Research report 190

DERIVING OPTICAL FLOW IN NOISY IMAGE SEQUENCES

WOOI BOON GOH
GRAHAM R. MARTIN

A technique is presented extracting local image motion utilising the properties of spatiotemporal orientation in the frequency domain. This technique is based on the eigenvalue analysis of the inertia tensor matrix of the frequency domain. An iterative velocity field smoothing algorithm was developed based on the properties of a variant of this inertia tensor matrix. This iterative algorithm is able to smooth noisy translational, rotational and other smoothly varying velocity flow fields, within the constraints of motion discontinuities. Some results from the application of this algorithm to noisy random dot sequences are presented.

Department of Computer Science
University of Warwick
Coventry CV4 7AL
United Kingdom

July 1991

CONTENTS

Deriving Optical Flow in Noisy Image Sequences

W. Boon Goh, Graham R. Martin
Dept. of Computer Science, University of Warwick, Coventry.
July 1991

1. Introduction

One of the key challenges in image sequence analysis is the estimation of instantaneous velocity vectors for each small region on an image plane. The derivation of this two dimensional velocity vector field, commonly termed 'optical flow' [*Horn & Shunck '81*], represents the first stage in many image sequence analysis problems, like extracting structure from motion, object tracking and image compression.

Braddick suggested two different motion mechanisms used by the human visual system; the long range and short range mechanisms [*Braddick '74*]. The long range mechanism operates over a larger spatial extent and is generally believed to be based on identifying and tracking features. The short range mechanism however, operates over a shorter spatial distances and involves lower level visual information such as image intensity. Correspondingly, over the years, two distinct approaches have been developed for computing motion from image sequences [*Aggarwal & Nandhakumar '88*]. They are broadly classified under feature-based methods and gradient-based methods. The former attempts to estimate image motion by establishing inter-frame correspondence between distinct features like corners and edges. Whereas the latter computes it directly from the grey-level values in the image sequence.

In this report, we will describe a method for estimating local image velocity which has characteristics similar to the short range motion mechanism. This method adopts a spatiotemporal gradient approach which utilises the frequency domain properties of several image frames that are closely sampled in time. We will show how local image velocity could be extracted by finding a tilted plane that fits the frequency power spectra in a least squared error manner.

An iterative velocity smoothing algorithm is also described. This algorithm is able to smooth noisy velocity flow fields while taking into account the presence of motion discontinuities. Finally, we present some results obtained from the application of this cooperative motion boundary detection and velocity field smoothing algorithm on noisy random dot image sequences.

2. Representation of Motion

2.1 Motion as Orientation

Image motion is the result of the translation of brightness patterns across the image plane. Assuming the changes in image intensity values are due solely to the translation of these brightness patterns over time, we have :

$$f(x, y, t) = f(x - r_x t, y - r_y t, t) \quad (1)$$

where r_x and r_y are the horizontal and vertical components of the image velocity **r.**

This change in intensity is characterised by a 3-dimensional orientation in x-y-t space [*Adelson & Bergen '85*] . Fig 1a shows a bar which moves to the left with time. In x-y-t space it forms a volume slanted in the temporal axis and the degree of slant corresponds to the velocity of the bar. The task of motion detection could thus be viewed as a problem of detecting spatiotemporal orientation.

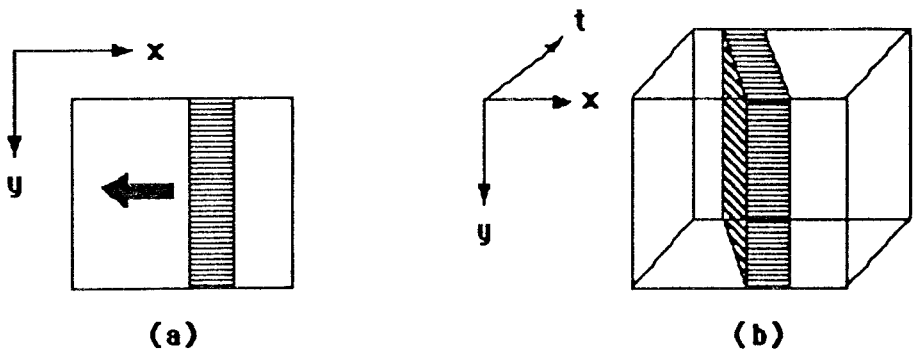


Fig. 1 - Image motion represented as spatiotemporal orientation in 3-dimensional x-y-t space

2.2 Motion in the Frequency Domain

Spatiotemporal orientation has a simple representation in the frequency domain [*Watson & Ahumada '85*]. Taking the Fourier transform of equation (1) gives :

$$\mathcal{F}\, f(x - r_x t, y - r_y t, t) = F(u, v, w + r_x u + r_y v) \quad (2)$$

Each temporal frequency is shifted by the amount - $(r_x u + r_y v)$, proportional to the local image velocity.

Fig. 2a shows the spectrum of a stationary pattern, where energy is concentrated in the uv-plane. As the pattern translates (Fig. 2b), the spectrum tilts into the temporal axis (w-axis), the larger the angle θ, the larger the velocity. Fig. 2c illustrates the case when there are both horizontal and vertical velocity components, where the horizontal component r_x is given by tan (α) and the vertical component r_y is given by tan(β). Finally, Fig. 2d illustrates the spectrum of a moving edge. The edge, being one dimensional, produces a one dimensional spectrum which is concentrated about a line through the origin. The plane cannot be determined unambiguously, hence only the velocity component orthogonal to the edge can be found. This is commonly termed the aperture problem [*Marr & Ullman '81*].

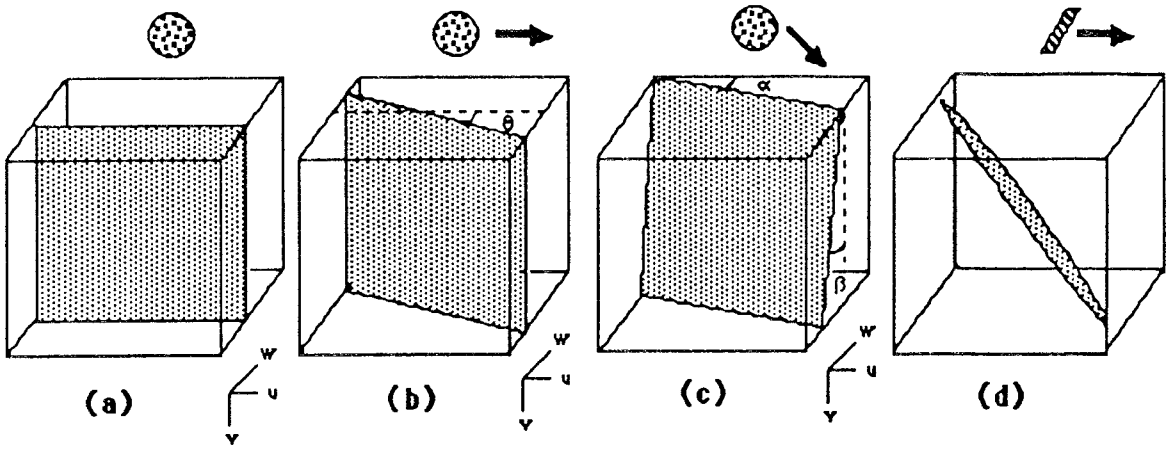


Fig. 2 - Representation of motion in the frequency domain (see text).

3. Extracting Image Motion

For the purposes of deriving an optical flow field, we require only the local translational information. This is because non-rigid motions, rotation and scaling produce a varying two dimensional flow field except at relatively small image regions. By taking a small spatiotemporal window at each of these image regions and fitting a plane to each of their frequency spectra, we can estimate the true velocity flow field of the image sequence.

