3D positional integration from image sequences

C G Harris and J M Pike

An explicit three-dimensional (3D) representation is constructed from feature points extracted from a sequence of images taken by a moving camera. The points are tracked through the sequence, and their 3D locations are accurately determined by use of Kalman filters. The egomotion of the camera is also determined.

Keywords: robot vision, 3D scene representation, Kalman filter, egomotion

Understanding three-dimensional (3D) scene geometry from a sequence of images requires careful selection and management of the information they offer. Many techniques are conceivable, offering different tradeoffs between complexity of implementation and detail of 3D scene representation. A suitable technique for computer vision must provide a compact representation which is robust and easy to update as further information is acquired from subsequent images. The approach we have taken is based on the REV (region, edge, vertex) graph, which works on the edge and point (i.e. vertex) information contained within an image. This provides a list-based representation which meets the above criteria, and which maintains 3D information for features closely related to those existing in the real world. The REV graph can be divided into two parts: the geometry, which contains all the metrical information (e.g. position and orientation); and the topology, which contains information concerning connectivity of points, lines and surfaces. This paper is concerned solely with the geometry part of the REV graph.

In a sequence of images, each observation of an image feature (e.g. a point, edge or region) provides data on the 3D analogue of the feature. This data permits the metrical representation of the 3D feature to be refined. An example of a (nonoptimal) refining procedure is that of estimating the range of a point feature by triangulation between successive pairs of images in the sequence, and then forming the average 3D position. The refining procedure that is developed here is based on Kalman filters, and thus makes optimal use of all the observations.

The state space of the present Kalman filter is the 3D location of points seen in the images. The main advantage of using 3D points is that they are uncoupled: positional error in one point does not affect any other. If N points are being tracked, then the Kalman filter separates into N 3D state spaces, instead of consisting of one $3N$-dimensional state space. The geometry part of the REV graph could also have included, for example, straight lines and planar surfaces[1,2]. Unfortunately, these would couple the terms in the state space, and would result in a high-order system. In addition, the variables describing straight lines and planes are complexly related to the observations, and suffer from singularities (e.g. when the length of a line is zero, and when a plane passes through the origin).

The processing involved with the geometry part of the REV graph is shown as a flowchart in Figure 1. The initiation of points into 3D from the first two images processed is described as 'bootstrap processing'. Successive processed images are used to determine the camera motion, to refine the estimated positions of 3D points and to instantiate new points into the representation. These processes comprise the 'run-mode processing'. Figure 2 shows (in raster order) the 16 images comprising the sequence.

BOOTSTRAP PROCESSING

The boostrap mode of processing is used to initiate the 3D representation of points. This uses the matched feature points from the first two images of the sequence to estimate the depth of these points, and hence to provide each with an initial 3D instantiation. Details of this computation are given in Harris[3]. Feature points from two images are extracted and matched on grounds of image plane proximity and attribute similarity. These matches are then used to estimate the relative camera motion between the two camera locations. This motion is a six-dimensional quantity, representing both vector

