IIB CRIBS
2003

4F12

Engineering Tripos Part IIB FOURTH YEAR

Module 4F12: Computer Vision and Robotics

Solutions to 2003 Tripos Paper

1. **Feature detection**

(a) $I(x, y)$ is a function of many variables, including the position of the camera; the properties of the lens and the CCD; the shape of the structures in the scene; the nature and distribution of light sources; and the reflectance properties of the visible surfaces.

Edge detection is commonly used in the first stage of many computer vision applications, since edges provide a compact representation of image structure and are invariant to illumination effects. Compared with raw images, edges offer significant data reduction while preserving much of the image's useful information content (it is possible to recognise many structures in a line drawing of a scene). In contrast, most of the discarded information is not useful for discovering scene structure and motion. 6

(b) The Marr–Hildreth operator convolves the image with a discrete version of the Laplacian of a Gaussian and then localises edges at the resulting zero-crossings. The Canny operator is a directional edge finder. It first localises the orientation of the edge by computing

$$\hat{\mathbf{n}} = \frac{\nabla\left(G_\sigma(x,y) * I(x,y)\right)}{\left|\nabla\left(G_\sigma(x,y) * I(x,y)\right)\right|}$$

and then searches for a local maximum of $|\nabla(G_\sigma * I)|$ in the direction $\hat{\mathbf{n}}$. This is equivalent to finding zero-crossings in the directional second derivative of $(G_\sigma * I)$ in the direction $\hat{\mathbf{n}}$.

The principle advantage of the Marr-Hildreth operator is computational simplicity and efficiency: edge detection requires only a single convolution and the detection of zero-crossings. Conversely, the Canny operator requires an additional, costly search for a local maximum normal to the gradient direction which requires the storage of both gradient magnitude and gradient direction.

The advantage of the Canny operator is that it localises the edges correctly and has some robustness to noise. Any differential operator amplifies noise. The Canny operator computes only first derivatives and then searches for a local maximum (which is equivalent to a zero-crossing of the second derivative) normal to the gradient. The Marr-Hildreth operator computes second derivatives both along and normal to the edge. Computation of the second derivative along the edge emphasizes noise in that direction while serving no purpose in edge detection. This latter noise leads to incorrect edge localisation. 6

(c) The rate of change of intensity I in the direction $\mathbf{n}$ is found by taking the scalar product of ∇I and $\hat{\mathbf{n}}$:

$$I_n \equiv \nabla I(x,y).\hat{\mathbf{n}} \;\Rightarrow\; I_n^2 = \frac{\mathbf{n}^T\,\nabla I\,\nabla I^T\,\mathbf{n}}{\mathbf{n}^T\mathbf{n}} = \frac{\mathbf{n}^T \begin{bmatrix} I_x^2 & I_xI_y \\ I_xI_y & I_y^2 \end{bmatrix} \mathbf{n}}{\mathbf{n}^T\mathbf{n}}$$

where $I_x \equiv \partial I/\partial x$, etc. [20%]

We smooth I_n^2 by convolution with a Gaussian kernel:

$$C_n(x,y) = G_\sigma(x,y) * I_n^2 = \frac{\mathbf{n}^T \begin{bmatrix} \langle I_x^2 \rangle & \langle I_x I_y \rangle \\ \langle I_x I_y \rangle & \langle I_y^2 \rangle \end{bmatrix} \mathbf{n}}{\mathbf{n}^T\mathbf{n}}$$

where $\langle\ \rangle$ is the smoothed value. The smoothed change in intensity in direction $\mathbf{n}$ is therefore given by

$$C_n(x,y) = \frac{\mathbf{n}^T \mathrm{A} \mathbf{n}}{\mathbf{n}^T\mathbf{n}}$$

where A is the 2×2 matrix

$$\begin{bmatrix} \langle I_x^2 \rangle & \langle I_x I_y \rangle \\ \langle I_x I_y \rangle & \langle I_y^2 \rangle \end{bmatrix}$$

Elementary eigenvector theory tells us that

$$\lambda_1 \leq C_n(x,y) \leq \lambda_2$$

where λ_1 and λ_2 are the eigenvalues of A. So, if we try every possible orientation $\mathbf{n}$, the maximum change in intensity we will find is λ_2, and the minimum value is λ_1.