3.1 Spatiotemporal Orientation Detection

There have been several methods proposed for extracting spatiotemporal orientation from a sequence of images. One method is the use of spatiotemporal filters [*Heeger '87*]. Heeger used a family of Gabor-energy filters that were tuned to different spatial orientations and temporal frequencies. By combining the responses of each of these filters in a least square manner and normalising the response from each spatial orientation separately, he was able to resolve the aperture problem and extract the unambiguous velocity for a great variety of highly textured translating patterns.

Bigun and Granlund proposed a method for determining local orientation in a multidimensional space. Their approach is analogous to the problem of reducing three dimensional tensors to their principal axes, but applied to the spectral energy in the Fourier domain [*Bigun & Granlund '87*].

As shown earlier in Fig. 2d, the spectral energy of a linearly moving edge is always concentrated on a line through the origin in the Fourier domain. If we consider the spectral energy as masses rotating about the axis $\mathbf{k}_p$, then the inertia about the axis $\mathbf{k}_p$ is given by :

$$I(\mathbf{k}_p) = \int_{-\infty}^{\infty} d^2(\mathbf{k},\mathbf{k}_p) \mid F(\mathbf{k}) \mid^2 d\mathbf{k} \tag{3}$$

where $d(\mathbf{k},\mathbf{k}_p)$ is the Euclidean distance function between each position in the Fourier space and the line represented by the axis $\mathbf{k}_p$. (To simplify expressions, the x-y-t space will be represented by the vector $\mathbf{x}$ and the u-v-w frequency space will be represented by the vector $\mathbf{k}$).

The problem here is to find the axis $\mathbf{k}_p$ that minimises the inertia function $I(\mathbf{k}_p)$. This could be solved by formulating an eigenvalue problem given by

$$I(\mathbf{k}_p) = \mathbf{k}_p^{\,t} \, \mathbf{J} \, \mathbf{k}_p \tag{4}$$

where $|\mathbf{k}_p| = 1$ and is orthogonal. $\mathbf{J}$ is the inertia tensor matrix in (3) whose diagonal elements are

$$J_{ii} = \sum_{j \neq i} \int_{-\infty}^{\infty} k_j^2 \mid F(\mathbf{k}) \mid^2 d\mathbf{k} \tag{5}$$

and non-diagonal elements are

$$J_{ij} = - \int_{-\infty}^{\infty} k_i \, k_j \mid F(\mathbf{k}) \mid^2 d\mathbf{k} \tag{6}$$

Finding the minimum inertia $I(\mathbf{k}_p)$, corresponds to finding the axis $\mathbf{k}_p$ that fits the spectral energy in the least squared error sense. This is given by the eigenvector of the least eigenvalue of $\mathbf{J}$.

It is possible to solve this eigenvalue problem in the spatial domain. Utilising the differentiation properties of the Fouries transform , it can be shown that [*Bracewell '86*] :

$$k_j^2 \, | \, F(\mathbf{k}) \, |^2 \; = - \frac{1}{4\pi^2} \left(\frac{\delta f}{\delta x_j}\right)^2 \qquad \text{and} \qquad k_i \, k_j \, | \, F(\mathbf{k}) \, |^2 \; = - \frac{1}{4\pi^2} \left(\frac{\delta f}{\delta x_i} \, \frac{\delta f}{\delta x_j}\right) \tag{7}$$

With (7) and Parseval's theorem, we can transform the integral in equations (5) & (6) from the Fourier **k**-space to the spatial **x**-space using the inverse Fourier transform.

Furthermore, to limit the determination of image velocity to a locality at image point $\mathbf{x_0}$, a weighting function $g(\mathbf{x})$ is utilised to control the size of the effective neighbourhood. The final expression for the diagonal elements of the inertia tensor matrix are

$$J_{ii} \, (\mathbf{x_0}) = \sum_{j \neq i} \int_{-\infty}^{\infty} g(\mathbf{x}-\mathbf{x_0}) \left(\frac{\delta f}{\delta x_j}\right)^2 d\mathbf{x} \tag{8}$$

and for non-diagonal elements :

$$J_{ij} \, (\mathbf{x_0}) = \; - \int_{-\infty}^{\infty} g(\mathbf{x}-\mathbf{x_0}) \left(\frac{\delta f}{\delta x_i} \, \frac{\delta f}{\delta x_j}\right) d\mathbf{x} \tag{9}$$

The inertia tensor matrix **J** could be expressed in another form :

$$\mathbf{J} = \; \mathbf{I} \; \text{trace} \; (\mathbf{A}) - \mathbf{A} \tag{10}$$

where the trace of matrix **A** is the sum of its diagonal elements and **A** has diagonal elements :

$$A_{ii} \, (\mathbf{x_0}) = \int_{-\infty}^{\infty} g(\mathbf{x}-\mathbf{x_0}) \left(\frac{\delta f}{\delta x_i}\right)^2 d\mathbf{x} \tag{11}$$

and non-diagonal elements:

$$A_{ij} \, (\mathbf{x_0}) = \int_{-\infty}^{\infty} g(\mathbf{x}-\mathbf{x_0}) \left(\frac{\delta f}{\delta x_i} \, \frac{\delta f}{\delta x_j}\right) d\mathbf{x} \tag{12}$$

It can be shown that both the **J** and **A** matrices share the same eigenvectors [*Bigun and Granlund '88*].

Finding the eigenvector corresponding to the least eigenvalue **J** is now equivalent to finding the eigenvector corresponding to the largest eigenvalue of **A**. There are several advantages in dealing with matrix **A** instead of **J** and this will be become apparent in section 4. The elements of **A** associated with the neighbourhood of the image point $\mathbf{x}_0$ can be evaluated by simple summation of all the partial derivatives within the weighted neighbourhood. These partial derivatives being first obtained through appropriate partial derivative filters.

In this work, the weighting window $g(\mathbf{x}_0)$ was a 5x5x5 Hamming window and the partial derivative filters were designed utilising the 3x3 edge detection kernel of Wilson and Bhaleroa. [*Wilson & Bhaleroa '90*]. The eigenvalues and eigenvectors of matrix **A** were evaluated using the NAG library routine F02ABF for a symmetric matrix. This routine reduces the matrix to upper Hessenberg form and uses the QR algorithm to determine all eigenvalues and eigenvectors.

3.2 Interpreting the eigenvalues and eigenvectors

The solution to the eigenvalue problem of matrix **A** yields 3 positive eigenvalues :

$$\lambda_2 \geq \lambda_1 \geq \lambda_0 \geq 0 \qquad \text{(in descending order)}$$

and their corresponding eigenvectors $\mathbf{u}_2$, $\mathbf{u}_1$, $\mathbf{u}_0$. When the grey-levels in the image sequence are constant, we have :

$$\lambda_2 = \lambda_1 = \lambda_0 = 0 \tag{13}$$

Under these conditions, no velocity vectors are obtained. In image regions where there are moving one dimensional features like lines and edges, the spectrum is concentrated about a line through the origin given by the eigenvector $\mathbf{u}_2$, corresponding to the largest eigenvalue λ_2 (see Fig. 3a). Under these conditions :

$$\lambda_2 >>> \lambda_1 \qquad \text{and} \qquad \lambda_0 \rightarrow 0 \tag{14}$$

