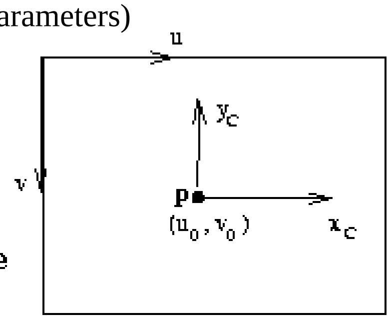
$$\begin{bmatrix} x_c \\ y_c \\ f \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix}$$

where a 3×4 **projection matrix** represents a map from 3D to 2D.

2. **Image**: (intrinsic/internal camera parameters)

$$\begin{aligned} k_u x_c &= u - u_0 \\ k_v y_c &= v_0 - v \end{aligned}$$

where the units of k are [pixels/length].



$$\mathbf{x}_i = \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} fk_u & 0 & u_0 \\ 0 & -fk_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_c \\ y_c \\ f \end{bmatrix} = \mathbf{C} \begin{bmatrix} x_c \\ y_c \\ f \end{bmatrix}$$

$\mathbf{C}$ is a 3×3 upper triangular matrix, called the **camera calibration matrix**:

$$\mathbf{C} = \begin{bmatrix} \alpha_u & 0 & u_0 \\ 0 & \alpha_v & v_0 \\ 0 & 0 & 1 \end{bmatrix}$$

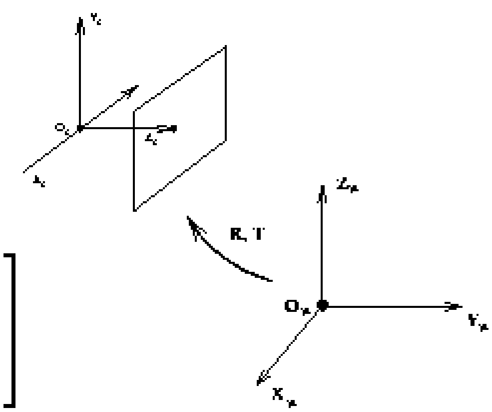
where $\alpha_u = f k_u$, $\alpha_v = -f k_v$.

- $\mathbf{C}$ provides the transformation between an image point and a ray in Euclidean 3-space.
- There are four parameters:
  1. The scaling in the image **x** and **y** directions, α_u and α_v.
  2. The *principal point* (u_0, v_0), which is the point where the optic axis intersects the image plane.

  The *aspect ratio* is α_v/α_u.
- Once $\mathbf{C}$ is known the camera is termed *calibrated*.
- A calibrated camera is a *direction sensor*, able to measure the direction of rays --- like a 2D protractor.

3. **World**: (extrinsic/external camera parameters)
   The Euclidean transformation between the camera and world coordinates is $\mathbf{X}_c = \mathbf{R}\mathbf{X}_w + \mathbf{T}$:

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{R} & \mathbf{T} \\ \mathbf{0}^\top & 1 \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}$$



Finally, concatenating the three matrices,

$$\begin{aligned} \mathbf{x} = \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} &= \mathbf{C} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{R} & \mathbf{T} \\ \mathbf{0}^\top & 1 \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \\ &= \mathbf{C}\,[\mathbf{R}|\ \mathbf{T}] \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \end{aligned}$$

which defines the 3×4 projection matrix from Euclidean 3-space to an image:

$$\mathbf{x} = \mathbf{P} \begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix} \qquad \mathbf{P} = \mathbf{C}\,[\mathbf{R}|\ \mathbf{T}]$$

Bob Fisher
Wed Apr 16 00:58:54 BST 1997

Summary and Properties of the Projection Matrix

3×4 projection matrix from Euclidean 3-space to an image:

$$\mathbf{x} = \mathbf{P} \begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix} \qquad \mathbf{P} = \mathbf{C}\,[\mathbf{R}|\ \mathbf{T}]$$

- The null-space of $\mathbf{P}$ is the optical centre of the camera.
- If the world coordinate system is aligned with the camera coordinate system, then $\mathbf{P} = \mathbf{C}\,[\mathbf{I}|\ \mathbf{0}]$.
- If the image measurements are *normalised*, then $\mathbf{P} = [\mathbf{I}|\ \mathbf{0}]$ (this corresponds to using points $\mathbf{x}_c = \mathbf{C}^{-1}\mathbf{x}_i$).

In the following sections it is often only the 3×4 *form* of $\mathbf{P}$ that is important, rather than its decomposition.

Bob Fisher
Wed Apr 16 00:58:54 BST 1997

Camera Calibration

The perspective projection from Euclidean 3-space to an image is represented as

$$\mathbf{x} = \mathbf{P} \begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \end{bmatrix} \begin{bmatrix} X_i \\ Y_i \\ Z_i \\ 1 \end{bmatrix}$$

$\mathbf{P}$ is a homogeneous 3×4 camera projection matrix (11 dof) with the decomposition

$$\mathbf{P} = \mathbf{C}\,[\mathbf{R}|\ \mathbf{T}]$$

The algorithm for camera calibration has two parts:

1. **Compute the matrix $\mathbf{P}$** from a set of points with known 3D positions and their measured image positions.
2. **Decompose $\mathbf{P}$ into $\mathbf{C}, \mathbf{R}$ and $\mathbf{T}$** via the $\mathbf{QR}$ decomposition.

1. Compute the matrix $\mathbf{P}$: Use correspondences between 3D points $\mathbf{X}_i$ and their 2D images $\mathbf{x}_i$ to determine the matrix $\mathbf{P}$.

1. Each correspondence generates two (linear) equations on the matrix elements of $\mathbf{P}$.

   $$x_i = \frac{p_{11}X_i + p_{12}Y_i + p_{13}Z_i + p_{14}}{p_{31}X_i + p_{32}Y_i + p_{33}Z_i + p_{34}} \qquad y_i = \frac{p_{21}X_i + p_{22}Y_i + p_{23}Z_i + p_{24}}{p_{31}X_i + p_{32}Y_i + p_{33}Z_i + p_{34}}$$

   multiplying out

   $$\begin{aligned} x_i(p_{31}X_i + p_{32}Y_i + p_{33}Z_i + p_{34}) &= p_{11}X_i + p_{12}Y_i + p_{13}Z_i + p_{14} \\ y_i(p_{31}X_i + p_{32}Y_i + p_{33}Z_i + p_{34}) &= p_{21}X_i + p_{22}Y_i + p_{23}Z_i + p_{24} \end{aligned}$$

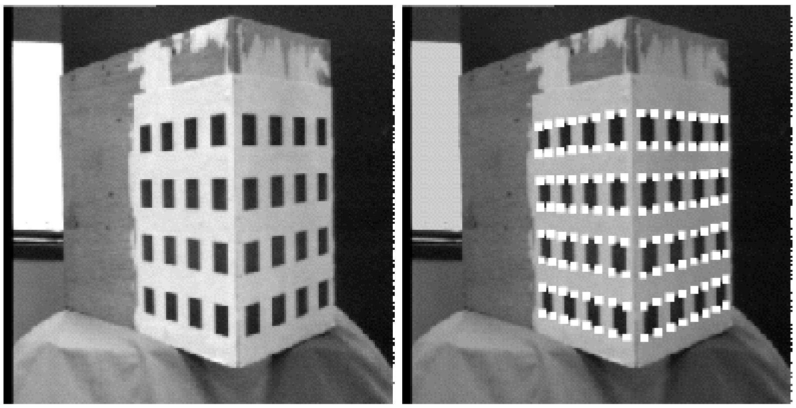
2. Given ($n \geq 6$) correspondences, a linear solution can be obtained for $\mathbf{P}$ from the set of **2n** linear simultaneous equations (cf computation of a projective transformation): $\mathbf{Ap} = \mathbf{0}$, where $\mathbf{p}$ is the 12-vector representation of the projection matrix $\mathbf{P}$, and $\mathbf{A}$ is a $2n \times 12$ matrix. The solution is the

eigenvector with least eigenvalue of $\mathbf{A}^\top \mathbf{A}$.

3. This linear solution is then used as the starting point for a non-linear minimisation of the difference between the measured and projected point:

$$\min_{\mathbf{P}} \Sigma_i \left((x_i, y_i) - P(X_i, Y_i, Z_i)\right)^2$$

Example - Calibration Object



Determine accurate corner positions by

1. Extract and link edges using Canny.
2. Fit lines to edges using orthogonal regression.
3. Intersect lines.

The final error between measured and projected points is typically less than 0.02 pixels.

2. Decompose $\mathbf{P}$ into $\mathbf{C}, \mathbf{R}$ and $\mathbf{T}$:

The first 3×3 submatrix, $\mathbf{M}$, of $\mathbf{P}$ is the product ($\mathbf{M} = \mathbf{CR}$) of an upper triangular and rotation matrix.

