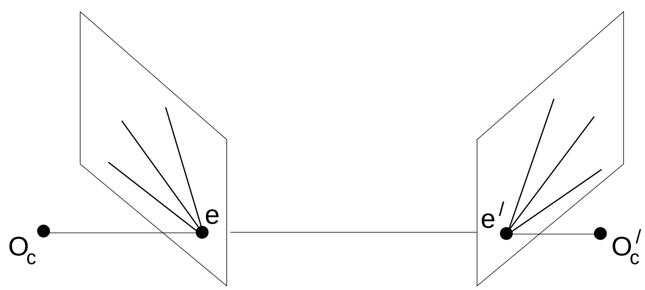
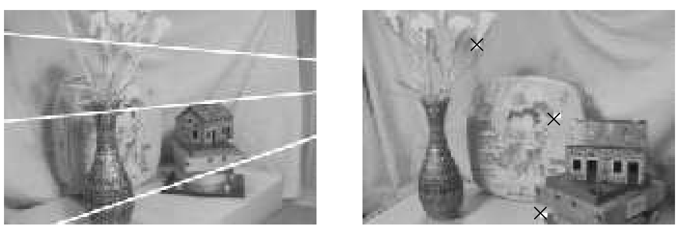
University of Cambridge
Engineering Part IIB & EIST Part II

Module I12: Computer Vision and Robotics

Handout 4: Stereo Vision

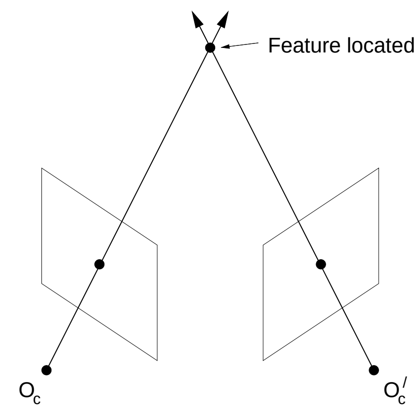




Andrew Gee & Roberto Cipolla
October 1999

Stereo vision

We have seen how it is impossible to recover 3D scene structure from a single image. Even if the camera is calibrated, we can only deduce the *ray* on which each image feature lies.



If we can observe the same feature in two views, however, we can solve for the intersection of the rays and recover the 3D location of the feature. This is the essence of **stereo vision**.

While this might sound straightforward, there are many subtleties to stereo vision. For instance, to what extent do we need to calibrate the cameras? How do we establish correspondences between features in the two views?

Recovering 3D structure

If the left and right cameras are calibrated with respect to the world coordinate system, then it is straightforward to recover 3D structure.

Recall from handout 3 that each point observed by one camera gives us two equations in three unknowns (X, Y, Z):

$$u = \frac{su}{s} = \frac{p_{11}X + p_{12}Y + p_{13}Z + p_{14}}{p_{31}X + p_{32}Y + p_{33}Z + p_{34}}$$

$$v = \frac{sv}{s} = \frac{p_{21}X + p_{22}Y + p_{23}Z + p_{24}}{p_{31}X + p_{32}Y + p_{33}Z + p_{34}}$$

Observing the same point with the other camera provides two further equations. The system of four equations in three unknowns is over-constrained.

To understand what is required for the equations to be consistent, we need to reformulate the equations in terms of 3D vectors. The analysis will also identify a key constraint to help with the correspondence problem.

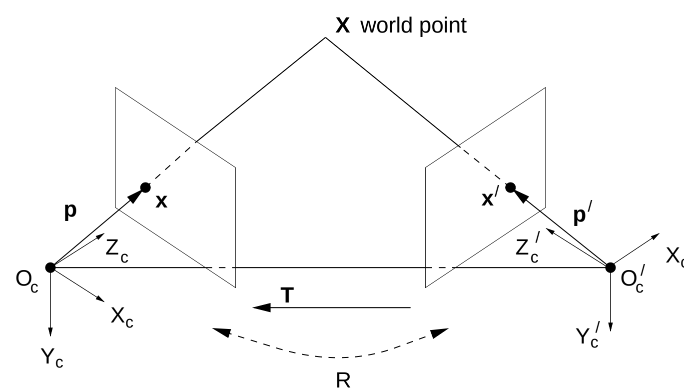
Triangulation

Suppose we know the *relative* positions of the cameras and their *intrinsic* parameters[1]. Given the CCD parameters, we can translate pixel coordinates (u, v) into image plane coordinates (x, y):

$$u = u_0 + k_u x \ , \quad v = v_0 + k_v y$$

With the focal length, we can translate image plane coordinates into a ray in 3D space. Let's define the ray by the point $\mathbf{p}$ (in camera-centered coordinates) where it pierces the image plane:

$$\mathbf{p} = \begin{bmatrix} x \\ y \\ f \end{bmatrix}$$



[1] We can extract most of this information from the two calibration matrices.

Triangulation

Ray vectors $\mathbf{p}$ and world positions $\mathbf{X}_{\mathrm{c}}$ are related via the unknown depth Z_c:

$$\mathbf{p} = \begin{bmatrix} x \\ y \\ f \end{bmatrix} = \begin{bmatrix} fX_c/Z_c \\ fY_c/Z_c \\ fZ_c/Z_c \end{bmatrix} = \frac{f}{Z_c}\mathbf{X}_{\mathrm{c}}$$

With two rays from two calibrated cameras, we can locate the point where the rays intersect (assuming they do). We assume that the two camera-centered coordinate systems are related by a *known* rotation R and translation $\mathbf{T}$ (defined in the right camera's coordinate system):

$$\mathbf{X}'_{\mathrm{c}} = \mathrm{R}\mathbf{X}_{\mathrm{c}} + \mathbf{T}$$

Since $\mathbf{X}'_{\mathrm{c}}$ and $\mathbf{p}'$ are parallel, we have

$$\begin{aligned} \mathbf{X}'_{\mathrm{c}} \times \mathbf{p}' &= \mathbf{0} \\ \Leftrightarrow (\mathrm{R}\mathbf{X}_{\mathrm{c}} + \mathbf{T}) \times \mathbf{p}' &= \mathbf{0} \\ \Leftrightarrow \left(\frac{Z_c}{f}\mathrm{R}\,\mathbf{p} + \mathbf{T}\right) \times \mathbf{p}' &= \mathbf{0} \end{aligned}$$

This provides us with three equations in the single unknown Z_c. If the equations are inconsistent, the rays do not intersect and the left and right image features do *not* correspond to the same world point. Given Z_c, we can recover full 3D scene structure using $\mathbf{X}_{\mathrm{c}} = (Z_c/f)\mathbf{p}$.