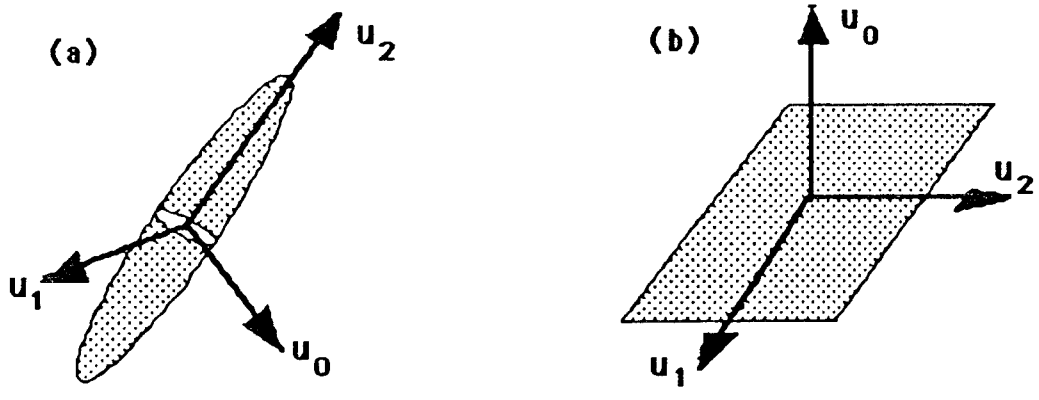


Fig. 3 - Eigenvectors corresponding to (a) a translating edge and (b) a translating 2-D pattern

In regions containing moving two dimensional features like corners and textured surfaces, the spectrum will be concentrated about a plane through the origin, the normal to which is given by the eigenvector $\mathbf{u_0}$, corresponding to the least eigenvalue λ_0 (see Fig. 3b). Under these conditions :

$$\lambda_2 \geq \lambda_1 \qquad \text{and} \qquad \lambda_0 \rightarrow 0 \tag{15}$$

Using simple vector geometry, it can be shown that the horizontal and vertical velocity components in the direction of the image gradient are given by :

$$V_x = -\frac{u_x u_t}{u_x^2 + u_y^2} \qquad \text{and} \qquad V_y = -\frac{u_y u_t}{u_x^2 + u_y^2} \tag{16}$$

where u_x, u_y, u_t are the components of the eigenvector $\mathbf{u_2}$.

This velocity cannot, under most circumstances, be regarded as the true image velocity. The true image velocity can only be extracted when the spectrum fits unambiguously about a plane and this velocity is given by :

$$V_x = \frac{u_x}{u_t} \qquad \text{and} \qquad V_y = \frac{u_y}{u_t} \tag{17}$$

where u_x, u_y, u_t are the components of the eigenvector $\mathbf{u_0}$.

Another useful measure that can be extracted from the three eigenvalues λ_2, λ_1 and λ_0 is a realibility measure of the estimated velocity at each image point. This certainty value $C(\mathbf{x_0})$, indicates how well the frequency spectrum fits a line or plane in a least squared error sense, and is given by :

$$C(\mathbf{x_0}) = 1 - \frac{\lambda_0}{\lambda_0 + \lambda_1 + \lambda_2} \tag{18}$$

The certainty measure $C(\mathbf{x_0})$ is normalised and thus takes a value between 0 and 1; the closer the value is to 1, the more reliable is the velocity estimate.

4. Cooperative boundary detection and velocity field smoothing

Studies of the human motion detection system [*Van Doorn & Koenderink '83*] suggest that high noise immunity is achieved through spatiotemporal integration of motion information over extensive spatial area and time duration. However, this integration process does not blur out the visibility of boundaries demarcating regions moving with different velocities. This implies that spatiotemporal integration of motion information is in some way guided by constraints such as the presence of motion discontinuities.

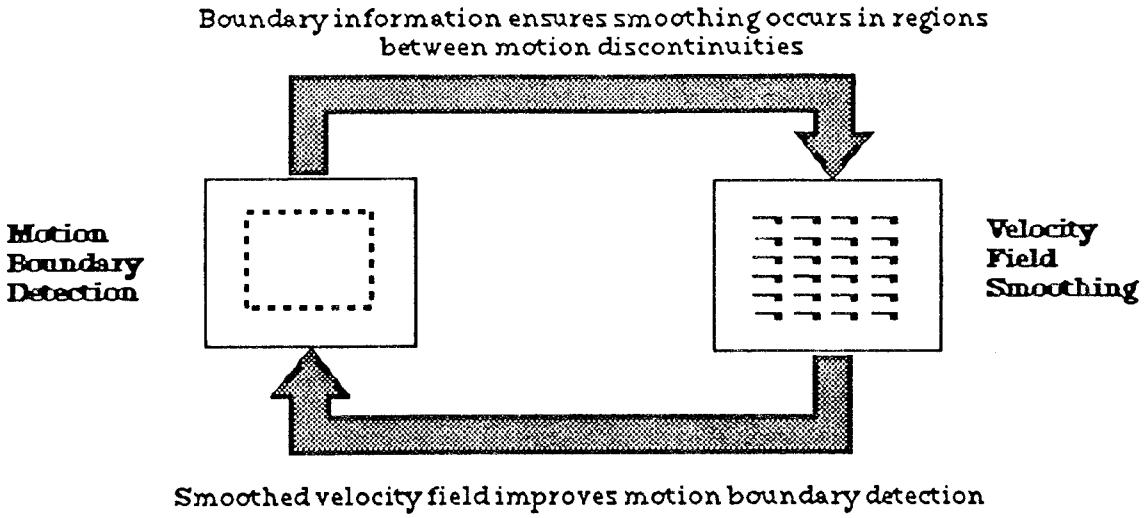


Fig. 4 - Cooperative interaction between motion boundary detection and velocity smoothing

A scheme is proposed here, as shown in Fig. 4, where both the detection of motion boundaries and the spatial integration of local velocity estimates occur in tandem. Each process interacts with the other in a manner that serves to improve both the detection of the motion boundaries and the smoothness of the velocity flow field.

4.1 Velocity field smoothing

This algorithm achieves velocity field smoothing in an iterative manner, propagating velocity information from each image locality to its spatial neighbourhood. The propagation of velocity information is weighted by its certainty measure, ensuring the suppression of poor initial velocity estimates and enhancing the influence of more reliable ones.

As shown earlier in equations (11) & (12), the matrix **A** at point $\mathbf{x}_0$ is obtained by summing all partial derivatives within its windowed neighbourhood. In order to achieve more extensive integration of motion information, this summation is extended to include the partial derivatives of the **A** matrices from the 8 image points immediately surrounding point $\mathbf{x}_0$. At the kth iteration, the summed elements of the **A** matrix at image point (x_0, y_0) is given by :

$$A_{ij}(x_0,y_0 ; k) = \sum_{n=-l}^{l} \sum_{m=-l}^{l} C^2(x_0+n,y_0+m ; k-1)\, A_{ij}(x_0+n,y_0+m ; k-1) \qquad (19)$$

The certainty measure $C(x_0+n, y_0+m ; k-1)$ for each of the **A** matrices in the neighbourhood is defined by equation (18). Its value is squared to increase the suppression of poor velocity estimates. After summation of all the neighbouring **A** matrices, each matrix element of the resulting **A** matrix is normalised appropriately as shown :

$$A_{ij}(x_0, y_0 ; k) = \frac{A_{ij}(x_0, y_0 ; k)}{\sum_{n=-l}^{l} \sum_{m=-l}^{l} C^2(x_0+n, y_0+m ; k-1)} \qquad (20)$$

With each successive iteration, the spatial extent of motion information integration is enlarged. This occurs because local velocity estimates are being progressively passed from one neighbourhood to another through the iterative neighbourhood summation process. With the use of a certainty weighting function $C(\mathbf{x}_0)$, the integration process enhances the propagation of good initial velocity estimates while suppressing the effects of unreliable ones.

Some of the results of the velocity smoothing algorithm are illustrated below. Fig. 5 shows the centre frame of the image sequence used. The sequence is corrupted with additive spatiotemporal Gaussian white noise and has a signal-to-noise ratio of 10dB.