1. Factor $\mathbf{M}$ into $\mathbf{CR}$ using the $\mathbf{QR}$ matrix decomposition. This determines $\mathbf{C}$ and $\mathbf{R}$.
2. Then

$$\mathbf{T} = \mathbf{C}^{-1}(p_{14}, p_{24}, p_{34})^\top$$

Note, this procedure produces a matrix with an extra parameter $\mathbf{k}$

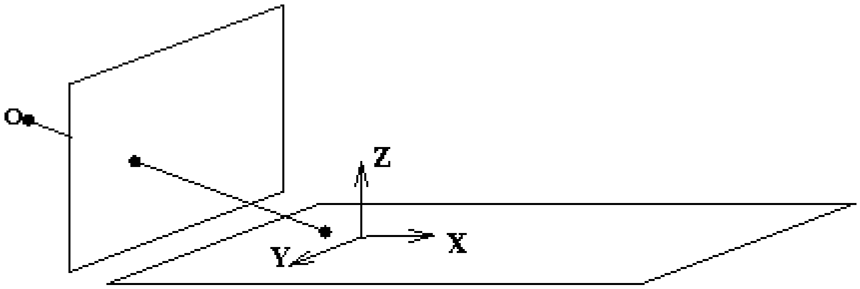
$$\mathbf{C} = \begin{bmatrix} \alpha_u & k & u_0 \\ 0 & \alpha_v & v_0 \\ 0 & 0 & 1 \end{bmatrix}$$

with $k = \tan\theta$, and θ the angle between the image axes.

Bob Fisher
Wed Apr 16 00:58:54 BST 1997

Next Up Previous
Next: Four points define **Up:** No Title **Previous:** Camera Calibration

Plane projective transformations



Choose the world coordinate system such that the plane of the points has zero $\mathbf{Z}$ coordinate. Then the 3×4 matrix $\mathbf{P}$ reduces to

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \end{bmatrix} \begin{bmatrix} X \\ Y \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} p_{11} & p_{12} & p_{14} \\ p_{21} & p_{22} & p_{24} \\ p_{31} & p_{32} & p_{34} \end{bmatrix} \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix}$$

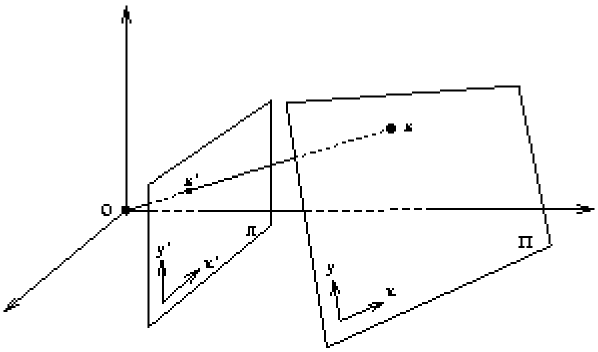
which is a 3×3 matrix representing a general plane to plane projective transformation.

It is often only the 3×3 *form* of the matrix that is important in establishing properties of this transformation.

Bob Fisher
Wed Apr 16 00:58:54 BST 1997

Next Up Previous
Next: Example 1: Removing **Up:** No Title **Previous:** Plane projective transformations

Four points define a planar projective transformation



$$\begin{bmatrix} x_1' \\ x_2' \\ x_3' \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

or $\mathbf{x}' = \mathbf{H}\mathbf{x}$, where $\mathbf{H}$ is a 3×3 non-singular homogeneous matrix.

$$x' = \frac{x_1'}{x_3'} = \frac{h_{11}x + h_{12}y + h_{13}}{h_{31}x + h_{32}y + h_{33}}, \qquad y' = \frac{x_2'}{x_3'} = \frac{h_{21}x + h_{22}y + h_{23}}{h_{31}x + h_{32}y + h_{33}}$$

Each point correspondence generates two linear equations for the elements of $\mathbf{H}$.

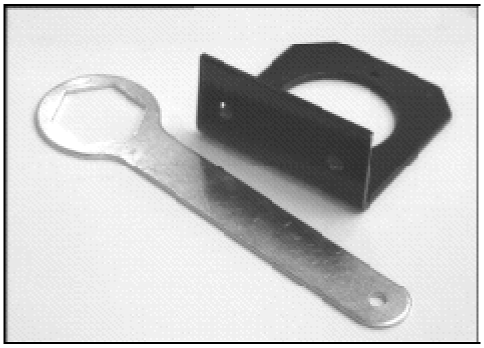
$$\begin{aligned} x'\left(h_{31}x + h_{32}y + h_{33}\right) &= h_{11}x + h_{12}y + h_{13} \\ y'\left(h_{31}x + h_{32}y + h_{33}\right) &= h_{21}x + h_{22}y + h_{23} \end{aligned}$$

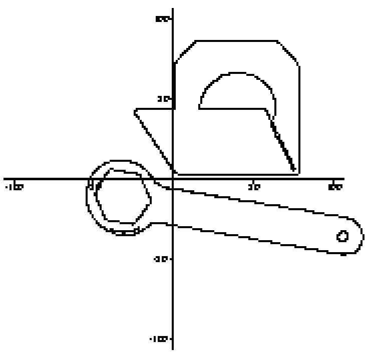
The converse of this is that it is possible to transform any four planar points in general position to any other four planar points in general position by a projectivity.

Bob Fisher
Wed Apr 16 00:58:54 BST 1997

Next Up Previous
Next: Rotation about the **Up:** No Title **Previous:** Four points define

Example 1: Removing Perspective Distortion



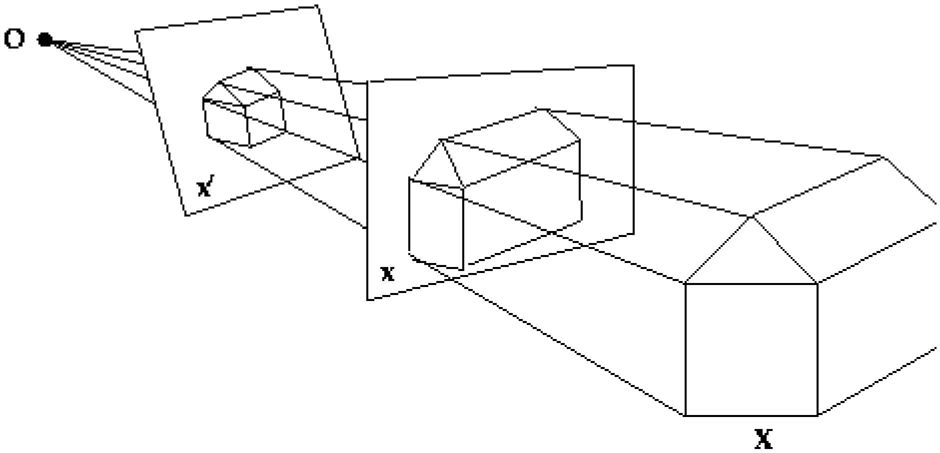


1. Have coordinates of four points on the object plane, $\mathbf{x}_i, \;\; i = 1, 4$ (in the above example they are the corners of the bracket).
2. Solve for $\mathbf{H}$ in $\mathbf{x}' = \mathbf{H}\mathbf{x}$ from (x_i, y_i) and corresponding image coordinates (x_i', y_i'). These are represented in homogeneous coordinates as $\mathbf{x}_i = (x_i, y_i, 1)^\top$ and $\mathbf{x}_i' = (x_i', y_i', 1)^\top$ respectively.
3. Then $\mathbf{x} = \mathbf{H}^{-1}\mathbf{x}'$

Bob Fisher
Wed Apr 16 00:58:54 BST 1997

Next Up Previous
Next: Example 2: Synthetic **Up:** No Title **Previous:** Example 1: Removing

Rotation about the Optical Centre



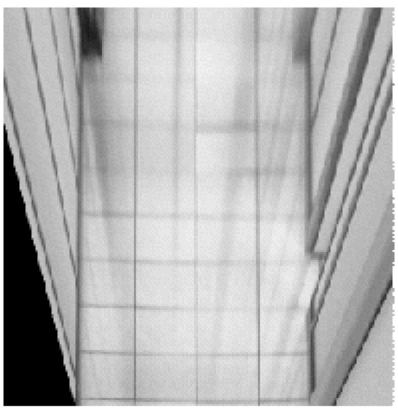
If the image plane is moved, with the optical centre fixed, then corresponding image points are related by a planar projective transformation $\mathbf{x}' = \mathbf{H}\mathbf{x}$:

$$\begin{aligned} \mathbf{x} &= \mathbf{C}\,[\mathbf{I}|\ \mathbf{0}] \begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix} = \mathbf{C}\mathbf{X} \\ \mathbf{x}' &= \mathbf{C}\,[\mathbf{R}|\ \mathbf{0}] \begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix} = \mathbf{C}\mathbf{R}\mathbf{X} \\ &= \mathbf{C}\mathbf{R}\mathbf{C}^{-1}\mathbf{x} \end{aligned}$$

Hence, $\mathbf{H} = \mathbf{C}\mathbf{R}\mathbf{C}^{-1}$ is a conjugate rotation (in general a projective transformation). NB, $\mathbf{H}$ does *not* depend on 3D structure.