Triangulation: example

Let's consider the case when the image planes of the two cameras are aligned and the cameras have the same focal length:

$$\mathrm{R} = \mathrm{I}\,, \quad \mathbf{T} = [\,-d \;\; 0 \;\; 0\,]^T$$

The triangulation equations reduce to:

$$\begin{aligned} \frac{Z_c}{f}(\mathbf{p} \times \mathbf{p}') &= -\mathbf{T} \times \mathbf{p}' \\ \Leftrightarrow Z_c(\mathbf{p} \times \mathbf{p}') &= f \begin{bmatrix} d \\ 0 \\ 0 \end{bmatrix} \times \mathbf{p}' \\ \Leftrightarrow Z_c \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ x & y & f \\ x' & y' & f \end{vmatrix} &= f \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ d & 0 & 0 \\ x' & y' & f \end{vmatrix} \end{aligned}$$

Equating coefficients in $\mathbf{i}$, $\mathbf{j}$ and $\mathbf{k}$:

$$Z_c f(y - y') = 0 \qquad (1)$$

$$Z_c f(x - x') = df^2 \qquad (2)$$

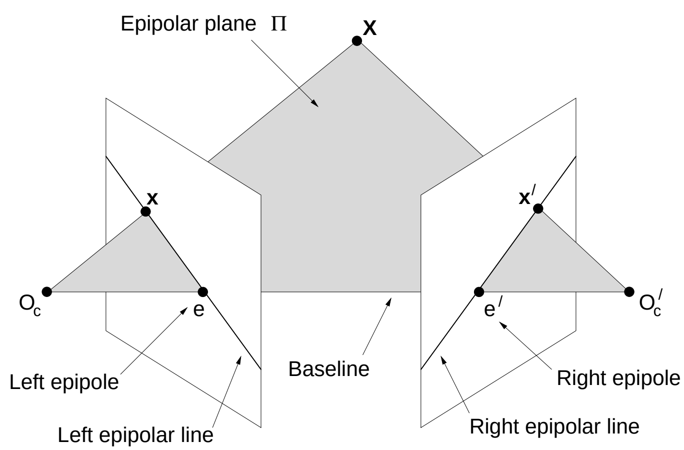
$$Z_c(xy' - yx') = fdy' \qquad (3)$$

(3) is not independent of (1) and (2). (2) allows us to recover the depth from the horizontal **disparity** $(x - x')$: $Z_c = df/(x - x')$. This result is intuitively correct: distant objects have smaller disparities than nearby objects.

Epipolar geometry

Equation (1) tells us that corresponding features lie on the same horizontal line in left and right image planes. This is an example of an **epipolar constraint**, and follows directly from the fact that the rays must intersect in 3D space.

Epipolar constraints are useful when searching for correspondences: they constrain the search to a line in each image. To derive general epipolar constraints, we'll investigate the **epipolar geometry** of two cameras.



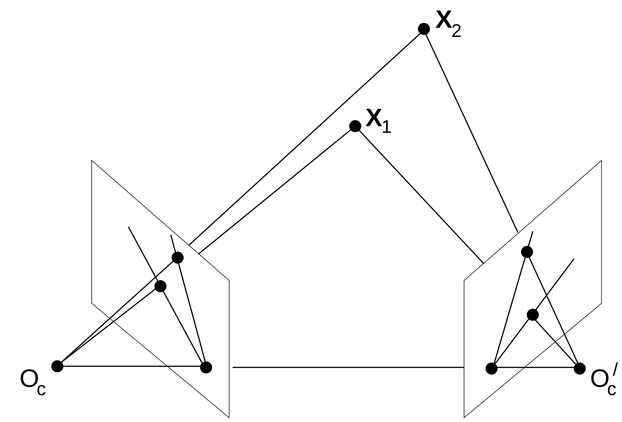
The **epipolar plane** is the plane defined by a 3D point **X** and the optical centres.

Epipolar geometry

The **baseline** is the line joining the optical centres.

An **epipole** is the point of intersection of the baseline with the image plane. There are two epipoles, one for each image.

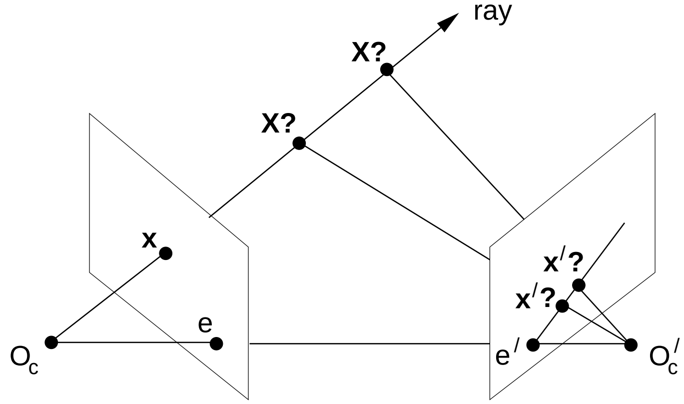
An **epipolar line** is a line of intersection of the epipolar plane with an image plane. It is the image in one camera of the ray from the other camera's optical centre to the point **X**.



For different world points **X**, the epipolar plane rotates about the baseline. All epipolar lines intersect at the epipole.

Epipolar geometry

The epipolar line constrains the search for correspondence from a region to a line. If a point feature $\mathbf{x}$ is observed in one image, then its location $\mathbf{x}'$ in the other image must lie on the epipolar line.