Plessey Electronic Systems Research Ltd, Roke Manor, Romsey, Hants SO5 0ZN, UK

0262–8856/88/02087–04 $03.00 © 1988 Butterworth & Co. (Publishers) Ltd

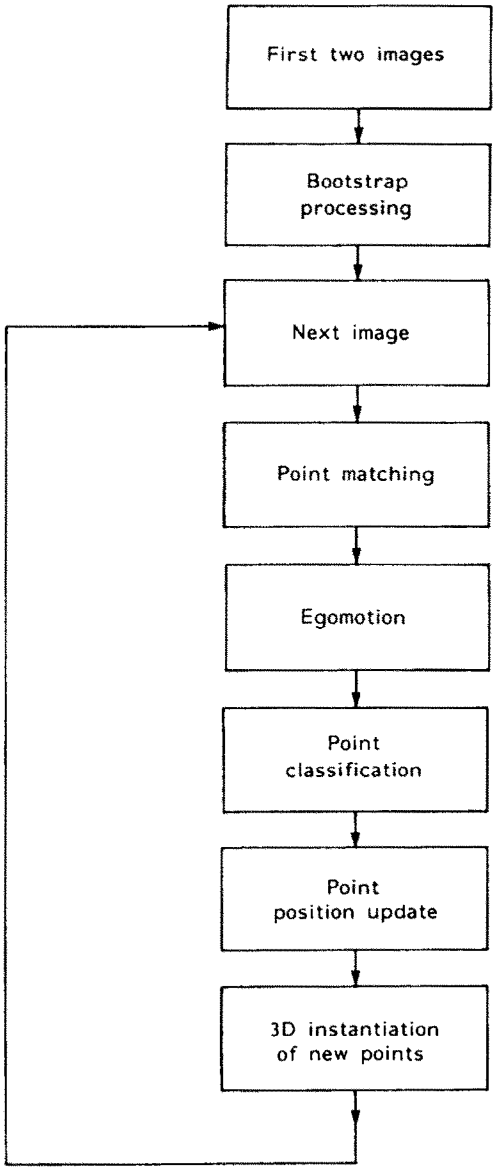

Figure 1. Processing flowchart for the geometry part of the REV graph

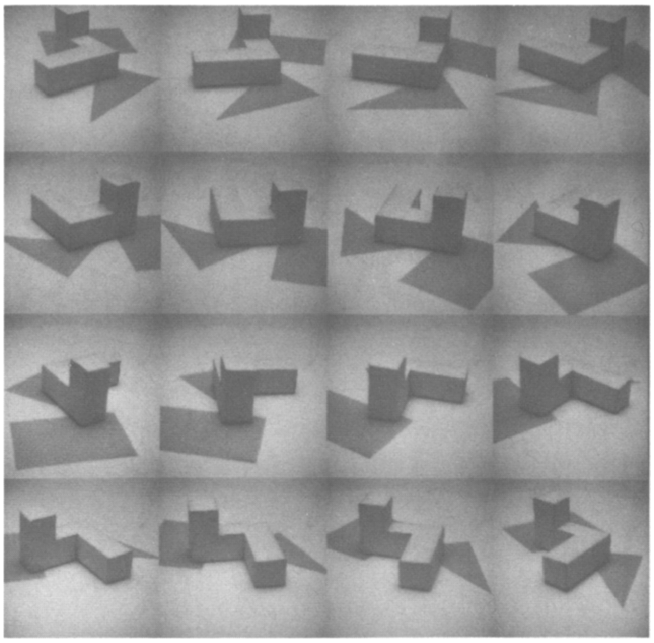

Figure 2. The 16 images of a widget comprising the run-mode processing sequence

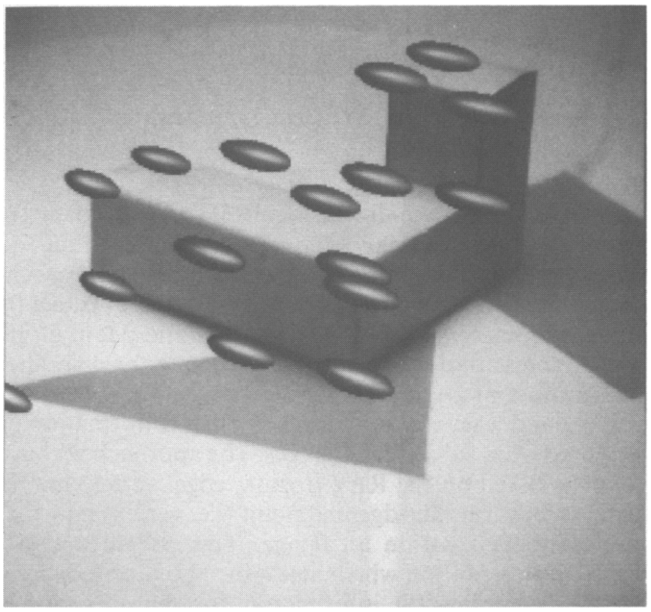

Figure 3. Bootstrap ellipsoids as seen at the fourth camera location

translations and rotations of the camera, and is generally referred to as the 'egomotion'. The egomotion algorithm may fail, or the resulting motion estimate may be too ill conditioned, because, for example, the camera translation may be too small. In this case, boot-mode processing will be attempted on another pair of images.

