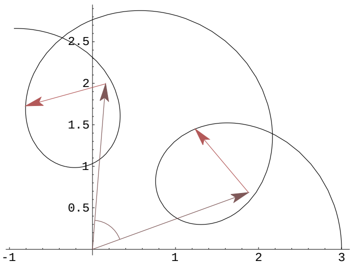
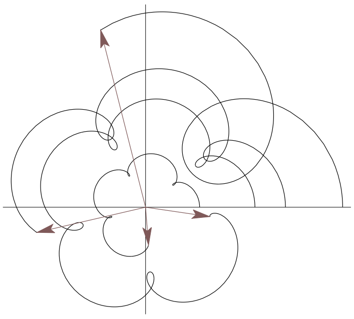
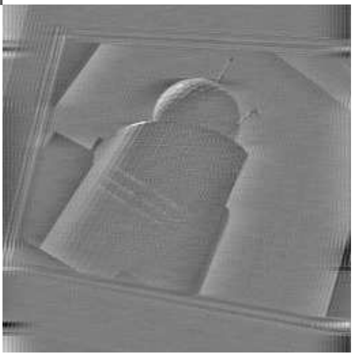
FOURIER VISION

Segmentation and Velocity Measurement using the Fourier Transform







DAVID VERNON

FOURIER VISION
Segmentation and Velocity Measurement using the Fourier Transform

DAVID VERNON
Department of Computer Science
National University of Ireland, Maynooth

Kluwer Academic Publishers
Boston/Dordrecht/London

Contents

Preface

Doing research is a little bit like hill-walking: it's a highly-rewarding endeavour which, with effort, leads you to heights you've never imagined and gives you the exhilaration of viewing the world from new and thrilling perspectives.

Like hill-walking, in research you know roughly where you want to get to but, if you are breaking new ground, you don't know exactly how best to get there — some turns lead to blind gulleys, some to rivers that can't be forded, but others yet present unanticipated glades and meadows. One thing is sure, though; if you want to get to the summit, you have to slog up some very uninviting slopes in the constant knowledge that, if the conditions become too bad, you'll simply have to turn around and return through the mist to where you started.

This book is the story of one such expedition on the hills of scientific research. The particular range of hills — segmentation and velocity estimation — is well-trodden and has been explored by many day-trippers, hardened scramblers, and experienced mountaineers, all of whom forged a path according to their needs, skills, and insight. In the following pages, you will find a description of a new path which offers a somewhat different perspective. It doesn't lead all the way to the highest summit in this range but I don't think that path has been discovered yet anyway. However, it does scale a number of difficult pinnacles, such as segmentation of transparent objects and optical flow estimation near occluding boundaries, and my hope is that the landscape will prove attractive enough for others to come and take a look. With a bit of luck, they may find that the path leads them at least part of the way to their own particular destination.

On my trip over these hills, I've met and been accompanied by many explorers and I've benefitted immensely from their conversation and advice. I am particularly indebted to Dermot Furlong, Trinity College, Dublin, long-time friend and fellow-traveller, who lent encouragement on the steeper slopes, technical advice with the map-reading, and continuous good humour especially when my spirits were flagging. I also want to mention my wife Keelin and my two daughters Ciana and Georgina who put up with my long absences when I was blundering about on the slopes and welcomed me home, tired and bedraggled, when the day was done. Thank you all.

DAVID VERNON

Chapter 1

INTRODUCTION

1. COMPUTER VISION

The goal of computer vision is to endow a machine with the ability to understand the structure and composition of its physical surroundings through the use of visual data: time-varying two-dimensional projections of light energy reflected off and radiated by the physical objects constituting the three-dimensional physical world.

This definition of computer vision seems straight-forward enough — but it isn't, for it side-steps one very important question: What does it mean for a computer system (or any system) to 'understand' something? There are two possible answers, each giving a different interpretation of the concept of understanding in the context of vision.

The first interpretation, which we will refer to as *structural vision*, takes understanding to be the faithful reconstruction or inference of the physical world which gives rise to the flux of visible light-data that is sensed by the computer vision system. To understand in this sense means to know somehow, with minimal ambiguity, what spatial and temporal configurations of objects gives rise to the sensed optical data. This is a non-trivial problem simply because the very process of sensing the ambient visible light radiation pattern is a projective process with an inherent loss of information. Consequently, many computer vision systems concern themselves less with the strict recovery of exact causative spatio-temporal configurations and more with the construction of a system of constraints which allows the computer vision system to maximise the likelihood that the inferences it makes about the structure and composition of the originating spatio-temporal environment are correct.

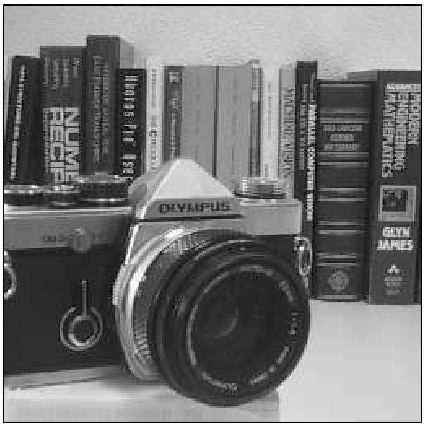


Figure 1.1. Structural composition of a scene: a camera appears in the foreground and the background comprises a shelf of books.

The second interpretation of the word understanding goes much further and embraces the possibility of inferring what the constituents of the physical world are used for. That it, it is explicitly concerned with both the structure of the spatio-temporal world and also with its purpose and intent (or, perhaps more pejoratively, its meaning). Such an approach might be termed *semantic vision.*

To illustrate the difference between the two approaches, consider the scene shown in Figure 1.1 which contains a camera standing in front of a shelf of books. The goal of a structural vision system would be to isolate each one of these two principal objects and to provide an accurate description of their relative position (or their absolute positions if we are using a calibrated vision system). Furthermore, the structural system might also assert that the foreground object has the same visual and spatial configuration of a class of objects which have been previously encountered and labelled as a camera by an outside agent.[1] Similarly, it would be able to assert that the background comprises a collection of objects which are particular instances of a general class of shape-determined (as opposed to functionally-determined) objects called books. All this is established through the processing and analysis of a 4-D spatio-temporal data-set comprising sensed light. This data-set inevitably carries much additional information caused by the presence of the shadows, additional objects, noise, and so on.

[1] Typically, the outside agent is the designer of the vision system or the user of the system.

On the other hand, the semantic interpretation would also require the computer vision system to somehow infer that the books are actually complex objects with pages containing information in written and graphic form, that the writing on the spines indicates the authors and subject matter, that the camera is an instrument for acquiring 2-D photographic images of the world, and that the two together are probably part of an office scene. Here, we go far beyond the structural and we take a leap into the world of epistemology. We require our computer vision system not only to be able to say what is the physical configuration of the scene it is viewing and perhaps to classify certain subsets of that configuration on the basis of *a priori* externally-mediated knowledge but also that it develop some concept of the usage or function of the constituents of the configuration.

In essence, structural computer vision is concerned with the physical organization of the constituents of a spatio-temporal world while semantic computer vision is concerned also with its purpose and general behaviour.

There can be little doubt but that the ultimate priority of computer vision is semantic vision, because only through the realization of systems with semantic capability can we fully integrate vision into complete robust intelligent physical systems which can act — and interact — as well as perceive. Unfortunately, semantic vision has proved to be a tough nut to crack, perhaps because we are not yet using the right tools. Since it seems reasonable that a semantic vision system should build upon resilient structural capabilities, our immediate priority is to construct robust models of structural vision in the expectation or hope that we will eventually get around to solving the problem of semantics.

This book is about structural computer vision. Important and challenging though semantic vision may be, we are interested here in the physical organization of the visual world and we seek to establish that structure without recourse to *a priori* or semantic knowledge. Even more, we would like to be able to accomplish as much as possible with as much elegance as possible and by deploying as few computational models as possible.

So what does it take to build a structural computer vision system? Arguably, three of the most important tasks in structural computer vision are

- Segmentation;
- The computation of stereo disparity;
- The computation of optical flow.

Segmentation is important since it provides explicit information regarding the localization of the objects in the image; stereo disparity provides the relative distance of the objects from the viewer; and optical flow provides a viewer-centred description of the dynamic movement of an object in its environment.

In this book, we shall see how we can accomplish all these tasks with one basic technique based on the Fourier transform. Note well, though, that we don't claim that any or all of them are accomplished optimally, just that they can be accomplished well and, in some circumstances, extremely well. What motivates the development of the technique to be described is its novelty, simplicity, and utility in appropriate circumstances.

2. OVERVIEW OF THE FOURIER VISION APPROACH

The Fourier technique described in this book is based on the behaviour of images which exhibit object movement. This movement may be either real or apparent. Real movement arises when an object changes position in an scene and that scene is imaged by one or more cameras (see Figure 1.2 (a)). Apparent motion arises in a static scene when the objects are viewed from different positions (see Figure 1.2 (b)). What binds all of these situations together is that there is relative movement between the constituents objects in images across the image sequence. In this book, we will concentrate on short sequences (two or four images) and simple translating movement over very short distances. Indeed, these two attributes are mutually compatible: unless the object is travelling very quickly and assuming that the images are taken close together — either in time or in space — the frame-to-frame movement can be well-approximated by small translations. Of course, if these conditions are violated and we are confronted with much more complex relative motions then, naturally, the approach will fail. Nonetheless, the type of motion we are speaking of occurs so frequently that it is worth paying special attention to it. In any case, there are no silver bullets in the field of computer vision (at least, not yet). Each approach, including this one, has its own strengths and weaknesses.

The strengths of the approach being presented here are several. First and foremost, it can deal with both transparent or translucent scenes — an area in which traditional space-domain approaches usually have significant difficulties — as well as conventional opaque scenes with occlusion. Second, segmentation is carried out in a transform domain so that the visual complexity of the scene is inconsequential (within certain limits). Third, the approach makes no *a priori* assumptions about

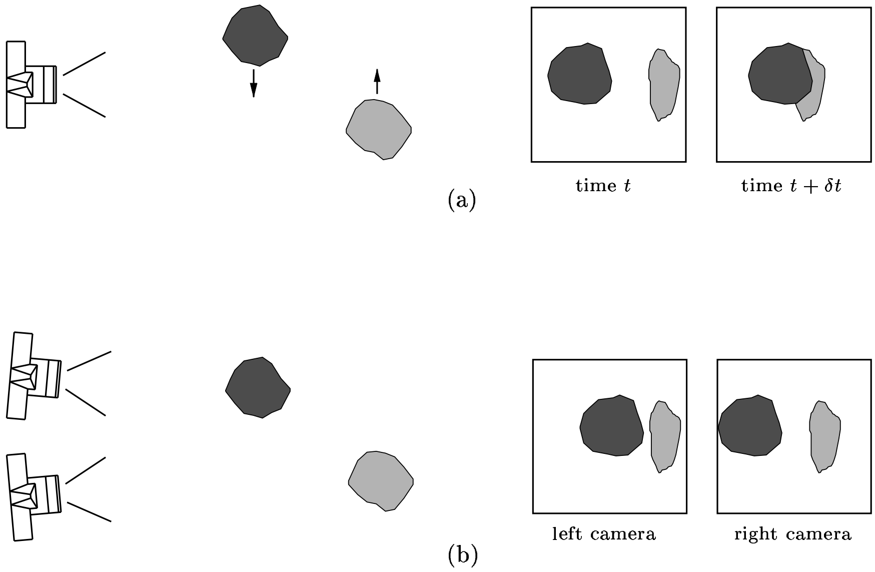


Figure 1.2. (a) Real movement arises when an object changes position in an scene and that scene is imaged by one or more cameras; (b) Apparent motion arises in a static scene when the objects are viewed from different positions.

the form or shape of the objects. Finally, it is applicable in most of the major visual configurations: single/multiple cameras, static/moving cameras, and static/dynamic scenes.

So, how does it work? Without pre-empting what is to come in subsequent chapters, we can summarise the approach as follows. Assume a visual scene comprises a number of objects. In a sense, an image of these objects is an image of each object added together. This is exactly true if the objects are transparent or translucent and it is figuratively true if we assume there is nothing behind an occluding foreground object (see Figure 1.3). If the objects are moving with real or apparent motion then we have an image sequence comprising a succession of 'added-together' object images. The goal then becomes one of finding the images which, when 'added together', produce the observed composite images. This is where the Fourier theory comes in.

Fourier processing is a transform technique and, therefore, it makes explicit in the transform domain certain useful or important properties or relationships which are implicit in the original domain. Just as the Laplace transform is used to turn difficult differential equations

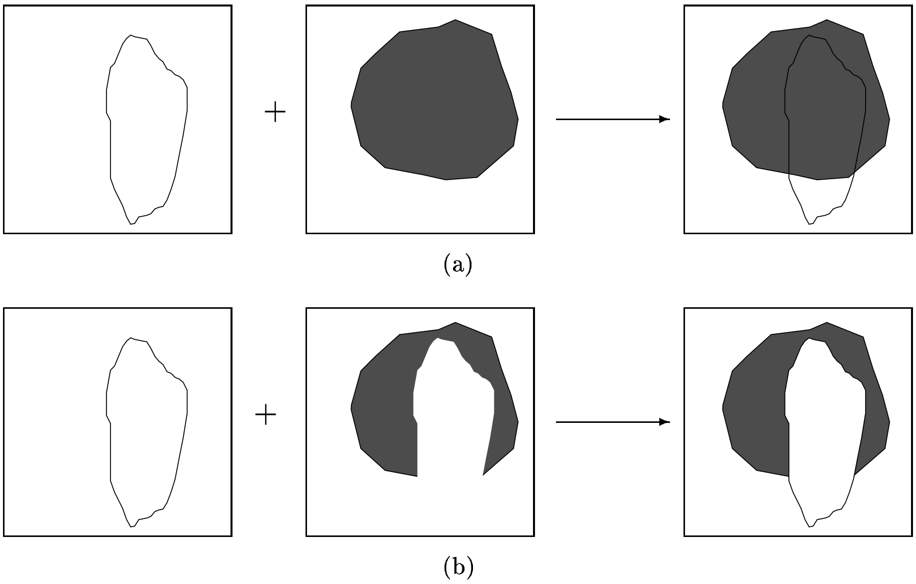


Figure 1.3. An image of composite objects is the sum of the two images: (a) exactly so in the case of transparent or translucent objects and (b) figuratively so in the case of occluding objects in which case there is 'hole' behind the foreground object.

into easy-to-handle algebraic equations, the Fourier transform takes the spatially-distributed information implicit in an iconic image representation and re-presents it in a fashion which reveals explicitly the information which we require.

In our case, and assuming we are dealing with just the foreground and the background objects, we need to know four things: the velocity or disparity of the foreground and background images and the structural composition — the appearance — of the images of the foreground and background. These four issues are inextricably linked in the spatio-temporal domain since the images change as the objects move. However, as we will see, all motion information is localized in one attribute of the Fourier domain: the phase. This allows us to decouple the motion problem from the structure problem. In essence, we find the velocities first and then, knowing these, we can solve the remainder of the problem, computing the images of the constituent objects, Fourier component by Fourier component. We can then reconstruct the images of the constituent objects from their respective decoupled Fourier components.

More formally, the Fourier representation makes it possible to formulate the segmentation problem as the solution to a set of equations based on individual Fourier components rather than as a difficult problem in the interpretation of distributed spatial information about form and behaviour.

3. CAMERA CONFIGURATIONS

We have already mentioned that the approach described in this book can be deployed with several camera configurations. In the following chapters, we will deal with four. These are:

1. Fixed monocular vision, in which a sequence of four images is acquired by a single static camera observing objects moving in the scene (Figure 1.4 (a));

2. Articulated monocular vision, in which a sequence of four images is acquired by a single translating camera observing a static scene (Figure 1.4 (b));

3. Articulated binocular vision, in which two pairs of images are acquired at two distinct positions by a binocular stereo rig fixating on a static point (Figure 1.4 (c));

4. Fixed binocular vision, in which a single pair of images is acquired by a forward-facing fronto-parallel binocular stereo rig (Figure 1.4 (d)).

Each of these four configurations requires a slightly different model but each is a variant of the basic theory which we will develop for the monocular case in Chapter 3.

In addition to these variations in camera configuration, the approach also deals both with the separation of additive images (*i.e.* images which are the result of additive superposition of two component images) and with the segmentation of occluding images (*i.e.* images which result from the occlusion of one object by another).

Since object/camera motion is relative, cases 1 and 2 above are effectively equivalent (as long as both camera and objects respect the assumption of translating fronto-parallel motion with constant velocity) and we will deal with them as two variants of the one scenario. In the case of articulated binocular vision, we are interested in modelling a pair of cameras each of which acquires a pair of stereo images at two different spatial positions. In the case of fixed binocular vision, we note that it has a corresponding relative-motion counterpart in which a single camera acquires images at two distinct positions. In the following, we will treat the equivalent four-image monocular cases together, first for additive images in Chapter 3 and then for occluding images in Chapter 4. We will then proceed to consider the four-image articulated binocular case in Chapter 5. In Chapter 6, we will look at the fixed two-image binocular case and its relative-motion counterpart.

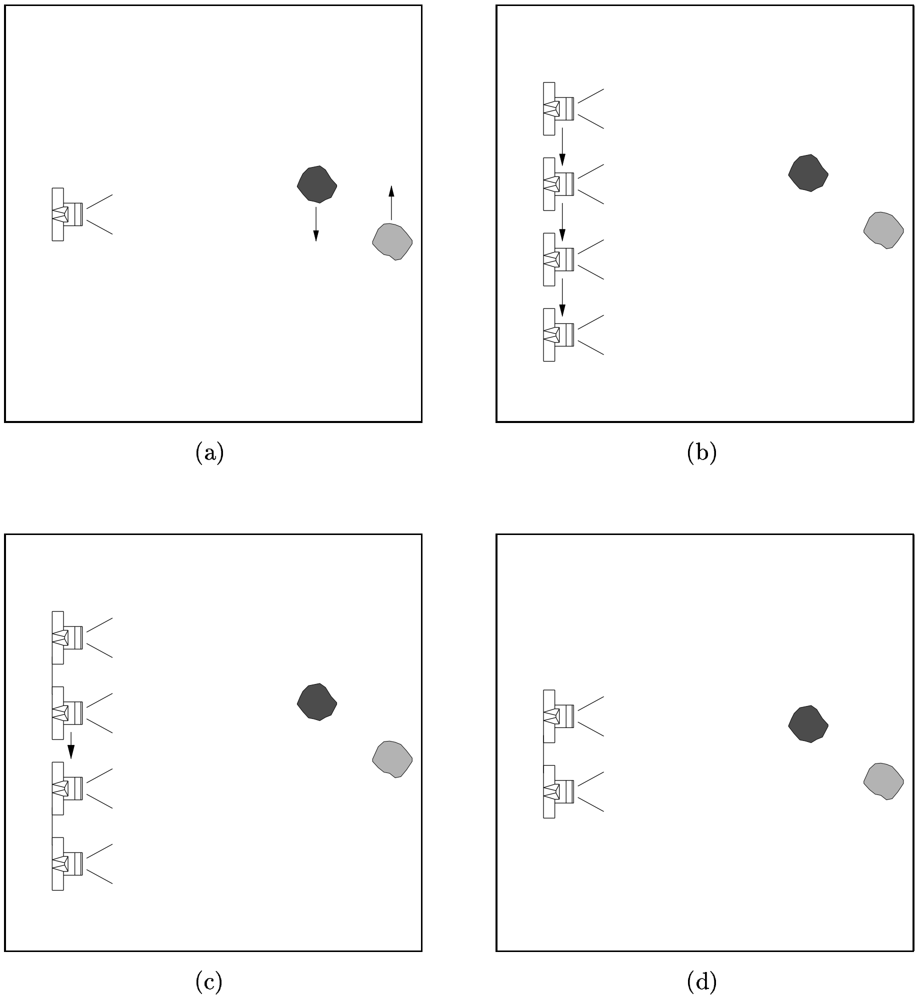


Figure 1.4. Camera configurations (a) Fixed monocular; (b) Articulated monocular; (c) Articulated binocular; (d) Fixed binocular.

4. SCOPE OF THE BOOK

This book describes a particular technique for figure-ground segmentation, velocity/disparity estimation, and computation of optical flow fields. It is *not* a general book on the segmentation problem; although we will make passing reference to alternative models and approaches in the following, such references will be made for illustration, contrast, and comparison only. It is not intended to that they constitute and exhaustive and comprehensive treatment of either the segmentation problem, the computation of stereo disparity, or the estimation of optical flow. With that said, a reasonably comprehensive bibliography is included at the end of the book and this will provide the reader with a good jumping-off point for further research.

Chapter 2

MATHEMATICAL PRELIMINARIES

We begin by reviewing the two principal mathematical tools which will be used in the subsequent chapters: the Fourier tranform and the Hough transform. Although both of these transforms will be familiar to most readers, and those who are expert in the area can safely skip over this chapter, many will benefit from at least a cursory glance at the material since it sets out the framework to be used throughout the rest of the book and introduces a number of ways in which the transform domains can be used to extract the required information. In particular, we will look at the relationship between the continuous Fourier transform and the discrete Fourier transform, the exponential representation of Fourier components and the associated magnitude/phase notation, and the effect of translation in the space domain on the phase of the Fourier components. We will also look at the general formulation of the Hough transform and the ways the Hough transform can be used to fit data to a given model and thereby identify the data-specific parameters of that model.

1. THE 2-D FOURIER TRANSFORM

1.1 THE CONTINUOUS FOURIER TRANSFORM

Consider a real-valued continuous two-dimensional spatial function $f(x,y)$, where the independent variables x and y denote displacement in orthogonal spatial directions. The Fourier transform of $f(x,y)$ is written as $\mathcal{F}(f(x,y))$ and the transformed function is written $\mathsf{F}(\omega_x,\omega_y)$. The independent variables in the Fourier domain, ω_x and ω_y, denote spatial frequencies in the x and y directions, respectively. The Fourier transform itself is defined as follows.

$$\begin{aligned}\mathcal{F}\left(f(x,y)\right) &= \mathsf{F}(\omega_x,\omega_y)\\ &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)e^{-i(\omega_x x+\omega_y y)}\mathrm{d}x\mathrm{d}y \qquad (2.1)\end{aligned}$$

where $i=\sqrt{-1}$.

The inverse Fourier transform is given by:

$$\begin{aligned}f(x,y) &= \mathcal{F}^{-1}\left(\mathsf{F}(\omega_x,\omega_y)\right)\\ &= \frac{1}{(2\pi)^2}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \mathsf{F}(\omega_x,\omega_y)e^{i(\omega_x x+\omega_y y)}\mathrm{d}\omega_x\mathrm{d}\omega_y \qquad (2.2)\end{aligned}$$

A few immediate remarks are necessary.

- First, equation 2.2 effectively states that the function $f(x,y)$ can be constructed from a linear combination of elementary functions having the form $e^{i(\omega_x x+\omega_y y)}$, each appropriately weighted in amplitude and phase by a complex factor $\mathsf{F}(\omega_x,\omega_y)$ which, in turn, is given by equation 2.1. Both of these equations, called the *Fourier transform pair*, apply if $f(x,y)$ is continuous and integrable and if $\mathsf{F}(\omega_x,\omega_y)$ is integrable.

- Second, although $f(x,y)$ need not necessarily be a real function and can be complex, we are interested here in real spatial functions. Specifically, we are interested in 2-D reflectance functions of a visual scene. On the other hand, as we just noted, the Fourier transform $\mathsf{F}(\omega_x,\omega_y)$ is generally a complex function with $\mathsf{F}(\omega_x,\omega_y) = \mathsf{A}(\omega_x,\omega_y)+i\mathsf{B}(\omega_x,\omega_y)$ where $\mathsf{A}(\omega_x,\omega_y)$ and $\mathsf{B}(\omega_x,\omega_y)$ are the real and imaginary components of $\mathsf{F}(\omega_x,\omega_y)$, respectively. It will turn out that this complex nature of the Fourier transform is the key to the solution of the vision problems we are addressing in this book.

- Third, we note that the domain of spatial frequecies includes negative spatial frequencies as well as positive ones, *i.e.* $-\infty \leq \omega_x,\omega_y \leq +\infty$.

The Fourier transform is sometimes defined a little differently (albeit equivalently) as follows.

$$\begin{aligned}\mathcal{F}\left(f(x,y)\right) &= \mathsf{F}(k_x,k_y)\\ &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)e^{-2\pi i(k_x x+k_y y)}\mathrm{d}x\mathrm{d}y \qquad (2.3)\end{aligned}$$

and

$$\begin{aligned}f(x,y) &= \mathcal{F}^{-1}\left(\mathsf{F}(k_x,k_y)\right)\\ &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \mathsf{F}(k_x,k_y)e^{2\pi i(k_x x+k_y y)}\mathrm{d}k_x\mathrm{d}k_y \qquad (2.4)\end{aligned}$$

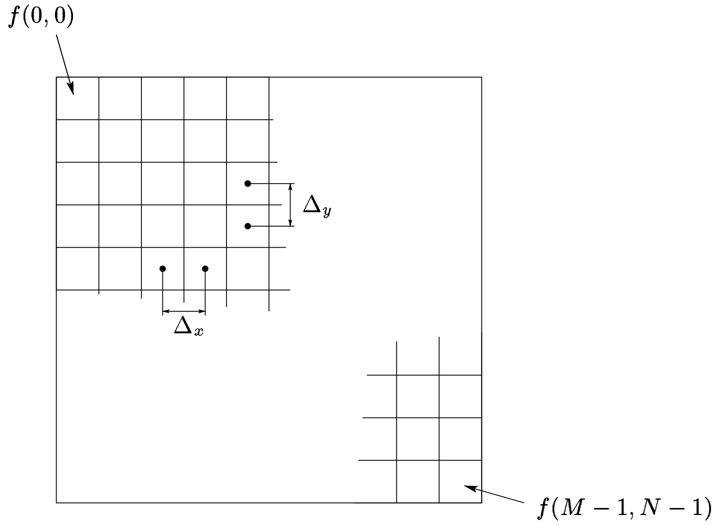


Figure 2.1. A discretely-sampled image $f(x,y)$ comprising $M \times N$ samples, with inter-sample distances Δx and Δy in the x and the y directions.

Apart from the absence of the scaling factor term $\frac{1}{(2\pi)^2}$ compared with equations (2.1) and (2.2), the essential difference in this form of the definition is that we have replaced ω_x and ω_y with $2\pi k_x$ and $2\pi k_y$: ω_x and ω_y are angular spatial frequencies (given in units of radians/unit distance) and k_x and k_y are spatial frequencies (given in units of [unit distance]$^{-1}$). In this book, we will use the former definition throughout and we will adopt the term spatial frequency for ω_x and ω_y on the understanding that we are actually referring to angular spatial frequencies.

1.2 THE DISCRETE FOURIER TRANSFORM

In most instances, and certainly throughout this book, we are dealing with sampled functions — digital images — rather than continuous functions. In this case, we require a discrete Fourier transform (DFT). Let's assume we are dealing with images with M uniformly-spaced samples in the x direction and N samples in the y direction, and that the inter-sample distances are Δx and Δy, respectively (see Figure 2.1). That is, our image is a function $f(x,y)$ such that

$$f(x,y) = f(x\Delta_x, y\Delta_y)$$

where $x = 0,1,2,\ldots M-1, y = 0,1,2,\ldots N-1$.

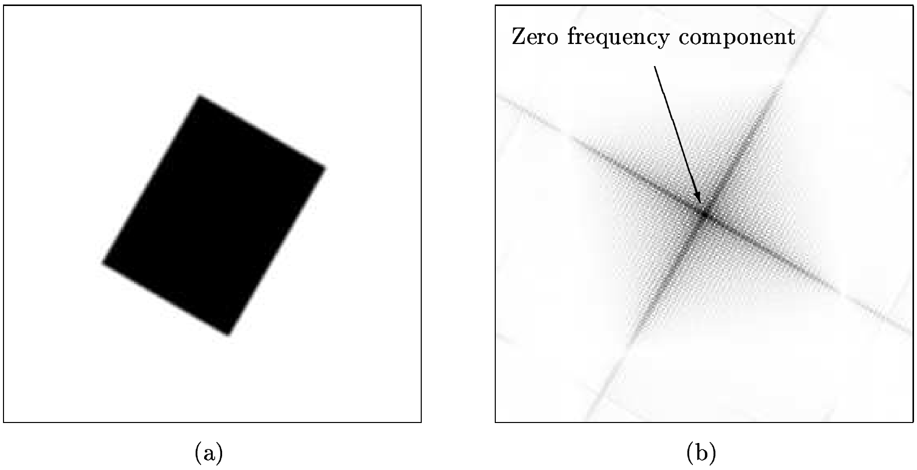


Figure 2.2. The Discrete Fourier transform: (a) a simple $N \times N$ pixel image; (b) Fourier transform with the zero frequency component located at pixel $\left(\frac{N}{2}, \frac{N}{2}\right)$.

The discrete Fourier transform of $f(x, y)$ is then defined as:

$$\begin{aligned}\mathcal{F}(f(x,y)) &= \mathsf{F}(\omega_x, \omega_y)\\ &= \mathsf{F}(\omega_x \Delta_{\omega_x}, \omega_y \Delta_{\omega_y})\\ &= \sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f(x,y) e^{-i(\frac{\omega_x x}{M} + \frac{\omega_y y}{N})} \qquad (2.5)\end{aligned}$$

where $\omega_x = 0, 1, 2, \ldots M-1, \omega_y = 0, 1, 2, \ldots N-1$. The inverse discrete Fourier transform is given by:

$$\begin{aligned}f(x,y) &= \mathcal{F}^{-1}(\mathsf{F}(\omega_x, \omega_y))\\ &= \frac{1}{(2\pi)^2 MN}\sum_{\omega_x=0}^{M-1}\sum_{\omega_y=0}^{N-1} \mathsf{F}(\omega_x, \omega_y) e^{i(\frac{\omega_x x}{M} + \frac{\omega_y y}{N})} \qquad (2.6)\end{aligned}$$

where $x = 0, 1, 2, \ldots M-1, y = 0, 1, 2, \ldots N-1$.

In most implementations of the discrete Fourier transform, or its algorithmically-faster version the Fast Fourier transform (FFT), the origin in the spatial-frequency domain is shifted from $(0, 0)$ to $(\frac{M}{2}, \frac{N}{2})$, so that the $(\frac{M}{2}, \frac{N}{2})$ component represents the zero frequency and the bounds on ω_x and ω_y become $-\frac{M}{2} \leq \omega_x \leq \frac{M}{2}$ and $-\frac{N}{2} \leq \omega_x \leq \frac{N}{2}$ (see Figure 2.2). Close inspection of the bounds of ω_x and ω_y reveals that there are $M+1$ and $N+1$ samples in the x and y spatial frequency directions rather than M and N, as we would have expected. It turns out that values $\pm\frac{M}{2}$ are equal, as are those of $\pm\frac{N}{2}$, so there are really $M \times N$ distinct samples in the frequency domain, just as there are in the spatial domain [275].

Note that the sampling periods Δ_x and Δ_y in the spatial domain and Δ_{ω_x} and Δ_{ω_y} in the spatial-frequency domain are related as follows:

$$\frac{\Delta_{\omega_x}}{2\pi} = \frac{1}{M\Delta_x} \tag{2.7}$$

$$\frac{\Delta_{\omega_y}}{2\pi} = \frac{1}{N\Delta_y} \tag{2.8}$$

Hence, if we have an image of dimensions $M \times N$ pixels, with $\Delta_x = \Delta_y = 1$ pixel, then $\Delta_{\omega_x} = \frac{2\pi}{M}$ radians/pixel and $\Delta_{\omega_y} = \frac{2\pi}{N}$ radians/pixel. The maximum spatial frequencies which can be represented by the DFT in the x direction, $\omega_x^{\max}$, occurs at $\omega_x = \pm\frac{M}{2}$:

$$\omega_x^{\max} = \pm\Delta_{\omega_x}\frac{M}{2}$$

From equation 2.7, we have therefore:

$$\omega_x^{\max} = \pm\pi \text{ radians/pixel}$$

Similarly,

$$\omega_y^{\max} = \pm\Delta_{\omega_y}\frac{N}{2} = \pm\pi \text{ radians/pixel}$$

Both of these maximum spatial frequencies correspond to a spatial sinusoid with period 2 pixels.[1] Thus, the sampling period is half that of the maximum representable spatial frequency or, equivalently, the sampling frequency is twice that of the maximum spatial frequency; this is the so-called *Nyquist frequency.*

For the sake of simplicity and brevity, all theoretical discussions in this book will use the notation of the continuous Fourier transform, neglecting the inclusion of terms in M or N on the explicit understanding that implementation of the theory requires a DFT or FFT and, consequently, when performing any computation, *e.g.* in computing velocities in unit of pixel/frame using spatial frequency values, we do need to include the terms in N and M as set out above.

[1] The period T is related to angular spatial frequency ω by $T = \frac{2\pi}{\omega}$; if $\omega = \pi$ radians/pixel, then $T = 2$ pixels.

1.3 PHASOR NOTATION

As we noted already, $\mathsf{F}(\omega_x, \omega_y)$ is defined on a complex domain with $\mathsf{F}(\omega_x, \omega_y) = \mathsf{A}(\omega_x, \omega_y) + i\mathsf{B}(\omega_x, \omega_y)$ where

$$\begin{aligned}\mathsf{A}(\omega_x, \omega_y) &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x', y')\cos(\omega_x x' + \omega_y y')\mathrm{d}x'\mathrm{d}y' \\ \mathsf{B}(\omega_x, \omega_y) &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x', y')\sin(\omega_x x' + \omega_y y')\mathrm{d}x'\mathrm{d}y'\end{aligned}$$

That is, each value of a Fourier transformed image $\mathsf{F}(\omega_x, \omega_y)$ is a complex number. We know that it is sometimes more convenient to think about complex numbers as vectors having a magnitude and a phase angle. In this instance, $\mathsf{F}(\omega_x, \omega_y)$ may be expressed in terms of its magnitude and phase:

$$\mathsf{F}(\omega_x, \omega_y) = |\mathsf{F}(\omega_x, \omega_y)|\, e^{i\phi(\omega_x, \omega_y)}$$

where:

$$\begin{aligned}|\mathsf{F}(\omega_x, \omega_y)| &= \sqrt{\mathsf{A}^2(\omega_x, \omega_y) + \mathsf{B}^2(\omega_x, \omega_y)} \\ \phi(\omega_x, \omega_y) &= \arctan\frac{\mathsf{B}(\omega_x, \omega_y)}{\mathsf{A}(\omega_x, \omega_y)}\end{aligned}$$

$|\mathsf{F}(\omega_x, \omega_y)|$ is the real-valued *Fourier spectrum* or *amplitude spectrum* and $\phi(\omega_x, \omega_y)$ is the real-valued *phase spectrum.* This exponential form which, on the face of it looks more complicated, is very useful as it allows us to think of each Fourier component, *i.e.* each complex Fourier value at a given spatial frequency (ω_x, ω_y), as a vector having magnitude $|\mathsf{F}(\omega_x, \omega_y)|$ and phase $\phi(\omega_x, \omega_y)$ (see Figure 2.3).

Thus, we can think about the Fourier transform of an image in two ways:

1. as a 2-D distribution of complex numbers;

2. as a 2-D distribution of vectors.

Conceiving the Fourier transform as a 2-D distribution of vectors — we will call them *phasors* from now on since this is what they are commonly referred to in the areas of electrical and computer engineering — is particularly appealing because it allows us to visualize a 2-D Fourier transform quite easily (see Figure 2.4) and, as we will see later in this chapter and throughout the remainder of the book, it facilitates a very intuititive understanding of the Fourier transform, its properties, and the behaviour of the Fourier transform of image sequences.

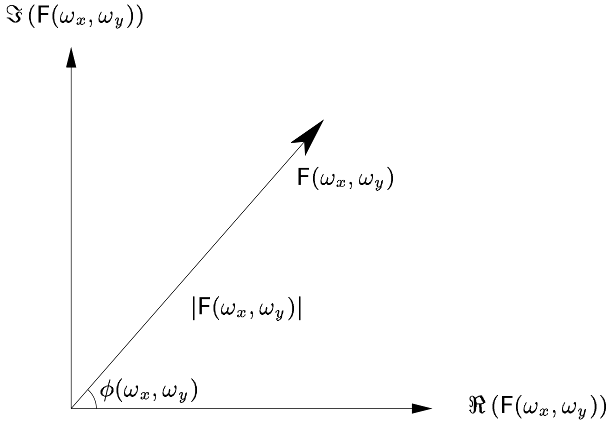

$\Im(\mathsf{F}(\omega_x, \omega_y))$
$\mathsf{F}(\omega_x, \omega_y)$
$|\mathsf{F}(\omega_x, \omega_y)|$
$\phi(\omega_x, \omega_y)$
$\Re(\mathsf{F}(\omega_x, \omega_y))$

Figure 2.3. A complex Fourier component with real and imaginary parts $\Re(\mathsf{F}(\omega_x, \omega_y))$ and $\Im(\mathsf{F}(\omega_x, \omega_y))$, respectively, viewed as a phasor $\mathsf{F}(\omega_x, \omega_y)$ with magnitude $|\mathsf{F}(\omega_x, \omega_y)|$ and phase $\phi(\omega_x, \omega_y)$.

1.4 PROPERTIES OF THE FOURIER TRANSFORM

The Fourier transform has many useful properties. However, we will not cite them all here and we will concentrate instead on the two properties which are required for the theoretical developments which will ensue in the following chapters. Readers who are interested in a more exhaustive treatment of the Fourier transform, its properties and applications might like to consult one of the many excellent texts which deal fully with all of the aspects of the Fourier transform. For example, Gonzalez and Woods [123] provide a thorough treatment of the Fourier transform in digital image processing; Poularikas [272] gives a very rigorous and exhaustive discussion of the 1-, 2-, and n-dimensional Fourier transforms and their applications, while Press *et al.* [275] provides one of the most lucid explanations of the Fourier transform, the Discrete Fourier transform, and its more efficient algorithmic counterpart, the Fast Fourier transform.

1.4.1 THE FOURIER SHIFT PROPERTY

We begin with the property of the Fourier transform which forms the cornerstone of the techniques described in this book: the shift property.

The shift property of the Fourier transform states that the Fourier transform of a shifted function $f(x - x_0, y - y_0)$, *i.e.* a function $f(x, y)$

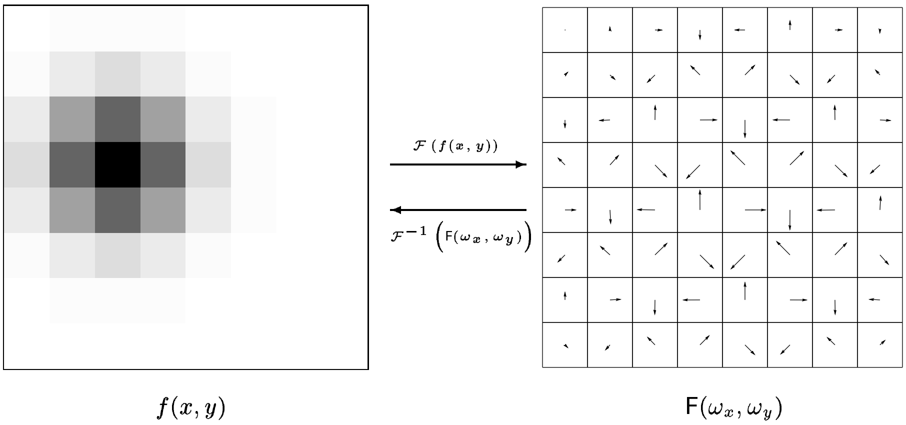

$\mathcal{F}\left(f(x,y)\right)$
$\mathcal{F}^{-1}\left(\mathsf{F}(\omega_x,\omega_y)\right)$
$f(x,y)$
$\mathsf{F}(\omega_x,\omega_y)$

Figure 2.4. Phasor-representation of the 2-D Discrete Fourier transform $\mathcal{F}\left(f(x,y)\right)$ of a discrete image $f(x,y)$. The image $f(x,y)$ is a Gaussian distribution of intensities with a standard deviation $\sigma = 1.0$ pixels; in this case, black signifies a high intensity. Each Fourier component, $\mathsf{F}(\omega_x,\omega_y)$, is represented by a phasor of magnitude $|\mathsf{F}(\omega_x,\omega_y)|$ and phase $\phi(\omega_x,\omega_y)$; to aid the visualization of the phasors of higher spatial frequency, the depicted phasor magnitude is actually $log\left(|\mathsf{F}(\omega_x,\omega_y)|\right)$ rather than $|\mathsf{F}(\omega_x,\omega_y)|$. Note the coarse quantization resolution of the phase of each phasor. This is due to the fact that the resolution of the intensity image is just 8 × 8 pixels and, from equations (2.7) and (2.8), the sampling period in the Fourier domain, Δ_ω, is $\frac{2\pi}{8} = \frac{\pi}{4}$ radians/pixel; this resolution increases as the resolution of $f(x,y)$ increases.

shifted so that is origin is at (x_0, y_0), is given by:

$$\begin{aligned} \mathcal{F}\left(f(x-x_0,y-y_0)\right) &= \mathcal{F}\left(f(x,y)\right)e^{-i(\omega_x x_0+\omega_y y_0)} \\ &= \mathsf{F}(\omega_x,\omega_y)e^{-i(\omega_x x_0+\omega_y y_0)} \qquad (2.9) \end{aligned}$$

That is, a spatial shift of a signal produces only a frequency-dependent phase change, *i.e.* $e^{-i(\omega_x x_0+\omega_y y_0)}$, in the Fourier domain.

The equivalent property for the discrete Fourier transform is

$$\mathcal{F}\left(f(x-x_0,y-y_0)\right) = \mathsf{F}(\omega_x,\omega_y)e^{-i(\frac{\omega_x x_0}{N}+\frac{\omega_y y_0}{N})}$$

assuming $f(x,y)$ is an $N \times N$ pixel image.

We can get a very intuitive understanding of this property if we view it from a phasor perspective. If we write the Fourier transform in exponential or phasor notation so that we have

$$\begin{aligned} \mathcal{F}\left(f(x-x_0,y-y_0)\right) &= |\mathsf{F}(\omega_x,\omega_y)|\, e^{i\phi(\omega_x,\omega_y)}e^{-i(\omega_x x_0+\omega_y y_0)} \\ &= |\mathsf{F}(\omega_x,\omega_y)|\, e^{i(\phi(\omega_x,\omega_y)-(\omega_x x_0+\omega_y y_0))} \qquad (2.10) \end{aligned}$$

we see that the shift by (x_0, y_0) doesn't have any influence on the magnitude of the Fourier components $|\mathsf{F}(\omega_x, \omega_y)|$ but it changes the phase $\phi(\omega_x, \omega_y)$ by an angle $-(\omega_x x_0 + \omega_y y_0)$. Thus, the shift property simply states that each phasor, $\mathsf{F}(\omega_x, \omega_y)$, is phase-shifted — rotated — by an angle given by $(\omega_x x_0 + \omega_y y_0)$, *i.e.* the exponent in the term $e^{-i(\omega_x x_0 + \omega_y y_0)}$. That is, the angle of rotation depends on both the shift (x_0, y_0) — as you would expect — and on the spatial frequency (ω_x, ω_y). Figures 2.5 and 2.6 illustrate this rotation by showing the phasors of a sample image and the same image shifted by one pixel to the right.

Let's take this one step further. Let's assume the image is not just shifted by a constant amount (x_0, y_0) but that it is translating with a constant velocity (v_x, v_y). The shift in the image over some time increment δt is given by $(v_x \delta t, v_y \delta t)$. The shift at any discrete time $t = n\delta t$ is $(v_x n\delta t, v_y n\delta t)$. Hence, the Fourier transform at some time $t = n\delta t$ of an image translating with constant velocity (v_x, v_y) is

$$\begin{aligned}\mathcal{F}\left(f(x - v_x n\delta t, y - v_y n\delta t)\right) &= \mathcal{F}\left(f(x, y)\right) e^{-i(\omega_x v_x n\delta t + \omega_y v_x n\delta t)} \\ &= \mathsf{F}(\omega_x, \omega_y) e^{-i(\omega_x v_x n + \omega_y v_y) n\delta t} \\ &= \mathsf{F}(\omega_x, \omega_y)\left(e^{-i(\omega_x v_x + \omega_y v_y)\delta t}\right)^n \quad (2.11)\end{aligned}$$

In other words, as the image translates with velocity (v_x, v_y) pixels per unit time, the phasors rotate with angular velocity $(\omega_x v_x + \omega_y v_y)$ radians per unit time. Using subscripts to explicitly denote the time at which the Fourier transform is effected, *i.e.* letting

$$\mathsf{F}_{t_0}(\omega_x, \omega_y) = \mathcal{F}\left(f(x, y)\right)$$

and

$$\mathsf{F}_{t_n}(\omega_x, \omega_y) = \mathcal{F}\left(f(x - v_x n\delta t, y - v_y n\delta t)\right)$$

from equation 2.11, we have therefore, in general, at time t_n(for a constant velocity):

$$\mathsf{F}_{t_n}(\omega_x, \omega_y) = \mathsf{F}_{t_0}(\omega_x, \omega_y)\left(e^{-i(\omega_x v_x \delta t + \omega_y v_y \delta t)}\right)^n \quad (2.12)$$

Figures 2.7 and 2.8 show two examples of the effect of image translation on the Fourier phasors. In the first case, the images are those of the 8 pixel Gaussian function shown in the previous examples, together with the phasors at time $t = 0, 1, 2$ and 3, *i.e* $\mathcal{F}\left(f(x - v_x n\delta t, y - v_y n\delta t)\right)$ where $\delta t = 1$ and $n = 0, 1, 2, 3$. The velocity of the image is $(1, 0)$ pixels per unit time. Note that, because of the low resolution of the image (8×8 pixels), the quantization resolution of the angular velocity of the phasors is very coarse ($\frac{2\pi}{8} = \frac{\pi}{4}$ radians/frame). On the other hand, Figure

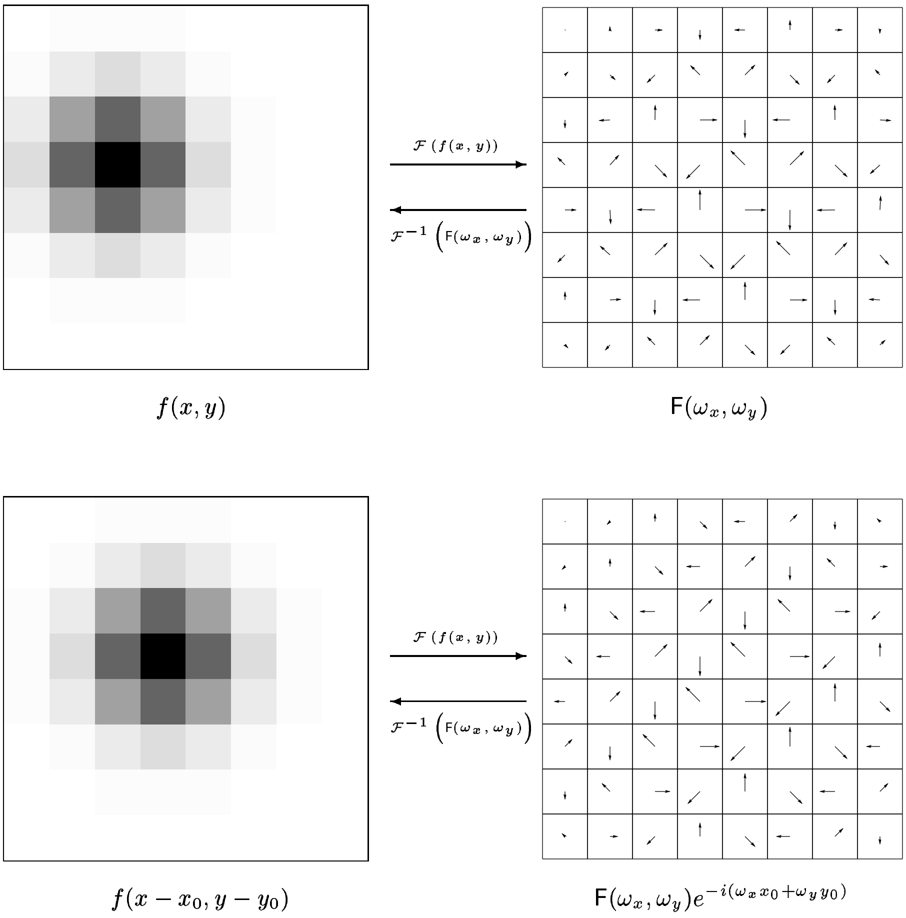

$\mathcal{F}\,(f(x,y))$
$\mathcal{F}^{-1}\left(\mathsf{F}(\omega_x,\omega_y)\right)$
$f(x,y)$
$\mathsf{F}(\omega_x,\omega_y)$
$\mathcal{F}\,(f(x,y))$
$\mathcal{F}^{-1}\left(\mathsf{F}(\omega_x,\omega_y)\right)$
$f(x-x_0,y-y_0)$
$\mathsf{F}(\omega_x,\omega_y)e^{-i(\omega_x x_0+\omega_y y_0)}$

Figure 2.5. The shift property of the Fourier transform states that if a function $f(x,y)$ is shifted or translated by (x_0,y_0), then each Fourier phasor is rotated by an angle $(\omega_x x_0+\omega_y y_0)$ radians: $\mathcal{F}\,(f(x-x_0,y-y_0)) = \mathsf{F}(\omega_x,\omega_y)e^{-i(\omega_x x_0+\omega_y y_0)}$. Note that there is no other effect: pure translation only affects the phase and has no effect on the magnitude of the constituent phasors. In the example shown above, $(x_0,y_0) = (1,0)$ pixels and, hence, there is no phase change in the (vertical) ω_y direction but the phase difference increases (by $\frac{\pi}{4}$ radians) for each sample in the (horizontal) ω_x direction; this is more evident in Figure 2.6 which shows both phasor respresentations superimposed.

2.8 shows a 256×256 pixel image translating with a velocity of $v_x = 3$ and $v_y = -3$ pixels per frame with the superimposed phasors depicting an 8×8 region in the Fourier domain, centered at pixel $(128,128)$, *i.e.*

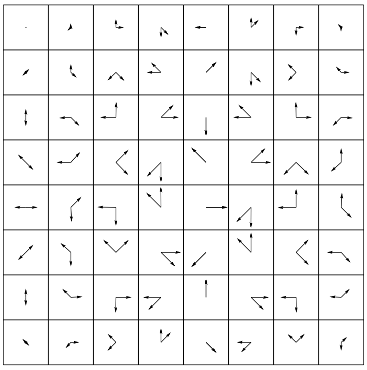


Figure 2.6. Superposition of the phasors of $\mathcal{F}(f(x,y))$ and $\mathcal{F}(f(x-x_0,y-y_0))$; refer to Figure 2.5 which shows both phasor respresentations and the corresponding images $f(x,y)$ and $f(x-x_0,y-y_0)$.

centred at the zero-frequency component. In this case, the quantization resolution of the angular velocity is much smaller and the phase change is more gradual in both the ω_x and ω_y directions.

1.4.2 LINEARITY

In addition to the Fourier shift property, we also need to be aware that the Fourier transform is a linear operator: that is, the transform of the sum of two functions is equal to the sum of the transform of the two functions:

$$\mathcal{F}\left(f(x,y)+g(x,y)\right)=\mathcal{F}\left(f(x,y)\right)+\mathcal{F}\left(g(x,y)\right) \tag{2.13}$$

i.e. the Fourier transform distributes over addition. Also, the transform of a constant times a function is equal to that constant times the transform of the function:

$$\mathcal{F}\left(cf(x,y)\right)=c\mathcal{F}\left(f(x,y)\right) \tag{2.14}$$

We now turn out attention to the second transform which we will use in subsequent chapters.

2. THE HOUGH TRANSFORM

The Hough transform [145] is used to identify the presence of particular patterns of data, usually shapes or curves, within some given data-set. The idea is to characterize the curve or shape by some appopriate equation and then to reverse the *rôles* of the independent variables, which

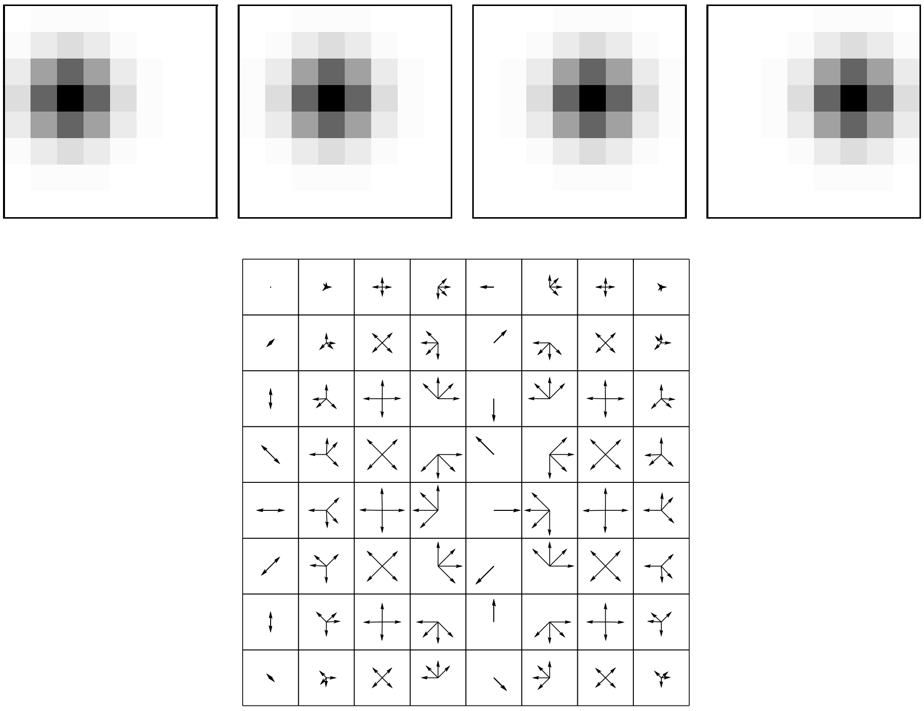


Figure 2.7. Fourier transform of an image $f(x,y)$ translating with constant velocity (v_x, v_y): $\mathcal{F}\left(f(x - v_x n\delta t, y - v_y n\delta t)\right)|_{n=0,1,2,3} = \mathsf{F}(\omega_x, \omega_y)e^{-i(\omega_x v_x n + \omega_y v_y)n\delta t}\Big|_{n=0,1,2,3}$ Each phasor rotates with a frequency-dependent and velocity-dependent angular velocity $(\omega_x v_x + \omega_y v_y)$. The velocity of the image is $(1,0)$ pixels per unit time. Note that, because of the low resolution of the image (8×8 pixels), the quantization resolution of the angular velocity of the phasors is very coarse ($\frac{2\pi}{8} = \frac{\pi}{4}$ radians/frame) and, consequently, the high ω_x spatial frequencies wrap around 2π.

we normally don't know or specify, and the coeffients or parameters in the equation, which we normally do know since they specify the curve in question. By reversing the *rôles*, we are saying we now do know or specify the independent variables, *e.g.* coordinates of sample points, and we don't know (but would like to find out) the parameters for a particular curve (corresponding to the sample points).

Specifically, equations which describe a curve accomplish the description by defining a spatial relationship between the independent variables. Often this relationship takes the form of multiplicative coefficients and exponents. For example, a line can be described by the well-known equation $y = mx + c$, and a circle of radius r and centre (x_0, y_0) is described

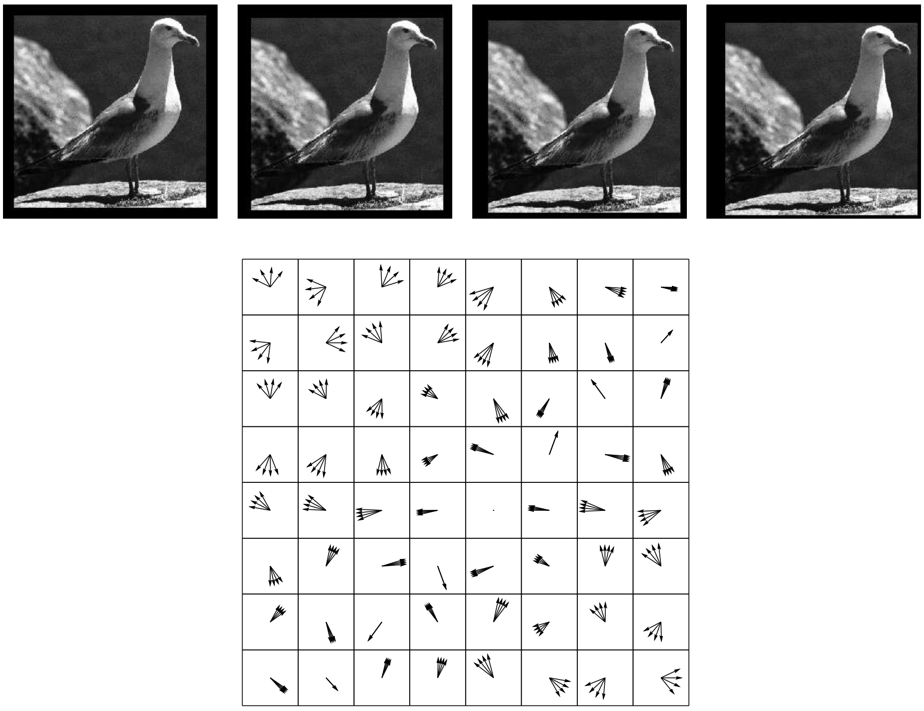
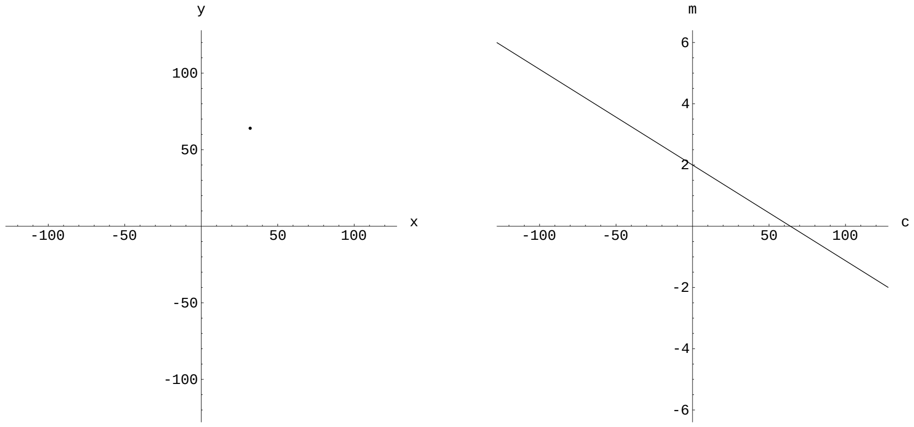


Figure 2.8. Fourier transform of a 256 × 256 pixel image $f(x,y)$ translating with constant velocity (v_x, v_y): $\mathcal{F}(f(x - v_x n\delta t, y - v_y n\delta t))|_{n=0,1,2,3} = \mathsf{F}(\omega_x, \omega_y)e^{-i(\omega_x v_x n + \omega_y v_y)n\delta t}\Big|_{n=0,1,2,3}$ Each phasor rotates with a frequency-dependent and velocity-dependent angular velocity $(\omega_x v_x + \omega_y v_y)$. The velocity of the image is $(3, -3)$ pixels per unit time. The superimposed phasors depict an 8 × 8 region in the Fourier domain, centred at pixel $(128, 128)$, *i.e.* centred at the zero-frequency component. In this case, because of the higher resolution of the image compared with that shown in Figure 2.7 (8 × 8 pixels), the quantization resolution of the angular velocity of the phasors is much smaller ($\frac{2\pi}{256}$ radians/frame) and, consequently, the phase differences of any given phasor are much smaller. In this case, it is easier to discern the linear relationship between angular velocity of rotation and spatial velocity & spatial frequency. Note the set of phasors which have a zero angular velocity. These are the components for which $\omega_x v_x = -\omega_y v_y$ (*i.e.* $\omega_x v_x + \omega_y v_y = 0$.) The magnitude of the zero-frequency component has been set to zero since it would otherwise have swamped the relative magnitude of the other components.

by the equation $(x - x_0)^2 + (y - y_0)^2 = r^2$. Both of these equations capture the definition of a particular curve by defining a relationship between a set of independent variables $\mathbf{v}$ (*e.g.* x and y) and a set of parameters $\mathbf{p}$, the value of which define a particular curve (*e.g.* m and c, or x_0, y_0, and r). The independent variables correspond to the domain



$x - y$ space

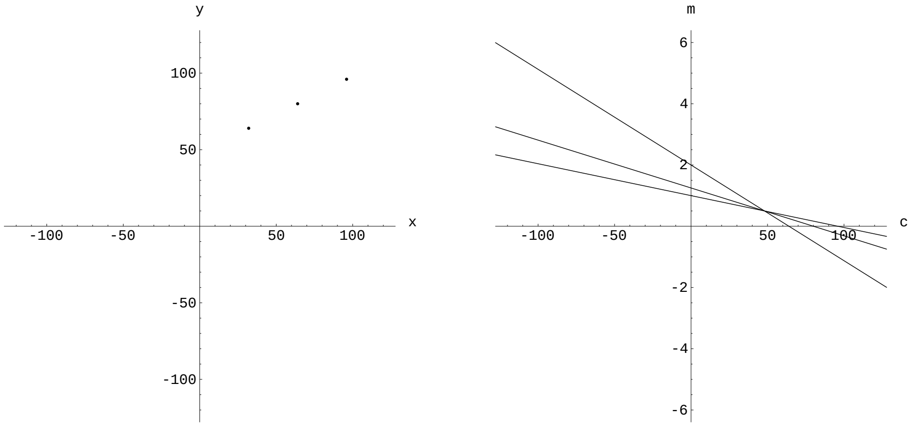
$m - c$ space

Figure 2.9. Hough transform equation $y = mx + c$: a given point (x_i, y_i) corresponds to a line in the $m - c$ Hough space. This line in $m - c$ space identifies the values of m and c of all lines in $x - y$ space containing the point (x_i, y_i).

in which the curve is defined, *e.g.* the $x - y$ plane, and the parameters distinguish one particular curve in that domain from the infinite set of possible curves of the type described by the equation in question; that is, a given set of parameter values defines a particular curve in the domain of the independent variables.