We can derive a mathematical expression for the epipolar line. The two camera-centered coordinate systems are related by a rotation R and translation $\mathbf{T}$:

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

Taking the vector product with $\mathbf{T}$, we obtain

$$\begin{aligned} \mathbf{T} \times \mathbf{X}'_c &= \mathbf{T} \times \mathrm{R}\mathbf{X}_c + \mathbf{T} \times \mathbf{T} \\ \Leftrightarrow \mathbf{T} \times \mathbf{X}'_c &= \mathbf{T} \times \mathrm{R}\mathbf{X}_c \end{aligned}$$

The essential matrix

Taking the scalar product with $\mathbf{X}'_c$, we obtain

$$\begin{aligned} \mathbf{X}'_c.(\mathbf{T} \times \mathbf{X}'_c) &= \mathbf{X}'_c.(\mathbf{T} \times \mathrm{R}\mathbf{X}_c) \\ \Leftrightarrow \mathbf{X}'_c.(\mathbf{T} \times \mathrm{R}\mathbf{X}_c) &= 0 \end{aligned} \quad (4)$$

Recall that a vector product can be expressed as a matrix multiplication:

$$\begin{aligned} \mathbf{T} \times \mathbf{X}_c &= \mathrm{T}_\times \mathbf{X}_c \\ \text{where } \mathrm{T}_\times &= \begin{bmatrix} 0 & -T_z & T_y \\ T_z & 0 & -T_x \\ -T_y & T_x & 0 \end{bmatrix} \end{aligned}$$

So equation (4) can be rewritten as

$$\begin{aligned} \mathbf{X}'_c.(\mathrm{T}_\times \mathrm{R}\mathbf{X}_c) &= 0 \\ \Leftrightarrow \mathbf{X}'^T_c \mathrm{E}\mathbf{X}_c &= 0\,, \quad \text{where } \mathrm{E} = \mathrm{T}_\times \mathrm{R} \end{aligned}$$

E is a 3×3 matrix known as the **essential matrix**. The constraint also holds for rays $\mathbf{p}$, which are parallel to the camera-centered position vectors $\mathbf{X}_c$:

$$\mathbf{p}'^T \mathrm{E}\mathbf{p} = 0 \quad (5)$$

This is the epipolar constraint. If we observe a point $\mathbf{p}$ in one image, then its position $\mathbf{p}'$ in the other image is constrained to lie on the line defined by (5).

Essential matrix: example

Let's calculate the essential matrix for the parallel camera configuration we examined in the triangulation example:

$$\mathrm{R} = \mathrm{I}\,, \quad \mathbf{T} = \begin{bmatrix} -d \\ 0 \\ 0 \end{bmatrix}\,, \quad \mathrm{E} = \mathrm{T}_{\times}\mathrm{R} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & d \\ 0 & -d & 0 \end{bmatrix}$$

The epipolar constraint $\mathbf{p}'^T \mathrm{E}\mathbf{p} = 0$ is therefore

$$\begin{bmatrix} x' & y' & f \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & d \\ 0 & -d & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ f \end{bmatrix} = 0$$

$$\Leftrightarrow \begin{bmatrix} x' & y' & f \end{bmatrix} \begin{bmatrix} 0 \\ df \\ -dy \end{bmatrix} = 0$$

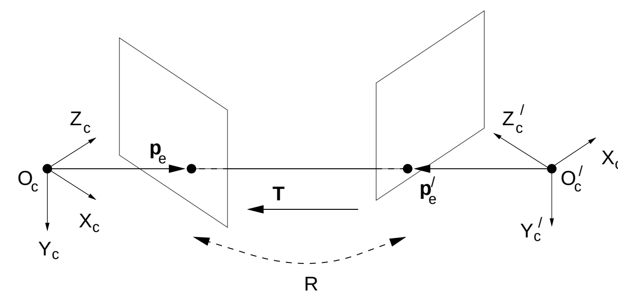
$$\Leftrightarrow y = y'$$

Hence the image of any point $\mathbf{X}$ must lie on the same horizontal line in each image plane.

For parallel cameras, the epipolar lines are parallel, and the epipole is at infinity. This is what we'd expect: neither camera can "see" the optical centre of the other camera.

The essential matrix

The essential matrix can also be used to find the locations of the epipoles.



Referring to the figure, the position of the left camera's epipole is $\mathbf{p}_\mathrm{e}$ in the left camera's coordinate system and $\lambda\mathbf{T}$ in the right camera's coordinate system. Relating the coordinate systems, we obtain

$$\lambda\mathbf{T} = \mathrm{R}\mathbf{p}_\mathrm{e} + \mathbf{T}$$

Taking the vector product with $\mathbf{T}$, we obtain

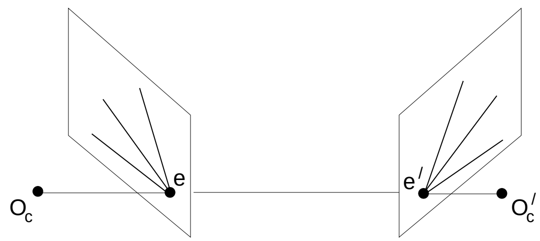
$$\mathbf{0} = \mathbf{T} \times \mathrm{R}\mathbf{p}_\mathrm{e}$$

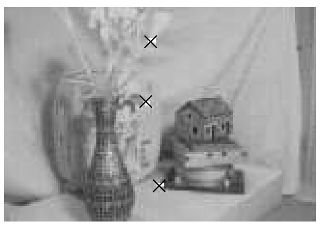
$$\Leftrightarrow \mathrm{E}\mathbf{p}_\mathrm{e} = \mathbf{0}$$

So the location of the epipole in the left image lies in the null space of E. It follows that E is non-invertible (det E = 0) and is therefore of maximum rank 2. The corresponding result for the other epipole is $\mathrm{E}^T\mathbf{p}'_\mathrm{e} = \mathbf{0}$.