We can detect a corner by looking at the eigenvectors of A. For a (corner) λ_1 and λ_2 both large. It is necessary to calculate A at every pixel and mark corners where the quantity $\lambda_1\lambda_2 - \kappa(\lambda_1 + \lambda_2)^2$ exceeds some threshold ($\kappa \approx 0.04$ makes the detector a little "edge-phobic"). Note that $\det \mathrm{A} = \lambda_1\lambda_2$ and trace $\mathrm{A} = \lambda_1 + \lambda_2$, so the required eigenvalue properties can be obtained directly from the elements of A. 8

2. **Projection matrices under perspective and weak perspective.**

(a) (i) Parallel planes meet at lines in the image, often referred to as horizon lines. To prove this, consider a plane in 3D space defined as follows:

$$\mathbf{X}_c.\mathbf{n} = d$$

where $\mathbf{n} = (n_x, n_y, n_z)$ is the normal to the plane. We can analyse horizon lines by writing the perspective projection in the following form:

$$\begin{bmatrix} x \\ y \\ f \end{bmatrix} = \frac{f\mathbf{X}_c}{Z_c}$$

Taking the scalar product of both sides with $\mathbf{n}$ gives:

$$\begin{bmatrix} x \\ y \\ f \end{bmatrix} \cdot \begin{bmatrix} n_x \\ n_y \\ n_z \end{bmatrix} = \frac{f\mathbf{X}_c.\mathbf{n}}{Z_c} = \frac{fd}{Z_c}$$

6

As $Z_c \to \infty$ we move away from the camera and we find

$$\begin{bmatrix} x \\ y \\ f \end{bmatrix} \cdot \begin{bmatrix} n_x \\ n_y \\ n_z \end{bmatrix} = 0$$

Thus the equation of the horizon line is ✓

$$n_x x + n_y y + f n_z = 0$$

which depends only on the orientation of the plane, and not its position. Thus a set of parallel planes meet at a horizon line in the image. [30%]

(b) The overall imaging process, from world $\tilde{\mathbf{X}}$ to image $\tilde{\mathbf{w}}$, can be written as a single matrix multiplication in homogeneous coordinates:

$$\tilde{\mathbf{w}} = \begin{bmatrix} k_u & 0 & u_0 \\ 0 & k_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \left[\begin{array}{ccc|c} & \mathrm{R} & & \mathbf{T} \\ \hline 0 & 0 & 0 & 1 \end{array} \right] \tilde{\mathbf{X}} = \mathrm{P}\,\tilde{\mathbf{X}}, \text{ say}$$

P is a 3 × 4 matrix, so the process can be expressed as

$$\begin{bmatrix} su \\ sv \\ s \end{bmatrix} = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

P can be estimated by observing the images of known 3D points. Each point we observe gives us a pair of equations:

$$\begin{aligned} u &= \frac{su}{s} = \frac{p_{11}X + p_{12}Y + p_{13}Z + p_{14}}{p_{31}X + p_{32}Y + p_{33}Z + p_{34}} \\ v &= \frac{sv}{s} = \frac{p_{21}X + p_{22}Y + p_{23}Z + p_{24}}{p_{31}X + p_{32}Y + p_{33}Z + p_{34}} \end{aligned}$$

Since we are observing a known scene, we know X, Y, and Z, and we observe the pixel coordinates u and v in the image. So we have two linear equations in the unknown camera parameters. Since there are 11 unknowns (the overall scale of P does not matter), we need to observe at least 6 points, in a non-degenerate configuration, to calibrate the camera. In practice, we would use more than 6 points to mitigate the effects of measurement noise.

The equations can be solved using orthogonal least squares. First, we write the equations in matrix form:

$$\mathrm{A}\mathbf{p} = \mathbf{0}$$

where $\mathbf{p}$ is the 12 × 1 vector of unknowns (the twelve elements of P), A is the $2n \times 12$ matrix of coefficients and n is the number of observed calibration points. The orthogonal least squares solution corresponds to the eigenvector of $\mathrm{A}^T\mathrm{A}$ with the smallest corresponding eigenvalue.

