Digital Video Processing

PRENTICE HALL SIGNAL PROCESSING SERIES

Alan V. Oppenheim, Series *Editor*

ANDREWS & HUNT *Digital Image Restoration*
BRACEWELL *Two Dimensional Imaging*
BRIGHAM *The Fast Fourier Transform and Its Applications*
BURDIC *Underwater Acoustic System Analysis 2/E*
CASTLEMAN *Digital Image Processing*
CROCHIERE & RABINER *Multirate Digital Signal Processing*
DUDGEON & MERSEREAU *Multidimensional Digital Signal Processing*
HAYKIN *Advances in Spectrum Analysis and Array Processing. Vols. I, II & III*
HAYKIN, ED. *Array Signal Processing*
JOHNSON & DUDGEON *Array Signal Processing*
KAY *Fundamentals of Statistical Signal Processing*
KAY *Modern Spectral Estimation*
KINO *Acoustic Waves: Devices, Imaging, and Analog Signal Processing*
LIM *Two-Dimensional Signal and Image Processing*
LIM, ED. *Speech Enhancement*
LIM & OPPENHEIM, EDS. *Advanced Topics in Signal Processing*
MARPLE *Digital Spectral Analysis with Applications*
MCCLELLAN & RADER *Number Theory in Digital Signal Processing*
MENDEL *Lessons in Estimation Theory for Signal Processing Communications and Control 2/E*
NIKIAS *Higher Order Spectra Analysis*
OPPENHEIM & NAWAB *Symbolic and Knowledge-Based Signal Processing*
OPPENHEIM, WILLSKY, WITH YOUNG *Signals and Systems*
OPPENHEIM & SCHAFER *Digital Signal Processing*
OPPENHEIM & SCHAFER *Discrete-Time Signal Processing*
PHILLIPS & NAGLE *Digital Control Systems Analysis and Design, 3/E*
PICINBONO *Random Signals and Systems*
RABINER & GOLD *Theory and Applications of Digital Signal Processing*
RABINER & SCHAFER *Digital Processing of Speech Signals*
RABINER & JUANG *Fundamentals of Speech Recognition*
ROBINSON & TREITEL *Geophysical Signal Analysis*
STEARNS & DAVID *Signal Processing Algorithms in Fortran and C*
TEKALP *Digital Video Processing*
THERRIEN *Discrete Random Signals and Statistical Signal Processing*
TRIBOLET *Seismic Applications of Homomorphic Signal Processing*
VIADYANATHAN *Multirate Systems and Filter Banks*
WIDROW & STEARNS *Adaptive Signal Processing*

Digital Video Processing

A. Murat Tekalp
University of Rochester

Prentice Hall PTR
Upper Saddle River, NJ 07458

Tekalp, A. Murat.
Digital video processing / A. Murat Tekalp.
p. cm. -- (Prentice-Hall signal processing series)
ISBN 0-13-190075-7 (alk. paper)
1. Digital video. I. Title. II. Series.
TK6680.5.T45 1995
621.388'33--dc20 95-16650
CIP

Editorial/production supervision: **Ann Sullivan**
Cover design: **Design Source**
Manufacturing manager: **Alexis R. Heydt**
Acquisitions editor: **Karen Gettman**
Editorial assistant: **Barbara Alfieri**

Printed on Recycled Paper

Prentice-Hall, Inc.
A Simon and Schuster Company
Upper Saddle River, NJ 07458

The publisher offers discounts on this book when ordered in bulk quantities.

For more information, contact:

Corporate Sales Department
Prentice Hall PTR
One Lake Street
Upper Saddle River, NJ 07458

Phone: 800-382-3419
Fax: 201-236-7141

email: corpsales@prenhall.com

Printed in the United States of America

1 0 9 8 7 6 5 4 3 2 1

ISBN: 0-13-190075-7

Prentice-Hall International (UK) Limited, *London*
Prentice-Hall of Australia Pty. Limited, *Sydney*
Prentice-Hall Canada Inc., *Toronto*
Prentice-Hall Hispanoamericana, S.A., Mexico
Prentice-Hall of India Private Limited, *New Delhi*
Prentice-Hall of Japan, Inc., *Tokyo*
Simon & Schuster Asia Pte. Ltd., *Singapore*
Editora Prentice-Hall do Brasil, Ltda., *Rio de Janeiro*

To *Sevim* and *Kaya Tekalp, my* **mom and** dad
and *to Özge, my beloved wife*

Contents

III THREE-DIMENSIONAL MOTION ESTIMATION AND SEGMENTATION

IV VIDEO FILTERING

V STILL IMAGE COMPRESSION

APPENDICES

Preface

At present, development of products and services offering full-motion digital video is undergoing remarkable progress, and it is almost certain that digital video will have a significant economic impact on the computer, telecommunications, and imaging industries in the next decade. Recent advances in digital video hardware and the emergence of international standards for digital video compression have already led to various desktop digital video products, which is a sign that the field is starting to mature. However, much more is yet to come in the form of digital TV, multimedia communication, and entertainment platforms in the next couple of years. There is no doubt that digital video processing, which began as a specialized research area in the 70s, has played a key role in these developments. Indeed, the advances in digital video hardware and processing algorithms are intimately related, in that it is the limitations of the hardware that set the possible level of processing in real time, and it is the advances in the compression algorithms that have made full-motion digital video a reality.

The goal of this book is to provide a comprehensive coverage of the principles of digital video processing, including leading algorithms for various applications, in a tutorial style. This book is an outcome of an advanced graduate level course in Digital Video Processing, which I offered for the first time at Bilkent University, Ankara, Turkey, in Fall 1992 during my sabbatical leave. I am now offering it at the University of Rochester. Because the subject is still an active research area, the underlying mathematical framework for the leading algorithms, as well as the new research directions as the field continues to evolve, are presented together as much as possible. The advanced results are presented in such a way that the application-oriented reader can skip them without affecting the continuity of the text.

The book is organized into six parts: i) Representation of *Digital* Video, including modeling of video image formation, spatio-temporal sampling, and sampling lattice conversion without using motion information; ii) *Two-Dimensional (2-D)* Motion Estimation; iii) *Three-Dimensional (3-D) Motion Estimation and* Segmentation; iv) *Video Filtering; v) Still Image* Compression; and vi) *Video Compression,* each of which is divided into four or five chapters. Detailed treatment of the mathematical principles behind representation of digital video as a form of computer data, and processing of this data for 2-D and 3-D motion estimation, digital video standards conversion, frame-rate conversion, de-interlacing, noise filtering, resolution enhancement, and motion-based segmentation are developed. The book also covers the fundamentals of image and video compression, and the emerging world standards for various image and video communication applications, including high-definition TV, multimedia workstations, videoconferencing, videophone, and mobile image communications. A more detailed description of the organization and the contents of each chapter is presented in Section 1.3.

As a textbook, it is well-suited to be used in a one-semester advanced graduate level course, where most of the chapters can be covered in one 75-minute lecture. A complete set of visual aids in the form of transparency masters is available from the author upon request. The instructor may skip Chapters 18-21 on still-image compression, if they have already been covered in another course. However, it is recommended that other chapters are followed in a sequential order, as most of them are closely linked to each other. For example, Section 8.1 provides background on various optimization methods which are later referred to in Chapter 11. Chapter 17 provides a unified framework to address all filtering problems discussed in Chapters 13-16. Chapter 24, "Model-Based Coding," relies on the discussion of 3-D motion estimation and segmentation techniques in Chapters 9-12. The book can also be used as a technical reference by research and development engineers and scientists, or for self-study after completing a standard textbook in image processing such as Two-Dimensional Signal and *Image* Processing by J. S. Lim. The reader is expected to have some background in linear system analysis, digital signal processing, and elementary probability theory. Prior exposure to still-frame image-processing concepts should be helpful but is not required. Upon completion, the reader should be equipped with an in-depth understanding of the fundamental concepts, able to follow the growing literature describing new research results in a timely fashion, and well-prepared to tackle many open problems in the field.

My interactions with several exceptional colleagues had significant impact on the development of this book. First, my long time collaboration with Dr. Ibrahim Sezan, Eastman Kodak Company, has shaped my understanding of the field. My collaboration with Prof. Levent Onural and Dr. Gozde Bozdagi, a Ph.D. student at the time, during my sabbatical stay at Bilkent University helped me catch up with very-low-bitrate and object-based coding. The research of several excellent graduate students with whom I have worked Dr. Gordana Pavlovic, Dr. Mehmet Ozkan, Michael Chang, Andrew Patti, and Yucel Altunbasak has made major contributions to this book. I am thankful to Dr. Tanju Erdem, Eastman Kodak Company, for many helpful discussions on video compression standards, and to Prof. Joel Trussell for his careful review of the manuscript. Finally, reading of the entire manuscript by Dr. Gozde Bozdagi, a visiting Research Associate at Rochester, and her help with the preparation of the pictures in this book are gratefully acknowledged. I would also like to extend my thanks to Dr. Michael Kriss, Carl Schauffele, and Gary Bottger from Eastman Kodak Company, and to several program directors at the National Science Foundation and the New York State Science and Technology Foundation for their continuing support of our research; Prof. Kevin Parker from the University of Rochester and Prof. Abdullah Atalar from Bilkent University for giving me the opportunity to offer this course; and Chip Blouin and John Youngquist from the George Washington University Continuing Education Center for their encouragement to offer the short-course version.

A. Murat Tekalp "tekalp@ee.rochester.edu"
Rochester, NY February 1995

About the Author

A. Murat Tekalp received B.S. degrees in electrical engineering and mathematics from Boğaziçi University, Istanbul, Turkey, in 1980, with the highest honors, and the M.S. and Ph.D. degrees in electrical, computer, and systems engineering from Rensselaer Polytechnic Institute (RPI), Troy, New York, in 1982 and 1984, respectively.

From December 1984 to August 1987, he was a research scientist and then a senior research scientist at Eastman Kodak Company, Rochester, New York. He joined the Electrical Engineering Department at the University of Rochester, Rochester, New York, as an assistant professor in September 1987, where he is currently a professor. His current research interests are in the area of digital image and video processing, including image restoration, motion and structure estimation, segmentation, object-based coding, content-based image retrieval, and magnetic resonance imaging.

Dr. Tekalp is a Senior Member of the IEEE and a member of Sigma Xi. He was a scholar of the Scientific and Technical Research Council of Turkey from 1978 to 1980. He received the NSF Research Initiation Award in 1988, and IEEE Rochester Section Awards in 1989 and 1992. He has served as an Associate Editor for IEEE Transactions on Signal Processing (1990-1992), and as the Chair of the Technical Program Committee for the 1991 MDSP Workshop sponsored by the IEEE Signal Processing Society. He was the organizer and first Chairman of the Rochester Chapter of the IEEE Signal Processing Society. At present he is the Vice Chair of the IEEE Signal Processing Society Technical Committee on Multidimensional Signal Processing, and an Associate Editor for IEEE Transactions on Image Processing, and Kluwer Journal on Multidimensional Systems and Signal Processing. He is also the Chair of the Rochester Section of IEEE.

Time-varying images

Continuous *spatio-temporal* image:
$s_c(x_1, x_2, t) = s_c(\mathbf{x}, t), \quad (\mathbf{x}, t) \in \mathbf{R}^3 = \mathrm{R} \times \mathrm{R} \times \mathbf{R}$

Image sampled *on* a *lattice* - continuous *coordinates:*
$s_p(x_1, x_2, t) = s_p(\mathbf{x}, t), \quad \begin{bmatrix} \mathbf{x} \\ t \end{bmatrix} = \mathbf{V} \begin{bmatrix} \mathbf{n} \\ k \end{bmatrix} \in \Lambda^3$

Discrete spatio-temporal *image:*
$s(n_1, n_2, k) = s(\mathbf{n}, k), \quad (\mathbf{n}, k) \in \mathbf{Z}^3 = \mathbf{Z} \times \mathbf{Z} \times \mathbf{Z}$

Still images

Continuous still image:
$s_k(x_1, x_2) = s_c(\mathbf{x}, t)|_{t=k\Delta t}, \mathrm{x} \in R^2, \quad k$ fixed integer

Still image sampled *on a lattice:*
$s_k(x_1, \mathrm{x2}) = s_p(\mathbf{x}, t)|_{t=k\Delta t}, \quad k$ **fixed integer**
$\mathbf{x} = [v_{11}n_1 + v_{12}n_2 + v_{13}k, v_{21}n_1 + v_{22}n_2 + v_{23}k]^T,$
where v_{ij} denotes elements of the matrix **V,**

Discrete still image:
$s_k(n_1, n_2) = s(\mathbf{n}, k), \mathbf{n} \in Z^2, \quad k$ fixed integer

The subscript k may be dropped, and/or subscripts "c" and "p" may be added to $s(x_1, x_2)$ depending on the context.
$\mathbf{s}_k$ denotes lexicographic ordering of all pixels in $s_k(x_1, x_2)$.

Displacement field from time t to $t + \ell\Delta t$:

$\mathbf{d}(x_1, x_2, t; \ell\Delta t) = [d_1(x_1, x_2, t; \ell\Delta t), d_2(x_1, x_2, t; \ell\Delta t)]^T,$
$\ell \in \mathbf{Z},$ At R, $(x_1, x_2, t) \in \mathbf{R}^3$ or $(x_1, x_2, t) \in \Lambda^3$

$\mathbf{d}_{k,k+\ell}(x_1, x_2) = \mathbf{d}(x_1, x_2, t; \ell\Delta t)|_{t=k\Delta t}, \quad k, \ell$ fixed integers

$\mathbf{d}_1$ and $\mathbf{d}_2$ denote lexicographic ordering of the components of the motion vector field for a particular $(k, k + \ell)$ pair.

Instantaneous velocity field

$\mathbf{v}(x_1, x_2, t) = [v_1(x_1, x_2, t), v_2(x_1, x_2, t)]^T,$
$(x_1, \mathrm{x2}, t) \in \mathbf{R}^3$ or $(x_1, x_2, t) \in \Lambda^3$

$\mathbf{v}_k(x_1, x_2) = \mathbf{v}(x_1, x_2, t)|_{t=k\Delta t} \quad k$ fixed integer

$\mathbf{v}_1$ and $\mathbf{v}_2$ denote lexicographic ordering of the components of the motion vector field for a given k.

Chapter 1

BASICS OF VIDEO

Video refers to pictorial (visual) information, including still images and time-varying images. A still image is a spatial distribution of intensity that is constant with respect to time. A time-varying image is such that the spatial intensity pattern changes with time. Hence, a time-varying image is a spatio-temporal intensity pattern, denoted by $s_c(x_1, x_2, t)$, where x_1 and x_2 are the spatial variables and t is the temporal variable. In this book video refers to time-varying images unless otherwise stated. Another commonly used term for video is "image sequence," since a time-varying image is represented by a time sequence of still-frame images (pictures). The "video signal" usually refers to a one-dimensional analog or digital signal of time, where the spatio-temporal information is ordered as a function of time according to a predefined scanning convention.

Video has traditionally been recorded, stored, and transmitted in analog form. Thus, we start with a brief description of analog video signals and standards in Section 1.1. We then introduce digital representation of video and digital video standards, with an emphasis on the applications that drive digital video technology, in Section 1.2. The advent of digital video opens up a number of opportunities for interactive video communications and services, which require various amounts of digital video processing. The chapter concludes with an overview of the digital video processing problems that will be addressed in this book.

1.1 Analog Video

Today most video recording, storage, and transmission is still in analog form. For example, images that we see on TV are recorded in the form of analog electrical signals, transmitted on the air by means of analog amplitude modulation, and stored on magnetic tape using videocasette recorders as analog signals. Motion pictures are recorded on photographic film, which is a high-resolution analog medium, or on laser discs as analog signals using optical technology. We describe the nature of the

analog video signal and the specifications of popular analog video standards in the following. An understanding of the limitations of certain analog video formats is important, because video signals digitized from analog sources are usually limited by the resolution and the artifacts of the respective analog standard.

1.1.1 Analog Video Signal

The analog video signal refers to a one-dimensional (1-D) electrical signal f(t) of time that is obtained by sampling $s_c(x_1, x_2, t)$ in the vertical x_2 and temporal coordinates. This periodic sampling process is called scanning. The signal f(t), then, captures the time-varying image intensity $s_c(x_1, x_2, t)$ only along the scan lines, such as those shown in Figure 1.1. It also contains the timing information and the blanking signals needed to align the pictures correctly.

The most commonly used scanning methods are progressive scanning and interlaced scanning. A progressive scan traces a complete picture, called a frame, at every **At** sec. The computer industry uses progressive scanning with **At** $= 1/72$ sec for high-resolution monitors. On the other hand, the TV industry uses 2:1 interlace where the odd-numbered and even-numbered lines, called the odd field and the even field, respectively, are traced in turn. A 2:1 interlaced scanning raster is shown in Figure 1.1, where the solid line and the dotted line represent the odd and the even fields, respectively. The spot snaps back from point B to C, called the horizontal retrace, and from D to E, and from F to A, called the vertical retrace.

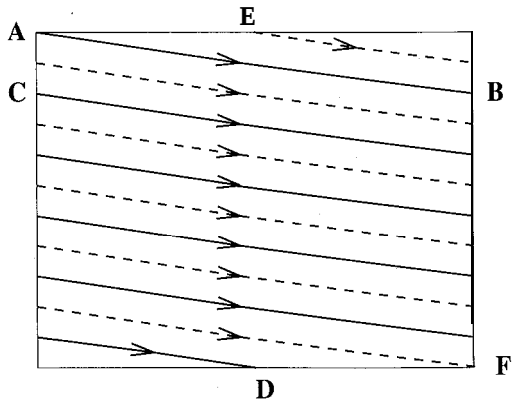


Figure 1.1: Scanning raster.

An analog video signal $f(t)$ is shown in Figure 1.2. Blanking pulses (black) are inserted during the retrace intervals to blank out retrace lines on the receiving CRT. Sync pulses are added on top of the blanking pulses to synchronize the receiver's

horizontal and vertical sweep circuits. The sync pulses ensure that the picture starts at the top left corner of the receiving CRT. The timing of the sync pulses are, of course, different for progressive and interlaced video.

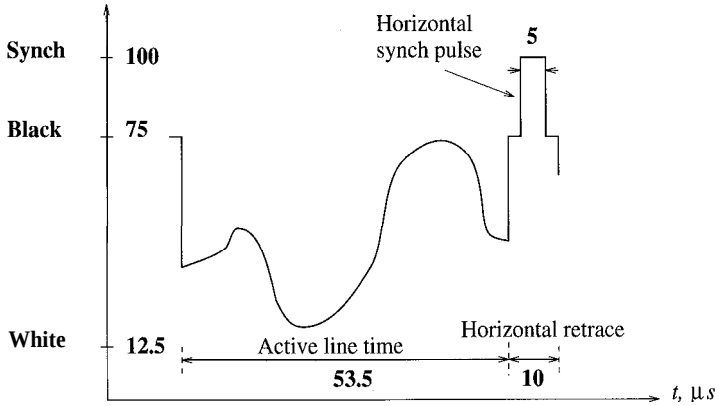


Figure 1.2: Video signal for one full line.

Some important parameters of the video signal are the vertical resolution, aspect ratio, and frame/field rate. The vertical resolution is related to the number of scan lines per frame. The aspect ratio is the ratio of the width to the height of a frame. Psychovisual studies indicate that the human eye does not perceive flicker if the refresh rate of the display is more than 50 times per second. However, for TV systems, such a high frame rate, while preserving the vertical resolution, requires a large transmission bandwidth. Thus, TV systems utilize interlaced scanning, which trades vertical resolution to reduced flickering within a fixed bandwidth.

An understanding of the spectrum of the video signal is necessary to discuss the composition of the broadcast TV signal. Let's start with the simple case of a still image, $s_c(x_1, x_2)$, where $(x_1, x_2) \in R^2$. We construct a doubly periodic array of images, $\tilde{s}_c(x_1, x_2)$, which is shown in Figure 1.3. The array $\tilde{s}_c(x_1, x_2)$ can be expressed in terms of a 2-D Fourier series,

$$\tilde{s}_c(x_1, x_2) = \sum_{k_1=-\infty}^{\infty} \sum_{k_2=-\infty}^{\infty} S_{k_1 k_2} \exp\left\{ j2\pi \left(\frac{k_1 x_1}{L} + \frac{k_2 x_2}{H} \right) \right\} \qquad (1.1)$$

where $S_{k_1 k_2}$ are the 2-D Fourier series coefficients, and L and H denote the horizontal and vertical extents of a frame (including the blanking intervals), respectively.

The analog video signal $f(t)$ is then composed of intensities along the solid line across the doubly periodic field (whichcorresponds to the scan line) in Figure 1.3.

Assuming that the scanning spot moves with the velocities v_1 and v_2 in the horizontal and vertical directions, respectively, the video signal can be expressed as

$$f(t) = \sum_{k_1=-\infty}^{\infty} \sum_{k_2=-\infty}^{\infty} S_{k_1 k_2} \exp\left\{ j2\pi \left(\frac{k_1 v_1 t}{L} + \frac{k_2 v_2 t}{H} \right) \right\} \tag{1.2}$$

where L/v_1 is the time required to scan one image line and H/v_2 is the time required to scan a complete frame. The still-video signal is periodic with the fundamentals $F_h = v_1/L$, called the horizontal sweep frequency, and $F_v = v_2/H$. The spectrum of a still-video signal is depicted in Figure 1.4. The horizontal harmonics are spaced at F_h Hz intervals, and around each harmonic is a collection of vertical harmonics F_v Hz apart.

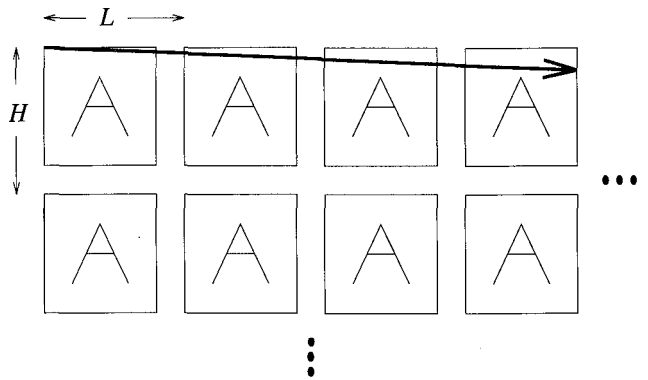


Figure 1.3: Model for scanning process.

In practice, for a video signal with temporal changes in the intensity pattern, every frame in the field shown in Figure 1.3 has a distinct intensity pattern, and the field is not doubly periodic. As a result, we do not have a line spectrum. Instead, the spectrum shown in Figure 1.4 will be smeared. However, empty spaces still exist between the horizontal harmonics at multiples of F_h Hz. For further details, the reader is referred to [Pro 94, Mil 92].

1.1.2 Analog Video Standards

In the previous section, we considered a monochromatic video signal. However, most video signals of interest are in color, which can be approximated by a superposition of three primary intensity distributions. The tri-stimulus theory of color states that almost any color can be reproduced by appropriately mixing the three additive primaries, red (R), green (G) and blue (B). Since display devices can only generate

nonnegative primaries, and an adequate amount of luminance is required, there is, in practice, a constraint on the gamut of colors that can be reproduced. An in-depth discussion of color science is beyond the scope of this book. Interested readers are referred to [Net 89, Tru 93].

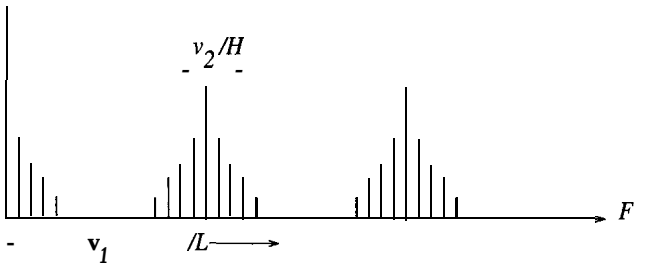


Figure 1.4: Spectrum of the scanned video signal for still images.

There exist several analog video signal standards, which have different image parameters (e.g., spatial and temporal resolution) and differ in the way they handle color. These can be grouped as:

- Component analog video
- Composite video
- S-video (Y/C video)

In component analog video (CAV), each primary is considered as a separate monochromatic video signal. The primaries can be either simply the R, G, and B signals or a luminance-chrominance transformation of them. The luminance component (Y) corresponds to the gray level representation of the video, given by

$$\mathrm{Y} = 0.30R + 0.59G + 0.11B \tag{1.3}$$

The chrominance components contain the color information. Different standards may use different chrominance representations, such as

$$\begin{aligned} I &= 0.60R + 0.28G - 0.328 \\ \mathrm{Q} &= 0.21R - 0.52G + 0.31B \end{aligned} \tag{1.4}$$

or

$$\begin{aligned} C_r &= R - Y \\ C_b &= B - Y \end{aligned} \tag{1.5}$$

In practice, these components are subject to normalization and gamma correction. The CAV representation yields the best color reproduction. However, transmission

of CAV requires perfect synchronization of the three components and three times more bandwidth.

Composite video signal formats encode the chrominance components on top of the luminance signal for distribution as a single signal which has the same bandwidth as the luminance signal. There are different composite video formats, such as NTSC (National Television Systems Committee), PAL (Phase Alternation Line), and SECAM (Systeme Electronique Color Avec Memoire), being used in different countries around the world.

NTSC

The NTSC composite video standard, defined in 1952, is currently in use mainly in North America and Japan. NTSC signal is a 2:1 interlaced video signal with 262.5 lines per field (525 lines per frame), 60 fields per second, and 4:3 aspect ratio. As a result, the horizontal sweep frequency, F_h, is 525 x 30 = 15.75 kHz, which means it takes $1/15,750$ sec = 63.5 μs to sweep each horizontal line. Then, from (1.2), the NTSC video signal can be approximately represented as

$$f(t) \approx \sum_{k_1=-\infty}^{\infty} \sum_{k_2=-\infty}^{\infty} S_{k_1 k_2} \exp\{j2\pi(15,750k_1 + 30k_2)t\} \qquad (1.6)$$

Horizontal retrace takes 10 μs, that leaves 53.5 μs for the active video signal per line. The horizontal sync pulse is placed on top of the horizontal blanking pulse, and its duration is 5 μs. These parameters were shown in Figure 1.2. Only 485 lines out of the 525 are active lines, since 20 lines per field are blanked for vertical backtrace [Mil 92]. Although there are 485 active lines per frame, the vertical resolution, defined as the number of resolvable horizontal lines, is known to be

$$485 \times 0.7 = 339.5\ (340)\ \text{lines/frame}, \qquad (1.7)$$

where 0.7 is known as the Kell factor, defined as

$$\text{Kell factor} = \frac{\text{number of perceived vertical lines}}{\text{number of total active scan lines}} \approx 0.7$$

Using the aspect ratio, the horizontal resolution, defined as the number of resolvable vertical lines, should be

$$339 \times \frac{4}{3} = 452\ \text{elements/line}. \qquad (1.8)$$

Then, the bandwidth of the luminance signal can be calculated as

$$\frac{452}{2 \times 53.5 \times 10^{-6}} = 4.2\ \text{MHz}. \qquad (1.9)$$

The luminance signal is vestigial sideband modulated (VSB) with a sideband, that extends to 1.25 MHz below the picture carrier, as depicted in Figure 1.5.

The chrominance signals, I and Q, should also have the same bandwidth. However, subjective tests indicate that the I and Q channels can be low-pass filtered to 1.6 and 0.6 MHz, respectively, without affecting the quality of the picture due to the inability of the human eye to perceive changes in chrominance over small areas (high frequencies). The I channel is separated into two bands, O-O.6 MHz and 0.6-1.6 MHz. The entire Q channel and the O-O.6 MHz portion of the I channel are quadrature amplitude modulated (QAM) with a color subcarrier frequency 3.58 MHz above the picture carrier, and the 0.6-1.6 MHz portion of the I channel is lower side band (SSB-L) modulated with the same color subcarrier. This color subcarrier frequency falls in midway between $227F_h$ and $228F_h$; thus, the chrominance spectra shift into the gaps midway between the harmonics of F_h. The audio signal is frequency modulated (FM) with an audio subcarrier frequency that is 4.5 MHz above the picture carrier. The spectral composition of the NTSC video signal, which has a total bandwidth of 6 MHz, is depicted in Figure 1.5. The reader is referred to a communications textbook, e.g., Lathi [Lat 89] or Proakis and Salehi [Pro 94], for a discussion of various modulation techniques including VSB, QAM, SSB-L, and FM.

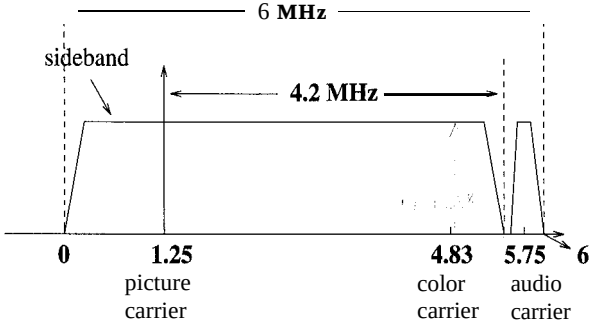


Figure 1.5: Spectrum of the NTSC video signal.

PAL and SECAM

PAL and SECAM, developed in the 1960s, are mostly used in Europe today. They are also 2:1 interlaced, but in comparison to NTSC, they have different vertical and temporal resolution, slightly higher bandwidth (8 MHz), and treat color information differently. Both PAL and SECAM have 625 lines per frame and 50 fields per second; thus, they have higher vertical resolution in exchange for lesser temporal resolution as compared with NTSC. One of the differences between PAL and SECAM is how they represent color information. They both utilize Cr and Cb components for the chrominance information. However, the integration of the color components with the luminance signal in PAL and SECAM are different. Both PAL and SECAM are said to have better color reproduction than NTSC.

In PAL, the two chrominance signals are QAM modulated with a color subcarrier at 4.43 MHz above the picture carrier. Then the composite signal is filtered to limit its spectrum to the allocated bandwidth. In order to avoid loss of high-frequency color information due to this bandlimiting, PAL alternates between +Cr and -Cr in successive scan lines; hence, the name phase alternation line. The high-frequency luminance information can then be recovered, under the assumption that the chrominance components do not change significantly from line to line, by averaging successive demodulated scan lines with the appropriate signs [Net 89]. In SECAM, based on the same assumption, the chrominance signals Cr and Cb are transmitted alternatively on successive scan lines. They are FM modulated on the color subcarriers 4.25 MHz and 4.41 MHz for Cb and Cr, respectively. Since only one chrominance signal is transmitted per line, there is no interference between the chrominance components.

The composite signal formats usually result in errors in color rendition, known as hue and saturation errors, because of inaccuracies in the separation of the color signals. Thus, S-video is a compromise between the composite video and the component analog video, where we represent the video with two component signals, a luminance and a composite chrominance signal, The chrominance signal can be based upon the (I,&) or (Cr,Cb) representation for NTSC, PAL, or SECAM systems. S-video is currently being used in consumer-quality videocasette recorders and camcorders to obtain image quality better than that of the composite video.

1.1.3 Analog Video Equipment

Analog video equipment can be classified as broadcast-quality, professional-quality, and consumer-quality. Broadcast-quality equipment has the best performance, but is the most expensive. For consumer-quality equipment, cost and ease of use are the highest priorities.

Video images may be acquired by electronic live pickup cameras and recorded on videotape, or by motion picture cameras and recorded on motion picture film (24 frames/sec), or formed by sequential ordering of a set of still-frame images such as in computer animation. In electronic pickup cameras, the image is optically focused on a two-dimensional surface of photosensitive material that is able to collect light from all points of the image all the time. There are two major types of electronic cameras, which differ in the way they scan out the integrated and stored charge image. In vacuum-tube cameras (e.g., vidicon), an electron beam scans out the image. In solid-state imagers (e.g., CCD cameras), the image is scanned out by a solid-state array. Color cameras can be three-sensor type or single-sensor type. Three-sensor cameras suffer from synchronicity problems and high cost, while single-sensor cameras often have to compromise spatial resolution. Solid-state sensors are particularly suited for single-sensor cameras since the resolution capabilities of CCD cameras are continuously improving. Cameras specifically designed for television pickup from motion picture film are called telecine cameras. These cameras usually employ frame rate conversion from 24 frames/sec to 60 fields/sec.

Analog video recording is mostly based on magnetic technology, except for the laser disc which uses optical technology. In magnetic recording, the video signal is modulated on top of an FM carrier before recording in order to deal with the nonlinearity of magnetic media. There exist a variety of devices and standards for recording the analog video signal on magnetic tapes. The Betacam is a component analog video recording standard that uses 1/2" tape. It is employed in broadcast- and professional-quality applications. VHS is probably the most commonly used consumer-quality composite video recording standard around the world. U-matic is another composite video recording standard that uses 3/4" tape, and is claimed to result in a better image quality than VHS. U-matic recorders are mostly used in professional-quality applications. Other consumer-quality composite video recording standards are the Beta and 8 mm formats. S-VHS recorders, which are based on S-video, recently became widely available, and are relatively inexpensive for reasonably good performance.

1.2 Digital Video

We have been experiencing a digital revolution in the last couple of decades. Digital data and voice communications have long been around. Recently, hi-fi digital audio with CD-quality sound has become readily available in almost any personal computer and workstation. Now, technology is ready for landing full-motion digital video on the desktop [Spe 92]. Apart from the more robust form of the digital signal, the main advantage of digital representation and transmission is that they make it easier to provide a diverse range of services over the same network [Sut 92]. Digital video on the desktop brings computers and communications together in a truly revolutionary manner. A single workstation may serve as a personal computer, a high-definition TV, a videophone, and a fax machine. With the addition of a relatively inexpensive board, we can capture live video, apply digital processing, and/or print still frames at a local printer [Byt 92]. This section introduces digital video as a form of computer data.

1.2.1 Digital Video Signal

Almost all digital video systems use component representation of the color signal. Most color video cameras provide RGB outputs which are individually digitized. Component representation avoids the artifacts that result from composite encoding, provided that the input RGB signal has not been composite-encoded before. In digital video, there is no need for blanking or sync pulses, since a computer knows exactly where a new line starts as long as it knows the number of pixels per line [Lut 88]. Thus, all blanking and sync pulses are removed in the A/D conversion.

Even if the input video is a composite analog signal, e.g., from a videotape, it is usually first converted to component analog video, and the component signals are then individually digitized. It is also possible to digitize the composite sig-

nal directly using one A/D converter with a clock high enough to leave the **color** subcarrier components free from aliasing, and then perform digital decoding to obtain the desired RGB or YIQ component signals. This requires sampling at a rate three or four times the color subcarrier frequency, which can be accomplished by special-purpose chip sets. Such chips do exist in some advanced TV sets for digital processing of the received signal for enhanced image quality.

The horizontal and vertical resolution of digital video is related to the number of pixels per line and the number of lines per frame. The artifacts in digital video due to lack of resolution are quite different than those in analog video. In analog video the lack of spatial resolution results in blurring of the image in the respective direction. In digital video, we have pixellation (aliasing) artifacts due to lack of sufficient spatial resolution. It manifests itself as jagged edges resulting from individual pixels becoming visible. The visibility of the pixellation artifacts depends on the size of the display and the viewing distance [Lut 88].

The arrangement of pixels and lines in a contiguous region of the memory is called a bitmap. There are five key parameters of a bitmap: the starting address in memory, the number of pixels per line, the pitch value, the number of lines, and number of bits per pixel. The pitch value specifies the distance in memory from the start of one line to the next. The most common use of pitch different from the number of pixels per line is to set pitch to the next highest power of 2, which may help certain applications run faster. Also, when dealing with interlaced inputs, setting the pitch to double the number of pixels per line facilitates writing lines from each field alternately in memory. This will form a "composite frame" in a contiguous region of the memory after two vertical scans. Each component signal is usually represented with 8 bits per pixel to avoid "contouring artifacts." Contouring results in slowly varying regions of image intensity due to insufficient bit resolution. Color mapping techniques exist to map 2^{24} distinct colors to 256 colors for display on 8-bit color monitors without noticeable loss of color resolution. Note that display devices are driven by analog inputs; therefore, D/A converters are used to generate component analog video signals from the bitmap for display purposes.

The major bottleneck preventing the widespread use of digital video today has been the huge storage and transmission bandwidth requirements. For example, digital video requires much higher data rates and transmission bandwidths as compared to digital audio. CD-quality digital audio is represented with 16 bits/sample, and the required sampling rate is 44kHz. Thus, the resulting data rate is approximately 700 kbits/sec (kbps). In comparison, a high-definition TV signal (e.g., the AD-HDTV proposal) requires 1440 pixels/line and 1050 lines for each luminance frame, and 720 pixels/line and 525 lines for each chrominance frame. Since we have 30 frames/s and 8 bits/pixel per channel, the resulting data rate is approximately 545 Mbps, which testifies that a picture is indeed worth 1000 words! Thus, the viability of digital video hinges upon image compression technology [Ang 91]. Some digital video format and compression standards will be introduced in the next subsection.

1.2.2 Digital Video Standards

Exchange of digital video between different applications and products requires digital video format standards. Video data needs to be exchanged in compressed form, which leads to compression standards. In the computer industry, standard display resolutions; in the TV industry, digital studio standards; and in the communications industry, standard network protocols have already been established. Because the advent of digital video is bringing these three industries ever closer, recently standardization across the industries has also started. This section briefly introduces some of these standards and standardization efforts.

Table 1.1: Digital video studio standards

Parameter	CCIR601 525/60 NTSC	CCIR601 625/50 PAL/SECAM	CIF
Number of active pels/line			
Lum (Y)	720	720	360
Chroma (U,V)	360	360	180
Number of active lines/pic			
Lum (Y)	480	576	288
Chroma (U,V)	480	576	144
Interlacing	2:1	2:1	1:1
Temporal rate	60	50	30
Aspect ratio	4:3	4:3	4:3

Digital video is not new in the broadcast TV studios, where editing and special effects are performed on digital video because it is easier to manipulate digital images. Working with digital video also avoids artifacts that would be otherwise caused by repeated analog recording of video on tapes during various production stages. Another application for digitization of analog video is conversion between different analog standards, such as from PAL to NTSC. CCIR (International Consultative Committee for Radio) Recommendation 601 defines a digital video format for TV studios for 525-line and 625-line TV systems. This standard is intended to permit international exchange of production-quality programs. It is based on component video with one luminance (Y) and two color difference (Cr and Cb) signals. The sampling frequency is selected to be an integer multiple of the horizontal sweep frequencies in both the 525- and 625-line systems. Thus, for the luminance component,

$$f_{s,lum} = 858 f_{h,525} = 864 f_{h,625} = 13.5\text{MHz}, \tag{1.10}$$

and for the chrominance,

$$f_{s,chr} = f_{s,lum}/2 = 6.75\text{MHz}. \quad (1.11)$$

The parameters of the CCIR 601 standards are tabulated in Table 1.1. Note that the raw data rate for the CCIR 601 formats is 165 Mbps. Because this rate is too high for most applications, the CCITT (International Consultative Committee for Telephone and Telegraph) Specialist Group (SGXV) has proposed a new digital video format, called the Common Intermediate Format (CIF). The parameters of the CIF format are also shown in Table 1.1. Note that the CIF format is progressive (noninterlaced), and requires approximately 37 Mbps. In some cases, the number of pixels in a line is reduced to 352 and 176 for the luminance and chrominance channels, respectively, to provide an integer number of 16 x 16 blocks.

In the computer industry, standards for video display resolutions are set by the Video Electronics Standards Association (VESA). The older personal computer (PC) standards are the VGA with 640 pixels/line x 480 lines, and TARGA with 512 pixels/line x 480 lines. Many high-resolution workstations conform with the S-VGA standard, which supports two main modes, 1280 pixels/line x 1024 lines or 1024 pixels/line x 768 lines. The refresh rate for these modes is 72 frames/sec. Recognizing that the present resolution of TV images is well behind today's technology, several proposals have been submitted to the Federal Communications Commission (FCC) for a high-definition TV standard. Although no such standard has been formally approved yet, all proposals involve doubling the resolution of the CCIR 601 standards in both directions.

Table 1.2: Some network protocols and their bitrate regimes

Network	Bitrate
Conventional Telephone	0.3-56 kbps
Fundamental BW Unit of Telephone (DS-0)	56 kbps
ISDN (Integrated Services Digital Network)	64-144 kbps (px64)
Personal Computer LAN (Local Area Network)	30 kbps
T-1	1.5 Mbps
Ethernet (Packet-Based LAN)	10 Mbps
Broadband ISDN	100-200 Mbps

Various digital video applications, e.g., all-digital HDTV, multimedia services, videoconferencing, and videophone, have different spatio-temporal resolution requirements, which translate into different bitrate requirements. These applications will most probably reach potential users over a communications network [Sut 92]. Some of the available network options and their bitrate regimes are listed in Table 1.2. The main feature of the ISDN is to support a wide range of applications

over the same network. Two interfaces are defined: basic access at 144 kbps, and primary rate access at 1.544 Mbps and 2.048 Mbps. As audio-visual telecommunication services expand in the future, the broadband integrated services digital network (B-ISDN), which will provide higher bitrates, is envisioned to be the universal information "highway" [Spe 91]. Asynchronous transfer mode (ATM) is the target transfer mode for the B-ISDN [Onv 94].

Investigation of the available bitrates on these networks and the bitrate requirements of the applications indicates that the feasibility of digital video depends on how well we can compress video images. Fortunately, it has been observed that the quality of reconstructed CCIR 601 images after compression by a factor of 100 is comparable to analog videotape (VHS) quality. Since video compression is an important enabling technology for development of various digital video products, three video compression standards have been developed for various target bitrates, and efforts for a new standard for very-low-bitrate applications are underway. Standardization of video compression methods ensures compatibility of digital video equipment by different vendors, and facilitates market growth. Recall that the boom in the fax market came after binary image compression standards. Major world standards for image and video compression are listed in Table 1.3.

Table 1.3: World standards for image compression.

Standard	Application
CCITT G3/G4	Binary images (nonadaptive)
J B I G	Binary images
JPEG	Still-frame gray-scale and color images
H.261	p x 64 kbps
MPEG-1	1.5 Mbps
MPEG-2	10-20 Mbps
MPEG-4	4.8-32 kbps (underway)

CCITT Group 3 and 4 codes are developed for fax image transmission, and are presently being used in all fax machines. JBIG has been developed to fix some of the problems with the CCITT Group 3 and 4 codes, mainly in the transmission of halftone images. JPEG is a still-image (monochrome and color) compression standard, but it also finds use in frame-by-frame video compression, mostly because of its wide availability in VLSI hardware. CCITT Recommendation H.261 is concerned with the compression of video for videoconferencing applications over ISDN lines. The target bitrates are p x 64 kbps, which are the ISDN rates. Typically, videoconferencing using the CIF format requires 384 kbps, which corresponds to p = 6. MPEG-1 targets 1.5 Mbps for storage of CIF format digital video on CD-ROM and hard disk. MPEG-2 is developed for the compression of higher-definition video at 10-20 Mbps with HDTV as one of the intended applications. We will discuss digital video compression standards in detail in Chapter 23.

Interoperability of various digital video products requires not only standardization of the compression method but also the representation (format) of the data. There is an abundance of digital video formats/standards, besides the CCITT 601 and CIF standards. Some proprietary format standards are shown in Table 1.4. A committee under the Society of Motion Picture and Television Engineers (SMPTE) is working to develop a universal header/descriptor that would make any digital video stream recognizable by any device. Of course, each device should have the right hardware/software combination to decode/process this video stream once it is identified. There also exist digital recording standards such as D1 for recording component video and D2 for composite video.

Table 1.4: Examples of proprietary video format standards

Video Format	Company
DVI (Digital Video Interactive), Indeo	Intel Corporation
QuickTime	Apple Computer
CD-I (Compact Disc Interactive)	Philips Consumer Electronics
Photo CD	Eastman Kodak Company
CDTV	Commodore Electronics

Rapid advances have taken place in digital video hardware over the last couple of years. Presently, several vendors provide full-motion video boards for personal computers and workstations using frame-by-frame JPEG compression. The main limitations of the state-of-the-art hardware originate from the speed of data transfer to and from storage media, and available CPU cycles for sophisticated real-time processing. Today most storage devices are able to transfer approximately 1.5 Mbps, although 4 Mbps devices are being introduced most recently. These numbers are much too slow to access uncompressed digital video. In terms of CPU capability, most advanced single processors are in the range of 70 MIPS today. A review of the state-of-the-art digital video equipment is not attempted here, since newer equipment is being introduced at a pace faster than this book can be completed.

1.2.3 Why Digital Video?

In the world of analog video, we deal with TV sets, videocassette recorders (VCR) and camcorders. For video distribution we rely on TV broadcasts and cable TV companies, which transmit predetermined programming at a fixed rate. Analog video, due to its nature, provides a very limited amount of interactivity, e.g., only channel selection in the TV, and fast-forward search and slow-motion replay in the VCR. Besides, we have to live with the NTSC signal format. All video captured on a laser disc or tape has to be NTSC with its well-known artifacts and very low still-frame image quality. In order to display NTSC signals on computer monitors or European TV sets, we need expensive transcoders. In order to display a smaller

version of the NTSC picture in a corner of the monitor, we first need to reconstruct the whole picture and then digitally reduce its size. Searching a video database for particular footage may require tedious visual scanning of a whole bunch of videotapes. Manipulation of analog video is not an easy task. It usually requires digitization of the analog signal using expensive frame grabbers and expertise for custom processing of the data.

New developments in digital imaging technology and hardware are bringing together the TV, computer, and communications industries at an ever-increasing rate. The days when the local telephone company and the local cable TV company, as well as TV manufactures and computer manufacturers, will become fierce competitors are near [Sut 92]. The emergence of better image compression algorithms, optical fiber networks, faster computers, dedicated video boards, and digital recording promise a variety of digital video and image communication products. Driving the research and development in the field are consumer and commercial applications such as:

- All-digital HDTV [Lip 90, Spe 95]
  @ 20 Mbps over 6 MHz taboo channels
- Multimedia, desktop video [Spe 93]
  @ 1.5 Mbps CD-ROM or hard disk storage
- Videoconferencing
  @ 384 kbps using p x 64 kbps ISDN channels
- Videophone and mobile image communications [Hsi 93]
  @ 10 kbps using the copper network (POTS)

Other applications include surveillance imaging for military or law enforcement, intelligent vehicle highway systems, harbor traffic control, cine medical imaging, aviation and flight control simulation, and motion picture production. We will overview some of these applications in Chapter 25.

Digital representation of video offers many benefits, including:

i) Open architecture video systems, meaning the existence of video at multiple spatial, temporal, and SNR resolutions within a single scalable bitstream.

ii) Interactivity, allowing interruption to take alternative paths through a video database, and retrieval of video.

iii) Variable-rate transmission on demand.

iv) Easy software conversion from one standard to another.

v) Integration of various video applications, such as TV, videophone, and so on, on a common multimedia platform.

vi) Editing capabilities, such as cutting and pasting, zooming, removal of noise and blur.

vii) Robustness to channel noise and ease of encryption.

All of these capabilities require digital processing at various levels of complexity, which is the topic of this book.

1.3 Digital Video Processing

Digital video processing refers to manipulation of the digital video bitstream. All known applications of digital video today require digital processing for data compression, In addition, some applications may benefit from additional processing for motion analysis, standards conversion, enhancement, and restoration in order to obtain better-quality images or extract some specific information.

Digital processing of still images has found use in military, commercial, and consumer applications since the early 1960s. Space missions, surveillance imaging, night vision, computed tomography, magnetic resonance imaging, and fax machines are just some examples. What makes digital video processing different from still image processing is that video imagery contains a significant amount of temporal correlation (redundancy) between the frames. One may attempt to process video imagery as a sequence of still images, where each frame is processed independently. However, utilization of existing temporal redundancy by means of multiframe processing techniques enables us to develop more effective algorithms, such as motion-compensated filtering and motion-compensated prediction. In addition, some tasks, such as motion estimation or the analysis of a time-varying scene, obviously cannot be performed on the basis of a single image. It is the goal of this book to provide the reader with the mathematical basis of multiframe and motion-compensated video processing. Leading algorithms for important applications are also included.

Part 1 is devoted to the representation of full-motion digital video as a form of computer data. In Chapter 2, we model the formation of time-varying images as perspective or orthographic projection of 3-D scenes with moving objects. We are mostly concerned with 3-D rigid motion; however, models can be readily extended to include 3-D deformable motion. Photometric effects of motion are also discussed. Chapter 3 addresses spatio-temporal sampling on 3-D lattices, which covers several practical sampling structures including progressive, interlaced, and quincunx sampling. Conversion between sampling structures without making use of motion information is the subject of Chapter 4.

Part 2 covers nonparametric 2-D motion estimation methods. Since motion compensation is one of the most effective ways to utilize temporal redundancy, 2-D motion estimation is at the heart of digital video processing. 2-D motion estimation, which refers to optical flow estimation or the correspondence problem, aims to estimate motion projected onto the image plane in terms of instantaneous pixel velocities or frame-to-frame pixel correspondences. We can classify nonparametric 2-D motion estimation techniques as methods based on the optical flow equation, block-based methods, pel-recursive methods, and Bayesian methods, which are presented in Chapters 5-8, respectively.

Part 3 deals with 3-D motion/structure estimation, segmentation, and tracking. 3-D motion estimation methods are based on parametric modeling of the 2-D optical flow field in terms of rigid motion and structure parameters. These parametric models can be used for either 3-D image analysis, such as in object-based image compression and passive navigation, or improved 2-D motion estimation. Methods

that use discrete point correspondences are treated in Chapter 9, whereas optical-flow-based or direct estimation methods are introduced in Chapter 10. Chapter 11 discusses segmentation of the motion field in the presence of multiple motion, using direct methods, optical flow methods, and simultaneous motion estimation and segmentation. Two-view motion estimation techniques, discussed in Chapters 9-11, have been found to be highly sensitive to small inaccuracies in the estimates of point correspondences or optical flow. To this effect, motion and structure from stereo pairs and motion tracking over long monocular or stereo sequences are addressed in Chapter 12 for more robust estimation.

Filtering of digital video for such applications as standards conversion, noise reduction, and enhancement and restoration is addressed in Part 4. Video filtering differs from still-image filtering in that it generally employs motion information. To this effect, the basics of motion-compensated filtering are introduced in Chapter 13. Video images often suffer from graininess, especially when viewed in freeze-frame mode. Intraframe, motion-adaptive, and motion-compensated filtering for noise suppression are discussed in Chapter 14. Restoration of blurred video frames is the subject of Chapter 15. Here, motion information can be used in the estimation of the spatial extent of the blurring function. Different digital video applications have different spatio-temporal resolution requirements. Appropriate standards conversion is required to ensure interoperability of various applications by decoupling the spatio-temporal resolution requirements of the source from that of the display. Standards conversion problems, including frame rate conversion and de-interlacing (interlaced to progressive conversion), are covered in Chapter 16. One of the limitations of CCIR 601, CIF, or smaller-format video is the lack of sufficient spatial resolution. In Chapter 17, a comprehensive model for low-resolution video acquisition is presented as well as a novel framework for superresolution which unifies most video filtering problems.

Compression is fundamental for all digital video applications. Parts 5 and 6 are devoted to image and video compression methods, respectively. It is the emergence of video compression standards, such as JPEG, H.261, and MPEG and their VLSI implementations, that makes applications such as all-digital TV, multimedia, and videophone a reality. Chapters 18-21 cover still-image compression methods, which form the basis for the discussion of video compression in Chapters 22-24, In particular, we discuss lossless compression in Chapter 18, DPCM and transform coding in Chapter 19, still-frame compression standards, including binary and gray-scale/color image compression standards, in Chapter 20, and vector quantization and subband coding in Chapter 21. Chapter 22 provides a brief overview of interframe compression methods. International video compression standards such as H.261, MPEG-1, and MPEG-2 are explained in Chapter 23. Chapter 24 addresses very-low-bitrate coding using object-based methods. Finally, several applications of digital video are introduced in Chapter 25.

Bibliography

[Ang 91] P. H. Ang, P. A. Ruetz, and D. Auld, "Video compression makes big gains," *IEEE Spectrum*, pp. 16-19, Oct. 1991.

[Byt 92] "Practical desktop video," *Byte*, Apr. 1992.

[Hsi 93] T. R. Hsing, C.-T. Chen, and J. A. Bellisio, "Video communications and services in the copper loop," *IEEE* Comm. Mag., pp. 63-68, Jan. 1993.

[Lat 89] B. P. Lathi, Modern *Digital* and Analog Communication *Systems*, Second Edition, HRW Saunders, 1989.

[Lip 90] A. Lippman, "HDTV sparks a digital revolution," *Byte*, Dec. 1990.

[Lut 88] A. C. Luther, *Digital Video* in *the PC* Environment, New York, NY: McGraw-Hill, 1988.

[Mil 92] G. M. Miller, *Modern Electronic* Communication, Fourth Edition, Regents, Prentice Hall, 1992.

[Net 89] A. N. Netravali and B. G. Haskell, *Digital* Pictures - Representation *and Compression*, New York, NY: Plenum Press, 1989.

[Onv 94] R. 0. Onvural, Asynchronous Transfer *Mode Networks:* Performance *Issues*, Norwood, MA: Artech House, 1994.

[Pro 94] J. G. Proakis and M. Salehi, Communication *Systems Engineering*, Englewood Cliffs, NJ: Prentice Hall, 1994.

[Spe 91] "B-1SDN and how it works," *IEEE Spectrum*, pp. 39-44, Aug. 1991.

[Spe 92] "Digital video," *IEEE Spectrum*, pp. 24-30, Mar. 1992.

[Spe 93] "Special report: Interactive multimedia," *IEEE* Spectrum, pp. 22-39, Mar. 1993.

[Spe 95] "Digital Television," *IEEE* Spectrum, pp. 34-80, Apr. 1995.

[Sut 92] J. Sutherland and L. Litteral, "Residential video services," *IEEE Comm. Mag.*, pp. 37–41, July 1992.

[Tru 93] H. J. Trussell, "DSP solutions run the gamut for color systems," *IEEE* Signal *Processing Mag.*, pp. 8-23, Apr. 1993.

Chapter 2

TIME-VARYING IMAGE FORMATION MODELS

In this chapter, we present models (in most cases simplistic ones) for temporal variations of the spatial intensity pattern in the image plane. We represent a time-varying image by a function of three continuous variables, $s_c(x_1, x_2, t)$, which is formed by projecting a time-varying three-dimensional (3-D) spatial scene into the two-dimensional (2-D) image plane. The temporal variations in the 3-D scene are usually due to movements of objects in the scene. Thus, time-varying images reflect a projection of 3-D moving objects into the 2-D image plane as a function of time. Digital video corresponds to a spatio-temporally sampled version of this time-varying image. A block diagram representation of the time-varying image formation model is depicted in Figure 2.1.

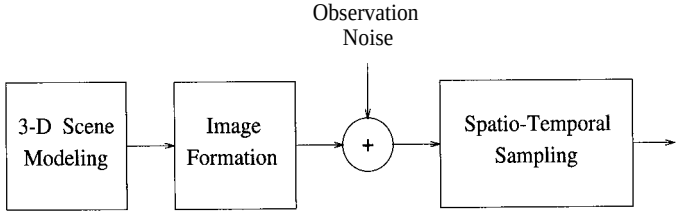


Figure 2.1: Digital video formation

In Figure 2.1, "3-D scene modeling" refers to modeling the motion and structure of objects in 3-D, which is addressed in Section 2.1. "Image formation," which includes geometric and photometric image formation, refers to mapping the 3-D scene into an image plane intensity distribution. Geometric image formation, discussed

in Section 2.2, considers the projection of the 3-D scene into the 2-D image plane. Photometric image formation, which is the subject of Section 2.3, models variations in the image plane intensity distribution due to changes in the scene illumination in time as well as the photometric effects of the 3-D motion. Modeling of the observation noise is briefly discussed in Section 2.4. The spatio-temporal sampling of the time-varying image will be addressed in Chapter 3. The image formation model described in this chapter excludes sudden changes in the scene content.

2.1 Three-Dimensional Motion Models

In this section, we address modeling of the relative 3-D motion between the camera and the objects in the scene. This includes 3-D motion of the objects in the scene, such as translation and rotation, as well as the 3-D motion of the camera, such as zooming and panning. In the following, models are presented to describe the relative motion of a set of 3-D object points and the camera, in the Cartesian coordinate system (Xi, X_2, Xs) and in the homogeneous coordinate system (kX_1, kX_2, kX_3, k), respectively. The depth X_3 of each point appears as a free parameter in the resulting expressions. In practice, a surface model is employed to relate the depth of each object point to reduce the number of free variables (see Chapter 9).

According to classical kinematics, 3-D motion can be classified as rigid motion and nonrigid motion. In the case of rigid motion, the relative distances between the set of 3-D points remain fixed as the object evolves in time. That is, the 3-D structure (shape) of the moving object can be modeled by a nondeformable surface, e.g., a planar, piecewise planar, or polynomial surface. If the entire field of view consists of a single 3-D rigid object, then a single set of motion and structure parameters will be sufficient to model the relative 3-D motion. In the case of independently moving multiple rigid objects, a different parameter set is required to describe the motion of each rigid object (see Chapter 11). In nonrigid motion, a deformable surface model (also known as a deformable template) is utilized in modeling the 3-D structure. A brief discussion about modeling deformable motion is provided at the end of this section.

2.1.1 Rigid Motion in the Cartesian Coordinates

It is well known that 3-D displacement of a rigid object in the Cartesian coordinates can be modeled by an affine transformation of the form [Rog 76, Bal 82, Fol 83]

$$\mathbf{X}' = \mathbf{R}\,\mathbf{X} + \mathbf{T} \tag{2.1}$$

where **R** is a 3 x 3 rotation matrix,

$$\mathbf{T} = \begin{bmatrix} T_1 \\ T_2 \\ T_3 \end{bmatrix}$$

is a 3-D translation vector, and

$$\mathbf{X} = \begin{bmatrix} X_1 \\ X_2 \\ X_3 \end{bmatrix} \text{ and } \mathbf{X}' = \begin{bmatrix} X_1' \\ X_2' \\ X_3' \end{bmatrix}$$

denote the coordinates of an object point at times t and t' with respect to the center of rotation, respectively. That is, the 3-D displacement can be expressed as the sum of a 3-D rotation and a 3-D translation. The rotation matrix **R** can be specified in various forms [Hor 86]. Three of them are discussed next.

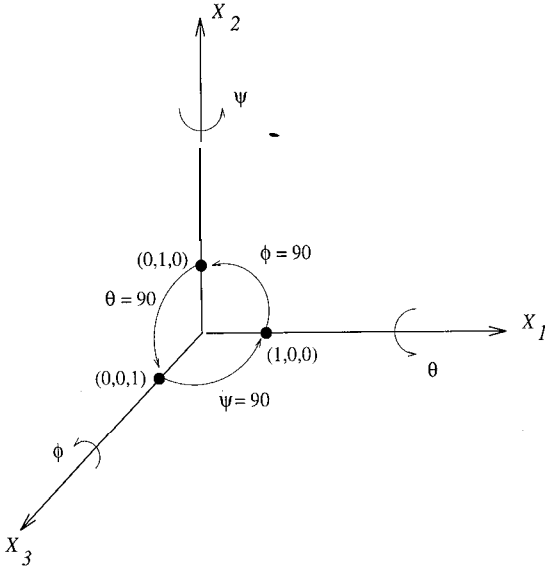


Figure 2.2: Eulerian angles of rotation.

The Rotation Matrix

Three-dimensional rotation in the Cartesian coordinates can be characterized either by the Eulerian angles of rotation about the three coordinate axes, or by an axis of rotation and an angle about this axis. The two descriptions can be shown to be equivalent under the assumption of infinitesimal rotation.

- *Eulerian* angles *in the Cartesian coordinates:* An arbitrary rotation in the 3-D space can be represented by the Eulerian angles, θ, ψ, and ϕ, of rotation about the X_1, X_2, and X_3 axes, respectively. They are shown in Figure 2.2.

The matrices that describe clockwise rotation about the individual axes are given by

$$\mathbf{R}_\theta = \begin{vmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{vmatrix} \tag{2.2}$$

$$\mathbf{R}_\psi = \begin{vmatrix} \cos\psi & 0 & \sin\psi \\ 0 & 1 & 0 \\ -\sin\psi & 0 & \cos\psi \end{vmatrix} \tag{2.3}$$

and

$$\mathbf{R}_\phi = \begin{bmatrix} \cos\phi & -\sin\phi & 0 \\ \sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.4}$$

Assuming that rotation from frame to frame is infinitesimal, i.e., $\phi = \Delta\phi$, etc., and thus approximating $\cos\Delta\phi \approx 1$ and $\sin\Delta\phi \approx \Delta\phi$, and so on, these matrices simplify as

$$\mathbf{R}_\theta = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & -\Delta\theta \\ 0 & \Delta\theta & 1 \end{bmatrix}$$

and

$$\mathbf{R}_\psi = \begin{vmatrix} 1 & 0 & \Delta\psi \\ 0 & 1 & 0 \\ -\Delta\psi & 0 & 1 \end{vmatrix}$$

$$\mathbf{R}_\phi = \begin{vmatrix} 1 & -\Delta\phi & 0 \\ \Delta\phi & 1 & 0 \\ 0 & 0 & 1 \end{vmatrix}$$

Then the composite rotation matrix $\mathbf{R}$ can be found as

$$\mathbf{R} = \mathbf{R}_\phi\mathbf{R}_\theta\mathbf{R}_\psi = \begin{bmatrix} 1 & -\Delta\phi & \Delta\psi \\ \Delta\phi & 1 & -\Delta\theta \\ -\Delta\psi & \Delta\theta & 1 \end{bmatrix} \tag{2.5}$$

Note that the rotation matrices in general do not commute. However, under the infinitesimal rotation assumption, and neglecting the second and higher-order cross-terms in the multiplication, the order of multiplication makes no difference.

• *Rotation about an* arbitrary *axis* in *the* Cartesian *coordinates:* An alternative characterization of the rotation matrix results if the 3-D rotation is described by an angle α about an arbitrary axis through the origin, specified by the directional cosines n_1, n_2, and n_3, as depicted in Figure 2.3.

Then it was shown, in [Rog 76], that the rotation matrix is given by

$$\mathbf{R} = \begin{bmatrix} n_1^2 + (1 - n_1^2)\cos\alpha & n_1 n_2(1-\cos\alpha) - n_3\sin\alpha & n_1 n_3(1-\cos\alpha) + n_2\sin\alpha \\ n_1 n_2(1-\cos\alpha) + n_3\sin\alpha & n_2^2 + (1 - n_2^2)\cos\alpha & n_2 n_3(1-\cos\alpha) - n_1\sin\alpha \\ n_1 n_3(1-\cos\alpha) - n_2\sin\alpha & n_2 n_3(1-\cos\alpha) + n_1\sin\alpha & n_3^2 + (1 - n_3^2)\cos\alpha \end{bmatrix} \tag{2.6}$$

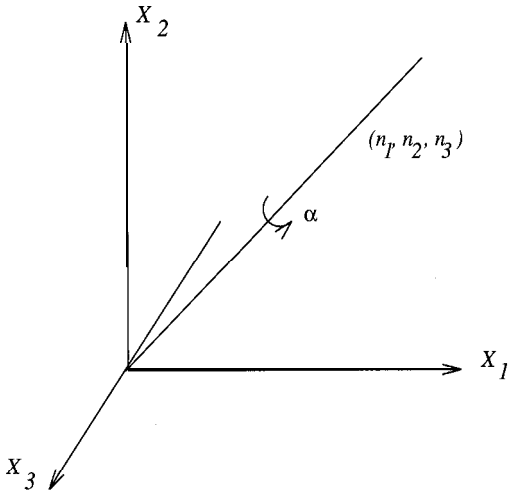


Figure 2.3: Rotation about an arbitrary axis.