Epipolar geometry examples

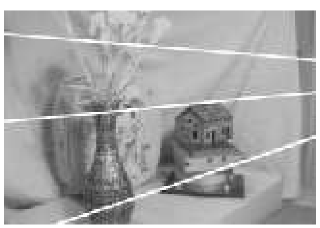
Converging cameras





3 corner features
in left image

Epipolar lines
in right image

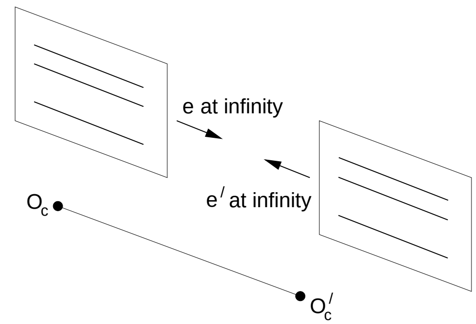


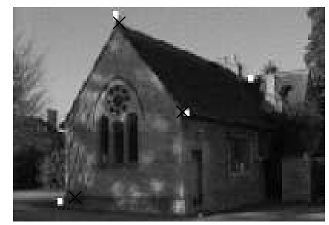
Epipolar lines
in left image

3 corner features
in right image

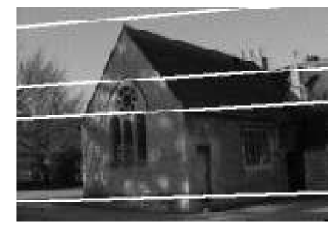
Epipolar geometry examples

Near parallel cameras

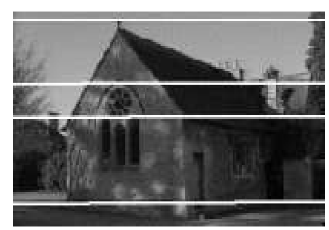




3 corner features
in left image



Epipolar lines
in right image



Epipolar lines
in left image

3 corner features
in right image

From rays to pixels

Up until now we have been assuming calibrated cameras, so we can go from pixel coordinates $\mathbf{w}$ to rays $\mathbf{p}$. But what if we do not know the calibration?

We have seen how pixel coordinates and image plane coordinates are related by the CCD calibration matrix:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} k_u & 0 & u_0 \\ 0 & k_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

We can modify this to derive a relationship between pixel coordinates and rays:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} k_u & 0 & u_0/f \\ 0 & k_v & v_0/f \\ 0 & 0 & 1/f \end{bmatrix} \begin{bmatrix} x \\ y \\ f \end{bmatrix}$$

If we define the matrix K as follows:

$$\mathrm{K} = \begin{bmatrix} fk_u & 0 & u_0 \\ 0 & fk_v & v_0 \\ 0 & 0 & 1 \end{bmatrix}$$

then we can write

$$\tilde{\mathbf{w}} = \mathrm{K}\mathbf{p}$$

The fundamental matrix

The epipolar constraint becomes

$$\begin{aligned} \mathbf{p}'^T \mathrm{E}\mathbf{p} &= 0 \\ \Leftrightarrow \tilde{\mathbf{w}}'^T \mathrm{K}'^{-T} \mathrm{E} \mathrm{K}^{-1} \tilde{\mathbf{w}} &= 0 \\ \Leftrightarrow \tilde{\mathbf{w}}'^T \mathrm{F} \tilde{\mathbf{w}} &= 0 \ , \quad \text{where } \mathrm{F} = \mathrm{K}'^{-T} \mathrm{E} \mathrm{K}^{-1} \end{aligned}$$

F is a 3×3 matrix known as the **fundamental matrix**.

For any given point $\tilde{\mathbf{w}}$ in the left image, if we know F we can derive an epipolar constraint on the point's location $\tilde{\mathbf{w}}'$ in the right image.

The locations of the epipoles $\tilde{\mathbf{w}}_e$ and $\tilde{\mathbf{w}}'_e$ (in pixels) are given by

$$\begin{aligned} \mathrm{E}\mathbf{p}_e &= \mathbf{0} \\ \Leftrightarrow \mathrm{E}\mathrm{K}^{-1}\tilde{\mathbf{w}}_e &= \mathbf{0} \\ \Leftrightarrow \mathrm{K}'^{-T}\mathrm{E}\mathrm{K}^{-1}\tilde{\mathbf{w}}_e &= \mathbf{0} \\ \Leftrightarrow \mathrm{F}\tilde{\mathbf{w}}_e &= \mathbf{0} \ \text{ and likewise } \ \mathrm{F}^T\tilde{\mathbf{w}}'_e = \mathbf{0} \end{aligned}$$

At first sight, F appears to have 9 degrees of freedom. However, its overall scale does not matter (so we could set f_{33} to 1) and, as with E, it has zero determinant (maximum rank 2). So F has only 7 degrees of freedom.

Computing F from correspondences

Since the cameras are uncalibrated, we do not know E, K or K′ and so we do not know F a-priori. However, we can estimate F from point correspondences.

Each point correspondence $\tilde{\mathbf{w}} \leftrightarrow \tilde{\mathbf{w}}'$ generates one constraint on F:

$$\begin{bmatrix} u' & v' & 1 \end{bmatrix} \begin{bmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{bmatrix} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = 0$$

n of these constraints can be arranged in the following form:

$$\begin{bmatrix} u'_1 u_1 & u'_1 v_1 & u'_1 & v'_1 u_1 & v'_1 v_1 & v'_1 & u_1 & v_1 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ u'_n u_n & u'_n v_n & u'_1 & v'_n u_n & v'_n v_n & v'_n & u_n & v_n & 1 \end{bmatrix} \begin{bmatrix} f_{11} \\ f_{12} \\ f_{13} \\ f_{21} \\ f_{22} \\ f_{23} \\ f_{31} \\ f_{32} \\ f_{33} \end{bmatrix} = \mathbf{0}$$

Computing F from correspondences

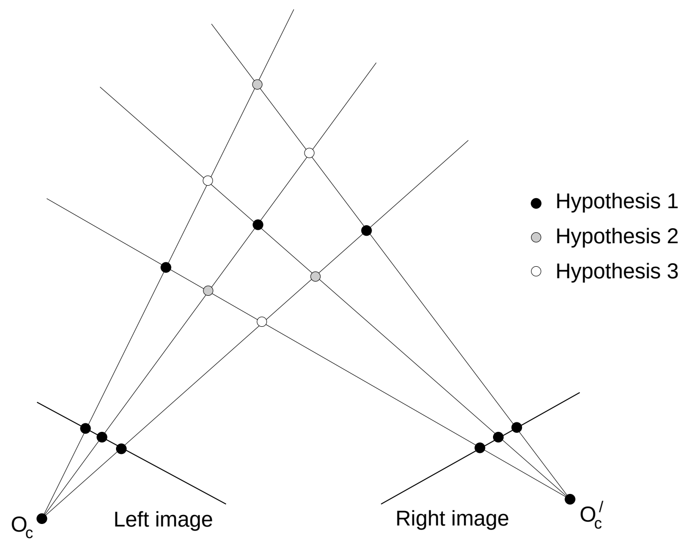
Given 8 or more perfect correspondences (image points in *general* position, no noise), F can be determined uniquely up to scale. In practice, we may have more than 8 correspondences and the image measurements will be noisy. The system of equations is then solved by least squares.

