2003

Use of multi-scale phase-based methods to determine optical flow in dynamic scene analysis

Robert Hastings
Edith Cowan University

Edith Cowan University

Copyright Warning

Use of Multi-Scale Phase-Based Methods to Determine Optical Flow in Dynamic Scene Analysis.

Robert Hastings
Grad. Dip. Applied Science (Computer Studies), Edith Cowan University 1991.
BSc(Hons), La Trobe University 1970.

Department of Computer Science.
Edith Cowan University.
Thesis submitted for the degree of Master of Science (Computer Studies).

Date: December 12, 2003

USE OF THESIS

The Use of Thesis statement is not included in this version of the thesis.

Abstract

Estimates of optical flow in images can be made by applying a complex periodic transform to the images and tracking the movement of points of constant phase in the complex output. This approach however suffers from the problem that filters of large width give information only about broad scale image features, whilst those of small spatial extent (high resolution) cannot track fast motion, which causes a feature to move a distance that is large compared to the filter size.

A method is presented in which the flow is measured at different scales, using a series of complex filters of decreasing width. The largest filter is used to give a large scale flow estimate at each image point. Estimates at smaller scales are then carried out by using the previous result as an *a priori* estimate. Rather than comparing the same region in different images in order to estimate flow, the regions to be compared are displaced from one another by an amount given by the most recent previous flow estimate. This results in an estimate of flow **relative** to the earlier estimate. The two estimates are then added together to give a new estimate of the absolute displacement. The process is repeated at successively smaller scales. The method can therefore detect small local velocity variations superimposed on the broad scale flow, even where the magnitude of the absolute displacement is larger than the scope of the smaller filters. Without the assistance of the earlier estimates in 'tuning' the smaller filters in this manner, a smaller filter could fail to capture these velocity variations, because the absolute displacement would carry the feature out of range of the filter during successive frames.

The output of the method is a series of scale-dependent flow fields corresponding to different scales, reflecting the fact that motion in the real world is a scale-dependent quantity.

Application of the method to some 1-dimensional test images gives good results, with realistic flow values that could be used as an aid to segmentation. Some synthetic 2-dimensional images containing only a small number of well defined features also yield good results, but the method performs poorly on a random-dot stereogram and on a real-world test image pair selected from the Hamburg Taxi sequence.

Declaration

I certify that this thesis does not, to the best of my knowledge and belief:

- incorporate without acknowledgment any material previously submitted for a degree or diploma in any institution of higher education;
- contain any material previously published or written by another person except where due reference is made in the text; or
- contain any defamatory material.

Signature [redacted]..

Date 29/1/2004.....

Acknowledgments

This research was greatly facilitated by the receipt of a HECS exemption scholarship for the first 12 months of the research program.

I am indebted to my initial supervisor, Dr. James Cooper, now of Curtin University of Technology, for all his theoretical and technical assistance, in particular for his help in setting up my home computer system so that I was able to continue working from home, and later from the other side of the country, on this research.

Contents

List of Figures

Chapter 1

Introduction

1.1 Concepts

1.1.1 Computer Vision

Various definitions have been proposed for **Computer Vision**, eg.:

"Computer vision is the construction of explicit, meaningful descriptions of physical objects from images." (Ballard and Brown, 1982, P.xiii)

"...computer vision is about image acquisition, processing, classification, recognition, and, to be all embracing, decision making subsequent to recognition as in, for example, when a space probe has to make its own decision about movement." (Low, 1991, P.2)

I have adopted the view that the function of a computer vision system is to:

1. Accept a **digital** image or a sequence of such images,
2. Derive from the image(s) a **real-world model** of the scene from which the images were derived.

The nature of the resultant model depends on the uses to which the model is to be put. For some applications the goal may be to locate in the image

a particular object whose appearance is known is advance, in which case the analysis output might be a description (in English or some other descriptive notation) of the location of occurrences of the object that is sought. An alternative goal might be to detect the motion of objects as revealed in a succession of images; here the output might be a tabulation of the measured velocities at different points in the image.

1.1.2 What is an Image?

A **digital image** may be defined as an array of points, each point being associated with a property that can be expressed numerically. In the case of a black-and-white photograph the property is grey-level intensity, i.e. the brightness at each point, whilst in a colour photograph the parameter is a set of numbers describing the colour and intensity of the light via the values of the different colour components, using a basis such as RGB or hue-saturation-intensity. However the image parameter may quantify something other than visible light, such as infra-red radiation, or radar or sonar echoes. More generally, any measured or derived quantity may be an image parameter. In this work we shall frequently refer to a **complex** image, this being one in which the image parameter is complex in the mathematical sense, i.e. having a real component and an imaginary component.

Images may have a number of dimensions other than two. The array of image points may conceivably have any number of dimensions, although images with two spatial dimensions are the most familiar. A time sequence of 2-dimensional images may be thought of as a single 3-D image with time comprising the third dimension.

It is sometimes useful in research to work with 1-dimensional images for the purpose of simplifying a problem.

1.1.3 Features and Objects

It is necessary to distinguish between the uses of these two terms.

An **object** is a part of the original scene that is thought of as being a separate entity in its own right, eg. a person, an animal, a house, a brick that forms

part of a building, or a speck of dust on the camera lens.

A **feature** is a part of the image that we identify as being significant for some reason when attempting to analyse the image. A feature may be, for example:

- A part of the image where the intensity values differ markedly from those in the surrounding parts of the image.
- A region of the image where the intensity values change rapidly with spatial coordinate.
- A region where intensity changes rapidly between frames in an image sequence.

Features in the image frequently correspond to objects in the original scene. For example a large dark region in an otherwise bright image may correspond to a dark coloured building in the scene. A feature may also correspond to a part of an object, eg. a region of discontinuous intensity change in the image may correspond in the scene to the corner of a building.

Movement of a feature in the image does not always indicate the movement of an object in the scene. Some examples are:

- **Shadows.** The movement of a shadow may be due to movement of the object casting the shadow, or to movement of the source of illumination.
- **Reflections.** These may move due to movement of the reflected object or a change of orientation of the reflecting surface.
- **Projected images,** such as those on a T.V. or cinema screen.

Image analysis initially generates information about **features** rather than **objects.** Using features to infer the nature of objects present in the original scene represents a high-level phase of the analysis. In some applications the identification of such objects is the chief goal of the analysis. Examples where this is the case include computer analysis of medical images to identify tumours, and use of military satellite images to identify enemy warships.

1.1.4 Image Velocity

An important aspect of a scene description is the specification of the **motion**, if any, that is occurring in the scene. By identifying the movement of features in the image, it may be possible to deduce the movement of objects in the original scene. It may also be possible to use motion as an aid to segmentation (see section 1.1.5) to assist in identification of features, and thence objects, or indeed to identify features where they cannot be delineated by other means.

Image velocity or **optical flow** may be defined as the apparent motion of features in a small region of an image. Some authors (eg. Fleet, 1992) use the term **image velocity**, whilst others (eg. Ballard and Brown, 1982) use the term **optical flow** to mean the same thing. In the present work the terms are used interchangeably.

Some reasons why we might wish to determine optical flow are:

- Analysis of stereoscopic image pairs can yield the between-image displacement at each point in the image. This can be used to calculate the distance between the observer and the object at each point in the image. This has various potential practical applications, such as in aerial surveying.
- Identification of movement in a temporal sequence of images, such as those produced by a camera attached to a moving robot, can be used to derive the position of objects by means of an analysis similar to that used for stereoscopic pairs. In such cases, where the apparent motion is due to the movement of the camera rather than the movement of objects in the scene, the motion is referred to as "egomotion". One possible application is in the exploration of other planets by moving robotic devices such as NASA's "Mars Rover".
- Applications where the motion in the image is due to the motion of objects rather than that of the observer include the monitoring of traffic, and the observation of the movement of clouds (Aggarwal and Duda, 1975).

1.1.5 Segmentation

Segmentation can be described as the splitting of an image into regions each of which is distinguished by some property from its neighbour (Low, 1991, P.84), or

> the process of partitioning an image into meaningful parts such that all points belonging to a part have some common property or can be represented using a mathematical or logical predicate.

(Jain, Martin, and Aggarwal, 1979)

The objective of segmentation is to identify the parts of the image that correspond to objects in the scene.

1.2 Methodology of This Work

The approach to optical flow measurement adopted by Fleet (1992) is to apply a number of "velocity tuned filters" — 3-dimensional Gabor filters that are skewed in the time dimension so that they are most sensitive to features that are moving with particular velocities. Because the filters are selected in advance, this limits the range of velocities that can be measured, and increases the cost of the method in being able to deal with any particular range of velocities, since sensitivity to a wider range of velocities requires a greater number of velocity tuned filters.

Our approach differs from that of Fleet in that we apply Gabor filters of widely varying scales, but limited to single frames, i.e. the filters have only a spatial component. Like Fleet, we compute the velocities from individual filters by dividing the time derivative of phase with the space derivative of phase. However we use a prior estimate $\tilde{d}$ of the displacement at point $\tilde{x}$ to apply the flow operator itself not to point $\tilde{x}$ in both frames, but to point $\tilde{x}$ in one frame and point $\tilde{x}+\tilde{d}$ in the other. This has the effect of producing a velocity adjusted flow operator "on the fly". The resulting velocity measured from the output of one filter is then used as the prior estimate $\tilde{d}$ of velocity for the next flow computation at the next finer scale. We use the term "velocity

adjusted" rather than "velocity tuned" to describe the filters used in our work, to emphasise the fact the filter used at each stage is not defined in advance based on a particular velocity, but rather is applied in a manner governed by the output from the previous filter(s).

The multi-scale methodology outlined above and described in more detail in section 3.2.2 is very general. It is not specified how the flow estimate is to be made at each step, whether by the differential behaviour of intensity, by region-based matching, or by some other method. It would be possible to specify a whole family of optical flow methologies based on the multi-scale approach, each one being characterised by the manner in which the velocity estimate is made. In the present work, however, each estimate is made using the phase behaviour of a Gabor filter. We therefore implement the methodology of section 3.2.2 as follows:

- A sequence of Gabor filters of decreasing width is used to perform the convolutions. Analysis proceeds from the largest scale through to the smallest. At each scale:
  - The input images are convolved with the appropriate Gabor filter, giving complex-valued output images.
  - A flow estimate is derived at each image point, by means of applying a differential technique to the phase of the output image. The previous estimate of the flow is used as the starting point in generating the new estimate. This is discussed in detail in sections 4.1.6 and 5.1.6.
- The above analysis also incorporates an uncertainty measurement at each point.

Chapter 2

Background and Literature Review

There are several different approaches to the measurement of image velocity, each with its own advantages and drawbacks. In this chapter we identify and compare these different strategies. Following this, we review the research that has been carried out to date in determination of image velocity. Some observations are made on the performance of the different techniques.

2.1 Measurement of Image Velocity

Algorithms for estimating optical flow can be classified into two types, which employ two different methodologies (Aggarwal and Nandhakumar, 1988; Vega-Riveros and Jabbour, 1989):

1. **Feature Tracking.**
    - Identify the features in an image on the basis of properties such as brightness or colour.
    - Derive the image motion by tracking features across frames. This generates a **sparse** flow field, so called because a flow vector is associated with only certain image features rather than with each point in the image.

The above approach is also known as the **token-based approach** (Cooper, 1992). It requires that each feature or token in one frame be associated with the correct token in the next frame for the purpose of computing the movement of the feature.

This approach is feasible if there are only a few tokens. The task becomes much more complex as the number of tokens increases, because the number of possible matches increases with the square of the number of tokens. The problem of identifying which feature in one frame corresponds to which feature in another frame is referred to as the **Correspondence Problem.**

There are a number of ways (Ballard and Brown, 1982, pp. 198-199) in which a constraint can be applied to narrow down the number of possible correspondences:

(a) If a maximum image velocity can be specified, based on physical considerations of the original scene, then this places a limit on the possible displacement of a feature between frames.

(b) It may be possible to place a constraint on the rate of change of velocity with time, in accordance with physical laws. This constraint is only useful when analysing a sequence of more than two images.

(c) Spatially coherent objects may be expected to exhibit a "common motion" across the object.

(d) Two points from one image generally cannot be said to match a single point from another image (exceptions arise in the case of occlusion).

(e) Depending on the kind of scene being analysed, a world model can supply information about motions of the actual objects being imaged. For example, if the scene consists of of traffic on a highway, the component of motion parallel to the direction of the road is (hopefully) much greater than the component perpendicular to the road.

Thompson and Barnard (1981) identify three basic ways of tackling this problem:

(a) **Differencing techniques** start with a point by point determination of significant changes in image intensity. This identifies portions of moving surfaces.The rate of translation can then be estimated by matching such regions in different frames.

(b) **Temporal-spatial gradient techniques** avoid the correspondence problem by deriving a dense motion field, as discussed in the next section, rather than by attempting to identify individual features and track them.

(c) **Matching techniques** identify a set of structures in one image frame, then attempt to search for the corresponding structures in subsequent frames, using some optimisation criteria based on the properties of structures or the relationships between them. "Structures" may be small image segments, derived features points or image regions likely to correspond to full object surfaces.

2. **Dense Motion Field.**

   The image motion is estimated at all points in the image.

   Because not all image regions contain the same amount of information with which to derive motion, the reliability of the estimate may vary greatly from one part of the image to another. This variation in uncertainty is not always acknowledged — authors (eg. Barron et al. (1994); del Bimbo et al. (1996)) historically have rated the performance of methodologies in terms of the difference between measured and "actual" flow averaged over the whole image. In this work, however, the flow estimate at each point incorporates an uncertainty estimate, which, like the flow, varies over the image. Other workers have recognised the need for a point-by-point uncertainty estimate; Roach and Aggarwal (1979), for example, derive motion estimates for a series of frames by propagating the velocity estimates and the associated uncertainties from one frame to the next (see section 2.2.4).

Whereas the first approach can lead to a partial segmentation (by identifying features with similar motion as being part of the same higher-order feature), it requires that features first be identified on the basis of something other than image velocity. This methodology, based on selection of features, has the inherent disadvantage that it is difficult to apply to certain images, such as images of motion of a water surface, which lack features such as edges and corners (Jahne, 1993, P.78). The second approach, on the other hand, can potentially achieve a dense segmentation by identifying whole regions in the image such that within each region the flow at all points is the same or nearly the same.

There is some evidence (Ullman, 1981) that both kinds of process are employed in the human visual system. Short-range motion, involving spatial

displacements of up to about 15 minutes of visual arc and temporal intervals of less than about 60-100 msec, appear to be detected by an intensity-based dense motion scheme, while the mechanism for detecting long-range motion appears to be a token-matching scheme.

The present work is concerned with the dense motion approach, and in particular with the task of obtaining, for all points in the image, a meaningful measure of image velocity that incorporates the uncertainty in the velocity estimate.

Motion analysis has traditionally been performed by analysis of just two consecutive images. It is claimed (eg. Jahne, 1993) that the analysis becomes much more robust when many images are considered rather than only two. The present work, however, employs only two-image sequences, since this is sufficient to demonstrate the methodology. In section 6 I note that a certain problem that manifested itself with one of the 1-D test images might have been eliminated by employing a sequence of more than two images.

A number of techniques exist for deriving a dense optical flow field. These can be categorised (Barron et al., 1994) as follows:

1. Differential Methods;
2. Region-Based Matching;
3. Energy-based Methods;
4. Phase-Based Methods.

It is also possible to combine two or more of these techniques.

2.1.1 Image Noise

Any real image (as opposed to a synthetically generated image) is subject to noise – random errors in the intensity registered at each pixel, due to imperfections in the image capture process. In addition, the fact that any image has only a finite resolution may also be regarded as a source of noise, since the part of the original scene that is mapped onto a single pixel would

normally not possess a totally uniform intensity across its area, but of course the image intensity is constant across the pixel by definition.

Measurements of image velocity are always affected by noise, and different methods of measurement are affected in different ways. Fermuller et al. (2000) show that there are limitations to determination of optical flow due to bias in the estimates, which are caused by noise in the data. The estimated velocity tends to be an underestimate in magnitude and to be closer in direction than the actual flow vector to the majority of gradients in the patch. Correction for the bias is difficult because the noise parameters can change in complex and unpredictable ways. The authors claim this problem exists for the human vision system as well, and that this is the source of an optical illusion known as the Ouchi illusion.

2.1.2 Differential Methods

These are sometimes referred to as **temporal-spatial gradient techniques** This technique is employed for example by Thompson and Barnard (1981), who use the change in intensity at an image point over both time and space to estimate the rate of translation of the underlying surface.

The technique rests on the assumption that intensity is conserved within a moving feature (Horn and Schunck, 1981). From this assumption the **Gradient Constraint Equation** is derived, relating the image velocity to the time and space derivatives of intensity:

$$\nabla I \cdot \mathbf{v} + \partial I/\partial t = 0 \tag{2.1}$$

where v is the velocity and ∇I and $\partial I/\partial t$ represent respectively the spatial component-wise derivative and the time derivative of image intensity I. In the case of 1-dimensional images the above equation reduces to:

$$\mathbf{v} = -\frac{\partial I/\partial t}{\partial I/\partial x}. \tag{2.2}$$

This relation is illustrated in Figure 2.1 for the case of a 1-dimensional image.

When this analysis is applied to a portion of a 2-D spatial image, it constrains the two components of $\mathbf{v}$ by only one linear equation, restricting $\mathbf{v}$ to a

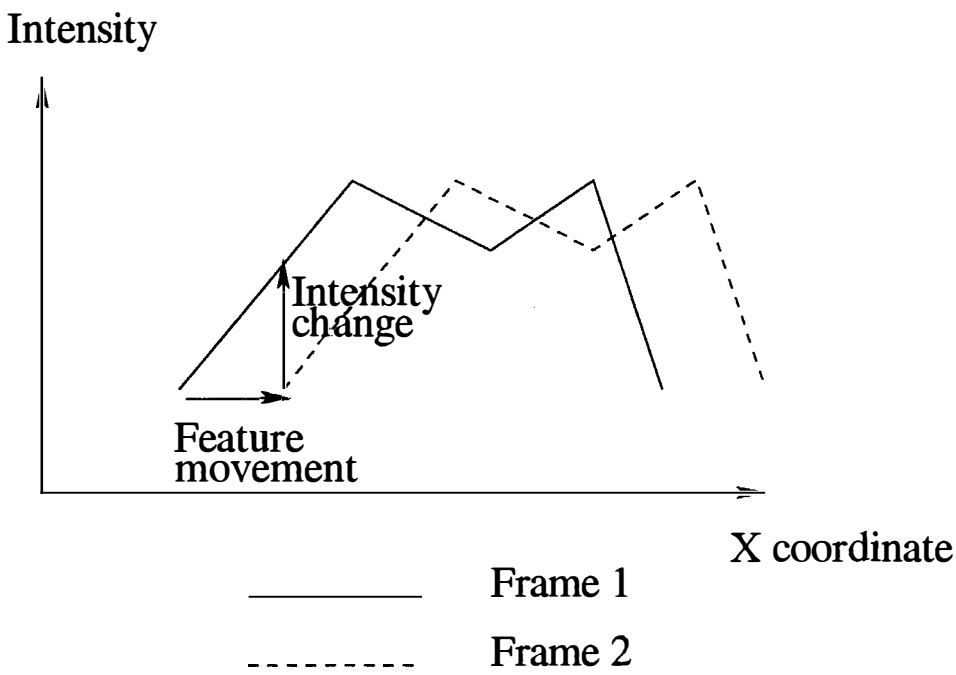


Figure 2.1: Image velocity derived from spatiotemporal derivatives.

particular set of values. The information derived from one portion of the image by this means is insufficient to obtain a unique value for **v**; this is referred to as the **aperture problem**. Further constraints are therefore required in order to determine **v** unambiguously. In some parts of the image such further constraints may be available; in other parts of the image they may not.

The **aperture problem** is illustrated in Figure 2.2. At the two points on the boundary of the rectangle, constraints on the velocity can be derived. At a third point in the interior, the velocity is indeterminate because there are no features to track.

One method of introducing further constraints is to use second-order derivatives to give a second constraint equation, in the manner of Nagel or Uras (McCane et al., 2001).

Another approach is to combine local estimates of component velocity through space and time, using assumptions of smoothness of the flow field across an image (eg. Horn and Schunck, 1981). A problem with this latter technique is that closely-spaced areas of the image may exhibit very different image

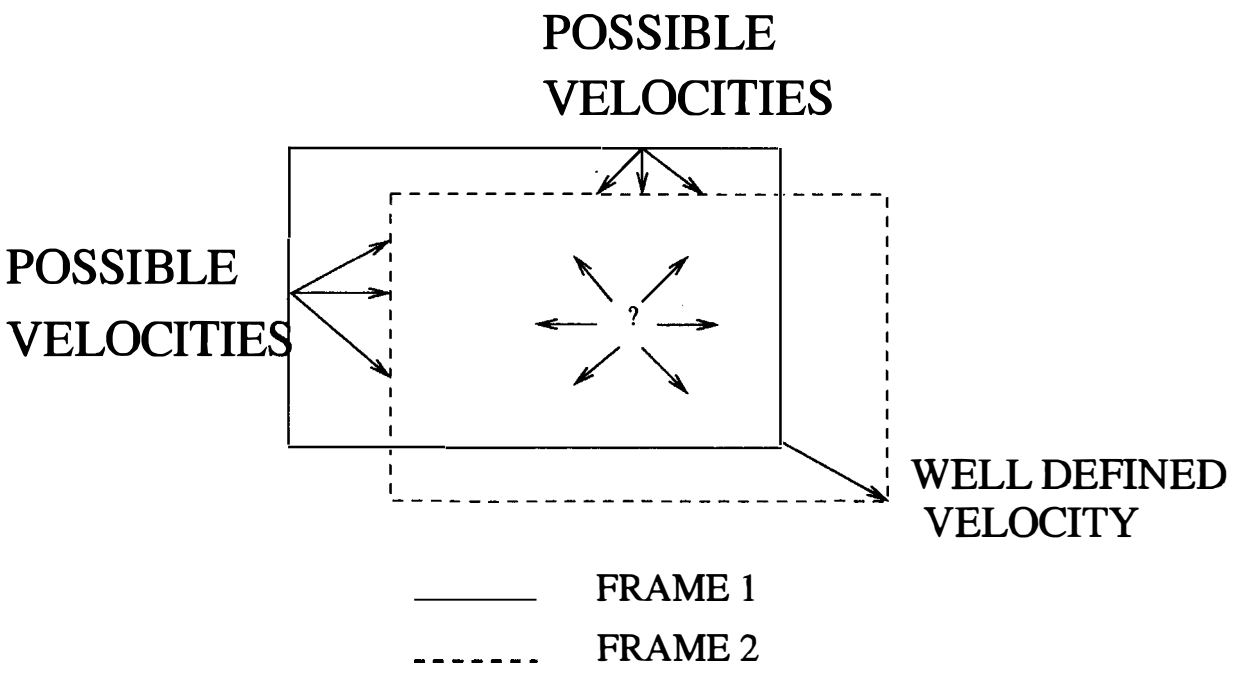


Figure 2.2: Optical Flow: Areas of high and low motion information content.

velocities, due to the occlusion of one object by another. The assumption of smoothness therefore breaks down at the boundaries of moving objects, i.e. at the most interesting parts of the scene.

Some of the problems with differential methods in general are:

- The methods rely on the assumption that image intensity variation is entirely due to movement in the scene, ignoring the effects of other changes such as changing illumination. Verri and Poggio (1989) point out that, for images of real-world scenes containing motion, the optical flow as determined from first order variations of the image brightness is exactly the same as the motion field only for Lambertian objects (i.e. objects showing the Lambert reflectance function) that translate under uniform, fixed illumination.
- Differential methods are highly sensitive to noise in the image, implying the necessity for some kind of prefiltering of the image to reduce the effect of noise.
- Differential methods require that the intensity function be differentiable, which may require a smoothing operation to be performed on

the input image before differentiation is carried out. This results in areas of ambiguity in the flow field, and indeed poses the question of how the flow should be defined at these points.

- Areas of the image containing no features give no information about the flow. It is possible to generate a flow field by smoothly propagating velocity values from areas of high information content to areas of low information content (Horn and Schunck, 1981), but this can give false results. As Xie et al. (1996) point out, by postulating a smooth motion field we may attempting to impose smoothness at places in the image where no smoothness exists, such as at sharp motion boundaries.

  The problem is that more than one real-world scene can produce the same image sequence. In Figure 2.2, for example, consider how the image sequence should be interpreted if the interior of the rectangle were filled with black. There are many real-world situations which would give rise to this image sequence; three possibilities are:

  1. A black rectangle moving across a stationary white background.
  2. A black rectangle superimposed on a white object that extends beyond the image boundary, both objects moving together.
  3. A white object, extending beyond the boundaries of the image, containing a rectangular hole, moving across a stationary black background.

  In the first two cases, the actual velocity at a point in the interior of the black rectangle is towards the lower right; in the third case it is zero. It is not possible to determine the correct velocity purely by studying the image; it would be necessary to make assunptions about the objects captured in the image. An optical flow determination that assigns a velocity to such an interior point must also acknowledge the inherent uncertainty, otherwise the determination is misleading because it implies the existence of information that is simply not present.

2.1.3 Region-Based Matching

Region-based matching methods use some measure of image region similarity to search for an image patch in frame 2 that closely matches a particular image patch in frame 1. The optical flow between the two frames for a certain patch is defined as that displacement which, when applied to the

patch in the first frame, yields the best fit with the corresponding patch in the second frame. A similarity measure, such as a sum-of-squared-differences or a normalized cross-correlation, is applied to the two patches in the two frames, in order to evaluate the quality of the "fit".

Some problems with this technique are:

- A point-by-point similarity measure will only find a good match in the case where the motion is a simple translation. It will not handle scaling and rotational transformations.
- Noise in the images affects the matching at finer resolutions. (This of course is a problem with any measurement technique applied at a fine resolution.)
- Occlusions in an image can result in portions of one image that have no counterpart in the other image. If the region we are attempting to match contains such an occluded portion, this will result in a poor similarity measurement for the region, whereas in fact the fit might have been a good one for those parts of the region that were not occluded.

An example of this technique is the work of Agarwal and Sklansky (1992). They divide velocity space into a set of regions, then for each point (x, y) in the image a confidence level is assigned to each region in velocity space. This confidence level is based on a function that measures the inter-frame disparity between between two image regions in two frames, the regions being offset from one another by the velociuty that corresponds to that region of velocity space. A velocity clustering technique is then used to refine the confidence level estimate, following which the velocity corresponding to the region with the highest confidence level is selected. They claim that the clustering technique partly overcomes the first of the problems listed above, in that it handles rotational motion as well as simple translation.

2.1.4 Energy-Based Methods

Energy-based motion analysis employs the technique of transforming the image using a number of velocity-tuned filters. The filters are complex-valued spatio-temporal filters, i.e. they are applied to a temporal image sequence,

not just a single image. They are referred to as "velocity-tuned" because each is designed so that the magnitude of the complex output (the "energy" of the output) is a maximum for a particular image velocity. The optical flow is then estimated by examining the outputs of all the filters.

Heeger (1988) developed a method based on this approach:

1. The image is operated on with a series of such filters.
2. The velocity is estimated by minimizing the sum of the squares of the differences (SSD) between the filter output and the expected output for all of the filters.

This work, and that of other researchers who adopted this approach, are described in section 2.2.3.

2.1.5 Phase-Based Methods

Phase-based methods may incorporate aspects of the four approaches described above, but they merit a separate classification because of their defining characteristic, which is that instead of using intensity as the basis for analysis, phase is used. A complex-valued filter is applied to the input image, resulting in an output image whose values at each point are also complex, and the phase of this complex value becomes the basis for estimating the flow. The phase ϕ is defined by

$$\phi = arctan(I/R) \tag{2.3}$$

where R and I are, respectively, the real and imaginary parts of the complex valued output.

In principle, it would be possible to derive motion by taking any well-defined varying function of the input image and performing a velocity analysis on the resultant output field. Intensity and phase are only two possible candidate functions; there are many others. For example, one could use the amplitude of the complex filter output. However, phase as a parameter has a particular property that makes it an attractive candidate for flow derivation:

All optical flow techniques estimate the flow by, in one way or another, finding the flow that best explains the change between one image frame and another. Therefore it is desirable to ignore as far as possible changes that are due to effects other than motion, such as changes in scene illumination. Fleet (1992, pages 56-61) found that in test image sequences, phase contours follow closely the movement of features in the images, whilst amplitude was very sensitive to variation in scene illumination and to local variations in speed, scale and orientation of the input, as well as to the scale of image features relative to the scale of the filter. Fleet and Jepson (1993) claim that phase exhibits this desirable behaviour because it is robust with respect to small geometric deformations such as those typically occurring in images of 3-D scenes. The fact that phase is robust with respect to variations in illumination should not surprise, since a change in illumination would cause equal fractional changes in the real and imaginary parts of the filter output, leaving their ratio, and hence the phase, unchanged.

Jepson and Jenkin (1989) show how to estimate velocity from phase differences between different image points.

A differential phase-based optical flow method can be described in general terms as follows:

- Operate on the image with one or more complex-valued filters.
- Measure the velocity in each filtered image using a differential technique applied to the phase of the output.
- Integrate measurements from all the filters to obtain the best velocity estimate.

The specifications of the filters used, and the manner in which the velocity estimates from different filters are integrated, define the particular methodology.

One candidate for a complex-valued filter is the Gabor filter, which consists of a complex periodic waveform modulated by a Gaussian envelope (See section 4.1.1 for the mathematical definition). The effect of the Gaussian envelope is that at any point the filter output is determined by the intensity values in a restricted region centred on the point, because the envelope function decreases with increasing distance from the centre (Figure 2.3). The

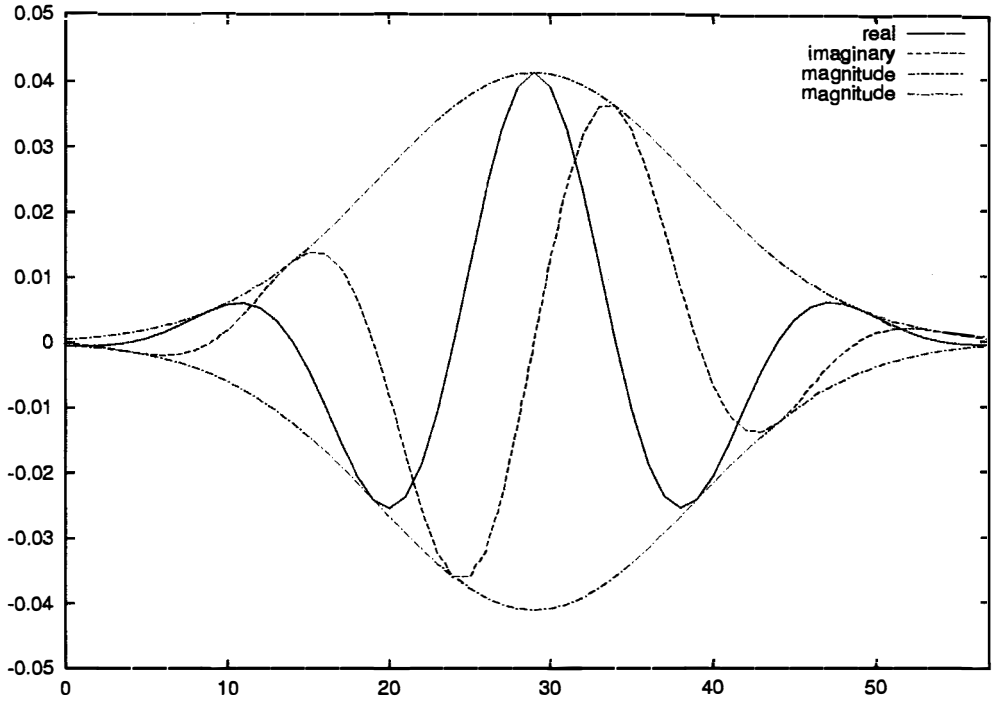


Figure 2.3: Gabor filter function formed by modulating a periodic waveform with a Gaussian envelope.

waveform component of the Gabor filter has the effect of making the output sensitive to intensity variations within this restricted region. In this respect the Gabor filter mimics the response of the human retina — each ganglion cell in the retina responds to light falling within a limited retinal region, and is sensitive to the variation in light intensity between the centre of the region and the surrounding area (Ullman, 1981).

In the present work we convolve the input images with a series of Gabor filters of decreasing width, and estimate the flow using the phase derivative in a manner analogous to the intensity-based method described in section 2.1.2. By analogy with equations 2.1 and 2.2, the gradient constraint condition applied to phase yields the relation

$$\nabla\phi \cdot \mathbf{v} + \partial\phi/\partial t = 0 \tag{2.4}$$

or, in the one-dimensional case,

$$\mathbf{v} = -\frac{\partial\phi/\partial t}{\partial\phi/\partial x} \tag{2.5}$$

Two problems arise in phase-based analysis: *phase singularities* and *flow singularities*. These can lead to unreliable velocity measurements. Areas of the image where either kind of singularity occurs must be identified, so that velocity measurements from these regions may be discarded or at least labelled as having high uncertainty. Methods to identify such regions have been developed (Fleet, 1992; Fleet and Jepson, 1993) and are discussed below.

Phase Singularities. These occur where both the real and imaginary parts of a complex filter output are zero, so that the phase as given by equation 4.11 is indeterminate. This occurs in regions of the image where there are no features to provide a response to a filter, such as in regions of uniform brightness, to which a periodic filter is insensitive. However, singularities can also occur in parts of an image where structure exists. Phase varies rapidly across the image in the neighbourhood of these singularities, and exhibits a discontinuous jump at the singularity itself. Regions near these points exhibit phase instability, in the sense that a small intensity variation results in a large change in the phase. This means that, in these neighbourhoods, phase behaves chaotically, and noise in the image dominates the phase of the filter output, making phase-based motion measurements unreliable in these regions.