The linear solution is, however, only approximate, since we have not taken into account the special structure of P. Ideally, the linear solution should be used as the starting point for nonlinear optimization, finding the parameters of the rigid body transformation, perspective projection and CCD mapping that minimize the errors between measured image points $(u_{i,} v_i)$ and projected (or modeled) image positions $(\hat{u}_i, \hat{v}_i)$:

$$\min_{\mathbf{P}} \sum_i ((u_i - \hat{u}_i)^2 + (v_i - \hat{v}_i)^2)$$

Given the projective camera matrix, we can attempt to recover the intrinsic and extrinsic parameters using QR decomposition. Writing

$$\begin{aligned} \mathrm{P} &= \begin{bmatrix} fk_u & 0 & u_0 & 0 \\ 0 & fk_v & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \left[\begin{array}{ccc|c} & \mathrm{R} & & \mathbf{T} \\ & & & \\ \hline 0 & 0 & 0 & 1 \end{array}\right] = \begin{bmatrix} fk_u & 0 & u_0 \\ 0 & fk_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \left[\begin{array}{c|c} \mathrm{R} & \mathbf{T} \end{array}\right] \\ &= \mathrm{C}\left[\, \mathrm{R} \,|\, \mathbf{T} \,\right] = \left[\, \mathrm{CR} \,|\, \mathrm{CT} \,\right] \end{aligned}$$

it is apparent that we need to decompose the left 3×3 sub-matrix of P into an upper triangular matrix C and an orthogonal (rotation) matrix R. This can be achieved using QR decomposition. $\mathbf{T}$ can then be recovered using 10

$$\mathbf{T} = \mathrm{C}^{-1} \left[\, p_{14} \;\; p_{24} \;\; p_{34} \,\right]^T$$

It is not possible to decouple the focal length f from the pixel scale factors k_u and k_v. [60%]

(c) Under weak perspective projection, we assume that all points lie at approximately the same depth Z_A from the camera. This allows the projection to be re-written as follows:

$$\begin{bmatrix} su_A \\ sv_A \\ s \end{bmatrix} = \begin{bmatrix} k_u f & 0 & 0 & u_0 Z_A \\ 0 & k_v f & 0 & v_0 Z_A \\ 0 & 0 & 0 & Z_A \end{bmatrix} \begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix}$$

Weak perspective is a good approximation when the depth range of objects in the scene is small compared with the viewing distance. A good rule of thumb is that the viewing distance should be at least ten times the depth range in the scene.

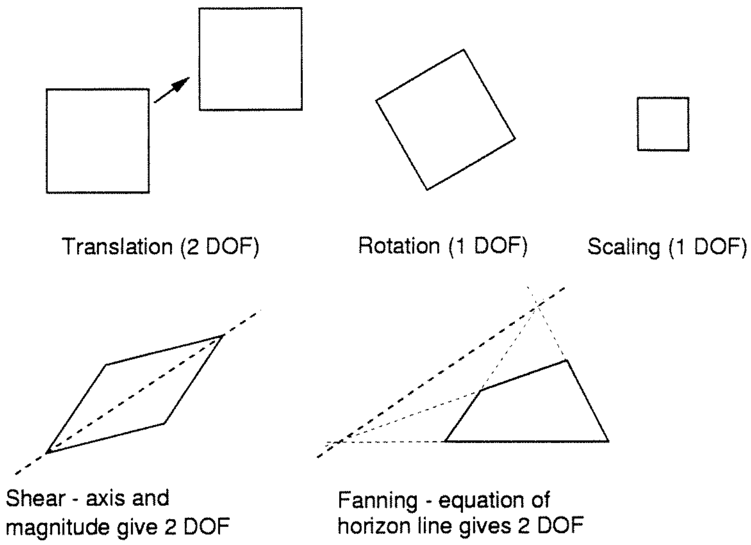
The main advantage of the weak perspective model is that it is easier to calibrate than the full perspective model. The calibration requires fewer points with known world position, and, since the model is linear, the calibration process is also better conditioned (less sensitive to noise) than the nonlinear full perspective calibration. 4

3. **Planar projective transformations**

(a)

(i) Since the transformation operates on homogeneous coordinates, the overall scale of the transformation matrix does not matter and we could, for instance, set t_{33} to 1. The transformation therefore has 8 degrees of freedom. 2

(ii) The image of a square could take any of the forms shown on the next page.