For an infinitesimal rotation by the angle $\Delta\alpha$, R reduces to

$$\mathbf{R} = \begin{bmatrix} 1 & -n_3\Delta\alpha & n_2\Delta\alpha \\ n_3\Delta\alpha & 1 & -n_1\Delta\alpha \\ -n_2\Delta\alpha & n_1\Delta\alpha & 1 \end{bmatrix} \tag{2.7}$$

Thus, the two representations are equivalent with

$$\begin{aligned} \text{A0} &= n_1\Delta\alpha \\ \Delta\psi &= n_2\Delta\alpha \\ \Delta\phi &= n_3\Delta\alpha \end{aligned}$$

In video imagery, the assumption of infinitesimal rotation usually holds, since the time difference between the frames are in the order of $1/30$ seconds.

• *Representation* in ♦ℳ□○· of quaternions: A quaternion is an extension of a complex number such that it has four components [Hog 92],

$$\mathbf{q} = q_0 + q_1\mathbf{i} + q_2\mathbf{j} + q_3\mathbf{k} \tag{2.8}$$

where q_0, q_1, q_2, and q_3 are real numbers, and

$$\mathbf{i}^2 = \mathbf{j}^2 = \mathbf{k}^2 = \mathbf{ijk} = -1$$

A unit quaternion, where $q_0^2 + q_1^2 + q_2^2 + q_3^2 = 1$, can be used to describe the change in the orientation of a rigid body due to rotation. It has been shown that the unit quaternion is related to the directional cosines n_1, n_2, n_3 and the solid angle of rotation α in (2.6) as [Hor 86]

$$\mathbf{q} = \begin{pmatrix} n_1 \sin(\alpha/2) \\ n_2 \sin(\alpha/2) \\ n_3 \sin(\alpha/2) \\ \cos(\alpha/2) \end{pmatrix} \tag{2.9}$$

The rotation matrix **R** can then be expressed as

$$\mathbf{R} = \begin{bmatrix} q_0^2 - q_1^2 - q_2^2 - q_3^2 & 2(q_0q_1 + q_2q_3) & 2(q_0q_2 - q_1q_3) \\ 2(q_0q_1 - q_2q_3) & -q_0^2 + q_1^2 - q_2^2 + q_3^2 & 2(q_1q_2 + q_0q_3) \\ 2(q_0q_2 + q_1q_3) & 2(q_1q_2 - q_0q_3) & -q_0^2 - q_1^2 + q_2^2 + q_3^2 \end{bmatrix} \tag{2.10}$$

The representation (2.10) of the rotation matrix in terms of the unit quaternion has been found most helpful for temporal tracking of the orientation of a rotating object (see Chapter 12).

Two observations about the model (2.1) are in order:

i) If we consider the motion of each object point X independently, then the 3-D displacement vector field resulting from the rigid motion can be characterized by a different translation vector for each object point. The expression (2.1), however, describes the 3-D displacement field by a single rotation matrix and a translation vector. Hence, the assumption of a rigid configuration of a set of 3-D object points is implicit in this model.

ii) The effects of camera motion, as opposed to object motion, can easily be expressed using the model (2.1). The camera pan constitutes a special case of this model, in that it is a rotation around an axis parallel to the image plane. Zooming is in fact related to the imaging process, and can be modeled by a change of the focal length of the camera. However, it is possible to incorporate the effect of zooming into the 3-D motion model if we assume that the camera has fixed parameters but the object is artificially scaled up or down. Then, (2.1) becomes

$$\mathbf{X'} = \mathbf{S}\,\mathbf{R}\,\mathbf{X} + \mathbf{T} \tag{2.11}$$

where

$$\mathbf{S} = \begin{bmatrix} S_1 & 0 & 0 \\ 0 & S_2 & 0 \\ 0 & 0 & S_3 \end{bmatrix}$$

is a scaling matrix.

Modeling 3-D Instantaneous Velocity

The model (2.1) provides an expression for the 3-D displacement between two time instants. It is also possible to obtain an expression for the 3-D instantaneous velocity by taking the limit of the 3-D displacement model (2.1) as the interval between the two time instants At goes to zero. Expressing the rotation matrix **R** in terms of infinitesimal Eulerian angles, we have

$$\begin{bmatrix} X_1' \\ X_2' \\ X_3' \end{bmatrix} = \begin{bmatrix} 1 & -\Delta\phi & \Delta\psi \\ \Delta\phi & 1 & -\Delta\theta \\ -\Delta\psi & \Delta\theta & 1 \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \\ X_3 \end{bmatrix} + \begin{bmatrix} T_1 \\ T_2 \\ T_3 \end{bmatrix} \tag{2.12}$$

Decomposing the rotation matrix as

$$\begin{bmatrix} 1 & -\Delta\phi & \Delta\psi \\ \Delta\phi & 1 & -\Delta\theta \\ -\Delta\psi & \Delta\theta & 1 \end{bmatrix} = \begin{bmatrix} 0 & -\Delta\phi & \Delta\psi \\ \Delta\phi & 0 & -\Delta\theta \\ -\Delta\psi & \Delta\theta & 0 \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.13}$$

substituting (2.13) into (2.12), and rearranging the terms, we obtain

$$\begin{bmatrix} X_1' - X_1 \\ X_2' - X_2 \\ X_3' - X_3 \end{bmatrix} = \begin{bmatrix} 0 & -\Delta\phi & \Delta\psi \\ \Delta\phi & 0 & -\Delta\theta \\ -\Delta\psi & \Delta\theta & 0 \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \\ X_3 \end{bmatrix} + \begin{bmatrix} T_1 \\ T_2 \\ T_3 \end{bmatrix} \tag{2.14}$$

Dividing both sides of (2.14) by At and taking the limit as At goes to zero, we arrive at the 3-D velocity model to represent the instantaneous velocity of a point (Xl, X_2, X_3) in the 3-D space as

$$\begin{bmatrix} \dot{X}_1 \\ \dot{X}_2 \\ \dot{X}_3 \end{bmatrix} = \begin{bmatrix} 0 & -\Omega_3 & \Omega_2 \\ \Omega_3 & 0 & -\Omega_1 \\ -\Omega_2 & \Omega_1 & 0 \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \\ X_3 \end{bmatrix} + \begin{bmatrix} V_1 \\ V_2 \\ V_3 \end{bmatrix} \tag{2.15}$$

where Ω_i and V_i denote the angular and linear velocities in the respective directions, i = 1, 2, 3. The model (2.15) can be expressed in compact form as

$$\dot{\mathbf{X}} = \boldsymbol{\Omega} \times \mathbf{X} + \mathbf{V} \tag{2.16}$$

where $\dot{\mathbf{X}} = [\dot{X}_1\ \dot{X}_2\ \dot{X}_3]^T$, $\boldsymbol{\Omega} = [\Omega_1\ \Omega_2\ \Omega_3]^T$, $\mathbf{V} = [V_1\ V_2\ V_3]^T$, and x denotes the cross-product. Note that the instantaneous velocity model assumes that we have a continuous temporal coordinate since it is defined in terms of temporal derivatives.

2.1.2 Rigid Motion in the Homogeneous Coordinates

We define the homogeneous coordinate representation of a Cartesian point $X = [X_1 X_2 X_3]^T$ as

$$\mathbf{X}_h = \begin{bmatrix} kX_1 \\ kX_2 \\ kX_3 \\ k \end{bmatrix} \tag{2.17}$$

Then the affine transformation (2.11) in the Cartesian coordinates can be expressed as a linear transformation in the homogeneous coordinates

$$\mathbf{X}_h' = \tilde{\mathbf{A}}\mathbf{X}_h \tag{2.18}$$

where

$$\tilde{\mathbf{A}} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & T_1 \\ a_{21} & a_{22} & a_{23} & T_2 \\ a_{31} & a_{32} & a_{33} & T_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

and the matrix **A**

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} = \mathbf{SR}$$

Translation in the Homogeneous Coordinates

Translation can be represented as a matrix multiplication in the homogeneous coordinates given by

$$\mathbf{X}_h' = \tilde{\mathbf{T}}\mathbf{X}_h \tag{2.19}$$

where

$$\tilde{\mathbf{T}} = \begin{bmatrix} 1 & 0 & 0 & T_1 \\ 0 & 1 & 0 & T_2 \\ 0 & 0 & 1 & T_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

is the translation matrix.

Rotation in the Homogeneous Coordinates

Rotation in the homogeneous coordinates is represented by a 4 x 4 matrix multiplication in the form

$$\mathbf{X}_h' = \tilde{\mathbf{R}}\mathbf{X}_h \tag{2.20}$$

where

$$\tilde{\mathbf{R}} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & 0 \\ r_{21} & r_{22} & r_{23} & 0 \\ r_{31} & r_{32} & r_{33} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

and r_{ij} denotes the elements of the rotation matrix R in the Cartesian coordinates

Zooming in the Homogeneous Coordinates

The effect of zooming can be incorporated into the 3-D motion model as

$$\mathbf{X}_h' = \tilde{\mathbf{S}}\mathbf{X}_h \tag{2.21}$$

where

$$\tilde{\mathbf{S}} = \begin{bmatrix} S_1 & 0 & 0 & 0 \\ 0 & S_2 & 0 & 0 \\ 0 & 0 & S_3 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

2.1.3 Deformable Motion

Modeling the 3-D structure and motion of nonrigid objects is a complex task. Analysis and synthesis of nonrigid motion using deformable models is an active research area today. In theory, according to the mechanics of deformable bodies [Som 50], the model (2.1) can be extended to include 3-D nonrigid motion as

$$\mathbf{X}' = (\mathbf{D} + \mathbf{R})\mathbf{X} + \mathbf{T} \tag{2.22}$$

where **D** is an arbitrary deformation matrix. Note that the elements of the rotation matrix are constrained to be related to the sines and cosines of the respective angles, whereas the deformation matrix is not constrained in any way. The problem with this seemingly simple model arises in defining the **D** matrix to represent the desired deformations.

Some examples of the proposed 3-D nonrigid motion models include those based on free vibration or deformation modes [Pen 91a] and those based on constraints induced by intrinsic and extrinsic forces [Ter 88a]. Pentland et *al.* [Pen 91a] parameterize nonrigid motion in terms of the eigenvalues of a finite-element model of the deformed object. The recovery of 3-D nonrigid motion using this model requires the knowledge of the geometry of the undeformed object. Terzopoulos *et al.* [Ter 88a] have exploited two intrinsic constraints to design deformable models: surface coherence and symmetry seeking. The former is inherent in the elastic forces prescribed by the physics of deformable continua, and the latter is an attribute of many natural and synthetic objects. Terzopoulos and Fleischer [Ter 88b] also proposed a physically based modeling scheme using mechanical laws of continuous bodies whose

shapes vary in time. They included physical features, such as mass and damping, in their models in order to simulate the dynamics of deformable objects in response to applied forces. Other models include the deformable superquadrics [Ter 91] and extensions of the physically based framework [Met 93]. The main applications of these models have been in image synthesis and animation.

A simple case of 3-D nonrigid models is that of flexibly connected rigid patches, such as a wireframe model where the deformation of the nodes (the so-called local motion) is allowed. In this book, we will consider only 2-D deformable models that is, the effect of various deformations in the image plane; except for the simple case of 3-D flexible wireframe models, that are discussed in Chapter 24.

2.2 Geometric Image Formation

Imaging systems capture 2-D projections of a time-varying 3-D scene. This projection can be represented by a mapping from a 4-D space to a 3-D space,

$$\begin{aligned} f: \quad & R^4 \rightarrow R^3 \\ & (X_1, X_2, X_3, t) \rightarrow (x_1, x_2, t) \end{aligned} \tag{2.23}$$

where (X_1, X_2, X_3), the 3-D world coordinates, (x_1, x_2), the 2-D image plane coordinates, and t, time, are continuous variables, Here, we consider two types of projection, perspective (central) and orthographic (parallel), which are described in the following.

2.2.1 Perspective Projection

Perspective projection reflects image formation using an ideal pinhole camera according to the principles of geometrical optics. Thus, all the rays from the object pass through the center of projection, which corresponds to the center of the lens. For this reason, it is also known as "central projection." Perspective projection is illustrated in Figure 2.4 when the center of projection is between the object and the image plane, and the image plane coincides with the (Xi, X_2) plane of the world coordinate system.

The algebraic relations that describe the perspective transformation for the configuration shown in Figure 2.4 can be obtained based on similar triangles formed by drawing perpendicular lines from the object point (Xi, X_2, Xs) and the image point $(x_1, x_2, 0)$ to the X_3 axis, respectively. This leads to

$$\frac{x_1}{\mathrm{f}} = -\frac{X_1}{X_3 - f} \quad \text{and} \quad \frac{x_2}{f} = -\frac{X_2}{X_3 - f}$$

or

$$x_1 = \frac{fX_1}{f - X_3} \quad \text{and} \quad x_2 = \frac{fX_2}{f - X_3} \tag{2.24}$$

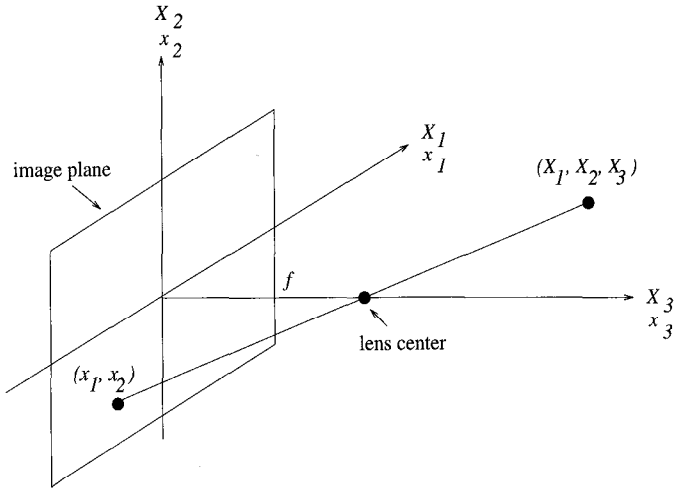


Figure 2.4: Perspective projection model.

where f denotes the distance from the center of projection to the image plane.

If we move the center of projection to coincide with the origin of the world coordinates, a simple change of variables yields the following equivalent expressions:

$$x_1 = \frac{fX_1}{X_3} \quad \text{and} \quad x_2 = \frac{fX_2}{X_3} \tag{2.25}$$

The configuration and the similar triangles used to obtain these expressions are shown in Figure 2.5, where the image plane is parallel to the (X_1, X_2) plane of the world coordinate system. Observe that the latter expressions can also be employed as an approximate model for the configuration in Figure 2.4 when $X_3 >> f$ with the reversal of the sign due to the orientation of the image being the same as the object, as opposed to being a mirror image as in the actual image formation. The general form of the perspective projection, when the image plane is not parallel to the (X_1, X_2) plane of the world coordinate system, is given in [Ver 89].

We note that the perspective projection is nonlinear in the Cartesian coordinates since it requires division by the X_3 coordinate. However, it can be expressed as a linear mapping in the homogeneous coordinates, as

$$\begin{bmatrix} \ell x_1 \\ \ell x_2 \\ \ell \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} kX_1 \\ kX_2 \\ kX_3 \\ k \end{bmatrix} \tag{2.26}$$

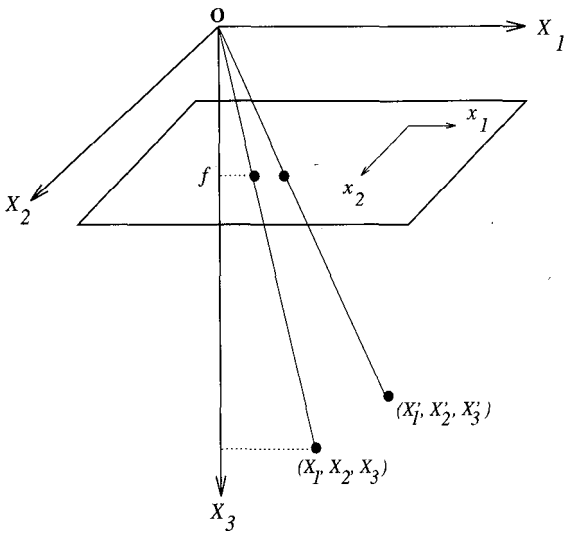


Figure 2.5: Simplified perspective projection model.

where

$$\mathbf{X}_h = \begin{bmatrix} kX_1 \\ kX_2 \\ kX_3 \\ k \end{bmatrix}$$

and

$$\mathbf{x}_h = \begin{bmatrix} \ell x_1 \\ \ell x_2 \\ \ell \end{bmatrix}$$

denote the world and image plane points, respectively, in the homogeneous coordinates.

2.2.2 Orthographic Projection

Orthographic projection is an approximation of the actual imaging process where it is assumed that all the rays from the 3-D object (scene) to the image plane travel parallel to each other. For this reason it is sometimes called the "parallel projection." Orthographic projection is depicted in Figure 2.6 when the image plane is parallel to the $X_1 - X_2$ plane of the world coordinate system.

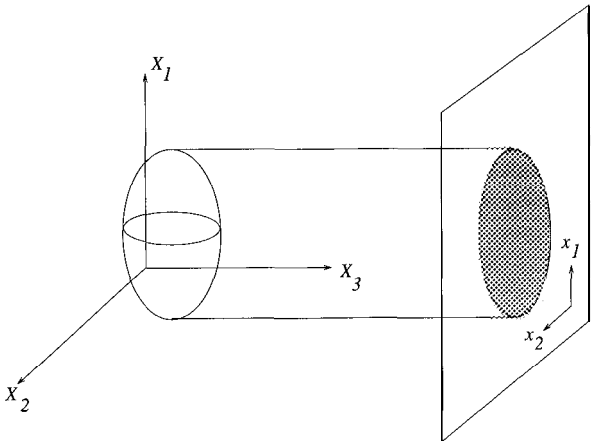


Figure 2.6: Orthographic projection model.

Provided that the image plane is parallel to the $X_1 - X_2$ plane of the world coordinate system, the orthographic projection can be described in Cartesian coordinates as

$$x_1 = X_1 \quad \text{and} \quad x_2 = X_2 \tag{2.27}$$

or in vector-matrix notation as

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \\ X_3 \end{bmatrix} \tag{2.28}$$

where x_1 and x_2 denote the image plane coordinates.

The distance of the object from the camera does not affect the image plane intensity distribution in orthographic projection. That is, the object always yields the same image no matter how far away it is from the camera. However, orthographic projection provides good approximation to the actual image formation process when the distance of the object from the camera is much larger than the relative depth of points on the object with respect to a coordinate system on the object itself. In such cases, orthographic projection is usually preferred over more complicated but realistic models because it is a linear mapping and thus leads to algebraically and computationally more tractable algorithms.

2.3 Photometric Image Formation

Image intensities can be modeled as proportional to the amount of light reflected by the objects in the scene. The scene reflectance function is generally assumed to contain a Lambertian and a specular component. In this section, we concentrate on surfaces where the specular component can be neglected. Such surfaces are called Lambertian surfaces. Modeling of specular reflection is discussed in [Dri 92]. More sophisticated reflectance models can be found in [Hor 86, Lee 90].

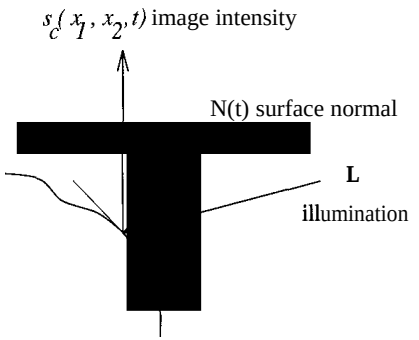


Figure 2.7: Photometric image formation model

2.3.1 Lambertian Reflectance Model

If a Lambertian surface is illuminated by a single point-source with uniform intensity (in time), the resulting image intensity is given by [Hor 86]

$$s_c(x_1, x_2, t) = \rho \mathbf{N}(t)\, \mathbf{L} \tag{2.29}$$

where ρ denotes the surface albedo, i.e., the fraction of the light reflected by the surface, $\mathbf{L} = (L_1, L_2, L_3)$ is the unit vector in the mean illuminant direction, and N(t) is the unit surface normal of the scene, at spatial location $(X_1, X_2, X_3(X_1, X_2))$ and time t, given by

$$\mathbf{N}(t) = (-p, -q, 1)/(p^2 + q^2 + 1)^{1/2} \tag{2.30}$$

in which $p = \partial X_3/\partial x_1$ and $q = \partial X_3/\partial x_2$ are the partial derivatives of depth $X_3(x_1, x_2)$ with respect to the image coordinates x_1 and x_2, respectively, under the orthographic projection. Photometric image formation for a static surface is illustrated in Figure 2.7.

The illuminant direction can also be expressed in terms of tilt and slant angles as [Pen 91b]

$$\mathbf{L} = (L_1, L_2, L_3) = (\cos\tau \sin\sigma, \sin\tau \sin\sigma, \cos\sigma) \tag{2.31}$$

where τ, the tilt angle of the illuminant, is the angle between **L** and the $X_1 - X_3$ plane, and σ, the slant angle, is the angle between **L** and the positive X_3 axis.

2.3.2 Photometric Effects of 3-D Motion

As an object moves in 3-D, the surface normal changes as a function of time; so do the photometric properties of the surface. Assuming that the mean illuminant direction **L** remains constant, we can express the change in the intensity due to photometric effects of the motion as

$$\frac{ds_c(x_1, x_2, t)}{dt} = \rho \mathbf{L} \cdot \frac{d\mathbf{N}(t)}{dt} \tag{2.32}$$

The rate of change of the normal vector N at the point (X_1, X_2, X_3) can be approximated by

$$\frac{d\mathbf{N}}{dt} \approx \frac{\mathbf{AN}}{\mathrm{At}}$$

where **AN** denotes the change in the direction of the normal vector due to the 3-D motion from the point (X_1, X_2, X_3) to (X_1', X_2', X_3') within the period At. This change can be expressed as

$$\begin{aligned} \mathbf{A}\,\mathbf{N} &= \mathbf{N}(X_1', X_2', X_3') - \mathbf{N}(X_1, X_2, X_3) \\ &= \frac{(-p', -q', 1)}{(p'^2 + q'^2 + 1)^{1/2}} - \frac{(-p, -q, 1)}{(p^2 + q^2 + 1)^{1/2}} \end{aligned} \tag{2.33}$$

where p' and q' denote the components of $\mathbf{N}(X_1', X_2', X_3')$ given by

$$\begin{aligned} p' &= \frac{\partial X_3'}{\partial x_1'} - \frac{\partial X_3'}{\partial x_1}\frac{\partial x_1}{\partial x_1'} \\ &= \frac{-\Delta\psi + p}{1 + \Delta\psi p} \\ q' &= \frac{\partial X_3'}{\partial x_2'} - \frac{\Delta\theta + q}{1 - \Delta\theta q} \end{aligned} \tag{2.34}$$

Pentland [Pen 91b] shows that the photometric effects of motion can dominate the geometric effects in some cases.

2.4 Observation Noise

Image capture mechanisms are never perfect. As a result, images generally suffer from graininess due to electronic noise, photon noise, film-grain noise, and quantization noise. In video scanned from motion picture film, streaks due to possible scratches on film can be modeled as impulsive noise. Speckle noise is common

in radar image sequences and biomedical tine-ultrasound sequences. The available signal-to-noise ratio (SNR) varies with the imaging devices and image recording media. Even if the noise may not be perceived at full-speed video due to the temporal masking effect of the eye, it often leads to poor-quality "freeze-frames."

The observation noise in video can be modeled as additive or multiplicative noise, signal-dependent or signal-independent noise, and white or colored noise. For example, photon and film-grain noise are signal-dependent, whereas CCD sensor and quantization noise are usually modeled as white, Gaussian distributed, and signal-independent. Ghosts in TV images can also be modeled as signal-dependent noise. In this book, we will assume a simple additive noise model given by

$$g_c(x_1, x_2, t) = s_c(x_1, x_2, t) + v_c(x_1, x_2, t) \qquad (2.35)$$

where $s_c(x_1, x_2,$ t) and $v_c(x_1, x_2, t)$ denote the ideal video and noise at time t, respectively.

The SNR is an important parameter for most digital video processing applications, because noise hinders our ability to effectively process the data. For example, in 2-D and 3-D motion estimation, it is very important to distinguish the variation of the intensity pattern due to motion from that of the noise. In image resolution enhancement, noise is the fundamental limitation on our ability to recover high-frequency information. Furthermore, in video compression, random noise increases the entropy hindering effective compression. The SNR of video imagery can be enhanced by spatio-temporal filtering, also called noise filtering, which is the subject of Chapter 14.

2.5 Exercises

1. Suppose a rotation by 45 degrees about the X_2 axis is followed by another rotation by 60 degrees about the X_1 axis. Find the directional cosines n_1, n_2, n_3 and the solid angle α to represent the composite rotation.
2. Given a triangle defined by the points (1,1,1), (1,-1,0) and (0,1,-1). Find the vertices of the triangle after a rotation by 30 degrees about an axis passing through (0,1,0) which is parallel to the X_1 axis.
3. Show that a rotation matrix is orthonormal.
4. Derive Equation (2.6).
5. Show that the two representations of the rotation matrix R given by (2.6) and (2.10) are equivalent.
6. Discuss the conditions under which the orthographic projection provides a good approximation to imaging through an ideal pinhole camera.
7. Show that the expressions for p' and q' in (2.34) are valid under the orthographic projection.

Bibliography

[Bal 82] D. H. Ballard and C. M. Brown, *Computer Vision,* Englewood Cliffs, NJ: Prentice-Hall, 1982.

[Dri 92] J. N. Driessen, *Motion Estimation for Digital Video,* Ph.D. Thesis, Delft University of Technology, 1992.

[Fol 83] J. D. Foley and A. Van Dam, *Fundamentals* of Interactive *Computer Graphics,* Reading, MA: Addison-Wesley, 1983.

[Hog 92] S. G. Hoggar, *Mathematics* for Computer *Graphics,* Cambridge University Press, 1992.

[Hor 86] B. K. P. Horn, *Robot Vision,* Cambridge, MA: MIT Press, 1986.

[Lee 90] H. C. Lee, E. J. Breneman, and C. P. Schutte, "Modeling light reflection for computer color vision," *IEEE Trans. Putt.* Anal. *Mach. Intel.,* vol. PAMI-12, pp. 402-409, Apr. 1990.

[Met 93] D. Metaxas and D. Terzopoulos, "Shape and nonrigid motion estimation through physics-based synthesis," *IEEE* Trans. *Patt. Anal. Mach. Intel.,* vol. 15, pp. 580-591, June 1993.

[Pen 91a] A. Pentland and B. Horowitz, "Recovery of nonrigid motion and structure," *IEEE Trans. Patt.* Anal. *Mach. Intel.,* vol. 13, pp. 730–742, 1991.

[Pen 91b] A. Pentland, "Photometric motion," *IEEE* Trans. *Patt.* Anal. Mach. *Intel.,* vol. PAMI-13, pp. 879–890, Sep. 1991.

[Rog 76] D. F. Rogers and J. A. Adams, *Mathematical Elements* for *Computer* Graphics, New York, NY: McGraw Hill, 1976.

[Som 50] A. Sommerfeld, Mechanics *of Deformable Bodies,* 1950.

[Ter 88a] D. Terzopoulos, A. Witkin, and M. Kass, "Constraints on deformable models: Recovering 3-D shape and nonrigid motion," *Artif. Intel.,* vol. 36, pp. 91-123, 1988.

[Ter 88b] D. Terzopoulos and K. Fleischer, "Deformable models," *Visual Comput.,* vol. 4, pp. 306–331, 1988.

[Ter 91] D. Terzopoulos and D. Metaxas, "Dynamic 3-D models with local and global deformations: Deformable superquadrics," *IEEE* Trans. *Putt. Anal. Mach. Intel.,* vol. PAMI-13, pp. 703-714, July 1991.

[Ver 89] A. Verri and T. Poggio, "Motion field and optical flow: Qualitative properties," *IEEE* Trans. Putt. *Anal. Mach. Intel.,* vol. PAMI-11, pp. 490–498, May 1989.

Chapter 3

SPATIO-TEMPORAL SAMPLING

In order to obtain an analog *or digital* video signal representation, the continuous time-varying image $s_c(x_1, x_2, t)$ needs to be sampled in both the spatial and temporal variables. An analog video signal representation requires sampling $s_c(x_1, x_2, t)$ in the vertical and temporal dimensions. Recall that an analog video signal is a 1-D continuous function, where one of the spatial dimensions is mapped onto time by means of the scanning process. For a digital video representation, $s_c(x_1, x_2, t)$ is sampled in all three dimensions. The spatio-temporal sampling process is depicted in Figure 3.1, where (n_1, n_2, k) denotes the discrete spatial and temporal coordinates, respectively.

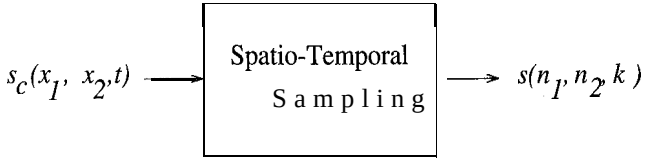


Figure 3.1: Block diagram.

Commonly used 2-D and 3-D sampling structures for the representation of analog and digital video are shown in Section 3.1. Next, we turn our attention to the frequency domain characterization of sampled video. In order to motivate the main principles, we start with the sampling of still images. Section 3.2 covers the case of sampling on a 2-D rectangular grid, whereas Section 3.3 treats sampling on arbitrary 2-D periodic grids. In Section 3.4, we address extension of this theory to sampling of

multidimensional signals on lattices and other periodic sampling structures. However, the discussion is limited to sampling of spatio-temporal signals $s_c(x_1, x_2, t)$ on 3-D lattices, considering the scope of the book. Finally, Section 3.5 addresses the reconstruction of continuous time-varying images from spatio-temporally sampled representations. The reader is advised to review the remarks about the notation used in this book on page xxi, before proceeding with this chapter.

3.1 Sampling for Analog and Digital Video

Some of the more popular sampling structures utilized in the representation of analog and digital video are introduced in this section.

3.1.1 Sampling Structures for Analog Video

An analog video signal is obtained by sampling the time-varying image intensity distribution in the vertical x_2, and temporal t directions by a 2-D sampling process known as scanning. Continuous intensity information along each horizontal line is concatenated to form the 1-D analog video signal as a function of time. The two most commonly used vertical-temporal sampling structures are the orthogonal sampling structure, shown in Figure 3.2, and the hexagonal sampling structure, depicted in Figure 3.3.

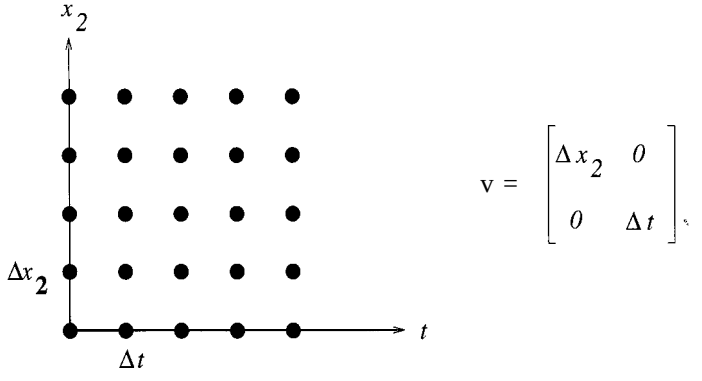


Figure 3.2: Orthogonal sampling structure for progressive analog video.

In these figures, each dot indicates a continuous line of video perpendicular to the plane of the page. The matrices **V** shown in these figures are called the sampling matrices, and will be defined in Section 3.3. The orthogonal structure is used in the representation of progressive analog video, such as that shown on workstation

monitors, and the hexagonal structure is used in the representation of 2:1 interlaced analog video, such as that shown on TV monitors. The spatio-temporal frequency content of these signals will be analyzed in Sections 3.2 and 3.3, respectively.

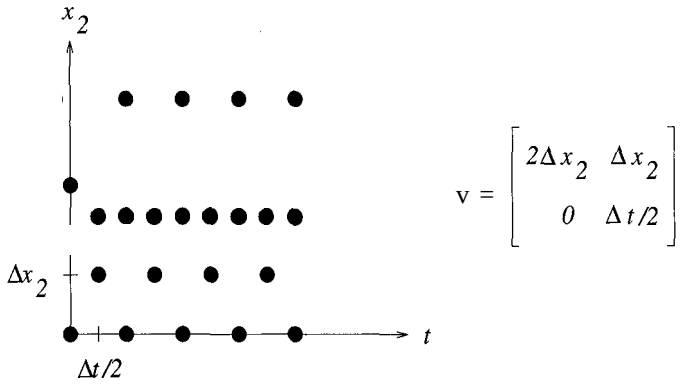


Figure 3.3: Hexagonal sampling structure for 2:1 interlaced analog video.

3.1.2 Sampling Structures for Digital Video

Digital video can be obtained by sampling analog video in the horizontal direction along the scan lines, or by applying an inherently 3-D sampling structure to sample the time-varying image, as in the case of some solid-state sensors. Examples of the most popular 3-D sampling structures are shown in Figures 3.4, 3.5, 3.6, and 3.7. In these figures, each circle indicates a pixel location, and the number inside the circle indicates the time of sampling. The first three sampling structures are lattices, whereas the sampling structure in Figure 3.7 is not a lattice, but a union of two cosets of a lattice. The vector c in Figure 3.7 shows the displacement of one coset with respect to the other. Other 3-D sampling structures can be found in [Dub 85].

The theory of sampling on lattices and other special M-D structures is presented in Section 3.4. It will be seen that the most suitable sampling structure for a time-varying image depends on its spatio-temporal frequency content. The sampling structures shown here are field- or frame-instantaneous; that is, a complete field or frame is acquired at one time instant. An alternative strategy is time-sequential sampling, where individual samples are taken one at a time according to a prescribed ordering which is repeated after one complete frame. A theoretical analysis of time-sequential sampling can be found in [Rah 92].

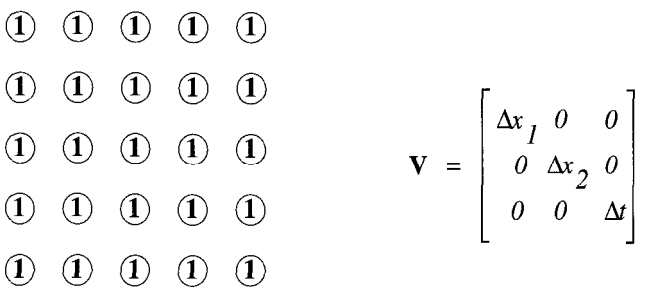


Figure 3.4: Orthogonal sampling lattice [Dub 85] (©1985 IEEE).

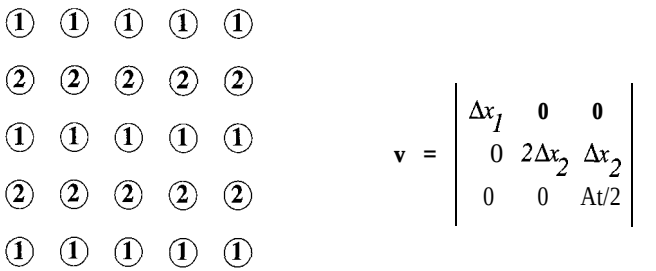


Figure 3.5: Vertically aligned 2:1 line-interlaced lattice [Dub 85] (©1985 IEEE).

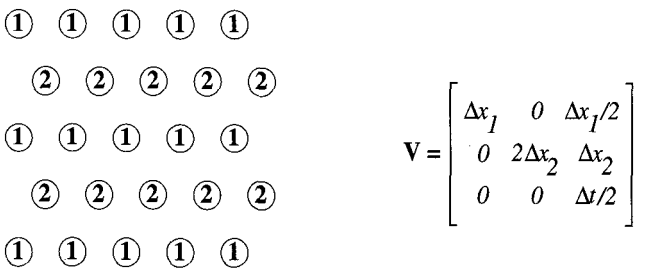
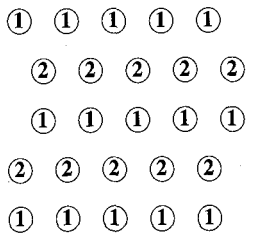


Figure 3.6: Field-quincunx sampling lattice [Dub 85] (©1985 IEEE).



$$\mathbf{V} = \begin{bmatrix} \Delta x_1 & \Delta x_1/2 & 0 \\ 0 & 2\Delta x_2 & 0 \\ 0 & 0 & \Delta t \end{bmatrix} \quad \mathbf{c} = \begin{bmatrix} 0 \\ \Delta x_2 \\ \Delta t/2 \end{bmatrix}$$

Figure 3.7: Line-quincunx sampling lattice [Dub 85] (©1985 IEEE).

3.2 Two-Dimensional Rectangular Sampling

In this section, we discuss 2-D rectangular sampling of a still image, $s_c(x_1, x_2)$ in the two spatial coordinates. However, the same analysis also applies to vertical-temporal sampling (as in the representation of progressive analog video). In spatial rectangular sampling, we sample at the locations

$$\begin{aligned} x_1 &= n_1 \Delta x_1 \\ x_2 &= n_2 \Delta x_2 \end{aligned} \tag{3.1}$$

where Δx_1 and Δx_2 are the sampling distances in the x_1 and x_2 directions, respectively. The 2-D rectangular sampling grid is depicted in Figure 3.8. The sampled signal can be expressed, in terms of the unitless coordinate variables (n_1, n_2), as

$$s(n_1, n_2) = s_c(n_1 \Delta x_1, n_2 \Delta x_2), \quad (n_1, n_2) \in \mathbf{Z}^2. \tag{3.2}$$

In some cases, it is convenient to define an intermediate sampled signal in terms of the continuous coordinate variables, given by

$$\begin{aligned} s_p(x_1, x_2) &= s_c(x_1, x_2) \sum_{n_1} \sum_{n_2} \delta(x_1 - n_1 \Delta x_1, x_2 - n_2 \Delta x_2) \\ &= \sum_{n_1} \sum_{n_2} s_c(n_1 \Delta x_1, n_2 \Delta x_2) \delta(x_1 - n_1 \Delta x_1, x_2 - n_2 \Delta x_2) \\ &= \sum_{n_1} \sum_{n_2} s(n_1, n_2) \delta(x_1 - n_1 \Delta x_1, x_2 - n_2 \Delta x_2) \end{aligned} \tag{3.3}$$

Note that $s_p(x_1, x_2)$ is indeed a sampled signal because of the presence of the 2-D Dirac delta function $\delta(\cdot, \cdot)$.

3.2.1 2-D Fourier Transform Relations

We start by reviewing the 2-D continuous-space Fourier transform (FT) relations. The 2-D FT $S_c(F_1, F_2)$ of a signal with continuous variables $s_c(x_1, x_2)$ is given by

$$S_c(F_1, F_2) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} s_c(x_1, x_2) \exp\{-j2\pi(F_1 x_1 + F_2 x_2)\} dx_1 dx_2 \tag{3.4}$$

where $(F_1, F_2) \in \mathbf{R}^2$, and the inverse 2-D Fourier transform is given by

$$s_c(x_1, x_2) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} S_c(F_1, F_2) \exp\{j2\pi(F_1 x_1 + F_2 x_2)\} dF_1 dF_2 \tag{3.5}$$

Here, the spatial frequency variables F_1 and F_2 have the units in cycles/mm and are related to the radian frequencies by a scale factor of 2π.

In order to evaluate the 2-D FT $S_p(F_1, F_2)$ of $s_p(x_1, x_2)$, we substitute (3.3) into (3.4), and exchange the order of integration and summation, to obtain

$$\begin{aligned} S_p(F_1, F_2) = \sum_{n_1} \sum_{n_2} & s_c(n_1 \Delta x_1, n_2 \Delta x_2) \\ & \int \int \delta(x_1 - n_1 \Delta x_1, x_2 - n_2 \Delta x_2) \exp\{-j2\pi(F_1 x_1 + F_2 x_2)\} dx_1 dx_2 \end{aligned}$$

which simplifies as

$$S_p(F_1, F_2) = \sum_{n_1} \sum_{n_2} s_c(n_1 \Delta x_1, n_2 \Delta x_2) \exp\{-j2\pi(F_1 n_1 \Delta x_1 + F_2 n_2 \Delta x_2)\} \tag{3.6}$$

Notice that $S_p(F_1, F_2)$ is periodic with the fundamental period given by the region $F_1 < |1/(2\Delta x_1)|$ and $F_2 < |1/(2\Delta x_2)|$.

Letting $f_i = F_i \Delta x_i$, i = 1, 2, and using (3.2), we obtain the discrete-space Fourier transform relation, in terms of the unitless frequency variables f_1 and f_2, as

$$S(f_1, f_2) = S_p\left(\frac{f_1}{\Delta x_1}, \frac{f_2}{\Delta x_2}\right) = \sum_{n_1=-\infty}^{\infty} \sum_{n_2=-\infty}^{\infty} s(n_1, n_2) \exp\{-j2\pi(f_1 n_1 + f_2 n_2)\} \tag{3.7}$$

The 2-D discrete-space inverse Fourier transform is given by

$$s(n_1, n_2) = \int_{-\frac{1}{2}}^{\frac{1}{2}} \int_{-\frac{1}{2}}^{4} S(f_1, f_2) \exp\{j2\pi(f_1 n_1 + f_2 n_2)\} df_1 df_2 \tag{3.8}$$

Recall that the discrete-space Fourier transform $S(f_1, f_2)$ is periodic with the fundamental period $f_1 < |1/2|$ and $f_2 < |1/2|$.

3.2.2 Spectrum of the Sampled Signal

We now relate the Fourier transform, $S_p(F_1, F_2)$ or $S(f_1, f_2)$, of the sampled signal to that of the continuous signal. The standard approach is to start with (3.3), and express $S_p(F_1, F_2)$ as the 2-D convolution of the Fourier transforms of the continuous signal and the impulse train [Opp 89, Jai 90], using the modulation property of the FT,

$$S_p(F_1, F_2) = S_c(F_1, F_2) * *\mathcal{F}\{\sum_{n_1}\sum_{n_2} \delta(x_1 - n_1\Delta x_1, x_2 - n_2\Delta x_2)\} \qquad (3.9)$$

where $\mathcal{F}$ denotes 2-D Fourier transformation, which simplifies to yield (3.11).

Here, we follow a different derivation which can be easily extended to other periodic sampling structures [Dud 84, Dub 85]. First, substitute (3.5) into (3.2), and evaluate x_1 and x_2 at the sampling locations given by (3.1) to obtain

$$s(n_1, n_2) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} S_c(F_1, F_2) \exp\{j2\pi(F_1 n_1 \Delta x_1 + F_2 n_2 \Delta x_2)\} dF_1 dF_2$$

After the change of variables $f_1 = F_1\Delta x_1$ and $f_2 = F_2\Delta x_2$, we have

$$s(n_1, n_2) = \frac{1}{\Delta x_1 \Delta x_2}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} S_c(\frac{f_1}{\Delta x_1}, \frac{f_2}{\Delta x_2}) \exp\{j2\pi(f_1 n_1 + f_2 n_2)\}\, df_1 df_2$$

Next, break the integration over the (f_1, f_2) plane into a sum of integrals each over a square denoted by $SQ(k_1, k_2)$,

$$s(n_1, n_2) = \sum_{k_1}\sum_{k_2} \frac{1}{\Delta x_1 \Delta x_2}\int\int_{SQ} S_c(\frac{f_1}{\Delta x_1}, \frac{f_2}{\Delta x_2}) \exp\{j2\pi(f_1 n_1 + f_2 n_2)\} df_1 df_2$$

where $SQ(k_1, k_2)$ is defined as

$$-\frac{1}{2} + k_1 \le f_1 \le \frac{1}{2} + k_1 \quad \text{and} \quad -\frac{1}{2} + k_2 \le f_2 \le \frac{1}{2} + k_2$$

Another change of variables, $f_1 = f_1 - k_1$ and $f_2 = f_2 - k_2$, shifts all the squares $SQ(k_1, k_2)$ down to the fundamental period $(-\frac{1}{2}, \frac{1}{2})$ x $(-\frac{1}{2}, \frac{1}{2})$, to yield

$$s(n_1, n_2) = \int_{-\frac{1}{2}}^{\frac{1}{2}}\int_{-\frac{1}{2}}^{\frac{1}{2}} \{\frac{1}{\Delta x_1 \Delta x_2}\sum_{k_1}\sum_{k_2} S_c(\frac{f_1 - k_1}{\Delta x_1}, \frac{f_2 - k_2}{\Delta x_2})\} \exp\{j2\pi(f_1 n_1 + f_2 n_2)\} \exp\{-j2\pi(k_1 n_1 + k_2 n_2)\} df_1 df_2 \quad (3.10)$$

But $\exp\{-j2\pi(k_1 n_1 + k_2 n_2)\} = 1$ for k_1, k_2, n_1, n_2 integers. Thus, the frequencies $(f_1 - k_1, f_2 - k_2)$ map onto (f_1, f_2). Comparing the last expression with (3.8), we therefore conclude that

$$S(f_1, f_2) = \frac{1}{\Delta x_1 \Delta x_2}\sum_{k_1}\sum_{k_2} S_c(\frac{f_1 - k_1}{\Delta x_1}, \frac{f_2 - k_2}{\Delta x_2}) \qquad (3.11)$$

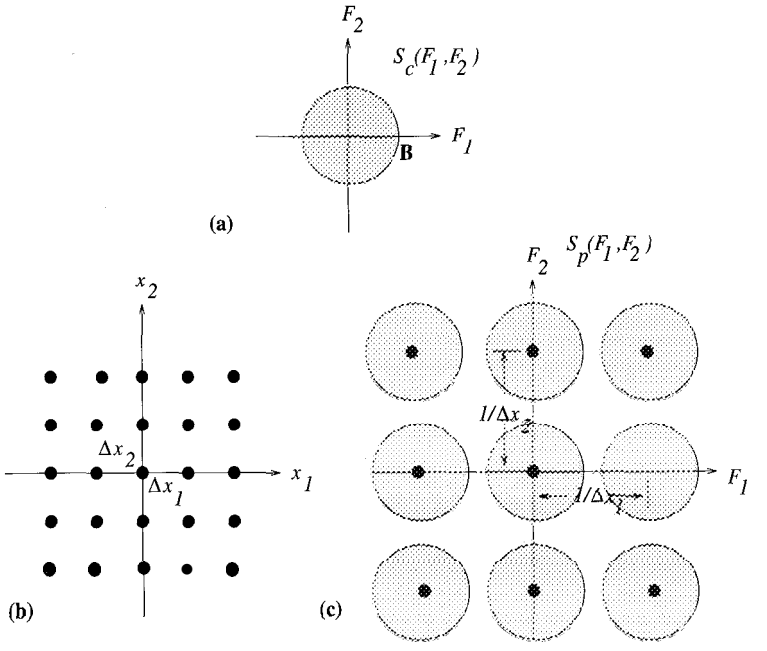


Figure 3.8: Sampling on a 2-D rectangular grid: a) support of the Fourier spectrum of the continuous image; b) the sampling grid; c) spectral support of the sampled image.

or, equivalently,

$$S_p(F_1, F_2) = \frac{1}{\Delta x_1 \Delta x_2}\sum_{k_1}\sum_{k_2} S_c(F_1 - \frac{k_1}{\Delta x_1}, F_2 - \frac{k_2}{\Delta x_2}) \qquad (3.12)$$

We see that, as a result of sampling, the spectrum of the continuous signal replicates in the 2-D frequency plane according to (3.11). The case when the continuous signal is bandlimited with a circular spectral support of radius $B < \max\{1/(2\Delta x_1), 1/(2\Delta x_2)\}$ is illustrated in Figure 3.8.

3.3 Two-Dimensional Periodic Sampling

In this section we extend the results of the previous section to arbitrary 2-D periodic sampling grids.

3.3.1 Sampling Geometry

An arbitrary 2-D periodic sampling geometry can be defined by two basis vectors $\mathbf{v}_1 = (v_{11}\ v_{21})^T$ and $\mathbf{v}_2 = (v_{12}\ v_{22})^T$, such that every sampling location can be expressed as a linear combination of them, given by

$$\begin{aligned} x_1 &= v_{11}n_1 + v_{12}n_2 \\ x_2 &= v_{21}n_1 + v_{22}n_2 \end{aligned} \tag{3.13}$$

In vector-matrix form, we have

$$\mathbf{x} = \mathbf{V}\mathbf{n} \tag{3.14}$$

where

$$\mathrm{x} = (x_1\ x_2)^T, \quad \mathbf{n} = (n_1\ n_2)^T$$

and

$$\mathbf{V} = [\mathbf{v}_1|\mathbf{v}_2]$$

is the sampling matrix. The sampling locations for an arbitrary periodic grid are depicted in Figure 3.9.

Then, analogous to (3.2) and (3.3), the sampled signal can be expressed as

$$\mathbf{s(n)} = s_c(\mathbf{Vn}), \quad \mathbf{n} \in \mathbf{Z}^2 \tag{3.15}$$

or as

$$\begin{aligned} s_p(\mathbf{x}) &= s_c(\mathbf{x}) \sum_{\mathbf{n}\in\mathbf{Z}^2} \delta(\mathbf{x} - \mathbf{Vn}) \\ &= \sum_{\mathbf{n}} s_c(\mathbf{Vn})\delta(\mathbf{x} - \mathbf{Vn}) = \sum_{\mathbf{n}} s(\mathbf{n})\delta(\mathbf{x} - \mathbf{Vn}) \end{aligned} \tag{3.16}$$

3.3.2 2-D Fourier Transform Relations in Vector Form

Here, we restate the 2-D Fourier transform relations given in Section 3.2.1 in a more compact vector-matrix form as follows:

$$\mathbf{S,(F)} = \int_{-\infty}^{\infty} s_c(\mathbf{x})\ \exp\{-j2\pi\mathbf{F}^T\mathbf{x}\}\ \boldsymbol{dx} \tag{3.17}$$

$$s_c(\mathbf{x}) = \int_{-\infty}^{\infty} S_c(\mathbf{F})\ \exp\{j2\pi\mathbf{F}^T\mathbf{x}\}\ d\mathbf{F} \tag{3.18}$$

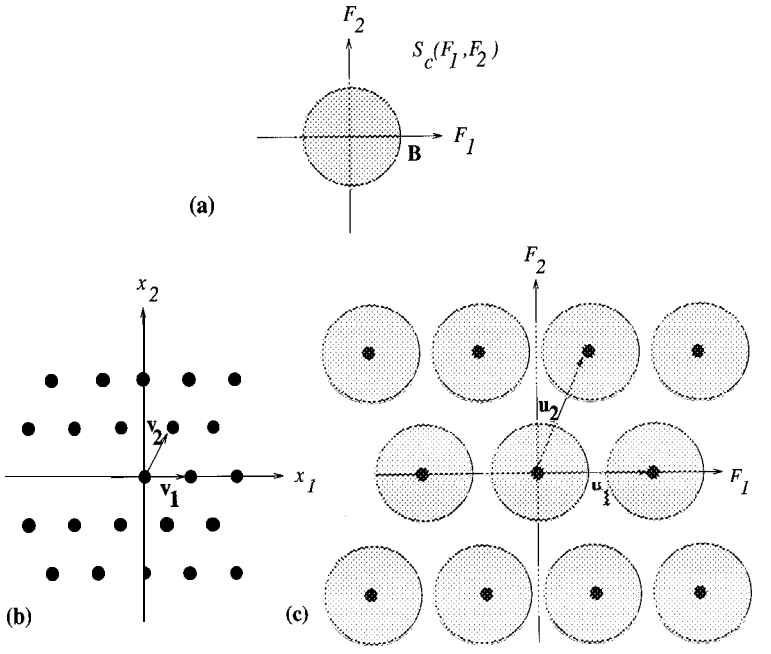


Figure 3.9: Sampling on an arbitrary 2-D periodic grid: a) support of the Fourier spectrum of the continuous image; b) the sampling grid; c) spectral support of the sampled image.

where $\mathrm{x} = (x_1\ x_2)^T$ and $\mathbf{F} = (F_1\ F_2)^T$. We also have

$$\mathbf{S,(F)} = \sum_{\mathbf{n}=-\infty}^{\infty} s_c(\mathbf{Vn})\ \exp\{-j2\pi\mathbf{F}^T\mathbf{Vn}\} \tag{3.19}$$

or

$$\mathbf{S(f)} = \sum_{\mathbf{n}=-\infty}^{\infty} \mathbf{s(n)}\ \exp\{-j2\pi\mathbf{f}^T\mathbf{n}\} \tag{3.20}$$

$$\mathbf{s(n)} = \int_{-\frac{1}{2}}^{\frac{1}{2}} \mathbf{S(f)}\ \exp\{j2\pi\mathbf{f}^T\mathbf{n}\}\ \mathbf{df} \tag{3.21}$$

where $\mathbf{f} = (f_1\ f_2)^T$. Note that the integrations and summations in these relations are in fact double integrations and summations.

3.3.3 Spectrum of the Sampled Signal

To derive the relationship between $\mathbf{S(f)}$ and $S_c(\mathbf{F})$, we follow the same steps as in Section 3.2. Thus, we start by substituting (3.18) into (3.15) as

$$\mathbf{s(n)} = s_c(\mathbf{Vn}) = \int_{-\infty}^{\infty} S_c(\mathbf{F}) \exp\{j2\pi \mathbf{F}^T \mathbf{Vn}\} d\mathbf{F}$$

Making the change of variables $\mathbf{f} = \mathbf{V}^T\mathbf{F}$, we have

$$\mathbf{s(n)} = \int_{-\infty}^{\infty} \frac{1}{|det\mathbf{V}|} S_c(\mathbf{Uf}) \exp\{j2\pi \mathbf{f}^T \mathbf{n}\} d\mathbf{f}$$

where $\mathbf{U} = \mathbf{V}^{T^{-1}}$ and $df = |det\mathbf{V}|d\mathbf{F}$.

Expressing the integration over the $\mathbf{f}$ plane as a sum of integrations over the squares $(-1/2, 1/2) \times (-1/2, 1/2)$, we obtain

$$\mathbf{s(n)} = \int_{-\frac{1}{2}}^{\frac{1}{2}} \sum_{\mathbf{k}} \frac{1}{|det\mathbf{V}|} S_c(\mathbf{U(f-k)}) \exp\{j2\pi \mathbf{f}^T \mathbf{n}\} \exp\{-j2\pi \mathbf{k}^T \mathbf{n}\} d\mathbf{f} \qquad (3.22)$$

where $\exp\{-j2\pi \mathbf{k}^T \mathbf{n}\} = 1$ for $\mathbf{k}$ an integer valued vector.

Thus, comparing this expression with (3.21), we conclude that

$$S(\mathbf{f}) = \frac{1}{|det\mathbf{V}|} \sum_{\mathbf{k}} S_c(\mathbf{U(f-k)}) \qquad (3.23)$$

or, equivalently,

$$S_p(\mathbf{F}) = \frac{1}{|det\mathbf{V}|} \sum_{\mathbf{k}} S_c(\mathbf{F - Uk}) \qquad (3.24)$$

where the periodicity matrix in the frequency domain $\mathbf{U}$ satisfies

$$\mathbf{U}^T\mathbf{V} = I \qquad (3.25)$$

and $\mathbf{I}$ is the identity matrix. The periodicity matrix can be expressed as $\mathbf{U} = [\mathbf{u}_1|\mathbf{u}_2]$, where $\mathbf{u}_1$ and $\mathbf{u}_2$ are the basis vectors in the 2-D frequency plane.

Note that this formulation includes rectangular sampling as a special case with the matrices $\mathbf{V}$ and $\mathbf{U}$ being diagonal. The replications in the 2-D frequency plane according to (3.24) are depicted in Figure 3.9.

3.4 Sampling on 3-D Structures

The concepts related to 2-D sampling with an arbitrary periodic geometry can be readily extended to sampling time-varying images $s_c(\mathbf{x}, t) = s_c(x_1, x_2, t)$ on 3-D sampling structures. In this section, we first elaborate on 3-D lattices, and the spectrum of 3-D signals sampled on lattices. Some specific non-lattice structures are also introduced. The theory of sampling on lattices and other structures has been generalized to M dimensions elsewhere [Dub 85].

3.4.1 Sampling on a Lattice

We start with the definition of a 3-D lattice. Let $\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3$ be linearly independent vectors in the 3-D Euclidean space $\mathbf{R}^3$. A lattice Λ^3 in $\mathbf{R}^3$ is the set of all linear combinations of $\mathbf{v}_1$, $\mathbf{v}_2$, and $\mathbf{v}_3$ with integer coefficients

$$\Lambda^3 = \{n_1\mathbf{v}_1 + n_2\mathbf{v}_2 + k\mathbf{v}_3 \mid n_1, n_2, k \in \mathbf{Z}\} \qquad (3.26)$$

The set of vectors $\mathbf{v}_1$, $\mathbf{v}_2$, and $\mathbf{v}_3$ is called a basis for Λ^3.

In vector-matrix notation, a lattice can be defined as

$$\Lambda^3 = \{\mathbf{V} \begin{bmatrix} \mathbf{n} \\ k \end{bmatrix} \mid \mathbf{n} \in \mathbf{Z}^2, k \in \mathbf{Z}\} \qquad (3.27)$$

where $\mathbf{V}$ is called a 3×3 sampling matrix defined by

$$\mathbf{V} = [\mathbf{v}_1 \mid \mathbf{v}_2 \mid \mathbf{v}_3] \qquad (3.28)$$

The basis, and thus the sampling matrix? for a given lattice is not unique. In particular, for every sampling matrix $\mathbf{V}$, $\mathbf{V} = \mathbf{EV}$, where $\mathbf{E}$ is an integer matrix with $\det\mathbf{E} = \pm 1$, forms another sampling matrix for Λ^3. However, the quantity $d(\Lambda^3) = |\det\mathbf{V}|$ is unique and denotes the reciprocal of the sampling density.

Then, similar to (3.15) and (3.16), the sampled spatio-temporal signal can be expressed as

$$s(\mathbf{n}, k) = s_c(\mathbf{V} \begin{bmatrix} \mathbf{n} \\ k \end{bmatrix}), (\mathbf{n}, k) = (n_1, n_2, k) \in \mathbf{Z}^3 \qquad (3.29)$$

or as

$$\begin{aligned} s_p(\mathbf{x}, t) &= s_c(\mathbf{x}, t) \sum_{(\mathbf{n},k)\in\mathbf{Z}^3} \delta(\begin{bmatrix} \mathbf{x} \\ t \end{bmatrix} - \mathbf{V} \begin{bmatrix} \mathbf{n} \\ k \end{bmatrix}) \\ &= \sum_{(\mathbf{n},k)\in Z^3} s(\mathbf{n}, k)\delta(\begin{bmatrix} \mathbf{x} \\ t \end{bmatrix} - \mathbf{V} \begin{bmatrix} \mathbf{n} \\ k \end{bmatrix}) \end{aligned} \qquad (3.30)$$

The observant reader may already have noticed that the 2-D sampling structures discussed in Sections 3.2 and 3.3 are also lattices. Hence, Sections 3.2 and 3.3 constitute special cases of the theory presented in this section.

3.4.2 Fourier Transform on a Lattice

Based on (3.19), we can define the spatio-temporal Fourier transform of $s_p(\mathbf{x}, t)$, sampled on Λ^3, as follows:

$$\begin{aligned} S_p(\mathbf{F}) &= \sum_{(\mathbf{x},t)\in\Lambda^3} s_c(\mathbf{x}, t) \exp\left\{-j2\pi \mathbf{F}^T \begin{bmatrix} \mathbf{x} \\ t \end{bmatrix}\right\}, \quad \mathbf{F} = [F_1\ F_2\ F_t]^T \in \mathbf{R}^3 \\ &= \sum_{(\mathbf{n},k)\in\mathbf{Z}^3} s_c(\mathbf{V} \begin{bmatrix} \mathbf{n} \\ k \end{bmatrix}) \exp\left\{-j2\pi \mathbf{F}^T \mathbf{V} \begin{bmatrix} \mathbf{n} \\ k \end{bmatrix}\right\} \end{aligned} \qquad (3.31)$$

In order to quantify some properties of the Fourier transform defined on a lattice, we next define the reciprocal lattice and the unit cell of a lattice. Given a lattice Λ^3, the set of all vectors $\mathbf{r}$ such that $\mathbf{r}^T \begin{bmatrix} \mathbf{x} \\ t \end{bmatrix}$ is an integer for all $(\mathbf{x}, t) \in \Lambda^3$ is called the reciprocal lattice Λ^{3*} of Λ^3. A basis for Λ^{3*} is the set of vectors $\mathbf{u}_1$, $\mathbf{u}_2$, and us determined by

$$\mathbf{u}_i^T \mathbf{v}_j = \delta_{ij}, \quad i, j = 1, 2, 3$$

or, equivalently,

$$\mathbf{U}^T \mathbf{V} = \mathbf{I}_3$$

where Is is a 3 x 3 identity matrix.

The definition of the unit cell of a lattice is not unique. Here we define the Voronoi cell of a lattice as a unit cell. The Voronoi cell, depicted in Figure 3.10, is the set of all points that are closer to the origin than to any other sample point.

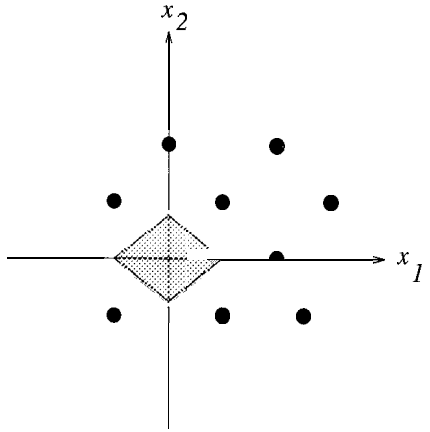


Figure 3.10: The Voronoi cell of a 2-D lattice.

The Fourier transform of a signal sampled on a lattice is a periodic function over $\mathbf{R}^3$ with periodicity lattice Λ^{3*},

$$S_p(\mathbf{F}) = S_p(\mathbf{F} + \mathbf{r}), \quad \mathbf{r} \in \Lambda^{3*}$$

This follows from $\mathbf{r}^T \begin{bmatrix} \mathbf{x} \\ t \end{bmatrix}$ is an integer for (x, t) $\in \Lambda^3$ and $\mathbf{r} \in \Lambda^{3*}$, by definition of the reciprocal lattice. Because of this periodicity, the Fourier transform need only

be specified over one unit cell $\mathcal{P}$ of the reciprocal lattice Λ^{3*}. Thus, the inverse Fourier transform of $S_p(\mathbf{F})$ is given by

$$s_p(\mathbf{x}, t) = d(\Lambda^3) \int_{\mathcal{P}} S_p(\mathbf{F}) \exp\left\{ j2\pi \mathbf{F}^T \begin{bmatrix} \mathbf{x} \\ t \end{bmatrix} \right\} d\mathbf{F}, \quad (\mathbf{x}, t) \in \Lambda^3 \tag{3.32}$$

Note that in terms of the normalized frequency variables, $\mathbf{f} = \mathbf{V}^T \mathbf{F}$, we have

$$S(\mathbf{f}) = S_p(\mathbf{U}\mathbf{f}) = \sum_{(\mathbf{n}, k) \in \mathbf{Z}^3} s(\mathbf{n}, k) \exp\left\{ -j2\pi \mathbf{f}^T \begin{bmatrix} \mathbf{n} \\ k \end{bmatrix} \right\} \tag{3.33}$$

where $\mathbf{U} = \mathbf{V}^{T^{-1}}$, with the fundamental period given by the unit cell $f_1 < |1/2|$, $f_2 < |1/2|$, and $f_t < |1/2|$.

