Technical Report

UCAM-CL-TR-436
ISSN 1476-2986

Number 436

UNIVERSITY OF CAMBRIDGE

Computer Laboratory

A new method for estimating optical flow

W.F. Clocksin

November 1997

15 JJ Thomson Avenue
Cambridge CB3 0FD
United Kingdom
phone +44 1223 763500

http://www.cl.cam.ac.uk/

Technical reports published by the University of Cambridge
Computer Laboratory are freely available via the Internet:

http://www.cl.cam.ac.uk/techreports/

ISSN 1476-2986

A New Method for Estimating Optical Flow

W.F. Clocksin

University of Cambridge

Computer Laboratory

Pembroke Street

Cambridge CB2 3QG, U.K.

Internet: wfc@CL.cam.ac.uk

WWW: http://www.CL.cam.ac.uk/users/wfc/

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 weighted mean near the mode of the velocity distribution function.

The method is demonstrated using image sequences obtained from the analysis of ceramic and metal materials under stress. The performance of the system under degenerate conditions is also analysed to provide insight into the behaviour of optical flow methods in general.

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. In the recent comprehensive survey by Beauchemin and Barron [2], at least 136 of the references concerned the computation of optical flow, and yet this list was not exhaustive.

There are two main methods for computing optical flow: gradient-based [3] and correlation-based [5]. Apart from these, there are also some methods involving the statistical estimation of motion parameters [3] and the use of phase information [4]. Gradient-based methods make the initial assumption that the intensities of local time-varying image regions are approximately constant for at least a short duration. Using the notation of Beauchemin and Barron, if $I(\mathbf{x}, t)$ is the image intensity function, then

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

where $\delta\mathbf{x}$ is the displacement of the local image region at $(\mathbf{x}, t)$ after time δt. Gradient-based methods for estimating optical flow usually follow Horn and Schunk [4] by expanding the left hand side in a Taylor series,

$$I(\mathbf{x}, t) = I(\mathbf{x}, t) + \nabla I \cdot \delta\mathbf{x} + \delta t I_t + O^2$$

where $\nabla I = (I_x, I_y)$ and I_t are the first order partial derivatives of $I(\mathbf{x},t)$, and the higher order terms are considered negligible. Neglecting higher-order terms yields

$$\nabla I \cdot \mathbf{v} + I_t = 0$$

where $\mathbf{v} = (u,v)$ is the velocity of the local image region. Although the two components of the velocity are constrained by one equation, this can be used to estimate $\mathbf{v}_\perp$, the motion component in the direction of the local gradient of the image intensity function. Because the normal velocity is in the direction of the spatial gradient ∇I, we may write

$$\mathbf{v}_\perp = \frac{-I_t \nabla I}{\|\nabla I\|^2} .$$

Thus, measuring spatiotemporal derivatives allows normal image velocity to be estimated.

Gradient-based methods are accurate, in general, only when the intensity conservation assumption (Eq. 1) is holds, and when frame-to-frame displacements due to motion are a fraction of a pixel so that the Taylor series approximation is meaningful. Furthermore, second-order differential methods make the assumption that first-order deformations of intensity (due to rotation or dilation) are not present. There are also problems with accuracy in determining second order derivatives due to the sensitivity of numerical integration. In practice, 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].

Correlation-based 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 [6]. Correlation-based methods have difficulty with subpixel displacements as their error depends on the proximity of image displacement to an integer number of pixels [1].

While the correlation surface computed by correlation-based methods is intrinsically multi-modal, and thus capable of representing multiple motions, the gradient-based and statistical estimation methods assume that locally there is only one coherent motion, and thus 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 studies referred to have shown, this is almost never the correct response.

Correlation-based methods have been useful in sequences where the assumptions required for gradient-based methods do not apply, for example in cloud [7] and combustion [6] images. The method proposed in the present paper 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 sub-pixel accuracy is required and large displacements and local distortions are encountered.

2. Method

The proposed method uses the fundamental assumption of Equation 1, 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 shall assume that each $\mathbf{x}$ refers to a single pixel. Given pixel values $p_1 = I(\mathbf{x}, t)$ and $p_2 = I(\mathbf{x}+\delta\mathbf{x},\ t+\delta t)$, let $P(p_1 \rightarrow p_2)$ denote the probability that the observation of p_1 and p_2 is due to the velocity $\mathbf{v} = \delta\mathbf{x}/\delta t$. A working definition for P is:

$$P(p_1 \rightarrow p_2) = \exp\left(-\frac{(p_1 - p_2)^2}{\alpha}\right). \qquad (2)$$

The probability is unity when p_1 and p_2 have identical 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 levels in the image.