For each of the matched points, the egomotion algorithm provides an estimate of depth relative to the camera, together with a 'figure of merit' indicating its 3D consistency. Points with a low figure of merit can arise from erroneous matches, and from the obscuration of a distant body by a closer body (obscuration points). Such points are discarded. The remaining points are instantiated in 3D, and this enables subsequent run-mode processing to be performed.

Each instantiated point is represented by a probability distribution function (PDF), indicating the likelihood that the point actually exists at a particular position in space. On the grounds of mathematical tractability, we have chosen to work with multivariate normal PDFs, specified by a centroid (i.e. most probable) position vector and a 3 × 3 covariance matrix. As surfaces of constant probability density are ellipsoidal in shape, we generally refer to the PDFs as ellipsoids. These ellipsoids are situated in a global coordinate system, the origin of which is not necessarily at either camera location.

The initial size and shape of an ellipsoid depends on its range and the relative camera translation between the two bootstrap images. In general, there will be little angular error associated with a matched point, but much radial error. This results in ellipsoids which are elongated towards the cyclopian camera position (i.e. half way between the two actual camera positions). These bootstrap ellipsoids are shown in Figure 3, as seen from the fourth camera location; the four standard deviation surfaces are also shown.

Points which are unmatched, or have been otherwise discarded, are retained for possible later use; these points are said to be in limbo. If, on a subsequent image, matching can be achieved to a point in limbo, then this point is instantiated into 3D with an appropriately sized and positioned ellipsoid. Sufficiently elderly points in limbo are discarded because the validity of their matching attributes decreases with time (i.e. with increasing camera motion) and this makes them prone to incorrect matching. Also, the number of points in limbo cannot be allowed to become too great, otherwise incorrect matching becomes too common.

FEATURE POINT MATCHING

The processing of each new image in the sequence starts with the extraction of feature points. To match these points to previously instantiated ellipsoids, we make use of an estimate of the location and attitude of the camera (called the camera egomotion). The location of the camera is specified by the vector displacement, $\mathbf{t}$, of the pinhole of the camera from a global coordinate origin. The camera attitude is defined by the rotation of the camera away from a reference position in which the optical axis and image axes are aligned with the global cartesian axes. This rotation is specified by the vector $\boldsymbol{\theta}$ whose direction is the axis of rotation and whose magnitude is the angle of rotation.

Given an estimate of the camera egomotion, the perspective projection (in this camera position) of each ellipsoid is computed, forming an ellipse on the image plane. Broadening each ellipse by a few pixels to cater for error in observed feature point positioning and error in the estimated egomotion defines a search region in which candidate points for matching must lie. The selection of one of these candidates as the correct match is performed by inspection of the grey-level attributes of the feature points.

EGOMOTION DETERMINATION

In the run mode, it is necessary to determine the egomotion of the camera for each new image. This is performed by finding the egomotion that brings the observed image plane positions of the matched feature points into alignment with the projection of the ellipsoids. The egomotion must be determined before the ellipsoids can be updated, as we do not assume that the *a priori* estimates of the camera motion are of sufficient accuracy. However, the *a priori* estimates of camera motion may be used as regularizing terms in cases where the image data alone would result in ill conditioning.

Consider a hypothesized camera egomotion, specified explicitly by the six-dimensional vector

$$\mathbf{q} = (\boldsymbol{\theta}, \mathbf{t})$$

We wish to align the hypothesized motion with the true camera egomotion. To do this, the ellipsoid PDFs are first projected into the image plane of the hypothesized camera, resulting in PDFs on the image plane. These projected PDFs are modified appropriately by the PDFs of the observed feature points to take account of the accuracy of positioning of the observed feature points. The 'goodness of fit', $E(\mathbf{q})$, of the hypothesized camera is defined to be the sum of the squared Mahalanobis distance for each of the matched points. Mathematically, this is

$$E(\mathbf{q}) = \sum_{k=1}^{N} [\mathbf{r}_k - \mathbf{r}_k'(\mathbf{q})]^{\mathrm{T}} \, n_k^{-1}(\mathbf{q}) \, [\mathbf{r}_k - \mathbf{r}_k'(\mathbf{q})]$$

where N is the number of matched points, $\mathbf{r}_k$ is the observed image plane position of the kth matched feature point, $\mathbf{r}_k'(\mathbf{q})$ is the position of the kth 3D feature point projected onto the image plane of the camera with hypothesized egomotion $\mathbf{q}$, and $n_k(\mathbf{q})$ is the covariance matrix of an ellipsoid, projected onto the image plane of the camera with hypothesized egomotion $\mathbf{q}$, and modified by the observed PDF.

