MATCHING CANNY EDGELS TO COMPUTE THE PRINCIPAL COMPONENTS OF OPTIC FLOW

D A Castelow, D W Murray, G L Scott*† and B F Buxton

GEC Research Ltd.,
Hirst Research Centre,
East Lane,
Wembley,
HA9 7PP.

ABSTRACT

A relaxation algorithm for the computation of optic flow at edge elements (edgels) is presented. Flow is estimated only at intensity edges of the image. Edge elements, extracted from an intensity image, are used as the basis for the algorithm. A matching strength or weight surface is computed around each edgel and neighbourhood support obtained to enhance the matching strength. A principal moments method is used to determine the flow from this weight surface.

The output of the algorithm is, for each edgel, a pair of orthogonal components of the estimate of the flow. Associated with each component is a confidence measure.

Examples of the output of the algorithm are given, and tests of its accuracy are discussed.

I. BACKGROUND

The aim of the *ISOR* system, described in the paper by Murray *et al*[1]. is to obtain a three-dimensional representation of a polyhedral object from an image sequence and then to recognise the object using 3-d model matching.

The algorithm described in this paper was developed to provide estimates of the optic flow near edges in a sequence of images. This method forms an essential precursor to the depth from flow (*dff*) algorithm[2] described in Murray *et al*[1]. and the subsequent 3-d edge matching[3] described in the paper by Murray and Cook[4]. One of the first stages in this process is to estimate the visual motion at intensity edges in the image.

Edges are extracted from raw images using an implementation[5, 6] of the Canny edge detector[7, 8]. This generates edge elements (edgels) which contain the following information;

(i) edge location to sub– pixel accuracy (approximately 0.03 pixels on clean step edges),

(ii) edge orientation (accurate to approximately 1.5° on an ideal image),

(iii) edge strength: the magnitude of the gradient at the point of inflection.

The edge maps are refined using the *hyster* programme[9] which removes edgels of low strength by thresholding and links edges of high strength by hysteresis using the method of Canny[7].

These refined edgels are used by the *mcs3* programme[10], the algorithm for which is described here.

II. AN OVERVIEW

Edgels are used to compute optic flow for three reasons.

(i) For the structure from motion calculations and subsequent processing we require estimates of flow only at the edges of polyhedral objects.

(ii) For the images we are considering, edgels are sparser than pixels, leading to reduced computation times.

(iii) Edgels are more distinctive than pixels, having both a strength and a direction. This information is used in the computation of the initial matching strength.

Along a straight line segment the match of an individual edgel is not unique. Only the displacement or motion perpendicular to the edge can be obtained. This is usually refered to as the "aperture problem." The principal moments decomposition used in the algorithm described here is used to capture this information in an unbiased manner, as suggested by Scott[11]. Although individual edgels do not encode any shape (curvature) information, both components of the motion may be obtained reliably at a distinctive shape such as a corner. This is for two reasons.

(a) The orientation of individual edgels near a corner may be sufficiently distinct to give a single very strong initial match.

(b) The support for each match that is sought from neighbouring matches in the relaxation stage of the algorithm may lead to a single strong match for an edgel near the distinctive shape.

The algorithm may be described by the following steps.

(i) The edges are matched against neighbouring edgels in the forward and backward frames to produce an initial weight surface. Matching to both the forward and the backward frames ensures that the solutions sought will be symmetric with respect to time inversion.

(ii) Support for each match (point on weight surface) is obtained from neighbouring edgels.

(iii) The resulting matching strengths are used to weight the least squares estimate of the visual motion or optic flow.

* G L Scott was a GEC Research Associate at the University of Sussex, Falmer, Brighton BN1 9RH, UK at the time this work was carried out.

† Current address – Dept Engineering Science, University of Oxford, Parks Rd, Oxford OX1 3PJ

This work was undertaken as part of the Alvey programme IKBS 013, "Spatio– temporal processing and optic flow for computer vision."

AVC 1987 doi:10.5244/C.1.27

The least-squares estimate of the velocity vector is found by computing the mean displacement of the edgel from the weight surface and projecting this onto the principal axes of the scatter matrix. These define the components of the motion that can be estimated with greatest and least reliability[11, 12]. The eigenvalues of the scatter matrix are used to compute the confidences of the motion estimates. This ensures that we obtain the appropriate information along an extended edge segment *and* at a distinctive shape such as a corner. It also ensures that the estimates are not biased by the choice of image co-ordinate axes.

These three stages are described in the next three sections.

III. INITIAL WEIGHTS

As mentioned above, three frames of edgels are used and are referred to as the central, backward and forward frames. An estimate of the visual motion is obtained for each edgel in the central frame.

Around each edgel, i, in the central frame is a neighbourhood, N_i, which contains all edgels that lie within a distance, L_N, of the edgel, i. Around the same position in the forward and backward frames are constructed two larger locales, F_i and B_i with radii L_F and L_B respectively. Figure 1 illustrates the neighbourhood and forward locale around an edgel. Each edgel, j, in the forward and backward frames has associated with it a neighbourhood set, N_j.

Computing the matching strengths of an edgel to the forward and backward frames uses the same method. We describe the computation for the forward frame.

Initial matches are made between the edgel, i, and its neighbours in the forward frame, F_i.