Instead of defining a *specific* curve by choosing or fixing a set of parameter values, the Hough transform reverses the *rôle* of the independent variables $\mathbf{v}$ and the parameters $\mathbf{p}$, making the parameter variables the independent variables and the independent variables the parameters. So, the equation $y = mx + c$, viewed as a Hough transform, instead of identifying the set of all points (x, y) which lie on a unique line defined by m and c, identifies the set of all possible parameters m and c which define the set of all lines containing (x_i, y_i); see Figure 2.9. Thus, an equation comprising parameters and independent variables defines a *parameter*-specific curve in the domain of the independent varibles or, alternatively, it defines an *independent-variable*-specific curve in the domain of the parameter variables: this domain is the Hough transform space.

Just as one set of specific parameters defines a unique curve in the domain of the independent variables, so too, viewing the same equation as a Hough transform, one set of specific independent variable values defines a unique curve in the domain of the parameter variables: the Hough



$x - y$ space

$m - c$ space

Figure 2.10. Hough transform equation $y = mx + c$: the three given points (x_i, y_i) correspond to three lines in the $m - c$ Hough space, the intersection of which give the values of m and c which define the line containing (x_i, y_i).

transform space. All we have done is to swap independent variables and parameters: the old independent variables become the 'parameters' in the Hough transform domain and the old parameter variables become the new 'independent variables' in the Hough domain. Thus, the Hough transform domain corresponds to the parameter variables in the Hough transform equation (*e.g.* m and c, and x_0, y_0, and r in the two preceding examples).

So where does all of this get us? As it stands, not very far. However, consider a situation where we have not one but two sets of specific values of the independent variables, *e.g.* $\{(x, y)\} = \{(32, 64), (96, 96)\}$ say; each defines a different curve in the Hough transform space. These curves will intersect and the point of intersection defines the parameter values of a curve in the original $x - y$ domain which contains both the points $(32, 64)$ and $(96, 96)$ — see Figure 2.10. By extension, n specific points $(x_i, y_i), i = 1 \dots n$ generate n curves in the Hough domain, the intersection of which gives the parameter values of the curve in $x{-}y$ space which contains (x_i, y_i). Thus, if we have a closed-form equation defining a curve and a set of points (x_i, y_i), the Hough transform technique can be used to identify the parameter values of that curve which contains those points by computing the Hough transform curve for each point (x_i, y_i) and then identifying the point or points (p_i, q_i) in the Hough space where the curves intersect.

From an algorithmic point of view, we can do this very simply by representing the sampled Hough transform space by an $k-$dimensional array, where k is the number of parameters (*e.g.* $k = 2$ in the case of a line: m and c; $k = 3$ in the case of a circle: x_0, y_0, and r). For a given value (x_i, y_i), we add 1 to each array element if the corresponding parameter value satisfies the Hough transform equation. If we repeat this process for all known points (x_i, y_i), the Hough transform array – which is referred to as an accumulator — will contain a maximal value at exactly the element where the curves intersect.

If the points (x_i, y_i) don't all lie exactly on the curve defining the Hough transform, then the maximum will occur over some local neighbourhood.

The Hough transform can be extended to deal with surfaces in n-dimensional space and isn't restricted to curves in 2-D $x - y$ space and k-D Hough transform space [372]. However, the computational complexity inherent in solving all of the Hough transform parameter values for each known independent variable values makes the technique virtually intractable if the dimensionality increases beyond three or four. Happily, in the following, we require only a 2-D Hough transform.

Note that we chose the line equation $y = mx + c$ to explain the Hough transform for illustrative purposes only (mainly because it is a very familiar equation) and, in real situations where we wished to use the Hough transform to identify co-linear points, we would not choose this particular form of equation because one of the parameters, the slope m, is not bounded; instead, we would choose an alternative formulation which parameterizes a line by its normal distance r from the origin and the angle that the normal makes with one of the axes [100, 366].

In summary, then, the Hough transform is a type of universal transform which relates any two domains linked by some closed-form equation. In the event that such an explicit closed-form equation is not available, a trainable look-up table can substituted for the parametric equation and the transform is then referred to as the *Generalized Hough Transform* [100].

Chapter 3

MONOCULAR VISION — SEGMENTATION IN ADDITIVE IMAGES

1. OVERVIEW

This chapter begins the presentation of the core material in the book. Here, we develop and apply a technique to separate additive images (*i.e.* images formed by the addition of a number of component images) into their additive components. We will assume that each component image is translating with some unknown and possibly zero velocity. Specifically, given a time sequence of a number of images, each formed by the addition of two distinct but perhaps identical images — distinct in the sense that the component images are moving with a different velocity — we will show how one can recover the two component images. In the next chapter, we will extend these results to the case where the images are not formed by additive superposition but by occluding superposition, *i.e.* the images comprise two objects or scenes, one occluding the other.

Since both of these applications are variants of the so-called *segmentation problem*, we will begin with a brief look at the general process of segmentation and place the technique to be developed in context.

2. THE SEGMENTATION PROBLEM

The term segmentation refers to a process of distinguishing and labelling distinct regions in an image such that each region has a corresponding physical counterpart in the visual scene. This counterpart might be a complete object (*e.g.* a road sign, a seagull, a chromosome, an integrated circuit) or a well-bounded part of an object (*e.g.* the letters on the road sign, the eye of the seagull, an arm of the chromosome, or the leads on the integrated circuit). The key point is that each segmented region should correspond on a one-to-one basis to some physically-meaningful region in the visual environment.

Traditional segmentation techniques typically exploit one of two broad approaches. These are (a) boundary detection, which depends on the detection of spatial discontinuities in the reflectance function of the scene or in some other image property (using first- or second-order gradient techniques) and their aggregation into contour-based object descriptions, and (b) region-growing, which depends on the identification of local regions that satisfy a regional similarity predicate of some image property. In dealing with three-dimensional scenes, the most commonly-used image properties for image segmentation are the local optical flow, the local stereo disparity, and the local depth or range. In the case of optical flow, one of the most powerful and frequently-used formulations of a similarity predicate is that the local flow field correspond to some parametric model of a moving surface, typically an affine or a quadratic model of the motion of a planar surface. A non-exhaustive sample of representative motion/stereo segmentation algorithms can be found in Section 6. at the end of the next chapter.

The goal of this book is to develop an alternative formulation of the motion segmentation problem using the Fourier analysis of spatio-temporal image sequences. In the following, we will present a new approach to the segmentation of both additive and occluding images. This is effected by processing the resultant Fourier phasors derived from the FFT of the composite image and by resolving each of them (*i.e.* at every spatial frequency) into the individual Fourier components corresponding to each object. Four distinct scenarios are addressed:

1. Monocular temporal sequences of additive scenes (Chapter 3);

2. Monocular temporal sequences of occluding scenes (Chapter 4);

3. Articulated binocular (spatio-temporal) sequences of static or dynamic scenes (chapter 5); and

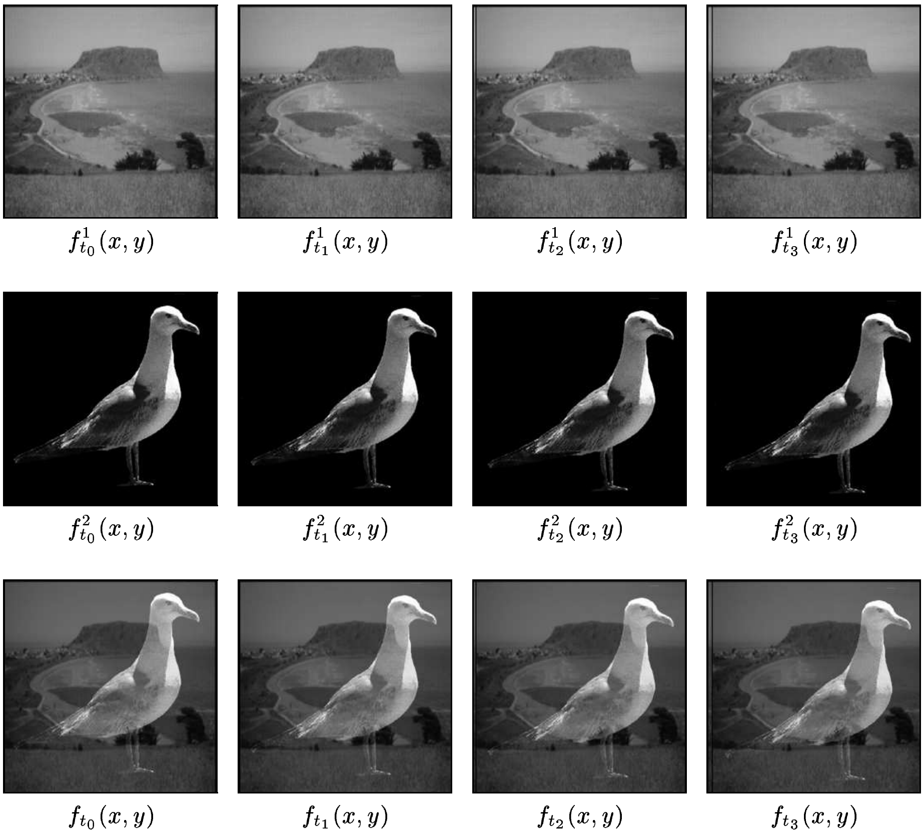
4. Binocular images of static scenes (chapter 6).

The significance of the approach is that the visual complexity of the foreground and background is relatively inconsequential since the segmentation is effected independently for each individual spatial frequency in the Fourier domain. It does not require the computation of an optical flow field and it can deal explicitly with the situation where there is more than one motion in a local region. Consequently, it can cope well with all of the elementary local motion configurations identified by Bergen *et al.* [39]: motion boundaries, transparent surfaces in motion, 'picket-fence' motion (where a moving background is visible through small gaps in a foreground which exhibits a different motion), masking (of a small moving pattern by a larger pattern), and the two-component aperture effect (the problem encountered when, in an effort to overcome the aperture problem inherent in all motion estimation techniques, the aperture is enlarged thus revealing a second independent motion).

As we noted in chapter 1, the approach assumes fronto-parallel translatory motion of objects in the scene; it does not cater explicitly for scaling or rotation due either to camera motion or zoom or due to the independent motion of the objects. Whilst this is certainly not the most general situation, it is nonetheless an important scenario, especially if we ensure that the successive images in an acquired sequence are sampled with a small-enough sampling period. In situations where this assumption is not valid, exactly because the motion field *within the field of view* is too complex, the usual approach in such situations is to deploy a more sophisticated model, for example, such as that proposed by Irani and Anandan [157]. An alternative strategy is to retain the simpler model, and to re-constrain the situation by appropriately narrowing the field of view and adjusting the focus of attention. This is particularly appropriate when conventional approaches cannot deal with the visual complexity (or lack of, as the case may be) of the objects to be segmented. Of course, if the object to be segmented is larger than the field of view, then segementation will have to proceed by processing appropriate sub-views and assembling these parts together.

3. THE FOURIER MODEL OF SEGMENTATION

We will begin by developing a model for the separation or decoupling of m additive images from a temporal sequence of the sum of these images, where the images are translating with distinct and unique velocities (including zero velocity). A solution is presented for the case where $m = 2$. In the next chapter, this model is then adapted to embrace situations where the images are not additive but are, instead, formed by the superposition of an occluding object or objects on an occluded background. That is, the approach is generalized to effect a model-free segmentation of objects undergoing translatory fronto-parallel motion in dynamic image sequences. Note that object velocities of one pixel per frame are sufficient to guarantee segmentation.



$f^1_{t_0}(x,y)$ $f^1_{t_1}(x,y)$ $f^1_{t_2}(x,y)$ $f^1_{t_3}(x,y)$

$f^2_{t_0}(x,y)$ $f^2_{t_1}(x,y)$ $f^2_{t_2}(x,y)$ $f^2_{t_3}(x,y)$

$f_{t_0}(x,y)$ $f_{t_1}(x,y)$ $f_{t_2}(x,y)$ $f_{t_3}(x,y)$

Figure 3.1. A composite image sequence $f_{t_j}(x,y) = \sum_{i=1}^{m} f^i_{t_j}(x,y)$. In this case, m, the number of images, is equal to 2. The goal is to compute $f^1_{t_j}(x,y)$ and $f^2_{t_j}(x,y)$, knowing only $f_{t_j}(x,y)$.

3.1 THE ADDITIVE MODEL FOR TEMPORAL SEQUENCES

In this section, we show that it is possible to separate (or recover) m additive images, given only a time sequence of the sum of these images and given the assumption that the images are translating, each with a distinct and unique velocity (which may be zero, in the case of a static foreground or background). That is, given a composite image $f_{t_j}(x,y)$ at time t_j in a temporally-ordered image sequence (refer also to Figure 3.1),

$$f_{t_j}(x,y) = \sum_{i=1}^{m} f^i_{t_j}(x,y) \tag{3.1}$$

where $f^i_{t_j}(x,y)$ is the unknown i^{th} additive component image at time t_j, and assuming

$$f^i_{t_j}(x,y) = f^i_{t_0}(x - v^i_x j\delta t, y - v^i_y j\delta t) \tag{3.2}$$

where (v^i_x, v^i_y) is the spatial velocity of the i^{th} component image and δt is the incremental time interval, it is possible to recover, or compute, each individual image $f^i_{t_0}(x,y), \forall i$.

This recovery is accomplished by computing the Fourier transform $\mathsf{F}_{t_j}(\omega_x, \omega_y)$ of each image in the image sequence

$$\mathsf{F}_{t_j}(\omega_x,\omega_y) = \mathcal{F}\left(f_{t_j}(x,y)\right) \tag{3.3}$$

and then by decoupling the resultant (composite) Fourier component at each spatial frequency into the m individual Fourier components $\mathsf{F}^i_{t_0}(\omega_x,\omega_y)$ at time t_0. The required individual images are then computed using the inverse Fourier transform

$$f^i_{t_0}(x,y) = \mathcal{F}^{-1}\left(\mathsf{F}^i_{t_0}(\omega_x,\omega_y)\right) \tag{3.4}$$

The kernel of the problem, then, is to take a resultant Fourier component $\mathsf{F}(\omega_x,\omega_y)$ which is the vector sum of each $\mathsf{F}^i(\omega_x,\omega_y)$ — see Figure 3.2 — and then determine each constituent Fourier component $\mathsf{F}^i(\omega_x,\omega_y)$. That is, given $\mathsf{F}_{t_j}(\omega_x,\omega_y)$,

$$\mathsf{F}_{t_j}(\omega_x,\omega_y) = \sum_{i=1}^{m} \mathsf{F}^i_{t_j}(\omega_x,\omega_y) \tag{3.5}$$

where m is the number of constituent images, we wish to compute $\mathsf{F}^i_{t_0}(\omega_x,\omega_y), \forall i$.

3.2 SOLVING FOR THE COMPONENT PHASORS

Clearly, we require additional information over and above what is embodied in equation 3.5 in order to effect the resolution of components. This information arises from an assumption that the individual images which give rise to the constituent Fourier components are undergoing a constant translation, each with different (possibly zero) velocity. We saw in Chapter 2 that the shift property of the Fourier transform states that the Fourier transform of a shifted function $f(x - v_x\delta t, y - v_y\delta t)$, *i.e.* an image translating with a constant velocity (v_x, v_y), is given by:

$$\begin{aligned}\mathcal{F}\left(f(x - v_x\delta t, y - v_y\delta t)\right) &= \left|\mathsf{F}(\omega_x,\omega_y)\right| e^{i\phi(\omega_x,\omega_y)} e^{-i(\omega_x v_x \delta t + \omega_y v_y \delta t)} \\ &= \mathcal{F}\left(f(x,y)\right) e^{-i(\omega_x v_x \delta t + \omega_y v_y \delta t)}\end{aligned}$$

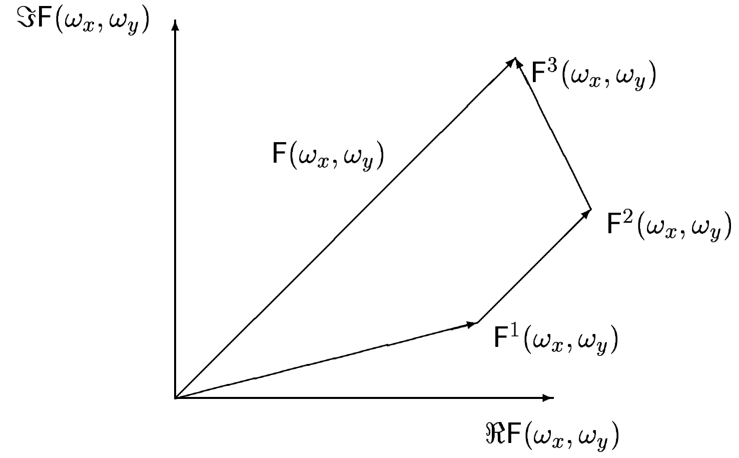

$\Im \mathsf{F}(\omega_x, \omega_y)$
$\mathsf{F}^3(\omega_x, \omega_y)$
$\mathsf{F}(\omega_x, \omega_y)$
$\mathsf{F}^2(\omega_x, \omega_y)$
$\mathsf{F}^1(\omega_x, \omega_y)$
$\Re \mathsf{F}(\omega_x, \omega_y)$

Figure 3.2. The Fourier transform generates $\mathsf{F}(\omega_x, \omega_y)$, the resultant of the sum of the Fourier components $\mathsf{F}^1(\omega_x, \omega_y), \mathsf{F}^2(\omega_x, \omega_y), \mathsf{F}^3(\omega_x, \omega_y)$ of objects 1, 2, and 3, respectively.

That is, a spatial shift of a signal only produces a frequency-dependent phase change in the Fourier domain. The consequence of this is that the Fourier components of $f^i(x, y)$ translating with uniform velocity only undergo a phase change, with no change in magnitude: the vector $\mathsf{F}^i(\omega_x, \omega_y)$ is only rotated by an angle given by the phase change $-(\omega_x v_x^i \delta t + \omega_y v_y^i \delta t)$. Thus,

$$\mathsf{F}^i_{t_1}(\omega_x, \omega_y) \quad = \quad \mathsf{F}^i_{t_0}(\omega_x, \omega_y) e^{-i(\omega_x v_x^i \delta t + \omega_y v_y^i \delta t)}$$

In general, for a constant velocity, at time t_n,

$$\mathsf{F}^i_{t_n}(\omega_x, \omega_y) \quad = \quad \mathsf{F}^i_{t_0}(\omega_x, \omega_y) \left(e^{-i(\omega_x v_x^i \delta t + \omega_y v_y^i \delta t)} \right)^n \tag{3.6}$$

Equation 3.6 (which is equivalent to equation 2.12 given in Chapter 2) indicates that a Fourier component of an individual image i at time t_n is simply the Fourier component at time t_0 rotated through an angle $n\phi$, where ϕ is the phase angle given by $-(\omega_x v_x^i \delta t + \omega_y v_y^i \delta t)$, *i.e.* it is a function of the velocity (v_x^i, v_y^i) of object i.

Rewriting equations 3.5 and 3.6, dropping the (ω_x, ω_y) for the sake of brevity whilst remembering that we are dealing with complex values

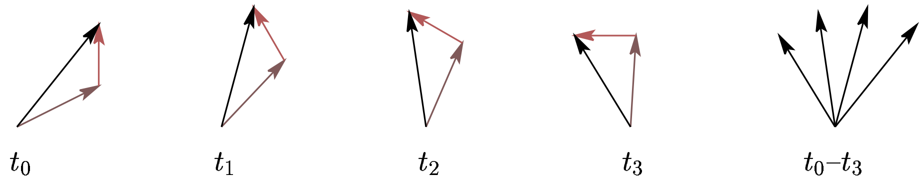


Figure 3.3. Evolution of the Fourier phasor components of two additive images using the resultant phasors. Each vector is shown at time t_0, t_1, t_2, and t_3. Both constituent phasors are rotating with distinct constant angular velocities since the images are undergoing a spatial shift. The angular velocities are dependent on both the spatial frequency and the velocity of image translation. The goal is to compute the constituent phasors at t_0 from the four resultants.

defined on a 2-D domain, and letting $e^{-i(\omega_x v_x^i \delta t + \omega_y v_y^i \delta t)} = \Delta\Phi^i$, a complex variable representing the frequency- and velocity-dependent phase change, we have:

$$\mathsf{F}_{t_j} = \sum_{i=1}^{m} \mathsf{F}_{t_j}^{i} \tag{3.7}$$

$$\mathsf{F}_{t_n}^{i} = \mathsf{F}_{t_0}^{i} \left(\Delta\Phi^i\right)^n \tag{3.8}$$

Combining 3.7 and 3.8, we have

$$\mathsf{F}_{t_j} = \sum_{i=1}^{m} \mathsf{F}_{t_0}^{i} \left(\Delta\Phi^i\right)^j \tag{3.9}$$

If we have $i = m$ distinct individual images, equation 3.9 implies that we have $2m$ complex unknowns (*i.e.* $\mathsf{F}_{t_0}^{i}$ and $\Delta\Phi^i$) and, consequently, we can solve for these $2m$ unknowns if we have $2m$ constraints. These constraints are derived from equation 3.9 by making $j = 2m$ observations for F_{t_j} (*i.e.* by using $j = 2m$ composite images in the temporal sequence). That is, for a given spatial frequency (ω_x, ω_y), we observe the Fourier transform F_{t_j} at time $t_0, t_1, \ldots, t_j, \ldots, t_{2m}$ and solve the resultant $2m$ simultaneous equations (of degree $2m-1$) in complex unknowns $\mathsf{F}_{t_0}^{i}$ and $\Delta\Phi^i$, each equation being derived from equation 3.9.

In the simplest non-trivial case, $m = 2$, there are two distinct objects (see Figure 3.3). This is the familiar figure-ground segmentation problem. In this case, we have

$$\mathsf{F}_{t_0} \quad = \quad \mathsf{F}_{t_0}^{1} \left(\Delta\Phi^1\right)^0 + \mathsf{F}_{t_0}^{2} \left(\Delta\Phi^2\right)^0 \tag{3.10}$$

$$
\begin{aligned}
\mathsf{F}_{t_1} &= \mathsf{F}^1_{t_0}\left(\Delta\Phi^1\right)^1 + \mathsf{F}^2_{t_0}\left(\Delta\Phi^2\right)^1 && (3.11)\\
\mathsf{F}_{t_2} &= \mathsf{F}^1_{t_0}\left(\Delta\Phi^1\right)^2 + \mathsf{F}^2_{t_0}\left(\Delta\Phi^2\right)^2 && (3.12)\\
\mathsf{F}_{t_3} &= \mathsf{F}^1_{t_0}\left(\Delta\Phi^1\right)^3 + \mathsf{F}^2_{t_0}\left(\Delta\Phi^2\right)^3 && (3.13)
\end{aligned}
$$

This set of four simultaneous equations has a closed-form solution:

$$
\begin{aligned}
\Delta\Phi^1 &= \frac{-b+\sqrt{z}}{2a} && (3.14)\\
\Delta\Phi^2 &= \frac{-b-\sqrt{z}}{2a} && (3.15)\\
\mathsf{F}^1_{t_0} &= \frac{\mathsf{F}_{t_0}\Delta\Phi^2 - \mathsf{F}_{t_1}}{(\Delta\Phi^2-\Delta\Phi^1)} && (3.16)\\
\mathsf{F}^2_{t_0} &= \frac{\mathsf{F}_{t_0}\Delta\Phi^1 - \mathsf{F}_{t_1}}{(\Delta\Phi^1-\Delta\Phi^2)} && (3.17)
\end{aligned}
$$

where $a = (\mathsf{F}_{t_1})^2 - \mathsf{F}_{t_0}\mathsf{F}_{t_2}$, $b = \mathsf{F}_{t_0}\mathsf{F}_{t_3} - \mathsf{F}_{t_1}\mathsf{F}_{t_2}$, $c = (\mathsf{F}_{t_2})^2 - \mathsf{F}_{t_1}\mathsf{F}_{t_3}$, and $z = b^2 - 4ac = \alpha + i\beta$ with

$$
\sqrt{z} = \begin{cases}
\pm\left[\left(\frac{\alpha+\sqrt{\alpha^2+\beta^2}}{2}\right)^{\frac{1}{2}} + i\left(\frac{-\alpha+\sqrt{\alpha^2+\beta^2}}{2}\right)^{\frac{1}{2}}\right], & \beta > 0\\
\pm\left[-\left(\frac{\alpha+\sqrt{\alpha^2+\beta^2}}{2}\right)^{\frac{1}{2}} + i\left(\frac{-\alpha+\sqrt{\alpha^2+\beta^2}}{2}\right)^{\frac{1}{2}}\right], & \beta < 0\\
\pm\sqrt{\alpha}, & \beta = 0, \alpha \geq 0\\
\pm i\sqrt{-\alpha}, & \beta = 0, \alpha < 0
\end{cases}
$$

To summarize, given the four *combined* images $f_{t_j}(x,y)$ or, actually, their Fourier transform $\mathsf{F}_{t_j}(\omega_x,\omega_y)$, at times t_0, t_1, t_2, and t_3, we can compute the two component Fourier phasors and their angular velocities (or, equivalently, the velocity of the two component images in the space domain).

3.3 COMBINING PHASORS

There remains, however, one problem. The assignment of $\Delta\Phi^1$ rather than $\Delta\Phi^2$ in equation 3.14 and *vice versa* in equation 3.15 (and, hence the assignment of $\mathsf{F}^1_{t_0}$ and $\mathsf{F}^2_{t_0}$ in equations 3.16 and 3.17) is arbitrary and the alternative assignment is equally valid. In general, there is no necessary consistency in their assignments at different spatial frequencies, *i.e.*

nothing in equations 3.14 - 3.17 guarantees that $\mathsf{F}^1_{t_0}$ belongs to image 1 and that $\mathsf{F}^2_{t_0}$ belongs to image 2. More importantly, if we *assume* that $\mathsf{F}^1_{t_0}$ belongs to image 1 and that $\mathsf{F}^2_{t_0}$ belongs to image 2 at a particular spatial frequency (ω_x, ω_y), there is no guarantee that this assumption will be valid at another spatial frequency. Consequently, once $\mathsf{F}^1_{t_0}(\omega_x, \omega_y)$ and $\mathsf{F}^2_{t_0}(\omega_x, \omega_y)$ have been solved for all spatial frequencies ω_x, ω_y, it is still necessary to sort these sets of $\mathsf{F}^1_{t_0}(\omega_x, \omega_y)$ and $\mathsf{F}^2_{t_0}(\omega_x, \omega_y)$ into two image-specific groups. That is, we actually have at this point two sets of phase changes $\{\Delta\Phi^A\}$ and $\{\Delta\Phi^B\}$ and two corresponding sets of Fourier components $\{\mathsf{F}^A\}$ and $\{\mathsf{F}^B\}$. These sets need to be sorted into two new sets $\{\mathsf{F}^1\}$ and $\{\mathsf{F}^2\}$ corresponding to the two individual component images. We can do this based on the regularity of the incremental phase change $\Delta\Phi$ as a function of frequency (ω_x, ω_y) [370]. Specifically, we have:

$$\begin{aligned}\Delta\Phi(\omega_x, \omega_y) &= e^{-i(v_x \omega_x \delta t + v_y \omega_y \delta t)} \\ &= e^{i\delta\phi(\omega_x, \omega_y)} \qquad (3.18)\end{aligned}$$

For a given image i, (v^i_x, v^i_y) is constant. Thus a given image i will exhibit a phase change $\delta\phi(\omega_x, \omega_y)$:

$$\delta\phi(\omega_x, \omega_y) = -(v^i_x \omega_x \delta t + v^i_y \omega_y \delta t) \qquad (3.19)$$

which will differ for each image i. Assuming $\delta t = 1$, equation 6.1 is the equation of a plane in $\omega_x - \omega_y$ space; the plane contains the origin $(\omega_x, \omega_y) = (0, 0)$ and the direction of its normal depends on the particular values of the velocity components v_x and v_y (see Figure 3.4). Since we require that each image is translating with a different velocity, *i.e.* $(v^i_x, v^i_y) \neq (v^j_x, v^j_y), i \neq j$ — otherwise there would be nothing to distinguish the two images — to sort the components of the two images we simply need to identify the two velocities (v^1_x, v^1_y) and (v^2_x, v^2_y) which will, in turn, allow us to identify the corresponding expected phase change for images 1 and 2. Let these expected phase changes be denoted $\delta\phi^1_e(\omega_x, \omega_y)$ and $\delta\phi^2_e(\omega_x, \omega_y)$, respectively. Then we assign a component $\mathsf{F}^A(\omega_x, \omega_y)$ to image 1, *i.e.* we include it in $\{\mathsf{F}^1\}$, if $\left|\delta\phi^1_e - \delta\phi^A\right| < \left|\delta\phi^2_e - \delta\phi^A\right|$, otherwise we assign it to $\{\mathsf{F}^2\}$; $\mathsf{F}^B(\omega_x, \omega_y)$ is assigned to the other image. Note that $\delta\phi^1_e$ and $\delta\phi^2_e$ are computed *modulo* 2π to ensure that these expected phase changes wrap in the same way as do the solved phase changes $\delta\phi^A$ and $\delta\phi^B$.

It only remains, then, to identify the two velocities (v^1_x, v^1_y) and (v^2_x, v^2_y). We do this using a Hough transform. From equation 3.19 we have:

$$v_y = \frac{1}{\omega_y \delta t}\left(\delta\phi(\omega_x, \omega_y) - \omega_x v_x \delta t\right) \qquad (3.20)$$

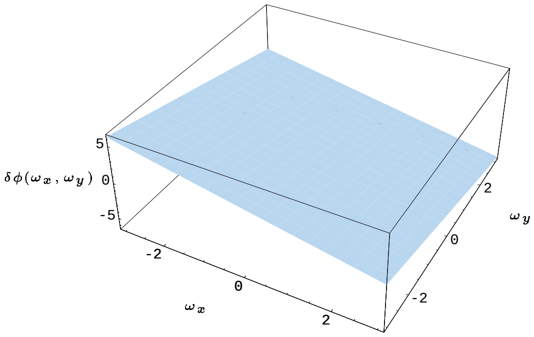


Figure 3.4. The variation of phase change $\delta\phi(\omega_x,\omega_y)$ with spatial frequency (ω_x,ω_y) is defined by the plane equation $\delta\phi(\omega_x,\omega_y) = -(v_x^i\omega_x\delta t + v_y^i\omega_y\delta t)$. The attitude of the plane is determined by the velocity components (v_x,v_y). Note that the plane passes through the origin $(\omega_x = 0, \omega_y = 0)$.

This equation represents a Hough transform in the two variables (v_x, v_y) which we solve for all $\delta\phi(\omega_x,\omega_y), \omega_x, \omega_y$, and v_x. Note that $\delta\phi(\omega_x,\omega_y)$ is computed as follows:

$$\delta\phi(\omega_x,\omega_y) = \arctan(\Im\Delta\Phi(\omega_x,\omega_y), \Re\Delta\Phi(\omega_x,\omega_y)) \tag{3.21}$$

where $\Im\Delta\Phi(\omega_x,\omega_y)$ and $\Re\Delta\Phi(\omega_x,\omega_y)$ are the imaginary and real parts of $\Delta\Phi(\omega_x,\omega_y)$, respectively.

Local maxima in this Hough transform space represent the velocities which are exhibited by the frequency components. In this case, there are two velocity maxima, one corresponding to image 1 and the other to image 2. The location of these maxima give us (v_x^1, v_y^1) and (v_x^2, v_y^2) and, thus, we can proceed to sort the components.

Note that the Hough transform equation 3.20 becomes degenerate if $\omega_y = 0$ in which case we use an alternative re-arrangement as follows:

$$v_x = \frac{1}{\omega_x\delta t}\delta\phi(\omega_x,\omega_y) \tag{3.22}$$

4. APPLICATION OF THE TECHNIQUE

4.1 SYNTHETIC IMAGES

In Figure 3.1, we showed a sequence of four composite or additive images $f_{t_j}(x,y)$, $j = 0,1,2,3$, together with the component images $f_{t_j}^1(x,y)$

and $f^2_{t_j}(x,y)$ which were used in forming that sequence. Figure 3.5 shows the same sequence of four composite images together with the results of applying the separation algorithm just described, *i.e.* it shows the reconstructed versions of images $f^1_{t_0}(x,y)$ and $f^2_{t_0}(x,y)$. For the sake of comparison, the original images are also shown. The Hough transform for this sequence is shown in Figure 3.6; the two maxima are clearly visible and their positions indicate the velocities of the two component images. The corresponding map of phase changes $\Delta\Phi$ for images $\mathsf{F}^1(\omega_x,\omega_y)$ and $\mathsf{F}^2(\omega_x,\omega_y)$ are shown in Figure 3.7.

Figures 3.8 and 3.9 show the result of applying the separation technique to the same images as those shown in Figure 3.5 but for a range of image velocities.

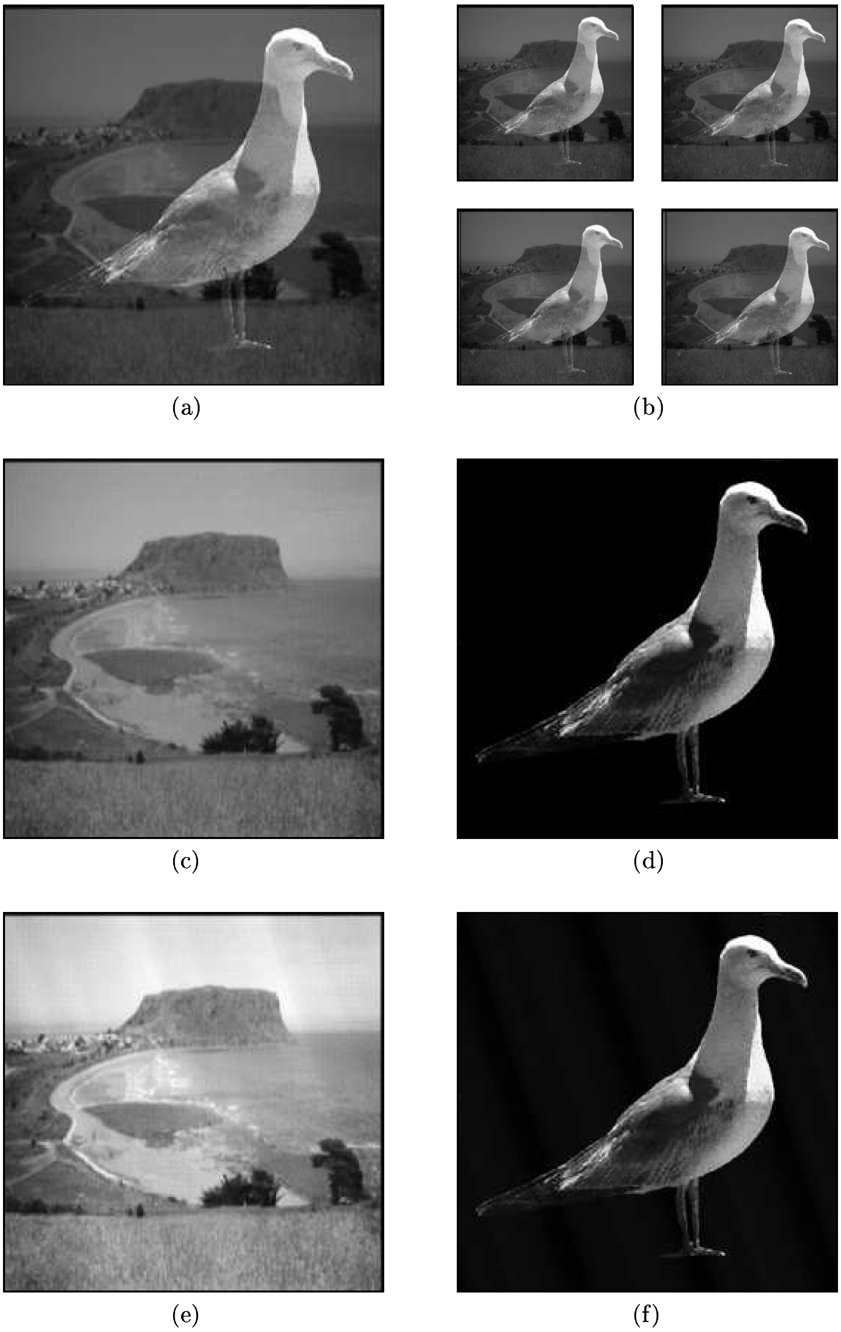
It is clear from Figures 3.5, 3.8, and 3.9 that excellent separation can be achieved. This is to be expected since the problem is well-posed and exactly determined. However, it is evident that there is a small residual error in all of the images manifested by a form of ‘streaking’. These errors are due to the impossibility of solving for some spatial frequencies, specifically those for which the phase change is identical in the forground and background. Note that there is always a set of such frequencies. Recall equation 3.19 which defines the phase change in terms of velocity and spatial frequency:

$$\delta\phi^i(\omega_x,\omega_y) \quad = \quad -(v^i_x\omega_x\delta t + v^i_y\omega_y\delta t)$$

For a given velocity, this equation defines a plane through the origin in (ω_x,ω_y) space and, thus, any two images travelling at distinct velocities will give rise to two intersecting planes (see Figure 3.10.) The intersection of these two planes defines a line in (ω_x,ω_y) space such that $\Delta\Phi^1(\omega_x,\omega_y) = \Delta\Phi^2(\omega_x,\omega_y)$, *i.e.*, the set of all spatial frequencies which exhibit identical phase changes. Clearly, there will always be a non-null intersection and, hence, a non-null set of frequencies for which it will be impossible to assign with certainty to either the foreground or the background. Indeed, we won't even be able to identify the corresponding phasors $\{\mathsf{F}^1\}$ and $\{\mathsf{F}^2\}$. The reason for this is rooted directly in the solution for the Fourier components in equations 3.16 - 3.17.

$$\begin{aligned}\mathsf{F}^1_{t_0} &= \frac{\mathsf{F}_{t_0}\Delta\Phi^2 - \mathsf{F}_{t_1}}{(\Delta\Phi^2 - \Delta\Phi^1)}\\ \mathsf{F}^2_{t_0} &= \frac{\mathsf{F}_{t_0}\Delta\Phi^1 - \mathsf{F}_{t_1}}{(\Delta\Phi^1 - \Delta\Phi^2)}\end{aligned}$$

Since the phase changes $\Delta\Phi^1$ and $\Delta\Phi^2$ in this instance are the same, the denominators in these two equations are zero and the solutions are not



(a) (b)

(c) (d)

(e) (f)

Figure 3.5. A sequence of four additive 256 × 256 pixel images $f_{t_j}(x, y)$ at times t_0, t_1, t_2, and t_3 is shown in (b) with the image at time t_0 shown in (a) at full resolution; (c) and (d) show the the original component images $f^1_{t_0}(x, y)$ and $f^2_{t_0}(x, y)$ while those extracted from the composite images are shown in (e) and (f).

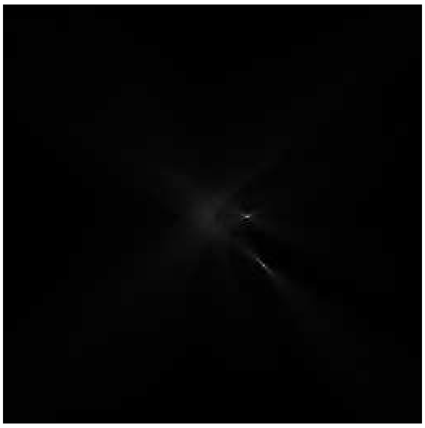
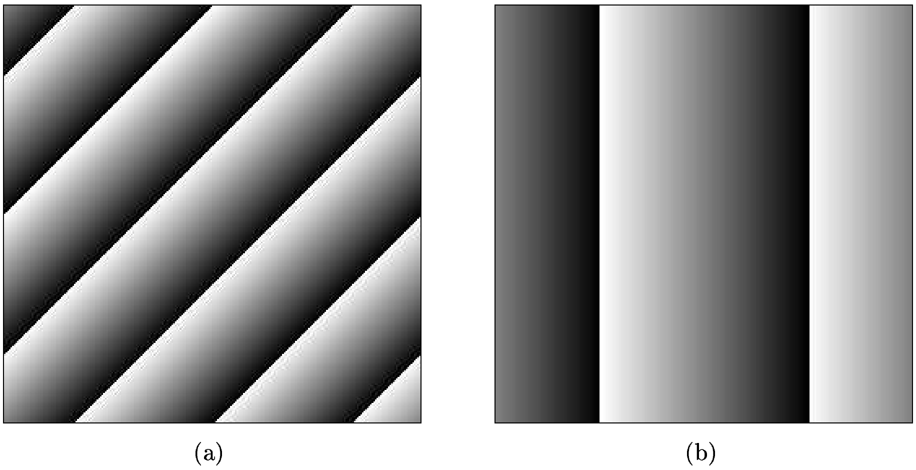


Figure 3.6. Hough transform of the image sequence in fig 3.5; the velocities of the component images correspond to the positions of the two maxima.



(a)

(b)

Figure 3.7. Phase change $\delta\phi(\omega_x, \omega_y) = -(v_x^i \omega_x \delta t + v_y^i \omega_y \delta t)$ for (a) the background and (b) the foreground images shown in figure 3.5; the phase change is proportional to intensity, *i.e.* black denotes zero phase change. Each value is computed individually using equation 3.14.

defined. The streaking in Figures 3.8 and 3.9 is caused by the absence of these components. If required, the streaking can be removed by appropriate post-processing, *e.g.* conditional averaging in a direction normal

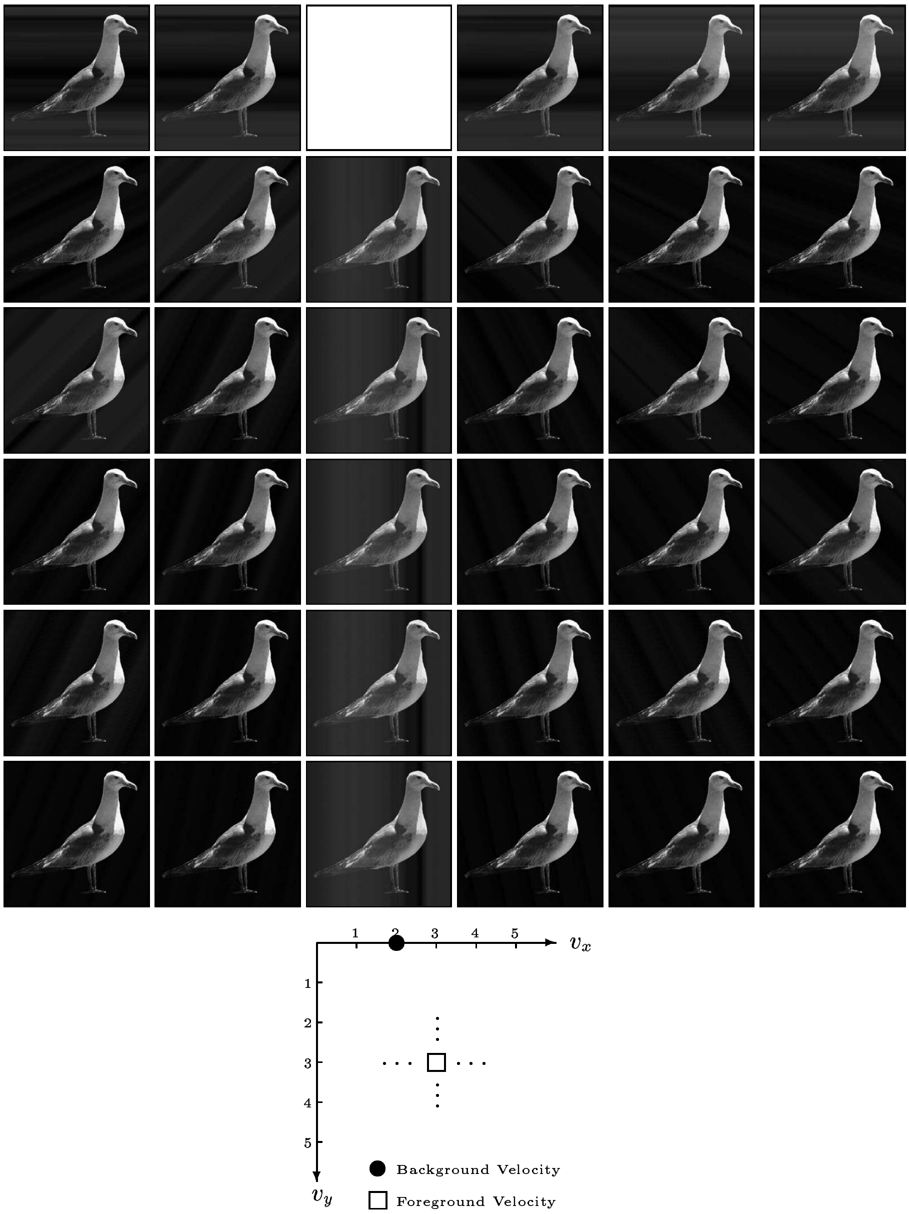


Figure 3.8. Monocular additive sequence: segmentation of the foreground object as a function of foreground image velocity which varies from (0, 0) pixels/frame to (5, 5) pixels/frame.

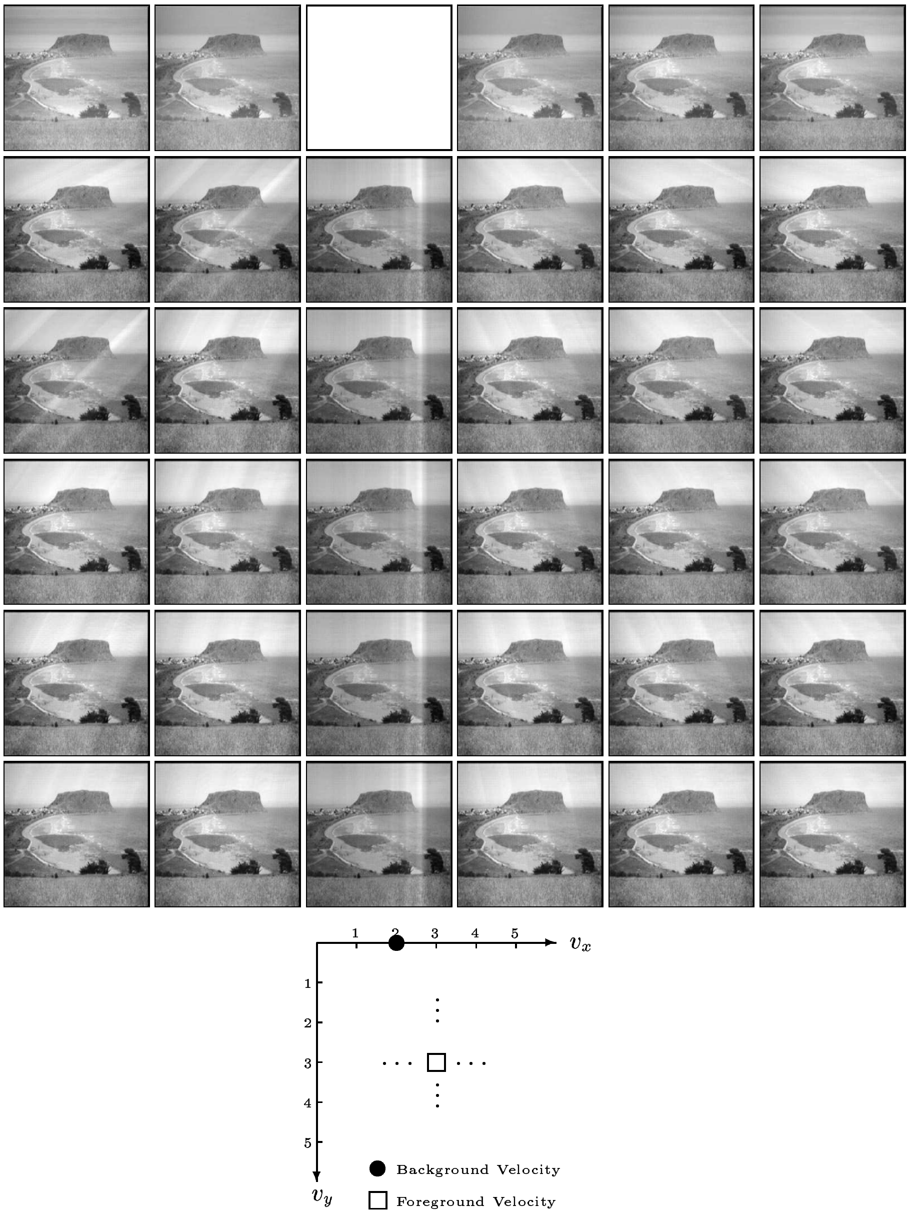


Figure 3.9. Monocular additive sequence: segmentation of the background object as a function of foreground image velocity which varies from (0, 0) pixels/frame to (5, 5) pixels/frame.

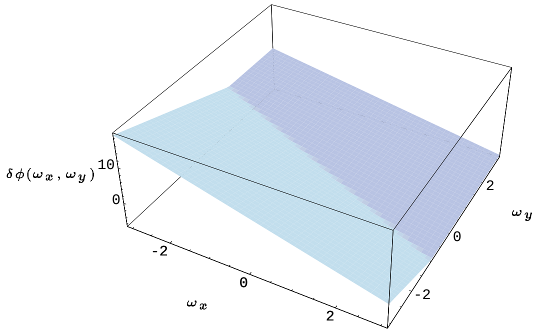


Figure 3.10. The variation of phase change $\delta\phi(\omega_x, \omega_y)$ with spatial frequency (ω_x, ω_y) is defined by the plane equation $\delta\phi(\omega_x, \omega_y) = -(v_x^i \omega_x \delta t + v_y^i \omega_y \delta t)$. The attitude of the plane is determined by the velocity components (v_x, v_y). In this figure, two intersecting planes are shown corresponding to velocities $(v_x, v_y) = (3, 3)$ and $(v_x, v_y) = (2, 0)$ pixels per frame, respectively.

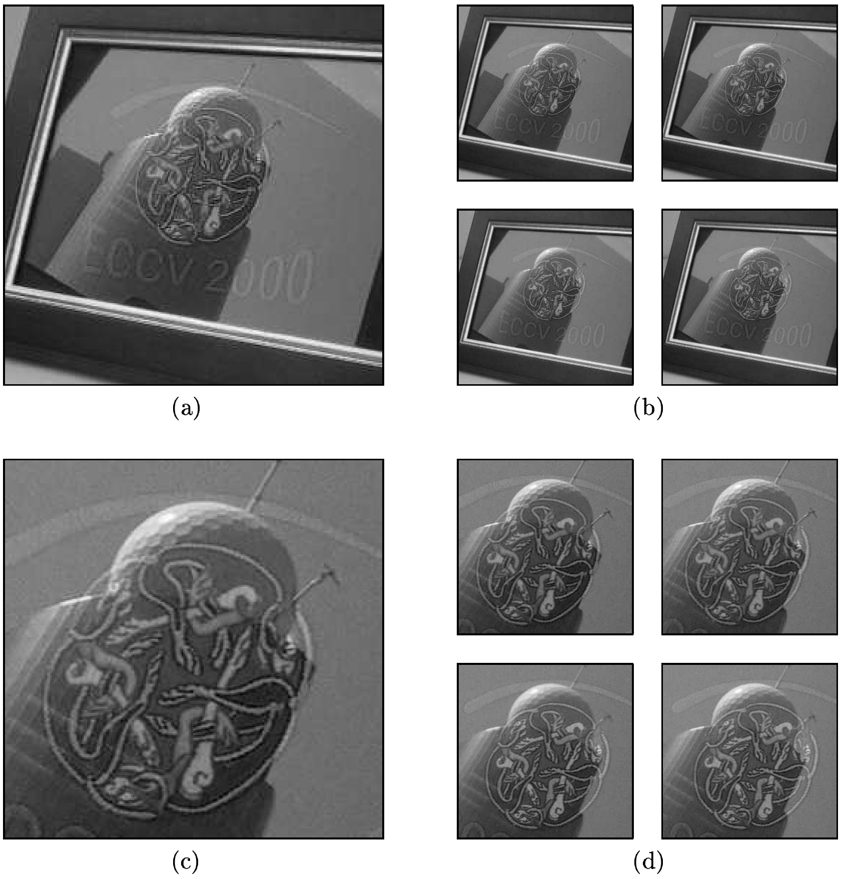
to the known streaking direction.[1] The streaking is stronger when the velocities of the foreground and background are either parallel or orthogonal because, in these two cases, the number of spatial frequencies for which the phase change is exactly equal is at a maximum and, hence, so too is the number of 'missing' phasors (*i.e.* phasors for which we can't produce a solution). This is a result of the sampled nature of the digital image representation of the spatial frequencies; in a continuous image, the number of frequencies lying on the line defined by the intersection of the two phase-change planes is infinite.

Note that there is one empty cell in Figures 3.8 and 3.9; this corresponds to the case where both constituent images are translating with the same velocity and, consequently, they can't be separated since there is nothing to distinguish them.

4.2 REAL IMAGES

Finally, Figures 3.11 to 3.15 show the result of applying the technique to real scenes.

[1] The direction of the streaking is parallel to the direction of the relative velocity of the foreground with respect to the background.


(a) (b)

(c) (d)

Figure 3.11. Example of removal of reflections: a sequence of four 256 × 256 pixel images $f_{t_j}(x, y)$ at times t_0, t_1, t_2, and t_3 is shown in (b) with the image at time t_0 shown in (a) at full resolution; these images show a framed *ECCV 2000* logo with a building reflected in the glass. A detail of this sequence is shown in (c) and (d).

Figure 3.11 shows a glazed picture-frame containing an embroidered *ECCV 2000* logo[2] with a telecommunications building reflected in the glass. Four images in a sequence are shown. Figure 3.11 also shows a detail taken from this sequence. The results of applying the image separation to the image sequence of the logo detail are shown in Figure 3.12 with the foreground and background shown in Figures 3.12 (a) and (b), respectively.

Inspection of these results will reveal imperfections in the image separation. In particular, there is a certain amount of low frequency noise and

[2] *ECCV 2000 — European Conference on Computer Vision*

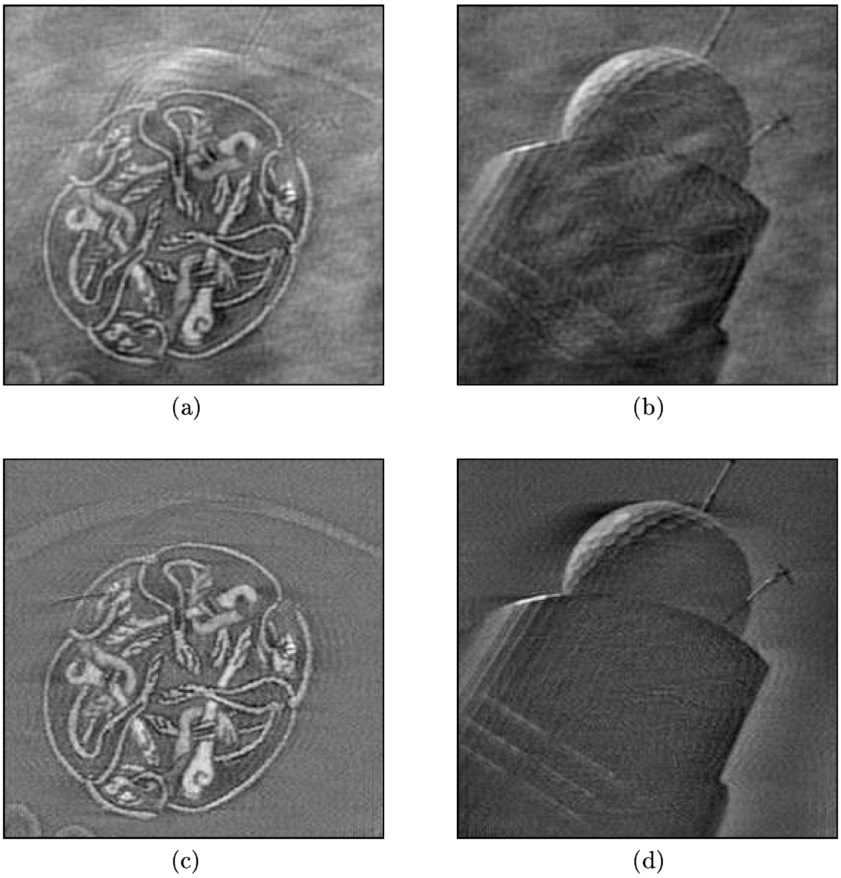


Figure 3.12. Example of removal of reflections: the detail of the *ECCV 2000 logo* in the frame is shown in (a) while the reflection is shown in (b); an improved result in (c) and (d) is achieved by using additional post-processing to be described in the next chapter.

a very minor amount of 'mixing', *i.e.*, some residual components of the foreground can be seen in the background and *vice versa.* Why has this occurred? The reason is again related to the solution for the Fourier components in equations 3.16 - 3.17 at spatial frequencies which exhibit identical or near identical phase changes. In the case of synthetic images, we can compute $\Delta\Phi^1(\omega_x, \omega_y)$ and $\Delta\Phi^2(\omega_x, \omega_y)$ exactly since the translations are exactly integer multiples of one pixel. However, for real images, there is an unavoidable error in computing $\Delta\Phi^1(\omega_x, \omega_y)$ and $\Delta\Phi^2(\omega_x, \omega_y)$, an error which propagates in to the computation of $\mathsf{F}^1_{t_0}$ and $\mathsf{F}^2_{t_0}$. Normally, this error is very small but as $\Delta\Phi^1(\omega_x, \omega_y) \rightarrow \Delta\Phi^2(\omega_x, \omega_y)$, the relative error in the denominator becomes large and, hence, so too does the error in the computation of $\mathsf{F}^1_{t_0}$ and $\mathsf{F}^2_{t_0}$. These errors are the cause of the low-frequency noise in Figures 3.12 (a) and 3.12 (b).

There is yet another problem with the phasors at the spatial frequencies for which $\Delta\Phi^1(\omega_x, \omega_y) \approx \Delta\Phi^2(\omega_x, \omega_y)$. This problem concerns the assignment of phasors to either the foreground or the background. You will recall from Section 3.3 that this sorting or assignment is effected on the basis of the expected phase changes (derived from the computed image velocities). If $\Delta\Phi^1(\omega_x, \omega_y) \approx \Delta\Phi^2(\omega_x, \omega_y)$ then we won't be able to guarantee the correct assignment and we have to assign the phasors at random to either the foreground or the background. If we make the wrong assignment, then we will get a 'mixing' of foreground and background images. Since $\Delta\Phi^1(\omega_x, \omega_y)$ and $\Delta\Phi^2(\omega_x, \omega_y)$ can be computed exactly without error in the case of the synthetic images, this mixing does not occur. However, in the case of the real images, with their attendant inaccuracies or errors in the computation of $\Delta\Phi^1(\omega_x, \omega_y)$ and $\Delta\Phi^2(\omega_x, \omega_y)$, the likelihood of incorrect assignment of phasors is increased for those spatial frequencies for which $\Delta\Phi^1(\omega_x, \omega_y) \approx \Delta\Phi^2(\omega_x, \omega_y)$. This results in the slight mixing evident in Figures 3.12 (a) and 3.12 (b). The solution to both of these problems is the same: remove those phasors at spatial frequencies for which $\Delta\Phi^1(\omega_x, \omega_y) \approx \Delta\Phi^2(\omega_x, \omega_y)$. If we do this in an appropriate manner, we can improve the quality of the results very significantly as shown in Figures 3.12 (c) and 3.12 (d). We will defer our discussion of the details of the removal of these phasors to the next chapter when we will meet this problem again, this time because of errors caused by occlusion.

Consider now figure 3.13 which shows the results of applying the separation technique to the full *ECCV* logo and picture-frame. Again, the presence of low-frequency noise is evident and, indeed, in this instance it is more pronounced. This is due to the fact that the image sequence is not purely additive: only the part of the image containing the logo and the reflection is additive and the surrounding frame and background is not. Actually, we are meeting here a version of the occlusion problem that we will address in the next chapter. However, we can see that, with appropriate removal of the phasors at frequencies where $\Delta\Phi^1(\omega_x, \omega_y) \approx \Delta\Phi^2(\omega_x, \omega_y)$, we can significantly improve the image separation.

Finally, Figure 3.14 shows an example where the reflection overpowers or swamps the figure of interest. Even so, good separation is possible.