Detection of these regions is carried out using two constraints: an **amplitude constraint** and a **frequency constraint**, with two associated dimensionless constants τ_ρ and τ_k.

- The **amplitude constraint** is a restriction on the size of the relative derivative of amplitude ρ. It requires that this relative derivative be small enough so that, over the width of the filter, the variation in amplitude is not too large compared with the amplitude itself. This criterion is expressed as:

$$\frac{1}{\rho}\frac{\partial \rho}{\partial x} * \sigma < \tau_\rho \tag{2.6}$$

  where

  - ρ is the amplitude
  - τ_ρ is the amplitude threshold
  - σ is the spread of the filter

- The **frequency constraint** compares the measured spatial frequency (i.e. spatial phase derivative) with the peak tuning angular frequency $k = 2\pi/\lambda$ of the filter. Large deviations from the peak frequency are rejected.

  By analogy with the amplitude constraint, the frequency constraint requires that the spatial derivative of the phase remain relatively close to the peak filter frequency over the spread of the filter. The constraint is expressed as:

  $$\frac{|\partial\phi/\partial x - k|}{k} \leq \tau_k \tag{2.7}$$

  i.e. the difference between the phase derivative and the peak filter frequency, expressed as a multiple of the filter frequency, is less than the threshold value.

  Fleet uses a value for τ_k of 1.25.

In the present work, however, we do not check explicitly for these two constraints to be satisfied. This is discussed further in section 4.1.4.

Flow Singularities. These occur where the phase can be determined (i.e. there is no phase singularity), but where the spatial phase derivative ($\partial\phi/\partial x$ in equation 2.5) is zero. If $\partial\phi/\partial t$ is also zero, v is indeterminate; if $\partial\phi/\partial t$ is non-zero, this equation gives $v = \pm\infty$. An additional consideration is that, if the filter wavelength is λ, a distance of $\lambda/2$ corresponds to a phase change of π. This is indistinguishable from $-\pi$. For this reason, when processing with each filter, derived velocities greater than $\lambda/2$ were rejected.

It should be noted that the problem of flow singularities is not restricted to phase-based techniques. It can occur wherever flow is determined by using a gradient constraint equation. In any region of the image where the intensity is nearly constant in the spatial dimension, any parameter derived from intensity will also be nearly constant, and its spatial derivative close to zero. The need to filter out velocities greater than $\lambda/2$, however, only arises where phase is used as the parameter.

A single test serves both to identify points where the spatial derivative is zero, and to detect points where this derivative is non-zero but is such as to give too high a value for v:

$$|\partial\phi/\partial t| > |(\partial\phi/\partial x| \times \lambda/2 \tag{2.8}$$

This test is applied prior to actually calculating v. For points where this inequality holds, the velocity is assigned a value (zero), but the estimate is labelled as having a high uncertainty. This automatically avoids the possibility of attempting a division by zero.

2.2 Early Work on Image Velocity Measurement

Jain et al. (1979) use motion as an aid to segmentation. They identify regions of the image in which motion is occurring, using difference operations. They then delineate the images of the moving objects by focusing the segmentation process on these regions. They do not derive a motion field in the sense of assigning a velocity to points in the image. A similar approach is taken by Hogg (1977), who captures simple laboratory scenes on camera and uses simple picture differencing to identify moving features. Provided the overall illumination remains constant, the system is capable of tracking the movement of a small number of objects in the scene, and in some cases to identify the objects — for example, in one experiment a particular moving object was identified as a person. (Jain and Nagel, 1979) employ a similar picture differencing strategy to identify and track moving features in a TV image sequence.

Thompson (1980) also uses motion as an aid to segmentation, though he uses a differential method to estimate image velocity (see section 2.1.2).

2.2.1 Differential Methods

Fennema and Thompson (1979) describe a technique for determining optical flow via a differential method (as defined in section 2.1.2). At regions in the image where there is an intensity gradient, it is possible to write a gradient constraint equation for the image velocity (equation 2.1). Only the component of optical flow orthogonal to the brightness gradient can be estimated from one such equation. However, by sampling the intensity behaviour at a number of points in a region, a number of such constraint equations can be derived. These are then combined, using a clustering technique, to give a velocity estimate.

Points in the image where the intensity behaviour matches the behaviour that would result from this image velocity, to within a specified threshold, are assigned this velocity. The analysis is then repeated with these points excluded, resulting in a second velocity estimate. In this way a number of regions are identified and are each assigned a velocity.

Horn and Schunck (1981) also adopt a differential approach. They also derive a gradient constraint equation at each point, but in order to further constrain the velocity they apply a smoothness constraint, which assumes a smooth variation of optical flow across the image.

The following initial assumptions are made:

- Surfaces are flat, so that there are no shading effects caused by the shape of the object.
- Uniform incident illumination is present over surfaces.
- Reflectance varies smoothly with no spatial discontinuities.

The first two assumptions imply that the variation of intensity over a surface is due solely to variations in the reflectance of the surface, and that therefore the motion of brightness patterns is determined solely by the motion of objects.

The x and y velocity components are denoted u and v respectively. The flow field is smoothed iteratively by:

- Minimizing the magnitude of gradient of optical velocity.
- Minimizing the sum of magnitude of Laplacians of u and v. The Laplacians are defined as:

$$\nabla^2 u = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \tag{2.9}$$

and

$$\nabla^2 v = \frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} \tag{2.10}$$

In this manner the flow is calculated at regions of brightness gradient, corresponding to edges in the image, and is eventually propagated inwards into the regions bounded by the edges.

A problem that Horn and Schunk fail to resolve is the treatment of occluding edges, where the optical flow is discontinuous, violating the assumption of smoothness.

Paquin and Dubois (1983) employed a similar method to the above, but using a slightly modified version of the gradient constraint equation that produces a velocity estimate at a point, based on the intensity values in a 3-dimensional neighbourhood of the point (the third dimension being the time coordinate in a sequence of frames). This is done by modifying the velocity estimate in a given frame by updating it according to the estimate from the corresponding point in the previous frame.

Horn and Schunck's approach was carried further by Nagel (1983). He uses grey-level intensity values to identify regions of the image that exhibit the properties of corners. At such points it is possible to write two simultaneous gradient constraint equations and to solve them explicitly for the 2-D image velocity. An iterative refinement procedure is then used to extend these estimates into image regions that do not exhibit the properties of grey-value corners. This is done by fitting a Taylor expansion to the observed grey values in successive frames. The methodology of Horn and Schunk can be seen as a special case of this approach — they attempt to achieve smoothness in the flow field by postulating a linear variation in velocity across regions where the motion cannot be explicitly determined. A Taylor expansion however is more general than a linear variation, since the Taylor expansion may contain second- and higher-order terms.

The work of different researchers attempting to derive a dense motion field can be characterised by the manner in which the flow estimates are propagated from regions of high information content to areas of low or zero information content.

Ghosal and Vaněk (1996) use a smoothness constraint which has a higher weighting at locations of low intensity gradient and vice versa.

Schunck (1989), in contrast, uses an algorithm for motion boundary detection due to Thompson et al. (1985), and restrict the smoothing to regions between the motion boundaries. This allows the resultant flow field to exhibit

discontinuities such as those due to occlusion.

The determination of disparity between stereo image pairs is a special case of the optical flow problem. Such an image pair is generated by two cameras whose spatial relation to each other is fixed; this is equivalent to two images from one camera that has suffered a displacement, or alternatively the camera can be considered to be stationary and the entire scene to be displaced in the opposite direction. The direction of the displacement being known, it is only necessary to consider motion along this axis, so that the images can be analysed line by line, the lines being taken parallel to the axis of displacement.

Poggio (1984) employs a multi-scale differential approach to the analysis of such stereo image pairs. He operates on the image pairs with a number of filters of descending sizes; the filters are designed so that, when applied to a particular pixel, nearby pixels contribute positively to the value of the convolution result, while those in a region slightly further from the central pixel make a negative contribution. Image disparities are then calculated by comparing "zero-crossings" in one image with those in the other. Zero-crossings are points where the second spatial derivative of intensity is zero; they are points where the intensity is changing most rapidly.

This method is in a sense a token-based approach, since it actually identifies features, viz. zero-crossings, and tracks their motion. Like the method that forms the basis of the present research, this method employs the strategem of using a coarse filter to form an approximate estimate of displacement, then refining the estimate using finer filters. However the method of deriving velocity at each scale is quite different from that used in the present work; we do not explicitly track zero-crossings or any other feature.

Lucas and Kanade use an alternative method to obtain a dense velocity field (Barron et al., 1992). They employ a weighted least-squares fit of local measurements to a constant model for the velocity in each local region of the image. The weighting factor is a window function that gives more weight to constraints at the centre of the neighbourhood than to those at the periphery. The effect of the weighting function is that measurements from regions of constant intensity gradient make a greater contribution to the final estimate.

Duncan and Chou (1992) base their work on the assumption that, in a sequence of several images, a moving edge corresponds to a zero-crossing in the second time derivative of a temporal Gaussian smoothing function. Their method is primarily directed towards the detection of motion, but can also

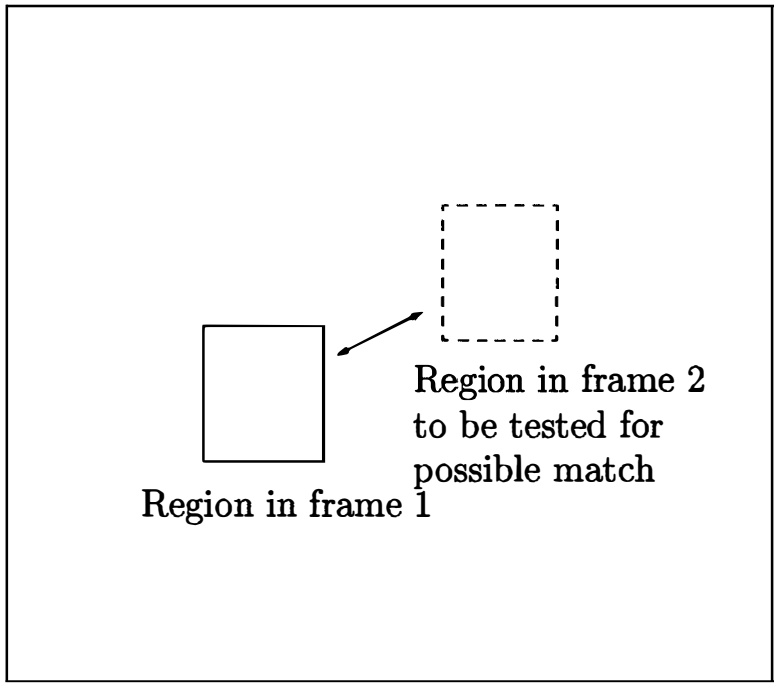


Figure 2.4: Deriving optical flow by region-based matching.

be used to determine the motion normal to the edge (but not motion parallel to the edge). This determination is insensitive to variations in the ovearall illumination of the image.

2.2.2 Region-Based Matching

This methodology consists in applying a particular displacement to a portion of one image, and comparing the result with the other image. The optical flow is estimated as the displacement that results in the best match (Figure 2.4). The result is therefore an estimate of the motion over a whole region, as opposed to the motion at a single point.

One way of measuring the closeness of the match is to compute a similarity measure such as the normalized cross-correlation between the two images (Glazer et al., 1983); another, employed by Heeger (1988), is to use the sum-of-squares of the intensity differences at all points in the images (Barron et al., 1994). Such matching is computationally expensive because of the

large number of point-by-point comparisons required.

A multi-scale approach to region-based matching is described by Glazer et al. (1983). They initially match large image structures using low-frequency filters, to generate a "ball-park" estimate of disparity, then match higher-frequency information at finer levels in order to refine the estimate. Three problems manifest themselves with this approach; these are described in section 2.1.3. These problems appear to be problems intrinsic to the region-based matching approach, rather than problems related to the use of multiple scales.

2.2.3 Energy-Based Methods

Heeger (1988) adopted a method based on the energy output of filters. To derive a best estimate of the image velocity over a given region of the image, the method proceeds as follows:

1. The image sequence is operated on with a series of spatio-temporal Gabor filters. Each filter is designed so that its output is a maximum for a particular image velocity.

2. The "motion energy" at for each of the filters calculated. This is defined as the sum (over the region) of the squares of the outputs from the sine and cosine phases of the Gabor filter.

3. The assumption of any particular 2-dimensional velocity (u, v) over the region results in a corresponding value for the motion energy for each filter. Using a method such as the Gauss-Newton gradient-descent method, find values for u and v that minimise the SSD (sum of squared differences) between the theoretical filter outputs and the actual outputs as measured in the previous step.

Good results are obtained with "natural" textures such as images of rotating spheres and rotating spirals.

A method is given for calculating the uncertainty in the flow estimates, however in some natural image sequences the actual errors turn out to be greater than these uncertainty estimates (Heeger, 1988).

It should be noted that this method is computationally expensive, since a number of velocities must be tested in order to select one that minimises the SSD, and a convolution operation must be performed for each of these candidate velocities.

Burt et al. (1983) implement a multi-resolution methodology in which the flow estimates at each resolution are derived in a manner similar to that described above. They use a number of motion channels corresponding to right, left, up, down and zero motion. They find that the displacement in, say, the x direction is roughly proportional to the ratio of differences between the outputs from the zero channel and the left and right channels. The estimates at each resolution need not be particularly accurate, since the estimates are refined at the level of higher resolution at subsequent stages.

2.2.4 Token-Based Methods

The token-based approach, described in section 2.1, differs fundamentally from the other approaches described in this section, inasmuch as it requires that one or more tokens, or features, be identified in the image and tracked between frames.

An early motivation for work on optical flow was the analysis of the motion of clouds as inferred from satellite photographs. Aggarwal and Duda (1975) illustrate a method for doing this. They work on images containing rigid white polygons on a black background. However they claim that this restriction is for the purposes of computational simplicity, and that the method could be applied to more general images. The problem is one of separating out the objects from one another, and is complicated by the fact that objects may occlude one another (as is common with clouds as seen from a satellite). They delineate the polygons in the individual image frames by identifying the vertices, and the movement of the polygons is then tracked across frames. Success was demonstrated with very simple images, such as an image consisting of two or three polygons moving so as to partially occlude one another.

Roach and Aggarwal (1980) describe work done in inferring the 3-dimensional motion of objects, based on the apparent 2-dimensional motion as captured in digital images. They achieve some success with synthetically generated images. However, they do not address the low-level task of feature recogni-

tion; their work assumes that the image velocities of various features in the image are already available. As stated in section 2.1, the problem of deciding which feature in one frame corresponds to a given feature in another frame is non-trivial, since the number of possible matches increases with the square of the number of features.

In an earlier paper (Roach and Aggarwal, 1979), they describe work done with images of a real-world scene containing blocks that change position between frames. Here the objects are located by finding lines, and then faces, in the image, and the objects are then tracked over a number of frames to determine their motion. Various assumptions are made about the scene; for example, objects are constrained to not move more than one-third of the way across the image between scenes. Clearly the fact that all the objects involved are blocks greatly simplifies the problem of identifying objects, compared with the difficulty of object identification in arbitrary real-world scenes.

Chow and Aggarwal (1977) analyse images of more general scenes. They use thresholding and edge detection to identify image regions that are expected to correspond with objects. They then apply two descriptors – the area and the principal object axes – to each object thus identified, so as to track it between frames. Their method can be applied to scenes with objects of arbitrary shape, but cannot deal with objects that contain holes.

Nagel (1981) also locates objects on the basis of certain features in the individual images. In his case he uses operators that identify points in the image that appear to correspond to corners of objects.

Potter (1975) performs a segmentation of sorts by tracking the movement of "reference details". These are parts of the image, such as grey-level discontinuities, that are expected to correspond to interesting parts of objects. Not every point in the image is associated with such a discontinuity – however, points are assigned a velocity based on the measured velocities of nearby or surrounding discontinuities. Points of the image are then classified on the basis of velocity, giving a segmented image.

The MATCH algorithm (Prager and Arbib, 1983), also tracks the movement of feature points such as edges and corners that have been identified in individual frames. However it sidesteps the Correspondence Problem by calculating the velocity of each feature point in one frame by matching it with each candidate feature point in another frame, but weighting the resultant velocity estimate using a function which rapidly decreases with both

the distance between the feature point positions and the difference between the feature types. A single velocity estimate for the feature point is then obtained using the weighted sum of these velocities. A relaxation operation is then applied to the velocity field so that the resultant field complies with certain smoothness constraints. Barnard and Thompson (1980) employed a similar technique.

2.3 Phase-Based Analysis

A method developed by Fleet and Jepson (Fleet, 1992; Fleet and Jepson, 1993) effectively applies a differential technique to phase rather than intensity. Like Heeger, they convolve the image with a series of velocity-tuned filters. Component velocity estimates are made from the outputs of the different filters. Effectively the features being tracked are points of constant phase.

Given the component velocity estimates from the different filters, a linear velocity model is made to fit each region of the image.

Fleet and Jepson (1993) found local phase information to be a robust quantity in the sense that it is stable with respect to small geometric transformations such as those occurring in projections of 3-D scenes. In other words, a small deformation in the input signal results in only a small change in the image property – in this case phase. This suggested that local phase was a suitable quantity for measuring image velocity and binocular disparity.

A problem with phase-based techniques is the occurrence of phase singularities in the filter outputs. It is necessary to identify where these occur. However Fleet (1992) has presented ways in which one can detect where a singularity will occur, based on the frequency and amplitude derivatives.

El Zaart, Ziou, and Dubeau (1997) also use phase-based analysis to perform disparity estimation between pairs of stereo images, though in their case they estimate the phase difference between two views of the same scene by convolution of the two views with a Gaussian function and its first two derivatives.

2.4 Performance of Optical Flow Measurement Techniques

Some studies have been done on performance comparisons of the various techniques for obtaining optical flow. One group of workers compared methods using both synthetic scenes and real scenes. Correct optical flow data was available for the synthetic scenes only, so that no objective analysis was possible on the real scenes. (Barron, Fleet, Beauchemin, and Burkitt (1992); Barron, Fleet, and Beauchemin (1994)).

The most accurate measurements were found to be:

- the local differential method of Lucas and Kanade,
- the phase-based method of Fleet and Jepson.

The latter method is much more computationally expensive, though with advances in computer hardware development this is likely to become less of a problem. The region-based matching methods, as well as being computationally expensive, were found to perform relatively poorly, particularly in the case of small velocities.

A later study (McCane et al., 2001) stressed the importance of using "ground truth" optical flow data for real-world scenes when performing qualitative assessment of optical flow algorithms. The researchers generated some synthetic scenes with a computed motion field, and also selected some real scenes with restrictive conditions that enable points in different frames to be matched so that the flow field could be derived. They assessed the performance of seven algorithms, and found that the ranking depended on the kind of motion.

For example, for sequences involving camera motion, the performance of a multiscale version of Proesman's algorithm, which employs an anisotropic diffusion method, was relatively insensitive to the complexity of the motion. On the other hand, Nagel's algorithm, which uses second-order derivatives of intensity, performed much worse for cases involving camera rotation than for simpler motions.

2.5 Recent Work

The technique of tracking phase contours of Gabor filters has been applied by Tsao and Chen (1992), who devised a neural computation scheme for extracting optical flow based on the Gabor phase differences of successive images. Their scheme works well both on synthetic images of moving textures and on natural scenes and is robust with changing illumination; however in their test cases the whole image pattern underwent the same displacement, so that the velocity field was constant across the image. Results were not given for images containing features with different velocities within the one image.

Later work by Chen and Bovik (1995) analyses motion in stereo image pairs and uses a multi-scale strategy to produce a dense flow map. Their motion estimates are performed using the reponse from a series of "disparity channels" — filters each of which is sensitive to a particular disparity. (Since the input images are stereo pairs, it is only necessary to consider motion along one axis.) The disparity at each pixel is assumed to be that corresponding to the channel giving the maximum response. The foregoing estimate was carried out at a number of different scales, and the final estimate at each pixel is made using a weighted mean of the estimates at the different scales. Useful results are obtained, but the method would not be applicable to the more general case in which motion may be in any direction of the image plane.

Ouali, Ziou, and Laurgeau (1999) also use a multi-scale strategy to determine optical flow in stereo image pairs. They use the phase output of Gabor filters to estimate disparity at the different scales, then adopt two different strategies for integrating these estimates into a final flow estimate. The first method selects the best disparity as the one having the highest magnitude of the Gabor coefficient, while the second method assumes that the best disparity is the one for which the local spatial frequency is closest to the peak tuning frequency of the filter. They show that these schemes produce roughly the same results.

Fleet (1994) treats binocular disparity using a method that combines his phase-difference technique with another method known as phase-correlation. In using this latter technique, Fleet notes that a Fourier transform is in fact a set of linear band-pass filters, each sensitive to a particular displacement. A Fourier transform is applied to a windowed subset of each of the left and right

images, and the two output functions are combined to yield a function whose peaks in frequency space correspond to disparity estimates. The window must be considerably larger than the expected displacements; this means that some initial information about disparity must be available, and also limits the extent to which the disparity can be localised. Phase correlation is also employed by Papadimitriou and Dennis (1994), who use it to obtain initial high-confidence disparity estimates of block subsets of the image before using these as input to a subsequent area-matching process. This avoids wasted comparison at the area-matching stage and thereby increases the speed and accuracy of stereo matching.

Gökstorp and Danielsson (1994) experimented with velocity-tuned Sobel operators (as an alternative to the velocity-tuned Gabor filters employed by Fleet) for multiresolution flow computation. The Sobel operators are (they claim) good approximations of the derivatives of Gaussian functions Results obtained were comparable with other existing methods. Ghosal and Mehrotra (1994) employed a different set of operators — the "Zernike moments", which are also robust in the presence of changing illumination. When applied to multiple image sequences this gave performance comparable to the Lucas-Kanade and Horn-Schunck techniques.

A different approach to multiresolution flow computation is taken by Yu-Te Wu; Kanade, Cohn, and Li (1998), who represent the motion field as a linear combination of motion wavelets of different scales. The coefficient of each wavelet is estimated for the entire image using a SSD (sum-of-squared-differences) approach to select the coefficient that produces the best match between image frames. The velocity field is then reconstructed by summing the wavelet components, weighted by the derived coefficients.

Chen, Liao, and Lin (1992) also perform a wavelet-based estimation, but in their case by optimising an energy function that is a function of an image constraint and a smoothness constraint, such as the one used by Horn and Schunck. Under the wavelet representation this problem becomes one of solving a linear system of a quadratic and convex function of wavelet scaling coefficients.

Solari, Sabatini, and Bisio (2001) show how phase-based disparity estimation for stereo image pairs can be carried out with no explicit calculation of individual phase values, leading to more efficient hardware and software implementation. Using trigonometric identities they establish a formula expressing the derivative of phase as a simple algebraic function of the real and

imaginary components of intensity that requires no trigonometric operations. This is essentially the same as the method used in sections 4.1.2 and 4.1.3 of the present work in calculating temporal and spatial phase derivatives, and thereby obtaining a velocity estimate.

Fleet and Langley (1995) have experimented with ways of reducing the computational and storage cost of implementing the gradient-based approach by designing a recursive set of filters based on a truncated exponential function. Repeated application of the filter is shown to be equivalent to the application of band pass filters of progressively lower frequency (i.e. giving progressively more smoothing). Although the method represents a substantial increase in efficiency, both in terms of computation time and of storage requirements, the results are in general less accurate than those obtained by the methods listed in Barron, Fleet, and Beauchemin (1994).

The Gradient Constraint Equation (Equation 2.1) is sometimes called the **Optical Flow Constraint**. An alternative constraint equation is sometimes used, called the **Extended Optical Flow Constraint**, which relates the divergence of the brightness field to that of the velocity field. In a paper by del Bimbo, Nesi, and Sanz (1996), they note that there is no agreement among authors about which of these two constraints provides the best velocity estimate, and go on to generate solutions for the velocity field based on both these constraints, smoothing the field according to the assumption that the flow follows a law that is approximately linear in the coordinates (x, y). The test images used included various types of motion — translation, expansion and contraction, rotation in the (x, y) plane and rotation about the x or y axis (i.e. the 2-D image that would result from a 3-D scene containing an object rotating about the x or y axis).

They find that expansion and contraction motions are better detected with their techniques than with earlier models such as the Horn and Schunck approach. For such motion, the estimates based on the Optical Flow Constraint gave better solutions close to the image centre, while the Extended Optical Flow Constraint gave better results in the outer parts of the image.

The uncertainty involved in the gradient constraint for optical flow detection is often modeled as constant Gaussian noise added to the time-derivative of the image intensity. Alternative models have been proposed in which the mean and variance of the Gaussian noise, instead of being assumed to be constant, are functions of the partial derivatives of image intensity with respect to the space and time coordinates. In Ohta (1996) the Lucas-Kanade

assumption was adopted, viz. that the optical flow is constant over a small patch of the image. The maximum likelihood estimates (MLE's) of optical flow were calculated for six different such noise models. Results were claimed to be more accurate than those obtained by the conventional least-squares method. The paper makes the point that only small (sub-pixel) motions were considered, and that for large motions it may be desirable to combine the methods with a multi-scale approach.

Hemmendorf, Andersson, and Knutsson (1999) have employed the phase-based differential approach with some real world images, viz. a medical X-ray angiography sequence, as well as some synthetic images. they find the performance superior to that of the conventional intensity-based gradient method, even for synthetic images with constant illumination, which might have been expected to show good results using the intensity-based approach.

In applying differential techniques to 2-D images, it is necessary to address the Aperture Problem by taking gradient constraint equations at a number of points and combining these equations in some fashion. The traditional method is to apply the gradient constraint equation over all the points in a square subset of the image, and select the best fit for the velocity by the method of least squares. Tsai, Galatsanos, and Katsaggelos (1999) have applied an alternative method called Constrained Total Least Squares, which takes better account of the noise in the image. When applied to video sequences, this method outperforms the least-squares method provided the image subset is large enough; the major disadvantage is its high computational cost.

Yang and Ma (1999) determine optical flow by using a multi-scale image representation and propose the concept of optical flow in "scale-space". This way of viewing optical flow has some commonality with the present work — in Chapter 3 we develop the idea that both optical flow and motion in the real world should be considered as scale-dependent phenomena.

2.6 Research Questions

Some of the problems associated with the determination of optical flow have been outlined above. In particular, the determination of optical flow at any point in the image carries with it an uncertainty, and the velocity estimate

should be accompanied by a statement of the uncertainty.

One source of this uncertainty is the noise inherent in the image data.

Another source is the lack of features in certain regions of the image. As pointed out by Mitiche and Bouthemy (1996, section 3.1), where no features exist in a region, that portion of the image will not change from one frame to the next irrespective of whether motion is present, so that many different optical flow values will be consistent with the observed image sequence. Some possible ways of dealing with this situation are:

- Use a token-based scheme. This avoids the problem of what velocity to assign to areas of the image containing little information, since velocity is assigned only to features (tokens), rather than to each image point.
- Use some means of "propagating" velocity from regions of high information content, for example by using some kind of smoothness constraint as done by Horn and Schunck. The problem with this approach is noted in section 2.2.1, viz. discontinuities in the motion field, such as those caused by occlusion, are not well handled.
- Incorporate an uncertainty measure into the velocity field. Regions containing little spatial variation in intensity, (i.e. having few features) are assigned high uncertainty values; regions with no features at all would be assigned an uncertainty large enough to be effectively infinite. This is the only realistic approach to extracting a dense flow field from image sequences having spatially varying information content.

A third source of uncertainty is the "aperture problem", which occurs for any image whose dimensionality is greater than one. For example, in a 2-dimensional image containing a moving straight edge, it is possible by differential methods (section 2.1.2) to derive a single constraint for motion perpendicular to the edge (figure 2.2). This constraint gives no information about motion parallel to the edge.

2.7 Summary

Of the two major approaches to velocity determination, the token-based approach, while appearing to offer the virtue of simplicity when the images are simple, increases rapidly in complexity as the number of identifiable features in the image increases. It also does not offer the possibility of using motion as an aid to segmentation, since it requires that features ("tokens") be identified prior to estimating their motion.

The dense motion field approach on the other hand has the potential to provide a motion field without the need to first identify image features. Not being based on features, it can be applied to arbitrarily complex images. The main problem with this approach is in deciding how to deal with areas of the image that have low information content, especially considering that real-world images invariably contain noise.

The dense motion field methodologies were further classified according to the method used to estimate the motion at each image point. It was noted that the differential approach, the one used in the present work, presents a number of problems, including the Aperture Problem and the question of how to deal with intensity changes that are caused by something other than the motion of features. Phase-based differential analysis is one way of addressing this latter problem.

Chapter 3

This Research

In this chapter I give some reasons why one might want to determine optical flow. I then proceed to discuss the methodology employed in this research, and the justification for this approach. I give reasons why the motion field itself should be regarded as a scale dependent quantity, a fact that does not appear to have been acknowledged by researchers to date.

3.1 Significance of this Research

Optical flow can be an important aid in interpreting images. If two points in the image are found to have the same or similar optical flow, this may be used to infer that they are part of the same object or feature. A derived image velocity field can be an important aid to segmentation:

- Two image points at which the image velocities are equal or nearly equal may be flagged as being part of the same image feature. They may therefore identify points in the scene that are both associated with the same object, which is translating as a rigid body. On the other hand, if the measured flows are very different at the two points, we might label them as belonging to different features (and therefore corresponding to different real-world objects).
- An object in the scene may be apparently growing or shrinking, per-

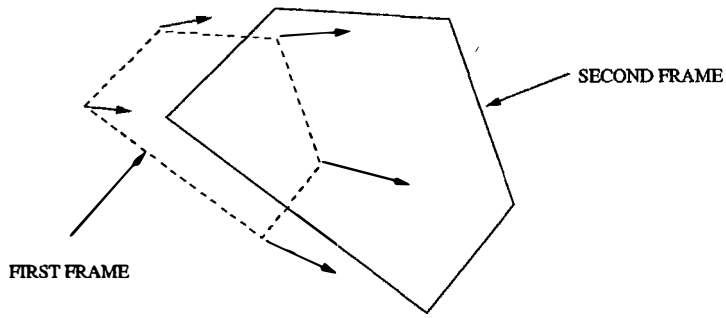


Figure 3.1: Object undergoing simultaneous translation and expansion. Arrows indicate motion at different parts of the object.

haps due to changing distance between the object and the observer (Figure 3.1).

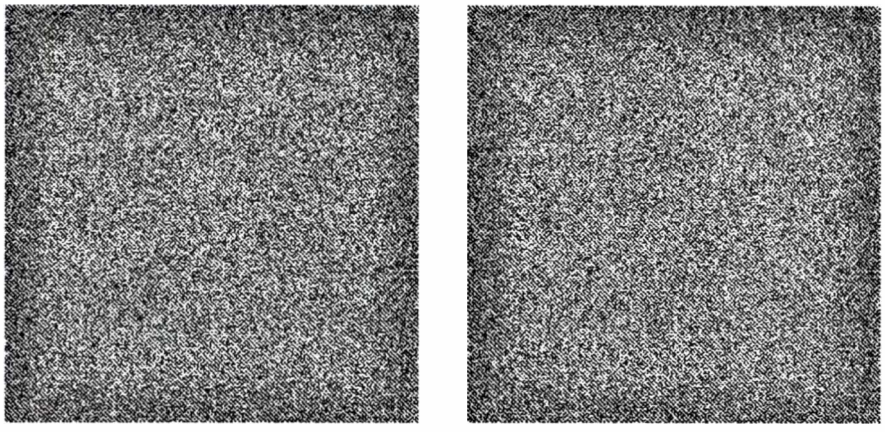
In this case different parts of the object would have different apparent motion, but there would be a smooth gradation of velocity across that part of the image corresponding to the object. This situation could be identified by a suitable analysis, again leading to a grouping of points in the image that we hypothesize as corresponding to points on a single object in the scene.

- A third possibility is an object undergoing solid rotation. Here the velocity vectors would again vary smoothly across the object, though in a different manner from the preceding situation.

- In the more general case an object may undergo any combination of the preceding types of transformation, each such combination resulting in a characteristic velocity distribution across the object.

The human vision system is capable of detecting motion in a series of frames, even where the individual still frames are devoid of identifiable features. This can be demonstrated for example by random dot stereograms (Ballard and Brown, 1982, P.90; Julesz, 1965).

An example of such a random dot stereogram is shown in figure 3.2. The two images are identical except in a square central region, where the pattern in one image is displaced horizontally relative to its position in the other image. (The area disoccluded by this displacement is then filled in with more random intensity values.) The images are viewed as a stereo pair by fixating one eye on the right image and the other on the left image until the two appear to merge into one image, as if they were two views of the



(a) Left image.

(b) Right image, generated from the left image by horizontally displacing a square central region

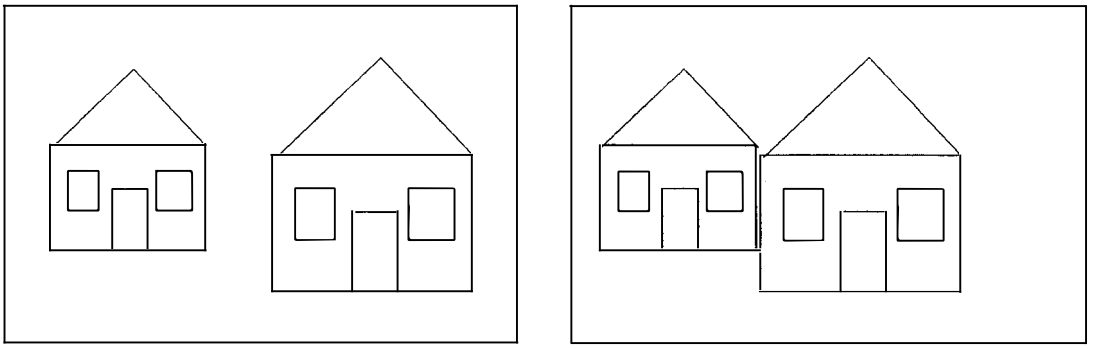
Figure 3.2: Random dot stereo pair giving the illusion of a raised central area.

same scene. When they are viewed in this way, a central square can be seen apparently "floating" above the rest of the pattern. Although neither image contains any perceptible features of a size larger than the individual pixels, the disparity between the two images produces a feature, and the human vision system is capable of perceiving this feature.