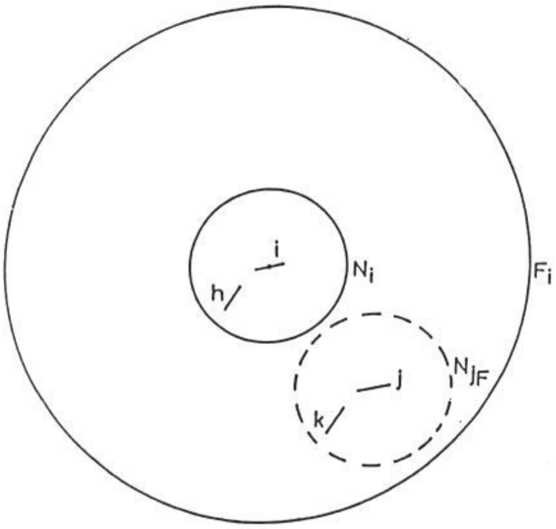


Figure 1. Edgels i and j and their neighbourhoods are shown. Edgel i is in the central frame, edgel j in i's locale, F_i or B_i in the forward or backward frame (here forward). Edgel $h \in N_i$, is a neighbour of i, and $k \in N_j$ is a neighbour of j (in the forward frame).

The initial matching strength of edgel i to an edgel j in the forward frame is

$$p_{ij}^{(0)}\ (j \in F_i) = \exp\left\{-\frac{1}{2}\left[\frac{\delta a_{ij}^2}{\sigma_a^2} + \frac{\delta s_{ij}^2}{\sigma_s^2}\right]\right\}, \qquad \text{(III.1)}$$

where $\delta a_{ij} = a_i - a_j$ modulo 2π and $\delta s_{ij} = s_i - s_j$ are, respectively,
the differences in orientation, a, and strength, s, of the pair of edgels, i and j. The range of edge strengths or orientations which lead to high initial match strengths are specified by the "widths", σ_s and σ_a respectively. This heuristic gives a high matching strength between those edgels which have similar orientation and edge strength.

This function implies that the best matches are ones where there is no rotation of the edgel and no change in its strength between successive frames. Typical values of the parameter, σ_a, are of order 5° to 10°. These values are chosen because of the variability in the directions of edgels as determined by the edge detection stage and to allow some rotation of the whole edge feature from frame to frame. For example, if we ignore the variability in the computed edge orientation and if our image sequence came from successive frames of a video, then specifying a value of $\sigma_a = 5°$ would only have a significant effect for objects rotating at a speed of a third of a revolution per second or faster.

IV. NEIGHBOURHOOD SUPPORT

Once the initial matching strengths, $p_{ij}^{(0)}$, have been found, support for every match is then sought from the neighbours of i. The matching strength is updated according to

$$p_{ij}{}^{(1)} = \frac{1}{n(N_i)+1}\left[p_{ij}{}^{(0)} + \sum_{h \in N_i} \max_{\substack{k \in N_j \\ k \in F_h}} \left[c(i,j;h,k)\,p_{hk}^{(0)}\right]\right], \qquad \text{(IV.1)}$$

where N_j is the neighbourhood of j in the forward frame and $n(N_i)$ is the number of edgels in N_i. The compatibility function, c, is based on differences in relative displacement:

$$c(i,j;h,k) = \frac{(\Delta \mathbf{r}_{ih})^2}{(\Delta \mathbf{r}_{ih})^2 + (\Delta \mathbf{r}_{ij} - \Delta \mathbf{r}_{hk})^2}, \qquad \text{(IV.2)}$$

where

$$\Delta \mathbf{r}_{ij} = \mathbf{r}_j - \mathbf{r}_i \qquad \text{(IV.3)}$$

and $\mathbf{r}_i$ is the position in the image of edgel i. The derivation for the backward frame is similar, with the locale F_i replaced by B_i.

This scheme may be iterated, allowing good matches to be enhanced relative to bad matches by the support of neighbours. This has been tested, and is discussed further in section VIII.

V. MATCHING SURFACES AND THE LEAST SQUARES CALCULATION

The least squares method used to determine the displacement at each pixel (the optic flow) requires weighted sums of displacements.

The algorithm is made symmetric with respect to forward and backward frames by associating displacements $\Delta \mathbf{r}_{ij}$ with $p_{ij}^{(1)}$ for forward matches $i \rightarrow j \in F_i$ but displacements $-\Delta \mathbf{r}_{ij'}$ with $p_{ij'}^{(1)}$ for backward matches $i \rightarrow j' \in B_i$.

First the mean displacement, $< \Delta \mathbf{r}_i >$, is found:

$$< \Delta \mathbf{r}_i > \; = \frac{\sum_{j \in F_i} p_{ij}^{(1)} \Delta \mathbf{r}_{ij} - \sum_{j' \in B_i} p_{ij'}^{(1)} \Delta \mathbf{r}_{ij'}}{\left[\sum_{j \in F_i} p_{ij}^{(1)} + \sum_{j' \in B_i} p_{ij'}^{(1)} \right]} \quad . \qquad \text{(V.1)}$$

Deviations from the mean,

$$\delta \mathbf{r}_{ij} = \Delta \mathbf{r}_{ij} - < \Delta \mathbf{r}_i > \quad , \qquad \text{(V.2)}$$

are then used to construct the scatter matrix, $\mathbf{M}$:

$$\mathbf{M} = \sum_{j \in F_i} p^{(n)_{ij}} \delta \mathbf{r}_{ij} \delta \mathbf{r}_{ij}^T + \sum_{j' \in B_i} p^{(n)_{ij'}} (-\delta \mathbf{r}_{ij'})(-\delta \mathbf{r}_{ij'})^T \; \text{(.V.3)}$$

The error term is

$$E^2 = \mathbf{v}^T \mathbf{M} \mathbf{v} \quad , \qquad \text{(V.4)}$$

to be minimised over unit vectors, $\mathbf{v}$. The minimum is λ_1, the smallest eigenvalue of $\mathbf{M}$. The corresponding eigenvector, $\mathbf{v}_1$, defines the direction along which we have highest confidence in our estimate of the optic flow at the edgel i, $< \delta \mathbf{r}_i >$. The estimate is projected into the co-ordinate system defined by the two eigenvectors, $\mathbf{v}_1$ and $\mathbf{v}_2$. The confidence measures in these components of the estimate are the inverses of the corresponding eigenvalues.