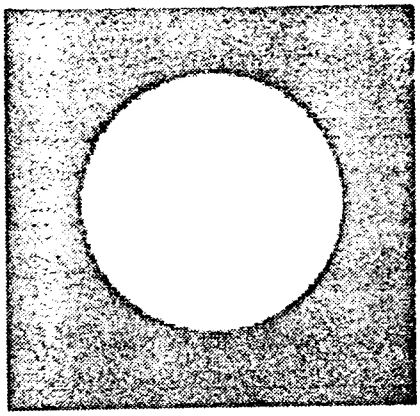
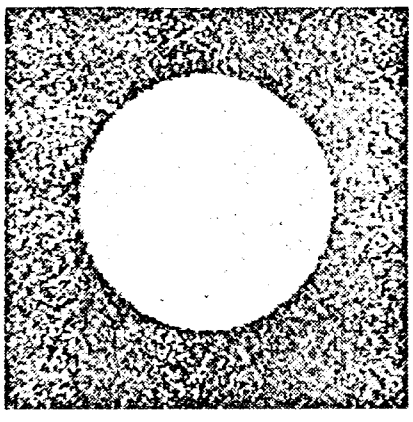
(a)  (b) 

Fig. 5 - (a) One frame from the sequence before addition of white noise, (b) same frame with SNR of 10dB.

The first sequence has the circle shown in Fig. 5b moving linearly with a velocity of 0.5 pixel/frame. The velocity estimates illustrated in the examples below are all taken from level 1 of the spatial lowpass pyramid of each frame in the image sequence [*Burt & Adelson '83*]. Fig. 6 shows the effect of the smoothing algorithm at various iterations. Good initial estimates are obtained in areas with a strong intensity gradient, such as the edges of the circle, and are progressively propagated inwards and outwards. By a combination of the propagation of strong motion energy at the circle edge and the suppression of poor velocity estimates, the random noise in the vicinity of the circle appears to move in a coherent fashion, in a similar way to that of the circle. This phenomena called motion capture [*Ramachandran & Anstis '83*] is observed by the human visual system.

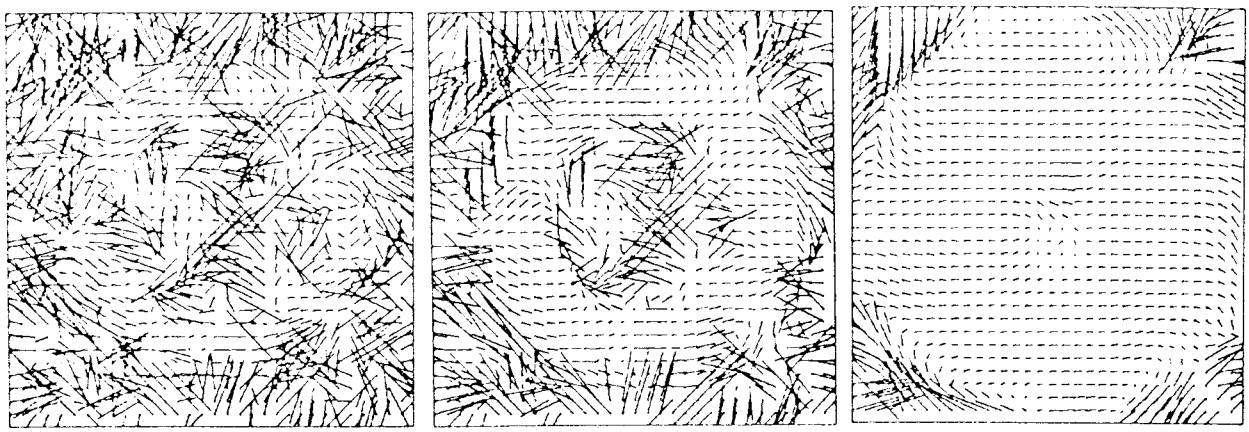


Fig. 6 - Velocity field of circle moving rightwards. (a),(b),(c) are results at iterations 1,5&20 respectively.

This smoothing algorithm is effective not only for translational motion. As shown below in Fig. 7, the radius of the circle is now expanding at 0.5 pixel/frame. As most natural velocity flow fields are smoothly varying, except at motion discontinuities, this algorithm is able to integrate neighbouring velocity estimates to produce a smooth velocity field.

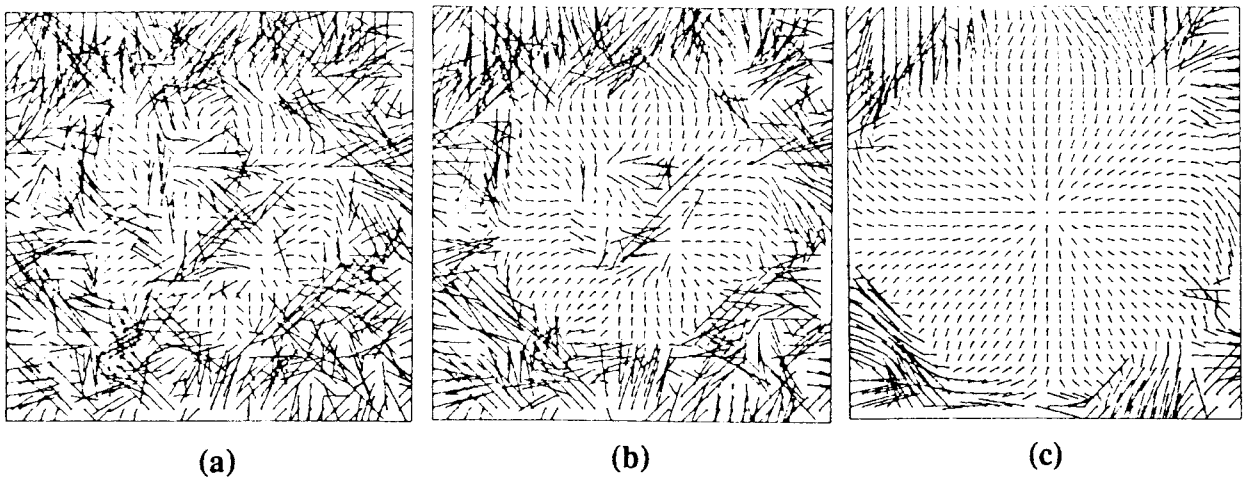


Fig. 7 - Velocity field of expanding circle. (a),(b),(c) are results at iterations 1, 5 & 20 respectively.

To avoid integration of motion information across motion discontinuities, image points that have been identified as boundary points by the motion boundary detection algorithm in section 4.3, are not considered in the neighbourhood summation process. In other words, if an image point $\mathbf{x_0}$ is surrounded by 8 neighbouring points, of which 3 are boundary points. Then the summation process in equation (19) will only involve the other 5 good neighbours and the resultant **A** matrix will be normalised accordingly.

4.2 Neighbourhood summation at boundary points

If an image point $\mathbf{x_0}$ is a boundary point, it will not undergo the certainty weighted neighbourhood summation described in equation (19). Instead, all its 8 neighbours will be summed with equal weighting irregardless of whether they are boundary points or points with good velocity estimates. The elements of the **A** matrix of a boundary point at the kth iteration is then :

$$A_{ij}(x_o,y_o ; k) = \frac{1}{9} \sum_{n=-1}^{1} \sum_{m=-1}^{1} A_{ij}(x_o+n,y_o+m ; k-1) \qquad (21)$$

If a boundary point actually lies on a motion discontinuity, the equal weighted summation of its neighbourhood will ensure its certainty measure remains poor throughout the iterative process, thus maintaining a barrier which will prevent motion information of different velocity regions from intermingling.

In cases where the boundary points are detected falsely due to the presence of noise, this summation of the neighbourhood will progressively improve its certainty measure, eventually reverting it back to an image point with a good velocity estimate.