Bob Fisher
Wed Apr 16 00:58:54 BST 1997

Example 2: Synthetic Rotations



The upper image is the original image of the corridor. The lower left image is produced by projectively warping the original image so that the floor tiles map to squares. This is simply achieved by computing the plane projective transformation that maps four corners of the imaged square tiles to corners of a square. Similarly, the right image is produced by a projective warping that maps the four imaged corners of the door surround to a rectangle. Both mappings correspond to a synthetic rotation of the camera about the (fixed) camera centre.

Bob Fisher
Wed Apr 16 00:58:54 BST 1997

Computing Plane Projective Transformations

Estimate $\mathbf{H}$ for $\mathbf{x}' = \mathbf{H}\mathbf{x}$,

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

where = is equality up to scale. **Minimum number of correspondences:** 4 points (non-collinear) or 4 lines (non-concurrent).

$n \geq 4$ correspondences

n = 4 will be covered as a special case. Each point correspondence generates two linear equations for the elements of $\mathbf{H}$ (dividing by the third component to remove the unknown scale factor)

$$x' = \frac{h_{11}x + h_{12}y + h_{13}}{h_{31}x + h_{32}y + h_{33}}, \qquad y' = \frac{h_{21}x + h_{22}y + h_{23}}{h_{31}x + h_{32}y + h_{33}}$$

and multiplying out

$$\begin{aligned} x'\left(h_{31}x + h_{32}y + h_{33}\right) &= h_{11}x + h_{12}y + h_{13} \\ y'\left(h_{31}x + h_{32}y + h_{33}\right) &= h_{21}x + h_{22}y + h_{23} \end{aligned}$$

Then $n \geq 4$ points generates **2 n** linear equations, which are sufficient to solve for $\mathbf{H}$. The problem with projective transformations is that the matrix $\mathbf{H}$ is only recovered up to scale.

Homogeneous Solutions There are two methods of dealing with the unknown scale factor in a homogeneous matrix

1. Choose one of the matrix elements to have a certain value. For example, $h_{33} = 1$.
2. Solve for the matrix up to scale.

Method I

If $h_{33} = 1$, then the above equations can be written:

$$\begin{pmatrix} x_1 & y_1 & 1 & 0 & 0 & 0 & -x'_1x_1 & -x'_1y_1 \\ 0 & 0 & 0 & x_1 & y_1 & 1 & -y'_1x_1 & -y'_1y_1 \\ x_2 & y_2 & 1 & 0 & 0 & 0 & -x'_2x_2 & -x'_2y_2 \\ 0 & 0 & 0 & x_2 & y_2 & 1 & -y'_2x_2 & -y'_2y_2 \\ x_3 & y_3 & 1 & 0 & 0 & 0 & -x'_3x_3 & -x'_3y_3 \\ 0 & 0 & 0 & x_3 & y_3 & 1 & -y'_3x_3 & -y'_3y_3 \\ x_4 & y_4 & 1 & 0 & 0 & 0 & -x'_4x_4 & -x'_4y_4 \\ 0 & 0 & 0 & x_4 & y_4 & 1 & -y'_4x_4 & -y'_4y_4 \end{pmatrix} \begin{pmatrix} h_{11} \\ h_{12} \\ h_{13} \\ h_{21} \\ h_{22} \\ h_{23} \\ h_{31} \\ h_{32} \end{pmatrix} = \begin{pmatrix} x'_1 \\ y'_1 \\ x'_2 \\ y'_2 \\ x'_3 \\ y'_3 \\ x'_4 \\ y'_4 \end{pmatrix}$$

for 4 points. A linear solution is then obtained, in the usual manner, by solving the set of linear simultaneous equations. Similarly, for **n** > **4** points, a solution can be obtained using a pseudo-inverse.

The problem with this approach is that if the true solution is $h_{33} = 0$, then this cannot be reached. Consequently, a poor quality estimate of $\mathbf{H}$ will be obtained.

Method II

The equations,

$$\begin{aligned} x'\,(h_{31}x + h_{32}y + h_{33}) &= h_{11}x + h_{12}y + h_{13} \\ y'\,(h_{31}x + h_{32}y + h_{33}) &= h_{21}x + h_{22}y + h_{23} \end{aligned}$$

can be rearranged as

$$\begin{pmatrix} x & y & 1 & 0 & 0 & 0 & -x'x & -x'y & -x' \\ 0 & 0 & 0 & x & y & 1 & -y'x & -y'y & -y' \end{pmatrix} \mathbf{h} = \mathbf{0}$$

where $\mathbf{h} = (h_{11}, h_{12}, h_{13}, h_{21}, h_{22}, h_{23}, h_{31}, h_{32}, h_{33})^\top$ is the matrix $\mathbf{H}$ written as a vector. For 4 points,

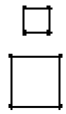
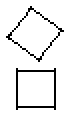
$$\begin{pmatrix} x_1 & y_1 & 1 & 0 & 0 & 0 & -x'_1x_1 & -x'_1y_1 & -x'_1 \\ 0 & 0 & 0 & x_1 & y_1 & 1 & -y'_1x_1 & -y'_1y_1 & -y'_1 \\ x_2 & y_2 & 1 & 0 & 0 & 0 & -x'_2x_2 & -x'_2y_2 & -x'_2 \\ 0 & 0 & 0 & x_2 & y_2 & 1 & -y'_2x_2 & -y'_2y_2 & -y'_2 \\ x_3 & y_3 & 1 & 0 & 0 & 0 & -x'_3x_3 & -x'_3y_3 & -x'_3 \\ 0 & 0 & 0 & x_3 & y_3 & 1 & -y'_3x_3 & -y'_3y_3 & -y'_3 \\ x_4 & y_4 & 1 & 0 & 0 & 0 & -x'_4x_4 & -x'_4y_4 & -x'_4 \\ 0 & 0 & 0 & x_4 & y_4 & 1 & -y'_4x_4 & -y'_4y_4 & -y'_4 \end{pmatrix} \mathbf{h} = \mathbf{0}$$

which has the form $\mathbf{Ah} = \mathbf{0}$, with $\mathbf{A}$ a 8×9 matrix. The solution $\mathbf{h}$ is the (one dimensional) kernel of $\mathbf{A}$.

For **n** > **4** real point correspondences, $\mathbf{A}$ is a $2n \times 9$ matrix, and there will not be a solution to $\mathbf{Ah} = \mathbf{0}$. In this case, a sensible procedure is to again minimise the residuals. It can be shown that the vector that minimises $\mathbf{h}^\top\mathbf{A}^\top\mathbf{Ah}$ subject to $||\mathbf{h}|| = 1$, is the eigenvector with least eigenvector of $\mathbf{A}^\top\mathbf{A}$. In the case of **n = 4** the least eigenvalue is zero, and the eigenvector is the kernel of $\mathbf{A}$.

Bob Fisher
Wed Apr 16 00:58:54 BST 1997

Summary of Plane Projective Transformations

Group	Matrix	Distortion	Invariant properties
projective 8 DOF	$\begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix}$		concurrency and collinearity, **order of** contact: intersection (1 pt contact); tangency (2 pt contact); inflections (3 pt contact with line); tangent discontinuities and cusps. cross-ratio (ratio of ratio of lengths).
affine 6 DOF	$\begin{bmatrix} a_{11} & a_{12} & t_x \\ a_{21} & a_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix}$		parallelism, ratio of areas, ratio of lengths on collinear or parallel lines (e.g. midpoints), linear combinations of vectors
similarity 4 DOF	$\begin{bmatrix} sr_{11} & sr_{12} & t_x \\ sr_{21} & sr_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix}$		ratio of lengths, angle
Euclidean 3 DOF	$\begin{bmatrix} r_{11} & r_{12} & t_x \\ r_{21} & r_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix}$		length, area

Geometric properties invariant to commonly occurring **plane** transformations. Transformations lower in the table inherit the invariants of those above, but the converse is not true. The matrix $\mathbf{A} = [a_{ij}]$ is an invertible 2×2 matrix, $\mathbf{R} = [r_{ij}]$ is a 2D rotation matrix, and (t_x, t_y) a 2D translation. The distortion column shows typical effects of the transformations on a square. Transformations higher in the table can produce all the actions of the ones below. These range from Euclidean, where only translations and rotations occur, to projective where the square can be transformed to any arbitrary quadrilateral (provided no three points are collinear).

Bob Fisher
Wed Apr 16 00:58:54 BST 1997

Two View Geometry

There are two cases of distinct viewpoint which are often treated separately, but have much in common:

- **Stereo**: two images acquired simultaneously.
- **Motion**: two images acquired sequentially.

There are three intertwined goals:

1. **Recovery of 3D structure**
   Recover the 3D position of scene structure from corresponding points in the two images.

2. **Motion Recovery**
   Compute the motion (rotation and translation) of the camera between the two views.

3. **Correspondence**
   Compute points in both images corresponding to the same 3D point.