3.4.3 Spectrum of Signals Sampled on a Lattice

In this section, we relate the Fourier transform $S_p(\mathbf{F})$ of the sampled signal to that of the continuous signal. Suppose that $s_c(\mathbf{x}, t) \in L^1(\mathbf{R}^3)$ has the Fourier transform

$$S_c(\mathbf{F}) = \int_{\mathbf{R}^3} s_c(\mathbf{x}, t) \exp\left\{ -j2\pi \mathbf{F}^T \begin{bmatrix} \mathbf{x} \\ t \end{bmatrix} \right\} \mathrm{dx}\, dt, \quad \mathrm{F} \in \mathbf{R}^3 \tag{3.34}$$

with the inverse transform

$$s_c(\mathbf{x}, t) = \int_{\mathbf{R}^3} S_c(\mathbf{F}) \exp\left\{ j2\pi \mathbf{F}^T \begin{bmatrix} \mathbf{x} \\ t \end{bmatrix} \right\} \mathbf{dF}, \quad (\mathbf{x}, t) \in \mathbf{R}^3 \tag{3.35}$$

We substitute (3.35) into (3.30), and express this integral as a sum of integrals over displaced versions of a unit cell P of Λ^{3*} to obtain

$$\begin{aligned} s_p(\mathbf{x}, t) &= \int_{\mathbf{R}^3} S_c(\mathbf{F}) \exp\left\{ j2\pi \mathbf{F}^T \begin{bmatrix} \mathbf{x} \\ t \end{bmatrix} \right\} \mathbf{dF}, \quad \mathbf{x} \in \Lambda^3 \\ &= \sum_{\mathbf{r} \in \Lambda^{3*}} \int_{\mathcal{P}} S_c(\mathbf{F} + \mathbf{r}) \exp\left\{ j2\pi (\mathbf{F} + \mathbf{r})^T \begin{bmatrix} \mathbf{x} \\ t \end{bmatrix} \right\} \mathbf{dF} \end{aligned}$$

Since $\exp(j2\pi \mathbf{r}^T \begin{bmatrix} \mathbf{x} \\ t \end{bmatrix}) = 1$ by the property of the reciprocal lattice, exchanging the order of summation and integration yields

$$s_p(\mathbf{x}, t) = \int_{\mathcal{P}} \left[\sum_{\mathbf{r} \in \Lambda^{3*}} S_c(\mathbf{F} + \mathbf{r}) \right] \exp\left\{ j2\pi \mathbf{F}^T \begin{bmatrix} \mathbf{x} \\ t \end{bmatrix} \right\} \mathbf{dF}, \ (\mathbf{x}, t) \in \Lambda^3 \tag{3.36}$$

Thus, we have

$$\mathrm{S,(F)} = \frac{1}{d(\Lambda^3)} \sum_{\mathbf{r} \in \Lambda^{3*}} S_c(\mathbf{F} + \mathbf{r}) = \frac{1}{d(\Lambda^3)} \sum_{\mathbf{k} \in \mathbf{Z}^3} S_c(\mathbf{F} + \mathbf{U}\mathbf{k}) \tag{3.37}$$

or, alternatively,

$$S(\mathbf{f}) = S_p(\mathbf{F})|_{\mathbf{F}=\mathbf{U}\mathbf{f}} = \frac{1}{d(\Lambda^3)} \sum_{\mathbf{k}\in\mathbf{Z}^3} S_c(\mathbf{U}(\mathbf{f}+\mathbf{k})) \qquad (3.38)$$

where $\mathbf{U}$ is the sampling matrix of the reciprocal lattice Λ^{3*}. As expected, the Fourier transform of the sampled signal is the sum of an infinite number of replicas of the Fourier transform of the continuous signal, shifted according to the reciprocal lattice Λ^{3*}.

Example

This example illustrates sampling of a continuous time-varying image, $s_c(\mathbf{x},t)$, $(\mathbf{x},t) \in \mathbf{R}^3$, using the progressive and the 2:1 line interlaced sampling lattices, shown in Figures 3.4 and 3.5 along with their sampling matrices **V,** respectively.

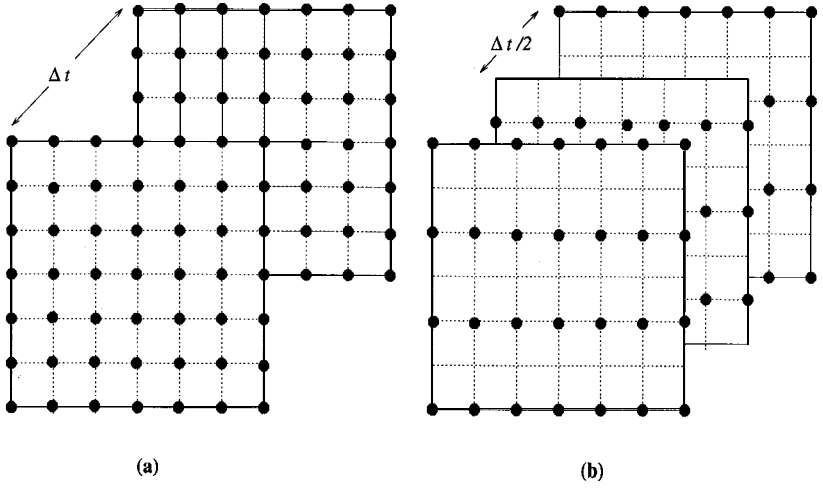


Figure 3.11: Sampling lattices for a) progressive and b) interlaced video.

The locations of the samples of $s_p(\mathbf{x},t)$, $\begin{bmatrix} \mathbf{x} \\ t \end{bmatrix} = \mathbf{V}\begin{bmatrix} \mathbf{n} \\ k \end{bmatrix} \in \Lambda^3$, or $s(\mathbf{n},k)$, $(\mathbf{n},k) \in \mathbf{Z}^3$, for the cases of progressive and interlaced sampling are depicted in Figure 3.11 (a) and (b), respectively. Observe that the reciprocal of the sampling density $d(\Lambda^3) = \Delta x_1 \Delta x_2 \Delta t$ is identical for both lattices. However, the periodicity matrices of the spatio-temporal Fourier transform of the sampled video, indicating the locations of the

replications, are different, and are given by

$$\mathbf{U} = \mathbf{V}^{-1^T} = \begin{bmatrix} \frac{1}{\Delta x_1} & 0 & 0 \\ 0 & \frac{1}{\Delta x_2} & 0 \\ 0 & 0 & \frac{1}{\Delta t} \end{bmatrix}$$

and

$$\mathbf{U} = \mathbf{V}^{-1^T} = \begin{bmatrix} \frac{1}{\Delta x_1} & 0 & 0 \\ 0 & \frac{1}{2\Delta x_2} & 0 \\ 0 & -\frac{1}{\Delta t} & \frac{2}{\Delta t} \end{bmatrix}$$

for the progressive and interlaced cases, respectively.

3.4.4 Other Sampling Structures

In general, the Fourier transform cannot be defined for sampling structures other than lattices; thus, the theory of sampling does not extend to arbitrary sampling structures [Dub 85]. However, an extension is possible for those sampling structures which can be expressed as unions of cosets of a lattice.

Unions of Cosets of a Lattice

Let Λ^3 and Γ^3 be two 3-D lattices. Λ^3 is a sublattice of Γ^3 if every site in Λ^3 is also a site of Γ^3. Then, $d(\Lambda^3)$ is an integer multiple of $d(\Gamma^3)$. The quotient $d(\Lambda^3)/d(\Gamma^3)$ is called the index of Λ^3 in Γ^3, and is denoted by $(\Lambda^3 : \Gamma^3)$. We note that if Λ^3 is a sublattice of Γ^3, then Γ^{3*} is a sublattice of Λ^{3*}.

The set

$$\mathbf{c} + \Lambda^3 = \left\{ \mathbf{c} + \begin{bmatrix} \mathbf{x} \\ t \end{bmatrix} \mid \begin{bmatrix} \mathbf{x} \\ t \end{bmatrix} \in \Lambda^3 \text{ and } \mathbf{c} \in \Gamma^3 \right\} \qquad (3.39)$$

is called a coset of Λ^3 in Γ^3. Thus, a coset is a shifted version of the lattice Λ^3.

The most general form of a sampling structure Ψ^3 that we can analyze for spatio-temporal sampling is the union of P cosets of a sublattice Λ^3 in a lattice Γ^3, defined by

$$\Psi^3 = \bigcup_{i=1}^{P} (\mathbf{c}_i + \Lambda^3) \qquad (3.40)$$

where $\mathbf{c}_1, \ldots, \mathbf{c}_P$ is a set of vectors in Γ^3 such that

$$\mathbf{c}_i - \mathbf{c}_j \notin \Lambda^3 \text{ for } i \neq j$$

An example of such a sampling structure is depicted in Figure 3.12. Note that Ψ^3 becomes a lattice if we take $\Lambda^3 = \Gamma^3$ and $P = 1$.

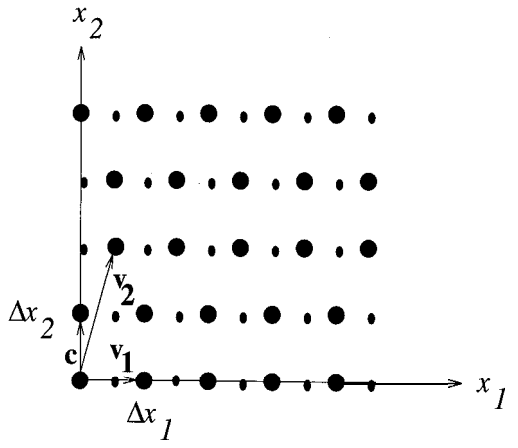


Figure 3.12: Union of cosets of a lattice [Dub 85] (©1985 IEEE).

Spectrum of Signals on Unions of Cosets of a Lattice

As stated, the Fourier transform is, in general, not defined for a sampling structure other than a lattice. However, for the special case of the union of cosets of a sublattice Λ^3 in Γ^3, we can assume that the signal is defined over the parent lattice Γ^3 with certain sample values set to zero. Then,

$$S_p(\mathbf{F}) = \sum_{i=1}^{P} \sum_{\mathbf{x}\in\Lambda^3} s_p(\mathbf{c}_i + \begin{bmatrix} \mathbf{x} \\ t \end{bmatrix}) \exp\left\{-j2\pi\mathbf{F}^T(\mathbf{c}_i + \begin{bmatrix} \mathbf{x} \\ t \end{bmatrix})\right\}$$

$$= \sum_{i=1}^{P} \exp\left\{-j2\pi\mathbf{F}^T\mathbf{c}_i\right\} \sum_{(\mathbf{x},t)\in\Lambda^3} s_p(\mathbf{c}_i + \begin{bmatrix} \mathbf{x} \\ t \end{bmatrix}) \exp\left\{-j2\pi\mathbf{F}^T \begin{bmatrix} \mathbf{x} \\ t \end{bmatrix}\right\} \quad (3.41)$$

The periodicity of this Fourier transform is determined by the reciprocal lattice Γ^{3*}.

It can be shown that the spectrum of a signal sampled on a structure which is in the form of the union of cosets of a lattice is given by [Dub 85]

$$S_p(\mathbf{F}) = \frac{1}{d(\Lambda^3)} \sum_{\mathbf{r}\in\Lambda^{3*}} g(\mathbf{r}) S_c(\mathbf{F}+\mathbf{r}) \quad (3.42)$$

where $\mathbf{V}$ is the sampling matrix of Λ^3, and the function

$$g(\mathbf{r}) = \sum_{i=1}^{P} \exp(j2\pi\mathbf{r}^T\mathbf{c}_i) \quad (3.43)$$

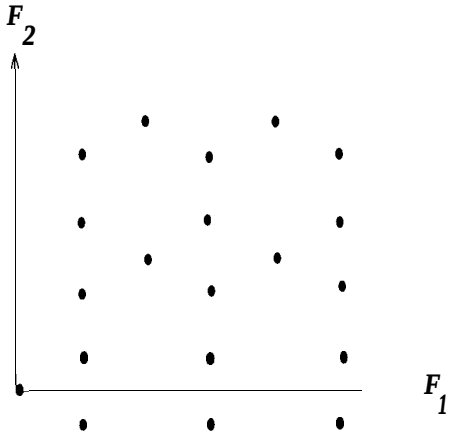


Figure 3.13: Reciprocal structure Ψ^{3*} [Dub 85] (©1985 IEEE).

is constant over cosets of Γ^{3*} in Λ^{3*}, and may be equal to zero for some of these cosets, so the corresponding shifted versions of the basic spectrum are not present.

Example The line-quincunx structure shown in Figure 3.7, which occasionally finds use in practical systems, is in the form of a union of cosets of a lattice. A 2-D version of this lattice was illustrated in Figure 3.12 where $P = 2$. The reciprocal structure of the sampling structure Ψ^3 shown in Figure 3.12 is depicted in Fig. 3.13.

3.5 Reconstruction from Samples

Digital video is usually converted back to analog video for display purposes. Furthermore, various digital video systems have different spatio-temporal resolution requirements which necessitate sampling structure conversion. The sampling structure conversion problem, which is treated in the next chapter, can alternatively be posed as the reconstruction of the underlying continuous spatio-temporal video, followed by its resampling on the desired spatio-temporal lattice. Thus, in the remainder of this chapter, we address the theory of reconstruction of a continuous video signal from its samples.

3.5.1 Reconstruction from Rectangular Samples

Reconstruction of a continuous signal from its samples is an interpolation problem. In ideal bandlimited interpolation, the highest frequency that can be represented in

the analog signal without aliasing, according to the Nyquist sampling theorem, is equal to one-half of the sampling frequency. Then, a continuous image, $s_r(x_1, x_2)$, can be reconstructed from its samples taken on a 2-D rectangular grid by ideal low-pass filtering as follows:

$$S_r(F_1, F_2) = \begin{cases} \Delta x_1 \Delta x_2 S(F_1 \Delta x_1, F_2 \Delta x_2) & \text{for } |F_1| < \frac{1}{2\Delta x_1} \text{ and } |F_2| < \frac{1}{2\Delta x_2} \\ 0 & \text{otherwise.} \end{cases} \quad (3.44)$$

The support of the ideal reconstruction filter in the frequency domain is illustrated in Figure 3.14.

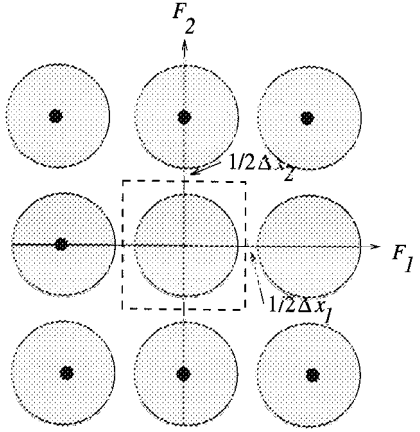


Figure 3.14: Reconstruction filter.

The reconstruction filter given by (3.44) is sometimes referred to as the "ideal bandlimited interpolation filter," since the reconstructed image would be identical to the original continuous image, that is

$$s_r(x_1, x_2) = s_c(x_1, x_2) \quad (3.45)$$

provided that the original continuous image was bandlimited, and Δx_1 and Δx_2 were chosen according to the Nyquist criterion.

The ideal bandlimited interpolation filtering can be expressed in the spatial domain by taking the inverse Fourier transform of both sides of (3.44)

$$s_r(x_1, x_2) = \int_{\frac{-1}{2\Delta x_1}}^{\frac{1}{2\Delta x_1}} \int_{\frac{-1}{2\Delta x_2}}^{\frac{1}{2\Delta x_2}} \Delta x_1 \Delta x_2 \, S(F_1 \Delta x_1, F_2 \Delta x_2) \exp\{j2\pi(F_1 x_1 + F_2 x_2)\} \, dF_1 dF_2$$

Substituting the definition of $S(F_1 \Delta x_1, F_2 \Delta x_2)$ into this expression, and rearranging the terms, we obtain

$$\begin{aligned} s_r(x_1, x_2) = \Delta x_1 \Delta x_2 \sum_{n_1} \sum_{n_2} s(n_1, n_2) \int_{\frac{-1}{2\Delta x_1}}^{\frac{1}{2\Delta x_1}} \int_{\frac{-1}{2\Delta x_2}}^{\frac{1}{2\Delta x_2}} & \exp\{-j2\pi(F_1 \Delta x_1 n_1 + F_2 \Delta x_2 n_2)\} \\ & \exp\{j2\pi(F_1 x_1 + F_2 x_2)\} dF_1 dF_2 \\ = \Delta x_1 \Delta x_2 \sum_{n_1} \sum_{n_2} & s(n_1, n_2) h(x_1 - n_1 \Delta x_1, x_2 - n_2 \Delta x_2) \end{aligned} \quad (3.46)$$

where $h(x_1, x_2)$ denotes the impulse response of the ideal interpolation filter for the case of rectangular sampling, given by

$$h(x_1, x_2) = \frac{\sin\ (\&xi)\ \sin\left(\frac{\pi}{\Delta x_2} x_2\right)}{\frac{\pi}{\Delta x_1} x_1 \qquad \frac{\pi}{\Delta x_2} x_2} \quad (3.47)$$

3.5.2 Reconstruction from Samples on a Lattice

Similar to the case of rectangular sampling, the reconstructed time-varying image $s_r(\mathbf{x}, t)$ can be obtained through the ideal low-pass filtering operation

$$S_r(\mathbf{F}) = \begin{cases} |det \mathbf{V}| S(\mathbf{V}^T \mathbf{F}) & \text{for } F \in \mathcal{P} \\ 0 & \text{otherwise.} \end{cases}$$

Here, the passband of the ideal low-pass filter is determined by the unit cell $\mathcal{P}$ of the reciprocal sampling lattice.

Taking the inverse Fourier transform, we have the reconstructed time-varying image

$$\begin{aligned} s_r(\mathbf{x}, t) &= \sum_{(\mathbf{n}, k) \in \mathbf{Z}^3} s(\mathbf{n}, k) \, h\left(\begin{bmatrix} \mathbf{x} \\ t \end{bmatrix} - \mathbf{V} \begin{bmatrix} \mathbf{n} \\ k \end{bmatrix} \right) \\ &= \sum_{(\mathbf{z}, \tau) \in \Lambda^3} s_p(\mathbf{z}, \tau) \, h\left(\begin{bmatrix} \mathbf{x} \\ t \end{bmatrix} - \begin{bmatrix} \mathbf{z} \\ \tau \end{bmatrix} \right) \end{aligned} \quad (3.48)$$

where

$$h(\mathbf{x}, t) = |det \mathbf{V}| \int_{\mathcal{P}} \exp\left\{ j2\pi \mathbf{F}^T \begin{bmatrix} \mathbf{x} \\ t \end{bmatrix} \right\} d\mathbf{F} \quad (3.49)$$

is the impulse response of the ideal bandlimited spatio-temporal interpolation filter for the sampling structure used. Unlike the case of rectangular sampling, this integral, in general, cannot be reduced to a simple closed-form expression. As expected, exact reconstruction of a continuous signal from its samples on a lattice Λ^3 is possible if the signal spectrum is confined to a unit cell $\mathcal{P}$ of the reciprocal lattice.

3.6 Exercises

1. Derive (3.11) using (3.9).

2. The expression (3.11) assumes impulse sampling; that is, the sampling aperture has no physical size. A practical camera has a finite aperture modeled by the impulse response $h_a(x_1, x_2)$. How would you incorporate the effect of the finite aperture size into (3.11)?

3. Suppose a camera samples with 20-micron intervals in both the horizontal and vertical directions. What is the highest spatial frequency in the sampled image that can be represented with less than 3 dB attenuation if a) $h_a(x_1, x_2)$ is a 2-D Dirac delta function, and b) $h_a(x_1, x_2)$ is a uniform circle with diameter 20 microns?

4. Strictly speaking, images with sharp spatial edges are not bandlimited. Discuss how you would digitize an image that is not bandlimited.

5. Find the locations of the spectral replications for each of the 3-D sampling lattices depicted in Figures 3.4 through 3.7.

6. Evaluate the impulse response (3.49) if $\mathcal{P}$ is the unit sphere.

Bibliography

[Dub 85] E. Dubois, "The sampling and reconstruction of time-varying imagery with application in video systems," *Proc. IEEE,* vol. 73, no. 4, pp. 502–522, Apr. 1985.

[Dud 84] D. E. Dudgeon and R. M. Mersereau, *Multidimensional Digital Signal Processing,* Englewood Cliffs, NJ: Prentice Hall, 1984.

[Jai 89] A. K. Jain, Fundamentals *of Digital* Image *Processing,* Englewood Cliffs, NJ: Prentice Hall, 1989.

[Lim 90] J. S. Lim, Two-Dimensional *Signal* and *Image Processing,* Englewood Cliffs, NJ: Prentice Hall, 1990.

[Opp 89] A. V. Oppenheim and R. W. Schafer, *Discrete-Time* Signal Processing, Englewood Cliffs, NJ: Prentice Hall, 1989.

[Rah 92] M. A. Rahgozar and J. P. Allebach, "A general theory of time-sequential sampling," Signal *Proc.*, vol. 28, no. 3, pp. 253-270, Sep. 1992.

Chapter 4

SAMPLING STRUCTURE CONVERSION

Various digital video systems, ranging from all-digital high-definition TV to videophone, have different spatio-temporal resolution requirements leading to the emergence of different format standards to store, transmit, and display digital video. The task of converting digital video from one format to another is referred to as the standards conversion problem. Effective standards conversion methods enable exchange of information among various digital video systems, employing different format standards, to ensure their interoperability.

Standards conversion is a 3-D sampling structure conversion problem, that is, a spatio-temporal interpolation/decimation problem. This chapter treats sampling structure conversion as a multidimensional digital signal processing problem, including the characterization of sampling structure conversion in the 3-D frequency domain and filter design for sampling structure conversion, without attempting to take advantage of the temporal redundancy present in the video signals. The theory of motion-compensated filtering and practical standards conversion algorithms specifically designed for digital video, which implicitly or explicitly use interframe motion information, will be presented in Chapters 13 and 16, respectively.

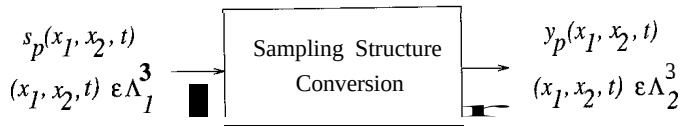


Figure 4.1: Block diagram for sampling structure conversion.

Chapter 5

OPTICAL FLOW METHODS

Motion estimation, which may refer to image-plane motion (2-D motion) or object-motion (3-D motion) estimation, is one of the fundamental problems in digital video processing. Indeed, it has been the subject of much research effort [Hua 81, Hua 83, Agg 88, Sin 91, Fle 92, Sez 93]. This chapter has two goals: it provides a general introduction to the 2-D motion estimation problem; it also discusses specific algorithms based on the optical flow equation. Various other nonparametric approaches for 2-D motion estimation will be covered in Chapters 6-8. In Section 5.1, we emphasize the distinction between 2-D motion and apparent motion (optical flow or correspondence). The 2-D motion estimation problem is then formulated as an ill-posed problem in Section 5.2, and a brief overview of a priori motion field models is presented. Finally, we discuss a number of optical flow estimation methods based on the optical flow equation (also known as the differential methods) in Section 5.3. Besides being an integral component of motion-compensated filtering and compression, 2-D motion estimation is often the first step towards 3-D motion analysis, which will be studied in Chapters 9-12.

5.1 2-D Motion vs. Apparent Motion

Because time-varying images are 2-D projections of 3-D scenes, as described in Chapter 2, 2-D motion refers to the projection of the 3-D motion onto the image plane. We wish to estimate the 2-D motion (instantaneous velocity or displacement) field from time-varying images sampled on a lattice Λ^3. However, 2-D velocity or displacement fields may not always be observable for several reasons, which are cited below. Instead, what we observe is the so-called "apparent" motion (optical flow or correspondence) field. It is the aim of this section to clarify the distinction between 2-D velocity and optical flow, and 2-D displacement and correspondence fields, respectively.

5.1.1 2-D Motion

2-D motion, also called "projected motion," refers to the perspective or the orthographic projection of 3-D motion into the image plane. 3-D motion can be characterized in terms of either 3-D instantaneous velocity (hereafter velocity) or 3-D displacement of the object points. Expressions for the projections of the 3-D displacement and velocity vectors, under the assumption of rigid motion, into the image plane are derived in Chapters 9 and 10, respectively.

The concept of a 2-D displacement vector is illustrated in Figure 5.1. Let the object point **P** at time t move to **P'** at time t'. The perspective projection of the points **P** and **P'** to the image plane gives the respective image points p and p'. Figure 5.2 depicts a 2-D view of the motion of the image point p at time t to p' at time t' as the perspective projection of the 3-D motion of the corresponding object points. Note that because of the projection operation, all 3-D displacement vectors whose tips lie on the dotted line would give the same 2-D displacement vector.

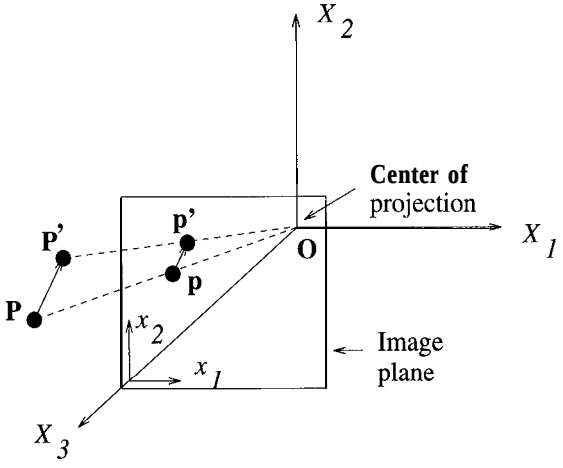


Figure 5.1: Three-dimensional versus two-dimensional.motion.

The projected displacement between the times t and $t' = t + \ell\Delta t$, where ℓ is an integer and At is the temporal sampling interval, can be defined for all $(\mathbf{x}, t) \in \mathbf{R}^3$, resulting in a real-valued 2-D displacement vector function $\mathbf{d}_c(\mathbf{x}, t; \ell\Delta t)$ of the continuous spatio-temporal variables. The 2-D displacement vector field refers to a sampled representation of this function, given by

$$\mathbf{d}_p(\mathbf{x}, t; \ell\Delta t) = \mathbf{d}_c(\mathbf{x}, t; \ell\Delta t), \quad (\mathbf{x}, t) \in \Lambda^3, \tag{5.1}$$

or, equivalently,

$$\mathbf{d}(\mathbf{n}, k; \ell) = \mathbf{d}_p(\mathbf{x}, t; \ell\Delta t)|_{[x_1\ x_2\ t]^T = \mathbf{V}[n_1\ n_2\ k]^T}, \quad (\mathbf{n}, k) \in \mathbf{Z}^3, \tag{5.2}$$

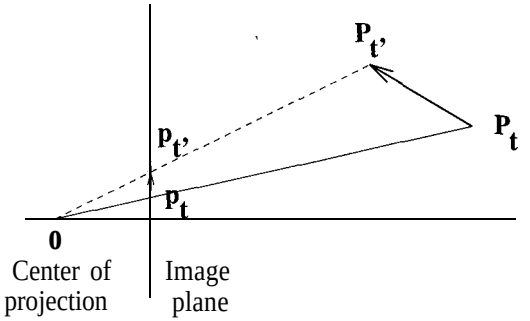


Figure 5.2: The projected motion.

where **V** is the sampling matrix of the lattice Λ^3. Thus, a 2-D displacement field is a collection of 2-D displacement vectors, $\mathbf{d}(\mathbf{x}, \mathbf{t}; \ell\Delta t)$, where $(x, t) \in \Lambda^3$.

The projected velocity function $\mathbf{v}_c(\mathbf{x}, t)$ at time t, and the 2-D velocity vector field $\mathbf{v}_p(\mathbf{x}, t) = v(\mathbf{n}, k)$, for $[x_1\, x_2\, t]^T = \mathbf{V}[n_1, n_2, k]^T \in \Lambda^3$ and $(\mathbf{n}, k) \in \mathbf{Z}^3$ can be similarly defined in terms of the 3-D instantaneous velocity (Xi, $\dot{X}_2$, $\dot{X}_3$), where the dot denotes a time derivative.

5.1.2 Correspondence and Optical Flow

The displacement of the image-plane coordinates x from time t to t', based on the variations of $s_c(\mathbf{x}, t)$, is called a correspondence vector. An optical flow vector is defined as the temporal rate of change of the image-plane coordinates, $(v_1, v_2) = (dx_1/dt, dx_2/dt)$, at a particular point $(\mathbf{x}, t) \in \mathbf{R}^3$ as determined by the spatio-temporal variations of the intensity pattern $s_c(\mathbf{x}, t)$. That is, it corresponds to the instantaneous pixel velocity vector. (Theoretically, the optical flow and correspondence vectors are identical in the limit $\mathbf{At} = t'$ -t goes to zero, should we have access to the continuous video.) In practice, we define the correspondence (optical flow) field as a vector field of pixel displacements (velocities) based on the observable variations of the 2-D image intensity pattern on a spatio-temporal lattice Λ^3. The correspondence field and optical flow field are also known as the "apparent 2-D displacement" field and "apparent 2-D velocity" field, respectively.

The correspondence (optical flow) field is, in general, different from the 2-D displacement (2-D velocity) field due to [Ver 89]:

. Lack of sufficient spatial image gradient: There must be sufficient gray-level (color) variation within the moving region for the actual motion to be observable. An example of an unobservable motion is shown in Figure 5.3, where a circle with uniform intensity rotates about its center. This motion generates no optical flow, and thus is unobservable.

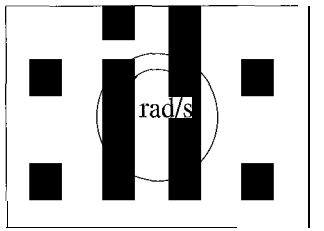


Figure 5.3: All projected motion does not generate optical flow.

. Changes in external illumination: An observable optical flow may not always correspond to an actual motion. For example, if the external illumination varies from frame to frame, as shown in Figure 5.4, then an optical flow will be observed even though there is no motion. Therefore, changes in the external illumination impair the estimation of the actual 2-D motion field.

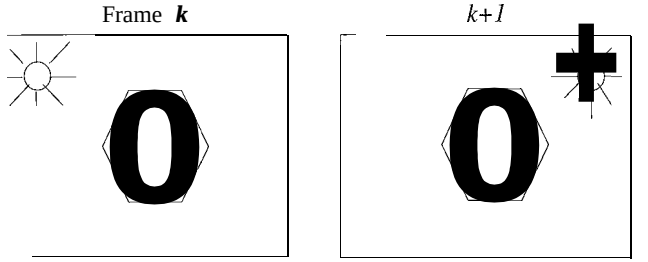


Figure 5.4: All optical flow does not correspond to projected motion.

In some cases, the shading may vary from frame to frame even if there is no change in the external illumination. For example, if an object rotates its surface normal changes, which results in a change in the shading. This change in shading may cause the intensity of the pixels along a motion trajectory to vary, which needs to be taken into account for 2-D motion estimation.

In conclusion the 2-D displacement and velocity fields are projections of the respective 3-D fields **into** the image plane, whereas the correspondence and optical flow fields are the velocity and displacement functions perceived from the time-varying image intensity pattern. Since we can only observe optical flow and correspondence fields, we assume they are the same as the 2-D motion field in the remainder of this book.

5.2 2-D Motion Estimation

In this section, we first state the pixel correspondence and optical flow estimation problems. We then discuss the ill-posed nature of these problems, and introduce some a priori 2-D motion field models.

The 2-D motion estimation problem can be posed as either: i) the estimation of image-plane correspondence vectors $\mathbf{d}(\mathbf{x}, t; \ell\Delta t) = [d_1(\mathbf{x}, t; \ell\Delta t)\ d_2(\mathbf{x}, t; \ell\Delta t)]^T$ between the times t and $t + \ell\Delta t$, for all $(\mathbf{x}, t) \in \Lambda^3$ and ℓ is an integer, or
ii) the estimation of the optical flow vectors $\mathbf{v}(\mathbf{x}, t) = [v_1(\mathbf{x}, t)\ v_2(\mathbf{x}, t)]^T$ for all $(\mathbf{x}, t) \in \Lambda^3$. Observe that the subscript "p" has been dropped for notational simplicity. The correspondence and optical flow vectors usually vary from pixel to pixel (space-varying motion), e.g., due to rotation of objects in the scene, and as a function of time, e.g., due to acceleration of objects.

The Correspondence Problem: The correspondence problem can be set up as a *forward* or *backward* motion estimation problem, depending on whether the motion vector is defined from time t to $t+\ell\Delta t$ or from t to $t-\ell\Delta t$, as depicted in Figure 5.5.

Forward Estimation: Given the spatio-temporal samples $s_p(\mathbf{x}, t)$ at times t and $t + \ell\Delta t$, which are related by

$$s_p(x_1, x_2, t) = s_c(x_1 + d_1(\mathbf{x}, t; \ell\Delta t), x_2 + d_2(\mathbf{x}, t; \ell\Delta t), t + \ell\Delta t) \tag{5.3}$$

or, equivalently,

$$s_k(x_1, x_2) = s_{k+\ell}(x_1 + d_1(\mathbf{x}), x_2 + d_2(\mathbf{x})), \quad \text{such that } t = k\Delta t$$

find the real-valued correspondence vector $\mathrm{d(x)} = [d_1(\mathbf{x})\ d_2(\mathbf{x})]^T$, where the temporal arguments of d(x) are dropped.

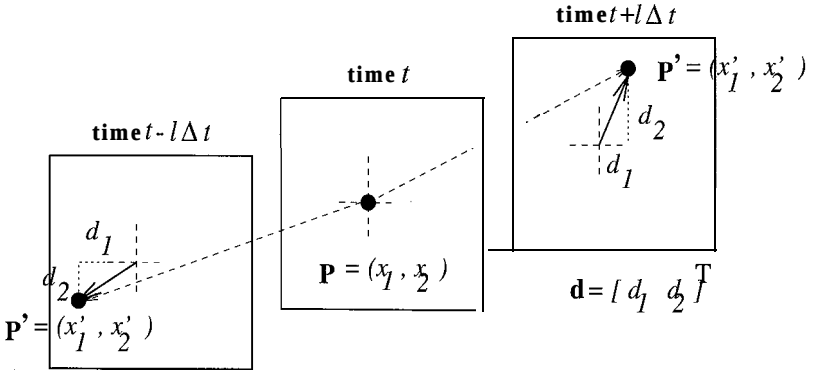


Figure 5.5: Forward and backward correspondence estimation

Backward Estimation: If we define the correspondence vectors from time t to $t - \ell\Delta t$ then, the 2-D motion model becomes

$$s_k(x_1, x_2) = s_{k-\ell}(x_1 + d_1(\mathbf{x}), x_2 + d_2(\mathbf{x})), \quad \text{such that } t = k\Delta t$$

Alternately, the motion vector can be defined from time $t - \ell\Delta t$ to t. Then we have

$$s_k(x_1, x_2) = s_{k-\ell}(x_1 - d_1(\mathbf{x}), x_2 - d_2(\mathbf{x})), \quad \text{such that } t = k\Delta t$$

Although we discuss both types of motion estimation in this book, backward motion estimation is more convenient for forward motion compensation, which is commonly employed in predictive video compression. Observe that because $\mathbf{x} \pm \mathbf{d}(\mathbf{x})$ generally does not correspond to a lattice site, the right-hand sides of the expressions are given in terms of the continuous video $s_c(x_1, x_2, t)$, which is not available. Hence, most correspondence estimation methods incorporate some interpolation scheme. The correspondence problem also arises in stereo disparity estimation (Chapter 12), where we have a left-right pair instead of a temporal pair of images.

Image *Registration:* The registration problem is a special case of the correspondence problem, where the two frames are globally shifted with respect to each other, for example, multiple exposures of a static scene with a translating camera.

Optical Flow Estimation: Given the samples $s_p(x_1, x_2, t)$ on a 3-D lattice Λ^3, determine the 2-D velocity $\mathbf{v}(\mathbf{x}, t)$ for all $(\mathbf{x}, t) \in \Lambda^3$. Of course, estimation of optical flow and correspondence vectors from two frames are equivalent, with $\mathbf{d}(\mathbf{x}, t; \ell\Delta t) = \mathbf{v}(\mathbf{x}, t)\ell\Delta t$, assuming that the velocity remains constant during each time interval $\ell\Delta t$. Note that one needs to consider more than two frames at a time to estimate optical flow in the presence of acceleration.

2-D motion estimation, stated as either a correspondence or optical flow estimation problem, based only on two frames, is an "ill-posed" problem in the absence of any additional assumptions about the nature of the motion. A problem is called ill-posed if a unique solution does not exist, and/or solution(s) do(es) not continuously depend on the data [Ber 88]. 2-D motion estimation suffers from all of the existence, uniqueness, and continuity problems:

- Existence of a solution: No correspondence can be established for covered/uncovered background points. This is known as the *occlusion* problem.
- Uniqueness of the solution: If the components of the displacement (or velocity) at each pixel are treated as independent variables, then the number of unknowns is twice the number of observations (the elements of the frame difference). This leads to the so-called "aperture" problem.
- Continuity of the solution: Motion estimation is highly sensitive to the presence of observation noise in video images. A small amount of noise may result in a large deviation in the motion estimates.

The occlusion and aperture problems are described in detail in the following.

5.2.1 The Occlusion Problem

Occlusion refers to the covering/uncovering of a surface due to 3-D rotation and translation of an object which occupies only part of the field of view. The covered and uncovered background concepts are illustrated in Figure 5.6, where the object indicated by the solid lines translates in the x_1 direction from time t to t'. Let the index of the frames at time t and t' be k and $k + 1$, respectively. The dotted region in the frame k indicates the background to be covered in frame $k + 1$. Thus, it is not possible to find a correspondence for these pixels in frame $k + 1$. The dotted region in frame $k + 1$ indicates the background uncovered by the motion of the object. There is no correspondence for these pixels in frame k.

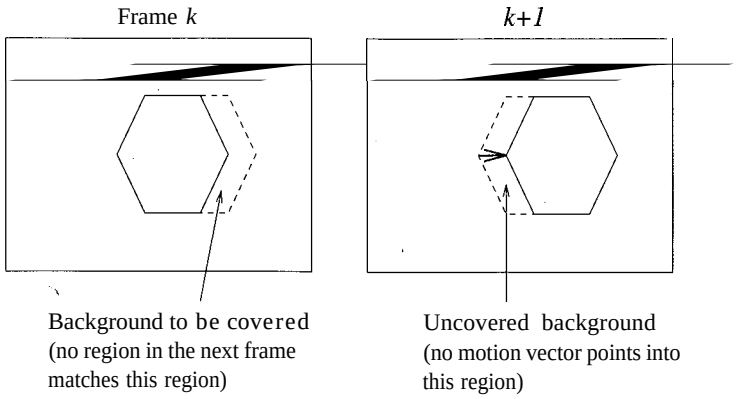


Figure 5.6: The covered/uncovered background problem.

5.2.2 The Aperture Problem

The aperture problem is a restatement of the fact that the solution to the 2-D motion estimation problem is not unique. If motion vectors at each pixel are considered as independent variables, then there are twice as many unknowns as there are equations, given by (5.3). The number of equations is equal to the number of pixels in the image, but for each pixel the motion vector has two components.

Theoretical analysis, which will be given in the next section, indicates that we can only determine motion that is orthogonal to the spatial image gradient, called the normal flow, at any pixel. The aperture problem is illustrated in Figure 5.7. Suppose we have a corner of an object moving in the x_2 direction (upward). If we estimate the motion based on a local window, indicated by Aperture 1, then it is not possible to determine whether the image moves upward or perpendicular to the edge. The motion in the direction perpendicular to the edge is called the normal flow.

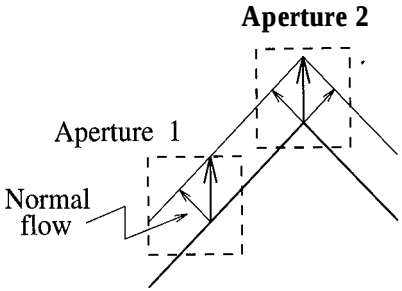


Figure 5.7: The aperture problem.

However, if we observe Aperture 2, then it is possible to estimate the correct motion, since the image has gradient in two perpendicular directions within this aperture. Thus, it is possible to overcome the aperture problem by estimating the motion based on a block of pixels that contain sufficient gray-level variation. Of course, implicit here is the assumption that all these pixels translate by the same motion vector. A less restrictive approach would be to represent the variation of the motion vectors from pixel to pixel by some parametric or nonparametric 2-D motion field models.

5.2.3 Two-Dimensional Motion Field Models

Because of the ill-posed nature of the problem, motion estimation algorithms need additional assumptions (models) about the structure of the 2-D motion field. We provide a brief overview of these models in the following.

Parametric Models

Parametric models aim to describe the orthographic or perspective projection of 3-D motion (displacement or velocity) of a surface into the image plane. In general, parametric 2-D motion field models depend on a representation of the 3-D surface. For example, a 2-D motion field resulting from 3-D rigid motion of a planar surface under orthographic projection can be described by a 6-parameter affine model, while under perspective projection it can be described by an 8-parameter nonlinear model [Ana 93]. There also exist more complicated models for quadratic surfaces [Agg 88]. We elaborate on these models in Chapters 9-12, where we discuss 3-D motion estimation.

A subclass of parametric models are the so-called quasi-parametric models, which treat the depth of each 3-D point as an independent unknown. Then the six 3-D motion parameters constrain the local image flow vector to lie along a spe-

cific line, while knowledge of the local depth value is required to determine the exact value of the motion vector [Ana 93]. These models may serve as constraints to regulate the estimation of the 2-D motion vectors, which lead to simultaneous 2-D and 3-D motion estimation formulations (see Chapter 11).

Nonparametric Models

The main drawback of the parametric models is that they are only applicable in case of 3-D rigid motion. Alternatively, nonparametric uniformity (smoothness) constraints can be imposed on the 2-D motion field without employing 3-D rigid motion models. The nonparametric constraints can be classified as deterministic versus stochastic smoothness models. The following is a brief preview of the nonparametric approaches covered in this book.

. Optical *flow* equation *(OFE) bused methods:* Methods based on the OFE, studied in the rest of this chapter, attempt to provide an estimate of the optical flow field in terms of spatio-temporal image intensity gradients. With monochromatic images, the OFE needs to be used in conjunction with an appropriate spatio-temporal smoothness constraint, which requires that the displacement vector vary slowly over a neighborhood. With color images, the OFE can be imposed at each color band separately, which could possibly constrain the displacement vector in three different directions [Oht 90]. However, in most cases an appropriate smoothness constraint is still needed to obtain satisfactory results. Global smoothness constraints cause inaccurate motion estimation at the occlusion boundaries. More advanced directional smoothness constraints allow sudden discontinuities in the motion field.

- *Block* **motion** *model:* It is assumed that the image is composed of moving blocks. We discuss two approaches to determining the displacement of blocks from frame to frame: the phase-correlation and block-matching methods. In the phase-correlation approach, the linear term of the Fourier phase difference between two consecutive frames determines the motion estimate. Block matching searches for the location of the best-matching block of a fixed size in the next (and/or previous) frame(s) based on a distance criterion. The basic form of both methods apply only to translatory motion; however, generalized block matching can incorporate other spatial transformations. Block-based motion estimation is covered in Chapter 6.

. *Pel-recursive methods:* Pel-recursive methods are predictor-corrector type displacement estimators. The prediction can be taken as the value of the motion estimate at the previous pixel location or as a linear combination of motion estimates in a neighborhood of the current pixel. The update is based on gradient-based minimization of the displaced frame difference (DFD) at that pixel. The prediction step is generally considered as an implicit smoothness constraint. Extension of this approach to block-based estimation results in the so-called Wiener-type estimation strategies. Pel-recursive methods are presented in Chapter 7.

- *Bayesian Methods:* Bayesian methods utilize probabilistic smoothness constraints, usually in the form of a Gibbs random field to estimate the displacement field. Their main drawback is the extensive amount of computation that is required. A maximum a *posteriori* probability estimation method is developed in Chapter 8.

5.3 Methods Using the Optical Flow Equation

In this section, we first derive the optical flow equation (OFE). Then optical flow estimation methods using the OFE are discussed.

5.3.1 The Optical Flow Equation

Let $s_c(x_1, x_2, t)$ denote the continuous space-time intensity distribution. If the intensity remains constant along a motion trajectory, we have

$$\frac{ds_c(x_1, x_2, t)}{dt} = 0 \tag{5.4}$$

where x_1 and x_2 varies by t according to the motion trajectory. Equation (5.4) is a total derivative expression and denotes the rate of change of intensity along the motion trajectory. Using the chain rule of differentiation, it can be expressed as

$$\frac{\partial s_c(\mathbf{x};t)}{\partial x_1} v_1(\mathbf{x},t) + \frac{\partial s_c(\mathbf{x};t)}{\partial x_2} v_2(\mathbf{x},t) + \frac{\partial s_c(\mathbf{x};t)}{\partial t} = 0 \tag{5.5}$$

where $v_1(\mathbf{x},t) = dx_1/dt$ and $v_2(\mathbf{x},t) = dx_2/dt$ denote the components of the coordinate velocity vector in terms of the continuous spatial coordinates. The expression (5.5) is known as the optical flow equation or the optical flow constraint.

It can alternatively be expressed as

$$\langle \nabla s_c(\mathbf{x};t), \mathbf{v}(\mathbf{x},t) \rangle + \frac{\partial s_c(\mathbf{x};t)}{\partial t} = 0 \tag{5.6}$$

where $\nabla s_c(\mathbf{x};t) \triangleq [\frac{\partial s_c(\mathbf{x};t)}{\partial x_1} \; \frac{\partial s_c(\mathbf{x};t)}{\partial x_2}]$ and $\langle \cdot, \cdot \rangle$ denotes vector inner product.

Naturally, the OFE (5.5) is not sufficient to uniquely specify the 2-D velocity (flow) field just like (5.3). The OFE yields one scalar equation in two unknowns, $v_1(\mathbf{x},t)$ and $v_2(\mathbf{x},t)$, at each site (x, t). Inspection of (5.6) reveals that we can only estimate the component of the flow vector that is in the direction of the spatial image gradient $\frac{\nabla s_c(\mathbf{x};t)}{\|\nabla s_c(\mathbf{x};t)\|}$, called the normal flow $v_\perp(\mathbf{x},t)$, because the component that is orthogonal to the spatial image gradient disappears under the dot product.

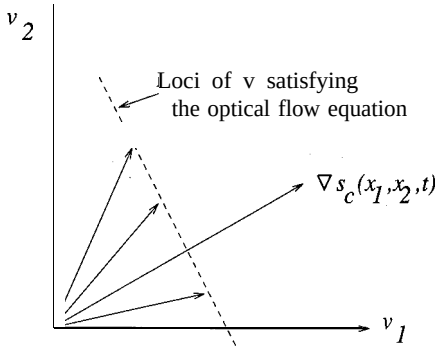


Figure 5.8: The normal flow.

This is illustrated in Figure 5.8, where all vectors whose tips lie on the dotted line satisfy (5.6). The normal flow at each site can be computed from (5.6) as

$$v_\perp(\mathbf{x},t) = \frac{\frac{\partial s_c(\mathbf{x};t)}{\partial t}}{||\nabla s_c(\mathbf{x};t)||} \tag{5.7}$$

Thus, the OFE (5.5) imposes a constraint on the component of the flow vector that is in the direction of the spatial gradient of the image intensity at each site (pixel), which is consistent with the aperture problem. Observe that the OFE approach requires that first, the spatio-temporal image intensity be differentiable, and second, the partial derivatives of the intensity be available. In practice, optical flow estimation from two views can be shown to be equivalent to correspondence estimation under certain assumptions (see Exercise 2). In the following we present several approaches to estimate optical flow from estimates of normal flow.

5.3.2 Second-Order Differential Methods

In search of another constraint to determine both components of the flow vector at each pixel, several researchers [Nag 87, Ura 88] suggested the conservation of the spatial image gradient, $\nabla s_c(\mathbf{x};\, t)$, stated by

$$\frac{d\,\nabla s_c(\mathbf{x};\,t)}{dt} = 0 \tag{5.8}$$

An estimate of the flow field, which is obtained from (5.8), is given by

$$\begin{bmatrix} \hat{v}_1(\mathbf{x};t) \\ \hat{v}_2(\mathbf{x};t) \end{bmatrix} = \begin{bmatrix} \frac{\partial^2 s_c(\mathbf{X},t)}{\partial x_1^2} & \frac{\partial^2 s_c(\mathbf{X},t)}{\partial x_2 x_1} \\ \frac{\partial^2 s_c(\mathbf{X},t)}{\partial x_1 x_2} & \frac{\partial^2 s_c(\mathbf{X},t)}{\partial x_2^2} \end{bmatrix}^{-1} \begin{bmatrix} -\frac{\partial^2 s_c(\mathbf{X},t)}{\partial t \partial x_1} \\ -\frac{\partial^2 s_c(\mathbf{X},t)}{\partial t \partial x_2} \end{bmatrix} \tag{5.9}$$

However, the constraint (5.8) does not allow for some common motion such as rotation and zooming (see Exercise 6). Further, second-order partials cannot always be estimated with sufficient accuracy. This problem is addressed in Section 5.3.5. As a result, Equation (5.9) does not always yield reliable flow estimates.

5.3.3 Block Motion Model

Another approach to overcoming the aperture problem is to assume that the motion vector remains unchanged over a particular block of pixels, denoted by $\mathcal{B}$ (suggested by Lucas and Kanade [Luc 81]); that is,

$$\mathbf{v}(\mathbf{x},\, t) = \mathrm{v}(t) = [v_1(t)\; v_2(t)]^T, \quad \text{for } \mathbf{x} \in \mathcal{B} \tag{5.10}$$

Although such a model cannot handle rotational motion, it is possible to estimate a purely translational motion vector uniquely under this model provided that the block of pixels contain sufficient gray-level variation.

Let's define the error in the optical flow equation over the block of pixels $\mathcal{B}$ as

$$E = \sum_{\mathbf{X}\in\mathcal{B}} \left(\frac{\partial s_c(\mathbf{x},t)}{\partial x_1} v_1(t) + \frac{\partial s_c(\mathbf{x},t)}{\partial x_2} v_2(t) + \frac{\partial s_c(\mathbf{x},t)}{\partial t} \right)^2 \tag{5.11}$$

Computing the partials of the error E with respect to $v_1(t)$ and $v_2(t)$, respectively, and setting them equal to zero, we have

$$\sum_{\mathbf{X}\in\mathcal{B}} \left(\frac{\partial s_c(\mathbf{x},t)}{\partial x_1} \hat{v}_1(t) + \frac{\partial s_c(\mathbf{x},t)}{\partial x_2} \hat{v}_2(t) + \frac{\partial s_c(\mathbf{x},t)}{\partial t} \right) \frac{\partial s_c(\mathbf{x},t)}{\partial x_1} = 0$$

$$\sum_{\mathbf{X}\in\mathcal{B}} \left(\frac{\partial s_c(\mathbf{x},t)}{\partial x_1} \hat{v}_1(t) + \frac{\partial s_c(\mathbf{x},t)}{\partial x_2} \hat{v}_2(t) + \frac{\partial s_c(\mathbf{x},t)}{\partial t} \right) \frac{\partial s_c(\mathbf{x},t)}{\partial x_2} = 0$$

where $\hat{\cdot}$ denotes the estimate of the respective quantity. Solving these equations simultaneously, we have

$$\begin{bmatrix} \hat{v}_1(t) \\ \hat{v}_2(t) \end{bmatrix} = \begin{bmatrix} \sum_{\mathbf{X}\in\mathcal{B}} \frac{\partial s_c(\mathbf{X},t)}{\partial x_1} \frac{\partial s_c(\mathbf{X},t)}{\partial x_1} & \sum_{\mathbf{X}\in\mathcal{B}} \frac{\partial s_c(\mathbf{X},t)}{\partial x_1} \frac{\partial s_c(\mathbf{X},t)}{\partial x_2} \\ \sum_{\mathbf{X}\in\mathcal{B}} \frac{\partial s_c(\mathbf{X},t)}{\partial x_1} \frac{\partial s_c(\mathbf{X},t)}{\partial x_2} & \sum_{\mathbf{X}\in\mathcal{B}} \frac{\partial s_c(\mathbf{X},t)}{\partial x_2} \frac{\partial s_c(\mathbf{X},t)}{\partial x_2} \end{bmatrix}^{-1} \begin{bmatrix} -\sum_{\mathbf{X}\in\mathcal{B}} \frac{\partial s_c(\mathbf{X},t)}{\partial x_1} \frac{\partial s_c(\mathbf{X},t)}{\partial t} \\ -\sum_{\mathbf{X}\in\mathcal{B}} \frac{\partial s_c(\mathbf{X},t)}{\partial x_2} \frac{\partial s_c(\mathbf{X},t)}{\partial t} \end{bmatrix} \tag{5.12}$$

It is possible to increase the influence of the constraints towards the center of the block $\mathcal{B}$ by replacing all summations with weighted summations. A suitable weighting function may be in the form of a 2-D triangular window. Clearly, the accuracy of the flow estimates depends on the accuracy of the estimated spatial and temporal partial derivatives (see Section 5.3.5). We next discuss the method of Horn and Schunck [Hor 81], which imposes a less restrictive global smoothness constraint on the velocity field.

5.3.4 Horn and Schunck Method

Horn and Schunck seek a motion field that satisfies the OFE with the minimum pixel-to-pixel variation among the flow vectors. Let

$$\mathcal{E}_{of}(\mathbf{v}(\mathbf{x},t)) = (\nabla s_c(\mathbf{x};t) , \mathbf{v}(\mathbf{x},t)) + \frac{\partial s_c(\mathbf{x};t)}{\partial t} \tag{5.13}$$

denote the error in the optical flow equation. Observe that the OFE is satisfied when $\mathcal{E}_{of}(\mathbf{v}(\mathbf{x}, t))$ is equal to zero. In the presence of occlusion and noise, we aim to minimize the square of $\mathcal{E}_{of}(\mathbf{v}(\mathbf{x}, t))$ in order to enforce the optical flow constraint.

The pixel-to-pixel variation of the velocity vectors can be quantified by the sum of the magnitude squares of the spatial gradients of the components of the velocity vector, given by

$$\begin{aligned}\mathcal{E}_s^2(\mathbf{v}(\mathbf{x},t)) &= ||\nabla v_1(\mathbf{x},t)||^2 + ||\nabla v_2(\mathbf{x},t)||^2 \\ &= (\frac{\partial v_1}{\partial x_1})^2 + (\frac{\partial v_1}{\partial x_2})^2 + (\frac{\partial v_2}{\partial x_1})^2 + (\frac{\partial v_2}{\partial x_2})^2 \end{aligned} \tag{5.14}$$

where we assume that the spatial and temporal coordinates are continuous variables. It can easily be verified that the smoother the velocity field, the smaller $\mathcal{E}_s^2(\mathbf{v}(\mathbf{x}, t))$.

Then the Horn and Schunck method minimizes a weighted sum of the error in the OFE and a measure of the pixel-to-pixel variation of the velocity field

$$\min_{\mathbf{v}(\mathbf{x},t)} \int_A (\mathcal{E}_{of}^2(\mathbf{v}) + \alpha^2 \mathcal{E}_s^2(\mathbf{v})) d\mathbf{x} \tag{5.15}$$

to estimate the velocity vector at each point x, where A denotes the continuous image support. The parameter α^2, usually selected heuristically, controls the strength of the smoothness constraint. Larger values of α^2 increase the influence of the constraint.

The minimization of the functional (5.15), using the calculus of variations, requires solving the two equations

$$\begin{aligned}(\frac{\partial s_c}{\partial x_1})^2 \hat{v}_1(\mathbf{x},t) + \frac{\partial s_c}{\partial x_1}\frac{\partial s_c}{\partial x_2}\hat{v}_2(\mathbf{x},t) &= \alpha^2\nabla^2\hat{v}_1(\mathbf{x},t) - \frac{\partial s_c}{\partial x_1}\frac{\partial s_c}{\partial t} \\ \frac{\partial s_c}{\partial x_1}\frac{\partial s_c}{\partial x_2}\hat{v}_1(\mathbf{x},t) + (\frac{\partial s_c}{\partial x_2})^2\hat{v}_2(\mathbf{x},t) &= \alpha^2\nabla^2\hat{v}_2(\mathbf{x},t) - \frac{\partial s_c}{\partial x_2}\frac{\partial s_c}{\partial t}\end{aligned} \tag{5.16}$$

simultaneously, where ∇^2 denotes the Laplacian and $\hat{\cdot}$ denotes the estimate of the respective quantity. In the implementation of Horn and Schunck [Hor 81], the Laplacians of the velocity components have been approximated by FIR highpass filters to arrive at the Gauss-Seidel iteration

$$\begin{aligned}\hat{v}_1^{(n+1)}(\mathbf{x},t) &= \bar{v}_1^{(n)}(\mathbf{x},t) - \frac{\partial s_c}{\partial x_1}\frac{\frac{\partial s_c}{\partial x_1}\bar{v}_1^{(n)}(\mathbf{x},t) + \frac{\partial s_c}{\partial x_2}\bar{v}_2^{(n)}(\mathbf{x},t) + \frac{\partial s_c}{\partial t}}{\alpha^2 + (\frac{\partial s_c}{\partial x_1})^2 + (\frac{\partial s_c}{\partial x_2})^2} \\ \hat{v}_2^{(n+1)}(\mathbf{x},t) &= \bar{v}_2^{(n)}(\mathbf{x},t) - \frac{\partial s_c}{\partial x_2}\frac{\frac{\partial s_c}{\partial x_1}\bar{v}_1^{(n)}(\mathbf{x},t) + \frac{\partial s_c}{\partial x_2}\bar{v}_2^{(n)}(\mathbf{x},t) + \frac{\partial s_c}{\partial t}}{\alpha^2 + (\frac{\partial s_c}{\partial x_1})^2 + (\frac{\partial s_c}{\partial x_2})^2}\end{aligned} \tag{5.17}$$

where n is the iteration counter, the overbar denotes weighted local averaging (excluding the present pixel), and all partials are evaluated at the point $(\mathbf{x}, t)$. The reader is referred to [Hor 81] for the derivation of this iterative estimator. The initial estimates of the velocities $v_1^{(0)}(\mathbf{x}, t)$ and $v_2^{(0)}(\mathbf{x}, t)$ are usually taken as zero. The above formulation assumes a continuous spatio-temporal intensity distribution. In computer implementation, all spatial and temporal image gradients need to be estimated numerically from the observed image samples, which will be discussed in the next subsection.

5.3.5 Estimation of the Gradients

We discuss two gradient estimation methods. The first method makes use of finite differences, and the second is based on polynomial fitting.

Gradient Estimation Using Finite Differences

One approach to estimating the partial derivatives from a discrete image $s(n_1, n_2, k)$ is to approximate them by the respective forward or backward finite differences. In order to obtain more robust estimates of the partials, we can compute the average of the forward and backward finite differences, called the average difference. Furthermore, we can compute a local average of the average differences to eliminate the effects of observation noise. Horn and Schunck [Hor 81] proposed averaging four finite differences to obtain

$$\begin{aligned}\frac{\partial s_c(x_1,x_2,t)}{\partial x_1} \approx \frac{1}{4}\{ & s(n_1+1,n_2,k) - s(n_1,n_2,k) + s(n_1+1,n_2+1,k) \\ & - s(n_1,n_2+1,k) + s(n_1+1,n_2,k+1) - s(n_1,n_2,k+1) \\ & + s(n_1+1,n_2+1,k+1) - s(n_1,n_2+1,k+1)\} \\ \frac{\partial s_c(x_1,x_2,t)}{\partial x_2} \approx \frac{1}{4}\{ & s(n_1,n_2+1,k) - s(n_1,n_2,k) + s(n_1+1,n_2+1,k) \\ & - s(n_1+1,n_2,k) + s(n_1,n_2+1,k+1) - s(n_1,n_2,k+1) \\ & + s(n_1+1,n_2+1,k+1) - s(n_1+1,n_2,k+1)\} \\ \frac{\partial s_c(x_1,x_2,t)}{\partial t} \approx \frac{1}{4}\{ & s(n_1,n_2,k+1) - s(n_1,n_2,k) + s(n_1+1,n_2,k+1) \\ & - s(n_1+1,n_2,k) + s(n_1,n_2+1,k+1) - s(n_1,n_2+1,k) \\ & + s(n_1+1,n_2+1,k+1) - s(n_1+1,n_2+1,k)\}\end{aligned} \tag{5.18}$$

Various other averaging strategies exist for estimating partials using finite differences [Lim 90]. Spatial and temporal presmoothing of video with Gaussian kernels usually helps gradient estimation.

Gradient Estimation by Local Polynomial Fitting

An alternative approach is to approximate $s_c(x_1, x_2, t)$ locally by a linear combination of some low-order polynomials in x_1, x_2 and t; that is,

$$s_c(x_1, x_2, t) \approx \sum_{i=0}^{N-1} a_i \phi_i(x_1, x_2, t) \tag{5.19}$$

where $\phi_i(x_1, x_2, t)$ are the basis polynomials, N is the number of basis functions used in the polynomial approximation, and a_i are the coefficients of the linear combination. Here, we will set N equal to 9, with the following choice of the basis functions:

$$\phi_i(x_1, x_2, t) = 1, x_1, x_2, t, x_1^2, x_2^2, x_1x_2, x_1t, x_2t \tag{5.20}$$

which are suggested by Lim [Lim 90]. Then, Equation (5.19) becomes,

$$s_c(x_1, x_2, t) \approx a_0 + a_1x_1 + a_2x_2 + a_3t + a_4x_1^2 + a_5x_2^2 + a_6x_1x_2 + a_7x_1t + a_8x_2t \tag{5.21}$$

The coefficients a_i, i = 0,. . ,8, are estimated by using the least squares method, which minimizes the error function

$$e^2 = \sum_{n_1}\sum_{n_2}\sum_{k}\left(s(n_1, n_2, k) - \sum_{i=0}^{\mathbf{N-1}} a_i\phi_i(x_1, x_2, t)|_{[x_1\ x_2\ t]^T=\mathbf{V}[n_1\ n_2\ k]^T}\right)^2 \tag{5.22}$$

with respect to these coefficients. The summation over (n_1, n_2, k) is carried within a local neighborhood of the pixel for which the polynomial approximation is made. A typical case involves 50 pixels, 5 x 5 spatial windows in two consecutive frames.

Once the coefficients a_i are estimated, the components of the gradient can be found by simple differentiation,

$$\frac{\partial s_c(x_1, x_2, t)}{\partial x_1} \approx a_1 + 2a_4x_1 + a_6x_2 + a_7t|_{x_1=x_2=t=0} = a_1 \tag{5.23}$$

$$\frac{\partial s_c(x_1, \mathbf{x2}, t)}{\partial x_2} \approx a_2 + 2a_5x_2 + a_6x_1 + a_8t|_{x_1=x_2=t=0} = a_2 \tag{5.24}$$

$$\frac{\partial s_c(x_1, x_2, t)}{\partial t} \approx a_3 + a_7x_1 + a_8x_2|_{x_1=x_2=t=0} = a_3 \tag{5.25}$$

Similarly, the second-order and mixed partials can be easily estimated in terms of the coefficients a_4 through us.