It is now necessary to provide support for P to model the velocity structure of the image sequence in a given region $I(\mathbf{x}, t)$. 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. 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

$$R = \{\delta\mathbf{n} = (n_1, n_2) \mid -s \le n_1 < s,\ -s \le n_2 < s\}$$

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:

$$R = \{\delta\mathbf{n} = (n_1, n_2) \mid n_1^2 + n_2^2 \le r^2\}.$$

We now define the velocity distribution function of P as

$$F(\delta\mathbf{x}) = \frac{1}{|R|} \sum_{\substack{\delta\mathbf{a}\in R, \delta\mathbf{b}\in R, \\ \text{such that } \delta\mathbf{b}-\delta\mathbf{a}=\delta\mathbf{x}}} P(I(\mathbf{x}+\delta\mathbf{a}, t) \rightarrow I(\mathbf{x}+\delta\mathbf{b}, t+\delta t)). \qquad (3)$$

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 $\delta\mathbf{x}$. This vote is accumulated by summing $P(p_1 \rightarrow p_2)$ 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 $\operatorname{argmax}_{\delta\mathbf{x}} F(\delta\mathbf{x})$, although in the implemented version a subpixel resolution is obtained by finding a weighted mean in the region of $\operatorname{argmax}_{\delta\mathbf{x}} F(\delta\mathbf{x})$. It is necessary to compute F at every location $\mathbf{x}$ for which a motion estimate is required.

F can be computed as suggested by the following examples (omitting normalisation). Suppose `I[x,y,t]` is the pixel value the neighbourhood is 32×32 pixels square, and δt = 1 frame. The velocity distribution function for `I[x,y,t]` can be accumulated in a 32×32 histogram `F` by an algorithm expressed in the following pseudocode (in which negative subscripts are permitted for `F`):

```
for (i1 = -16 while i1 < 16 step 1)
  for (j1 = -16 while j1 < 16 step 1)
    for (i2 = -16 while i2 < 16 step 1)
      for (j2 = -16 while j2 < 16 step 1)
        d = I[x+i1,y+j1,t] - I[x+i2,y+j2,t+1];
        dx = i2 - i1;
        dy = j2 - j1
        F[dx,dy] = F[dx,dy] + exp(d * d / -a);
```

Of course, a more general implementation will store the n neighborhood offsets in a pair of $1 \times n$ arrays Nx and Ny. This is useful when neighbourhoods are arbitrary collections of pixel offsets such as those contained within a disc of a given radius.

```
for (i = 0 while i < n; step 1)
  p1 = I[x+Nx[i], y+Ny[i], t];
  for (j = 0 while j < n; step 1)
    d = p1 - I[x+Nx[j], y+Ny[j], t+1];
    dx = Nx[j] - Nx[i];
    dy = Ny[j] - Ny[i];
    F[dx,dy] = F[dx,dy] + exp(d * d / -a);
```

Another optimisation is to predefine a 256×256 array g containing the exponentiall function as a function of the two pixel values as

$$g[i,j] = \exp\left(-\frac{(i-j)^2}{\alpha}\right)$$

so that the incrementing line in the pseudocode then can be written

```
F[dx,dy] = F[dx,dy] + g[I[x+i1,y+j1,t], I[x+i2,y+j2,t+1]].
```

Before presenting the results of using this method, it is necessary to make a refinement of the velocity distribution function. The voting system of Equation 3 is biassed 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

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

votes independent of the pixel values (V may be calculated as a histogram in a way similar to the above code). The pixel values will determine the amount of vote according to Eq 2. Thus, a uniform random vote has the effect of biassing 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 biassed 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.