VI. A COMPARISON WITH OTHER MATCHING TECHNIQUES

In the computation of the weight surface, an edgel can match, with various probabilities, to many edgels in the subsequent or preceding frames. It may not be possible to decide upon a unique match when the edgel belongs to an extended straight edge segment. For this reason one-to-one or many-to-one matches cannot be insisted upon. Even with the neighbourhood support, it can take very many iterations of the support algorithm, (IV.1), for the unique information from corners and other distinctive parts of an edge to diffuse to the middle of a straight portion.

One-to-one matching constraints are applied in the matching algorithms of Ibison and Zapalowski[13], Pollard, Mayhew and Frisby (*PMF*)[14], and Lloyd[15].

These algorithms, however, are quite different to ours.

The *PMF* algorithm and that of Lloyd are designed to find unique matches in a pair of stereo images. Even though they match edge elements as here, one-to-one matching can be insisted upon because in binocular stereo the camera geometry is usually known accurately from calibration of the system. The epipolar lines are therefore known and an edgel can only match to another edgel on the same epipolar (usually chosen to be a raster or arranged to be so after rectification of the images). Ambiguous matches along an extended edge segment do not therefore arise unless the edge is nearly parallel to the epipolar, when it leads to the well known "horizontal edge problem."

The relaxation labelling scheme of Ibison and Zapalowski[13] matches distinct feature points from one frame to another. It therefore does not suffer from the aperture problem associated with the ambiguity of matching edgels along a line and hence can utilize a constraint insisting on at most one-to-one matching. and can thus avoid the aperture problem, and hence insist upon one to one matching.

Instead of trying to solve the matching problem, the algorithm described here uses one to many matches to construct a weight surface, the mean and principal axes of which are identified using the least squares technique described in section 5.

VII. ACCURACY OF THE ALGORITHM

Figures 2 to 14 are the results of applying the method to three simple test image sequences.

The first test was on the motion of some simple planar image shapes. One image from the sequence used is shown in Figure 2. Each of the shapes was moved one pixel in both the vertical and horizontal directions between frames. Edges were extracted using the Canny edge detector[5]. The edge map obtained is shown in Figure 3. The estimate of the flow at each edgel is shown in Figures 4 to 6. In Figure 4 the estimate of the full flow is plotted. Neither component of the flow has been thresholded. Compare this with the estimate of the major component of the flow, shown in Figure 5.

In all these diagrams, the estimated flow is **not** drawn to scale.

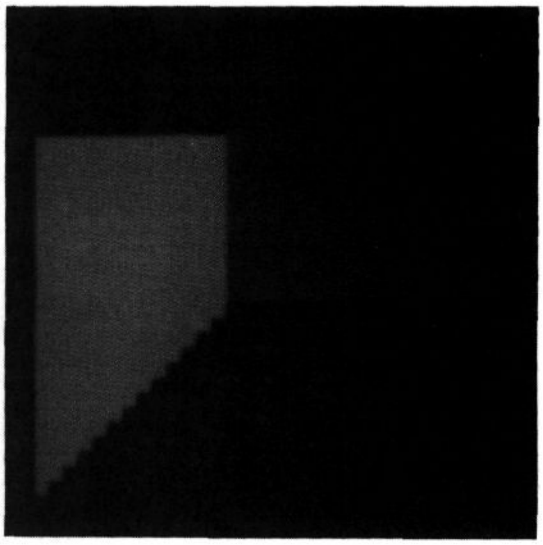


Figure 2. The image used in the accuracy experiments.

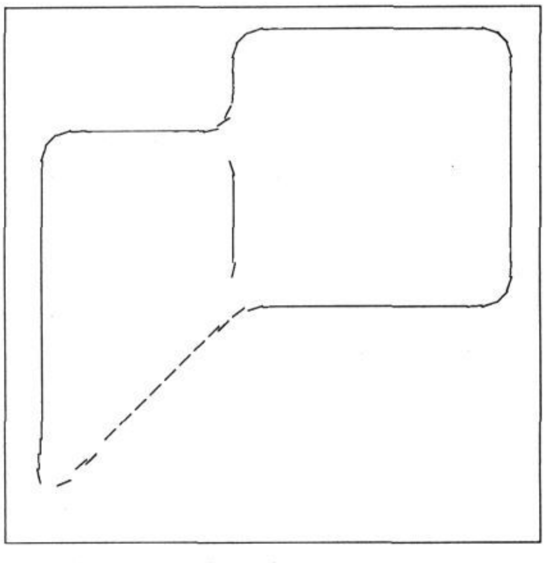


Figure 3. The edge map of Figure 2, produced using the Canny edge detector. Parameters were $\sigma = 1.0$, thresholds 6 and 2.