Note that we have not attempted to enforce the constraint that det F = 0. If the 8 image points are noisy, then we will find that our estimate of F does *not* have zero determinant and the epipolar lines do not meet at a point. Nonlinear techniques exist to estimate F from 7 point correspondences, enforcing the rank 2 constraint.

Given F, we can establish correspondences with *relative* ease. If we know the intrinsic camera parameters K, we can also find the essential matrix, decompose E into $T_\times$ and R, and recover metric structure by triangulation. Without K we can only recover structure up to a 3D projective transformation, which is not ideal but nevertheless useful for object recognition.

The correspondence problem

Even with the epipolar constraint, establishing correspondences between points in the left and right image is not trivial. Comparing image patches by correlation is unreliable since the grey levels are viewpoint dependent.



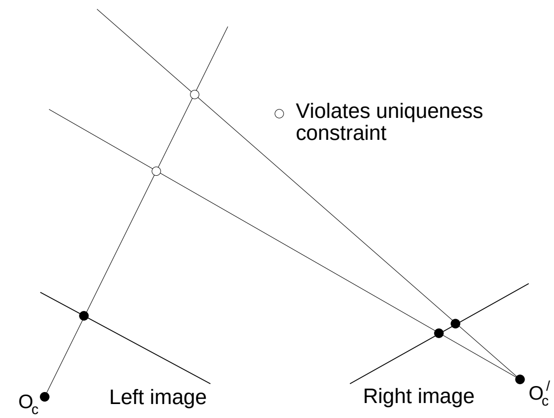
In the illustration, we are trying to match three corners in the left image with three corners in the right image. We have three hypotheses, all of which satisfy the epipolar constraint. How can we discover which hypothesis is correct?

The correspondence problem

The correspondence problem is very difficult to solve, but we can make some progress by identifying more constraints.

Uniqueness

The most obvious constraint is uniqueness. For opaque objects, each point in the left image has at most one match in the right image.

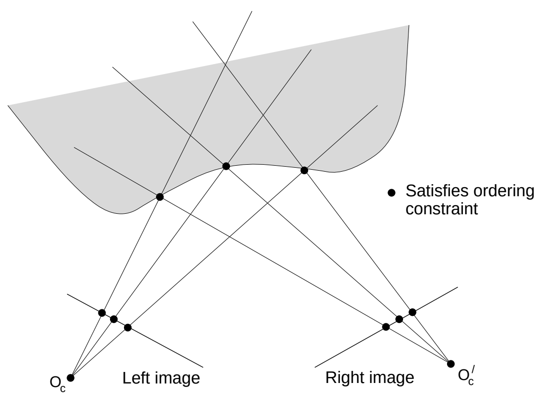


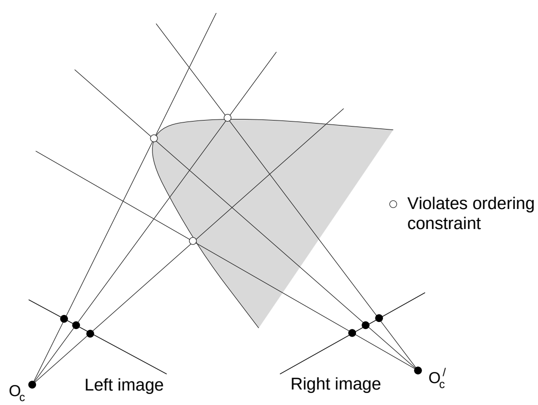
For transparent objects, we cannot rely on the uniqueness constraint. Two features may be visible in the right image but instantaneously aligned in the left image.

The correspondence problem

Ordering

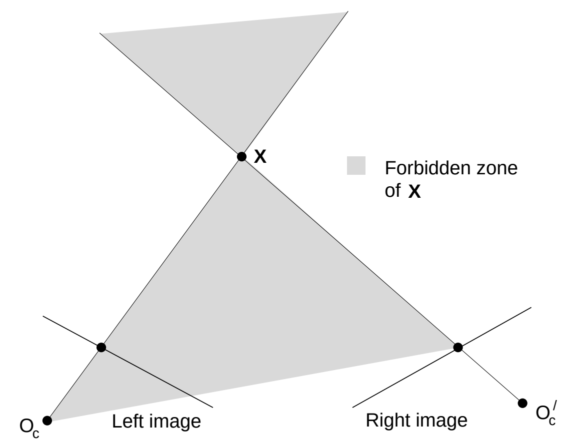
Corresponding points lying on the surface of an opaque object will be ordered identically in left and right images.





The correspondence problem

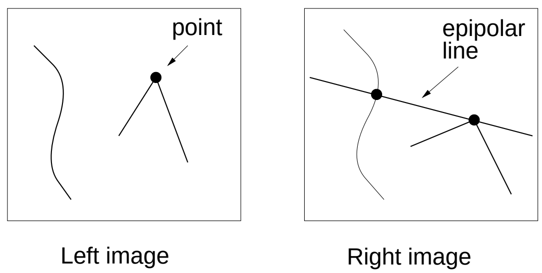
The ordering constraint will not necessarily hold if the points do not lie on the surface of the same opaque object. Given point **X** observed in both images, any point lying in **X**'s "forbidden zone" will violate the ordering constraint.



