A New Method for Computing Optical Flow

William Clocksin
Computer Laboratory
University of Cambridge
Cambridge CB2 3QG, UK
wfc@CL.cam.ac.uk

Abstract

Accurate and high density estimation of optical flow vectors in an image sequence is accomplished by a method that estimates the velocity distribution function for small overlapping regions of the image. Because the distribution is multimodal, the method can accurately estimate the change in velocity near motion contrast borders. Large spatiotemporal support without sacrificing spatial resolution is a feature of the method, so it is not necessary to smooth the resulting flow vectors in a subsequent operation, and there is a certain degree of resistance to aperture and aliasing effects. Spatial support also provides for the accurate estimation of long-range displacements, and subpixel accuracy is achieved by a simple maximum likelihood estimation of the velocity distribution function. A confidence value is also estimated, which can help to predict flow ambiguities. The method is demonstrated using image sequences obtained from the analysis of ceramic and metal materials under stress.

1 Introduction

Optical flow is a representation of the instantaneous motion of intensity points in a sequence of images. Since 1980, considerable attention has been devoted to accurately estimating the optical flow given a sequence of images (see the comprehensive surveys [1, 3]). There are two main methods for computing optical flow: gradient methods and correlation methods. Apart from these, other known methods involve the statistical estimation of motion parameters [7] and the use of phase information [8]. Hayton [9] illustrates the relationship between optical flow and image registration methods, surveying the use of non-rigid correlations, mutual information, and multi-scale methods.

Suppose $I(\mathbf{x}, t)$ is the image intensity function returning the pixel grey value at location $\mathbf{x}$ of the image at time t. Gradient-based methods are based on the intensity conservation assumption

$$I(\mathbf{x}, t) = I(\mathbf{x}+\delta\mathbf{x}, t+\delta t) \tag{1}$$

which can be expanded in a Taylor series neglecting higher order terms [10]. In general gradient-based methods are accurate only when this assumption holds, and when frame-to-frame displacements due to motion are a fraction of a pixel so that the Taylor series approximation is meaningful. There are also problems with accuracy in determining second order derivatives due to the sensitivity of numerical integration. In practice,

BMVC 2000 doi:10.5244/C.14.13

gradient-based methods require iteration to reduce the error and need to be augmented by a subsequent process of smoothing the resulting flow vectors over a larger image region. The need to take into account flow information over a large image region for smoothing, together with the need to cope with large displacements, has led to gradient-based approaches being supplemented by hierarchical, coarse-to-fine resolution searches, in which a first approximation to image motion is estimated using a coarse grid, and then using this information to define finer grids to limit the area in which displacements are found [2]. Another approach is to apply anisotropic smoothing, so that flow vectors are not smoothed across motion boundaries.

Correlation-based methods have been useful in sequences where the assumptions required for gradient-based methods do not apply, for example in cloud [14] and combustion [13] images. Such methods try to establish correspondences between invariant features between frames. Typical features might be blobs, corners, and edges. Alternatively, patches of the image itself may be used directly as templates to be matched in subsequent images. The main difficulties with correspondence-based methods are the uncertainty in finding invariant features due to distortion of the image, and the uncertainty in finding corresponding features. Again the spurious results obtained due to this uncertainty need to be removed or smoothed.

While the correlation surface computed by correlation-based methods is intrinsically multi-modal, and thus capable of representing multiple motions, gradient methods assume that locally there is only one coherent motion, and they attempt to find the best values for certain parameters that describe the motion. Such single-motion methods are unreliable near motion contrast borders because the best value amounts to the mean of the various velocities present. As the surveys referred to above have shown, this is almost never the correct response. Instead, one can consider distributed representations of optical flow. Scott [11] uses an iterative procedure that finds least-squares solutions to the intersection of four constraint lines in velocity space, giving much improved estimates near motion boundaries. Simoncelli [12] uses flow information to find a probability distribution in velocity space, but uses Gaussian parameters for the entire distribution to estimate a single motion result. Black and Anadan [4] use robust estimation to reduce the erroneous influence of outliers at motion boundaries.