5.3.6 Adaptive Methods

The Horn-Schunck method imposes the optical flow and smoothness constraints globally over the entire image, or over a motion estimation window. This has two undesired effects:

i) The smoothness constraint does not hold in the direction perpendicular to an occlusion boundary. Thus, a global smoothness constraint blurs "motion edges." For example, if an object moves against a stationary background, there is a sudden change in the motion field at the boundary of the object. Motion edges can be preserved by imposing the smoothness constraint only in the directions along which the pixel intensities do not significantly change. This is the basic concept of the so-called directional or oriented smoothness constraint.

ii) The nonadaptive method also enforces the optical flow constraint at the occlusion regions, where it should be turned off. This can be achieved by adaptively varying α to control the relative strengths of the optical flow and smoothness constraints. For example, at occlusion regions, such as the dotted regions shown in Figure 5.6, the optical flow constraint can be completely turned off, and the smoothness constraint can be fully on.

Several researchers proposed to impose the smoothness constraint along the boundaries but not perpendicular to the occlusion boundaries. Hildreth [Hil 84] minimized the criterion function of Horn and Schunck given by (5.15) along object contours. Nagel and Enkelman [Nag 86, Enk 88] introduced the concept of directional smoothness, which suppresses the smoothness constraint in the direction of the spatial image gradient. Fogel [Fog 91] used directional smoothness constraints with adaptive weighting in a hierarchical formulation. Note that adaptive weighting methods require strategies to detect moving object (occlusion) boundaries. Recently, Snyder [Sny 91] proposed a general formulation of the smoothness constraint that includes some of the above as special cases.

Directional-Smoothness Constraint

The directional smoothness constraint can be expressed as

$$\mathcal{E}_{ds}^2(\mathbf{v}(\mathbf{x}, t)) = (\nabla v_1)^T\mathbf{W}(\nabla v_1) + (\nabla v_2)^T\mathbf{W}(\nabla v_2) \tag{5.26}$$

where $\mathbf{W}$ is a weight matrix to penalize variations in the motion field depending on the spatial changes in gray-level content of the video. Various alternatives for the weight matrix $\mathbf{W}$ exist [Nag 86, Nag 87, Enk 88]. For example, $\mathbf{W}$ can be chosen as

$$\mathbf{W} = \frac{\mathbf{F} + \delta\mathbf{I}}{\mathrm{trace}(\mathbf{F} + \delta\mathbf{I})} \tag{5.27}$$

where $\mathbf{I}$ is the identity matrix representing a global smoothness term to ensure a nonzero weight matrix at spatially uniform regions, 6 is a scalar, and

$$\mathbf{F} = \begin{bmatrix} (\frac{\partial s_c}{\partial x_1})^2 + b^2((\frac{\partial^2 s_c}{\partial x_1^2})^2 + (\frac{\partial^2 s_c}{\partial x_1 \partial x_2})^2) & \frac{\partial s_c}{\partial x_1}\frac{\partial s_c}{\partial x_2} + b^2\frac{\partial^2 s_c}{\partial x_1 \partial x_2}(\frac{\partial^2 s_c}{\partial x_1^2} + \frac{\partial^2 s_c}{\partial x_2^2}) \\ \frac{\partial s_c}{\partial x_1}\frac{\partial s_c}{\partial x_2} + b^2\frac{\partial^2 s_c}{\partial x_1 \partial x_2}(\frac{\partial^2 s_c}{\partial x_1^2} + \frac{\partial^2 s_c}{\partial x_2^2}) & (\frac{\partial s_c}{\partial x_2})^2 + b^2((\frac{\partial^2 s_c}{\partial x_1 \partial x_2})^2 + (\frac{\partial^2 s_c}{\partial x_2^2})^2) \end{bmatrix}^{-1}$$

with b^2 a constant.

Then, the directional-smoothness method minimizes the criterion function

$$\min_{\mathbf{v}(\mathbf{x},t)} \int_{\mathcal{A}} (\mathcal{E}_{of}^2(\mathbf{v}) + \alpha^2 \mathcal{E}_{ds}^2(\mathbf{v}))d\mathbf{x} \tag{5.28}$$

to find an estimate of the motion field, where A denotes the image support and α^2 is the smoothness parameter. Observe that the method of Horn and Schunck (5.15) is a special case of this formulation with $\delta = 1$ and $\mathbf{F} = \mathbf{0}$. A Gauss-Seidel iteration to minimize (5.28) has been described in [Enk 88], where the update term in each iteration is computed by means of a linear algorithm. The performance of the directional-smoothness method depends on how accurately the required second and mixed partials of image intensity can be estimated.

Hierarchical Approach

Fogel [Fog 91] used the concepts of directional smoothness and adaptive weighting in an elegant hierarchical formulation. A multiresolution representation $s_c^\alpha(x_1, x_2, t)$ of the video was defined as

$$s_c^\alpha(x_1, x_2, t) \doteq \int_{\mathcal{A}} s_c(x_1, x_2, t) h(\frac{\mu - x_1}{\alpha}, \frac{\eta - x_2}{\alpha}) d\mu d\eta \tag{5.29}$$

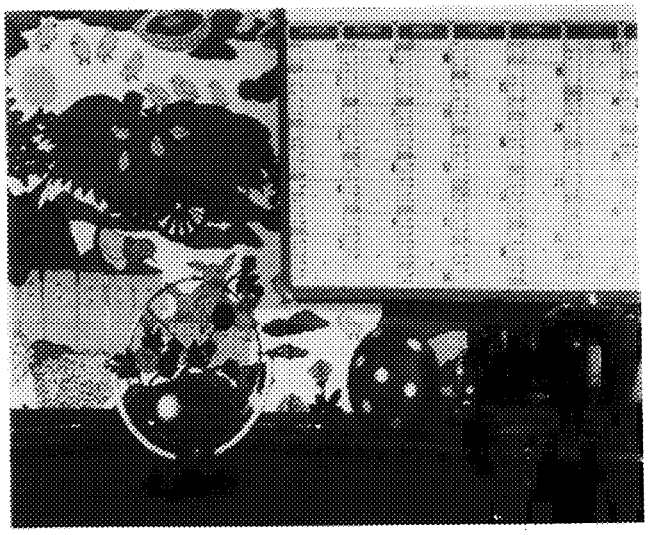
where α is the resolution parameter, $\mathcal{A}$ denotes the image support, and

$$h(x_1, x_2) = \begin{cases} A \exp\{\frac{-(C^2x_1^2 + D^2x_2^2)}{B^2 - (C^2x_1^2 + D^2x_2^2)}\} & C^2x_1^2 + D^2x_2^2 < B^2 \\ 0 & \text{otherwise} \end{cases} \tag{5.30}$$

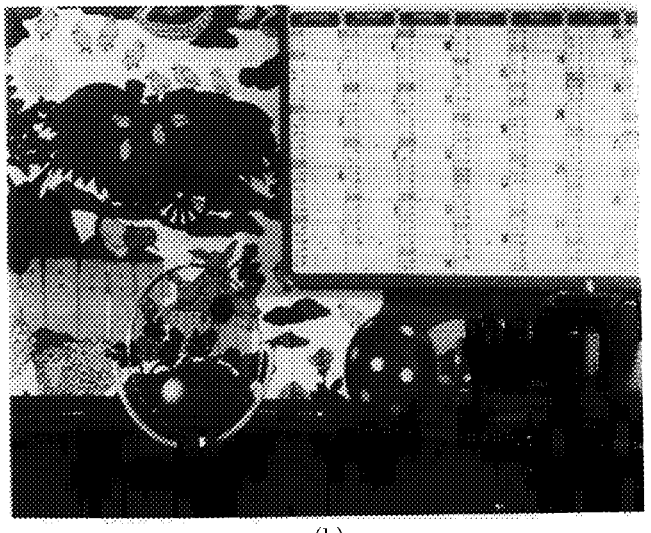
denotes a low pass filter with parameters A, B, C and D. It can readily be seen that the spatial resolution of $s_c^\alpha(x_1, x_2, t)$ decreases as α increases. Observe that spatial partial derivatives of $s_c^\alpha(x_1, x_2, t)$ can be computed by correlating $s_c(x_1, x_2, t)$ with the partial derivatives of $h(x_1, x_2)$ which can be computed analytically. A nonlinear optimization problem was solved at each resolution level using a Quasi-Newton method. The estimate obtained at one resolution level was used as the initial estimate at the next higher resolution level. Interested readers are referred to [Fog 91] for implementational details.

5.4 Examples

We compare the results of three representative methods of increasing complexity: a simple Lucas-Kanade (L-K) type estimator given by (5.12) based on the block motion model, the Horn-Schunck (H-S) method (5.17) imposing a global smoothness constraint, and the directional-smoothness method of Nagel. In all cases, the spatial and temporal gradients have been approximated by both average finite differences and polynomial fitting as discussed in Section 5.3.5. The images are spatially presmoothed by a 5 x 5 Gaussian kernel with the variance 2.5 pixels.



(a)



(b)

Figure 5.9: a) First and b) second frames of the Mobile and Calendar sequence. (source: CCETT, Cesson Sevigne Cedex, France.)

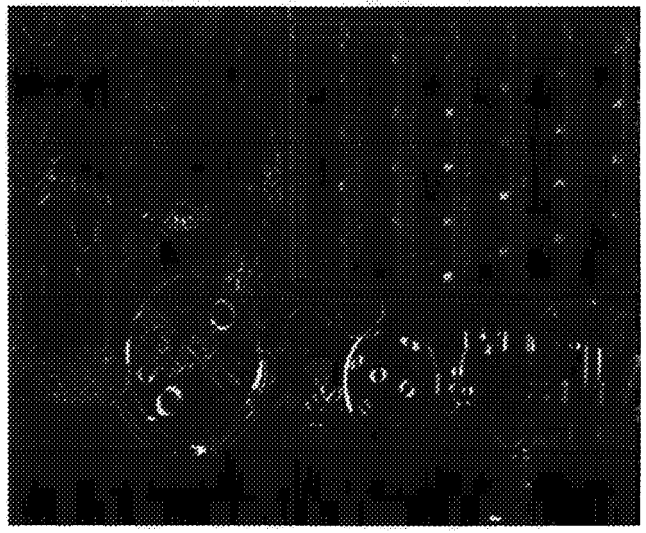


Figure 5.10: Absolute value of the frame difference. (Courtesy Gozde Bozdagi)

Our implementation of the L-K method considers 11 x 11 blocks with no weighting. In the H-S algorithm, we set α^2 = 625, and allowed for 20 to 150 iterations. The parameters of the Nagel algorithm were set to $\alpha^2 = 25$ and $\delta = 5$ with 20 iterations.

These methods have been applied to estimate the motion between the seventh and eighth frames of a progressive video, known as the "Mobile and Calendar" sequence, shown in Figure 5.9 (a) and (b), respectively. Figure 5.10 (a) shows the absolute value of the frame difference (multiplied by 3), without any motion compensation, to indicate the amount of motion present. The lighter pixels are those whose intensity has changed with respect to the previous frame due to motion. Indeed, the scene contains multiple motions: the train is moving forward (from right to left), pushing the ball in front of it; there is a small ball in the foreground spinning around a circular ring; the background moves toward the right due to camera pan; and the calendar moves up and down. The motion fields estimated by the L-K and the H-S methods are depicted in Figure 5.11 (a) and (b), respectively. It can be seen that the estimated fields capture most of the actual motion.

We evaluate the goodness of the motion estimates on the basis of the peak signal-to-noise ratio (PSNR) of the resulting displaced frame difference (DFD) between the seventh and eighth frames, defined by

$$PSNR = 10 \log_{10} \frac{255 \times 255}{\sum [s_8(n_1, n_2) - s_7(n_1 + d_1(n_1, n_2), n_2 + d_2(n_1, n_2))]^2} \quad (5.31)$$

where d_1 and d_2 are the components of the motion estimates at each pixel. We also computed the entropy of the estimated 2-D motion field, given by

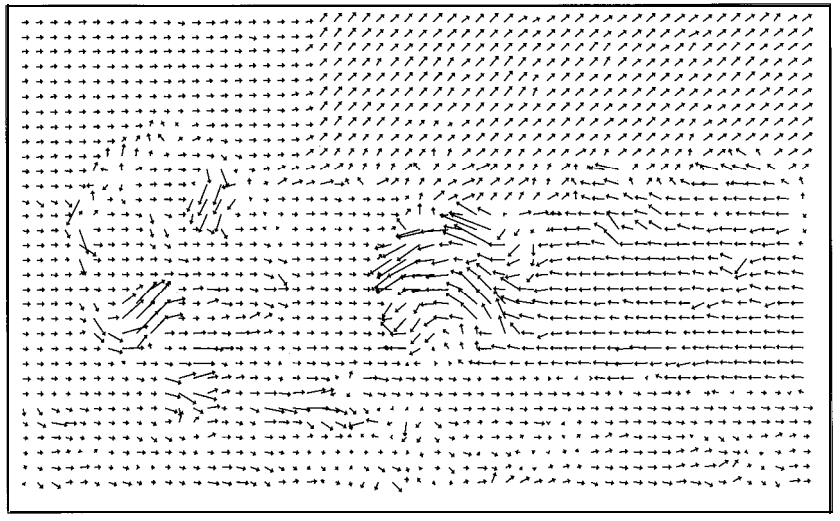
$$H = -\sum_{d_1} P(d_1) log_2 P(d_1) - \sum_{d_2} P(d_2) log_2 P(d_2) \quad (5.32)$$

where $P(d_1)$ and $P(d_2)$ denote the relative frequency of occurence of the horizontal and vertical components of the motion vector **d.** The entropy, besides serving as a measure of smoothness of the motion field, is especially of interest in motion-compensated video compression, where one wishes to minimize both the entropy of the motion field (for cheaper transmission) and the energy of the DFD. The PSNR and the entropy of all methods are listed in Table 5.1 after 20 iterations (for H-S and Nagel algorithms). In the case of the H-S method, the PSNR increases to 32.23 dB after 150 iterations using average finite differences. Our experiments indicate that the Nagel algorithm is not as robust as the L-K and H-S algorithms, and may diverge if some stopping criterion is not employed. This may be due to inaccuracies in the estimated second and mixed partials of the spatial image intensity.

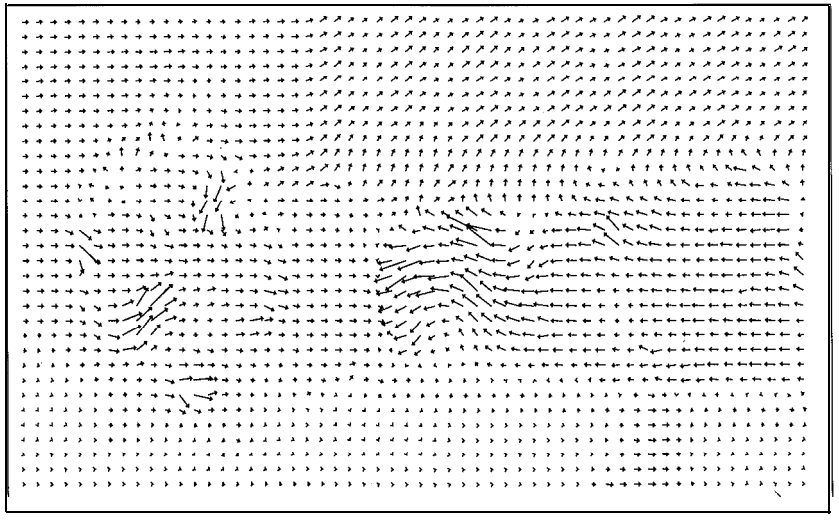
Table 5.1: Comparison of the differential methods. (Courtesy Gozde Bozdagi)

Method	PSNR (dB)		[illegible]its)	
	Polynomial	Differences	Polynomial	Differences
Frame-Difference	23.45	-	-	-
Lucas-Kanade	30.89	32.09	6.4[illegible]	6.82
Horn-Schunck	28.14	30.71	4.22	5.04
Nagel	29.08	31.84	5.83	5.95

The reader is alerted that the mean absolute DFD provides a measure of the goodness of the pixel correspondence estimates. However, it does not provide insight about how well the estimates correlate with the projected 3-D displacement vectors. Recall that the optical flow equation enables estimation of the normal flow vectors at each pixel, rather than the actual projected flow vectors.



(a)



(b)

Figure 5.11: Motion field obtained by a) the Lucas-Kanade method and b) the Horn-Schunck method. (Courtesy Gozde Bozdagi and Mehmet Ozkan)

5.5 Exercises

1. For a color image, the optical flow equation (5.5) can be written for each of the R, G, and B channels separately. State the conditions on the (R,G,B) intensities so that we have at least two linearly independent equations at each pixel. How valid are these conditions for general color images?

2. State the conditions on spatio-temporal image intensity and the velocity under which the optical flow equation can be used for displacement estimation. Why do we need the small motion assumption?

3. What are the conditions for the existence of normal flow (5.7)? Can we always recover optical flow from the normal flow? Discuss the relationship between the spatial image gradients and the aperture problem.

4. Suggest methods to detect occlusion

5. Derive (5.9) from (5.8).

6. Show that the constraint (5.8) does not hold when there is rotation or zoom.

7. Find the least squares estimates of a_1, a_2 and us in (5.23)-(5.25).

Bibliography

[Agg 88] J. K. Aggarwal and N. Nandhakumar, "On the computation of motion from sequences of images," *Proc. IEEE*, vol. 76, pp. 917–935, Aug. 1988.

[Ana 93] P. Anandan, J. R. Bergen, K. J. Hanna, and R. Hingorani, "Hierarchical model-based motion estimation," in *Motion* Analysis and Image Sequence *Processing*, M. I. Sezan and R. L. Lagendijk, eds., Norwell, MA: Kluwer, 1993.

[Bar 94] J. L. Barron, D. J. Fleet, and S. S. Beauchemin, "Systems and experiment: Performance of optical flow techniques," *Int. J. Comp. Vision*, vol. 12:1, pp. 43–77, 1994.

[Ber 88] M. Bertero, T. A. Poggio and V. Torre, "Ill-posed problems in early vision," *Proc. IEEE*, vol. 76, pp. 869-889, August 1988.

[Enk 88] W. Enkelmann, "Investigations of multigrid algorithms for the estimation of optical flow fields in image sequences," *Comp. Vis.* **Graph** Image *Proc.*, vol. 43, pp. 1506177, 1988.

[Fle 92] D. J. Fleet, Measurement *of* **Image** *Velocity*, Norwell, MA: Kluwer, 1992.

[Fog 91] S. V. Fogel, "Estimation of velocity vector fields from time-varying image sequences," *CVGIP:* Image Understanding, vol. 53, pp. 253-287, 1991.

[Hil 84] E. C. Hildreth, 'Computations underlying the measurement of visual motion," *Artif.* **Intel.,** *vol. 23,* pp. 309-354, 1984.

[Hor 81] B. K. P. Horn and B. G. Schunck, "Determining optical flow," *Artif. Intell.*, vol. 17, pp. 185-203, 1981.

[Hua 81] T. S. Huang, ed., *Image Sequence Analysis,* Springer Verlag, 1981.

[Hua 83] T. S. Huang, ed., Image *Sequence Processing* and Dynamic *Scene Analysis,* Berlin, Germany: Springer-Verlag, 1983.

[Lim 90] J. S. Lim, Two-Dimensional Signal *and* Image Processing, Englewood Cliffs, NJ: Prentice Hall, 1990.

[Luc 81] B. D. Lucas and *T.* Kanade, "An iterative image registration technique with an application to stereo vision," *Proc. DARPA Image Understanding Workshop,* pp. 121–130, 1981.

[Nag 86] H. H. Nagel and W. Enkelmann, "An investigation of smoothness constraints for the estimation of displacement vector fields from image sequences," *IEEE* Trans. *Patt.* Anal. Mach. *Intel.,* vol. 8, pp. 565-593, 1986.

[Nag 87] H. H. Nagel, "On the estimation of optical flow: Relations between different approaches and some new results," *Artificial Intelligence,* vol. 33, pp. 2999324, 1987.

[Oht 90] N. Ohta, "Optical flow detection by color images," *NEC Res.* and *Dev.,* no. 97, pp. 78-84, Apr. 1990.

[Sez 93] M. I. Sezan and R. L. Lagendijk, eds., *Motion* Analysis *and* Image *Sequence Processing,* Norwell, MA: Kluwer, 1993.

[Sin 91] A. Singh, *Optic* Flow Compatation, Los Alamitos, CA: IEEE Computer Soc. Press, 1991.

[Sny 91] M. A. Snyder, "On the mathematical foundations of smoothness constraints for the determination of optical flow and for surface reconstruction," *IEEE* Trans. *Patt. Anal. Mach. Intel.,* vol. 13, pp. 1105–1114, 1991.

[Ura 88] S. Uras, F. Girosi, A. Verri, and V. Torre, "A computational approach to motion perception," *Biol. Cybern.,* vol. 60, pp. 79-97, 1988.

[Ver 89] A. Verri and T. Poggio, "Motion field and optical flow: Qualitative properties," *IEEE* Trans. *Patt.* Anal. *Mach. Intel.,* vol. PAMI-11, pp. 490–498, May 1989.

Chapter 6

BLOCK-BASED METHODS

Block-based motion estimation and compensation are among the most popular approaches. Block-based motion compensation has been adopted in the international standards for digital video compression, such as H.261 and MPEG 1-2. Although these standards do not specify a particular motion estimation method, block-based motion estimation becomes a natural choice. Block-based motion estimation is also widely used in several other digital video applications, including motion-compensated filtering for standards conversion.

We start with a brief introduction of the block-motion models in Section 6.1. A simple block-based motion estimation scheme, based on the translatory block model, was already discussed in Chapter 5, in the context of estimation using the optical flow equation. In this chapter, we present two other translatory-block-based motion estimation strategies. The first approach, discussed in Section 6.2, is a spatial frequency domain technique, called the phase-correlation method. The second, presented in Section 6.3, is a spatial domain search approach, called the block-matching method. Both methods can be implemented hierarchically, using a multiresolution description of the video. Hierarchical block-motion estimation is addressed in Section 6.4. Finally, in Section 6.5, possible generalizations of the block-matching framework are discussed, including 2-D deformable block motion estimation, in order to overcome some shortcomings of the translatory block-motion model.

6.1 Block-Motion Models

The block-motion model assumes that the image is composed of moving blocks. We consider two types of block motion: i) simple 2-D translation, and ii) various 2-D deformations of the blocks.

6.1.1 Translational Block Motion

The simplest form of this model is that of translatory blocks, restricting the motion of each block to a pure translation. Then an N x N block $\mathcal{B}$ in frame k centered about the pixel $\mathbf{n} = (n_1, n_2)$ is modeled as a globally shifted version of a same-size block in frame $k + \ell$, for an integer ℓ. That is,

$$s(n_1, n_2, k) = s_c(x_1 + d_1, x_2 + d_2, t + \ell\Delta t)\Big|_{\begin{bmatrix} \mathbf{x} \\ t \end{bmatrix} = \mathbf{V}\begin{bmatrix} \mathbf{n} \\ k \end{bmatrix}} \tag{6.1}$$

for all $(n_1, n_2) \in \mathcal{B}$, where d_1 and d_2 are the components of the displacement (translation) vector for block $\mathcal{B}$. Recall from the previous chapter that the right-hand side of (6.1) is given in terms of the continuous time-varying image $s_c(x_1, x_2, \mathrm{t})$, because d_1 and d_2 are real-valued. Assuming the values of d_1 and d_2 are quantized to the nearest integer, the model (6.1) can be simplified as

$$s(n_1, n_2, k) = s(n_1 + d_1, n_2 + d_2, k + \ell) \tag{6.2}$$

Observe that it is possible to obtain $1/2^L$ pixel accuracy in the motion estimates, using either the phase-correlation or the block-matching methods, if the frames k and $k + \ell$ in (6.2) are interpolated by a factor of L.

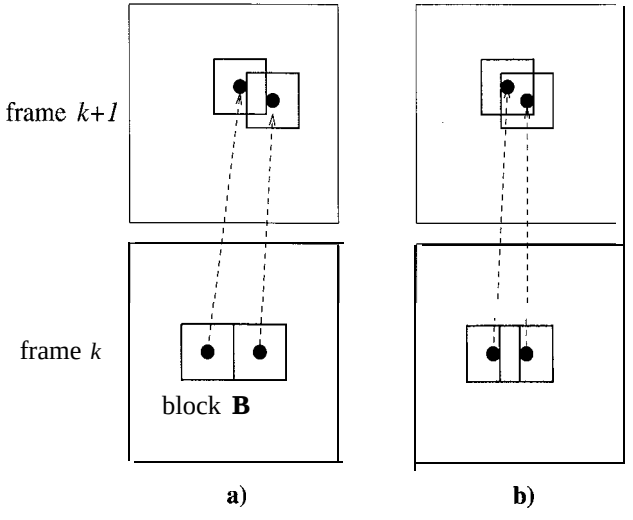


Figure 6.1: Block-motion models: a) nonoverlapping and b) overlapping blocks.

In the model (6.1), the blocks $\mathcal{B}$ may be nonoverlapping or overlapping as shown in Figure 6.1 (a) and (b), respectively. In the nonoverlapping case, the entire block

is assigned a single motion vector. Hence, motion compensation can be achieved by copying the gray-scale or color information from the corresponding block in the frame $k + 1$ on a pixel-by-pixel basis. In the case of overlapping blocks, we can either compute the average of the motion vectors within the overlapping regions, or select one of the estimated motion vectors. Motion compensation in the case of overlapping blocks was discussed in [Sul 93], where a multihypothesis expectation approach was proposed.

The popularity of motion compensation and estimation based on the model of translational blocks originates from:

- low overhead requirements to represent the motion field, since one motion vector is needed per block., and
- . ready availability of low-cost VLSI implementations.

However, motion compensation using translational blocks i) fails for zoom, rotational motion, and under local deformations, and ii) results in serious blocking artifacts, especially for very-low-bitrate applications, because the boundaries of objects do not generally agree with block boundaries, and adjacent blocks may be assigned substantially different motion vectors.

6.1.2 Generalized/Deformable Block Motion

In order to generalize the translational block model (6.1), note that it can be characterized by a simple frame-to-frame pixel coordinate (spatial) transformation of the form

$$\begin{aligned} x_1' &= x_1 + d_1 \\ x_2' &= x_2 + d_2 \end{aligned} \tag{6.3}$$

where (x_1', x_2') denotes the coordinates of a point in the frame $k + l$. The spatial transformation (6.3) can be generalized to include affine coordinate transformations, given by

$$\begin{aligned} x_1' &= a_1x_1 + a_2x_2 + d_1 \\ x_2' &= a_3x_1 + a_4x_2 + d_2 \end{aligned} \tag{6.4}$$

The affine transformation (6.4) can handle rotation of blocks as well as 2-D deformation of squares (rectangles) into parallelograms, as depicted in Figure 6.2. Other spatial transformations include the perspective and bilinear coordinate transformations. The perspective transformation is given by

$$\begin{aligned} x_1' &= \frac{a_1x_1 + a_2x_2 + a_3}{a_7x_1 + a_8x_2 + 1} \\ x_2' &= \frac{a_4x_1 + a_5x_2 + a_6}{a_7x_1 + a_8x_2 + 1} \end{aligned} \tag{6.5}$$

whereas the bilinear transformation can be expressed as

$$\begin{aligned} x_1' &= a_1x_1 + a_2x_2 + a_3x_1x_2 + a_4 \\ x_2' &= a_5x_1 + a_6x_2 + a_7x_1x_2 + a_8 \end{aligned} \tag{6.6}$$

We will see in Chapter 9 that the affine and perspective coordinate transformations correspond to the orthographic and perspective projection of the 3-D rigid motion of a planar surface, respectively. However, the bilinear transformation is not related to any physical 3-D motion. Relevant algebraic and geometric properties of these transformations have been developed in [Wol 90].

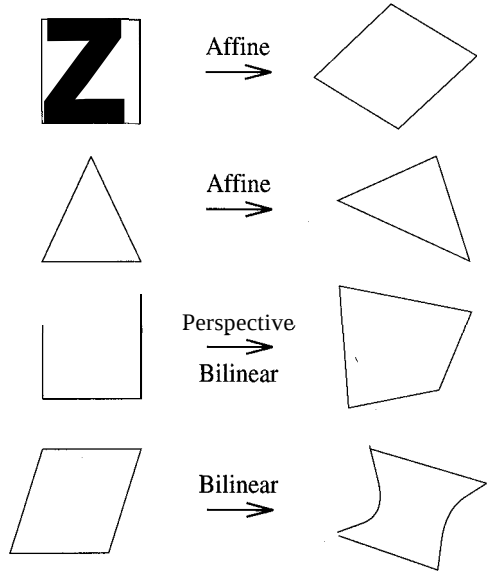


Figure 6.2: Examples of spatial transformations.

While the basic phase-correlation and block-matching methods are based on the translational model (6.3), generalizations of the block-matching method to track 2-D deformable motion based on the spatial transformations depicted in Figure 6.2 will be addressed in Section 6.5. We note that various block-motion models, including (6.4) and (6.5), fall under the category of parametric motion models (discussed in Chapter 5), and may be considered as local regularization constraints on arbitrary displacement fields to overcome the aperture problem, in conjunction with the optical-flow-equation-based methods (see Section 5.3.3), or pel-recursive methods (which are described in Chapter 7).

6.2 Phase-Correlation Method

Taking the 2-D Fourier transform of both sides of the discrete motion model (6.2), with $\ell = 1$, over a block $\mathcal{B}$ yields

$$S_k(f_1, f_2) = S_{k+1}(f_1, f_2) \exp\{j2\pi(d_1f_1 + d_2f_2)\} \tag{6.7}$$

where $S_k(f_1, f_2)$ denotes the 2-D Fourier transform of the frame k with respect to the spatial variables x_1 and x_2. It follows that, in the case of translational motion, the difference of the 2-D Fourier phases of the respective blocks,

$$\arg\{S(f_1, f_2, k)\} - \arg\{S(f_1, f_2, k+1)\} = 2\pi(d_1f_1 + d_2f_2) \tag{6.8}$$

defines a plane in the variables (f_1, f_2). Then the interframe motion vector can be estimated from the orientation of the plane (6.8). This seemingly straightforward approach runs into two important problems: i) estimation of the orientation of the plane in general requires 2-D phase unwrapping, which is not trivial by any means; and ii) it is not usually easy to identify the motion vectors for more than one moving object within a block. The phase-correlation method alleviates both problems [Tho 87]. Other frequency-domain motion estimation methods include those based on 3-D spatio-temporal frequency-domain analysis using Wigner distributions [Jac 87] or a set of Gabor filters [Hee 87].

The phase-correlation method estimates the relative shift between two image blocks by means of a normalized cross-correlation function computed in the 2-D spatial Fourier domain. It is also based on the principle that a relative shift in the spatial domain results in a linear phase term in the Fourier domain. In the following, we first show thk derivation of the phase-correlation function, and then discuss some issues related to its implementation. Although an extension of the phase-correlation method to include rotational motion was also suggested [Cas 87], that will not be covered here.

6.2.1 The Phase-Correlation Function

The cross-correlation function between the frames k and $k + 1$ is defined as

$$c_{k,k+1}(n_1, n_2) = s(n_1, n_2, k+1) ** s(-n_1, -n_2, k) \tag{6.9}$$

where $**$ denotes the 2-D convolution operation. Taking the Fourier transform of both sides, we obtain the complex-valued cross-power spectrum expression

$$C_{k,k+1}(f_1, f_2) = S_{k+1}(f_1, f_2)S_k^*(f_1, f_2) \tag{6.10}$$

Normalizing $C_{k,k+1}(f_1, f_2)$ by its magnitude gives the phase of the cross-power spectrum

$$\tilde{C}_{k,k+1}(f_1, f_2) = \frac{S_{k+1}(f_1, f_2)S_k^*(f_1, f_2)}{|S_{k+1}(f_1, f_2)S_k^*(f_1, f_2)|} \tag{6.11}$$

Assuming translational motion, we substitute (6.7) into (6.11) to obtain

$$\tilde{C}_{k,k+1}(f_1, f_2) = \exp\{-j2\pi(f_1 d_1 + f_2 d_2)\} \tag{6.12}$$

Taking the inverse 2-D Fourier transform of this expression yields the phase-correlation function

$$\tilde{c}_{k,k+1}(n_1, n_2) = \delta(n_1 - d_1, n_2 - d_2) \tag{6.13}$$

We observe that the phase-correlation function consists of an impulse whose location yields the displacement vector.

6.2.2 Implementation Issues

Implementation of the phase-correlation method in the computer requires replacing the 2-D Fourier transforms by the 2-D DFT, resulting in the following algorithm:

1. Compute the 2-D DFT of the respective blocks from the kth and $k+1$th frames.
2. Compute the phase of the cross-power spectrum as in (6.11).
3. Compute the 2-D inverse DFT of $\tilde{C}_{k,k+1}(f_1, f_2)$ to obtain the phase-correlation function $\tilde{c}_{k,k+1}(n_1, n_2)$.
4. Detect the location of the peak(s) in the phase-correlation function

Ideally, we expect to observe a single impulse in the phase-correlation function indicating the relative displacement between the two blocks. In practice, a number of factors contribute to the degeneration of the phase-correlation function to contain one or more peaks. They are the use of the 2-D DFT instead of the 2-D Fourier transform, the presence of more than one moving object within the block, and the presence of observation noise.

The use of the 2-D DFT instead of the 2-D Fourier transform has a number of consequences:

- Boundary effects: In order to obtain a perfect impulse, the shift must be cyclic. Since things disappearing at one end of the window generally do not reappear at the other end, the impulses degenerate into peaks. Further, it is well known that the 2-D DFT assumes periodicity in both directions. Discontinuities from left to right boundaries, and from top to bottom, may introduce spurious peaks.
- Spectral leakage due to noninteger motion vectors: In order to observe a perfect impulse, the components of the displacement vector must correspond to an integer multiple of the fundamental frequency. Otherwise, the impulse will degenerate into a peak due to the well-known spectral leakage phenomenon.

- Range of displacement estimates: Since the 2-D DFT is periodic with the block size (N_1, N_2), the displacement estimates need to be unwrapped as

$$\hat{d}_i = \begin{cases} d_i & \text{if } |d_i| \le N_i/2,\ N_i \text{ even, or } |d_i| \le (N_i-1)/2,\ N_i \text{ odd} \\ d_i - 2N_i & \text{otherwise} \end{cases} \tag{6.14}$$

  to accommodate negative displacements. Thus, the range of estimates is $[-N_i/2+1,\ N_i/2]$ for N_i even. For example, to estimate displacements within a range [-31,32], the block size should be at least 64 x 64.

The block size is one of the most important parameters in any block-based motion estimation algorithm. Selection of the block size usually involves a tradeoff between two conflicting requirements. The window must be large enough in order to be able to estimate large displacement vectors. On the other hand, it should be small enough so that the displacement vector remains constant within the window. These two contradicting requirements can usually be addressed by hierarchical methods [Erk 93]. Hierarchical methods will be treated for the case of block matching that also faces the same tradeoff in window size selection.

The phase-correlation method has some desirable properties:

Frame-to-frame intensity changes: The phase-correlation method is relatively insensitive to changes in illumination, because shifts in the mean value or multiplication by a constant do not affect the Fourier phase. Since the phase-correlation function is normalized with the Fourier magnitude, the method is also insensitive to any other Fourier-magnitude-only degradation.

Multiple moving *objects:* It is of interest to know what happens when multiple moving objects with different velocities are present within a single window. Experiments indicate that multiple peaks are observed, each indicating the movement of a particular object [Tho 87]. Detection of the significant peaks then generates a list of candidate displacement vectors for each pixel within the block. An additional search is required to find which displacement vector belongs to which pixel within the block. This can be verified by testing the magnitude of the displaced frame difference with each of the candidate vectors.

6.3 Block-Matching Method

Block matching can be considered as the most popular method for practical motion estimation due to its lesser hardware complexity [Jai 81, Ghr 90]. As a result it is widely available in VLSI, and almost all H.261 and MPEG l-2 codecs are utilizing block matching for motion estimation. In block matching, the best motion vector estimate is found by a pixel-domain search procedure.

The basic idea of block matching is depicted in Figure 6.3, where the displacement for a pixel (n_1, n_2) in frame k (the present frame) is determined by considering an N_1 x N_2 block centered about (n_1, n_2), and searching frame $k+1$ (the search frame) for the location of the best-matching block of the same size. The search is

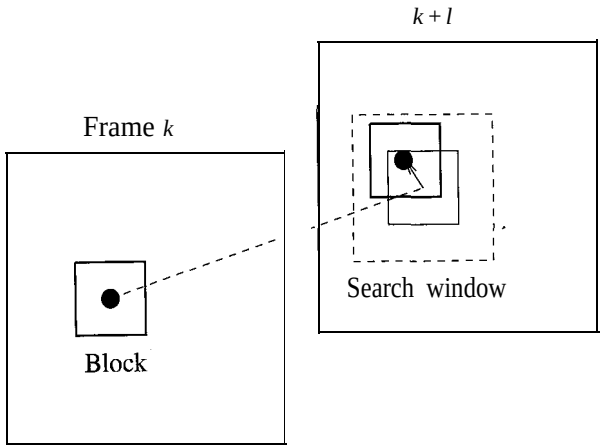


Figure 6.3: Block matching

usually limited to an $N_1 + 2M_1 \times N_2 + 2M_2$ region called the search window for computational reasons.

Block-matching algorithms differ in:

- the matching criteria (e.g., maximum cross-correlation, minimum error)
- the search strategy (e.g., three-step search, cross search), and
- the determination of block size (e.g., hierarchical, adaptive)

We discuss some of the popular options in the following.

6.3.1 Matching Criteria

The matching of the blocks can be quantified according to various criteria including the maximum cross-correlation (similar to the phase-correlation function), the minimum mean square error (MSE), the minimum mean absolute difference (MAD), and maximum matching pel count (MPC).

In the minimum MSE criterion, we evaluate the MSE, defined as

$$MSE(d_1, d_2) = \frac{1}{N_1 N_2} \sum_{(n_1,n_2)\in\mathcal{B}} [s(n_1, n_2, k) - s(n_1 + d_1, n_2 + d_2, k+1)]^2 \quad (6.15)$$

where $\mathcal{B}$ denotes an N_1 x N_2 block, for a set of candidate motion vectors (d_1, d_2). The estimate of the motion vector is taken to be the value of (d_1, d_2) which minimizes

the MSE. That is,

$$[\hat{d}_1\ \hat{d}_2]^T = \arg \min_{(d_1,d_2)} MSE(d_1, d_2) \quad (6.16)$$

Minimizing the MSE criterion can be viewed as imposing the optical flow constraint on all pixels of the block. In fact, expressing the displaced frame difference,

$$dfd_{k,k+1}(n_1, n_2) = s(n_1, n_2, k) - s(n_1 + d_1, n_2 + d_2, k+1) \quad (6.17)$$

in terms of a first order Taylor series (see Section 7.1), it can be easily seen that minimizing the MSE (6.15) is equivalent to minimizing $\mathcal{E}_{of}(\mathbf{v}(\mathbf{x}, t))$ given by (5.13) in the Horn and Schunck method. However, the minimum MSE criterion is not commonly used in VLSI implementations because it is difficult to realize the square operation in hardware.

Instead, the minimum MAD criterion, defined as

$$MAD(d_1, d_2) = \frac{1}{N_1 N_2} \sum_{(n_1,n_2)\in\mathcal{B}} |s(n_1, n_2, k) - s(n_1 + d_1, n_2 + d_2, k+1)| \quad (6.18)$$

is the most popular choice for VLSI implementations. Then the displacement estimate is given by

$$[\hat{d}_1\ \hat{d}_2]^T = \arg \min_{(d_1,d_2)} MAD(d_1, d_2) \quad (6.19)$$

It is well-known that the performance of the MAD criterion deteriorates as the search area becomes larger due to the presence of several local minima.

Another alternative is the maximum matching pel count (MPC) criterion. In this approach, each pel within the block $\mathcal{B}$ is classified as either a matching pel or a mismatching pel according to

$$T(n_1, n_2; d_1, d_2) = \begin{cases} 1 & \text{if } |s(n_1, n_2, k) - s(n_1 + d_1, n_2 + d_2, k+1)| \le t \\ 0 & \text{otherwise} \end{cases} \quad (6.20)$$

where t is a predetermined threshold. Then the number of matching pels within the block is given by

$$MPC(d_1, d_2) = \sum_{(n_1,n_2)\in\mathcal{B}} T(n_1, n_2; d_1, d_2) \quad (6.21)$$

and

$$[\hat{d}_1, \hat{d}_2]^T = \arg \max_{(d_1,d_2)} MPC(d_1, d_2) \quad (6.22)$$

That is, the motion estimate is the value of (d_1, d_2) which gives the highest number of matching pels. The MPC criterion requires a threshold comparator, and a $\log_2(N_1 \times N_2)$ counter [Gha 90].

6.3.2 Search Procedures

Finding the best-matching block requires optimizing the matching criterion over all possible candidate displacement vectors at each pixel (n_1, n_2). This can be accomplished by the so-called "full search," which evaluates the matching criterion for all values of (d_1, d_2) at each pixel, and is extremely time-consuming.

As a first measure to reduce the computational burden, we usually limit the search area to a

$$-M_1 \leq d_1 \leq M_1 \quad \text{and} \quad -M_2 \leq d_2 \leq M_2$$

"search window" centered about each pixel for which a motion vector will be estimated, where M_1 and M_2 are predetermined integers. A search window is shown in Figure 6.3. Another commonly employed practice to lower the computational load is to estimate motion vectors on a sparse grid of pixels, e.g., once every eight pixels and eight lines using a 16×16 block, and to then interpolate the motion field to estimate the remaining motion vectors.

In most cases, however, search strategies faster than the full search are utilized, although they lead to suboptimal solutions. Some examples of faster search algorithms include the

- three-step search and,
- cross-search.

These faster search algorithms evaluate the criterion function only at a predetermined subset of the candidate motion vector locations. Let's note here that the expected accuracy of motion estimates varies according to the application. In motion-compensated compression, all we seek is a matching block, in terms of some metric, even if the match does not correlate well with the actual projected motion. It is for this reason that faster search algorithms serve video compression applications reasonably well.

Three-Step Search

We explain the three-step search procedure with the help of Figure 6.4, where only the search frame is depicted with the search window parameters $M_1 = M_2 = 7$. The "0" marks the pixel in the search frame that is just behind the present pixel. In the first step, the criterion function is evaluated at nine points, the pixel "0" and the pixels marked as "1." If the lowest MSE or MAD is found at the pixel "0," then we have "no motion." In the second step, the criterion function is evaluated at 8 points that are marked as "2" centered about the pixel chosen as the best match in the first stage (denoted by a circled "1"). Note that in the initial step, the search pixels are the corners of the search window, and then at each step we halve the distance of the search pixels from the new center to obtain finer-resolution estimates. The motion estimate is obtained after the third step, where the search pixels are all 1 pixel away from the center.

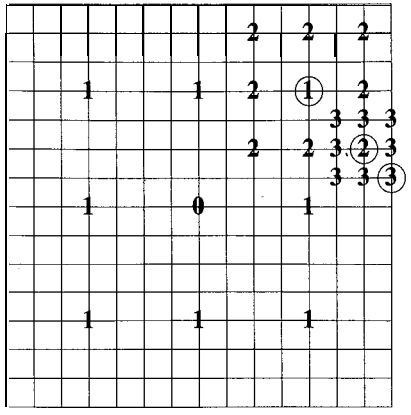


Figure 6.4: Three-step search.

Additional steps may be incorporated into the procedure if we wish to obtain subpixel accuracy in the motion estimates. Note that the search frame needs to be interpolated to evaluate the criterion function at subpixel locations. Generalization of this procedure to other search window parameters yields the so-called "n-step search" or "log-D search" procedures.

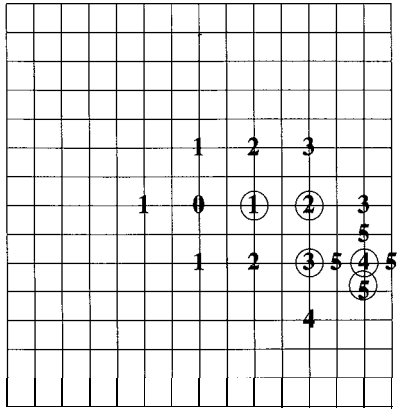


Figure 6.5: Cross-search

Cross-Search

The cross-search method is another logarithmic search strategy, where at each step there are four search locations which are the end points of an (x)-shape cross or a (+)-shape cross [Gha 90]. The case of a (+)-shape cross is depicted in Figure 6.5.

The distance between the search points is reduced if the best match is at the center of the cross or at the boundary of the search window. Several variations of these search strategies exist in the literature [Ghr 90]. Recently, more efficient search algorithms that utilize the spatial and temporal correlations between the motion vectors across the blocks have been proposed [Liu 93]. Block matching has also been generalized to arbitrary shape matching, such as matching contours and curvilinear shapes using heuristic search strategies [Cha 91, Dav 83].

The selection of an appropriate block size is essential for any block-based motion estimation algorithm. There are conflicting requirements on the size of the search blocks. If the blocks are too small, a match may be established between blocks containing similar gray-level patterns which are unrelated in the sense of motion. On the other hand, if the blocks are too large, then actual motion vectors may vary within a block, violating the assumption of a single motion vector per block. Hierarchical block matching, discussed in the next section, addresses these conflicting requirements.

6.4 Hierarchical Mot ion Estimation

Hierarchical (multiresolution) representations of images (frames of a sequence) in the form of a Laplacian pyramid or wavelet transform may be used with both the phase-correlation and block-matching methods for improved motion estimation. A pyramid representation of a single frame is depicted in Figure 6.6 where the full-resolution image (layer-l) is shown at the bottom. The images in the upper levels are lower and lower resolution images obtained by appropriate low-pass filtering and subsampling. In the following, we only discuss the hierarchical block-matching method. A hierarchical implementation of the phase-correlation method follows the same principles.

The basic idea of hierarchical block-matching is to perform motion estimation at each level successively, starting with the lowest resolution level [Bie 88]. The lower resolution levels serve to determine a rough estimate of the displacement using relatively larger blocks. Note that the "relative size of the block" can be measured as the size of the block normalized by the size of the image at a particular resolution level. The estimate of the displacement vector at a lower resolution level is then passed onto the next higher resolution level as an initial estimate. The higher resolution levels serve to fine-tune the displacement vector estimate. At higher resolution levels, relatively smaller window sizes can be used since we start with a good initial estimate.

In practice, we may skip the subsampling step. Then the pyramid contains images that are all the same size but successively more blurred as we go to the

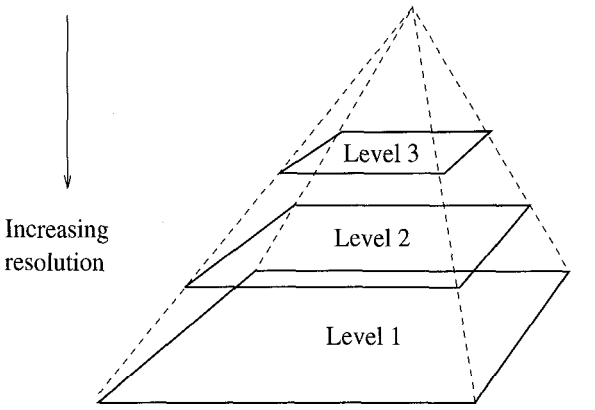


Figure 6.6: Hierarchical image representation.

lower resolution levels. Hierarchical block-matching in such a case is illustrated in Figure 6.7, where the larger blocks are applied to more blurred versions of the image. For simplicity, the low-pass filtering (blurring) may be performed by a box filter which replaces each pixel by a local mean. A typical set of parameters for

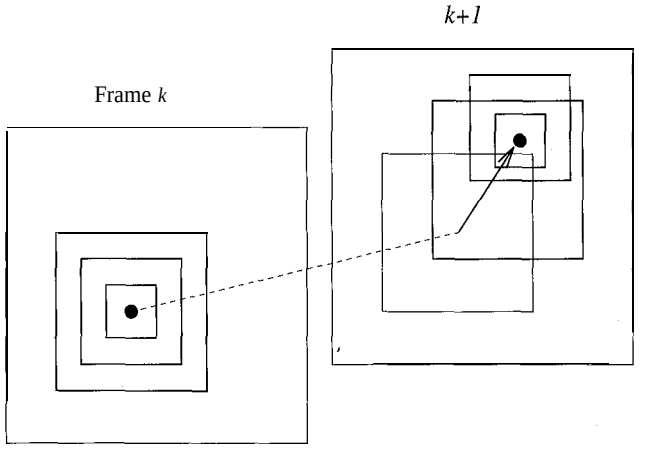


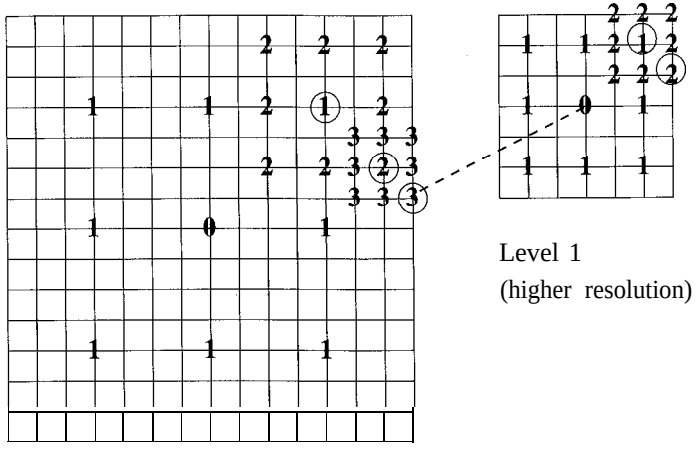
Figure 6.7: Hierarchical block-matching

Table 6.1: Typical set of parameters for 5-Level hierarchical block-matching.

LEVEL:	5	4	3	2	1
Filter Size	10	10	5	5	3
Maximum Displacement	±31	±15	±7	±3	±1
Block Size	64	32	16	8	4

5-level hierarchical block-matching (with no subsampling) is shown in Table 6.1. Here, the filter size refers to the size of a square window used in computing the local mean.

Figure 6.8 illustrates hierarchical block-matching with 2 levels, where the maximum allowed displacement $M = 7$ for level 2 and $M = 3$ for level 1. The best estimate at the lower resolution level (level 2) is indicated by the circled "3." The center of the search area in level 1 (denoted by "0") corresponds to the best estimate from the second level. The estimates in the second and first levels are $[7, 1]^T$ and $[3, 1]^T$, respectively, resulting in an overall estimate of $[10, 2]^T$. Hierarchical block-matching can also be performed with subpixel accuracy by incorporating appropriate interpolation.



Level 1
(higher resolution)

Level 2 (lower resolution)

Figure 6.8: Example of hierarchical block-matching with 2 levels.

6.5 Generalized Block-Motion Estimation

While motion estimation based on the translatory block model is simple, it deals poorly with rotations and deformations of blocks from frame to frame, as well as discontinuities in the motion field. We discuss two approaches in the following for improved motion tracking and compensation using block-based methods: a postprocessing approach, and a generalized block-matching approach using spatial transformations.

6.5.1 Postprocessing for Improved Motion Compensation

Block-based motion representation and compensation have been adopted in international standards for video compression, such as H.261, MPEG-1, and MPEG-2, where a single motion vector is estimated for each 16 x 16 square block, in order to limit the number of motion vectors that need to be transmitted. However, this low-resolution representation of the motion field results in inaccurate compensation and visible blocking artifacts, especially at the borders of the blocks.

To this effect) Orchard [Orc 93] proposed a postprocessing method to reconstruct a higher-resolution motion field based on image segmentation. In this method, a single motion vector is estimated per block, as usual, in the first pass. Next, image blocks are segmented into K regions such that each region is represented by a single motion vector. To avoid storage/transmission of additional motion vectors, the K candidate motion vectors are selected from a set of already estimated motion vectors for the neighboring blocks. Then, the motion vector that minimizes the DFD at each pixel of the block is selected among the set of K candidate vectors. A predictive MAP segmentation scheme has been proposed to avoid transmitting overhead information about the boundaries of these segments, where the decoder/receiver can duplicate the segmentation process. As a result, this method allows for motion compensation using a pixel-resolution motion field, while transmitting the same amount of motion information as classical block-based methods.

6.5.2 Deformable Block Matching

In deformable (or generalized) block matching, the current frame is divided into triangular, rectangular, or arbitrary quadrilateral patches. We then search for the best matching triangle or quadrilateral in the search frame under a given spatial transformation. This is illustrated in Figure 6.9. The choice of patch shape and the spatial transformation are mutually related. For example, triangular patches offer sufficient degrees of freedom (we have two equations per node) with the affine transformation, which has only six independent parameters. Perspective and bilinear transformations have eight free parameters. Hence, they are suitable for use with rectangular or quadrilateral patches. Note that using affine transformation with quadrilateral patches results in an overdetermined motion estimation problem in that affine transformation preserves parallel lines (see Figure 6.2).

The spatial transformations (6.4), (6.5), and (6.6) clearly provide superior motion tracking and rendition, especially in the presence of rotation and zooming, compared to the translational model (6.3). However, the complexity of the motion estimation increases significantly. If a search-based motion estimation strategy is adopted, we now have to perform a search in a 6- or 8-D parameter space (affine or perspective/bilinear transformation) instead of a 2-D space (the x_1 and x_2 components of the translation vector). Several generalized motion estimation schemes have been proposed, including a full-search method [Sef 93], a faster hexagonal search [Nak 94], and a global spline surface-fitting method using a fixed number of point correspondences [Flu92].

The full-search method can be summarized as follows [Sef 93]:

1. Segment the current frame into rectangular (triangular) blocks.
2. Perturb the coordinates of the corners of a matching quadrilateral (triangle) in the search frame starting from an initial guess.
3. For each quadrilateral (triangle), find the parameters of a prespecified spatial transformation that maps this quadrilateral (triangle) onto the rectangular (triangular) block in the current frame using the coordinates of the four (three) matching corners.
4. Find the coordinates of each corresponding pixel within the quadrilateral (triangle) using the computed spatial transformation, and calculate the MSE between the given block and the matching patch.
5. Choose the spatial transformation that yields the smallest MSE or MAD.

In order to reduce the computational load imposed by generalized block-matching, generalized block-matching is only used for those blocks where standard block-matching is not satisfactory. The displaced frame difference resulting from standard block matching can be used as a decision criterion. The reader is referred to [Flu92] and [Nak 94] for other generalized motion estimation strategies.

The generalized block-based methods aim to track the motion of all pixels within a triangular or quadrilateral patch using pixel correspondences established at the corners of the patch. Thus, it is essential that the current frame is segmented into triangular or rectangular patches such that each patch contains pixels only from a single moving object. Otherwise, local motion within these patches cannot be tracked using a single spatial transformation. This observation leads us to feature- and/or motion-based segmentation of the current frame into adaptive meshes (also known as irregular-shaped meshes). Regular and adaptive mesh models are depicted in Figure 6.10 for comparison.

There are, in general, two segmentation strategies: i) Feature-based segmentation: The edges of the patches are expected to align sufficiently closely with gray-level and/or motion edges. Ideally, adaptive mesh fitting and motion tracking should be performed simultaneously, since it is not generally possible to isolate motion rendition errors due to adaptive mesh misfit and motion estimation. A simultaneous optimization method was recently proposed by Wang et al. [Wan 94].

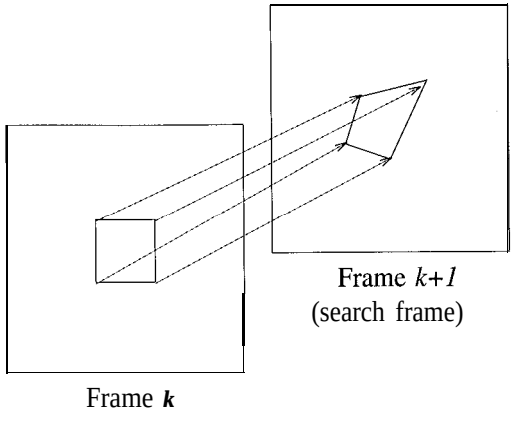


Figure 6.9: Generalized block-matching.

ii) Hierarchical segmentation: It is proposed to start with an initial coarse mesh, and perform a synthesis of the present frame based on this mesh. Then this mesh is refined by successive partitioning of the patches where the initial synthesis error is above a threshold. A method based on the hierarchical approach was proposed by Flusser [Flu92].

Fitting and tracking of adaptive mesh models is an active research area at present. Alternative strategies for motion-based segmentation (although not necessarily into mesh structures) and parametric motion model estimation will be discussed in Chapter 11.

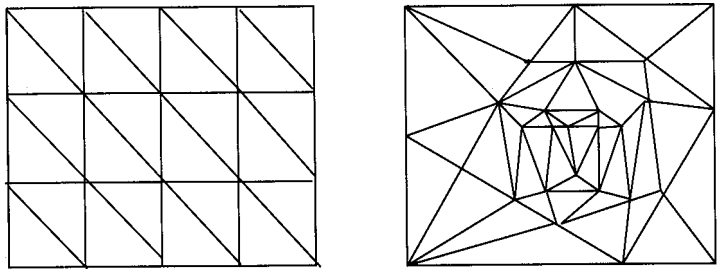


Figure 6.10: a) Regular and b) adaptive mesh models fitted to frame k.

6.6 Examples

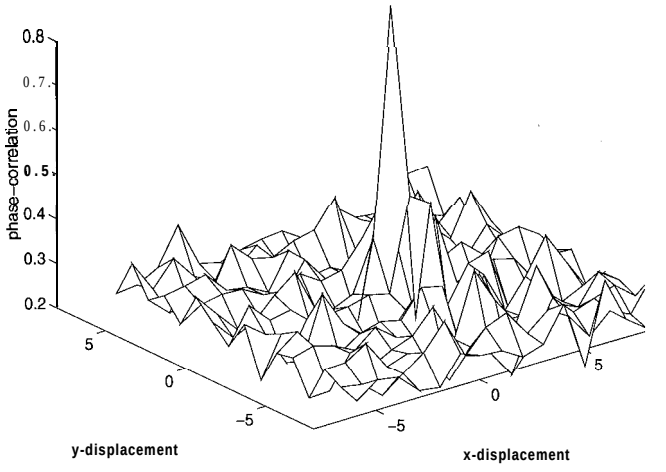
We have evaluated four algorithms, phase correlation, block matching (BM), hierarchical BM, and the generalized BM, using the same two frames of the Mobile and Calendar sequence shown in Figure 5.9 (a) and (b).

An example of the phase-correlation function, computed on a 16 x 16 block on the calendar, using a 32 x 32 DFT with appropriate zero padding, which allows for a maximum of ± 8 for each component of the motion vector, is plotted in Figure 6.11 (a). The impulsive nature of the function can be easily observed. Figure 6.11 (b) depicts the motion estimates obtained by applying the phase-correlation method with the same parameters centered about each pixel. In our implementation of the phase-correlation method, the motion vector corresponding to the highest peak for each block has been retained. We note that improved results could be obtained by postprocessing of the motion estimates. For example, we may consider two or three highest peaks for each block and then select the vector yielding the smallest displaced frame difference at each pixel,

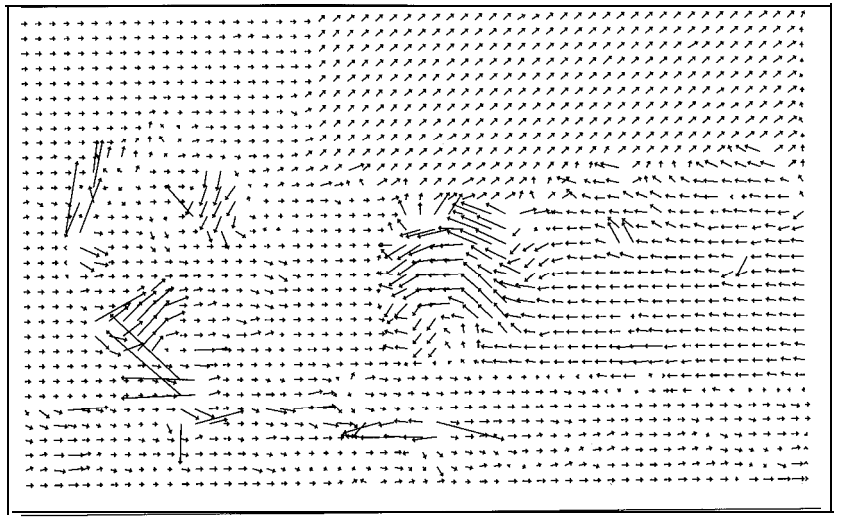
The BM and 3-level hierarchical BM (HBM) methods use the mean-squared error measure and the three-step search algorithm. The resulting motion fields are shown in Figure 6.12 (a) and (b), respectively. The BM algorithm uses 16 x 16 blocks and 3 x 3 blurring for direct comparison with the phase-correlation method. In the case of the HBM, the parameter values given in Table 6.1 (for the first three levels) have been employed, skipping the subsampling step. The generalized BM algorithm has been applied only to those pixels where the displaced frame difference (DFD) resulting from the S-level HBM is above a threshold. The PSNR of the DFD and the entropy of the estimated motion fields are shown in Table 6.2 for all four methods. The 3-level HBM is prefered over the generalized BM algorithm, since the latter generally requires an order of magnitude more computation.