Figural continuity

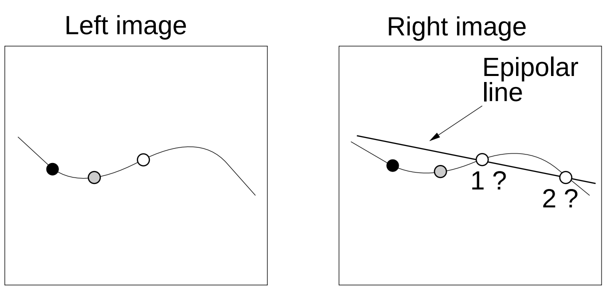
When distinguished points lie on image contours, we can sometimes use figural continuity as a matching constraint. In the following example, the point in the left image must match the point towards the right of the right image.

The correspondence problem



Disparity gradient

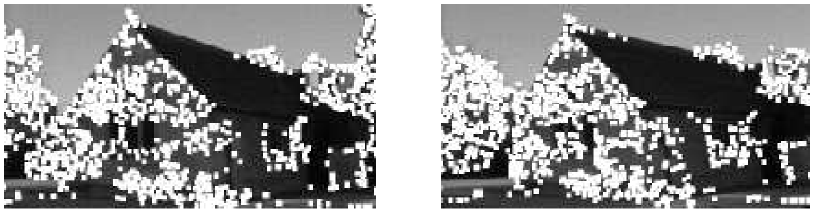
If surfaces are smooth, then disparities (differences in location between points in the left and right images) must be locally smooth. So, away from occluding boundaries, a further constraint comes from imposing a limit on the allowable spatial derivatives of disparity.



Given matches ● and ◎, point ○ in the left image must match point 1 in the right image. Point 2 would exceed the disparity gradient limit.

Finding correspondences

Here is the outline of an algorithm for finding correspondences between corners (typically 200–300 per image).



1. Unguided matching.

Seed matches

Use local search and normalized cross-correlation to obtain a small number of seed matches.

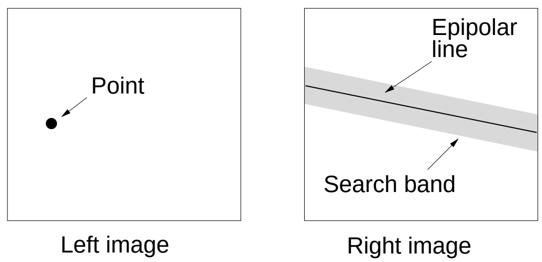
2. Compute epipolar geometry. Use seed matches and robust regression to compute F.

Find an F which is consistent with many of the seed matches — reject the rest as outliers.

Matches consistent with F

Finding correspondences

3. Guided matching. Now that we know F, the search for matches can be restricted to a narrow band around epipolar lines.



Using the epipolar and other constraints (ordering, grey level correlation, etc.), we obtain a large number of matches.

With calibrated cameras, we can now recover the structure of the building by triangulation. We only recover structure at the detected corners: to reconstruct more of the scene, we could try **intensity-based** matching between corners.

Affine stereo

Recall that when depth variations in the scene are small compared with the viewing distance, an affine camera is appropriate:

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

The affine camera can be calibrated by observing four points in space.

With two calibrated affine cameras, it is straightforward to triangulate to recover structure:

$$\begin{bmatrix} u \\ v \\ u' \\ v' \end{bmatrix} = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p'_{11} & p'_{12} & p'_{13} & p'_{14} \\ p'_{21} & p'_{22} & p'_{23} & p'_{24} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (6)$$

Each point observed in left and right images gives us 4 equations in the 3 unknowns (X, Y, Z). These can be solved using least squares.

But what about an epipolar constraint to help with the correspondence problem?

Affine stereo

Assume (without loss of generality), that the left camera is aligned with the world coordinate system: this will simplify the algebra considerably. It is straightforward to show (by inspection of the weak perspective camera matrix) that the left camera matrix reduces to

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} p_{11} & 0 & 0 & p_{14} \\ 0 & p_{22} & 0 & p_{24} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

We can now easily eliminate X and Y from the equations for u' and v' in (6):

$$\begin{aligned} u' &= p'_{11}\frac{(u-p_{14})}{p_{11}} + p'_{12}\frac{(v-p_{24})}{p_{22}} + p'_{13}Z + p'_{14} \\ v' &= p'_{21}\frac{(u-p_{14})}{p_{11}} + p'_{22}\frac{(v-p_{24})}{p_{22}} + p'_{23}Z + p'_{24} \end{aligned}$$

Rewriting these equations, we obtain

$$\begin{bmatrix} u' \\ v' \end{bmatrix} = \begin{bmatrix} p'_{11}\frac{(u-p_{14})}{p_{11}} + p'_{12}\frac{(v-p_{24})}{p_{22}} + p'_{14} \\ p'_{21}\frac{(u-p_{14})}{p_{11}} + p'_{22}\frac{(v-p_{24})}{p_{22}} + p'_{24} \end{bmatrix} + Z \begin{bmatrix} p'_{13} \\ p'_{23} \end{bmatrix}$$

Affine stereo

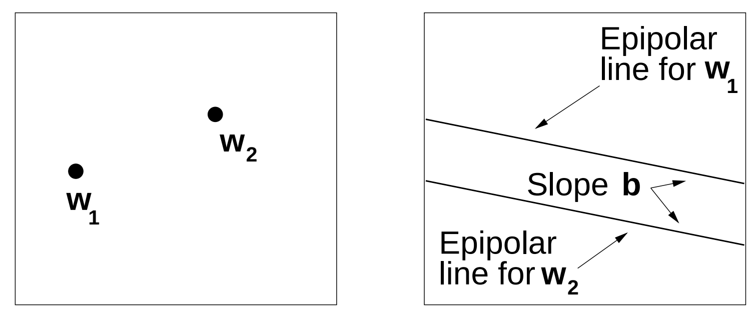
We can rewrite the preceding equation as

$$\mathbf{w}' = \mathbf{a} + Z\mathbf{b} \tag{7}$$

This is one form of the epipolar constraint for affine stereo.

Given calibrated cameras and a point $\mathbf{w}$ in the left image, we do not know Z but we do know $\mathbf{a}$ and $\mathbf{b}$. Thus, the corresponding point $\mathbf{w}'$ must lie on the epipolar line in the right image described by (7).