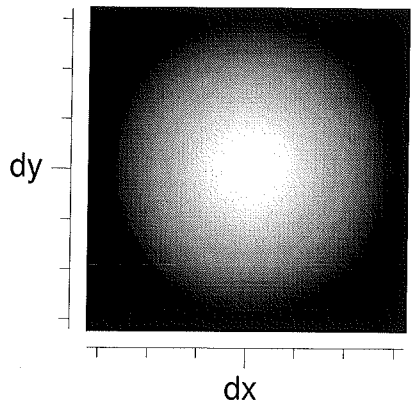


Figure 1

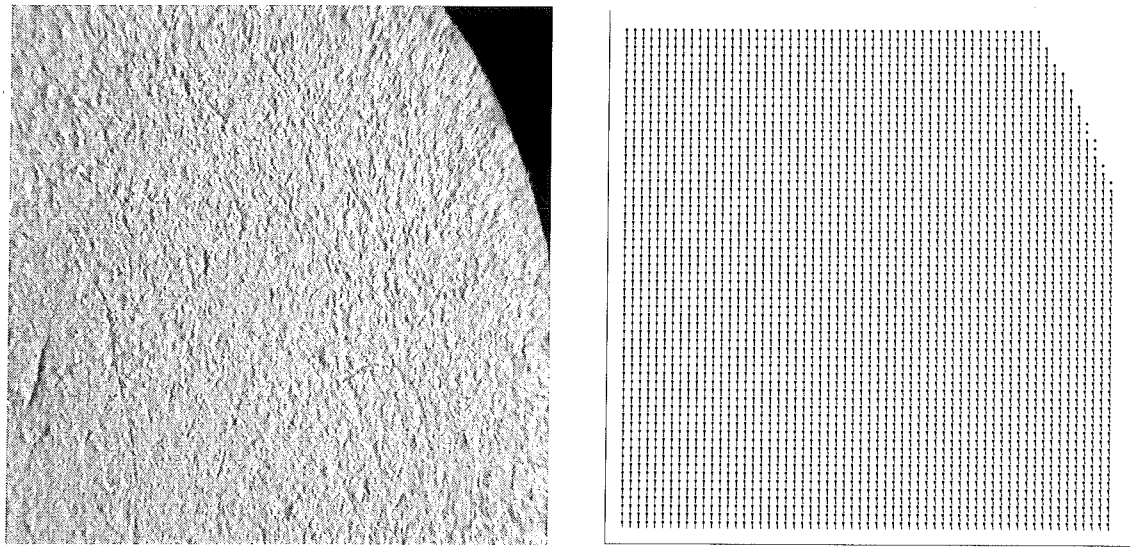
The effects of bias may be corrected by estimating the response due to bias alone, and subtracting this from Eq. 3. The response due to bias may be modelled by multiplying V by the probability that given pixel values are drawn from the image. Equation 2 assumes that pixel values are drawn uniformly from 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 $f_t(i)$ the frequency of pixel value i in the image at time (frame) t, estimate the discrete pdf $h_t(i)$ 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

$$F(\delta\mathbf{x}) = \left(\sum_{\substack{\mathbf{a} \in R,\, \mathbf{b} \in R, \\ \text{such that } \mathbf{b}-\mathbf{a}=\delta\mathbf{x}}} P(p_1 \rightarrow p_2) \right) - h_t(p_1) h_{t+\delta t}(p_2) P(p_1 \rightarrow p_2) V(\delta\mathbf{x}) \,. \quad (5)$$

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 term of Eq. 5 (the bias correction term) is precomputed from the histogram of each image in the sequence, and is used for each computation of F. 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 closeup 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 ($\mathrm{p\,f^{-1}}$) to 19 $\mathrm{p\,f^{-1}}$. 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 stretched. 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.



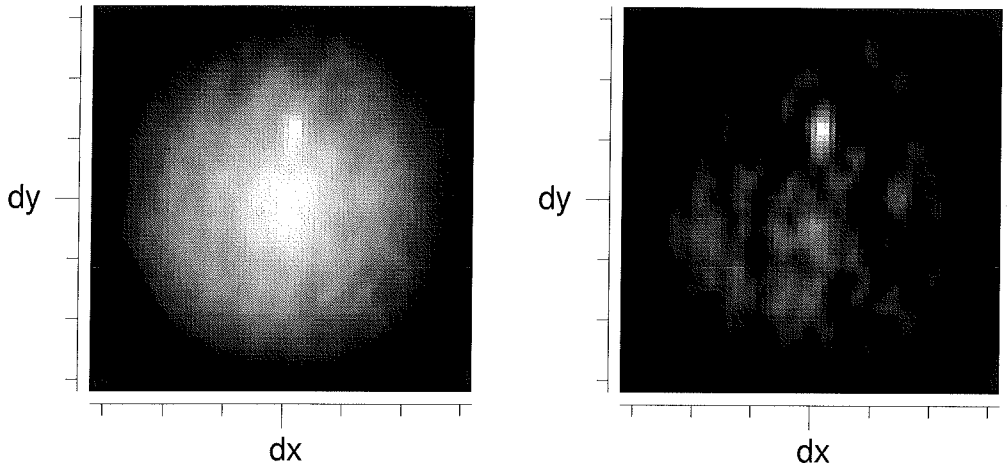
(a) (b)