4.3 Motion boundary detection

The motion boundary detection algorithm was inspired by the models proposed by Reichardt and Poggio from their studies of relative motion discrimination abilities in flies [*Reichardt and Poggio '79*]. They suggested that large field pool cells sum information from many small field motion detector cells and it is this summation of motion detectors over an extensive spatial area that allows the large field cells to detect the presence of motion discontinuities within that spatial extent. A similar summation scheme is used here.

The detection of boundaries between two regions of differing velocities relies on the fact that the certainty measure at these image locations is generally much poorer than in areas with smoothly varying velocity fields. By summing over a large spatial area at motion discontinuities, we would obtain a frequency spectra similar to that shown in Fig. 8. Since the spectral energy is concentrated in two different planes, attempting a least square error fit of this spectra into a single plane will result in large errors. This is shown up in the eigenvalues of the matrix **A**. Eigenvalues λ_2, λ_1 and λ_0 will all be much greater than 0, resulting in a low certainty measure.

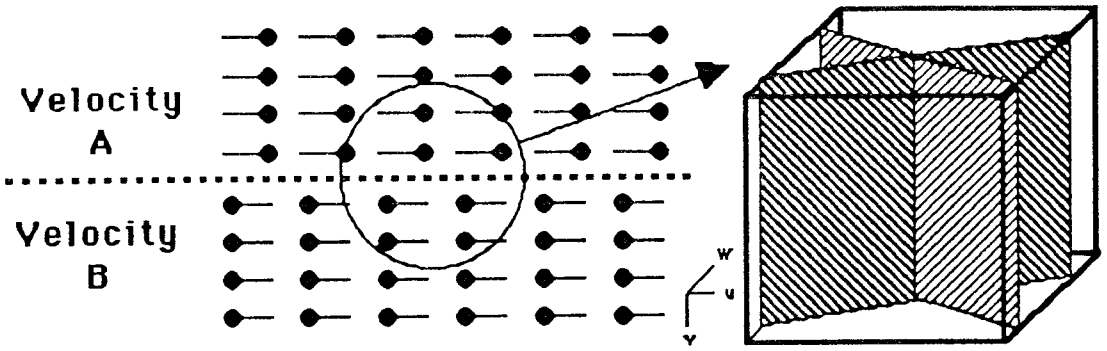


Fig. 8 - The frequency spectra at a motion discontinuity contains spectral energy that is concentrated about two different planes corresponding to two distinct velocity regions. Poor certainty measures are obtained in this region because an optimal plane fit results in large errors.

Similar neighbourhood summation of **A** matrices as in equation (19) is used here, except over a larger spatial neighbourhood. In our implementation, an 8x8 neighbourhood was used. Another variance from equation (19) is the normalisation of each element of matrix **A** by the total sum of all its elements. This reduces the bias in the frequency spectra that may arise due to a non-uniform intensity gradient within the neighbourhood.

A circular window function is used to reduce the influence of regions that are further away from the centre.

The elements of the summed matrix **A** at point (x_o, y_o) on the kth iteration is given by :

$$A_{ij}(x_o, y_o ; k) = \sum_{n=-4}^{4} \sum_{m=-4}^{4} \frac{A_{ij}(x_o+n, y_o+m ; k-1)\, w(n,m)}{N_A(x_o+n, y_o+m ; k-1)} \tag{22}$$

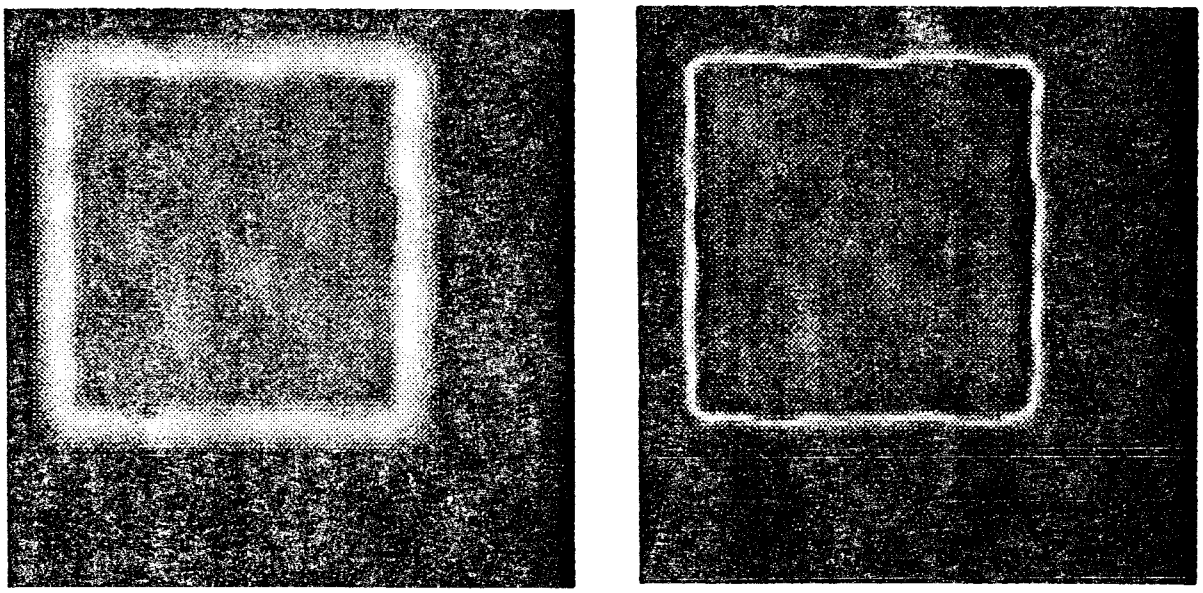
where the normalising factor :

$$N_A(x_o+n, y_o+m ; k-1) = \frac{1}{9} \sum_{i=1}^{3} \sum_{j=1}^{3} |A_{ij}(x_o+n, y_o+m ; k-1)| \tag{23}$$

and the windowing function :

$$w(n,m) = 6.66 - \sqrt{n^2 + m^2} \tag{24}$$

After neighbourhood summation of **A** matrices for every image point, their corresponding eigenvalues are evaluated and their certainty measure $C(\mathbf{x_o})$ found using equation (18). Fig. 9a shows a scaled plot of the certainty measure for a random dot image sequence. As a result of summating over a large neighbourhood, the motion boundary is fairly wide. In order to detect the motion boundaries more precisely, boundary thinning must first be performed on the certainty image in Fig. 9a.



(a) (b)

Fig. 9 - A noise free random dot image of a moving square. (a) shows a linearly scaled plot of its certainty values (scaling is between the limits of its maximun and minimum certainty value). (b) Show a boundary image obtained after performing boundary thinning on the certainty image in (a).

Boundary thinning is done in the following manner. Firstly the gradient, $C'(x_0)$ at each point of the certainty image $C(x_0)$ is obtained :

$$C'(x_0) = \sqrt{\left(\frac{\delta C(x_0)}{\delta x}\right)^2 + \left(\frac{\delta C(x_0)}{\delta y}\right)^2} \tag{25}$$

The boundary image is then obtained from the expression :

$$B(x_0) = C(x_0)\,(\,C'(x_0)_{max} - C'(x_0)\,)^2 \tag{26}$$

where $C'(x_0)_{max}$ is the largest gradient value in the certainty image and is used to scale the boundary image $B(x_0)$.

Boundary points are selected by setting a threshold value on the scaled boundary image $B(x_0)$. The selection of a suitable threshold value is not too cirtical. If the threshold is set too high, weak motion boundaries may not be detected and 'leakage' may occur between different motion regions. When neighbourhood summation is done in these regions, the resulting certainty measure of the summed **A** matrix will deteriorate (see Fig. 8). This progressive drop in the certainty value will eventually 'plug up' the leak. This is a benefit of the adaptive nature of the velocity smoothing algorithm; as the certainty measure at the kth iteration is obtained from the neighbourhood summed **A** matrix generated during the (k-1)th iteration. However, the resulting motion boundary will be slightly biased towards the region which had a poorer initial certainty measure. An effective technique to prevent leakage is to start with a low threshold value and progressively raise this threshold with each iteration.