The method described here also uses distributions in velocity space, and was developed in response to a need to compute flow vectors for surface strain mapping of materials, where both gradient-based and correlation-based method give poor results because both large displacements and local distortions are encountered, and because of the need for accurate measurement near motion boundaries. More background information on the method described in this paper is given elsewhere [6], and an evaluation of the method by comparison with other standard optical flow algorithms has been carried out [5].

2. Method

The proposed method uses the fundamental intensity conservation assumption, but proceeds by estimating the probability that $I(\mathbf{x}, t)$ and $I(\mathbf{x}+\delta\mathbf{x}, t+\delta t)$ are observations of the same event which can be assigned the interpretation that image region $\mathbf{x}$ is displaced to $\mathbf{x}+\delta\mathbf{x}$ from frame t to $t+\delta t$. We can assume without loss of generality that each $\mathbf{x}$ refers to the location of a single pixel. Given pixel grey values $p_1 = I(\mathbf{x}, t)$ and $p_2 = I(\mathbf{x}+\delta\mathbf{x}, t+\delta t)$, let $P(p_1 = I(\mathbf{x}, t)\ \&\ p_2 = I(\mathbf{x}+\delta\mathbf{x}, t+\delta t)\ |\ \delta\mathbf{x}=\mathbf{v}\delta t)$ denote the probability that the observation of p_1 and p_2 is due to the velocity $\mathbf{v} = \delta\mathbf{x}/\delta t$. Henceforth we will abbreviate $P(p_1 = I(\mathbf{x}, t)\ \&\ p_2 = I(\mathbf{x}+\delta\mathbf{x}, t+\delta t) \mid \delta\mathbf{x}=\mathbf{v}\delta t)$ by $P(\delta\mathbf{x} \mid \mathbf{v})$ without loss of generality. A definition for P that is a generalisation of the conservation assumption is:

$$P(\delta\mathbf{x} \mid \mathbf{v}) = \exp(-(p_1 - p_2)^2/\alpha). \qquad (2)$$

The probability is unity when p_1 and p_2 have identical grey values; the probability decreases as the difference between the values of p_1 and p_2 increases. The rate of decrease depends on parameter α, a reasonable value for which is an estimate of the variance σ^2 of the grey values in the image.

It is necessary to provide support for P to model the velocity structure of the image sequence in a given region. This is done by defining a neighbourhood of pixels in the (spatial) vicinity of $\mathbf{x}$. This neighbourhood (which can also be called a patch or receptive field) can be defined as a set of displacements from a central reference point. For example, if the neighbourhood is a square of side $2s$ centred on $\mathbf{x}$ (but offset by one pixel), we may have the neighbourhood

$$\mathrm{R} = \{\delta\mathbf{n} = (\mathrm{n}_1, \mathrm{n}_2) \mid -\mathrm{s} \leq \mathrm{n}_1 < \mathrm{s}, -\mathrm{s} \leq \mathrm{n}_2 < \mathrm{s}\}$$

so that $\mathbf{x}+\delta\mathbf{n}$ ($\delta\mathbf{n} \in R$) refers to a pixel in the neighbourhood of $\mathbf{x}$. In practice, $\delta\mathbf{n}$ will have integer components and represent a pixel offset, and without loss of generality we shall identify offsets using the δ notation. Alternatively, a disc-shaped neighbourhood of radius r may be used, where again integer components are assumed. We now estimate the velocity distribution function of P by

$$F(\delta\mathbf{x}) = 1/|\mathrm{R}| \sum P(\delta\mathbf{x} \mid \mathbf{v}) \qquad (3)$$