Here we concentrate on the third point. In particular we will answer the question: given a point in one image, where does the corresponding point in the second image lie? It will be shown that for images of *rigid* scenes this question can be answered very simply and precisely. There are two cases: first, for images of points on a ``known'' plane, the corresponding point is uniquely determined; second, for any image point the corresponding point is constrained to lie on a line (an epipolar line). In both cases these constraints can be computed from image correspondences directly, without requiring camera calibration.

Bob Fisher
Wed Apr 16 00:58:54 BST 1997

Two View Notation

- Euclidean transformation between cameras coordinate systems

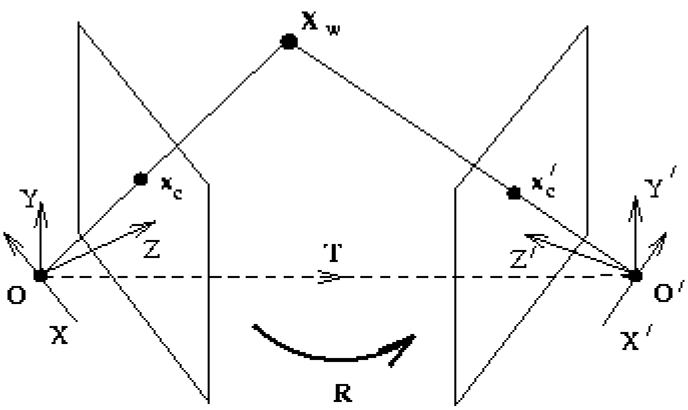
$$\mathbf{X}' = \mathbf{R}\mathbf{X} + \mathbf{T}$$

- Image point in camera coordinate systems:

$$\mathbf{x}_c = f\frac{\mathbf{X}}{Z} \qquad \mathbf{x}'_c = f'\frac{\mathbf{X}'}{Z'}$$

- Image point in pixels:

$$\mathbf{x}_i = \mathbf{P}\mathbf{X}_w = \mathbf{C}[\mathbf{I} \mid \mathbf{0}]\mathbf{X}_w \qquad \mathbf{x}'_i = \mathbf{P}'\mathbf{X}_w = \mathbf{C}'[\mathbf{R} \mid \mathbf{T}]\mathbf{X}_w$$

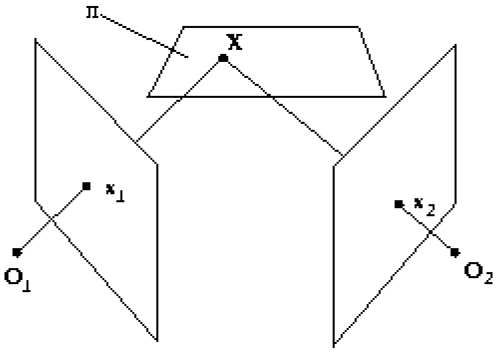


- In the case of a stereo rig, $\mathbf{C}, \mathbf{C}', \mathbf{R}$ and $\mathbf{T}$ are generally known.
- In the case of motion, $\mathbf{C} = \mathbf{C}'$ in general.

Bob Fisher
Wed Apr 16 00:58:54 BST 1997

Images of Planes

Projective transformation between images induced by a plane



$$\mathbf{x}_1 = \mathbf{H}_{1w}\mathbf{X} \qquad \mathbf{x}_2 = \mathbf{H}_{2w}\mathbf{X}$$

$$\begin{aligned} \mathbf{x}_2 &= \mathbf{H}_{2w}\mathbf{X} = \mathbf{H}_{2w}\mathbf{H}_{1w}^{-1}\mathbf{x}_1 \\ &= \mathbf{H}_{21}\mathbf{x}_1 \end{aligned}$$

- Compute $\mathbf{H}_{21}$ from the correspondence of four points on the plane.
- Or, if calibration is known,

$$\mathbf{H}_{21} = \mathbf{C}'(\mathbf{R} + \mathbf{T}\mathbf{n}^\top/d)\mathbf{C}^{-1}$$

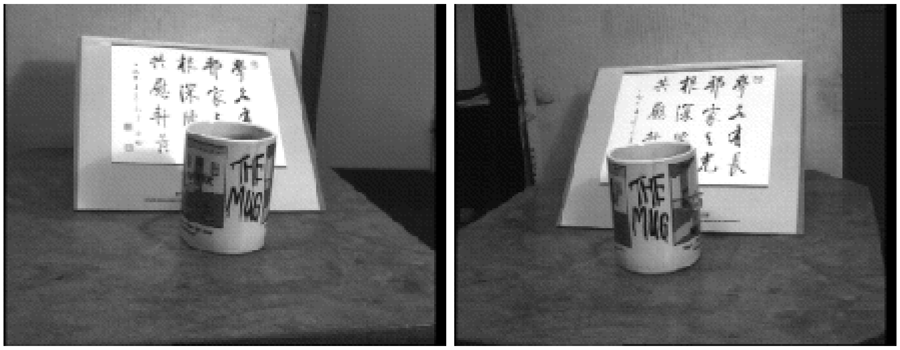
where the equation of the plane in the coordinate system of the first camera is $\mathbf{n}.\mathbf{X} = d$.

Bob Fisher
Wed Apr 16 00:58:54 BST 1997

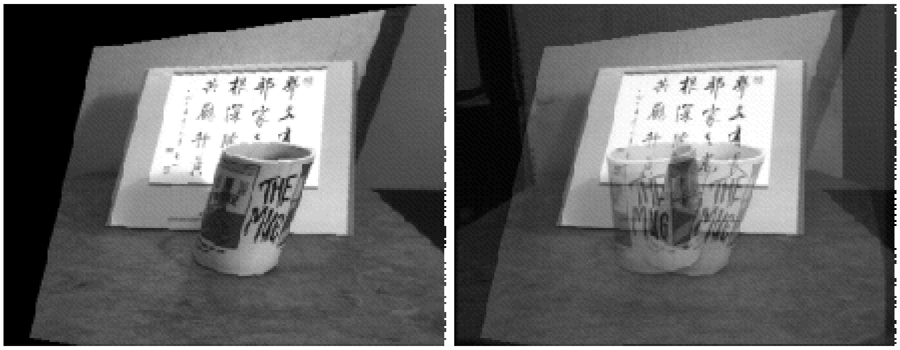
Next Up Previous
Next: Rotation about optical **Up:** No Title **Previous:** Images of Planes

Plane Projective Transfer

Original images (left and right)



Transfer and superimposed images



- The ``transfer" image is the left image projectively warped so that points on the plane containing the Chinese text are mapped to their position in the right image.
- The ``superimpose" image is a superposition of the transfer and right image. The planes exactly cooincide. However, points off the plane (such as the mug) do not cooincide.
- This is an example of planar projectively induced parallax. Lines joining corresponding points off the plane in the ``superimposed" image intersect at the epipole.

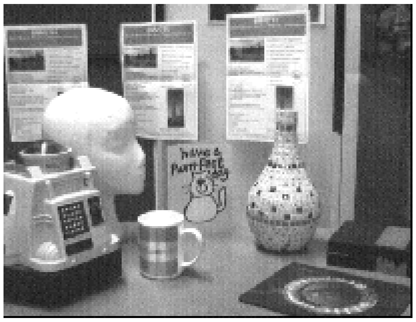
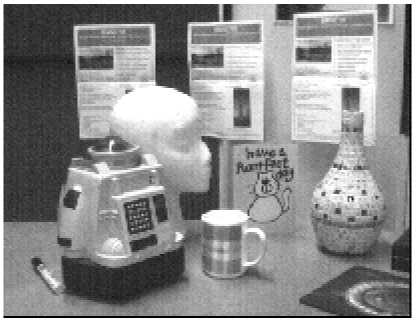
Bob Fisher
Wed Apr 16 00:58:54 BST 1997

Next Up Previous

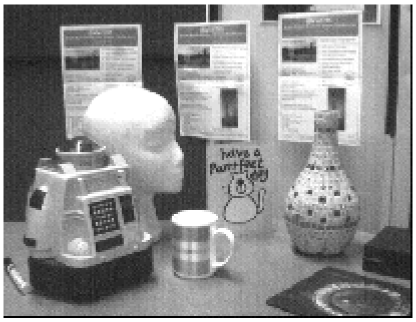
Next: Epipolar Geometry **Up:** No Title **Previous:** Plane Projective Transfer

Rotation about optical centre

Original and rotated views

Rotated and translated views

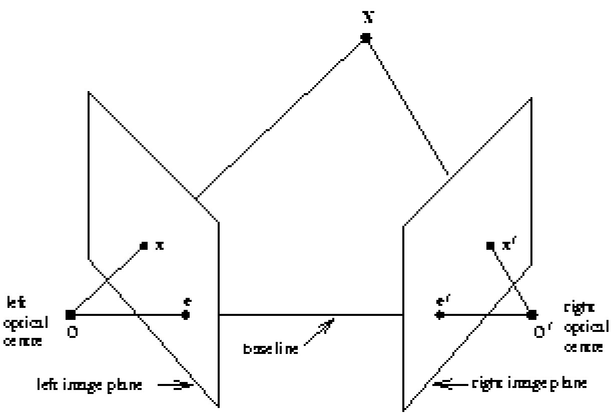
Small and Large translation

- Between the upper images the camera rotates about its centre but does not translate. The images are thus related by a plane projective transformation. There is no parallax.
- Between the upper left and lower images the camera rotates about its centre and translates. The images are not related by a plane projective transformation, and motion parallax is evident. The multiple view relation here is epipolar geometry, which is described on the next slide.