For completeness, the Hough transforms for both the examples in Figures 3.12 and 3.13 are shown in Figure 3.15. Comparing these with the Hough transform for the synthetic image sequence in figure 3.6, we see that the maxima are not quite as well defined as they are for the

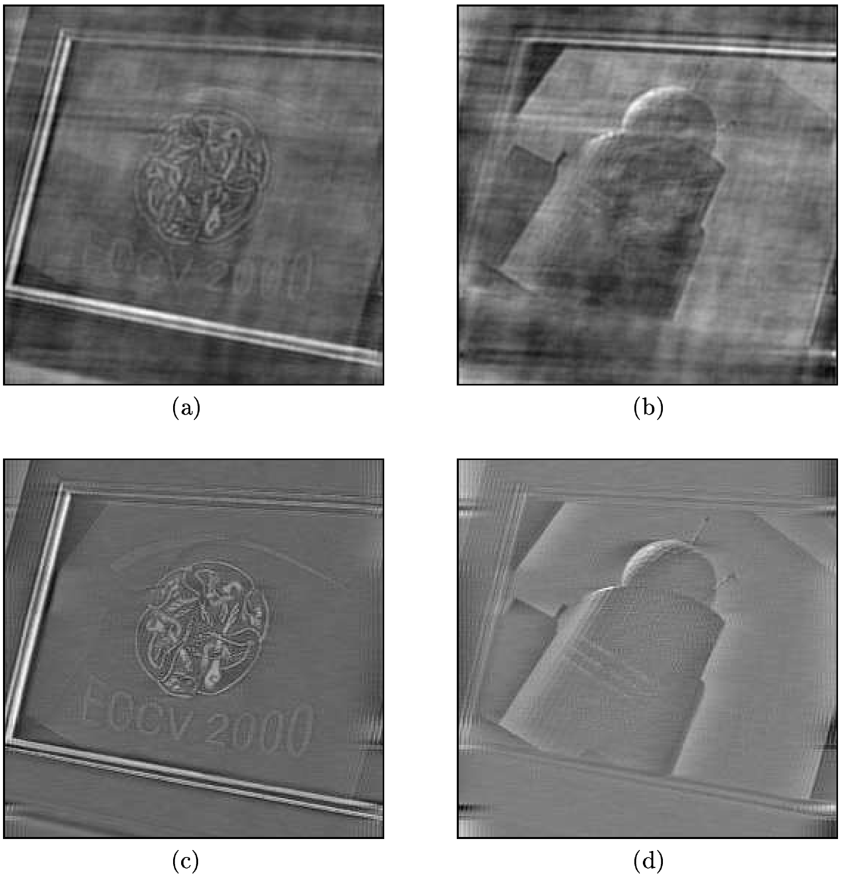
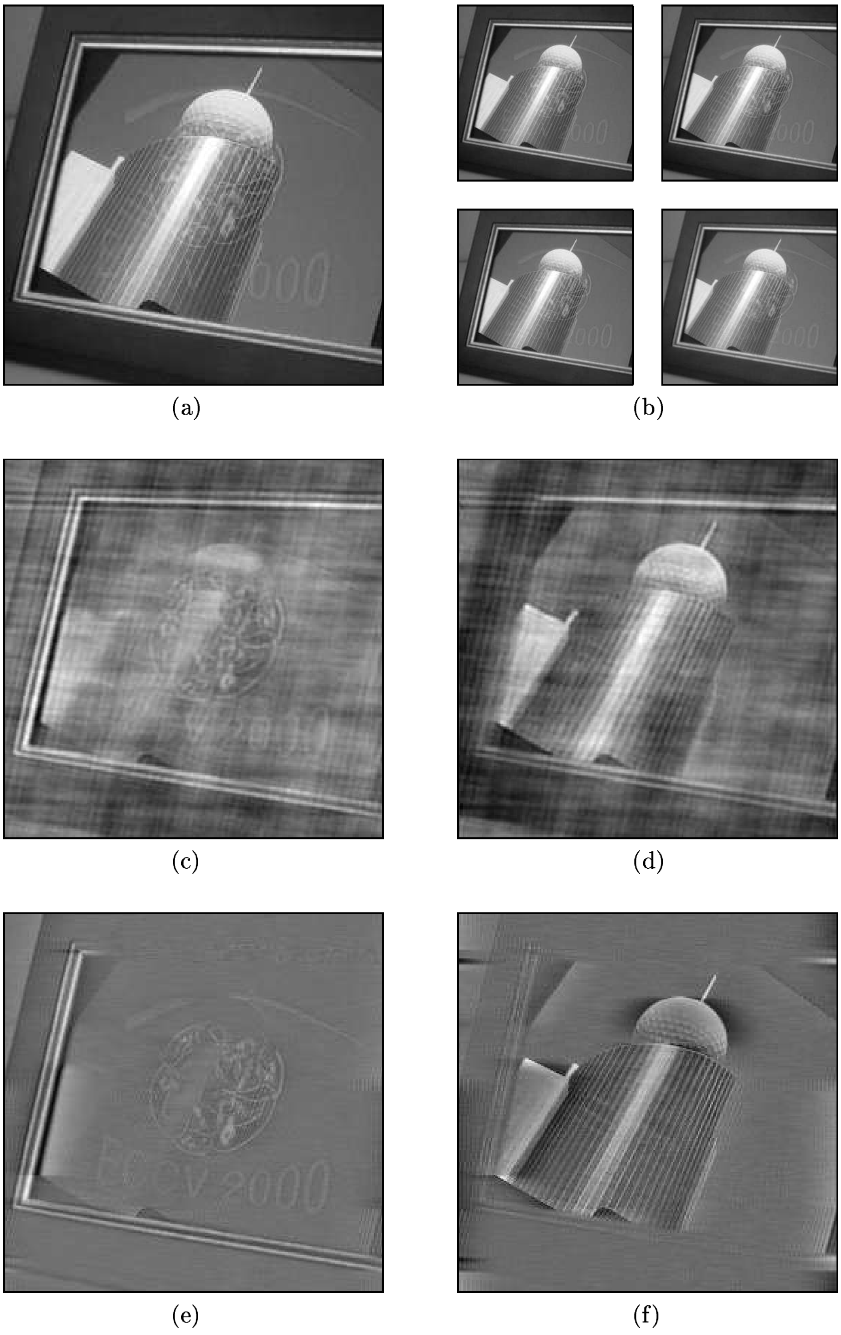


Figure 3.13. Example of removal of reflections: the full *ECCV 2000* logo and frame is shown in (a) while the reflection is shown in (b); again, an improved result in (c) and (d) is achieved by using additional post-processing to be described in the next chapter.

synthetic images, especially for the background which has a velocity close to zero. This smearing of the Hough transform space is an issue we will meet again in the next chapter when we consider occluding images.

5. CONCLUSION

Although additive signals occur more often in, for example, acoustic domains, additive visual signals do occasionally occur in computer vision, such as in the case of a scene viewed through a reflective transparent medium or in digital angiography where images of organs are effectively superimposed. Such visual signals cause significant difficulties for traditional filtering techniques but the Fourier technique just presented copes



(a) (b)

(c) (d)

(e) (f)

Figure 3.14. Example of removal of reflections where the reflection overpowers the figure of interest (a) and (b): the *ECCV 2000* logo in the frame is shown in (c) while the reflection is shown in (d); as before, an improved result in (e) and (f) is achieved by using additional post-processing to be described in the next chapter.

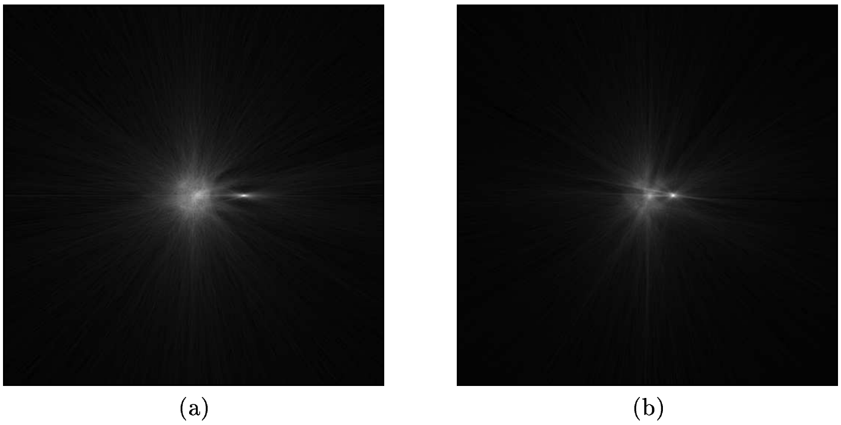


Figure 3.15. Hough transform spaces for the examples shown in Figures 3.11 to 3.13: (a) Hough transform for the extracted detail image sequence and (b) Hough transform for the full frame image sequence.

extremely well in such circumstances. That said, for applications where we are dealing not with the additive superposition of figure on ground but with the occluding superposition of figure on ground, we must turn to the occlusion model to be presented in the next chapter.

Chapter 4

MONOCULAR VISION — SEGMENTATION IN OCCLUDING IMAGES

1. OVERVIEW

In the previous chapter, we developed a model for the separation or decoupling of m additive images from a temporal sequence of the sum of these images, where the images are translating with distinct and unique velocities (including zero velocity). We presented a solution to the problem for the case where $m = 2$, *i.e.* the figure/ground scenario, and we demonstrated this solution for a number of images. In this chapter, we will adapt the additive model to embrace situations where the images are not additive but are, instead, formed by the superposition of an occluding object or objects on an occluded background. That is, the approach will be modified to effect a model-free segmentation of objects undergoing translatory fronto-parallel motion in dynamic image sequences. As with the additive case, object velocities of one pixel per frame are sufficient to achieve segmentation.

We begin by extending the Fourier model presented in the previous chapter to include explicit occlusion components. It turns out that the resultant model is ill-posed in the sense that it is not possible to provide a sufficient number of constraints to fully determine the attendant system of equations and, consequently, we are forced to develop an approximate solution. Based on an analysis of the errors which result from a direct application of the additive model, it emerges that inaccuracies are due not so much to the incorrect solution for the component phasors (and, in particular, to the inevitably incorrect solution for the occluded phasors) as they are to the solution for the angular velocity of each phasor. This means that occlusion can be dealt with by using a more robust approach for computing the angular velocities. Specifically, we can first use the additive model to compute an approximate independent estimate of the angular velocity of each phasor (*i.e.* the phase change per time increment for at each spatial frequency), then these approximations can be pooled and used to compute the object velocities, and finally, knowing an object's velocity we can compute the true angular velocity for each object. This, in turn, allows us to solve for each Fourier component, as before. In a manner which is analogous to that described under identical circumstances in the previous chapter, residual errors arise at frequencies where the phase changes are almost equal and these are eliminated in the same way by well-formed frequency-specific filtering.

2. FIGURE-GROUND SEGMENTATION OF OCCLUDING TRANSLATING OBJECTS

The model developed so far assumes that the component images combine additively to form the resultant image. Although there are important situations or applications where this assumption is valid, *e.g.* superposition of reflections on an image acquired through a transparent medium, there are many more situations where it is not. In particular, the common situation where a moving object occludes a (possibly moving) background violates the assumption (see Figure 4.1). Although the approach described in the previous chapter does in fact produce a reasonable segmentation of the occluding forground objects from the partially occluded background, the results are inevitably imperfect. Our goal in this chapter is to develop the approach so that it can deal with occlusion.

We begin by identifying the cause of the errors which occur when the additive model is applied directly. Recall equations 3.10 to 3.13:

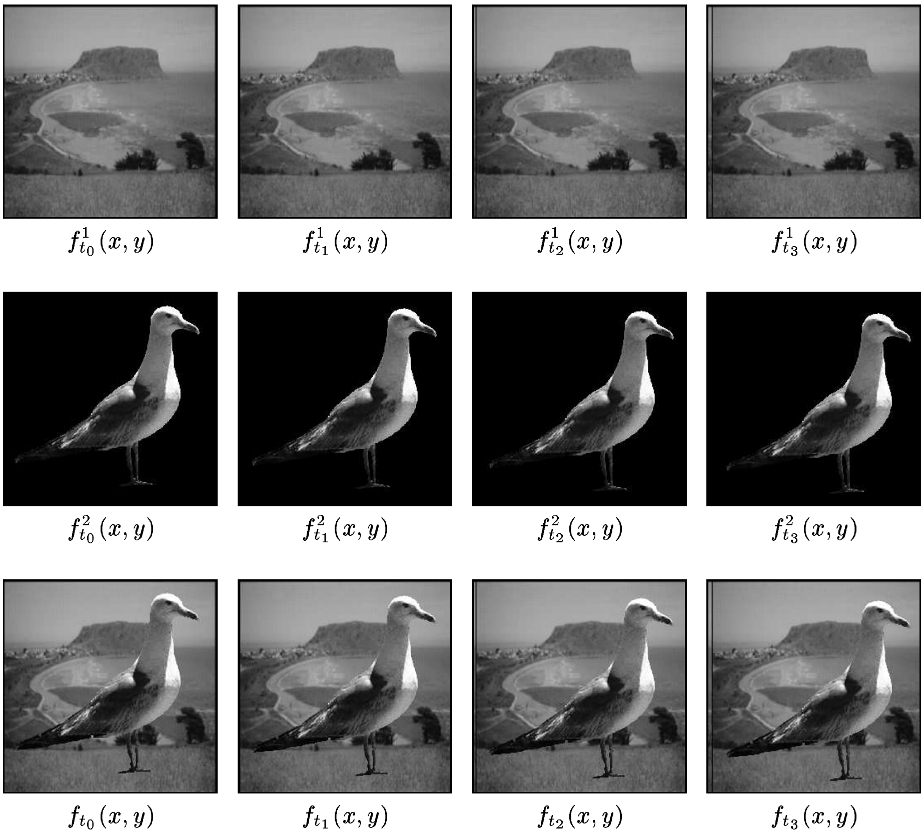
$$\mathsf{F}_{t_0} = \mathsf{F}^1_{t_0}\left(\Delta\Phi^1\right)^0 + \mathsf{F}^2_{t_0}\left(\Delta\Phi^2\right)^0 \tag{4.1}$$

$$\mathsf{F}_{t_1} = \mathsf{F}^1_{t_0}\left(\Delta\Phi^1\right)^1 + \mathsf{F}^2_{t_0}\left(\Delta\Phi^2\right)^1 \tag{4.2}$$

$$\mathsf{F}_{t_2} = \mathsf{F}^1_{t_0}\left(\Delta\Phi^1\right)^2 + \mathsf{F}^2_{t_0}\left(\Delta\Phi^2\right)^2 \tag{4.3}$$

$$\mathsf{F}_{t_3} = \mathsf{F}^1_{t_0}\left(\Delta\Phi^1\right)^3 + \mathsf{F}^2_{t_0}\left(\Delta\Phi^2\right)^3 \tag{4.4}$$

At time t_0, an occluded image is still accurately represented by equation 3.10, *i.e.* the composite image is the sum of the occluded object (or image) and the background object (or image), assuming the background occluded image signal has a zero value wherever the occluding image is non-zero, *i.e.* assuming that there is nothing behind the occluding object. Figure 4.2 shows a characterization of this situation. Of course, such a situation rarely happens in reality since there is almost always



$f^1_{t_0}(x,y)$ $f^1_{t_1}(x,y)$ $f^1_{t_2}(x,y)$ $f^1_{t_3}(x,y)$

$f^2_{t_0}(x,y)$ $f^2_{t_1}(x,y)$ $f^2_{t_2}(x,y)$ $f^2_{t_3}(x,y)$

$f_{t_0}(x,y)$ $f_{t_1}(x,y)$ $f_{t_2}(x,y)$ $f_{t_3}(x,y)$

Figure 4.1. A composite occluding image sequence $f_{t_j}(x,y)$. In this case, m, the number of images, is equal to 2. The goal is to compute $f^1_{t_j}(x,y)$ and $f^2_{t_j}(x,y)$, knowing only $f_{t_j}(x,y)$.

a non-zero occluded signal. Nonetheless, if we aren't interested in this occluding signal, *i.e.* if we don't need to recover what is hidden by the foreground and if we then allow the assumption of a zero 'hole' behind the foreground, then equation 3.10 does hold.

However, there is still going to be a problem with equation 3.11. At time t_1, when the object and/or the background object has moved, there will be a change in the spectral content of the image due to the appearance of visual information in the background which was previously occluded at time t_0 and to the disappearance of visual information which is now occluded at time t_1 (remember that the velocities of the occluded and the occluding object are different). We will call this revealed/hidden signal the *occluded residue* or simply the *residue.* Consequently, to ren-

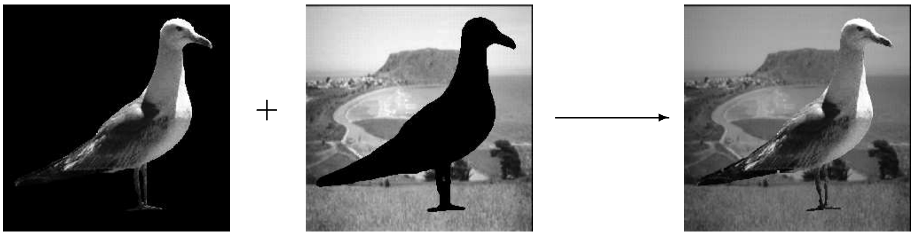


Figure 4.2. An image of composite objects is the sum of the two images in a figurative sense for an occluding foreground if we assume that there is 'hole' behind the foreground object.

der equation 3.11 accurate again we must include a new term F^3 to represent the residue. Equation 3.11 now becomes

$$\mathsf{F}_{t_1} = \mathsf{F}^1_{t_0}\left(\Delta\Phi^1\right)^1 + \mathsf{F}^2_{t_0}\left(\Delta\Phi^2\right)^1 + \mathsf{F}^3 \tag{4.5}$$

Similarly, at time t_2, we must again add a new residue term F^4 and, in addition, we must alter the phase of the previous residue F^3 to reflect the translation of the background. Consequently, we must alter the phase either by $\Delta\Phi^1$ or by $\Delta\Phi^2$ depending on whether $\mathsf{F}^1_{t_0}$ or $\mathsf{F}^2_{t_0}$ is the background. Since we don't know which is the case at this point, we choose, arbitrarily, $\mathsf{F}^2_{t_0}$ as the background occluded image and $\mathsf{F}^1_{t_0}$ as the foreground occluding image. Equation 3.12 then becomes:

$$\mathsf{F}_{t_2} = \mathsf{F}^1_{t_0}\left(\Delta\Phi^1\right)^2 + \mathsf{F}^2_{t_0}\left(\Delta\Phi^2\right)^2 + \mathsf{F}^3\left(\Delta\Phi^2\right)^1 + \mathsf{F}^4 \tag{4.6}$$

In the same way, equation 3.13 becomes:

$$\mathsf{F}_{t_3} = \mathsf{F}^1_{t_0}\left(\Delta\Phi^1\right)^3 + \mathsf{F}^2_{t_0}\left(\Delta\Phi^2\right)^3 + \mathsf{F}^3\left(\Delta\Phi^2\right)^2 + \mathsf{F}^4\left(\Delta\Phi^2\right)^1 + \mathsf{F}^5 \tag{4.7}$$

Now, instead of four complex unknowns as in the additive situation, we now have three additional unknowns, making a total of seven. Consequently, the system is underdetermined. Unfortunately, every time we add a new constraint, we also introduce an additional unknown representing a new residue — there is always a hidden signal which is revealed when the foreground moves relative to the background. As a result, we can't proceed to solve the new set of equations in a straightforward manner.

In order to make the problem tractable, we must adopt an alternative strategy, the basis of which lies in the observation that, for small velocities, the residue forms a very small part of the overall image. From

Parseval's theorem, this equates to saying that the change in the spectral energy in the images is very small and the error introduced by ignoring the residue should also be small. But when we do ignore the residue by using the additive model, it is clear that the errors are not small at all. On the contrary, they are quite significant. This leads us to the conclusion that the errors are due not so much to the spectral energy and the computation of the magnitude of the Fourier components but instead to the computation of their phases and, by extension, to the computation of their temporal phase changes $\Delta\Phi^1$ and $\Delta\Phi^2$. This has been borne out by empirical investigation. Consequently, the solution to the residue problem is to adopt a new strategy to estimate the correct phase changes $\Delta\Phi^1$ and $\Delta\Phi^2$, and then to proceed as before to solve for the Fourier components $\mathsf{F}^1_{t_0}$ and $\mathsf{F}^2_{t_0}$. Essentially, after identifying $\Delta\Phi^1$ and $\Delta\Phi^2$ for each phasor in isolation, we exploit the velocity-dependent and frequency-dependent regularity of $\Delta\Phi^1$ and $\Delta\Phi^2$ to allow us to pool the incorrectly solved values and estimate the true values. We will set out this procedure in detail in the next section.

2.1 SOLUTION FOR THE PHASE CHANGES

Recall equation 3.19 in Chapter 3:

$$\delta\phi(\omega_x, \omega_y) = -(v^i_x \omega_x \delta t + v^i_y \omega_y \delta t) \tag{4.8}$$

As we have seen previously, this equation implies that the phase changes for a given velocity lie in a plane through the origin in the spatial frequency domain (ω_x, ω_y). In the occluded case which we are addressing in this chapter, we first solve for $\Delta\Phi^1$ and $\Delta\Phi^2$ as in the additive case using equations 3.14 and 3.15 in the previous chapter, *i.e.* by assuming the residue is zero and that the additive model is faithful. For convenience, we reproduce equations 3.14 and 3.15 again here:

$$\begin{aligned}
\Delta\Phi^1 &= \frac{-b+\sqrt{z}}{2a} && (4.9)\\
\Delta\Phi^2 &= \frac{-b-\sqrt{z}}{2a} && (4.10)\\
\mathsf{F}^1_{t_0} &= \frac{\mathsf{F}_{t_0}\Delta\Phi^2 - \mathsf{F}_{t_1}}{(\Delta\Phi^2 - \Delta\Phi^1)} && (4.11)\\
\mathsf{F}^2_{t_0} &= \frac{\mathsf{F}_{t_0}\Delta\Phi^1 - \mathsf{F}_{t_1}}{(\Delta\Phi^1 - \Delta\Phi^2)} && (4.12)
\end{aligned}$$

where $a = (\mathsf{F}_{t_1})^2 - \mathsf{F}_{t_0}\mathsf{F}_{t_2}$, $b = \mathsf{F}_{t_0}\mathsf{F}_{t_3} - \mathsf{F}_{t_1}\mathsf{F}_{t_2}$, $c = (\mathsf{F}_{t_2})^2 - \mathsf{F}_{t_1}\mathsf{F}_{t_3}$, and $z = b^2 - 4ac = \alpha + i\beta$ with

$$\sqrt{z} = \begin{cases} \pm\left[\left(\frac{\alpha+\sqrt{\alpha^2+\beta^2}}{2}\right)^{\frac{1}{2}} + i\left(\frac{-\alpha+\sqrt{\alpha^2+\beta^2}}{2}\right)^{\frac{1}{2}}\right], & \beta > 0 \\ \pm\left[-\left(\frac{\alpha+\sqrt{\alpha^2+\beta^2}}{2}\right)^{\frac{1}{2}} + i\left(\frac{-\alpha+\sqrt{\alpha^2+\beta^2}}{2}\right)^{\frac{1}{2}}\right], & \beta < 0 \\ \pm\sqrt{\alpha}, & \beta = 0, \alpha \geq 0 \\ \pm i\sqrt{-\alpha}, & \beta = 0, \alpha < 0 \end{cases}$$

However, since the residue is not zero, the solution for each $\Delta\Phi^1$ and $\Delta\Phi^2$ will include an error $\delta_{\Delta\Phi^1}$ and $\delta_{\Delta\Phi^2}$. Since these errors are not systematic for all spatial frequencies (ω_x, ω_y), we can estimate the true value of $\Delta\Phi^1$ and $\Delta\Phi^2$ by fitting a plane to the erroneous values $\Delta\Phi^{1'} = \Delta\Phi^1 + \delta_{\Delta\Phi^1}$ and $\Delta\Phi^{2'} = \Delta\Phi^2 + \delta_{\Delta\Phi^2}$. Actually, we use the values $\Delta\Phi^{1'}(\omega_x, \omega_y)$ and $\Delta\Phi^{2'}(\omega_x, \omega_y)$ to estimate the velocity (v_x^1, v_y^1) of the foreground occluding object and (v_x^2, v_y^2) of the background occluded image using a Hough transform as described in the previous chapter, thus yielding the parameters of the plane defining the phase changes for that image and, hence, the true values of $\Delta\Phi^1$ and $\Delta\Phi^2$ for any frequency (ω_x, ω_y).

We note here one small but important difference between this use of equation 3.19 in solving the phase changes for the occluded model and its use in the additive model. In the previous case (*i.e.* the additive case), we used all spatial frequencies as input to the Hough transform. However, since we necessarily have to use an estimate in the solution to the occlusion model, it is clear that each spatial frequency will give rise to a solution which includes an error. This error manifests itself as a small variation in the Hough transform solution for v_x and v_y and, hence, there is a consequent smearing of the Hough transform space. This yields a distributed maximum rather than a point-like maximum at the two velocities (v_x^1, v_y^1) and (v_x^2, v_y^2). It is significant that these errors grow with frequency and the higher spatial frequencies give rise to incorrect solutions for $\Delta\Phi^1$ and $\Delta\Phi^2$, *i.e.* the relative error is larger than the phase difference. This, in turn, gives rise to erroneous maxima in the Hough space which swamp the true maxima since the number of high spatial frequencies quickly out-grows the number of low frequencies. To combat this, in the occlusion model, we only use the low spatial frequencies to estimate (v_x^1, v_y^1) and (v_x^2, v_y^2) using the Hough transform.

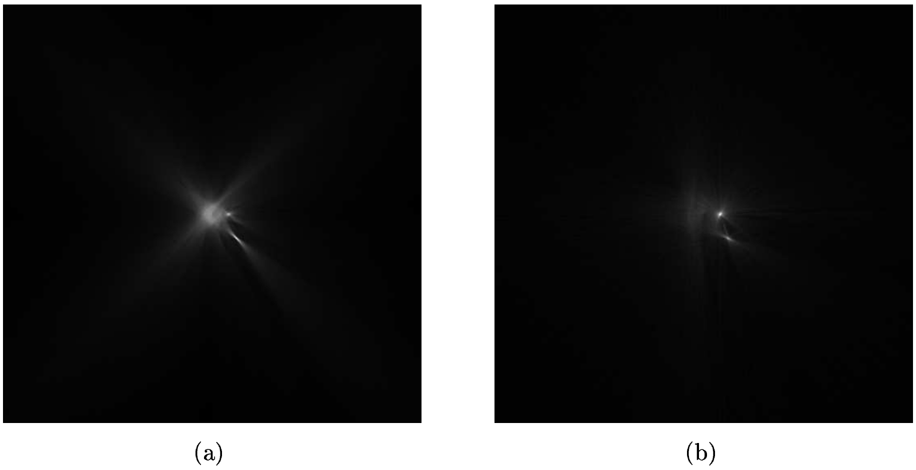


Figure 4.3. Two Hough tranforms computed from the occluded image sequence shown in Figure 4.1: (a) transform computed with the solutions for the phase changes at all spatial frequencies; (b) transform computed using only those frequencies (ω_x, ω_y) where $|\omega_x| + |\omega_y| \leq \frac{2}{3}\omega_{x,y}^{\max}$ where $\omega_{x,y}^{\max}$ is the maximum spatial frequency.

Figure 4.3 shows two Hough transforms computed from the occluded image sequence shown in Figure 4.1. The first transform was computed with the solutions for the phase changes at all spatial frequencies while the second was computed using only those frequencies (ω_x, ω_y) where $|\omega_x| + |\omega_y| \leq \frac{2}{3}\omega_{x,y}^{\max}$ where $\omega_{x,y}^{\max}$ is the maximum spatial frequency (see Chapter 2, Section 1.2). The factor $\frac{2}{3}$ is a somewhat arbitrary cut-off value which yields good results.

Having computed the velocities (v_x^1, v_y^1) and (v_x^2, v_y^2), we can then compute an estimate of the true incremental phase change $\Delta\Phi^1(\omega_x, \omega_y)$ and $\Delta\Phi^2(\omega_x, \omega_y)$ for each Fourier component, *i.e.* at each spatial frequency. Specifically, we have

$$\Delta\Phi^i = e^{-i(\omega_x v_x^i \delta t + \omega_y v_y^i \delta t)} \tag{4.13}$$

and, hence,

$$\Re(\Delta\Phi^i) = \cos(\omega_x v_x^i \delta t + \omega_y v_y^i \delta t) \tag{4.14}$$

$$\Im(\Delta\Phi^i) = \sin(\omega_x v_x^i \delta t + \omega_y v_y^i \delta t) \tag{4.15}$$

2.2 SOLUTION FOR THE FOURIER COMPONENTS

At this point, we have 'solved' (or, rather, estimated) two of the unknowns $\Delta\Phi^1$ and $\Delta\Phi^2$, and it remains simply to solve for the Fourier components $\mathsf{F}^1_{t_0}$ and $\mathsf{F}^2_{t_0}$ using equations 4.11 and 4.12. Note that we now solve these equations for all spatial frequencies, even though we only used the low spatial frequencies to estimate the phase changes $\Delta\Phi^1$ and $\Delta\Phi^2$.

There remains one problem, however, a problem which is analogous to the one we met with the additive images concerning the solution of Fourier components at frequencies which exhibit identical phase changes $\Delta\Phi^1$ and $\Delta\Phi^2$. As before, equations 4.11 and 4.12 are degenerate if $\Delta\Phi^1 = \Delta\Phi^2$, that is, if the incremental phase change of the occluded and occluding objects are equal. Unfortunately, matters are somewhat worse here since significant errors also arise as $\Delta\Phi^1 \to \Delta\Phi^2$ because the numerators in equations 4.11 and 4.12 involve an estimate of $\Delta\Phi^1$ and $\Delta\Phi^2$ rather than an exact value. Hence, the relative magnitude of the error inherent in this estimate grows exponentially as $\Delta\Phi^1 \to \Delta\Phi^2$, that is, as the denominator approaches zero. To eliminate the influence of these errors on the computation of the occluded and occluding signal, $\mathsf{F}^2_{t_0}$ and $\mathsf{F}^1_{t_0}$, we must modify them as $\Delta\Phi^1 \to \Delta\Phi^2$. In particular, we modify all Fourier components $\mathsf{F}^2_{t_0}$ and $\mathsf{F}^1_{t_0}$ whose associated phase changes satisfy the following condition:

$$\left|\Delta\Phi^1 - \Delta\Phi^2\right| \leq \tau v_{\max} \tag{4.16}$$

where τ is some specified tolerance and $v_{\max}$ is the magnitude of the maximum velocity exhibited by either the occluded or the occluding image.

We do not, however, simply remove these components (*i.e.* assign them a value of zero) since this would be equivalent to the implementation of an ideal pulse-shaped band-stop filter. Instead, we attenuate progressively the band-stop frequencies before computing the inverse FFT of the occluding and background images. The approach we have adopted is to attenuate frequencies satisfying equation 4.16 according to the function

$$\begin{aligned} \mathsf{G}^{1,2}_{t_0}(\omega_x,\omega_y) &= \mathsf{F}^{1,2}_{t_0}(\omega_x,\omega_y)\left(\sin\left(\frac{\left|\Delta\Phi^1 - \Delta\Phi^2\right|}{\tau v_{\max}} \times \frac{\pi}{2}\right)\right)^{2n} && (4.17)\\ &\quad \text{if} \left|\Delta\Phi^1 - \Delta\Phi^2\right| \leq \tau v_{\max} \\ \mathsf{G}^{1,2}_{t_0}(\omega_x,\omega_y) &= \mathsf{F}^{1,2}_{t_0}(\omega_x,\omega_y), \text{otherwise} && (4.18) \end{aligned}$$

The exponent $n \geq 1$ determines the slope of the filter cutoff. As $n \to \infty$, the filter approaches an ideal filter. In the following, unless stated otherwise, we use values of $\tau = 0.15$ and $n = 1$ throughout.

3. APPLICATION OF THE TECHNIQUE

In the following, we will evaluate the technique using both synthetic images and real images: the synthetic to assess the performance as a function of object velocity and image complexity and the real to assess the performance in dealing with scenarios which violate the assumptions of homogeneous fronto-parallel object translation.

3.1 SYNTHETIC IMAGES

Figure 4.4 shows a sequence of four composite occluding images $f_{t_j}(x, y)$, $j = 0, 1, 2, 3$, together with the component images $f^1_{t_j}(x, y)$ and $f^2_{t_j}(x, y)$ which were used in forming that sequence. This test scenario comprises two images: a background seaside scene and a foreground seagull. The background scene and the foreground seagull move independently of one another with velocities[1] (v_{x_1}, v_{y_1}) and (v_{x_2}, v_{y_2}), respectively. The technique was tested for velocities in the ranges $(0, 0) \leq (v_{x_1}, v_{y_1}) \leq (4, 0)$ and $(0, 0) \leq (v_{x_2}, v_{y_2}) \leq (5, 5)$, in increments of one pixel. Figure 4.4 also shows the results of applying the segmentation algorithm just described for $(v_{x_1}, v_{y_1}) = (2, 0)$ and $(v_{x_2}, v_{y_2}) = (3, 3)$, *i.e.* it shows the reconstructed versions of images $f^1_{t_0}(x, y)$ and $f^2_{t_0}(x, y)$. The Hough transform for this sequence is shown in Figure 4.5; the positions of the two maxima indicate the velocities of the two component images. The corresponding map of phase changes $\Delta\Phi$ for images $\mathsf{F}^1(\omega_x, \omega_y)$ and $\mathsf{F}^2(\omega_x, \omega_y)$ are shown in Figure 4.6.

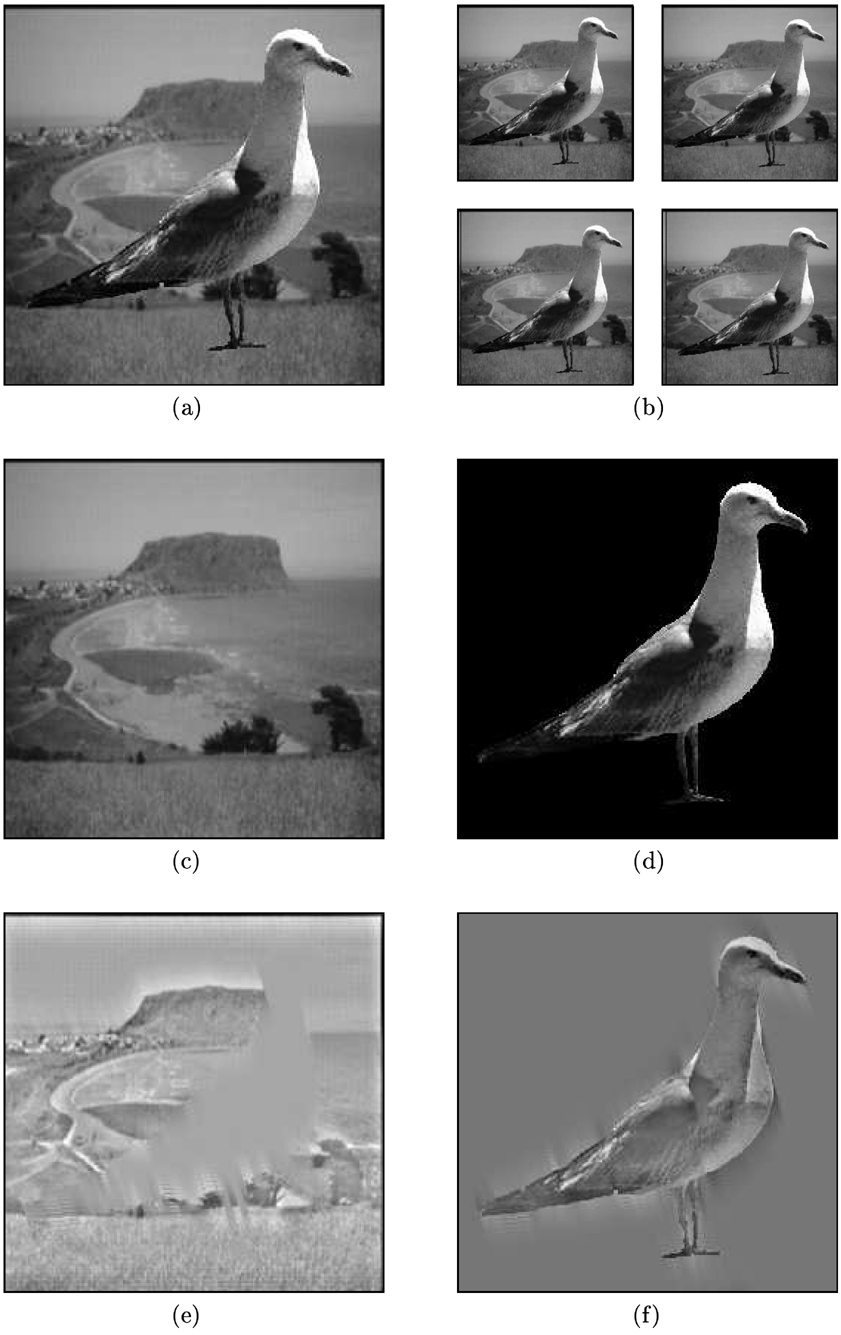
Figures 4.7 and 4.8 show the result of applying the separation technique to the same images as those shown in Figure 4.4 but the full range of test velocities $(0, 0) \leq (v_{x_1}, v_{y_1}) \leq (4, 0)$ and $(0, 0) \leq (v_{x_2}, v_{y_2}) \leq (5, 5)$.

Note that there is one empty cell in Figures 4.7 and 4.8; this corresponds to the case where both constituent images are translating with the same velocity and, consequently, they can't be segmented since there is no relative motion by which they can be distinguished.

3.2 REAL IMAGES

Figure 4.9 shows a sequence of four images of an indoor scene where an SLR camera occludes a set of books. Although the background,

[1] Velocities or displacements/frame.



(a) (b)

(c) (d)

(e) (f)

Figure 4.4. A sequence of four occluding 256×256 pixel images $f_{t_j}(x, y)$ at times t_0, t_1, t_2, and t_3 is shown in (b) with the image at time t_0 shown in (a) at full resolution; (c) and (d) show the the original component images $f^1_{t_0}(x, y)$ and $f^2_{t_0}(x, y)$ while those extracted from the composite images are shown in (e) and (f).

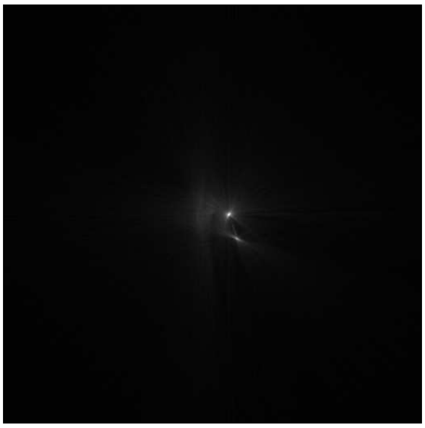


Figure 4.5. Hough transform of the image sequence in fig 4.4; the velocities of the component images correspond to the positions of the two maxima.

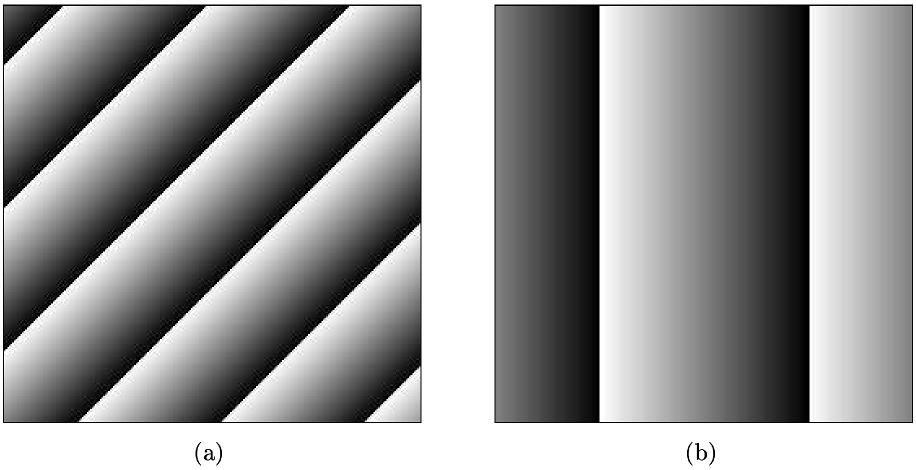


Figure 4.6. Phase change $\delta\phi(\omega_x, \omega_y) = -(v_x^i \omega_x \delta t + v_y^i \omega_y \delta t)$ for the (a) background and (b) foreground images shown in Figure 4.4; the phase change is proportional to intensity, *i.e.* black denotes zero phase change.

comprising the spines of each book, is almost planar and will thus have a homogeneous velocity as required by the model, the foreground, *i.e.* the camera, is a 3-D object and each part will have a slightly different velocity, thereby violating the assumption on which the model is based. Nonetheless, the technique successfully segments the camera foreground from the book background.

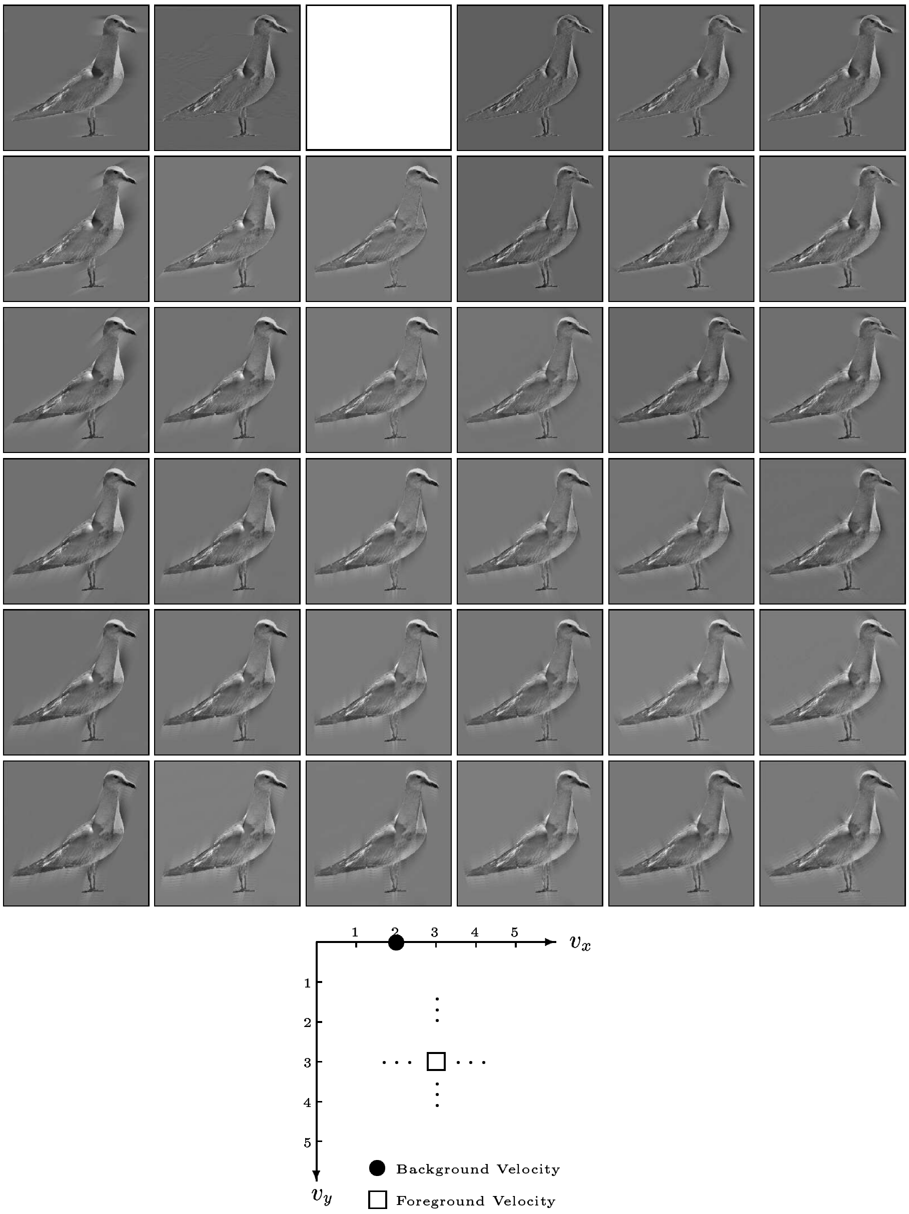


Figure 4.7. Occlusion model: segmentation of the foreground object as a function of foreground image velocity.

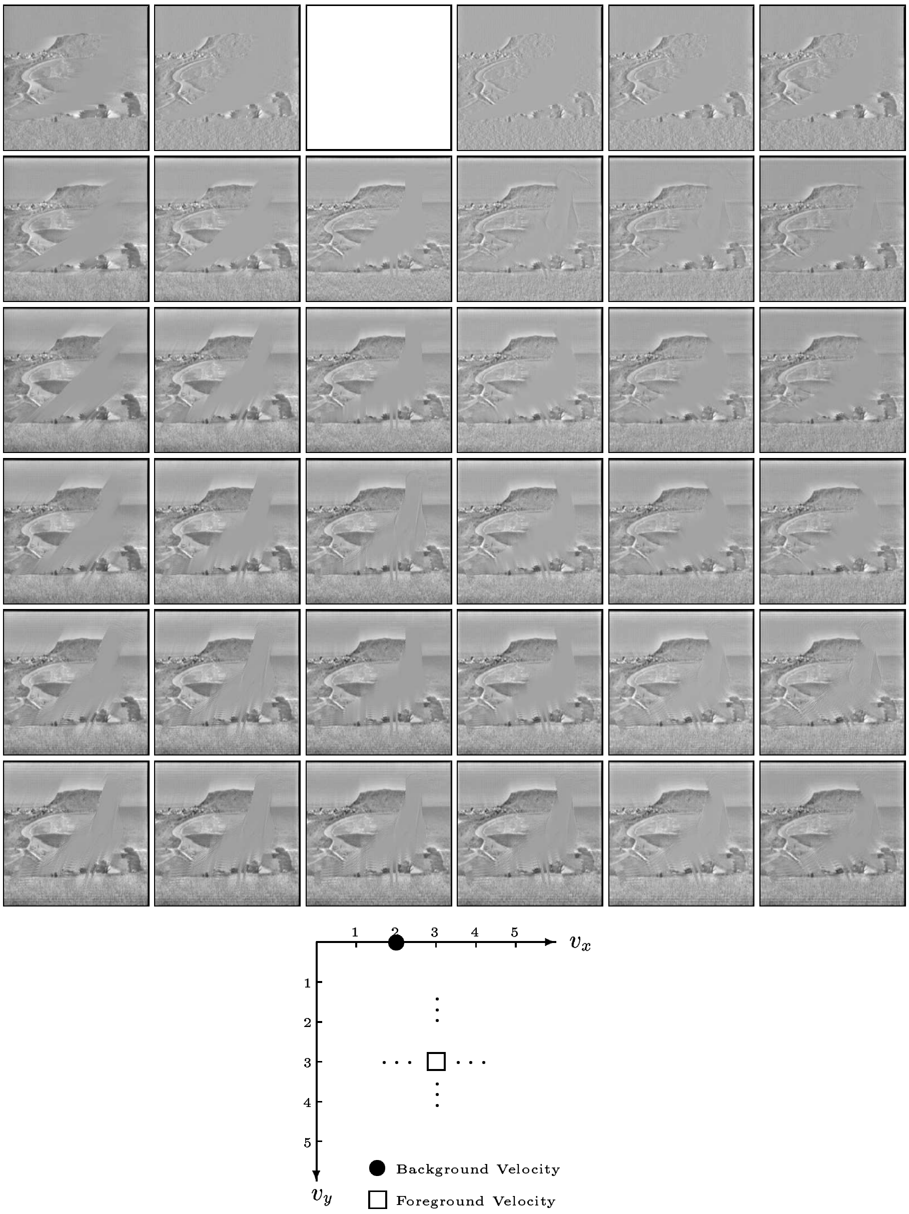
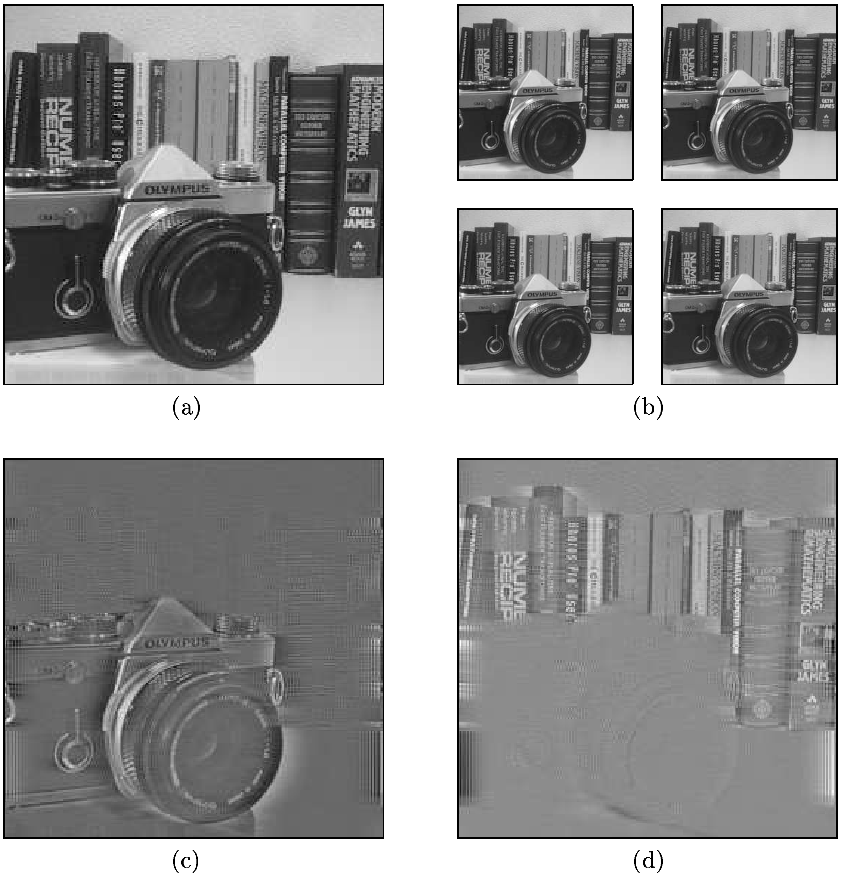


Figure 4.8. Occlusion model: segmentation of the background object as a function of foreground image velocity.



(a) (b)

(c) (d)

Figure 4.9. Example of figure/ground segmentation in real scenes: a sequence of four 256×256 pixel images $f_{t_j}(x, y)$ at times t_0, t_1, t_2, and t_3 is shown in (b) with the image at time t_0 shown in (a) at full resolution. The computed figure and ground images are shown in (c) and (d).

A similar scenario is shown in Figure 4.10 where the camera is replaced by a coke can with almost equivalent results.

In the scene shown in Figure 4.11, the camera is positioned in the background in front of the books but behind the coke can so that the background is now quite definitely non-planar and the model's assumptions are violated to a greater extent than in the scenes shown in Figures 4.9 and 4.10. Nonetheless, we have again achieved successful segmentation. Note though that there is a faint 'ghosting' of the Coca-Cola writing due to its high contrast in the original sequence.

Finally, we come to an example which breaks many of the assumptions upon which the technique is based. Figure 4.12 shows a scene of a

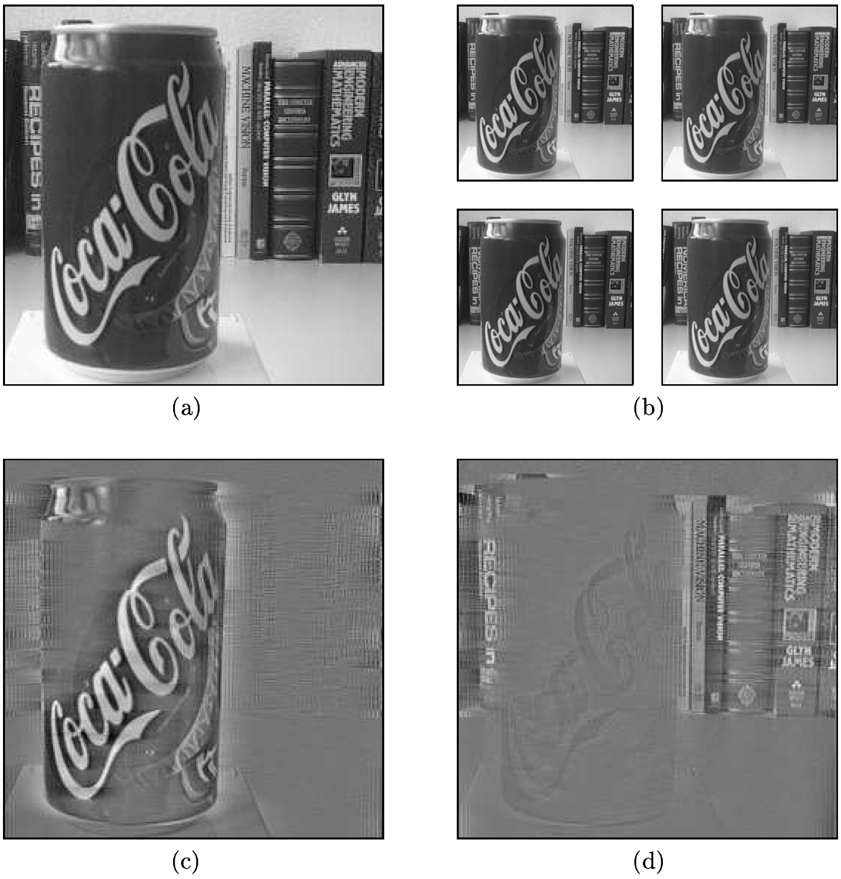


Figure 4.10. Example of figure/ground segmentation in real scenes: a sequence of four 256 × 256 pixel images $f_{t_j}(x, y)$ at times t_0, t_1, t_2, and t_3 is shown in (b) with the image at time t_0 shown in (a) at full resolution. The computed figure and ground images are shown in (c) and (d).

building and a car park viewed through a window with the foreground comprising three objects on the window-sill. In this case, the camera is translating laterally and the objects in the scene therefore move relative to the camera. However, some of the vehicles and the trees in the background are moving independently. In this instance, with so many violations of the assumptions of unique fronto-parallel object motion, it is inevitable that the technique will not work perfectly. Nevertheless, the foreground and the background are successfully segmented albeit that there are clear artifacts introduced in the process. In the next section, we will look at ways in which we can overcome these problems and

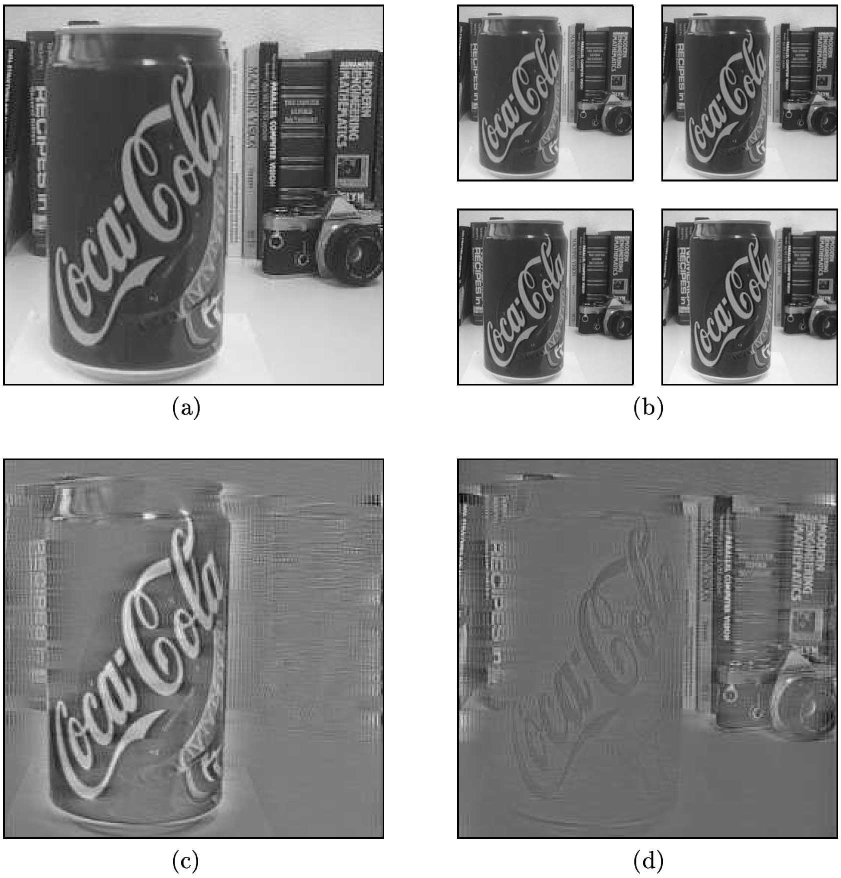


Figure 4.11. Example of figure/ground segmentation in real scenes: a sequence of four 256 × 256 pixel images $f_{t_j}(x, y)$ at times t_0, t_1, t_2, and t_3 is shown in (b) with the image at time t_0 shown in (a) at full resolution. The computed figure and ground images are shown in (c) and (d).

improve the results shown in Figure 4.12 (see Figure 4.17 for the final outcome).

For completeness, the Hough transforms for all four of the examples in Figures 4.9 to 4.12 are shown in Figure 4.13.

4. IMAGE COMPLEXITY

At the outset, in Chapter 1, we claimed that the visual space-domain complexity of the images[2] is relatively inconsequential because the seg-

[2] By visual space-domain complexity, we mean the complexity of the 2-D image pattern, *not* the spatial complexity of the underlying 3-D scene.

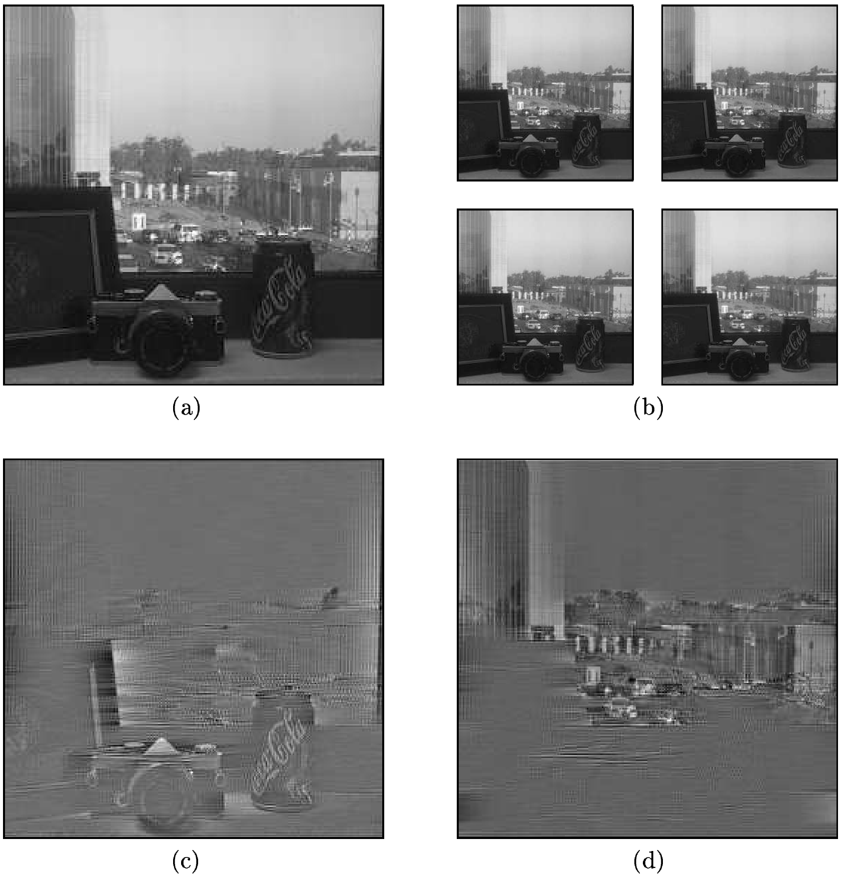


Figure 4.12. Example of figure/ground segmentation in real scenes: a sequence of four 256 × 256 pixel images $f_{t_j}(x, y)$ at times t_0, t_1, t_2, and t_3 is shown in (b) with the image at time t_0 shown in (a) at full resolution. The computed figure and ground images are shown in (c) and (d). See Figure 4.17 for an improved segmentation of this scene.

mentation is effected in the Fourier domain. In this section, we will provide some empirical substantiation for this claim.

Figure 4.14 shows a foreground and a background image of Gaussian noise ($\mu = 0; \sigma = 10$), each image having been created independently, together with their additive and occluding superposition at time t_0. It also shows the result of separating the additively-generated composite images based on the sequence at time t_0, t_1, t_2, and t_3 using the additive model and the result of segmenting the occlusion-based composite images using the adapted occlusion model. These results are based on a background image velocity of (2, 0) pixels per frame in the x and y

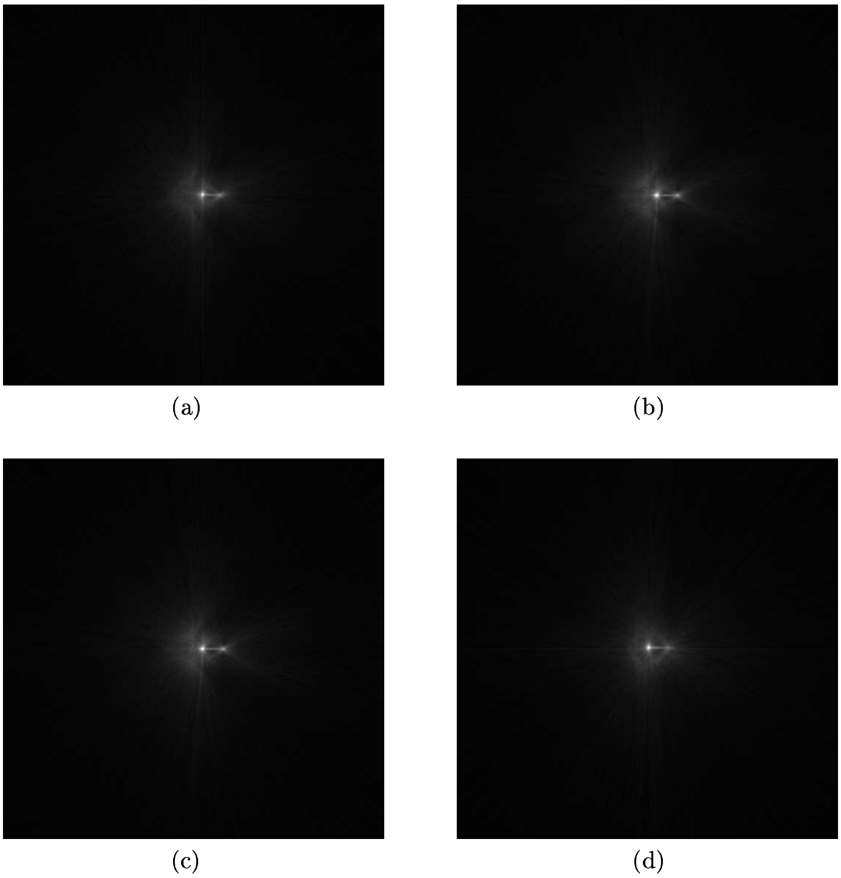


Figure 4.13. Hough transform spaces for the examples shown in Figures 4.9 to 4.12.

direction, and a foreground image velocity of (3, 3) pixels per frame. In both cases, almost perfect segmentation has been achieved.