The ability to duplicate this result with a computer-based analysis technique would be a good demonstration of the capabilities of the technique.

One potential application of this is in the interpretation of photographic stereo pairs, such as aerial photographs taken from a survey aircraft from two different vantage points. The apparent motion of a feature between two frames (disparity) is related to the distance of the object from the camera.

Another application is autonomous vehicle navigation (Low, 1991, P.219), in which the movement of a vehicle is controlled by software which makes decisions about the surrounding terrain and the objects in its path, based on the images of the scene as sensed by the vehicle. The motion of the vehicle gives rise to apparent motions of objects in the scene; the apparent motion of an object is dependent on the position of the object relative to the vehicle. Identification of regions of apparent motion can therefore be used to identify objects and their positions (Figure 3.3).



(a) Frame 1: First view.

(b) Frame 2: Second view, seen from a different vantage point than the first.

Figure 3.3: Apparent motion of objects as seen from a moving vehicle. The objects exhibit different apparent motion due to their different distances from the vehicle.

Other applications include (Weng et al., 1993, P.1):

- Monitoring of traffic speed and traffic flow.
- Defence applications: moving target detection and tracking (Snyder, 1981).
- Reduction of required bandwidth for image transmission. Instead of transmitting every image in the sequence, transmit only certain images plus a calculated flow field. The flow field can then be used to reconstruct the intermediate images that were not transmitted.
- Diagnosis of heart problems by analysis of heart motion (Tsotsos et al., 1980).
- Velocity estimation of fluid flow. This is useful, for example, in meteorology. One of the earliest dynamic scene analysis problems to gain attention was the automated detection and measurement of cloud motions from satellite photographs (Martin and Aggarwal, 1978).

3.2 Multi-Scale Optical Flow Analysis

3.2.1 The Role of Scale; The Need For a Multi-Scale Approach

All of the techniques that have been used to derive a dense motion field require that one choose a scale or scales at which to measure flow.

For example, in using a differential method, it is necessary to select the size of the interval over which the temporal and spatial derivatives of intensity are to be estimated, or, equivalently, the degree of smoothing to be applied to the input before differential analysis.

In the case of region-based matching, the size of the regions to be matched defines the scale of the analysis.

In phase-based analysis such as the present work, the scale is represented by the width of the complex filter used to perform the convolution.

Any local image velocity estimate is characterized by the scale of the measurement. This refers to the size of the image region over which the measurement is carried out. The size of the region over which the optical flow measurement is made critically influences both the measurement and its interpretation.

Two conflicting considerations affect the choice of scale:

1. Measurement over a large region results in a motion estimate that is characteristic of the region as a whole, leaving a large spatial uncertainty as to the location of the feature(s) associated with the motion. A broad scale measurement will detect movement of only broad scale features, while smaller scale features and their movement may go undetected.

2. A small scale measurement operates on only a small portion of the image at a time. Measurement over a smaller region has the potential to more accurately locate the places where motion is occurring. However, as the focus of analysis is narrowed, a number of problems arise:

    - Large motions may confuse the smaller scale operator, or it may

simply not capture them, because a feature that appears in the filter's spatial window in both frames must of necessity have a motion less than the size of the window.

- As the size of the region decreases, fewer and fewer image points are being used to form a motion estimate, so that the effect of quantization increases, as does the influence of noise in the image data. A further problem with fine-scale analysis is that because the window of observation is smaller, the variation across an image will generally be smaller, giving fewer features to assist in motion estimation. This also means that image noise becomes more dominant at the finer scales.
- A more fundamental problem is the fact that in a complex scene there may be different motions occurring simultaneously at different scales. As an example, consider a moving crowd of people involved in a street demonstration. An observer looking down from an aircraft may be unable to discern individual people but might identify the movement of the crowd as a whole. A closer observer would be able to discern the motions of individuals jostling back and forth within the crowd. At a still finer resolution, given the right measuring equipment, it might in theory be possible to observe the chaotic thermal motions of the atoms of which the people are composed.

  A swarm of bees provides another example; an entomologist might be interested in measuring the speed at which the individual bees move, but a lay person seeing the swarm and worrying about being stung might only be interested in the speed of movement of the entire swarm. At observation scales of about one centimetre, it would be possible, at least in theory, to associate each bee with its own velocity vector. At this scale, there would be many widely differing velocities at different points inside the swarm, while at some points no meaningful velocity could be defined owing to the fact that there was no bee located at that point at that particular moment.

  At a smaller scale, of the order of millimetres, different parts of the insect would be found to have quite different velocities. The wings, for example, would be moving rapidly up and down at speeds considerably higher than the average forward speed of the insect. A physicist would no doubt remark that in fact the individual molecules within the insects are in random thermal motion, with speeds of many hundreds of metres per second. These velocities

exist only at the molecular scale (about 10^{-9} of a metre). They are meaningful velocities to the physicist, but to someone interested in the motion of an individual insect, or of the swarm as a whole, they are irrelevant.

Depending on the purpose of the measurement, small-scale information may be useful, or it may merely serve to obscure the broader-scale flow and make it more difficult to interpret the scene in a coherent way.

In order to be useful, a motion estimate must include some kind of statement of the reliability of the estimate. There are two aspects to this:

- Noise in the original image propagates as uncertainty in the motion estimate. As mentioned above, this effect becomes more important at smaller scales.
- Even in a "clean" image (by which we mean that, apart from pixel quantization effects, the image is a true representation of the light actually emanating from the scene), there may be regions of the image in which there is insufficient information on which to base a flow estimate. For example, in an totally featureless part of the image, it is not possible to have a meaningful concept of motion since there is no feature that can move.

  This effect is also intimately bound up with scale. A textured surface exhibits features at the scale of the texture; at both larger and smaller scales the features may not be apparent.

Clearly there exists a need to measure motion at multiple scales, and to somehow integrate the information from these measurements into a coherent flow model. An important consideration when analysing an image at more than one scale is the manner in which the optical flow and uncertainty measurements at the different scales should be integrated into an overall picture. This is discussed in section 3.2.1.

For phase analysis, the characteristics of the filter function determine the scale of measurement. Referring to the Gabor function shown in Figure 2.3, if the wavelength λ is large, the width, or standard deviation σ, of the filter envelope must be broad so as to accomodate the wave, and only broad scale

information can be retrieved. If λ is small, large movements may transport a feature across a distance comparable to $\lambda/2$, giving a local phase change approaching or exceeding π, and resulting in a flow singularity due to phase ambiguity.

Determination of optical flow is important because of its application in scene segmentation, depth from stereo, determination of egomotion etc. Unfortunately current optical flow algorithms suffer from a tradeoff between resolution and the ability to respond correctly to large motions in the scene. This is because, as stated in section 3.2.1, a flow operator that uses a small enough region to give the high resolution will only be able to capture a moving feature in the two consecutive frames of a sequence if the feature has a motion less than the size of the region used by the operator.

The multi-scale approach is an attempt to combine the advantages of the large scale with those of the small scale by taking measurements at several scales and integrating them into a coherent flow model.

Large-scale filters, which are able to detect large motions, do so by integrating information over a large area of the scene. In the process they lose detail.

Small-scale filters, on the other hand, resolve fine detail but fail to respond at all to large disparities which cause features to be detectable in one frame in a sequence but not the other. There are two ways to address this problem:

- A number of filters can be used, each designed to respond maximally to a particular velocity. All the filters are applied independently to the input, then all the outputs are combined to give a best estimate of the velocity. Fleet (1992), for example, has adopted this approach.
- A multi-scale approach can be used, whereby the filters are applied successively, the output from the broader filters being used to control the application of the small filter. This methodology can be applied to region-based matching (Glazer et al., 1983), by matching large regions to obtain a crude estimate of velocity, then matching smaller regions to refine the measurement.

The present work takes the second approach. The thrust of the work presented here is the implementation of multi-scale phase-based optical flow analysis in one and two dimensions, whereby the velocity output at one scale

is used to control the application of the next (finer resolution) convolution operator. This involves estimating the uncertainty in the flow estimate at each stage. In section 3.2.2 we present our multi-scale approach and the background theory for computing flow and uncertainty from phase and magnitude of the filter output. In sections 4.3 and 5.3, the results of this work are presented.

The rationale for this approach is that the real world seems to consist of objects, which are piece-wise continuous at various scales. If a large-scale feature is detected with a particular velocity, then a reasonable hypothesis is that small-scale features in the immediate neighbourhood are associated with the large feature in some way, and will therefore have the same or similar velocity. Without the above approach, these smaller scale features would go undetected because their motion would carry them out of the range of the narrow filters.

Note that this methodology contrasts with other that of, for example, Ouali, Ziou, and Laurgeau (1999), who derive a number of independent estimates at different scales before combining them in some fashion to produce a best estimate of velocity.

3.2.2 Multi-Scale Analysis Methodology

The methodology of multi-scale optical flow analysis can be summarized as follows:

- A number of spatial scales are selected.
- Analysis is performed at each scale in turn, beginning at the coarsest scale and proceeding to the finest.
- At each step, the output from previous scales is used, along with the information available at the current scale, to derive a new estimate of the flow applicable to the new scale.

The significance of scale was referred to in section 3.2.1, where it was noted that both the velocity measurement and the quality of the measurement depend on the scale over which the measurement is taken. Motion analysis

at the broad scale may completely miss small-scale features of the image, resulting in large areas where the motion field is uncertain because no features are detectable. Analysis at a finer scale may detect new features and may, by estimating the motion of these features, give rise to a velocity field quite different from what would have been obtained through assumptions of continuity in the broad-scale motion field.

On the other hand, if the magnitude of the optical flow between frames is large compared with the measurement scale, we should not expect this motion to be measurable at this scale without some sort of *a priori* estimate being available.

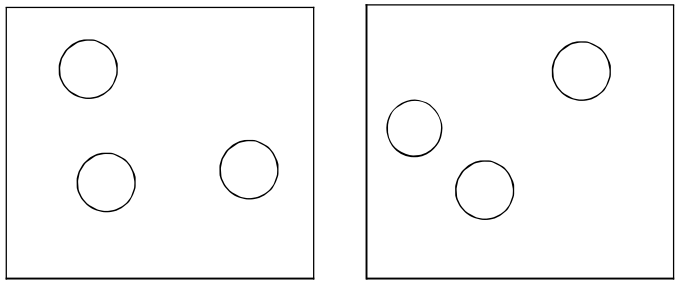
It is misleading to suggest that the velocity measurement made at one scale is more or less "correct" than one made at another scale. Optical flow is in fact a scale-dependent quantity, a fact recognised by Fleet when designing his velocity-tuned filters (Fleet, 1992). He uses a family of 23 filter types of various speeds and orientations.

This reflects the fact that motion in the real world is also scale-dependent, as explained in section 3.2.1. In many real-world situations, it is necessary to specify the scale at which one wishes to examine a system before one can meaningfully discuss the motion occurring within the system.

The dependence of the velocity measurement on scale should therefore not be viewed as being a problem that makes it difficult to find the "true" velocity, but rather as an indicator that the concept of velocity is not meaningful when considered in isolation from scale.

The implication of the above is that when making a velocity estimate we need to clear about the purpose of the analysis. We may wish to measure velocity at the broad scale, at a fine scale, or even at a number of scales.

Given a sequence of images, then, how should the scale-dependent velocity be defined? Since the ultimate goal of image analysis is to infer the real-world system that gave rise to the image, we wish to explain the changes between frames of a sequence as being due to the movement of entities in the real world. Image velocity, then, should be defined as the motion which best explains the evolution between frames of the features present in the image at the particular scale of measurement. There is more than one way to define the term "best explains".



(a) Frame 1. (b) Frame 2.

Figure 3.4: Three moving circles. Which object has moved where?

The **goal** of an image motion methodology is to derive an image motion field for a given sequence or two or more images.

In the simplest case, the intensity changes are due **solely** to motion, i.e. changes of illumination, noise etc. do not play a part. Even in this case it is not always possible to derive a unique motion field. For example if each of two frames contains several identical features in motion (Figure 3.4), it is not possible to be certain which feature in frame 2 corresponds to a given feature in frame 1.

Further complications arise from other effects that cause intensity change, such as noise, changes in general illumination, or occlusion.

In principle we could derive a motion field by applying all possible motion fields to frame 1 and comparing the result with frame 2, then picking the motion field that gives the best match, using some measure such as sum-squared-differences or normalized correlation. This is the region-based matching approach described in section 2.1.3. Leaving aside consideration of the amount of computation required, the problems with this approach are:

- Different motion fields may give similar match (ambiguity).
- Some kind of constraint must be enforced in order to link the motion at neighbouring points, otherwise we may obtain a chaotic motion field that gives a good match measurement but is useless from the point of view of image description and almost certainly does not correspond to the actual real-world motion situation that produced the given image sequence.

The above considerations suggest that although a value of motion is derived for each point, it must be related to the behaviour of the image in a neighbourhood of the point, not just at the point itself.

In proposing the alternative multi-scale approach to velocity measurement, we make the following observations:

- The application of certain filters, including Gabor filters, has a smoothing effect on the image. The degree of smoothing is dependent on the scale of the filter. This should result in the velocity field also being smoothed out to the same degree.
- In derivation of motion at each scale **relative** to the output of the preceding scale, we are measuring the amount by which the motion in a small neighbourhood differs from the motion smoothed over a larger surrounding neighbourhood.
- If motion is indeed continuous over a region, then consideration of successively smaller sub-regions should give converging velocity estimates until the scale is reached where noise begins to dominate the image signal. Conversely, if below a certain scale there are some points where velocity is ill-defined, while at neighbouring points it converges to a well-defined value, this indicates a discontinuity in the motion field, and can be interpreted as two features in the vicinity of the point moving with different velocities. A third possibility is that, below a certain scale, velocity is ill-defined over an entire neighbourhood. This indicates that at this scale no moving features are discernible.
- Even if a smooth motion exists at all scales in the original scene, the presence of noise in the image means that we can only expect the velocity measurements to converge down to a certain scale, since at smaller scales the intensity values will be dominated by noise variations causing the velocity to fluctuate.

3.2.3 Implementation of the Methodology in this Work

The methodology adopted in this research can be summarized as follows:

- Start at the largest scale by applying the coarsest filter.

- At each scale:
    1. Use the velocity estimate already obtained from larger scale measurements to give an *a priori* estimate (except of course at the largest scale, where there is no *a priori* estimate).
    2. Apply a filter corresponding to the current scale.
    3. Use the *a priori* velocity estimate to decide which neighbourhoods in one frame to compare with which ones in other frames.
    4. By this means derive a velocity correction which is added to the previous velocity estimate to produce a new estimate. At the same time derive a new uncertainty estimate.

Another way of describing steps 2 and 3 above is to say that a **spatio-temporal** filter is applied to the image sequence, this filter being **velocity-adjusted** using the previous velocity estimate. This gives fine-scale measurements that could not otherwise be obtained. The reason is that, in deriving the finer-scale motion, the earlier estimate is used to decide which parts of different image frames to compare with one another. What is being measured at each scale, therefore, is the motion **relative** to the previous motion estimate, and this is then added vectorially to the previous estimate to obtain a new estimate of **absolute** motion.

The methodology is described in more detail in section 4.1.6 and is illustrated in figure 4.3. It is implemented by means of a hierarchy of smoothing filters of different sizes. First, the broad-scale motion is estimated using a broad filter, then the estimates are progressively refined by analysing with successively narrower filters.

This multi-scale approach is reminiscent of the work of Bergholm (1987), who carries out edge detection and location by first detecting edges at a coarse scale using a Gasussian smoothing filter, then progressively refining the scale and matching the edges detected at the finer scales with those found at broader scales.

The present research proceeded in the following stages:

1. Test image sequences were generated.
2. The image processing filters were generated, and their operation tested by applying them to the test images.

3. A measure of local optical flow was defined, incorporating an uncertainty measure.

4. Optical flow was estimated in the test images as follows:

   - The images were analysed at the coarsest scale, giving a coarse optical flow estimate and an uncertainty estimate.
   - This was used to assign parameters for analysis at the next scale down, so as to measure optical flow at the finer scale **relative** to motion at the coarser scale.
   - The procedure was repeated at successively finer scales. The process terminated at the finest level of resolution.

5. The results were compared with what theory suggested the motion estimates should be for these test images, and conclusions were drawn as to the performance of the method.

3.2.4 The Output

The output of the analysis is not a single velocity field but a hierarchy of velocity fields at a series of scales. For each point in each field, the analysis gives a motion estimate along with an uncertainty matrix which defines the confidence in the estimate. Each such velocity field output at each scale is potentially useful (section 3.2.1.

Consider an image sequence generated by a white square moving over a black background. Over most of the background, no information exists on which to base a flow estimate, because there are no features. In and around the square, we would expect the velocity estimate to be equal to the actual velocity of the square, at scales comparable with the size of the square. However in the interior of the square, again there is a lack of features and therefore a lack of velocity information. At such points the estimated velocity value and the uncertainty thereof will be related in some way to the distance of the point from the moving edge, which is the only place where motion information actually exists because it is the only feature in the image.

This output format, viz. a succession of estimates at different scales, would be appropriate as part of an image understanding system. Decisions such as whether or not two points are both located on the one feature, and therefore

correspond to points on the one real-world object, could be made by comparing the image velocities at those points, at the scale of the feature whose existence is hypothesized.

This strategy is referred to as a "coarse-to-fine" strategy, which means that:

- Analysis first takes place at larger scales, then proceeds to smaller scales.
- The output is a **series** of velocity/uncertainty measurements at a number of scales.

3.3 Facilities

The work was carried out on an IBM RISC System/6000 desktop computer running the AIX operating system, and later on a Pentium PC running Red Hat Linux 5.2. With these systems it was possible to view an image in one window on the screen while programming in another window, making it possible to monitor in real time the effects of modifying the processing algorithms.

3.4 Software

- The image processing utilities, including the programs to implement the optical flow analysis, were written in C by the author.
- Extensive use was made of the existing IPRS image-processing program library which was developed by Craig Dillon at Melbourne University.

3.5 Summary

This chapter has emphasised that what we refer to as image velocity is dependent on the measurement scale, and indeed that motion in the real world

is a scale-dependent quantity. In recognition of this, the present work measures velocity at a series of successively smaller scales, and the output is not a single velocity field but rather a series of such velocity fields representing image motion at each of the various scales. The estimate at each scale is used to assist the analysis at the next smaller scale; the manner in which this is done was described.

Chapter 4

Optical Flow in One-dimensional Images

Although for most practical applications one would be dealing with 2-dimensional images, analysis of 1-dimensional images is a useful starting point. The restriction to a single dimension results in significant simplifications:

- A 1-dimensional image of size 256 pixels, as used in this work, contains far fewer points than a 2-dimensional 256×256 image. This results in far shorter computation times, an important consideration when developing and testing algorithms and programs.
- The Aperture Problem does not occur in one dimension.
- A single filter function suffices as a convolution kernel. By comparison, in the 2-dimensional case, any one filter will fail to respond to some velocities, so that it becomes necessary to employ at least two different filters at each measurement scale, and combine the filter outputs in some fashion.
- The output velocity estimates can be very easily displayed on a 2-dimensional graph, where the X-axis represents the image coordinate and the Y-axis indicates the velocity estimate.
- The uncertainty of the estimate can be expressed by a single number, compared with the 2-dimensional case where a 2×2 matrix is required.

Accordingly the method was first developed and tested for the one-dimensional case. We describe below the application of a series of convolution kernels at different scales, the phase-based velocity determination at each scale, and the manner in which the output at each scale is is used to assist the estimate at the next smaller scale. We describe the synthetic test images that were generated, and present the results of applying the method to these images.

4.1 Theory

4.1.1 Choice of Convolution Kernels

The class of functions known as Gabor Functions has been shown to have optimum filter behaviour in the sense that they achieve the minimum possible value of the product of spatial extent and frequency bandwidth (Turner, 1986).

A 1-dimensional Gabor function centred on zero has the form

$$\texttt{Gabor}(x;\sigma,k) = G(x;\sigma)e^{ixk} \tag{4.1}$$

where:

- k is the spatial angular frequency of the Gabor function (the spatial frequency is then $k/2\pi$)
- G is a Gaussian function defined by

$$G(x;\sigma) = \frac{1}{\sqrt{2\pi}\sigma}e^{(-x^2/2\sigma^2)} \tag{4.2}$$

- σ is the characteristic width of the Gaussian envelope (Figure 2.3).

Equation 4.1 can therefore be expanded to

$$\texttt{Gabor}(x;\sigma,k) = \frac{1}{\sqrt{2\pi}\sigma}e^{(-x^2/2\sigma^2)}e^{ixk} \tag{4.3}$$

Alternatively we may characterize the waveform by its wavelength λ, which is the reciprocal of the spatial frequency, i.e.

$$\lambda = 2\pi/k \tag{4.4}$$

The parameters of a Gabor filter are:

- the envelope width σ,
- the wavelength λ.

The filter function is given by:

$$\texttt{Gabor}(x;\sigma,\lambda) = \frac{1}{\sqrt{2\pi}\sigma} e^{(-x^2/2\sigma^2)} e^{i2\pi x/\lambda} \tag{4.5}$$

Instead of λ we may employ the spatial angular frequency k, in which case the filter function is given by:

$$\texttt{Gabor}(x;\sigma,k) = \frac{1}{\sqrt{2\pi}\sigma} e^{(-x^2/2\sigma^2)} e^{ikx} \tag{4.6}$$

The span of the theoretical Gabor filter is infinite, since the Gaussian envelope function never actually reaches zero. In practice of course we must truncate the filters. For practical filters, therefore, a third parameter is the length or span l of the filter.

In selecting the parameters of a Gabor filter to use as a convolution kernel, there are two criteria that need to be met:

1. For a particular value of σ, the span l should be large enough so as to include all points that can significantly affect the result of the convolution.
2. We wish the "DC" component of the kernel to be close to zero. The DC component is the result of applying the filter to a region of constant unit brightness in the image. A non-zero DC component would imply that the filter output would be affected by the overall brightness of the region, but we wish the filter to be sensitive only to variations within the region. The DC component is the sum of the filter function

values across the span of the kernel. (It has only a real component; the imaginary component is zero by virtue of the fact that the imaginary component of the filter is antisymmetric about the origin.)

This DC component is dependent on both σ and λ; therefore if σ is specified, this restricts our choice of λ, and vice versa.

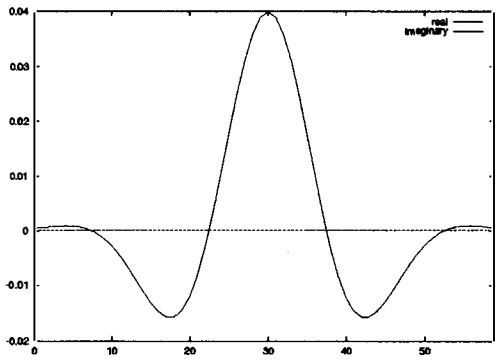
To meet the first criterion, it was found empirically that the kernel function was very close to zero at the boundary provided the length was at least six times the standard deviation σ of the Gaussian envelope. Using the notation of Section 4.1.1:

$$l \geq 6.0 * \sigma \tag{4.7}$$

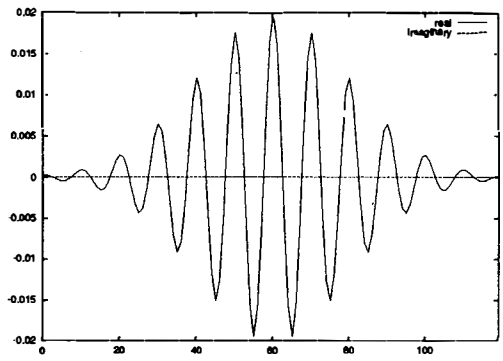
To address the second criterion, we note that the DC component is larger when the wavelength of the Gabor function is comparable to the size of the envelope. This can be understood intuitively as follows: for small values of σ/λ, the Gaussian envelope decays over a distance much less than one wavelength, so that the Gabor function is effectively a Gaussian with an area under the curve equal to 1.0 (Figure 4.1(a)). In the extreme case where the spatial frequency is zero, i.e. the wavelength is infinite, the Gabor function degenerates into a Gaussian curve which of course has a large DC component since the function is positive everywhere. As σ/λ becomes large, the function oscillates between positive and negative values many times before the envelope dies away, allowing the area under the curve to approach zero (Figure 4.1(b)).

A measure of the disparity from zero was obtained for a number of Gabor kernels by means of performing a Fourier transform $\mathbf{F}$ on the kernels. If we denote the resulting function by $(F \circ Gabor)(k')$ we note that $(F \circ Gabor)(0)$ is the output for an image with a single constant intensity of 1, i.e. a filter whose spatial frequency is zero. This is the DC output. On the other hand, the maximum output should occur when $k' = k$. This suggests that an appropriate measure of the significance of the DC output is given by dividing the DC output $(F \circ Gabor)(0)$ by the maximum output $(F \circ Gabor)(k)$.

Experimentally it is found that this ratio depends only on the ratio of σ and λ. In particular, setting a criterion that the DC response be less than 1% of the maximum response, it is found that this condition is met when:



(a) Gabor filter with large DC component: algebraic area under curve is large



(b) Gabor filter with small DC component: positive and negative areas sum to nearly zero

Figure 4.1: Gabor filters with different wavelength-to-envelope ratios, showing the effect of the ratio on the DC component of the filter.

$$\sigma/\lambda \geq 0.485 \tag{4.8}$$

It is not surprising that, since σ/λ characterises the shape of the kernel, this result is independent of the absolute size of the kernel over a wide range of kernel sizes. For very small kernels with a length less than four pixels, the relationship breaks down; this is no doubt due to the fact that our kernels are actually only discrete approximations to a continuous function. For very small kernels, quantisation effects become important.

The foregoing discussion implies that, once the Gabor wavelength of the filter is chosen, we should choose the other two parameters as follows:

$$\sigma = 0.485 \times \lambda \tag{4.9}$$

and

$$l = 6.0 * \sigma \tag{4.10}$$

The final decision to be made, then, is the selection of the wavelength for each filter in the hierarchy. In this we are guided by the following considerations:

- The broadest filter should respond primarily to features of the same order of size as that of the entire image. This means that the filter

span needs to be at least equal to the size of the image. In the case of our test images this width is 256 pixels.

- The envelope width of each filter (i.e. the distance over which its amplitude decays) governs the extent to which we can localise the motion that it can detect; the next filter down in size should therefore have a span at least equal to this width.
- The envelope width of the narrowest filter governs the extent to which we can localise the finest motion that is detected; this should therefore be of the order of one pixel.

Based on the above criteria, a total of 13 filters were employed. The filter parameters λ (wavelength), σ (envelope width) and l (filter span) were set as follows:

Stage	λ	σ	l
1	160.0	77.60	465.6
2	113.0	54.80	328.8
3	80.0	38.80	232.8
4	56.0	27.16	163.0
5	40.0	19.40	116.4
6	28.0	13.58	81.48
7	20.0	9.70	58.20
8	14.0	6.79	40.74
9	10.0	4.85	29.10
10	7.0	3.40	20.40
11	5.0	2.42	14.52
12	3.5	1.70	10.20
13	2.5	1.21	7.26

All the filters have the same shape, differing only in scale. The ratio between wavelengths of successive kernels is approximately $1/\sqrt{2}$. This ensures that the envelope of each filter lies well within the span of the next smaller filter.

Experimentally it was noted that the DC output from the smallest filter appears to be significant, contradicting the result given above that $\sigma > 0.485 \times \lambda$ gives a minimal DC response. This is probably due to quantization effects: the DC response is the discrete sum of a number of terms, which differs from the area under a smooth curve. For detection of very small variations in image velocity, further work may need to be done to fine-tune the smallest

filters in order to minimize this potential source of error. Future work could be directed towards finding whether the use of more or fewer filters significantly affects the accuracy of the results. Clearly the fewer filters the better, since this lessens the processing time, provided the output is still accurate and useful.

4.1.2 Determination of Phase and Phase Derivative

Let the complex filter output be denoted by $C = R + iI$ where R and I are the real and imaginary parts of the output, and $i = \sqrt{-1}$.

Then the phase ϕ is defined by

$$\phi = \arctan(I/R) \tag{4.11}$$

Using the standard identities:

$$\frac{\partial}{\partial x}[f(x)/g(x)] = \frac{g(x)f'(x) - f(x)g'(x)}{[g(x)]^2} \tag{4.12}$$

and

$$\frac{\partial}{\partial x}[\arctan(x)] = \frac{1}{1+x^2} \tag{4.13}$$

we deduce that the spatial derivative of phase is:

$$\partial\phi/\partial x = \frac{R(\partial I/\partial x) - I(\partial R/\partial x)}{R^2 + I^2} \tag{4.14}$$

Similarly, the time derivative of phase is given by:

$$\partial\phi/\partial t = \frac{R(\partial I/\partial t) - I(\partial R/\partial t)}{R^2 + I^2} \tag{4.15}$$

4.1.3 Motion from Phase at One Scale

Any of a wide variety of filters might be used to calculate the optical flow. In our derivation we use the Gabor filter, defined by Equation 4.5.

Application of the Gabor filter to the input image results in a complex valued output image. Our method of velocity estimation employs the phase of this output; velocity is calculated from the spatial and temporal derivatives of phase.

In the manner of Fleet (1992, P. 62), we use the equations for the spatial and temporal derivatives of phase (Equations 4.14 and 4.15) to estimate velocity. This is done by applying the Gradient Constraint Equation (Equation 2.1) to the phase of the filter output. For the case of a one-dimensional image, this equation reduces to

$$\frac{\partial \phi}{\partial t} + v\frac{\partial \phi}{\partial x} = 0 \tag{4.16}$$

which transforms to

$$v = -\frac{\partial \phi / \partial t}{\partial \phi / \partial x} \tag{4.17}$$

to give an explicit expression for v.

Substituting from Equations 4.14 and 4.15, the denominators obligingly cancel, giving:

$$\frac{\partial \phi / \partial t}{\partial \phi / \partial x} = \frac{R(\partial I/\partial t) - I(\partial R/\partial t)}{R(\partial I/\partial x) - I(\partial R/\partial x)} \tag{4.18}$$

Because terms like $R(\partial I/\partial t) - I(\partial R/\partial t)$ appear so frequently here, we introduce the following notation:

$$\psi_t = R(\partial I/\partial t) - I(\partial R/\partial t) \tag{4.19}$$

$$\psi_x = R(\partial I/\partial x) - I(\partial R/\partial x) \tag{4.20}$$

This enables us to write Equation 4.18 more simply as

$$v = -\frac{\psi_t}{\psi_x} \tag{4.21}$$

Since we must work with discrete data, the derivatives $\partial R/\partial t, \partial I/\partial t$, etc. are approximated by finite differences using the values at neighbouring points in space and time. In our work, we restrict ourselves to considering only two image frames. To compute the velocity at a point, the values at four

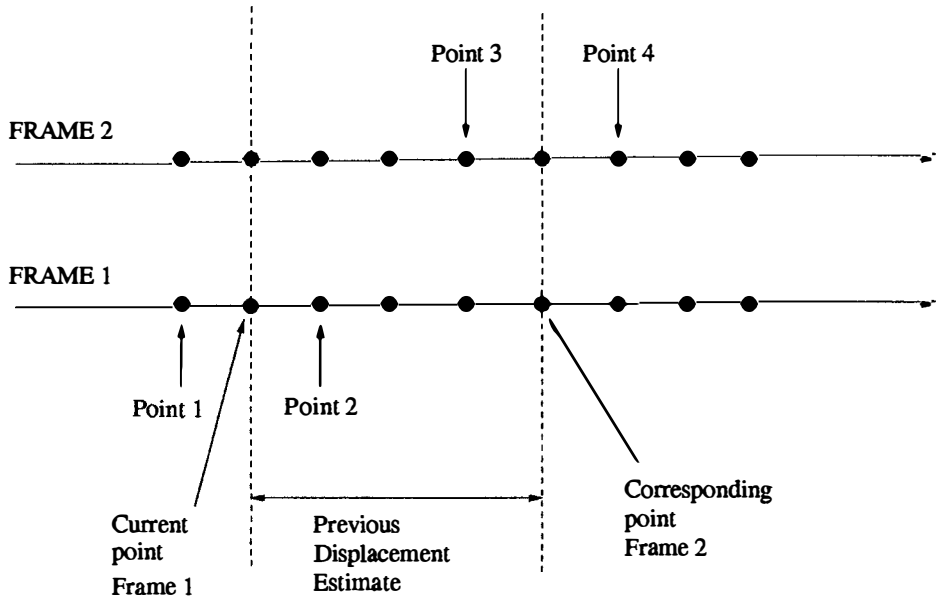


Figure 4.2: Points used to compute spatial and temporal derivatives in using a previous displacement estimate to derive a new estimate.

points are used in taking finite differences (Figure 4.2). The points used in computing the finite differences are shown in Figure 4.2; they are:

Point	Spatial coordinate	Frame
1	$x_1 - 1$	1
2	$x_1 + 1$	1
3	$x_2 - 1$	2
4	$x_2 + 1$	2

Points 1 and 2 are those immediately to the left and right respectively of the current point in frame 1. Points 3 and 4 are the points immediately to the left and right of the corresponding point in frame 2. The velocity estimate is made by comparing the phase behaviour at point x_1 in frame 1 with the behaviour at point x_2 in frame 2, where x_1 and x_2 are offset from each other by a displacement that depends on the velocity estimate at the preceding scale; this is elaborated on in section 4.1.6.