The position of boundary points found by this algorithm is fed back to the velocity field smoothing algorithm. Locations of new boundary points are updated before proceeding to the next iteration of smoothing. Likewise, boundary points that have fallen below the new threshold value would revert back to points having good velocity estimates.

5. Experimental Results

All results presented in this section were computed using the lowest level of the spatial lowpass pyramid. Additive spatiotemporal Gaussian white noise was added to all test image sequences to obtain a signal-to-noise ratio (SNR) of 10dB. The SNR is defined by :

$$SNR = 10 \log \left(\frac{\sigma_i^2}{\sigma_n^2}\right) \tag{27}$$

where σ_i^2 and σ_n^2 are the variances of the clean image sequence and added noise respectively.

5.1 Translational flow field

A square was extracted from a static random dot image and made to translate diagonally with each consecutive image frame. The moving square's horizontal and vertical velocity components were both 0.5 pixel/frame. The velocity flow fields are shown below in Fig. 10.

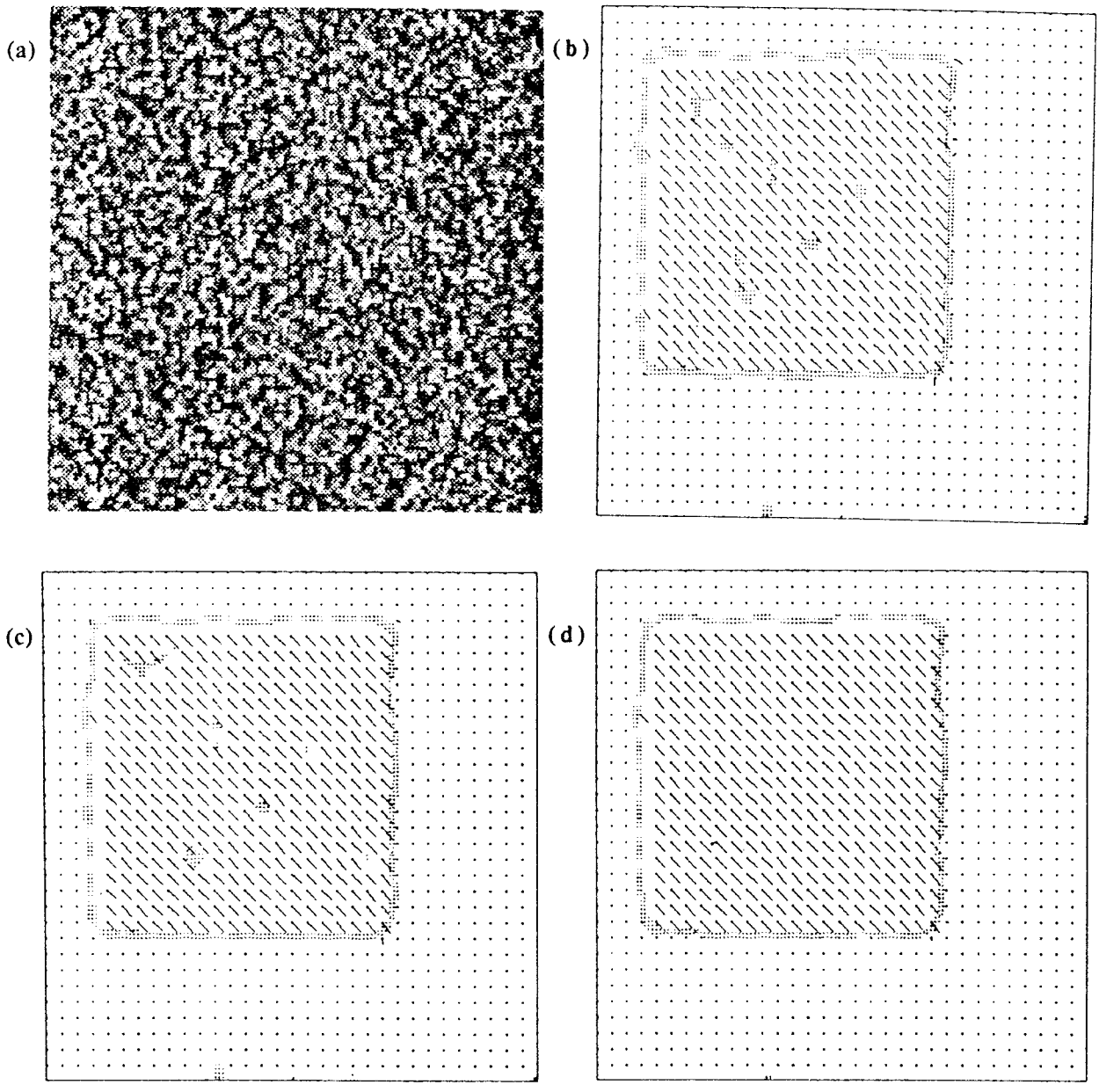


Fig. 10 - Moving random dot square. Each vector in the flow field is depicted by a line whose length and direction is proportional to the velocity estimated at that image point. (a) 1 frame from the image sequence with a SNR of 10dB. (b),(c)&(d) Velocity flow field at iterations 1,5 and 11 respectively. The boundary points seen as light grey dots reveal the structure of the moving square patch. False boundary points at image regions badly affected by noise are progressively removed with each iteration.

At the 10th iteration, the mean horizontal and vertical velocity components within the moving square patch were estimated at 0.534 pixel/frame (V_x) and 0.538 pixel/frame (V_y) respectively. Their respective variances were 0.0004 and 0.0004. The mean percentage error estimated for V_x was 6.9% and 7.7% for V_y. This accuracy is comparable to the results obtained by Heeger [*Heeger '87*].

The boundary images for the same moving random dot square at various smoothing stages are shown below. The position of the motion boundary has a tendency to be biased towards the side of the region with poorer initial certainty measure, they are usually areas with a weaker intensity gradient.