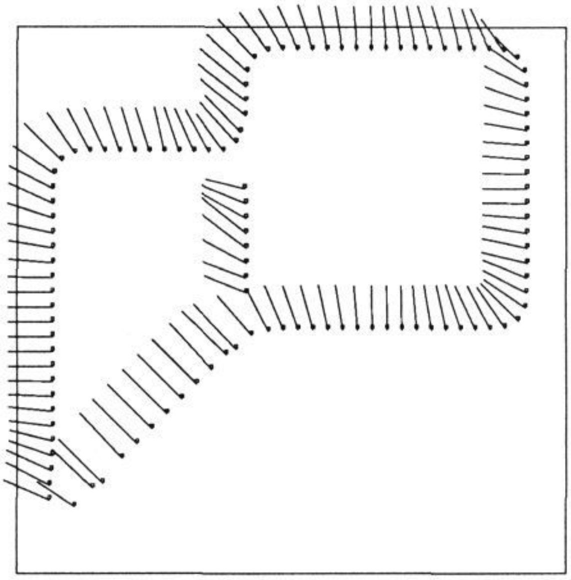


Figure 4. The estimate of the full flow from the geometric shapes image. The neighbourhood support region was of radius 3.0 pixels, the locale size was 5.0 pixels. Matching strength parameters were $\sigma_a = 10°$, $\sigma_s = 10$. No thresholding was applied to the components of the estimate. A single iteration of equation (IV.1) was used to obtain the neighbourhood support.

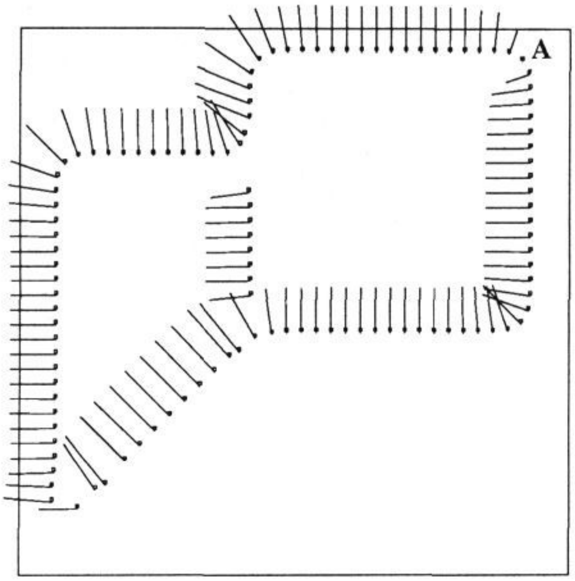


Figure 5. The estimate of the major component of the flow for the shapes image. Parameters as in Figure 4.

At the corners, the direction of the major axis is not perpendicular to the edge, partly because of the change in the estimated orientation of the edgels (by the Canny operator), but mainly due to the support obtained from adjacent vertical edgels. Hence the major component estimates show a deviation towards the true motion.

At *A* in Figure 5, the estimate of the major component is almost zero. In addition, this estimate has a moderate weight (0.44). The minor component estimate has a weight of 0.22. The full flow estimate at this point is the worst one in the figure. It was (-0.75,-0.53) pixels/frame, which must be compared to the veridical motion of (-1.0,-1.0).

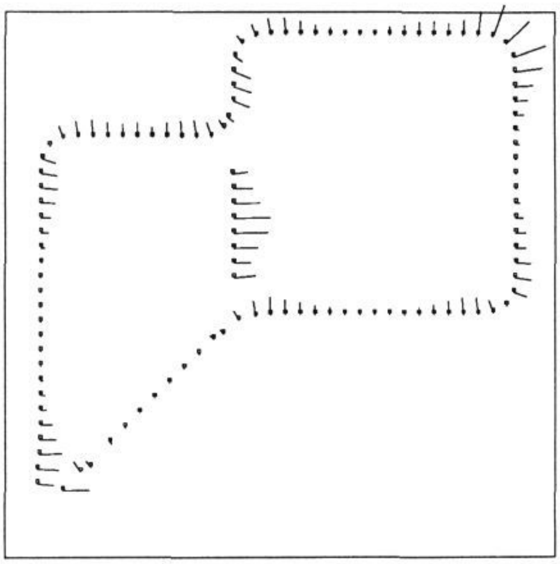


Figure 6. The minor component of the flow vector from the shapes image. To show these clearly they have been rotated by 90°. Parameters as in Figure 4.

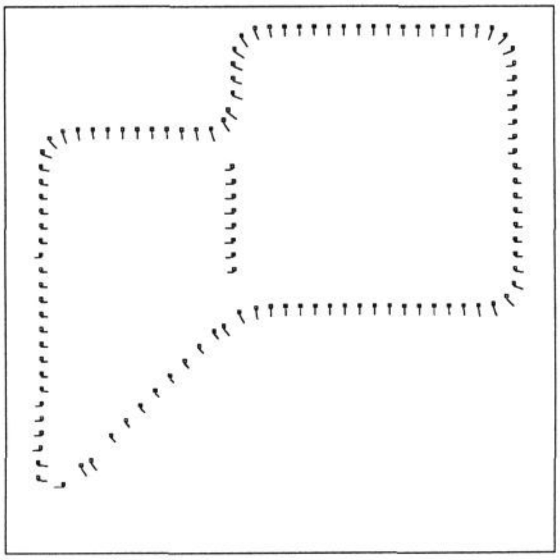


Figure 7. A map of the confidences associated with the minor component of the flow estimate, plotted along the major component axes. Parameters as in Figure 4.