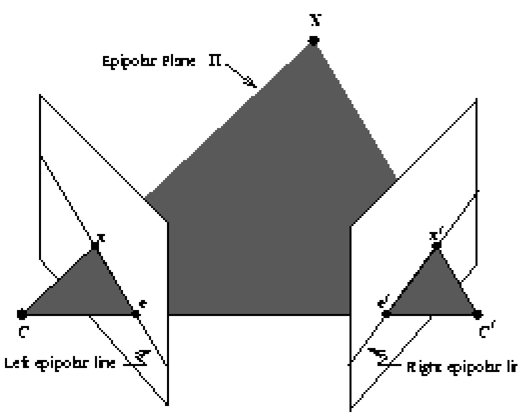
Bob Fisher
Wed Apr 16 00:58:54 BST 1997

Epipolar Geometry

The fundamental **geometric** relationship between two perspective cameras.



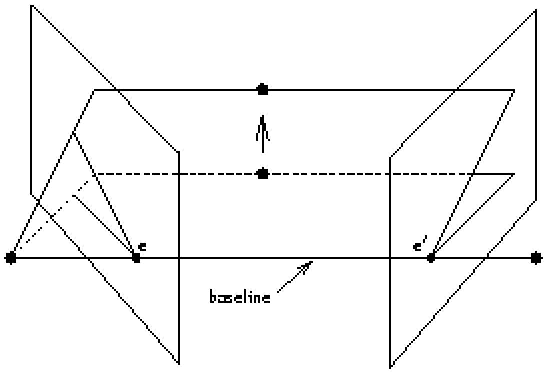
- The **epipole**: is the *point* of intersection of the line joining the optical centres---the *baseline*---with the image plane. The epipole is the image in one camera of the optical centre of the other camera.



- The **epipolar plane**: is the *plane* defined by a 3D point and the optical centres. Or, equivalently, by an image point and the optical centres.

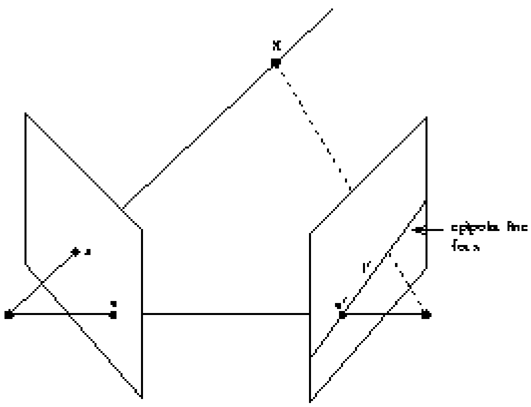
- The **epipolar line**: is the *straight line* of intersection of the epipolar plane with the image plane. It is the image in one camera of a ray through the optical centre and image point in the other camera. All epipolar lines intersect at the epipole.

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.

Correspondences between images

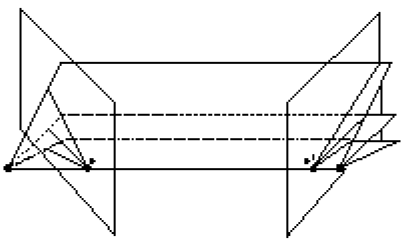


A **point** in one image generates a **line** in the other on which its corresponding point must lie. The search for correspondences is thus reduced from a region to a line. This **epipolar constraint** arises because, for image points corresponding to the same 3D point, the image points, 3D point and optical centres are coplanar.

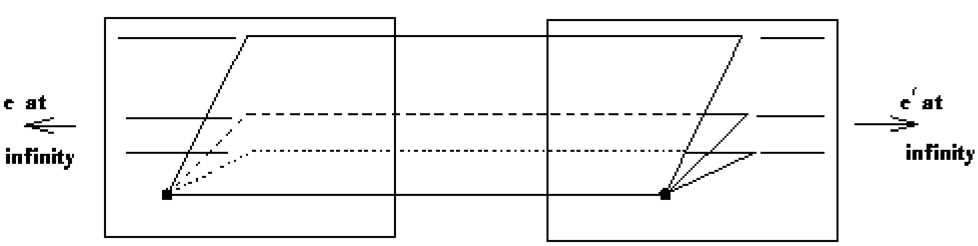
Bob Fisher
Wed Apr 16 00:58:54 BST 1997

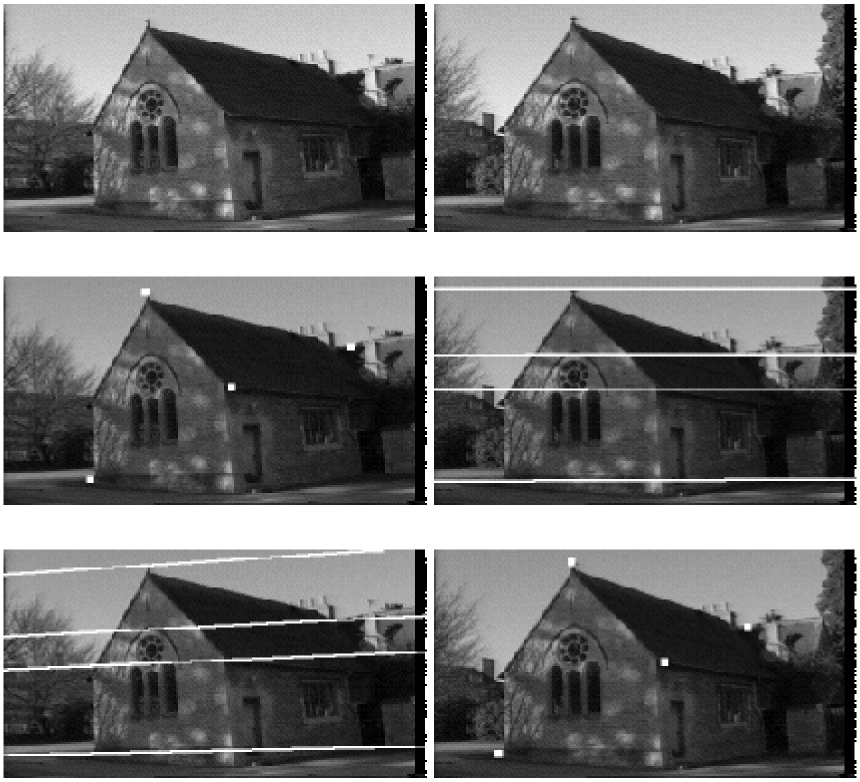
Epipolar geometry examples

Example 1: Converging cameras



Example 2: Near parallel cameras





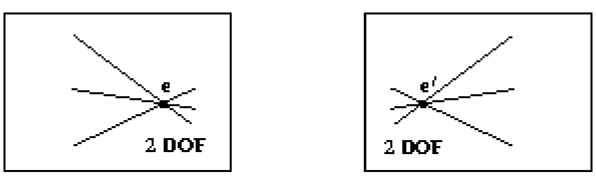
In general epipolar lines are *not* parallel.

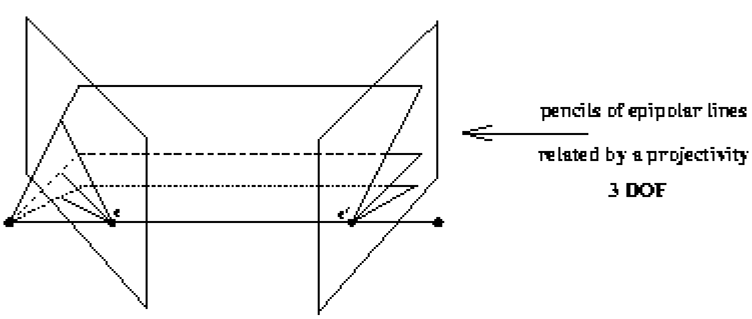
Degrees of Freedom

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

There are 7 degrees of freedom:

- 2 to specify the epipole in each image.
- 3 to specify the projectivity between epipolar line pencils.





The **fundamental matrix** is the algebraic representation of epipolar geometry.

Bob Fisher
Wed Apr 16 00:58:54 BST 1997

Homogeneous Notation Interlude

Notation for plane (i.e. 2D) projective geometry. NB ``=" means equality up to scale.

- **point**: a point $\mathbf{x}$ is represented by the homogeneous 3-vector

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

  where $x = x_1/x_3 \quad y = x_2/x_3$ and (x, y) are the inhomogeneous plane coordinates. Only the ratio of the homogeneous coordinates is significant.