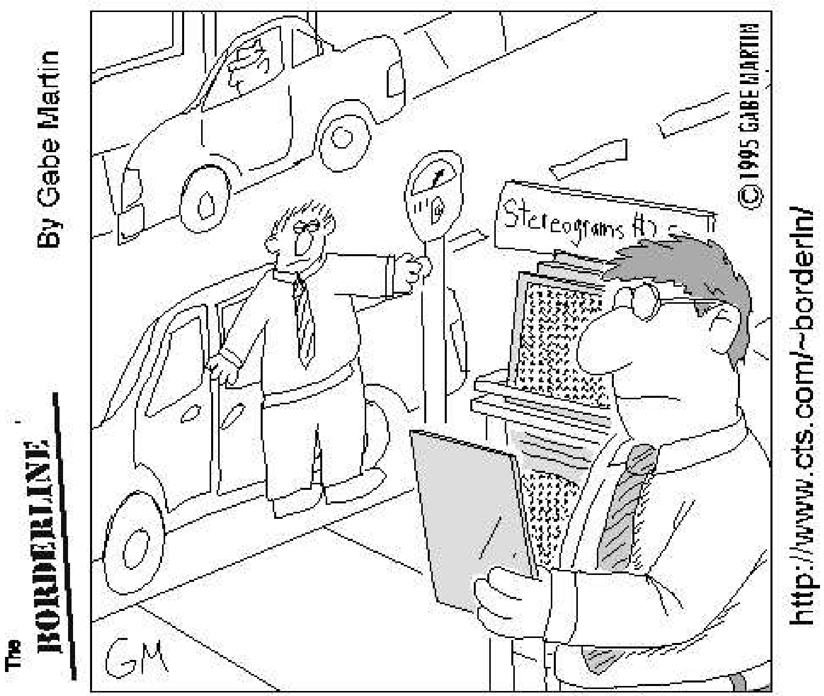
Since $\mathbf{b}$ is independent of $\mathbf{w}$, it follows that all epipolar lines are parallel under affine stereo.



Left image Right image

Random dot stereograms

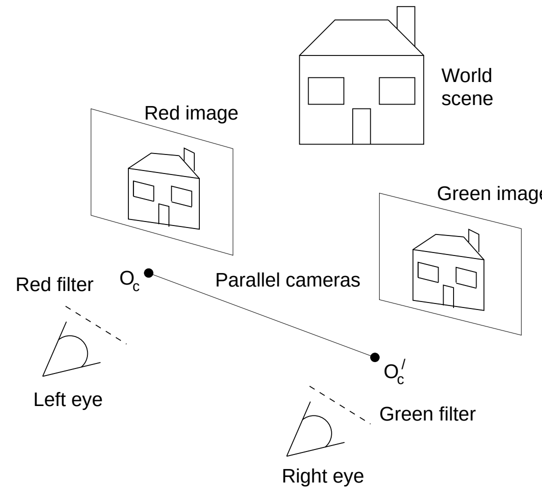
We won't be looking at random dot stereograms.



Frank finally got the hang of seeing the 3-D pictures, but the problem was now the rest of the world looked 2-D.

Case study: 3D stereograms

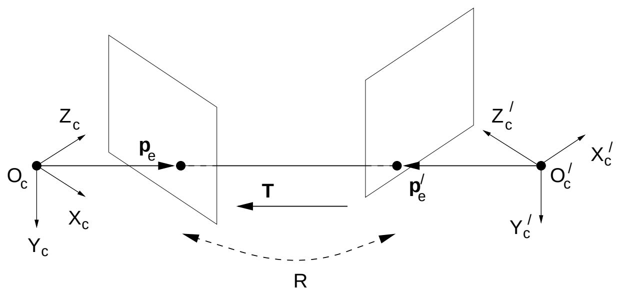
3D stereograms give the illusion of depth, even though the viewer is in fact looking at a planar image. They can take several forms, including random dot stereograms and red/green stereograms. Here we focus on the latter.



To see an RG stereogram, the viewer is presented with a different image of the scene in each eye. To achieve this, the two images are printed in red and green and the viewer wears appropriate filter glasses. If the two images are acquired by parallel cameras, mimicing the human visual system, the brain is able to fuse the two images together and perceive depth.

Case study: 3D stereograms

Traditionally, RG stereograms are created using highly calibrated stereoscopic cameras, ensuring parallel image planes. However, epipolar geometry can be exploited to create an RG stereogram from *any* stereo pair of images.



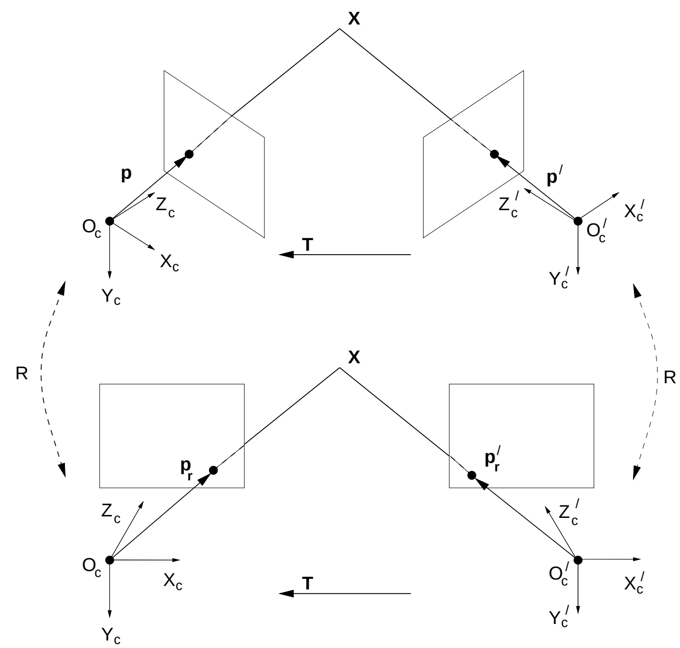
The idea is to take two pictures with the same camera and estimate the fundamental matrix F by searching for corresponding features in the left and right images. Assuming we know the camera calibration matrix K, we can recover the essential matrix:

$$\mathrm{E} = \mathrm{K}^T \mathrm{F} \mathrm{K}$$

We can then find the epipoles $\mathbf{p}_\mathrm{e}$ and $\mathbf{p}'_\mathrm{e}$ from the null space of E and E^T, and hence the direction of translation $\hat{\mathbf{T}}$.