where the sum is taken over all $\delta\mathbf{a}\ \varepsilon\ \mathrm{R}$, $\delta\mathbf{b}\ \varepsilon\ \mathrm{R}$, such that $\delta\mathbf{b}-\delta\mathbf{a} = \delta\mathbf{x}$. The value of $F(\delta\mathbf{x})$ is determined by all pairwise pixels at $\mathbf{x}+\delta\mathbf{a}$ (a neighbour of $\mathbf{x}$ at time t) and $\mathbf{x}+\delta\mathbf{b}$ (a neighbour of $\mathbf{x}$ at time $t+\delta t$), such that $\delta\mathbf{b}$ and $\delta\mathbf{a}$ are separated by displacement $\delta\mathbf{x}$**.** The value of $F(\delta\mathbf{x})$ is therefore the support for the hypothesis that the image velocity is caused by displacement $\delta\mathbf{x}$, and can be thought of as the number of 'votes' for the hypothesis. This vote is accumulated by summing Eq. 2 for every pair of pixels in the neighbourhood of $\mathbf{x}$ that are displaced by $\delta\mathbf{x}$**,** for all $\delta\mathbf{x}$. The estimated displacement is therefore the index of the mode of F, or argmax $F(\delta\mathbf{x})$, although in the implemented version a subpixel resolution is obtained by finding a maximum likelihood estimate for $F(\delta\mathbf{x})$. It is necessary to compute F at every location $\mathbf{x}$ for which a motion estimate is required. Optimised algorithms for computing F together with a biologically related model are given in [6].

It is necessary to make a refinement of the velocity distribution function. The voting system of Equation 3 is biased because different displacements are not equally represented within the neighbourhood. For example, in a 32×32 square neighbourhood, 1,024 votes will contribute to $\delta\mathbf{x} = (0,0)$, but only 783 votes will contribute to $\delta\mathbf{x} = (5,3)$. In general, a displacement of $\delta\mathbf{x}$ will receive

$$V(\delta\mathbf{x}) = |\ \{\delta\mathbf{x} \mid\ \mathbf{b} - \mathbf{a} = \delta\mathbf{a}\ \varepsilon\ \mathrm{R}, \delta\mathbf{b}\ \varepsilon\ \mathrm{R}\ \}\ | \qquad (4)$$

votes independent of the pixel values. The pixel values will determine the amount of vote according to Eq 2. Thus, a uniform random vote has the effect of biasing the response towards small values of $|\delta\mathbf{x}|$, even if the voted amount is a small one because the difference between pixel values is large. Consequently, the presence of noise in the image may conspire with a biased voting scheme to conceal the correct response. The effect of bias is illustrated in Figure 1 by the histogram of V for a disc neighbourhood of radius 16. The axes of this plot are labelled such that (dx, dy) corresponds to $\delta\mathbf{x}$. The zero value is in the middle of each axis (*i.e.* $\delta\mathbf{x}=\mathbf{0}$ is in the centre of the F plot). The graduations are in units of 10 pixels displacement. The central bright region has a maximum of 795 votes for $\delta\mathbf{x} = (0,0)$. The overall distribution of votes is cone-shaped.

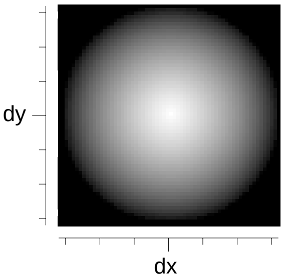


Figure 1. Histogram of systematic voting bias for a disc neighbourhood of radius 16 pixels.