Table 6.2: Comparison of the block-based methods. (Courtesy Gozde Bozdagi)

Method	PSNR (dB)	Entropy (bits)
Frame-difference	23.45	-
Phase correlation	32.70	5.64
Block matching (BM)	29.76	4.62
Hierarchical BM	36.29	7.32
Generalized BM	36.67	7.31

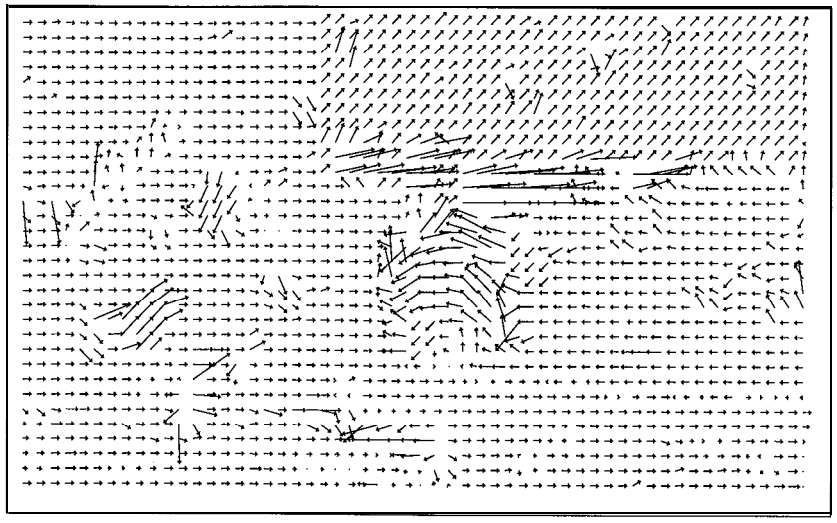


(a)

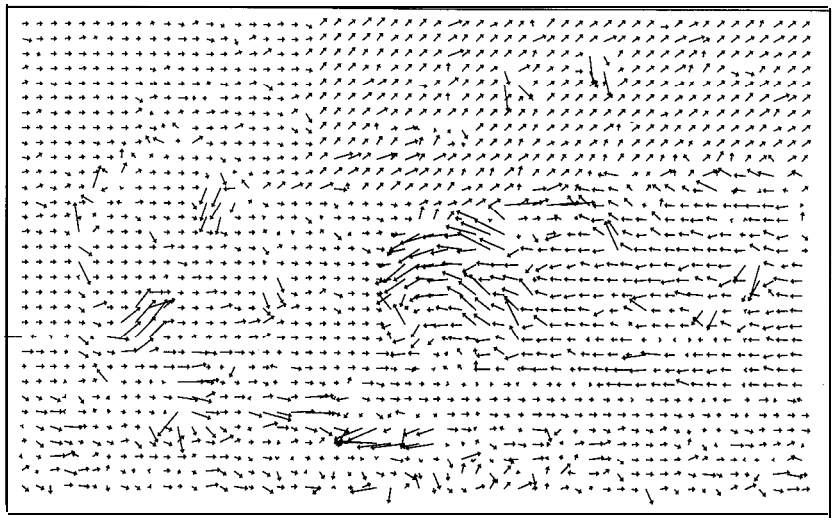


(b)

Figure 6.11: a) The phase-correlation function, and b) the motion field obtained by the phase-correlation method. (Courtesy Gozde Bozdagi)



(a)



(b)

Figure 6.12: The motion field obtained by a) the block matching and b) the hierarchical block-matching methods. (Courtesy Mehmet Ozkan and Gozde Bozdagi)

6.7 Exercises

1. How do you deal with the boundary effects in the phase-correlation method? Can we use the discrete cosine transform (DCT) instead of the DFT?

2. Suggest a model to quantify the spectral leakage due to subpixel motion in the phase-correlation method.

3. State the symmetry properties of the DFT for N even and odd, respectively. Verify Equation (6.14).

4. Discuss the aperture problem for the cases of i) single pixel matching, ii) line matching, iii) curve matching, and iv) corner matching.

5. Compare the computational complexity of full search versus i) the three-step search, and ii) search using integral projections [Kim 92].

6. Propose a method which uses the optical flow equation (5.5) for deformable block matching with the affine model. (Hint: Review Section 5.3.3.)

Bibliography

[Bie 88] M. Bierling, "Displacement estimation by hierarchical block-matching," *Proc.* Visual Comm. and Image *Proc.*, SPIE vol. 1001, pp. 942–951, 1988.

[Cas 87] E. De Castro and C. Morandi, "Registration of translated and rotated images using finite Fourier transforms," *IEEE* Trans. *Patt.* Anal. *Mach.* Mel., vol. 9, no. 5, pp. 700-703, Sep. 1987.

[Cha 91] S. Chaudhury, S. Subramanian, and G. Parthasarathy, "Heuristic search approach to shape matching in image sequences," *IEE Proc.-E*, vol. 138, no. 2, pp. 97–105, 1991.

[Dav 83] L. S. Davis, Z. Wu, and H. Sun, "Contour-based motion estimation," *Comput. Vis.* Graphics Image *Proc.*, vol. 23, pp. 313-326, 1983.

[Erk 93] Y. M. Erkam, M. I. Sezan, and A. T. Erdem, "A hierarchical phase-correlation method for motion estimation," *Proc. Conf.* on Info. *Scien.* and Systems, Baltimore MD, Mar. 1993, pp. 419-424.

[Flu92] J. Flusser, "An Adaptive Method for Image Registration," *Patt. Recog.*, vol. 25, pp. 45–54, 1992.

[Gha 90] M. Ghanbari, "The cross-search algorithm for motion estimation," *IEEE* Trans. Commun., vol. 38, pp. 950-953, 1990.

[Ghr 90] H. Gharavi and M. Mills, "Block-matching motion estimation algorithms: New results," *IEEE Trans. Circ.* and *Syst.*, vol. 37, pp. 6499651, 1990.

[Hee 87] D. J. Heeger, "Model for the extraction of image flow," *J. Opt. Soc. Am. A*, vol. 4, no. 8, pp. 145551471, Aug. 1987.

[Jac 87] L. Jacobson and H. Wechsler, "Derivation of optical flow using a spatio-temporal frequency approach," *Comp. Vision* **Graph.** *Image Proc.*, vol. 38, pp. 57-61, 1987.

[Jai 81] J. R. Jain and A. K. Jain, "Displacement measurement and its application in interframe image coding," *IEEE Trans. Commun.*, vol. 29, pp. 1799-1808, 1981.

[Kim 92] J.-S. Kim and R.-H. Park, "A fast feature-based block-matching algorithm using integral projections," *IEEE J. Selected* **Areas in Comm.,** vol. 10, pp. 968-971, June 1992.

[Liu 93] B. Liu and A. Zaccarin, "New fast algorithms for the estimation of block motion vectors," *IEEE* Trans. *Circ. and Syst. Video Tech.*, vol. 3, no. 2, pp. 148-157, Apr. 1993.

[Nak 94] Y. Nakaya and H. Harashima, "Motion compensation based on spatial transformations," *IEEE Trans. CAS Video Tech.*, vol. 4, pp. 339-356, June 1994.

[Orc 93] M. Orchard, "Predictive motion-field segmentation for image sequence coding," *IEEE* Trans. *Circ. and Syst: Video Tech.*, vol. 3, pp. 54–70, Feb. 1993.

[Sef 93] V. Seferidis and M. Ghanbari, "General approach to block-matching motion estimation," Optical *Engineering*, vol. 32, pp. 1464–1474, July 1993.

[Sul 93] G. Sullivan, "Multi-hypothesis motion compensation for low bit-rate video coding," *Proc. IEEE Int. Conf. ASSP*, Minneapolis, MN, vol. 5, pp. 437-440, 1993.

[Tho 87] G. A. Thomas and B. A. Hons, "Television motion measurement for DATV and other applications," Tech. Rep. BBC-RD-1987-11, 1987.

[Wan 94] Y. Wang and 0. Lee, "Active mesh: A feature seeking and tracking image sequence representation scheme," *IEEE* Trans. Image *Proc.*, vol. 3, pp. 610-624, Sep. 1994.

[Wol 90] G. Wolberg, *Digital* Image *Warping*, Los Alamitos, CA: IEEE Comp. Soc. Press, 1990.

Chapter 7

PEL-RECURSIVE METHODS

All motion estimation methods, in one form or another, employ the optical flow constraint accompanied by some smoothness constraint. Pel-recursive methods are predictor-corrector-type estimators, of the form

$$\hat{\mathbf{d}}_a(\mathbf{x}, t; \text{At}) = \hat{\mathbf{d}}_b(\mathbf{x}, \text{t}; \text{At}) + \mathbf{u}(\mathbf{x}, t; \text{At}) \tag{7.1}$$

where $\hat{\mathbf{d}}_a(\mathbf{x}, t; \text{At})$ denotes the estimated motion vector at the location x and time t, $\mathbf{d}_b(\mathbf{x}, t; \text{At})$ denotes the predicted motion estimate, and $\mathbf{u}(\mathbf{x}, t; \text{At})$ is the update term. The subscripts "a" and "b" denote after and before the update at the pel location $(\mathbf{x}, t)$. The prediction step, at each pixel, imposes a local smoothness constraint on the estimates, and the update step enforces the optical flow constraint.

The estimator (7.1) is usually employed in a recursive fashion, by performing one or more iterations at (x, t) and then proceeding to the next pixel in the direction of the scan; hence the name pel-recursive. Early pel-recursive approaches focused on ease of hardware implementation and real-time operation, and thus employed simple prediction and update equations [Rob 83]. Generally, the best available estimate at the previous pel was taken as the predicted estimate for the next pel, followed by a single gradient-based update to minimize the square of the displaced frame difference at that pel. Later, more sophisticated prediction and update schemes that require more computation were proposed [Wal 84, Bie 87, Dri 91, Bor 91].

We start this chapter with a detailed discussion of the relationship between the minimization of the displaced frame difference and the optical flow constraint in Section 7.1. We emphasize that the update step, which minimizes the displaced frame difference at the particular pixel location, indeed enforces the optical flow equation (constraint) at that pixel. Section 7.2 provides an overview of some gradient-based minimization methods that are an integral part of basic pel-recursive methods. Basic pel-recursive methods are presented in Section 7.3. An extension of these methods, called Wiener-based estimation, is covered in Section 7.4.

7.1 Displaced Frame Difference

The fundamental principle in almost all motion estimation methods, known as the optical flow constraint, is that the image intensity remains unchanged from frame to frame along the true motion path (or changes in a known or predictable fashion). The optical flow constraint may be employed in the form of the optical flow equation (5.5), as in Chapter 5, or may be imposed by minimizing the displaced frame difference (6.15), as in block-matching and pel-recursive methods. This section provides a detailed description of the relationship between the minimization of the displaced frame difference (DFD) and the optical flow equation (OFE).

Let the DFD between the time instances t and $t' = t + \Delta t$ be defined by

$$dfd(\mathbf{x}, \mathbf{d}) \doteq s_c(\mathbf{x} + \mathrm{d}(\mathbf{x}, t; \mathbf{At}), t + \mathrm{At}) - s_c(\mathbf{x}, \mathrm{t}) \tag{7.2}$$

where $s_c(x_1, x_2, t)$ denotes the time-varying image distribution, and

$$\mathrm{d}(\mathbf{x}, t; \mathrm{At}) \doteq \mathbf{d(x)} = [d_1(\mathbf{x})\ d_2(\mathbf{x})]^T$$

denotes the displacement vector field between the times t and $t + \Delta t$. We observe that i) if the components of $\mathbf{d(x)}$ assume noninteger values, interpolation is required to compute the DFD at each pixel location; and ii) if $\mathbf{d(x)}$ were equal to the true displacement vector at site x and there were no interpolation errors, the DFD attains the value of zero at that site under the optical flow constraint.

Next, we expand $s_c(\mathbf{x} + \mathbf{d(x)}, t + \mathrm{At})$ into a Taylor series about $(\mathbf{x}; t)$, for $\mathbf{d(x)}$ and At small. as

$$s_c(x_1 + d_1(\mathbf{x}), x_2 + d_2(\mathbf{x}); t + \mathrm{At}) = s_c(\mathbf{x}; t) + d_1(\mathbf{x})\frac{\partial s_c(\mathbf{x}; t)}{\partial x_1} + d_2(\mathbf{x})\frac{\partial s_c(\mathbf{x}; t)}{\partial x_2} + \Delta t\frac{\partial s_c(\mathbf{x}; t)}{\partial t} + h.o.t. \tag{7.3}$$

Substituting (7.3) into (7.2), and neglecting the higher-order terms $(h.o.t.)$,

$$dfd(\mathbf{x}, \mathbf{d}) = \frac{\partial s_c(\mathbf{x}; t)}{\partial x_1} d_1(\mathbf{x}) + \frac{\partial s_c(\mathbf{x}; t)}{\partial x_2} d_2(\mathbf{x}) + \Delta t\frac{\partial s_c(\mathbf{x}; t)}{\partial t} \tag{7.4}$$

We investigate the relationship between the DFD and OFE in two cases:

1. *Limit* At approaches 0: Setting $dfd(\mathbf{x}, \mathbf{d}) = 0$, dividing both sides of (7.4) by At, and taking the limit as At approaches 0, we obtain the OFE

$$\frac{\partial s_c(\mathbf{x}; t)}{\partial x_1} v_1(\mathbf{x}, t) + \frac{\partial s_c(\mathbf{x}; t)}{\partial x_2} v_2(\mathbf{x}, t) + \frac{\partial s_c(\mathbf{x}; t)}{\partial t} = 0 \tag{7.5}$$

where $\mathbf{v}(\mathbf{x}, t) = [v_1(\mathbf{x}, t)\ v_2(\mathbf{x}, t)]^T$ denotes the velocity vector at time t. That is, velocity estimation using the OFE and displacement estimation by setting the DFD equal to zero are equivalent in the limit At goes to zero.

2. For At finite: An estimate of the displacement vector $\hat{\mathbf{d}}(\mathbf{x})$ between any two frames that are At apart can be obtained from (7.4) in a number of ways:

   (a) Search for $\hat{\mathbf{d}}(\mathbf{x})$ which would set the left-hand side of (7.4), given by (7.2), to zero over a block of pixels (block-matching strategy).

   (b) Compute $\hat{\mathbf{d}}(\mathbf{x})$ which would set the left-hand side of (7.4) to zero on a pixel-by-pixel basis using a gradient-based optimization scheme (pel-recursive strategy).

   (c) Set $\mathrm{At} = 1$ and $dfd(\mathbf{x}, \hat{\mathbf{d}}) = 0$; solve for $\hat{\mathbf{d}}(\mathbf{x})$ using a set of linear equations obtained from the right-hand side of (7.4) using a block of pixels.

   All three approaches can be shown to be identical if i) local variation of the spatio-temporal image intensity is linear, and ii) velocity is constant within the time interval At; that is,

$$d_1(\mathbf{x}) = \hat{v}_1(\mathbf{x}, t)\Delta t \quad \text{and} \quad d_2(\mathbf{x}) = \hat{v}_2(\mathbf{x}, t)\Delta t$$

In practice, the DFD, $dfd(\mathbf{x}, \mathbf{d})$, hardly ever becomes exactly zero for any value of $\mathbf{d(x)}$, because: i) there is observation noise, ii) there is occlusion (covered/uncovered background problem), iii) errors are introduced by the interpolation step in the case of noninteger displacement vectors, and iv) scene illumination may vary from frame to frame. Therefore, we generally aim to minimize the absolute value or the square of the dfd (7.2) or the left-hand side of the OFE (7.5) to estimate the 2-D motion field. The pel-recursive methods presented in this chapter employ gradient-based optimization techniques to minimize the square of the dfd (as opposed to a search method used in block matching) with an implicit smoothness constraint in the prediction step.

7.2 Gradient-Based Optimization

The most straightforward way to minimize a function $f(u_1, \ldots, u_n)$ of several unknowns is to calculate its partials with respect to each unknown, set them equal to zero, and solve the resulting equations

$$\frac{\partial f(\mathbf{u})}{\partial u_1} = 0$$

$$\frac{\partial f(\mathbf{u})}{\partial u_n} = 0 \tag{7.6}$$

simultaneously for $u_1, \ldots, u_n$. This set of simultaneous equations can be expressed as a vector equation,

$$\nabla_{\mathbf{u}} f(\mathbf{u}) = \mathbf{0} \tag{7.7}$$

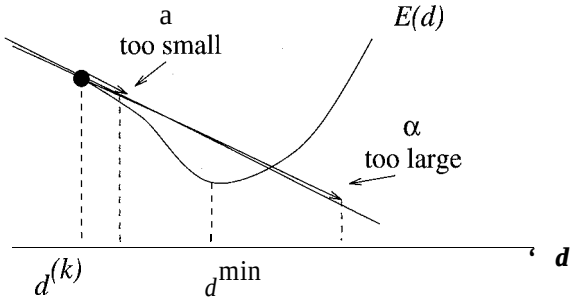


Figure 7.1: An illustration of the gradient descent method

where $\nabla_{\mathbf{u}}$ is the gradient operator with respect to the unknown vector $\mathbf{u}$. Because it is difficult to define a closed-form criterion function $f(u_1, , u_n)$ for motion estimation, and/or to solve the set of equations (7.7) in closed form, we resort to iterative (numerical) methods. For example, the DFD is a function of pixel intensities which cannot be expressed in closed form.

7.2.1 Steepest-Descent Method

Steepest descent is probably the simplest numerical optimization method. It updates the present estimate of the location of the minimum in the direction of the negative gradient, called the steepest-descent direction. Recall that the gradient vector points in the direction of the maximum. That is, in one dimension (function of a single variable), its sign will be positive on an "uphill" slope. Thus, the direction of steepest descent is just the opposite direction, which is illustrated in Figure 7.1.

In order to get closer to the minimum, we update our current estimate as

$$\mathbf{u}^{(k+1)} = \mathbf{u}^{(k)} - \alpha \nabla_{\mathbf{u}} f(\mathbf{u})|_{\mathbf{u}^{(k)}} \tag{7.8}$$

where a is some positive scalar, known as the step size. The step size is critical for the convergence of the iterations, because if α is too small, we move by a very small amount each time, and the iterations will take too long to converge. On the other hand, if it is too large the algorithm may become unstable and oscillate about the minimum. In the method of steepest descent, the step size is usually chosen heuristically.

7.2.2 Newton-Raphson Method

The optimum value for the step size α can be estimated using the well-known Newton-Raphson method for root finding. Here, the derivation for the case of

a function of a single variable is shown for simplicity. In one dimension, we would like to find a root of $f'(u)$. To this effect, we expand f'(u) in a Taylor series about the point $u^{(k)}$ to obtain

$$f'(u^{(k+1)}) = f'(u^{(k)}) + (u^{(k+1)} - u^{(k)}) f''(u^{(k)}) \tag{7.9}$$

Since we wish $u^{(k+1)}$ to be a zero of $f'(u)$, we set

$$f'(u^{(k)}) + (u^{(k+1)} - u^{(k)}) f''(u^{(k)}) = 0 \tag{7.10}$$

Solving (7.10) for $u^{(k+1)}$, we have

$$u^{(k+1)} = u^{(k)} - \frac{f'(u^{(k)})}{f''(u^{(k)})} \tag{7.11}$$

This result can be generalized for the case of a function of several unknowns as

$$\mathbf{u}^{(k+1)} = \mathbf{u}^{(k)} - \mathbf{H}^{-1} \nabla_{\mathbf{u}} f(\mathbf{u})|_{\mathbf{u}^{(k)}} \tag{7.12}$$

where **H** is the Hessian matrix

$$\mathbf{H}_{ij} = \left[\frac{\partial^2 f(\mathbf{u})}{\partial u_i \partial u_j} \right]$$

The Newton-Raphson method finds an analytical expression for the step-size parameter in terms of the second-order partials of the criterion function. When a closed-form criterion function is not available, the Hessian matrix can be estimated by using numerical methods [Fle 87].

7.2.3 Local vs. Global Minima

The gradient descent approach suffers from a serious drawback: the solution depends on the initial point. If we start in a "valley," it will be stuck at the bottom of that valley, even if it is a "local" minimum. Because the gradient vector is zero or near zero, at or around a local minimum, the updates become too small for the method to move out of a local minimum. One solution to this problem is to initialize the algorithm at several different starting points, and then pick the solution that gives the smallest value of the criterion function.

More sophisticated optimization methods, such as simulated annealing, exist in the literature to reach the global minimumregardless of the starting point. However, these methods usually require significantly more processing time. We will discuss simulated annealing techniques in detail in the next chapter.

7.3 Steepest-Descent-Based Algorithms

Pel-recursive motion estimation is usually preceeded by a change detection stage, where the frame difference at each pixel is tested against a threshold. Estimation

is performed only at those pixels belonging to the changed region. The steepest-descent-based pel-recursive algorithms estimate the update term $\mathbf{u}(\mathbf{x}, t; \mathrm{At})$ in (7.1), at each pel in the changed region, by minimizing a positive-definite function E of the frame difference dfd with respect to d. The function E needs to be positive definite, such as the square function, so that the minimum occurs when the dfd is zero. The dfd converges to zero locally, when $\hat{\mathbf{d}}_a(\mathbf{x}, t; \mathrm{At})$ converges to the actual displacement. Therefore, the update step corresponds to imposing the optical flow constraint locally. In the following, we present some variations on the basic steepest-descent-based pel-recursive estimation scheme.

7.3.1 Netravali-Robbins Algorithm

The Netravali-Robbins algorithm finds an estimate of the displacement vector, which minimizes the square of the DFD at each pixel, using a gradient descent method. Then the criterion function to be minimized is given by

$$E(\mathbf{x}; \mathbf{d}) = [dfd(\mathbf{x}, \mathrm{d})]^2 \tag{7.13}$$

From Section 7.2, minimization of $E(\mathbf{x}; \mathbf{d})$ with respect to d, at pixel $\mathbf{x}$, by the steepest descent method yields the iteration

$$\begin{aligned}\hat{\mathbf{d}}^{i+1}(\mathbf{x}) &= \hat{\mathbf{d}}^i(\mathbf{x}) - (1/2)\epsilon\ \nabla_{\mathbf{d}}[dfd(\mathbf{x}, \mathbf{d})|_{\mathbf{d}=\hat{\mathbf{d}}^i}]^2 \\ &= \hat{\mathbf{d}}^i(\mathbf{x}) - \epsilon\, dfd(\mathbf{x}, \hat{\mathbf{d}}^i)\, \nabla_{\mathbf{d}} dfd(\mathbf{x}, \mathbf{d})|_{\mathbf{d}=\hat{\mathbf{d}}^i}\end{aligned} \tag{7.14}$$

where V is the gradient with respect to d, and ϵ is the step size. Recall that the negative gradient points to the direction of steepest descent.

We now discuss the evaluation of $\nabla_{\mathbf{d}} dfd(\mathbf{x}, \mathbf{d})$. From (7.2), we can write

$$dfd(\mathbf{x}, \mathbf{d}) - dfd(\mathbf{x}, \hat{\mathbf{d}}^i) = s_c(\mathbf{x} + \mathbf{d}, t + \mathrm{At}) - s_c(\mathbf{x} + \hat{\mathbf{d}}^i, t + \mathrm{At}) \tag{7.15}$$

Now, expanding the intensity $s_c(\mathbf{x} + \mathrm{d}, t + \mathrm{At})$ at an arbitrary point x + d into a Taylor series about $\mathrm{x} + \hat{\mathbf{d}}^i$, we have

$$\begin{aligned}s_c(\mathbf{x} + \mathrm{d}, t + \Delta t) &= s_c(\mathbf{x} + \hat{\mathbf{d}}^i, t + \Delta t) + \\ &(\mathbf{d} - \hat{\mathbf{d}}^i)^T \nabla_{\mathbf{x}} s_c(\mathbf{x} - \mathbf{d}; t - \Delta t)|_{\mathbf{d}=\hat{\mathbf{d}}^i} + o(\mathbf{x}, \hat{\mathbf{d}}^i)\end{aligned} \tag{7.16}$$

where $o(\mathbf{x}, \hat{\mathbf{d}}^i)$ denotes the higher-order terms in the series. Substituting the Taylor series expansion into (7.15), we obtain the linearized DFD expression

$$dfd(\mathbf{x}, \mathbf{d}) = dfd(\mathbf{x}, \hat{\mathbf{d}}^i) + \nabla_{\mathbf{x}}^T s_c(\mathbf{x} - \hat{\mathbf{d}}^i; t - \mathrm{At})(\mathbf{d} - \mathbf{ii}) + o(\mathbf{x}, \hat{\mathbf{d}}^i) \tag{7.17}$$

where $\nabla_{\mathbf{x}} s_c(\mathbf{x} - \hat{\mathbf{d}}^i; t - \mathrm{At}) \doteq \nabla_{\mathbf{x}} s_c(\mathbf{x} - \mathbf{d}; t - \Delta t)|_{\mathbf{d}=\hat{\mathbf{d}}^i}$.

Using (7.17) and ignoring the higher-order terms, we can express the gradient of the DFD with respect to $\mathbf{d}$ in terms of the spatial gradient of image intensity as

$$\nabla_{\mathbf{d}} dfd(\mathbf{x}, \mathrm{d})|_{\mathbf{d}=\hat{\mathbf{d}}^i} = \nabla_{\mathbf{x}} s_c(\mathbf{x} - \hat{\mathbf{d}}^i; t - \mathrm{At}) \tag{7.18}$$

and the pel-recursive estimator becomes

$$\hat{\mathbf{d}}^{i+1}(\mathbf{x}) = \hat{\mathbf{d}}^i(\mathbf{x}) - \epsilon\, dfd(\mathbf{x}, \hat{\mathbf{d}}^i)\, \nabla_{\mathbf{x}} s_c(\mathbf{x} - \hat{\mathbf{d}}^i; t - \mathrm{At}) \tag{7.19}$$

In (7.19), the first and second terms are the prediction and update terms, respectively. Note that the evaluation of the frame difference $dfd(\mathbf{x}, \hat{\mathbf{d}}^i)$ and the spatial gradient vector may require interpolation of the intensity value for noninteger displacement estimates.

The aperture problem is also apparent in the pel-recursive algorithms. The update term is a vector along the spatial gradient of the image intensity. Clearly, no correction is performed in the direction perpendicular to the gradient vector.

In an attempt to further simplify the structure of the estimator, Netravali and Robbins also proposed the modified estimation formula

$$\hat{\mathbf{d}}^{i+1}(\mathbf{x}) = \hat{\mathbf{d}}^i(\mathbf{x}) - \epsilon\, sgn\{dfd(\mathbf{x}, \hat{\mathbf{d}}^i)\} sgn\{\nabla_{\mathbf{x}} s_c(\mathbf{x} - \hat{\mathbf{d}}^i; t - \mathrm{At})\} \tag{7.20}$$

where the update term takes one of the three values, $\pm\epsilon$ and zero. In this case, the motion estimates are updated only in 0, 45, 90, 135, degrees directions.

The convergence and the rate of convergence of the Netravali-Robbins algorithm depend on the choice of the step size parameter ϵ. For example, if $\epsilon = 1/16$, then at least 32 iterations are required to estimate a displacement by 2 pixels. On the other hand, a large choice for the step size may cause an oscillatory behavior. Several strategies can be advanced to facilitate faster convergence of the algorithm.

7.3.2 Walker-Rao Algorithm

Walker and Rao [Wal 84] proposed an adaptive step size motivated by the following observations: i) In the neighborhood of an edge where $|\nabla s_c(x_1, x_2, t)|$ is large, ϵ should be small if the DFD is small, so that we do not make an unduly large update. Furthermore, in the neighborhood of an edge, the accuracy of motion estimation is vitally important, which also necessitates a small step size. ii) In uniform image areas where $|\nabla s_c(x_1, x_2, t)|$ is small, we need a large step size when the DFD is large. Both of these requirements can be met by a step size of the form

$$\epsilon = \frac{1}{2\,||\nabla_{\mathbf{x}} s_c(\mathbf{x} - \hat{\mathbf{d}}^i; t - \Delta t)||^2} \tag{7.21}$$

In addition, Walker and Rao have introduced the following heuristic rules:

1. If the DFD is less than a threshold, the update term is set equal to zero.
2. If the DFD exceeds the threshold, but the magnitude of the spatial image gradient is zero, then the update term is again set equal to zero.
3. If the absolute value of the update term (for each component) is less than 1/16, then it is set equal to $\pm 1/16$.

4. If the absolute value of the update term (for each component) is more than 2, then it is set equal to ± 2.

Caffario and Rocca have also developed a similar step-size expression [Caf 83]

$$\epsilon = \frac{1}{||\nabla_{\mathbf{x}} s_c(\mathbf{x} - \hat{\mathbf{d}}^i; t - \Delta t)||^2 + \eta^2} \tag{7.22}$$

which includes a bias term η^2 to avoid division by zero in areas of constant intensity where the spatial gradient is almost zero. A typical value for $\eta^2 = 100$.

Experimental results indicate that using an adaptive step size greatly improves the convergence of the Netravali-Robbins algorithm. It has been found that five iterations were sufficient to achieve satisfactory results in most cases.

7.3.3 Extension to the Block Motion Model

It is possible to impose a stronger regularity constraint on the estimates at each pixel $\mathbf{x}$, by assuming that the displacement remains constant locally over a sliding support $\mathcal{B}$ about each pixel. We can, then, minimize the DFD over the support $\mathcal{B}$, defined as

$$E(\mathbf{x}, \mathbf{d}) = \sum_{\mathbf{x}_B \in \mathcal{B}} [dfd(\mathbf{x}_B, \mathbf{d})]^2 \tag{7.23}$$

as opposed to on a pixel-by-pixel basis. Notice that the support $\mathcal{B}$ needs to be "causal" (in the sense of recursive computability) in order to preserve the pel-recursive nature of the algorithm. A typical causal support with N = 7 pixels is shown in Figure 7.2.

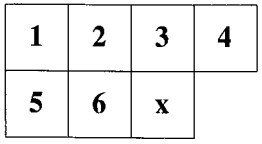


Figure 7.2: A causal support $\mathcal{B}$ for N = 7.

Following steps similar to those in the derivation of the pixel-by-pixel algorithm, steepest-descent minimization of this criterion function yields the iteration

$$\hat{\mathbf{d}}^{i+1}(\mathbf{x}) = \hat{\mathbf{d}}^i(\mathbf{x}) - \epsilon \sum_{\mathbf{x}_B \in \mathcal{B}} dfd(\mathbf{x}_B, \hat{\mathbf{d}}^i(\mathbf{x}))\ \nabla_{\mathbf{x}}\ s_c(\mathbf{x}_B - \hat{\mathbf{d}}^i(\mathbf{x}); t\ \text{-At}) \tag{7.24}$$

Observe that this formulation is equivalent to that of block matching, except for the shape of the support $\mathcal{B}$. Here, the solution is sought for using the steepest-descent minimization rather than a search strategy.

7.4 Wiener-Estimation-Based Algorithms

The Wiener-estimation-based method is an extension of the Netravali-Robbins algorithm in the case of block motion, where the higher-order terms in the linearized DFD expression (7.17) are not ignored. Instead, a linear least squares error (LLSE) or linear minimum mean square error (LMMSE) estimate of the update term

$$\mathbf{u}^i(\mathbf{x}) = \mathbf{d(x)} - \hat{\mathbf{d}}^i(\mathbf{x}) \tag{7.25}$$

where **d(x)** denotes the true displacement vector, is derived based on a neighborhood $\mathcal{B}$ of a pel x. In the following, we provide the derivation of the Wiener estimator for the case of N observations with a common displacement vector **d(x).**

Writing the linearized DFD (7.17), with $dfd(\mathbf{x}_B, \mathbf{d}) = \mathbf{0}$, **at** all **N** pixels $\mathbf{x}_B$ within the support $\mathcal{B}$, we have N equations in the two unknowns (the components of $\mathbf{u}^i$) given by

$$\begin{aligned}
-dfd(\mathbf{x}_B(1), \hat{\mathbf{d}}^i(\mathbf{x})) &= \nabla^T s_c(\mathbf{x}_B(1) - \hat{\mathbf{d}}^i(\mathbf{x}), t - \Delta t)\mathbf{u}^i + o(\mathbf{x}_B(1), \hat{\mathbf{d}}^i(\mathbf{x})) \\
-dfd(\mathbf{x}_B(2), \hat{\mathbf{d}}^i(\mathbf{x})) &= \nabla^T s_c(\mathbf{x}_B(2) - \hat{\mathbf{d}}^i(\mathbf{x}), t - \Delta t)\mathbf{u}^i + o(\mathbf{x}_B(2), \hat{\mathbf{d}}^i(\mathbf{x})) \\
&\vdots = \vdots \\
-dfd(\mathbf{x}_B(N), \hat{\mathbf{d}}^i(\mathbf{x})) &= \nabla^T s_c(\mathbf{x}_B(N) - \hat{\mathbf{d}}^i(\mathbf{x}), t - \Delta t)\mathbf{u}^i + o(\mathbf{x}_B(N), \hat{\mathbf{d}}^i(\mathbf{x}))
\end{aligned}$$

where $\mathbf{x}_B(1), , \mathbf{x}_B(N)$ denote an ordering of the pixels within the support $\mathcal{B}$ (shown in Figure 7.2). These equations can be expressed in vector-matrix form as

$$\mathbf{z} = \mathbf{\Phi u}(\mathbf{x}) + \mathbf{n} \tag{7.26}$$

where

$$\mathbf{z} = \begin{bmatrix} -dfd(\mathbf{x}_B(1), \hat{\mathbf{d}}^i(\mathbf{x})) \\ -dfd(\mathbf{x}_B(2), \hat{\mathbf{d}}^i(\mathbf{x})) \\ \vdots \\ -dfd(\mathbf{x}_B(N), \hat{\mathbf{d}}^i(\mathbf{x})) \end{bmatrix}$$

$$\mathbf{\Phi} = \begin{bmatrix} \frac{\partial s_c(\mathbf{x}_B(1) - \hat{\mathbf{d}}^i, t - \Delta t)}{\partial x_1} & \frac{\partial s_c(\mathbf{x}_B(1) - \hat{\mathbf{d}}^i, t - \Delta t)}{\partial x_2} \\ \frac{\partial s_c(\mathbf{x}_B(2) - \hat{\mathbf{d}}^i, t - \Delta t)}{\partial x_1} & \frac{\partial s_c(\mathbf{x}_B(2) - \hat{\mathbf{d}}^i, t - \Delta t)}{\partial x_2} \\ \cdot & \\ \frac{\partial s_c(\mathbf{x}_B(N) - \hat{\mathbf{d}}^i, t - \Delta t)}{\partial x_1} & \frac{\partial s_c(\mathbf{x}_B(N) - \hat{\mathbf{d}}^i, t - \Delta t)}{\partial x_2} \end{bmatrix}$$

and

$$\mathbf{n} = \begin{bmatrix} o(\mathbf{x}_B(1), \hat{\mathbf{d}}^i) \\ o(\mathbf{x}_B(2), \hat{\mathbf{d}}^i) \\ \vdots \\ o(\mathbf{x}_B(N), \hat{\mathbf{d}}^i) \end{bmatrix}$$

Assuming that the update term $\mathbf{u}(\mathbf{x})$ and the truncation error n are uncorrelated random vectors, and using the principle of orthogonality, the LMMSE estimate of the update term is given by [Bie 87]

$$\hat{\mathbf{u}}(\mathbf{x}) = [\mathbf{\Phi}^T \mathbf{R}_n^{-1} \mathbf{\Phi} + \mathbf{R}_u^{-1}]^{-1} \mathbf{\Phi}^T \mathbf{R}_n^{-1} \mathbf{z} \quad (7.27)$$

Note that the solution requires the knowledge of the covariance matrices of both the update term $\mathbf{R}_u$ and the linearization error $\mathbf{R}_n$. In the absence of exact knowledge of these quantities, we will make the simplifying assumptions that both vectors have zero mean value, and their components are uncorrelated among each other. That is, we have $\mathbf{R}_u = \sigma_u^2 \mathbf{I}$ and $\mathbf{R}_n = \sigma_n^2 \mathbf{I}$, where $\mathbf{I}$ is a 2 x 2 identity matrix, and σ_u^2 and σ_n^2 are the variances of the components of the two vectors, respectively. Then the LMMSE estimate (7.27) simplifies to

$$\hat{\mathbf{u}}(\mathbf{x}) = [\mathbf{\Phi}^T \mathbf{\Phi} + \mu \mathbf{I}]^{-1} \mathbf{\Phi}^T \mathbf{z} \quad (7.28)$$

where $\mu = \sigma_n^2 / \sigma_u^2$ is called the damping parameter. Equation (7.28) gives the least squares estimate of the update term. The assumptions that are used to arrive at the simplified estimator are not, in general, true; for example, the linearization error is not uncorrelated with the update term, and the updates and the linearization errors at each pixel are not uncorrelated with each other. However, experimental results indicate improved performance compared with other pel-recursive estimators [Bie 87].

Having obtained an estimate of the update term, the Wiener-based pel-recursive estimator can be written as

$$\hat{\mathbf{d}}^{i+1}(\mathbf{x}) = \hat{\mathbf{d}}^{i}(\mathbf{x}) + [\mathbf{\Phi}^T \mathbf{\Phi} + \mu \mathbf{I}]^{-1} \mathbf{\Phi}^T \mathbf{z} \quad (7.29)$$

It has been pointed out that the Wiener-based estimator is related to the Walker-Rao and Caffario-Rocca algorithms [Bie 87]. This can easily be seen by writing (7.29) for the special case of N = 1 as

$$\hat{\mathbf{d}}^{i+1}(\mathbf{x}) = \hat{\mathbf{d}}^{i}(\mathbf{x}) - \frac{dfd(\mathbf{x}, \hat{\mathbf{d}}^i) \nabla^T s_c(\mathbf{x} - \hat{\mathbf{d}}^i, t - \mathbf{At})}{|\nabla^T s_c(\mathbf{x} - \hat{\mathbf{d}}^i, t - \Delta t)|^2 + \mu} \quad (7.30)$$

The so-called simplified Caffario-Rocca algorithm results when $\mu = 100$. We obtain the Walker-Rao algorithm when we set $\mu = 0$ and multiply the update term by $1/2$. The convergence properties of the Wiener-based estimators have been analyzed in [Bor 91].

The Wiener-estimation-based scheme presented in this section employs the best motion vector estimate from the previous iteration as the prediction for the next iteration. An improved algorithm with a motion-compensated spatio-temporal vector predictor was proposed in [Dri 91]. Pel-recursive algorithms have also been extended to include rotational motion [Bie 88]. As a final remark, we note that all pel-recursive algorithms can be applied hierarchically, using a multiresolution representation of the images to obtain improved results. Pel-recursive algorithms have recently evolved into Bayesian motion estimation methods, employing stochastic motion-field models, which will be introduced in the next chapter.

7.5 Examples

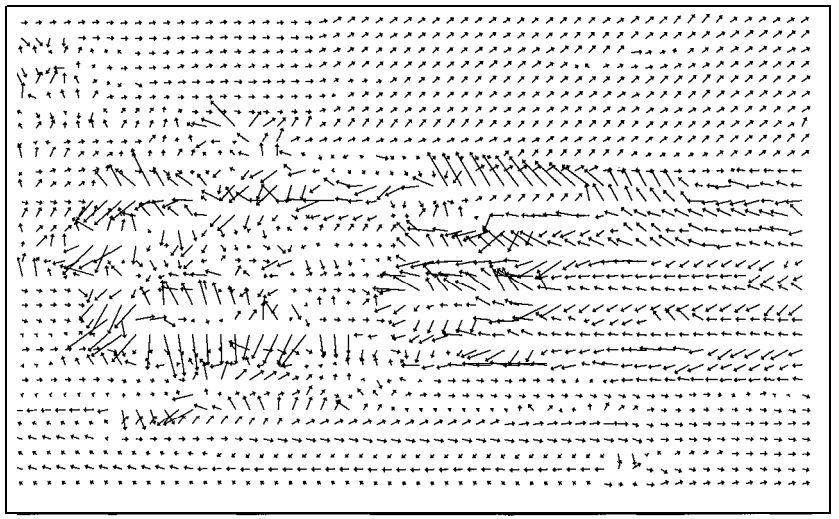
We have applied the Walker-Rao algorithm, given by (7.19), (7.22), and the heuristic rules stated in Section 7.3.2, and the Wiener-based method given by (7.29) to the same two frames of the Mobile and Calendar sequence that we used in Chapters 5 and 6. We generated two sets of results with the Walker-Rao algorithm, where we allowed 2 and 10 iterations at each pixel, respectively. In both cases, we have set the threshold on the DFD (see heuristic rules 1 and 2) equal to 3, and limited the maximum displacement to ± 10 as suggested by [Wal 84]. In the Wiener-estimation-based approach, we employed the support shown in Figure 7.2 with N = 7, allowed 2 iterations at each pixel, and set the damping parameter $\mu = 100$.

The effectiveness of the methods is evaluated visually, by inspection of the motion vector fields shown in Figure 7.3 (for the case of two iterations/pixel, I=2), and numerically, by comparing the PSNR and entropy values tabulated in Table 7.1. A few observations about the estimated motion fields are in order: i) Inspection of the upper left corner of the Walker-Rao estimate, shown in Figure 7.3 (a) indicates that the displacement estimates converge to the correct background motion after processing about 10 pixels. ii) In the Walker-Rao estimate, there are many outlier vectors in the regions of rapidly changing motion. iii) The Wiener estimate, shown in Figure 7.3 (b), contains far fewer outliers but gives zero motion vectors in the uniform areas (see flat areas in the background, the foreground, and the calendar). This may be overcome by using a larger support $\mathcal{B}$. In pel-recursive estimation, propagation of erroneous motion estimates may be overcome by resetting the predicted estimate to the zero vector, if the frame difference is smaller than the DFD obtained by the actual predicted estimate at that pixel. However, we have not implemented this option in our software.

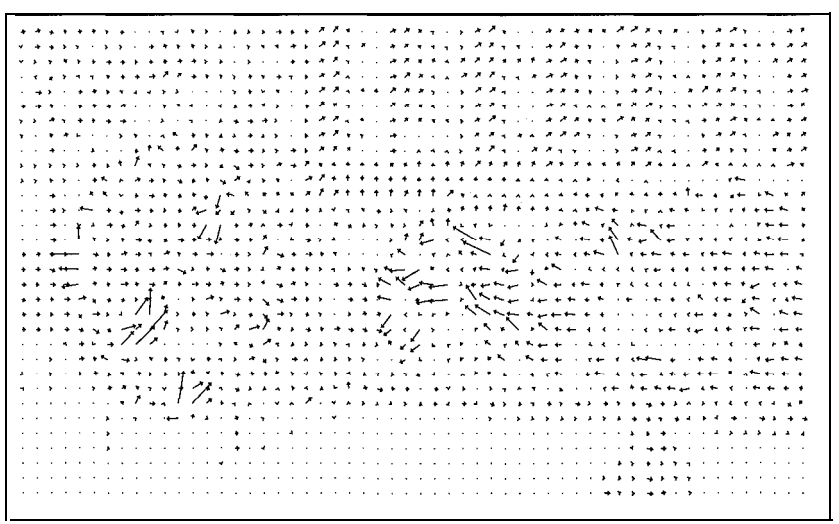
In terms of motion compensation, the Wiener-based method is about 0.8 dB better than the Walker-Rao method if we allow 2 iterations/pixel. However, if we allow 10 iterations/pixel, the performance of the Wiener-based method remains about the same, and both methods perform similarly. Observe that the entropy of the Wiener-based motion field estimate is smaller than half of both Walker-Rao estimates, for I=2 and I=10, by virtue of its regularity. The two Walker-Rao motion field estimates are visually very close, as indicated by their similar entropies.

Table 7.1: Comparison of the pel-recursive methods. (Courtesy Gozde Bozdagi)

Method	PSNR (dB)	Entropy (bits)
Frame-Difference	23.45	
Walker-Rao (I=2)	29.01	**8.51**
Wiener (I=2)	29.82	4.49
Walker-Rao (I=10)	29.92	**8.54**



(a)



(b)

Figure 7.3: Motion field obtained by a) the Walker-Rao method and b) the Wiener-based method. (Courtesy Gozde Bozdagi and Mehmet Ozkan)

7.6 Exercises

1. Derive (7.12). How would you compute the optimal step size for the Netravali-Robbins algorithm?

2. Derive an explicit expression for (7.24). Comment on the relationship between block matching and the algorithm given by (7.24).

3. Comment on the validity of the assumptions made to arrive at the Wiener-based estimator (7.29).

4. Derive (7.30) from (7.29).

Bibliography

[Bie 87] J. Biemond, L. Looijenga, D. E. Boekee, and R. H. J. M. Plompen, "A pel-recursive Wiener-based displacement estimation algorithm," Sign. *Proc.*, vol. 13, pp. 399–412, Dec. 1987.

[Bie 88] J. Biemond, J. N. Driessen, A. M. Geurtz, and D. E. Boekee, "A pel-recursive Wiener based algorithm for the simultaneous estimation of rotation and translation," *Proc. SPIE Conf.* Visual Commun. *Image Proc.*, pp. 917-924, Cambridge MA, 1988.

[Bor 91] L. Boroczky, *Pel-Recursive* Motion *Estimation for* Image *Coding*, Ph.D. thesis, Delft University of Technology, 1991.

[Caf 83] C. Cafforio and F. Rocca, "The differential method for image motion estimation," in Image *Sequence* Processing and Dynamic Scene Analysis, T. S. Huang, ed., pp. 104-124, Berlin, Germany: Springer-Verlag, 1983.

[Dri 91] J. N. Driessen, L. Boroczky, and J. Biemond, "Pel-recursive motion field estimation from image sequences," *J.* Vis. Comm. Image *Rep.*, vol. 2, no. 3, pp. 259-280, 1991.

[Fle 87] R. Fletcher, Practical *Methods of Optimization*, Vol. 1, 2nd ed., Chichester, UK: John Wiley and Sons, 1987.

[Rob 83] J. D. Robbins and A. N. Netravali, "Recursive motion compensation: A review," in Image *Sequence* Processing and Dynamic Scene Analysis, T. S. Huang, ed., pp. 76-103, Berlin, Germany: Springer-Verlag, 1983.

[Wal 84] D. R. Walker and K. R. Rao, "Improved pel-recursive motion compensation," *IEEE* Trans. Commun., vol. COM-32, pp. 1128–1134, Oct. 1984.

Chapter 8

BAYESIAN METHODS

In this chapter, 2-D motion estimation is formulated and solved as a Bayesian estimation problem. In the previous chapters, where we presented deterministic formulations of the problem, we minimized either the error in the optical flow equation or a function of the displaced frame difference (DFD). Here, the deviation of the DFD from zero is modeled by a random process that is exponentially distributed. Furthermore, a stochastic smoothness constraint is introduced by modeling the 2-D motion vector field in terms of a Gibbs distribution. The reader is referred to Appendix A for a brief review of the definitions and properties of Markov and Gibbs random fields. The clique potentials of the underlying Gibbsian distribution are selected to assign a higher a priori probability to slowly varying motion fields. In order to formulate directional smoothness constraints, more structured Gibbs random fields (GRF) with line processes are also introduced.

Since Bayesian estimation requires global optimization of a cost function, we study a number of optimization methods, including simulated annealing (SA), iterated conditional modes (ICM), mean field annealing (MFA), and highest confidence first (HCF) in Section 8.1. Section 8.2 provides the basic formulation of the maximum *a posteriori* probability (MAP) estimation problem. Extensions of the basic formulation to deal with motion discontinuities and occlusion areas are discussed in Section 8.3. It will be seen that block/pel matching and Horn-Schunck algorithms form special cases of the MAP estimator under certain assumptions.

8.1 Optimization Methods

Many motion estimation/segmentation problems require the minimization of a nonconvex criterion function **E(u),** where u is some N-dimensional unknown vector. Then, the motion estimation/segmentation problem can be posed so as to find

$$\hat{\mathbf{u}} = \arg\{\min_{\mathbf{u}} E(u)\} \tag{8.1}$$

This minimization is exceedingly difficult due to large dimensionality of the unknown vector and the presence of local minima. With nonconvex criterion functions, gradient descent methods, discussed in Chapter 7, generally cannot reach the global minimum, because they get trapped in the nearest local minimum.

In this section, we present two simulated (stochastic) annealing algorithms, the Metropolis algorithm [Met 53] and the Gibbs sampler [Gem 84], which are capable of finding the global minimum; and three deterministic algorithms, the iterative conditional modes (ICM) algorithm [Bes 74], the mean field annealing algorithm [Bil 91a], and the highest confidence first (HCF) algorithm [Cho 90], to obtain faster convergence. For a detailed survey of popular annealing procedures the reader is referred to [Kir 83, Laa 87].

8.1.1 Simulated Annealing

Simulated annealing (SA) refers to a class of stochastic relaxation algorithms known as Monte Carlo methods. They are essentially prescriptions for a partially random search of the solution space. At each step of the algorithm, the previous solution is subjected to a random perturbation. Unlike deterministic gradient-based iterative algorithms which always move in the direction of decreasing criterion function, simulated annealing permits, on a random basis, changes that increase the criterion function. This is because an uphill move is sometimes necessary in order to prevent the solution from settling in a local minimum.

The probability of accepting uphill moves is controlled by a temperature parameter. The simulated annealing process starts by first "melting" the system at a high enough temperature that almost all random moves are accepted. Then the temperature is lowered slowly according to a "cooling" regime. At each temperature, the simulation must proceed long enough for the system to reach a "steady-state." The sequence of temperatures and the number of perturbations at each temperature constitute the "annealing schedule." The convergence of the procedure is strongly related to the annealing schedule. In their pioneering work, Geman and Geman [Gem 84] proposed the following temperature schedule:

$$T = \frac{\tau}{\ln(i+1)}, \quad i = 1, \ldots \tag{8.2}$$

where τ is a constant and i is the iteration cycle. This schedule is overly conservative but guarantees reaching the global minimum. Schedules that lower the temperature at a faster rate have also been shown to work (without a proof of convergence).

The process of generating random perturbations is referred to as sampling the solution space. In the following, we present two algorithms that differ in the way they sample the solution space.

The Metropolis Algorithm

In Metropolis sampling, at each step of the algorithm a new candidate solution is generated at random. If this new solution decreases the criterion function, it is

always accepted; otherwise, it is accepted according to an exponential probability distribution. The probability P of accepting the new solution is then given by

$$P = \begin{cases} \exp(-\Delta E/T) & \text{if } \Delta E > 0 \\ 1 & \text{if } \Delta E \leq 0 \end{cases}$$

where AE is the change in the criterion function due to the perturbation, and T is the temperature parameter. If T is relatively large, the probability of accepting a positive energy change is higher than when T is small for a given AE. We provide a summary of the Metropolis algorithm in the following [Met 53]:

1. Set $i = 0$ and $T = T_{max}$. Choose an initial $\mathbf{u}^{(0)}$ at random
2. Generate a new candidate solution $\mathbf{u}^{(i+1)}$ at random.
3. Compute AE = $E(\mathbf{u}^{(i+1)}) - E(\mathbf{u}^{(i)})$.
4. Compute P from

$$P = \begin{cases} \exp(-\Delta E/T) & \text{if AE} > 0 \\ 1 & \text{if } \Delta E \leq 0 \end{cases}$$

5. If P = 1, accept the perturbation; otherwise, draw a random number that is uniformly distributed between 0 and 1. If the number drawn is less than P, accept the perturbation. .

6. Set i = i + 1. If i $\leq I_{max}$, where I_{max} is predetermined, go to 2.

7. Set i = 0, and $\mathbf{u}^{(0)} = \mathbf{u}^{(I_{max})}$. Reduce T according to a temperature schedule. If T > T_{min}, go to 2; otherwise, terminate.

Because the candidate solutions are generated by random perturbations, the algorithm typically requires a large number of iterations for convergence. Thus, the computational load of simulated annealing is significant, especially when the set of allowable values Γ (defined in Appendix A for $\mathbf{u}$ discrete) contains a large number of values or $\mathbf{u}$ is a continuous variable. Also, the computational load increases with the number of components in the unknown vector.

The Gibbs Sampler

Let's assume that u is a random vector composed of lexicographic ordering of the elements of a scalar GRF $u(\mathbf{x})$. In Gibbs sampling, the perturbations are generated according to local conditional probability density functions (pdf) derived from the given Gibbsian distribution, according to (A.5) in Appendix A, rather than making totally random perturbations and then deciding whether or not to accept them. The Gibbs sampler method is summarized in the following.

1. Set $T = T_{max}$. Choose an initial u at random.
2. Visit each site x to perturb the value of u at that site as follows:

a) At site x, first compute the conditional probability of U(X) to take each of the allowed values from the set Γ, given the present values of its neighbors using (A.5) in Appendix A. This step is illustrated for a scalar $u(\mathbf{x})$ by an example below.

Example: Computation *of* local conditional *probabilities*

Let $\Gamma = \{0, 1, 2, 3\}$. Given a 3 x 3 binary GRF, we wish to compute the conditional probability of the element marked with "x" in Figure 8.1 being equal to "0," "1," "2," or "3" given the values of its neighbors. If we define

$$\text{P(Y)} = P(u(\mathbf{x}_i) = \gamma, u(\mathbf{x}_j), \mathbf{x}_j \in N_{\mathbf{x}_i}), \quad \text{for all } \gamma \in \Gamma$$

to denote the joint probabilities of possible configurations, then, using (A.4), we have

$$P(u(\mathbf{x}_i) = 0 \mid u(\mathbf{x}_j), \mathbf{x}_j \in N_{\mathbf{x}_i}) = \frac{P(0)}{P(0) + \text{P(1)} + P(2) + P(3)}$$

$$P(u(\mathbf{x}_i) = 1 \mid u(\mathbf{x}_j), \mathbf{x}_j \in N_{\mathbf{x}_i}) = \frac{P(1)}{P(0) + \text{P(1)} + P(2) + P(3)}$$

$$P(u(\mathbf{x}_i) = 2 \mid u(\mathbf{x}_j), \mathbf{x}_j \in N_{\mathbf{x}_i}) = \frac{P(2)}{P(0) + P(1) + P(2) + P(3)}$$

and

$$P(u(\mathbf{x}_i) = 3 \mid u(\mathbf{x}_j), \mathbf{x}_j \in N_{\mathbf{x}_i}) = \frac{\text{P(3)}}{P(0) + P(1) + P(2) + P(3)}$$

The evaluation of $P(y)$ for all $\gamma \in \Gamma$, using the 4-pixel neighborhood system shown in Figure A.1 and the 2-pixel clique potential specified by (8.11), is left as an exercise. You may assume that the a *priori* probabilities of a "0" and a "1" are equal.

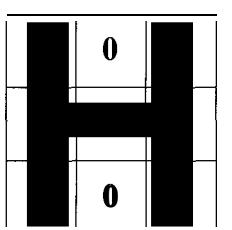


Figure 8.1: Illustration of local probability computation.

b) Once the probabilities for all elements of the set Γ are computed, draw the new value of $u(\mathbf{x})$ from this distribution. To clarify the meaning of "draw," again an example is provided.

Example: Suppose that $\Gamma = \{0, 1, 2, 3\}$, and it was found that

$$P(u(\mathbf{x}_i) = 0 \mid u(\mathbf{x}_j), \mathbf{x}_j \in \mathcal{N}_{\mathbf{x}_i}) = 0.2$$
$$P(u(\mathbf{x}_i) = 1 \mid u(\mathbf{x}_j), \mathbf{x}_j \in \mathcal{N}_{\mathbf{x}_i}) = 0.1$$
$$P(u(\mathbf{x}_i) = 2 \mid u(\mathbf{x}_j), \mathbf{x}_j \in \mathcal{N}_{\mathbf{x}_i}) = 0.4$$
$$P(u(\mathbf{x}_i) = 3 \mid u(\mathbf{x}_j), \mathbf{x}_j \in \mathcal{N}_{\mathbf{x}_i}) = 0.3$$

Then a random number R, uniformly distributed between 0 and 1, is generated, and the value of $u(\mathbf{x}_i)$ is decided as follows: if $0 \leq R \leq 0.2$ then $u(\mathbf{x}_i) = 0$ if $0.2 < R \leq 0.3$ then $u(\mathbf{x}_i) = 1$ if $0.3 < R \leq 0.7$ then $u(\mathbf{x}_i) = 2$ if $0.7 < R \leq 1$ then $u(\mathbf{x}_i) = 3$.

3. Repeat step 2 a sufficient number of times at a given temperature, then lower the temperature, and go to 2. Note that the conditional probabilities depend on the temperature parameter.

Perturbations through Gibbs sampling lead to very interesting properties that have been shown by Geman and Geman [Gem 84]:
(i) For any initial estimate, Gibbs sampling will yield a distribution that is asymptotically Gibbsian, with the same properties as the Gibbs distribution used to generate it. This result can be used to simulate a Gibbs random field.
(ii) For the particular temperature schedule (8.2), the global optimum will be reached. However, in practice, convergence with this schedule may be too slow.

8.1.2 Iterated Conditional Modes

Iterated conditional modes (ICM), also known as the greedy algorithm, is a deterministic procedure which aims to reduce the computational load of the stochastic annealing methods. It can be posed as special cases of both the Metropolis and Gibbs sampler algorithms.

ICM can best be conceptualized as the "instant freezing" case of the Metropolis algorithm, that is, when the temperature T is set equal to zero for all iterations. Then the probability of accepting perturbations that increase the value of the cost function is always 0 (refer to step 4 of the Metropolis algorithm). Alternatively, it has been shown that ICM converges to the solution that maximizes the local conditional probabilities given by (A.5)at each site. Hence, it can be implemented as in Gibbs sampling, but by choosing the value at each site that gives the maximum local conditional probability rather than drawing a value based on the conditional probability distribution.

ICM convergences much faster than the stochastic SA algorithms. However, because ICM only allows those perturbations yielding negative ΔE, it is likely to get trapped in a local minimum, much like gradient-descent algorithms. Thus, it is critical to initialize ICM with a reasonably good initial estimate. The use of ICM has been reported for image restoration [Bes 74] and image segmentation [Pap 92].

8.1.3 Mean Field Annealing

Mean field annealing is based on the "mean field approximation" (MFA) idea in statistical mechanics. MFA allows replacing each random variable (random field evaluated at a particular site) by the mean of its marginal probability distribution at a given temperature. Then mean field annealing is concerned about the estimation of these means at each site. Because the estimation of each mean is dependent on the means of the neighboring sites, this estimation is performed using an annealing schedule. The algorithm for annealing the mean field is similar to SA except that stochastic relaxation at each temperature is replaced by a deterministic relaxation to minimize the so-called mean field error, usually using a gradient-descent algorithm.

Historically, the mean field algorithms were limited to Ising-type models described by a criterion function involving a binary vector. However, it has recently been extended to a wider class of problems, including those with continuous variables [Bil 91b]. Experiments suggest that the MFA is valid for MRFs with local interactions over small regions. Thus, computations of the means and the mean field error are often based on Gibbsian distributions. It has been claimed that mean field annealing converges to an acceptable solution approximately 50 times faster than SA. The implementation of MFA is not unique. Covering all seemingly different implementations of the mean field annealing [Orl 85, Bil 92, Abd 92, Zha 93] is beyond the scope of this book.

8.1.4 Highest Confidence First

The highest confidence first (HCF) algorithm proposed by Chou and Brown [Cho 90] is a deterministic, noniterative algorithm. It is guaranteed to reach a local minimum of the potential function after a finite number of steps.

In the case of a discrete-valued GRF, the minimization is performed on a site-by-site basis according to the following rules: 1) Sites with reliable data can be labeled without using the prior probability model. 2) Sites where the data is unreliable should rely on neighborhood interaction for label assignment. 3) Sites with unreliable data should not affect sites with reliable data through neighborhood interaction. Guided by these principles, a scheme that determines a particular order for assigning labels and systematically increases neighborhood interaction is designed. Initially, all sites are labeled "uncommitted." Once a label is assigned to an uncommitted site, the site is committed and cannot return to the uncommitted state. However, the label of a committed site can be changed through another assignment. A "stability" measure is calculated for each site based on the local conditional *a posteriori* probability of the labels at that site, to determine the order in which the sites are to be visited. The procedure terminates when the criterion function can no longer be decreased by reassignment of the labels.

Among the deterministic methods, HCF is simpler and more robust than MFA, and more accurate than the ICM. Extensions of HCF for the case of continuous variables also exist.

8.2 Basics of MAP Motion Estimation

In this section, 2-D motion estimation is formulated as a maximum a *posteriori* probability (MAP) estimation problem. The MAP formulation requires two pdf models: the conditional pdf of the observed image intensity given the motion field, called the likelihood model or the observation model, and the a *priori* pdf of the motion vectors, called the motion field model. The basic formulation assumes a Gaussian observation model to impose consistency with the observations, and a Gibbsian motion field model to impose a probabilistic global smoothness constraint.

In order to use a compact notation, we let the vector $\mathbf{s}_k$ denote a lexicographic ordering of the ideal pixel intensities $s_k(\mathbf{x})$ in the kth picture (frame or field), such that (x, $t) \in \Lambda^3$ for $t = k\Delta t$. If $\mathbf{d}(\mathrm{x}) = (d_1(\mathbf{x}), d_2(\mathbf{x}))$ denotes the displacement vector from frame/field $k-1$ to k at the site $\mathrm{x} = (x_1, x_2)$, then we let $\mathbf{d}_1$ and $\mathbf{d}_2$ denote a lexicographic ordering of $d_1(\mathbf{x})$ and $d_2(\mathbf{x})$, respectively. Hence, ignoring covered/uncovered background regions and intensity variations due to changes in the illumination and shading, we have

$$s_k(\mathbf{x}) = s_{k-1}(\mathbf{x} - \mathrm{d(x)}) \doteq s_{k-1}(x_1 - d_1(\mathbf{x}), x_2 - d_2(\mathbf{x})) \tag{8.3}$$

which is a restatement of the optical flow constraint.

In general, we can only observe video that is corrupted by additive noise, given by

$$g_k(\mathbf{x}) = s_k(\mathbf{x}) + v_k(\mathbf{x}) \tag{8.4}$$

and need to estimate 2-D motion from the noisy observations. Then the basic MAP problem can be stated as: given two frames $\mathbf{g}_k$ and $\mathbf{g}_{k-1}$, find

$$(\hat{\mathbf{d}}_1, \hat{\mathbf{d}}_2) = \arg\max_{\mathbf{d}_1, \mathbf{d}_2} p(\mathbf{d}_1, \mathbf{d}_2 | \mathbf{g}_k, \mathbf{g}_{k-1}) \tag{8.5}$$

where $p(\mathbf{d}_1, \mathbf{d}_2 | \mathbf{g}_k, \mathbf{g}_{k-1})$ denotes the a *posterior-i* pdf of the motion field given the two frames. Using the Bayes theorem,

$$p(\mathbf{d}_1, \mathbf{d}_2 | \mathbf{g}_k, \mathbf{g}_{k-1}) = \frac{p(\mathbf{g}_k | \mathbf{d}_1, \mathbf{d}_2, \mathbf{g}_{k-1}) p(\mathbf{d}_1, \mathbf{d}_2 | \mathbf{g}_{k-1})}{p(\mathbf{g}_k | \mathbf{g}_{k-1})} \tag{8.6}$$

where $p(\mathbf{g}_k | \mathbf{d}_1, \mathbf{d}_2, \mathbf{g}_{k-1})$ is the conditional probability, or the "consistency (likelihood) measure," that measures how well the motion fields $\mathbf{d}_1$ and $\mathbf{d}_2$ explain the observations $\mathbf{g}_k$ through (8.3) given $\mathbf{g}_{k-1}$; and $p(\mathbf{d}_1, \mathbf{d}_2 | \mathbf{g}_{k-1})$ is the a priori pdf of the motion field that reflects our prior knowledge about the actual motion field. Since the denominator is not a function of $\mathbf{d}_1$ and $\mathbf{d}_2$, it is a constant for the purposes of motion estimation. Then the MAP estimate (8.5) can be expressed as

$$(\hat{\mathbf{d}}_1, \hat{\mathbf{d}}_2) = \arg\max_{\mathbf{d}_1, \mathbf{d}_2} p(\mathbf{g}_k | \mathbf{d}_1, \mathbf{d}_2, \mathbf{g}_{k-1}) p(\mathbf{d}_1, \mathbf{d}_2 | \mathbf{g}_{k-1}) \tag{8.7}$$

or as

$$(\hat{\mathbf{d}}_1, \hat{\mathbf{d}}_2) = \arg\max_{\mathbf{d}_1, \mathbf{d}_2} p(\mathbf{g}_{k-1} | \mathbf{d}_1, \mathbf{d}_2, \mathbf{g}_k) p(\mathbf{d}_1, \mathbf{d}_2 | \mathbf{g}_k)$$

Next, we develop models for the conditional and the prior pdfs.