Case study: 3D stereograms

The next stage is to **rectify** the images: that is, warp the images to recover what they would have looked like had the cameras been parallel.



It is relatively straightforward to find the rotation matrix R which aligns the X_c-axis of each camera's coordinate system with $\mathbf{T}$. We can now relate rays $\mathbf{p}$ in the raw images to rays $\mathbf{p}_\mathrm{r}$ in the rectified images:

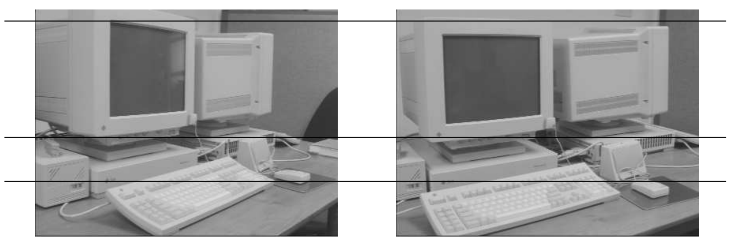
$$\mathbf{p} = \mathrm{R}\mathbf{p}_\mathrm{r}$$

Case study: 3D stereograms

Next, using the camera calibration, we relate pixels in the raw ($\mathbf{w}$) and rectified ($\mathbf{w}_r$) images:

$$
\begin{aligned}
\mathbf{p} &= \mathrm{R}\mathbf{p}_r \\
\mathrm{K}^{-1}\tilde{\mathbf{w}} &= \mathrm{R}\mathrm{K}^{-1}\tilde{\mathbf{w}}_r \\
\Leftrightarrow \tilde{\mathbf{w}} &= \mathrm{K}\mathrm{R}\mathrm{K}^{-1}\tilde{\mathbf{w}}_r \qquad (8)
\end{aligned}
$$

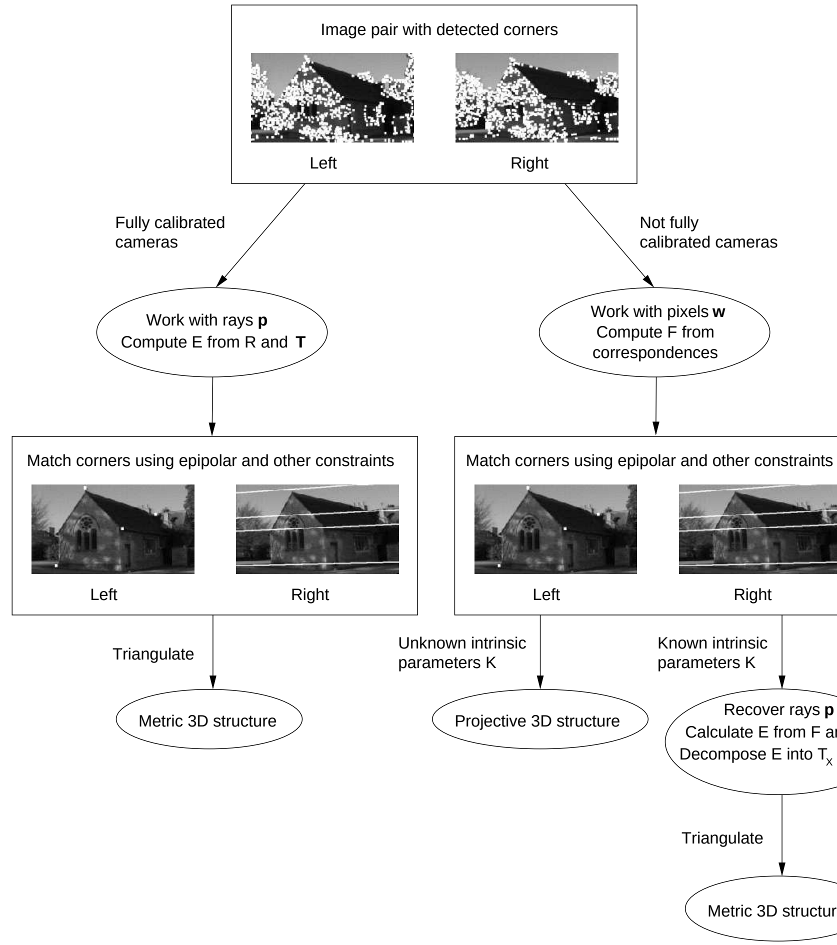
The warped images are created using (8) to find the grey level $I(\mathbf{w})$ associated with each pixel $\mathbf{w}_r$ in the rectified image. Finally, a similar warping is applied to rectify any relative rotation about the cameras' X_c-axes.



Raw images - epipolar lines not horizontal

Rectified images - epipolar lines horizontal. Can be fused if superimposed in red and green and viewed through filtered glasses.

Summary



Bibliography

The figures on pages 12, 13, 23 and 24 were reproduced (with thanks) from Andrew Zisserman's vision course notes. The 3D stereogram case study was inspired by Paul Smith's fourth year project at Cambridge (1995–96). The following publications make good further reading.

Epipolar geometry and scene reconstruction

H. C. Longuet-Higgins. A computer algorithm for reconstructing a scene from two projections. *Nature*, 293:133–135, 1981.

O. D. Faugeras and G. Toscanini. The calibration problem for stereo. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, 15–20, 1986.

O. D. Faugeras. What can be seen in three dimensions with an uncalibrated stereo rig? In *Proceedings of the 2nd European Conference on Computer Vision*, 563–578, 1992.

Stereo matching

S. B. Pollard, J. E. W. Mayhew and J. P. Frisby. PMF: a stereo correspondence algorithm using a disparity gradient. *Perception*, 14:449–470, 1985.

3D stereograms

P. Smith, *3D stereograms*. Fourth-year project in Group E, Cambridge University Engineering Department, May 1996.