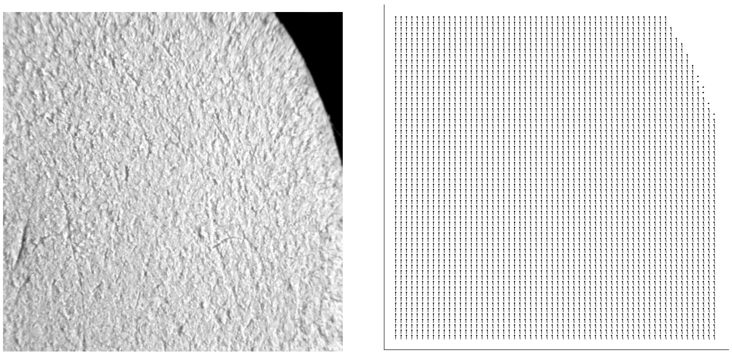
The effects of bias may be corrected by estimating the response due to bias alone, and correcting the response from Eq. 3. This is implemented by subtracting histograms. The response due to bias may be modelled by multiplying V by the probability that given pixel values are drawn from the image. Equation 2 assumes that pixel values are drawn uniformly from the range of values (typically 0 to 255). However, in practice, pixel values are drawn from the image, and the discrete probability distribution function (pdf) of pixel values can be estimated by the histogram of the image. With the frequency of pixel value i in the image at time (frame) t, estimate the discrete pdf of a pixel value i as $h_t(i) = f_t(i)/\sum_j f_t(j)$, where index j ranges over all the pixels in the image. Again letting $p_1 = I(\mathbf{x}, t)$ and $p_2 = I(\mathbf{x}+\delta\mathbf{x}, t+\delta t)$, Equation 3 may be redefined as

$$F(\delta\mathbf{x})\ |\mathrm{R}| = (\sum P(\delta\mathbf{x} \mid \mathbf{v}))(1.0 - h_t(p_i)\ h_{t+\delta t}(p_i)V(\delta\mathbf{x})) \qquad (5)$$

Where the sum is taken the same as Eq. 3. The effect of this is to greatly enhance the signal-to-noise ratio of the response, which will be seen below. For efficiency, the right-hand factor of Eq. 5 (the bias correction factor) is precomputed from the histogram of each image in the sequence, and is used for each computation of F. Image subtraction can be used instead of multiplying probabilities. In the results that follow, the bias correction term was computed for each vector, by using the elements of R to estimate the pdf and the parameter α (Eq 2). This requires a substantial increase in computation time compared to estimating one pdf and value of α for all flow vectors; any benefit in terms of increased accuracy has so far been indiscernable.

3. Results

Image sequences were obtained from photomicroscopic views of various heterogeneous and composite materials under mechanical stress. The detection of image flow from these sequences is of interest to materials scientists because of a requirement to map surface strain. Monochrome images were acquired, digitised to 7 bits, and histogram equalised. Pixel displacements varied from less than one pixel per frame (pf^{-1}) to 19 pf^{-1}. Eq. 5 was computed for each pixel using a pair of images. Figure 2a shows one of the 512×512 pixel images in a sequence of a rubber-type material being translated on the stage. Figure 2b shows a set of velocity vectors obtained by computing F at each point of a 64×64 grid centred on the 512×512 images. (*i.e.* one vector computed every 8 pixels of the image). The neighbourhood was a disc of radius 16. Flow magnitude has been scaled by a constant (0.5) for legibility.

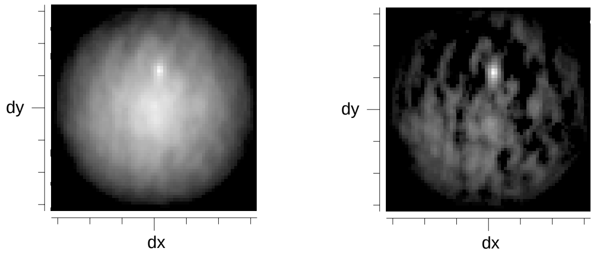


(a) (b)

Figure 2 (a) A frame from the rubber material displacement sequence; (b) flow vectors.

In Fig. 2(b) the flow is clearly defined except for a few isolated locations, and a few aliasing errors near the light/dark boundary in the upper right quadrant of the image. Apart from a few aliasing errors, the computed flow matches the ground truth (obtained from direct measurement of the microscope stage displacement) to within 0.001 pixel. Note that no result is given for some points in a region in the upper right corner of the image. For each point, the dispersion of its F map is compared with a threshold, giving a confidence value for the point. Flow vectors of low confidence are not displayed.