Figure 2

In Fig. 2(b) the flow is clearly defined except for a few isolated locations, and a few aliassing errors near the light/dark boundary in the upper right quadrant of the image. These will be discussed later. Note that no result is available for some points in a region in the upper right corner of the image. The method does not return a result if there is no maximum value in the F plot. This is a crude way to provide a confidence in the result, but does not work in all cases. A better way would be to compare the normalised maximum value of the F plot with the probability p of drawing a pixel pair from the image. If the maximum value is less than some function of p, then no result would be returned. This is a simple yet justifiable way of providing an estimate of confidence in the result. Confidence

in the result may also be obtained by observing the dispersion of the plot; this is discussed later.

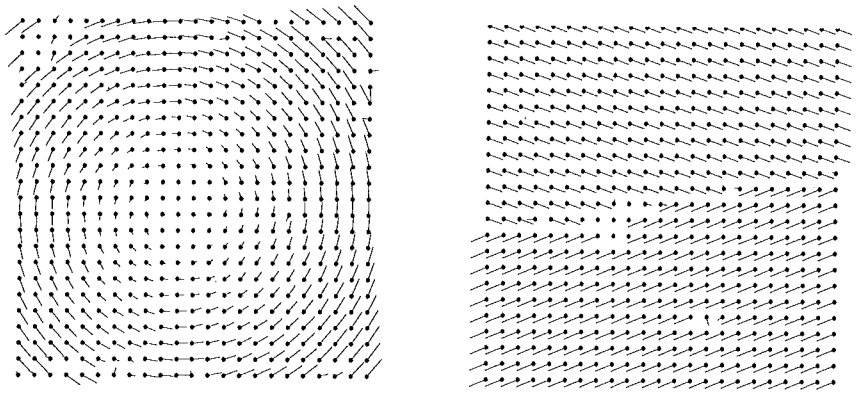
Figure 3 shows the values of F over the range of displacements for one pixel near the centre of the image in Fig. 2(a). The axes of each F plot are labelled such that dx, dy corresponds to $\delta\mathbf{x}$. The zero value is in the middle of each axis (*i.e.* $\delta\mathbf{x}=\mathbf{0}$ is in the centre of the F plot). The graduations are in units of 10 pixels displacement. In Figure 3(a) the maximum value of F is obtained at $\delta\mathbf{x}=(1.06, -11.05)$ by a weighted average about the mode. 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 $\delta\mathbf{x}=(1.11, -11.12)$ 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

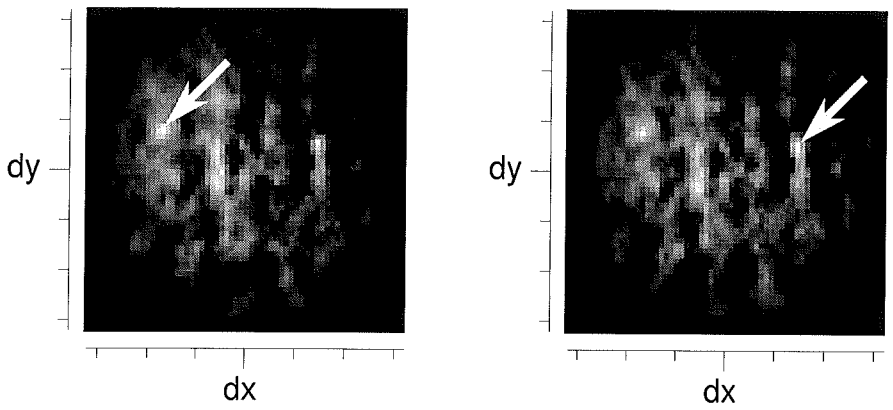
The flow fields in Figure 4 show examples of rotation (approx 0.06π radf^{-1}) and shear. Each was generated by applying the algorithm to every eighth pixel position on a pair of 256×256 pixel images of a surface similar to Figure 2. The rotation field in 2(a) was obtained by rotating the material in 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.



(a) (b)

Figure 4

In the shear field, the flow is clearly defined near the boundary except for a group of six patches. A better illustration of what happens at motion boundaries is as follows. The pair of images in Figure 5 show 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).