We denote the real and imaginary parts of the convolution outputs at these points by R_1, I_1, R_2, I_2 etc. Then the numerator in Equation 4.18 is calculated as follows:

- Use points 1 and 3 to calculate a value for the numerator,
- Use points 2 and 4 to calculate a second value,
- Take the mean of these to give a final value.

An analogous procedure is used to calculate the denominator.

$$R = \frac{1}{4}(R_1 + R_2 + R_3 + R_4) \tag{4.22}$$

$$\partial R/\partial t = \frac{1}{2}(R_3 - R_1 + R_4 - R_2) \tag{4.23}$$

$$\partial R/\partial x = \frac{1}{4}(R_2 - R_1 + R_4 - R_3) \tag{4.24}$$

with similar expressions holding for I, $\partial I/\partial t$ and $\partial I/\partial x$. Note that the expressions for the temporal and spatial derivatives have different denominators; this is because the spatial interval is twice that of the temporal interval (two pixels as against one frame).

Substituting from the above into equations 4.19 and 4.20 gives expressions for ψ_t and ψ_x in terms of the R_i and I_i:

$$\begin{aligned} \psi_t &= \frac{1}{8}[(R_1 + R_2 + R_3 + R_4)(I_3 - I_1 + I_4 - I_2) \\ &\quad -(I_1 + I_2 + I_3 + I_4)(R_3 - R_1 + R_4 - R_2)] \end{aligned} \tag{4.25}$$

$$\begin{aligned} \psi_x &= \frac{1}{16}[(R_1 + R_2 + R_3 + R_4)(I_2 - I_1 + I_4 - I_3) \\ &\quad -(I_1 + I_2 + I_3 + I_4)(R_2 - R_1 + R_4 - R_3)] \end{aligned} \tag{4.26}$$

Substituting these expressions for ψ_t and ψ_x into Equation 4.21 results in the long-winded but numerically simple expression:

$$v = -2\frac{N}{D} \tag{4.27}$$

where

$$N = (R_3+R_1)(I_3-I_1)-(R_3-R_1)(I_3+I_1)+(R_4+R_2)(I_4-I_2)-(R_4-R_2)(I_4+I_2) \tag{4.28}$$

and

$$D = (R_2+R_1)(I_2-I_1)-(R_2-R_1)(I_2+I_1)+(R_4+R_3)(I_4-I_3)-(R_4-R_3)(I_4+I_3) \tag{4.29}$$

4.1.4 Singularities

Two kinds of singularities can occur in the data: **phase singularities** and **flow singularities**. Both lead to unreliable flow measurements.

Phase singularities occur where both the real and imaginary parts R and I of a complex filter output are zero, so that the phase output, which depends on I/R as given by Equation 4.11, is indeterminate. In the neighbourhood of such points, where R and I are close to zero, phase varies chaotically due to the influence of noise, and the phase derivative becomes useless for measuring velocity. Fleet (1992, pp 124-125) detects these regions using an **amplitude constraint** (Equation 2.6) and a **frequency constraint** (Equation 2.7). In our work, however, we estimate the uncertainty in the flow at every point, at the same time as we calculate the flow itself. Values of I and R close to zero will have a large relative uncertainty, so that they will result in a large uncertainty in the ratio I/R and hence in the phase. This uncertainty is calculated at each point; there is no need to do an additional check for proximity to a phase singularity.

Flow singularities occur where the spatial phase derivative $\partial\phi/\partial x$ is zero. If the time derivative $\partial\phi/\partial t$ is also zero, v is indeterminate; if $\partial\phi/\partial t$ is non-zero, this equation gives $v = \pm\infty$. For an image with phase derivative close to the frequency k of the filter, a displacement of $\lambda/2$ corresponds to a phase change of π at a given point. This is indistinguishable from a phase change of $-\pi$. For this reason, when processing with each filter, derived velocities greater than $\lambda/2$ are rejected. They are rejected not because they are necessarily false, but because we cannot have confidence in the measurement.

In this context, "rejected" means that the velocity is assigned some value (eg. zero), but the uncertainty is set to a very large value. In our case a value of 256 pixels per frame was selected as representing high uncertainty, since this is the width of our test images. A movement of this amount between frames would not be meaningful; there is no way that it could be detected.

A single test identifies points where either the spatial derivative is zero, or is non-zero but such as to give too high a value for v: they are points that satisfy the inequality

$$\left|\frac{\partial\phi}{\partial t}\right| > \left|\frac{\partial\phi}{\partial x}\right| \left(\frac{\lambda}{2}\right) \tag{4.30}$$

This test is applied prior to actually calculating v, avoiding the possibility of attempting to divide by zero.

4.1.5 Uncertainty Estimation

In any natural image, the intensity values will be affected by noise superimposed on the input signal. Such noise can be due, for example, to imperfections in the camera used to capture the image, or electrical fluctuations in the equipment used to digitise the image. In our work we assume the existence of additive Gaussian noise of variance σ_I^2 superimposed on the input signal. (σ_I is not to be confused with σ, the width of the convolution kernel.) A value for σ_I^2 of 1.0 is common with modern digitizers (personal communication, James Cooper, April 1997). Since the input is digitized, we could expect σ_I^2 to be at least of order 1.0. The actual value may be higher, and will depend on the characteristics of the equipment used to capture the image.

In the present work a value of $\sigma_I^2 = 1.0$ is assumed.

We need to determine how this variance translates into a variance σ_v^2 of the velocity measurement. We need to know this quantity, firstly because it forms part of our desired output, and secondly because the uncertainty in the velocity estimate at each stage is used in obtaining the velocity estimate at the next stage, using the next smaller scale filter.

First we note that when the input image is convolved with a Gabor filter of the form given in Equation 4.5, it can be shown that the variance of the output at each point is inversely proportional to the width σ of the filter envelope. A detailed analysis can be found in Cooper and Hastings (1997), where it is shown that variances of both the real and imaginary parts of the filter response are given by

$$\sigma_r^2 \approx \sigma_i^2 \approx \frac{\sigma_I^2}{4\sigma\sqrt{\pi}} \tag{4.31}$$

We denote both σ_r^2 and σ_i^2 by σ_C^2, the C indicating that this is the variance of a component of the convolution output. Now referring to Equations 4.27, 4.28 and 4.29, the variance of terms like $(R_2 - R_1)(I_2 + I_1)$ cannot be calculated exactly since the R_i and I_i are not independent. However, estimating the variance of these terms by expressions like $(R_2 - R_1)^2 \times \mathtt{Var}(I_2 + I_1) + (I_2 + I_1)^2 \times \mathtt{Var}(R_2 - R_1)$ should give a good order-of-magnitude estimate. Using the identities

$$\sigma_{x+y}^2 = \sigma_x^2 + \sigma_y^2 \tag{4.32}$$

and

$$\sigma_{xy} = |x|\sigma_y + |y|\sigma_x \tag{4.33}$$

we can show that the term N in Equation 4.27 has approximately the variance

$$\sigma_N^2 \approx 2\sigma_C^2[(R_3 + R_1)^2 + (I_3 - I_1)^2 + (R_3 - R_1)^2 + \cdots] \tag{4.34}$$

A similar expression gives σ_D^2, the variance of the term D. Taking the square root of each gives the standard deviation.

Finally the uncertainty in the derived value of v is calculated. Using the identity for the standard deviation of the quotient of two quantities:

$$\sigma_{x/y} = \frac{\sigma_x}{|y|} + \frac{|x|\sigma_y}{y^2} \tag{4.35}$$

we obtain the standard deviation of v in terms of the deviations in N and D:

$$\sigma_v = \sigma_{(2N/D)} = 2\left(\frac{\sigma_N}{|D|} + \frac{|N|\sigma_D}{D^2}\right) \tag{4.36}$$

This calculation is performed in conjunction with each velocity estimate to give an optical flow field and an associated uncertainty field. Where the velocity was rejected as unreliable, the uncertainty value was treated as being equal to the image width. The rationale for this is that since the image width is the size of our observation window, a displacement greater than this is undetectable, so that an uncertainty of this size signifies that no useful displacement information is available.

4.1.6 Combining Measurements at Multiple Scales

Analysis proceeds as follows:

- At some stage in the processing, let $d_n(x)$ denote the current estimate of the displacement at the point x.
- The new estimate $d_{n+1}(x)$ is made by comparing the neighbourhood of the point x in frame 1 with the neighbourhood of the point $x + d_n(x)$ in frame 2 (Figure 4.2). In other words, the frames are offset from one another by an amount $d(x)$. The points 1, 2, 3 and 4 referred to in section 4.1.3 are selected in this way.
- Applying Equation 4.27 at these points gives an estimate of **relative inter-image displacement**. We denote this by $\delta(x)$. Since we are dealing with only a 2-image sequence, this quantity can also be referred to as the **relative velocity**.
- This correction is added to the previous estimate to give $d_{n+1}(x) = d_n(x) + \delta(x)$.
- An uncertainty measure for $d_{n+1}(x)$ is also calculated at this time.

The relative displacement $\delta(x)$ can be thought of as a correction to the previous displacement estimate, based on measurement at a finer scale. This mimics the methodology of Glazer et al. (1983) by using the coarse scale estimate to select the points to be compared with one another in producing the finer scale estimate.

Each of the displacement estimates, d_n and d_{n+1}, carries its own variance estimate; these can be denoted σ_n^2 and σ_{n+1}^2.

The flow of data through the multi-scale algorithm is illustrated in Figure 4.3. At the first stage (the largest scale), the displacement and uncertainty are calculated from two convolution inputs, one from each image frame. At subsequent stages, relative displacement and the associated uncertainty are calculated using three inputs; the convolution outputs from the two frames, plus the previous displacement estimate. The new flow estimate is then derived from the relative displacement by simple addition.

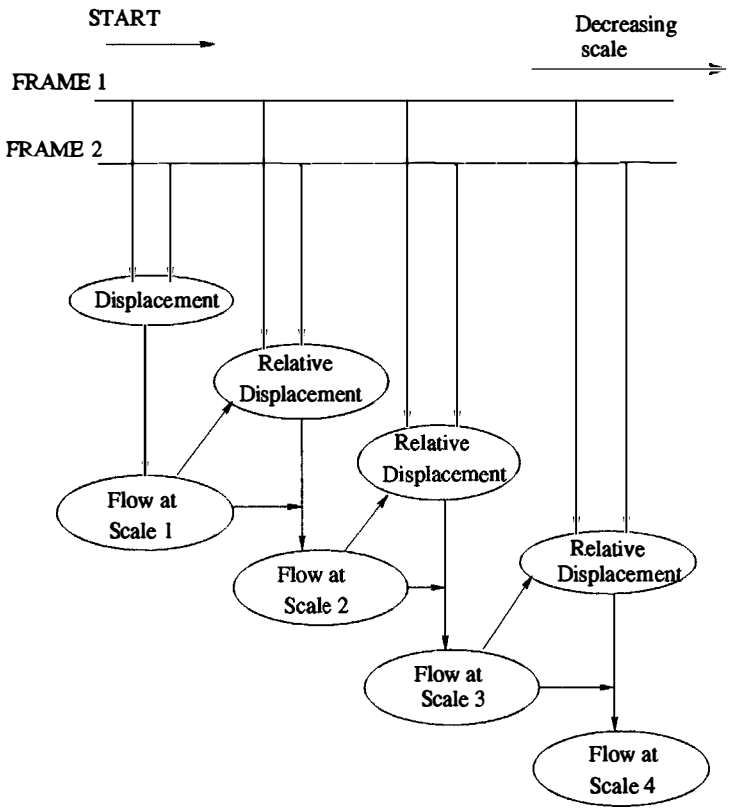


Figure 4.3: Data flow through the algorithm. From stage two onward, the velocity and uncertainty information from the previous stage is used to guide application of the smaller filters so that they do not try to measure velocities larger than about half a wavelength.

4.2 Test Methodology

4.2.1 Test Design

To evaluate our methodology of deriving image motion as applied to 1-dimensional image sequences, a number of synthetic 1-dimensional image sequences were generated for testing purposes. They are listed and presented in section 4.2.2.

The images were designed to test the following desirable characteristics of a motion detection method:

- In regions of the image where features are well defined, the correct motion should be detected. The "correct" motion here means the motion that was actually used in generating the test image sequence.
- In regions of low information content, it is not important that the derived motion have the correct value, but the calculated uncertainty should reflect the lack of information.
- The method should be insensitive to changes in overall scene illumination.
- Where different features have different motions, the different motions should be detected, and the resultant flow field should allow the features to be resolved.
- Small-scale feature motion superimposed on large-scale feature motion should be detected at finer scales even if it was missed at the coarser scales. This should hold true even if the magnitude of the absolute motion is greater than the span of the finer scale filters.
- The method should be able to determine large velocities with high resolution. We expect our method to do this by virtue of its multi-scale nature; the broad scale filters should identify regions of the image that exhibit large velocities, then the smaller scale filters should resolve neighbouring points in these regions that have large but slightly different velocities.

It would have been possible to use real-world images for testing, by using 1-dimensional "slices" of 2-dimensional image sequences. However it is ques-

tionable how "natural" such images would be, and we would not expect our 1-D analysis to extract two components of motion, or even to do well at extracting a single component, owing to the extremely small amount of information available in a single image slice as compared to the whole 2-D image. For these reasons it was decided to postpone the use of real-world test images until the image motion methodology was extended to two dimensions.

4.2.2 Test Images

The test images are described in this section, the results for these images being presented in the following section. The test images used were all 1-dimensional images with a width of 256 pixels. Each test sequence consisted of a pair of such images.

In order of increasing complexity, the test sequences are:

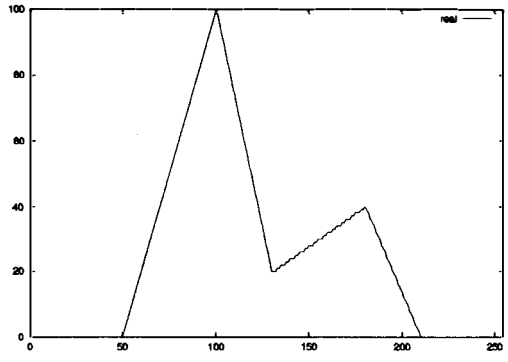
1. Rigid object moving with $v = +5$, i.e. the motion between frames 1 and 2 is 5 units to the right. Constant illumination. (Figure 4.4)

2. Rigid object moving with $v = +5$, as above, but with changing overall illumination. Note the change in the vertical scale between frames 1 and 2. (Figure 4.5)

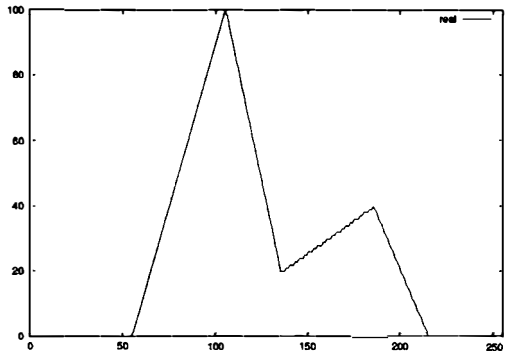
3. Multiple features moving at different velocities. In order from left to right, the spike-shaped features have velocities of $+5$, $+1$, -3 and -6. This kind of velocity distribution would be observed by a camera moving away from a collection of objects located at different distances from the camera. (Figure 4.6)

4. Large moving feature on a moving patterned background. A large region of intermediate intensity and velocity $v = +3$ is superimposed on a background of alternating bright and dark regions with $v = +5$. This is a 1-dimensional analogue of what might be seen by a moving observer looking at a large moving object in front of a picket fence. (Figure 4.7)

5. Superimposed features of similar scales with different velocities. A small object with $v = -1$ is superimposed on a larger object with $v = +5$. (Figure 4.8)



(a) Frame 1.



(b) Frame 2.

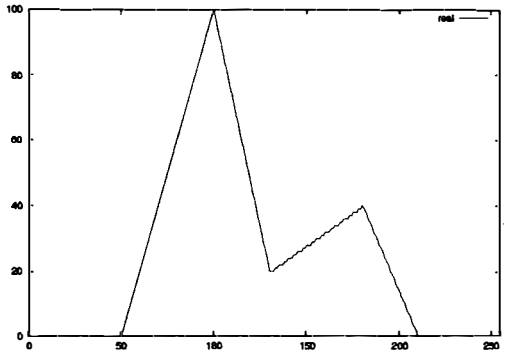
Figure 4.4: Test image 1 – Rigid moving object, constant illumination. Velocity = +5 (5 units to the right).

6. 1-D random dot stereogram. This is a 1-D analogue of the stereo pairs used to demonstrate the ability of the human vision system to obtain information from image pairs even when there is no useful information in either of the images taken in isolation (Julesz, 1965). The intensity values at all points in Frame 1 were set using a random number generator. Frame 2 was generated from Frame 1 by shifting the central portion (between $x = 64$ and $x = 192$) to the right by 2 units ($v = +2$), and the remainder by 5 units ($v = +5$). (Figure 4.9)

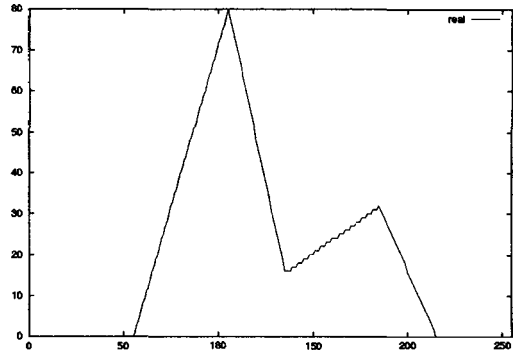
4.3 Test Results

The objectives of this research were to produce a multi-scale one dimensional optical flow algorithm whose output would include both flow values and uncertainty estimates at each point. The algorithm needs to avoid the tradeoff between the resolution of output and the size of disparity that it can deal with. Results are presented to demonstrate the extent to which these goals were achieved.

The output of the multi-scale analysis is not a single velocity at each image point, but rather a **sequence** of flow fields obtained at different scales, each with its corresponding measure of uncertainty. Only for those regions where



(a) Frame 1.



(b) Frame 2.

Figure 4.5: Test image 2 – Rigid moving object with velocity +5, changing illumination. Note the change in the scale between frames.

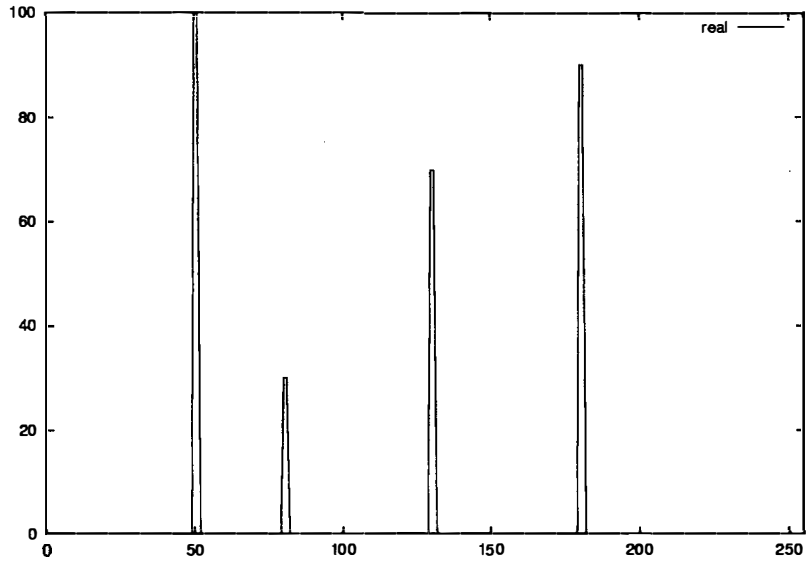
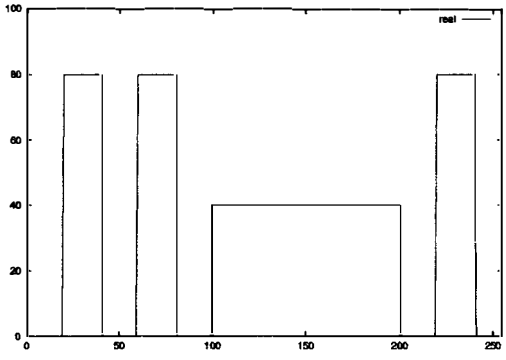
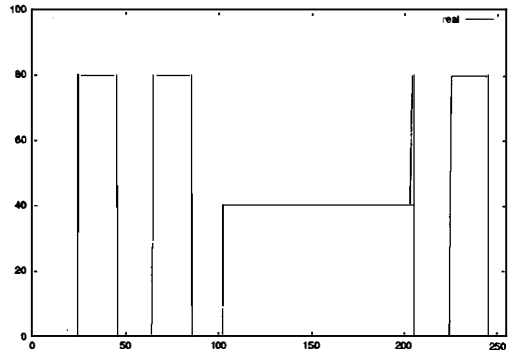


Figure 4.6: Test image 3 – Multiple features, different velocities (Frame 1). From left to right, the four objects have velocities of +5, +1, -3 and -6.

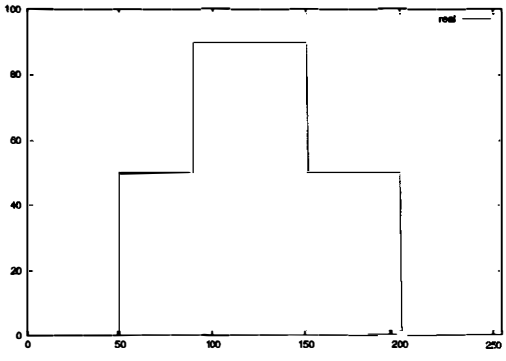


(a) Frame 1.

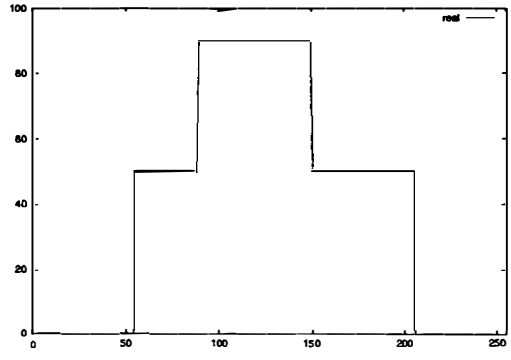


(b) Frame 2. A small object has been partly disoccluded due to its movement relative to the large object.

Figure 4.7: Test image 4 – Large central object with velocity +3, superimposed on a background pattern with velocity +5



(a) Frame 1.



(b) Frame 2.

Figure 4.8: Test image 5 – Small object, velocity -1, overlaying large object, velocity +5.

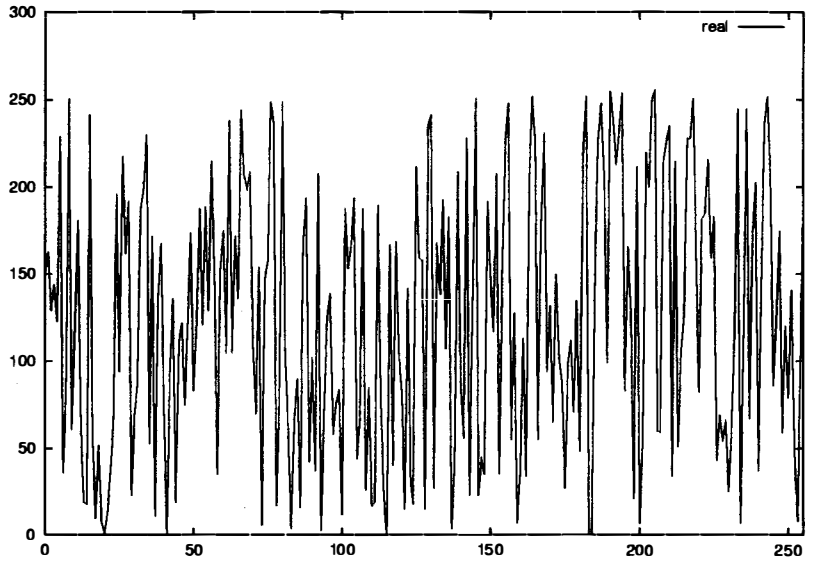


Figure 4.9: Test image 6 – 1-dimensional random dot stereogram (Frame 1). Central portion has velocity +2, remainder has velocity +5.

information is available at the fine scale do these velocities converge to a well-defined value. Elsewhere they fail to converge and the uncertainty becomes very large. This reflects the fact that there are some points in the image where information at the fine scale is not available.

A selection of test results is presented, exhibiting the performance of the method for different kinds of image input sequence. Further results are presented in Appendix A, where the complete output sequences for test images 1, 3 and 6 are given.

In the figures showing the outputs, the estimated flow between frames 1 and 2 is shown as a function of spatial coordinate x, with dark shaded areas above and below this function showing the calculated standard error. For clarity, the values are omitted where the uncertainty exceeded a threshold value — in all cases shown here, the threshold uncertainty value was 3.0 pixels per frame. These regions appear in the figures as gaps in the output.

The first test image (Figure 4.4) showed a rigid object moving with $v = +5$. The output from one of the larger filters (Figure 4.10) shows that a uniform rigid movement is detected very early and with a high degree of precision as regards the value of v. However, information is lacking about which parts of

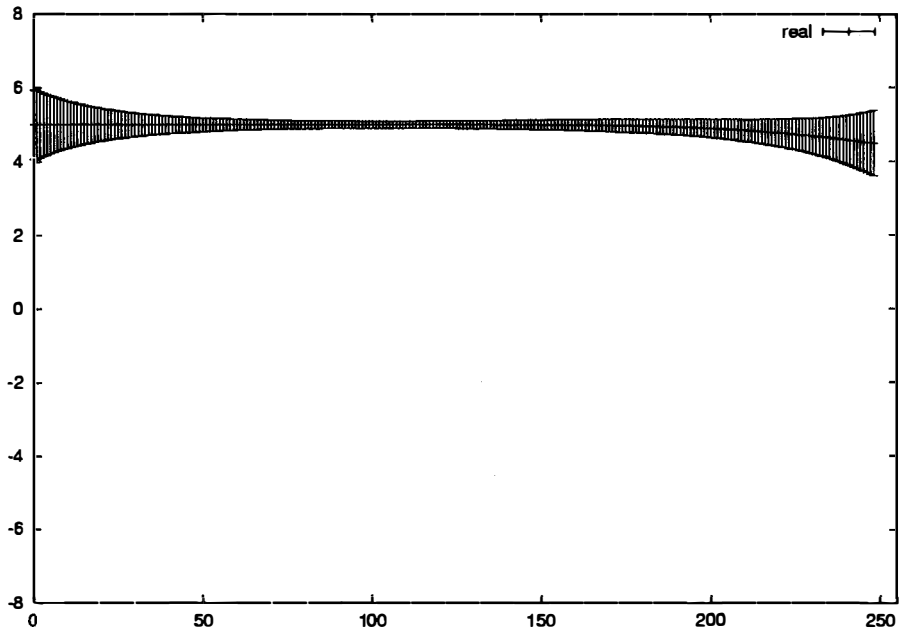


Figure 4.10: Test image 1, output stage 3. Filter wavelength $\lambda = 80$.

the image are contributing to this velocity measurement.

Figure 4.11 shows the output at a later stage of the algorithm, after several successively finer filters have been applied.

It will be noted that the velocity estimates at five regions in the image are still consistent. These regions are those corresponding to significant features in the original image, namely those places where there is a rapid change in the behaviour of the intensity profile. Given no information except Figure 4.11, one could reasonably conclude that the five regions are all part of the one object moving at one velocity.

By the time analysis has proceeded down to filter 10 with $\lambda = 7$ (Figure 4.12), the only reliable filter response occurs at the points near the centre where the intensity profile behaviour changes very rapidly. Elsewhere in the image, the filter output is swamped by noise or singularity errors.

Test image 2 (Figure 4.5) shows the same situation as test image 1, except that it simulates the effects of changing overall illumination. Frame 1 is identical with frame 1 of the first test image; frame 2 however is derived from test image 1 frame 2 by scaling down all the intensity values by a factor of 0.8.

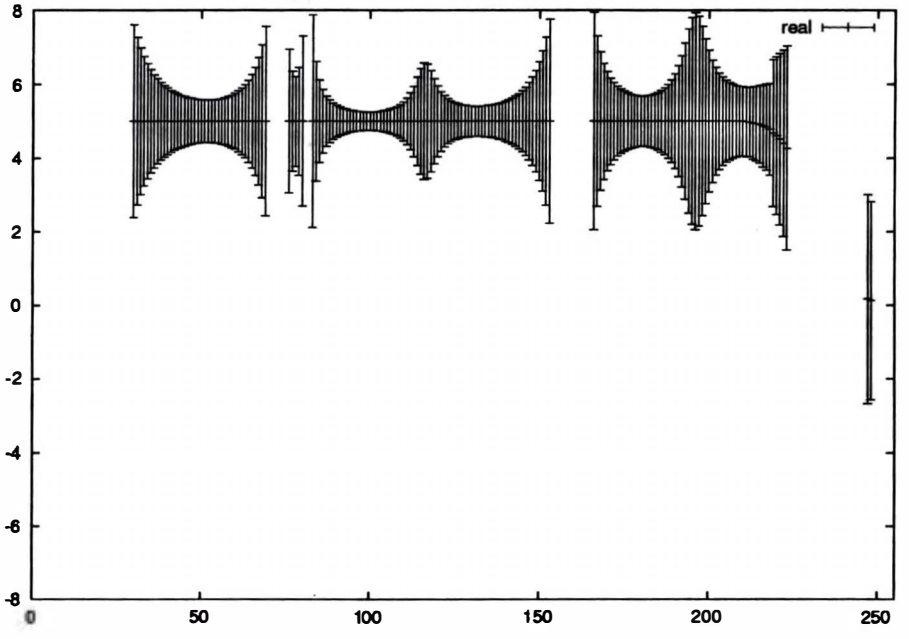


Figure 4.11: Test image 1, output stage 6. Filter wavelength $\lambda = 28$.

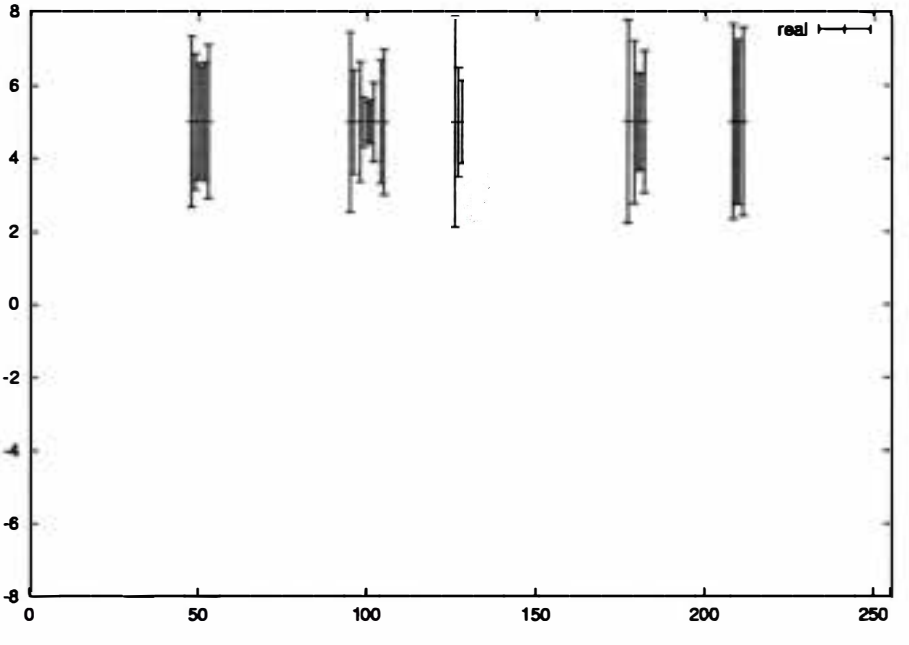


Figure 4.12: Test image 1, output stage 10. Filter wavelength $\lambda = 7$.

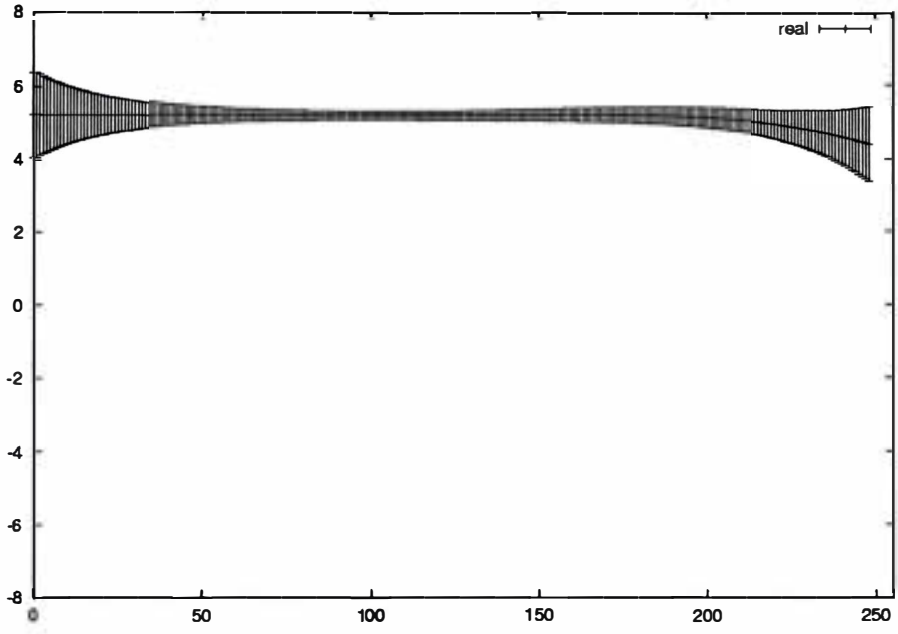


Figure 4.13: Test image 2, output stage 3. Filter wavelength $\lambda = 80$.