Figure 3 shows the values of *F* over the range of displacements for one pixel near the centre of the image in Fig. 2(a). The axes of each F plot are labelled such that (dx, dy) = δ**x**. The zero value is in the middle of each axis (i.e. δ**x**=**0** is in the centre of the *F* plot). The graduations are in units of 10 pixels displacement. In Figure 3(a) the maximum value of *F* is obtained at δ**x**=(1.06, -11.05) by a weighted average about the mode (an estimate of the maximum likelihood). This is taken to mean that the optical flow where the *F* plot was sampled has a displacement of (1.06, -11.05) pixels per frame (pf^{-1}). To illustrate the favourable effect of bias correction on Figure 3(a), Figure 3(b) shows the value of *F* as computed by Eq. 3, where bias is not corrected. The obtained δ**x** =(1.11, -11.12) in Figure 3(b) corresponds to a (0.05, 0.06) pixel error between the displacement measurements of the two histograms.



(a) (b)

Figure 3. It is possible to see how subtracting the image in Fig. 1 from that in (a) gives the image (b).

The flow fields in Figure 4 show examples of rotation (approx 0.06π rad f^{-1}) and shear obtained from ceramic samples. Each was generated by applying the algorithm to every eighth pixel position on a pair of 256×256 pixel images of a surface similar to Figure 2. The rotation field in 2(a) was obtained by rotating the material in the frontoparallel plane; Figure 2(b) shows a motion boundary cause by moving two overlapping plates of the material in different directions in the frontoparallel plane. Note the inaccuracies in the rotation field owing to corner effects: during rotation, pixels pass in and out of the image at the corners, and so the conservation condition (Eq. 1) is violated.

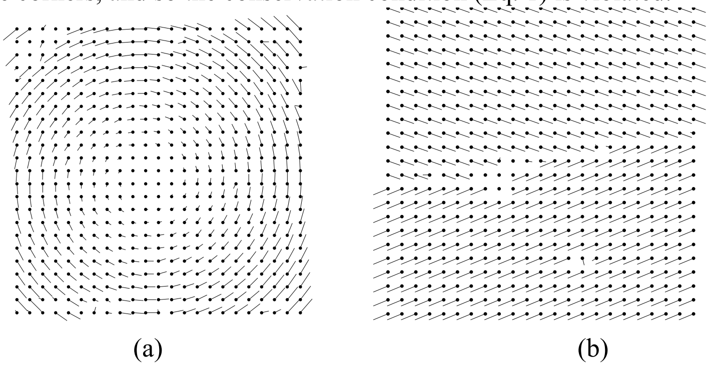


Figure 4. Illustration of (a) rotary and (b) shear flow.

In the shear field, the flow is clearly defined near the boundary except for a group of six patches along the actual dislocation of the material. A better illustration of what happens at motion boundaries is as follows. The pair of images in Figure 5 shows *F* plots at positions 1 pixel either side of the boundary. Both images look similar, as they are integrating the support in locations separated by only 2 pixels. In Figure 5(a), the maximum value is at (-17.03, -7.05). This is the flow value for a pixel on the lower side of the boundary, which has a true velocity of (-17.0, -7.0). In Figure 5(b) the maximum value is at (13.99, -4.86). This is the flow value for a pixel on the upper side of the boundary, which has a true velocity of (13.95, -4.85).