Translation (2 DOF)
Rotation (1 DOF)
Scaling (1 DOF)
Shear - axis and magnitude give 2 DOF
Fanning - equation of horizon line gives 2 DOF

[20%] 6

(iii) The equation of the conic in the first image is $\tilde{\mathbf{w}}^T \mathrm{C}\, \tilde{\mathbf{w}} = 0$, where

$$\mathrm{C} \equiv \begin{bmatrix} a & b/2 & d/2 \\ b/2 & c & e/2 \\ d/2 & e/2 & f \end{bmatrix}$$

Using again the relationship $\tilde{\mathbf{w}} = \mathrm{T}^{-1}\, \tilde{\mathbf{w}}'$, we find the equation of the corresponding conic in the second image as follows:

$$(\mathrm{T}^{-1}\, \tilde{\mathbf{w}}')^T \mathrm{C}\, \mathrm{T}^{-1}\, \tilde{\mathbf{w}}' = 0 \quad \Leftrightarrow \quad \tilde{\mathbf{w}}'^T\, \mathrm{T}^{-T} \mathrm{C}\, \mathrm{T}^{-1}\, \tilde{\mathbf{w}}' = 0$$

Alternatively, the conic in the second image can be expressed simply as $\mathrm{C}' = \mathrm{T}^{-T} \mathrm{C}\, \mathrm{T}^{-1}$. [20%] 4

(b) Assume, without loss of generality, that before the camera is rotated, the camera is aligned with the world coordinate system and hence

$$\tilde{\mathbf{w}} = \left[\, \mathrm{I} \mid \mathbf{O} \,\right] \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \mathbf{X}$$

where K is the 3×3 matrix of intrinsic camera parameters:

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

It follows that

$$\mathbf{X} =^{-1} \tilde{\mathbf{w}}$$

After rotating by R about the optical centre, the same world point $\mathbf{X}$ projects to a different image point $\tilde{\mathbf{w}}'$ as follows:

$$\tilde{\mathbf{w}}' = \left[\, \mathrm{R} \mid \mathbf{O} \,\right] \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \mathrm{R} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \mathrm{R}\mathbf{X} = \mathrm{R}^{-1}\tilde{\mathbf{w}} = \mathrm{T}\tilde{\mathbf{w}}$$

4

where T $\equiv$ R^{-1}. Hence the relationship between points in the original image and corresponding points in the second image is a 2D projective transformation. [20%]

1

4. **Stereo vision**

(a) The stereo camera geometry constrains each point feature identified in one image to lie on a corresponding *epipolar line* in the other image. If the cameras are calibrated, then the equation of the epipolar line can be derived from the essential matrix. For uncalibrated cameras, it is possible to estimate the fundamental matrix from point correspondences and derive epipolar lines from the fundamental matrix. Epipolar lines meet at the *epipole*: this is the image of one camera's optical centre in the other camera's image plane. There are two epipoles, one for each image.

4

(b) The essential matrix E describes the epipolar geometry of a stereo rig in terms of rays $\mathbf{p} = [\ x \ \ y \ \ f\ ]^T$, where (x, y) are the metric image plane coordinates of an observed point and f is the camera's focal length.

To derive the essential matrix in terms of R and **T**, we start with the equation relating the two coordinate systems:

$$\begin{aligned}
\mathbf{X}'_\mathbf{C} &= \mathrm{R}\mathbf{X}_\mathbf{C} + \mathbf{T} \Rightarrow \mathbf{T} \times \mathbf{X}'_\mathbf{C} = \mathbf{T} \times \mathrm{R}\mathbf{X}_\mathbf{C} \\
\Rightarrow \mathbf{X}'_\mathbf{C}.(\mathbf{T} \times \mathrm{R}\mathbf{X}_\mathbf{C}) &= 0 \Leftrightarrow \mathbf{X}'_\mathbf{C}.(\mathrm{T}_\times \mathrm{R}\mathbf{X}_\mathbf{C}) = 0\ , \quad \text{where } \mathrm{T}_\times = \begin{bmatrix} 0 & -T_z & T_y \\ T_z & 0 & -T_x \\ -T_y & T_x & 0 \end{bmatrix} \\
\Leftrightarrow \mathbf{p}'^T[\mathrm{T}_\times \mathrm{R}]\mathbf{p} &= 0\ , \quad \text{since rays and camera-centered positions are parallel.}
\end{aligned}$$

The essential matrix is therefore given by E = T$_\times$R.

Epipolar geometry can be expressed in pixel coordinates and the epipolar constraint leads to the fundamental matrix. The essential and fundamental matrices are related by the internal calibration matrices K and of the left and right cameras, where

6

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

f is the focal length, k_u and k_v the pixel scale factors and (u_0, v_0) the point where the optical axis intersects the image plane. F= $'^{-T}$T$_\times$R^{-1}

(c) F can be estimated from point correspondences. Each point correspondence $\tilde{\mathbf{w}} \leftrightarrow \tilde{\mathbf{w}}'$ generates one constraint on F:

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

This is a linear equation in the unknown elements of F. Given eight or more perfect correspondences (image points in *general* position, no noise), F can be determined uniquely up to scale by solving the simultaneous linear equations. In practice, there may be more than eight correspondences and the image measurements will be noisy. The system of equations can then be solved by least squares, or using a robust regression scheme to reject outliers.