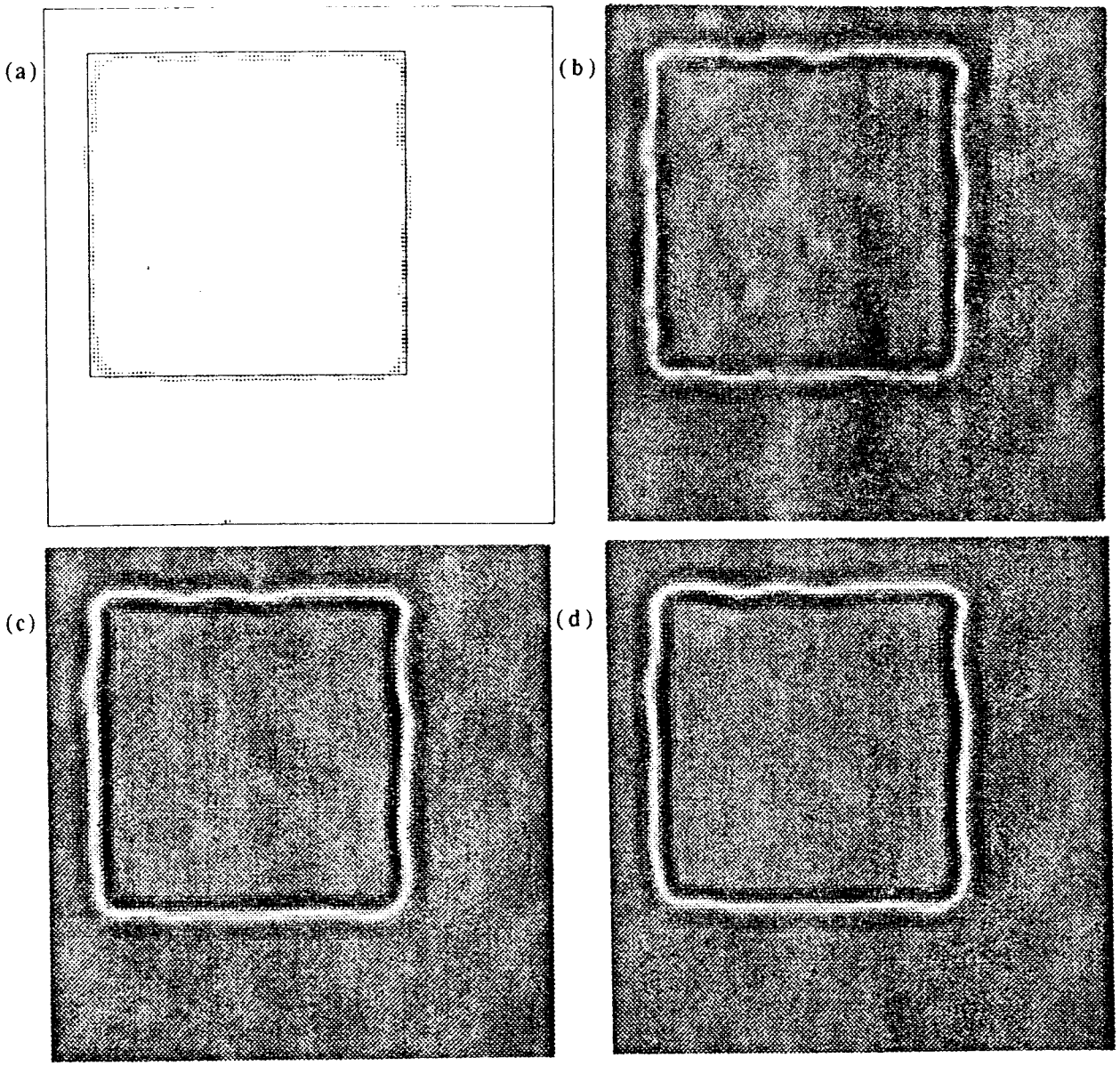


Fig. 11 - Boundary images of the moving random dot square sequence. (a) shows the estimated motion boundary overlaid over the actual motion boundary of the centre image frame. (b),(c)&(d) are boundary images at iterations 1, 5 & 11 respectively. Observe how the weak motion boundaries (eg: top left edge, just below the corner) were progressively reinforced with each iteration. Regions within the moving square patch that were significantly corrupted by added noise were gradually smoothed out.

5.2 Rotational flow field

A rotational velocity flow field was generated by applying rotational transformation to the random dots within a circle of radius 50 pixels, centred at the middle of a static random dot image. This rotation has an angular velocity of 1 deg/frame. The rotational velocity flow field obtained by application of the cooperative boundary detection and velocity field smoothing algorithm is shown below in Fig. 12.

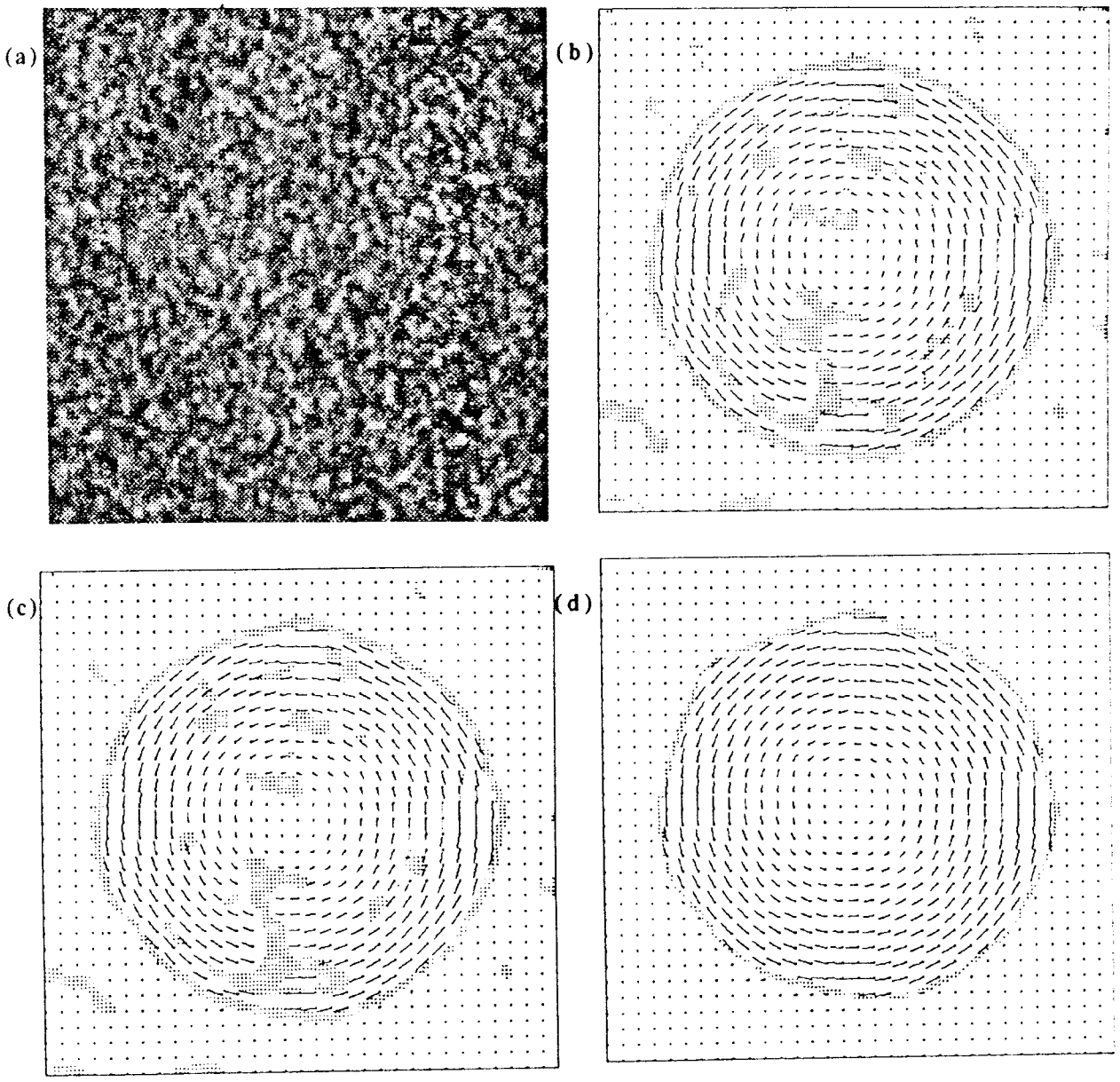


Fig. 12 - Rotating random dot circle with SNR of 10dB. (a) Shows one frame from the sequence. The rotational transformation was done on an image twice the size and was then spatially lowpass subsampled to the required size for generating the rotating sequence. This was to reduce errors caused by the discrete nature of the transformation algorithm. (b),(c)&(d) Shows the rotational velocity flow field at iterations 1,5 and 20 respectively.

Error estimates for the rotational velocity field were obtained at the 15th iteration. The mean percentage error for the velocity components V_x and V_y were 12.8% and 15.0% respectively (only non-boundary points and points with velocity components exceeding 0.01 pixel/frame were considered in evaluating the mean percentage error).

The boundary images for the same rotating random dot circle at various smoothing stages are shown below.