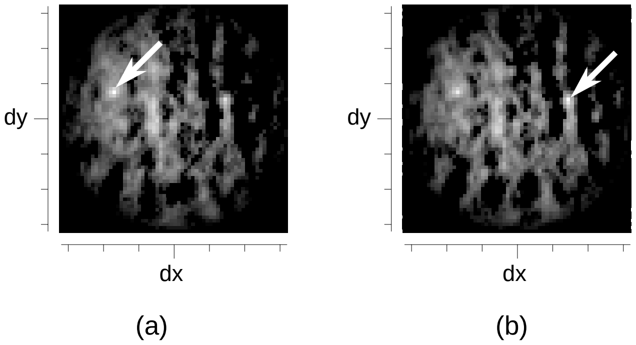
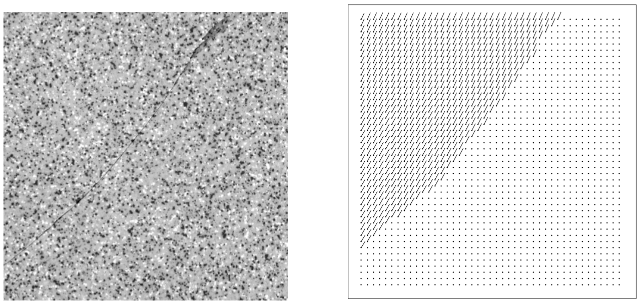


Figure 5. *F* plots for pixels at two sides of a motion boundary. Arrows point to the maximum value of each *F* plot.

This shows that the multimodal response of the function is important in discriminating the boundary even at close proximity.

Figure 6(a) shows an image from a sequence in which a ceramic (particulate composite) material has developed a hairline fracture under stress. The flow diagram in Fig. 6(b), derived from a consecutive pair of timelapsed images, shows that only the material on one side of the fracture has moved: this can give an indication useful in determining the cause of the fracture.



(a) (b)

Figure 6 Ceramic fracture: (a) an image in the sequence; (b) flow vectors.

4. Discussion

Three main data conditions affect the accuracy of the method. These are ambiguity, aliasing, and the aperture effect. Since the early 1980s [10], the aperture effect, popularly called the ‘aperture problem’, is widely discussed in the optical flow literature, but the other two conditions are not generally acknowledged. The vulnerability of any flow estimation method to these conditions should not be seen as a shortcoming of the method, but as an inherent condition of the data. Algorithms can take into account more assumptions to resist these conditions, but it is misleading to see these conditions as ‘problems’ that need to be ‘solved’, because there can be no solution in the usual sense of the word. After all, the human visual system is also vulnerable to the same conditions. Clocksin [6] discusses the sensitivity of the method to these conditions, illustrating with pathological conditions. A brief summary follows.

4.1 Data Conditions

The method described here exposes degenerate conditions as the dispersion of the probability distribution obtaining at a given spatiotemporal location. Ambiguity is a situation where, for example, the same pair of images may represent one region of pixels moving right or another group of pixels moving left, or both, or swapping of left and right. There is no consistent motion interpretation. In this case, ambiguity is revealed as a pair of peaks in F with wide dispersion.

Aliasing is where the drift of a periodic pattern may be variously interpreted as motions of integral multiples of the period of the pattern. In this case, aliasing is revealed as a sequence of peaks having the same period as the pattern. Again, the wide dispersion is a clue to interpretation.

The well known aperture effect is illustrated by the image sequence in Figure 7(a,b). This is ambiguous because the direction of motion of the light/dark boundary is undefined. The same image sequence could be caused by the light (or dark) region moving vertically down, or down and to the right, or to the right.

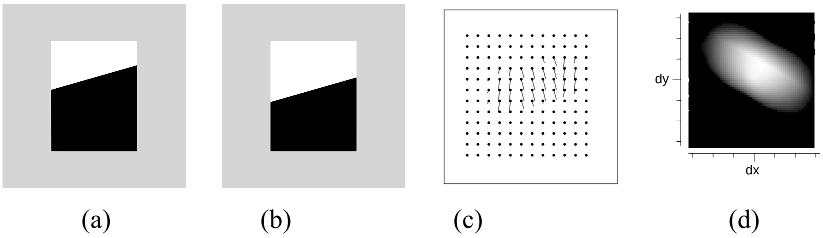


Figure 7. The flow vectors in (c) are unscaled.