The linear technique does not enforce the constraint that det F = 0. If the eight image points are noisy, then the linear estimate of F will *not* necessarily have zero determinant 4
and the epipolar lines will not meet at a point. Nonlinear techniques exist to estimate F from 7 point correspondences, enforcing the rank 2 constraint.

(d) The epipoles lie in the null spaces of F and F^T. So, for the left epipole we have:

$$\mathrm{F}\tilde{\mathbf{w}}_{\mathrm{e}} = \mathbf{0}$$

If F were invertible, we would be able to write

$$\tilde{\mathbf{w}}_{\mathrm{e}} = \mathrm{F}^{-1}\mathbf{0} = \mathbf{0}$$

which is a contradiction. It follows that F is non-invertible and therefore has maximum 4.
rank 2.

(e) With unknown internal parameters structure is recovered up to 3D projective transformation which can be removed by 5 known 3D points or calibration parameters of the 2
cameras. Scale can only be recovered by knowledge of a length.

5. **Applications**

(a) The camera should look down on the ground plane (the floor). The image-to-ground mapping has 8 degrees of freedom:

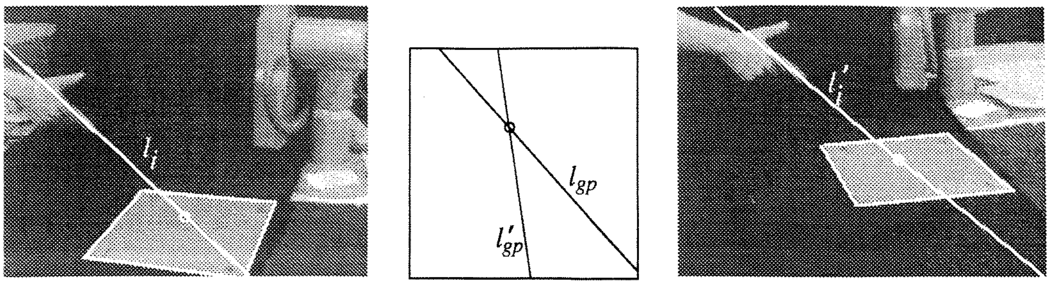
$$\begin{bmatrix} su \\ sv \\ s \end{bmatrix} = \begin{bmatrix} p_{11} & p_{12} & p_{13} \\ p_{21} & p_{22} & p_{23} \\ p_{31} & p_{32} & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix}$$

This can be calibrated using four known points on the floor (and their corresponding image positions), though for greater accuracy more correspondences should be used (with least squares).

Consecutive images can be subtracted from each other (time differencing) to detect moving people. The bottom of the moving region should correspond to the point of contact between the person and the floor. The calibration can then be used to translate this into a world position on the floor.

People can be tracked using cross-correlation of the moving blobs, or using B-spline snakes. Kalman filter is used to integrate noisy measurements.

(c) A single view of a pointing hand (or arm) is ambiguous: the 'piercing point', where the line defined by the hand intersects the screen, cannot be uniquely determined but is constrained to a line, which is the projection of the hand's line in the image (see (a) below).



(a) (b) (c)

With a second camera we obtain a similar constraint in the other image (c). There exists a planar projective transformation (8 degrees of freedom) that maps one image of the screen onto the other. This transformation can be calibrated by observing the four corners of the screen. We exploit this to transform the constraint lines into a common 'canonical' view of the screen, and hence find their intersection (b).

The piercing point can then be projected back into the two images (small circles in (a) and (b)); if the four reference points are known its world coordinates can also be calculated.

The user's arm can be tracted using B-spline snakes (book work, see Handout 2).

(d) Given that the cameras are some distance from the workspace, an affine model 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 left and right cameras can be calibrated by moving the gripper to four predetermined points in 3D space, and tracking its image position using affine B-spline snakes. More points (and least squares) could be used for better accuracy.

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

The user needs to specify the target in each view. The calibration can then be used to determine the world position of the target, and the gripper moved to the right location for a grasping manoeuvre.

Roberto Cipolla
February 2003