8.2.1 The Likelihood Model

Based on the models (8.3) and (8.4), the change in the intensity of a pixel along the true motion trajectory is due to observation noise. Assuming that the observation noise is white, Gaussian with zero mean and variance σ^2, the conditional pdf $p(\mathbf{g}_k | \mathbf{d}_1, \mathbf{d}_2, \mathbf{g}_{k-1})$ can be modeled as

$$p(\mathbf{g}_k | \mathbf{d}_1, \mathbf{d}_2, \mathbf{g}_{k-1}) = (2\pi\sigma^2)^{-\frac{1}{2d(\Lambda)}} \exp\left\{ - \sum_{\mathbf{x} \in \Lambda} \frac{[g_k(\mathbf{x}) - g_{k-1}(\mathbf{x} - \mathbf{d}(\mathbf{x}))]^2}{2\sigma^2} \right\} \tag{8.8}$$

where d(h) denotes the determinant of A which gives the reciprocal of the sampling density. The conditional pdf (8.8) gives the likelihood of observing the intensity $\mathbf{g}_k$ given the true motion field, $\mathbf{d}_1$ and $\mathbf{d}_2$, and the intensity vector of the previous frame $\mathbf{g}_{k-1}$.

8.2.2 The Prior Model

The motion field is assumed to be a realization of a continuous-valued GRF, where the clique potential functions are chosen to impose a local smoothness constraint on pixel-to-pixel variations of the motion vectors. Thus, the joint a *priori* pdf of the motion vector field can be expressed as

$$p(\mathbf{d}_1, \mathbf{d}_2 | \mathbf{g}_{k-1}) = \frac{1}{Q_d} \exp\{-U_d(\mathbf{d}_1, \mathbf{d}_2 | \mathbf{g}_{k-1})\} \tag{8.9}$$

where Q_d is the partition function, and

$$U_d(\mathbf{d}_1, \mathbf{d}_2 | \mathbf{g}_{k-1}) = \lambda_d \sum_{c \in \mathcal{C}_d} V_d^c(\mathbf{d}_1, \mathbf{d}_2 | \mathbf{g}_{k-1})$$

Here $\mathcal{C}_d$ denotes the set of all cliques for the displacement field, *Vi(.)* represents the clique potential function for c $\in \mathcal{C}_d$, and λ_d is a positive constant. The clique potentials will be chosen to assign smaller probability to configurations where the motion vector varies significantly from pixel to pixel. This is demonstrated by two examples in the following.

Example: The case of a continuous-valued *GRF*

This example demonstrates that a spatial smoothness constraint can be formulated as an a *priori* pdf in the form of a Gibbs distribution. Let us employ a four-point neighborhood system, depicted in Figure A.1, with two-pixel cliques (see Appendix A). For continuous-valued GRF, a suitable potential function for the two-pixel cliques may be

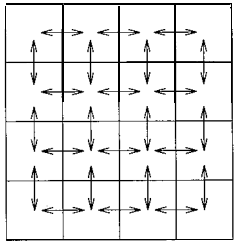
$$V_d^{c_2}(\mathbf{d}(\mathbf{x}_i), \mathbf{d}(\mathbf{x}_j)) = ||\mathbf{d}(\mathbf{x}_i) - \mathbf{d}(\mathbf{x}_j)||^2 \tag{8.10}$$

where $\mathbf{x}_i$ and $\mathbf{x}_j$ denote the elements of any two-pixel clique c_2, and $||.||$ is the Euclidian distance. In Equation 8.9, $V_d^{c_2}(\mathbf{d}(\mathbf{x}_i), \mathbf{d}(\mathbf{x}_j))$ needs to

be summed over all two-pixel cliques. Clearly, a spatial configuration of motion vectors with larger potential would have a smaller a priori probability.

Example: The case of a discrete-valued *GRF*

If the motion vectors are quantized, say, to 0.5 pixel accuracy, then we have a discrete-valued GRF. The reader is reminded that the definition of the pdf (8.9) needs to be modified to include a Dirac delta function in this case (see (A.2)). Suppose that a discrete-valued GRF z is defined over the 4 x 4 lattice, shown in Figure 8.2 (a). Figure 8.2 (b) and (c) show two realizations of 4 x 4 binary images.



24 two-pixel cliques

(a)

2	2	2	2
2	2	2	2
2	2	2	2
2	2	2	2

$V = -24\ \beta$

(b)

1	2	1	2
2	1	2	1
1	2	1	2
2	1	2	1

$V = 24\ \beta$

(c)

Figure 8.2: Demonstration of a Gibbs model

Let the two-pixel clique potential be defined as

$$V_C(z(\mathbf{x}_i), z(\mathbf{x}_j)) = \begin{cases} -\beta & \text{if } z(\mathbf{x}_i) = z(\mathbf{x}_j) \\ +\beta & \text{otherwise} \end{cases} \tag{8.11}$$

where β is a positive number.

There are a total of 24 two-pixel cliques in a 4 x 4 image (shown by double arrows). It can be easily verified, by summing all clique potentials, that the configurations shown in Figures 8.2 (b) and (c) have the Gibbs potentials -24β and $+24\beta$, respectively. Clearly, with the choice of the clique potential function (8.11), the spatially smooth configuration in Figure 8.2 (b) has a higher a priori probability.

The basic formulation of the MAP motion estimation problem can be obtained by substituting the likelihood model (8.8) and the a priori model (8.9) into (8.7). Simplification of the resulting expression indicates that maximization of (8.7) is

equivalent to minimization of a weighted sum of the square of the displaced frame difference and a global smoothness constraint term. The MAP estimation is usually implemented by minimizing the corresponding criterion function using simulated annealing.

A practical problem with the basic formulation is that it imposes a global smoothness constraint over the entire motion field, resulting in the blurring of optical flow boundaries. This blurring also adversely affects the motion estimates elsewhere in the image. Extensions of the basic approach to address this problem are presented in the following.

8.3 MAP Motion Estimation Algorithms

Three different approaches to account for the presence of optical flow boundaries and occlusion regions are discussed. First, a formulation which utilizes more structured motion field models, including an occlusion field and a discontinuity (line) field, is introduced. While the formulation with the discontinuity models is an elegant one, it suffers from a heavy computational burden because the discontinuity models introduce many more unknowns. To this effect, we also present two other noteworthy algorithms, namely the Local Outlier Rejection method proposed by Iu [Iu 93] and the region-labeling method proposed by Stiller [Sti 93].

8.3.1 Formulation with Discontinuity Models

We introduce two binary auxilary MRFs, called the occlusion field o and the line field 1, to improve the accuracy of motion estimation. The a *priori* pdf (8.9) penalizes any discontinuity in the motion field. In order to avoid oversmoothing actual optical flow boundaries, the line field is introduced, which marks the location of all allowed discontinuities in the motion field. The line field, **1,** has sites between every pair of pixel sites in the horizontal and vertical directions. Figure 8.3 (a) illustrates a 4-line clique of a line field, composed of horizontal and vertical lines indicating possible discontinuities in the vertical and horizontal directions, respectively. The state of each line site can be either ON (l = 1) or *OFF* (I = 0), expressing the presence or absence of a discontinuity in the respective direction. The actual state of each line field site is a *priori* unknown and needs to be estimated along with the motion vector field.

While the line field is defined to improve the a *priori* motion field model, the occlusion field, o, is used to improve the likelihood model. The occlusion field, which occupies the same lattice as the pixel sites, is an indicator function of occlusion pixels (refer to Section 5.2.1 for a discussion of occlusion) defined by

$$o(\mathbf{x}) = \begin{cases} 0 & \text{if } \mathbf{d}(x_1, x_2) \text{ is well defined} \\ 1 & \text{if x} = (x_1, x_2) \text{ is an occlusion point} \end{cases} \tag{8.12}$$

Because the pixels where o(x) is ON are also a priori unknown, the occlusion field

needs to be estimated along with the motion vector field and the line field.

With the introduction of the occlusion and line fields, the MAP motion estimation problem can be restated as: given the two frames, $\mathbf{g}_k$ and $\mathbf{g}_{k-1}$, find

$$\{\hat{\mathbf{d}}_1, \hat{\mathbf{d}}_2, \hat{\mathbf{o}}, \hat{\mathbf{l}}\} = \arg\max_{\mathbf{d}_1, \mathbf{d}_2, \mathbf{o}, \mathbf{l}} p(\mathbf{d}_1, \mathbf{d}_2, \mathbf{o}, \mathbf{l} | \mathbf{g}_k, \mathbf{g}_{k-1}) \tag{8.13}$$

Using the Bayes rule to factor the a *posteriori* probability and following the same steps as in the basic formulation, the MAP estimate can be expressed as

$$\begin{aligned}\{\hat{\mathbf{d}}_1, \mathbf{d}_2, \hat{\mathbf{o}}, \hat{\mathbf{l}}\} = \arg\max_{\mathbf{d}_1, \mathbf{d}_2, \mathbf{o}, \mathbf{l}} & p(\mathbf{g}_k | \mathbf{d}_1, \mathbf{d}_2, \mathbf{o}, \mathbf{l}, \mathbf{g}_{k-1}) \\ & p(\mathbf{d}_1, \mathbf{d}_2 | \mathbf{o}, \mathbf{l}, \mathbf{g}_{k-1}) p(\mathbf{o} | \mathbf{l}, \mathbf{g}_{k-1}) p(\mathbf{l} | \mathbf{g}_{k-1})\end{aligned} \tag{8.14}$$

where the first term is the improved likelihood model, the second, third, and fourth terms are the displacement field, the occlusion field, and the discontinuity field models, respectively. Next, we develop expressions for these models.

The Likelihood Model

The conditional probability model (8.8) fails at the occlusion areas, since the displaced frame difference in the occlusion areas cannot be accurately modeled as white, Gaussian noise. The modeling error at the occlusion points can be avoided by modifying (8.8) based on the knowledge of the occlusion field as

$$\begin{aligned}p(\mathbf{g}_k | \mathbf{d}_1, \mathbf{d}_2, \mathbf{o}, \mathbf{l}, \mathbf{g}_{k-1}) = \\ (2\pi\sigma^2)^{-\frac{1}{2}N} \exp\left\{ - \sum_{\mathbf{x} \in \Lambda} \frac{(1 - o(\mathbf{x}))[g_k(\mathbf{x}) - g_{k-1}(\mathbf{x} - \mathbf{d}(\mathbf{x}))]^2}{2\sigma^2} \right\}\end{aligned} \tag{8.15}$$

where the contributions of the pixels in the occluded areas are discarded, and N is the number of sites that do not experience occlusion. Note that this improved likelihood model does not depend on the line field explicitly. The pdf (8.15) can be expressed more compactly in terms of a "potential function" as

$$p(\mathbf{g}_k | \mathbf{d}_1, \mathbf{d}_2, \mathbf{o}, \mathbf{l}, \mathbf{g}_{k-1}) = \exp\{-U_g(\mathbf{g}_k | \mathbf{d}_1, \mathbf{d}_2, \mathbf{o}, \mathbf{g}_{k-1})\}$$

where

$$\begin{aligned}U_g(\mathbf{g}_k | \mathbf{d}_1, \mathbf{d}_2, \mathbf{o}, \mathbf{g}_{k-1}) = & \frac{\log(2\pi\sigma^2)}{2N} \\ & + \frac{1}{2\sigma^2} \sum_{\mathbf{x} \in \Lambda} (1 - o(\mathbf{x}))[g_k(\mathbf{x}) - g_{k-1}(\mathbf{x} - \mathbf{d}(\mathbf{x}))]^2\end{aligned} \tag{8.16}$$

We must assign an appropriate penalty for the use of the occlusion state "ON." Otherwise, the displaced frame difference can be made arbitrarily small by using more occlusion states. This penalty is imposed by the occlusion model discussed below.

The Motion Field Model

We incorporate the line field model to the prior motion vector field model (8.9) in order not to unduly penalize optical flow boundaries. The improved a priori motion field model can be expressed as

$$p(\mathbf{d}_1, \mathbf{d}_2 | \mathbf{o}, \mathbf{l}, \mathbf{g}_{k-1}) = \frac{1}{Q_d} \exp\{-U_d(\mathbf{d}_1, \mathbf{d}_2 | \mathbf{o}, \mathbf{l}, \mathbf{g}_{k-1}) / \beta_d\} \tag{8.17}$$

where Q_d is the partition function, and

$$U_d(\mathbf{d}_1, \mathbf{d}_2 | \mathbf{o}, \mathbf{l}, \mathbf{g}_{k-1}) = \sum_{c \in \mathcal{C}_d} V_d^c(\mathbf{d}_1, \mathbf{d}_2 | \mathbf{o}, \mathbf{l}, \mathbf{g}_{k-1}) \tag{8.18}$$

Here $\mathcal{C}_d$ denotes the set of all cliques for the displacement field, and *Vi(.)* represents the clique potential function for c $\in \mathcal{C}_d$.

Typically, the dependency of the clique potentials on o and $\mathbf{g}_{k-1}$ are omitted. We present two examples of such potential functions for 2-pixel cliques,

$$V_d^{c_2}(\mathbf{d}(\mathbf{x}_i), \mathbf{d}(\mathbf{x}_j) | \mathbf{l}) = ||\mathbf{d}(\mathbf{x}_i) - \mathbf{d}(\mathbf{x}_j)||^2 (1 - l(\mathbf{x}_i, \mathbf{x}_j)) \tag{8.19}$$

or

$$V_d^{c_2}(\mathbf{d}(\mathbf{x}_i), \mathbf{d}(\mathbf{x}_j) | \mathbf{l}) = (1 - \exp(-\gamma_d ||\mathbf{d}(\mathbf{x}_i) - \mathbf{d}(\mathbf{x}_j)||^2)) (1 - l(\mathbf{x}_i, \mathbf{x}_j)) \tag{8.20}$$

where γ_d is a scalar, $\mathbf{x}_i$ and $\mathbf{x}_j$ denote the elements of the two-pixel cliques c_2, and $l(\mathbf{x}_i, \mathbf{x}_j)$ denotes the line field site that is in between the pixel sites $\mathbf{x}_i$ and $\mathbf{x}_j$. As can be seen, no penalty is assigned to discontinuities in the motion vector field, if the line field site between these motion vectors is "ON." However, we must assign an appropriate penalty for turning the line field state "ON." Otherwise, the smoothness constraint can be effectively turned off by setting all of the line field sites "ON." This penalty is introduced by the motion discontinuity model discussed below.

The Occlusion Field Model

The occlusion field models the spatial distribution of occlusion labels as a discrete-valued GRF described by

$$P(\mathbf{o} | \mathbf{l}, \mathbf{g}_{k-1}) = \frac{1}{Q_o} \exp\{-U_o(\mathbf{o} | \mathbf{l}, \mathbf{g}_{k-1}) / \beta_o\} \tag{8.21}$$

where

$$U_o(\mathbf{o} | \mathbf{l}, \mathbf{g}_{k-1}) = \sum_{c \in \mathcal{C}_o} V_o^c(\mathbf{o} | \mathbf{l}, \mathbf{g}_{k-1}) \tag{8.22}$$

Here $\mathcal{C}_o$ denotes the set of all cliques for the occlusion field, and $V_o^c(.)$ represents the clique potential function for c $\in$ C,.

The potential functions $Vi(.)$ for all possible cliques are chosen to provide a penalty for turning the associated occlusion states "ON." For example, we will define the potential for singleton cliques as

$$V_o^{c_1}(o(\mathbf{x})) = o(\mathbf{x})T_o \tag{8.23}$$

which penalizes each "ON" state by an amount T_o. The quantity T_o can be viewed as a threshold on the normalized displaced frame difference,

$$\epsilon^2(\mathbf{x}) = \frac{[g_k(\mathbf{x}) - g_{k-1}(\mathbf{x} - \mathbf{d}(\mathbf{x}))]^2}{2\sigma^2} \tag{8.24}$$

That is, the occlusion state should be turned "ON" only when $\epsilon^2(\mathbf{x})$ is larger than T_o. The spatial distribution of the "ON" occlusion states and their interaction with line field states can be controlled by specifying two or more pixel cliques. Examples for the choices of occlusion field cliques can be found in Dubois and Konrad [Dub 93].

The Line Field Model

The allowed discontinuities in the motion field are represented by a line field, which is a binary GRF, modeled by the joint probability distribution

$$P(\mathbf{l}|\mathbf{g}_{k-1}) = \frac{1}{Q_l} \exp\{-U_l(\mathbf{l}|\mathbf{g}_{k-1})/\beta_l\} \tag{8.25}$$

where Q_l is the partition function and

$$U_l(\mathbf{l}|\mathbf{g}_{k-1}) = \sum_{c \in \mathcal{C}_l} V_l^c(\mathbf{l}|\mathbf{g}_{k-1}) \tag{8.26}$$

Here $\mathcal{C}_l$ denotes the set of all cliques for the line field, and $V_l^c(.)$ represents the clique function for $c \in \mathcal{C}_l$.

Like the occlusion model, the motion discontinuity model assigns a penalty for the use of the "ON" state in the line field. The potential function $V_l^c(\mathbf{l}|\mathbf{g}_{k-1})$ may take different forms depending on the desired properties of the motion discontinuity field. Here, we will consider only singleton and four-pixel cliques; hence,

$$V_l^c(\mathbf{l}|\mathbf{g}_{k-1}) = V_l^{c_1}(\mathbf{l}|\mathbf{g}_{k-1}) + V_l^{c_4}(\mathbf{l}) \tag{8.27}$$

The singleton cliques assign a suitable penalty for the use of the "ON" state at each individual site. The motion discontinuities are, in general, correlated with intensity discontinuities; that is, every motion edge should correspond to an intensity edge, but not vice versa. Thus, a line-field site should only be turned "ON" when there is a significant gray-level gradient, resulting in the following singleton clique potential:

$$V_l^{c_1}(\mathbf{l}|\mathbf{g}_{k-1}) = \begin{cases} \frac{\alpha}{(\nabla_v \mathbf{g}_k)^2} l(\mathbf{x}_i, \mathbf{x}_j) & \text{for horizontal cliques} \\ \frac{\alpha}{(\nabla_h \mathbf{g}_k)^2} l(\mathbf{x}_i, \mathbf{x}_j) & \text{for vertical cliques} \end{cases} \tag{8.28}$$

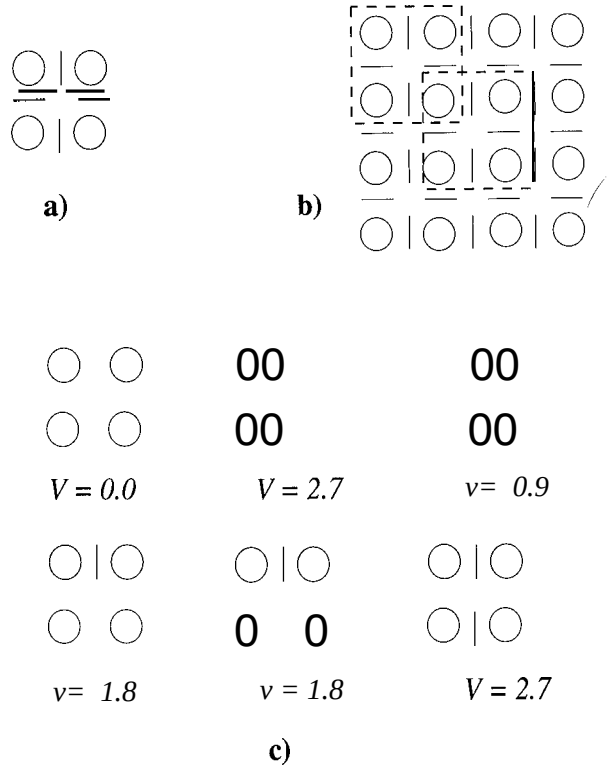


Figure 8.3: Illustration of the line field: a) four-line clique, b) all four-line cliques on a 4 x 4 image, c) potentials for four-line cliques.

where $\nabla_v \mathbf{g}_k$ and $\nabla_h \mathbf{g}_k$ denote the vertical and horizontal image gradient operators, respectively.

An example of potentials assigned to various rotation-invariant four-pixel cliques is shown in Figure 8.3 (c). More examples can be found in [Kon 92]. Next, we demonstrate the use of the line field.

Example: Demonstration of *the line field*

The line field model is demonstrated in Figure 8.3 using a 4 x 4 image. As shown in Figure 8.3 (b), a 4 x 4 image has 24 singleton line cliques and 9 distinct four-line cliques, indicating possible discontinuities between every pair of horizontal and vertical pixel sites.

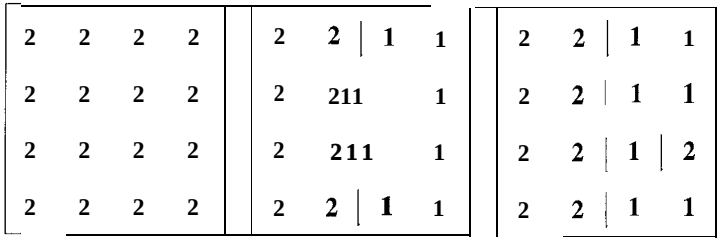


Figure 8.4: Prior probabilities with and without the line field: a) no edges, b) a vertical edge, c) a vertical edge and an isolated pixel.

The potentials shown in Figure 8.3 (c) assign a priori probabilities to all possible four-pixel discontinuity configurations reflecting our a *priori* expectation of their occurence. Note that these configurations are rotation-invariant; that is, the potential is equal to 2.7 whenever only one of the four line sites is ON. These potentials slightly penalize straight lines ($V = 0.9$), penalize corners ($V = 1.8$) and "T" junctions ($V = 1.8$), and heavily penalize end of a line ($V = 2.7$) and "crosses" ($V = 2.7$).

Figure 8.4 shows three pictures where there are no edges, there is a single vertical edge, and there is a vertical edge and an isolated pixel, respectively. The potential function $V_l^{c_4}(\mathbf{l})$ evaluated for each of these configurations are 0, 3 x 0.9 = 2.7, and 2.7 + 1.8 x 2 = 6.3, respectively. Recalling that the a priori probability is inversely proportional to the value of the potential, we observe that a smooth configuration has a higher a priori probability.

Konrad-Dubois Method

Given the above models, the MAP estimate of $\mathbf{d}_1$, $\mathbf{d}_2$, o, and **1** can be expressed (neglecting the dependency of the partition functions on the unknowns, e.g., Q_d depends on the number of sites at which **1** is "ON") in terms of the potential functions as

$$\hat{\mathbf{d}}_1, \hat{\mathbf{d}}_2, \hat{\mathbf{o}}, \hat{\mathbf{l}} = \arg \min_{\mathbf{d}_1, \mathbf{d}_2, \mathbf{o}, \mathbf{l}} U(\mathbf{d}_1, \mathbf{d}_2, \mathbf{o}, \mathbf{l} | \mathbf{g}_k, \mathbf{g}_{k-1}) \quad \textbf{(8.29)}$$

$$= \arg \min_{\mathbf{d}_1, \mathbf{d}_2, \mathbf{o}, \mathbf{l}} \{U_g(\mathbf{g}_k | \mathbf{d}_1, \mathbf{d}_2, \mathbf{o}, \mathbf{l}, \mathbf{g}_{k-1}) + \lambda_d U_d(\mathbf{d}_1, \mathbf{d}_2 | \mathbf{o}, \mathbf{l}, \mathbf{g}_{k-1}) + \lambda_o U_o(\mathbf{o}|\mathbf{l}, \mathbf{g}_{k-1}) + \lambda_l U_l(\mathbf{l}|\mathbf{g}_{k-1})\} \quad \textbf{(8.30)}$$

where λ_d, λ_o, and λ_l are positive constants. The minimization of (8.30) is an exceedingly difficult problem, since there are several hundreds of thousands of unknowns for a reasonable size image, and the criterion function is nonconvex. For example, for a 256 x 256 image, there are 65,536 motion vectors (131,072 components), 65,536 occlusion labels, and 131,072 line field labels for a total of 327,680 unknowns. An additional complication is that the motion vector components are continuous-valued, and the occlusion and line field labels are discrete-valued.

To address these difficulties, Dubois and Konrad [Dub 93] have proposed the following three-step iteration:

1. Given the best estimates of the auxilary fields $\hat{\mathbf{o}}$ and $\hat{\mathbf{l}}$, update the motion field $\mathbf{d}_1$ and $\mathbf{d}_2$ by minimizing

$$\min_{\mathbf{d}_1, \mathbf{d}_2} U_g(\mathbf{g}_k | \mathbf{d}_1, \mathbf{d}_2, \hat{\mathbf{o}}, \mathbf{g}_{k-1}) + \lambda_d U_d(\mathbf{d}_1, \mathbf{d}_2 | \hat{\mathbf{l}}, \mathbf{g}_{k-1}) \quad (8.31)$$

   The minimization of (8.31) can be done by Gauss-Newton optimization, as was done in [Kon 91].

2. Given the best estimates of $\hat{\mathbf{d}}_1$, $\hat{\mathbf{d}}_2$, and $\hat{\mathbf{l}}$, update o by minimizing

$$\min_{\mathbf{o}} U_g(\mathbf{g}_k | \hat{\mathbf{d}}_1, \hat{\mathbf{d}}_2, \mathbf{o}, \mathbf{g}_{k-1}) + \lambda_o U_o(\mathbf{o}|\hat{\mathbf{l}}, \mathbf{g}_{k-1}) \quad \textbf{(8.32)}$$

   An exhaustive search or the ICM method can be employed to solve this step.

3. Finally, given the best estimates of $\hat{\mathbf{d}}_1$, $\hat{\mathbf{d}}_2$, and $\hat{\mathbf{o}}$, update **1** by minimizing

$$\min_{\mathbf{l}} \lambda_d U_d(\hat{\mathbf{d}}_1, \hat{\mathbf{d}}_2 | \mathbf{l}, \mathbf{g}_{k-1}) + \lambda_o U_o(\hat{\mathbf{o}}|\mathbf{l}, \mathbf{g}_{k-1}) + \lambda_l U_l(\mathbf{l}|\mathbf{g}_{k-1}) \quad \textbf{(8.33)}$$

   Once all three fields are updated, the process is repeated until a suitable criterion of convergence is satisfied. This procedure has been reported to give good results.

In an earlier paper Konrad and Dubois [Kon 92] have derived a solution, using the Gibbs sampler, for the minimization of

$$U_g(\mathbf{g}_k | \mathbf{d}_1, \mathbf{d}_2, \mathbf{l}, \mathbf{g}_{k-1}) + \lambda_d U_d(\mathbf{d}_1, \mathbf{d}_2 | \mathbf{l}, \mathbf{g}_{k-1}) + \lambda_l U_l(\mathbf{l}|\mathbf{g}_{k-1}) \quad \textbf{(8.34)}$$

without taking the occlusion field into account. They have shown that the Horn and Schunck iteration (5.17) constitutes a special case of this Gibbs sampler solution. Furthermore, if the motion vector field is assumed discrete-valued, then the Gibbs sampler solution generalizes the pixel-/block-matching algorithms.

While the MAP formulation proposed by Konrad and Dubois provides an elegant approach with realistic constraints for motion estimation, the implementation of this algorithm is far from easy. Alternative approaches proposed for Bayesian motion estimation include those using mean field annealing by Adelqader et al. [Abd 92] and Zhang et al. [Zha 93] to reduce the computational load of the stochastic relaxation procedure, and the multimodal motion estimation and segmentation algorithm of Heitz and Bouthemy [Hei 90]. Pixel-recursive estimators have been derived by Driessen et al. [Dri 92], and Kalman-type recursive estimators have been used by Chin *et* al. [Chi 93]. In the following, we discuss two other strategies to prevent blurring of motion boundaries without using line fields.

8.3.2 Estimation with Local Outlier Rejection

The local outlier rejection approach proposed by Iu [Iu 93] is an extension of the basic MAP formulation, given by (8.7), which aims at preserving the optical flow boundaries without using computationally expensive line field models. At site x, the basic MAP method clearly favors the estimate **d(x)** that is closest to all other motion estimates within the neighborhood of site x. The local outlier rejection method is based on the observation that if the pixels within the neighborhood of site x exhibit two different motions, the estimate **d(x)** is then "pushed" towards the average of the two, resulting in blurring. To eliminate this undesirable effect, it is proposed that all the values of the clique potential function, one for each site, are ranked, and the outlier values are rejected. The outlier rejection procedure is relatively simple to incorporate into the local Gibbs potential calculation step.

To describe the approach in more detail, we revisit the basic MAP equations (8.7)-(8.9). While the likelihood model (8.8) remains unchanged, we rewrite the potential function of the prior distribution (8.9) as

$$U_{\mathbf{d}}(\mathbf{d}_1, \mathbf{d}_2|\mathbf{g}_{k-1}) = \lambda_d \sum_{\mathbf{x}_i} \frac{1}{N_h} \sum_{\mathbf{x}_j \in \mathcal{N}_{\mathbf{x}_i}} \|\mathbf{d}(\mathbf{x}_i) - \mathbf{d}(\mathbf{x}_j)\|^2 \tag{8.35}$$

where the summation $\mathbf{x}_i$ is over the whole image and the summation $\mathbf{x}_j$ runs over all the neighbors of the site $\mathbf{x}_i$, and N_h is the number of neighbors of the site $\mathbf{x}_i$. The term $\|\mathbf{d}(\mathbf{x}_i) - \mathbf{d}(\mathbf{x}_j)\|^2$ is the clique potential for the two-pixel clique containing sites $\mathbf{x}_i$ and $\mathbf{x}_j$. Observe that (8.35) is different than (8.9) because: i) the summation is not over all cliques, but over all pixels to simplify the outlier rejection process (it can be easily shown that in (8.35) every clique is counted twice); and ii) the potential function includes the mean clique potential, rather than the sum of all clique potentials, for each site (indeed, this corresponds to scaling of λ_d).

To incorporate outlier rejection, the potential function (8.35) is further modified as

$$U(\mathbf{d}_1, \mathbf{d}_2|\mathbf{g}_{k-1}) = \lambda_d \sum_{\mathbf{x}_i} \frac{1}{N_h} \sum_{\mathbf{x}_j \in \mathcal{N}_{\mathbf{x}_i}} \delta_j \|\mathbf{d}(\mathbf{x}_i) - \mathbf{d}(\mathbf{x}_j)\|^2 \tag{8.36}$$

where

$$\delta_j = \begin{cases} 1 \text{ if } \|\mathbf{d}(\mathbf{x}_i) - \mathbf{d}(\mathbf{x}_j)\|^2 \leq T_{OR} \\ 0 \text{ otherwise} \end{cases} \tag{8.37}$$

is the indicator function of the rejected cliques, T_{OR} is a threshold, and

$$N_h = \sum \delta_j \tag{8.38}$$

The expression (8.36) can be used in two ways: i) given a fixed threshold T_{OR}, in which case the number of cliques N_h varies from pixel to pixel, or ii) given a fixed number of cliques, in which case the threshold T_{OR} varies from pixel to pixel.

The selection of the right threshold or number of cliques is a compromise between two conflicting requirements. A low threshold preserves the optical flow boundaries better, while a high threshold imposes a more effective smoothness constraint. For an 8-pixel neighborhood system, Iu suggests setting $N_h = 3$. For most neighborhood systems, the ranking of clique potentials and the outlier rejection procedure only account for a small computational overhead.

8.3.3 Estimation with Region Labeling

Yet another extension of the basic MAP approach (8.7) to eliminate blurring of optical flow boundaries can be formulated based on region labeling [Sti 93]. In this approach, each motion vector is assigned a label $z(\mathbf{x})$ such that the label field z designates regions where the motion vectors are expected to remain constant or vary slowly. Region labeling differs from the line-field approach in that here the labels belong to the motion vectors themselves as opposed to indicating the presence of discontinuities between them. Clearly, the label field is unknown, and has to be estimated along with the motion field.

The region-labeling formulation can be expressed as

$$p(\mathbf{d}_1, \mathbf{d}_2, \mathbf{z} \mid \mathbf{g}_k, \mathbf{g}_{k-1}) \propto p(\mathbf{g}_k \mid \mathbf{d}_1, \mathbf{d}_2, \mathbf{z}, \mathbf{g}_{k-1}) p(\mathbf{d}_1, \mathbf{d}_2 \mid \mathbf{z}) p(\mathbf{z}) \tag{8.39}$$

where $p(\mathbf{g}_k \mid \mathbf{d}_1, \mathbf{d}_2, \mathbf{z}, \mathbf{g}_{k-1})$ is again the likelihood model and $p(\mathbf{d}_1, \mathbf{d}_2 \mid \mathbf{z})p(\mathbf{z}) = p(\mathbf{d}_1, \mathbf{d}_2, \mathbf{z})$ is the joint prior pdf of the motion and label fields. While the likelihood model follows straightforwardly from (8.8), we will examine the prior pdf model in more detail. The prior pdf of the motion field conditioned on the label field $p(\mathbf{d}_1, \mathbf{d}_2 \mid \mathbf{z})$ is modeled by a Gibbs distribution,

$$p(\mathbf{d}_1, \mathbf{d}_2 \mid \mathbf{z}) = \frac{1}{Q} \exp\{-U(\mathbf{d}_1, \mathbf{d}_2 \mid \mathbf{z})\} \tag{8.40}$$

where

$$U(\mathbf{d}_1, \mathbf{d}_2 \mid \mathbf{z}) = \sum_{\mathbf{x}_i} \sum_{\mathbf{x}_j \in \mathcal{N}_{\mathbf{x}_i}} \|\mathbf{d}(\mathbf{x}_i) - \mathbf{d}(\mathbf{x}_j)\|^2 \delta(z(\mathbf{x}_i) - z(\mathbf{x}_j)) \tag{8.41}$$

in which $\mathbf{x}_i$ ranges over all pixel sites, and $\mathbf{x}_j$ over all neighbors of $\mathbf{x}_i$. The function $\delta(z)$ is the Kronecker delta function, that is, 1 when $z = 0$ and 0 otherwise. It ensures that the local smoothness constraint is imposed only when both pixels in the clique have the same label, thus avoiding the smoothness constraint across region boundaries. The other part of this prior pdf, p(z), enforces a connected configuration of regions (labels) over the image. A suitable prior pdf model is a discrete-valued Gibbs distribution with the potential function given by (8.11).

The MAP estimate of the triplet $\mathbf{d}_1$, $\mathbf{d}_2$, and z can be obtained by minimizing the potential function corresponding to the a posteriori pdf (8.39). An ICM optimization method has been used by Stiller [Sti 93]. While this approach provides a simpler formulation than that of Konrad-Dubois, the performance is dependent on

the appropriate region assignments. Since the region labels are used only as a token to limit the motion smoothness constraint, there exists a certain degree of latitude and arbitrariness in the assignment of labels.

8.4 Examples

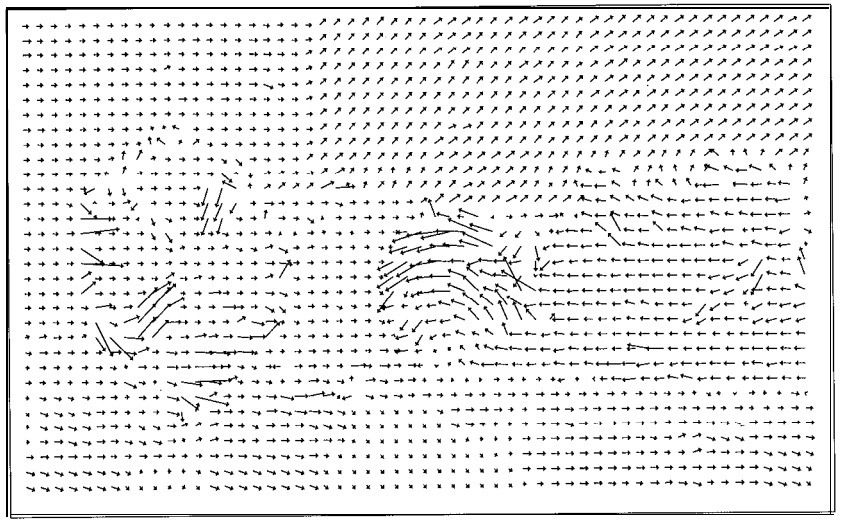
We have implemented four Bayesian motion estimation algorithms; namely, the basic method, the method of Konrad-Dubois (K-D), the outlier-rejection method of Iu, and the segmentation method of Stiller, on the same two frames of the Mobile and Calendar sequence that were used in the previous three chapters. The basic algorithm using a smoothness constraint evaluates (8.7) given the pdf models (8.8) and (8.9) with the potential function (8.10). In our implementation, the DFD in the exponent of (8.8) has been smoothed over the 8-neighborhood of each pixel using the motion estimate at that pixel. The initial iteration does not impose the smoothness constraint; that is, it resembles block matching. The optimization is performed by the ICM method with a temperature schedule of the form T = $1000(1 - i/I)$, where i = 0,. and I stand for the iteration index and the maximum number of iterations, respectively. The algorithm usually converges after $I = 5$ iterations.

For the K-D, Iu, and Stiller algorithms, the initial motion field is set equal to the result of the basic algorithm. Our implementation of the K-D algorithm did not include an occlusion field. Hence, it is a two-step iteration, given by (8.31) and (8.33). The horizontal and vertical line fields are initialized by simple edge detection on the initial motion estimates. The method of Iu is a variation of the basic algorithm, where an outlier-rejection stage is built into the smoothness constraint. At each site, the distance between the present motion vector and those within the S-neighborhood of that site are rank-ordered. The smallest third (N_h = 3) are used in imposing the smoothness constraint. Stiller's algorithm employs a segmentation field, which is initialized by a K-means segmentation of the initial motion estimates with K = 9. In the ICM iterations, we have used an exponential temperature schedule of the form T = $1000(0.8)^i$, i = 0,.

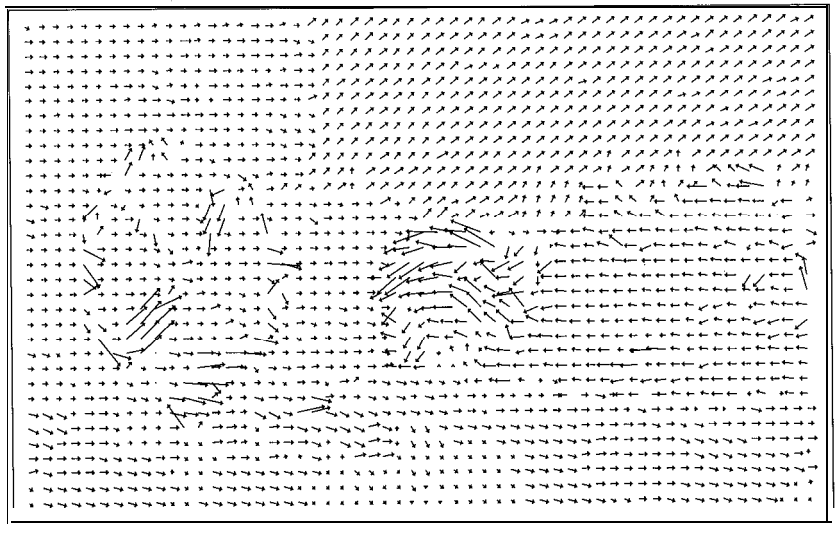
The motion fields obtained by the K-D and the Iu estimators are shown in Figure 8.5 (a) and (b), respectively. Observe that Iu's motion field is slightly more regular. A numerical comparison of the motion fields is presented in Table 8.1.

Table 8.1: Comparison of the Bayesian methods. (Courtesy Gozde Bozdagi)

Method	PSNR (dB)	Entropy (bits)
Frame-Difference	23.45	-
Global Smoothness	33.82	3.92
Konrad-Dubois	34.10	5.02
Iu	34.54	4.99
Stiller	34.14	4.03



(a)



(b)

Figure 8.5: Motion field obtained by a) the Konrad-Dubois method and b) Iu's method. (Courtesy Gozde Bozdagi and Michael Chang)

8.5 Exercises

1. How do you define the joint pdf of a discrete-valued GRF?

2. The initial motion field for the MAP estimator is usually computed by a deterministic estimator such as the Horn-Schunck estimator or block matching. Suggest methods to initialize the line field and the occlusion field.

3. The MAP estimator (8.30) has been found to be highly sensitive to the values of the parameters λ_d, λ_o, and λ_l, which are free parameters. How would you select them?

4. Discuss the relationship between the MAP estimator and the Horn-Schunck algorithm. (Hint: see [Kon 92].)

5. Compare modeling motion discontinuities by line fields versus region labeling.

Bibliography

[Abd 92] I. Abdelqader, S. Rajala, W. Snyder, and G. Bilbro, "Energy minimization approach to motion estimation," Signal Processing, vol. 28, pp. 291–309, Sep. 1992.

[Bes 74] J. Besag, "Spatial interaction and the statistical analysis of lattice systems," *J. Royal* Stat. *Soc. B*, vol. 36, no. 2, pp. 1922236, 1974.

[Bil 91a] G. L. Bilbro and W. E. Snyder, "Optimization of functions with many minima," *IEEE* Trans. Syst., Man *and Cyber.*, vol. 21, pp. 8406849, Jul/Aug. 1991.

[Bil 91b] G. L. Bilbro, W. E. Snyder, and R. C. Mann, "Mean field approximation minimizes relative entropy," *J. Opt. Soc. Am. A*, vol. 8, pp. 290-294, Feb. 1991.

[Bil 92] G. L. Bilbro, W. E. Snyder, S. J. Garnier, and J. W. Gault, "Mean field annealing: A formalism for constructing GNC-like algorithms," *IEEE Trans.* Neural *Networks*, vol. 3, pp. 131-138, Jan. 1992.

[Chi 93] T. M. Chin, M. R. Luettgen, W. C. Karl, and A. S. Willsky, "An estimation theoretic prespective on image processing and the calculation of optical flow," in *Motion Analysis* and Image Sequence Processing, M. I. Sezan and R. L. Lagendijk, eds., Norwell, MA: Kluwer, 1993.

[Cho 90] P. B. Chou and C. M. Brown, "The theory and practice of Bayesian image labeling," *Int. J. Comp. Vis.*, vol. 4, pp. 185-210, 1990.

[Dep 92] R. Depommier and E. Dubois, "Motion estimation with detection of occlusion areas," *Proc. ICASSP*, pp. 111.2699272, Mar. 1992.

[Dri 92] J. N. Driessen, Motion *Estimation* for Digital *Video*, Ph.D. Thesis, Delft University of Technology, 1992.

[Dub 93] E. Dubois and J. Konrad, "Estimation of 2-D motion fields from image sequences with application to motion-compensated processing," in *Motion* Analysis and Image *Sequence Processing*, M. I. Sezan and R. L. Lagendijk, eds., Norwell, MA: Kluwer, 1993.

[Gem 84] S. Geman and D. Geman, "Stochastic relaxation, Gibbs distribution, and Bayesian restoration of images," *IEEE* Trans. *Patt.* Anal. *Mach. Intel.*, vol. 6, pp. 721-741, Nov. 1984.

[Hei 90] F. Heitz and P. Bouthemy, "Motion estimation and segmentation using a global Bayesian approach," *Proc.* Int. Conf. *ASSP*, Albuquerque, NM, pp. 230552308, April 1990.

[Kir 83] S. Kirkpatrick, C. D. G. Jr., and M. P. Vecchi, "Optimization by simulated annealing," Science, vol. 220, pp. 671-679, May 1983.

[Kon 91] J. Konrad and E. Dubois, "Comparison of stochastic and deterministic solution methods in Bayesian estimation of 2D motion," Image and Vis. *Comput.*, vol. 9, pp. 215-228, Aug. 1991.

[Kon 92] J. Konrad and E. Dubois, "Bayesian estimation of motion vector fields," *IEEE* Trans. *Patt.* Anal. Mach. *Intel.*, vol. 14, pp. 910-927, Sep. 1992.

[Laa 87] P. J. M. Van Laarhoven and E. H. Aarts, Simulated Annealing: *Theory and Applications*, Dordrecht, Holland: Reidel, 1987.

[Met 53] **N.** Metropolis, A. Rosenbluth, M. Rosenbluth, H. Teller, and E. Teller, "Equation of state calculations by fast computing machines," *J. Chem. Phys.*, vol. 21, pp. 1087-1092, June 1953.

[Orl 85] H. Orland, "Mean-field theory for optimization problems," J. Physique Lett., vol. 46, 17, pp. 763-770, 1985.

[Pap 92] T. N. Pappas, "An Adaptive Clustering Algorithm for Image Segmentation," *IEEE* Trans. *Signal Proc.*, vol. SP-40, pp. 901-914, April 1992.

[Iu 93] S.-L. Iu, "Robust estimation of motion vector fields with discontinuity and occlusion using local outliers rejection," *SPIE*, vol. 2094, pp. 588-599, 1993.

[Sti 93] C. Stiller and B. H. Hurtgen, "Combined displacement estimation and segmentation in image sequences," *Proc. SPIE/EUROPTO Video Comm.* and *PACS* for Medical *Applications*, Berlin, Germany, vol. SPIE 1977, pp. 276-287, 1993.

[Zha 93] J. Zhang and J. Hanauer, "The mean field theory for image motion estimation," *Proc. IEEE Int. Conf. ASSP*, vol. 5, pp. 197-200, Minneapolis, MN, 1993.

Chapter 9

METHODS USING POINT CORRESPONDENCES

3-D motion estimation refers to estimating the actual motion of objects in a scene from their 2-D projections (images). Clearly, the structure (depth information) of the objects in the scene affect the projected image. Since the structure of the scene is generally unknown, 3-D motion and structure estimation need to be addressed simultaneously. Applications of 3-D motion and structure estimation include robotic vision, passive navigation, surveillance imaging, intelligent vehicle highway systems, and harbor traffic control, as well as object-based video compression. In some of these applications a camera moves with respect to a fixed environment, and in others the camera is stationary, but the objects in the environment move. In either case, we wish to determine the 3-D structure and the relative motion parameters from a time sequence of images.

Needless to say, 3-D motion and structure estimation from 2-D images is an ill-posed problem, which may not have a unique solution without some simplifying assumptions, such as rigid motion and a parametric surface. It is well-known that 3-D rigid motion can be modeled by three translation and three rotation parameters (see Chapter 2). The object surface may be approximated by a piecewise planar or quadratic model, or represented by a collection of independent 3-D points. Then, 3-D motion estimation refers to the estimation of the six rigid motion parameters, and structure estimation refers to the estimation of the parameters of the surface model, or the depth of each individual point from at least two 2-D projections.

The 3-D motion and structure estimation methods can be classified into two groups: those which require a set of feature point correspondences to be determined a priori, and those which do not [Agg 88]. This chapter is about the former class. Furthermore, it is assumed in this chapter that the field of view contains a single moving object. The segmentation problem in the presence of multiple moving objects will be dealt with in Chapter 11. Section 9.1 introduces parametric models for the projected displacement field that are based on the orthographic and perspective projection, respectively. 3-D motion and structure estimation using the

orthographic displacement field model is covered in Section 9.2, whereas Section 9.3 deals with estimation using the perspective displacement field model.

9.1 Modeling the Projected Displacement Field

We start by deriving parametric models for the 2-D projected displacement field based on the orthographic and perspective projections of the 3-D rigid motion model, respectively.

Suppose a point X $= [X_1, X_2, X_3]^T$ on a rigid object at time t moves to X' = $[X_1', X_2', X_3']^T$ at time t' subject to a rotation described by the matrix **R** and a translation by the vector **T.** Then, from Chapter 2, we have

$$\begin{bmatrix} X_1' \\ X_2' \\ X_3' \end{bmatrix} = \mathbf{R} \begin{bmatrix} X_1 \\ X_2 \\ X_3 \end{bmatrix} + \mathbf{T} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \\ X_3 \end{bmatrix} + \begin{bmatrix} T_1 \\ T_2 \\ T_3 \end{bmatrix} \quad (9.1)$$

Recall that in the case of small rotation, the composite rotation matrix can be expressed, in terms of the Eulerian angles, as

$$\mathbf{R} = \begin{bmatrix} 1 & -\Delta\phi & \Delta\psi \\ \Delta\phi & 1 & -\Delta\theta \\ -\Delta\psi & \Delta\theta & 1 \end{bmatrix} \quad (9.2)$$

where $\Delta\theta$, $\Delta\psi$, and $\Delta\phi$ denote small clockwise angular displacements about the Xi, X_2, and X_3 axes, respectively.

9.1.1 Orthographic Displacement Field Model

The orthographic displacement field refers to the orthographic projection of the 3-D displacement vectors into the image plane, which is obtained by substituting X_1' and X_2' from (9.1) into

$$x_1' = X_1' \quad \text{and} \quad x_2' = X_2' \quad (9.3)$$

that define the orthographic projection as discussed in Chapter 2. Since we have Xi $= x_1$ and $X_2 = x_2$ under the orthographic projection, the resulting model is given by

$$\begin{aligned} x_1 &= r_{11}x_1 + r_{12}x_2 + (r_{13}X_3 + T_1) \\ x_2 &= r_{21}x_1 + r_{22}x_2 + (r_{23}X_3 + T_2) \end{aligned} \quad (9.4)$$

The model (9.4) is an affine mapping of the pixel (xi, x_2) at frame t to the pixel (x_1', x_2') at frame t' defined in terms of the six parameters r_{11}, r_{12}, $(r_{13}X_3 + T_1)$, r_{21}, r_{22}, and $(r_{23}X_3 + T_2)$. Thus, it constitutes a parametric 2-D motion field model (see Section 5.2).

It should be noted that the actual distance of the object points from the image plane is not observable in orthographic projection, since it is a parallel projection. Thus, if we express the actual depth of a point as $\tilde{X}_3 = \bar{X}_3 + X_3$, where $\bar{X}_3$ is the depth of a reference point on the object, and X_3 is the relative depth of all other object points with respect to the reference point, we can only expect to estimate X_3 associated with each image point. Further observe that in (9.4), X_3 multiplies r_{13} and r_{23}. Obviously, if we scale r_{13} and r_{23}, a new value of $\bar{X}_3$ can be found for each pixel that would also satisfy (9.4). Hence, these variables cannot be uniquely determined for an arbitrary surface from two views, as stated in [Hua 89a].

Orthographic projection is a reasonable approximation to the actual imaging process described by the perspective projection, when the depth of object points does not vary significantly. Other approximations, such as weak perspective, para-perspective, and orthoperspective projections, also exist in the literature [Dem 92].

9.1.2 Perspective Displacement Field Model

The perspective displacement field can be derived by substituting X_1', X_2', and X_3' from (9.1) into the perspective projection model given by (from Chapter 2)

$$x_1' = f\frac{X_1'}{X_3'} \text{ and } x_2' = f\frac{X_2'}{X_3'} \tag{9.5}$$

to obtain

$$\begin{aligned} x_1 &= f\frac{r_{11}X_1 + r_{12}X_2 + r_{13}X_3 + T_1}{r_{31}X_1 + r_{32}X_2 + r_{33}X_3 + T_3} \\ x_2' &= f\frac{r_{21}X_1 + r_{22}X_2 + r_{23}X_3 + T_2}{r_{31}X_1 + r_{32}X_2 + r_{33}X_3 + T_3} \end{aligned} \tag{9.6}$$

Letting $f = 1$, dividing both the numerator and the denominator by X_3, and expressing the object-space variables in terms of image plane coordinates using the perspective transform expression, we obtain

$$\begin{aligned} x_1 &= \frac{r_{11}x_1 + r_{12}x_2 + r_{13} + \frac{T_1}{X_3}}{r_{31}x_1 + r_{32}x_2 + r_{33} + \frac{T_3}{X_3}} \\ x_2' &= \frac{r_{21}x_1 + r_{22}x_2 + r_{23} + \frac{T_2}{X_3}}{r_{31}x_1 + r_{32}x_2 + r_{33} + \frac{T_3}{X_3}} \end{aligned} \tag{9.7}$$

The expressions (9.7) constitute a nonlinear model of the perspective projected motion field in terms of the image-plane coordinates because of the division by xi and x_2. Notice that this model is valid for arbitrary shaped moving surfaces in 3-D, since the depth of each point X_3 remains as a free parameter. Observe, however, that X_3 always appears in proportion to T_3 in (9.7). That is, the depth information is observable only when $T_3 \neq 0$. Furthermore, it can only be determined up to

a scale factor. For example, an object at twice the distance from the image plane and moving twice as fast yields the same image under the perspective projection.

We develop 3-D motion and structure estimation algorithms based on the affine model (9.4) and the nonlinear model (9.7) in the next two sections. Because of the aforementioned limitations of these projections, we can estimate depth only up to an additive constant at best, using the orthographic displacement field, and only up to a multiplicative scale factor under the perspective displacement field model.

9.2 Methods Based on the Orthographic Model

It is well known that, from two orthographic views, we can estimate the depth of a feature point up to a scale factor α and an additive constant $\bar{X}_3$. The scale ambiguity arises because scaling X_{i3} by α, and r_{13} and r_{23} by $1/\alpha$ results in the same orthographic projection as can be seen from Equation (9.4). That is, if $\tilde{X}_{i3}$ denotes the true depth value, we expect to estimate

$$X_{i3} = \bar{X}_3 + \alpha\tilde{X}_{i3}, \quad \text{for } i = 1, \ldots, N \tag{9.8}$$

from two views. It is not possible to estimate $\bar{X}_3$ under any scheme, because this information is lost in the projection. However, the scale ambiguity may be overcome if more than two views are utilized. In his classical book Ullman [Ull 79] proved that four point correspondences over three frames, i.e. four points each traced from t_1 to t_2 and then to t_3, are sufficient to yield a unique solution to motion and structure up to a reflection. Later, Huang and Lee [Hua 89a] proposed a linear algorithm to obtain the solution in this case. Furthermore, they showed that with three point correspondences over three frames, there are 16 different solutions for motion and four for the structure plus their reflections.

In this section, we concentrate on the two-view problem, and first discuss a simple two-step iteration which was proposed by Aizawa et al. [Aiz 89] as part of the MBASIC algorithm for 3-D model-based compression of video. Two-step iteration is a simple and effective iterative algorithm for 3-D motion and depth estimation when the initial depth estimates are relatively accurate. However, it has been found to be particularly sensitive to random errors in the initial depth estimates. Thus, after introducing the two-step iteration algorithm, we propose an improved algorithm which yields significantly better results with only a small increase in the computational load.

9.2.1 Two-Step Iteration Method from Two Views

The two-step iteration method, proposed by Aizawa et al. [Aiz 89], is based on the following model of the projected motion field:

$$\begin{aligned} x_1' &= x_1 - \Delta\phi x_2 + \Delta\psi X_3 + T_1 \\ x_2' &= \Delta\phi x_1 + x_2 - \Delta\theta X_3 + T_2 \end{aligned} \tag{9.9}$$

which can be obtained by substituting the Eulerian angles definition of the matrix **R** given by (9.2) into (9.4).

In Equation (9.9), there are five unknown global motion parameters $\Delta\theta, \Delta\psi, \Delta\phi$, T_1, and T_2, and an unknown depth parameter X_3 for each given point correspondence (x_1, x_2). This is a bilinear model, since X_3 multiplies the unknown motion parameters. It is thus proposed to solve for the unknowns in two steps: First, determine the motion parameters given the depth estimates from the previous iteration, and then update the depth estimates using the new motion parameters. They are implemented as follows:

1) Given N corresponding coordinate pairs $\mathbf{x}_i = (x_{i1}, x_{i2})$ and $\mathbf{x}'_i = (x'_{i1}, x'_{i2})$ where N $\geq$ 3, and the associated depth estimates X_{i3}, i = 1,. . , N, estimate the five global motion parameters.

This can be accomplished by rearranging Equation (9.9) as

$$\begin{bmatrix} x'_1 - x_1 \\ x'_2 - x_2 \end{bmatrix} = \begin{bmatrix} 0 & X_3 & -x_2 & 1 & 0 \\ -X_3 & 0 & x_1 & 0 & 1 \end{bmatrix} \begin{bmatrix} \Delta\theta \\ \Delta\psi \\ \Delta\phi \\ T_1 \\ T_2 \end{bmatrix} \tag{9.10}$$

Then, writing Equation (9.10) for N corresponding point pairs, we obtain 2N equations in five unknowns, which can be solved for the motion parameters using the method of least squares.

The initial estimates of the depth parameters can be obtained from an a *priori* model of the scene, and the depth estimates are not allowed to vary from these values by more than a predetermined amount, apparently because of the nonuniqueness of the solution. For example, in the case of head-and-shoulder type images, they can be obtained from a scaled wireframe model, as shown in Chapter 24.

2) Once the motion parameters are found, we can estimate the new X_{i3}, $i = 1, \ldots,$ N, using

$$\begin{bmatrix} x'_1 - x_1 + \Delta\phi x_2 - T_1 \\ x'_2 - x_2 - \Delta\phi x_1 - T_2 \end{bmatrix} = \begin{bmatrix} \Delta\psi \\ -\Delta\theta \end{bmatrix} \begin{bmatrix} X_3 & 1 \end{bmatrix} \tag{9.11}$$

which is again obtained by rearranging Equation (9.9). Here, we have an equation pair, for each given point correspondence in one unknown, the depth. The depth for each point correspondence can be solved in the least squares sense from the associated pair (9.11).

The procedure consists of repeating steps 1 and 2 until the estimates no longer change from iteration to iteration. Although theoretically three point correspondences are sufficient, in practice six to eight point correspondences are necessary to obtain reasonably good results due to possible fractional errors in finding the point correspondences. However, as stated even with that many points or more, the two-step iteration may converge to an erronous solution, unless we have very good initial depth estimates X_{i3}, i = 1,. , N.

Suppose that the coordinate system is centered on the object so that $\bar{X}_3 = 0$, and we model the initial depth estimates as

$$X_{i3}^M = \beta X_{i3} + n_i \tag{9.12}$$

where β indicates a systematic error corresponding to a global underscaling or overscaling of the depth, and n_i represents the random errors, which are Gaussian distributed with zero mean. Clearly, it is not possible to estimate the scaling factor β unless the correct depth value of at least one of the N points is known. Assuming that β = 1, the performance of the MBASIC algorithm has been reported to be good when the initial depth estimates contain about 10% random error or less. However, its performance has been observed to degrade with increasing amounts of random error n_i in the initial depth estimates [Boz 94].

9.2.2 An Improved Iterative Method

In the two-step iteration there is strong correlation between the errors in the motion estimates and in the depth estimates. This can be seen from Equations (9.10) and (9.11), where the random errors in the depth estimates are fed back on the motion estimates and vice versa, repeatedly. Thus, if the initial depth estimates are not accurate enough, then the algorithm may converge to an erroneous solution.

To address this problem, we define an error criterion (9.13), and update X_{i3} in the direction of the gradient of the error with an appropriate step size, instead of computing from Equation (9.11), at each iteration. To avoid convergence to a local minimum, we also add a random perturbation to the depth estimates after each update, similar to simulated annealing. The update in the direction of the gradient increases the rate of convergence in comparison to totally random perturbations of X_{i3}. The motion parameters are still computed from Equation (9.10) at each iteration. The improved algorithm can be summarized as:

1. Initialize the depth values X_{i3} for i = 1,. , N. Set the iteration counter m = 0.
2. Determine the motion parameters from (9.10) using the given depth values.
3. Compute $(\tilde{x}_{i1}^{\prime(m)}, \tilde{x}_{i2}^{\prime(m)})$, the coordinates of the matching points that are predicted by the present estimates of motion and depth parameters, using (9.9). Compute the model prediction error

$$E_m = \frac{1}{N} \sum_{i=0}^{N} e_i \tag{9.13}$$

where

$$e_i = (x'_{i1} - \tilde{x}_{i1}^{\prime(m)})^2 + (x'_{i2} - \tilde{x}_{i2}^{\prime(m)})^2 \tag{9.14}$$

Here (x'_{i1}, x'_{i2}) are the actual coordinates of the matching points which are known.

4. If $E_m < \epsilon$, stop the iteration. Otherwise, set m = m + 1, and perturb the depth parameters as

$$\hat{X}_{i3}^{(m)} \leftarrow \hat{X}_{i3}^{(m-1)} - \beta \frac{\partial e_i}{\partial X_3} + \alpha \Delta_i^{(m)} \tag{9.15}$$

where $\Delta_i^{(m)} = N_i(0, \sigma_i^{2(m)})$ is a zero-mean Gaussian random variable with variance $\sigma_i^{2(m)} = e_i$, and α and β are constants.

5. Go to step (2)

A comparison of the performance of the improved algorithm and the two-step iteration is given in [Boz 94], where it was shown that the improved algorithm converges to the true motion and depth estimates even with 50% error in the initial depth estimates.

9.3 Methods Based on the Perspective Model

In the case of perspective projection, the equations that relate the motion and structure parameters to the image-plane coordinates (9.7) are nonlinear in the motion parameters. Early methods to estimate the motion and structure parameters from these expressions required an iterative search which might often diverge or converge to a local minimum [Roa 80, Mit 86]. Later, it was shown that with eight or more point correspondences, a two-step linear algorithm can be developed, where we first estimate some intermediate unknowns called the essential parameters [Hua 86]. The actual motion and structure parameters (for an arbitrary surface) can be subsequently derived from these essential parameters. To this effect, in the following we first present the epipolar constraint and define the essential parameters. Then, a linear least squares algorithm will be given for the estimation of the essential parameters from at least eight pixel correspondences. Finally, methods will be discussed for the estimation of the actual motion and structure parameters from the essential parameters. It should be noted that nonlinear algorithms for motion and structure estimation exist when the number of available point correspondences is five, six, or seven. One such algorithm, the homotopy method, is also briefly introduced.

9.3.1 The Epipolar Constraint and Essential Parameters

We begin by observing that the vectors X', T, and **RX** are coplanar, since X' = **RX** + **T** from (9.1). Then, because (T x **RX)** is orthogonal to this plane, we have

$$\mathbf{X' \ (T \times RX) = 0} \tag{9.16}$$

where x indicates vector product, and indicates dot product. It follows that

$$\mathbf{X'}^T \mathbf{E X} = \mathbf{0} \tag{9.17}$$

where

$$\mathbf{E} \doteq \begin{bmatrix} e_1 & e_2 & e_3 \\ e_4 & e_5 & e_6 \\ e_7 & e_8 & e_9 \end{bmatrix} = \begin{bmatrix} 0 & -T_3 & T_2 \\ T_3 & 0 & -T_1 \\ -T_2 & T_1 & 0 \end{bmatrix} \mathbf{R}$$

The elements of the matrix **E** are called the essential parameters [Hua 86]. Although there are nine essential parameters they are not all independent, because **E** is the product of a skew-symmetric matrix and a rotation matrix.

We divide both sides of (9.17) by $X_3 X_3'$ to obtain a relationship in terms of the image plane coordinates as

$$\begin{bmatrix} x_1' & x_2' & 1 \end{bmatrix} \mathbf{E} \begin{bmatrix} x_1 \\ x_2 \\ 1 \end{bmatrix} = 0 \tag{9.18}$$

which is a linear, homogeneous equation in terms of the nine essential parameters. Equation (9.18) is known as the epipolar constraint for 3-D motion estimation, and is the basis of the linear estimation methods.

The epipolar constraint can alternatively be obtained by eliminating X_3 from the two expressions in the model (9.7) to obtain a single equation

$$\begin{aligned} (T_1 - x_1' T_3) \left[x_2' (r_{31} x_1 + r_{32} x_2 + r_{33}) - (r_{21} x_1 + r_{22} x_2 + r_{23}) \right] \\ = (T_2 - x_2' T_3) \left[x_1' (r_{31} x_1 + r_{32} x_2 + r_{33}) - (r_{11} x_1 + r_{12} x_2 + r_{13}) \right] \end{aligned}$$

which can then be expressed as in (9.18).

It is well known that linear homogeneous equations have either no solution or infinitely many solutions. Thus, we set one of the essential parameters equal to one in (9.18), and solve for the remaining eight essential parameters, which is equivalent to using the essential parameter that is set equal to one as a scale factor. Recall that due to the scale ambiguity problem in the perspective projection, we can estimate the translation and depth parameters only up to a scale factor.