As another example, Figures 4.15 and 4.16 show two variations of the so-called 'Picket Fence' scenario where a background is visible only through gaps in the foreground object [39]. In the first variation, the background seaside scene is occluded by a foreground comprising a series of randomly textured horizontal bars where the foreground and background velocities are (3, 0) and (1, 0) pixels/frame, respectively. Applying the occlusion model with a very low threshold on attenuation $\tau = 0.01$ results in excellent segmentation. This is primarily due to the fact that the changes in the background scene due to occlusion are very small since the direction of motion is parallel to the major axes of the bars; in effect, we have a good approximation to two additive images, albeit that they are not spatially contiguous. The second variation shows an

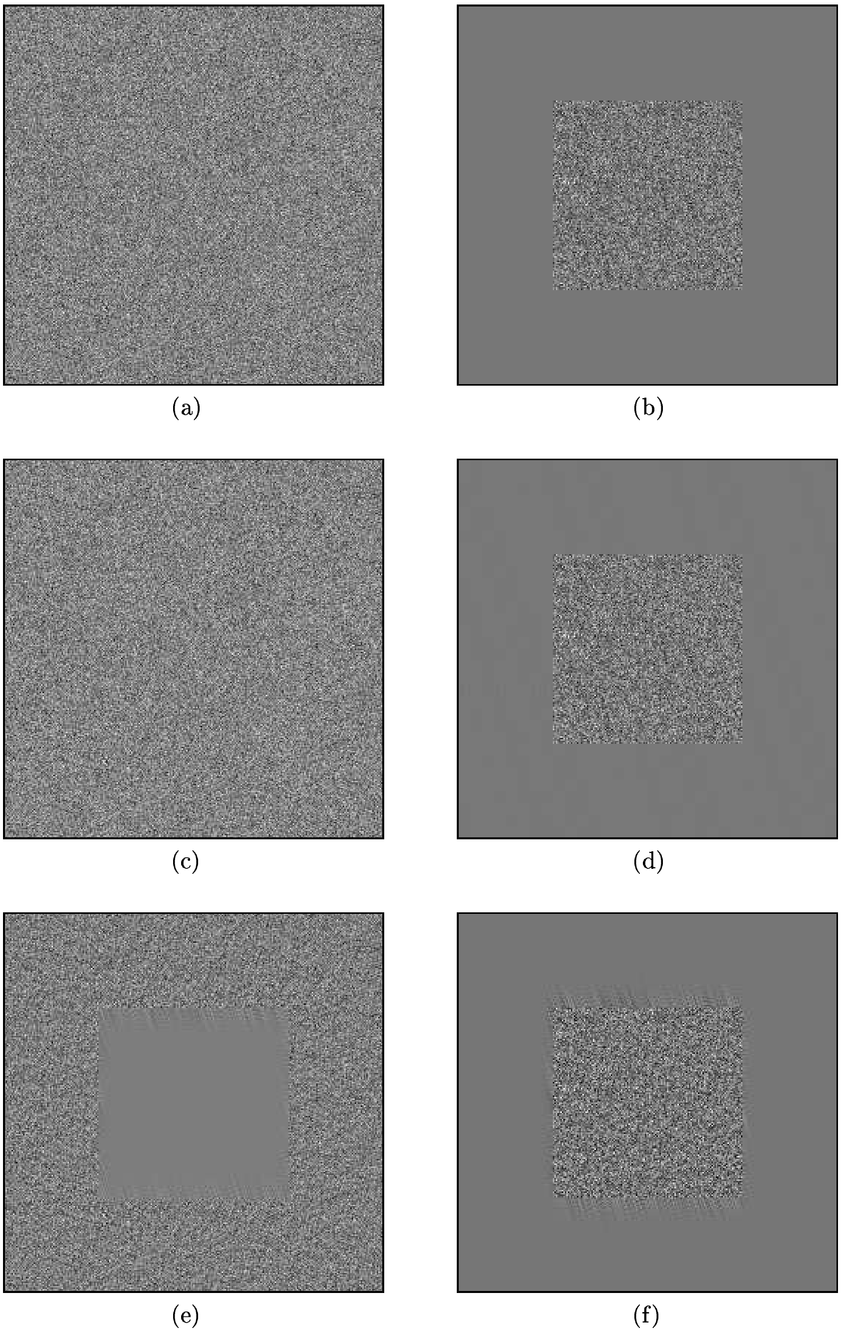


Figure 4.14. Separation and segmentation of Gaussian noise ($\mu = 0$; $\sigma = 10$): (a) and (b) background and foreground at time t_0 translating with velocities (2, 0) and (3, 3) pixels/frame, respectively; (c) and (d) separation of the additively-generated composite images; (e) and (f) segmentation of the occlusion-based composite images.

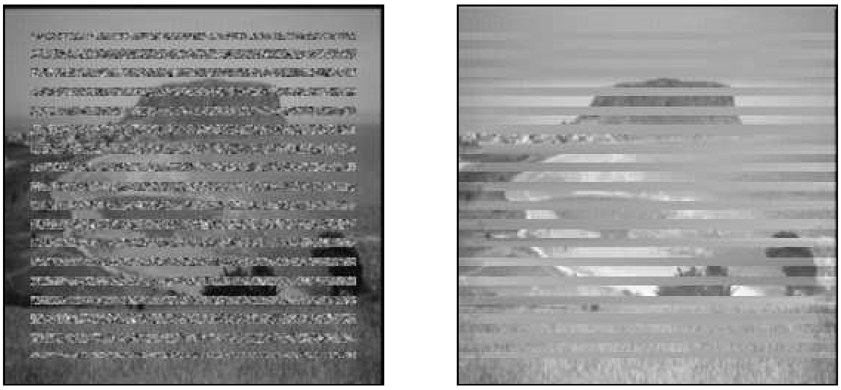


Figure 4.15. 'Picket Fence' test scenario. (a) The background seaside scene occluded by a foreground comprising a series of randomly textured horizontal bars (foreground and background velocities are (3, 0) and (1, 0) pixels/frame, respectively.) (b) The result of segmenting the sequence using the occlusion model; the threshold on attenuation $\tau = 0.01$.

alternative foreground where change in the occluded background is very significant (again, we use the same the foreground and background velocities of (3, 0) and (1, 0) pixels/frame, respectively). Here, the direction of motion is orthogonal to the major axis of the bars. Note that we require a larger gap between the bars to ensure that the background signal does not change completely over the four-image sequence. The occlusion model (here with the usual threshold on attenuation $\tau = 0.15$) shows reasonable segmentation but artifacts due to the change in the occluded signal are evident, despite the attenuation of frequencies which exhibit approximately the same phase changes. In this case, we can remedy the situation by automatically identifying the background pixels and by using them as a mask to gate (remove) the foreground image. Background pixels can be very easily identified since these show up as uniform regions in the segmented foreground image. We have simply used the gradient magnitude of the foreground image to label the pixels: those less than a small threshold are deemed to belong to the background. Note that, in order to maximise the effectiveness of this technique, we have used a threshold on attenuation $\tau = 1.0$ to ensure that all occlusion artifacts have been removed.

Finally, it is worth observing that the masking approach can be used with all of the foregoing results if the quality of the segmented images which have been reconstructed from the resolved Fourier components is deemed to be of an insufficient standard. For example, Figure 4.17 shows

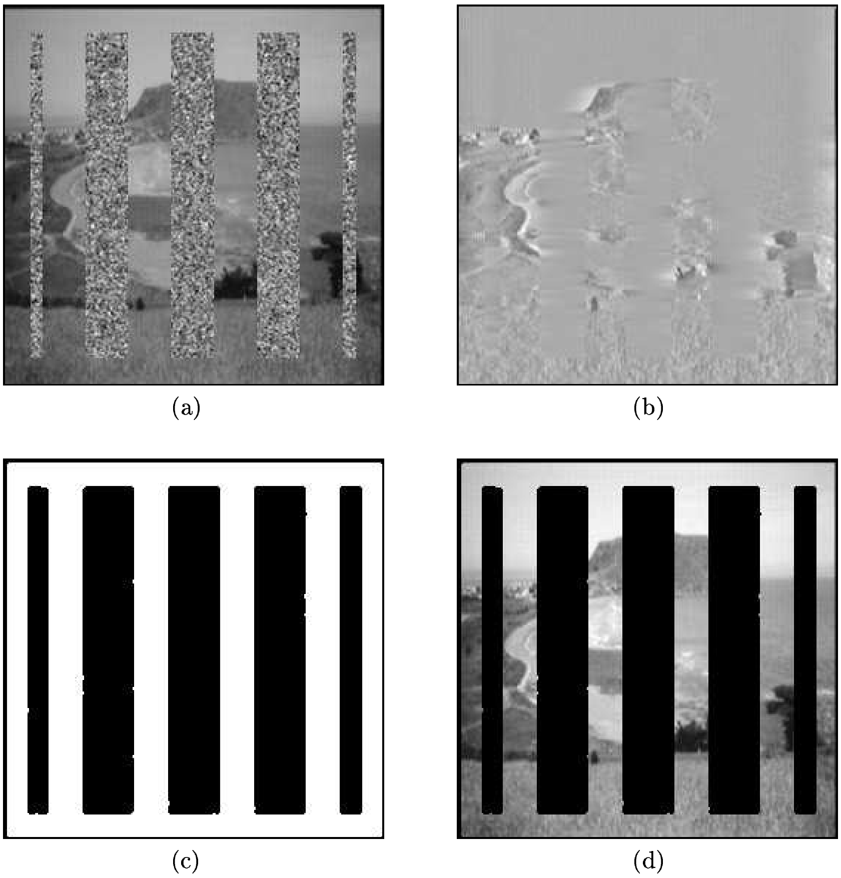
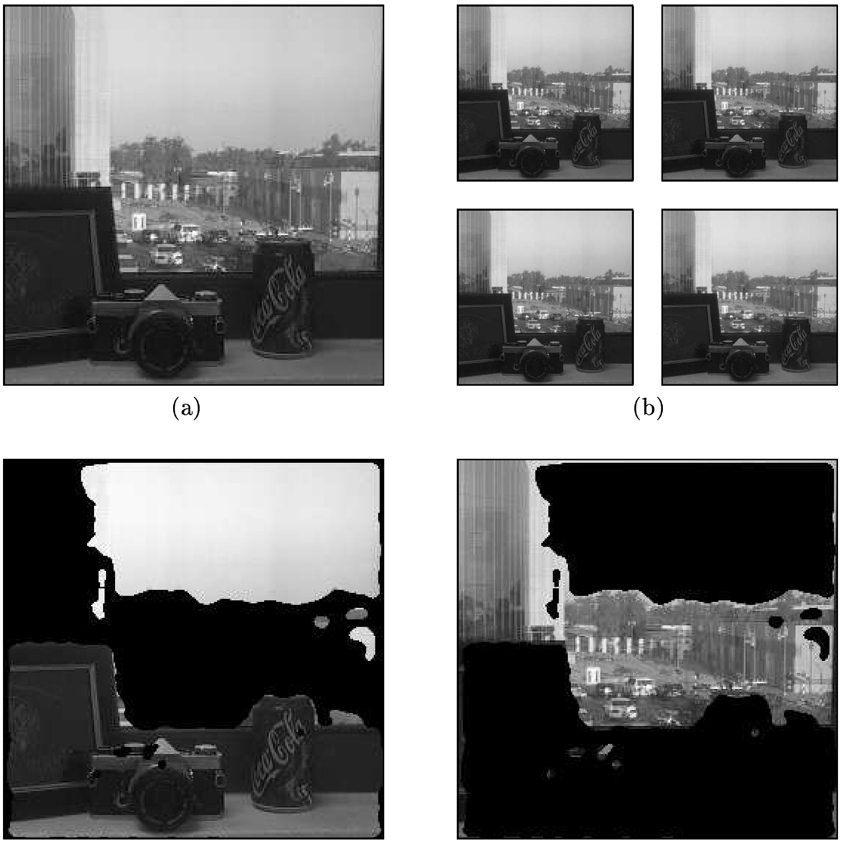


Figure 4.16. ‘Picket Fence’ test scenario. (a) An alternative foreground resulting in significant change in the occluded background. (b) The result of segmenting the sequence using the occlusion model ($\tau = 0.15$); artifacts due to the change in the occluded signal are evident. (c) Mask generated by isolating uniform regions in the segmented foreground when all the artifacts have removed ($\tau = 1.0$). (d) The segmented background computed by gating with the foreground mask.

the results of applying this masking technique to the window scene in Figure 4.12.

5. OUTSTANDING ISSUES

Although the results of the technique are more than satisfactory, the approach presented so far does suffer from a number of constraints concerning the number of independent motions exhibited in the scene and the nature of those motions. We have seen that we can overcome these to an extent using the ‘masking’ approach shown in the previous section.



(a) (b)

(c) (d)

Figure 4.17. Example of figure/ground segmentation in real scenes: a sequence of four 256 × 256 pixel images $f_{t_j}(x, y)$ at times t_0, t_1, t_2, and t_3 is shown in (b) with the image at time t_0 shown in (a) at full resolution. The computed figure and ground images are shown in (c) and (d).

However, before finishing up, it's worth taking a little time to consider how we might tackle the problem in a more general way.

5.1 THE THREE-OBJECT MODEL

In this book, we deal only with figure-ground segmentation and the solutions presented are restricted to this scenario. However, there is nothing in the theory which requires this restriction and it is possible to extend the model to deal with, for example, the segmentation of three objects moving with distinct velocities. In this case, we need a system of six equations of the same form as equation 3.10 to 3.13 and a six-image sequence, rather than a four-image sequence. The solution of this system of equations is left as an exercise for the author!

5.2 COMPLEX MOTION

The approach that has been presented assumes pure image/object translation. Many situations, at least to a first approximation, satisfy these conditions, especially if the sampling period is small enough. In general, however, objects may exhibit a more complex motion, especially if they are deformable or if they are comprised of multiple linked parts, such as a person running. So, what can we do to deal with these situations? Is the approach of any value in the general case? The answer is 'potentially yes', if we adopt an active approach the image acquisition and employ a focus of attention technique, windowing (or apodizing) the area of interest, typically using a Gaussian function. Again, a sustantiation of this claim is left as an exercise for future work although it's worth remarking that we have used exactly this approach in Chapter 8 to deal with the computation of multiple optical flow vectors in the vicinity of occluding boundaries.

Having completed this presentation of the Fourier segmentation technique, it is probably appropriate to finish up by considering very briefly the alternatives for motion segmentation.

6. A SAMPLE OF MOTION/STEREO SEGMENTATION TECHNIQUES

The computer vision literature is replete with various techniques for segmentation using optical flow and stereopsis. In this section, we take a cursory look at some representative approaches.

Our first example is due to Santos-Victor and Sandini [291, 292] who use the normal component of an optical flow field to estimate the affine transformation approximating the motion of the ground-plane on which a camera-bearing robot is travelling; this allows the imaged optical flow field to be inversely-projected onto the ground plane yielding a very regular flow-field. Obstacles lying on the ground-plane can then be easily detected by deviations away from this expected flow-field.

Wang and Adelson [382] have developed an affine-clustering technique for motion segmentation which estimates the affine motion parameters for a set of non-overlapping rectangular regions based on a dense optical flow field. Affine parameter sets which are considered sufficiently reliable are then clustered into a small number of classes, following which each flow vector is assigned to one of these classes, thus accomplishing the segmentation.

Borshukov *et al.* [47] have introduced a multi-pass variation on this approach which, instead of clustering the reliable affine parameter sets, finds the dominant affine motion, assigns the matching flow vectors to that class, removes them from further consideration, and then, in exactly the same way, identifies the next dominant motion and labels vectors accordingly. In effect, and allowing for isolated regions, this continues until all flow vectors (and, hence, image points) are classified.

Motion segmentation techniques which exploit optical flow are strongly dependent on the accuracy and validity of the flow field. Most optical flow estimation algorithms assume a single motion in the local region in which the flow is being estimated. This can cause significant errors in a number of important situations, such as at motion boundaries and in transparent and reflective scenes. Bergen *et al.* [39] have addressed this problem directly and they have introduced a technique which can compute two motions in a single neighbourhood. This is achieved by assuming an initial estimate of one motion, using this to warp or register three images in an image sequence to null that component, forming two new images representing the difference between the three 'registered' images, and then applying a conventional single-motion optical flow technique to the two difference images to compute the second motion. New difference images are then formed from images registered with the second component, based on which a more accurate estimate of the first motion is computed. This iterative process continues until successive estimates converge (typically within a few iterations). Whilst this approach works well, it does not effect any explicit segmentation of the two patterns which give rise to the motion in the first place although, as the authors point out, it would provide a powerful component of a subsequent segmentation process.

Rather than first estimating the optical flow field and then performing the segmentation, Chang *et al.* [72] have proposed a Bayesian strategy for simultaneously estimating the segmentation and the optical flow of the image sequence. This is accomplished by considering the flow field to be the sum of a parametric (eight-parameter bilinear or six-parameter affine) motion – which is interdependent with the segmentation – and a residual field. The approach proceeds by iteratively maximising the *a posteriori* probability of the parametric and residual flow fields and the segmentation mapping.

Recently, Irani and Anadan [157] have proposed a strategy for the integration of techniques for the detection of moving objects imaged by a non-stationary camera to facilitate the analysis of scenes of incrementally-increasing complexity (ranging from approximately planar 2-D scenes to 3-D scenes exhibiting 3-D parallax). The key idea is that independently moving objects can be detected by determining local misalignments of two images which have been registered to compensate for the camera motion and the 3-D parallax. In the 2-D case, the dominant motion due to camera movement is ultimately modelled as an eight-parameter quadratic transformation. This is then extended to encompass 3-D scenes which admit the modelling of camera-induced motion by a number of distinct 2-D parametric models, each corresponding to a distinct (depth) layer in the 3-D scene. In more complex situations, where such an assumption of multi-layered scenes is not valid, the 3-D parallax is explicitly modelled by identifying the epipolar geometry after compensation for the camera rotation and zoom. Independent object motion is detected by identifying regions which violate the epipolar image motion. Images exhibiting both sparse and dense 3-D parallax are catered for.

Other representative approaches to motion segmentation may be found in, for example, [6, 235, 346, 224, 158, 354] and in the many other references listed in the bibliography at the end of the book.

Before proceeding, we note that in recent years the signal processing community has also addressed the problem of signal separation from a slightly different perspective: that of blind systems identification and, in particular, that of blind signal separation. When one is concerned with the separation of additive (mixed) source signals impinging on a phased array of sensors (typically antennae), the problem is often referred to as blind beamforming which can be tackled using both deterministic algebraic methods [365, 278] or statistically using classical approaches [68] or neural networks [14]. Phased array algebraic techniques typically only deal with modulated narrow-band signals (where the time delay in arrival of the signal results in a phase shift of the base-band signal), whilst statistical techniques require that the signals are statistically-independent.

Chapter 5

ARTICULATED BINOCULAR VISION

1. MOTIVATION

The previous two chapters have shown that, in order to accomplish figure/ground segmentation based on the relative motion between objects in a scene, four distinct images are required to provide sufficient constraints on the four 'unknowns' of foreground, background, and their respective velocities or displacements. This is true irrespective of whether the relative object motion arises from independent movement of the objects or from camera motion. Essentially, a *quadocular* configuration is required, *i.e.* we need four images acquired at four equally-spaced positions x_0, x_1, x_2, x_3. If we are dealing with a translating camera rather than four distinct fixed cameras, then $x_i = vt_i$ where v is the camera velocity and t_i represents the time at which image i is acquired. Often, a binocular configuration is desirable. The question which arises then is: Can we effect image segmentation if we have two samples from each of two cameras? That is, can we combine spatial and temporal sampling in a unified spatio-temporal framework? Intuitively, it would seem that we can and the purpose of this chapter is to show formally how this can be achieved.

In Chapter 6, we will show how we can go even further and how, in certain circumstances, we can in fact accomplish figure/ground segmentation with just *two* images acquired either from a binocular camera configuration with a small inter-camera baseline or by translating a single camera by a small amount. Even though this development depends on a restricted form of movement of the camera and is therefore less general than the formulation presented in this chapter, it is in some senses more useful since it requires less information to effect segmentation. So, with this in mind, we will present the theoretical development of articulated binocular vision in this chapter without empirical validation and then proceed directly to the two-image scenario in the next chapter.

2. OVERVIEW

Our goal here is to show that a moving binocular stereoscopic camera configuration is indeed capable of yielding sufficient information to effect figure-ground image segmentation if we have two temporal samples. As we shall see, the approach depends on the use of both fixation and vergence, and it provides a sound theoretical basis for understanding why both are of practical use in constraining the segmentation problem.

We will begin by identifying all of the possible constraint equations which arise when we sample the scene in two dimensions: one spatial and one temporal (*i.e.*, left and right cameras at two different times). Thus, we have two types of displacement: spatial and temporal. Note that temporal displacement may or may not involve a spatial movement of the binocular camera configuration. For example, if we are observing a dynamic scene, we simply require two samples a two different times without any camera movement. On the other hand, if we are observing a static scene, then the camera configuration must be articulated and samples will be acquired at two different spatial positions and, of course, at two different times. Once the constraint equations which capture the model have been set up, the next step is to remove the linear dependence between them and we end up with four simultaneous equations with eight unknowns. Adding more equations (by moving the head and making more observations) doesn't help as every equation introduces a new unknown. This leads us to the need to consider an alternative way of constraining the problem. The solution is through the use of vergence and fixation. If we assume that the foreground object remains static as we articulate the camera head, *i.e.*, that we fixate on it during motion, and if we assume also that the motion parallax of the foreground object is zero, *i.e.*, that vergence angle of the two cameras is appropriately set, then we can remove four of the unknowns. This results in an exactly determined system of equations, the solution of which yields the Fourier components of the foreground and background objects and their spatial and temporal displacements.

3. THEORETICAL FRAMEWORK

To begin with, let us establish some notation. Since we are using two sampling dimensions, spatial and temporal, we will use two trailing subscripts to denote the particular sample. Thus, the image acquired at time $t = 0$ by the right camera in an articulated binocular configuration will be written $\mathsf{F}_{x_r t_0}$. Similarly, we have $\mathsf{F}_{x_l t_0}$, $\mathsf{F}_{x_l t_1}$, and $\mathsf{F}_{x_r t_1}$ for the left images at time $t = 0, 1$ and the right image at time $t = 1$. The notation for the phase change must identify whether the shift is due to a spatial binocular dispacement of the left and right cameras, or due to a temporal displacement.[1] We use a trailing subscript to denote this. In addition, we must identify the time at which the spatial displacement takes place or, *vice versa*, the spatial position at which the temporal displacement takes place. We denote this by appropriately subscripting the subscript. Thus, the phase change due to camera motion (*i.e.* temporal sampling) for the left camera is written $\Delta\Phi_{t_{x_l}}$. Similarly, $\Delta\Phi_{t_{x_r}}$ denotes the phase change due to right camera motion, and $\Delta\Phi_{x_{t_0}}$ and $\Delta\Phi_{x_{t_1}}$ denote the phase change due to change in viewing perspectives between left and right cameras (*i.e.* spatial sampling) at times $t = 0, 1$, respectively.

Next, we adapt equations 3.10 and 3.11 to deal with the four distinct cases of the motion of the left and right camera and stereopsis at time $t = 0$ and time $t = 1$.

Motion of the left camera is given by

$$\begin{aligned} \mathsf{F}_{x_l t_0} &= \mathsf{F}^1_{x_l t_0} + \mathsf{F}^2_{x_l t_0} && (5.1) \\ \mathsf{F}_{x_l t_1} &= \mathsf{F}^1_{x_l t_0} \Delta\Phi^1_{t_{x_l}} + \mathsf{F}^2_{x_l t_0} \Delta\Phi^2_{t_{x_l}} && (5.2) \end{aligned}$$

Motion of the right camera is given by

$$\begin{aligned} \mathsf{F}_{x_r t_0} &= \mathsf{F}^1_{x_r t_0} + \mathsf{F}^2_{x_r t_0} && (5.3) \\ \mathsf{F}_{x_r t_1} &= \mathsf{F}^1_{x_r t_0} \Delta\Phi^1_{t_{x_r}} + \mathsf{F}^2_{x_r t_0} \Delta\Phi^2_{t_{x_r}} && (5.4) \end{aligned}$$

Binocular stereopsis at time $t = 0$ is given by

$$\begin{aligned} \mathsf{F}_{x_l t_0} &= \mathsf{F}^1_{x_l t_0} + \mathsf{F}^2_{x_l t_0} && (5.5) \\ \mathsf{F}_{x_r t_0} &= \mathsf{F}^1_{x_l t_0} \Delta\Phi^1_{x_{t_0}} + \mathsf{F}^2_{x_l t_0} \Delta\Phi^2_{x_{t_0}} && (5.6) \end{aligned}$$

Binocular stereopsis at time $t = 1$ is given by

$$\begin{aligned} \mathsf{F}_{x_l t_1} &= \mathsf{F}^1_{x_l t_1} + \mathsf{F}^2_{x_l t_1} && (5.7) \\ \mathsf{F}_{x_r t_1} &= \mathsf{F}^1_{x_l t_1} \Delta\Phi^1_{x_{t_1}} + \mathsf{F}^2_{x_l t_1} \Delta\Phi^2_{x_{t_1}} && (5.8) \end{aligned}$$

[1] The term *spatial displacement* refers to the binocular samples at the left and right cameras, and the term *temporal displacement* refers to the sampling in time, *i.e.*, at times t_0 and t_1. Temporal displacement may or may not involve a spatial movement of the binocular camera configuration. In particular, if we are observing a dynamic scene, we simply require two samples at times t_0 and t_1, without any camera movement. On the other hand, if we are observing a static scene, then the camera configuration must be articulated and samples at t_0 and t_1 will be acquired at two different (spatial) positions.

Apparently, we have fourteen unknowns and eight constraint equations. However there is significant linear dependence between these equations and there are in fact only four linearly-independent equations since we have only four independent observations. This becomes explicit when we note the following equivalences between images:

$$\mathsf{F}^1_{x_l t_1} = \mathsf{F}^1_{x_l t_0} \Delta\Phi^1_{t_{x_l}} \tag{5.9}$$

$$\mathsf{F}^2_{x_l t_1} = \mathsf{F}^2_{x_l t_0} \Delta\Phi^2_{t_{x_l}} \tag{5.10}$$

$$\mathsf{F}^1_{x_r t_0} = \mathsf{F}^1_{x_l t_0} \Delta\Phi^1_{x_{t_0}} \tag{5.11}$$

$$\mathsf{F}^2_{x_r t_0} = \mathsf{F}^2_{x_l t_0} \Delta\Phi^2_{x_{t_0}} \tag{5.12}$$

Combining equations 5.1 – 5.8 and 5.9 – 5.12, we get the following:

$$\mathsf{F}_{x_l t_0} = \mathsf{F}^1_{x_l t_0} + \mathsf{F}^2_{x_l t_0} \tag{5.13}$$

$$\mathsf{F}_{x_l t_1} = \mathsf{F}^1_{x_l t_0} \Delta\Phi^1_{t_{x_l}} + \mathsf{F}^2_{x_l t_0} \Delta\Phi^2_{t_{x_l}} \tag{5.14}$$

$$\mathsf{F}_{x_r t_0} = \mathsf{F}^1_{x_l t_0} \Delta\Phi^1_{x_{t_0}} + \mathsf{F}^2_{x_l t_0} \Delta\Phi^2_{x_{t_0}} \tag{5.15}$$

$$\mathsf{F}_{x_r t_1} = \mathsf{F}^1_{x_l t_0} \Delta\Phi^1_{x_{t_0}} \Delta\Phi^1_{t_{x_r}} + \mathsf{F}^2_{x_l t_0} \Delta\Phi^2_{x_{t_0}} \Delta\Phi^2_{t_{x_r}} \tag{5.16}$$

We now have four constraints and eight unknowns. In order to determine the system exactly, we need to reduce the number of unknowns by four. We do this by assuming vergence and fixation. That is, we assume that the principal rays of the stereoscopic configuration converge on a single point of attention on the surface of one of the objects to be segmented (vergence) and that they remain fixed on this point as the camera configuration moves or as the object moves (fixation). These two assumptions then imply that all the phase changes associated with object 1 (say) are unity, *i.e.* object 1 does not translate as a result of either the spatial sampling (it has zero disparity) or as a result of the camera motion (it has zero velocity). Thus, we have:

$$\Delta\Phi^1_{x_{t_0}} = \Delta\Phi^1_{x_{t_1}} = 1 \tag{5.17}$$

$$\Delta\Phi^1_{t_{x_l}} = \Delta\Phi^1_{t_{x_r}} = 1 \tag{5.18}$$

Since $\Delta\Phi^1_{x_{t_1}}$ does not appear in equations 5.13 - 5.16, we have thus reduced the number of unknowns by only three and we need one further restriction. We achieve this by assuming that the translation of the background occluded image due to the motion of the stereoscopic configuration is the same for both the left and right cameras:

$$\Delta\Phi^2_{t_{x_l}} = \Delta\Phi^2_{t_{x_r}} \tag{5.19}$$

This will be a reasonable approximation if the camera motion is small and the background is not very close; it will be exactly true if the direction of motion is orthogonal to the stereo baseline (or, more correctly, if it is normal to the plane containing the principal rays). We now have an exactly determined system of equations:

$$\begin{align}
\mathsf{F}_{x_l t_0} &= \mathsf{F}^1_{x_l t_0} + \mathsf{F}^2_{x_l t_0} \tag{5.20}\\
\mathsf{F}_{x_l t_1} &= \mathsf{F}^1_{x_l t_0} + \mathsf{F}^2_{x_l t_0} \Delta\Phi^2_{t_{x_l}} \tag{5.21}\\
\mathsf{F}_{x_r t_0} &= \mathsf{F}^1_{x_l t_0} + \mathsf{F}^2_{x_l t_0} \Delta\Phi^2_{x_{t_0}} \tag{5.22}\\
\mathsf{F}_{x_r t_1} &= \mathsf{F}^1_{x_l t_0} + \mathsf{F}^2_{x_l t_0} \Delta\Phi^2_{x_{t_0}} \Delta\Phi^2_{t_{x_l}} \tag{5.23}
\end{align}$$

Equations 5.20 - 5.23 for $\mathsf{F}^1_{x_l t_0}$, $\mathsf{F}^2_{x_l t_0}$, $\Delta\Phi^2_{t_{x_l}}$, and $\Delta\Phi^2_{x_{t_0}}$ have the following solution:

$$\begin{align}
\Delta\Phi^2_{t_{x_l}} &= \frac{\mathsf{F}_{x_l t_1} - \mathsf{F}_{x_r t_1}}{\mathsf{F}_{x_l t_0} - \mathsf{F}_{x_r t_0}} \tag{5.24}\\
\Delta\Phi^2_{x_{t_0}} &= \frac{\mathsf{F}_{x_r t_0} - \mathsf{F}_{x_r t_1}}{\mathsf{F}_{x_l t_0} - \mathsf{F}_{x_l t_1}} \tag{5.25}\\
\mathsf{F}^2 x_l t_0 &= \frac{\mathsf{F}_{x_l t_1} - \mathsf{F}_{x_l t_0}}{\Delta\Phi^2_{t_{x_l}} - 1} \tag{5.26}\\
\mathsf{F}^1 x_l t_0 &= \mathsf{F}_{x_l t_0} - \mathsf{F}^2 x_l t_0 \tag{5.27}
\end{align}$$

We adopt exactly the same approach to the computation of these phase changes and Fourier components as we did in the previous chapters, with the small difference in this case that, for situations involving occlusion, we attenuate frequencies which satisfy the following condition:

$$\left|\Delta\Phi^2_{t_{x_l}} - 1\right| \leq \tau d \tag{5.28}$$

where τ is some specified toleralance and d is the magnitude of the displacement exhibited by the occluded background image due to the articulated (temporal) shift of the binocular system, *i.e.* it is the displacement corresponding to $\Delta\Phi^2_{t_{x_l}}$.

3.1 EFFECTING VERGENCE AND FIXATION

In the foregoing, we assumed that the vergence of the binocular stereo configuration on a point of interest and the fixation of the configuration on that point over time. How then are vergence and fixation to be effected? There are two issues:

1. The selection of the fixation point, *i.e.* the point of intersection of the principal rays;

2. The achievement of camera vergence.

The first issue belongs to the domain of pre-attentive vision: it is an area of much active and historical research, within the computer, the neurophysiological, and the psychological vision communities, and lies outside the scope of the work described in this book.

On the other hand, the second issue — achievement of camera vergence and fixation — can be dealt with using techniques which are directly related to the theory being developed here. If we make the assumption that we can extract an image sub-region[2] surrounding, and centred on, the camera centre through which passes the principal ray, then clearly we can also estimate the phase change $\Delta\Phi$ of spatial frequencies in this region. Since a phase change of zero implies that the image shift is zero — and assuming that otherwise the spectral content is identical — then vergence and fixation are achieved exactly when the phase change between the Fourier components of a centre sub-region of spatially- and temporally-successive image is minimized.

Alternatively, we can identify the vergence and fixation angles simply by finding the angle which minimises the correlation between the decoupled images (*i.e.* which maximises the decoupling). We can implement this directly in the Fourier domain by forming the product of the decoupled images and choosing the configuration which gives the minimal product. We note in passing that, if we know the binocular baseline distance, we need search for only one angle since the vergence angle and the fixation angle are related.

[2] In order to minimize 'edge effects', the region of interest will typically be an apodized window, of some appropriate dimension, surrounding the camera centre.

Chapter 6

FRONTO-PARALLEL BINOCULAR VISION

1. FORMULATION OF THE PROBLEM

In the previous chapters, we discussed how one can segment or decouple foreground and background objects in both occluding and transparent additive images using monocular and articulated binocular configurations. In both instances, we required four images to achieve the segmentation:

- four images at times t_0, t_1, t_2, and t_3, respectively in the case of the monocular configuration;

- the left and right images at times t_0 and t_1 in the case of the articulated binocular configuration.

We saw that the need for four images arises directly from the formulation of the problem itself. We have four unknowns: the two objects and their respective velocities (or their displacements in the case of multiocular vision). Alternatively, in terms of the Fourier theory which we have been discussing, the four unknowns are the Fourier components of each of the two objects and their corresponding phase changes:

- monocular: $\mathsf{F}^1_{t_0}, \mathsf{F}^2_{t_0}, \Delta\Phi^1, \Delta\Phi^2$

- articulated binocular: $\mathsf{F}^1_{x_l t_0}, \mathsf{F}^2_{x_l t_0}, \Delta\Phi^2_{t_{x_l}}, \Delta\Phi^2_{x_{t_0}}$

It would be very appealing to solve the segmentation problem with just two images but, on the face of it, this doesn't seem possible. Since we have four unknowns, we need four constraints and, hence, four images and there doesn't seem to be much we can do to change this. But it turns out that in fact there is and, in certain circumstances, we can accomplish the segmentation with just two images.

In all of the foregoing techniques, we treated each spatial frequency independently. That is, we solved for each of the four unknown quantities *independently* at each spatial frequency (ω_x, ω_y) and then we sorted the Fourier components into two distinct sets (corresponding to the two distinct objects) on the basis of their respective phase changes. We did this knowing that the phase change for each constituent image or object varies regularly with spatial frequency. In fact, as we saw, it lies on a plane through the origin in (ω_x, ω_y) space and the plane is determined by the velocities of the two objects or images. Using the solved, but unsorted, phase changes as input, we identify the two velocities or displacements using a Hough transform based on the known relationship between phase change and spatial frequency. These computed velocities then form the basis for the final process of assigning the solved Fourier components into one or other of the distinct object-specific sets.

The key point to note here is that, in the approach we have adopted so far, we use the phase-change/spatial-frequency relationship to sort the components *after* we have solved for them. In this chapter, we will show that it is possible to solve the figure-ground segmentation problem with just two binocular images by using the phase-change/spatial-frequency relationship *to compute the values of the two unknown phase-changes* as well as using it as we did before to sort the Fourier phasors once we have computed them. That is, we use the regularity of the phase-change/spatial-frequency relationship to identify two parts of the solution. This reduces the number of required constraints from four to two and, hence, we can accomplish the segmentation with two binocular images.

We need to make one simple assumption to allow us to develop a solvable model: that the binocular stereo configuration has a vergence angle of zero degrees, *i.e.* that both cameras are pointing in the same direction so that their optical axes are parallel. This is often referred to as fronto-parallel binocular stereo and is one of the most common and simple binocular stereo configurations. Why do we make this assumption? Recall the relationship between phase change, velocity, and spatial frequency:

$$\delta\phi(\omega_x, \omega_y) = -(v_x^i \omega_x \delta t + v_y^i \omega_y \delta t) \tag{6.1}$$

with spatial frequencies (ω_x, ω_y) as the independent variables. In this case, we are dealing with the actual image velocity due to real motion of the image/object over time. If we are dealing with the apparent velocity due to the parallax associated with camera displacement in a binocular configuration, we substitute the δx for δt and write the relationship as

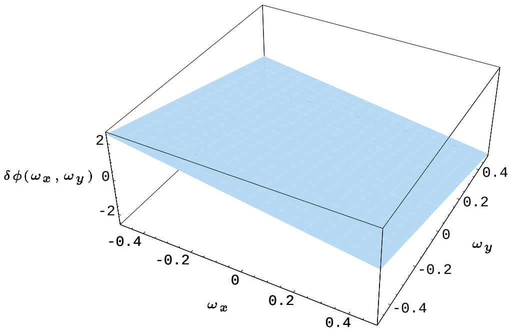


Figure 6.1. The variation of phase change $\delta\phi(\omega_x, \omega_y)$ with spatial frequency (ω_x, ω_y) is defined by the plane equation $\delta\phi(\omega_x, \omega_y) = -(v_x^i \omega_x \delta x + v_y^i \omega_y \delta x)$. The attitude of the plane is determined by the velocity components (v_x, v_y). Note that the plane passes through the origin $(\omega_x = 0, \omega_y = 0)$.

follows:

$$\delta\phi(\omega_x, \omega_y) = -(v_x^i \omega_x \delta x + v_y^i \omega_y \delta x) \tag{6.2}$$

Both of these equations define a plane through the origin where the orientation of the plane is defined by the velocity (or displacement) components v_x and v_y (see Figure 6.1).

Now, in the case of a fronto-parallel stereo configuration, the epipolar lines are parallel to the line joining the optical centres of the two images. That is, the displacement of each pixel in the image is horizontal. Thus, the vertical displacement is zero and the plane defining the phase changes is defined by:

$$\delta\phi(\omega_x, \omega_y) = -v_x^i \omega_x \delta x \tag{6.3}$$

Hence, for some displacement v_x and at a given horizontal spatial frequency ω_x the phase changes are identical at every vertical spatial frequency ω_y. Put another way, the phase changes for a given velocity or displacement is a linear function of the x spatial frequency ω_x and *only* the x spatial frequency (see Figure 6.2). We'll see how we can make use this fact in just a moment. Let's return for now to the model of our figure/ground segmentation problem.

2. THE COMPUTATIONAL MODEL

Recall that, with two objects or images, we have:

$$\mathsf{F}_{t_n}(\omega_x, \omega_y) = \mathsf{F}^1_{t_0}(\omega_x, \omega_y)\Big(\Delta\Phi^1(\omega_x, \omega_y)\Big)^n + \mathsf{F}^2_{t_0}(\omega_x, \omega_y)\Big(\Delta\Phi^2(\omega_x, \omega_y)\Big)^n$$

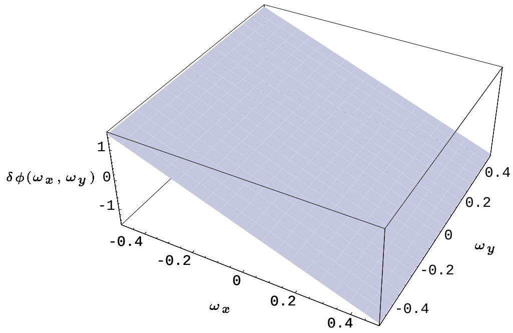


Figure 6.2. The variation of phase change $\delta\phi(\omega_x, \omega_y)$ with spatial frequency (ω_x, ω_y) for a fronto-parallel stereo configuration: the plane is defined by the plane equation $\delta\phi(\omega_x, \omega_y) = -v_x^i \omega_x \delta x$. Note that the phase change is a linear function of only the x spatial frequency ω_x.

where $\mathsf{F}_{t_0}^1(\omega_x, \omega_y)$, $\mathsf{F}_{t_0}^2(\omega_x, \omega_y)$, $\Delta\Phi^1(\omega_x, \omega_y)$, and $\Delta\Phi^2(\omega_x, \omega_y)$ are the Fourier phasors and phase changes of the two constituent objects or images, respectively, and $\mathsf{F}_{t_n}(\omega_x, \omega_y)$ is the resultant phasor at time t_n. That is, the resultant phasor at any given spatial frequency (ω_x, ω_y) is the vectorial sum of two phasors, each rotating with distinct and constant angular velocity. For a fronto-parallel binocular configuration, the angular velocities are determined by the phase changes:

$$\left(\Delta\Phi^i\right)^n = e^{-i\delta\phi^i n} \tag{6.4}$$

$$= e^{-i\omega_x v_x^i n\delta x} \tag{6.5}$$

Consequently, the resultant phasor at some spatial frequency traces out a curve similar to an epicycloid, the shape of which depends on the relative angular velocities (phase changes) of the two constituent phasors and on the magnitudes of the two phasors (see Figure 6.3). Note that the shape of this curve is periodic. Clearly, we will have different curves at different spatial frequencies, simply because the magnitude of the constituent phasors will vary. However, the periodicity of the curve is dependent only on the relative angular velocities: as long as the ratio of the angular velocities is the same, the periodicity will be identical. The number of rotations of the 'outer' phasor completes for every rotation of the 'inner' phasor in Figure 6.4 will be constant and, hence, so will the period of the occurrence of the loops in the resultant be constant. Note that this is true irrespective of the absolute angular velocity of the two

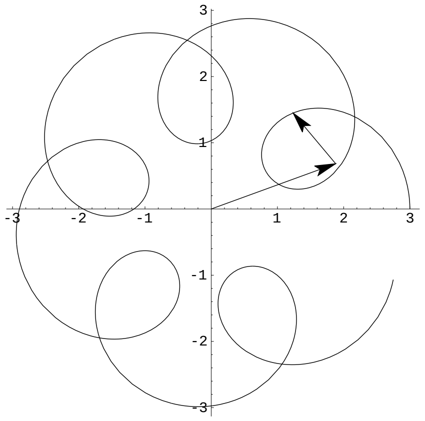


Figure 6.3. The resultant phasor at some spatial frequency traces out a curve, the shape of which depends on the relative angular velocities (phase changes) of the two constituent phasors and on the magnitudes of the two phasors.

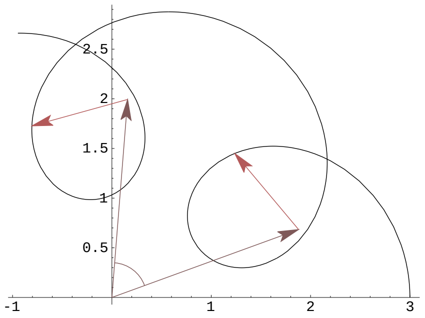


Figure 6.4. The period of the resultant's curve depends only on the relative angular velocities (phase changes) of the two constituent phasors.

phasors. Also, we always get the same periodic looping no matter what the magnitude of the constituent phasors.

Now we come to the critical part. The absolute angular velocities are a linear function of ω_x and, thus, both of the constituent phasors rotate at a rate which is proportional to the x or horizontal spatial frequency. Thus, the geometry of the curve doesn't change with frequency *but the rate at which it is generated does.* Now, we sample the image, *i.e.* the resultant phasor's curves, at some constant rate (*i.e.* δx = constant).

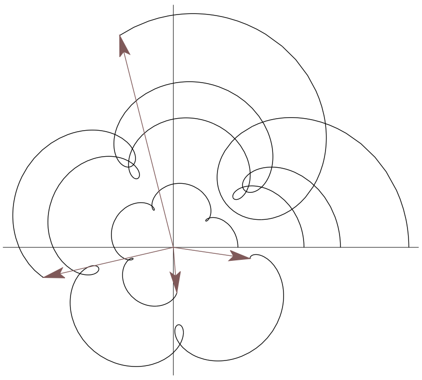


Figure 6.5. The curves generated by a series of exemplar resultants, each at a different spatial frequency ω_x. Note that even though each resultant comprises two constitutent phasors of different magnitudes, all curves display the same periodicity exactly because their relative angular velocities are identical. The phases of the resultants increase with spatial frequency ω_x but they do so non-monotonically.

This means that, although the resultant phasor looping curve retains the same periodicity, the phase or angle between the two sampled resultants will increase (albeit non-monotonically) with increase in horizontal angular frequency ω_x.

Referring to figure 6.5, which depicts the curve traced out by four exemplar resultants, each at a different frequency, we can see that, for some horizontal spatial frequency ω_x:

1. The difference in phase of the resultant will eventually be 2π, *i.e.* the phase of the resultant will wrap (and it will do so periodically). That is, the phase change of the resultant will be zero periodically as a function of ω_x; we will refer to these frequencies as $n\omega_{x_p}, n = 1, 2, 3, \ldots$.

2. Equally, for some other horizontal spatial frequencies, the difference in the magnitude of the resultant will be zero. That is, a zero difference in the resultant magnitude will occur periodically as a function of ω_x; $n\omega_{x_m}, n = 1, 2, 3, \ldots$.

We make use of these two facts to compute the two phase changes of our two constituent Fourier phasors $\Delta\Phi^1$ and $\Delta\Phi^2$ or, equivalently, $\delta\phi^1 = \omega_x v_x^1 \delta x$ and $\delta\phi^2 = \omega_x v_x^2 \delta x$. Specifically, we note that the period at which the phase of the resultant wraps (*i.e.* the phase change of the resultant) is equal to the period of one of the constituent Fourier components.

Equivalently, there exists a set of spatial frequencies $n\omega_{x_p}, n = 1, 2, 3, \ldots$ at which the phase change of the resultant phasor is $2\pi n$. We identify these frequencies empirically by computing the difference in the phase of the resultant for all spatial frequencies ω_x and by isolating those for which the difference equals zero, modulo $2\pi n$. Knowing ω_{x_p}, we can compute the displacement v_x from

$$\delta\phi = 2\pi n = (n\omega_{x_p})v_x\delta x \tag{6.6}$$

Assuming $\delta x = 1$, we have:

$$v_{x_1} = \frac{2\pi}{\omega_{x_p}} \tag{6.7}$$

Note that we assume arbitrarily that the displacement we are computing is v_{x_1}. Of course, knowing v_{x_1}, we can then compute $\Delta\Phi^1(\omega_x, \omega_y)$ — which is what we really want — as before from $\Delta\Phi^1 = e^{-i(v_{x_1}\omega_x\delta x)}$, and we can do so for all spatial frequencies (ω_x, ω_y).

Now, it only remains to identify v_{x_2}. This we do using the *difference* in the magnitude of the resultant.

Referring again to figure 6.5 we note that the period at which the difference in the magnitude of the resultant phasor equals zero is the same as that as the period of the loops in the resultant curve. In turn, and as we saw earlier, the period of these loops is related to the angular velocity of the 'outer' component phasor relative to that of the 'inner' phasor, or, equivalently, it is related to the spatial displacement of one object with respect to the other, *i.e.* $|v_{x_1} - v_{x_2}|$. That is, and in the same vein as before, there exists a set of x spatial frequencies $n\omega_{x_m}, n = 1, 2, 3, \ldots$ at which the difference of the magnitude of the resultant phasor equals zero. Again, we identify these frequencies empirically by computing the difference in the magnitude of the resultant for all spatial frequencies ω_x and isolating those for which the difference equals zero, modulo $2\pi n$. Knowing ω_{x_m}, we can compute the *relative* displacement $|v_{x_1} - v_{x_2}|$ from

$$\delta\phi = 2\pi n = (n\omega_{x_m})\,|v_{x_1} - v_{x_2}|\,\delta x \tag{6.8}$$

Again, assuming $\delta x = 1$, we have

$$|v_{x_1} - v_{x_2}| = \frac{2\pi}{\omega_{x_m}} \tag{6.9}$$

Since we already know v_{x_1}, we can easily compute v_{x_2} and, hence, we can compute $\Delta\Phi^2 = e^{-i(v_{x_2}\omega_x\delta x)}$.

Finally, we note that, since each curve (*i.e.* each frequency-dependent resultant phasor) exhibits the same periodicity at all vertical spatial frequency ω_y for any given horizontal spatial frequency ω_{x_i}, the periodicity of all phasors at frequency (ω_{x_p}, ω_y) is identical. Similarly, the periodicity of all phasors at frequency at ω_{x_m} is identical. We use these facts to eliminate the presence of any degenerate solutions by averaging both the phase changes and the magnitude changes over all ω_y and we find ω_{x_p} and ω_{x_m} by analysing the periodicity of the resulting 1-D signature — *e.g.* see Figures 6.8, 6.10, 6.12, and 6.14 (c) and (d).

3. APPLICATION OF THE TECHNIQUE

The following pages (Figures 6.7 – 6.24) show the result of applying the technique to four data-sets:

1. A synthetic scene comprising two additive images;

2. A synthetic scene comprising two occluding objects;

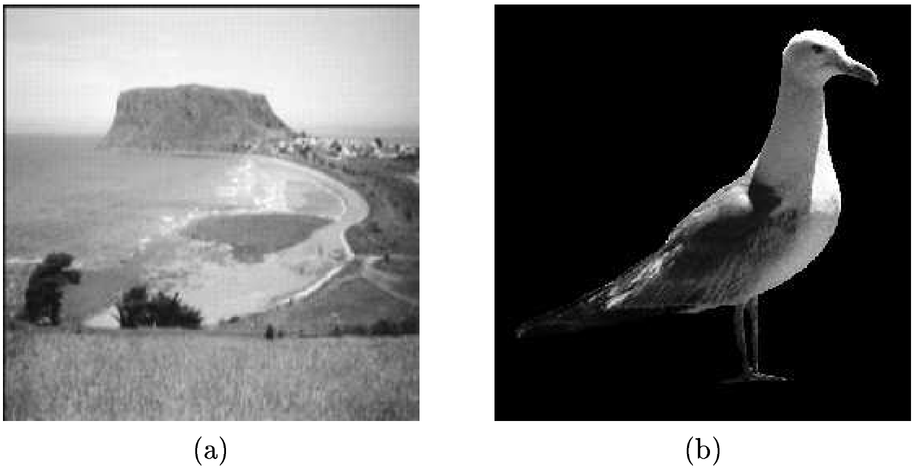
3. A real scene comprising two occluding binocular images.

4. A real scene comprising two additive binocular images.

The technique evidently works as well as the approaches developed in the previous chapters. The need to average in the direction of the vertical spatial frequency ω_y in order to make the periodicity more accessible can be clearly seen in all of the following figures.

4. CAVEAT

In the foregoing, we have managed to achieve a useful simplification of the theory developed in the previous chapters and, in the process, we reduced the number of images or samples required to accomplish the segmentation from four to two. It is tempting to assume that we can apply this two-image approach in general to, for example, the segmentation of scenes with two independently moving objects such as we discussed in Chapters 3 and 4. Unfortunately, we can't apply it, at least not directly. The approach described in this chapter depends on the fact that the angular velocities of both constituent phasors increase linearly with frequency and, furthermore, that the variation is confined to one axis (in this case the x-axis), thereby removing one component of velocity vector (the y component in this case). This then leaves just two independent unknown quantities: the two x components of the velocity of each object. These conditions are satisfied if and only if the velocities of the two images/objects are parallel, which is indeed the case for fronto-parallel binocular vision. That said, it is plausible to extend the approach somewhat, relaxing the assumption that the variation is aligned with the x-axis, by allowing an arbitrary but common axis of



(a) (b)

Figure 6.6. Synthetic test images: (a) background (b) foreground.

variation: in this case, we would need to first find the axis of variation (typically by identifying the orthogonal axis along which there is no variation in phase difference). Unfortunately, the general case where the objects move independently won't yield to this line of attack because the phase differences will vary along every direction.

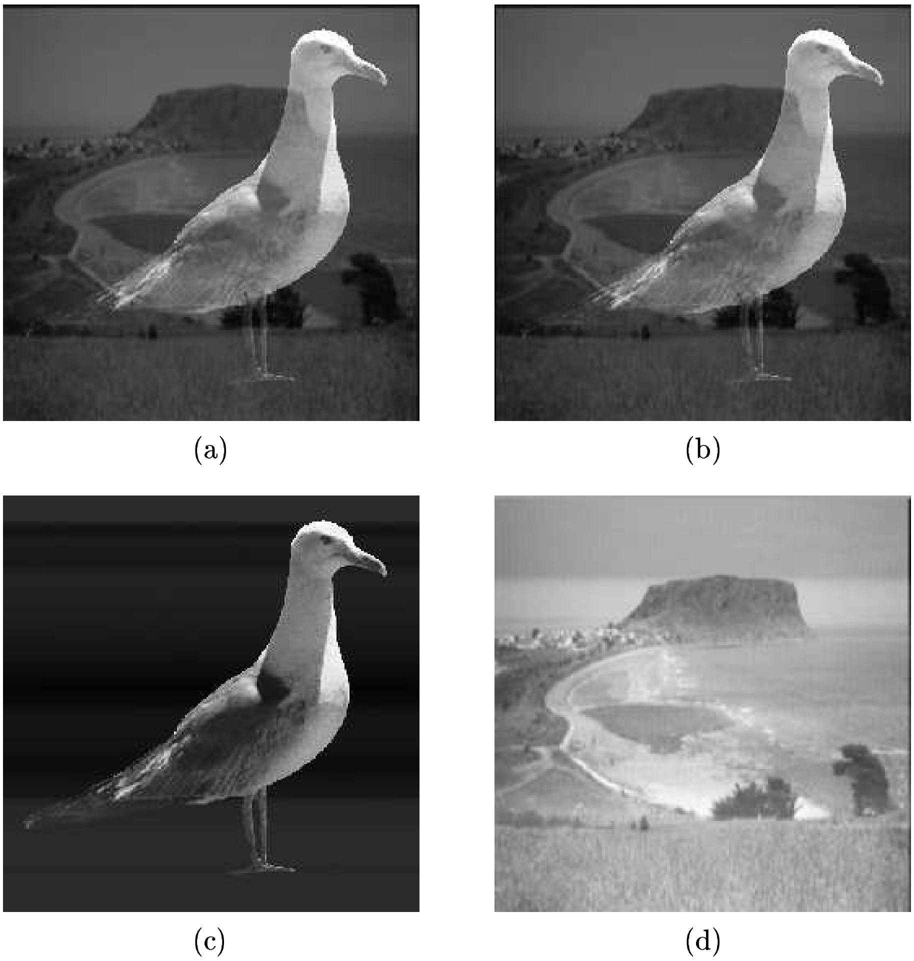
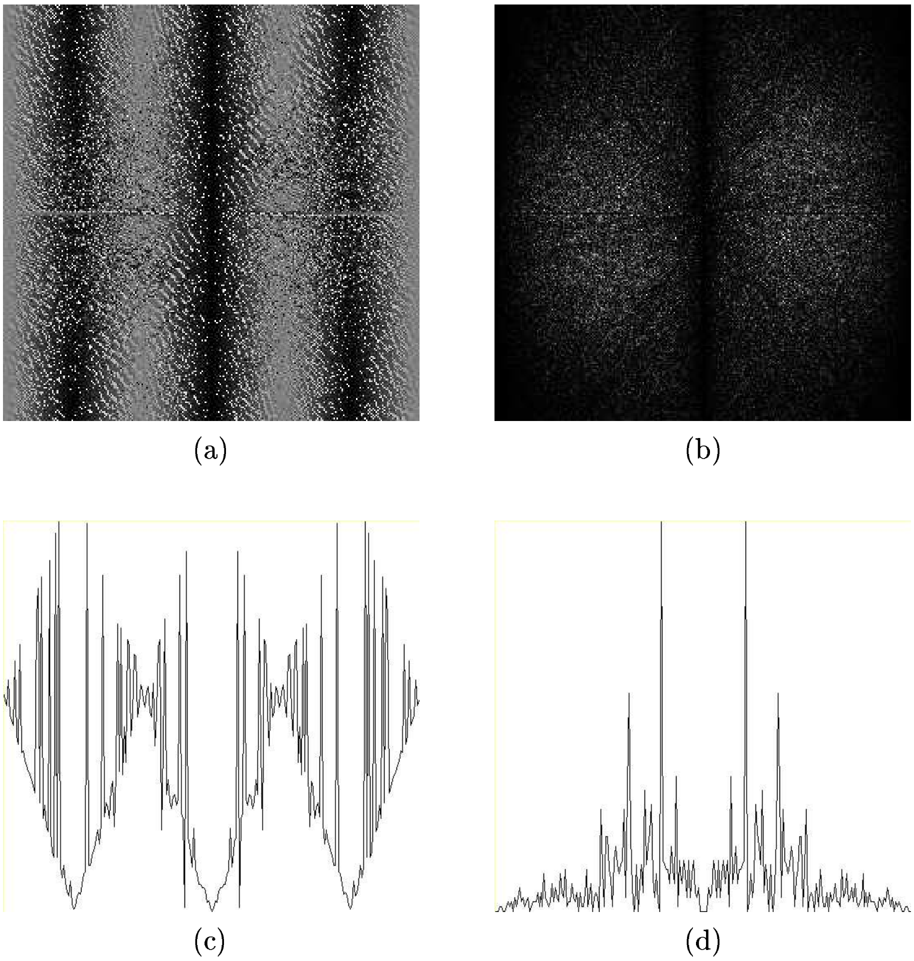


Figure 6.7. Synthetic additive test. (a) Left image. (b) Right image. (c) Segmented foreground. (d) Segmented background. The foreground displacement is 3 pixels and the background displacement is 1 pixel.



(a)

(b)

(c)

(d)

Figure 6.8. Synthetic additive test: computation of displacements from the difference of phases and magnitudes of the resultants. (a) Unprocessed phase difference. (b) Unprocessed magnitude difference. (c) Cross-section through the phase difference image. (d) Cross-section through the magnitude difference image.

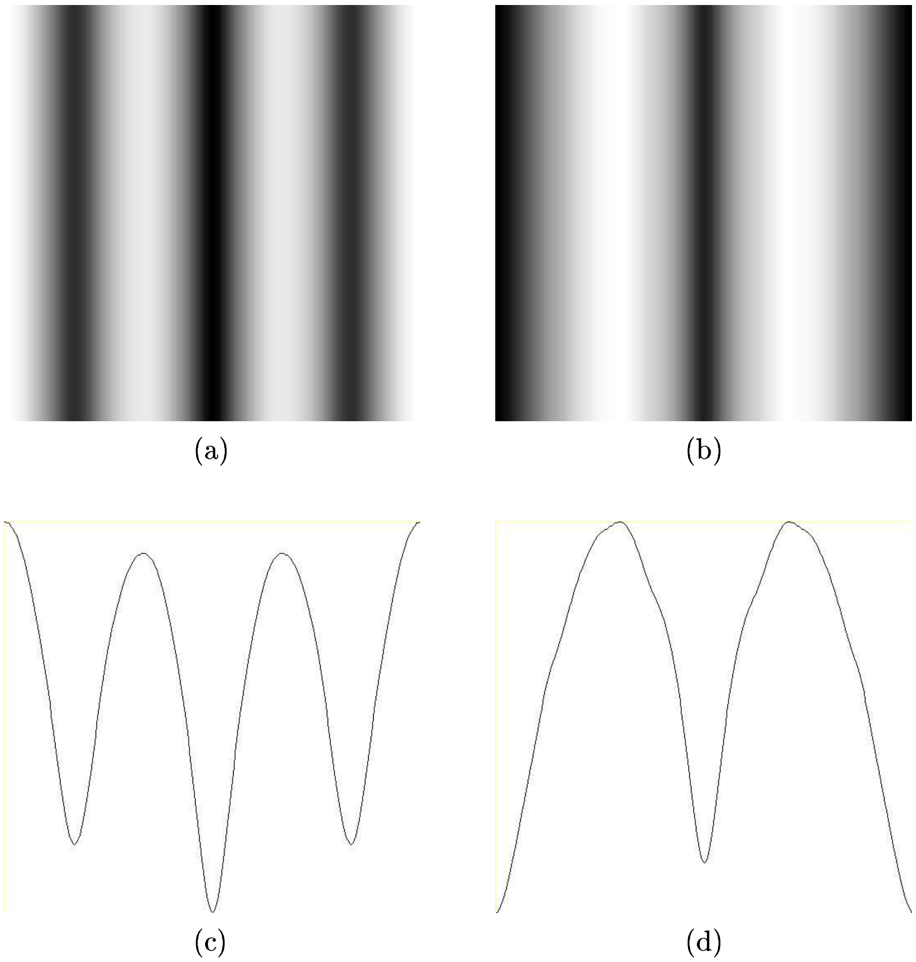
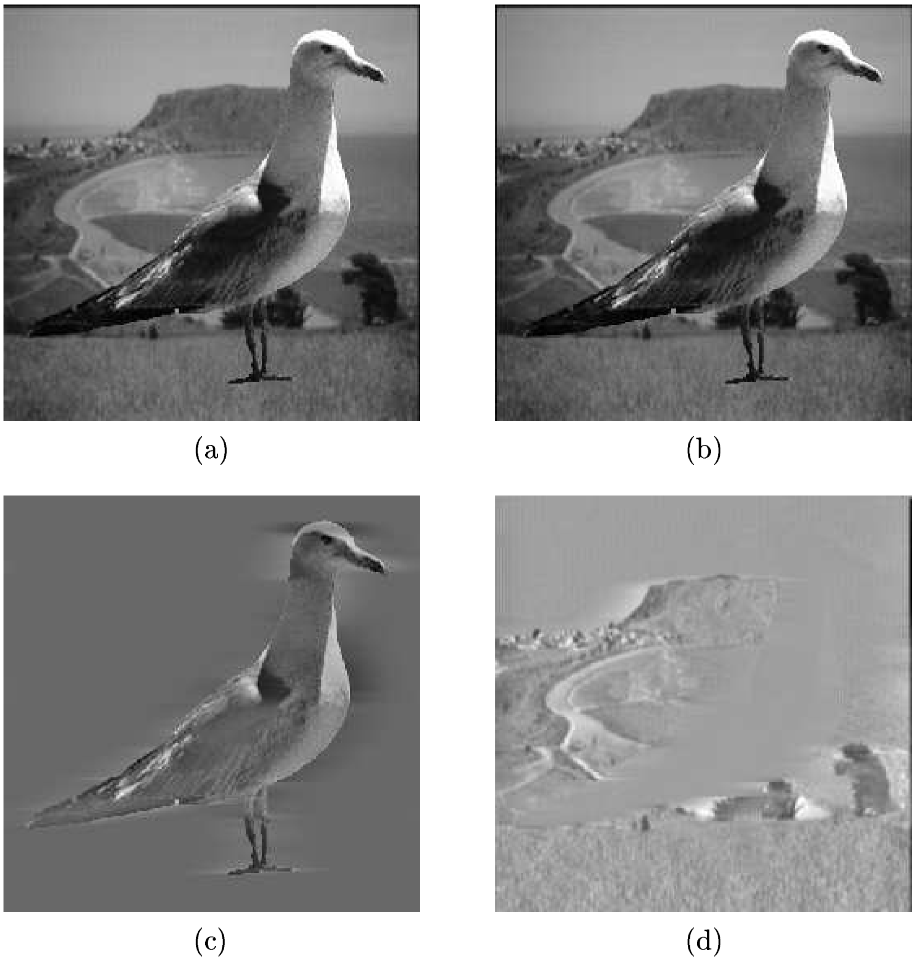


Figure 6.9. Synthetic additive test: computation of displacements from the difference of phases and magnitudes of the resultants. (a) The phase difference averaged in vertical spatial frequency (ω_y) direction; the displacement is computed from $v_{x_1} = \frac{2\pi n}{n\omega_{x_p}}$ where $n\omega_{x_p}$ are the frequencies at which the phase difference equals zero — these occur at the periodic mimima. (b) The averaged magnitude difference; the displacement is computed from $|v_{x_1} - v_{x_2}| = \frac{2\pi n}{\omega_{x_m}}$ where $n\omega_{x_m}$ are the frequencies at which the magnitude difference equals zero — again, these occur at the periodic mimima. (c) Cross-section through the phase difference image. (d) Cross-section through the magnitude difference image.