Minimizing E with respect to the egomotion parameters, $\mathbf{q}$, results in the egomotion estimate of highest joint probability for all the matched points (i.e. the most likely estimate). This minimization is performed iteratively, using either the Newton–Raphson or steepest descent techniques, depending on which performs best at each step of the iteration. At each step of the iteration, a camera egomotion is hypothesized, and a new (hopefully better) estimate is calculated. The minimization techniques require explicit evaluation of the zero-order, first and second differentials of E with respect to $\mathbf{q}$; these are derived analytically.

Incorrect matching is overcome by robust weighting, which adaptively and gracefully reduces the contribution of poorly fitting points in the above summation. No use is made of *a priori* egomotion estimates except for initiating the iteration loop. This is primarily because of the very high accuracy of the visual data, but also because it aids the formation of a selfconsistent 3D representation.

POINT POSITION UPDATE

Each time an instantiated feature point is observed and matched, a more precise estimate of its 3D position is obtained. This is because the new observation provides further information relating to the 3D position of the point, which enables its PDF to be reduced in size. As an analogy, this process may be thought of as each point being associated with a volume in which it is believed to reside, and this volume being pared down by each new view of the point. The updating of the ellipsoids with the new observations is performed by Kalman filters[4], which make optimal use of the information.

An observed feature point will not in general be located precisely at the position of the projection of its causative 3D feature: associated with the observed feature point will be a positional uncertainty. At minimum, this will be a circle of radius half a pixel (since feature points are situated at integral pixels), but a more meaningful expression for the uncertainty could be derived from, say, the size and shape of the local autocorrelation function. As before, we shall write the positional uncertainty as a normal PDF centred on the observed position, and with an appropriate covariance matrix.

The observation of the feature point provides no information about the range of the point, except that it is in front of the camera. Hence we can express the 3D PDF of the observation by a function which has surfaces of equal probability which are nested elliptical cones with their common apex at the pinhole of the camera. The cross-section of the cone is given by the aforementioned 2D covariance matrix.

The current observation may be used to update the ellipsoid by forming the joint PDF of the conical PDF, and the previous estimate of the ellipsoid, and then normalizing for unit probability. This, however, would result in a non-normal PDF (i.e. not an ellipsoid), because one of the constituent PDFs (the conical one) was not itself normal. Normality is regained by approximating the conical PDF to one that is cylindrical, possessing the same cross-section as the cone at the range of the ellipsoid. The centroid and covariance of the resultant ellipsoid are easily calculated.

This approach has problems with 3D points that are distant; their ellipsoids fail to reach to infinity, where the point could lie. The use of ellipsoids will thus introduce a nearness bias for distant points. This problem is overcome by working in disparity space, the axes of which are the current image plane coordinates and the current reciprocal depth. In disparity space, the PDF function of the observed feature point is an exact elliptical cylinder, with cross-section equal to that of the image plane PDF. The transformation of the ellipsoid to and from disparity space is not exact, but is a good approximation when the ellipsoid is small. The joint PDF is calculated as before.