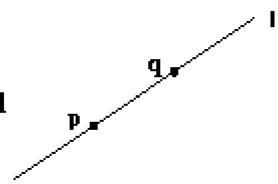
- **line**: a line $\mathbf{l}$ is represented by the homogeneous 3-vector

$$\mathbf{l} = \begin{bmatrix} l_1 \\ l_2 \\ l_3 \end{bmatrix}$$

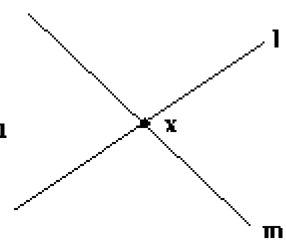
  for the line $l_1 x + l_2 y + l_3 = 0$. Again, only the ratio of the homogeneous line coordinates is significant.

- **point on line**: $\mathbf{l}.\mathbf{x} = 0$ or $\mathbf{l}^\top \mathbf{x} = 0$ or $\mathbf{x}^\top \mathbf{l} = 0$

- **two points define a line**: $\mathbf{l} = \mathbf{p} \times \mathbf{q}$



- **two lines define a point**: $\mathbf{x} = \mathbf{l} \times \mathbf{m}$



- **matrix notation for vector product**: The vector product $\mathbf{v} \times \mathbf{x}$ can be represented as a matrix multiplication

$$\mathbf{v} \times \mathbf{x} = [\mathbf{v}]_{\times} \mathbf{x}$$

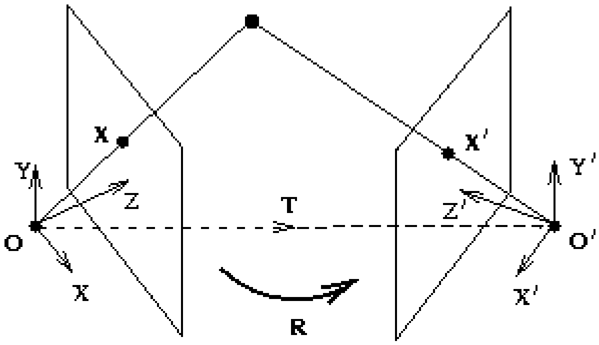
where

$$[\mathbf{v}]_{\times} = \begin{bmatrix} 0 & -v_z & v_y \\ v_z & 0 & -v_x \\ -v_y & v_x & 0 \end{bmatrix}$$

$[\mathbf{v}]_{\times}$ is a 3×3 skew-symmetric matrix of rank 2. $\mathbf{v}$ is the one dimensional kernel of $[\mathbf{v}]_{\times}$, since $\mathbf{v} \times \mathbf{v} = [\mathbf{v}]_{\times} \mathbf{v} = 0$.

Bob Fisher
Wed Apr 16 00:58:54 BST 1997

The Essential Matrix



There are two camera coordinate systems, related by a rotation $\mathbf{R}$ and a translation $\mathbf{T}$:

$$\mathbf{X}' = \mathbf{R}\mathbf{X} + \mathbf{T}$$

Taking the vector product with $\mathbf{T}$, followed by the scalar product with $\mathbf{X}'$

$$\mathbf{X}'.(\mathbf{T} \times \mathbf{R}\mathbf{X}) = 0,$$

which expresses that the vectors $\mathbf{OX}$, $\mathbf{O'X}$ and $\mathbf{OO'}$ are coplanar. Using

$$\mathbf{x}_c = f\frac{\mathbf{X}}{Z} \qquad \mathbf{x}'_c = f'\frac{\mathbf{X}'}{Z'}$$

gives a relation between image points in the camera coordinate systems, $\mathbf{x}'_c.(\mathbf{T} \times \mathbf{R}\mathbf{x}_c) = 0$. This can be written ${\mathbf{x}'_c}^\top \mathbf{E}\mathbf{x}_c = 0$, where

$$\mathbf{E} = [\mathbf{T}]_\times \mathbf{R}$$

is the **Essential** matrix. This is the algebraic representation of epipolar geometry for known calibration.

Then using the calibration matrices to relate the image point in pixels to the point in the camera coordinate system:

$$\mathbf{x}'^\top \mathbf{C}'^{-\top} \mathbf{E} \mathbf{C}^{-1} \mathbf{x} = 0$$

which defines the **Fundamental** matrix, $\mathbf{F}$ by

$$\mathbf{x}'^\top \mathbf{F} \mathbf{x} = 0$$

where

$$\mathbf{F} = \mathbf{C}'^{-\top}\mathbf{E}\mathbf{C}^{-1}$$

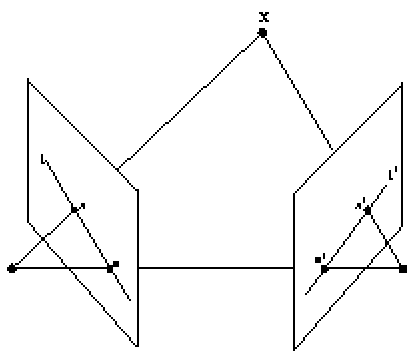
Bob Fisher
Wed Apr 16 00:58:54 BST 1997

The Fundamental Matrix

If $\mathbf{x}$ and $\mathbf{x}'$ are corresponding image points, then

$$\begin{bmatrix} u' & v' & 1 \end{bmatrix} \begin{bmatrix} & & \\ & \mathbf{F} & \\ & & \end{bmatrix} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = 0$$

i.e. $\mathbf{x}'^{\top}\mathbf{F}\mathbf{x} = 0$, where $\mathbf{F}$ is the 3×3 fundamental matrix of maximum rank 2.

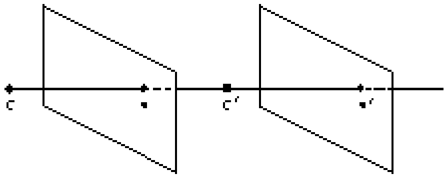


Properties

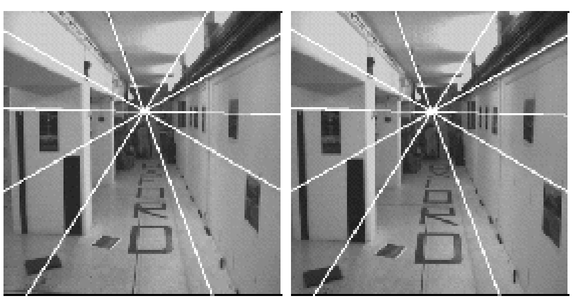
- **Epipolar lines:**
  - $\mathbf{l}' = \mathbf{F}\mathbf{x}$ **is the epipolar line corresponding to $\mathbf{x}$, since $\mathbf{x}'^{\top}\mathbf{l}' = 0$.**
  - $\mathbf{l} = \mathbf{F}^{\top}\mathbf{x}'$ **is the epipolar line corresponding to $\mathbf{x}'$, since $\mathbf{l}^{\top}\mathbf{x} = 0$.**
- **Epipoles:**
  - $\mathbf{F}\mathbf{e} = 0$
  - $\mathbf{F}^{\top}\mathbf{e}' = 0$
- **$\mathbf{F}$ has 7 dof --- there are 9 matrix elements but only their ratio is significant, which leaves 8 dof. In addition the elements satisfy the constraint $\det \mathbf{F} = 0$ which leaves 7 dof.**
- **$\mathbf{F}$ depends only on the *direction* of translation (unaffected by the magnitude of translation).**

Example 1: Pure translation of a `fixed' camera $\mathbf{C} = \mathbf{C}'$, $\mathbf{R} = \mathbf{I}$, then

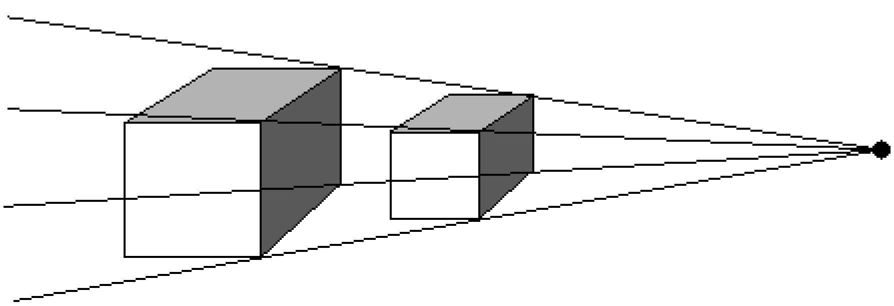
$$\begin{aligned} \mathbf{F} &= \mathbf{C}'^{-\top}[\mathbf{T}]_\times \mathbf{R}\mathbf{C}^{-1} \\ &= \mathbf{C}^{-\top}[\mathbf{T}]_\times \mathbf{C}^{-1} = [\mathbf{C}\mathbf{T}]_\times = [\mathbf{e}']_\times = [\mathbf{e}]_\times \end{aligned}$$



- **$\mathbf{F}$ has only 2 dof corresponding to the position of the epipole.**
- **Points lie on their corresponding epipolar line drawn *in the same image*, since $\mathbf{l}' = [\mathbf{e}]_\times \mathbf{x} = \mathbf{e} \times \mathbf{x}$, and $\mathbf{x}.(\mathbf{e} \times \mathbf{x}) = 0$. This is termed `auto-epipolar'.**
- **Points appear to move along lines radiating from the epipole. The epipole in this case is termed the Focus of Expansion (FOE).**



Connection with vanishing points The epipole is the vanishing point of the translation direction.