(a)

(b)

(c)

(d)

Figure 6.10. Synthetic occlusion test. (a) Left image. (b) Right image. (c) Segmented foreground. (d) Segmented background. The foreground displacement is 3 pixels and the background diplacement is 1 pixel.

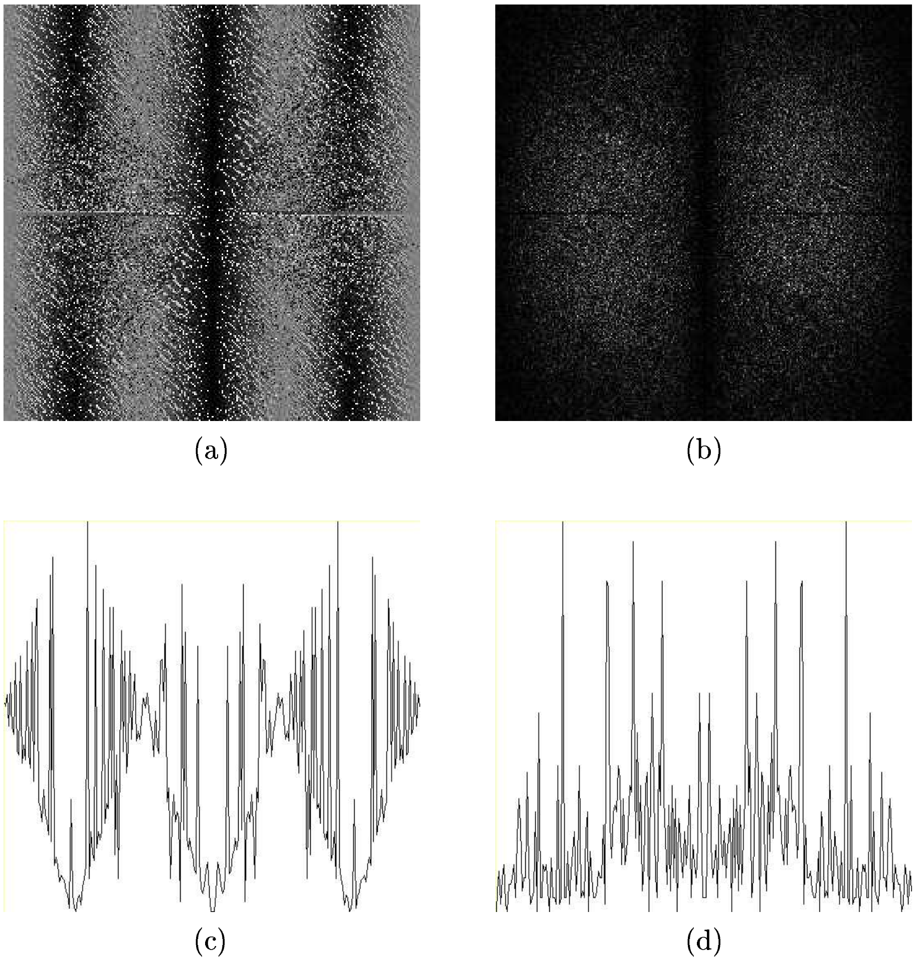
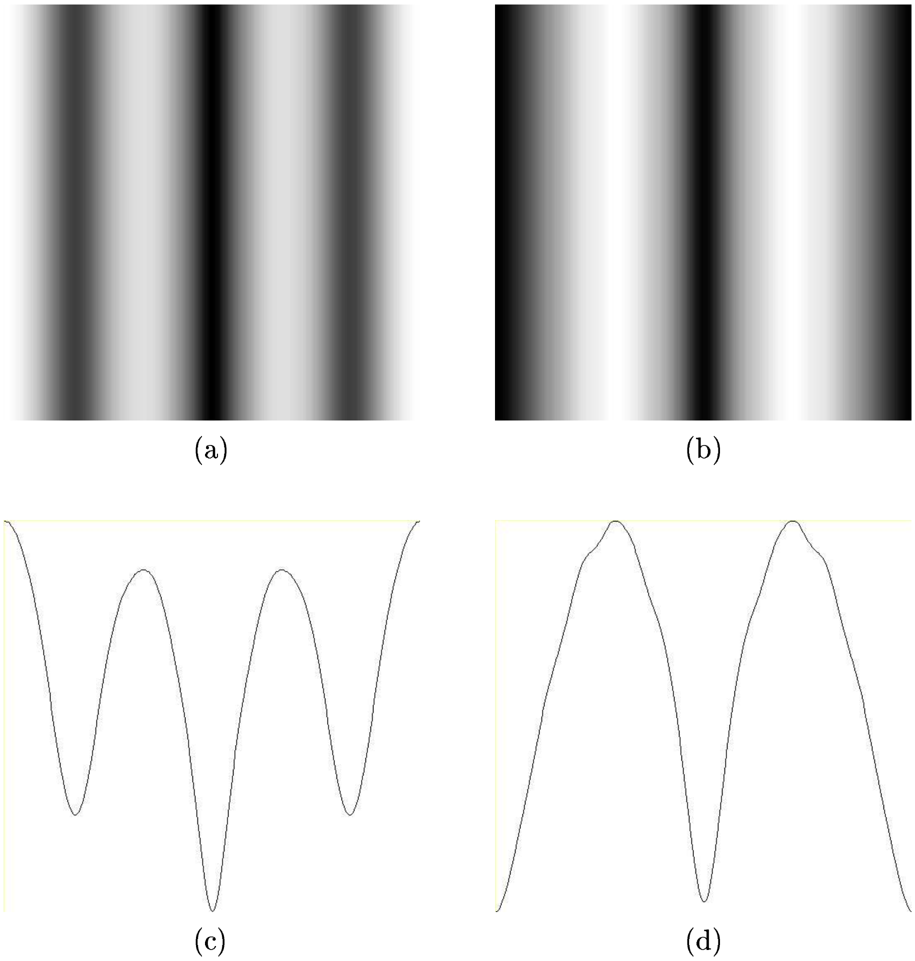


Figure 6.11. Synthetic occlusion test: computation of displacements from the difference of phases and magnitudes of the resultants. (a) Unprocessed phase difference. (b) Unprocessed magnitude difference. (c) Cross-section through the phase difference image (d) Cross-section through the magnitude difference image.



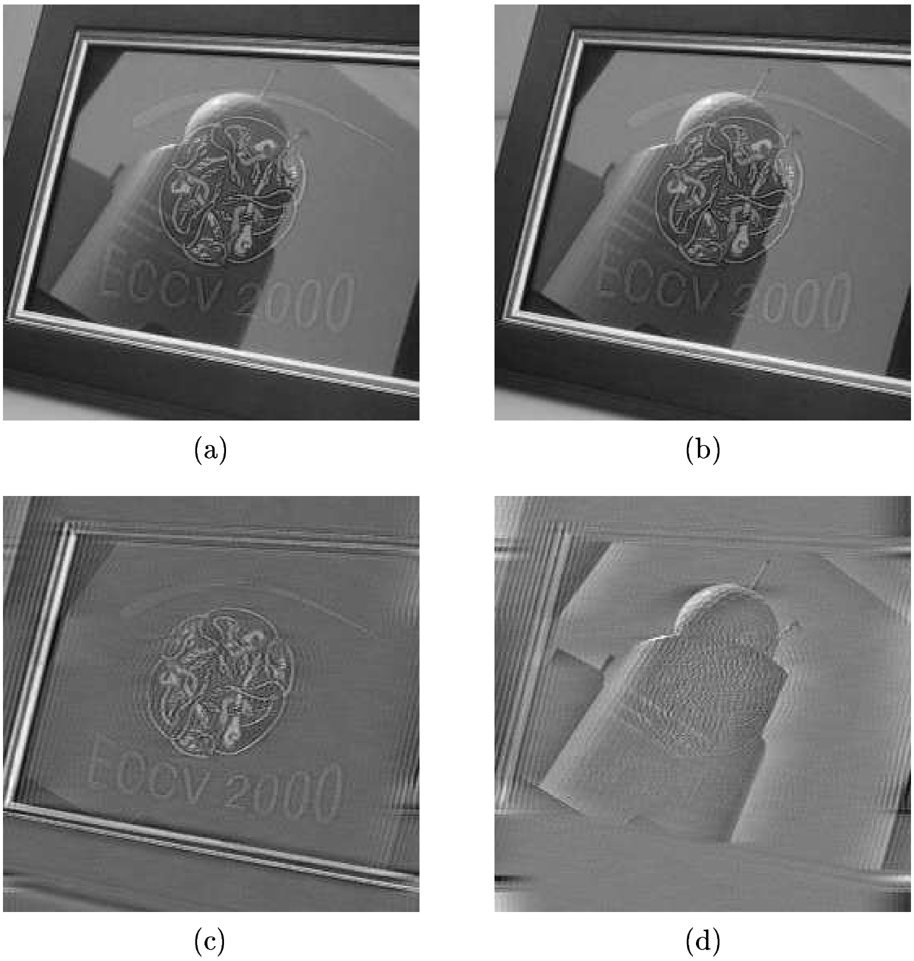
(a)

(b)

(c)

(d)

Figure 6.12. Synthetic occlusion test: computation of displacements from the difference of phases and magnitudes of the resultants. (a) The phase difference averaged in vertical spatial frequency (ω_y) direction; the displacement is computed from $v_{x_1} = \frac{2\pi n}{n\omega_{x_p}}$ where $n\omega_{x_p}$ are the frequencies at which the phase difference equals zero — these occur at the periodic mimima. (b) The averaged magnitude difference; the displacement is computed from $|v_{x_1} - v_{x_2}| = \frac{2\pi n}{\omega_{x_m}}$ where $n\omega_{x_m}$ are the frequencies at which the magnitude difference equals zero — again, these occur at the periodic mimima. (c) Cross-section through the phase difference image. (d) Cross-section through the magnitude difference image.



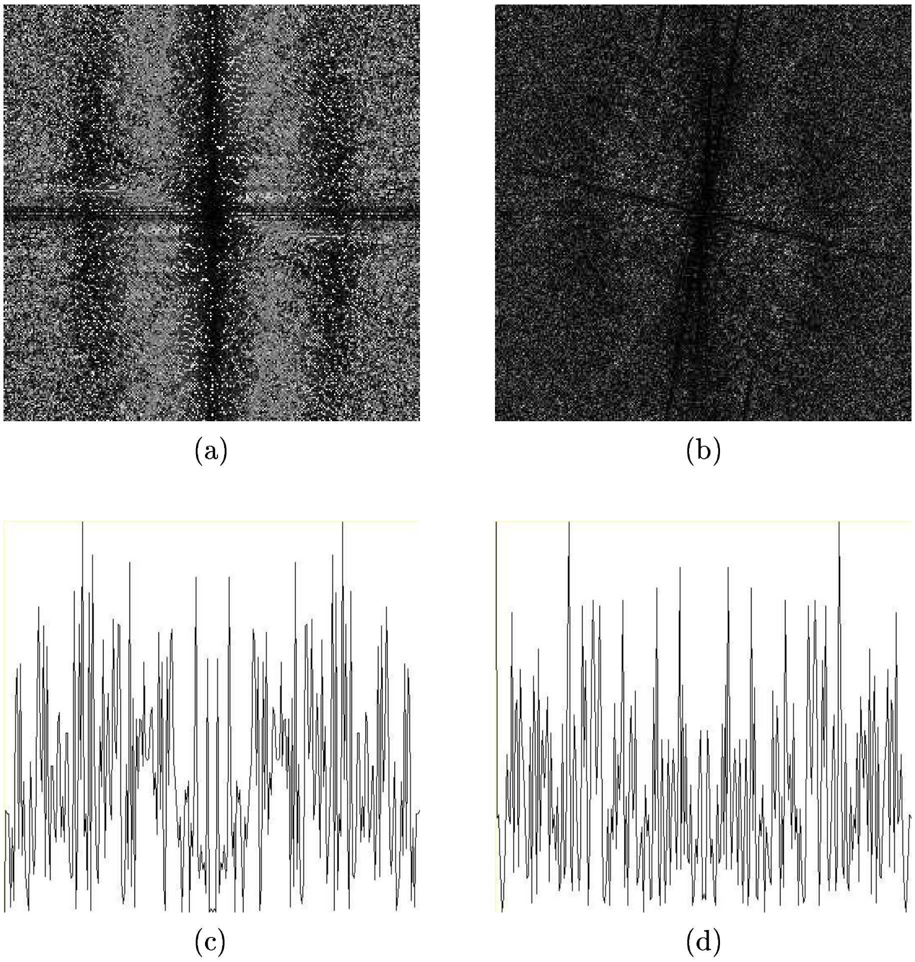
(a)

(b)

(c)

(d)

Figure 6.13. Real additive test. (a) Left image. (b) Right image. (c) Segmented foreground. (d) Segmented background.



(a)

(b)

(c)

(d)

Figure 6.14. Real additive test: computation of displacements from the difference of phases and magnitudes of the resultants. (a) Unprocessed phase difference. (b) Unprocessed magnitude difference. (c) Cross-section trhough the phase difference image. (d) Cross-section through the magnitude difference image.

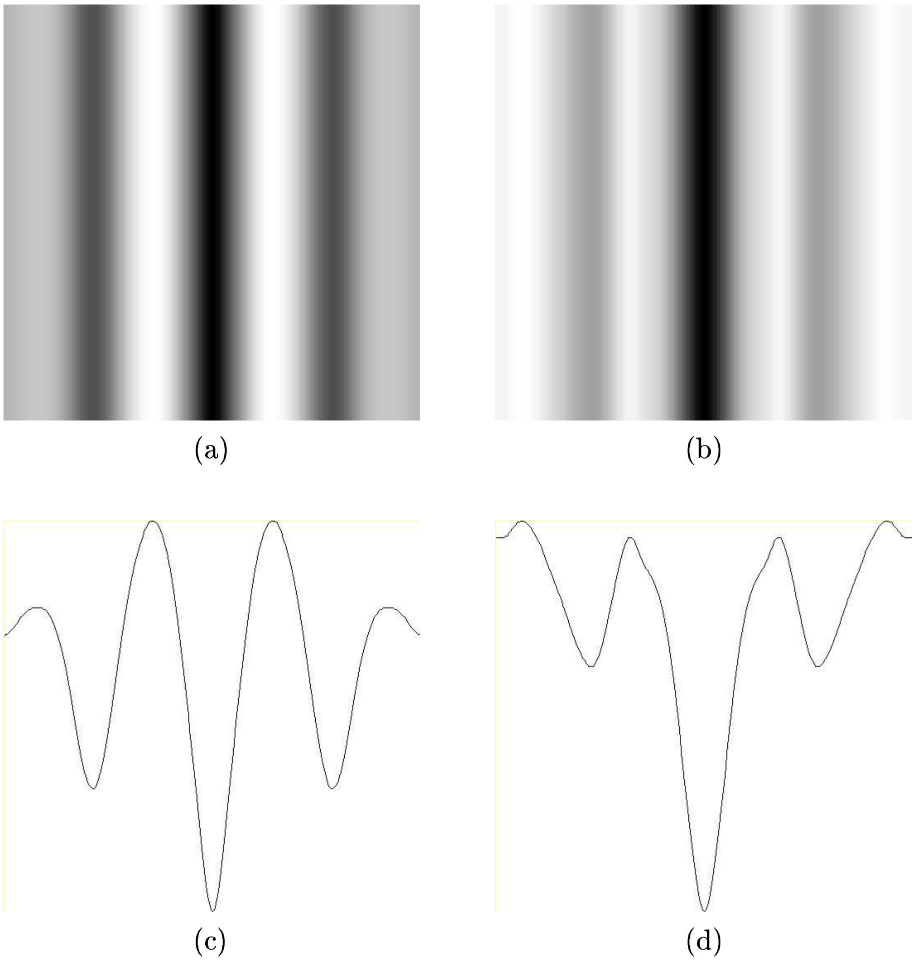


Figure 6.15. Real additive test: computation of displacements from the difference of phases and magnitudes of the resultants. (a) The phase difference averaged in vertical spatial frequency (ω_y) direction; the displacement is computed from $v_{x_1} = \frac{2\pi n}{n\omega_{x_p}}$ where $n\omega_{x_p}$ are the frequencies at which the phase difference equals zero — these occur at the periodic mimima. (b) The averaged magnitude difference; the displacement is computed from $|v_{x_1} - v_{x_2}| = \frac{2\pi n}{\omega_{x_m}}$ where $n\omega_{x_m}$ are the frequencies at which the magnitude difference equals zero — again, these occur at the periodic mimima. (c) Cross-section through the phase difference image. (d) Cross-section through the magnitude difference image.

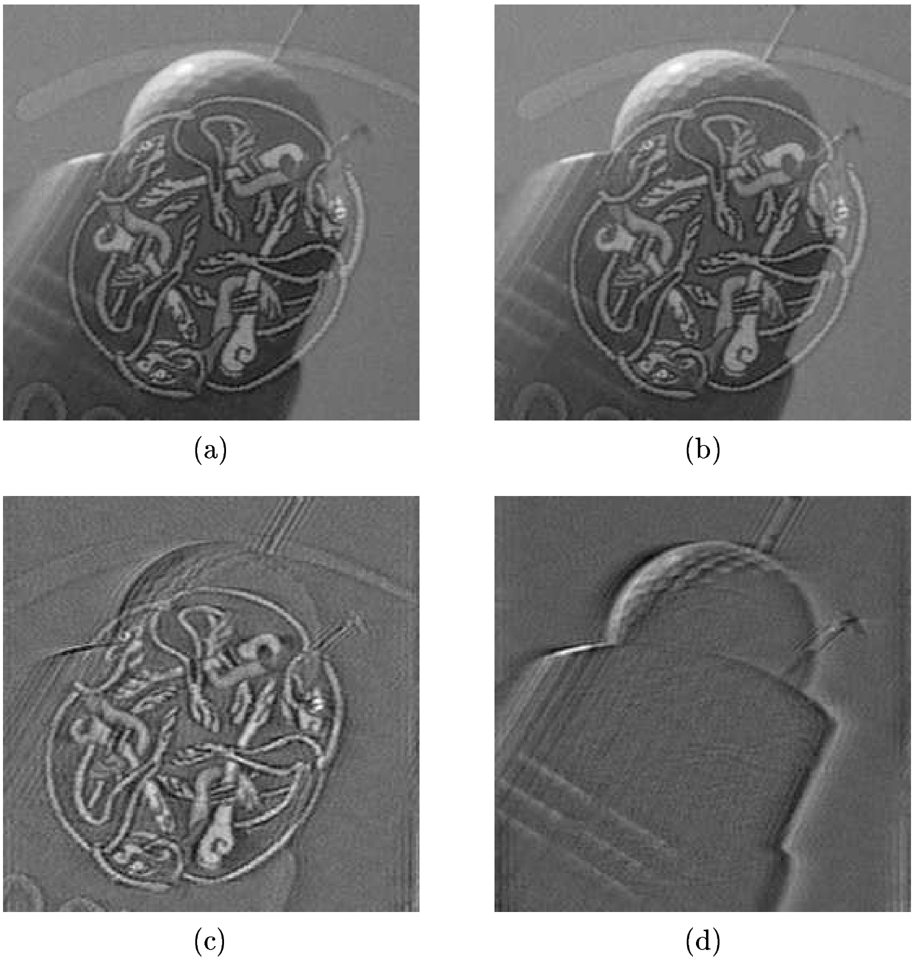
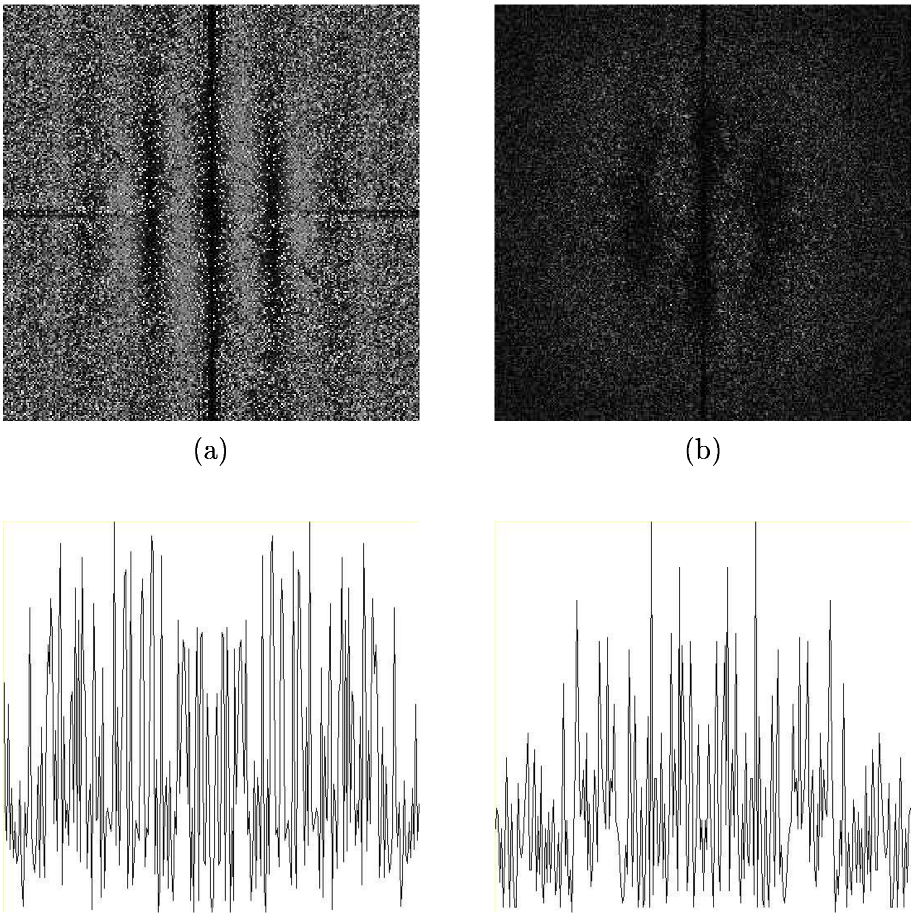


Figure 6.16. Real additive test. (a) Left image. (b) Right image. (c) Segmented foreground. (d) Segmented background.



(a) (b)

(c) (d)

Figure 6.17. Real additive test: computation of displacements from the difference of phases and magnitudes of the resultants. (a) Unprocessed phase difference. (b) Unprocessed magnitude difference. (c) Cross-section trhough the phase difference image. (d) Cross-section through the magnitude difference image.

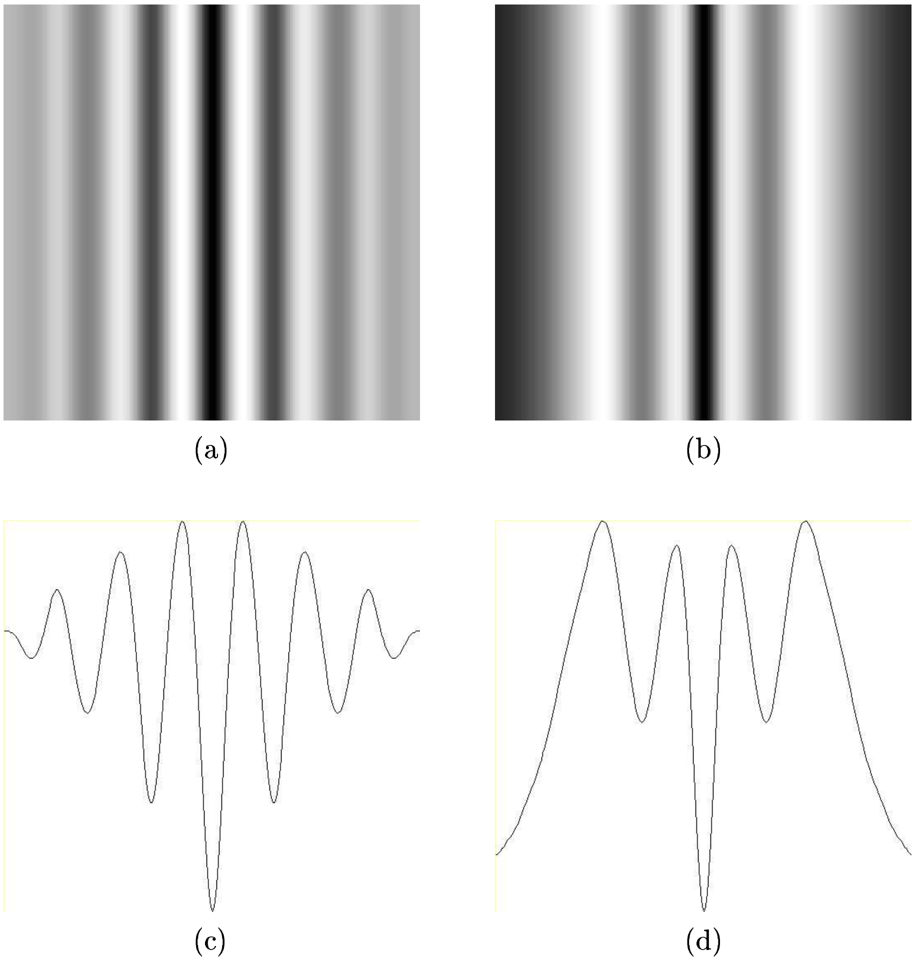
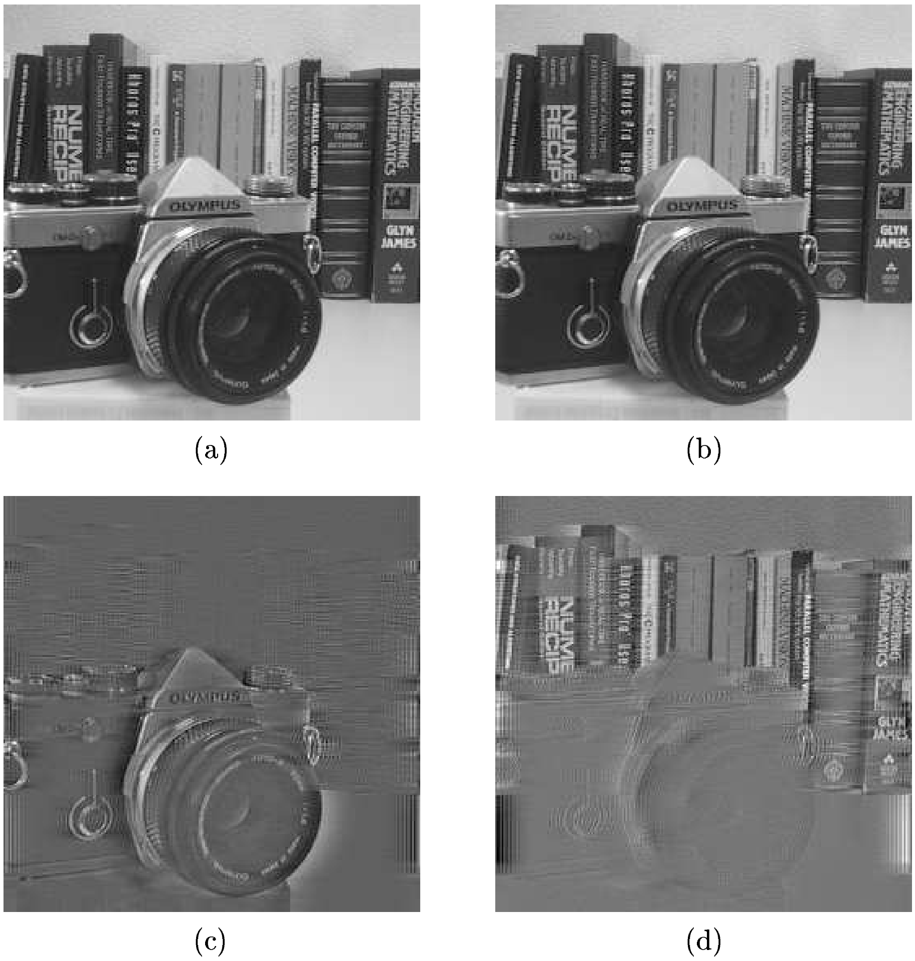


Figure 6.18. Real additive test: computation of displacements from the difference of phases and magnitudes of the resultants. (a) The phase difference averaged in vertical spatial frequency (ω_y) direction; the displacement is computed from $v_{x_1} = \frac{2\pi n}{n\omega_{x_p}}$ where $n\omega_{x_p}$ are the frequencies at which the phase difference equals zero — these occur at the periodic mimima. (b) The averaged magnitude difference; the displacement is computed from $|v_{x_1} - v_{x_2}| = \frac{2\pi n}{\omega_{x_m}}$ where $n\omega_{x_m}$ are the frequencies at which the magnitude difference equals zero — again, these occur at the periodic mimima. (c) Cross-section through the phase difference image. (d) Cross-section through the magnitude difference image.



(a)

(b)

(c)

(d)

Figure 6.19. Real occlusion test. (a) Left image. (b) Right image. (c) Segmented foreground. (d) Segmented background.

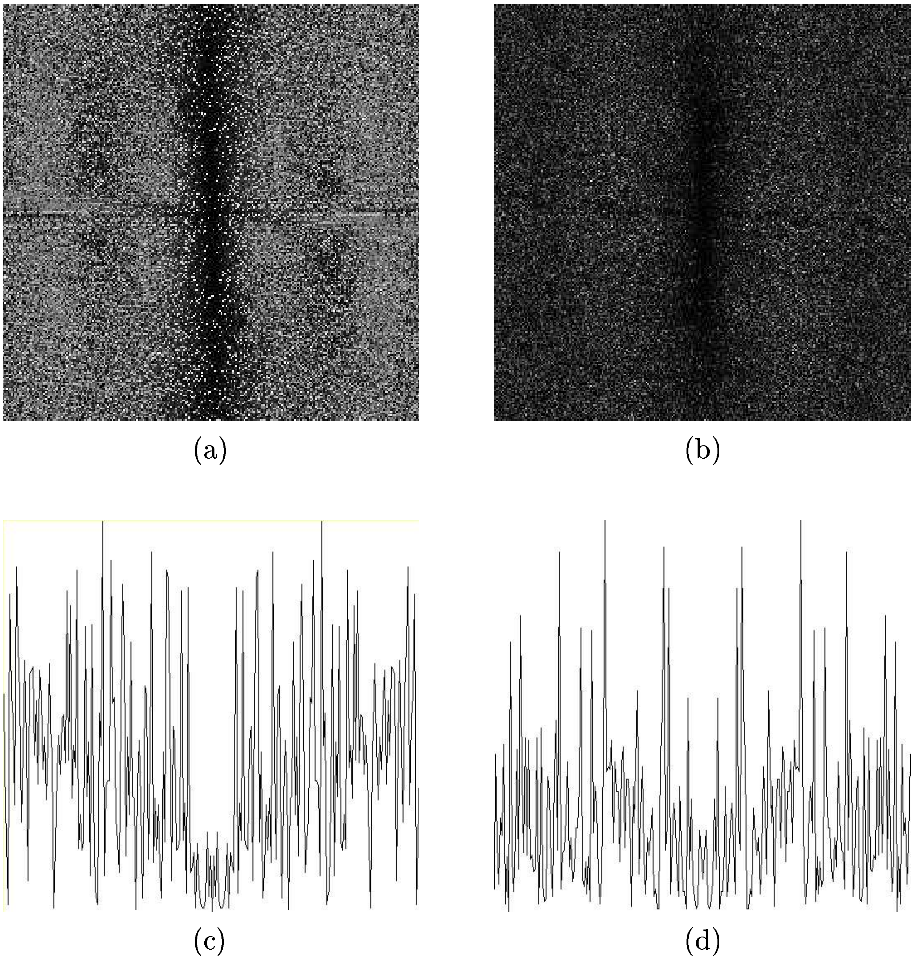


Figure 6.20. Real occlusion test: computation of displacements from the difference of phases and magnitudes of the resultants. (a) Unprocessed phase difference. (b) Unprocessed magnitude difference. (c) Cross-section trhough the phase difference image. (d) Cross-section through the magnitude difference image.

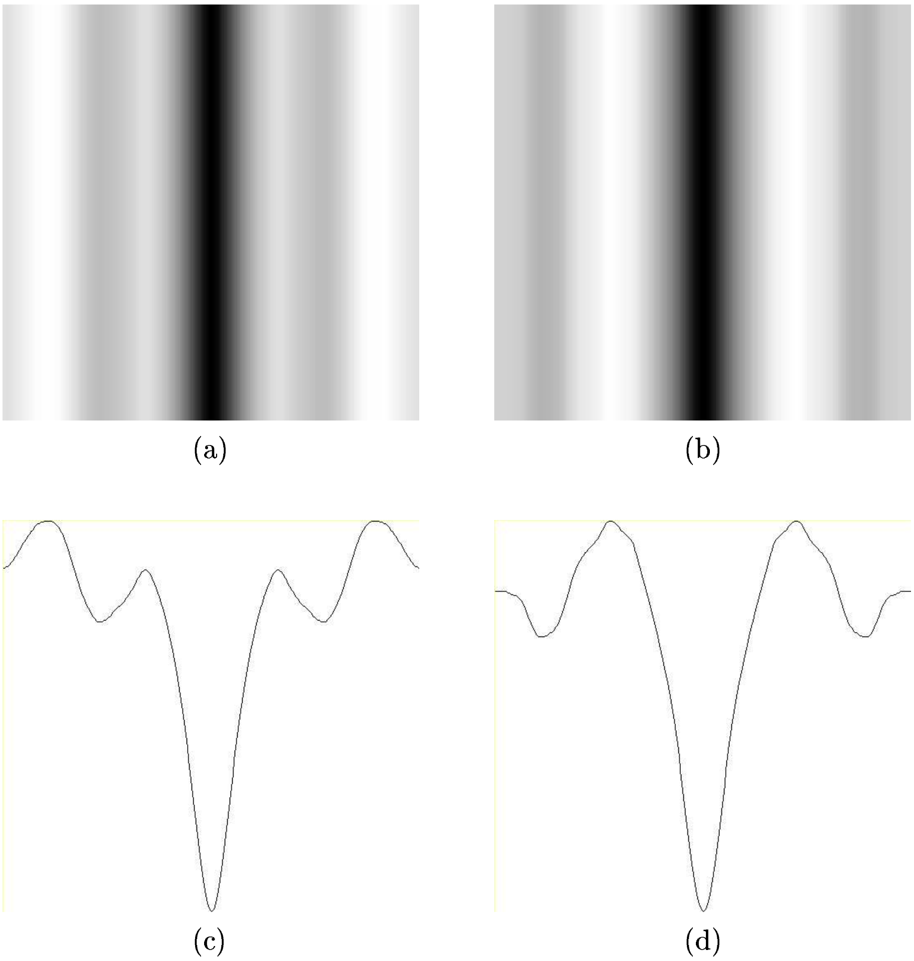
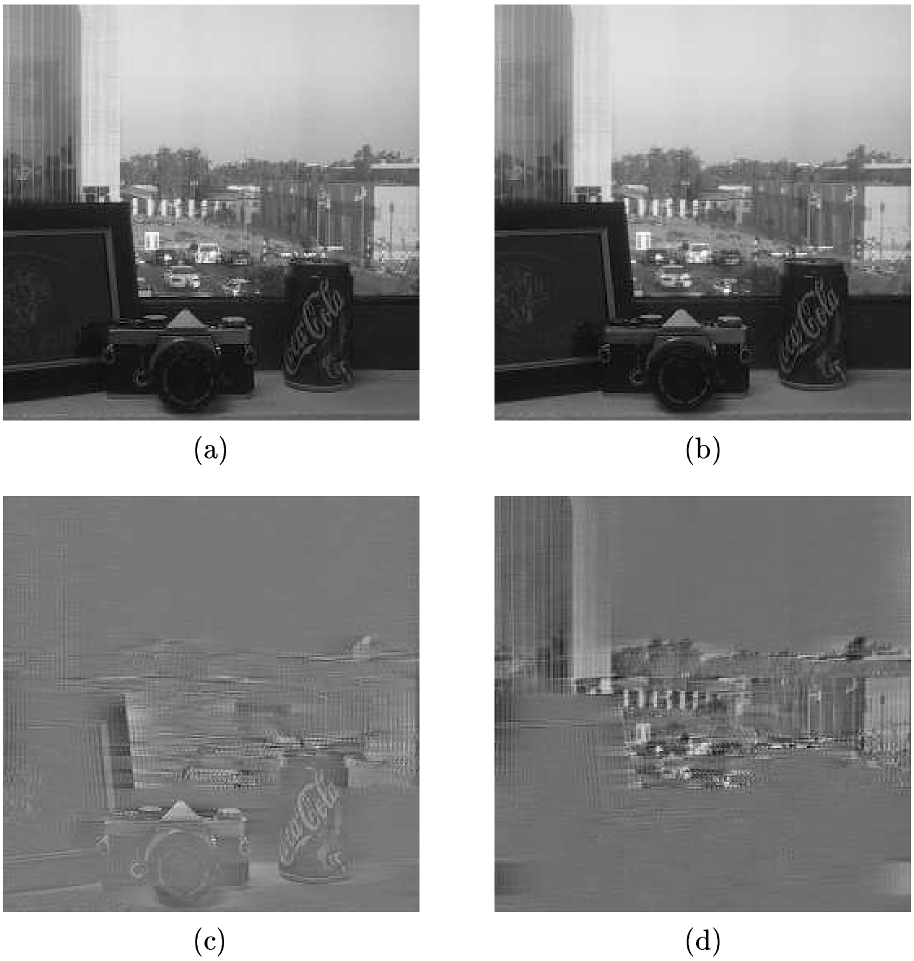


Figure 6.21. Real occlusion test: computation of displacements from the difference of phases and magnitudes of the resultants. (a) The phase difference averaged in vertical spatial frequency (ω_y) direction; the displacement is computed from $v_{x_1} = \frac{2\pi n}{n\omega_{x_p}}$ where $n\omega_{x_p}$ are the frequencies at which the phase difference equals zero — these occur at the periodic mimima. (b) The averaged magnitude difference; the displacement is computed from $|v_{x_1} - v_{x_2}| = \frac{2\pi n}{\omega_{x_m}}$ where $n\omega_{x_m}$ are the frequencies at which the magnitude difference equals zero — again, these occur at the periodic mimima. (c) Cross-section through the phase difference image. (d) Cross-section through the magnitude difference image.



(a) (b) (c) (d)

Figure 6.22. Real occlusion test. (a) Left image. (b) Right image. (c) Segmented foreground. (d) Segmented background.

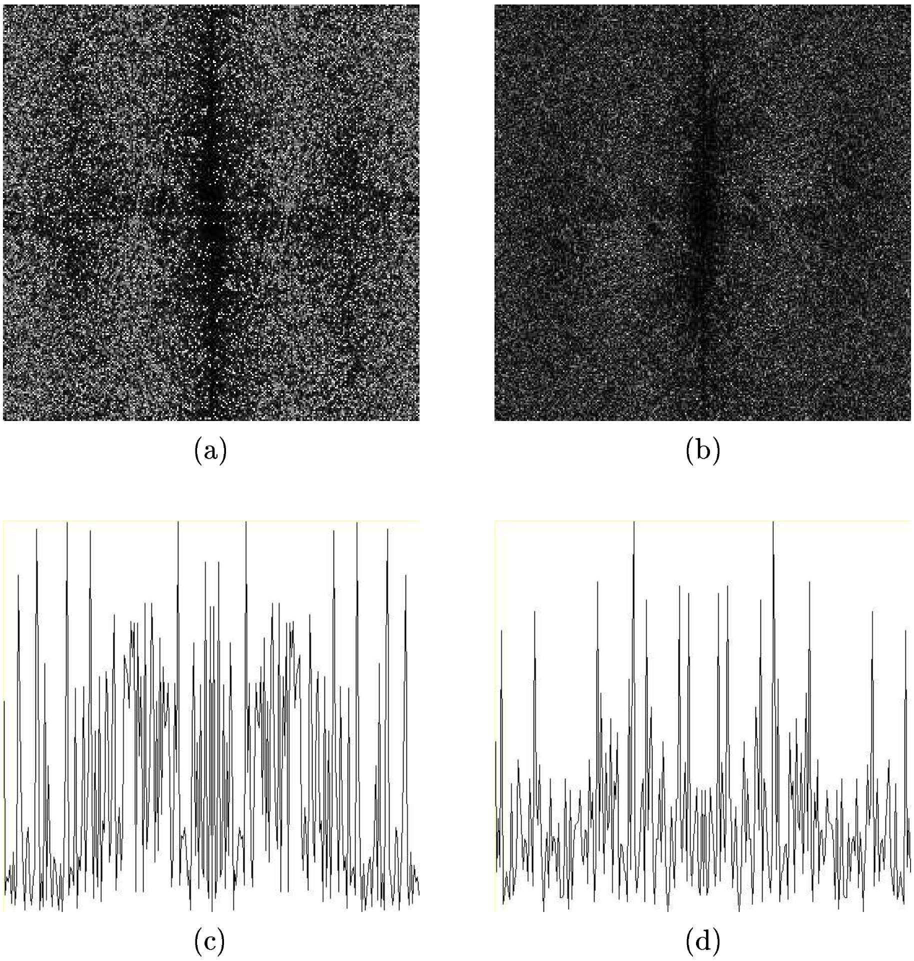


Figure 6.23. Real occlusion test: computation of displacements from the difference of phases and magnitudes of the resultants. (a) Unprocessed phase difference. (b) Unprocessed magnitude difference. (c) Cross-section trhough the phase difference image. (d) Cross-section through the magnitude difference image.

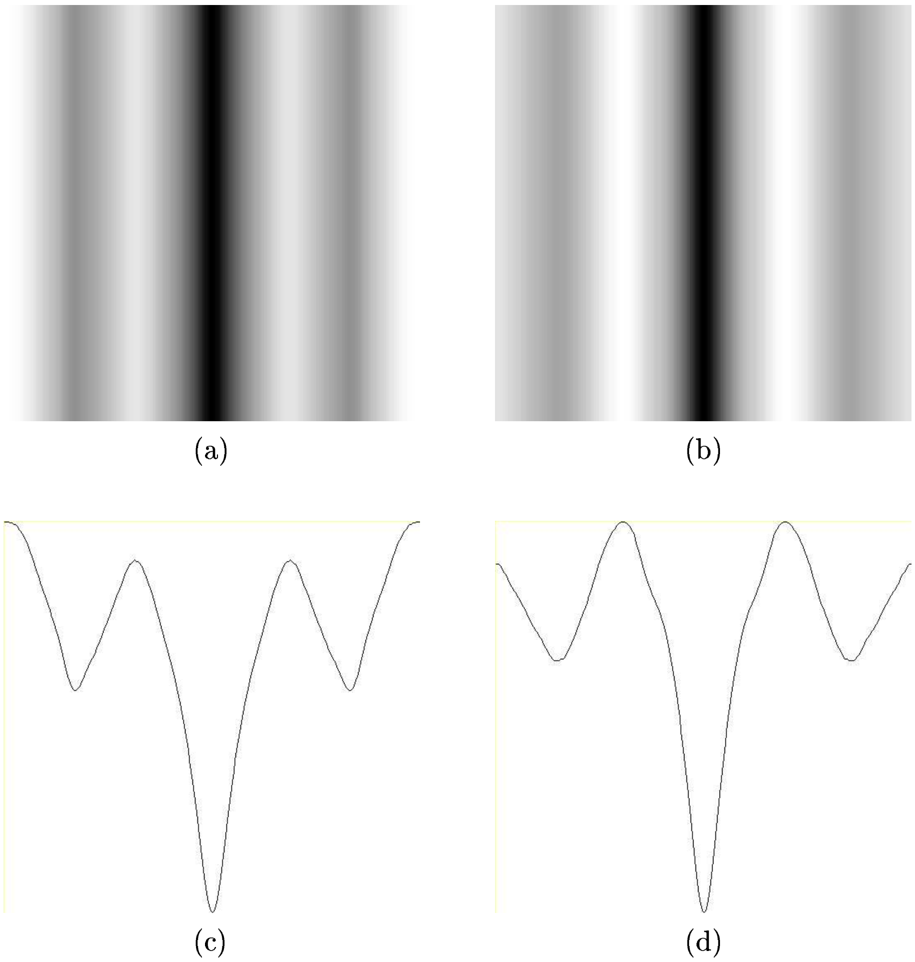


Figure 6.24. Real occlusion test: computation of displacements from the difference of phases and magnitudes of the resultants. (a) The phase difference averaged in vertical spatial frequency (ω_y) direction; the displacement is computed from $v_{x_1} = \frac{2\pi n}{n\omega_{x_p}}$ where $n\omega_{x_p}$ are the frequencies at which the phase difference equals zero — these occur at the periodic mimima. (b) The averaged magnitude difference; the displacement is computed from $|v_{x_1} - v_{x_2}| = \frac{2\pi n}{\omega_{x_m}}$ where $n\omega_{x_m}$ are the frequencies at which the magnitude difference equals zero — again, these occur at the periodic mimima. (c) Cross-section through the phase difference image. (d) Cross-section through the magnitude difference image.

Chapter 7

INSTANTANEOUS OPTICAL FLOW

1. MOTIVATION

So far in this book, we have been concerned with the development of a technique for segmenting images. However, as we have seen, this technique yields not only the segmented components of the constituent objects or images but it also extracts their global spatial velocities or disparities. In all of this, we have assumed that the scene comprises two distinct objects or images, each having a different velocity or disparity. However, if the scene which is being viewed doesn't exhibit this figure-ground composition, we can still perform some useful work and establish the structure of the scene. How? Well, if we are able to compute global image velocity, then we should also be able to compute local image velocity, *i.e.*, we should be able to use the technique to compute the local optical flow of the image. This in turn, together with some assumptions about the motion of either the observer or the objects in the scene, allows one to make inferences about the structure of the scene (see [169] for an excellent introduction to this subject). In this chapter we will show how the optical flow can be computed using a fairly well-established phase-based Fourier method. In the next chapter, we look at the possibility of deploying the full theory developed in this book and we'll consider the circumstances under which it would be appropriate to do so.

2. VELOCITY FROM PHASE CHANGE

To compute the optical flow of an image sequence, we need to measure the velocity of every point in the image (or at least a representative sample of them.) Unfortunately, this is an ill-posed problem because the motion of a distinct image point, taken in isolation from any other, is often ambiguous. To overcome this problem, we consider instead the velocity of a local region, treating it as a sub-image which is translating with some velocity, and we estimate the velocity of a point as the velocity of the region centred at that point. In this chapter, we are going to measure the velocity of the region by analysing the phase change between image regions and we will exploit the Fourier shift property to estimate the velocity.

2.1 OVERVIEW – TWO IMAGES OR FOUR?

Up to this point, and excepting the special case in Chapter 6, we have needed four image samples to segment the two images and to compute their respective velocities. That is, we effectively had four unknown quantities and we needed four constraints to determine them. Here, however, we don't wish to know the component images and we only want to recover the velocity of some local image region. This allows us to relax the requirement for four image samples and we can compute the region velocity or displacement from only two image regions [369, 373]. For this reason, we refer to the technique as *instantaneous* optical flow estimation.

2.2 MEASUREMENT OF PHASE CHANGE & IMAGE VELOCITY

We already know that a function $f(x, y)$ translating with velocity (v_x, v_y) can be written $f(x - v_x\delta t, y - v_y\delta t)$. We know that, by the shift property, its Fourier transform is given by:

$$\mathcal{F}\left(f(x - v_x\delta t, y - v_y\delta t)\right) = |\mathsf{F}(\omega_x, \omega_y)|\, e^{i\phi(\omega_x,\omega_y)} e^{-i(\omega_x v_x \delta t + \omega_y v_y \delta t)}$$

Thus, a spatial shift of $(v_x\delta t, v_y\delta t)$ of a signal in the spatial domain, *i.e.* $f(x, y)$ shifted to $f(x - v_x\delta t, y - v_y\delta t)$, only produces a change in the phase of the Fourier components in the frequency domain. This phase change is $e^{-i(\omega_x v_x \delta t + \omega_y v_y \delta t)}$.

In order to estimate the velocity of a signal translating with constant velocity in the image, we simply need to identify the phase shift, *i.e.* $e^{-i(\omega_x v_x \delta t + \omega_y v_y \delta t)}$. To accomplish this, we note that the phase spectrum for the shifted image at time $t + \delta t$ is equal to the phase spectrum of the image at time t multiplied by the phase change given above:

$$\begin{aligned} e^{i\phi_{t+\delta t}(\omega_x,\omega_y)} &= e^{-i(\omega_x v_x \delta t + \omega_y v_y \delta t)} e^{i\phi_t(\omega_x,\omega_y)} \\ &= e^{i(\phi_t(\omega_x,\omega_y) - (\omega_x v_x \delta t + \omega_y v_y \delta t))} \end{aligned}$$

Hence:

$$\phi_{t+\delta t}(\omega_x, \omega_y) = \phi_t(\omega_x, \omega_y) - (\omega_x v_x \delta t + \omega_y v_y \delta t) \tag{7.1}$$

That is, the phase at time $t + \delta t$ is equal to the initial phase at time t minus $(\omega_x v_x \delta t + \omega_y v_y \delta t)$. Since we require v_x and v_y, we rearrange as follows:

$$v_y = \frac{1}{\omega_y \delta t} \left(\phi_t(\omega_x, \omega_y) - \phi_{t+\delta t}(\omega_x, \omega_y) - \omega_x v_x \delta t \right) \tag{7.2}$$

This equation is degenerate if $\omega_y = 0$ in which case we substitute $\omega_y = 0$ in equation 7.1 and use an alternative re-arrangement as follows:

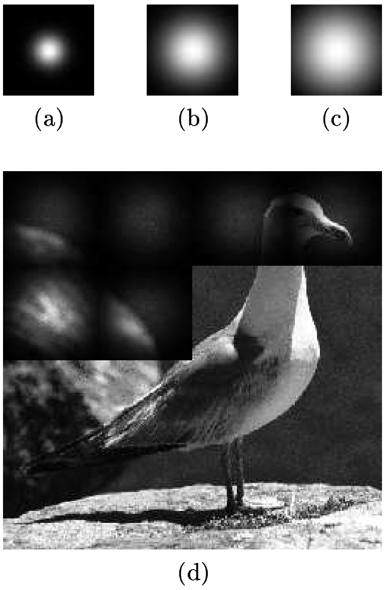
$$v_x = \frac{1}{\omega_x \delta t} \left(\phi_t(\omega_x, \omega_y) - \phi_{t+\delta t}(\omega_x, \omega_y) \right) \tag{7.3}$$

If we have two images taken at time $t = t_0$ and $t = t_0 + \delta t$, we can compute ϕ_{t_0} and $\phi_{t_0+\delta t}$ from their Fourier transforms. Treating equation 7.2 above as a Hough transform, with a 2-D Hough transform space defined on v_x, v_y, then we can compute v_y for all possible values of v_x and for all (known) values of $\omega_x, \omega_y, \phi_t(\omega_x, \omega_y), \phi_{t+\delta t}(\omega_x, \omega_y)$. Local maxima in this v_x, v_y Hough transform space signify Fourier components which comprise signals in the spatial domain that are moving with constant velocity v_x, v_y. Note that, in equation 7.2, $\phi(\omega_x, \omega_y)$ represents the absolute phase of frequency (ω_x, ω_y). However, in the Fourier domain, phase is bounded by $\pm 2\pi$ and phase values will 'wrap' as they cross this threshold. In effect, phase values are represented modulo 2π. We have allowed for this by solving equation 7.2 for the given phase values $\phi(\omega_x, \omega_y) + 2n\pi, \phi > 0; \phi(\omega_x, \omega_y) - 2n\pi, \phi < 0$, for all n such that $2n\pi <\mid \omega_x v_{x_{max}} \delta t \mid + \mid \omega_y v_{y_{max}} \delta t \mid$.

It is important to note here that we don't have to solve any equations as we did in previous chapters. We simply need to compute the phases of the Fourier components of the image region at two different times t and $t + \delta t$ and then compute v_y for all v_x, according to equation 7.2.

2.3 LOCAL APPLICATION USING APODIZED WINDOWS

The above theory is based on one fundamental assumption: that the only thing which changes between each image sample is the phase of the individual Fourier components. It assumes no change in the spectral content. However, it is inevitable that when we are dealing with the Fourier transform of a moving image there will be some spectral changes due to the introduction and removal of visual information at the boundary of that image. If the size of the image is large compared to its velocity, then these boundary effects will not be significant and can be ignored. However, when we are attempting to compute the optical flow in a small region, this is not the case and we must do something about them. The solution is to apply a multiplicative apodizing windowing function which



(a) (b) (c)

(d)

Figure 7.1. Gaussian windowing functions: (a), (b), (c) 50% weighting at $\frac{w}{8}$, $\frac{2w}{8}$, $\frac{3w}{8}$ pixels from the region centre, respectively (w is the size of the window); (d) each region in the image is multiplied by the windowing function before its Fourier transform is computed and its velocity estimated.

attenuates the visual information at the periphery of the region in question. This windowing function is applied before the Fourier transform is computed.

In this book, a Gaussian windowing function is used and the Gaussian's standard deviation σ is chosen such that the weighting at a some distance from the region centre is 50% of that at the region centre, where w is the length (in pixels) of the side of the 2-D region. Results are presented for Gaussian windowing functions of three standard deviations, each representing decreased attenuation of image data toward the edge of the image (the three functions provide 50% weighting at $\frac{w}{8}$, $\frac{2w}{8}$, and $\frac{3w}{8}$ pixels from the region centre) — see Figure 7.1. In the following, we will denote the three Gaussian functions by $G_{\frac{w}{8}}$, $G_{\frac{2w}{8}}$, and $G_{\frac{3w}{8}}$.

2.4 ALGORITHM

The computation of the instantaneous optical flow is accomplished in the following steps:

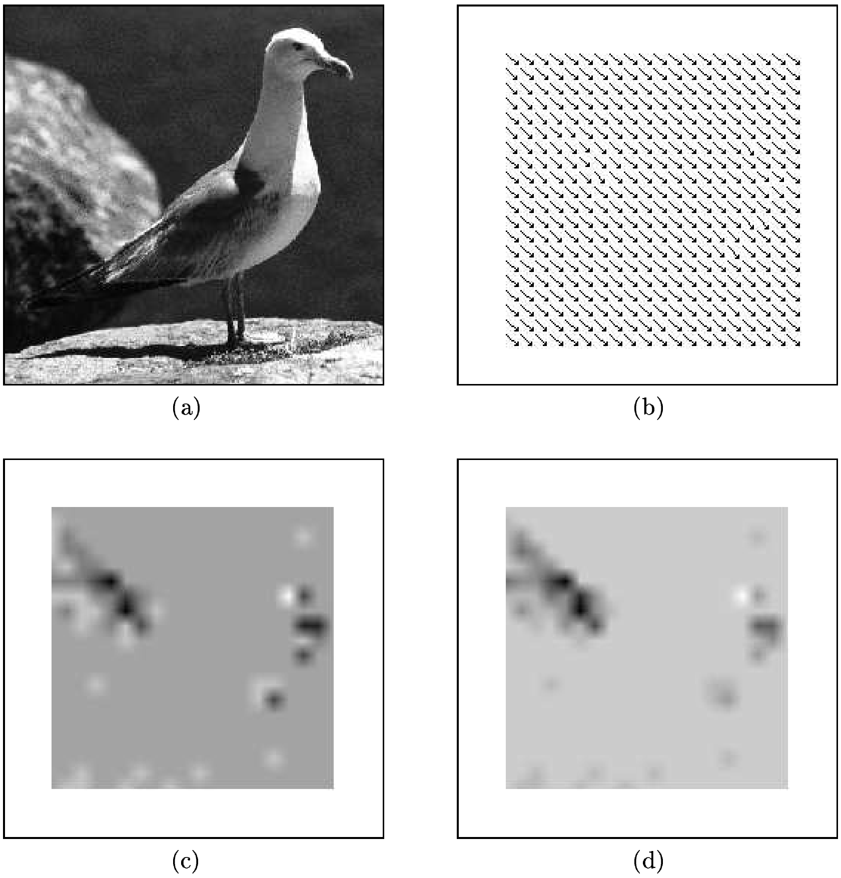
1. Sample the image at some pre-determined sampling period;

2. Extract a region of some size $w \times w$ pixels centred at some coordinates (i, j) (actually we extract two regions, one at time t and the other at time $t + \delta t$);
3. Multiply these regions by the Gaussian window function;
4. Compute their Fourier tranforms;
5. Extract their phases;
6. Update the Hough space for each spatial frequency (ω_x, ω_y) using equation 7.2.

The position of the maximum in the Hough transform space provides the estimate of the velocity of the region.

The full algorithm can be summarized as follows.

```
/* compute optical flow at coordinates i and j in images f1(i,j) and f2(i,j) */
/* where i and j are the coordinates of the centre of a 64 x 64 pixel region */
/* i and j effectively sample the image with a sampling period sp            */
/* (sp = 10 pixels in the results presented in this chapter)                 */
/* The dimensions of f(i,j) are assumed to be given by variables d_i and d_j */

initial_i = 32; final_j = d_i - 32;
initial_j = 32; final_j = d_j - 32;

for (i = initial_i; i < final_i; i = i + sp)
   for (j = initial_j; j < final_j; j = j + sp)

      extract 64x64 pixel regions f1' and f2', centred at i,j, from f1 and f2

      apodized/window f1' and f2' by computing

         g1(x,y) = f1'(x,y) x G(x,y)  // G(x,y) is a 64x64 pixel Gaussian
         g2(x,y) = f2'(x,y) x G(x,y)  // = 0.5 at nw/8 pixels from centre
                                      // (n=1, 2, 3 and w = 64)

      compute G1(wx, wy), the Fourier transform of g1(x,y)
      compute G2(wx, wy), the Fourier transform of g2(x,y)

      compute the phases of G1 and G2:  P1(wx, wy) and P2(wx, wy)

      for (wx = inital_wx; wx < final_wx; wx = wx + 1)      // -32 < wx < 32
         for (wy = inital_k; wy < final_wy; wy = wy + 1)    // -32 < wy < 32

            compute phase difference: pd = P1(wx, wy) - P2(wx, wy);

            if (wy != 0)
               for (vx = 0; vx < 10; vx = vx + 0.1)   // vx is the x velocity
                  vy = (pd - (wx * vx)) / wy
                  Hough_accumulator[vx,vy] += 1
            else if (wx != 0)
               vx = pd / wx;
               for (vy = 0; vy < 10; vy = vy + 0.1)   // vy is the y velocity
                  Hough_accumulator[vx,vy] += 1
```



(a) (b)

(c) (d)

Figure 7.2. (a) Simple test sequence: optical flow is computed using this image and an identical image translated by 2 pixels in the x and y direction. (b) Optical flow field computed using phase information ($G_{\frac{2w}{8}}$ Gaussian windowing function). (c) and (d) Error images for the Fleet and Nagel error metrics, respectively (refer to Table 7.1 for details).

```
Identify vx_max, vy_max such that, for all vx, vy,
Hough_accumulator[vx_max][vy_max] >= Hough_accumulator[vx][vy],

vx_max, vy_max is the velocity of the 64x64 region centred at i,j
```

3. EXAMPLES

3.1 SIMPLE GULL SEQUENCE

Figure 7.2 shows the result of applying the technique to two gull images where the second image is simply the first translated in both the x and y direction by 2 pixels. This test is an elementary baseline benchmark and is intended to do no more than demonstrate the accuracy and the repeatability of the technique on data with a known (and trivial) flow-field. Figure 7.2 (a) shows the test image. Figure 7.2 (b) shows the computed flow field using Gaussian weighting functions with 50% weight at $\frac{2w}{8}$ pixels from window centre (w = window size). Note that all of the results shown in this chapter were computed with a window size of 64 × 64 pixels. The flow field has been computed at ten-pixel intervals.