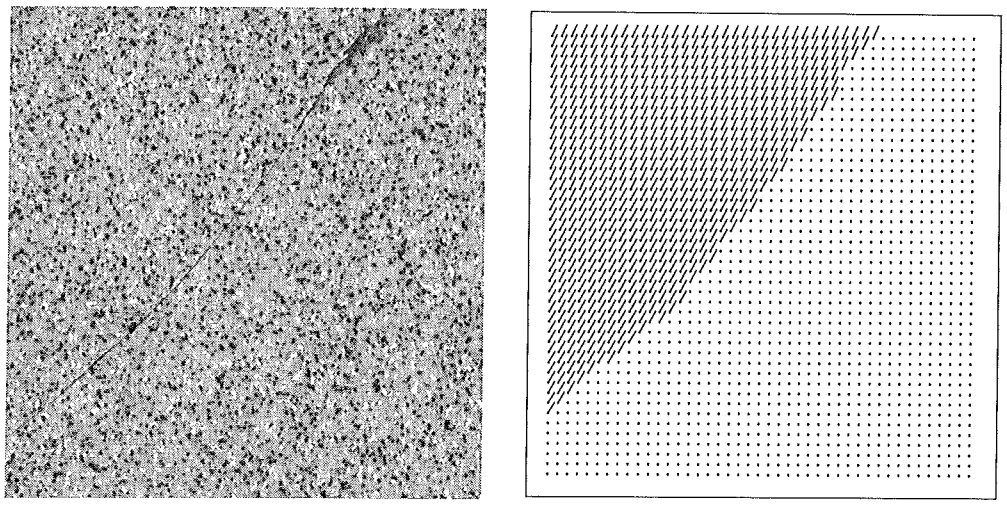
(a) (b)

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

Figure 5

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)

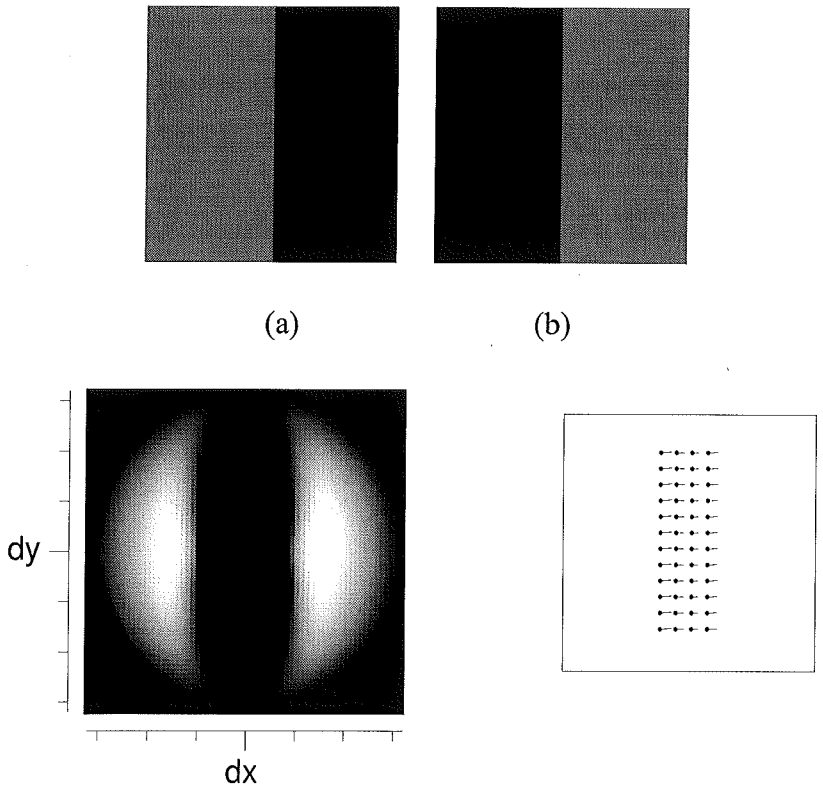
Figure 6

4. Discussion

There are three main data conditions that affect the accuracy of the method. These are ambiguity, aliassing, and the so-called aperture effect. The aperture effect is widely discussed in the optical flow literature, although the other two conditions have been less widely acknowledged. The vulnerability of a 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. In this section I discuss the sensitivity of the method to these conditions, illustrating with pathological conditions.