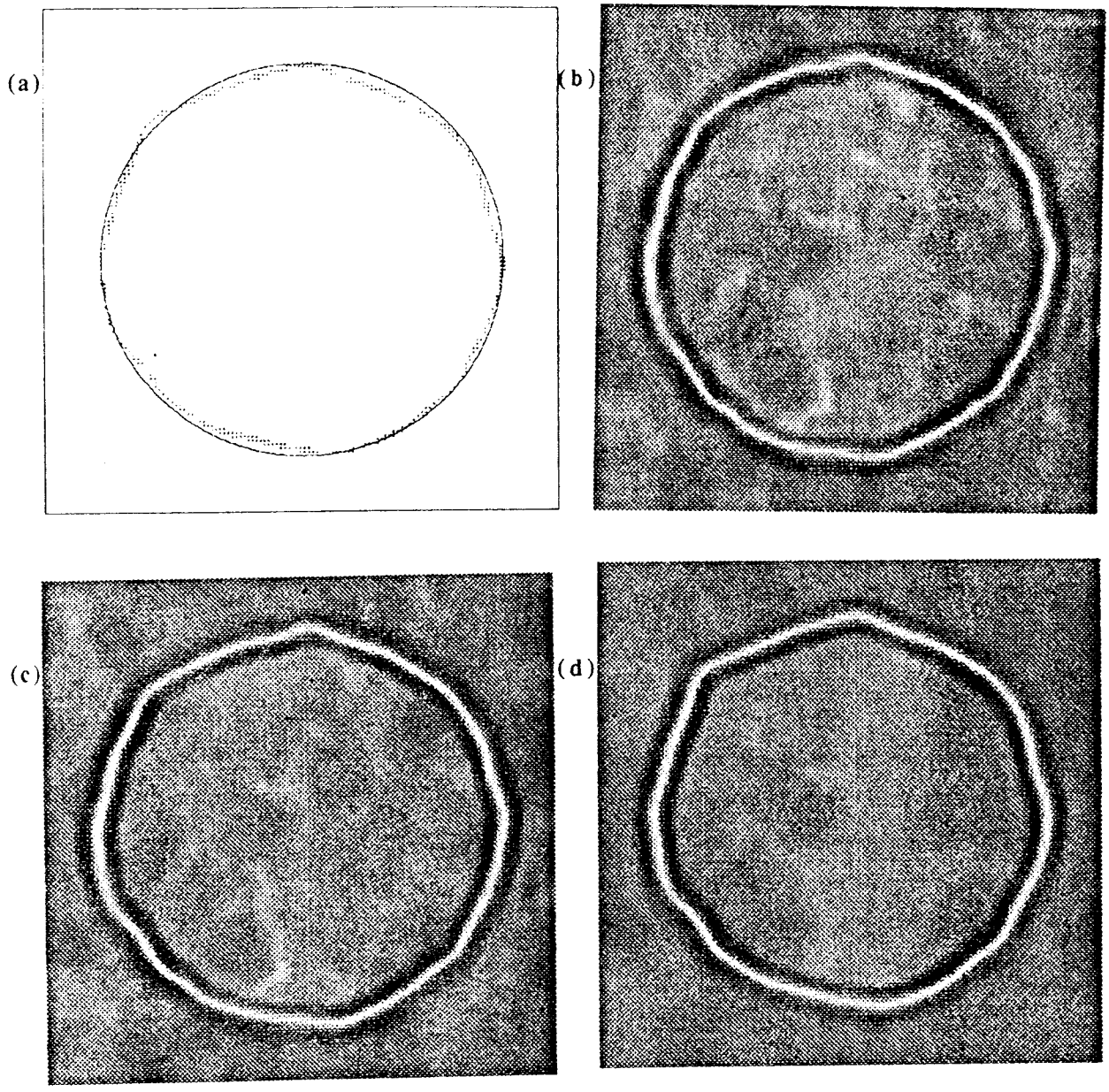


Fig. 13 - Boundary images of rotating random dot sequence. (a) shows the estimated motion boundary overlaid on the actual motion boundary of the centre image frame.(b),(c)&(d) are boundary images at iterations 1, 5 & 20, respectively.

6. Summary

This report has described a technique for deriving the optical flow field in noisy image sequences which contain motion discontinuities. It has been shown how local image motion can be characterised as spatiotemporal orientation, and this orientation can be detected in the spatiotemporal frequency domain as spectral energy concentrated about a plane or a line through the origin. A technique analogous to an eigenvalue analysis of the inertia tensor was used to determine this orientation.

This technique was further extended to smooth velocity flow fields by integrating motion information at each locality with that of its immediate neighbourhood. An iterative algorithm was developed that ensured this smoothing was constrained by the detection of motion discontinuities. A further strength of this algorithm is its ability to propagate good initial velocity estimates while suppressing unreliable ones. This has the effect of improving the overall accuracy of the optical flow field obtained, even under fairly noisy conditions.

The cooperative motion boundary detection and velocity field smoothing algorithm was shown to perform well on both translational and rotational random dot image sequences. The mean percentage error of the velocity estimates for translational motion was less that 8%, which is comparable to existing motion estimation techniques applied to similar noisy random dot sequences moving at subpixel velocities [*Heeger '87*].

The present work described here has only taken into account the spatial integration of motion information. Further improvement in noise immunity could be attained through the combined spatial and temporal integration of motion information. There is also a need to combine in some meaningful way, the velocity estimates at various levels of the lowpass spatial pyramid. This will enable the detection of a greater range of velocities within the image sequence without unduly sacrificing the detection resolution of motion boundaries at regions moving with slower velocities.

7. References

[*EH Adelson & JR Bergen '85*] - "Spatiotemporal energy models for the perception of motion", J. Opt. Soc. Am A/Vol 2, No.2, Feb '85, pp 284-299.

[*JK Aggarwal & N Nandhakumar '88*] - "On computation of motion from sequences of images- a review", Proc. of IEEE Vol.76, No. 8 Aug '88 ,pp 917-935.

[*O Braddick '74*] - "A short range process in apparent motion", Vision Res. 14 , pp 519-527.

[*J Bigun & GH Granlund '87*] - "Optimal orientation detection of linear symmetry", Proc of IEEE ICCV , Jun '87, pp 433-438.

[*J Bigun & GH Granlund '88*] - "Optical flow based on the inertia matrix of the frequency domain", Proc of SSAB on Image processing, Mar '88.

[*RN Bracewell '86*] - "The Fourier transform and its application", McGraw Hill books.

[*PJ Burt & EH Adelson '83*] - "The Laplacian pyramid as a compact image code", IEEE Trans. COM, Apr '83, pp 532-540.

[*DJ Heeger '87*] - "Optical flow using spatiotemporal filters", J. Opt. Soc. Am A/Vol 4, No.8, Aug '87, pp 1454-1471.

[*BKP Horn & BG Schunck '81*] - "Determining optical flow", AI, Vol. 17 Aug '81, pp 185-203.

[*D Marr & S Ullman '81*] - "Directional selectivity and its use in early visual processing", Proc. R. Soc. Lond. B 211 '81, pp 151-180.

[*VS Ramachandran & SM Anstis '86*] - "The perception of apparent motion", Sci. Am. Jun '86, pp 80-87.

[*W Reichardt & T Poggio '79*] - "Figure-ground discrimination by relative movement in the visual system of the fly", Bio. Cyber. Vol. 35, '79, pp 81-100.

[*AJ VanDoorn & JJ Koenderink '83*] - "The structure of the human motion detection system", IEEE Trans. SMC 13, No. 5, Sep '83, pp 918-922

[*AB Watson & AJ Ahumada '85*] - "Model of human visual motion sensing", J. Opt. Soc. Am A/Vol 2, No.2, Feb '85, pp 322-341.

[*RG Wilson & A Bhalerao '90*] - "Kernel design for efficient multiresolution edge detection and orientation estimation", July 90, in publication, IEEE Trans. PAMI.