Figures 7.2 (c) and (d) show the errors in the computed flow field according to the Fleet and the Nagel error metrics, respectively. These error images were produced by computing the error of the flow field at five-pixel intervals and then interpolating between these samples to complete the image.

The Fleet error metric is defined as follows ([29]). Let $\vec{v_t}$ denote the true optical flow vector at a point and let $\vec{v_c}$ denote the corresponding computed flow vector. Let θ_t and θ_c be their respective directions. The Fleet error, e_F, is defined to be

$$e_F = \cos^{-1}\left(\frac{u_t u_c + v_t v_c + 1}{\sqrt{{u_t}^2 + {v_t}^2 + 1}\sqrt{{u_c}^2 + {v_c}^2 + 1}}\right) \tag{7.4}$$

where

$$\begin{aligned} u_c &= |\vec{v_c}|\cos\theta_c \\ v_c &= |\vec{v_c}|\sin\theta_c \\ u_t &= |\vec{v_t}|\cos\theta_t \\ v_t &= |\vec{v_t}|\sin\theta_t \end{aligned}$$

The Nagel error metric [257], e_N, is defined in a slightly different way to be

$$e_N = \sqrt{{e_{\cos}}^2 + {e_{\sin}}^2} \tag{7.5}$$

where

$$\begin{aligned} e_{\cos} &= |\vec{v_c}|\cos\theta_c - |\vec{v_t}|\cos\theta_t \\ e_{\sin} &= |\vec{v_c}|\sin\theta_c - |\vec{v_t}|\sin\theta_t \end{aligned}$$

Table 7.1 summarizes the mean and standard deviation of the Fleet and Nagel error metrics, together with the Root Mean Square (RMS) error of the magnitude and direction values. The chief point to note about these results is that the correct flow field is computed to approximately 0.1 pixels (magnitude) and 1.4 degrees when the $G_{\frac{2w}{8}}$ windowing

Gaussian Weighting	RMS Error Magnitude (pixels)	RMS Error Direction (degrees)	Fleet Error Mean (degrees)	Fleet Error Std. Dev. (degrees)	Nagel Error Mean (pixels)	Nagel Error Std. Dev. (pixels)
$\frac{3w}{8}$	0.21	8.34	2.02	4.61	0.12	0.25
$\frac{2w}{8}$	0.10	1.44	1.48	0.25	0.11	0.05
$\frac{w}{8}$	0.31	6.34	2.83	3.99	0.23	0.30

Table 7.1. Summary of error metrics for the gull image translation benchmark.

function is used. Note that error measures are computed only for points which exhibit non-zero magnitude in both the computed and the true flow fields.

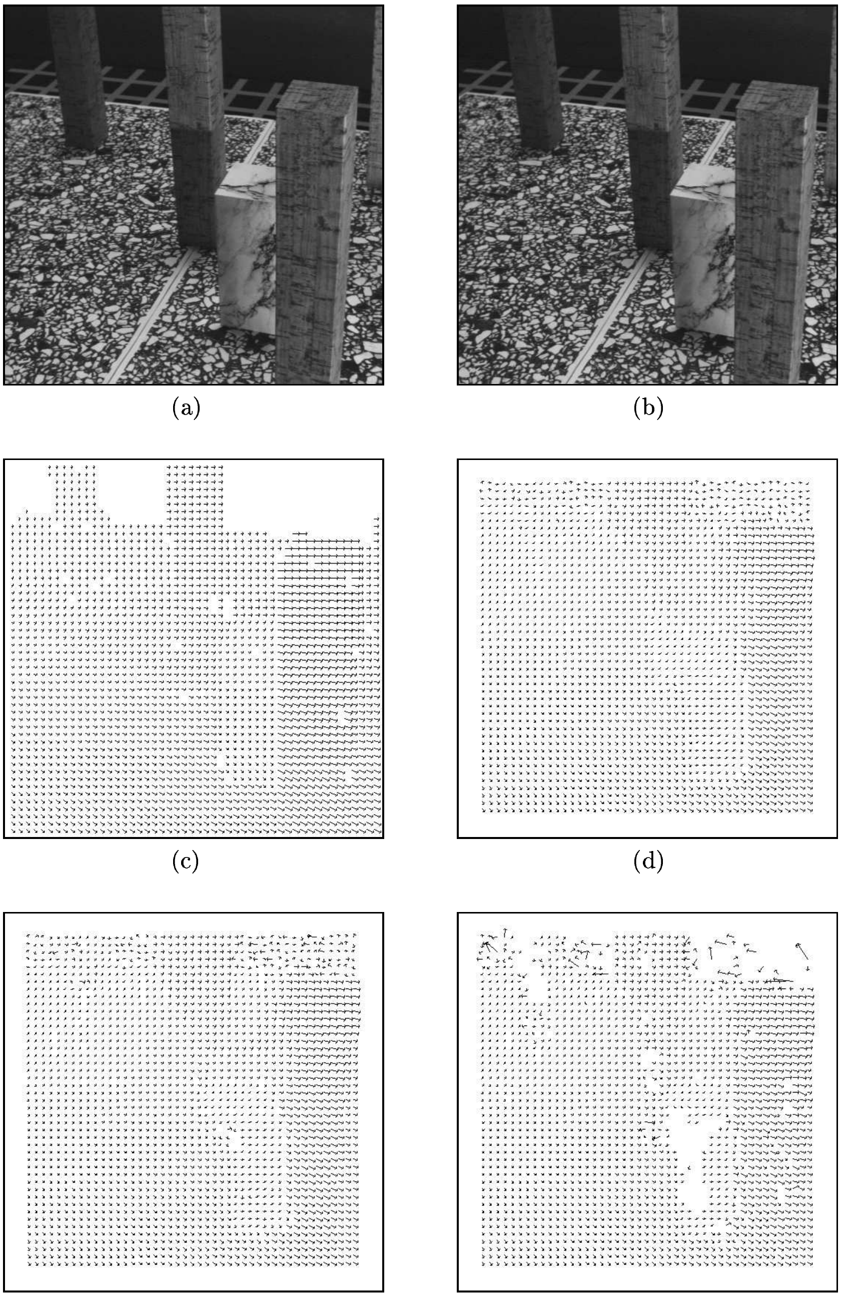
3.2 OTTE AND NAGEL TEST SEQUENCE

Figures 7.3 through 7.6 demonstrate the results of applying the technique to two images in Otte and Nagel's ground-truth test sequence [257]. This sequence comprises images of a scene acquired with a camera mounted on a moving robot arm. The camera motion is a pure 3-D translation toward the scene which comprises a stationary ground plane, four stationary pillars, and a fifth pillar which is translating to the left. The scene exhibits strong occlusion by both the stationary and the moving pillars. The magnitude and direction of the complete optical flow fields of the sequence were computed by Otte and Nagel on the basis of camera calibration data and the robot motion parameters.

Figure 7.3 (a) and 7.3 (b) are images number 40 and 41 in the sequence. Figure 7.3 (c) is the true optical flow field extracted directly from the ground-truth data (sampled every ten pixels). Figures 7.3 (d), (e), and (f) show the optical flow field computed in the manner described above and using the three Gaussian weighting functions $G_{\frac{3w}{8}}$, $G_{\frac{2w}{8}}$, and $G_{\frac{w}{8}}$, respectively. Flow vectors are plotted every ten pixels and their magnitude has been scaled by a factor of four.

Figure 7.4 (a) shows the true magnitude of the optical flow field extracted from Otte and Nagel's ground-truth data, while Figures 7.4 (b), (c), and (d) show the magnitude of the computed optical flow field using the three Gaussian windowing functions $G_{\frac{3w}{8}}$, $G_{\frac{2w}{8}}$, and $G_{\frac{w}{8}}$, respectively. These images were generated by computing the optical flow vectors at ten-pixel intervals and then by interpolating bi-linearly between these computed values.

Figure 7.5 shows the true direction of the optical flow field extracted from Otte and Nagel's ground-truth data, while Figures 7.5 (b), (c), and



(a)

(b)

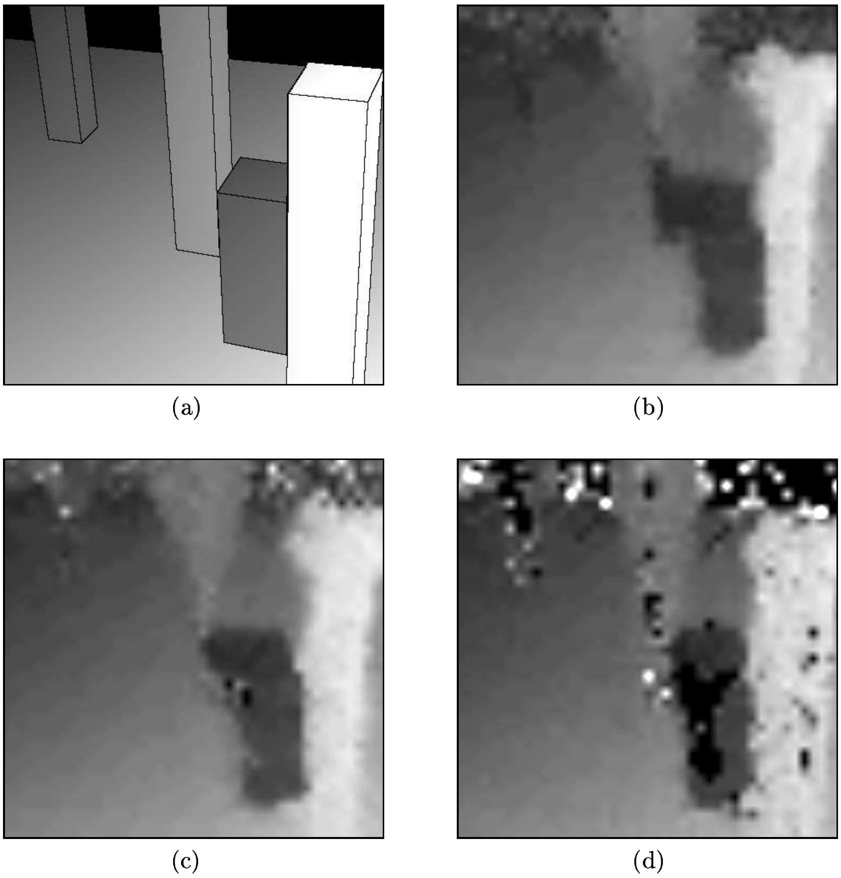
(c)

(d)

(e)

(f)

Figure 7.3. (a) and (b) Images number 40 and 43 of Otte and Nagel's ground-truth motion sequence. (c) True optical flow field given by Otte and Nagel's ground-truth data (sampled every ten pixels). (d), (e), and (f) Optical flow field computed using $G_{\frac{3w}{8}}$, $G_{\frac{2w}{8}}$, and $G_{\frac{w}{8}}$ Gaussian windowing functions, respectively.



(a)

(b)

(c)

(d)

Figure 7.4. (a) Magnitude of the optical flow field extracted from Otte and Nagel's ground-truth data. (b), (c), and (d) Magnitude of the optical flow field computed using the $G_{\frac{3w}{8}}$, $G_{\frac{2w}{8}}$, and $G_{\frac{w}{8}}$ Gaussian windowing functions, respectively.

(d) show the direction of the computed optical flow field. Again, these images were generated by bi-linear interpolation between the optical flow vector values estimated every ten pixels.

Figure 7.6 shows the Fleet and Nagel error images for the sequence using the three Gaussian windowing functions. Here we can see that the greatest error occurs in the regions which correspond to mutual occlusion between the objects. It is also worth noting that the errors are particularly significant for the results obtained using the $G_{\frac{w}{8}}$ windowing function.

Table 7.2 provides a summary of the RMS, Fleet, and Nagel error measures.

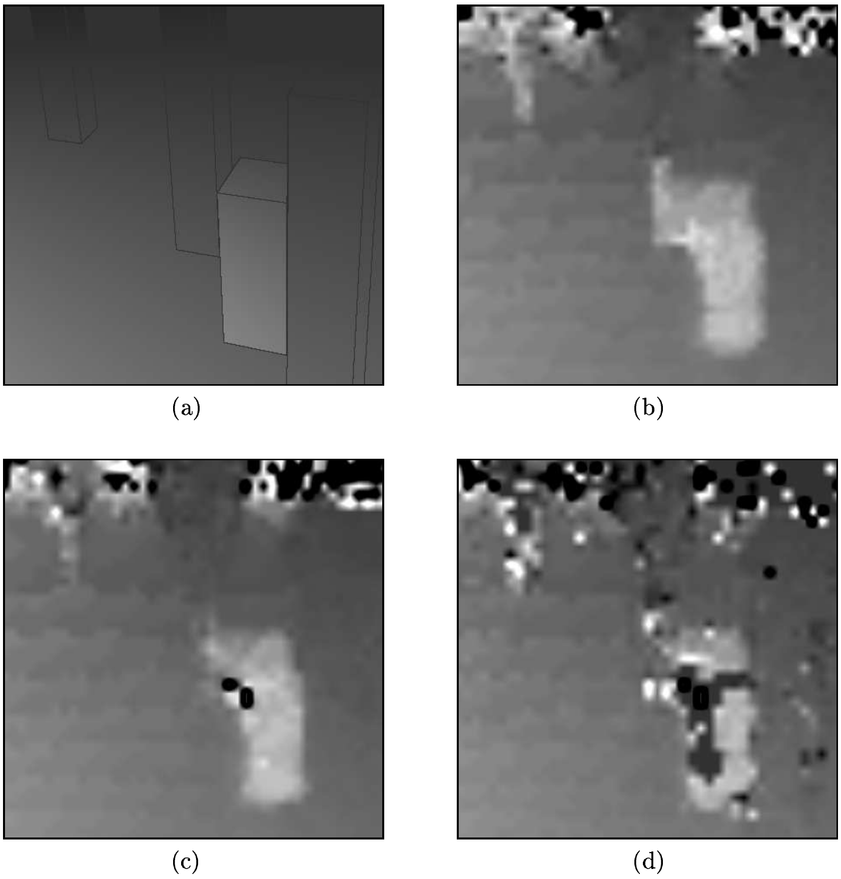


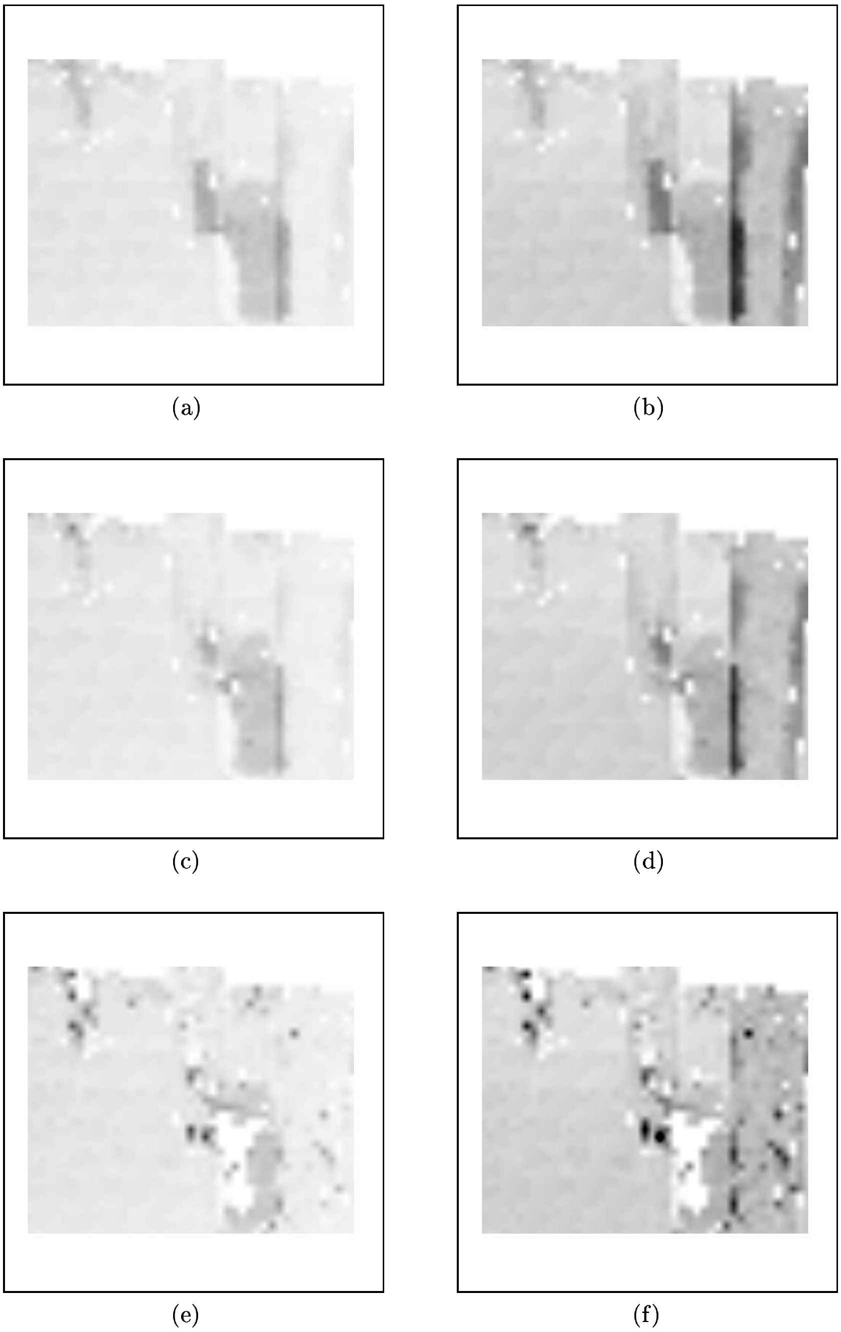
Figure 7.5. (a) Direction of the optical flow field extracted from Otte and Nagel's ground-truth data. (b), (c), and (d) Direction of the optical flow field computed using the $G_{\frac{3w}{8}}$, $G_{\frac{2w}{8}}$, and $G_{\frac{w}{8}}$ Gaussian windowing functions, respectively.

Gaussian	RMS Error		Fleet Error		Nagel Error	
Weighting	Magnitude	Direction	Mean	Std. Dev.	Mean	Std. Dev.
	(pixels)	(degrees)	(degrees)	(degrees)	(pixels)	(pixels)
$\frac{3w}{8}$	0.50	22.83	15.05	5.95	0.55	0.30
$\frac{2w}{8}$	0.44	21.01	14.12	5.24	0.51	0.25
$\frac{w}{8}$	0.45	24.90	15.13	7.14	0.55	0.30

Table 7.2. Summary of error metrics for the Otte and Nagel benchmark sequence.

3.3 SUNFLOWER GARDEN TEST SEQUENCE

The third scenario we have used to demonstrate the technique is the sunflower garden (SFLOWG) sequence provided by the Berkeley Mul-



(a) (b) (c) (d) (e) (f)

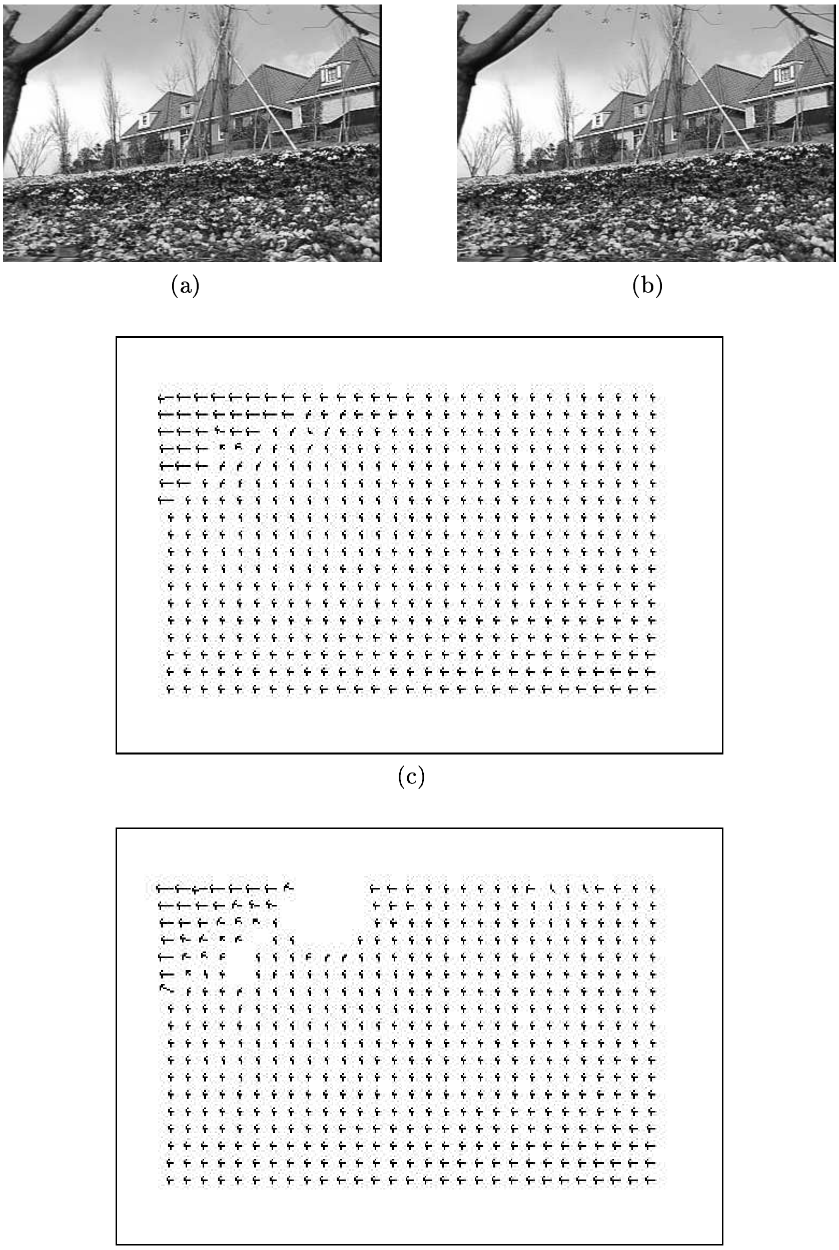
Figure 7.6. (a), (c), (e) Fleet error metric and (b), (d), (f) Nagel error metrics based on computed and ground-truth values using $G_{\frac{3w}{8}}$, $G_{\frac{2w}{8}}$, and $G_{\frac{w}{8}}$ Gaussian windowing functions, respectively. Note that darkness is proportional to error.

timedia Research Center, University of California, Berkeley. This is an extended image sequence and we have used three extracts from it to illustrate various aspects of the technique's performance. In Figure 7.7 we see a rural scene with a hedgerow in the foreground, receding into the distance on the left. Behind this, there are a house and trees. At the left, we can see some overhanging branches of a tree in the foreground. The instantaneous optical flow for the Gaussian windowing functions $G_{\frac{3w}{8}}$ and $G_{\frac{2w}{8}}$ are also shown in Figure 7.7. The interpolated magnitude and direction of the velocity fields are shown in Figure 7.8.

The magnitude of the flow field for this extract decreases from right to left in the region of the hedgerow, as one would expect since it recedes into the distance. Similarly, the region around the foreground tree exhibits flow vectors with large magnitudes and the distant background near the house exhibits flow vectors with small magnitudes. Note that with the narrower $G_{\frac{2w}{8}}$ window function the technique is unable to estimate the flow field in sky region. The $G_{\frac{3w}{8}}$ window function is able to do so be only because it uses information from nearby objects such as the house.

In the next extract, we see the left-hand tree encroaching into the foreground (see Figures 7.9 and 7.10). While the flow field near the top half of the tree is correct, we can see that there are problems near its base and the flow vectors represent the velocity of the neighbouring hedgerow rather than the trunk of the tree. In effect, the hedgerow signal swamps the trunk signal in the local window region. This doesn't happen at the top of the tree because of the featureless sky background. This invasion of neighbouring signal is alleviated somewhat with the narrower $G_{\frac{2w}{8}}$ window function (see Section 4.1 for a further discussion of this problem).

This problem becomes even more pronounced in the third extract in Figures 7.11 and 7.12 where the foreground tree stands centrally in front of the house. With the wider $G_{\frac{3w}{8}}$ window function we fail completely to detect the trunk's flow field. The narrower $G_{\frac{2w}{8}}$ window function is better but still fails in the region near the base where the hedgerow forms the background. Clearly, a narrower window improves the ability of the technique to localize the flow field but, unfortunately, if we decrease the window too much we encounter problems in areas where there are large velocities or displacements. This is an issue to which we will return in Section 4.1.



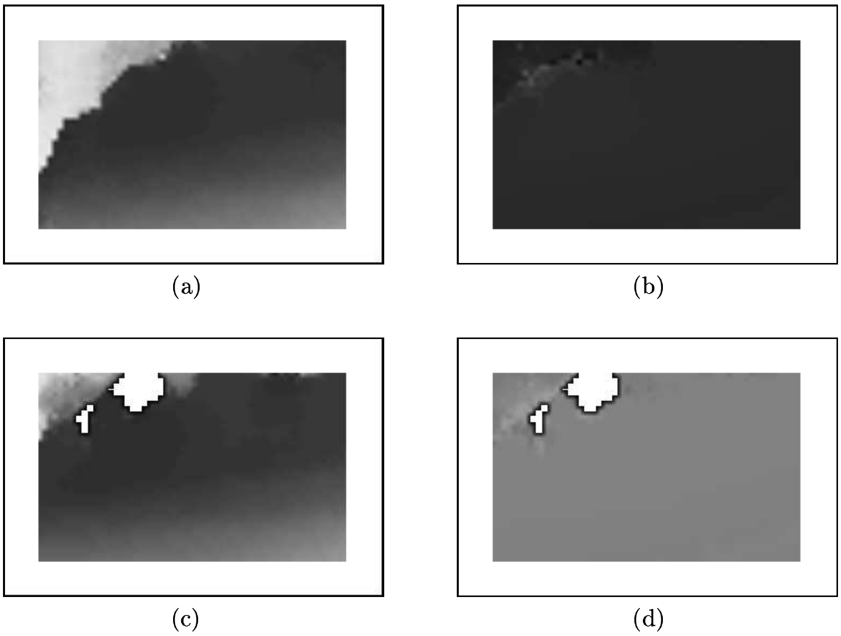
(a)

(b)

(c)

(d)

Figure 7.7. Instantaneous optical flow field for the SFLOWG sequence: (a) frame 50; (b) frame 51; (c) and (d) optical flow field computed using the $G_{\frac{3w}{8}}$ and $G_{\frac{2w}{8}}$ Gaussian windowing functions, respectively. Image sequence courtesy of the Berkeley Multimedia Research Center, University of California, Berkeley.



(a) (b)

(c) (d)

Figure 7.8. Instantaneous optical flow field for the house sequence: (a) & (c) magnitude, (b) & (d) direction, using the $G_{\frac{3w}{8}}$ and $G_{\frac{2w}{8}}$ Gaussian windowing functions, respectively.

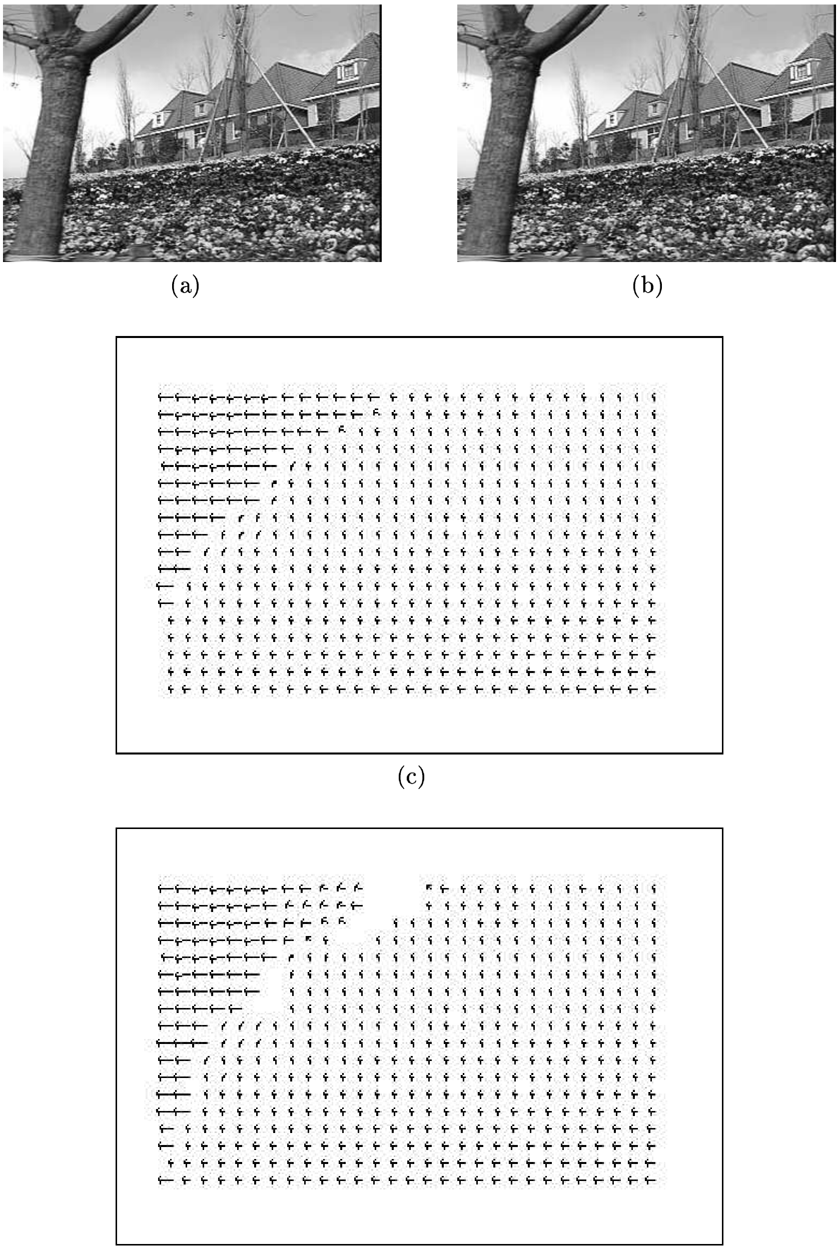
3.4 POST-PROCESSING

We note in passing that all of the results presented in this chapter are the unprocessed output of the algorithm (apart from interpolation); each velocity vector has been estimated independently and the vector field has not been subjected to median or mean filtering. Such filtering would, in fact, help to remove outlying estimates and would improve the overall error measures.

4. DISCUSSION

4.1 DISCONTINUITIES IN THE VELOCITY PROFILE

The preceding results demonstrate that the approach described produces a dense, accurate, repeatable, and reasonably complete flow field. However, in the case of the flow field produced with wider weighting functions, *i.e.* the Gaussians with standard deviation $G_{\frac{3w}{8}}$ and $G_{\frac{2w}{8}}$, the velocity represents an average or aggregate velocity in the windows. Normally, this is not a problem, especially where the velocity profile in the window is either constant or varying linearly since the average will



(a)

(b)

(c)

(d)

Figure 7.9. Instantaneous optical flow field for the SFLOWG sequence: (a) frame 36; (b) frame 37; (c) and (d) optical flow field computed using the $G_{\frac{3w}{8}}$ and $G_{\frac{2w}{8}}$ Gaussian windowing functions, respectively. Image sequence courtesy of the Berkeley Multimedia Research Center, University of California, Berkeley.

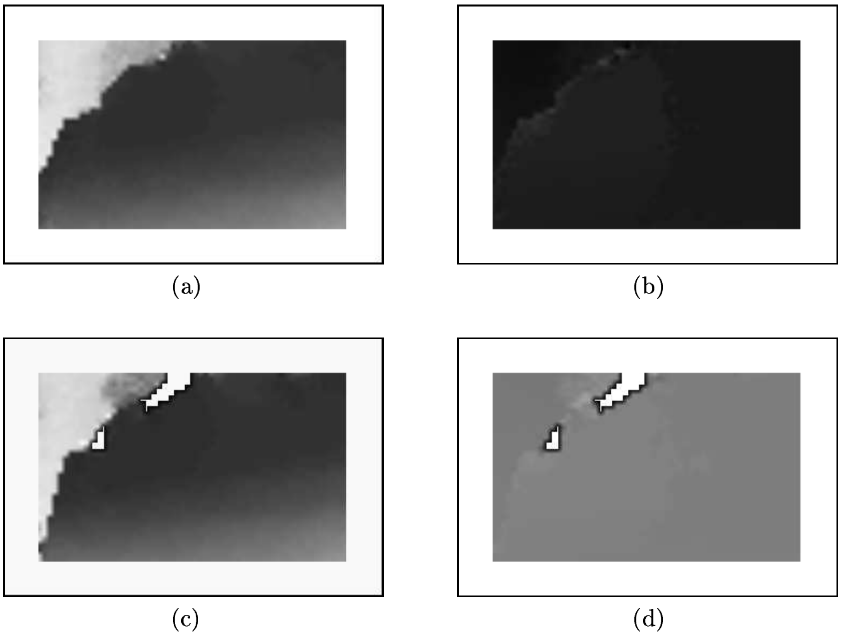


Figure 7.10. Instantaneous optical flow field for the SFLOWG sequence: (a) & (c) magnitude, (b) & (d) direction, using the $G_{\frac{3w}{8}}$ and $G_{\frac{2w}{8}}$ Gaussian windowing functions, respectively.

represent a good estimate at the centre of the window over which the estimate is taken. On the other hand, if there is a local variation in the velocity profile in the window or, worse still, if there is a discontinuity in the velocity profile such as is the case where there are two or more objects moving in the window, then this estimate will be unreliable. The reason for this becomes clear when we reflect on the manner in which we are computing the velocity estimate, *i.e.* by computing the phase changes of the Fourier frequency components. We have made the tacit assumption that there is just one single object moving in the window or, equivalently, that the image function in the window translates with unique and uniform velocity. Thus, the Fourier components should all exhibit the (frequency-dependent) phase change associated with this velocity. When there are two or more velocities, the Fourier components are the resultants of the two individual components of each object (assuming common spectral support) and the technique described in this chapter computes the velocity *based on the phase change of the resultant, and not on the the individual components.* Since the phase change of the resultant Fourier component will depend on the phase changes of the individual components *and* on their magnitudes, there won't be

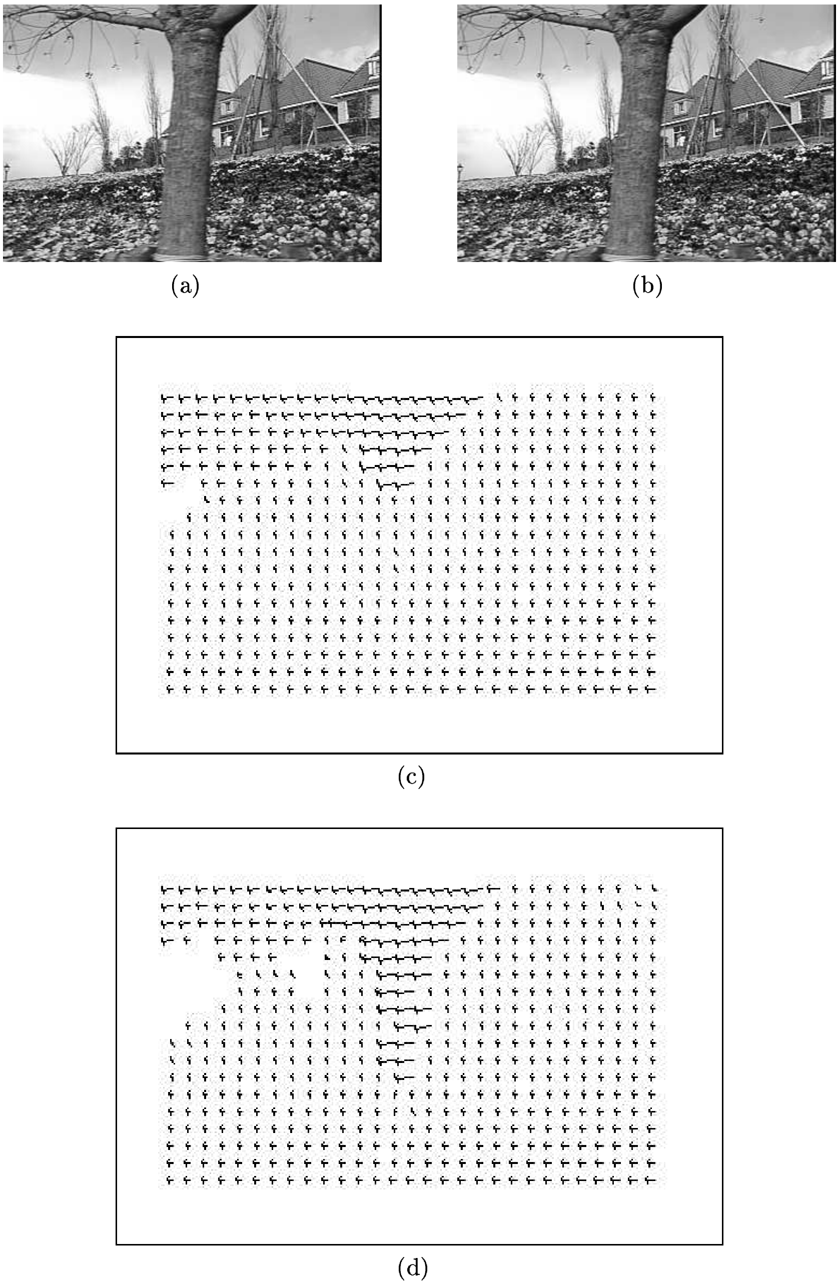
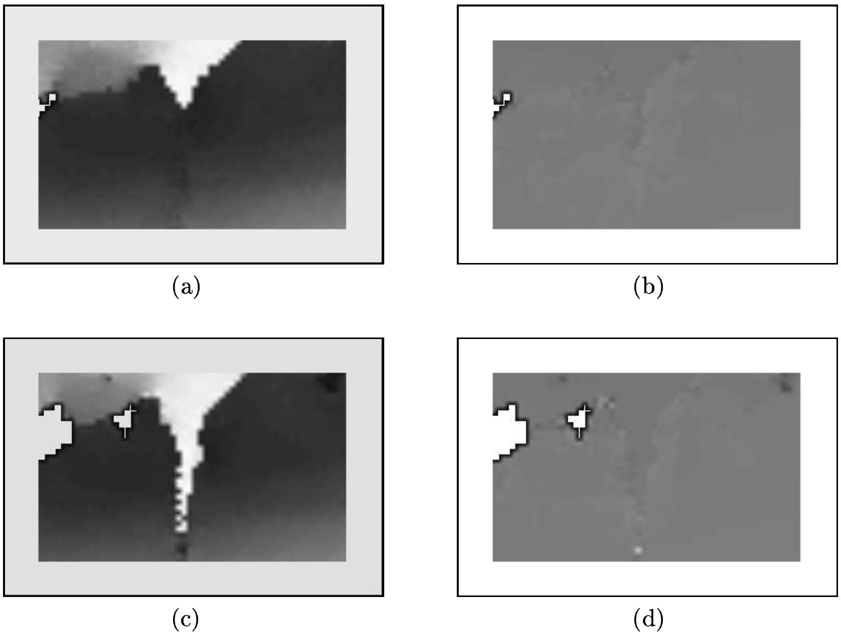


Figure 7.11. Instantaneous optical flow field for the SFLOWG sequence: (a) frame 10; (b) frame 11; (c) and (d) optical flow field computed using the $G_{\frac{3w}{8}}$ and $G_{\frac{2w}{8}}$ Gaussian windowing functions, respectively. Image sequence courtesy of the Berkeley Multimedia Research Center, University of California, Berkeley.



(a) (b)

(c) (d)

Figure 7.12. Instantaneous optical flow field for the SFLOWG sequence: (a) & (c) magnitude, (b) & (d) direction, the $G_{\frac{3w}{8}}$ and $G_{\frac{2w}{8}}$ Gaussian windowing functions, respectively.

a single consistent phase change for all Fourier components — different relative magnitudes will yield a different phase change in the resultant even if the phase changes of the components are consistent — and hence there won't be a well-defined local maximum in the Hough transform space. Consequently, the computed velocity will be very unreliable. If the signal corresponding to one of the sub-regions in the local window is significantly stronger than that of the other then its Fourier components will tend to dominate and the phase change of the resultant may be a better approximation to the phase change of the dominant component. In this case, the measured velocity will approximate that of the dominant sub-region. This is exactly what happens at the base of the tree in Figures 7.9 and 7.11.

As the window in which the velocity is being estimated encounters and crosses a velocity discontinuity, as in the case of an occluding boundary, the computed velocity will change from a true estimate of the velocity of the first object to a true estimate of the velocity of the second object, passing through probably unreliable estimates of the aggregate velocity of the two. This is particularly evident in the velocity fields associated with the 'wider' weighting functions, $G_{\frac{2w}{8}}$ and $G_{\frac{3w}{8}}$ (see Figures 7.3

(d) and 7.3 (e)), since they incorporate more components from both objects as the window passes over them. The flow-field associated with the $\sigma_{\frac{w}{8}}$ weighting function represents an attempt to reduce the effective support, computing just one velocity. However, the reduced support introduces its own problems. The velocity estimate is more prone to produce spurious results and, given its reduced 'window of visibility', it encounters problems in identifying a velocity in regions where the image function is approximately constant and, hence, there is no detectable phase change (see Figure 7.3 (f)). It will also encounter problems if the velocity is large with respect to the size of the window since there will be a much higher change in the spectral content due to the large amount of image data entering and exiting the window at its periphery. In essence, if we try to solve the velocity discontinuity problem by reducing the size of the window (or the width of the window function) we also reduce the range of velocities which can be measured.

An alternative strategy is to *increase* the resolution of the image itself. This has the effect of reducing the relative size of the window *vis-à-vis* the objects in the image and, as long as the object velocity does not exceed the point at which the change in spectral content in the window becomes a problem, this approach will improve the ability of the technique to localize the velocity measurement. For example, Figures 7.13 and 7.14 show the results of applying a 64 × 64 pixel window and the $G_{\frac{3w}{8}}$ and $G_{\frac{2w}{8}}$ Gaussian window functions to an enlarged version of the SFLOWG sequence. In this case, the local estimate of velocity near the tree trunk is much improved and the problems associated with the invasion by the background hedgerow signal have been alleviated somewhat.

Note that doubling the size of the image is *not* the same as halving the size of the window (*e.g.* from 64 × 64 to 32 × 32 pixels). Recall from Chapter 2 that the quantization resolution of the phase angles is proportional to the resolution of the image (in this case, to the resolution of the window). If we halve the window size, we halve the resolution at which we can measure phase and phase changes and, consequently, we reduce the accuracy with which we can measure velocity. It has been found empirically that a 64 × 64 pixel window is the smallest image size with which one can reliably estimate local velocities. In an ideal world, one where the speed of computation is not an issue, we would greatly increase the resolution with which the image sequence is acquired and use a considerably larger window.

Another solution to the problem of local variation and discontinuities in velocity profiles in the window is to identify explicitly the two (or more) velocities of those objects translating in the window and then to assign the appropriate velocity to that being estimated in the window.

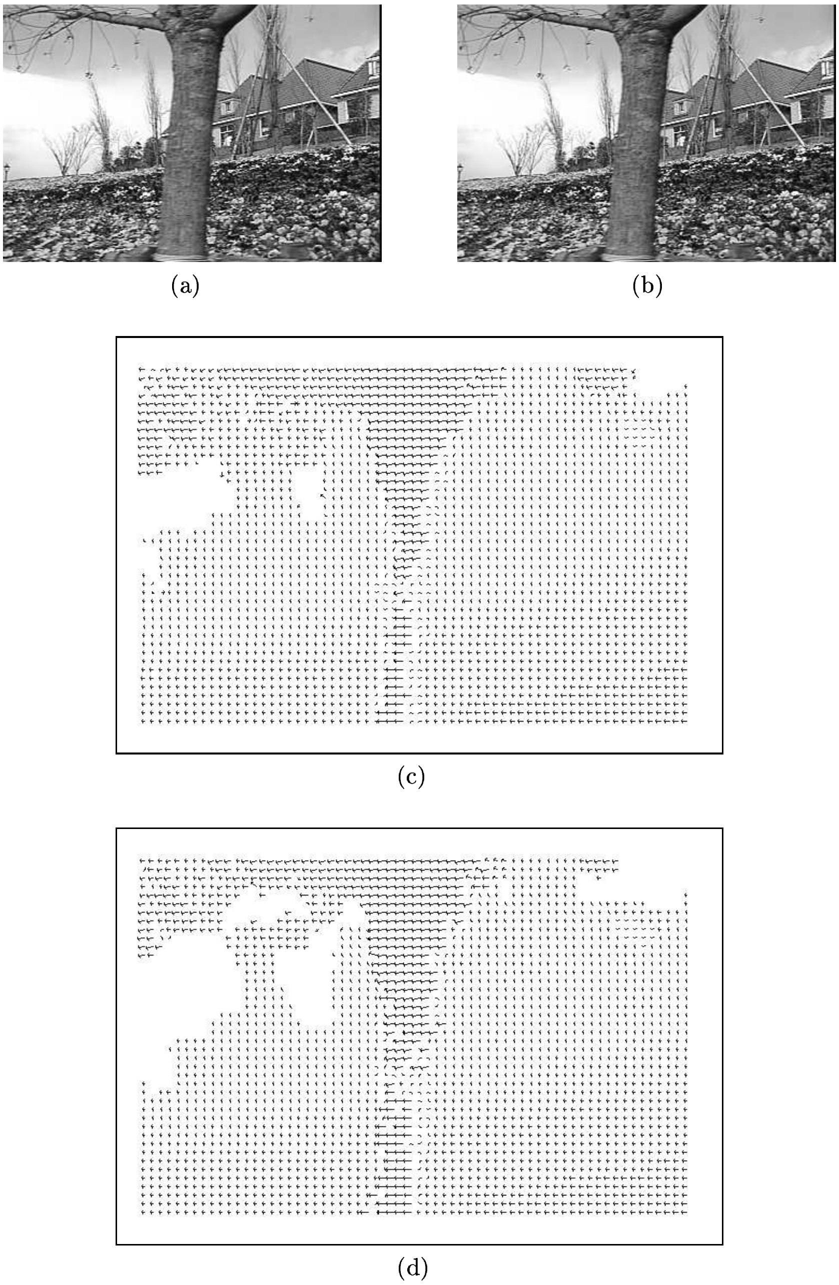


Figure 7.13. Instantaneous optical flow field for the SFLOWG sequence: (a) frame 10; (b) frame 11; (c) and (d) optical flow field computed using the $G_{\frac{3w}{8}}$ and $G_{\frac{2w}{8}}$ Gaussian windowing functions, respectively. Image sequence courtesy of the Berkeley Multimedia Research Center, University of California, Berkeley.

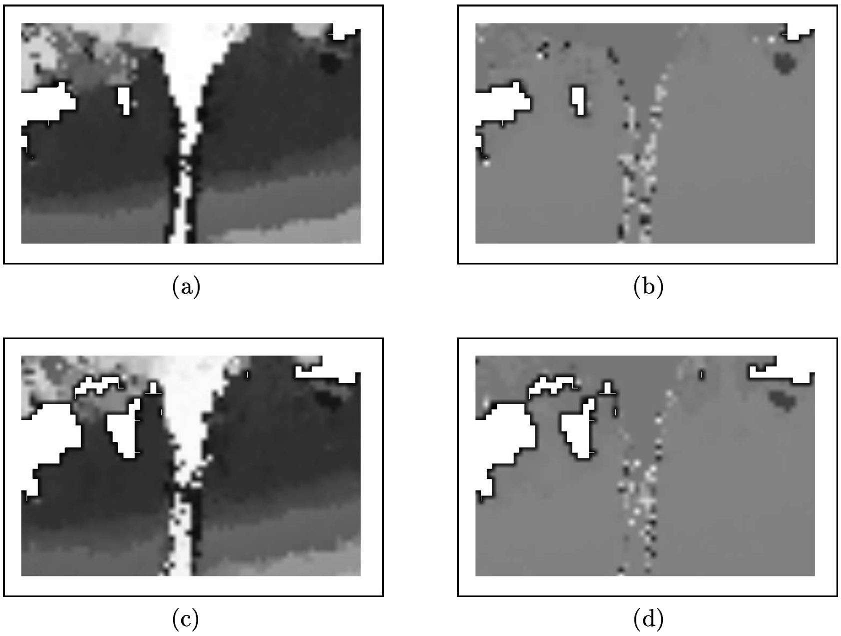

(a) (b) (c) (d)

Figure 7.14. Instantaneous optical flow field for the house sequence: (a) & (c) magnitude, (b) & (d) direction, the $G_{\frac{3w}{8}}$ and $G_{\frac{2w}{8}}$ Gaussian windowing functions, respectively.

This requires that the resultant Fourier component and its phase change, which forms the basis of the velocity computation, be decomposed into its constituent components, where each component corresponds directly to a distinct object, and these components and their phase changes then be used to identify the velocity. This is where the technique introduced in Chapters 3 and 4 comes in and we will use it explicitly in the next chapter to improve the quality of the computed optical flow field, particularly in these regions.

4.2 THE APERTURE PROBLEM

All techniques for computing the optical flow of an image sequence get into difficulties when there is insufficient information contained in the image window to estimate the velocity. This is clearly the case if the image in that region is homogeneous and contains little or no variation in intensity. However, it also arises when the image window contains a single discontinuity or edge between two homogeneous regions (*e.g.* see Figure 7.15.) In such a situation, the only thing we can compute is what is referred to as the orthogonal component of the flow vector, *i.e.* the component in a direction at right angles to the direction of the edge.

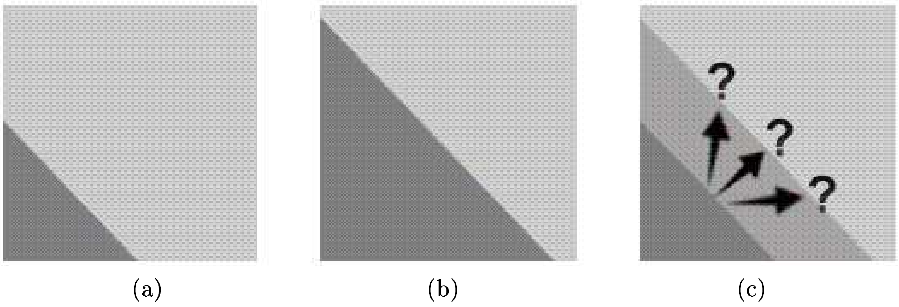


Figure 7.15. The aperture problem: (a) a discontinuity between two homogeneous regions at time t; (b) and at time $t + \delta t$; (c) its true velocity is ambiguous and all that can be computed with certainty is the orthogonal component of its velocity, *i.e.* the component in a direction which is normal to to the edge direction.

This is known as the aperture problem and it is exemplified well in the camera image sequence (Figure 7.16).

In particular, if we look at the region containing the prism (this is the triangular region at the top of the camera) we can see that the left edge of the prism separates two relatively homogeneous regions. The Fourier phase technique produces different results in this region depending on the size of the Gaussian windowing function. In the case of the windowing function with 50% weight $\frac{2w}{8}$ pixels from the centre (w, the window size, is 64 pixels), we can see that the technique fails to compute the true flow (which is horizontal, from left to right) and it produces instead the normal flow in a direction perpendicular to the edge of the prism.

4.3 ACCURACY

What of the accuracy of the technique? Assuming that we are using a reasonably-sized window, the accuracy is dependent on the resolution of the Hough accumulator since it effectively samples the velocity space. We have adopted, somewhat arbitrarily, a velocity sampling period of 0.1 pixel (*i.e.* the interpixel-distance in the velocity Hough space is equivalent to 0.1 pixels/frame). Figures 7.2 (e) and 7.2 (f), together with tables 7.1 and 7.2, show clearly that the technique is indeed capable of consistent computation of the flow to within this tolerance, at least for the trivial flow field shown in Figure 7.2. For a more demanding assessment of the technique, we have used Otte and Nagel's [257] benchmark sequence. This sequence has the major benefit that ground truth optical flow is available (*i.e.* the magnitude and direction of the optical flow of almost every point in the image). To compare the optical flow computed with the algorithm presented in this paper and ground-truth, the optical

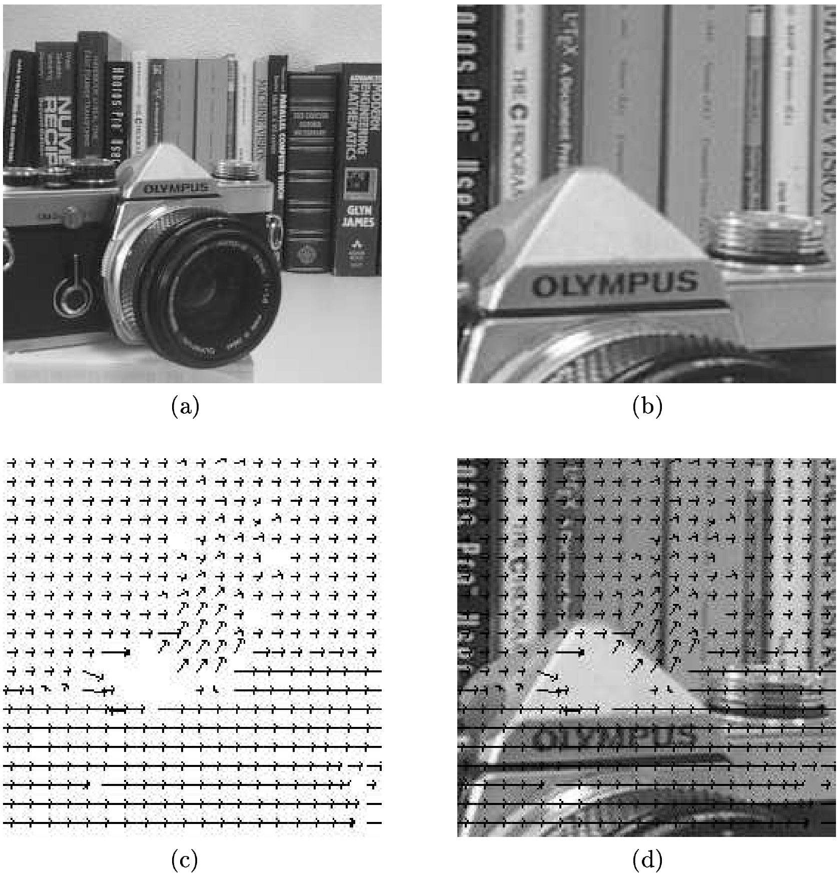

(a)
(b)
(c)
(d)

Figure 7.16. An example of the aperture problem: (a) frame 1 of the camera sequence; (b) detail from the sequence showing two reasonably uniform regions separated by a straight contour; (c) and (d) optical flow field computed using the $G_{\frac{2w}{8}}$ windowing function — only the orthogonal component has been extracted near the sloped edge.

flow was estimated every ten pixels (for the three Gaussian weighting functions) and then a complete optical flow image was produced for both magnitude and direction by interpolating bi-linearly among these points (see Figures 7.4 and 7.5). These were then compared to the ground-truth magnitude and direction images by estimating the RMS, Fleet, and Nagel errors (see Table 7.2).

Referring to these images and tables, a number of points can be noted.

First, it is clear that the main errors occur, as one would expect, at the occluding contours and, in particular, at the contour where the two objects are moving with significantly velocities (as, for example, is the case with the white block and the large dark block in the foreground). Again, as expected, this error is greater for the wider weighting functions and, because the velocity estimate is based on a larger effective support, the error propagates into a bigger region around the occluding contour.

Second, the RMS, Fleet, and Nagel error images in Figure 7.6 all show a consistent and almost constant error (except for the points close to occluding contours). Clearly, there is a bias in the measurement (assuming, as we must, that the ground-truth data is correct). On the other hand, it also has to be said that the algorithm reliably and consistently produces the correct magnitude and direction (close to 0.1 pixel) when tested on artificial test images *i.e.* the gull sequence.

5. CONCLUSIONS

Notwithstanding the apparent bias of the results of the approach when tested with Otte and Nagel's ground truth data, the technique presented produces dense, consistent, and accurate instantaneous optical flow fields. The use of a Gaussian weighting function which provides a 50% weighting at $\frac{2w}{8}$ pixels from the centre of the window in which the velocity estimate is being computed provides a good compromise between distortion introduced due to boundary-effects and the inability to compute an estimate due to the small support of the estimate.

The major problem of the approach is that it produces at best an aggregate velocity estimate in regions comprising objects with two distinct velocities (such as in the local region around occluding contours). We will deal with this problem in the next chapter.

6. POSTSCRIPT: OTHER APPROACHES

The measurement of optical flow has received a great deal of attention by the computer vision community and the literature is replete with a very large number of publications on the topic. Surveys and comparisons of the different approaches can be found in comprehensive works by, for example, Barron *et al.* [29] and Otte and Nagel [257], while Redert *et al.* present a very accessible overview in [283]. References to other approaches can be found in the bibliography at the end of the book.

Most approaches to the measurement of optical flow in images normally exploit one of two primary techniques. The first involves the computation of the spatio-temporal derivatives (first-order or second-order), differentiating the (filtered or unfiltered) image sequence with respect to time and thus computing the optical flow field (*e.g.* [102]). The second involves either feature- or region-based matching (such as normalized cross-correlation) of local iconic information such as raw image data or segmented features or objects (*e.g.* [376]). Comparisons of the many variations of these approaches and the relationship between them can be found in [29, 257, 348, 160].

An alternative approach (which is directly related to the technique presented in this chapter) exploits the regularity in spatio-temporal frequency representations of the image, such as the spatio-temporal Fourier Transform Domain, resulting from certain types of image motion. It can be shown that the spatio-temporal Fourier Transform of an image sequence in which the image content is moving with constant velocity results in a spatio-temporal frequency representation which is equal to the spatial Fourier Transform of the first image multiplied by a δ-Dirac function in the temporal-frequency domain. This δ-Dirac function is dependent of the image velocity which can be computed if one knows the position of the δ-Dirac function and any spatial frequency [64]. Because this approach is based on image motion, rather than object motion, it normally assumes uniform (zero) background when evaluating object motion. Extensions of the technique have been developed to allow it to cater for situations involving noisy backgrounds [65], several objects [211, 210], and non-uniform cluttered backgrounds [280].

The use of spatio-temporal frequency representations for the measurement of optical flow has been developed in depth by Fleet and Jepson [29, 111, 112, 113] who have extended the spatio-temporal frequency framework to deal with situations where the normal assumption of a single pure (local) image translation is no longer valid. Their technique, which is based on the application of a bank of spatio-temporal band-limited velocity-tuned linear filters, is able to distinguish different velocities within a given neighbourhood and it is resilient to small affine geometric deformations of the image neighbourhood.

Chapter 8

DECOUPLED OPTICAL FLOW

1. THE PROBLEM OF NON-UNIQUE MULTIPLE LOCAL VELOCITIES

We saw in the last chapter that we can compute the instantaneous optical flow between two images acquired at different times by analyzing the phase of the Fourier tranform of Gaussian-windowed regions in the image. This technique makes the assumption that there is one unique velocity associated with the region in question, *i.e.* that the entire region is translating uniformly with one velocity. In such circumstances, it works very well indeed, producing an arbitrarily dense accurate flow field. However, as has been pointed out by several people and as we saw ourselves in the previous chapter, these circumstances don't always prevail: there are instances where two or more velocities can and should be associated with a single local region, for example at occluding boundaries and with transparent surfaces in motion. As one would expect, the technique presented in the previous chapter does encounter problems in such cases. The goal of this chapter is to show how the theory presented in Chapters 3 and 4 can be deployed to produce better estimates of optical flow, especially in regions around occluding boundaries, by decoupling the components which give rise to the multiple velocities. We will also look at some of the limitations of deploying this approach and we will conclude with a comparison of the instantaneous and the decoupled optical flow estimation techniques in terms of their relative strengths and weaknesses.

2. ALGORITHM

The algorithm for decoupled optical flow is similar to that for instantaneous optical flow except that, instead of looking at the phase difference between the Fourier transform of each windowed region at times t_0 and t_1, we use the image segmentation technique described in Chapters 3 and 4 and apply it to each windowed region at times t_0, t_1, t_2, and t_3. Letting $g_{t_i}(x, y)$ represent a windowed region at time t_i, we compute the Fourier transform of each region $g_{t_i}(x, y)$ yielding $\mathsf{G}_{t_i}(\omega_x, \omega_y)$ and then we decouple each phasor to find $\mathsf{G}^1_{t_0}(\omega_x, \omega_y)$, $\mathsf{G}^2_{t_0}(\omega_x, \omega_y)$, $\Delta\Phi^1_{t_0}(\omega_x, \omega_y)$, and $\Delta\Phi^2_{t_0}(\omega_x, \omega_y)$. Since we are not interested in reconstructing the decoupled images $g^1_{t_0}(x, y)$ and $g^2_{t_0}(x, y)$, we can discard $\mathsf{G}^1_{t_0}(\omega_x, \omega_y)$ and $\mathsf{G}^2_{t_0}(\omega_x, \omega_y)$, and we just use $\Delta\Phi^1_{t_0}(\omega_x, \omega_y)$ and $\Delta\Phi^2_{t_0}(\omega_x, \omega_y)$ as input to the Hough transform equation 3.20 to find the velocity as described in Chapter 3. In this way, we can extract two velocities in one local region.