4.1 Ambiguity

Ambiguity is depicted in the image sequence of Figure 7(a,b). This sequence may represent a motion of a region of lighter pixels moving right, or a region of darker pixels moving left, or both. Figure 7(c) shows an F plot for a pixel on the light/dark boundary. The ambiguity is revealed by the bimodal form of F. The precise shape of the twin peaks is a result of the bias-corrected voting taking place over the patch size, a disc of radius 16.



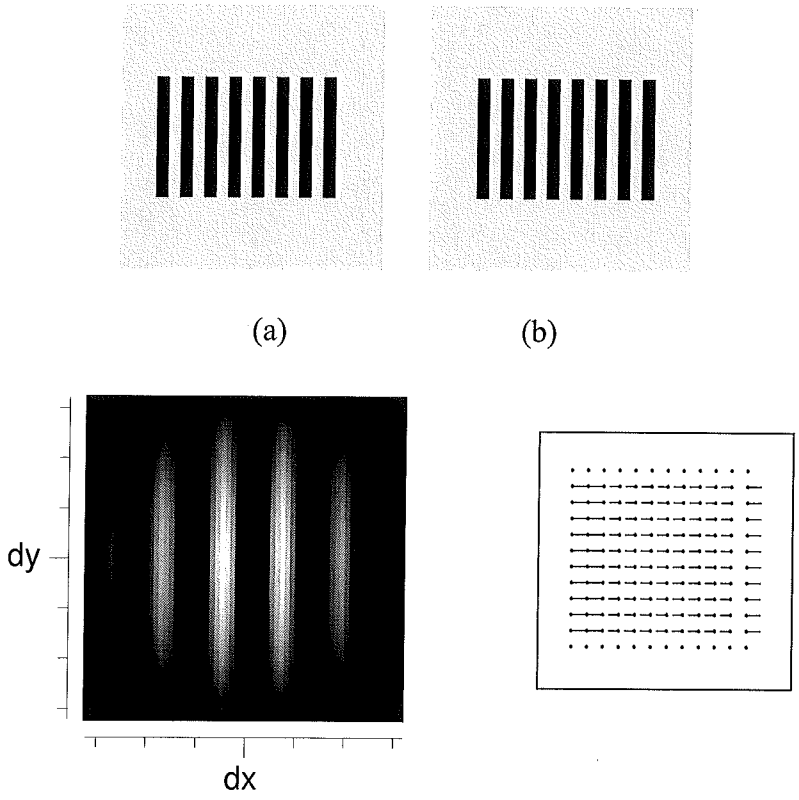
The flow vectors shown in (d) have been scaled by 0.2 for legibility.

Figure 7

The flow vectors in Fig. 7(d) are probably not meaningful, as suggested by the large variance in the F plot of Fig 7(c).

4.2 Aliassing

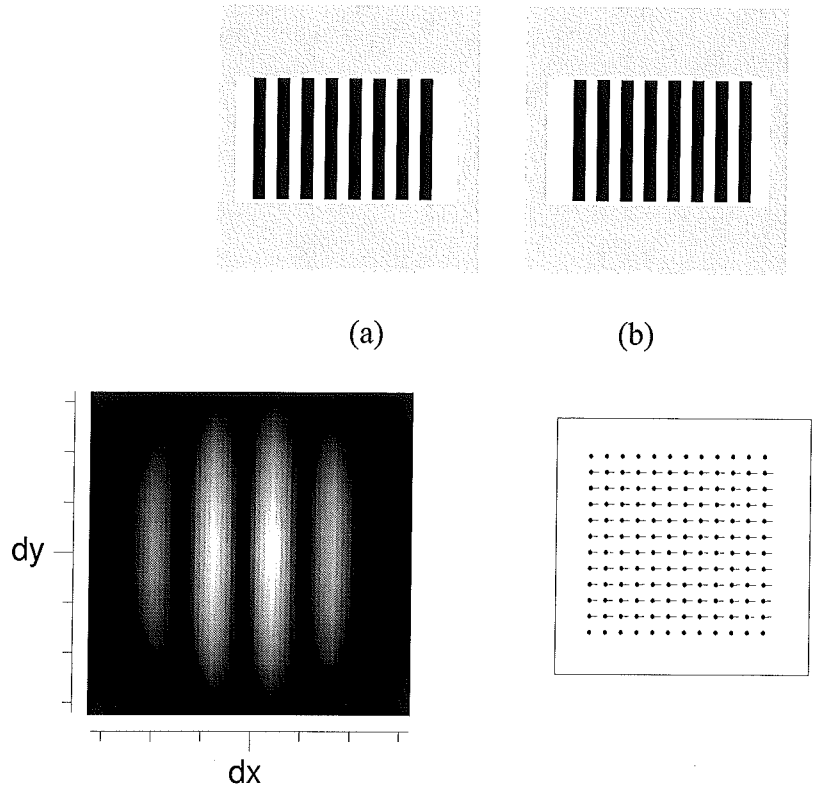
Aliassing is illustrated by the image sequence in Figure 8. The vertical bars are w pixels wide, separated by a distance $2w$, and move to the right by $w\,\mathrm{pf}^{-1}$. Here w=6. Figure 8(c) is an F plot of a point in the centre of the image. Alias peaks in the F plot are separated by $\pm\delta w$ dx. Again the amplitude of the F plot peaks is determined by voting within the 16-pixel diameter disc.