To explain why this estimate is so poor in comparison with those for other edgels, note that this is the only point where the motion is approximately parallel to the edgel direction. In addition, the edgel below and to its right is oriented similarly. As a result, a strong match will be made from edgel *A* not only to the translation of itself but also to the translation of this neighbour. Because the motion is almost parallel to the edgel, this reduces the estimate (if the two matches were of identical strength, the estimate of the flow would be (-0.5,-0.5)).

Along the horizontal line of edges at the top of the square, the estimates of the major components are all within 2% of the correct value (-1.0 pixels/frame), with the majority of the results within 0.1%. The major component estimates are best at the centre of the edge, where there is no influence from the corners.

Along straight edges the minor component has a low confidence associated with it. This is shown graphically in Figure 7.

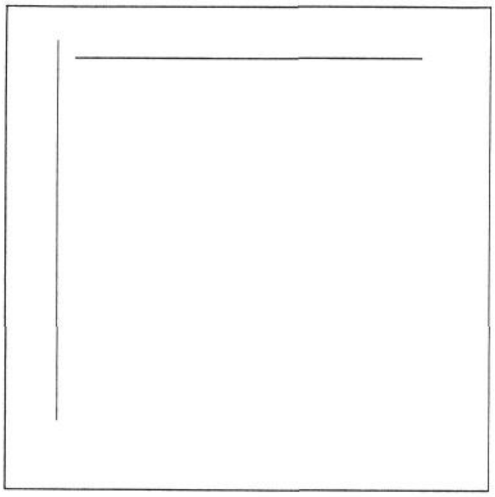


Figure 8. A synthetic edge map for an idealised corner feature.

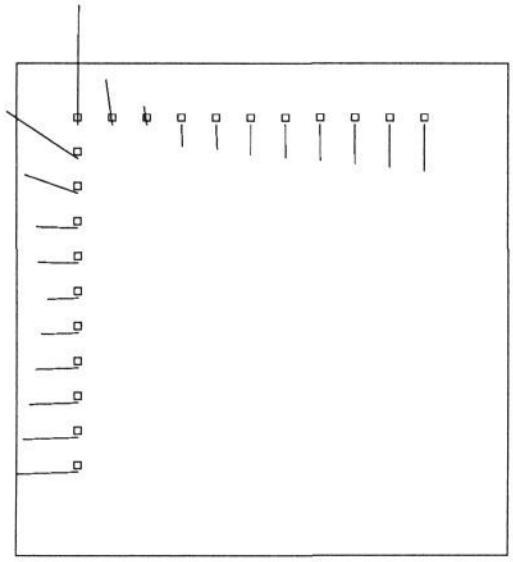


Figure 9. The major component of the flow estimate for a pure rotation of the corner feature in of Figure 8. The rotation was 2° per frame about the apex of the corner. $L_N = 5.$, $L_F = L_B = 3.$, $\sigma_a = 5°$, $\sigma_s = 5.$ 1 iteration of (IV.1).

Further tests have been carried out using sythetic edge maps. These edges are not the output of an edge detector, but are drawn to sub-pixel accuracy. The motion of each extended edge in the images may be specified to sub-pixel accuracy. This was done to show the quality of the results when edges are moved distances which are not an integer number of pixels.

These experiments are of particular relevence in testing the response of the algorithm near a rotating feature. The edges used are shown in Figure 8, and the two estimates of the optic flow shown in Figures 9 and 10.

In Figure 9, the motion of the corner was a pure rotation about the point of intersection of the two line segments. Far from the corner the estimates of the velocity are correct, but near the corner they are systematically altered. All the minor components of the optic flow are badly estimated, as expected on such a featureless image.

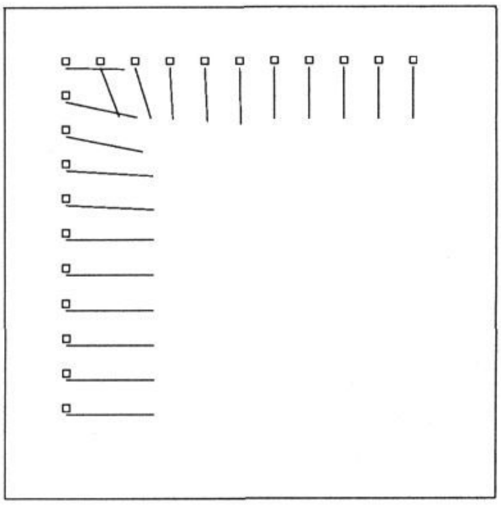


Figure 10. The estimate of the flow for pure translation of the corner feature for Figure 8. The corner was moved diagonally, as described in the text. Other parameters as for Figure 9.

The experiments showed that because of the way the matching surface is used to compute the mean displacement, $< \Delta \mathbf{r}_i >$, the algorithm very badly estimates the motion of points close to a high curvature corner which is also the centre of rotation. However, the occurrence of a corner which has high curvature at a point which appears to be a centre of rotation is a rare event and hence this apparent defect of the algorithm is not thought to be significant. Any possible problem is further alleviated because we are using edge elements from a Canny edge detector[7, 8] which reduces the curvature of corners.