3. EXAMPLES

3.1 SIMPLE GULL SEQUENCE

Figure 8.1 shows the result of applying the technique to an extended version of the gull sequence shown in the previous chapter. In this sequence, the image is translating by two pixels per frame in both the x and y directions by 2 pixels. Figure 8.1 (a) shows the test image. Figure 8.1 (b) shows the computed flow field using Gaussian weighting functions with 50% weight at $\frac{2w}{8}$ pixels from window centre (w = window size). As before, all of the results shown in this chapter were computed with a window size of 64×64 pixels. The flow field has been computed at ten-pixel intervals.

As before, Figures 8.1 (c) and (d) show the errors in the computed flow field according to the Fleet and the Nagel error metrics, respectively. These error images were produced by computing the error of the flow field at five-pixel intervals and then interpolating to complete the image.

Table 8.1 summarizes the mean and standard deviation of the Fleet and Nagel error metrics, together with the Root Mean Square (RMS) error of the magnitude and direction values. For the sake for comparison, the error metrics for the instantaneous flow computation are also included. We can see that the decoupled approach outperforms the instantaneous approach for both the $G_{\frac{3w}{8}}$ and $G_{\frac{2w}{8}}$ windowing functions but, significantly, not for the $G_{\frac{w}{8}}$ windowing function. This degradation in performance is due to the fact that the decoupled approach requires a sequence of four images. With the narrower $G_{\frac{w}{8}}$ windowing function, the change in image information over four frames is relatively greater and, hence, one the assumptions upon which the technique is based (*i.e.*

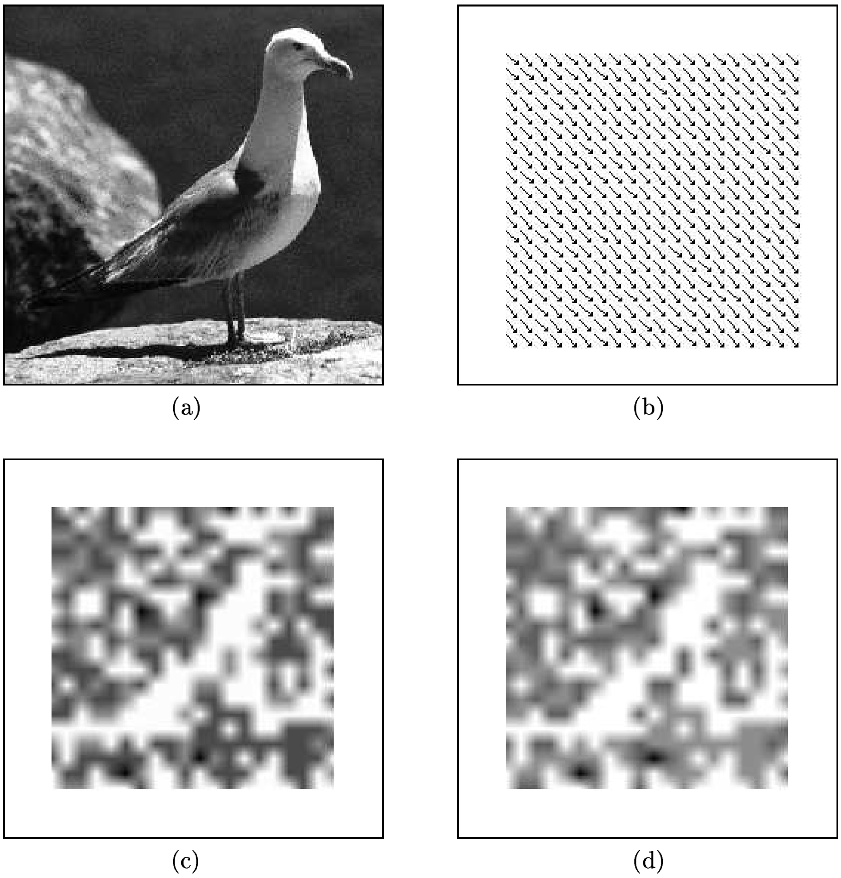


Figure 8.1. (a) Simple test sequence: optical flow is computed using this image and this image translated by 2 pixels in the x and y direction. (b) Optical flow field computed using phase information ($G_{\frac{2w}{8}}$ Gaussian windowing function). (c) and (d) Error images for the Fleet and Nagel error metrics, respectively (refer to Table 8.1 for details).

that there is no spectral change) is violated to a greater extent than in the case of the wider windowing functions. This is one of the most serious limitations of the decoupled approach: because it requires a longer sequence of images, it performs well only with relative low velocities and/or with wider windowing functions. On the other had, because the instantaneous approach requires a sequence of only two images, it can deal with larger velocities before the change in spectral content causes it to fail.

Gaussian	RMS Error		Fleet Error		Nagel Error	
Weighting	Magnitude	Direction	Mean	Std. Dev.	Mean	Std. Dev.
	(pixels)	(degrees)	(degrees)	(degrees)	(pixels)	(pixels)
Instantaneous Optical Flow						
$\frac{3w}{8}$	0.21	8.34	2.02	4.61	0.12	0.25
$\frac{2w}{8}$	0.10	1.44	1.48	0.25	0.11	0.05
$\frac{w}{8}$	0.31	6.34	2.83	3.99	0.23	0.30
Decoupled Optical Flow						
hline $\frac{3w}{8}$	0.21	11.74	1.31	3.75	0.09	0.24
$\frac{2w}{8}$	0.08	0.88	0.75	0.46	0.07	0.06
$\frac{w}{8}$	0.44	4.53	5.11	1.18	0.47	0.10

Table 8.1. Summary of error metrics for the gull image translation benchmark.

3.2 OTTE AND NAGEL TEST SEQUENCE

The improvement in performance of the technique compared to the instantaneous optical flow approach is even better with the Otte and Nagel ground truth test sequence (see Figures 8.2 through 8.5.) Qualitatively, both the flow fields (Figures 8.2 (d), (e), and (f)) and the interpolated magnitude images (Figures 8.3 (b), (c), and (d)) and interpolated direction images(Figures 8.4 (b), (c), and (d)) are better for the decoupled approach compared with the instantaneous approach. This is confirmed by the quantitative error metrics (*i.e.* the RMS, Fleet, and Nagel error measures) shown in Table 8.2 where we can see that the decoupled approach produces better results for all three metrics and for all three windowing functions. The decoupled approach works better here with the narrow $G_{\frac{w}{8}}$ windowing function that it did in the gull sequence because there is a strong preponderance of points with small velocities.

Figure 8.5 shows the Fleet and Nagel error images for the sequence using the three Gaussian windowing functions. While there are still errors around the occluding boundaries, they are much more localized. In fact, if we compute the optical flow at every pixel on row 255 (see Figure 8.6) using the decoupled approach, we can see that there is a much sharper discontinuity in the flow fields close to the occluding boundaries compared to the corresponding results obtained using the instantaneous technique (see Figures 8.7 and 8.8).

So why are there residual errors in the flow fields close to the occluding boundaries? Part of the reason is that, although we do compute two velocity vectors in each region using the decoupled approach, all of the above results show the 'stronger' vectors where by 'stronger' we mean

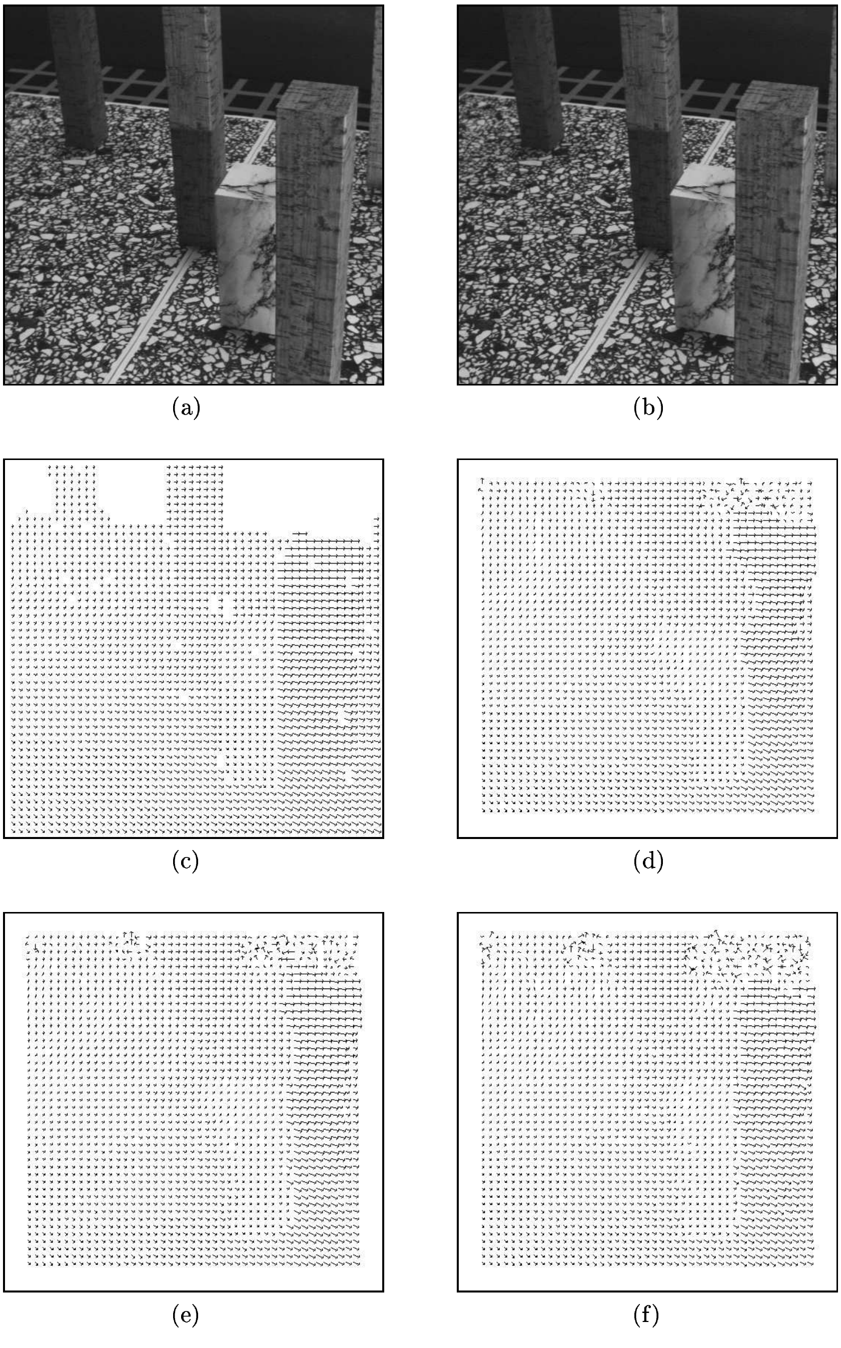


Figure 8.2. (a) and (b) Images number 40 and 43 of Otte and Nagel's ground-truth motion sequence. (c) True optical flow field given by Otte and Nagel's ground-truth data (sampled every ten pixels). (d), (e), and (f) Optical flow field computed using $G_{\frac{3w}{8}}$, $G_{\frac{2w}{8}}$, and $G_{\frac{w}{8}}$ Gaussian windowing functions, respectively.

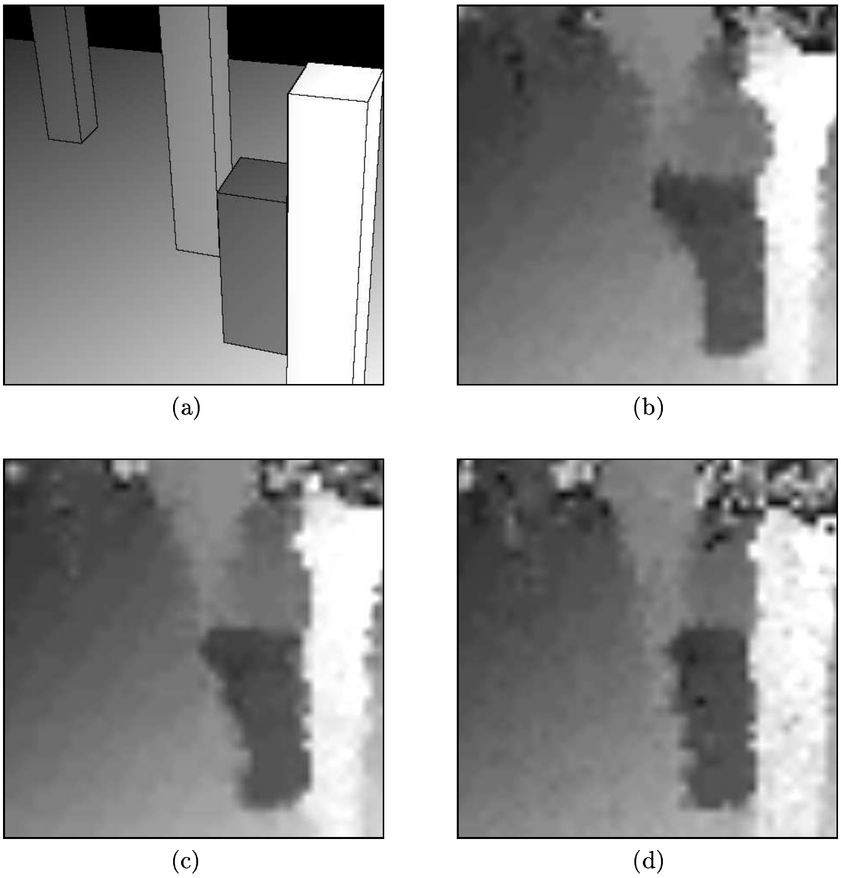


Figure 8.3. (a) Magnitude of the optical flow field extracted from Otte and Nagel's ground-truth data. (b), (c), and (d) Magnitude of the optical flow field computed using the $G_{\frac{3w}{8}}$, $G_{\frac{2w}{8}}$, and $G_{\frac{w}{8}}$ Gaussian windowing functions, respectively.

that they correspond to the largest local maximum in the Hough transform space; the second 'weaker' vector being discarded. Quite often, of course, and especially when there is little difference between the values of the two maxima, this weaker vector may well be the more consistent vector at that point and the stronger vector may correspond to the vector of the neighbouring region.

This is illustrated clearly in Figure 8.9 which shows both vectors in the vicinity of an occluding boundary (windowing function $G_{\frac{2w}{8}}$). Figure 8.10 shows the sequence of Hough transforms which are generated as we cross the occluding boundary shown in Figure 8.9. A Hough transform is shown for every second point on the central line of the image. We can see how a well-defined maximum (indicating one velocity in the region)

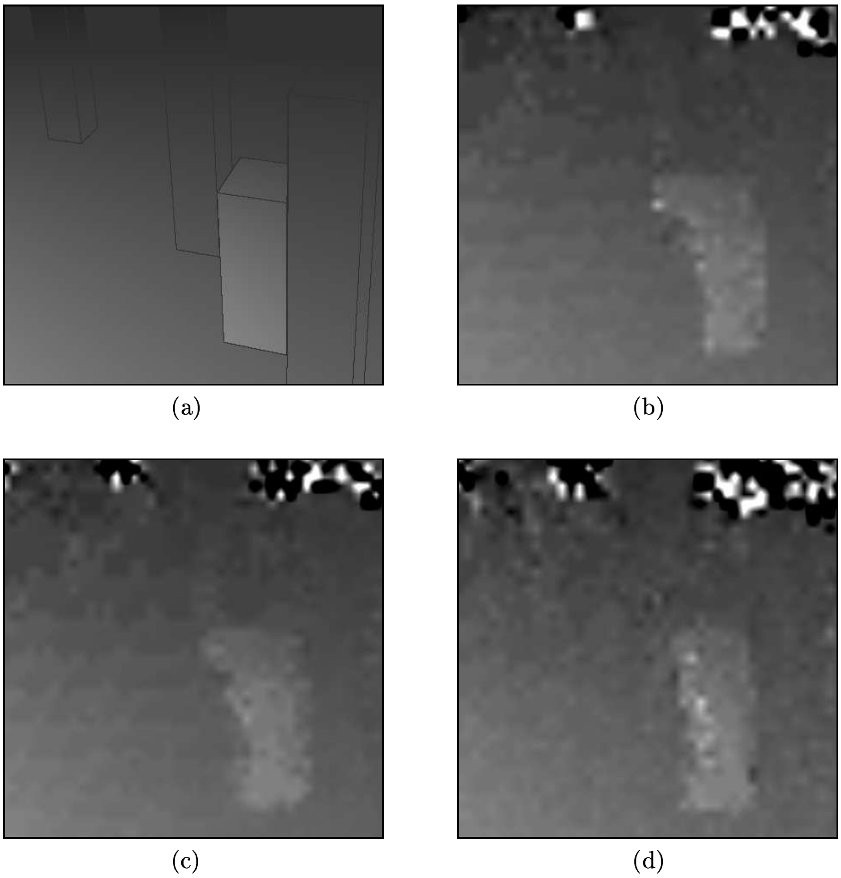
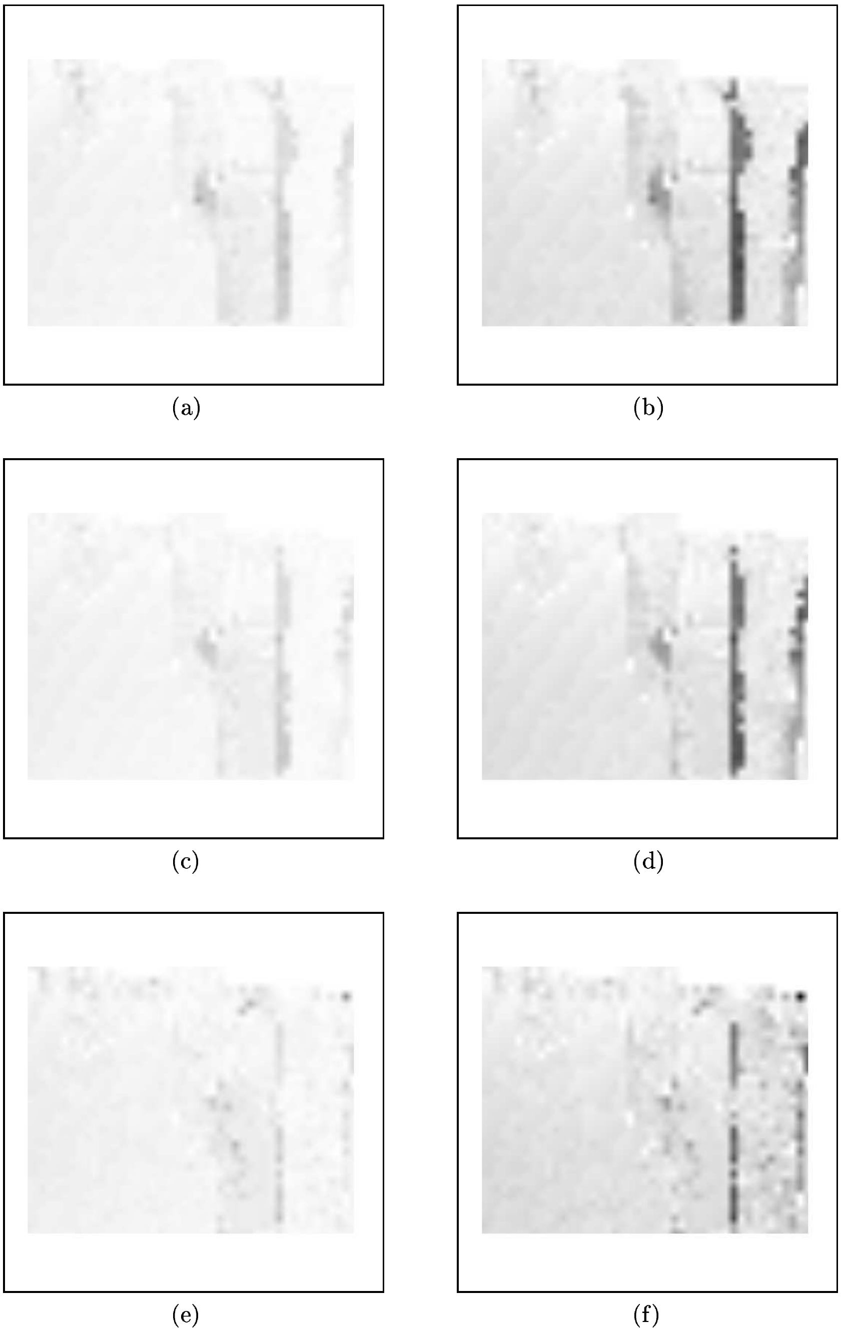


Figure 8.4. (a) Direction of the optical flow field extracted from Otte and Nagel's ground-truth data. (b), (c), and (d) Direction of the optical flow field computed using the $G_{\frac{3w}{8}}$, $G_{\frac{2w}{8}}$, and $G_{\frac{w}{8}}$ Gaussian windowing functions, respectively.

evolves into two less-well defined maxima (indicating two velocities in the region) as we approach the occluding boundary, and these evolve into one maximum again as we pass by the occluding boundary.

The lack of definition of the two maxima is caused by the occlusion in the region and the consequent violation of the assumption of purely additive signals upon which the technique is based. If the signals are additive, then the situation improves significantly. For example, Figure 8.11 shows two flow fields computed for an image sequence which is formed by the additive superposition of one image on another. In this case, we are able to extract two velocity vectors at every point and, furthermore, the Hough transform maxima are much better defined (see Figure 8.12).



(a) (b)

(c) (d)

(e) (f)

Figure 8.5. (a), (c), (e) Fleet error metric and (b), (d), (f) Nagel error metrics based on computed and ground-truth values using $G_{\frac{3w}{8}}$, $G_{\frac{2w}{8}}$, and $G_{\frac{w}{8}}$ Gaussian windowing functions, respectively. Note that darkness is proportional to error.

Gaussian Weighting	RMS Error		Fleet Error		Nagel Error	
	Magnitude (pixels)	Direction (degrees)	Mean (degrees)	Std. Dev. (degrees)	Mean (pixels)	Std. Dev. (pixels)
			Instantaneous Optical Flow			
$\frac{3w}{8}$	0.50	22.83	15.05	5.95	0.55	0.30
$\frac{2w}{8}$	0.44	21.01	14.12	5.24	0.51	0.25
$\frac{w}{8}$	0.45	24.90	15.13	7.14	0.55	0.30
			Decoupled Optical Flow			
$\frac{3w}{8}$	0.40	5.79	6.78	3.61	0.31	0.30
$\frac{2w}{8}$	0.37	5.40	6.43	3.09	0.29	0.26
$\frac{w}{8}$	0.34	7.48	7.22	3.13	0.31	0.21

Table 8.2. Summary of error metrics for the Otte and Nagel benchmark sequence.

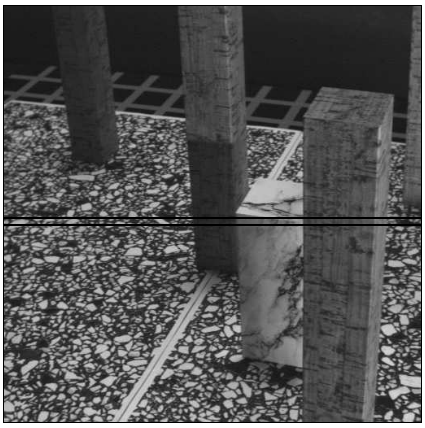


Figure 8.6. Row 255 of the Otte and Nagel sequence which is used to generate the optical flow components shown in Figures 8.7 and 8.8.

Clearly, we can improve the quality of the final flow fields by choosing the flow vectors which are more consistent with the structure of the scene. For example, one could choose the flow vectors to align the discontinuity in the flow field with a discontinuity in the image intensity.

Note that in Figures 8.9 and 8.11 we have shown the secondary vector only if the relative heights of their corresponding maxima in the Hough transform are within 30% of each other, otherwise we would have included spurious vectors in regions where there is really only one velocity.

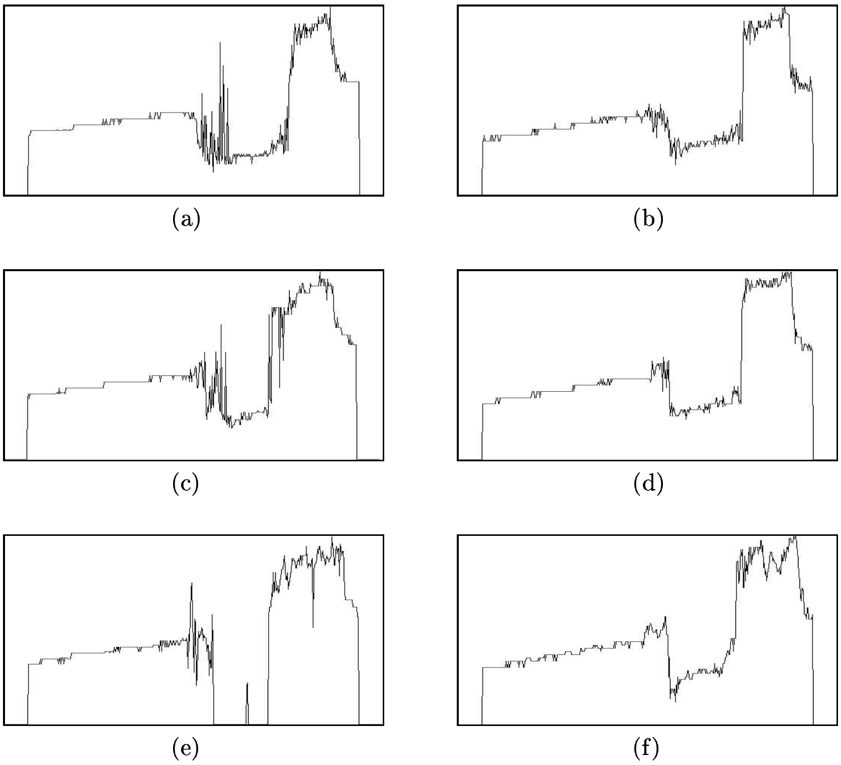


Figure 8.7. Magnitude of the optical flow vector measured at every point on row 255 of the Otte and Nagel sequence — (a), (c), (e) Instantaneous optical flow and (b), (d), (f) Decoupled optical flow, using $G_{\frac{3w}{8}}$, $G_{\frac{2w}{8}}$, and $G_{\frac{w}{8}}$ Gaussian windowing functions, respectively.

3.3 SUNFLOWER GARDEN TEST SEQUENCE

Finally, we come to the third scenario: the sunflower garden sequence provided by the Berkeley Multimedia Research Center, University of California, Berkeley.

The results for the extract with the tree juxtaposed directly in front of the house are shown in Figures 8.13 and 8.14. The limitations we mentioned previously regarding the velocities for which the decoupled approach will work are now immediately apparent. The decoupled approach has failed completely to extract the flow field for the high-velocity foreground tree-trunk and, instead, it only extracts the low-velocity background flow field. We get similar results for the other two extracts we looked at in the previous chapter. Since the high branches in the

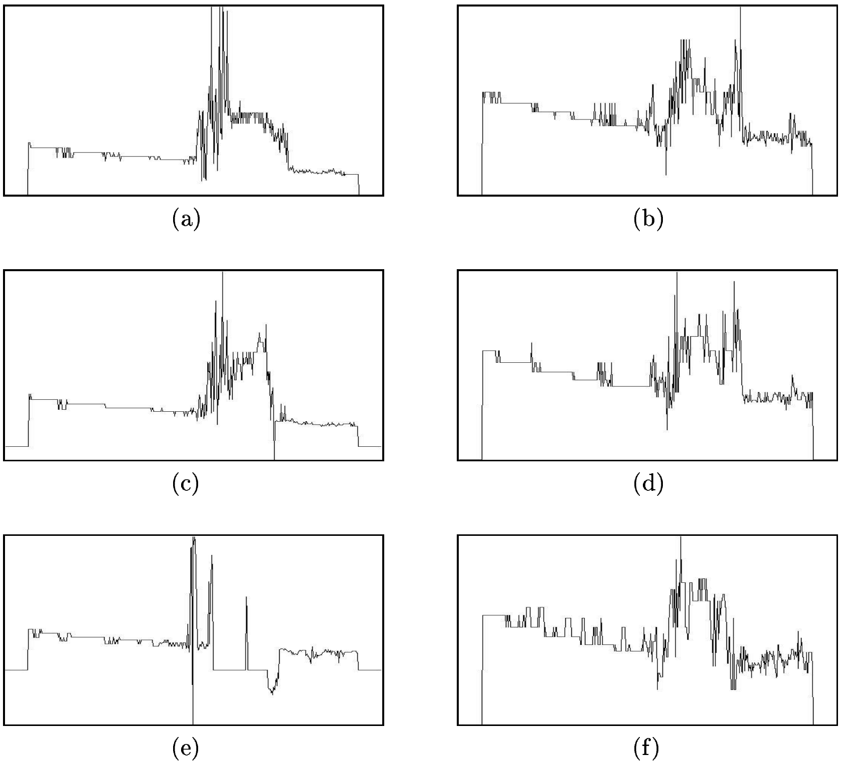


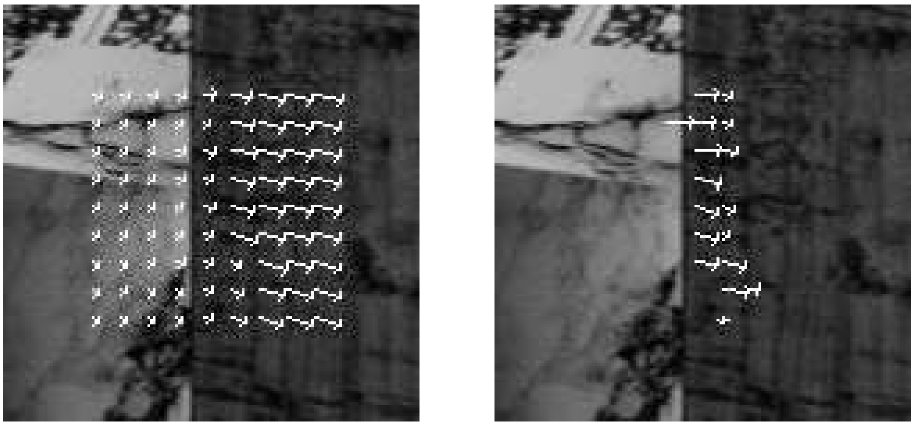
Figure 8.8. Direction of the optical flow vector measured at every point on row 255 of the Otte and Nagel sequence — (a), (c), (e) Instantaneous optical flow and (b), (d), (f) Decoupled optical flow, using $G_{\frac{3w}{8}}$, $G_{\frac{2w}{8}}$, and $G_{\frac{w}{8}}$ Gaussian windowing functions, respectively.

tree are further in the distance, and hence have a lower velocity, the decoupled approach does extract their velocity vectors correctly.

In the previous chapter, we improved the results for this sequence by increasing the resolution of the image, effectively reducing the relative size of the window to improve the localization of the flow field. Attempting to do the same here with the decoupled approach only makes matters worse because we are effectively increasing the object velocity and, as we have seen, this causes major problems with change in spectral content over the four images used in this approach (see Figures 8.15 and 8.16).

4. CONCLUSION

So, having presented two different Fourier-based techniques for computing optical flow, where do we stand?



(a) (b)

Figure 8.9. Dual flow fields in the vicinity of an occluding boundary (a) primary flow field (b) secondary flow field ($G_{\frac{2w}{8}}$ Gaussian windowing function).

The instantaneous approach, requiring only two images, is clearly more resilient when dealing with higher velocities. However, because it assumes a single unique velocity in the windowed region, it has problems when dealing with regions containing two or more objects moving with different velocities such as in the vicinity of occluding boundaries.

On the other hand, the decoupled approach, requiring four image samples, produces significantly better results around occluding boundaries but at the cost of reducing the range of velocities with which it can operate: high velocity means greater change in image and spectral content and, therefore, greater violation of the assumption on which both techniques are based. In addition to its improved performance in the vicinity of occluding boundaries, the decoupled approach also yields more accurate flow vectors, again by virtue of the extended number of image samples which it uses in the computation.

So which is best? It depends — if one has a low velocity image sequence, the decoupled approach is clearly superior but if the velocity increases much beyond four or five pixels per frame, the instantaneous technique is the better choice.

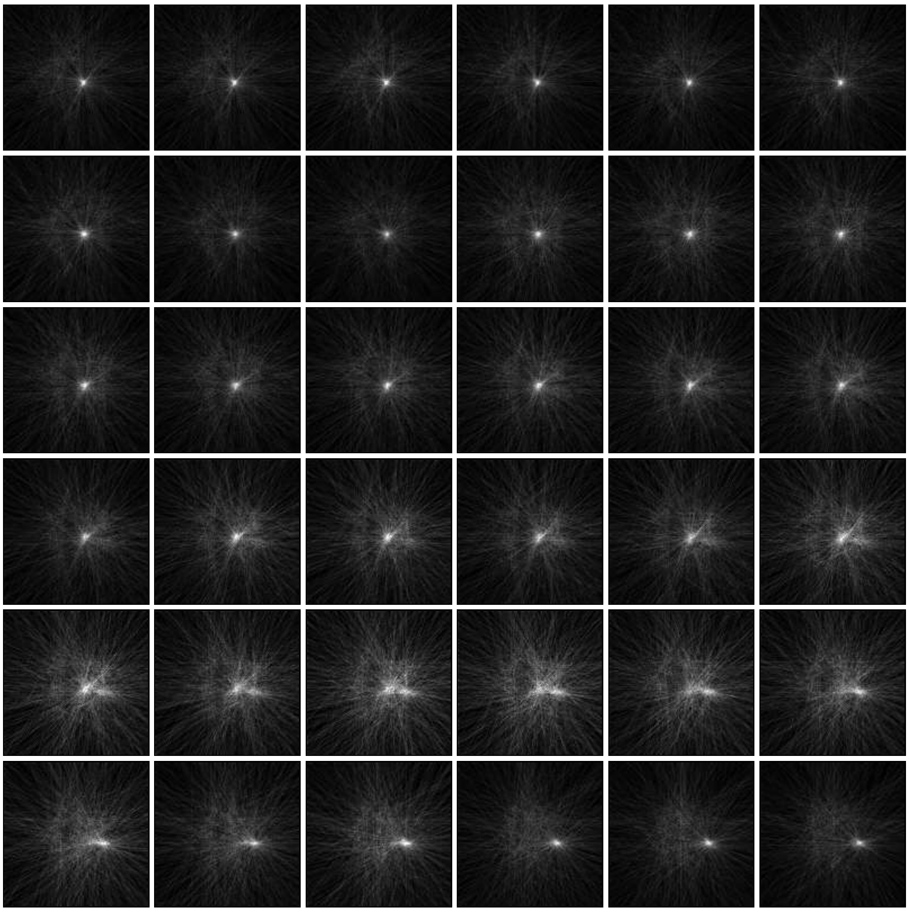


Figure 8.10. Sequence of Hough transforms produced as the window in which optical flow is being estimated passes over the occluding boundary shown in Figure 8.9.

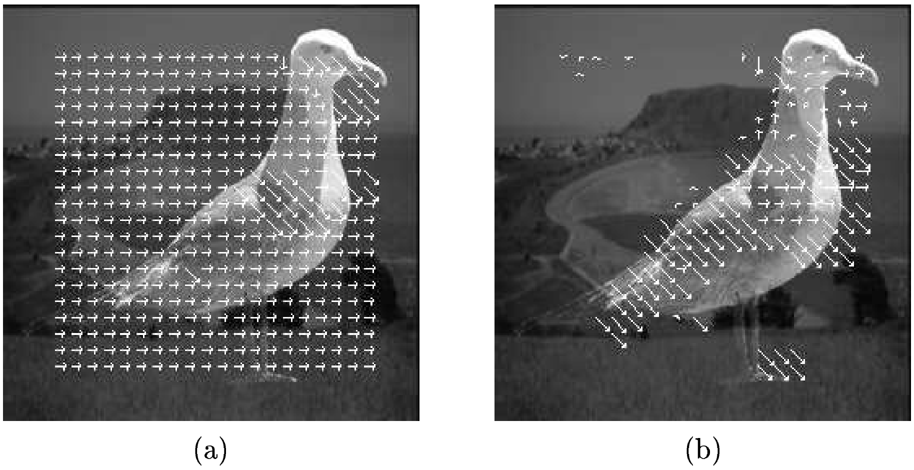


Figure 8.11. Dual flow fields in superimposed images (a) primary flow field (b) secondary flow field ($G_{\frac{2w}{8}}$ Gaussian windowing function).

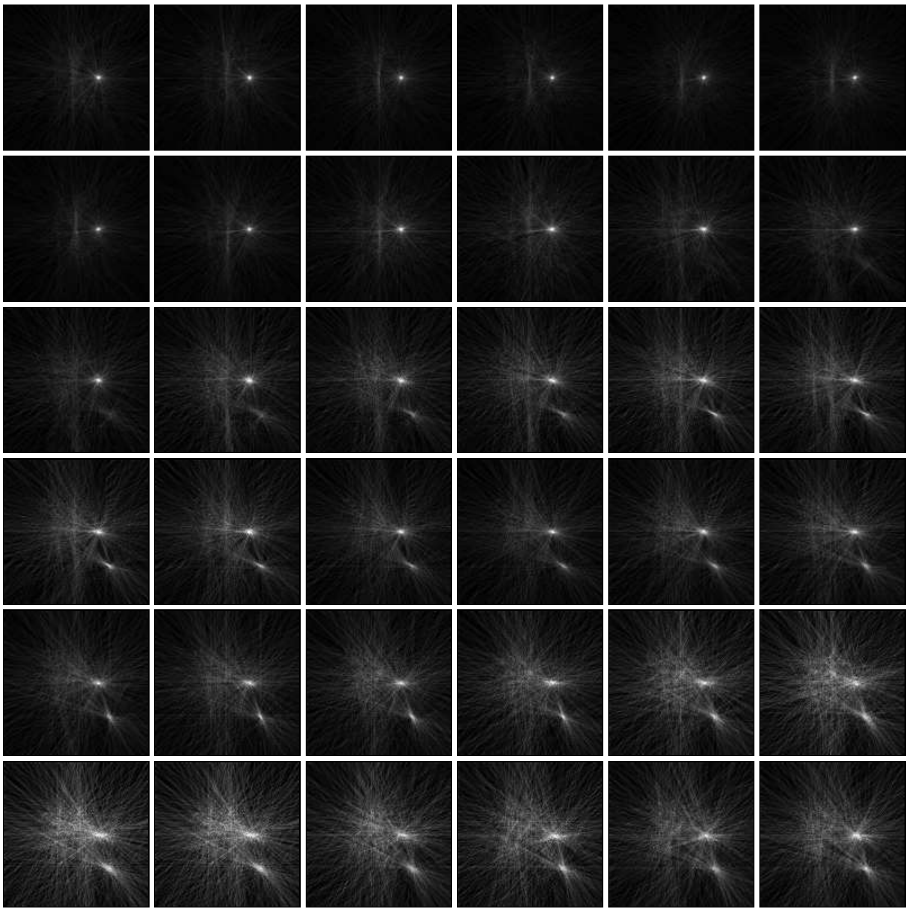


Figure 8.12. Sequence of Hough transforms produced as the window in which optical flow is being estimated passes over the occluding boundary shown in Figure 8.11.

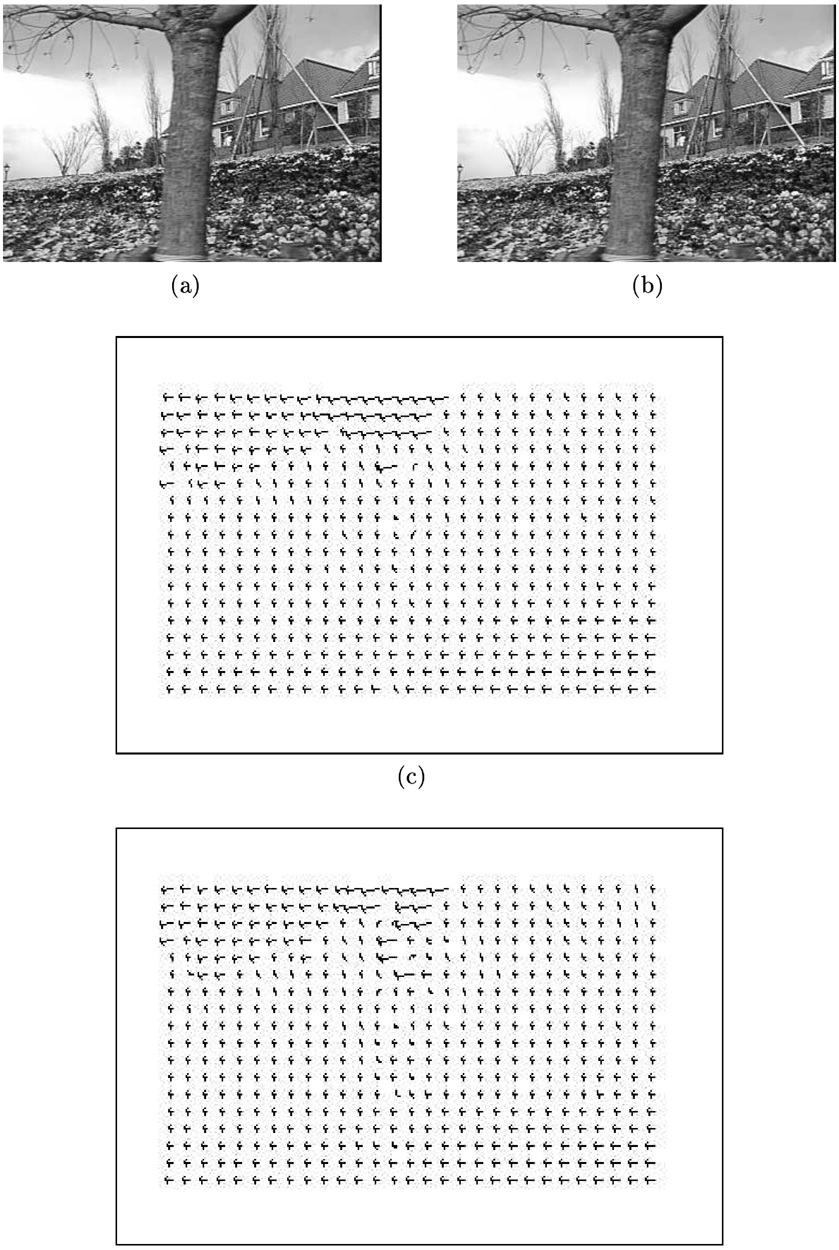


Figure 8.13. Decoupled optical flow field for the SFLOWG sequence: (a) frame 10; (b) frame 11; (c) and (d) optical flow field computed using the $G_{\frac{3w}{8}}$ and $G_{\frac{2w}{8}}$ Gaussian windowing functions, respectively. Image sequence courtesy of the Berkeley Multimedia Research Center, University of California, Berkeley.

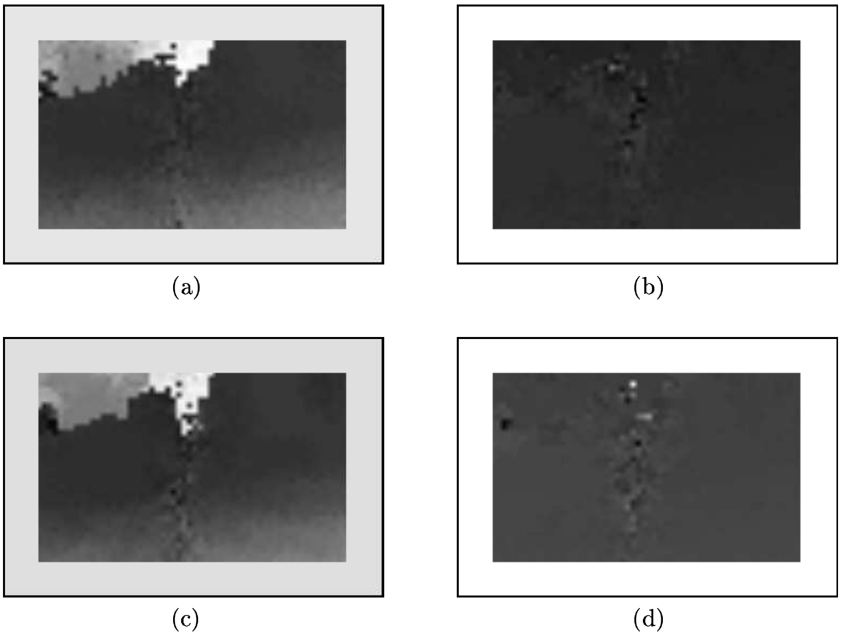
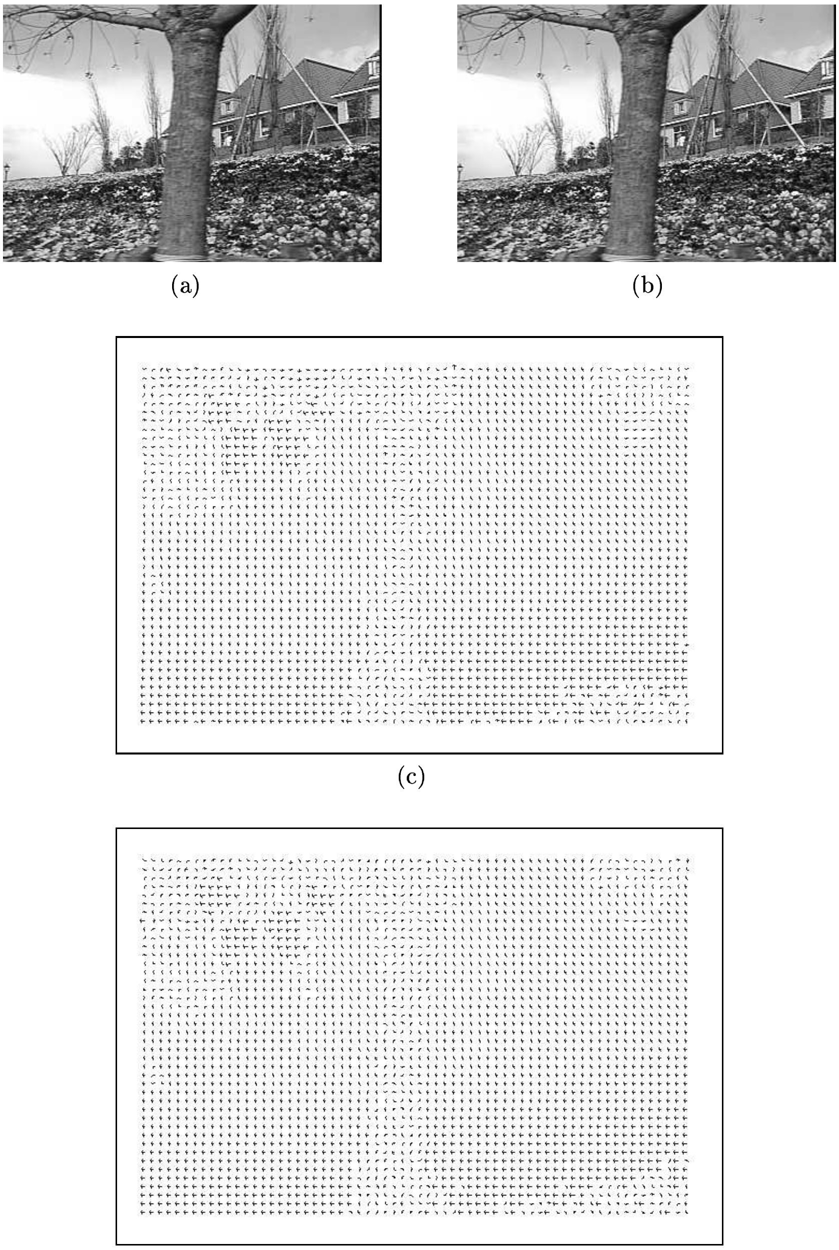


Figure 8.14. Decoupled optical flow field for the SFLOWG sequence: (a) & (c) magnitude, (b) & (d) direction, the $G_{\frac{3w}{8}}$ and $G_{\frac{2w}{8}}$ Gaussian windowing functions, respectively.



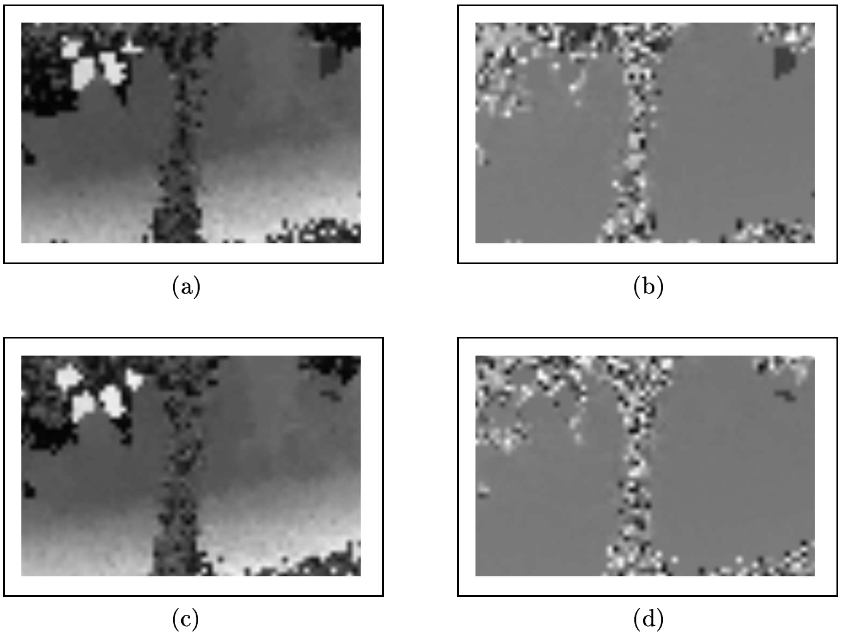
(a)

(b)

(c)

(d)

Figure 8.15. Decoupled optical flow field for the SFLOWG sequence: (a) frame 10; (b) frame 11; (c) and (d) optical flow field computed using the $G_{\frac{3w}{8}}$ and $G_{\frac{2w}{8}}$ Gaussian windowing functions, respectively. Image sequence courtesy of the Berkeley Multimedia Research Center, University of California, Berkeley.



(a)

(b)

(c)

(d)

Figure 8.16. Decoupled optical flow field for the house sequence: (a) & (c) magnitude, (b) & (d) direction, the $G_{\frac{3w}{8}}$ and $G_{\frac{2w}{8}}$ Gaussian windowing functions, respectively.

Chapter 9

EPILOGUE

In this book, we have travelled a short but eventful journey from Fourier and Hough transforms, through separation of additive images, segmentation of figure-ground scenes, via object motion, camera motion, and binocular stereopsis, to instantaneous and decoupled optical flow estimation. The goal of this journey was to present the development of a new perspective on these areas and to develop an approach with offers a small but potentially significant advance in dealing with these issues.

The key feature of the approach is that the separation/segmentation is accomplished in the Fourier domain so that we are not concerned with the visual appearance of the scene. The strengths of the technique are that it explictly models the difficult situation of decoupling additive signals and makes no assumptions about the spatial or spectral content of the constituent images (unlike other techniques such as independent component analysis). It works with several camera configurations, such as camera motion and short-baseline binocular stereo. The approach works best on images with small velocities/displacements, typically zero to ten pixels per frame.

Of course, like all other approaches, this one has weaknesses too. For example, it assumes no scale change in the scene so that if objects are moving toward or away from the camera then one has to enforce a very short sampling period so that the velocities are small and the scale changes negligible. If there is significant rotational motion, then the approach will fail since the requirement of constant translationary motion is not satisfied.

However, even is such situations we can still apply the technique on a local basis to compute the optical flow and hence infer the structure of the scene (assuming some knowledge of the camera configuration). Naturally, one can compute optical flow in many other ways, and even by using Fourier techniques to analyse the phase difference between two consecutive images. However, the approach developed in this book has the advantage that it allows you to compute optical flow more accurately and more reliably in regions where there are actually two distinct motions (*e.g.* in regions around occluding boundaries and in situations where we might have reflective transparent surfaces such as a scene viewed through a window).

The disadvantage of the decoupled optical flow technique is that it requires four image samples rather than the minimal two and, hence, it can't deal with large velocities because the amount of image data passing into and out of the measurement window will be too great and will cause the technique to fail due to excessive change in spectral content.

The motivation for taking this trip was to present the full theory of the technique together with some examples so that others with applications which exhibit the characteristics of these examples can consider the use of the approach in their domain. In the end, I don't imagine for a moment that this new technique will change the worlds of segmentation and optical flow estimation but it may well provide a way of dealing with some problems which would otherwise have been difficult to tackle, perhaps in a specialized area of computer vision — scenes with superimposed reflections or optical flow estimation near occluding boundaries, for example — and perhaps also in other areas such as acoustics, medical imaging, and signal processing — separating the voice signals of two simultaneous speakers or segmentation in digital angiography, for example.

References

[1] Inc. 3-D Video. Manufacturer of the nuview stereo camcorder adaptor. http://www.3-dvideo.com.

[2] M. Accame, F. G. B. De Natale, and D. D. Giusto. High performance hierarchical block-based motion estimation for real-time video coding. *Real-Time Imaging*, 4(1):67–79, February 1998.

[3] E. H. Adelson. Mechanisms for motion perception. *Optics and Photonics News*, 2(8):24–30, 1991.

[4] E. H. Adelson and J. R. Bergen. Spatiotemporal energy models for the perception of motion. *J. of the Optical Society of America A*, 2(2):284–299, 1985.

[5] E. H. Adelson and J. Y. A. Wang. Single lens stereo with a plenoptic camera. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):99–106, 1992.

[6] G. Adiv. Determining three-dimensional motion and structure from optical flow generated by several moving objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 7(4):384–401, 1985.

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

[8] J. K. Aggarwal and N. Nandhakumar. On the computation of motion from sequences of images - A review. *Proceedings of the IEEE*, 76(8):917–935, 1988.

[9] R. C. Agrawal, R. K. Shevgaonkar, and S. C. Sahasrabudhe. A fresh look at the Hough transform. *Pattern Recognition Letters*, 17:1065–1068, 1996.

[10] N. Ahuja and A. L. Abbott. Active stereo: Integrating disparity, vergence, focus, aperture, and calibration for surface estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(10):1007–1029, 1993.

[11] J. Aisbett. Optical flow with an intensity-weighted smoothing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(5):512–522, May 1989.

[12] S. Alibhai and S. W. Zucker. Contour-based correspondence for stereo. In *Computer Vision – ECCV 2000*, volume 1842 of *Lecture Notes in Computer Science*, pages 314–330, Berlin, 2000. Springer-Verlag.

[13] Y. Altunbasak, P. E. Eren, and A. M. Tekalp. Region-based parametric motion segmentation using color information. *Journal of Graphical Models and Image Processing*, 60(1):13–23, January 1998.

[14] S-I. Amari and A. Cichocki. Adaptive blind signal processing – neural network approaches. *Proceeding of the IEEE*, 86(10):2026–2048, 1998.

[15] P. Anandan. A computational framework and an algorithm for the measurement of visual motion. *International Journal of Computer Vision*, 2(3):283–310, 1989.

[16] N. Ancona and T. Poggio. Optical flow from 1-D correlation: application to a simple time-to-crash detector. *International Journal of Computer Vision*, 14(2):131–146, 1995.

[17] B. L. Anderson and K. Nakayama. Toward a general theory of stereopsis: Binocular matching, occluding contours, and fusion. *Psych. Review*, 101:414–445, 1994.

[18] A. A. Argyros, P. E. Trahanias, and S. C. Orphanoudakis. Robust regression for the detection of independent 3D motion by a binocular observer. *Real-Time Imaging*, 4(2):125–141, April 1998.

[19] N. Ayache. *Artificial Vision for Mobile Robots: Stereo Vision and Multisensory Perception.* The MIT Press, Cambridge, MA, 1991.

[20] N. Ayache and F. Lustman. Fast and reliable passive trinocular stereovision. In *Proc. Int. Conference on Computer Vision*, pages 422–427, 1987.

[21] S. Ayer, P. Schroeter, and J. Bigün. Segmentation of moving object by robust motion parameter estimation over multiple frames. In *Proc. 3rd European Conference on Computer Vision*, volume II, pages 317–327. Springer-Verlag, 1994.

[22] A. Azarbayejani, T. Starner, B. Horowitz, and A. P. Pentland. Visually controlled graphics. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(6):602–605, June 1993.

[23] A. Bainbridge-Smith and R. G. Lane. Measuring confidence in optical flow estimation. *Electronics Letters*, 32(10):882–884, 1996.

[24] A. Bainbridge-Smith and R. G. Lane. Determining optical flow using a differential method. *Image and Vision Computing*, 15(1):11–22, January 1997.

[25] D. H. Ballard. Generalizing the Hough tranform to detect arbitrary shapes. *Pattern Recognition*, 13(2):111–122, 1981.

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

[27] N. Barnes and G. Sandini. Directional control for an active docking behaviour based on the rotational component of log-polar optic flow. In D. Vernon, editor, *Computer Vision – ECCV 2000*, volume 1843 of *Lecture Notes in Computer Science*, pages 167–181, Berlin, 2000. Springer-Verlag.

[28] Y. Barniv. Error analysis of combined optical-flow and stereo passive ranging. *IEEE Transactions on Aerospace and Electronic Systems*, 28(4):978–989, 1992.

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

[30] J. L. Barron and A. Liptay. Optical flow to measure minute increments in plant growth. *Bioimaging*, 2(1):57–61, 1994.

[31] F. Bartolini, V. Cappellini, C. Colombo, and A. Mecocci. Multiwindow least-squares approach to the estimation of optical flow with discontinuities. *Optical Engineering*, 32(6):1250–1256, 1993.

[32] F. Bartolini, V. Cappellini, and A. Mecocci. An optical flow algorithm preserving motion boundaries and its parallel implementation. *European Transactions on Telecommunications and Related Technologies*, 3(6):583–591, 1992.

[33] F. Bartolini and A. Piva. Median based relaxation of smoothness constraints in optic flow computation. *Pattern Recognition Letters*, 18(7):649–655, July 1997.

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

[35] S. S. Beauchemin and J. L. Barron. The computation of optical flow. *ACM Computing Surveys*, 27(3):433–467, September 1995.

[36] S. S. Beauchemin and J. L. Barron. The frequency structure of one-dimensional occluding image signals. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 22(2):200–206, February 2000.

[37] D. Ben-Tzvi, A. Del Bimbo, and P. Nesi. Optical flow from constraint lines parametrization. *Pattern Recognition*, 26(10):1549–1561, 1993.

[38] J. R. Bergen and E. H. Adelson. Hierarchical, computationally efficient motion estimation algorithm. *J. of the Optical Society of America A*, 4:35, 1987.

[39] J. R. Bergen, P. J. Burt, R. Hingorani, and S. Peleg. A three-frame algorithm for estimating two-component image motion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(9):886–895, 1992.

[40] J. R. Bergen, P. J. Burt, R. Hingorani, and S. Peleg. A three-frame algorithm for estimating two-component image motion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(9):886–895, 1992.

[41] L. Bergen and F. Meyer. Motion segmentation and depth ordering based on morphological segmentation. In *Proc. 5th European Conference on Computer Vision*, volume II, pages 531–547. Springer-Verlag, 1998.

[42] F. Bergholm and S. Carlsson. A theory of optical flow. *Computer Vision, Graphics and Image Processing: Image Understanding*, 53(2):171–188, March 1991.

[43] S. S. Beuchemin and J. L. Barron. The computation of optical flow. *ACM Computing Surveys*, 27(3):433–467, September 1995.

[44] J. Bigün, G. H. Granlund, and J. Wiklund. Multidimensional orientation estimation with applications to texture analysis and optical flow. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(8):775–790, August 1991.

[45] M. J. Black and P. Anandan. The robust estimation of multiple motions: Parametric and piecewise-smooth flow fields. *Computer Vision and Image Understanding*, 63(1):75–104, January 1996.

[46] M. J. Black and A. D. Jepson. Estimating optical flow in segmented images using variable-order parametric models with local deformations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(10):972–986, October 1996.

[47] G. D. Borshukov, G. Bozdagi, Y. Altunbasak, and A. M. Tekalp. Motion segmentation by multistage affine classification. *IEEE Transactions on Image Processing*, 6(11):1591–1594, November 1997.

[48] P. Bouthemy. Maximum likelihood framework for determining moving edges. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(5):499–511, May 1989.

[49] P. Bouthemy and E. François. Motion segmentation and qualitative dynamic scene analysis from an image sequence. *International Journal of Computer Vision*, 10(2):157–182, April 1993.

[50] P. Bouthemy and P. Lalande. Recovery of moving object masks in an image sequence using local spatiotemporal contextual information. *Optical Engineering*, 32(6):1205–1212, June 1993.

[51] G. Bozdagi, A. M. Tekalp, and L. Onural. 3-D motion estimation and wireframe model adaptation including photometric effects for model-based coding of facial image sequences. *IEEE Transactions on Circuits and Systems for Video Technology*, 4(3):246–257, June 1994.

[52] G. Bozdagi, A. M. Tekalp, and L. Onural. An improvement to MBASIC algorithm for 3-D motion and depth estimation. *IEEE Transactions on Image Processing*, 3(5):711–717, September 1994.

[53] J. W. Brandt. Improved accuracy in gradient-based optical flow estimation. *International Journal of Computer Vision*, 25(1):5–22, 1997.

[54] R. W. Brockett. Gramians, generalized inverses, and the least-squares approximation of optical flow. *Journal of Visual Communication and Image Representation*, 1(1):3–11, 1990.

[55] T. J. Broida and R. Chellappa. Estimation of object motion parameters from noisy images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(1):90–99, 1986.

[56] T. J. Broida and R. Chellappa. Uniqueness and performance measure results for 3-D motion estimation from a monocular sequence of noisy images. *J. of the Optical Society of America*, 6:879–889, 1989.

[57] H. H. Bulthoff, J. J. Little, and T. Poggio. A parallel algorithm for real-time computation of optical flow. *Nature*, 337(6207):549–553, February 1989.

[58] T. J. Burns, S. K. Rogers, D. W. Ruck, and M. E. Oxley. Discrete, spatiotemporal, wavelet multiresolution analysis method for computing optical flow. *Optical Engineering*, 33(7):2236–2247, July 1994.