The flow vectors in (d) have been magnitude scaled by 0.6 for legibility.

Figure 8

Figure 8(d) shows the maximum values of F sampled every 8 pixels in both directions. Interestingly, the field shows a drift to the left, in the opposite direction of bar drift. This also can be discerned by the slightly higher amplitude of the dx=–6 peak in Figure 8(c). In this situation, the interpretation is that the inter-bar regions are moving to the left at a velocity of 6 pf^{-1}. The reason is that the surround has the same colour as the inter-bar regions, and so additional support is lent by the surround because figure and ground are not otherwise labelled. This explanation is confirmed by the sequence in Figure 9, in which the inter-bar regions are coloured to distinguish them from the surround, which removes the spurious support.



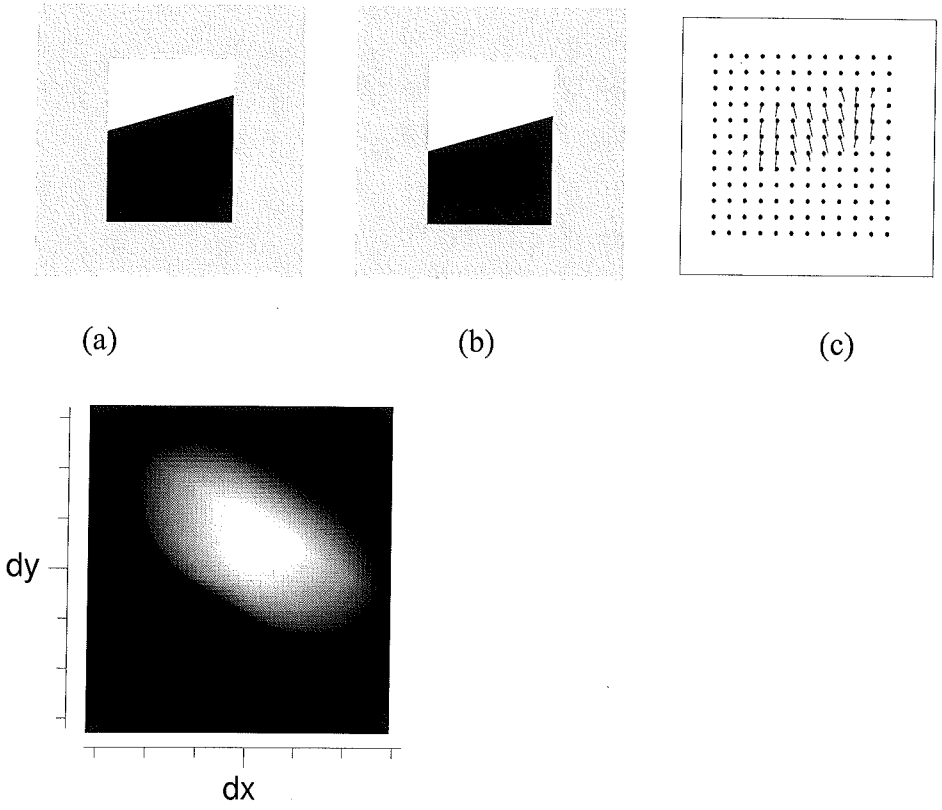
The flow vectors in (d) are unscaled.

Figure 9

So here the 'correct' behaviour is obtained by modifying the pathology of the data.