To demonstrate that this error is indeed the result of a fortuitous coincidence of the corner and centre of rotation, Figure 10 shows the estimate of the motion obtained when the corner feature of Figure 8 was translated, rather than being rotated. The motion of the corner was (0.5,0.3) pixels/frame. As can be seen, the estimates of the flow are good. The accuracy may be assessed by noting that the major component of the velocity estimate for edgels further than 2 pixels away from the corner are within 1% of the veridical motion. Close to the corner the estimates are reduced, to about 50% of the true velocity, but the directions of the estimates approach that of the veridical motion.

The most important question to be asked of the algorithm is whether the accuracy achieved is good enough for the purpose for which it was devised.

This computation of optic flow from the edge map of an image was undertaken in order to determine the structure and motion of a polyhedral object, and to recognise the object by matching the 3-d representation of its shape so obtained to models stored in the computer memory. Using the algorithm described here to compute the optic flow, a balsa wood chipped block, identical in shape to the block of Figure 11, was recognised and its motion estimated correctly, as is described in the paper by Murray *et al*[1]. in this conference.

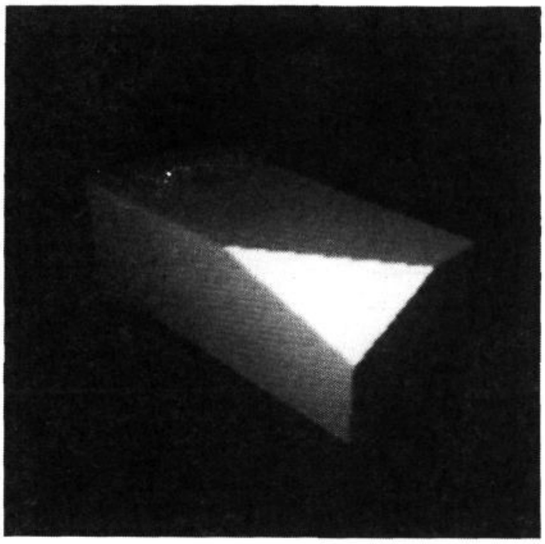


Figure 11. A photograph of the block image used in the experiments concerning the stability of the iteration scheme.

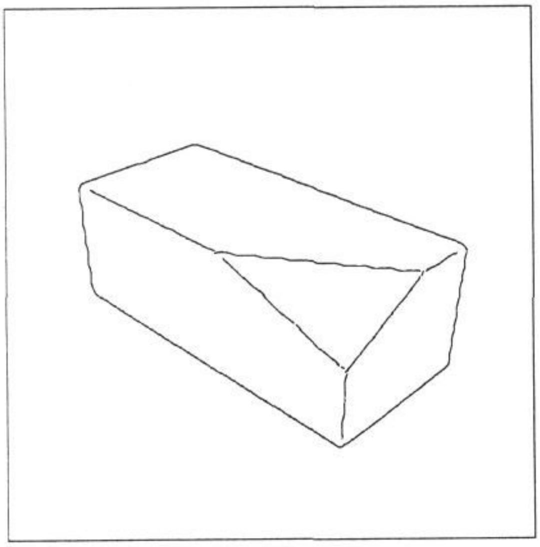
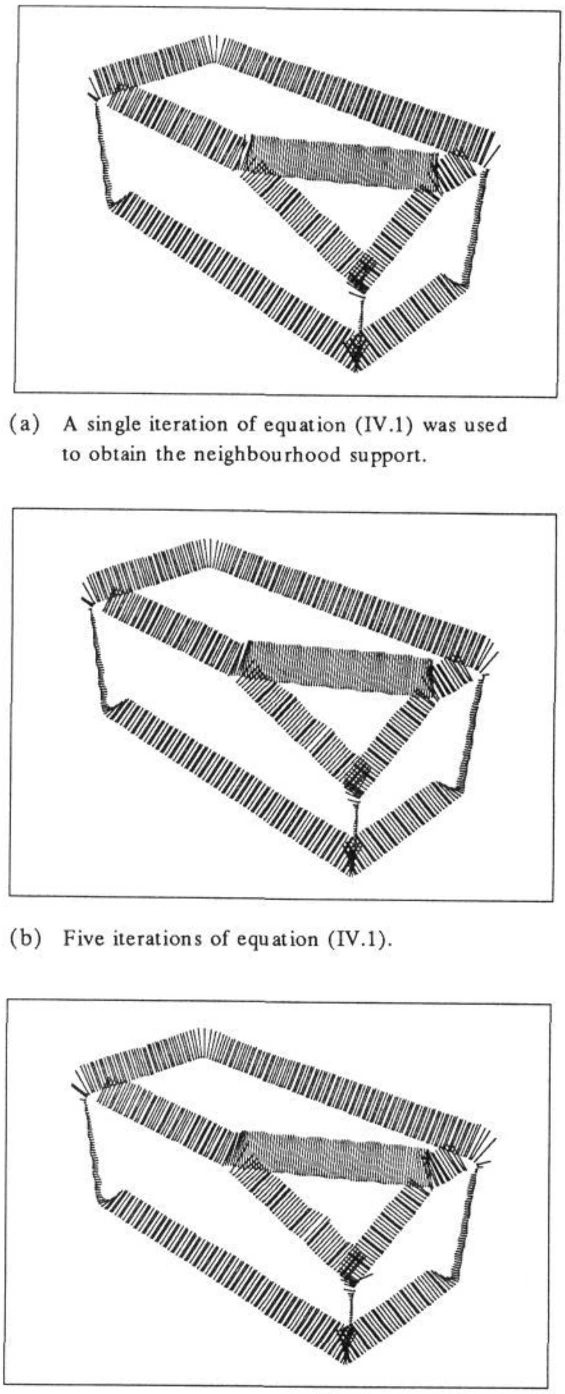


Figure 12. A thresholded edge map from the image in Figure 11. The AIVRU implentation of Canny was used, with a smoothing Gaussian of width 2.0 pixels. The image has been thresholded with thresholds 2.0 and 6.0. See the documentation of *canny*[5] and *hyster*[9].