The behaviour of the outputs is indistinguishable from that of the previous image, indicating that the change in illumination has no effect on our particular method. The output produced by filter 3, for example (Figure 4.13) looks exactly the same as that from the first test image (Figure 4.10). This supports the assertion that phase robust with respect to illumination levels.

In test image 3 (Figure 4.6), four spike-shaped features were moving with different velocities.

At $\lambda = 80$ (Figure 4.14), the flow field in the central region of the image has no well-defined value. This is because the broad filter can "see" more than one object and cannot resolve the velocities. At $\lambda = 14$ (Figure 4.15), the velocities are well resolved, and even more so at $\lambda = 7$ (Figure 4.16), where the positions of the moving objects have been located with high accuracy.

For the purpose of comparison, Figure 4.17 shows the result of applying only stage 10 of the process to this test image, without any of the preceding stages. Image regions of high motion information content are still identified, but the derived motion is incorrect. The actual displacements of the features in this figure are comparable to the filter half-wavelength ($\lambda/2 = 3.5$), and in some cases are greater than this half-wavelength. Without the assistance

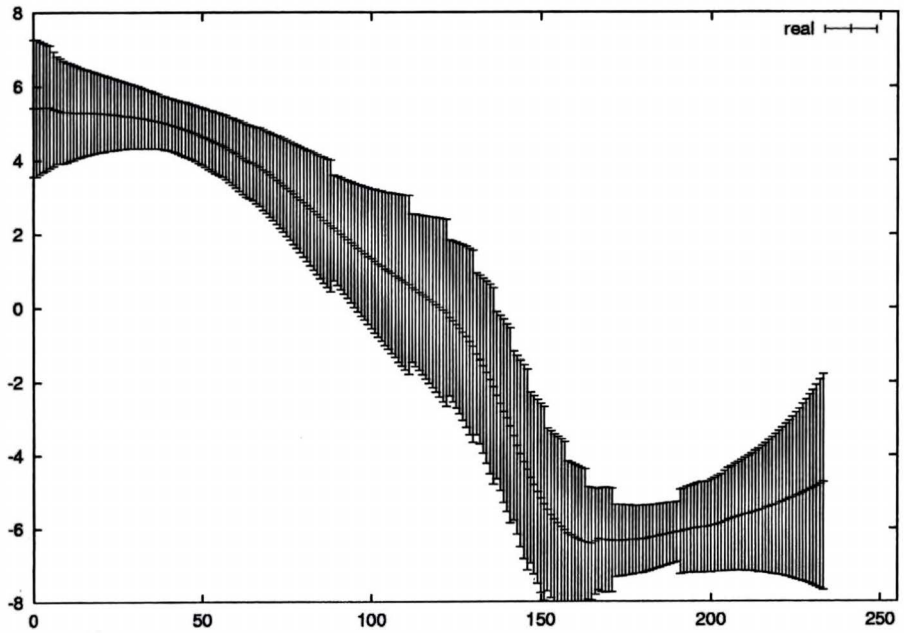


Figure 4.14: Test image 3, output stage 3. Filter wavelength $\lambda = 80$.

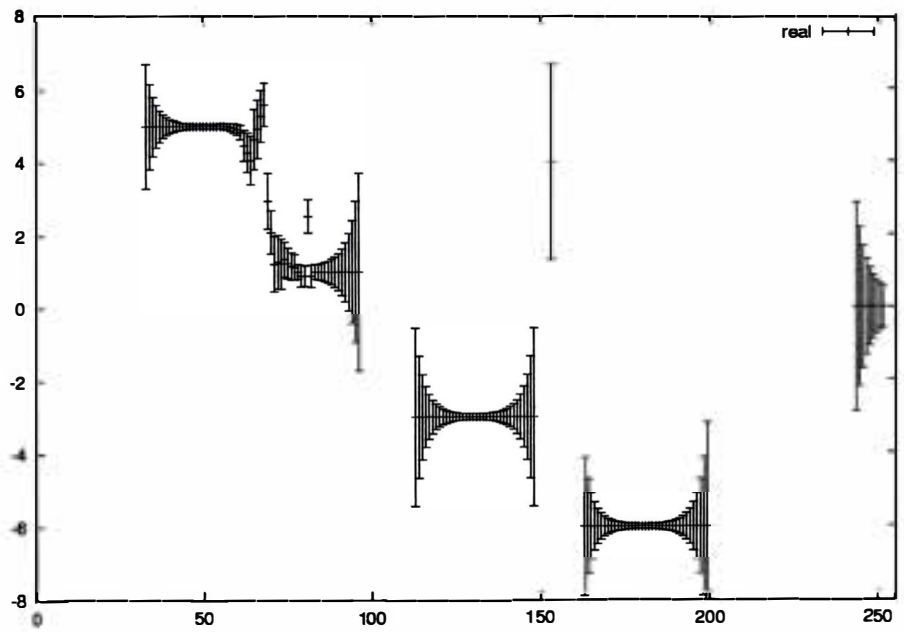


Figure 4.15: Test image 3, output stage 8. Filter wavelength $\lambda = 14$.

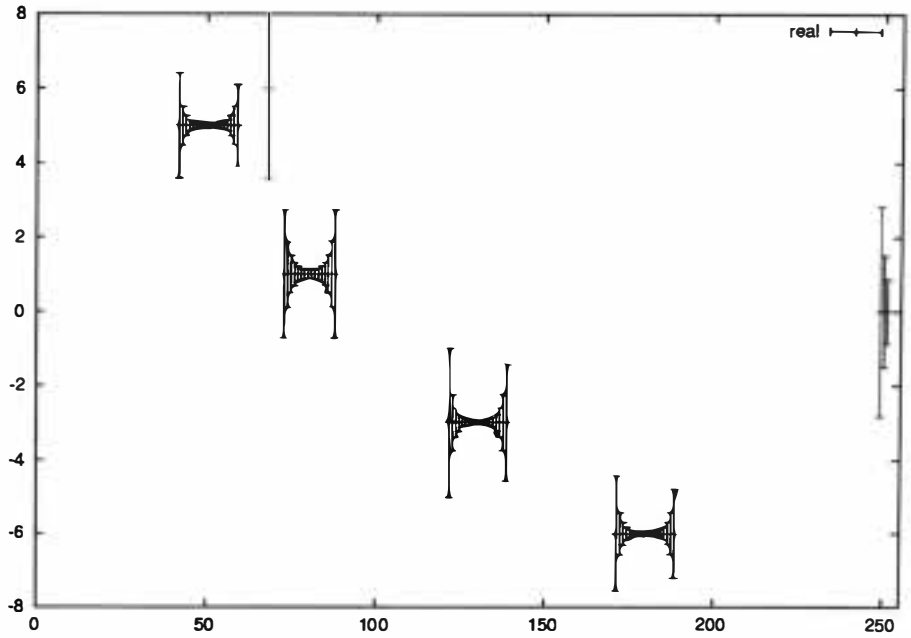


Figure 4.16: Test image 3, output stage 10. Filter wavelength $\lambda = 7$.

of broader filters, the method is not able to handle motions of this size.

The fourth test image (Figure 4.7) consisted of a large moving feature on a moving patterned background.

The method is partially successful in analyzing this rather complex scene (Figures 4.18 to 4.20). The movement of the "pickets" is captured well, as is the motion corresponding to the left hand edge of the large object. The motion associated with the right edge of the large object was missed, probably because the filter responded strongly to the disocclusion of one of the "pickets", so that this obscured the response from the moving edge of the large object.

It is easy to understand intuitively why this is so; in the vicinity of the left edge of the large object, the changes in image intensity at each point can be explained fully by the movement of this object. At the right edge, however, this is not the case. An intensity-based differential method would not be expected to give a correct result at the point of disocclusion. It is therefore not surprising that this difficulty is also experienced with a phase-based differential flow method. Admittedly this particular test image could be regarded as a pathological case, exhibiting the rare circumstance where a

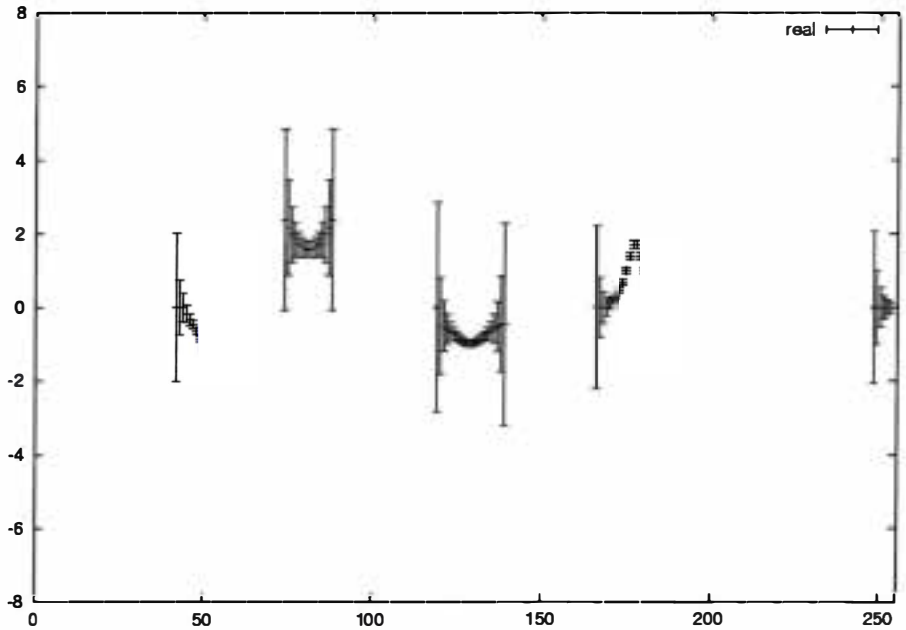


Figure 4.17: Test image 3. Output from stage 10 filter used in isolation — Filter wavelength $\lambda = 7$. Velocities are incorrect.

certain feature (i.e. the occluded feature) appears in one frame and is entirely absent in the other.

Test image 5 (Figure 4.8) simulated the scenario of one object moving across another of similar size. The output from $\lambda = 160$ is given (Figure 4.21), to show that at the large scale the filters can neither resolve the individual velocities nor localize their source. At $\lambda = 40$ (Figure 4.22), however, three regions of more or less constant velocity profile are already identifiable, and at $\lambda = 10$ (Figure 4.23), the location of the moving features has been well defined. Note that Figures 4.22 and 4.23 could well lead to the formulation of a hypothesis that the rightmost and leftmost features belong to a single object with another object moving in front of it.

The final image (Figure 4.9) was a one-dimensional analogue of a random dot stereogram.

Perhaps surprisingly, even by stage 5, with a filter as large as $\lambda = 40$ (Figure 4.24), the flow estimates are starting to converge over much of the central region of the image, though the values are misleading elsewhere. Finer resolution with smaller filters (Figures 4.25 and 4.26) gives realistic results right

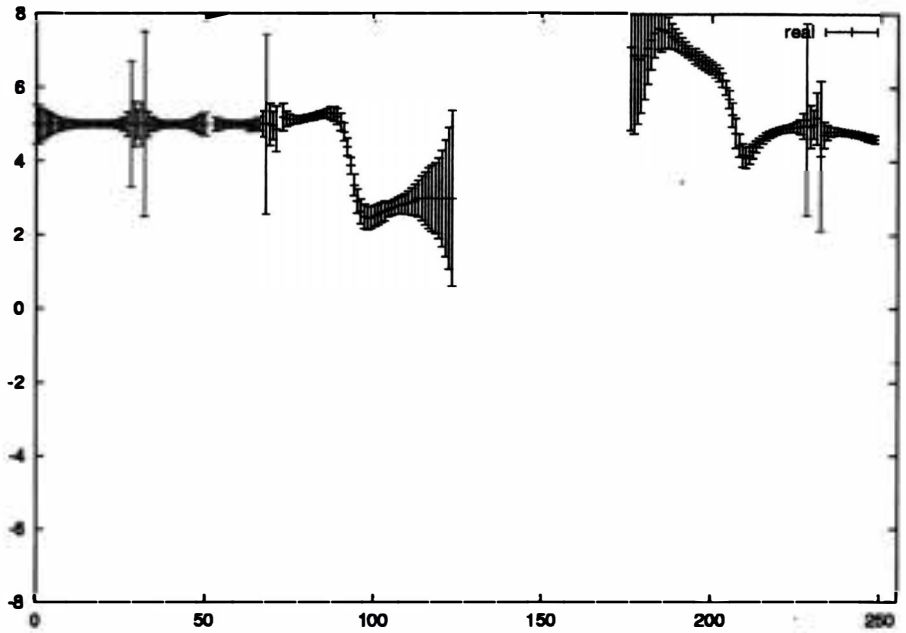


Figure 4.18: Test image 4, output stage 7. Filter wavelength $\lambda = 20$.

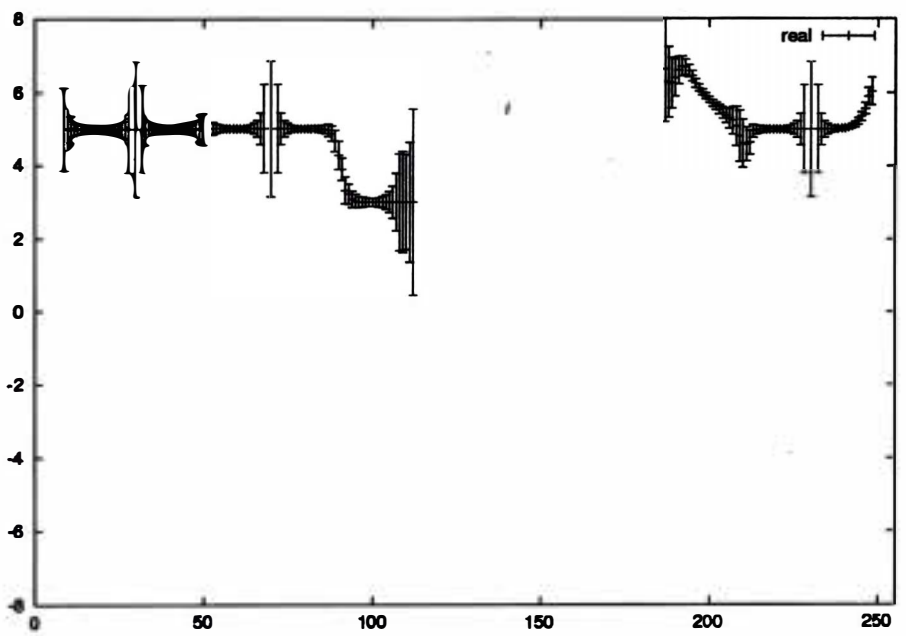


Figure 4.19: Test image 4, output stage 9. Filter wavelength $\lambda = 10$.

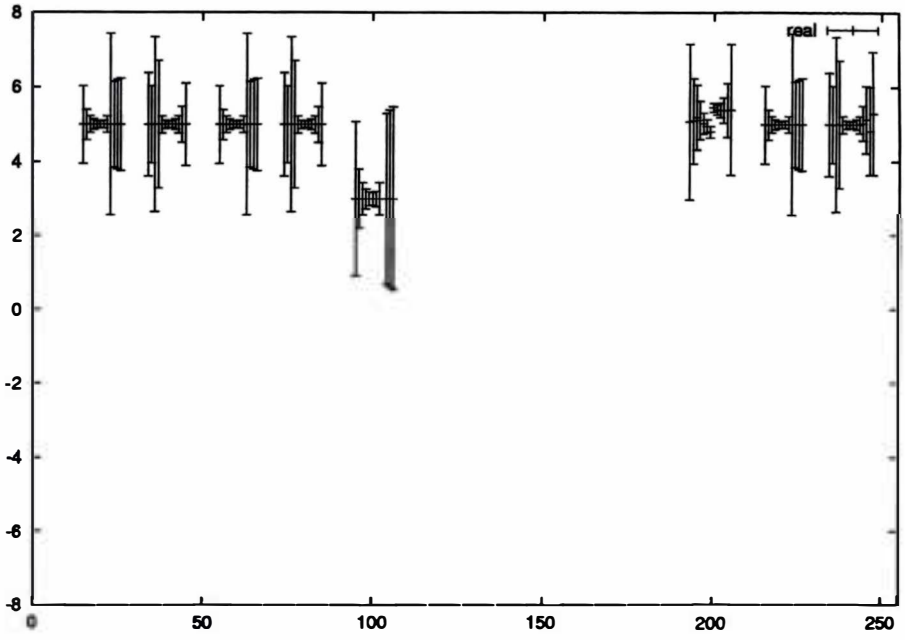


Figure 4.20: Test image 4, output stage 11. Filter wavelength $\lambda = 5$.

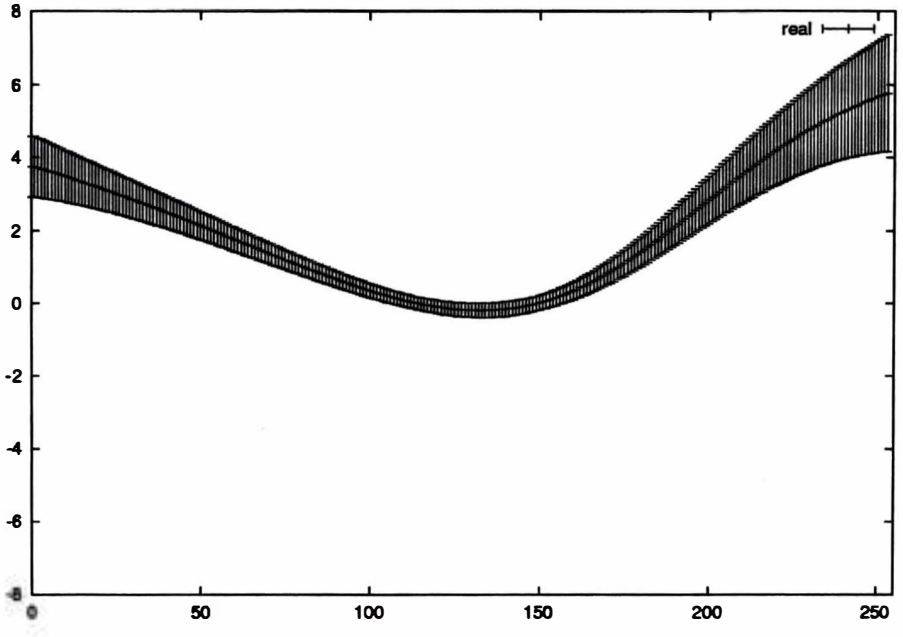


Figure 4.21: Test image 5, output stage 1. Filter wavelength $\lambda = 160$.

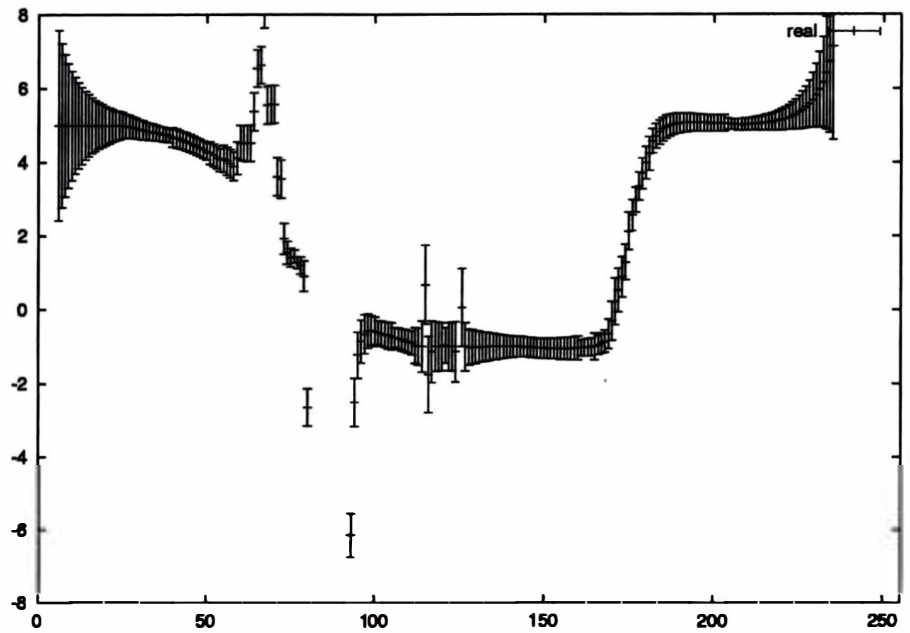


Figure 4.22: Test image 5, output stage 5. Filter wavelength $\lambda = 40$.

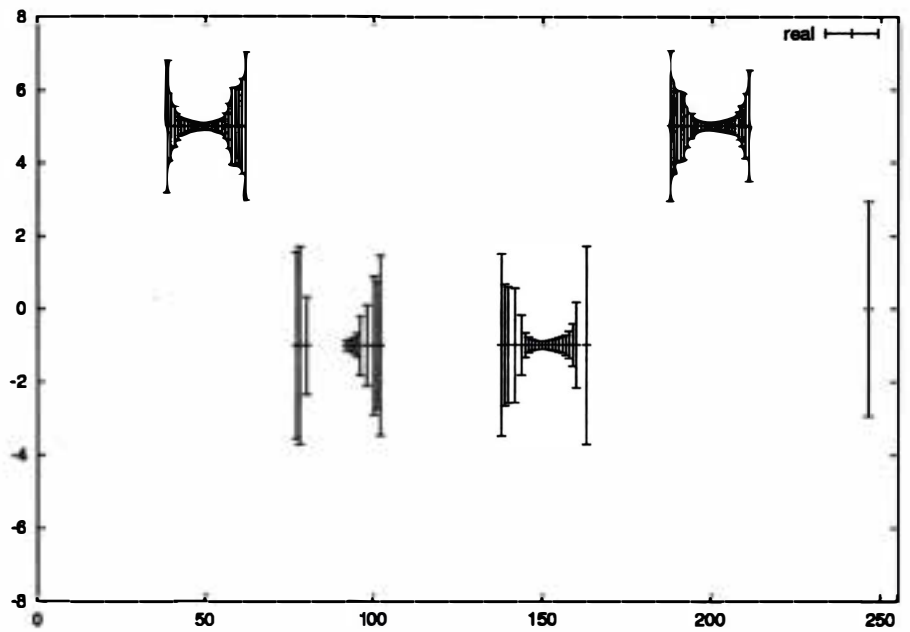


Figure 4.23: Test image 5, output stage 9. Filter wavelength $\lambda = 10$.

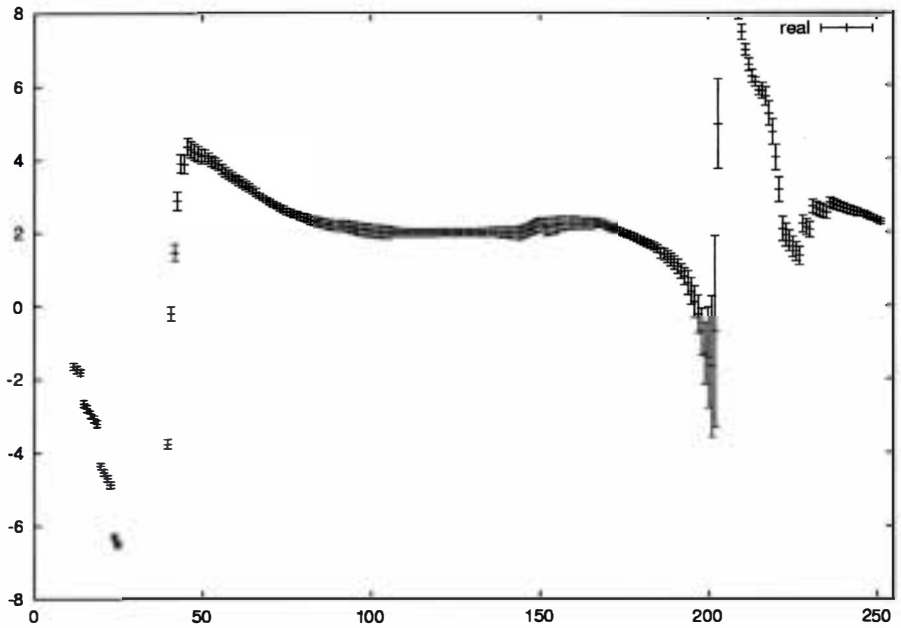


Figure 4.24: Test image 6, output stage 5. Filter wavelength $\lambda = 40$.

across the image, and although there are many gaps in the information due to uncertainty, the estimates that are available would certainly be sufficient to segment the scene into two or three well-defined regions.

For this test image, we again show (Figure 4.27) the output that results when we attempt to derive a flow field using only one of the finer filters. In this case no useful velocity estimates are produced at all. This is understandable — this test image was a randomly generated pattern. Without the benefit of *a priori* velocity estimates, the stage 12 filter can only attempt a velocity estimate by comparing points in the two frames that have the same coordinate. The actual disparity between the two frames is sufficient so that, over the window represented by the width of the filter, the pattern of intensity values in frame 1 is uncorrelated with that in frame 2.

There is an interesting effect, discernible in several of the output images but particularly noticeable in Figure 4.14, that should be commented on. Examination of this figure shows that there are a number of discontinuous jumps in the widths of the error bands. The initial suspicion was that this resulted from an error in the implementation of the algorithm at the programming level. Investigation revealed, however, that these jumps occur at places in the image where the rounded off value of the velocity estimate from the previous

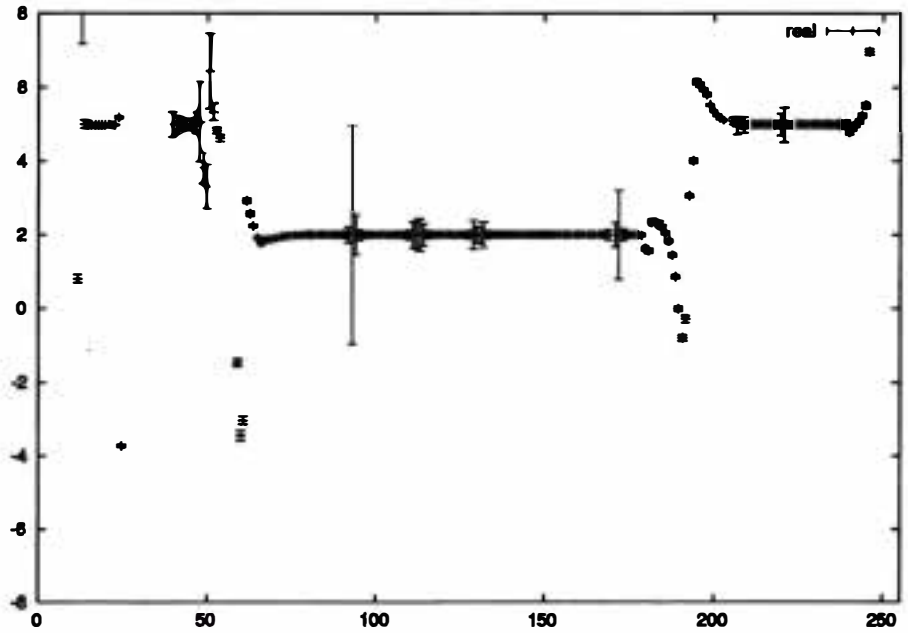


Figure 4.25: Test image 6, output stage 9. Filter wavelength $\lambda = 10$.

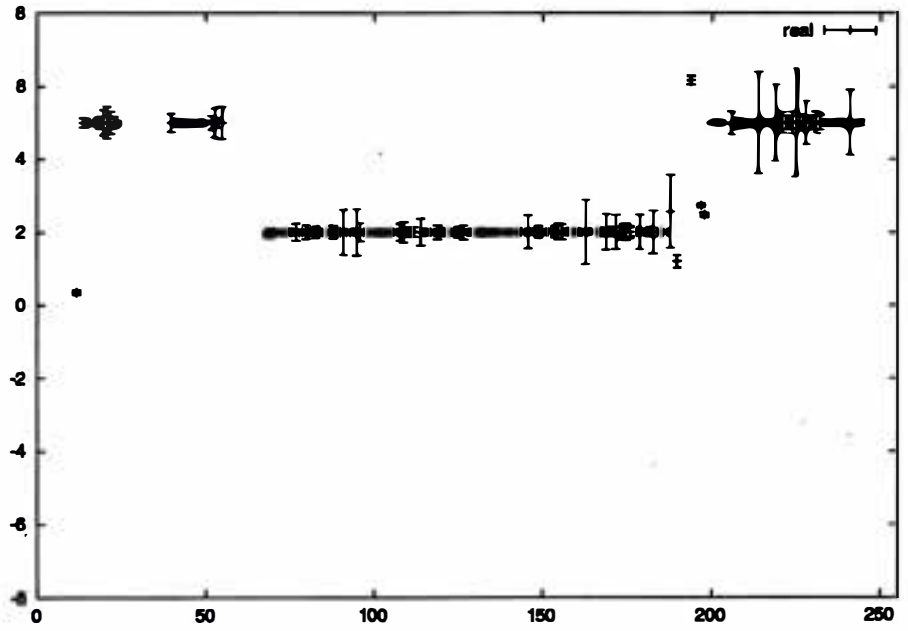


Figure 4.26: Test image 6, output stage 12. Filter wavelength $\lambda = 3.5$.

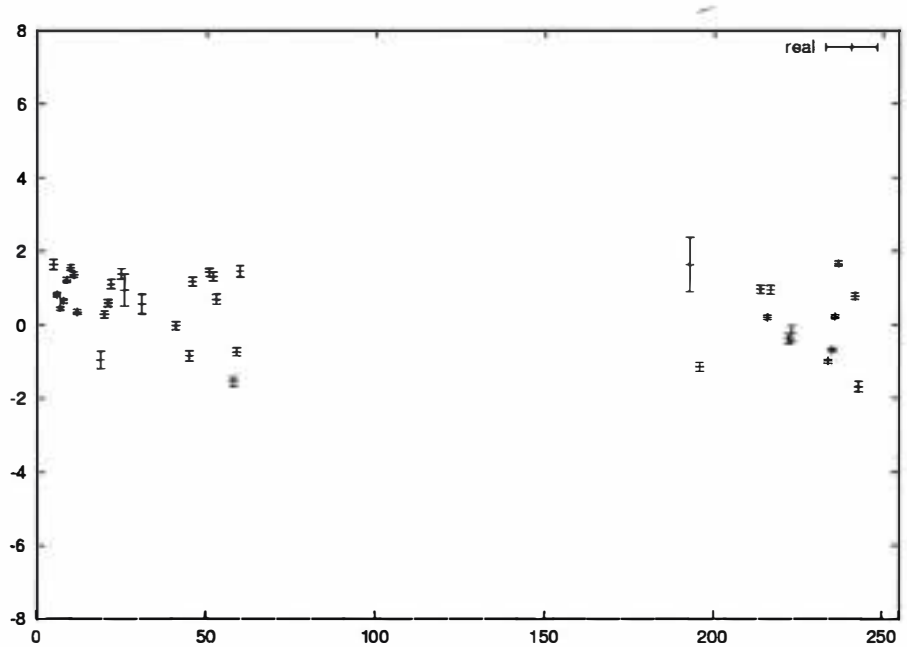


Figure 4.27: Test image 6. Output from stage 12 filter used in isolation — Filter wavelength $\lambda = 3.5$. No useful velocities are obtained.

stage switches from one value to another. Recall that this value is used to determine the offset to be used when comparing points in different frames (Figure 4.2). A one-dimensional velocity estimate is a real number, which may have a fractional component, but of course the image coordinates are integral, so the velocity must be rounded to an integer when selecting the offset.

For example, suppose that at point x in the image the *a priori* velocity estimate was +4.499, while at the neighbouring point $x + 1$ it was +4.501. These values are rounded to 4 and 5 respectively when selecting offsets. This means that the new velocity estimate for point x is derived by comparing image neighbourhoods in the two frames that are offset from one another by 4 units, but for point $x + 1$ the separation is 5 units. This can result in a discontinuous change in the values that are input to the Equations 4.28 and 4.29 to determine velocity, and can hence cause the uncertainty to behave in a discontinuous fashion between neighbouring points. The fact that the velocity estimate itself appears to behave in a continuous fashion at these image points encourages confidence in the strategy of deriving velocity at each stage using the estimate from the preceding stage.

4.4 Interpretation of the Results

The interpretation of the outputs requires some care. Traditionally the performance assessment of an optical flow method has been done via measurement of the disparity between the derived motion and the "ground-truth" motion at every point in the image (eg. Barron, Fleet, Beauchemin, and Burkitt, 1992; Barron, Fleet, and Beauchemin, 1994; McCane, Novins, Crannitch, and Galvin, 2001). This is not an appropriate way of assessing the present work. The output of the multi-scale analysis presented here is not a single velocity at each image point, but rather a **sequence** of velocities obtained at different scales, each with its corresponding measure of uncertainty. For those regions where information is available at the fine scale, these velocities converge to a well-defined value. For other points, where fine-scale information is not available, the velocities converge down to a certain scale but then fail to converge as one proceeds to the finer scales, and the uncertainty becomes very large. This reflects the fact that there are some points in the image where it is not possible to have a meaningful definition of velocity at the finer scale, because the information is simply not available.

Moreover, in chapter 3 the point is made that motion in the real world is in fact a scale-dependent quantity; at any point there is no single "ground-truth" motion with which to compare the estimate.

Devising a suitable numerical quantity that expresses the performance of a multi-scale analysis method remains a suitable topic for future research. In the meantime, a visual inspection of the test results presented should be sufficient to assess the performance.

Examining the test results, then, it appears that our method of determining 1-D optical flow does indeed offer the advantages claimed for the phase-based multi-scale approach, viz:

- Being based on phase, it appears to be highly responsive to significant features in the image intensity profile, while at the same time being apparently unaffected by smooth changes caused by changing scene illumination. This is demonstrated for example by the results for test images 1 and 2.
- Being a multi-scale method, it makes no prior assumptions about the scale of the features being used to detect motion, and therefore is able to

use information at whatever scale the information exists in the images. It deals reasonably successfully with features ranging in size from 1 or 2 pixels (test image 3) up to dimensions of the order of the image width (test images 1, 2 and 4).