[59] B. F. Buxton and H. Buxton. Low-level edge detection for the extraction of optical flow data and passive ranging. *GEC Journal of Research*, 1(3):184–187, 1983.

[60] B. F. Buxton and H. Buxton. Computation of optic flow from the motion of edge features in image sequences. *Image and Vision Computing*, 2(2):59–76, 1984.

[61] B. F. Buxton, H. Buxton, and B. K. Stephenson. Parallel computations of optic flow in early image processing. *IEE Proceedings F*, 131(6):593–602, 1984.

[62] B. F. Buxton and D. W. Murray. Optic flow segmentation as an ill-posed and maximum likelihood problem. *Image and Vision Computing*, 3(4):163–169, 1985.

[63] C. Cafforo and F. Rocca. Methods for measuring small displacements of television images. *IEEE Transactions on Information Theory*, 22(5):573–579, 1976.

[64] M. P. Cagigal, L. Vega, and P. Prieto. Object movement characterization from low-light-level images. *Optical Engineering*, 33(8):2810–2812, 1994.

[65] M. P. Cagigal, L. Vega, and P. Prieto. Movement characterization with the spatiotemporal Fourier transform of low-light-level images. *Applied Optics*, 34(11):1769–1774, 1995.

[66] M. Campani and A. Verri. Motion analysis from first-order properties of optical flow. *Computer Vision, Graphics and Image Processing: Image Understanding*, 56(1):90–107, 1992.

[67] T. Camus. Real-time quantized optical flow. *Real-Time Imaging*, 3(2):71–86, April 1997.

[68] J-F. Cardoso. Blind signal separation: Statistical principles. *Proceedings of the IEEE*, 86(10):2009–2025, 1998.

[69] D. A. Castelow, D. W. Murray, G. L. Scott, and B. F. Buxton. Matching canny edgels to compute the principal components of optic flow. *Image and Vision Computing*, 6(2):129–136, 1988.

[70] B. Cernuschi-Frias, D. B. Cooper, Y. P. Hung, and P. N. Belhumeur. Toward a model-based Bayesian theory for estimating and recognizing parameterized 3D objects using two or more images taken from different position. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11:1028–1052, 1989.

[71] Y.-L. Chan and W.-C. Siu. On block motion estimation using a novel search strategy for an improved adaptive pixel decimation. *Journal of Visual Communication and Image Representation*, 9(2):139–154, June 1998.

[72] M. M. Chang, A. M. Tekalp, and M. I. Sezan. Simultaneous motion estimation and segmentation. *IEEE Transactions on Image Processing*, 6(9):1326–1333, 1997.

[73] K. Chaudhury and R. Mehrotra. Optical flow estimation using smoothness of intensity trajectories. *Computer Vision, Graphics and Image Processing: Image Understanding*, 60(2):230–244, 1994.

[74] K. Chaudhury and R. Mehrotra. A trajectory-based computational model for optical flow estimation. *IEEE Transactions on Robotics and Automation*, 11(5):733–741, 1995.

[75] A. K. Chhabra. Real-time computation of optical flow along contours of significant intensity change. *Real-Time Imaging*, 3(2):87–99, April 1997.

[76] A. K. Chhabra and T. A. Grogan. On poisson solvers and semi-direct methods for computing area based optical flow.

IEEE Transactions on Pattern Analysis and Machine Intelligence, 16(11):1133–1138, November 1994.

[77] Y. Chih-Ho and F. M. Caimi. Determination of horizontal motion through optical flow computations. *Journal of Computer Science and Technology*, 12(2):133–144, 1997.

[78] T. M. Chin, W. C. Karl, and A. S. Willsky. Probabilistic and sequential computation of optical flow using temporal coherence. *IEEE Transactions on Image Processing*, 3(6):773–788, 1994.

[79] J. G. Choi and S.-D. Kim. Multi-stage segmentation of optical flow field. *Signal Processing*, 54(2):109–118, October 1996.

[80] Y.-M. Chou and H.-M. Hang. A new motion estimation method using frequency components. *Journal of Visual Communication and Image Representation*, 8(1):83–96, March 1997.

[81] C. Colombo, A. Del Bimbo, and S. Santini. Optical flow through relaxation in the velocity space. *Pattern Recognition Letters*, 15(4):373–382, April 1994.

[82] C. Colombo, a. Del Bimbo, and S. Santini. Optical flow by non-linear relaxation. *Pattern Recognition*, 28(7):977–988, 1995.

[83] G. Convertino, E. Stella, A. Branca, and A. Distante. Optic flow estimation by a hopfield neural network using geometrical constraints. *Machine Vision and Applications*, 10(3):114–122, 1997.

[84] V. Cornilleau-Peres and J. Droulez. Velocity-based correspondence in stereokinetic images. *Computer Vision, Graphics and Image Processing: Image Understanding*, 58(2):137–146, September 1993.

[85] J. Cryer, P.-S. Tsai, and M. A. Shah. Combining stereo and shading. *Pattern Recognition*, 28(7):1033–1043, July 1995.

[86] A. Cumani, A. Guiducci, and P. Grattoni. Image description of dynamic scenes. *Pattern Recognition*, 24(7):661–673, 1991.

[87] T. Darrell and A. P. Pentland. Cooperative robust estimation using layers of support. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(5):474–487, May 1995.

[88] E. De Micheli, V. Torre, and S. Uras. The accuracy of the computation of optical flow and of the recovery of motion parameters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15:434–447, 1993.

[89] A. Del Bimbo, P. Nesi, and J. L. C. Sanz. Analysis of optical flow constraints. *IEEE Transactions on Image Processing*, 4(4):460–469, 1995.

[90] A. Del Bimbo, P. Nesi, and J. L. C. Sanz. Optical flow computation using extended constraints. *IEEE Transactions on Image Processing*, 5(5):720–739, 1996.

[91] T. S. Denney and J. L. Prince. A frequency domain performance analysis of Horn and Schunck's optical flow algorithm for deformable motion. *IEEE Transactions on Image Processing*, 4(9):1324–1327, 1995.

[92] T. S. Jr. Denney and J. L. Prince. Optimal brightness functions for optical flow estimation of deformable motion. *IEEE Transactions on Image Processing*, 3(2):178–191, March 1994.

[93] U. R. Dhond and J. K. Aggarwal. Stereo matching in the presence of narrow occluding objects using dynamics disparity search. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 17(7):719–724, 1995.

[94] J. Dias, H. Araújo, C. Paredes, and J. Batista. Optical normal flow estimation on log-polar images – A solution for real-time binocular vision. *Real-Time Imaging*, 3(3):213–228, June 1997.

[95] S. Dickinson, A. P. Pentland, and A. Rosenfeld. 3-D shape recovery using distributed aspect matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):174–198, 1992.

[96] N. Diehl. Object-oriented motion estimation and segmentation in image sequences. *Signal Processing: Image Communication*, 3(1):23–56, 1991.

[97] L. Dreschler and H.-H. Nagel. Volumetric model and 3D trajectory of a moving car derived from monocular TV frame sequences of a street scene. *Computer Vision, Graphics and Image Processing*, 20:199–228, 1982.

[98] J. N. Driessen, L. Boroczky, and J. Biemond. Pel-recursive motion field estimation from image sequences. *Journal of Visual Communication and Image Representation*, 2(3):259–280, 1991.

[99] M.-P. Dubuisson and A. K. Jain. Contour extraction of moving objects in complex outdoor scenes. *International Journal of Computer Vision*, 14(1):83–105, 1995.

[100] R. O. Duda and P. E. Hart. *Pattern Classification and Scene Analysis.* Wiley, New York, 1973.

[101] F. Dufaux, I. Moccagatta, F. Moscheni, and H. Nicolas. Vector quantization-based motion field segmentation under the entropy criterion. *Journal of Visual Communication and Image Representation*, 5(4):356–369, 1994.

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

[103] R. C. Emerson, J. R. Bergen, and E. H. Adelson. Directionally selective complex cells and the computation of motion energy in cat visual cortex. *Vision Research*, 32:203–218, 1992.

[104] W. Enkelmann. Investigations of multigrid algorithms for the estimation of optical flow fields in image sequences. *Computer Vision, Graphics and Image Processing*, 43(2):150–177, 1988.

[105] S. M. Fairley, I. D. Reid, and D. W. Murray. Transfer of fixation using affine structure: Extending the analysis to stereo. *International Journal of Computer Vision*, 29(1):47–58, 1998.

[106] O. D. Faugeras, P. Fua, B. Hotz, R. Ma, L. Robert, M. Thonnat, and Z. Zhang. Quantitative and qualitative comparison of some area and feature-based stereo algorithms. In W. Forstner and S Ruwiedel, editors, *International Workshop on Robust Computer Vision: Quality of Vision Algorithms*, pages 1–26, Karlsruhe, Germany, March 1992.

[107] O. D. Faugeras, E. Le Bras-Mehlman, and J. D. Boissonnat. Representing stereo data with the Delaunay triangulation. *Artificial Intelligence*, 44(1-2):41–87, July 1990.

[108] O. D. Faugeras and S. J. Maybank. Motion from point matches: Multiplicity of solution. *International Journal of Computer Vision*, 4(3):225–246, 1990.

[109] O. D. Faugeras and L. Robert. What can two images tell us about a third one? *International Journal of Computer Vision*, 18(1):5–20, 1996.

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

[111] D. J. Fleet and A. D. Jepson. Hierarchical construction of orientation and velocity selective filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(3):315–325, 1989.

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

[113] D. J. Fleet and A. D. Jepson. Stability in phase information. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(12):1253–1268, 1993.

[114] D. J. Fleet and K. Langley. Recursive filters for optical flow. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(1):61–67, January 1995.

[115] W. T. Freeman, Edward H. Adelson, and D. J. Heeger. Motion without movement. *ACM Computer Graphics*, 25(4):27–30, 1991.

[116] P. Fua. From multiple stereo views to multiple 3-D surfaces. *International Journal of Computer Vision*, 24(1):19–35, 1997.

[117] S. Ghosal and R. Mehrotra. Robust optical flow estimation using semi-invariant local features. *Pattern Recognition*, 30(2):229–237, 1997.

[118] S. Ghosal and P. Vanek. A fast scalable algorithm for discontinuous optical flow estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(2):181–194, February 1996.

[119] D. D. Giusto and G. Vernazza. Optical flow calculation from feature space analysis through an automatic segmentation process. *Signal Processing*, 16:41–51, 1989.

[120] J. Gluckman and S. K. Nayar. Planar catadioptric stereo: Geometry and calibration. *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 22–28, 1999.

[121] J. Gluckman and S. K. Nayar. Rectified catadioptric stereo sensors. *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 380–387, 2000.

[122] J. Gluckman and S. K. Nayar. Rectified catadioptric stereo sensors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(2):224–236, February 2002.

[123] R. C. Gonzalez and R. E. Woods. *Digital Image Processing.* Addison Wesley, 1992.

[124] A. Goshtasby and W. A. Gruver. Design of a single-lens stereo camera system. *Pattern Recognition*, 26:923–936, 1993.

[125] N. Grammalidis and M. G. Strintzis. Disparity and occlusion estimation in multiocular systems and their coding for the communication of multiview image sequences. *IEEE Transactions on Circuits and Systems for Video Technology*, 8(3):327–344, June 1998.

[126] W. E. L. Grimson. Computational experiments with a feature based stereo algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 7(1):17–34, 1985.

[127] E. Grosso and M. Tistarelli. Active/dynamic stereo vision. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 17(9):868–889, 1995.

[128] E. Grosso and M. Tistarelli. Log-polar stereo for anthropomorphic robots. In D. Vernon, editor, *Computer Vision – ECCV 2000*, volume 1842 of *Lecture Notes in Computer Science*, pages 299–313, Berlin, 2000. Springer-Verlag.

[129] H. Gu, Y. Shirai, and M. Asada. MDL-based segmentation and motion modeling in a long image sequence of scene with multiple independently moving objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(1):58–64, January 1996.

[130] N. Haddadi and C.-C. Kuo. Computation of dense optical flow with a parametric smoothness model. *Journal of Visual Communication and Image Representation*, 4(4):309–323, 1993.

[131] L. Hahn. *Complex Numbers and Geometry.* The Mathmatical Association of America, Washington, D.C., 1994.

[132] R. Hartley. Segmentation of optical flow fields by pyramid linking. *Pattern Recognition Letters*, 3(4):253–262, 1985.

[133] S. M. Haynes and R. C. Jain. Detection of moving edges. *Computer Vision, Graphics and Image Processing*, 21(3):345–367, 1983.

[134] G. Healey. Hierarchical segmentation-based approach to motion analysis. *Image and Vision Computing*, 11(9):570–576, 1993.

[135] D. J. Heeger. Model for the extraction of image flow. *J. of the Optical Society of America*, 4(8):1455–1471, 1987.

[136] D. J. Heeger. Optical flow using spatio-temporal filters. *International Journal of Computer Vision*, 1(4):279–302, 1988.

[137] F. Heitz and P. Bouthemy. Multimodal motion estimation of discontinuous optical flow using markov random fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(12):1217–1232, December 1993.

[138] E. C. Hildreth. The computation of the velocity field. *Proceedings of the Royal Society of London, Series B*, 221:189–220, 1984.

[139] E. C. Hildreth. Computations underlying the measurement of visual motion. *Artificial Intelligence*, 23(3):309–354, August 1984.

[140] B. K. P. Horn and B. G. Schunck. Determining optical flow. *Artificial Intelligence*, 17(1-3):185–203, August 1981.

[141] B. K. P. Horn and B. G. Schunck. 'Determining optical flow': A retrospective. *Artificial Intelligence*, 59(1-2):81–87, January 1993.

[142] J. Hornegger and D. W. R. Paulus. Detecting elliptical objects using inverse Hough-transform. In g. Vernazza, A. N. Venetsanopoulos, and C. Braccini, editors, *Image Processing: Theory and Applications*, pages 155–158. Elsevier Science Publishers B.V., 1993.

[143] M. Hötter. Object-oriented analysis synthesis coding based on moving two-dimensional objects. *Signal Processing: Image Communication*, 2:409–428, 1990.

[144] M. Hötter and R. Thoma. Image segmentation based on object oriented mapping parameter estimation. *Signal Processing*, 15(3):315–334, 1988.

[145] P. V. C. Hough. *Method and Means for Recognising Complex Patterns.* U.S. Patent 3,069,654, 1962.

[146] Y. Z. Hsu, H.-H. Nagel, and G. Rekers. New likelihood test methods for change detection in image sequences. *Computer Vision, Graphics and Image Processing*, 26:73–106, 1984.

[147] T. S. Huang and C. B. Lee. Motion and structure from orthographic projections. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11:536–540, 1989.

[148] T. S. Huang and A. N. Netravali. Motion and structure from feature correspondences: A review. *Proceedings of the IEEE*, 82(2):252–268, 1994.

[149] Y. Huang, K. Palaniappan, X. Zhuang, and J. E. Cavanaugh. Optic flow field segmentation and motion estimation using a robust genetic partitioning algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(12):1177–1190, December 1995.

[150] T. L. Huntsberger and S. N. Jayaramamurthy. Determination of the optic flow field using the spatiotemporal deformation of region properties. *Pattern Recognition Letters*, 6(3):169–177, 1987.

[151] T. L. Huntsberger and S. N. Jayaramamurthy. Determination of the optic flow field in the presence of occlusion. *Pattern Recognition Letters*, 8(5):325–333, 1988.

[152] S. H. Hwang and S. Lee. A hierarchical optical flow estimation algorithm based on the inter-level motion smoothness constraint. *Pattern Recognition*, 26(6):939–952, April 1993.

[153] M. Inaba, T. Hara, and H. Inoue. A stereo viewer based on a single camera with view control mechanism. In *Proc. International Conference on Robots and Systems*, 1993.

[154] Point Grey Research Inc. Stereo head vendor. http://www.ptgrey.com.

[155] Videre Design Inc. Stereo head vendor. http://www.videredesign.com.

[156] D. Ioannou, E. T. Dugan, and A. F. Laine. On the uniqueness of the representation of a convex polygon by its Hough transform. *Pattern Recognition Letters*, 17:1259–1264, 1996.

[157] M. Irani and P. Anandan. A unified approach to moving object detection in 2D and 3D scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(6):577, June 1998.

[158] M. Irani, B. Rousso, and S. Peleg. Computing occluding and transparent motions. *International Journal of Computer Vision*, 12(1):5–16, February 1994.

[159] T.-R. Isao and V. Chen. A neural scheme for optical flow computation based on gabor filters and generalized gradient method. *Neurocomputing*, 6(3):305–325, June 1994.

[160] L. Jacobson and H. Wechsler. Derivation of optical flow using a spatiotemporal-frequency approach. *Computer Vision, Graphics and Image Processing*, 38(1):29–65, April 1987.

[161] C. J. Jacobus, R. T. Chien, and J. M. Selander. Motion detection and analysis by matching graphs of intermediate-level primitives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2(6):495–510, 1980.

[162] R. C. Jain. Extraction of motion information from peripheral processes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 3(5):489–503, 1981.

[163] R. C. Jain. Difference and accumulative difference pictures in dynamic scene analysis. *Image and Vision Computing*, 2(2):99–108, 1984.

[164] R. C. Jain. Segmentation of frame sequences obtained by a moving observer. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6:624–629, 1984.

[165] R. C. Jain, W. N. Martin, and J. K. Aggarwal. Segmentation through the detection of changes due to motion. *Computer Graphics and Image Processing*, 11(1):13–34, 1979.

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

[167] R. S. Jasinschi. The properties of space-time sampling and the extraction of the optical flow: the effects of motion uncertainty. *Journal of Visual Communication and Image Representation*, 2(3):222–229, 1991.

[168] S. N. Jayaramamurthy and R. C. Jain. Approach to the segmentation of textured dynamic scenes. *Computer Vision, Graphics and Image Processing*, 21(2):239–261, February 1983.

[169] T. Jebara, A. Azarbayejani, and A. Pentland. 3D structure from 2D motion. *IEEE Signal Processing Magazine*, 16(3):66–84, 1999.

[170] M. Jenkin and J. K. Tsotsos. Applying temporal constraints to the dynamic stereo problem. *Computer Vision, Graphics and Image Processing*, 24:16–32, 1986.

[171] G. Johansson. Visual perception of biological motion and a model for its analysis. *Perception and Psychophysics*, 14:201–211, 1973.

[172] C. Kambhamettu and D. B. Goldgof. Curvature-based approach to point correspondence recovery in conformal nonrigid motion.

Computer Vision, Graphics and Image Processing: Image Understanding, 60(1):26–43, 1994.

[173] K. Kanatani. Detection of surface orientation and motion from texture by a stereo-logical technique. *Artificial Intelligence*, 23(2):213–237, July 1984.

[174] K. Kanatani. Detecting the motion of a planar surface by line and surface integrals. *Computer Vision, Graphics and Image Processing*, 29(1):13–22, 1985.

[175] K. Kanatani. Tracing planar surface motion frome a projection without knowing the correspondence. *Computer Vision, Graphics and Image Processing*, 29(1):1–12, 1985.

[176] K. Kanatani. Transformation of optical flow by camera rotation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(2):131–143, March 1988.

[177] M. S. Kang, R.-H. Park, and K.-H. Lee. Recovering an elevation map by stereo modeling of the aerial image sequence. *Optical Engineering*, 33(11):3793–3802, 1994.

[178] A. M. L. Kappers, S. F. te Pas, J. J. Koenderink, and A. J. van Doorn. Simulating the detection of first-order optical flow components. *Vision Research*, 36(21):3539–3547, 1996.

[179] J. K. Kearney, W. B. Thompson, and D. L. Boley. Optical flow estimation: An error analysis of gradient-based methods with local optimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(2):229–244, March 1987.

[180] B.-H. Kim and R.-H. Park. Shape from shading and photometric stereo using surface approximation by Legendre polynomials. *Computer Vision and Image Understanding*, 66(3):255–270, June 1997.

[181] D. H. Kim, W. Y. Choi, and R.-H. Park. Stereo matching technique based on the theory of possibility. *Pattern Recognition Letters*, 13(10):735–744, 1992.

[182] D. H. Kim and R.-H. Park. Analysis of quantization error in line-based stereo matching. *Pattern Recognition*, 27(7):913–924, 1994.

[183] J. D. Kim, S.-D. Kim, and J.-K. Kim. Fast convergent method for optical flow estimation in noisy image sequences. *Electronics Letters*, 25(1):74–75, 1989.

[184] Y. C. Kim and J. K. Aggarwal. Determining object motion in a sequence of stereo images. *IEEE Transactions on Robotics and Automation*, 3(6):599–614, 1987.

[185] Y.-H. Kim and S.-D. Kim. Image flow segmentation and estimation using desplaced spatial gradient. *Electronics Letters*, 28(24):2213–2215, 1992.

[186] H. Kirchner and H. Niemann. Finite element method for determination of optical flow. *Pattern Recognition Letters*, 13(2):131–141, 1992.

[187] R. Koch. Automatic reconstruction of buildings from stereoscopic image sequences. *Computer Graphics Forum*, 12(3):C339–C350, 1993.

[188] J. J. Koenderink. Optic flow. *Vision Research*, 26(1):161–168, 1986.

[189] I. Kompatsiaris, D. Tzovaras, and M. G. Strintzis. 3D model-based segmentation of videoconference image sequences. *IEEE Transactions on Circuits and Systems for Video Technology*, 8(5):547, October 1998.

[190] K. Konlige. Small vision systems: Hardwared and implementation. In *Eighth International Symposium on Robotics Research*, Hayama, Japan, 1997.

[191] J. Konrad and E. Dubois. Bayesian estimation of vector motion fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(9):910–927, 1992.

[192] A. Kumar, A. R. Tannenbaum, and G. J. Balas. Optical flow: a curve evolution approach. *IEEE Transactions on Image Processing*, 5(4):598–610, 1996.

[193] K. Kutulakos. Approximate n-view stereo. In *Computer Vision – ECCV 2000*, volume 1842 of *Lecture Notes in Computer Science*, pages 67–83, Berlin, 2000. Springer-Verlag.

[194] M.S. Langer and R. Mann. Dimensional analysis of image motion. *Proc. International Conference on Computer Vision*, pages 155–162, July 2001. Discusses optical snow dealing with relative motion of highly complex objects in a scene and argues for the superiority of frequency-domain techniques.

[195] M. Lappe and A. Grigo. How stereo vision interacts with optic flow perception: Neural mechanisms. *Neural Networks*, 12(9), 1999.

[196] M. Lappe and J. P. Rauschecker. A neural network for the processing of optic flow from ego-motion in man and higher mammals. *Neural Computation*, 5(3):374–391, 1993.

[197] D. T. Lawton. Processing translational motion sequences. *Computer Vision, Graphics and Image Processing*, 22(1):116–144, 1983.

[198] D. H. Lee, I. S. Kweon, and Cipolla R. A biprism-stereo camera system. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'99)*, 1999.

[199] D. N. Lee. The optic flow field: The foundation of vision. *Philosophical Transactions of the Royal Society of London, Series B*, B290:169–179, 1980.

[200] J. H. Lee and S.-D. Kim. An error analysis of gradient-based methods. *Signal Processing*, 35(2):157–162, 1994.

[201] K.-H. Lee and K. Wohn. Adaptive model for image motion estimation. *Electronics Letters*, 31(6):426–428, March 1995.

[202] S.-W. Lee, J. G. Choi, and S.-D. Kim. Scene segmentation using a combined criterion of motion and intensity. *Optical Engineering*, 36(8):2346–2352, 1997.

[203] M. D. Levine, D. A. O'Handley, and G. M. Yagi. Computer determination of depth maps. *Computer Graphics and Image Processing*, 2:131–150, 1973.

[204] H. Y. Li and J. Y. A. Wang. Computing optical flow with a recurrent neural network. *International Journal of Pattern Recognition and Artificial Intelligence*, 7(4):801–814, 1993.

[205] J. Li, X. Lin, and C.-C. Kuo. Boundary-control vector (BCV) motion field representation and estimation by using a markov random field model. *Journal of Visual Communication and Image Representation*, 7(3):230–243, September 1996.

[206] S.-P. P. Liou and R. C. Jain. Motion detection in spatio-temporal space. *Computer Vision, Graphics and Image Processing*, 45(2):227–250, February 1989.

[207] H. Liu, T.-H. Hong, M. Herman, and R. Chellappa. A general motion model and spatio-temporal filters for computing optical

flow. *International Journal of Computer Vision*, 22(2):141–172, 1997.

[208] M. R. Luettgen, W. C. Karl, and A. S. Willsky. Efficient multiscale regularization with applications to the computation of optical flow. *IEEE Transactions on Image Processing*, 3(1):41–64, 1994.

[209] R. C. Luo and R. E. Mullen. A modified optical flow approach for robotic tracking and acquisition. *Journal of Robotic Systems*, 6(5):489–508, 1989.

[210] S. A. Mahmoud. A new technique for velocity estimation of large moving objects. *IEEE Transactions on Signal Processing*, 39(3):741–743, 1991.

[211] S. A. Mahmoud, M. S. Afifi, and R. J. Green. Recognition and velocity computation of large moving objects in images. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 36(11):1790–1791, 1988.

[212] S. Malassiotis and M. G. Strintzis. Joint motion/disparity MAP estimation for stereo image sequences. *IEE Proceedings on Vision, Image and Signal Processing*, 143(2):101–108, 1996.

[213] S. Malassiotis and M. G. Strintzis. Model based joint motion and structure estimation from stereo images. *Computer Vision and Image Understanding*, 65(1):79–94, January 1997.

[214] S. Malassiotis and M. G. Strintzis. Model based joint motion and structure estimation from stereo images. *Computer Vision and Image Understanding*, 65(1):79–94, January 1997.

[215] S. Malassiotis and M. G. Strintzis. Motion estimation based on spatio- temporal warping for very low bit-rate coding. *IEEE Transactions on Communication*, 45(10):1172–1176, 1997.

[216] D. Marr and T. Poggio. Cooperative computation of stereo disparity. *Science*, 194(4262):283–287, 1976.

[217] D. Marr and T. Poggio. A theory of human stereo vision. *Proceedings of the Royal Society of London*, B 204:301–328, 1979.

[218] D. Marr and S. Ullman. Directional selectivity and its use in early visual processing. *Proceedings of the Royal Society of London, Series B*, 211:151–180, 1981.

[219] H. Mathieu and F. Devernay. Systeme de miroirs pour la stereoscopie. *INRIA Technical Report 172*, 1995.

[220] S. J. Maybank. Optical flow and the taylor expansion. *Pattern Recognition Letters*, 4:243–246, 1986.

[221] J. E. W. Mayhew and J. P. Frisby. Psychophysical and computational studies towards a theory of human stereopsis. *Artificial Intelligence*, 17(1-3):349–385, August 1981.

[222] J. H. McIntosh and K. M. Mutch. Matching straight lines. *Computer Vision, Graphics and Image Processing*, 43(3):386–408, September 1988.

[223] G. Medioni and R. Nevatia. Segment-based stereo matching. *Computer Vision, Graphics, and Image Processing*, 31:2–18, 1985.

[224] F. Meyer and P. Bouthemy. Region-based tracking in image sequences. In *Proceedings of the 2nd European Conference on Computer Vision – ECCV '92*, pages 476–484. Springer-Verlag, 1992.

[225] F. G. Meyer and S. Beucher. Morphological segmentation. *Journal of Visual Communication and Image Representation*, 1(1):21–46, 1990.

[226] E. D. Micheli, V. Torre, and S. Uras. The accuracy of the computation of optical flow and of the recovery of motion parameters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(5):434–447, 1993.

[227] A. Mitiche and J. K. Aggarwal. Image segmentation by conventional and information-integrating techniques: a synopsis. *Image and Vision Computing*, 3(2):50–62, 1985.

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

[229] A. Mitiche, R. Grisell, and J. K. Aggarwal. On smoothness of a vector field-application to optical flow. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(6):943–949, 1988.

[230] A. Mitiche, Y. F. Wang, and J. K. Aggarwal. Experiments in computing optical flow with the gradient-based, multiconstraint method. *Pattern Recognition*, 20(2):173–179, 1987.

[231] H. Mitsumoto, S. Tamura, K. Okazaki, N. Kajimi, and Y. Fukui. 3-d reconstruction using mirror images based on a plane symmetry recovering method. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(9):941–946, 1992.

[232] K. Mück, H.-H. Nagel, and M. Middendorf. Data-driven extraction of curved intersection lanemarks from road traffic image sequences. In D. Vernon, editor, *Computer Vision – ECCV 2000*, volume 1843 of *Lecture Notes in Computer Science*, pages 411–427, Berlin, 2000. Springer-Verlag.

[233] J. Mulligan and K. Daniilidis. Predicting disparity windows for real-time stereo. In D. Vernon, editor, *Computer Vision – ECCV 2000*, volume 1842 of *Lecture Notes in Computer Science*, pages 220–235, Berlin, 2000. Springer-Verlag.

[234] D. W. Murray. Recovering range using virtual multicamera stereo. *Computer Vision and Image Understanding*, 61(2):285–291, March 1995.

[235] D. W. Murray and B. F. Buxton. Scene segmentation from visual motion using global optimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(2):220–228, 1987.

[236] D. W. Murray and N. S. Williams. Detecting the image boundaries between optical flow fields from several moving planar facets. *Pattern Recognition Letters*, 4(2):87–92, 1986.

[237] K. M. Mutch and W. B. Thompson. Analysis of accretion and deletion at boundaries in dynamic scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 7(2):133–138, 1985.

[238] H.-H. Nagel. Displacement vectors derived from 2nd order intensity variations in image sequences. *Computer Vision, Graphics and Image Processing*, 21(1):85–117, 1983.

[239] H.-H. Nagel. On the estimation of optical flow: Relations between different approaches and some new results. *Artificial Intelligence*, 33(3):299–324, November 1987.

[240] H.-H. Nagel. On a constraint equation for the estimation of displacement rates in image sequences. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(1):13–30, 1989.

[241] H.-H. Nagel. Optical flow estimation and the interaction between measurement errors at adjacent pixel positions. *International Journal of Computer Vision*, 15(3):271–288, 1995.

[242] H.-H. Nagel and W. Enkelmann. An investigation of smoothness constraints for the estimation of displacement vector fields from images sequences. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(5):565–593, 1986.

[243] K. Nakayama and J. M. Loomis. Optical velocity patterns, velocity sensitive neurons, and space perception: A hypothesis. *Perception*, 3:63–80, 1974.

[244] K. M. Nam, J.-S. Kim, R.-H. Park, and Y. S. Shim. A fast hierarchical motion vector estimation algorithm using mean pyramid. *IEEE Transactions on Circuits and Systems for Video Technology*, 5(4):344–351, 1995.

[245] N. M. Nasrabadi, S. P. Clifford, and Y. Liu. Integration of stereo vision and optical flow by using an energy-minimization approach. *J. of the Optical Society of America A*, 6(6):900–907, 1989.

[246] S. Negahdaripour. Multiple interpretations of the shape and motion of objects from two perspectives images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(11), November 1990.

[247] S. Negahdaripour, B. Y. Hayashi, and Y. Aloimonos. Direct motion stereo for passive navigation. *IEEE Transactions on Robotics and Automation*, 11(6):829, December 1995.

[248] S. Negahdaripour and S. Lee. Motion recovery from image sequences using only first order optical flow information. *International Journal of Computer Vision*, 9(3):163–184, December 1992.

[249] S. Negahdaripour, C. H. Yu, and A. H. Shokrollahi. Recovering shape and motion from undersea images. *IEEE Journal of Oceanic Engineering*, 15(3):189, July 1990.

[250] P. Nesi. Variational approach to optical flow estimation managing discontinuities. *Image and Vision Computing*, 11(7):419–439, 1993.

[251] P. Nesi, A. Del Bimbo, and D. Ben-Tzvi. A robust algorithm for optical flow estimation. *Computer Vision and Image Understanding*, 62(1):59–68, July 1995.

[252] B. Neumann. Optical flow. *Computers and Graphics*, 18(1):17–19, 1984.

[253] A. Nomura, H. Miike, and K. Koga. Field theory approach for determining optical flow. *Pattern Recognition Letters*, 12(3):183–190, 1991.

[254] J.-M. Odobez and P. Bouthemy. Direct incremental model-based image motion segmentation for video analysis. *Signal Processing*, 66(2):143–155, April 1998.

[255] Y. Ohta and T. Kanade. Stereo by intra- and inter-scanline search. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 7(2):139–154, 1985.

[256] M. T. Orchard. Predictive motion-field segmentation for image sequence coding. *IEEE Transactions on Circuits and Systems for Video Technology*, 3(1):54–70, 1993.

[257] M. Otte and H.-H. Nagel. Optical flow estimation: advances and comparisons. In *Proceedings of the 3rd European Conference on Computer Vision – ECCV '98*, pages 51–60. Springer-Verlag, 1994.

[258] M. Otte and H.-H. Nagel. Estimation of optical flow based on higher-order spatiotemporal derivatives in interlaced and non-interlaced image sequences. *Artificial Intelligence*, 78(1):5–43, November 1995.

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

[260] M. Pardàs and P. Salembier. 3-D morphological segmentation and motion estimation for image sequences. *Signal Processing*, 38(1):31–43, July 1994.

[261] D. Park. Adaptive Bayesian decision model for motion segmentation. *Pattern Recognition Letters*, 15(12):1183–1189, December 1994.

[262] J. S. Park and J. H. Han. Estimating optical flow by tracking contours. *Pattern Recognition Letters*, 18(7):641–648, July 1997.

[263] M. S. Park, D. Y. Kim, K. S. Roh, and T. S. Choi. Motion stereo based on Fourier local phase adaptive matching. *Optical Engineering*, 39(4):866–871, 2000.

[264] P. Peer and F. Solina. Panoramic depth imaging: single standard camera approach. *International Journal of Computer Vision*, 47:149–160, 2002.

[265] S.-C. Pei and L.-G. Liou. Tracking a planar patch in 3-D space by affine transformation in binocular vision. *Pattern Recognition*, 26(1):23–31, 1993.

[266] S.-C. Pei and L.-G. Liou. Motion-based grouping of optical flow fields: the extrapolation and subtraction technique. *IEEE Transactions on Image Processing*, 6(10):1358–1363, October 1997.

[267] S. Peleg and M. Ben-Ezra. Stereo panorama with a single camera. *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 395–401, 1999.

[268] J. A. Perrone. Simple technique for optical flow estimation. *J. of the Optical Society of America A*, 7(2):264–278, 1990.

[269] S. B. Pollard, J. E. W. Mayhew, and J. P. Frisby. PMF: A stereo correspondence algorithm using a disparity gradient constraint. *Perception*, 14:449–470, 1985.

[270] J. L. Potter. Velocity as a cue to segmentation. *IEEE Transactions on Systems, Man and Cybernetics*, 5:390–394, 1975.

[271] J. L. Potter. Scene segmentation using motion information. *Computer Graphics and Image Processing*, 6:558–581, 1977.

[272] A. D. Poularikas, editor. *The Transforms and Applications Handbook.* IEEE Press, 1996.

[273] J. M. Prager and M. A. Arbib. Computing the optic flow: the MATCH algorithm and prediction. *Computer Vision, Graphics and Image Processing*, 24(3):271–304, December 1983.

[274] K. Prazdny. Detection of binocular disparities. *Biological Cybernetics*, 52:93–99, 1985.

[275] W. H Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes in C.* Cambridge University Press, 1992.

[276] N. Qian, R. C. Anderson, and E. H. Adelson. Transparent motion perception as detection of unbalanced motion signals III: Modeling. *The Journal of Neuroscience*, 14(12):7381–7392, December 1994.

[277] N. Qian and S. Mikaelian. Relationship between phase and energy methods for disparity computation. *Neural Computation*, 1999. (in press).

[278] Roy. R. and T. Kailath. Esprit – estimation of signal parameters via rotational invariance techniques. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 37:984–995, 1989.

[279] C. J. Radford. Optical flow fields in Hough transform space. *Pattern Recognition Letters*, 4:293–303, 1986.

[280] S. A. Rajala, A. N. Riddle, and W. E. Snyder. Application of one-dimensional fourier transform for tracking moving objects in noisy environments. *Computer Vision, Graphics, and Image Processing*, 21:280–293, 1983.

[281] S. Rakshit and C. H. Anderson. Computation of optical flow using basis functions. *IEEE Transactions on Image Processing*, 6(9):1246–1254, 1997.

[282] S. Reddi and G. Loizou. First-order algorithm with three clusters of optical flow vectors. *International Journal of Imaging Systems and Technology*, 7(1):33–40, 1996.

[283] A. Redert, E. Hendriks, and J. Biemond. Correspondence estimation in image pairs. *IEEE Signal Processing Magazine*, 16(3):29–46, 1999.

[284] D. Regan and K. I. Beverley. Visual responses to vorticity and the neural analysis of optic flow. *J. of the Optical Society of America A*, 2(2):280–283, 1985.

[285] A. Rognone, M. Campani, and A. Verri. Detecting moving objects from optical flow. *Pattern Recognition and Image Analysis*, 2(1), 1992.

[286] A. Rosenfeld and A. Kak. *Digital Picture Processing*, volume 1. Academic Press, New York, 1982.

[287] A. Rosenfeld and I. Weiss. A convex polygon is determined by its Hough transform. *Pattern Recognition Letters*, 16:305–306, 1995.

[288] C. Rothwell, O. D. Faugeras, and G. Csurka. A comparison of projective reconstruction methods for pairs of views. *Computer Vision and Image Understanding*, 68(1):37–58, October 1997.

[289] B. Sabata and J. K. Aggarwal. Surface correspondence and motion computation from a pair of range images. *Computer Vision and Image Understanding*, 63(2):232–250, March 1996.

[290] P. Salembier, F. Marqués, M. Pardàs, J. R. Morros, I. Corset, S. Jeannin, L. Bouchard, F. G. Meyer, and B. Marcotegui. Segmentation-based video coding system allowing the manipulation of objects. *IEEE Transactions on Circuits and Systems for Video Technology*, 7(1):60–74, 1997.

[291] J. Santos-Victor and G. Sandini. Uncalibrated obstacle detection using normal flow. *Machine Vision and Applications*, 9:130–137, 1996.

[292] J. Santos-Victor and G. Sandini. Visual behaviors for docking. *Computer Vision and Image Understanding*, 67(3):223–238, 1997.

[293] A. Scheuing and H. Niemann. Computing depth from stereo images by using optical flow. *Pattern Recognition Letters*, 4(3):205–212, 1986.

[294] C. Schnörr. Determining optical flow for irregular domains by minimizing quadratic functionals of a certain class. *International Journal of Computer Vision*, 6(1):25–38, April 1991.

[295] C. Schnörr. Computation of discontinuous optical flow by domain decomposition and shape optimization. *International Journal of Computer Vision*, 8(2):153–165, 1992.

[296] C. Schnörr. On functionals with greyvalue-controlled smoothness terms for determining optical flow. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(10):1074–1079, 1993.

[297] R. R. Schultz, L. Meng, and R. L. Stevenson. Subpixel motion estimation for super-resolution image sequence enhancement. *Journal of Visual Communication and Image Representation*, 9(1):38–50, March 1998.

[298] B. G. Schunck. The image flow constraint equation. *Computer Vision, Graphics and Image Processing*, 35(1):20–46, July 1986.

[299] B. G. Schunck. Image flow segmentation and estimation by constraint line clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(10):1010–1027, October 1989.

[300] G. L. Scott. 'four-line' method of locally estimating optic flow. *Image and Vision Computing*, 5(2):67–72, 1987.

[301] P.-K. Ser and W.-C. Siu. A new generalized Hough transform for the detection of irregular objects. *Journal of Visual Communication and Image Representation*, 6(3):256–264, 1995.

[302] M. A. Shah and R. C. Jain. Detecting time varying corners. *Computer Vision, Graphics and Image Processing*, 28:345–355, December 1984.

[303] M. A. Shah, K. Rangarajan, and P.-S. Tsai. Motion trajectories. *IEEE Transactions on Systems, Man and Cybernetics*, 23(4):1138–1150, August 1993.

[304] H. Shariat and K. E. Price. Motion estimation with more than two frames. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(5):417–434, 1990.

[305] D.-F. Shen. Segmentation-based motion estimation and residual coding for packet video. *Journal of Information Science and Engineering*, 12(3):397–413, 1996.

[306] D. Sherman and S. Peleg. Stereo by incremental matching of contours. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 12(11):1102–1106, 1990.

[307] J. Shi and J. Malik. Motion segmentation and tracking using normalized cuts. In *Proc. 6th International Conference on Computer Vision*, pages 1154–1160, 1998.

[308] Y. Q. Shi, C. Q. Shu, and J. N. Pan. Unified optical flow field approach to motion analysis from a sequence of stereo images. *Pattern Recognition*, 27(12):1577–1590, 1994.

[309] M. Shizawa and K. Mase. Multiple optical flow-fundamental constraint equations and a unified computational theory for detecting motion transparency and motion boundaries. *Systems and Computers in Japan*, 25(5):52–74, 1994.

[310] C. Q. Shu and Y. Q. Shi. On unified optical flow field. *Pattern Recognition*, 24(6):579–586, 1991.

[311] C. Silva and J. Santos-Victor. Intrinsic images for dense stereo matching with occlusions. In D. Vernon, editor, *Computer Vision - ECCV 2000*, volume 1842 of *Lecture Notes in Computer Science*, pages 100–114, Berlin, 2000. Springer-Verlag.

[312] D.-G. Sim and R.-H. Park. Antisotropic hierarchical motion estimation method based on decomposition of the functional domain. *Journal of Visual Communication and Image Representation*, 7(3):259–272, September 1996.

[313] D.-G. Sim and R.-H. Park. A two-stage algorithm for motion discontinuity-preserving optical flow estimation. *Computer Vision and Image Understanding*, 65(1):19–37, January 1997.

[314] D.-G. Sim and R.-H. Park. Robust reweighted MAP motion estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(4):353–365, April 1998.

[315] A. Singh and P. Allen. Image-flow computation: An estimation-theoretic framework and a unified perspective. *Computer Vision, Graphics and Image Processing: Image Understanding*, 56(2):152–177, September 1992.

[316] P. Smith, T. Drummond, and R. Cipolla. Motion segmentation by tracking edge information over multiple frames. In D. Vernon, editor, *Computer Vision – ECCV 2000*, volume 1843 of *Lecture Notes in Computer Science*, pages 396–410, Berlin, 2000. Springer-Verlag.

[317] S. M. Smith. Note on small angle approximations for stereo disparity. *Image and Vision Computing*, 11(6), 1993.

[318] S. M. Smith. ASSET-2: Real-time motion segmentation and object tracking. *Real-Time Imaging*, 4(1):21–40, February 1998.

[319] S. M. Smith and J. M. Brady. A scene segmenter: visual tracking of moving vehicles. *Engineering Applications of Artificial Intelligence*, 7(2):191–204, April 1994.

[320] S. M. Smith and J. M. Brady. Asset-2: Real-time motion segmentation and object tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(8):814–820, August 1995.

[321] M. A. Snyder. On the mathematical foundations of smoothness constraints for the determination of optical flow and for surface reconstruction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(11):1105–1114, November 1991.

[322] P. J. Sobey, M. G. Nagle, Y. V. Venkatesh, and M. V. Srinivasan. Measurement of complex optical flow with use of an augmented generalized gradient scheme. *J. of the Optical Society of America A*, 11(11):2787–2798, 1994.

[323] P. J. Sobey and M. V. Srinivasan. Measurement of optical flow by a generalized gradient scheme. *J. of the Optical Society of America A*, 8(9):1488–1498, 1991.

[324] P. D. Sozou. Image perturbations and optical flow in time-dependent refraction. *Journal of Modern Optics*, 39(10):2131–2148, 1992.

[325] P. D. Sozou and G. Loizou. New perspectives on optical flow. *Pattern Recognition*, 26(8):1125–1136, 1993.

[326] M. E. Spetsakis. Optical flow estimation using discontinuity conforming filters. *Computer Vision and Image Understanding*, 68(3):276–289, December 1997.

[327] A. Spinei, D. Pellerin, and J. Herault. Spatiotemporal energy-based method for velocity estimation. *Signal Processing*, 65:347–362, 1998.

[328] M. V. Srinivasan. Generalized gradient schemes for the measurement of two-dimensional image motion. *Biological Cybernetics*, 63:421–431, 1990.

[329] M. V. Srinivasan. An image-interpolation technique for the computation of optic flow and egomotion. *Biological Cybernetics*, 71(5):401–415, 1994.

[330] R. N. Strickland and Z. Mao. Contour motion estimation using relaxation matching with a smoothness constraint on the velocity field. *Computer Vision, Graphics and Image Processing: Image Understanding*, 60(2):157–167, September 1994.

[331] R. N. Strickland and D. W. Sweeney. Optical flow computation in combustion image sequences. *Applied Optics*, 27(24):5213–5220, 1988.

[332] M. G. Strintzis and I. Kokkinidis. Maximum likelihood motion estimation in ultrasound image sequences. *IEEE Signal Processing Letters*, 4(6), 1997.

[333] M. Subbarao and A. M. Waxman. Closed form solution to image flow equations for planar surfaces in motion. *Computer Vision, Graphics and Image Processing*, 36(2/3):208–228, November 1986.

[334] G. Sudhir, S. Banerjee, K. K. Biswas, and R. Bahl. Cooperative integration of stereopsis and optic flow computation. *J. of the Optical Society of America A*, 12(12):2564–2572, 1995.

[335] S. Sull and N. Ahuja. Integrated matching and segmentation of multiple features in two views. *Computer Vision and Image Understanding*, 62(3):279–297, November 1995.

[336] I. D. Svalbe. Natural representations for straight lines and the Hough tranform on discrete arrays. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(9):941–950, Sept 1989.

[337] R. Szeliski and P. Golland. Stereo matching with transparency and matching. In *Proc. 6th Int. Conf. on Computer Vision*, pages 517–524, 1998.

[338] R. Szeliski and R. Zabih. An experimental comparison of stereo algorithms. In *Proceedings of the Vision Algorithms: Theory and Practice Workshop (ICCV '99)*, Corfu, Greece, 1999.

[339] D. Terzopoulos and A. Witkin. Deformable models: Physically based models with rigid and deformable components. *IEEE Transactions on Computer Graphics & Applications*, pages 41–51, November 1988.

[340] W. M. Theimer and H. A. Mallot. Phase-based binocular vergence control and depth reconstruction using active vision. *Computer Vision, Graphics and Image Processing: Image Understanding*, 60(3):343–358, 1994.

[341] W. B. Thompson. Combining motion and contrast for segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2(6):543–549, 1980.

[342] W. B. Thompson and S. T. Barnard. Lower-level estimation and interpretation of visual motion. *Computer*, 14(8):20–28, 1981.

[343] W. B. Thompson, P. Lechleider, and E. R. Stuck. Detecting moving objects using the rigidity constraint. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(2):162–166, 1993.

[344] W. B. Thompson, K. M. Mutch, and V. A. Berzins. Dynamic occlusion analysis in optical flow field. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 7(4):374–383, 1985.

[345] W. B. Thompson and T.-C. Pong. Detecting moving objects. *International Journal of Computer Vision*, 4(1):39–57, January 1990.

[346] W. B. Thompson and T.-C. Pong. Detecting moving objects. *International Journal of Computer Vision*, 4(1):39–57, January 1990.

[347] T. Y. Tian and M. A. Shah. Motion estimation and segmentation. *Machine Vision and Applications*, 9(1):32–42, 1996.

[348] M. Tistarelli. Multiple constraints for optical flow. In J.-O Eklundh, editor, *Computer Vision – ECCV '94*, Lecture Notes in Computer Science, pages 61–70, Berlin, 1994. springer-Verlag.

[349] M. Tistarelli. Multiple constraints to compute optical flow. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(12):1243–1250, December 1996.

[350] M. Tistarelli, E. Grosso, and G. Sandini. Dynamic stereo in visual navigation. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 186–193, 1991.

[351] C. Tomasi and J. Shi. Image deformations are better than optical flow. *Mathematical and Computer Modelling*, 24(5-6):165–175, 1996.

[352] P. H. S. Torr and D. W. Murray. Statistical detection of independent movement from a moving camera. *Image and Vision Computing*, 11(4):180–187, 1993.

[353] P. H. S. Torr, R. Szeliski, and P. Anandan. An integrated Bayesian approach to layer extraction from image sequences. In *Proc. 7th International Conference on Computer Vision*, volume II, pages 983–990, Kerkyra, Greece, September 1999.

[354] P. H. S. Torr and A. Zisserman. Stochastic motion clustering. In *Proceedings of the 3rd European Conference on Computer Vision – ECCV '94*, pages 328–337. Springer-Verlag, 1994.

[355] G. Tziritas. Recursive and/or iterative estimation of the two-dimensional velocity field and reconstruction of three-dimensional motion. *Signal Processing*, 16(1):53–72, 1989.

[356] D. Tzovaras, N. Grammalidis, and M. G. Strintzis. Joint three-dimensional motion/disparity segmentation for object-based stereo image sequence coding. *Optical Engineering*, 35(1):137–143, 1996.

[357] D. Tzovaras, N. Grammalidis, and M. G. Strintzis. 3-D camera motion estimation and foreground/background separation for stereoscopic image sequences. *Optical Engineering*, 36(2), 1997.

[358] D. Tzovaras, N. Grammalidis, and M. G. Strintzis. Object-based coding of stereo image sequences using joint 3-D motion/disparity compensation. *IEEE Transactions on Circuits and Systems for Video Technology*, 7(2):312, April 1997.

[359] D. Tzovaras, N. Grammalidis, and M. G. Strintzis. Disparity field and depth map coding for multiview 3D image generation. *Signal Processing: Image Communication*, 11(3):205–230, January 1998.

[360] D. Tzovaras, I. Kompatsiaris, and M. G. Strintzis. 3D object articulation and motion estimation in model-based stereoscopic videoconference image sequence analysis and coding. *Signal Processing: Image Communication*, 14(9), August 1999.

[361] D. Tzovaras, M. G. Strintzis, and H. Sahinoglou. Evaluation of multiresolution techniques for motion and disparity estimation. *Signal Processing: Image Communication*, 6(1):59–67, March 1994.

[362] D. Tzovaras and Michael G. Strintzis. Motion and disparity field estimation using rate distortion optimization. *IEEE Transactions on Circuits and Systems for Video Technology*, 8(2):171–181, April 1998.

[363] S. Uras, F. Girosi, A. Verri, and V. Torre. A computational approach to motion perception. *Biological Cybernetics*, 60:79–97, 1989.

[364] F. Valentinotti, G. Di Caro, and B. Crespi. Real-time parallel computation of disparity and optical flow using phase difference. *Machine Vision and Applications*, 9(3):87–96, 1996.

[365] A-J. van der Veen. Algebraic methods for deterministic blind beamforming. *Proceeding of the IEEE*, 86(10):1987–2008, 1998.

[366] D. Vernon. *Machine Vision.* Prentice-Hall International, London, 1991.

[367] D. Vernon. Phase-based measurement of object velocity in image sequences using the Hough transform. *Optical Engineering*, 35(9):2620–2626, 1996.

[368] D. Vernon. Segmentation in dynamic image sequences by isolation of coherent wave profiles. In *Proceedings of the 4th European Conference on Computer Vision*, pages 293–303. springer-Verlag, 1996.

[369] D. Vernon. Estimation of optical flow using the Fourier transform. In *Proceedings of the First Irish Machine Vision and Image Processing Conference – IMVIP '97*, pages 117–123. Magee College, University of Ulster, 1997.

[370] D. Vernon. Removal of reflections from dynamic image sequences. In *Proceedings of the First Irish Machine Vision and Image Processing Conference – IMVIP '97*, pages 124–130. Magee College, University of Ulster, 1997.

[371] D. Vernon. Decoupling Fourier components of dynamic image sequences: A theory of signal separation, image segmentation, and estimation of optical flow. In *Proceedings of the 5th European Conference on Computer Vision – ECCV '98*, volume 2, pages 69–85. Springer-Verlag, 1998.

[372] D. Vernon. A new Hough transform for the detection of arbitrary 3-dimensional objects. In *Proceedings of the Optical Engineering Society of Ireland and Irish Machine Vision and Image Processing Joint Conference*, pages 243–255. National University of Ireland, Maynooth, 1998.

[373] D. Vernon. Computation of instantaneous optical flow using the phase of Fourier components. *Image and Vision Computing*, 17(3–4):189–198, 1999.

[374] D. Vernon. *Fourier Vision — Segmentation and Velocity Measurement using the Fourier Transform.* Kluwer Academic Publishers, Boston, 2001.

[375] D. Vernon and G. Sandini. *Parallel Computer Vision — The* VIS $\vec{a}$ VIS *System.* Ellis Horwood, London, 1992.

[376] D. Vernon and M. Tistarelli. Using camera motion to estimate range for robotic parts manipulation. *IEEE Transactions on Robotics and Automation*, 6(5):509–521, 1990.

[377] A. Verri, F. Girosi, and V. Torre. Differential techniques for optical flow. *J. of the Optical Society of America A*, 7(5):912–922, 1990.

[378] A. Verri and T. Poggio. Motion field and optical flow: qualitative properties. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(5):490–498, May 1989.

[379] A. Verri and V. Torre. Absolute depth estimate in stereopsis. *J. of the Optical Society of America A*, 3:297–299, 1986.

[380] Alessandro Verri, M. Straforini, and V. Torre. Computational aspects of motion perception in artificial and natural visual systems. *Philosophical Transactions of the Royal Society of London, Series B*, 337:429–443, 1992.

[381] T. Viéville and O. D. Faugeras. The first order expansion of motion equations in the uncalibrated case. *Computer Vision and Image Understanding*, 64(1):128–146, July 1996.

[382] J. Y. A. Wang and E. H. Adelson. Representing moving images with layers. *IEEE Transactions on Image Processing*, 3(5):625–638, 1994.

[383] W. Wang and J. H. Duncan. Recovering the three-dimensional motion and structure of multiple moving objects from binocular image flows. *Computer Vision and Image Understanding*, 63(3):430–446, May 1996.

[384] A. M. Waxman and J. H. Duncan. Binocular image flows: Steps toward stereo-motion fusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(6):715–729, 1986.

[385] A. M. Waxman and S. S. Sinha. Dynamic stereo: Passive ranging to moving objects from relative image flows. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8:406–412, 1986.

[386] J. Weber and J. Malik. Robust computation of optical flow in a multi-scale differential framework. *International Journal of Computer Vision*, 14(1):67–81, 1995.

[387] J. Weber and J. Malik. Rigid body segmentation and shape description from dense optical flow under weak perspective. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(2):139–143, February 1997.

[388] Y. Weiss and E. H. Adelson. A unified mixture framework for motion segmentation: Incorporating spatial coherence and estimating the number of models. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 321–326, San Francisco, CA, June 1996.

[389] J. Weng, N. Ahuja, and T. S. Huang. Matching two perspective views. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(8):806–825, 1992.

[390] L. Westberg. Hierarchical contour-based segmentation of dynamic scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(9):946–952, 1992.

[391] D. Wetzel, A. Zins, and H. Niemann. Edge- and motion-based segmentation for traffic scene analysis. *Pattern Recognition and Image Analysis*, 3(3):385–387, 1993.

[392] T. D. Williams. Depth for camera motion in a real world scene. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2(6):511–516, 1980.

[393] L. Wiskott. Segmentation from motion: Combining gabor- and mallat-wavelets to overcome the aperture and correspondence problems. *Pattern Recognition*, 32(10):1751–1766, 1999.

[394] K. Wohn, L. S. Davis, and P. Thrift. Motion estimation based on multiple local constraints and nonlinear smoothing. *Pattern Recognition*, 16(6):563–570, 1983.

[395] K. Wohn and A. M. Waxman. The analytic structure of image flows: Deformation and segmentation. *Computer Vision, Graphics and Image Processing*, 49(2):127–151, February 1990.

[396] K. Wohn and J. Wu. Estimating the finite displacement using moments. *Pattern Recognition*, 11:371–378, 1990.

[397] K. Wohn, J. Wu, and R. Brackett. A contour-based recovery of image flow: Iterative transformation method. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(8):746–760, August 1991.

[398] W. N. Worthy and J. K. Aggarwal. Volumetric descriptions of objects from multiple views. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 5(2):150–158, March 1983.

[399] Q. X. Wu. A correlation-relaxation-labeling framework for computing optical flow-template matching from a new perspective. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(9):843–853, September 1995.

[400] S. F. Wu and J. Kittler. A gradient-based method for general motion estimation and segmentation. *Journal of Visual Communication and Image Representation*, 4(1):25–38, 1993.

[401] Y. Xiong and S. A. Shafer. Moment and hypergeometric filters for high precision computation of focus, stereo and optical flow. *International Journal of Computer Vision*, 22(1):25–59, 1997.

[402] C. Xu and S.A. Valastin. A comparison between the standard Hough transform and the Mahalanobis distance Hough transform. In J.-O. Eklundh, editor, *Computer Vision – ECCV '94*, volume 800 of *Lecture Notes in Computer Science*, pages 95–100, Berlin, 1994. Springer-Verlag.

[403] M. Yachida, M. Asada, and S. Tsuji. Automatic analysis of moving images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 3(1):12–20, 1981.

[404] Y. Yakimovsky and R. Cunningham. A system for extracting three-dimensional measurements from a stereo pair of TV cameras. *Computer Graphics and Image Processing*, 7:195–210, 1978.

[405] S. Yalamanchili, W. N. Martin, and J. K. Aggarwal. Extraction of moving object descriptions via differencing. *Computer Graphics and Image Processing*, 18:188–201, 1982.

[406] A. J. Yezzi and S. Soatto. Stereoscopic segmentation. *Proc. International Conference on Computer Vision*, pages 59–66, July 2001. Variational technique for multi-frame stereo reconstruction of smooth shapes. Complementary to traditional techniques and doesn't work in regions with strong texture.

[407] R. K. K. Yip, P. K. S. Tam, and D. N. K. Leung. Modification of Hough tranform for object recognition using a 2-dimensional array. *Pattern Recognition*, 28(11):1733–1744, 1995.

[408] G. S. Young and R. Chellappa. 3-D motion estimation using a sequence of noisy stereo images: Models, motion estimation and uniqueness results. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12:735–759, 1990.

[409] C. H. Yu and S. Negahdaripour. Motion recovery in light-attenuating media from image-shading variations. *J. of the Optical Society of America A*, 9(7):1062, July 1992.

[410] H. K. Yuen, J. Illingworth, and J. Kittler. Detecting partially occluded ellipses using the Hough transform. *Image and Vision Computing*, 7(1):31–37, Feb 1989.

[411] R. Zhang and M. A. Shah. Iterative shape recovery from multiple images. *Image and Vision Computing*, 15(11):801, November 1997.

[412] Z Zhang. Estimating motion and structure from correspondences of line segments between two perspective images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(12), 1995.

[413] Z. Zhang. Motion and structure of four points from one motion of a stereo rig with unknown extrinsic parameters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(12):1222–1227, 1995.

[414] Z. Zhang. Determining the epipolar geometry and its uncertainty: A review. *International Journal of Computer Vision*, 27(2):161, 1997.

[415] Z. Zhang. Motion and structure from two perspective views: From essential parameters to euclidean motion via fundamental matrix. *J. of the Optical Society of America A*, 14(11):2938, 1997.

[416] Z. Zhang. A stereovision system for a planetary rover: Calibration, correlation, registration, and fusion. *Machine Vision and Applications*, 10(1):27–34, 1997.

[417] Z. Zhang, R. Deriche, O. D. Faugeras, and Q.-T. Luong. A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry. *Artificial Intelligence*, 78(1/2):87–119, November 1995.

[418] Z. Zhang and O. D. Faugeras. Estimation of displacements from two 3-D frames obtained from stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(12):1141–1156, 1992.

[419] Z. Zhang and O. D. Faugeras. Three-dimensional motion computation and object segmentation in a long sequence of stereo frames. *International Journal of Computer Vision*, 7(3):211–241, 1992.

[420] Z. Zhang and O. D. Faugeras. Three-dimensional motion computation and object segmentation in a long sequence of stereo frames. *International Journal of Computer Vision*, 7(3):211–241, 1992.

[421] Z. Zhang and O. D. Faugeras. Finding planes and clusters of objects from 3D line segments with application to 3D motion determination. *Computer Vision, Graphics and Image Processing: Image Understanding*, 60(3):267–284, 1994.

[422] Z. Zhang, Q.-T. Luong, and O. D. Faugeras. Motion of an uncalibrated stereo rig: Self-calibration and metric reconstruction. *IEEE Transactions on Robotics and Automation*, 12(1):103–113, 1996.

[423] Z. Y. Zhang and H. T. Tsui. 3d construction from a single view of an object and its image in a plane mirror. In *Proc. International Conference on Pattern Recognition*, pages 1174–1176, 1998.

[424] W.-Z. Zhao, F.-H. Qi, and T. Y. Young. Dynamic estimation of optical flow field using objective functions. *Image and Vision Computing*, 7(4):259–267, 1989.

[425] H. Zheng and S. D. Blostein. An error-weighted regularization algorithm for image motion-field estimation. *IEEE Transactions on Image Processing*, 2(2):246–252, 1993.

[426] H. Zheng and S. D. Blostein. Motion-based object segmentation and estimation using the MDL principle. *IEEE Transactions on Image Processing*, 4(9):1223–1235, September 1995.

[427] Y. Zheng, S. A. Billings, J. E. W. Mayhew, and J. P. Frisby. Motion parameter recovery and 3D scene segmentation. *Electronics Letters*, 32(10):884–885, 1996.

[428] X. Zhuang, T. S. Huang, N. Ahuja, and R. M. Haralick. A simplified linear optic flow-motion algorithm. *Computer Vision, Graphics and Image Processing*, 42(3):334–344, June 1988.

[429] S. W. Zucker and L. Iverson. From orientation selection to optical flow. *Computer Vision, Graphics and Image Processing*, 37(2):196–220, February 1987.