Example - pure translation continued If the camera translation is parallel to the x axis, then

$$\mathbf{e} = \mathbf{e}' = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

$$\mathbf{F} = [\mathbf{e}]_\times = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix}$$

From $\mathbf{x'}^\top \mathbf{F} \mathbf{x} = 0$

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

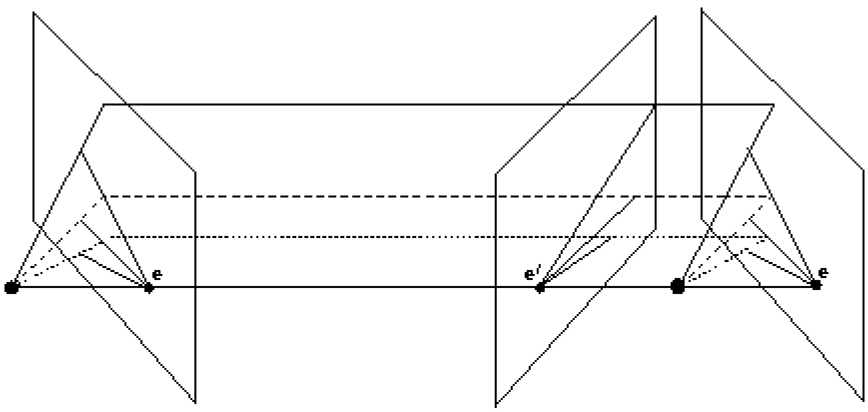
So that $y_i = y_i'$**, i.e. the epipolar lines are corresponding rasters.**

Bob Fisher
Wed Apr 16 00:58:54 BST 1997

Example 2 - General Motion

$\mathbf{R} \neq \mathbf{I}$:

- A point $\mathbf{x}'$ corresponding to $\mathbf{x}$ *always* lies on its epipolar line $\mathbf{l}'$, but in general $\mathbf{x}$ does not lie on $\mathbf{l}'$.
- A general motion can be thought of as a pure translation of the optical centre, followed by a rotation of the image plane about the optical centre. The effects of the rotation can be ``undone'' by a projective transformation



Computing $\mathbf{F}$: I Number of Correspondences Given perfect image points (no noise) in general position. Each point correspondence $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$ generates one constraint on $\mathbf{F}$.

$$\begin{bmatrix} x'_i & y'_i & 1 \end{bmatrix} \begin{bmatrix} f_1 & f_2 & f_3 \\ f_4 & f_5 & f_6 \\ f_7 & f_8 & f_9 \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} = 0$$

This can be rearranged as $\mathbf{A}\mathbf{f} = \mathbf{0}$ where $\mathbf{A}$ is a $n \times 9$ measurement matrix, and $\mathbf{f}$ is the fundamental matrix represented as a 9-vector.

$$\begin{bmatrix} x'_1x_1 & x'_1y_1 & x'_1 & y'_1x_1 & y'_1y_1 & y'_1 & x_1 & y_1 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \\ x'_nx_n & x'_ny_n & x'_n & y'_nx_n & y'_ny_n & y'_n & x_n & y_n & 1 \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \\ f_6 \\ f_7 \\ f_8 \\ f_9 \end{bmatrix} = \mathbf{0}$$

- **8 or more points in general position**: The dimension of the null-space of $\mathbf{A}$ is one. $\mathbf{f}$, and consequently $\mathbf{F}$, is determined uniquely up to scale.

- **7 points in general position**: The dimension of the null-space is two.

  Suppose the 9-vectors $\mathbf{u_1}$ and $\mathbf{u_2}$, corresponding to the matrices $\mathbf{U_1}$ and $\mathbf{U_2}$ respectively, span the null-space. Then, up to scale, there is a one-parameter family of matrices: $\alpha\mathbf{U_1} + (1-\alpha)\mathbf{U_2}$. Imposing $\det \mathbf{F} = 0$

$$\det |\alpha\mathbf{U}_1 + (1-\alpha)\mathbf{U}_2| = 0$$

  which is a cubic in α, and thus has either one or three real solutions.

Bob Fisher
Wed Apr 16 00:58:54 BST 1997

Computing F: II Least Squares Solutions

Given **n** corresponding points (**n** is typically hundreds) --- inevitably the points will be `` **noisy**'' and there will be **mis-matches** (outliers).

Linear solution --- ``The 8-point Algorithm''

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

This equation will not be satisfied exactly (for **n** > **8**). A solution is obtained by:

$$\min_{\mathbf{f}} ||\mathbf{A}\mathbf{f}||^2 \quad \text{subject to } ||\mathbf{f}|| = 1$$

This is a standard linear algebra problem, the solution is the eigenvector with minimum eigenvalue of $\mathbf{A}^\top\mathbf{A}$.

This computation is generally poorly conditioned, and it is very important to **pre-condition** the matrix: i.e. (affine) transform the image points:

$$\tilde{\mathbf{x}}_i = \mathbf{H}\mathbf{x}_i \quad \tilde{\mathbf{x}}'_i = \mathbf{H}'\mathbf{x}'_i$$

where $\mathbf{H}$ and $\mathbf{H}'$ are 3×3 matrices, such that

$$\begin{aligned} <\tilde{x}_i> &= <\tilde{x}'_i> = 0 \\ <\tilde{y}_i> &= <\tilde{y}'_i> = 0 \\ <\tilde{x}_i^2> &= <\tilde{x}_i'^2> = 1 \\ <\tilde{y}_i^2> &= <\tilde{y}_i'^2> = 1 \end{aligned}$$

and < > indicates averages. A fundamental matrix $\tilde{\mathbf{F}}$ is then computed from the transformed points $\tilde{\mathbf{x}}_i, \tilde{\mathbf{x}}'_i$ and $\mathbf{F}$ obtained by

$$\mathbf{F} = \mathbf{H}'^\top \tilde{\mathbf{F}} \mathbf{H}$$

Pre-conditioning can make a difference of an order of magnitude to the average distance of a point from its epipolar lines (e.g. reduce an average of 3.5 to 0.3 pixels) and hundreds of pixels in the position of the epipoles.

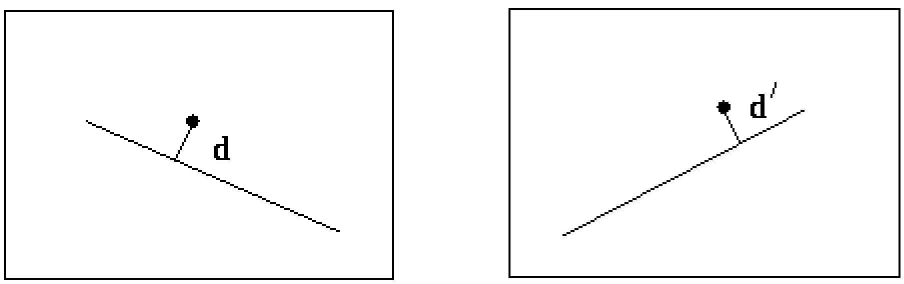
Non-linear solution

Minimise (average) squared perpendicular distance of points from their epipolar lines:

$$\min_{\mathbf{f}} \sum_i (d_{\perp i}^2 + d'^2_{\perp i})$$

where

$$(d_{\perp i}^2 + d'^2_{\perp i}) = (\mathbf{x}'^{\top}_i \mathbf{F}\mathbf{x}_i)^2 \left(\frac{1}{(\mathbf{F}\mathbf{x}_i)_1^2 + (\mathbf{F}\mathbf{x}_i)_2^2} + \frac{1}{(\mathbf{F}^{\top} \mathbf{x}'_i)_1^2 + (\mathbf{F}^{\top} \mathbf{x}'_i)_2^2} \right)$$



Use Levenberg-Marquardt or Powell for non-linear optimisation.

Summary Point

- Epipolar geometry is the fundamental geometric relationship between two images.
- It is represented algebraically by the fundamental matrix $\mathbf{F}$.
- $\mathbf{F}$ can be computed using image correspondences alone (7 or more).

Bob Fisher
Wed Apr 16 00:58:54 BST 1997

Mismatches

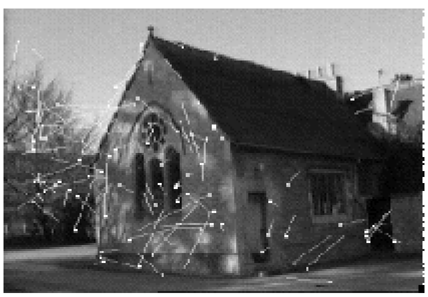
How to detect and eliminate mismatches? Use a *robust* technique, e.g.\

1. Eliminate points far from their epipolar lines *after* computing **F**.
2. Use RANSAC (random sampling consensus) approach --- compute putative **F** using minimum number of matches (**n=7**), and measure support. Repeat over randomly chosen sets of 7 matches (use proximity). Choose **F** with largest support.