The technique appears capable of calculating velocities where meaningful velocities exist, and of identifying the regions associated with those velocities. Confusion can occur in certain situations, such as where occlusion causes a feature to appear in one frame and not the other (eg. test image 4). In the concluding chapter we discuss how this problem might be overcome.

The results with the random dot stereogram (test image 6) are particularly encouraging, as they indicate that motion can be calculated in situations where motion is the **only** information available.

It was mentioned earlier that optical flow output could be used as a basis for image segmentation, taking the interpretation of an image to the next higher level. This is demonstrated for example by the outputs at later stages for image 4 (Figures 4.18 to 4.20 and 4.23). In the first two of these figures, several regions are identified with a velocity of +5, and one might reasonably suppose they correspond to parts of the same object in the real world. In the case of this particular image, the supposition would be supported by the fact that these regions also possess roughly the same grey-level intensity. In carrying out a segmentation we would therefore mark these regions as part of the same compound feature. Another region, with velocity = +3, has been identified, and a segmentation process based on flow would therefore flag this as being part of a separate feature.

In the case of image 6, Figures 4.25 and 4.26 demonstrate the possibility of separating the central region from the remainder of the image, based solely on the optical flow, with no additional features to assist.

4.5 Summary

The restriction of the methodology to images of only one dimension, while offering substantial simplifications and savings in computation time, retains the two most important features of the method, viz. the multi-scale technique

and the choice of phase as the parameter to which differential analysis is applied.

The results indicate success in localising a moving feature and assigning the velocity value. At the same time, the method assigns a high uncertainty to the velocity at places in the image where little information is available at that scale. This is in keeping with the philosophy expressed in chapter 3, and contrasts with the traditional approach of assigning a velocity at every point in the image by some method of propagation. The results for the random dot stereogram, in particular, suggest the probability that image segmentation could be achieved where motion was the only available guide, there being no discernible static features.

The success in the one-dimensional case gave encouragement to carry the work to two dimensions.

Chapter 5

Optical Flow in Two-Dimensional Images

Most images that we deal with in the real world are 2-dimensional. Moreover, although there are some special cases (such as images from aerial surveys) where the disparity between adjacent images is known to lie along one line (eg. the direction of motion of the aircraft), in general we would require that the method be able to extract a full 2-dimensional velocity field where the displacements may be in any direction in the image plane.

The steps required to generalise the method to two dimensions are described in this chapter. Extra complications arise, such as the Aperture Problem, and while the philosophy of the method remains unchanged, the details of its application need to be modified to handle these complications.

Some synthetic test images were used, as was done for the one-dimensional case; however a portion of an image sequence taken from a real-world scene is also used as a test case.

5.1 Theory

5.1.1 Issues in Generalising from the 1-D Case

The methodology of extracting optical flow in 1-dimensional images can easily be generalised to two dimensions. The concept is the same — a series of motion estimates at different spatial scales is obtained. At each scale the estimate obtained from the preceding (larger) scale is used as the initial estimate, and the estimate is then fine-tuned by comparing the image parameters at points in the two frames that are offset from one another, the offset being based on the previous flow estimate. In the two-dimensional case the offset has both an x and a y component.

There are two extra complications that arise in moving from one to two dimensions:

The Aperture Problem. As pointed out in section 2.1.2, a single spatial gradient and a single temporal rate of change are insufficient to determine a 2-dimensional velocity vector. They can only constrain the velocity to lie on a particular line. Another way of stating this is to say that there are two unknown quantities, v_x and v_y, so that at least two equations are needed to solve for the two unknowns. Further measurements of some kind are necessary in order to derive a velocity estimate.

Uncertainty in Two Dimensions. In the one-dimensional case, the flow estimate was expressed as a scalar velocity coupled with a scalar variance σ^2.

In expressing error or uncertainty of a 2-dimensional velocity, it is necessary not only to deal with the variance of each component σ_x^2 and σ_y^2, but also the cross-correlation σ_{xy} between the two components v_x and v_y. In general the uncertainty of a two-dimensional quantity is expressed by a 2 x 2 matrix Λ, where

$$\Lambda = \begin{pmatrix} \sigma_x^2 & \sigma_{xy} \\ \sigma_{xy} & \sigma_y^2 \end{pmatrix} \tag{5.1}$$

Although the matrix has 4 elements, it is diagonally symmetric, so that it contains only 3 independent scalar quantities.

There is also the question of what convolution operator(s) should be applied to a 2-dimensional image prior to performing a phase-based analysis. It is not sufficient to simply use a single 2-dimensional analogue of the Gabor filter to produce a single complex convolution output. This is because any 2-dimensional complex image will possess contours of constant phase, and a phase-based method will be unable to detect motion along a phase contour.

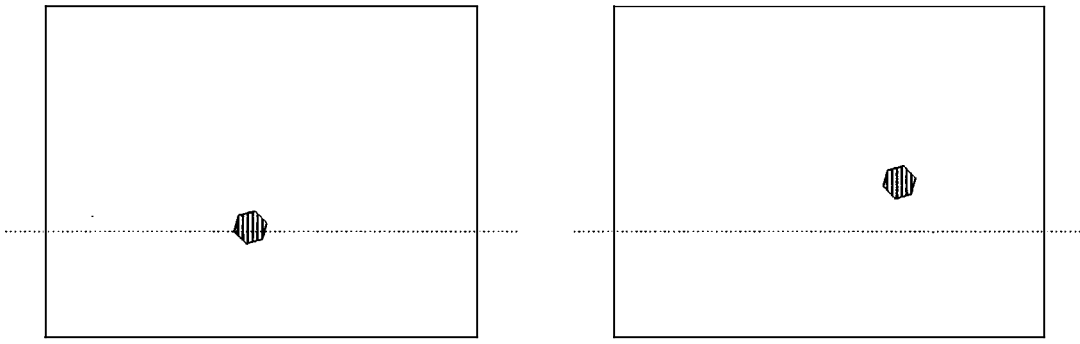
In order to calculate flow in an image of more than one dimension, we need 2 or more filters that give phase contours with different orientations. We choose to use two filters oriented at right angles to one another. This leads to two gradient constraint equations, which are combined to give an estimate of the 2-D flow as well as a corresponding uncertainty matrix.

5.1.2 Convolution Operators for 2-D Flow Estimation

A first attempt to derive flow in a 2-D image might proceed as follows, using 1-dimensional complex kernels:

- Select a 1-dimensional kernel, such as the 1-D Gabor filters employed for the 1-D image analysis.
- Generate a "horizontal convolution" by convolving each horizontal line of the image with this kernel.
- Similarly, generate a "vertical convolution" by convolving each vertical line of the image.
- Generate two gradient constraint equations at each point, one from the horizontal convolution output and one from the vertical convolution.
- Combine these two equations at each point to solve for the velocity.

Unfortunately such a methodology would not work. To understand why, take the example of a small object in the image, say a feature 5 pixels wide, and suppose that between frames it moves 20 pixels in the x direction and 10 pixels in the y direction. There is then no horizontal line that intersects the object in both frames, so it is not possible to obtain any meaningful velocity estimate from the horizontal convolution (Figure 5.1). The same is true of the vertical convolution.



(a) Frame 1: Object lies across the horizontal line

(b) Frame 2: Object no longer intersects the line

Figure 5.1: The horizontal line intersects the object in frame 1 but not in frame 2, so a 1-D convolution along this line gives no information about the movement.

The problem arises because the y component of the movement is greater than the size of the object, carrying it out of the "field of view" of this particular horizontal line. While such a method might work on a certain subclass of images, viz. those in which the displacement of a feature is small compared with the size of the feature, in general we cannot assume this restriction will hold.

The solution we adopt is to convolve the image with two different 2-dimensional kernels. One kernel is the result of the composition of a vertically oriented Gabor filter with a horizontally oriented Gaussian kernel. The other is the composition of a horizontally oriented Gabor filter with a vertically oriented Gaussian. By "vertically oriented" we mean that the filter function is a function only of x and is independent of y, so that phase contours are vertical (as are amplitude contours). Similarly a "horizontally oriented" kernel has a filter function that depends only on y.

It is not clear what the relation should be between the parameters of the two filters – one Gabor, the other Gaussian – that are applied at each step. For this work, the two filters were set to have the same size envelope, i.e. the same σ value.

In mathematical notation, then, at a given stage of the analysis, let σ and k denote the width and the spatial angular frequency of the filters to be used. Then the vertically and horizontally oriented 2-D Gaussian kernels are

defined by:

$$G_x(x, y; \sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{(-x^2/2\sigma^2)} \tag{5.2}$$

and

$$G_y(x, y; \sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{(-y^2/2\sigma^2)} \tag{5.3}$$

while the horizontally and vertically oriented 2-D Gabor kernels are given by:

$$\mathtt{Gabor_x}(x, y; \sigma, k) = \frac{1}{\sqrt{2\pi}\sigma} e^{(-x^2/2\sigma^2)} e^{ixk} \tag{5.4}$$

and

$$\mathtt{Gabor_y}(x, y; \sigma, k) = \frac{1}{\sqrt{2\pi}\sigma} e^{(-y^2/2\sigma^2)} e^{iyk} \tag{5.5}$$

Then the two kernels that are applied to the image at each stage are the results of composition:

$$(G_x) \circ (\mathtt{Gabor_y}) \tag{5.6}$$

and

$$(G_y) \circ (\mathtt{Gabor_x}) \tag{5.7}$$

Each of these kernels has the effect of "blurring" the image in both the x and y directions, so that even a small feature can be "seen" in both frames provided the kernel width is large enough to capture the motion of the feature.

For each point in the image, one velocity contraint is obtained from each of the two convolutions. The two constraints are then combined to give a single velocity estimate and uncertainty matrix, as described in section 5.1.5 and illustrated in figure 5.3.

The uncertainty in the flow estimates v_x and v_y is carried out in a manner analogous to the one-dimensional case as described in section 4.1.5, with the added complication that there are two estimates, derived from two different convolution processes, that are then combined using a Kalman Filter technique.

5.1.3 Flow Constraint Line for One Convolution Sequence

In the previous section we described how two convolved image sequences are produced — one by convolving the image frames with the composition of a horizontally oriented Gabor filter with a vertically oriented Gaussian filter, the other by convolution with the composition of a vertically oriented Gabor with a horizontally oriented Gaussian.

In this section we show how we derive the gradient constraint equation for either one of these resultant complex output image sequences.

In the absence of image noise, an optical flow estimate at a single point would consist of a single velocity constraint line. Figure 5.2 illustrates how the velocity constraint line can be determined if the function behaviour (in our case the phase behaviour) is precisely known. The temporal derivative of the function can be combined with the spatial derivatives along the x and y axes using the 2-dimensional gradient constraint equation:

$$\frac{\partial \phi}{\partial t} + v_x \frac{\partial \phi}{\partial x} + v_y \frac{\partial \phi}{\partial y} = 0 \tag{5.8}$$

In Figure 5.2, the velocity constraint line is defined by the points **A** and **B** at which it crosses the x and y axes. These points in turn are defined by the x and y spatial derivatives of the phase. **OA** and **OB** represent the velocities of the points of intersection of a phase contour with the x and y axes respectively.

We define w_x as the phase velocity measured along the x axis, as if that line were a single 1-dimensional image. Similarly we define w_y as the phase velocity measured along the y axis. These velocities should not be confused with the x and y components of the 2-D velocity, which are the projections of the vector **OP** on the x and y axes.

Then the velocity vectors $(w_x, 0)$ and $(0, w_y)$ both lie on the flow constraint line. Referring to Figure 5.2, w_x and w_y are the lengths **OA** and **OB** respectively. They define the flow constraint line. Moreover, w_x and w_y can be

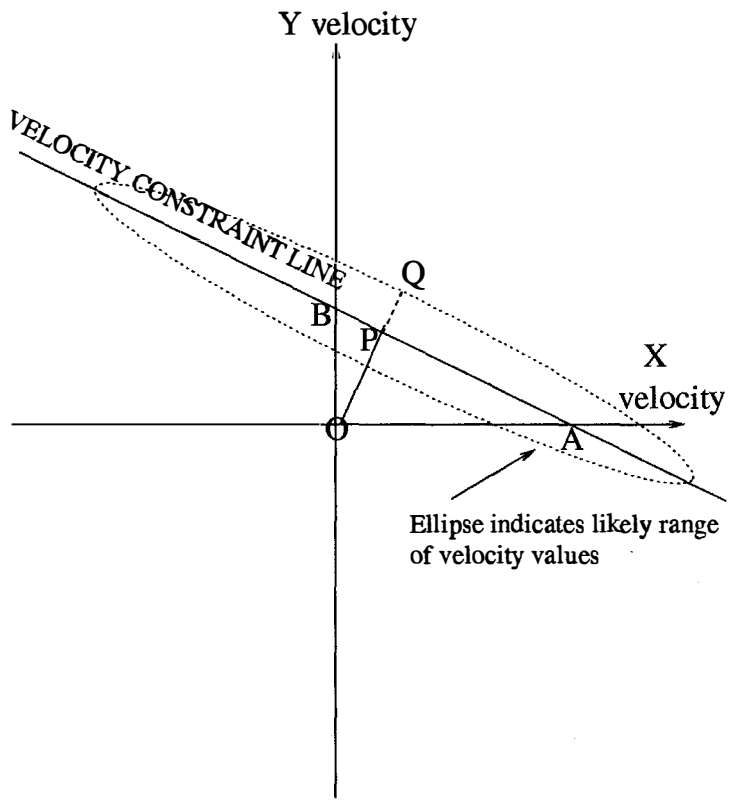


Figure 5.2: The result in velocity space of one application of the gradient constraint equation to a 2-D image.

calculated using two one-dimensional constraint equations:

$$\frac{\partial \phi_x}{\partial t} + w_x \frac{\partial \phi_x}{\partial x} = 0 \tag{5.9}$$

$$\frac{\partial \phi_y}{\partial t} + w_y \frac{\partial \phi_y}{\partial y} = 0 \tag{5.10}$$

To simplify notation, we define the following terms:

$$\phi_{x,t} = \partial \phi_x / \partial t \tag{5.11}$$

$$\phi_{x,x} = \partial \phi_x / \partial x \tag{5.12}$$

$$\phi_{y,t} = \partial \phi_y / \partial t \tag{5.13}$$

$$\phi_{y,y} = \partial \phi_y / \partial y \tag{5.14}$$

We also extend the ψ notation introduced in section 4.1.3 as follows: if p is allowed to be either of the spatial variables x and y, and q can be x, y or t, then

$$\psi_{p,q} = R_p \frac{\partial I_p}{\partial q} - I_p \frac{\partial R_p}{\partial q} \tag{5.15}$$

For example,

$$\psi_{x,t} = R_x \frac{\partial I_x}{\partial t} - I_x \frac{\partial R_x}{\partial t} \tag{5.16}$$

where R_x and I_x are the real and imaginary parts of the x convolution and R_y and I_y are the real and imaginary parts of the y convolution.

By analogy with equation 4.21, we can write expressions for w_x and w_y using the ψ notation:

$$w_x = -\frac{\psi_{x,t}}{\psi_{x,x}} \tag{5.17}$$

$$w_y = -\frac{\psi_{y,t}}{\psi_{y,y}} \tag{5.18}$$

The flow constraint line is now defined by w_x and w_y, and its equation can be written:

$$\frac{v_x}{w_x} + \frac{v_y}{w_y} = 1 \tag{5.19}$$

The perpendicular line OP has slope w_x/w_y and passes through the origin, so that its equation is

$$\frac{v_y}{v_x} = \frac{w_x}{w_y} \tag{5.20}$$

To find the point of intersection P, we solve these last two equations simultaneously, giving expressions for v_x and v_y at P:

$$v_x = \frac{w_x w_y^2}{w_x^2 + w_y^2} \tag{5.21}$$

$$v_y = \frac{w_x^2 w_y}{w_x^2 + w_y^2} \tag{5.22}$$

Substituting the expressions for w_x and w_y from equations 5.17 and 5.18 gives v_x and v_y in terms of ψ values:

$$\begin{aligned} v_x &= -(\frac{\psi_{x,t}}{\psi_{x,x}} \frac{\psi_{y,t}^2}{\psi_{y,y}^2})/(\frac{\psi_{x,t}^2}{\psi_{x,x}^2} + \frac{\psi_{y,t}^2}{\psi_{y,y}^2}) \\ &= -(\psi_{x,t}\psi_{x,x}\psi_{y,t}^2)/(\psi_{x,t}^2\psi_{y,y}^2 + \psi_{y,t}^2\psi_{x,x}^2) \end{aligned} \tag{5.23}$$

$$\begin{aligned} v_y &= -(\frac{\psi_{y,t}}{\psi_{y,y}} \frac{\psi_{x,t}^2}{\psi_{x,x}^2})/(\frac{\psi_{x,t}^2}{\psi_{x,x}^2} + \frac{\psi_{y,t}^2}{\psi_{y,y}^2}) \\ &= -(\psi_{y,t}\psi_{y,y}\psi_{x,t}^2)/(\psi_{x,t}^2\psi_{y,y}^2 + \psi_{y,t}^2\psi_{x,x}^2) \end{aligned} \tag{5.24}$$

These equations should be compared wth equation 4.21 which is the one-dimensional equivalent. Note that if $\psi_{y,y}$ is zero, equation 5.23 reduces to

$$v_x = -\frac{\psi_{x,t}}{\psi_{x,x}} \tag{5.25}$$

which is of the same form as equation 4.21. This is to be expected, since in the absence of any spatial variation in the y direction, the calculation of v_x reduces to a one-dimensional problem.

Note that in the two-dimensional case the denominators contain two non-negative terms, both of which must approach zero in order for the denominator to be small. This means that the requirements for flow non-singularity are less stringent than in the one-dimensional case; for example, it is possible for $\psi_{x,x}$ to be zero (which would give an infinite value for w_x) but for both v_x and v_y to be well-defined. However it is still necessary to test the individual behaviour of the terms $\psi_{x,t}/\psi_{x,x}$ and $\psi_{y,t}/\psi_{y,y}$.

Considering first the expression for v_x, four cases arise:

- Both w_x and w_y are well-behaved.
  v_x can be calculated directly from equation 5.23.
- w_x is well-behaved but w_y exhibits a singularity because $\psi_{y,y}$ is too small.
  Equation 5.23 simplifies to 5.25, so that v_x is simply equal to w_x.
- w_y is well-behaved, but w_x has a singularity because $\psi_{x,x}$ is too small.
  Here v_x should be set to zero, as can be seen from the second form of equation 5.23.
- Both w_x and w_y show singularities.
  In this case v_x cannot be determined and the uncertainty is infinite.

Similar conditions apply to the determination of v_y.

5.1.4 Calculation of the Flow Covariance Matrix

In a measurement situation, the function and its spatial derivatives are subject to uncertainty, and we need to determine not only a constraint line for the velocity vector $\tilde{v} = (v_x, v_y)$ but a covariance matrix for $\tilde{v}$. Referring to Figure 5.2, uncertainties in the positions of **A** and **B** affect not only the position but the orientation of the velocity constraint line, giving a 2-D probability distribution for $\tilde{v}$. A single point measurement results in a covariance matrix that is singular. The large eigenvalue of the matrix is infinite and the

measured velocity may lie anywhere along the line. When two measurements from two different convolution outputs at the point are combined, however, the resultant covariance matrix may be nonsingular and the velocity estimate become localised around a point. The actual behaviour in this regard depends on how the phase behaviour varies between the two convolution images, which governs the orientation of the two velocity constraint lines relative to one another. Two parallel constraint lines give no more information than a single constraint line.

A model for this probability distribution can be obtained as follows:

1. Calculate the velocity that lies on the velocity constraint line and is perpendicular to it. This is the vector **OP** in Figure 5.2, **P** being the closest point on the constraint line to the origin in velocity space. It therefore represents the smallest possible magnitude for a velocity that satisfies the constraint equation. Being a solution of the equation, it is as good a choice as any other at this stage for a velocity estimate. Denote this vector $\tilde{v_\perp}$. The point **P** is determined by the axial velocities **OA** and **OB**, since these define the constraint line.

2. Calculate the associated uncertainty in $\tilde{v_\perp}$, using the uncertainties of **OA** and **OB**.

3. Adopt a rotated coordinate system (x', y') where x' and y' are parallel and perpendicular respectively to $\tilde{v_\perp}$.

4. Calculate the component $v_{x'}$ of $\tilde{v_\perp}$ in the x' direction. This is simply the magnitude of $\tilde{v_\perp}$, represented by the length **OP** in Figure 5.2.

5. Calculate the associated uncertainty $\sigma_{v'_x}$ of v'_x. This is length **PQ** in the figure.

6. Model the distribution of (v'_x, v'_y) by a Gaussian distribution with variance σ'_x in the x' direction and a very large (effectively infinite) spread σ_{max} in the y' direction.

7. Represent this by the uncertainty matrix Λ' in (x', y') coordinates:

$$\Lambda' = \begin{pmatrix} \sigma^2_{x'} & 0 \\ 0 & \sigma^2_{max} \end{pmatrix} \tag{5.26}$$

8. Operate on this matrix with a rotation matrix, giving the covariance matrix in Λ in the (x, y) coordinate system:

$$\Lambda = \begin{pmatrix} \sigma_x^2 & \sigma_{xy} \\ \sigma_{xy} & \sigma_y^2 \end{pmatrix}$$

Carrying out the above steps, then:

1. Recall that, in the one-dimensional case, v was calculated from the ratio of two phase derivatives, using equation 4.18. We were able to transform this expression into an expression in the real and imaginary parts of the convolution output (equations 4.25 and 4.26).

   In the two-dimensional case, v_x and v_y are calculated in an analogous manner, using equations 5.23 and 5.24 above.

2. In the 2-dimensional case, the input image is convolved with a Gabor filter then a Gaussian filter, as opposed to the one-dimensional case where only a Gabor filter is used. This means that, by analogy with equation 4.31, the intensity noise σ_C^2 at each point in the convolved image is given by

$$\sigma_C^2 = \sigma_I^2 / 16\pi\sigma^2 \tag{5.27}$$

   The division factor is the square of the division factor for the 1-D case, because each convolution introduces a factor of $1/4\sqrt{\pi}\sigma$.

   The uncertainties σ_x and σ_y of v_x and v_y respectively are then calculated in the same way as σ_v in the 1-dimensional case, using Equations 4.34 and 4.36.

3. Adopt the rotated coordinate system (x', y') as defined above, so that $\tilde{v_\perp}$ has component v_x' in the x' direction and zero in the y' direction.

4. Taking equations 5.23 and 5.24 and combining these, making use of Pythagoras' theorem, gives

$$v_{x'} = \sqrt{v_x^2 + v_y^2}$$

$$= \frac{w_x w_y}{\sqrt{w_x^2 + w_y^2}}$$

$$= \frac{\psi_{x,t}\psi_{y,t}}{\sqrt{\psi_{x,t}^2\psi_{y,y}^2 + \psi_{y,t}^2\psi_{x,x}^2}} \tag{5.28}$$

5. To calculate the uncertainty in $v_{x'}$, we denote the numerator in Equation 5.28 by N and the denominator by D, and use the shorthand notation $\sigma_{x,t}$, $\sigma_{x,x}$ etc. to denote the uncertainties in $\sigma_{x,t}$, $\psi_{x,x}$ etc.

   Using identities for the variances of sums and products of two quantities (equations 4.32 and 4.33), we derive the uncertainty in N and D:

$$\sigma_N = \psi_{x,t}\sigma_{y,t} + \psi_{y,t}\sigma_{x,t} \tag{5.29}$$

   and

$$\begin{aligned}\sigma_D = &\left(1/\sqrt{\psi_{x,t}^2\psi_{y,y}^2 + \psi_{y,t}^2\psi_{x,x}^2}\right) \\ &\times[\psi_{x,t}^2\psi_{y,y}\sigma_{y,y} + \psi_{y,y}^2\psi_{x,t}\sigma_{x,t} \\ &+\psi_{y,t}^2\psi_{x,x}\sigma_{x,x} + \psi_{x,x}^2\psi_{x,x}\sigma_{y,t}]\end{aligned} \tag{5.30}$$

   These expressions are evaluated by expanding the expressions for $\psi_{x,t}$, $\psi_{y,y}$ etc. in terms of real and imaginary parts of the outputs, in the same manner as described in section 4.1.5 for the 1-dimensional case. Then, using the identity for the standard deviation of the quotient of two quantities (equation 4.35), we find the uncertainty in $\sigma_{v'_x}$:

$$\sigma_{v'_x} = \frac{\sigma_N}{|D|} + \frac{|N|\sigma_D}{D^2} \tag{5.31}$$

6. All points on the velocity constraint line have the same value of v'_x but may take any value of v'_y. This means that, for a point satisfying the constraint equation, the uncertainty in the x' coordinate is $\sigma_{x'}$, while the uncertainty in the y' coordinate is ∞. In practice of course our equations cannot handle ∞, but it is sufficient to use some value σ_{max} large enough so that it will be interpreted as an unacceptable uncertainty. (In the present research the value used was the square of the image width, adopting the reasoning that an error in the flow estimate equal to the image width equates to complete lack of information about the flow.)

7. We then set up the uncertainty matrix Λ' for the velocity in the rotated (x', y') coordinates as per equation 5.26, repeated here:

$$\Lambda' = \begin{pmatrix} \sigma_{x'}^2 & 0 \\ 0 & \sigma_{max}^2 \end{pmatrix}$$

8. It now remains to derive the uncertainty matrix Λ expressed in the original coordinates (x, y):

$$\Lambda = \begin{pmatrix} \sigma_x^2 & \sigma_{xy} \\ \sigma_{xy} & \sigma_y^2 \end{pmatrix}$$

This is given by:

$$\Lambda = Q^T \Lambda Q \tag{5.32}$$

where Λ is the uncertainty matrix in (x, y) coordinates: and Q is the rotational transformation matrix:

$$Q = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}$$

θ being the angle **XOP** in Figure 5.2, so that:

$$\cos\theta = v_x / v_{x'}$$

and

$$\sin\theta = v_y / v_{x'}$$

5.1.5 Combining Two Velocity Estimates Using the Kalman Filter

In the previous section we show how the output from one convolution is used to derive a single velocity estimate and uncertainty covariance matrix. Because we are only able to say that the velocity lies somewhere along a constraint line, the large eigenvalue of this uncertainty matrix is effectively infinite.

We now describe how the output from two convolutions may be combined. Two velocity estimates, together with the two associated uncertainty matrices, are combined to produce a single "best" estimate and associated uncertainty matrix. In many cases the resultant uncertainty matrix has manageably finite eigenvalues, reflecting the fact that the simultaneous application

of two velocity constraint equations effectively confines the velocity estimate to a small area in velocity space.

The two estimates are combined using the Kalman Filter technique. This was introduced by R. E. Kalman in 1960, and is a method for combining a series of measurements of a physical system to provide a best estimate of the state of the system.

The technique is described in Du Plessis (1967). If we have a current estimate of the state of the system, and a covariance matrix that specifies the uncertainty in that estimate, the Kalman filter describes how to use an additional measurement (with its own uncertainty) to derive a new best estimate of the system state.

In a generalised physical measurement situation, a series of readings would be taken of the variables that constitute the state of the system. In the present case, we consider the system as being characterised by the two velocity components v_x and v_y, and we deal with only two "measurements", these being the estimates derived from the two convolution outputs.

Let the velocity meaurements be denoted by $\tilde{v_1}$ and $\tilde{v_2}$, and the corresponding covariance matrices by Λ_1 and Λ_2. We wish to derive a final estimate $\tilde{v}$ and its covariance matrix Λ.

Following Du Plessis (1967), we adopt the following notation:

Let

$\tilde{x}$ = the multi-dimensional quantity being measured or estimated

P = the covariance matrix for $\tilde{x}$

(Du Plessis also uses M to denote the "measurement matrix", which relates the components of the quantity being measured to the outputs from the measuring devices. In our case we are making direct estimates of $\tilde{x}$ rather than inferring $\tilde{x}$ from the readout of a scientific instrument, so we can consider M to be the identity matrix and omit it from the analysis.)

Suppose that we have a prior estimate $\tilde{x_1}$ of $\tilde{x}$, with an assigned uncertainty covariance matrix of P_1, and that a new measurement gives an estimate of $\tilde{x_2}$ with uncertainty matrix of P_2. Then Du Plessis shows how to combine

the two estimates to derive the "maximum likelihood estimate":

1. Define a weighting matrix B:

$$B = P_1(P_1 + P_2)^{-1} \tag{5.33}$$

2. The weighting factor is applied to the difference between $\tilde{x_2}$ and $\tilde{x_1}$. This is then added as a correction to $\tilde{x_1}$ giving a new estimate $\tilde{x}$:

$$\tilde{x} = \tilde{x_1} + B(\tilde{x_2} - \tilde{x_1}) \tag{5.34}$$

3. The new covariance matrix P for $\tilde{x}$ is found by reducing P_1 using the same weighting matrix B:

$$P = (I - B)P_1 \tag{5.35}$$

   where I is the identity matrix.

Substituting for B in equation 5.34 gives

$$\begin{aligned} \tilde{x} &= \tilde{x_1} + [P_1(P_1 + P_2)^{-1}](\tilde{x_2} - \tilde{x_1}) \\ &= [P_2\tilde{x_1} + P_1\tilde{x_2}](P_1 + P_2)^{-1} \end{aligned} \tag{5.36}$$

while substituting for B in equation 5.35 gives

$$P = [I - P_1(P_1 + P_2)^{-1}]P_1 \tag{5.37}$$

$$= P_1P_2(P_1 + P_2)^{-1} \tag{5.38}$$

In our work, the quantity $\tilde{x}$ is the image velocity, which we have denoted $\tilde{v}$, and the covariance matrix P is the velocity uncertainty matrix Λ. Rewriting the above equations using this notation gives the expressions for the new velocity estimate and uncertainty matrix:

$$\tilde{v} = [\Lambda_2\tilde{v_1} + \Lambda_1\tilde{v_2}][\Lambda_1 + \Lambda_2]^{-1} \tag{5.39}$$

and

$$\Lambda = \Lambda_1\Lambda_2[\Lambda_1 + \Lambda_2]^{-1} \tag{5.40}$$

These expressions are symmetric; that is, they are unchanged when $\tilde{v_1}$ and Λ_1 are transposed with $\tilde{v_2}$ and Λ_2. This is not surprising, since we would expect

it to make no difference which of the two measurements is considered to be the first. (This symmetry would not necessarily apply in a scientific laboratory situation in which measurements of a dynamically changing system are being taken at different times.)

Figure 5.3 illustrates the process of deriving a 2-dimensional flow and error estimate at one scale.

5.1.6 Combining Measurements at Different Scales

The multi-scale analysis process is carried out in a manner exactly analogous to that for the 1-dimensional case as described in section 4.1.6. Figure 4.3 illustrates the process for both the 1-D and 2-D cases. The only technical difference is that, when a point in one frame is compared with a point in the second frame that is offset from the first, the offset being the displacement calculated from the previous stage, the offset now has an x and a y component (figure 5.4).

5.2 Test Methodology

5.2.1 Test Design

The criteria for designing test images are the same as for the 1-dimensional case, as given in section 4.2.1. They will not be repeated here.

Four synthetic image sequences, each consisting of two frames, were generated. In addition, to test the performance on a real-world scene, an image pair consisting or two frames from the well-known Hamburg Taxi Sequence was included.

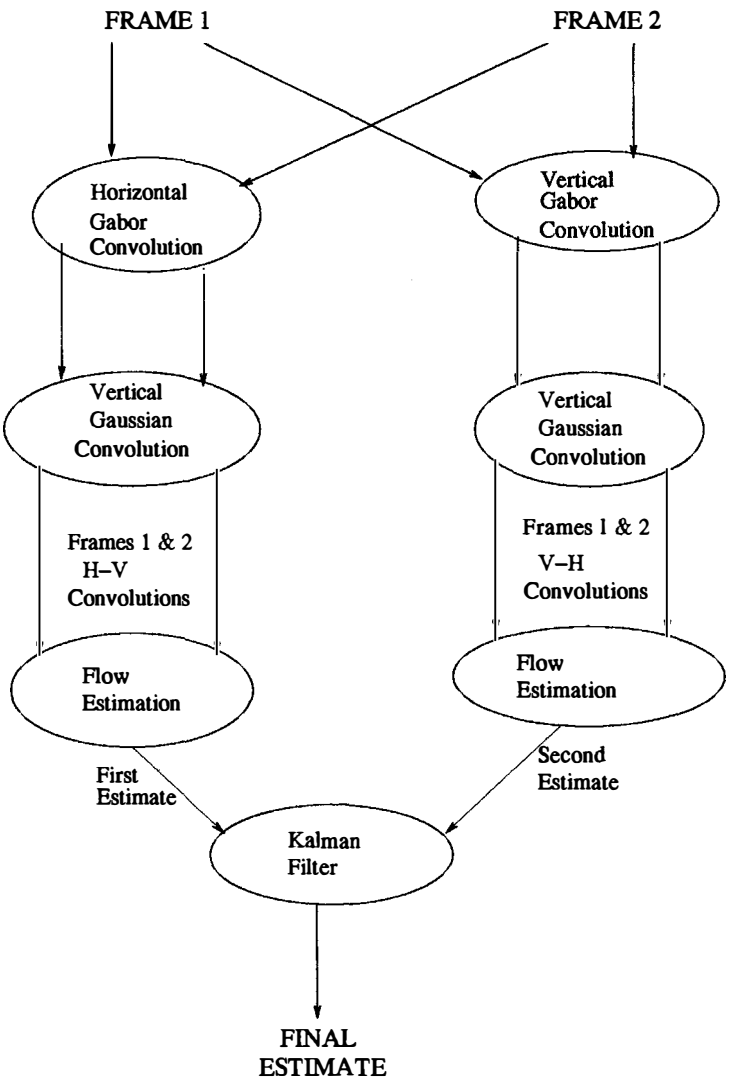


Figure 5.3: Data Flow Diagram, showing the steps in the process of deriving a 2-dimensional flow estimate using convolutions in both the horizontal and vertical directions.