4.3 Aperture Effect

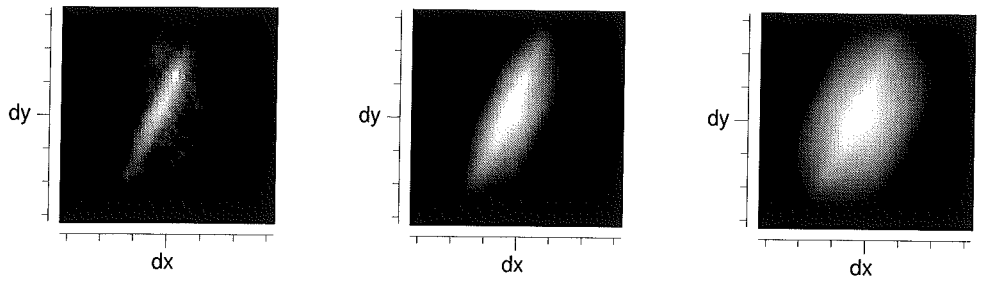
The well known aperture effect is illustrated by the image sequence in Figure 10(a,b). This is ambiguous because the direction of motion of the light/dark boundary is undefined. The same image sequence could be caused by the light (or dark) region moving vertically down, or down and to the right, or to the right.



The flow vectors in (c) are unscaled.

Figure 10

The resulting flow diagram in Fig. 10(c) shows the method's attempt to deal with this. 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. This results is not usable as it stands. However, Fig. 10(d) shows that the extremely large variance of the F plots for any of the nonzero vectors immediately betrays the dubiousness of the result. Similar situations are seen in Figure 11, which shows the F plots for selected pixels near the light/dark boundary in the upper right quadrant of Figure 2(a).



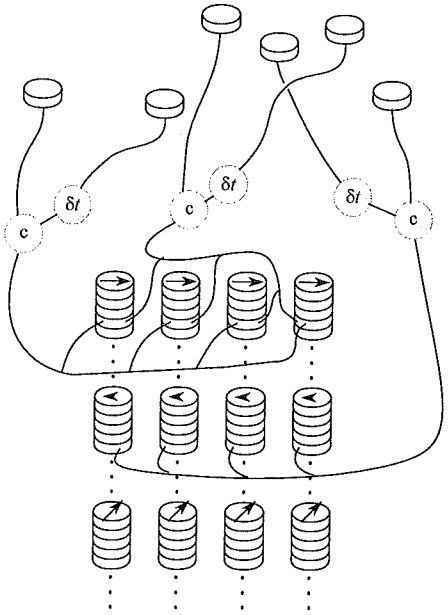
(a) (b) (c)

Figure 11

Fig. 11(a) is the F plot for a location 3 pixels away from the border into the moving object. Although some background indicative of aperture effects is apparent, the small variance peak (at the correct value of $\delta\mathbf{x}$) is more than twice the height of the highest background, suggesting a reliable flow measurement. By contrast, *F* plots for locations on the boundary (Fig. 11(b)) and 3 pixels into the dark background region (Fig. 11(c)) show a large variance about the maximum, suggesting that these F plots should not be taken as evidence for a flow vector. In all the situations I have investigated (which include timelapse imagery taked 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.

4.4 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. Figure 12 shows a simple asychronous model in which *F* plots have a columnar organisation, with pairwise comparisons being channeled 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.



There are three levels of cells: the light sensitive input cells at the top of the illustration, the computation cells that perform time delay (marked δt) and comparison (marked c) functions, and the summing cells organised into columns. Each column of summing cells implements the displacement distribution function for one receptive field. Within a receptive field, the response from pairs of cells projects via the computation cells to the summing cell representing the spatial displacement between the pair of input cells. Because receptive fields overlap, the response from a given pair of input cells may project to a summing cell in more than one column. It is assumed that summing cells leak with a time constant that can be related to the temporal discrimination of the system.

Figure 12

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 time is too large.

5. Conclusions

The approach proposed 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.

The proposed method shares some similarities with correlation-based approaches. Like correlation approaches, a multi-modal response surface is computed, but unlike previous correlation approaches, it accumulates the probability of pairwise observations instead of matching templates or features. However, because pairwise observations are performed rather than the comparison of image patches, performance is better than correlation-based approaches when first-order deformations (e.g. rotation and dilation) within the image are encountered. This property not only makes the method feasible for the test application of surface strain mapping of heterogeneous materials, but also leads to a more reliable method for estimating optical flow. The use of a multi-modal response surface with a noise model leads naturally to the derivation of a confidence factor, which is useful in disambiguating 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 BVS 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 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.

5 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.

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

7 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.

8 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.