Matches computed from corners alone

Matches using RANSAC (Torr)

Mismatches

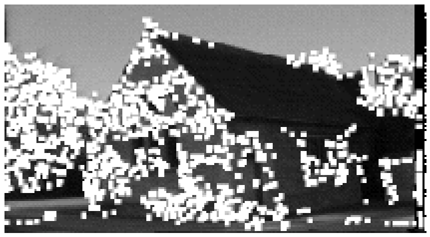
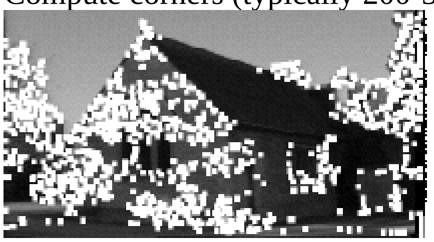


Bob Fisher
Wed Apr 16 00:58:54 BST 1997

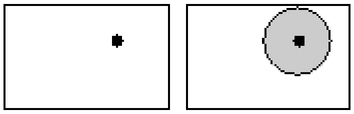
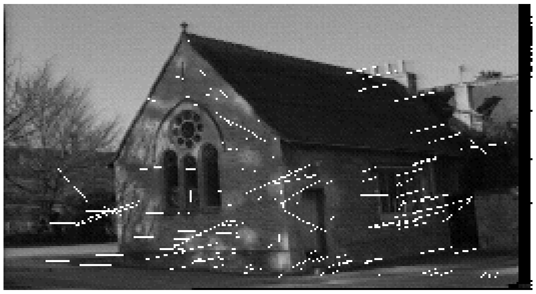
Next Up Previous
Next: Application --- Binary **Up:** No Title **Previous:** Mismatches

How to Establish Correspondences

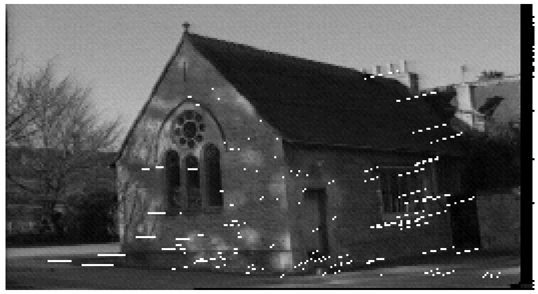
 

Compute corners (typically 200-300 per image): 


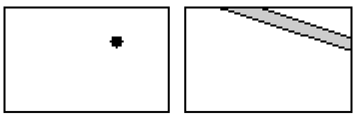
Correspondences are computed automatically and robustly in three stages:

1. **Unguided matching:** obtain a small number of *seed* matches using a local search and normalised cross-correlation with a conservative threshold.
    

2. **Compute epipolar geometry:** use seed matches to robustly compute **F** (using RANSAC).

Matches consistent with $\mathbf{F}$ 

3. **Guided matching:** Search for matches is then restricted to a band about epipolar lines using the computed $\mathbf{F}$.




Typically: number of corners in a 512×512 image is about 300

number of seed matches is about 50-100
final number of matches is about 200-250
distance of a point from its epipolar line is $\sim$0.2pixels (using corners computed to sub-pixel accuracy).

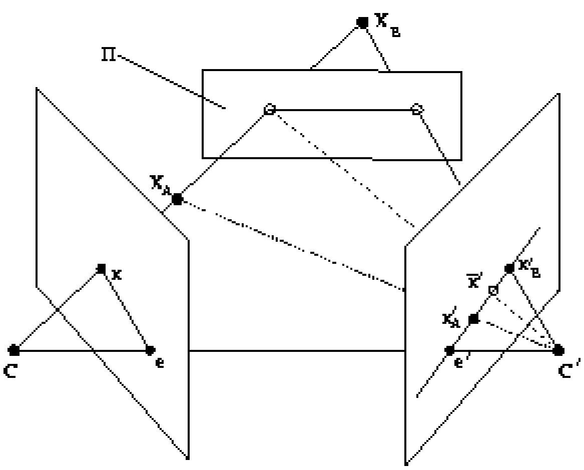
Bob Fisher
Wed Apr 16 00:58:54 BST 1997

Application --- Binary Space Partition (BSP)

Three points in 3-space define a plane π. The images of these points, together with the epipoles, provide the four correspondences necessary to compute the compatible transformation $\mathbf{H}$ for π. That is, the projective transformation $\mathbf{H}$ such that

$$\begin{aligned} \mathbf{x}'_i &= \mathbf{H}\mathbf{x}_i, i \in \{1,..,3\} \\ \mathbf{e}' &= \mathbf{H}\mathbf{e} \end{aligned}$$

transfers points coplanar with $\mathbf{X}_i, i \in \{1,..,3\}$. Given $\mathbf{H}$ it is then possible to distinguish points in 3-space on either side of π, using only their image projections and $\mathbf{F}$ i.e. a binary partition of 3-space.

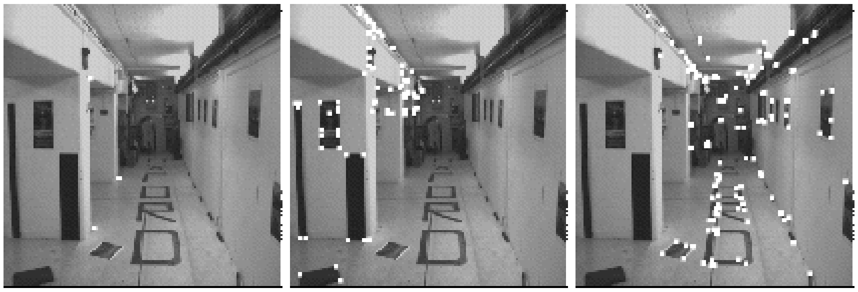


- The three points need not actually correspond to images of physical points, the method can be applied to ``virtual'' planes.
- Points contained in *regions* of 3-space can be identified using a set of binary space partitions for the planes which bound the region.

Original images and points

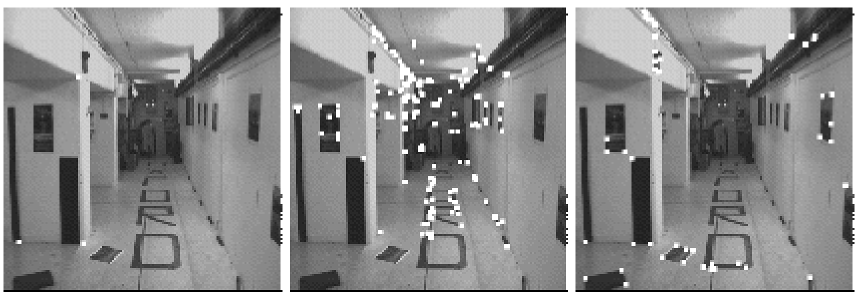
First and second images and corners

BSP 1



Selected planar points, on one side and on the other side

BSP 2



More selected planar points, on one side and on the other side

---

Bob Fisher
Wed Apr 16 00:58:54 BST 1997

Related Reading

- Beardsley, P., Torr, P. H. S., and Zisserman, A. 3D model acquisition from extended image sequences. In B. Buxton and Cipolla R., editors, *Proc. 4th European Conference on Computer Vision, LNCS 1065, Cambridge*, pages 683--695. Springer--Verlag, 1996.
- Deriche R., Zhang Z., Luong Q. T., and Faugeras O., Robust recovery of the epipolar geometry for an uncalibrated stereo rig. In J. O. Eckland, editor, *Proc. 3rd European Conference on Computer Vision, LNCS 800/801, Stockholm*, pages 567--576. Springer-Verlag, 1994.
- Faugeras, O. ``What can be seen in three dimensions with an uncalibrated stereo rig?'', *Proc. ECCV*, LNCS 588. Springer-Verlag, 1992.
- Faugeras, O., ``Three-Dimensional Computer Vision'', MIT Press, 1993.
- Faugeras, O. ``Stratification of three-dimensional vision: projective, affine, and metric representation'', *J. Opt. Soc. Am.*, A12:465--484, 1995.
- Hartley, R., Gupta, R., and Chang, T. ``Stereo from uncalibrated cameras'', *Proc. CVPR*, 1992.
- Hartley, R. I., In defence of the 8-point algorithm. In *ICCV5*, 1995.
- Kanatani, K., *Geometric Computation for Machine Vision*. Oxford University Press, Oxford, 1992.
- Longuet-Higgins H.C., A computer algorithm for reconstructing a scene from two projections. *Nature*, vol.293:133--135, 1981.
- Mundy, J. and Zisserman, A., ``Geometric invariance in computer vision'', MIT Press, 1992. Introduction and Chapter 23 (projective geometry).
- Semple, J. and Kneebone, G. *Algebraic Projective Geometry*. Oxford University Press, 1979.
- Torr P. H. S., *Outlier Detection and Motion Segmentation*. PhD thesis, University of Oxford, Engineering Dept., 1995.

Bob Fisher
Wed Apr 16 00:58:54 BST 1997