9.3.2 Estimation of the Essential Parameters

We first present a linear least squares method and an optimization method, which require that at least eight point correspondences be known. Then we briefly mention a nonlinear method where only five, six, or seven point correspondences may be sufficient to estimate the essential parameters.

Linear Least Squares Method

Setting es = 1 arbitrarily, we can rearrange (9.18) as

$$\begin{bmatrix} x_1'x_1 & x_1'x_2 & x_1' & x_2'x_1 & x_2'x_2 & x_2' & x_1 & x_2 & 1 \end{bmatrix} \begin{vmatrix} e_1 \\ e_2 \\ e_3 \\ e_4 \\ e_5 \\ e_6 \\ e_7 \\ e_8 \end{vmatrix} = -1$$

which is a linear equation in the remaining eight essential parameters. Given N $\geq$ 8 point correspondences, we can set up a system of N $\geq$ 8 linear equations in 8 unknowns. Conditions for the coefficient matrix to have full rank were investigated by Longuet-Higgins [Lon 81]. Briefly stated, the coefficient matrix has full rank if $\mathbf{T} \neq 0$ and the shape of the 3-D surface satisfies certain conditions, called the surface constraint [Zhu 89]. The solution, then, gives the essential matrix **E** up to a scale factor. We note here that any of the essential parameters could have been chosen as the scale factor. In practice, it is advisable to select the e_i, $i = 1,\ldots,9$, which would yield the coefficient matrix with the smallest condition number as the scale factor.

Optimization Method

Because it is not obvious, in general, which essential parameter should be chosen as a scale factor, an alternative is to use the norm of the solution as a scale factor. Then the estimation of the essential parameters can be posed as a constrained minimization problem [Wen 93a], such that

$$\mathbf{e} = \arg\min_{\mathbf{e}} ||\, \mathrm{Ge} \,||, \qquad \text{subject to } ||\, \mathrm{e} \,|| = 1 \tag{9.19}$$

where $||\ ||$ denotes vector or matrix norm,

$$\mathbf{G} = \begin{bmatrix} x_{11}'x_{11} & x_{11}'x_{12} & x_{11}' & x_{12}'x_{11} & x_{12}'x_{12} & x_{12}' & x_{11} & x_{12} & 1 \\ \vdots & & & & & & & & \vdots \\ x_{N1}'x_{N1} & x_{N1}'x_{N2} & x_{N1}' & x_{N2}'x_{N1} & x_{N2}'x_{N2} & x_{N2}' & x_{N1} & x_{N2} & 1 \end{bmatrix} \tag{9.20}$$

is the observation matrix (derived from (9.18)), N $\geq$ 8, and
$\mathrm{e} = [e_1\ e_2\ \mathrm{es}\ e_4\ e_5\ e_6\ e_7\ \mathrm{es}\ e_9]^T$ denotes the 9 x 1 essential parameter vector. It is well-known that the solution of this problem is the unit eigenvector of $\mathbf{G}^T\mathbf{G}$ associated with its smallest eigenvalue.

The Homotopy Method

When fewer than eight point correspondences are known, or the rank of the coefficient matrix is less than 8, the system of linear equations is underdetermined. However, the fact that the matrix **E** can be decomposed into a skew-symmetric' matrix postmultiplied by a rotation matrix (see Equation 9.21) can be formulated in the form of additional polynomial equations. In particular, the decomposability implies that one of the eigenvalues of **E** is zero, and the other two are equal. Capitalizing on this observation, Huang and Netravali [Hua 90] introduced the homotopy method to address the cases where five, six, or seven point correspondences are available. For example, with five point correspondences, they form five linear equations and three cubic equations, which are then solved by the homotopy method. It has been shown that there are at most ten solutions in this case. The interested reader is referred to [Hua 90].

9.3.3 Decomposition of the E-Matrix

Theoretically, from (9.17), the matrix **E** can be expressed as

$$\mathbf{E} = [\mathbf{e}_1 \mid \mathbf{e}_2 \mid \mathbf{es}] = \left[k\hat{\mathbf{T}} \times \mathbf{r}_1 \mid k\hat{\mathbf{T}} \times \mathbf{r}_2 \mid k\hat{\mathbf{T}} \times \mathbf{r}_3\right] \tag{9.21}$$

where the vectors $\mathbf{r}_i$, $i = 1,\ldots,3$ denote the columns of the rotation matrix **R,** k denotes the length of the translation vector **T,** and **T** is the unit vector along **T.** In the following, we discuss methods to recover **R** and $\hat{\mathbf{T}}$ given **E** computed from noise-free and noisy point correspondence data, respectively.

Noise-Free Point Correspondences

It can easily be observed from (9.21) that each column of **E** is orthogonal to **T.** Then the unit vector along **T** can be obtained within a sign by taking the cross product of two of the three columns as

$$\hat{\mathbf{T}} = \pm\frac{\mathbf{e}_i \times \mathbf{e}_j}{||\mathbf{e}_i \times \mathbf{e}_j||} \qquad i \neq j \tag{9.22}$$

Furthermore, it can be shown that [Hua 86]

$$k^2 = \frac{1}{2}(\mathbf{e}_1\,\mathbf{e}_1 + \mathbf{e}_2\,.\,\mathbf{e}_2 + \mathbf{e}_3\,\mathbf{e}_3) \tag{9.23}$$

Obviously, finding k from (9.23) cannot overcome the scale ambiguity problem, since es contains an arbitrary parameter. However, it is needed to determine the correct rotation parameters.

In order to determine the correct sign of the translation vector, we utilize

$$X_3'\hat{\mathbf{T}} \times [x_1'\ x_2'\ 1]^T = X_3\hat{\mathbf{T}} \times \mathbf{R}[x_1\ x_2\ 1]^T$$

which is obtained by cross-multiplying both sides of (9.1) with $\hat{\mathbf{T}}$, after applying the perspective projection. Then, Zhuang [Zhu 89] shows that the vector $\hat{\mathbf{T}}$ with the correct sign satisfies

$$\sum \left[\hat{\mathbf{T}} \times [x_1' \; x_2' \; 1]^T\right]^T \mathbf{E}[x_1 \; x_2 \; 1]^T > 0 \tag{9.24}$$

where the summation is computed over all observed point correspondences.

Once the sign of the unit translation vector $\hat{\mathbf{T}}$ is determined, the correct rotation matrix can be found uniquely. To this effect, we observe that

$$\mathbf{e}_1 \times \mathbf{e}_2 = k\hat{\mathbf{T}}(k\hat{\mathbf{T}} \cdot \mathbf{r}_3) \tag{9.25}$$

$$\mathbf{e}_2 \times \mathbf{e}_3 = k\hat{\mathbf{T}}(k\hat{\mathbf{T}} \cdot \mathbf{r}_1) \tag{9.26}$$

$$\mathbf{e}_3 \times \mathbf{e}_1 = k\hat{\mathbf{T}}(k\hat{\mathbf{T}} \cdot \mathbf{r}_2) \tag{9.27}$$

The expressions (9.25)-(9.27) can be derived by employing the vector identity

$$\mathbf{A} \times (\mathbf{B} \times \mathbf{C}) = (\mathbf{A} \cdot \mathbf{C})\mathbf{B} - (\mathbf{A} \cdot \mathbf{B})\mathbf{C}$$

In particular,

$$\begin{aligned} \mathbf{e}_1 \times \mathbf{e}_2 &= \mathbf{e}_1 \times (k\hat{\mathbf{T}} \times \mathbf{r}_2) \\ &= [(k\hat{\mathbf{T}} \times \mathbf{r}_1) \cdot \mathbf{r}_2]k\hat{\mathbf{T}} - [(k\hat{\mathbf{T}} \times \mathbf{r}_1)\, k\hat{\mathbf{T}}]\mathbf{r}_2 \end{aligned}$$

Using the properties of the cross product, the first term simplifies as

$$[(\mathbf{r}_1 \times \mathbf{r}_2) \cdot k\hat{\mathbf{T}}]k\hat{\mathbf{T}} = (\mathbf{r}_3 \, k\hat{\mathbf{T}})k\hat{\mathbf{T}}$$

since $\mathbf{r}_1$, $\mathbf{r}_2$, and $\mathbf{r}_3$ are mutually orthogonal and have unity length (recall that they are the column vectors of a rotation matrix). The second term is zero since $(k\hat{\mathbf{T}} \times \mathbf{r}_1)$ is orthogonal to $k\hat{\mathbf{T}}$, yielding the relation (9.25).

Next, observe that we can express the vector $\mathbf{r}_1$ as

$$\mathbf{r}_1 = (\hat{\mathbf{T}} \, \mathbf{r}_1)\hat{\mathbf{T}} + (\hat{\mathbf{T}} \times \mathbf{r}_1) \times \hat{\mathbf{T}} \tag{9.28}$$

Here, the first term denotes the orthogonal projection of $\mathbf{r}_1$ onto $\hat{\mathbf{T}}$, and the second term gives the orthogonal complement of the projection, since

$$(\mathbf{T} \times \mathbf{r}_1) \times \mathbf{T} = ||\mathbf{r}_1|| \sin\beta = \sin\beta$$

where β denotes the angle between $\mathbf{r}_1$ and $\hat{\mathbf{T}}$, as depicted in Figure 9.1.

It follows from (9.28) that if we know the cross product and the dot product of an unknown vector $\mathbf{r}_1$ with a known vector $\hat{\mathbf{T}}$, we can determine the unknown through the relation (9.28). Evaluating the dot product of $\hat{\mathbf{T}}$ with both sides of (9.26) yields

$$\hat{\mathbf{T}} \cdot \mathbf{r}_1 = \frac{1}{k^2}\hat{\mathbf{T}} \cdot (\mathbf{e}_2 \times \mathbf{e}_3)$$

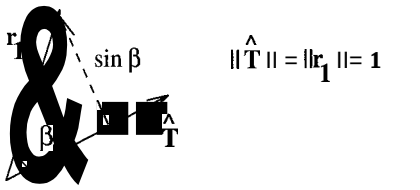


Figure 9.1: Orthogonal projection of $\mathbf{r}_1$ onto $\hat{\mathbf{T}}$.

Recall that we already have

$$\hat{\mathbf{T}} \times \mathbf{r}_1 = \frac{1}{k}\mathbf{e}_1$$

from the definition of the E-matrix (9.21). Substituting these dot and cross product expressions in (9.28), we find

$$\mathbf{r}_1 = \left[\frac{1}{k^2}\hat{\mathbf{T}} \cdot (\mathbf{e}_2 \times \mathbf{e}_3)\right]\hat{\mathbf{T}} + \frac{1}{k}(\mathbf{e}_1 \times \hat{\mathbf{T}}) \tag{9.29}$$

The other column vectors of the rotation matrix $\mathbf{R}$ are given similarly as

$$\mathbf{r}_2 = \left[\frac{1}{k^2}\hat{\mathbf{T}} \, (\mathbf{e}_3 \times \mathbf{e}_1)\right]\hat{\mathbf{T}} + \frac{1}{k}(\mathbf{e}_2 \times \hat{\mathbf{T}}) \tag{9.30}$$

$$\mathbf{r}_3 = \left[\frac{1}{k^2}\hat{\mathbf{T}} \cdot (\mathbf{e}_1 \times \mathbf{e}_2)\right]\hat{\mathbf{T}} + \frac{1}{k}(\mathbf{e}_3 \times \hat{\mathbf{T}}) \tag{9.31}$$

Noisy Point Correspondences

When feature point correspondences are contaminated by noise, we may obtain different estimates of the unit translation vector by using different combinations of the column vectors e_i, $i = 1, 2, 3$ in (9.22), and the estimate of the rotation matrix obtained from (9.29)-(9.31) may no longer satisfy the properties of a rotation matrix. To address these problems, let

$$\mathbf{W} = [\mathbf{w}_1 \; \mathbf{w}_2 \; \mathbf{w}_3] \tag{9.32}$$

denote an estimate of the rotation matrix where $\mathbf{w}_1$, $\mathbf{w}_2$, and $\mathbf{w}_3$ are obtained from (9.29))(9.31). It has been shown that the estimates of $\hat{\mathbf{T}}$ and $\mathbf{R}$, in the case of noisy point correspondences, are given by [Wen 93a]

$$\hat{\mathbf{T}} = \arg\min_{\hat{\mathbf{T}}} \; || \; \mathbf{E}^T\hat{\mathbf{T}} \; ||, \quad \text{subject to } || \; \hat{\mathbf{T}} \; || = 1 \tag{9.33}$$

and

$$\hat{\mathbf{R}} = \arg\min_{\mathbf{R}} || \mathbf{R} - \mathbf{W} ||, \quad \text{subject to } \mathbf{R} \text{ is a rotation matrix} \tag{9.34}$$

respectively.

The solution $\hat{\mathbf{T}}$ of (9.33) is the unit eigenvector of $\mathbf{E}\mathbf{E}^T$ associated with its smallest eigenvalue. Note that the correct sign of $\hat{\mathbf{T}}$ must satisfy (9.24). Let the solution of (9.34) be expressed in terms of the quaternion representation (2.10). Then, $\mathrm{q} = [q_0\ q_1\ q_2\ q_3]^T$ is the unit eigenvector of the 4 x 4 matrix

$$\mathbf{B} = \sum_{i=1}^{3} \mathbf{B}_i^T \mathbf{B}_i$$

where

$$\mathbf{B}_i = \begin{bmatrix} 0 & (\mathbf{I}_i - \mathbf{w}_i)^T \\ \mathbf{w}_i - \mathbf{I}_i & [\mathbf{w}_i - \mathbf{I}_i]_\times \end{bmatrix}$$

such that $\mathbf{I}_i$, $i = 1, 2, 3$ denote the columns of a 3 x 3 identity matrix, and

$$[(x_1\ x_2\ x_3)^T]_\times = \begin{bmatrix} 0 & -x_3 & x_2 \\ x_3 & 0 & -x_1 \\ -x_2 & x_1 & 0 \end{bmatrix}$$

The reader is referred to [Wen 93a] for a derivation of this result.

9.3.4 Algorithm

The method can then be summarized as:

1) Given eight or more point correspondences, estimate **E** up to a scale factor using either the least squares or the optimization method.

2) Compute **T** (up to a scale factor) using (9.33).

3) Find **W** from (9.29)-(9.31) and **R** from (9.34).

Given the rotation matrix **R,** the axis of rotation in terms of its directional cosines (n_1, n_2, n_3) and the incremental angle of rotation $\Delta\alpha$ about this axis can then be determined, if desired, from

$$trace\{\mathbf{R}\} = 1 + 2\ \cos\ \Delta\alpha \tag{9.35}$$

and

$$\mathbf{R} - \mathbf{R}^T = 2 \sin \Delta\alpha \begin{pmatrix} 0 & -n_3 & n_2 \\ n_3 & 0 & -n_1 \\ -n_2 & n_1 & 0 \end{pmatrix} \tag{9.36}$$

4) Once the motion parameters are estimated, solve for the depth parameter X_3 in the least squares sense (up to a scale factor) from

$$\begin{bmatrix} T_1 - x_1' T_3 \\ T_2 - x_2' T_3 \end{bmatrix} = \begin{bmatrix} x_1'(r_{31}x_1 + r_{32}x_2 + r_{33}) - (r_{11}x_1 + r_{12}x_2 + r_{13}) \\ x_2'(r_{31}x_1 + r_{32}x_2 + r_{33}) - (r_{21}x_1 + r_{22}x_2 + r_{23}) \end{bmatrix} (X_3) \tag{9.37}$$

We note that the translation vector T and the depth parameters can only be determined up to a scale factor due to the scale ambiguity inherent in the perspective projection.

Alternative algorithms for the estimation of the 3-D motion and structure parameters from feature correspondences exist in the literature. The two-step linear algorithm presented in this section is favored because it provides an algebraic solution which is simple to compute. However, it is known to be sensitive to noise in the pixel correspondences [Wen 89, Phi 91, Wen 92]. This sensitivity may be attributed to the fact that the epipolar constraint constrains only one of the two components of the image plane position of a 3-D point [Wen 93]. It is well known that the matrix **E** possesses nice properties, such as that two of its eigenvalues must be equal and the third has to be zero [Hua 89b], which the linear method is unable to use. To this effect, Weng et al. [Wen 93] proposed maximum likelihood and minimum variance estimation algorithms which use properties of the matrix **E** as constraints in the estimation of the essential parameters to improve the accuracy of the solution in the presence of noise.

9.4 The Case of 3-D Planar Surfaces

Planar surfaces are an important special case because most real-world surfaces can be approximated as planar at least on a local basis. This fact leads to representation of arbitrary surfaces by 3-D mesh models composed of planar patches, such as the wireframe model widely used in computer graphics. The main reason for treating planar surfaces as a special case is that they do not satisfy the surface assumption required in the general case provided above. Fortunately, it is possible to derive simple linear algorithms for the case of planar surfaces, as described in this section.

9.4.1 The Pure Parameters

We start by deriving a simplified model for the case of planar surfaces. Let the 3-D points that we observe all lie on a plane described by

$$aX_1 + bX_2 + cX_3 = 1 \tag{9.38}$$

where [a ***b*** c]^T denotes the normal vector of this plane. Then, the 3-D displacement model (9.1) can be rewritten as

$$\begin{bmatrix} X_1' \\ X_2' \\ X_3' \end{bmatrix} = \mathbf{R} \begin{bmatrix} X_1 \\ X_2 \\ X_3 \end{bmatrix} + \mathbf{T} [\ a\ b\ c\ ] \begin{bmatrix} X_1 \\ X_2 \\ X_3 \end{bmatrix}$$

or

$$\begin{bmatrix} X_1' \\ X_2' \\ X_3' \end{bmatrix} = \begin{bmatrix} a_1 & a_2 & a_3 \\ a_4 & a_5 & a_6 \\ a_7 & a_8 & a_9 \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \\ X_3 \end{bmatrix} = \mathbf{A} \begin{bmatrix} X_1 \\ X_2 \\ X_3 \end{bmatrix} \tag{9.39}$$

where

$$A = R + T[\, a\ b\ c\,]$$

Next, we map the 3-D displacements onto the 2-D image plane using the perspective transformation, and normalize as $= 1$ due to the well-known scale ambiguity, to obtain the image plane mapping from t to t' given by

$$\begin{aligned} x_1' &= \frac{a_1x_1 + a_2x_2 + a_3}{a_7x_1 + a_8x_2 + 1} \\ x_2' &= \frac{a_4x_1 + a_5x_2 + a_6}{a_7x_1 + a_8x_2 + 1} \end{aligned} \tag{9.40}$$

The constants $a_1, \ldots,$ us are generally known as the pure parameters [Tsa 81]. Next we present a linear algorithm for the estimation of the pure parameters.

9.4.2 Estimation of the Pure Parameters

By cross-multiplying each equation, we can rearrange (9.40) for each given point correspondence, as follows:

$$\begin{bmatrix} x_1 & x_2 & 1 & 0 & 0 & 0 & -x_1x_1' & -x_2x_1' \\ 0 & 0 & 0 & x_1 & x_2 & 1 & -x_1x_2' & -x_2x_2' \end{bmatrix} \begin{vmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \\ a_7 \\ a_8 \end{vmatrix} = \begin{bmatrix} x_1' \\ x_2' \end{bmatrix} \tag{9.41}$$

Therefore, given at least four point correspondences, we can set up eight or more linear equations to solve for the eight pure parameters. It has been shown that the rank of the coefficient matrix is 8 if and only if no three of the four observed points are collinear in three dimensions [Hua 86].

9.4.3 Estimation of the Motion and Structure Parameters

Once the matrix A has been determined, the motion parameters R and T, and the structure parameters, i.e., the normal vector of the plane, can be estimated by

means of a singular value decomposition (SVD) of the matrix A as described in [Tsa 82]. We start by expressing the matrix A as

$$\mathbf{A} = \mathbf{U} \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{bmatrix} \mathbf{V}^T \tag{9.42}$$

where U = [$\mathbf{u}_1$ | $\mathbf{u}_2$ | us] and V = [vi | $\mathbf{v}_2$ | $\mathbf{v}_3$] are 3×3 orthogonal matrices and $\lambda_1 \geq \lambda_2 \geq \lambda_3 \geq 0$ are the singular values of A. There are three possibilities depending on the singular values:

Case 1) The singular values are distinct with $\lambda_1 > \lambda_2 > \lambda_3$, which indicates that the motion can be decomposed into a rotation about an axis through the origin followed by a translation in a direction other than the direction of the normal vector of the plane. Then two solutions for the motion parameters exist, which can be expressed as

$$\begin{aligned} \mathrm{R} &= \mathrm{U} \begin{bmatrix} \alpha & 0 & -s\beta \\ 0 & 1 & 0 \\ -s\beta & 0 & s\alpha \end{bmatrix} \mathbf{V}^T \\ \mathbf{T} &= k\left(-\beta\mathbf{u}_1 + \left(\frac{\lambda_3}{\lambda_2} - s\alpha\right)\mathbf{u}_3\right) \\ [\, a\ b\ c\,]^T &= \frac{1}{k}(\delta\mathbf{v}_1 + \mathbf{v}_3) \end{aligned}$$

where

$$\begin{aligned} \delta &= \pm\left(\frac{\lambda_1^2 - \lambda_2^2}{\lambda_2^2 - \lambda_3^2}\right)^{\frac{1}{2}} \\ \alpha &= \frac{\lambda_1 + s\lambda_3\delta^2}{\lambda_2(1+\delta^2)} \\ \beta &= \frac{1}{\delta}\left(\alpha - \frac{\lambda_1}{\lambda_2}\right) \\ \mathrm{s} &= \det(\mathbf{U})\ \det(\mathbf{V}) \end{aligned}$$

and k is an arbitrary scale factor (positive or negative). The sign ambiguity may be resolved by requiring $1/X_3 > 0$ for all points.

Case 2) If the multiplicity of the singular values is two, e.g., $\lambda_1 = \lambda_2 \neq \lambda_3$, then a unique solution for the motion parameters exist, which is given by

$$\begin{aligned} \mathbf{R} &= \frac{1}{\lambda_1}\mathbf{A} - \left(\frac{\lambda_3}{\lambda_1} - s\right)\mathbf{u}_3\mathbf{v}_3^T \\ \mathbf{T} &= k\left(\frac{\lambda_3}{\lambda_1} - s\right) \\ [\mathrm{a}\ \ \boldsymbol{b}\ \ c\,]^T &= \frac{1}{k}\mathbf{v}_3 \\ s &= \det(\mathbf{U})\ \det(\mathbf{V}) \end{aligned}$$

and k is an arbitrary scale factor, In this case, the motion can be described by a rotation about an axis through the origin followed by a translation along the normal vector of the plane.

Case 3) If the multiplicity of the singular values is three, i.e., $\lambda_1 = \lambda_2 = \lambda_3$, then the motion is a pure rotation around an axis through the origin, and $\mathbf{R}$ is uniquely determined by

$$\mathbf{R} = (\frac{1}{\lambda_1})\mathbf{A}$$

However, it is not possible to determine $\mathbf{T}$ and $[\ a\ b\ c\ ]^T$.

As an alternative method, Tsai and Huang [Tsa 81] obtained a sixth-order polynomial equation to solve for one of the unknowns, and then solved for the remaining unknowns. However, the SVD method provides a closed-form solution.

9.5 Examples

This section provides an experimental evaluation of the performances of some leading 3-D motion and structure estimation algorithms in order to provide the reader with a better understanding of their relative strengths and weaknesses. Results are presented for numerical simulations as well as with two frames of the sequence "Miss America." In each case, we have evaluated methods based on both the orthographic projection (two-step iteration and the improved method) and the perspective projection (E-matrix and A-matrix methods).

9.5.1 Numerical Simulations

Simulations have been performed to address the following questions: 1) Under what conditions can we successfully use the methods based on orthographic projection? 2) Which method, among the four that we compare, performs the best? 3) How sensitive are these methods to errors in the point correspondences? To this effect, we have generated two 3-D data sets, each containing 30 points. The first set consists of 3-D points that are randomly selected within a 100 cm × 100 cm × 0.6 cm rectangular prism whose center is 51.7 cm away from the center of projection; that is, we assume X_1 and X_2 are uniformly distributed in the interval [-50,50] cm, and X_3 is uniformly distributed within [51.4,52] cm. The second set consists of points such that X_1 and X_2 are uniformly distributed in the interval [-25,25] cm, and X_3 is uniformly distributed within [70,100] cm. The image-plane points are computed via the perspective projection, $x_i = fX_i/X_3$, i = 1, 2, with f = 50 mm. In order to find the matching points in the next frame, the 3-D data set is rotated and translated by the "true" motion parameters, and the corresponding image-plane points are recomputed. Clearly, the orthographic projection model provides a good approximation for the first data set, where the range of X_3 (ΔX_3 = 0.6 cm) is small compared to the average value of X_3 ($\bar{X}_3 = 51.7$ cm), since $x_i = fX_i/51.7$, i = 1, 2,

Table 9.1: Comparison of the motion parameters. All angles are in radians, and the translations are in pixels. (Courtesy Gozde Bozdagi)

True	10% Error		30% Error		50% Error	
	Two-Step	Improved	Two-Step	Improved	Two-Step	Improved
A8 = 0.007	0.00689	0.00690	0.00626	0.00641	0.00543	0.00591
$\Delta\psi$ = 0.010	0.00981	0.00983	0.00898	0.00905	0.00774	0.00803
$\Delta\phi$ = 0.025	0.02506	0.02500	0.02515	0.02500	0.02517	0.02501
T_1 = 0.100	0.09691	0.09718	0.08181	0.08929	0.06172	0.07156
T_2 = *0.180*	0.18216	0.18212	0.19240	0.19209	0.20660	0.17011

more-or-less describes all image-plane points. On the contrary, for the second set ΔX_3 = 30 cm is not small in comparison to $\bar{X}_3$ = 85 cm; hence, the orthographic projection would not be a good approximation.

We have tested the performance of the methods based on the orthographic projection, the two-step iteration, and the improved algorithm on the first data set. The true motion parameters that are used in the simulations are listed in Table 9.1. Recall that both the two-step and the improved iterative algorithms require an initial estimate of the depth values for each point pair. In order to test the sensitivity of these algorithms to random errors in the initial depth estimates, $\pm 10\%$, $\pm 30\%$, or $\pm 50\%$ error has been added to each depth parameter X_{i3}. The sign of the error (+ or -) was chosen randomly for each point. The parameter values $\alpha = 0.95$ and β = 0.3 have been used to obtain the reported results. In the case of the improved algorithm, we iterate until E_m given by (9.13) is less than an acceptable level. In order to minimize the effect of random choices, the results are repeated three times using different seed values. The average results are reported.

Table 9.1 provides a comparison of the motion parameter estimates obtained by the two-step algorithm and the improved algorithm at the conclusion of the iterations. In order to compare the results of depth estimation, we define the following error measure:

$$\text{Error} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\frac{(X_{i3} - \hat{X}_{i3})^2}{(X_{i3})^2}} \tag{9.43}$$

where N is the number of matching points, and X_{i3} and $\hat{X}_{i3}$ are the "true" and estimated depth parameters, respectively. Figures 9.2 (a) and (b) show the error measure plotted versus the iteration number for the cases of 30% and 50% initial error, respectively. Note that the scale of the vertical axis is not the same in both plots. Although the results are depicted for 500 iterations, convergence has resulted in about 100 iterations in almost all cases.

It can be seen from Table 9.1 that the error in the initial depth estimates directly

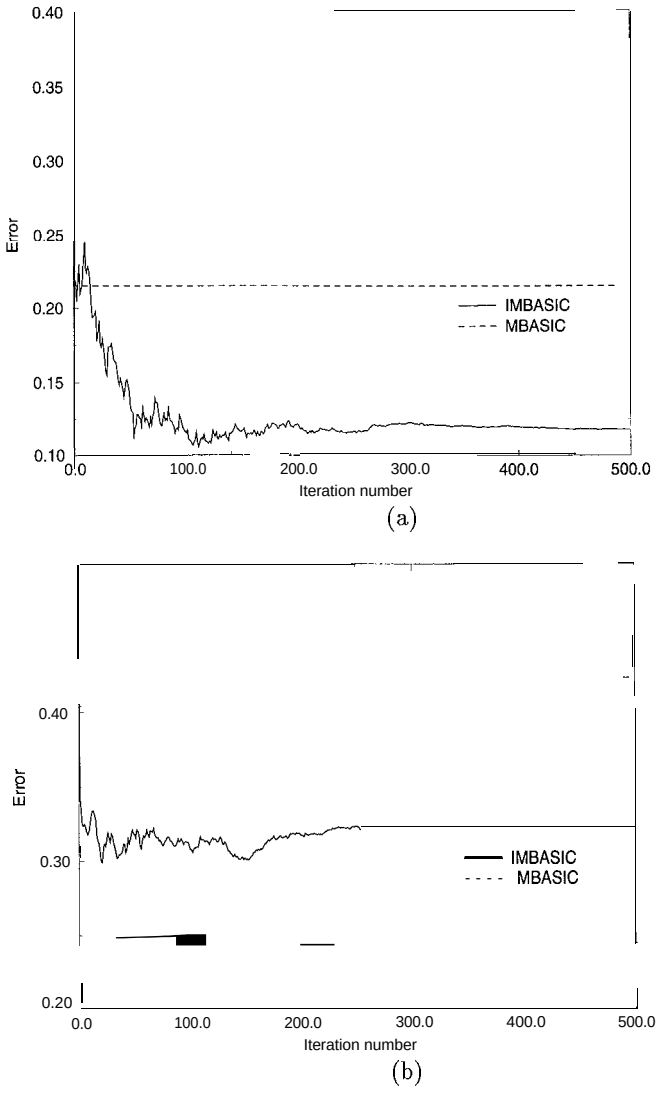


Figure 9.2: Comparisonof depth parameters for a) 30% and b) 50% error. (Courtesy Gozde Bozdagi)

affects the accuracy of A8 and $\Delta\psi$, which are multiplied by X_3 in both equations. Thus, in the two-step algorithm, the error in $\Delta\theta$ and $\Delta\psi$ estimates increases as we increase the error in the initial depth estimates. Furthermore, the error in the depth estimates (at the convergence) increases with increasing error in the initial depth parameters (see Figure 9.2). However, in the improved algorithm, as can be seen from Figure 9.2, the depth estimates converge closer to the correct parameters even in the case of 50% error in the initial depth estimates. For example, in the case of 50% error in the initial depth estimates, the improved method results in about 10% error after 500 iterations, whereas the two-step algorithm results in 45% error, demonstrating the robustness of the improved method to errors in the initial depth estimates.

The maximum difference in the image plane coordinates due to orthographic versus perspective projection of the 3-D object points is related to the ratio of the width of the object to the average distance of the points from the center of projection, which is $\Delta X_3/2\bar{X}_3 \approx 1\%$ for the first data set used in the above experiments. However, this ratio is approximately 18% for the second set, on which our experiments did not yield successful results using either the two-step or the improved algorithm. To provide the reader with a feeling of when we can use methods based on the orthographic projection successfully, we have shown the overall error in the motion and depth estimates given by

$$\begin{aligned}\text{Error} &= (\Delta\phi - \hat{\Delta\phi})^2 + (\Delta\psi - \hat{\Delta\psi}/\alpha)^2(\Delta\theta - \hat{\Delta\theta}/\alpha)^2 + (T_1 - \hat{T_1})^2 \\ &\quad + (T_2 - \hat{T_2})^2 + 1/N \sum_{i=1}^{N} (X_{i3} - \alpha\hat{X}_{i3})^2\end{aligned}$$

where α is a scale factor, as a function of the ratio $\Delta X_3/2\bar{X}_3$ in Figure 9.3. The results indicate that the orthographic approximation yields acceptable results for $\Delta X_3/2\bar{X}_3 < 5\%$, while methods based on the perspective projection are needed otherwise.

We have tested two methods based on the perspective projection, namely the E-matrix and the A-matrix methods, on the second data set. In order to test the sensitivity of both methods to random errors in the point correspondences, $P\%$ error is added to the coordinates of the matching points according to

$$x_i' = x_i + (x_i' - x_i)(1 \pm \frac{P}{100}), \quad i = 1, 2$$

Table 9.2 shows the results of estimation with the true point correspondences as well as with 3% and 10% error in the point correspondences. Recall that the A-matrix method assumes all selected points lie on a planar surface, which is not the case with our data set. As a result, when the amount of noise is small, the E-matrix method outperforms the A-matrix method. Interestingly, the A-matrix method seems to be more robust in the presence of moderate noise in the coordinates of matching points, which may be due to use of a surface model.

Table 9.2: Comparison of the motion parameters. All angles are in radians, and the translations are in pixels. (Courtesy Yucel Altunbasak)

	E-Matrix			A-Matrix		
True	No Error	3% Error	10% Error	No Error	3% Error	10% Error
$\Delta\theta$ =0.007	0.00716	0.00703	0.00938	0.00635	0.00587	0.00470
$\Delta\psi$ = 0.010	0.00991	0.00920	0.00823	0.01467	0.01450	0.01409
$\Delta\phi$ = 0.025	0.02500	0.02440	0.02600	0.02521	0.02513	0.02501
T_2/T_1 = 1.80	1.80014	1.89782	2.15015	1.05759	1.12336	1.28980
T_3/T_1 = 0.48	0.47971	0.50731	-0.98211	0.25627	0.29159	0.37145
Depth error	0.00298	0.02284	0.15549	0.09689	0.09744	0.09974
Match error	1.73 E-6	2.20 E-4	6.57 E-4	1.15 E-3	1.20 E-3	1.61 E-3

The "depth error" refers to the root mean square (RMS) error in the X_3 coordinates of the 30 points selected. We have also calculated the "matching error" which

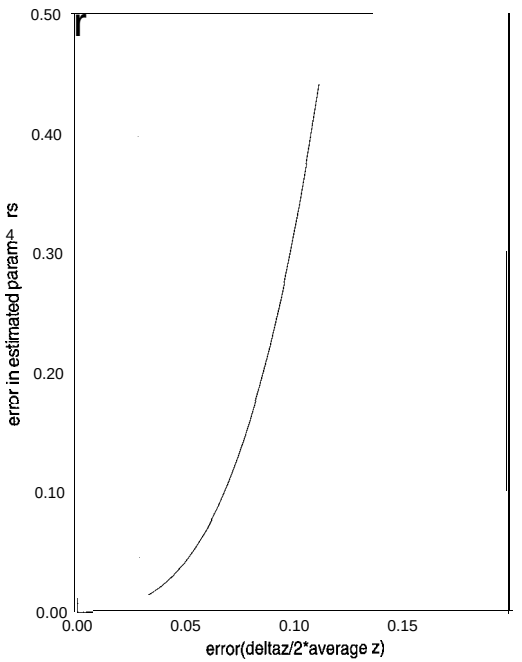


Figure 9.3: The performance of methods based on the orthographic projection (Courtesy Gozde Bozdagi)

is the RMS difference between the coordinates of the matching points calculated with the true parameter values and those predicted by the estimated 3-D motion and structure parameters. The matching error with no motion compensation has been found as 1.97 $E-3$.

9.5.2 Experiments with Two Frames of Miss America

We next tested all four methods on two frames of the "Miss America" sequence, which are shown in Figure 9.4 (a) and (b), respectively. We note that the ratio of the maximum depth on a front view of a typical face to the average distance of the face from the camera falls within the range for which the orthographic projection is a reasonable approximation. Marked in Figure 9.4 (a) are the coordinates of 20 points (x_1, x_2) which are selected as features. The corresponding feature points (x_1', x_2') are likewise marked in Figure 9.4 (b). The coordinates of the matching points have been found by hierarchical block matching to the nearest integer. This truncation corresponds to adding approximately 15% noise to the matching point coordinates.

Table 9.3: Comparison of methods on two frames of the Miss America sequence.

Method	Coordinate RMSE
No Motion	4.4888
Two-Step	1.7178
Improved	0.5731
E-matrix	1.2993
A-matrix	0.7582

Since we have no means of knowing the actual 3-D motion parameters between the two frames or the actual depth of the selected feature points, we have calculated the RMS difference between the coordinates of the matching points determined by hierarchical block matching and those predicted by estimated 3-D motion and the structure parameters in order to evaluate the performance of the four methods.

Inspection of the RMSE values in Table 9.3 indicates that the improved iterative algorithm based on the orthographic projection performed best on the head-and-shoulder-type sequence. The reader is reminded that this sequence can be well approximated by the orthographic projection. The A-matrix method has been found to be the second best. We note that in the case of the A-matrix method, the image-plane points were compensated by the "a-parameters" without actually decomposing the A-matrix to determine the rotation and translation parameters. In the case of the E-matrix method, the E-matrix needs to be decomposed to find the motion parameters in order to be able compensate the image-plane points. We have observed that in the presence of errors in the coordinates of the matching points, the decomposition step tends to increase the coordinate RMSE.

Figure 9.4: a) The first and b) the third frames of Miss America with matching points marked by white circles. (Courtesy Yucel Altunbasak)

9.6 Exercises

1. Is it possible to estimate $\Delta\theta$, $\Delta\psi$, and X_3 uniquely from (9.9)? Discuss all ambiguities that arise from the orthographic projection.

2. Observe from (9.7) that no depth information can be estimated when there is no translation, i.e., T = 0. What is the minimum number of point correspondences in this case to determine the rotation matrix R uniquely?

3. Discuss the sensitivity of the proposed methods to noise in the point correspondences. Is the sensitivity related to the size of the object within the field of view? Explain.

Bibliography

[Agg 88] J. K. Aggarwal and N. Nandhakumar, "On the computation of motion from sequences of images," *Proc. IEEE,* vol. 76, pp. 917–935, Aug. 1988.

[Aiz 89] K. Aizawa, H. Harashima, and T. Saito, "Model-based analysis-synthesis image coding (MBASIC) system for a person's face," Signal *Processing:* Image Communication, no. 1, pp. 139-152, Oct. 1989.

[Boz 94] G. Bozdagi, A. M. Tekalp, and L. Onural, "An improvement to MBASIC algorithm for 3-D motion and depth estimation," *IEEE Trans.* on Image *Processing* (special issue), vol. 3, pp. 711-716, June 1994.

[Dem 92] D. DeMenthon and L. S. Davis, "Exact and approximate solutions of the perspective-three-point problem," *IEEE* Trans. *Patt. Anal. Mach. Intel.,* vol. 14, pp. 1100-1104, Nov. 1992.

[Hua 86] T. S. Huang, "Determining three-dimensional motion and structure from two perspective views," Chp. *14* in *Handbook of Putt. Recog.* and Image *Proc.*, Academic Press, 1986.

[Hua 89a] T. S. Huang and C. H. Lee, "Motion and structure from orthographic projections," *IEEE Trans.* Putt. *Anal. Mach. Intel.,* vol. 11, pp. 536-540, May 1989.

[Hua 89b] T. S. Huang and 0. D. Faugeras, "Some properties of the E-matrix in two-view motion estimation," *IEEE Trans. Patt. Anal. Mach. Intel.,* vol. 11, No. 12, pp. 1310-1312, Dec. 1989.

[Hua 90] T. S. Huang and A. N. Netravali, "3D motion estimation," in *Machine Vision for Three-Dimensional Scenes,* Academic Press, 1990.

[Hua 94] T. S. Huang and A. N. Netravali, "Motion and structure from feature correspondences: A review," *Proc. IEEE*, vol. 82, pp. 2522269, Feb. 1994.

[Lon 81] H. C. Longuet-Higgins, "A computer program for reconstructing a scene from two projections," Nature, vol. 392, pp. 133-135, 1981.

[Mit 86] A. Mitiche and J. K. Aggarwal, "A computational analysis of time-varying images," in Handbook of Pattern *Recognition* and *Image* Processing, T. Y. Young and K. S. Fu, eds., New York, NY: Academic Press, 1986.

[Phi 91] J. Philip, "Estimation of three-dimensional motion of rigid objects from noisy observations," *IEEE Trans. Putt. Anal.* Mach. *Intel.*, vol. 13, no. 1, pp. 61-66, Jan. 1991.

[Roa 80] J. W. Roach and J. K. Aggarwal, "Determining the movement of objects from a sequence of images," *IEEE Trans.* Putt. *Anal. Mach. Intel.*, vol. 6, pp. 554–562, 1980.

[Tsa 81] R. Y. Tsai and T. S. Huang, "Estimating three-dimensional motion parameters of a rigid planar patch," *IEEE Trans. Acoust. Speech Sign. Proc.*, vol. 29, no. 6, pp. 1147-1152, Dec. 1981.

[Tsa 82] R. Y. Tsai, T. S. Huang, and W. L. Zhu, "Estimating 3-D motion parameters of a rigid planar patch II: Singular value decomposition," *IEEE Trans. Acoust. Speech Sign. Proc.*, vol. 30, no. 4, pp. 525-534, 1982; and vol. ASSP-31, no. 2, p. 514, 1983.

[Ull 79] S. Ullman, *The Interpretation of* Visual *Motion*, Cambridge, MA: MIT Press, 1979.

[Wen 89] J. Weng, T. S. Huang, and N. Ahuja, "Motion and structure from two perspective views: Algorithms, error analysis, and error estimation," *IEEE Trans. Putt. Anal.* Mach. *Intel.*, vol. 11, no. 5, pp. 451-476, May 1989.

[Wen 91] J. Weng, N. Ahuja, and T. S. Huang, "Motion and structure from point correspondences with error estimation: Planar surfaces," *IEEE* Trans. Sign. *Proc.*, vol. 39, no. 12, pp. 2691-2717, Dec. 1991.

[Wen 92] J. Weng, N. Ahuja, and T. S. Huang, "Motion and structure from line correspondences: Closed-form solution, uniqueness, and optimization," *IEEE* Trans. *Putt.* Anal. Mach. *Intel.*, vol. 14, no. 3, pp. 3188336, Mar. 1992.

[Wen 93] J. Weng, N. Ahuja, and T. S. Huang, "Optimal motion and structure estimation," *IEEE* Trans. *Putt. Anal.* Mach. *Intel.*, vol. 15, no. 9, pp. 864–884, Sep. 1993.

[Wen 93a] J. Weng, T. S. Huang, and N. Ahuja, Motion *and Structure from* Image Sequences, Springer-Verlag (Series in Information Sciences), 1993.

[Zhu 89] X. Zhuang, "A simplification to linear two-view motion algorithms," *Comp. Vis.* Graph. *Image Proc.*, vol. 46, pp. 1755178, 1989.

Chapter 10

OPTICAL FLOW AND DIRECT METHODS

This chapter discusses 3-D motion and structure estimation from two orthographic or perspective views based on an estimate of the optical flow field (optical flow methods) or on spatio-temporal image intensity gradients using the optical flow constraint (direct methods). The main differences between the methods in this chapter and the previous one are: optical flow methods utilize a projected velocity model as opposed to a projected displacement model (see Section 10.1.3 for a comparison of the two models); and optical flow methods require a dense flow field estimate or estimation of the image intensity gradient everywhere, rather than selecting a set of distinct feature points and matching them in two views. Here, we assume that the optical flow field is generated by a single object subject to 3-D rigid motion. The case of multiple moving objects, hence motion segmentation, will be dealt with in the next chapter.

Section 10.1 introduces 2-D velocity field models under orthographic and perspective projections. Estimation of the 3-D structure of the scene from the optical flow field in the case of pure 3-D translational motion is discussed in Section 10.2. Motion and structure estimation from the optical flow field in the case of general 3-D rigid motion using algebraic methods and optimization methods are the subjects of Section 10.3 and 10.4, respectively. Finally, in Section 10.5, we cover direct methods which do not require estimation of the optical flow field or any feature correspondence, but utilize spatio-temporal image intensity gradients directly.

10.1 Modeling the Projected Velocity Field

In this section, we present models for the 2-D (projected) velocity field starting with the 3-D velocity expression. Recall from Chapter 2 that, for the case of small

angular rotation, the 3-D velocity vector $(\dot{X}_1, \dot{X}_2, \dot{X}_3)$ of a rigid object is given by

$$\begin{bmatrix} \dot{X}_1 \\ \dot{X}_2 \\ \dot{X}_3 \end{bmatrix} = \begin{bmatrix} 0 & -\Omega_3 & \Omega_2 \\ \Omega_3 & 0 & -\Omega_1 \\ -\Omega_2 & \Omega_1 & 0 \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \\ X_3 \end{bmatrix} + \begin{bmatrix} V_1 \\ V_2 \\ V_3 \end{bmatrix}$$

which can be expressed in vector notation as a cross product

$$\dot{\mathbf{X}} = \boldsymbol{\Omega} \times \mathbf{X} + \mathbf{V} \tag{10.1}$$

where $\boldsymbol{\Omega} = [\Omega_1 \ \Omega_2 \ \Omega_3]^T$ is the angular velocity vector and $\mathrm{V} = [V_1 \ V_2 \ V_3]^T$ represents the translational velocity vector. In scalar form, we have

$$\begin{aligned} \dot{X}_1 &= \Omega_2 X_3 - \Omega_3 X_2 + V_1 \\ \dot{X}_2 &= \Omega_3 X_1 - \Omega_1 X_3 + V_2 \\ \dot{X}_3 &= \Omega_1 X_2 - \Omega_2 X_1 + V_3 \end{aligned} \tag{10.2}$$

We will present two models for the 2-D velocity field, based on the orthographic and perspective projections of the model (10.2), respectively, in the following.

10.1.1 Orthographic Velocity Field Model

The orthographic projection of the 3-D velocity field onto the image plane can be computed from

$$\begin{aligned} v_1 &= \dot{x}_1 = \dot{X}_1 \\ v_2 &= \dot{x}_2 = \dot{X}_2 \end{aligned}$$

which results in

$$\begin{aligned} v_1 &= V_1 + \Omega_2 X_3 - \Omega_3 x_2 \\ v_2 &= V_2 + \Omega_3 x_1 - \Omega_1 X_3 \end{aligned} \tag{10.3}$$

in terms of the image plane coordinates. The orthographic model can be thought of as an approximation of the perspective model as the distance of the object from the image plane gets larger and the field of view becomes narrower.

10.1.2 Perspective Velocity Field Model

In order to obtain the perspective projection of the 3-D velocity field, we first apply the chain rule of differentiation to the perspective projection expression as

$$\begin{aligned} v_1 &= \dot{x}_1 = f\frac{X_3\dot{X}_1 - \dot{X}_1 X_3}{X_3^2} = f\frac{\dot{X}_1}{X_3} - x_1\frac{\dot{X}_3}{X_3} \\ v_2 &= \dot{x}_2 = f\frac{X_3\dot{X}_2 - \dot{X}_2 X_3}{X_3^2} = f\frac{\dot{X}_2}{X_3} - x_2\frac{\dot{X}_3}{X_3} \end{aligned} \tag{10.4}$$

Now, substituting the 3-D velocity model (10.2) in (10.4), and rewriting the resulting expressions in terms of the image plane coordinates, we have

$$\begin{aligned} v_1 &= f\left(\frac{V_1}{X_3} + \Omega_2\right) - \frac{V_3}{X_3}x_1 - \Omega_3 x_2 - \frac{\Omega_1}{f}x_1 x_2 + \frac{\Omega_2}{f}x_1^2 \\ v_2 &= f\left(\frac{V_2}{X_3} - \Omega_1\right) + \Omega_3 x_1 - \frac{V_3}{X_3}x_2 + \frac{\Omega_2}{f}x_1 x_2 - \frac{\Omega_1}{f}x_2^2 \end{aligned} \tag{10.5}$$

When the projection is normalized, $f = 1$, the model (10.5) can be rearranged as

$$\begin{aligned} v_1 &= \frac{-V_1 + x_1 V_3}{X_3} + \Omega_1 x_1 x_2 - \Omega_2(1 + x_1^2) + \Omega_3 x_2 \\ v_2 &= \frac{-V_2 + x_2 V_3}{X_3} + \Omega_1(1 + x_2^2) - \Omega_2 x_1 x_2 - \Omega_3 x_1 \end{aligned} \tag{10.6}$$

Note that the perspective velocities in the image plane depend on the X_3 coordinate of the object. The dependency of the model on X_3 can be relaxed by assuming a parametric surface model, at least on a local basis (see Sections 10.3.1 and 10.3.2).

For arbitrary surfaces, the depth X_3 of each scene point can be eliminated from the model (10.6) by solving both expressions for X_3 and equating them. This results in a single nonlinear expression, which relates each measured flow vector to the 3-D motion and structure parameters, given by

$$\begin{aligned} -v_2 e_1 + v_1 e_2 - x_1(\Omega_1 + \Omega_3 e_1) - x_2(\Omega_2 + \Omega_3 e_2) - x_1 x_2(\Omega_2 e_1 + \Omega_1 e_2) \\ +(x_1^2 + x_2^2)\Omega_3 + (1 + x_2^2)\Omega_1 e_1 + (1 + x_1^2)\Omega_2 e_2 = v_1 x_2 - v_2 x_1 \end{aligned} \tag{10.7}$$

where $e_1 = V_1/V_3$ and $e_2 = V_2/V_3$ denote the focus of expansion (see Section 10.2). Recall from the discussion of perspective projection that we can find the translational velocities and the depth up to a scaling constant, which is set equal to V_3 above.

10.1.3 Perspective Velocity vs. Displacement Models

Because optical flow estimation from two views indeed corresponds to approximate displacement estimation using spatio-temporal image intensity gradients (see Section 7.1), it is of interest to investigate the relation between the perspective velocity (10.5) and displacement (9.7) models for finite At. To this effect, let

$$v_1 = \dot{x}_1 \approx \frac{x_1' - x_1}{\Delta t} \tag{10.8}$$

Substituting (9.7) into (10.8), we can write

$$\frac{x_1' - x_1}{\Delta t} = \frac{x_1 + \Delta x_1}{\Delta t(1 + \Delta X_3)} - \frac{x_1}{\Delta t}$$

where

$$\Delta x_1 = -\Delta\phi x_2 + \Delta\psi + \frac{T_1}{X_3} \quad (10.9)$$

$$\text{A X 3} = -\Delta\psi x_1 + \Delta\theta x_2 + \frac{T_3}{X_3} \quad (10.10)$$

Now, invoking the approximation $1/(1+\text{x}) \approx 1 - x$ for $x << 1$, we have

$$\frac{x_1 + \Delta x_1}{1 + \Delta X_3} \approx (x_1 + \Delta x_1)(1 - \Delta X_3)$$
$$\approx x_1 + \Delta x_1 - x_1 \Delta X_3 \quad (10.11)$$

where we assume $\Delta X_3 << 1$, and $\Delta x_1 \Delta X_3$ is negligible. Substituting (10.11) into (10.8) we obtain

$$v_1 = \frac{\Delta x_1}{\Delta t} - x_1 \frac{\Delta X_3}{\Delta t} \quad (10.12)$$

which yields (10.6) after replacing Δx_1 and ΔX_3 with (10.9) and (10.10), respectively. This leads us to two conclusions: the relations (10.5) and (9.7) are equivalent in the limit as At goes to zero; and second, for At finite, the two relations are equivalent only when the approximation (10.11) is valid.

10.2 Focus of Expansion

Estimation of the structure of a 3-D scene from a set of images is a fundamental problem in computer vision. There exist various means for structure estimation, such as structure from stereo, structure from motion, and structure from shading. In the previous chapter, we were able to estimate the depth (from motion) at only selected feature points. Recovery of the structure of the moving surface then requires 3-D surface interpolation. In this section, we estimate the structure of a moving surface by analyzing a dense optical flow field in the special case of 3-D translational motion. Structure from optical flow in the more general case, with rotational motion, is treated in the subsequent sections.

For the case of pure 3-D translation, the optical flow vectors (in the image plane) all appear to either emanate from a single point, called the focus of expansion (FOE), or converge to a single point, called the focus of contraction. Here, we consider only the case of expansion. The FOE can be defined as the intersection of the 3-D vector representing the instantaneous direction of translation and the image plane. More specifically, if an object is in pure translation, the 3-D coordinates of a point on the object at time t are

$$\begin{bmatrix} X_1(t) \\ X_2(t) \\ X_3(t) \end{bmatrix} = \begin{bmatrix} X_1(0) + V_1 t \\ X_2(0) + V_2 t \\ X_3(0) + V_3 t \end{bmatrix}$$

where $[X_1(0), X_2(0), X_3(0)]^T$ denote the coordinates of the point at time $t = 0$. Under perspective projection, the image of this point is given by

$$\begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} = \begin{bmatrix} \frac{X_1(0)+V_1 t}{X_3(0)+V_3 t} \\ \frac{X_2(0)+V_2 t}{X_3(0)+V_3 t} \end{bmatrix}$$

Then it appears that all points on the object emanate from a fixed point e in the image plane, called the FOE, given by

$$\mathbf{e} \doteq lim_{t \to -\infty} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} = \begin{bmatrix} V_1/V_3 \\ V_2/V_3 \end{bmatrix} \quad (10.13)$$

Observe that the vector $[V_1/V_3 \; V_2/V_3 \; 1]^T$ indicates the instantaneous direction of the translation. Several approaches exist to calculate the FOE from two or more frames, either by first estimating the optical flow vectors [Law 83] or by means of direct search [Law 83, Jai 83].

Given two frames at times t and t', we can determine the relative depths of image points in the 3-D scene as follows:

1. Estimate the optical flow vectors from time t to t', and the location of FOE at t.

2. The depth $X_{3,i}(t')$ of a single image point $\mathbf{x}_i = [x_{1,i}, x_{2,i}]^T$ at time t' can be determined in terms of ΔX_3, called time-until-contact by Lee [Lee 76], as

$$\frac{X_{3,i}(t')}{\Delta X_3} = \frac{d_i}{\Delta d_i} \quad (10.14)$$

where d_i is the distance of the image point from the FOE at time t, and Δd_i is the displacement of this image point from t to t'. Note that (10.14) is derived by using the similar triangles in the definition of the perspective transform, and AX3 corresponds to the displacement of the 3-D point from t to t' along the X_3 axis, which cannot be determined.

3. The relative depths of two image points $\mathbf{x}_i$ and $\mathbf{x}_j$ at time t' is then given by the ratio

$$\frac{X_{3,i}(t')}{X_{3,j}(t')} = \frac{d_i}{d_j} \frac{\Delta d_i}{\Delta d_j} \quad (10.15)$$

canceling ΔX_3.

10.3 Algebraic Methods Using Optical Flow

Optical flow methods consist of two steps: estimation of the optical flow field, and recovery of the 3-D motion parameters and the scene structure by analyzing the

estimated optical flow. Methods to estimate the optical flow were discussed in Chapter 5. Various approaches exist to perform the second step which differ in the assumptions they make about the structure of the optical flow field and the criteria they employ. We can broadly classify these techniques in two groups: algebraic methods, which seek for a closed-form solution; and optimization methods, which can be characterized as iterative refinement methods towards optimization of a criterion function.

This section is devoted to algebraic methods. After a brief discussion about the uniqueness of the solution, we first assume a planar surface model to eliminate X_3 from (10.2). In this case, the orthographic and perspective projection of the 3-D velocity field results in an affine flow model and a quadratic flow model, respectively [Ana 93]. A least squares method can then be employed to estimate the affine and quadratic flow parameters. We then present two linear algebraic methods to estimate 3-D motion and structure from arbitrary flow fields,

10.3.1 Uniqueness of the Solution

Because 3-D motion from optical flow requires the solution of a nonlinear equation (10.7), there may, in general, be multiple solutions which yield the same observed optical flow. Observe that (10.7) has five unknowns, $\Omega_1, \Omega_2, \Omega_3, e_1$ and e_2. However, in the presence of noise-free optical flow data, it has been shown, using algebraic geometry and homotopy continuation, that [Hol 93]:

- there are at most 10 solutions with five optical flow vectors,
- optical flow at six or more points almost always determines 3-D motion uniquely,
- if the motion is purely rotational, then it is uniquely determined by two optical flow values, and
- in the case of 3-D planar surfaces, optical flow at four points almost always gives two solutions.

With these uniqueness results in mind, we now present a number of algorithms to determine 3-D motion and structure from optical flow.

10.3.2 Affine Flow

A planar surface undergoing rigid motion yields an affine flow field under the orthographic projection. This can easily be seen by approximating the local surface structure with the equation of a plane

$$x_3 = z_0 + z_1 X_1 + 22x2 \tag{10.16}$$

Substituting (10.16) into the orthographic velocity expressions (10.3), we obtain the six-parameter affine flow model

$$\begin{aligned} v_1 &= a_1 + a_2 x_1 + a_3 x_2 \\ v_2 &= a_4 + a_5 x_1 + a_6 x_2 \end{aligned} \tag{10.17}$$

where

$$a_1 = V_1 + z_0\Omega_2, \quad a_2 = z_1\Omega_2, \quad a_3 = z_2\Omega_2 - \Omega_3$$
$$a_4 = V_2 - z_0\Omega_1, \quad a_5 = \Omega_3 - z_1\Omega_1, \quad a_6 = -z_2\Omega_1$$

Observe that, if we know the optical flow at three or more points, we can set up six or more equations in six unknowns to solve for $(a_1, \ldots, a_6)$. However, because of the orthographic projection, it is not possible to determine all eight motion and structure parameters from $(a_1, \ldots, a_8)$ uniquely. For example, z_0 is not observable under the orthographic projection.

10.3.3 Quadratic Flow

The quadratic flow is the most fundamental flow field model, because it is an exact model for the case of planar surfaces under perspective projection; otherwise, it is locally valid under a first-order Taylor series expansion of the surface [Wax 87]. In the case of a planar surface, we observe from (10.16) that

$$\frac{1}{X_3} = \frac{1}{z_0} - \frac{z_1}{z_0}x_1 - \frac{z_2}{z_0}x_2 \tag{10.18}$$

Substituting the expression (10.18) into the perspective velocity field model (10.5), we obtain the eight-parameter quadratic flow field

$$\begin{aligned} v_1 &= a_1 + a_2 x_1 + a_3 x_2 + a_7 x_1^2 + a_8 x_1 x_2 \\ v_2 &= a_4 + a_5 x_1 + a_6 x_2 + a_7 x_1 x_2 + a_8 x_2^2 \end{aligned} \tag{10.19}$$

where

$$a_1 = f\left(\frac{V_1}{z_0} + \Omega_2\right), \quad a_2 = -\left(f\frac{V_1 z_1}{z_0} + \frac{V_3}{z_0}\right), \quad a_3 = -\left(f\frac{V_1 z_2}{z_0} + \Omega_3\right)$$
$$a_4 = f\left(\frac{V_2}{z_0} - \Omega_1\right), \quad a_5 = -\left(f\frac{V_2 z_1}{z_0} - \Omega_3\right), \quad a_6 = -\left(f\frac{V_2 z_2}{z_0} + \frac{V_3}{z_0}\right)$$
$$a_7 = \left(\frac{V_3 z_1}{z_0} + \frac{\Omega_2}{f}\right), \quad a_8 = \left(\frac{V_3 z_2}{z_0} - \frac{\Omega_1}{f}\right)$$

If we know the optical flow on at least four points that lie on a planar patch, we can set up eight or more equations in eight unknowns to solve for $(a_1, \ldots, a_8)$. Note that z_0, which is the distance between the surface point and the camera, always appears as a scale factor. Other approaches have also been reported to solve for the model parameters [Die 91, Ana 93].

In order to recover the 3-D motion and structure parameters using the second-order flow model, Waxman *et* al. [Wax 87] defined 12 kinematic deformation parameters that are related to the first- and second-order partial derivatives of the optical flow. Although closed-form solutions can be obtained, partials of the flow estimates are generally sensitive to noise.

10.3.4 Arbitrary Flow

When the depth of each object point varies arbitrarily, the relationship between the measured flow vectors and the 3-D motion and structure parameters is given by (10.6) or (10.7). An important observation about the expressions (10.6) is that they are bilinear; that is, they are linear in $(V_1, V_2, V_3, \Omega_1, \Omega_2, \Omega_3)$ for a given value of X_3. We discuss two methods, for the solution of (10.6) and (10.7), respectively, in the following. The first method obtains a set of linear equations in terms of 8 intermediate unknowns, three of which are redundant, whereas the second method proposes a two-step least squares solution for the motion parameters.

A Closed-Form Solution

Following Zhuang et al. [Zhu 88] and Heikkonen [Hei 93], the expression (10.7) can be expressed in vector-matrix form as

$$[-v_2\ v_1\ -x_1\ -x_2 - x_1x_2\ (x_1^2 + x_2^2)\ (1 + x_2^2)\ (1 + x_1^2)]\mathbf{H} = v_1x_2 - v_2x_1 \quad \textbf{(10.20)}$$

where

$$\begin{aligned}\mathbf{H} &\doteq [\, h_1\ h_2\ h_3\ h_4\ h_5\ h_6\ h_7\ h_8\,] \\ &= [e_1\ e_2\ \Omega_1 + \Omega_3 e_1\ \Omega_2 + \Omega_3 e_2\ \Omega_2 e_1 + \Omega_1 e_2\ \Omega_3\ \Omega_1 e_1\ \Omega_2 e_2]^T\end{aligned}$$

Given the optical flow vectors at a minimum of eight image points, we can set up eight linear equations to solve for H. Usually we need more than eight image points to alleviate the effects of errors in the optical flow estimates, in which case the intermediate unknowns h_i can be solved in the least squares sense. The parameters h_i can be estimated uniquely as long as the rank of the coefficient matrix is 8, which is almost always the case provided that not all the selected points are coplanar. It can be easily seen from (10.20) that when all points are coplanar, we have a quadratic flow field, and the columns of the coefficient matrix are dependent on each other.

The five motion parameters, $\Omega_1, \Omega_2, \Omega_3, e_1$, and e_2, can subsequently be recovered from h_i, i = 1, , 8. Next, the depth X_3 of each point can be estimated from the model (10.6). We note here, however, that the estimation of the motion parameters from h_i is not unique in the presence of errors in the optical flow estimates. This is due to the fact that only five of the eight h_i are independent. For example, motion parameters estimated from h_1, h_2, h_6, h_7, and h_8 do not necessarily satisfy ha, h_4, and h_5. Estimation of the motion parameters to satisfy all h_i in the least squares sense can be considered, though it is not trivial.