Working in the disparity space of the ith camera location, the operation of the Kalman filter is as follows. Write the centroid and covariance of an ellipsoid before incorporation of the data from the ith image as $\mathbf{R}_i$ and C_i respectively. The observed feature point is located in the image at $\mathbf{r}' = (x', y')$, with observation covariance c' (as 2×2 matrix). These are extended to disparity space as $\mathbf{r}$ (a 3-vector) and c (a 3×3 matrix) by inserting zeros appropriately in the disparity coordinates

$$c^{-1} = \begin{bmatrix} c'^{-1} & \mathbf{0} \\ \mathbf{0}^{\mathrm{T}} & 0 \end{bmatrix}$$

$$\mathbf{r} = (\mathbf{r}', 0)$$

After incorporation of the observation, the centroid and covariance of the ellipsoid, $\mathbf{R}_{i+1}$ and C_{i+1}, are given by

$$C_{i+1} = (C_i^{-1} + c^{-1})^{-1}$$

$$\mathbf{R}_{i+1} = C_{i+1}\,(C_i^{-1}\,\mathbf{R}_i + c^{-1}\,\mathbf{r})$$

POINT CLASSIFICATION

Observed image points can be divided into two classes: those that originate from actual 3D events (such as corners and surface markings), and obscuration points which arise from the conjunction of a pair of edges as seen from a particular camera viewpoint. The obscuration points do not in general correspond to a consistent 3D position, and do not directly give any useful 3D information. Indeed, it is necessary to exclude such points from the egomotion calculation, as they are a major source of spurious information. Points are classified as arising from obscurations if their cumulative positional inaccuracy becomes excessive. Points with positional inaccuracy near the threshold are excluded from the egomotion calculation, though their positions continue to be updated.

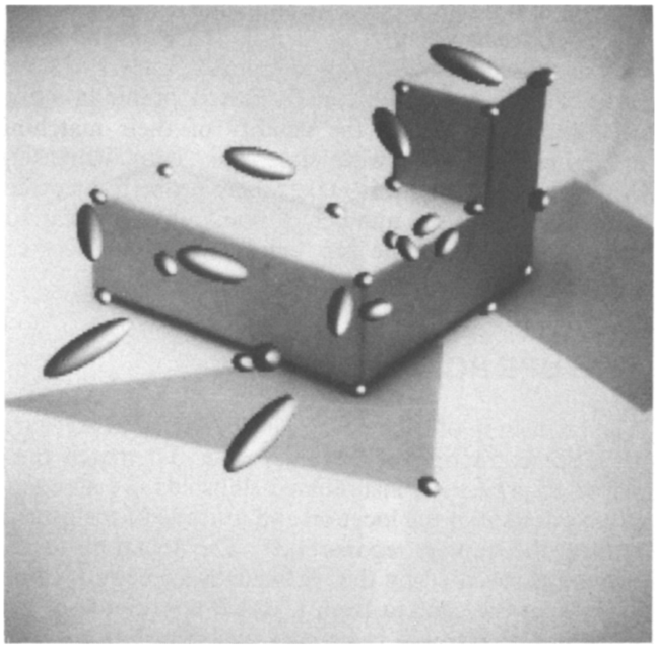

Figure 4. Ellipsoids after processing of all 16 images in the sequence

RESULTS AND CONCLUSIONS

Processing the sequence of images in run mode results in a point representation that rapidly settles down to a solution. Correctly located points end up with tight ellipsoids, whereas spuriously matched points result in large elongated ellipsoids. This is illustrated by Figure 4, showing the ellipsoids after processing of all 16 images of the sequence. The spurious (large) ellipsoids can easily be identified and removed by noting the number of images that contributed to their existence.

The boostrap estimate of camera motion (used to regularize the bootstrap egomotion) has only needed to be accurate enough to ensure that matching is achieved. The determination of egomotion is generally very stable and works well even with only a few matched points, provided they adequately span the space, both across the image and in depth.

The processing for the geometry part of the REV graph has shown itself to be both stable and accurate, and to be able to usefully process image sequences of arbitrary length.

REFERENCES

1 **Porrill, J, Pollard, S B and Mayhew, J E W** 'Optimal combination of multiple sensors including stereo vision' *Image Vision Comput.* Vol 5 No 2 (May 1987) pp 174–180

2 **Ayache, N and Faugeras, O** 'Building registering, and fusing noisy visual imagery' *Proc. IEEE Int. Conf. Computer Vision* (1987) pp 73–82

3 **Harris, C G** 'Determination of ego-motion from matched points' *Proc. Alvey Vision Conf.* (1987)

4 **Gelb, A (ed.)** *Applied optimal estimation* MIT Press, Cambridge, MA, USA (1974)