VIII. ITERATIONS

We have looked at the effects of iterating the algorithm (IV.1) before computing the optic flow. In this case we applied the algorithm to the image of the chipped block, Figure 11. This image was synthesised using the constructive solid body modeller, *WINSOM*[16].
The motion of the block in three dimensions was vertical, with no rotation. Note that this motion is not parallel to any of the edges in the image. The block was moved so that the displacement of its image was approximately 1% of the image size (2.56 pixels) between successive frames.

The edge map is shown in Figure 12. The "Canny edgels" were produced using the Canny operator[5] and refined using Canny's hysteresis algorithm[9].



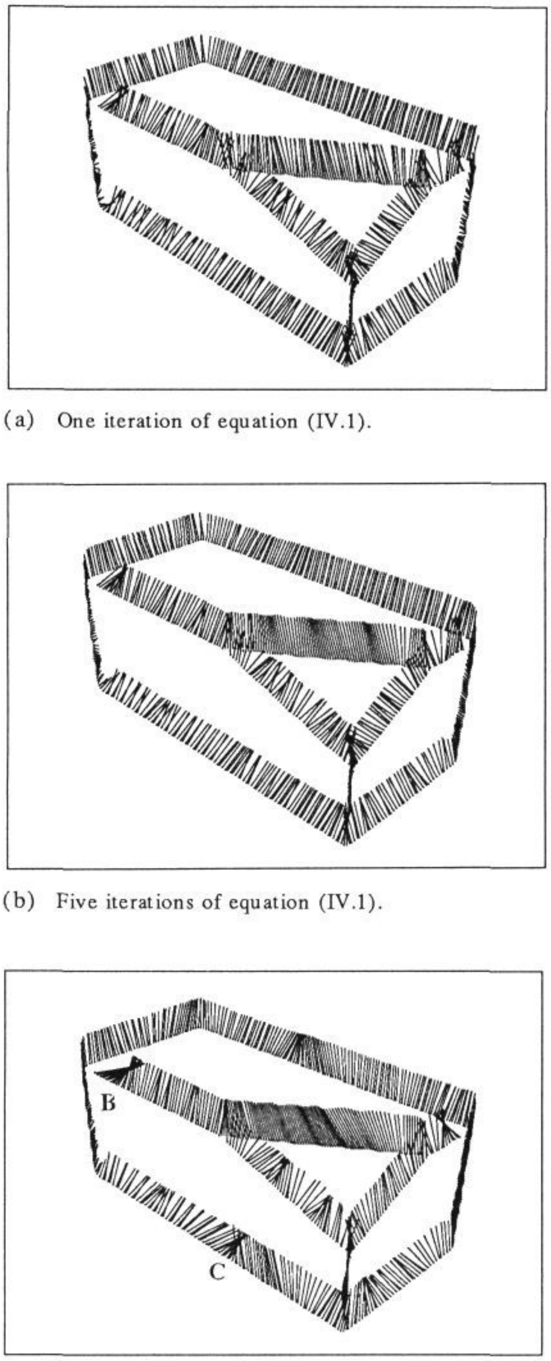
(a) A single iteration of equation (IV.1) was used to obtain the neighbourhood support.

(b) Five iterations of equation (IV.1).

(c) Fifty iterations of equation (IV.1).

Figure 13. The major component of the flow estimated from a sequence of images of the block, Figure 11. The neighbourhood support region was of radius 3.0 pixels, the locale size was 10.0 pixels. No thresholding was applied to the components of the estimate.

The major component of the estimates of the flow are shown in Figure 13, after 1, 5 and 50 iterations of equation (IV.1). Apart from a few points near vertices, these results are in essence the same. The effect of iterating the algorithm has not been to alter the major components of the estimate significantly. The full flow estimates are altered as the neighbourhood support resolves ambiguities in the matches, reflected in the minor components of the flow. This can be seen in Figure 14, where the full flow is plotted ater 1, 5 and 50 iterations.



(a) One iteration of equation (IV.1).

(b) Five iterations of equation (IV.1).

(c) Fifty iterations of equation (IV.1).

Figure 14. The full flow estimate from a sequence of images of the block, Figure 11. Other details as in Figure 13.

The initial effect of iterating equation (IV.1) is to smooth the full flow estimates. This is particularly evident along the top edge of the triangular facet.

After 50 iterations the estimates are much smoother, but there is a definite bias. For example, in some places, the effect of iterating the algorithm (IV.1) is to decrease the correctness of the full flow estimate. The region marked *B* in Figure 14(c), which is close to a vertex, is an area of this type. Because there are no good matches to an edgels' left, the least squares solution is biased to the right.

In other places, the full flow estimates approach the veridical motion (vertical). An example of this is seen in the region marked *C* in Figure 14(c). The reason for this region of good estimates of the full flow is probably due to a distinct edgel, associated with a discontinuity in the original image and to a lack of anti-aliasing of the diagonal edge.