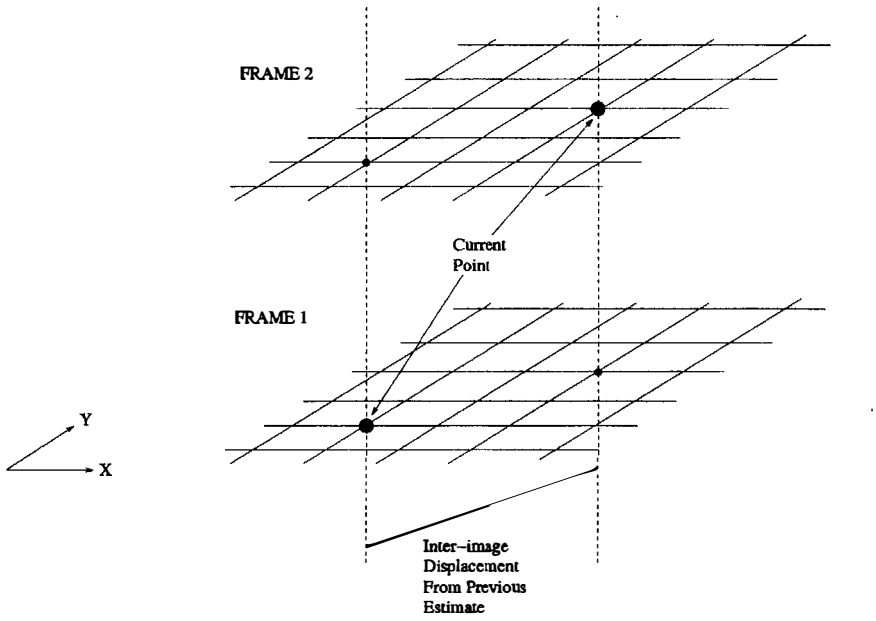


Figure 5.4: Points in frames 1 and 2 used to compute flow are offset from one another by the displacement vector got from the output at the preceding scale.

5.2.2 Test Images

All test images, with the exception of image 5, are squares of 256×256 pixels. Image 5 is 256×190 pixels.

The following test images were used:

- Test image 1 (Figure 5.5) shows a black circle on a white background. The circle is displaced upwards and to the right between frames 1 and 2.

  This figure was designed to be perhaps the simplest possible figure that would test the flow output for a situation where the motion of a feature is significantly greater than the size of the feature.

- Test Image 2 (Figure 5.6) is slightly more complex. Two circles, one black and the other white, are moving with different velocities across a background of intermediate grey intensity. This introduces the added requirement that the model must be able to separate out the velocity in one part of the image from the velocity in another region, so that segmentation may be carried out.

- Test Image 3 (Figure 5.7) shows a scene where an object is not only undergoing translation but is also changing its size. The object is growing and also moving towards the right, so that parts of the object at the right hand edge should exhibit a greater movement towards the right than the points at the left hand edge.

- Test image 4 (Figure 5.8) is a random dot stereo pair. The first frame was generated by using a random number generator. For the second frame, a square subset of the image was shifted 40 pixels to the right and 10 pixels upwards. (The disoccluded region was filled with random values.) The square has a width of 100 pixels and is centred at $(120, 120)$ in frame 1 and $(160, 130)$ in frame 2.

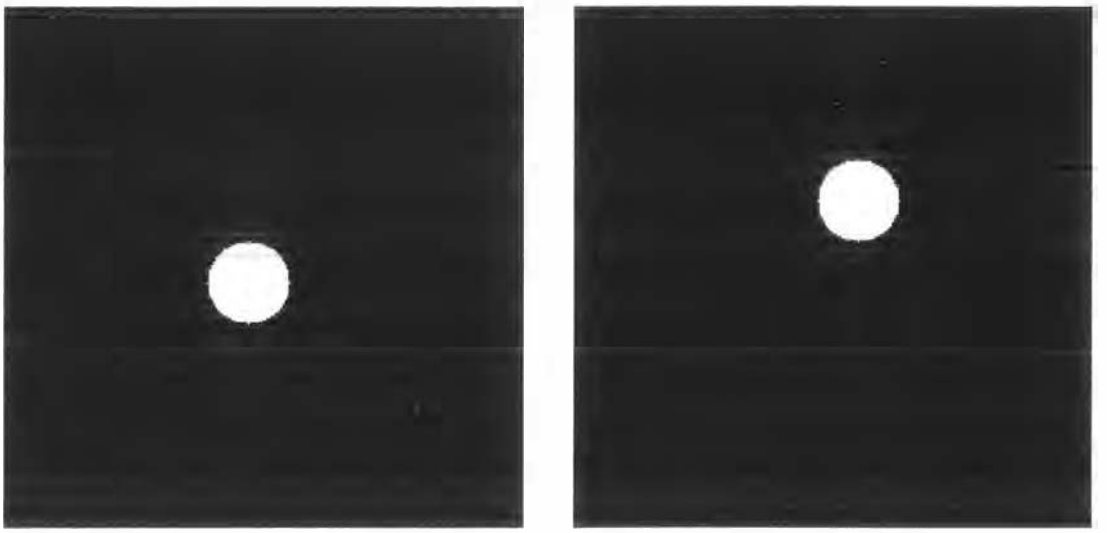
- Test image 5 (Figure 5.9) consists of two frames (frames 1 and 10) extracted from the well-known Hamburg Taxi Sequence. There are 4 moving objects discernible in this scene:

  - A white taxi near the centre of the image moving to the left and also moving with a lesser velocity in the y direction,
  - A dark vehicle at lower left, moving towards the right,
  - A dark vehicle at lower right, moving towards the left.
  - A walking person at top left, moving to the left and also slightly down (parallel to the kerb).

It would have been possible to add synthetic noise to one or more of the first three images, in order to test the sensitivity of the method to noise, which may have produced spurious features at certain scales. This was not done, however, since at the stage of design of the test images it was expected that at least some useful results would be obtained from the random dot image and the real-world image, and that this would give an indication of the way in which the method is affected by noise.

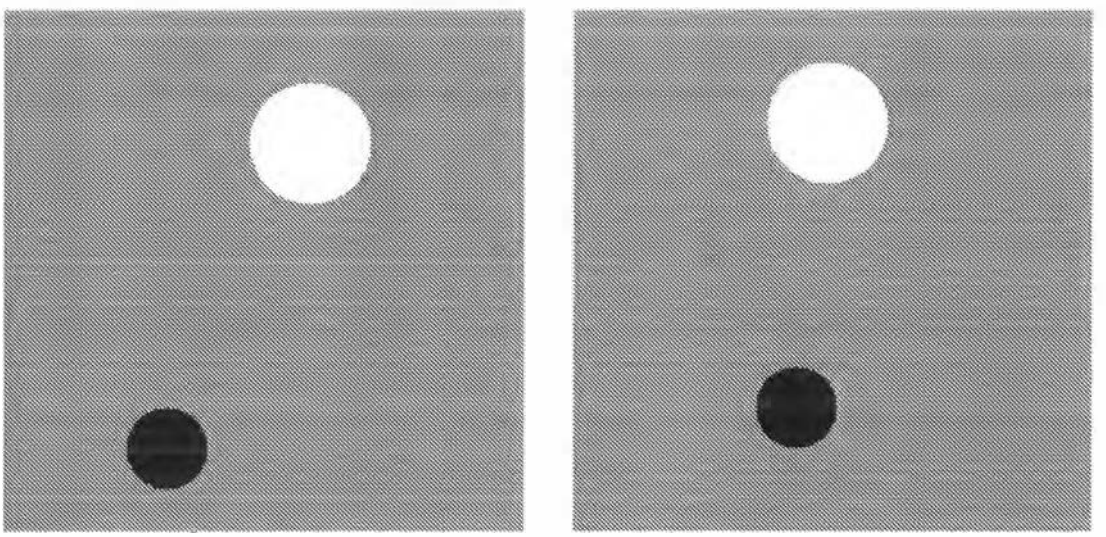
5.3 Results

The problem of displaying the results of a 2-D optical flow analysis visually in a form easily comprehensible is more difficult than for the 1-dimensional case. Velocity can be represented by a vector drawn in an image, but it is



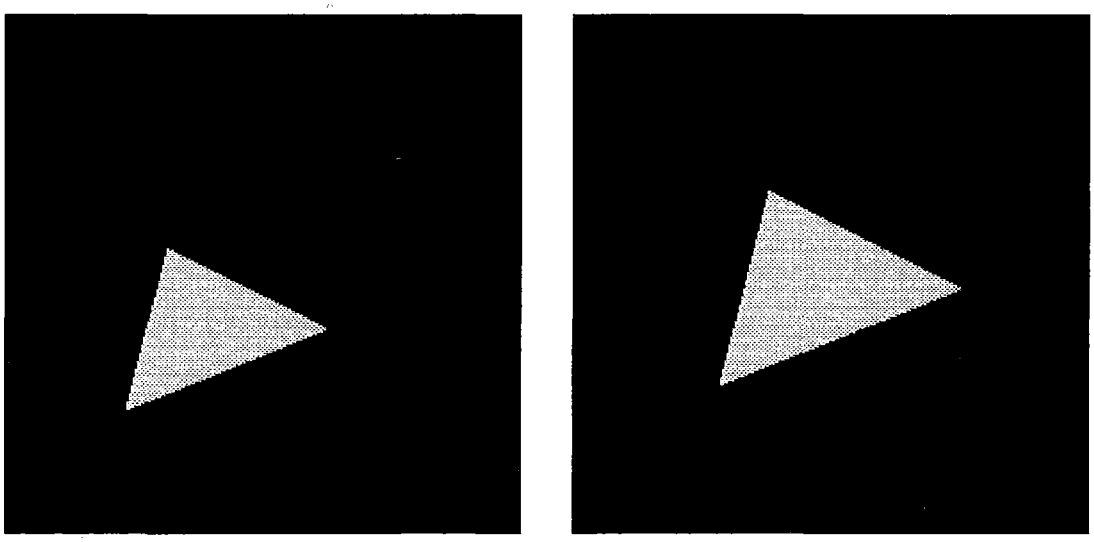
(a) Frame 1 (b) Frame 2

Figure 5.5: Test Image 1 – Single moving object. The displacement is large compared to the size of the object. Velocity $\vec{v} = (+20, +40)$.



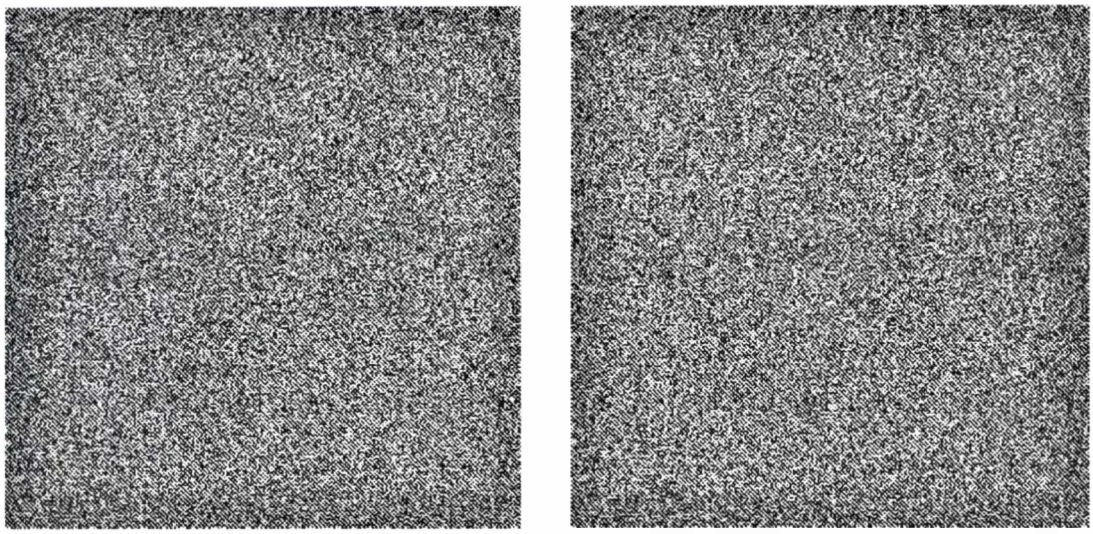
(a) Frame 1 (b) Frame 2

Figure 5.6: Test Image 2 – Two moving objects, White circle has velocity $(-25, +10)$, black circle has velocity (+30,+20).



(a) Frame 1 (b) Frame 2

Figure 5.7: Test Image 3 – The triangular object is moving but simultaneously undergoing scaling, increasing in size by 20% from one frame to the next. Apex at bottom left has velocity $(+12, +12)$, rightmost apex has velocity $(+32, +20)$.



(a) Frame 1 (b) Frame 2

Figure 5.8: Test Image 4 – Random dot stereogram. A square of width 100 pixels, centred at $(120, 120)$ in frame 1 has moved with velocity $(+40, +10)$ to be at $(160, 130)$ in frame 2.

(a) Frame 1 (b) Frame 2

Figure 5.9: Test Image 5 – Two frames from the Hamburg taxi Sequence. See the text for description of the objects and motion in the scene.

not possible to display the velocity at every image point in this way, since the vectors would overlie one another. The conventional approach is to display the output in the form of "needle diagrams", in which a vector is drawn at each one of an array of lattice points corresponding to points in the input image. The vector that is drawn is the velocity vector multiplied by a specified scaling factor. (In all the results presented here, the scaling factor is 1.0.)

Some authors (e.g. Cooper and Venkatesh, 1996) superimpose the needles onto the original image. This allows a reader to see at a glance the flow estimate at any point in the image and to associate it with an image feature. This works well where the image intensity is mostly dark or mostly light—the needles can be drawn with an intensity value that gives good contrast, e.g. white needles on a dark background. However, where there is a broad range of intensities over a region, the needles would not be easily visible using this scheme. Other researchers (e.g. Barnard and Thompson, 1980; Horn and Schunck, 1981) use needle diagrams separate from the original image — these are sometimes clearer, but need to be examined alongside the original image. In the present work, we superimpose the flow vectors onto Frame 1 of the original image. An exception is made for test figure 4. This figure consisted of a random dot pattern, and the flow vectors would have been difficult to distinguish visually from the background of the original image. For this reason the output from this figure is presented without being superimposed on the original image.

A needle image as described above does not give any indication of the measurement uncertainty estimate at each point. This problem can be addressed by selecting a threshold estimate σ_{MAX} for the error; if the threshold exceeds this value, no vector is drawn at that point. In the 2-dimensional case there are actually four error parameters involved (two of them equal); these are the elements of the 2 x 2 uncertainty matrix. We use the determinant of the matrix to decide whether to display the flow vector; if the determinant exceeds the fourth power of the selected error threshold, no vector is drawn. This cutoff value is used because it corresponds to what the determinant would be if the x and y estimates were uncorrelated and the standard deviation of each was equal to σ_{MAX}. The value of the determinant, $\sigma_x^2\sigma_y^2 - (\sigma_{xy})^2$, would then equate to $(\sigma_{MAX})^4$.

In the images presented here, the threshold error was set to 50 pixels per frame. Since the velocities in our sample images were not much less than this in some cases, such a threshold avoided the the early rejection of estimates

made with with the broadband filters. A threshold of 50 may seem excessive for estimates of velocity corrections at the finer scales; it was found, however, that adjusting this threshold either way appeared to have little effect on the output. It appears that the flow uncertainty at any point tends to be either very small or very large, so that the uncertainty threshold is not critical.

Bearing in mind the limitations of this output display method, a selection of results from the test images of the previous section is presented to demonstrate the performance of the method applied in two dimensions.

The first test image (Figure 5.5) showed a single object moving with a well defined velocity $v = (+20, +40)$.

The output from one of the larger filters (Figure 5.10(a)) is reminiscent of figure 4.10 for the 1-D case. It shows that a uniform rigid movement is detected very early in the analysis, and with an apparently accurate value for the flow; however the areas that gave rise to this flow estimate are not identifiable.

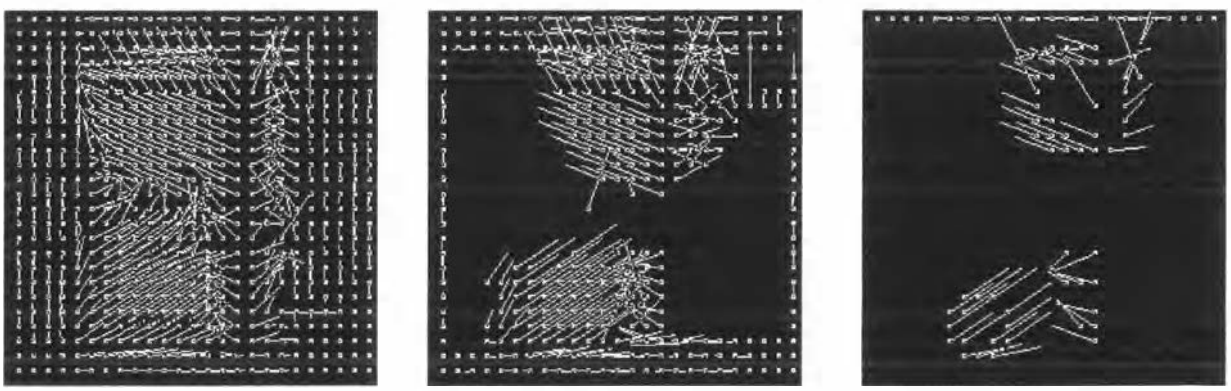
At a later stage (figure 5.10(b)), a small region has been identified as being associated with this flow; the rest of the image lacks flow information. In the next image the source of the information has been localized further still; it is confined to the periphery of the moving circular object, with no information found in the interior. Recall from section 2.1.2 and figure 2.2 that there is no motion information available in the interior of an object of uniform intensity – information is present only at the boundary discontinuity.

Test image 2 showed two objects moving with different velocities, instead of only one object. The output images (Figure 5.11), showing the results at successively finer scales, again demonstrate that the method succeeds in identifying two regions that are associated with two different flow vectors. They also show that in this slightly more complex image, there are more points giving spurious results. The model appears confused by the existence of two different motions in the image, more so than in the 1-D case.

Test image 3 showed a moving triangle that was also undergoing scaling. In this case, as is the case for test image 1, the motion and the variation of the motion across the object are captured well (Figure 5.12). Again, the later filter outputs succcessfully confine the flow to a region close to the edge of the object; points in the interior drop out of the output as the analysis proceeds to finer scales.

(a) Output at stage 6, $\lambda = 28$ (b) Output at stage 10, $\lambda = 7$ (c) Output at stage 13, $\lambda = 2.5$

Figure 5.10: Test image 1: Outputs at stages 6, 10 and 13.



(a) Output at stage 5, $\lambda = 40$ (b) Output at stage 9, $\lambda = 10$ (c) Output at stage 13, $\lambda = 2.5$

Figure 5.11: Test image 2: outputs at stages 5, 9 and 13.

(a) Output at stage 7, $\lambda = 20$ (b) Output at stage 10, $\lambda = 7$ (c) Output at stage 12, $\lambda = 3.5$

Figure 5.12: Test image 3: outputs at stages 7, 10 and 12.

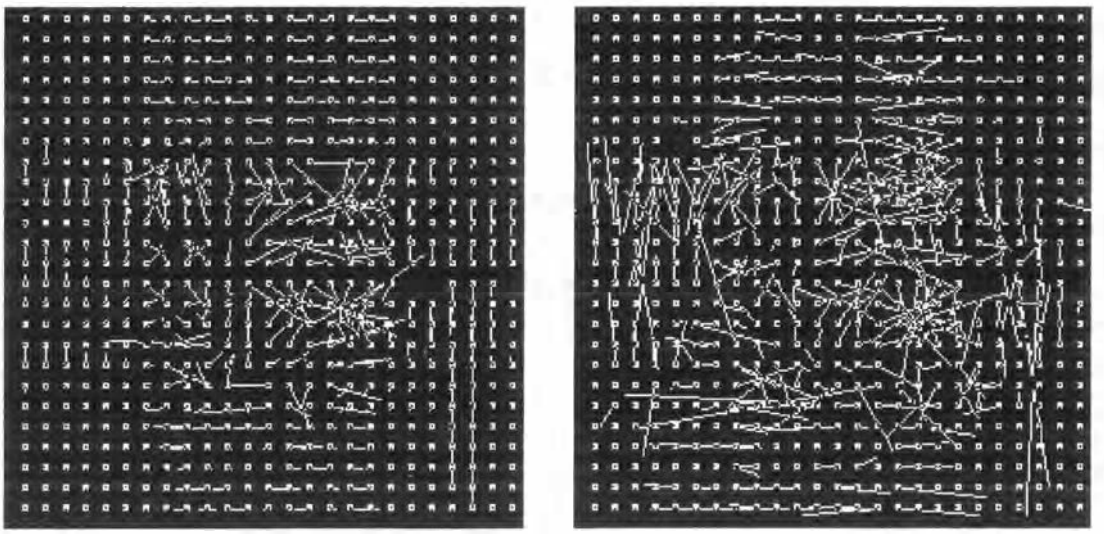
Results for images 4 (figure 5.13) and 5 (figure 5.14) are less encouraging. The best that can be said is that, for the most part, regions where the flow is zero or close to zero are correctly flagged as such. Where motion exists, it is detected, however the derived flow vectors are mostly unreliable, both in magnitude and direction. It is not certain why the performance for these kinds of images was worse than for other kinds, and also significantly worse than for the 1-D random dot stereogram. This is discussed further in the following section.

5.4 Interpretation of the Results

The method performs well in 2-D on synthetic images that are not too complex. The results from the first three test images show the possibility of using the flow field as an aid to segmentation.

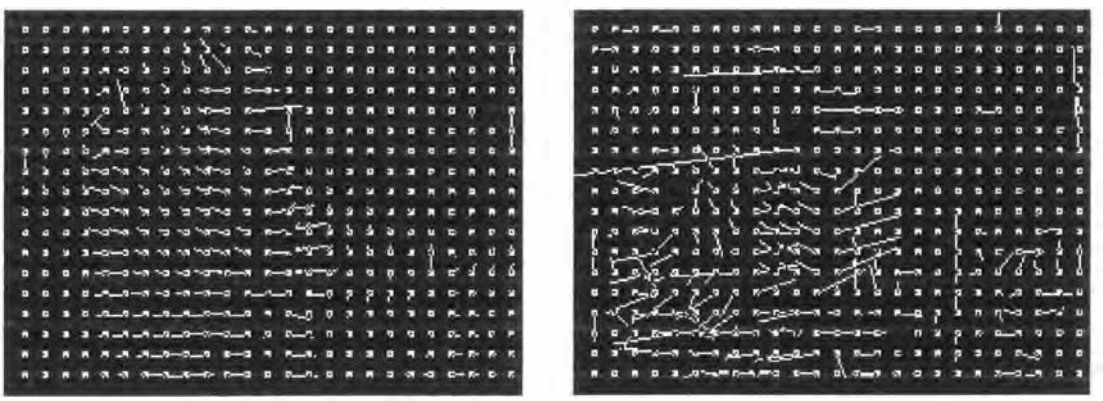
For more complex scenes, such as the random dot stereogram, or a real-world scene such as the taxi sequence, it is less successful. The method seems to identify well those regions where there is zero motion in the image. (There are of course far simpler ways to do this.) When motion exists, the actual value of the motion is not well captured.

It appears that the method is well suited to images where there are large ar-



(a) Output at stage 4, $\lambda = 56$ (b) Output at stage 9, $\lambda = 10$

Figure 5.13: Test image 4: Outputs at stages 4 and 9.



(a) Output at stage 4, $\lambda = 56$ (b) Output at stage 9, $\lambda = 10$

Figure 5.14: Test image 5: Outputs at stages 4 and 9.

eas of almost uniform brightness, but it needs refinement in order to handle images with more complex characteristics. The application of the combination of a horizontal Gabor filter and a vertical Gaussian (or vice versa) is designed to propagate information across the image in both the x and y directions, so that the motion of an object does not carry it out of the range of the filter being used in the flow derivation. It may be, however, that too much extraneous information from distant parts of the image is being fed to the filter. Particularly in the case of the random dot stereogram, the Gaussian spreading function would have averaged out the random dot data, making the filter responses very small and therefore possibly unreliable.

There is scope for tinkering with the filter parameters in various ways to see whether this improves the quality of the analysis for images with high information content. One possibility is to use filters with a much narrower Gaussian spreading function, to decrease the amount of extraneous information contributing to the filter response. To compensate for the narrow spread, and avoid the problem described in section 5.1.2 and depicted in figure 5.1, it would be necessary to employ additional filters oriented in various directions, as opposed to only the horizontal and vertical directions as used in the present work.

5.5 Summary

The generalisation of the method to two dimensions introduces extra complications. The Gradient Constraint Equation applied to a 2-dimensional image provides only a single velocity constraint, which is insufficient to determine the velocity (the "Aperture Problem"). It is therefore necessary to obtain at least two velocity constraints at each scale. In the present work this is done by convolving with two different complex kernels, viz. a horizontal Gabor kernel combined with a vertical Gaussian, and a vertical Gabor combined with a horizontal Gaussian. These two convolution operations result in two different resultant image sequences. The application of the Gradient Constraint Equation to these sequences gives two different velocity constraints, which are in general independent, and these constraints are then combined to give a single velocity estimate.

The results show that this particular combination performs well on certain types of images – those containing a small number of well-defined features

– but not on others. Disappointingly it yielded little in the way of useful results for the real-world test image sequence. Some reasons for this were postulated in section 5.4, along with some ideas for alternative choices of convolution kernels. This is a topic for future research.

Chapter 6

Conclusion

The multi-scale phase-based approach has been demonstrated to be feasible, though some problems remain.

In summary, the experiments on 1-dimensional test images demonstrate that:

- A phase-based differential analysis is capable of detecting motion and assigning correct motion values at points where the image contains sufficient information.
- The phase-based method is insensitive to changes in overall illumination level between succcessive image frames.
- The multi-scale analysis can detect fine motion in circumstances where the application of a narrow convolution kernel alone would miss the motion or give incorrect values.
- Velocity derivation by this method does not require the presence of discernible static features. It could therefore be used to segment an image based on the motions of the different segments.

For 2-dimensional images, the method shows promise, in that it works well on a restricted class of images — those containing only a few well defined features and with large areas of uniform brightness level. Images with many features or those where the intensity is variable across the image are not handled well.

Some suitable candidate topics for future research are:

- It needs to be determined why the analysis works well on some kinds of 2-D images and not others. The fact that some very realistic results were obtained, both for the 1-D case and for some of the 2-D images, suggests that for those 2-D images that gave poor results there may be a problem with the manner in which we have used 1-dimensional convolution kernels to propagate information across the image. We need to do such propagation if the motion is comparable with the size of the moving features — however experiments should be carried out using 2-D Gabor filters with a narrower Gaussian spreading function, and incorporating several orientations of such filters, not just horizontal and vertical orientations.

- Although a total of 13 filters were used in our work, the impression gained is that this is more than enough, and that computation time could be shortened without a significant drop in performance by using a smaller number of filters at scales more widely spaced. There is scope for formal investigation of the relation between the ratio of successive filter sizes and the accuracy of the motion estimates.

- It was mentioned in section 4.4 that there is a need for a new way of numerically describing the performance of a multi-scale filter, since existing methods based on disparity between measured flow and "true" flow are too crude and do not take into account the variation in information content across the image, or indeed the difference in information content available at different scales at the one point in the image.

- In some of our filter outputs, DC output from the smallest filter appears to be significant, contradicting the result given above that $\sigma > 0.485\lambda$ gives a minimal DC response. This is probably due to quantization effects; the DC response is the discrete sum of a number of terms, which differs from the area under a smooth curve, the difference being greater for a smaller number of terms. Further work needs to be done in fine-tuning the smallest filters in order to minimize this potential source of error.

- In most of the tests, no new information was obtained with the very smallest filters ($\lambda = 5$, $\lambda = 3.5$ and $\lambda = 2.5$). This is because the phase normally varies so rapidly across these filter outputs that most points fail the phase singularity detection tests.

In section 2.1.5 we suggested that the fact that phase contours appear to follow the movement of features in the image is due to the fact that phase is derived from a ratio rather than being related to absolute values of intensity. It is possible that other quantities based on ratios would have the same advantages as phase without some of the problems, such as the problem of dealing with points where phase flips from $-\pi$ to π. There are several candidates for such a parameter; one would be to take the intensity difference between neighbouring parts of an image divided by the average intensity over this whole neighbourhood.

- In one of the 1-D test cases (test image 4, section 4.2.2), the method performed poorly in a part of the image where disocclusion of a feature was occurring. There is scope for research into how occlusion and disocclusion could be better handled. In this work, each test sequence consisted of only two images. This particular test image pair happened to contain a feature in one image that was entirely absent in the other image. It is likely that a method that employed longer image sequences could handle this situation better — if the occluded feature appeared in several frames, the method might be able to determine its motion and thereby distinguish this from the motion of the occluding feature.

- Fleet sometimes denotes velocity by a velocity angle θ, where $\tan\theta = x/t$. This avoids the infinities problem: equation 4.18 becomes:

$$\tan\theta = -\frac{\partial\phi/\partial t}{\partial\phi/\partial x}. \tag{6.1}$$

  so that when $\partial\phi/\partial x$ is much smaller in magnitude than $\partial\phi/\partial t$, the velocity angle approaches $\pm\pi/2$ rather than tending toward infinity.

  In describing his velocity filters, Fleet refers to one filter called a "flicker channel" which captures velocity angles close to $\pi/2$, which we would refer to as infinite velocity. This condition would exist at points in the image where the function (in our case the phase) was constant in space but changing with time. Such phenomena can occur in real scenes, eg. a flashing light. We feel that such a velocity estimate should be retained, since it provides real information about the scene. One way to store this velocity would be to use a velocity angle, as Fleet does; another would be to simply store the pair (x, t).

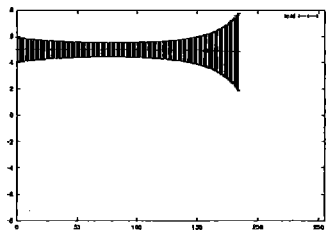
Appendix A

Selected 1-D Output Sequences

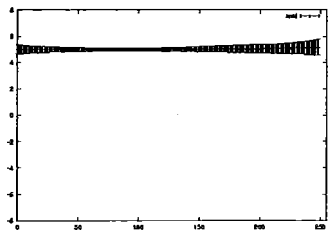
The complete output image sequences for test images 1, 3 and 6 are presented.

The image x coordinate is represented on the horizontal scale, while the value of the velocity estimate v is shown on the vertical scale. The velocity is represented as a vertical error bar; v is marked by the centre of the bar, while the extent of the bar above and below v give the uncertainty.

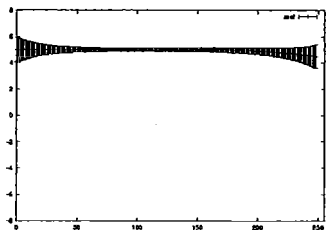
For clarity, where the uncertainty exceeds a selected threshold value, the output is omitted. Gaps in the output therefore indicate regions where the uncertainty was high because insufficient flow information was available at the relevant scale. In each of the outputs presented here, the threshold uncertainty was 3.0 pixels per frame unless otherwise stated.