The resulting flow diagram in Fig. 7(c) shows the method's attempt to deal with this situation. The flow vectors near the middle of the boundary are perpendicular to the boundary (this interpretation has most local support), while the flow vectors at the edge of the boundary point vertically down because of the edge effect of the surround. Such a result is not usable as it stands, and in the lack of other support no one interpretation can be favoured. The vectors shown in Fig 7(c) are simply the ones corresponding to the maximum likelihood of their corresponding *F* plots. However, Fig. 7(d) shows that the extremely large dispersion of the *F* plots for any of the nonzero vectors immediately suggests a low confidence for the choice of any particular displacement

In each degenerate case, a wide dispersion is the key to estimating the confidence with which the maximum likelihood of *F* can be used to estimate a flow vector value. A large dispersion tells us that *F* is unlikely to be meaningful. In all the situations I have investigated (which include timelapse imagery taken from video cameras, scanning electron microscopes and magnetic resonance imaging devices) it appears that an unreliable flow vector is always betrayed by an *F* plot showing a maximal peak of relatively large variance. The studies in [6] used the variance as a measure of dispersion. This is not suitable in general because *F* is not gaussian and is often multimodal. A more suitable measure of dispersion, which is used in current work, is illustrated in a one-dimensional version in Figure 8.

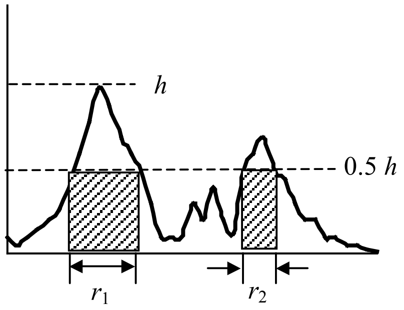


Figure 8. A one-dimensional illustration of the measure of dispersion suitable for nonparametric distributions. It is computed by integrating all r_i, the intervals over which the amplitude exceeds half h, then normalising by dividing by h.

4.2 Implementation Issues

Several practical factors influence the efficiency and accuracy of the method. First, to obtain a dense flow field, an *F* plot is required at each pixel location where a flow vector is required. This means that a patch will overlap with several others. Every pairwise comparison that is inside the overlap of two or more patches need be computed only once, and the result used in each of the overlapping *F* plots. For the implementation used to obtain the results reported in this paper, this more efficient scheme was not used.

Instead, an *F* plot was computed anew for each patch. However, in a parallel architecture such as custom integration or the human visual system, great computational savings can be made by sharing the result of the pairwise comparison. Clocksin [6] describes a asynchronous model in which *F* plots have a columnar organisation, with pairwise comparisons being channelled to a number of *F* plots associated with overlapping patches (receptive fields). A time delay element is used to provide the temporal separation of one of the pair of inputs.

Second, the sensitivity of the method in picking up evidence for various velocities is determined by shape and size of the patch. In the implementation, a disc of radius 16 (pixels) was used for the patch. This has the benefit that vote bias is radially symmetric, but resources (*i.e.* amount of computing time in a sequential implementation; number of components in a parallel implementation) are used to ensure that the patch is equally sensitive to motion in any direction. However, in situations where there is prior information about the preferred range of motions, different patch shapes may be used to improve displacement range, accuracy, or efficiency. An example is given by an ellipsoidal receptive field, which is sensitive to long range displacements in the direction of the long axis, and shorter range displacements in the direction of the short axis. Thus, the conventional idea of velocity receptive fields in the human visual system is a special case of this method, although physiological models use the concept of a 'preferred' direction and magnitude.

Third, there is the question of support in the temporal dimension. All the examples shown in this paper use the minimum of two frames of temporal support. Providing increased temporal support is very easy to implement, and involves accumulating the *F* plot values using the pairwise comparisons derived from as many pairs of images in the sequence as are required for temporal support. This suggests that temporal support may perform noise averaging as well as accentuating the signal, at the risk of 'smearing' the *F* plot if the interframe interval is too large.