IX. POSSIBLE IMPROVEMENTS AND ADDITIONAL TESTS

The algorithm as described here has performed impressively to date and met the (very) stringent requirements imposed by the subsequent structure-from-motion calculations carried out in the *ISOR* system. However, we note the following points where improvements might be made or additional tests carried out.

(i) The matching strategy does not include the disparity gradient limit[14], despite the use of pairs of matches. Inclusion of this limit could reduce the weight given to false matches.

(ii) Multiplicative updating of the matching strengths of the form used by Ibison and Zapalowski[13] could be used in place of the updating scheme (equation IV.1).

(iii) The algorithm in use[10] already includes provision to threshold the major and minor components of the flow, but this aspect of the programme has not been investigated.

(iv) The matching function described gives equal weight for matches to edge elements of equal strength and orientation, and does not take any account of their length. A geometric factor could be added to the initial weight function, equation (III.1), to take account of this. However, this might reduce the discrimination of the algorithm where the edge element is to be matched to a slightly curved line.

(v) The algorithm has not been extensively tested on highly textured scenes, where an edgel might match not only to many edgels on a line in the subsequent or previous frames, but also to edgels on nearby edges. However, initial experiments on images of a textured wooden block suggest that the optic flow may be estimated sufficiently well that the orientation of the planes may be determined to reasonable accuracy using a planar facet algorithm[17]. The motion of the block was given, and the three visable facets of the block were hand segmented. Two of the three planes were at right angles, and their relative orientation was estimated as 94°.

X. CONCLUSIONS

We have described a method for determining optic flow from a sequence of images, and indicated the quality of the results obtainable.

The full flow estimates are good at feature points in the edge image, such as corners, while along extended straight edges the major component of the flow is the only significant contributor to the estimate.

We have experimented with iterating the neighbourhood support function, equation (IV.1). The increase in support for particular matches seems not to alter the major component of the flow on a straight edge segment, which remains perpendicular to the edge. The full flow estimate is altered. In the image studied in detail, the effect is almost always to move the full flow estimate towards the veridical motion.

ACKNOWLEDGEMENTS

The authors thank the AIVRU at Sheffield University and IBM UK, GEC Research's Alvey partners on IKBS Project 025 for use of the "Canny" edge detector and the *WINSOM* CSG body modeller, respectively.

REFERENCES

1. D W Murray, D A Castelow, and B F Buxton, "From an Image Sequence to a Recognised Polyhedral Model," Submitted to AVC '87, 1987.
2. D W Murray, "Dff – depth from optic flow," in *ISOR User Documentation*, ed. D A Castelow, GEC Research Ltd, Hirst Research Centre, Wembley, 1987. In preparation.
3. D W Murray, "Edgeprune – edge interpretation tree pruner," in *ISOR User Documentation*, ed. D A Castelow, GEC Research Ltd, Hirst Research Centre, Wembley, 1987. In preparation.
4. D W Murray and D B Cook, "Using the orientation of fragmentary 3D edge segments for polyhedral object recognition," Submitted to *Int Journal of Computer Vision*, 1986.
5. University of Sheffield AIVRU, "Canny – an implementation of the Canny Edge detector," in *GDB User Documentation release 1.0*, vol. AIRVU 014, 1985.
6. B P D Ruff, "Icanny – Integer arithmetic Canny edge detector," in *ISOR User Documentation*, ed. D A Castelow, GEC Research Ltd, Hirst Research Centre, Wembley, 1987. In preparation.
7. John F Canny, "Finding Lines and Edges in Images," MIT Artificial Intelligence Laboratory Technical Report AI-TR-720, June 1983.
8. J F Canny, "A computational approach to edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI– 8, no. 6, pp. 679– 698, November 1986.
9. University of Sheffield AIVRU, "Hyster – edge thresholding with hysteresis," in *GDB User Documentation release 1.0*, vol. AIRVU 014, 1985.
10. D A Castelow and D W Murray, "Mcs3 – A method for computing optic flow from Canny edgels," in *ISOR User Documentation*, ed. D A Castelow, GEC Research Ltd, Hirst Research Centre, Wembley, 1987. In preparation.
11. G L Scott, "'Four– line' method of locally estimating optic flow," *Image and Vision Computing*, vol. 5, no. 2, pp. 67-72, 1987.
12. D H Ballard and C M Brown, *Computer Vision*, Prentice– Hall Inc, New Jersey, USA, 1982.
13. M C Ibison and L Zapalowski, "On the use of relaxation labelling in the correspondence problem," *Pattern Recognition Letters*, vol. 4, no. 2, pp. 103-109, North Holland, 1986.
14. Stephen B Pollard, John E W Mayhew, and John P Frisby, "PMF - A Stereo Correspondence Algorithm Using a Disparity Gradient Limit," *Perception*, vol. 14, no. 4, pp. 449-470, 1985.
15. S A Lloyd, "Binocular stereo algorithm based on the disparity– gradient limit and using optimisation theory," *Image and Vision Computing*, vol. 3, no. 4, pp. 177-182, 1985.
16. P Quarendon, "WINSOM User's Guide," UK Scientific Centre report number UKSC 123, IBM, Winchester, Hants, 1984.
17. B F Buxton, Buxton H, Murray D W, and Williams N S, "3D solutions to the aperture problem," in *Proceedings of ECAI-84: Advances in Artificial Intelligence*, ed. T O'Shea, pp. 631-640, Elsevier Science Publishers BV, Amsterdam, 1984.