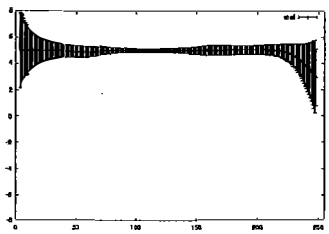
(a) Stage 1



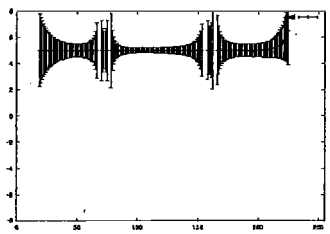
(b) Stage 2



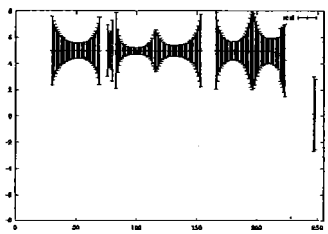
(c) Stage 3



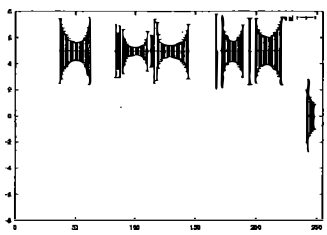
(d) Stage 4



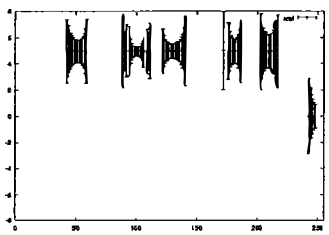
(e) Stage 5



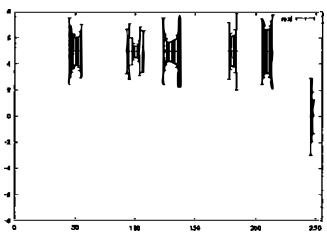
(f) Stage 6



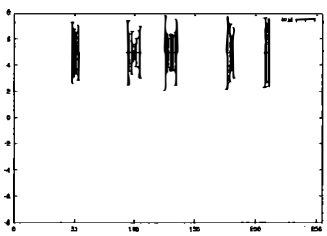
(g) Stage 7



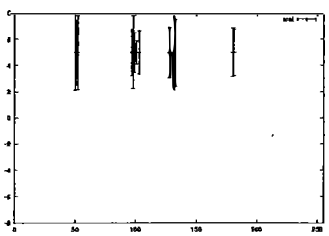
(h) Stage 8



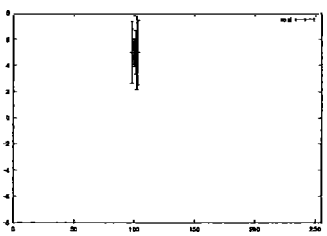
(i) Stage 9



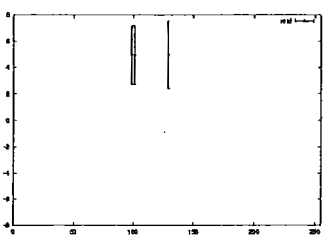
(j) Stage 10



(k) Stage 11

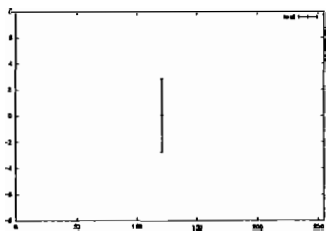


(l) Stage 12

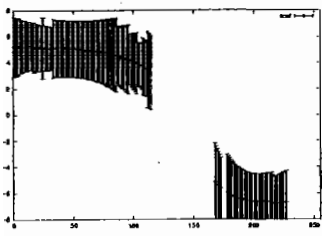


(m) Stage 13

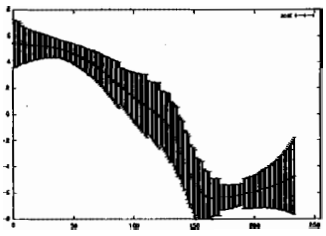
Figure A.1: Output flow estimates for 1-D Test Image 1



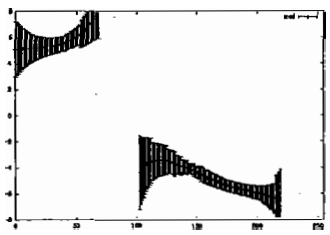
(a) Stage 1



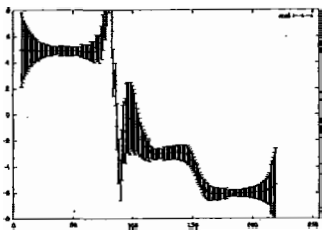
(b) Stage 2



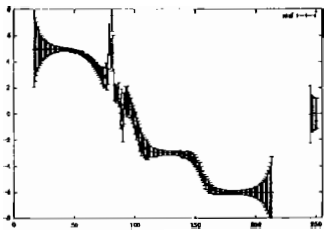
(c) Stage 3



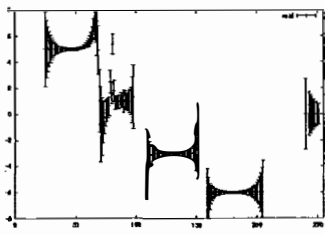
(d) Stage 4



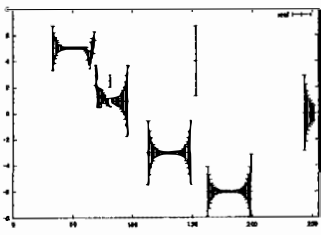
(e) Stage 5



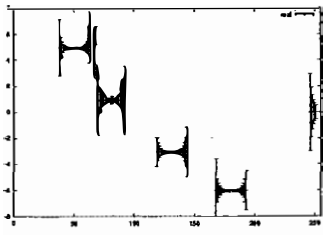
(f) Stage 6



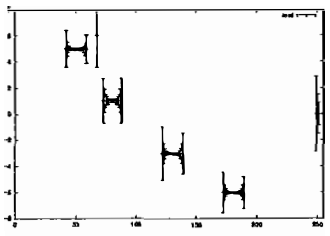
(g) Stage 7



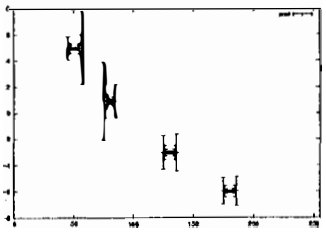
(h) Stage 8



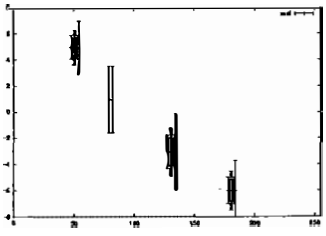
(i) Stage 9



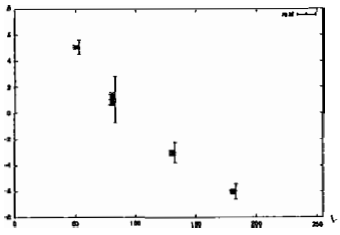
(j) Stage 10



(k) Stage 11

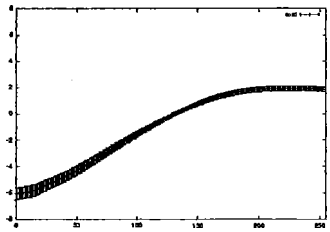


(l) Stage 12

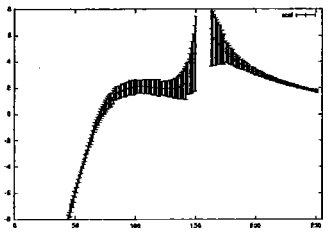


(m) Stage 13

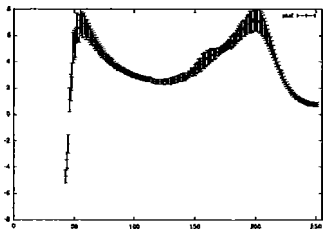
Figure A.2: Output flow estimates for 1-D Test Image 3



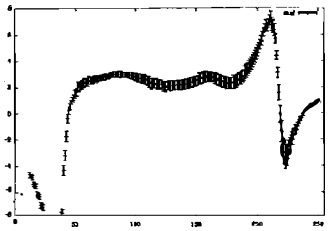
(a) Stage 1



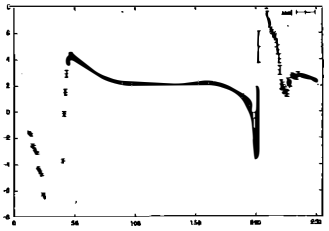
(b) Stage 2



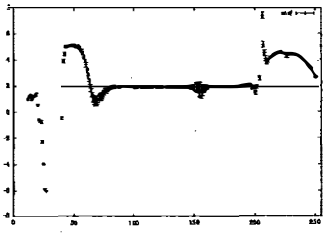
(c) Stage 3



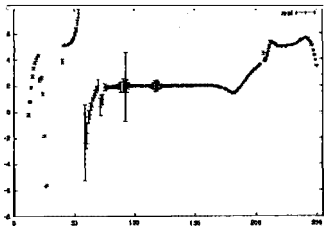
(d) Stage 4



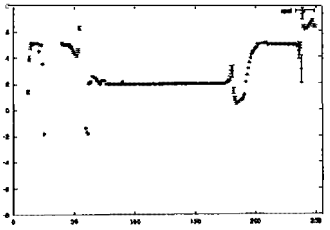
(e) Stage 5



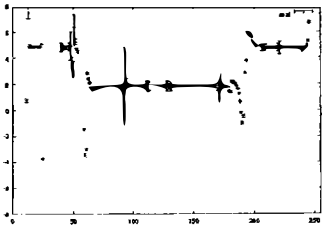
(f) Stage 6



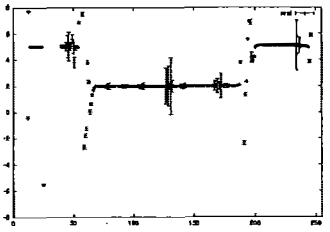
(g) Stage 7



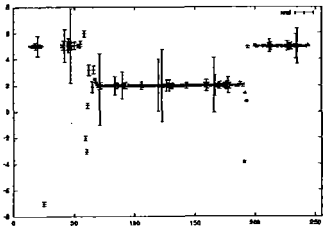
(h) Stage 8



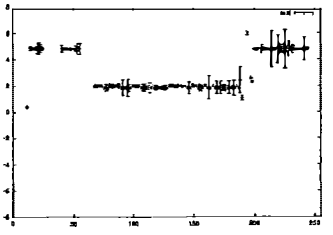
(i) Stage 9



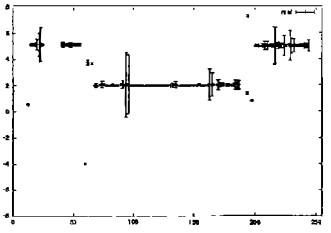
(j) Stage 10



(k) Stage 11



(l) Stage 12



(m) Stage 13

Figure A.3: Output flow estimates for 1-D Test Image 6

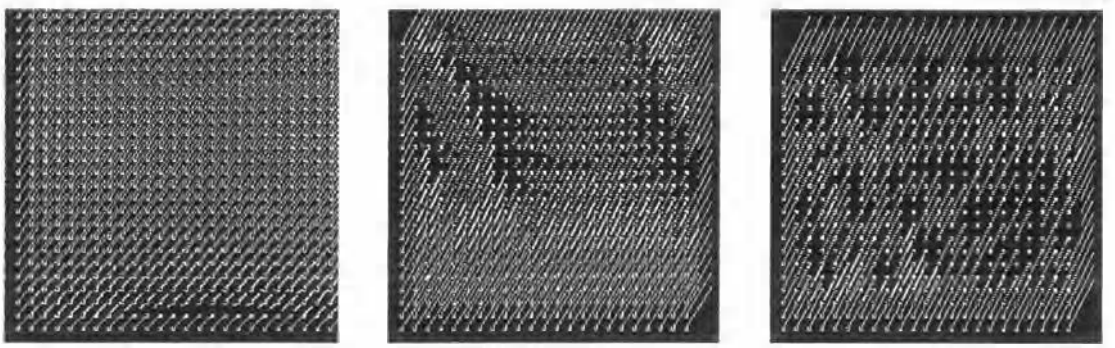
Appendix B

Selected 2-D Output Sequences

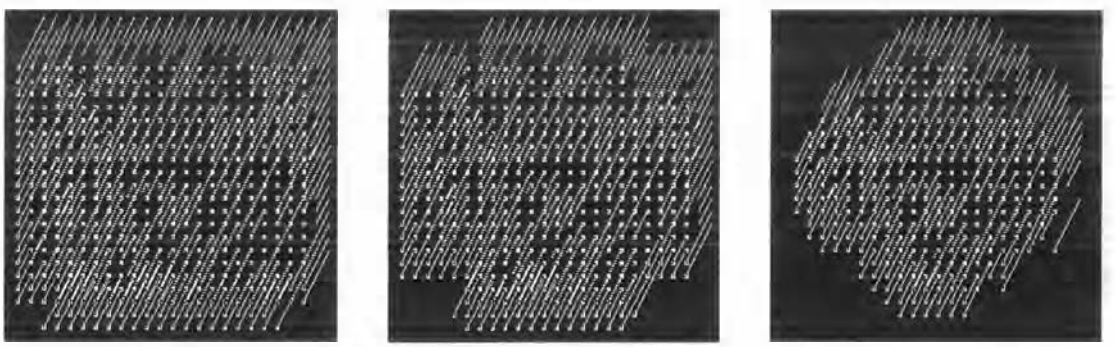
The complete output image sequences for 2-D test images 1, 3 and 5 are presented here.

The image velocity estimates are represented by "needles" — lines superimposed on the first image frame of the sequence. The origin of each needle is marked by a dot; this is the point to which the velocity estimate refers. The vector from the origin to the end point of the needle indicates the flow vector.

For clarity, as for the 1-dimensional test results, the flow estimates are omitted where the uncertainty is large. In this case, however, we deal with an uncertainty covariance matrix rather than a scalar value of uncertainty. The determinant of the covariance matrix is therefore used to set the display criterion, as explained in section 5.3. The corresponding threshold value is 50 pixels per frame in all cases.

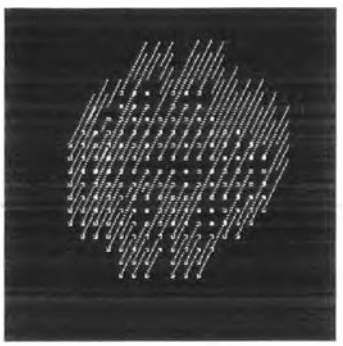


(a) Stage 1 (b) Stage 2 (c) Stage 3

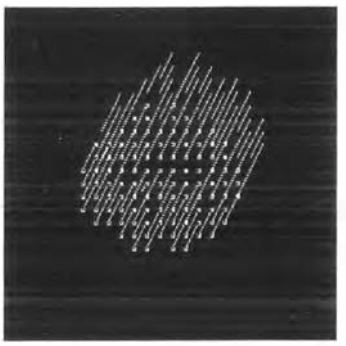


(d) Stage 4 (e) Stage 5 (f) Stage 6

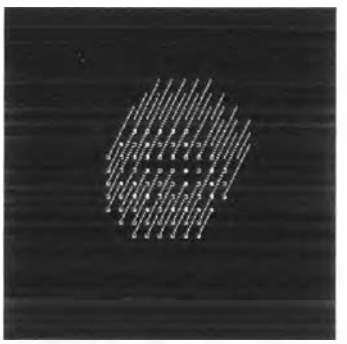
Figure B.1: Output flow estimates for 2-D Test Image 1, stages 1-6



(a) Stage 7



(b) Stage 8



(c) Stage 9

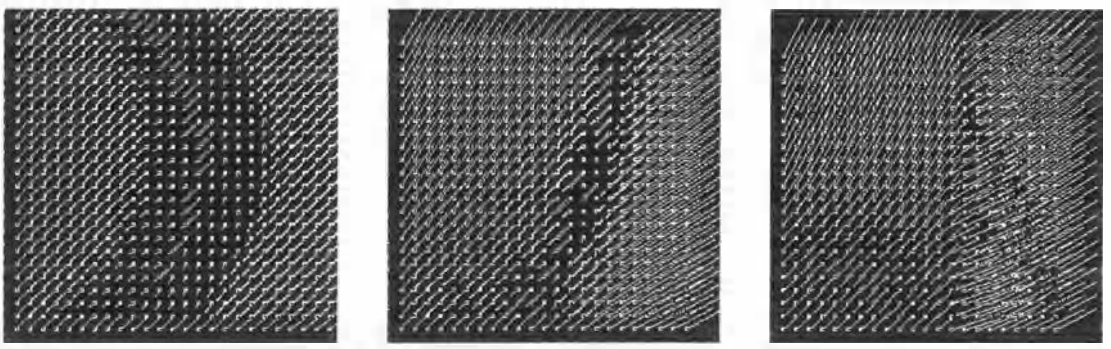
(d) Stage 10

(e) Stage 11

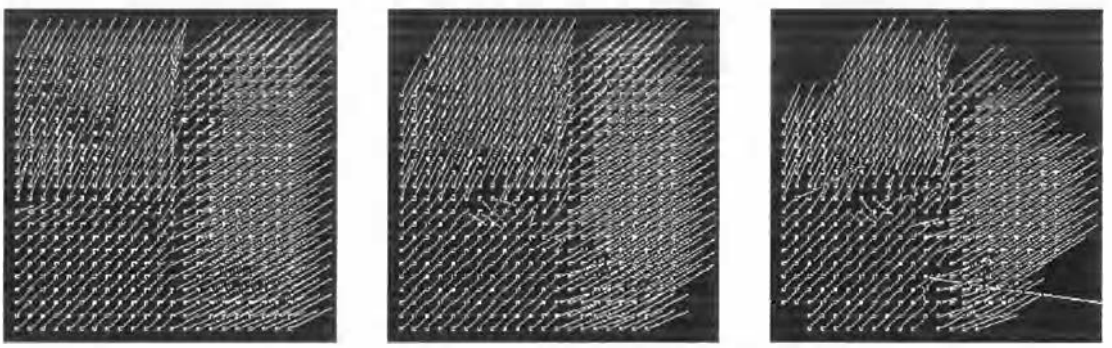
(f) Stage 12

(g) Stage 13

Figure B.2: Output flow estimates for 2-D Test Image 1, stages 7-13

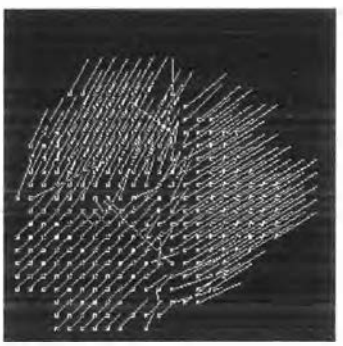


(a) Stage 1 (b) Stage 2 (c) Stage 3



(d) Stage 4 (e) Stage 5 (f) Stage 6

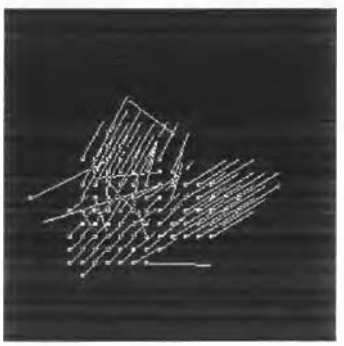
Figure B.3: Output flow estimates for 2-D Test Image 3, stages 1-6



(a) Stage 7

(b) Stage 8

(c) Stage 9



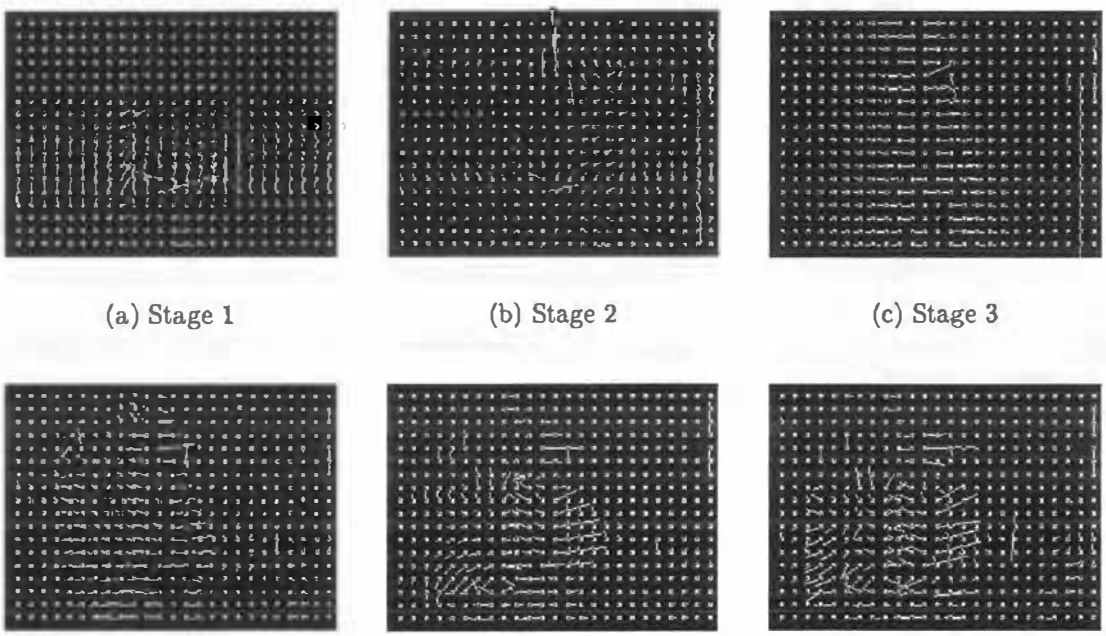
(d) Stage 10

(e) Stage 11

(f) Stage 12

(g) Stage 13

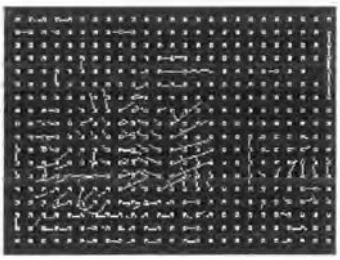
Figure B.4: Output flow estimates for 2-D Test Image 3, stages 7-13



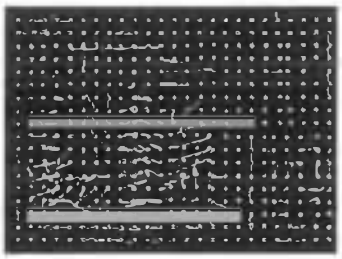
(a) Stage 1 (b) Stage 2 (c) Stage 3

(d) Stage 4 (e) Stage 5 (f) Stage 6

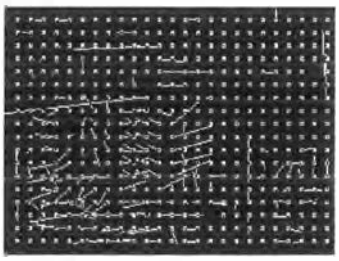
Figure B.5: Output flow estimates for 2-D Test Image 5, stages 1-6



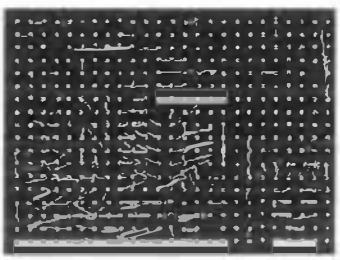
(a) Stage 7



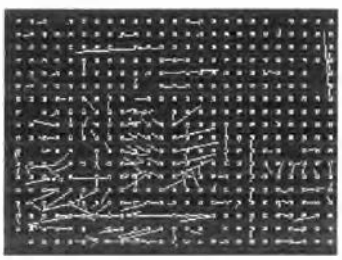
(b) Stage 8



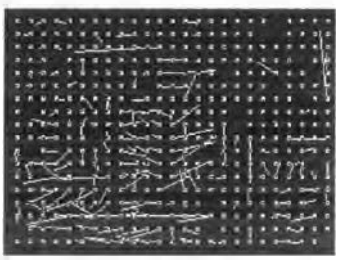
(c) Stage 9



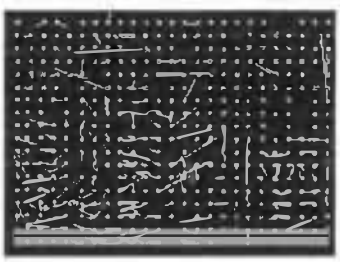
(d) Stage 10



(e) Stage 11



(f) Stage 12



(g) Stage 13

Figure B.6: Output flow estimates for 2-D Test Image 5, stages 7-13

Bibliography

Agarwal, R. and J. Sklansky (1992). Estimating optical flow from clustered trajectories in velocity-time. In *Proceedings, 11th IAPR International Conference on Pattern Recognition, 1992, Conference A: Computer Vision and Applications*, Volume 1, pp. 215–219.

Aggarwal, J. and N. Nandhakumar (1988, Aug). On the computation of motion from sequences of images-a review. *Proceedings of the IEEE 76*, 917–935.

Aggarwal, J. K. and R. O. Duda (1975, Oct). Computer analysis of moving polygonal images. *IEEE Transactions on Computers C-24*(10), 966–976.

Ballard, D. and C. Brown (1982). *Computer Vision*. Englewood Cliffs, N.J.: Prentice-Hall.

Barnard, S. T. and W. B. Thompson (1980). Disparity analysis of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence PAMI-2*(4), 333–340.

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

Barron, J., D. Fleet, S. Beauchemin, and T. Burkitt (1992, June). Performance of optical flow techniques. In *Proceedings. 1992 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Champaign, Illinois, USA, pp. 236–242. IEEE: IEEE Computer Society Press.

Bergholm, F. (1987, Nov). Edge focusing. *IEEE Transactions on Pattern Analysis and Machine Intelligence 9*(6), 726–741.

Burt, P., C. Yen, and X. Xu (1983). Multi-resolution flow-through motion analysis. In *IEEE Computer Vision and Pattern Recognition Conference Proceedings, Washington D.C.*, pp. 246–252. IEEE.

Chen, L.-F., H.-Y. M. Liao, and J.-C. Lin (1992, Jan). Wavelet-based optical flow estimation. *IEEE Transactions on Circuits and Systems for Video Technology 12*, 1–12.

Chen, T.-Y. and A. Bovik (1995). Stereo disparity from multiscale processing of local image phase. In *Proceedings, International Symposium on Computer Vision, 21-23 Nov. 1995*, pp. 188–193.

Chow, W. M. and J. K. Aggarwal (1977, February). Computer analysis of planar curvilinear moving images. *IEEE Transactions on Computers C-26*, 179–185.

Cooper, J. (1992, August). *Real-time Task-directed Robot Vision.* Ph. D. thesis, Department of Computer Science, University of Western Australia.

Cooper, J. and S. Venkatesh (1996, December). Uncertainty in the optical flow using the kalman filter. In *Proceedings. Fourth international conference on Control, Automation, Robotics and Vision*, Volume 3, Westin Stamford, Singapore, pp. 1681–1686. School of Electrical and Electronic Engineering, Nanyang Technical University, Singapore: Nanyang Technical University, Singapore.

Cooper, J. R. and R. O. Hastings (1997, Dec). Kalman filtering from a phase based optical flow operator. In *Proceedings of the Tenth Australian Joint Conference on Artificial Intelligence.*

del Bimbo, A., P. Nesi, and J. L. Sanz (1996, May). Optical flow computation using extended constraints. *IEEE Transactions on Image Processing 5*(5), 720–739.

Du Plessis, M. (June 1967). Poor man's explanation of Kalman filtering or how I stopped worrying and learned to love matrix inversion. Technical report, North American Aviation, Inc., Autonetics Division 3370 Miraloma Avenue, California 92803.

Duncan, J. and T.-C. Chou (1992). On the detection of motion and the computation of optical flow. *IEEE Transactions on Pattern Analysis and Machine Intelligence 14*(3), 346–352.

El Zaart, A., D. Ziou, and F. Dubeau (1997). Phase-based disparity estimation: a spatial approach. In *Proceedings, IEEE International Conference on Image Processing, 1997*, Volume 3, pp. 244–247.

Fennema, C. L. and W. B. Thompson (1979). Velocity determination in scenes containing several moving objects. *Computer Graphics and Image Processing 9*, 301–315.

Fermuller, C., D. Shulman, and Y. Aloimonos (2000). The statistics of optical flow. *Computer Vision and Image Understanding 82*, 1–32.

Fleet, D. and A. Jepson (1993, December). Stability of phase information. *IEEE Transactions on Pattern and Machine Intelligence 15*(12), 1253–1268.

Fleet, D. and K. Langley (1995, Jan). Recursive filters for optical flow. *IEEE Transactions on Pattern and Machine Intelligence 17*(1), 61–67.

Fleet, D. J. (1992). *Measurement of Image Velocity*. Kluwer Academic Publishers: Boston, USA.

Fleet, D. J. (1994). Disparity from local weighted phase-correlation. In *IEEE International Conference on Systems, Man, and Cybernetics, 1994. 'Humans, Information and Technology'*, Volume 1, pp. 48–54.

Ghosal, S. and R. Mehrotra (1994). Robust optical flow estimation. In *Proceedings, IEEE International Conference on Image Processing, 1994*.

Ghosal, S. and P. Vaněk (1996, Feb). A fast scalable algorithm for discontinuous optical flow estimation. *IEEE transaction on pattern matching and machine intelligence 18*(2), 181–194.

Glazer, F., G. Reynolds, and P. Anandan (1983). Scene matching by hierarchical correlation. In *CVPR*, pp. 432–444.

Gökstorp, M. and P.-E. Danielsson (1994). Velocity tuned generalized sobel operators for multiresolution computation of optical flow. In *Proceedings, IEEE International Conference on Image Processing, 1994*.

Heeger, D. (1988). Optical flow using spatiotemporal filters. *International Journal of Computer Vision 1*, 279–302.

Hemmendorf, M., M. T. Andersson, and H. Knutsson (1999, Mar). Phase-based image motion and registration. In *IEEE Proceedings of International Conference on Acoustics, Speech, and Signal Processing, 15-19 March 1999*, Volume 6, pp. 3345–3348.

Hogg, D. C. (1977). A methodology for real time analysis. In *Proceedings of the 5th International Joint Conference on Artificial Intelligence, Aug. 22-25, 1977*, Cambridge, MA, pp. 627.

Horn, B. and B. Schunck (1981). Determining optical flow. *Artificial Intelligence 17*, 185–203.

Jahne, B. (1993). *Spatio-Temporal Image Processing - Theory and Scientific Applications*. Berlin/Heidelberg, Germany: Springer-Verlag.

Jain, R., W. N. Martin, and J. K. Aggarwal (1979). Segmentation through the detection of changes due to motion. *Computer Graphics and Image Processing 11*, 13–34.

Jain, R. and H. H. Nagel (1979, April). On the analysis of accumulative difference pictures from image sequences of real world scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence PAMI-1*(2), 206–214.

Jepson, A. D. and M. R. M. Jenkin (1989). The fast computation of disparity from phase differences. In *CVPR'89*, pp. 398–403.

Julesz, B. (1965, Feb). Texture and visual perception. *Scientific American 231*(2), 38–48.

Low, A. (1991). *Introductory Computer Vision and Image Processing.* UK: McGraw-Hill.

Martin, W. N. and J. K. Aggarwal (1978). Dynamic scene analysis: A survey. *Computer Graphics and Image Processing 7*, 365–374.

McCane, B., K. Novins, D. Crannitch, and B. Galvin (2001). On benchmarking optical flow. *Computer Vision and Image Understanding 84*, 126–143.

Mitiche, A. and P. Bouthemy (1996). Computation and analysis of image motion: a synopsis of current problems and methods. *International Journal of Computer Vision 19*(1), 29–55.

Nagel, H. H. (1981, August). Representation of moving rigid objects based on visual observations. *IEEE Computer*, 29–38.

Nagel, H. H. (1983). Displacement vectors derived from second-order intensity variations in image sequences. *Computer Vision, Graphics and Image Processing 21*, 85–117.

Ohta, N. (1996, Jul). Uncertainty models of the gradient constraint for optical flow computation. *IEICE Transactions on Information and Systems E79D*(7), 958–964.

Ouali, M., D. Ziou, and C. Laurgeau (1999). A cooperative multiscale phase-based disparity algorithm. In *Proceedings, IEEE International Conference on Image Processing, 1999*, Volume 3, pp. 24–28.

Papadimitriou, D. and T. Dennis (1994, Sep). Stereo disparity analysis using phase correlation. *Electronics Letters 30*(18), 1475–1477.

Paquin, R. and E. Dubois (1983). A spatio-temporal gradient method for estimating the displacement field in time-varying imagery. *Computer Vision, Graphics and Image Processing 21*, 205–221.

Poggio, T. (1984, Apr). Vision by man and machine. *Scientific American 250*, 68–78.

Potter, J. L. (1975, May). Velocity as a cue to segmentation. *IEEE Transactions on Systems, Man and Cybernetics*, 390–394.

Prager, J. M. and M. A. Arbib (1983). Computing the optic flow: The MATCH algorithm and prediction. *Computer Vision, Graphics and Image Processing 24*, 271–304.

Roach, J. W. and J. K. Aggarwal (1979, April). Computer tracking of objects moving in space. *IEEE Transactions on Pattern Analysis and Machine Intelligence PAMI-1*(2), 127–135.

Roach, J. W. and J. K. Aggarwal (1980). Determining the movement of objects from a sequence of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence PAMI-2*, 554–562.

Schunck, B. (1989). Image flow segmentation and estimation by constraint line clustering. *IEEE Transactions on Pattern and Machine Intelligence 11*(10), 1019–1027.

Snyder, W. E. (1981, Aug). Computer analysis of time-varying images. *IEEE Computer*, 7–9.

Solari, F., S. Sabatini, and G. Bisio (2001). Fast technique for phase-based disparity estimation with no explicit calculation of phase. *Electronics Letters 37*(23), 1382–1383.

Thompson, W. B. (1980, November). Combining motion and contrast for segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence PAMI-2*(6), 543–549.

Thompson, W. B. and S. T. Barnard (1981, Aug). Lower-level estimation and interpretation of visual motion. *IEEE Computer*, 20–28.

Thompson, W. B., K. M. Mutch, and V. A. Berzins (1985, July). Dynamic occlusion analysis in optical flow fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence PAMI-7*(4), 374–383.

Tsai, C.-J., N. Galatsanos, and A. Katsaggelos (1999). Optical flow estimation from noisy data using differential techniques. In *IEEE Proceedings of International Conference on Acoustics, Speech, and Signal Processing, 15-19 March 1999*, Volume 6, pp. 3393–3396.

Tsao, T.-R. and V. C. Chen (1992). A neural computational scheme for extracting optical flow from the gabor phase differences of successive images. In *IJCNN*, pp. 450–456.

Tsotsos, J. K., J. Mylopoulos, H. D. Convvey, and S. W. Zucker (1980). A framework for visual motion understanding. *IEEE Transactions on Pattern Analysis and Machine Intelligence PAMI-2*, 563–573.

Turner, R. (1986). Texture discrimination by gabor functions. *Biological Cybernetics 55*, 71–82.

Ullman, S. (1981, Aug). Analysis of visual motion by biological and computer systems. *IEEE Computer*, 57–69.

Vega-Riveros, J. and K. Jabbour (1989, Dec). Review of motion analysis techniques. *IEE Proceedings-Communications 136*, 397–404.

Verri, A. and T. Poggio (1989, Mar). Motion field and optical flow: Qualitative properties. *IEEE Transactions on Pattern Analysis and Machine Intelligence 11*(3), 490–498.

Weng, J., T. S. Huang, and N. Ahuja (1993). *Motion and Structure from Image Sequences*. Berlin: Springer-Verlag.

Xie, K., L. Naneycken, and A. Oosterlink (1996, Jan). Hierarchical motion estimation with smoothness constraints and postprocessing. *Optical Engineering 35*(1), 145–155.

Yang, Q. and S. D. Ma (1999, Mar). Intrinsic multiscale representation using optical flow in the scale-space. *IEEE Transactions on Image Processing 8*(3), 444–447.

Yu-Te Wu; Kanade, T., J. Cohn, and C.-C. Li (1998). Optical flow estimation using wavelet motion model. In *Sixth International Conference on Computer Vision, 4-7 Jan. 1998.*, pp. 992–998.