5. Conclusions

The approach described here considers optical flow at only a single local scale of resolution, but instead uses a probabilistically inspired model, with large spatiotemporal support for the decision. It uses the conservation assumption of Eq. 1 to motivate the model of Eq. 2 and 5 instead of motivating the use of gradients. The spatial area of support is related to the maximum displacement observable by the method. Temporal support can be extended to an arbitrary number of frames, yet good results are obtained using only two frames. There is no reason why the method cannot be extended to a multi-scale approach either spatially or temporally or both.

The method shares some similarities with correlation-based approaches. Like correlation approaches, a multi-modal response surface is computed, but unlike correlation approaches, it accumulates the probability of pairwise observations instead of matching templates or features, and therefore can deal with non-rigid displacements. Because pairwise observations are performed rather than the comparison of image patches, performance is better than correlation-based approaches when deformations within the image are encountered. This property not only makes the method feasible for the test application of surface strain mapping of heterogeneous materials [5], but also leads to a more reliable method for estimating optical flow in general. The use of a multi-modal response surface leads naturally to the derivation of a confidence factor, which is useful in assessing the ambiguity of the response.

Acknowledgements

I thank Katherine Chivers and Phil Withers for providing the image sequences from their research, which is supported by a grant from British Aerospace. I thank Massimo Gaetani of Axiom Bildverarbeitungssysteme for supplying the ceramic surfaces. Ceramic imaging was done on equipment provided by EC ESPRIT Project CERIND. I also thank Chris Dance and Tom Körner for suggestions and advice.

References

1 Barron, J.L., Fleet, D.J., and Beauchemin, S.S., 1994. Performance of optical flow techniques. *International Journal of Computer Vision* **12**(1), 43-77.

2 Battiti, R., Amaldi, E and Koch, C., 1991. Computing optical flow across multiple scales: An adaptive coarse-to-fine strategy. *International Journal of Computer Vision* **6**(2), 133-145.

3 Beauchemin, S.S. and Barron, J.L., 1995. Computation of optical flow. *ACM Computing Surveys* **27**(3), 433-467.

4 Black, M.J. and P. Anadan, 1996. The robust estimation of multiple motions: parametric and piecewise-smooth flow fields. *Computer Vision and Image Processing* **63**(1), 75-104.

5 Chivers, K.F., 1997. Surface strain mapping of heterogeneous materials. CPGS Report, Department of Materials Science and Metallurgy, University of Cambridge.

6 Clocksin, W.F., 1997. A new method for estimating optical flow. Technical Report 436, Computer Laboratory, University of Cambridge.

7 Fan, C.M., Namazi, N.M. and Penafiel, P.B., 1996. A new image motion estimation algorithm based on the EM technique. *IEEE Transactions on Pattern Analysis and Machine Intelligence,* **18**(3), 348-352.

8 Fleet, D.J. and Jepson, A.D., 1990. Computation of component image velocity from local phase information. *International Journal of Computer Vision* **5**(1), 77-104.

9 Hayton, P.M., Brady, M., Smith, S.M. and Moore, N., 1999. A non-rigid registration algorithm for dynamic breast MR images. *Artificial Intelligence* **114**, 125-156.

10 Horn, B.K.P. and Schunck, B.G., 1981. Determining optical flow. *Artificial Intelligence* **17**, 185-204.

11 Scott, G.L., 1987. 'Four-line' method of locally estimating optic flow. *Image and Vision Computing* **5**(2), 67-72.

12 Simoncelli, E.P, E.H. Adelson and D.J. Heeger, 1991. Probability distributions of optical flow. *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, June, Mauii, Hawaii, 310-315.

13 Sun, J.H., Yates, D.A. and Winterbone, D.E., 1996. Measurement of the flow field in a diesel engine combustion chamber after combustion by cross-correlation of high-speed photographs. *Experiments in Fluids* **20**, 335-345.

14 Wu, Q.X., 1995. A correlation-relaxation-labeling framework for computing optical flow -- template matching from a new perspective. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **17**(8), 843-853.