Heeger and Jepson Method

Alternatively, Heeger and Jepson [Hee 92] develop a two-step method where no redundant variables are introduced, They express the arbitrary flow equations (10.6)

in vector-matrix form as

$$\begin{bmatrix} v_1(x_1, x_2) \\ v_2(x_1, x_2) \end{bmatrix} = p(x_1, x_2)\mathbf{A}(x_1, x_2)\mathbf{V} + \mathbf{B}(x_1, x_2)\boldsymbol{\Omega} \quad (10.21)$$

where

$$\mathbf{A}(x_1, x_2) = \begin{bmatrix} f & 0 & -x_1 \\ 0 & f & -x_2 \end{bmatrix}, \mathbf{B}(x_1, x_2) = \begin{bmatrix} -x_1x_2/f & f + x_1^2/f & -x_2 \\ -(f + x_2^2/f) & x_1x_2/f & x_1 \end{bmatrix}$$

$$\mathbf{V} = [\, V_1\ V_2\ V_3\,]^T, \quad \boldsymbol{\Omega} = [\, \Omega_1\ \Omega_2\ \Omega_3\,]^T, \text{ and } p(x_1, x_2) = \frac{1}{X_3}$$

Consider stacking (10.21) for flow vectors at N different spatial locations, $(x_{11}, x_{12}), \ldots, (x_{N1}, x_{N2})$, to obtain

$$\begin{aligned}\mathbf{v} &= \mathbf{A}(\mathbf{V})\mathbf{p} + \mathbf{B}\boldsymbol{\Omega} \\ &= \mathbf{C}(\mathbf{V})\mathbf{q} \end{aligned} \quad (10.22)$$

where

$$\mathbf{A(V)} = \begin{bmatrix} \mathbf{A}(x_{11}, x_{12})\mathbf{V} & \cdots & 0 \\ \vdots & \vdots & \vdots \\ 0 & \cdots & \mathbf{A}(x_{N1}, x_{N2})\mathbf{V} \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} \mathbf{B}(x_{11}, x_{12}) \\ \vdots \\ \mathbf{B}(x_{N1}, x_{N2}) \end{bmatrix}$$

$$\mathbf{C}(\mathbf{V}) = \begin{bmatrix} | & | \\ \mathbf{A}(\mathbf{V}) & \mathbf{B} \\ | & | \end{bmatrix}, \quad \mathbf{q} = \begin{bmatrix} \mathbf{p} \\ \boldsymbol{\Omega} \end{bmatrix}$$

In order to find the least squares estimates of the motion and structure parameters **V** and **q, we** minimize the error in (10.22) given by

$$E(\mathbf{V}, \mathbf{q}) = ||\mathbf{v} - \mathbf{C}(\mathbf{V})\mathbf{q}||^2 \quad (10.23)$$

Evidently, to minimize (10.23) with respect to **V** and **q** we need to compute the partial derivatives of (10.23) with respect to each variable and set them equal to zero, and solve the resulting equations simultaneously. Each equation would give a surface in a multidimensional space, and the solution lies at the intersection of all of these surfaces.

Alternatively, we can compute the partial of (10.23) with respect to **q only.** Setting it equal to zero yields

$$\hat{\mathbf{q}} = \left[\mathbf{C}(\mathbf{V})^T\mathbf{C}(\mathbf{V})\right]^{-1}\mathbf{C}(\mathbf{V})^T\mathbf{v} \quad (10.24)$$

which is a surface in terms of **V.** Obviously, Equation (10.24) cannot be directly used to find an estimate of **q,** since **V** is also unknown. However, because the actual solution lies on this surface, it can be substituted back into (10.23) to express the criterion function (10.23) in terms of V only. Now that we have a nonlinear

criterion function in terms of **V** only, the value of **V** that minimizes (10.23) can be determined by means of a search procedure. Straightforward implementation of this strategy suffers from a heavy computational burden, since Equation (10.24) needs to be solved for every perturbation of **V** during the search process. Observe that the dimensions of the matrix in (10.24) to be inverted is related to the number of optical flow vectors used.

Fortunately, Heeger and Jepson have proposed an efficient method to search for the minimum of this index as a function of only **V,** without evaluating (10.24) at each step, using some well-known results from linear algebra. Furthermore, since we can estimate **V** only up to a scale factor, they have restricted the search space, without loss of generality, to the unit sphere, which can be described by two angles in the spherical coordinates. Once the best estimate of **V** is determined, the least squares estimate of q can be evaluated from (10.24). Experimental results suggest that this method is quite robust to optical flow estimation errors.

10.4 Optimization Methods Using Optical Flow

Optimization methods are based on perturbing the 3-D motion and structure parameters independently until the projected velocity field is consistent with the observed optical flow field. These techniques are most successful for tracking small changes in the motion and structure parameters from frame to frame in a long sequence of frames, starting with reasonably good initial estimates, usually obtained by other means. Some sort of smoothness constraint on the 3-D motion and structure parameters is usually required to prevent the optimization algorithm from diverging or converging to a local minimum of the cost function.

Morikawa and Harashima [Mor 91] have proposed an optimization method using the orthographic velocity field (v_1, v_2) given by

$$\begin{aligned} v_1 &= \dot{x_1} = V_1 + \Omega_2 X_3 - \Omega_3 x_2 \\ v_2 &= \dot{x_2} = V_2 + \Omega_3 x_1 - \Omega_1 X_3 \end{aligned} \tag{10.25}$$

Note that the motion parameters are global parameters assuming there is a single rigid object in motion. However, the depth parameters vary spatially. Given initial estimates of the 3-D motion and depth parameters, we can update them incrementally using

$$\begin{aligned} \Omega_1(k) &= \Omega_1(k-1) + \Delta\Omega_1 \\ \Omega_2(k) &= \Omega_2(k-1) + \Delta\Omega_2 \\ \Omega_3(k) &= \Omega_3(k-1) + \Delta\Omega_3 \\ V_1(k) &= V_1(k-1) + \Delta V_1 \\ V_2(k) &= V_2(k-1) + \Delta V_2 \\ X_3(x_1, x_2)(k) &= X_3(x_1, x_2)(k-1) + \Delta X_3(x_1, x_2) \end{aligned}$$

where the problem reduces to finding the incremental parameters from frame to frame. The smoothness of motion constraint means that the update terms (the incremental parameters) should be small. Consequently, we introduce a measure of the smoothness in terms of the functional

$$||P||^2 = \frac{\alpha}{N}||\mathbf{\Delta\Omega}||^2 + \frac{\beta}{N}||\mathbf{\Delta V}||^2 + \gamma\sum_{i=1}^{N}(\Delta X_{i3})^2$$

where α, β, and γ are scale parameters, $||.||$ is the L_2 norm, and N is the number of points considered.

Then the overall cost functional measures the sum of how well the projected velocity field conforms with the observed flow field, and how smooth the variations in the 3-D motion and structure parameters are from frame to frame, as follows

$$E(\mathbf{\Delta\Omega}, \mathbf{AT}, \Delta X_3) = \sum\left[(v_{i1} - \hat{u}_{i1})^2 + (v_{i2} - \hat{u}_{i2})^2\right] + ||P||^2 \tag{10.26}$$

where

$$\begin{aligned} \hat{u}_{i1} &= (V_1 + \Delta V_1) + (\Omega_2 + \Delta\Omega_2)(X_{i3} + (\Delta X)_{i3}) - (\Omega_3 + \Delta\Omega_3)x_2 \\ \hat{u}_{i2} &= (V_2 + \Delta V_2) + (\Omega_3 + \Delta\Omega_3)x_1 - (\Omega_1 + \Delta\Omega_1)(X_{i3} + \mathbf{(AX),,)} \end{aligned}$$

The optimization can be performed by gradient-based methods or simulated annealing procedures. We note that because this procedure uses the orthographic velocity field, the depth parameters can be found up to an additive constant and a scale parameter, while two of the rotation angles can be determined up to the reciprocal of this scaling constant. Other successful methods also exist for estimating motion and structure under the orthographic model [Kan 86].

10.5 Direct Methods

Direct methods utilize only the spatio-temporal image intensity gradients to estimate the 3-D motion and structure parameters. In this section, we present two examples of direct methods, one as an extension of optical-flow-based methods, and another starting from the projected motion field model and the optical flow equation. For other examples of direct methods, the reader is referred to the literature [Hor 88].

10.5.1 Extension of Optical Flow-Based Methods

Almost all optical-flow-based estimation methods can be extended as direct methods by replacing the optical flow vectors with their estimates given in terms of spatio-temporal image intensity gradients as derived in Chapter 5. Recall that an estimate

of the flow field was given by (5.12)

$$\begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \begin{bmatrix} \sum \frac{\partial s_c(\mathbf{X})}{\partial x_1}\frac{\partial s_c(\mathbf{X})}{\partial x_1} & \sum \frac{\partial s_c(\mathbf{X})}{\partial x_1}\frac{\partial s_c(\mathbf{X})}{\partial x_2} \\ \sum \frac{\partial s_c(\mathbf{X})}{\partial x_1}\frac{\partial s_c(\mathbf{X})}{\partial x_2} & \sum \frac{\partial s_c(\mathbf{X})}{\partial x_2}\frac{\partial s_c(\mathbf{X})}{\partial x_2} \end{bmatrix}^{-1} \begin{bmatrix} -\sum \frac{\partial s_c(\mathbf{X})}{\partial x_1}\frac{\partial s_c(\mathbf{X})}{\partial t} \\ -\sum \frac{\partial s_c(\mathbf{X})}{\partial x_2}\frac{\partial s_c(\mathbf{X})}{\partial t} \end{bmatrix} \quad (10.27)$$

Substituting this expression in any optical-flow-based method for the optical flow vectors, we can obtain an estimate of the 3-D motion and structure parameters directly in terms of the spatio-temporal image intensity gradients. However, the reader is alerted that (5.12) may not provide the best possible estimates for the optical flow field. An alternative approach is presented in the following starting with the projected motion and optical flow equations.

10.5.2 Tsai-Huang Method

Tsai and Huang [Tsa 81] considered the case of planar surfaces, where we can model the displacement of pixels from frame t to t' by (9.40)

$$x_1' = T_1(a_1, \ldots, a_8) = \frac{a_1 x_1 + a_2 x_2 + a_3}{a_7 x_1 + a_8 x_2 + 1}$$

$$x_2' = T_2(a_1, \ldots, a_8) = \frac{a_4 x_1 + a_5 x_2 + a_6}{a_7 x_1 + a_8 x_2 + 1} \quad (10.28)$$

where the pure parameters $a_1, , a_8$ are usually represented in vector notation by a. Note that a = e, where e = $(1, 0, 0, 0, 1, 0, 0, 0)^T$ corresponds to no motion, i.e., $x_1 = T_1(\mathbf{e})$, $x_2 = T_2(\mathbf{e})$.

In order to obtain a linear estimation algorithm, the mapping (10.28) will be linearized by means of a first-order Taylor series expansion about a = e, assuming small motion, as

$$T_1(\mathbf{a}) - T_1(\mathbf{e}) = \Delta x_1 = \sum_{i=1}^{8} \frac{\partial T_1(\mathbf{a})}{\partial a_i}\Big|_{\mathbf{a}=\mathbf{e}} (a_i - e_i)$$

$$T_2(\mathbf{a}) - T_2(\mathbf{e}) = \Delta x_2 = \sum_{i=1}^{8} \frac{\partial T_2(\mathbf{a})}{\partial a_i}\Big|_{\mathbf{a}=\mathbf{e}} (a_i - e_i) \quad (10.29)$$

where a = $(a_1, . , a_8)$, $\Delta x_1 = x_1' - x_1$ and $\Delta x_2 = x_2' - x_2$.

Now, referring to the optical flow equation (5.5) and approximating the partials by finite differences, we obtain the discrete expression

$$\frac{s_k(x_1, x_2) - s_{k+1}(x_1, x_2)}{\mathrm{At}} = \frac{\partial s_{k+1}(x_1, x_2)}{\partial x_1}\frac{\Delta x_1}{\mathrm{a\ t}} + \frac{\partial s_{k+1}(x_1, x_2)}{\partial x_2}\frac{\Delta x_2}{\Delta t} \quad (10.30)$$

where At is the time interval between the frames k and $k + 1$ at times t and t', respectively. After cancelling At, we obtain

$$s_k(x_1, x_2) = s_{k+1}(x_1, x_2) + \frac{\partial s_{k+1}(x_1, x_2)}{\partial x_1}\Delta x_1 + \frac{\partial s_{k+1}(x_1, x_2)}{\partial x_2}\mathrm{Ax}, \quad (10.31)$$

Observe that (10.31) may also be interpreted as the linearization of image intensity function in the vicinity of (x_1, x_2) in frame $k + 1$, as

$$s_{k+1}(x_1', x_2') - s_{k+1}(x_1, x_2) = \frac{\partial s_{k+1}(x_1, x_2)}{\partial x_1}\Delta x_1 + \frac{\partial s_{k+1}(x_1, x_2)}{\partial x_2}\Delta x_2$$

since we have

$$s_{k+1}(x_1', x_2') = s_k(x_1, x_2)$$

Substituting the linearized expressions for Δx_1 and Δx_2 from (10.29) into (10.31)) we can write the frame difference between the frames k and $k + 1$ as

$$\begin{aligned} FD(x_1, x_2) &= s_k(x_1, x_2) - s_{k+1}(x_1, x_2) \\ &= \frac{\partial s_{k+1}(x_1, x_2)}{\partial x_1}\left[\sum_{i=1}^{8} \frac{\partial T_1(\mathbf{a})}{\partial a_i}(a_i - e_i)\right] \\ &\quad + \frac{\partial s_{k+1}(x_1, x_2)}{\partial x_2}\left[\sum_{i=1}^{8} \frac{\partial T_2(\mathbf{a})}{\partial a_i}(a_i - e_i)\right] \end{aligned} \quad (10.32)$$

In order to express (10.32) in vector matrix form, we define

$$\mathbf{H} = [H_1\ H_2\ H3\ \boldsymbol{H4}\ H_5\ H_6\ H_7\ H_8]^T \quad (10.33)$$

where

$$H_1 = \frac{T_1(\mathbf{a})}{\partial a_1}\frac{\partial s_{k+1}}{\partial x_1} = x_1\frac{\partial s_{k+1}}{\partial x_1}, \qquad H_2 = \frac{T_1(\mathbf{a})}{\partial a_2}\frac{\partial s_{k+1}}{\partial x_1} = x_2\frac{\partial s_{k+1}}{\partial x_1}$$

$$H_3 = \frac{T_1(\mathbf{a})}{\partial a_3}\frac{\partial s_{k+1}}{\partial x_1} = \frac{\partial s_{k+1}}{\partial x_1}, \qquad H_4 = \frac{T_2(\mathbf{a})}{\partial a_4}\frac{\partial s_{k+1}}{\partial x_2} = \frac{\partial s_{k+1}}{\partial x_2}x_1$$

$$H_5 = \frac{T_2(\mathbf{a})}{\partial a_5}\frac{\partial s_{k+1}}{\partial x_2} = x_2\frac{\partial s_{k+1}}{\partial x_2}, \qquad H_6 = \frac{T_2(\mathbf{a})}{\partial a_6}\frac{\partial s_{k+1}}{\partial x_2} = \frac{\partial s_{k+1}}{\partial x_2}$$

$$H_7 = \frac{T_1(\mathbf{a})}{\partial a_7}\frac{\partial s_{k+1}}{\partial x_1} + \frac{T_2(\mathbf{a})}{\partial a_7}\frac{\partial s_{k+1}}{\partial x_2} = -x_1^2\frac{\partial s_{k+1}}{\partial x_1} - x_1x_2\frac{\partial s_{k+1}}{\partial x_2}$$

$$H_8 = \frac{T_1(\mathbf{a})}{\partial a_8}\frac{\partial s_{k+1}}{\partial x_1} + \frac{T_2(\mathbf{a})}{\partial a_8}\frac{\partial s_{k+1}}{\partial x_2} = -x_1x_2\frac{\partial s_{k+1}}{\partial x_1} - \frac{\partial s_{k+1}}{\partial x_2}x_2$$

Then

$$FD(x_1, x_2) = \mathbf{H} \cdot (\mathbf{a} - \mathbf{e}) \tag{10.34}$$

Note that this relation holds for every image point that originates from a single planar object.

To summarize, the following algorithm is proposed for the estimation of pure parameters without the need to identify any point correspondences:

1. Select at least eight points (x_1, x_2) in the kth frame that are on the same plane.
2. Compute $FD(x_1, x_2)$ between the frames k and $k + 1$ at these points
3. Estimate the image gray level gradients $\frac{\partial s_{k+1}(x_1,x_2)}{\partial x_1}$ and $\frac{\partial s_{k+1}(x_1,x_2)}{\partial x_2}$ at these eight points in the frame k + 1.
4. Compute the vector **H.**
5. Solve for $\mathbf{\Delta a}$ = **a** — e in the least squares sense.

Once the pure parameters have been estimated, the corresponding 3-D motion and structure parameters can be determined by a singular value decomposition of the matrix **A** as described in Section 9.4. We note here that the estimator proposed by Netravali and Salz [Net 85] yields the same results in the case of a planar object.

10.6 Examples

Optical-flow-based and direct methods do not require establishing feature point correspondences, but they rely on estimated optical flow field and spatio-temporal image intensity gradients, respectively. Recall that the estimation of the optical flow and image intensity gradients are themselves ill-posed problems which are highly sensitive to observation noise.

We compare the performances of four algorithms, three optical flow-based methods and one direct method. They are: i) Zhuang's closed-form solution and ii) the Heeger-Jepson method, which are optical-flow-based methods using a perspective model; iii) the iterative method of Morikawa, which is based on the orthographic flow model; and iv) the direct method of Tsai and Huang, using the perspective projection and a planar surface model. It is also of interest to compare these methods with the feature-based techniques discussed in Chapter 9 to determine which class of techniques is more robust in the presence of errors in the point correspondences, optical flow field, and image intensity values, respectively.

The methods will be assessed on the basis of the accuracy of the resulting 3-D motion and depth estimates, and the goodness of the motion compensation that can be achieved by synthesizing the second frame from the first, given the 3-D motion and depth estimates. We start with some numerical simulations to provide the

Table 10.1: Comparison of the motion parameters. All angles are in radians, and the translations are in pixels. (Courtesy Yucel Altunbasak)

True	Zhuang		Heeger-Jepson		
	No Error	3% Error	No Error	3% Error	10% Error
Ω_1 = 0.007	0.0070	0.0133	0.0064	0.0094	0.0036
Ω_2 = 0.010	0.0099	0.0058	0.0105	0.0066	0.0008
Ω_3 = 0.025	0.0250	0.0232	0.0249	0.0251	0.0249
V_2/V_1 = 1.80	1.7999	2.9571	1.7872	1.9439	2.7814
V_3/V_1 = *0.48*	*0.4799*	10.121	0.4693	0.5822	0.8343
Depth Error	1.73 E-5	0.824	1.41 E-3	0.0181	0.0539
Match Error	2.17 E-7	0.0152	2.71 E-5	5.99 E-4	0.0018

reader with a comparison of the true and estimated 3-D motion and depth estimates obtained by these methods. The accuracy of the resulting motion compensation will be demonstrated on both simulated and real video sequences.

10.6.1 Numerical Simulations

We have simulated an optical flow field using the perspective flow model (10.5) and a set of 3-D motion and depth parameters, which are given in Table 10.1 under the "True" column. The flow field has been simulated for 30 points such that X_1 and X_2 are uniformly distributed in the interval [-25,251 cm, and X_3 is uniformly distributed within [70,100] cm. The focal length parameter has been set to f = 50 mm. Indeed, the simulation parameters are identical to those used in Chapter 9 to facilitate comparison of the results with those of the methods using point correspondences. In order to test the sensitivity of the methods to the errors in optical flow estimation, 3% and 10% errors have been added to the synthesized flow vectors, respectively.

Table 10.1 provides a comparison of the results obtained by the methods of Zhuang and Heeger-Jepson (H-J). In the table, "Depth Error" corresponds to the normalized RMS error in the depth estimates given by

$$\text{Depth Error} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\frac{(X_3^{(i)} - \hat{X}_3^{(i)})^2}{(X_3^{(i)})^2}} \tag{10.35}$$

where N = 30, and the "Match Error" measures the RMS deviation of the flow field $\hat{\mathbf{v}}$ generated by the estimated 3-D motion and depth parameters from the input flow field v, given by

$$\text{Match Error} = \frac{1}{N}\sqrt{\sum_{i=1}^{N}(v_1 - \hat{v}_1)^2 + (v_2 - \hat{v}_2)^2}$$

It can easily be seen that the method of Zhuang is sensitive to noise. This is mainly because it introduces three redundant intermediate unknowns, which makes the linear formulation (10.20) possible. Observe that only five of the eight intermediate unknowns are sufficient to solve for the motion parameters, and different combinations of the five intermediate unknowns may yield different motion estimates in the presence of noise. Although the intermediate unknowns are solved in the least squares sense, the final motion estimates are not really least squares estimates. On the other hand, the H-J method always finds the motion and depth parameters that minimize the least squares error criterion, and hence is more robust to errors in the optical flow estimates. We remark that the accuracy of the estimates in the H-J method depends on the step size (angular increments on the unit sphere) used in the search procedure. Two-degree increments were used to obtain the results given in Table 10.1.

Because the method of Morikawa is based on the orthographic flow model, its performance depends on how much the given flow vectors deviate from the orthographic model (10.3). Recall from Chapter 9 that this deviation is related to the ratio of the width of the object to the average distance of the points from the center of projection, given by $R = \Delta X_3 / 2\bar{X}_3$. For the data set used in the above simulation, we have $R \approx 18\%$, and Morikawa's method did not give successful results. Table 10.2 provides a comparison of the performance of Morikawa's method on a perfectly orthographic flow field simulated from (10.3) and a perspective flow field with $R \approx 1\%$, where the true parameter values are identical to those of the first simulation set in Chapter 9. Another issue of practical importance in Morikawa's method is how to initialize the unknown parameters. In the above simulations, all parameters have been initialized at $\pm 5\%$ of their true values. Note that, in general, the cost function has multiple minima, and the results very much depend on the initial point. It is thus recommended that Morikawa's method be used for frame-to-frame tracking of small motion parameters, where the initial estimates for the first frame are obtained through another method, such as the improved two-step algorithm in Chapter 9.

Table 10.2: Comparison of the motion parameters. All angles are in radians, and the translations are in pixels. (Courtesy Gozde Bozdagi)

True	Orthographic	Perspective, $R \approx 1\%$
$\Omega_1 = 0.007$	0.00713	0.00677
$\Omega_2 = 0.010$	0.01013	0.00977
$\Omega_3 = 0.025$	0.02513	0.02477
$V_1 = 0.100$	0.09987	0.10023
$V_2 = 0.180$	0.17987	0.18023
Depth Error	0.00011	0.00023
Match Error	0.00322	0.00489

To demonstrate the performance of the Tsai-Huang method, we have mapped the first frame of "Miss America" onto the plane given by $n_1X_1 + n_2X_2 + n_3X_3 = 1$. The resulting planar object is subjected to 3-D rigid motion, and an image of the new object is computed by means of the prespective projection with $f = 50$ mm. In this simulation, we have a planar object, no local motion, and no uncovered background. The only sources of error are gradient estimation and linearized model assumptions. Two sets of parameters for the 3-D motion and the planar surface used in the simulation are tabulated in Table 10.3. For the first simulation, the plane is approximately 70 cm away from the image plane. The size of the object is assumed to be 50 cm x 50 cm, and 1 pel = 0.1395 mm. A requirement for the success of the direct methods is undoubtedly the accuracy of the spatio-temporal gradient estimates. To ensure reasonably accurate gradient estimates, pixels where the Laplacian of the image intensity is above a threshold have been selected for 3-D motion estimation. This is because in flat regions, gradient values are in the same order of magnitude as the noise. The gradient estimation is performed by 9th order spatio-temporal polynomial fitting.

Table 10.3 shows the results of 3-D motion and structure estimation using the Tsai-Huang method for two different motions. Clearly, the results are better in the

Table 10.3: Comparison of the motion parameters. All angles are in radians, and the translations are in pixels. (Courtesy Yucel Altunbasak)

Motion 1: True							
Ω_1	Ω_2	Ω_3	V_2/V_1	V_3/V_1	n_1	n_2	n_3
0.0070	0.0100	0.0250	1.8000	0.4800	0	0	0.01429
a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8
0.9964	-0.0249	6.0981	0.0249	0.9964	2.5109	-2.7 E-5	1.9 E-5
Estimate							
Ω_1	Ω_2	Ω_3	V_2/V_1	V_3/V_1	n_1	n_2	n_3
0.0355	0.1034	-0.0086	-0.3775	-0.1599	-15.871	17.8351	115.99
a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8
0.9983	-0.0065	1.4815	-0.0099	0.9952	0.7108	-0.0003	0.0001
Motion 2: True							
Ω_1	Ω_2	Ω_3	V_2/V_1	V_3/V_1	n_1	n_2	n_3
0.004	0.005	0.006	-1.25	0.625	0	0	0.01
a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8
0.9975	-0.0059	0.0049	0.0059	0.9975	0.0049	-0.0009	0.0008
Estimate							
Ω_1	Ω_2	Ω_3	V_2/V_1	V_3/V_1	n_1	n_2	n_3
0.0019	0.0053	0.0055	-0.6552	0.6842	-0.3325	0.0542	-1.4870
a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8
0.9972	-0.0054	0.0049	0.0062	0.9980	0.0049	-0.0009	0.0004

case of Motion 2. To see how well we can compensate for the motion using the estimated u-parameters, we synthesized the second frame from the first for both motions, The resulting mean square displaced frame differences (MS-DFD) were compared with plain frame differences (MS-FD) (i.,e no motion compensation). In the first case, the MS-DFD was 16.02 compared to MS-FD equal to 19.84. However, in the second case, the MS-DFD has dropped to 0.24 in comparison to MS-FD equal to 4.63. Inspection of the results indicates that the Tsai-Huang method performs better when both the frame difference and the u-parameter values are relatively small. In fact, this is expected since the method relies on the linearization of both the frame difference (the discrete optical flow equation) and the pixel-to-pixel mapping in terms of the u-parameters.

10.6.2 Experiments with Two Frames of Miss America

These methods have also been tested on the same two frames of "Miss America" that were used in Chapter 9, shown in Figure 9.4 (a) and (b). The optical flow field between these two frames has been estimated by 3-level hierarchical block matching. All algorithms have been applied to those pixels, where the Laplacian of the image intensity is above a predetermined threshold (T=50) to work with reliable optical

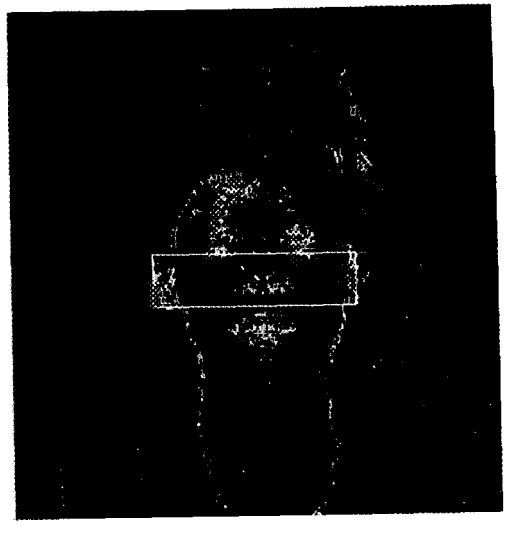


Figure 10.1: The map of pixels selected for 3-D motion estimation marked on the first frame of Miss America. (Courtesy Yucel Altunbasak)

flow estimates, In order to exclude points which may exhibit nonrigid motion, only pixels within the white box depicted in Figure 10.1 have been used. Although we expect to have reliable optical flow and/or spatio-temporal gradient estimates at these pixels, some of these pixels may still fall into regions of local motion or uncovered background, violating the assumption of rigid motion. However) because we work with at least 500 pixels, it is expected that the effect of such pixels will be negligible. The results are compared on the basis of how well we can synthesize the estimated optical flow field at those pixels shown in Figure 10.1 through the estimated 3-D motion and structure parameters and in some cases how well we can synthesize the next frame from the first and estimated motion parameters,

We have initialized Morikawa's algorithm with the 3-D motion estimates obtained from the improved algorithm discussed in Chapter 9. The initial depth estimates at all selected points have been computed using the optical flow and the initial motion parameter values. After 100 iterations, the criterion function has dropped from 1.94 to 1.04. The value of the criterion function with zero motion estimation was computed as 6.79. Clearly, Morikawa's method is suitable to estimate small changes in the motion and depth parameters, provided a good set of initial estimates are available.

Next, the method of Heeger and Jepson (H-J) was used with the same two frames where the search for the best **V** was performed with angular increments of 1 degree on the unit sphere. The RMSE difference between the optical flow synthesized using the parameters estimated by the H-J method and the one estimated by hierarchical block matching was found to be 0.51, which ranks among the best results. Finally, we have applied the direct method of Tsai-Huang on the same set of pixels in the two frames and compare the results with those of the A-matrix method where each optical flow vector is used to determine a point correspondence. Table 10.4 tabulates the respective velocity and intensity synthesis errors. The results indicate that the A-matrix method is more successful in compensating for the motion.

Table 10.4: Comparison of sythesis errors. (Courtesy Yucel Altunbasak)

Method	No Compensation	Tsai-Huang	A-matrix
Velocity Sythesis Error	6.96	5.52	2.32
Intensity Sythnesis Error	20.59	17.63	8.10

10.7 Exercises

1. How do you compare 3-D motion estimation from point correspondences versus that from optical flow? List the assumptions made in each case. Suppose we have 3-D motion with constant acceleration and precession; which method would you employ?
2. Which one can be estimated more accurately, token matching or spatio-temporal image intensity gradients?
3. Suppose we assume that the change in X_3 (ΔX_3) in the time interval between the two views can be neglected. Can you propose a linear algorithm, similar to the one in Section 9.3, to estimate the 3-D motion and structure parameters from optical flow?
4. Show that (10.31) may also be interpreted as the linearization of the image intensity function in the vicinity of (x_1, x_2) in frame $k + 1$.

Bibliography

[Adi 85] G. Adiv, "Determining three-dimensional motion and structure from optical flow generated by several moving objects," *IEEE Trans. Putt.* Anal. *Mach. Intel.,* vol. 7, pp. 384-401, July 1985.

[Agg 88] J. K. Aggarwal and N. Nandhakumar, "On the computation of motion from sequences of images," *Proc. IEEE,* vol. 76, pp. 917935, Aug. 1988.

[Ana 93] P. Anandan, J. R. Bergen, K. J. Hanna, and R. Hingorani, "Hierarchical model-based motion estimation," in Motion Analysis and *Image* Sequence *Processing,* M. I. Sezan and R. L. Lagendijk, eds., Kluwer, 1993.

[Cam 92] M. Campani and A. Verri, "Motion analysis from first-order properties of optical *flow," CVGIP:* Image Understanding, vol. 56, no. 1, pp. 90–107, Jul. 1992.

[Die 91] N. Diehl, "Object-oriented motion estimation and segmentation in image sequences," Signal *Processing:* Image Comm., vol. 3, pp. 23-56, 1991.

[Hee 92] D. J. Heeger and A. Jepson, "Subspace methods for recovering rigid motion I: Algorithm and implementation," *Int. J.* Comp. *Vis.,* vol. 7, pp. 95–117, 1992.

[Hei 93] J. Heikkonen, "Recovering 3-D motion from optical flow field," Image *Processing:* Theory and Applications, Vernazza, Venetsanopoulos, and Braccini, eds., Elsevier, 1993.

[Hol 93] R. J. Holt and A. N. Netravali, "Motion from optic flow: Multiplicity of solutions," *J. Vis. Comm. Image Rep.,* vol. 4, no. 1, pp. 14–24, 1993.

[Hor 88] B. K. P. Horn and E. J. Weldon Jr., "Direct methods for recovering motion," *Int. J. Comp. Vision,* vol. 2, pp. 51-76, 1988.

[Jai 83] R. Jain, "Direct computation of the focus of expansion," *IEEE Trans. Patt. Anal.* Mach. *Intel.,* vol. 5, pp. 58863, Jan. 1983.

[Jer 91] C. P. Jerian and R. Jain, "Structure from motion - A critical analysis of methods," *IEEE System,* Man and *Cyber.,* vol. 21, pp. 572-588, 1991.

[Kan 86] K. Kanatani, "Structure and motion from optical flow under orthographic projection," *Comp. Vision Graph.* Image *Proc.,* vol. 35, pp. 181-199, 1986.

[Law 83] D. T. Lawton, "Processing translational motion sequences," *Comp. Vis. Grap. Image Proc.,* vol. 22, pp. 1166144, 1983.

[Lee 76] D. N. Lee, "A theory of visual control of braking based on information about time to collision," *Perception,* vol. 5, pp. 4377459, 1976.

[Mor 91] H. Morikawa and H. Harashima, "3D structure extraction coding of image sequences," *J. Visual* Comm. *and Image Rep.,* vol. 2, no. 4, pp. 3322344, Dec. 1991.

[Net 85] A. N. Netravali and J. Salz, "Algorithms for estimation of three-dimensionalmotion," *AT&T Tech. J.,* vol. 64, no. 2, pp. 335-346, Feb. 1985.

[Tsa 81] R. Y. Tsai and T. S. Huang, "Estimating three-dimensional motion parameters of a rigid planar patch," *IEEE* Trans. *Acoust. Speech Sign. Proc.,* vol. 29, no. 6, pp. 1147-1152, Dec. 1981.

[Ver 89] A. Verri and T. Poggio, "Motion field and optical flow: Qualitative properties," *IEEE* Trans. Patt. *Anal.* Mach. *Intel.,* vol. PAMI-11, no. 5, pp. 490-498, May 1989.

[Wax 87] A. M. Waxman, B. Kamgar-Parsi, and M. Subbarao, "Closed-form solutions to image flow equations for 3-D structure and motion," *Int. J. Comp. Vision* vol. 1, pp. 2399258, 1987.

[Zhu 88] X. Zhuang, T. S. Huang, N. Ahuja, and R. M. Haralick, "A simplified linear optic-flow motion algorithm," *Comp. Vis. Graph.* and Image *Proc.,* vol. 42, pp. 334–344, 1988.

Chapter 11

MOTION SEGMENTATION

Most real image sequences contain multiple moving objects or multiple motions. For example, the Calendar and Train sequence depicted in Chapter 5 exhibits five different motions, shown in Figure 11.1. Optical flow fields derived from multiple motions usually display discontinuities (motion edges). Motion segmentation refers to labeling pixels that are associated with each independently moving 3-D object in a sequence featuring multiple motions. A closely related problem is optical flow segmentation, which refers to grouping together those optical flow vectors that are associated with the same 3-D motion and/or structure. These two problems are identical when we have a dense optical flow field with an optical flow vector for every pixel.

It should not come as a surprise that motion-based segmentation is an integral part of many image sequence analysis problems, including: i) improved optical flow estimation, ii) 3-D motion and structure estimation in the presence of multiple moving objects, and iii) higher-level description of the temporal variations and/or the content of video imagery. In the first case, the segmentation labels help to identify optical flow boundaries and occlusion regions where the smoothness constraint should be turned off. The approach presented in Section 8.3.3 constitutes an example of this strategy. Segmentation is required in the second case, because a distinct parameter set is needed to model the flow vectors associated with each independently moving 3-D object. Recall that in Chapters 9 and 10, we have assumed that all feature points or flow vectors belong to a single rigid object. Finally, in the third case, segmentation information may be considered as a high-level (object-level) description of the frame-to-frame motion information as opposed to the low-level (pixel-level) motion information provided by the individual flow vectors.

As with any segmentation problem, proper feature selection facilitates effective motion segmentation. In general, application of standard image segmentation

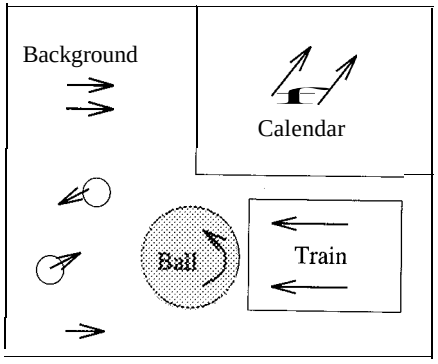


Figure 11.1: Example of optical flow generated by multiple moving objects.

methods (see Appendix B for a brief overview) directly to optical flow data may not yield meaningful results, since an object moving in 3-D usually generates a spatially varying optical flow field. For example, in the case of a single rotating object, there is no flow at the center of the rotation, and the magnitude of the flow vectors grows as we move away from the center of rotation. Therefore, in this chapter, a parametric model-based approach has been adopted for motion-based video segmentation where the model parameters constitute the features. Examples of parametric mappings that can be used with direct methods include the &parameter mapping (10.28), and affine mapping (9.4). Recall, from Chapter 9, that the mapping parameters depend on: i) the 3-D motion parameters, the rotation matrix R and the translation vector T, and ii) the model of the object surface, such as the orientation of the plane in the case of a piecewise planar model. Since each independently moving object and/or different surface structure will best fit a different parametric mapping, parameters of a suitably selected mapping will be used as features to distinguish between different 3-D motions and surface structures.

Direct methods, which utilize spatio-temporal image gradients, are presented in Section 11.1. These techniques may be considered as extensions of the direct methods discussed in Chapter 10 to the case of multiple motion. In Section 11.2, a two-step procedure is followed, where first the optical flow field is estimated using one of the techniques covered in Chapters 5 through 8. A suitable parametric motion model has subsequently been used for optical flow segmentation using clustering or maximum *a posteriori* (MAP) estimation. The accuracy of segmentation results clearly depends on the accuracy of the estimated optical flow field. As mentioned earlier, optical flow estimates are usually not reliable around moving object boundaries due to occlusion and use of smoothness constraints. Thus, optical flow estimation and segmentation are mutually interrelated, and should be addressed simultaneously for best results. We present methods for simultaneous optical flow estimation and segmentation using parametric flow models in Section 11.3.

11.1 Direct Methods

In this section, we consider direct methods for segmentation of images into independently moving regions based on spatio-temporal image intensity and gradient information. This is in contrast to first estimating the optical flow field between two frames and then segmenting the image based on the estimated optical flow field. We start with a simple thresholding method that segments images into "changed" and "unchanged regions." Methods using parametric displacement field models will be discussed next.

11.1.1 Thresholding for Change Detection

Thresholding is often used to segment a video frame into "changed" versus "unchanged" regions with respect to the previous frame. The unchanged regions denote the stationary background, while the changed regions denote the moving and occlusion areas.

We define the frame difference $FD_{k,k-1}(x_1, x_2)$ between the frames k and $k-1$ as

$$FD_{k,k-1}(x_1, x_2) = s(x_1, x_2, k) - s(x_1, x_2, k-1) \tag{11.1}$$

which is the pixel-by-pixel difference between the two frames. Assuming that the illumination remains more or less constant from frame to frame, the pixel locations where $FD_{k,k-1}(x_1, x_2)$ differ from zero indicate "changed" regions. However, the frame difference hardly ever becomes exactly zero, because of the presence of observation noise.

In order to distinguish the nonzero differences that are due to noise from those that are due to a scene change, segmentation can be achieved by thresholding the difference image as

$$z_{k,k-1}(x_1, x_2) = \begin{cases} 1 & \text{if } |FD_{k,k-1}(x_1, x_2)| > T \\ 0 & \text{otherwise} \end{cases} \tag{11.2}$$

where T is an appropriate threshold. The value of the threshold T can be chosen according to one of the threshold determination algorithms described in Appendix B. Here, $z_{k,k-1}(x_1, x_2)$ is called a segmentation label field, which is equal to "1" for changed regions and "0" otherwise. In practice, thresholding may still yield isolated 1s in the segmentation mask $z_{k,k-1}(x_1, x_2)$, which can be eliminated by postprocessing; for example, forming 4- or &connected regions, and discarding any region(s) with less than a predetermined number of entries.

Example: **Change** *Detection*

The changed region for the same two frames of the Miss America that were used in Chapters 9 and 10, computed as described above, is depicted in Figure 11.2. The threshold was set at 7, and no post-filtering was performed.

Figure 11.2: Changed region between the first and third frames of the Miss America sequence. (Courtesy Gozde Bozdagi)

Another approach to eliminating isolated l's is to consider accumulative differences which add memory to the motion detection process. Let $s(x_1, x_2, k)$, $s(x_1, x_2, k-1)$, ., $s(x_1, x_2, k-N)$ be a sequence of N frames, and let $s(x_1, x_2, k)$ be the reference frame. An accumulative difference image is formed by comparing this reference image with every subsequent image in the sequence. A counter for each pixel location in the accumulative image is incremented every time the difference between the reference image and the next image in the sequence at that pixel location is bigger than the threshold. Thus, pixels with higher counter values are more likely to correspond to actual moving regions.

11.1.2 An Algorithm Using Mapping Parameters

The method presented here, based on the works of Hotter and Thoma [Hoe 88] and Diehl [Die 91], can be considered as a hierarchically structured top-down approach. It starts by fitting a parametric model, in the least squares sense, to the entire changed region from one frame to the next, and then breaks this region into successively smaller regions depending on how well a single model fits each region or subregion. This is in contrast to the clustering and MAP approaches, to be discussed in the next section, which start with many small subregions and group them

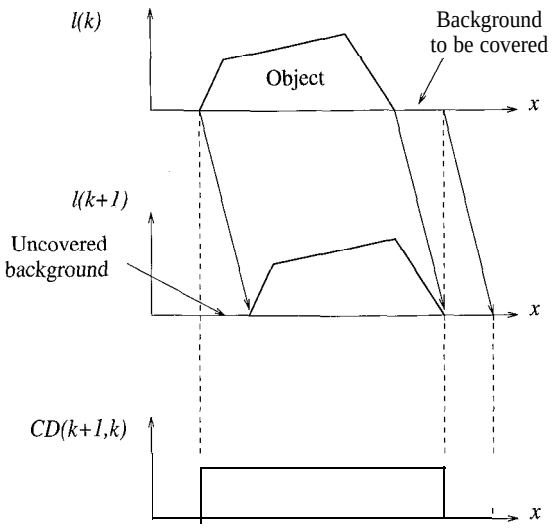


Figure 11.3: Detection of uncovered background [Hoe 88].

together according to some merging criterion to form segments. The hierarchically structured approach can be summarized through the following steps:

1) In the first step, a change detector, described above, initializes the segmentation mask separating the changed and unchanged regions from frame k to $k+1$. Median filtering or morphological filtering can be employed to eliminate small regions in the change detection mask. Each spatially connected changed region is interpreted as a different object.

2) For each object a different parametric model is estimated. The methods proposed by Hotter and Thoma [Hoe 88] and Diehl [Die 91] differ in the parametric models employed and in the estimation of the model parameters. Estimation of the parameters for each region is discussed in the next subsection.

3) The changed region(s) found in step 1 is (are) divided into moving region(s) and the uncovered background using the mapping parameters computed in step 2. This is accomplished as follows: All pixels in frame $k+1$ that are in the changed region are traced backwards, with the inverse of the motion vector computed from the mapping parameters found in step 2. If the inverse of the motion vector points to a pixel in frame k that is within the changed region, then the pixel in frame $k+1$ is classified as a moving pixel; otherwise, it is assigned to the uncovered background. Figure 11.3 illustrates this process, where *CD* refers to the change detection mask between the lines $l(k)$ and $l(k+1)$.

Next, the validity of the model parameters for those pixels within the moving region is verified by evaluating the displaced frame difference. The regions where

the respective parameter vector is not valid are marked,& independent objects for the second hierarchical level. The procedure iterates between steps 2 and 3 until the parameter vectors for each region are consistent with the region.

11.1.3 Estimation of Model Parameters

Let the kth frame of the observed sequence be expressed as

$$g_k(\mathbf{x}) = s_k(\mathbf{x}) + n_k(\mathbf{x}) \tag{11.3}$$

where $n_k(\mathbf{x})$ denotes the observation noise. Assuming no occlusion effects, we have $s_{k+1}(\mathbf{x}) = s_k(\mathbf{x}')$, where

$$\mathbf{x}' = h(\mathbf{x}, \theta) \tag{11.4}$$

is a tranformation of pixels from frame k to $k+1$, with the parameter vector θ. Then

$$g_{k+1}(\mathbf{x}) = s_k(\mathbf{x}') + n_{k+1}(\mathbf{x}) \tag{11.5}$$

The transformation $h(\mathbf{x}, \theta)$ must be unique and invertible. Some examples for the transformation are as follows:

i) Assuming a planar surface and using the perspective projection, we obtain the eight-parameter mapping

$$\begin{aligned} x_1' &= \frac{a_1 x_1 + a_2 x_2 + a_3}{a_7 x_1 + a_8 x_2 + 1} \\ x_2 &= \frac{a_4 x_1 + a_5 x_2 + a_6}{a_7 x_1 + a_8 x_2 + 1} \end{aligned} \tag{11.6}$$

where $\theta = (a_1, a_2, a_3, a_4, \text{as}, a_6, a_7, a_8)^T$ is the vector of so-called pure parameters.

ii) Alternatively, assuming a planar surface and using the orthographic projection, we have the affine transform

$$\begin{aligned} x_1' &= c_1 x_1 + c_2 x_2 + c_3 \\ x_2' &= c_4 x_1 + c_5 x_2 + c_6 \end{aligned} \tag{11.7}$$

where $\theta = (c_1, c_2, c_3, c_4, c_5, c_6)^T$ is the vector of mapping parameters.

iii) Finally, let's assume a quadratic surface, given by

$$X_3 = a_{11}X_1^2 + a_{12}X_1X_2 + a_{22}X_2^2 + a_{13}X_1 + a_{23}X_2 + a_{33} \tag{11.8}$$

Substituting (11.8) into the 3-D motion model, using the orthographic projection equations, and grouping the terms with the same exponent, we arrive at the quadratic transform

$$\begin{aligned} x_1' &= a_1 x_1^2 + a_2 x_2^2 + a_3 x_1 x_2 + a_4 x_1 + a_5 x_2 + a_6 \\ x_2' &= b_1 x_1^2 + b_2 x_2^2 + b_3 x_1 x_2 + b_4 x_1 + b_5 x_2 + b_6 \end{aligned} \tag{11.9}$$

which has 12 parameters.

Note that we may not always be able to determine the actual 3-D motion and surface structure parameters from the mapping parameters. However, for image coding applications this does not pose a serious problem, since we are mainly interested in predicting the next frame from the current frame. Furthermore, the approach presented here is not capable of handling occlusion effects.

The method of Hotter and Thoma uses the eight-parameter model (11.6). They estimated the model parameters for each region using a direct method which is similar to the method of Tsai and Huang [Tsa 81] that was presented in Chapter 10. On the other hand, Diehl [Die 91] suggests using the quadratic transform, because it provides a good approximation to many real-life images. He proposed estimating the mapping parameters to minimize the error function

$$J(\hat{\theta}) = \frac{1}{2} E\left\{ \left(g_{k+1}(\mathbf{x}) - g_k(\mathbf{x}', \hat{\theta}) \right)^2 \right\}$$

where $E\{.\}$ is the expectation operator, and $g_k(\mathbf{x}', \hat{\theta})$ denotes the prediction of frame $k+1$ from frame k using the mapping $h(\mathbf{x}, \theta)$. A gradient-based minimization algorithm, the modified Newton's method, is used to find the best parameter vector $\hat{\theta}$. The contents of the images $g_k(\mathbf{x})$ and $g_{k+1}(\mathbf{x})$ must be sufficiently similar in order for the error function to have a unique minimum.

11.2 Optical Flow Segmentation

In this section, we treat segmentation of a given flow field using parameters of a flow field model as features. We assume that there are K independently moving objects, and each flow vector corresponds to the projection of a 3-D rigid motion of a single opaque object. Then each distinct motion can be accurately described by a set of mapping parameters. Most common examples of parametric models, such as eight-parameter mapping (10.19) and affine mapping (10.17), implicitly assume a 3-D planar surface in motion. Approximating the surface of a real object by a union of a small number of planar patches, the optical flow generated by a real object can be modeled by a piecewise quadratic flow field, where the parameters $a_1, \ldots, a_8$ in (10.19) vary in a piecewise fashion. It follows that flow vectors corresponding to the same surface and 3-D motion would have the same set of mapping parameters, and optical flow segmentation can be achieved by assigning the flow vectors with the same mapping parameters into the same class.

The underlying principle of parametric, model-based segmentation methods can be summarized as follows: Suppose we have K sets of parameter vectors, where each set defines a correspondence or a flow vector at each pixel. Flow vectors defined by the mapping parameters are called model-based or synthesized flow vectors. Thus, we have II synthesized flow vectors at each pixel. The segmentation procedure then assigns the label of the synthesized vector which is closest to the estimated flow vector at each site. However, there is a small problem with this simple scheme: both the number of classes, K, and the mapping parameters for each class are not known

a priori. Assuming a particular value for K, the mapping parameters for each class could be computed in the least squares sense provided that the estimated optical flow vectors associated with the respective classes are known. That is, we need to know the mapping parameters to find the segmentation labels and the segmentation labels are needed to find the mapping parameters. This suggests an iterative procedure, similar to the K-means clustering algorithm where both the segmentation labels and the class means are unknown (see Appendix B). The modified Hough transform approach of Adiv [Adi 85], the modified K-means approach of Wang and Adelson [Wan 94], and the MAP method of Murray and Buxton [Mur 87], which are described in the following, all follow variations of this strategy.

11.2.1 Modified Hough Transform Method

The Hough transform is a well-known clustering technique where the data samples "vote" for the most representative feature values in a quantized feature space. In a straightforward application of the Hough transform method to optical flow segmentation using the six-parameter affine flow model (10.17), the six-dimensional feature space $a_1, \ldots, a_6$ would be quantized to certain parameter states after the minimal and maximal values for each parameter are determined. Then, each flow vector $\mathbf{v}(\mathbf{x}) = [v_1(\mathbf{x}) \; v_2(\mathbf{x})]^T$ votes for a set of quantized parameters which minimizes

$$\eta^2(\mathbf{x}) \doteq \eta_1^2(\mathbf{x}) + \eta_2^2(\mathbf{x}) \tag{11.10}$$

where $\eta_1(\mathbf{x}) = v_1(\mathbf{x}) - a_1 - a_2 x_1 - a_3 x_2$ and $\eta_2(\mathbf{x}) = v_2(\mathbf{x}) - a_4 - a_5 x_1 - a_6 x_2$. The parameter sets that receive at least a predetermined amount of votes are likely to represent candidate motions. The number of classes K and the corresponding parameter sets to be used in labeling individual flow vectors are hence determined. The drawback of this scheme is the significant amount of computation involved.

In order to keep the computational burden at a reasonable level, Adiv proposed a two-stage algorithm that involves a modified Hough transform procedure. In the first stage of his algorithm, connected sets of flow vectors are grouped together to form components which are consistent with a single parameter set. Several simplifications were proposed to ease the computational load, including: i) decomposition of the parameters space into two disjoint subsets $\{a_1, a_2, \text{us}\} \times \{a_4, a_5, \text{us}\}$ to perform two 3-D Hough transforms, ii) a multiresolution Hough transform, where at each resolution level the parameter space is quantized around the estimates obtained at the previous level, and iii) a multipass Hough technique, where the flow vectors which are most consistent with the candidate parameters are grouped first. In the second stage, those components formed in the first stage which are consistent with the same quadratic flow model (10.19) in the least squares sense are merged together to form segments. Several merging criteria have been proposed. In the third and final stage, ungrouped flow vectors are assimilated into one of their neighboring segments.

In summary, the modified Hough transform approach is based on first clustering the flow vectors into small groups, each of which is consistent with the flow generated

by a moving planar facet. The fusion of these small groups into segments is then performed based on some ad-hoc merging criteria.

11.2.2 Segmentation for Layered Video Representation

A similar optical flow segmentation method using the K-means clustering technique has been employed in the so-called layered video representation **strategy.** Instead of trying to capture the motion of multiple overlapping objects in a single motion field, Wang and Adelson [Wan 94] proposed a layered video representation in which the image sequence is decomposed into layers by means of an optical-flow-based image segmentation, and ordered in depth along with associated maps defining their motions, opacities, and intensities, In the layered representation, the segmentation labels denote the layer in which a particular pixel resides.

The segmentation method is based on the affine motion model (10.17) and clustering in a six-dimensional parameter space. The image is initially divided into small blocks. Given an optical flow Geld (e.g., computed by using (5.12)) a set of affine parameters are estimated for each block. To determine the reliability of the parameter estimates, the sum of squared distances between the synthesized and estimated flow vectors is computed as

$$\bar{\eta}^2 = \sum_{\mathbf{x}\in\mathcal{B}} ||\mathbf{v}(\mathbf{x}) - \tilde{\mathbf{v}}(\mathbf{x})||^2 \tag{11.11}$$

where $\mathcal{B}$ refers to a block of pixels. Obviously, if the Row within the block complies with an affine model, the residual will be small. On the other hand, if the block falls on the boundary between two distinct motions, the residual will be large. The motion parameters for blocks with acceptably small residuals are selected as the candidate layer models. To determine the appropriate number Ii of layers, the motion parameters of the candidate layers are clustered in the six-dimensional parameter space. The initial set of affine model parameters are set equal to the mean of the K clusters. Then the segmentation label of each pixel site is selected as the index of the parameter set that yields the closest optical flow vector at that site. After all sites are labeled, the affine parameters of each layer are recalculated based on the new segmentation labels. This procedure is repeated until the segmentation labels no longer change or a fixed number of iterations is reached.

It should be noted that if the smallest difference between an observed vector and its parameter-based estimate exceeds a threshold, then the site is not labeled in the above iterative procedure, and the observed flow vector is ignored in the parameter estimation that follows. After the iterative procedure converges to reliable affine models, all sites without labels are assigned one according to the motion compensation criterion, which assigns the label of the parameter vector that gives the best motion compensation at that site. This feature ensures more robust parameter estimation by eliminating the outlier vectors. A possible limitation of this segmentation method is that it lacks constraints to enforce spatial and temporal continuity of the

segmentation labels. Thus, rather ad-hoc steps are needed to eliminate small, isolated regions in the segmentation label field. The Bayesian segmentation strategy promises to impose continuity constraints in an optimization framework.

11.2.3 Bayesian Segmentation

The Bayesian method searches for the maximum of the a *posteriori* probability of the segmentation labels given the optical flow data, which is a measure of how well the current segmentation explains the observed optical flow data and how well it conforms to our prior expectations. Murray and Buxton [Mur 87] first proposed a MAP segmentation method where the optical flow data was modeled by a piecewise quadratic flow field, and the segmentation field modeled by a Gibbs distribution. The search for the labels that maximize the a *posteriori* probability was performed by simulated annealing. Here we briefly present their approach.

Problem Formulation

Let vi, $\mathbf{v}_2$, and z denote the lexicographic ordering of the components of the flow vector v(x) $= [v_1(\mathbf{x})\ v_2(\mathbf{x})]^T$ and the segmentation labels $z(\mathbf{x})$ at each pixel. The a *posteriori* probability density function (pdf) $p(\mathrm{z}|\mathrm{vi}, \mathbf{v}_2)$ of the segmentation label field z given the optical flow data vi and $\mathbf{v}_2$ can be expressed, using the Bayes theorem, as

$$p(\mathbf{z}|\mathbf{v}_1, \mathbf{v}_2) = \frac{p(\mathbf{v}_1, \mathbf{v}_2|\mathbf{z})p(\mathbf{z})}{p(\mathbf{v}_1, \mathbf{v}_2)} \tag{11.12}$$

where $p(\mathbf{v}_1, \mathbf{v}_2|\mathbf{z})$ is the conditional pdf of the optical flow data given the segmentation z, and p(z) is the *a priori* pdf of the segmentation. Observe that, i) z is a discrete-valued random vector with a finite sample space Ω, and ii) $p(\mathbf{v}_1, \mathbf{v}_2)$ is constant with respect to the segmentation labels, and hence can be ignored for the purposes of segmentation. The MAP estimate, then, maximizes the numerator of (11.12) over all possible realizations of the segmentation field $\mathrm{z} = \omega, \omega \in \Omega$.

The conditional probability $p(\mathbf{v}_1, \mathbf{v}_2|\mathbf{z})$ is a measure of how well the piecewise quadratic flow model (10.19), where the model parameters $a_1, .$, us depend on the segmentation label z, fits the estimated optical flow field $\mathbf{v}_1$ and $\mathbf{v}_2$. Assuming that the mismatch between the observed flow v(x) and the synthesized flow,

$$\begin{aligned} \tilde{v}_1(\mathbf{x}) &= a_1x_1 + a_2x_2 - a_3 + a_7x_1^2 + a_8x_1x_2 \\ \tilde{v}_2(\mathbf{x}) &= a_4x_1 + a_5x_2 - \mathbf{a6} + a_7x_1x_2 + a_8x_2^2 \end{aligned} \tag{11.13}$$

is modeled by white, Gaussian noise with zero mean and variance σ^2, the conditional pdf of the optical flow field given the segmentation labels can be expressed as

$$p(\mathbf{v}_1, \mathbf{v}_2|\mathbf{z}) = \frac{1}{(2\pi\sigma^2)^{M/2}} \exp\left\{-\sum_{i=1}^{M} \eta^2(\mathbf{x}_i)/2\sigma^2\right\} \tag{11.14}$$

where M is the number of flow vectors available at the sites $\mathbf{x}_i$, and

$$\eta^2(\mathbf{x}_i) = (v_1(\mathbf{x}_i) - \tilde{v}_1(\mathbf{x}_i))^2 + (v_2(\mathbf{x}_i) - \tilde{v}_2(\mathbf{x}_i))^2 \tag{11.15}$$

is the norm-squared deviation of the actual flow vectors from what is predicted by the quadratic flow model. Assuming that the quadratic flow model is more or less accurate, this deviation is due to segmentation errors and the observation noise.

The prior pdf is modeled by a Gibbs distribution which effectively introduces local constraints on the interpretation (segmentation). It is given by

$$p(\mathbf{z}) = \frac{1}{Q} \sum_{\omega \in \Omega} \exp\ \{-\mathrm{U(z)}\}\ \delta(\mathbf{z} - \omega) \tag{11.16}$$

where Ω denotes the discrete sample space of $\mathbf{z}$, Q is the partition function

$$Q = \sum_{\omega \in \Omega} \exp\{-U(\omega)\} \tag{11.17}$$

and $U(\omega)$ is the potential function which can be expressed as a sum of local clique potentials $V_C(z(\mathbf{x}_i), z(\mathbf{x}_j))$. The prior constraints on the structure of the segmentation labels can be specified in terms of local clique potential functions as discussed in Chapter 8. For example, a local smoothness constraint on the segmentation labels can be imposed by choosing $V_C(z(\mathbf{x}_i), z(\mathbf{x}_j))$ as in Equation (8.11). Temporal continuity of the labels can similarly be modeled [Mur 87].

Substituting (11.14) and (11.16) into the criterion (11.12) and taking the logarithm of the resulting expression, maximization of the a *posteriori* probability distribution can be performed by minimizing the cost function

$$E = \frac{1}{2\sigma^2} \sum_{i=1}^{M} \eta^2(\mathbf{x}_i) + U(\omega) \tag{11.18}$$

The first term describes how well the predicted data fit the actual optical flow measurements (in fact, optical flow is estimated from the image sequence at hand), and the second term measures how much the segmentation conforms to our prior expectations.

The Algorithm

Because the model parameters corresponding to each label are not known a priori, the MAP segmentation alternates between estimation of the model parameters and assignment of the segmentation labels to optimize the cost function (11.18) based on a simulated annealing (SA) procedure. Given the flow field v and the number of independent motions K, MAP segmentation via the Metropolis algorithm can be summarized as follows:

1. Start with an initial labeling z of the optical flow vectors. Calculate the mapping parameters $\mathbf{a} = [a_1\ a_8]^T$ for each region using least squares fitting as in Section 10.3.2. Set the initial temperature for SA.

2. Scan the pixel sites according to a predefined convention. At each site $\mathbf{x}_i$:

   (a) Perturb the label $z_i = z(\blacksquare_i)$ randomly.

   (b) Decide whether to accept or reject this perturbation, as described in Section 8.1.1, based on the change **AE** in the cost function (11.18),

   $$\Delta E = \frac{1}{2\sigma^2} \Delta\eta^2(\mathbf{x}_i) + \sum_{\mathbf{x}_j \in \mathcal{N}_{\mathbf{x}_i}} \Delta V_C(z(\mathbf{x}_i),\ z(\mathbf{x}_j)) \tag{11.19}$$

   where $\mathcal{N}_{\mathbf{x}_i}$ denotes a neighborhood of the site $\mathbf{x}_i$ and $V_C(z(\mathbf{x}_i),\ z(\mathbf{x}_j))$ is given by Equation (8.11). The first term indicates whether or not the perturbed label is more consistent with the given flow field determined by the residual (11.15), and the second term reflects whether or not it is in agreement with the prior segmentation field model.

3. After all pixel sites are visited once, re-estimate the mapping parameters for each region in the least squares sense based on the new segmentation label configuration.

4. Exit if a stopping criterion is satisfied. Otherwise, lower the temperature according to the temperature schedule, and go to step 2.

The following observations about the MAP segmentation algorithm are in order:
i) The procedure proposed by Murray-Buxton suggests performing step 3, the model parameter update, after each and every perturbation. Because such a procedure will be computationally more demanding, the parameter updates are performed only after all sites have been visited once.
ii) This algorithm can be applied to any parametric model relating to optical flow, although the original formulation has been developed on the basis of vernier velocities [Mur 87] and the associated eight-parameter model.
iii) The actual 3-D motion and depth parameters are not needed for segmentation purposes. If desired, they can be recovered from the parameter vector **a** for each segmented flow region at convergence.

We conclude this section by noting that all methods discussed so far are limited by the accuracy of the available optical flow estimates. Next, we introduce a novel framework, in which optical flow estimation and segmentation interact in a mutually beneficial manner.

11.3 Simultaneous Estimation and Segmentation

By now, it should be clear that the success of optical flow segmentation is closely related to the accuracy of the estimated optical flow field, and vice versa. It follows

that optical flow estimation and segmentation have to be addressed simultaneously for best results. Here, we present a simultaneous Bayesian approach based on a representation of the motion field as the sum of a parametric field and a residual field. The interdependence of optical flow and segmentation fields are expressed in terms of a Gibbs distribution within the MAP framework. The resulting optimization problem, to find estimates of a dense set of motion vectors, a set of segmentation labels, and a set of mapping parameters, is solved using the highest confidence first (HCF) and iterated conditional mode (ICM) algorithms. It will be seen that several existing motion estimation and segmentation algorithms can be formulated as degenerate cases of the algorithm presented here.

11.3.1 Motion Field Model

Suppose that there are K independently moving, opaque objects in a scene, where the 2-D motion induced by each object can be approximated by a parametric model, such as (11.13) or a 6-parameter affine model. Then, the optical flow field v(x) can be represented as the sum of a parametric flow field $\tilde{\mathbf{v}}(\mathbf{x})$ and a nonparametric residual field $\mathbf{v}_r(\mathbf{x})$, which accounts for local motion and other modeling errors [Hsu 94]; that is,

$$\mathbf{v}(\mathbf{x}) = \tilde{\mathbf{v}}(\mathbf{x}) + \mathrm{VT(X)} \tag{11.20}$$

The parametric component of the motion field clearly depends on the segmentation label $z(\mathbf{x})$, which takes on the values $1, \cdot, K$.

11.3.2 Problem Formulation

The simultaneous MAP framework aims at maximizing the a posteriori pdf

$$p(\mathbf{v}_1, \mathbf{v}_2, \mathbf{z} \mid \mathbf{g}_k\ \mathbf{g}_{k+1}) = \frac{p(\mathbf{g}_{k+1} \mid \mathbf{g}_k, \mathbf{v}_1, \mathbf{v}_2, \mathbf{z}) p(\mathbf{v}_1, \mathbf{v}_2 \mid \mathbf{z}, \mathbf{g}_k) p(\mathbf{z} \mid \mathbf{g}_k)}{p(\mathbf{g}_{k+1} \mid \mathbf{g}_k)} \tag{11.21}$$

with respect to the optical flow vi, $\mathbf{v}_2$ and the segmentation labels z. Through careful modeling of these pdfs, we can express an interrelated set of constraints that help to improve the estimates.

The first conditional pdf $p(\mathbf{g}_{k+1} \mid \mathbf{g}_k$, vi, $\mathbf{v}_2$, z) provides a measure of how well the present displacement and segmentation estimates conform with the observed frame $k+1$ given frame le. It is modeled by a Gibbs distribution as

$$p(\mathbf{g}_{k+1}\ \mathrm{I}\ \mathbf{g}_k, \mathbf{v}_1, \mathbf{v}_2, \mathbf{z}) = \frac{1}{Q_1} \exp\{-U_1(\mathbf{g}_{k+1}\ \mathrm{I}\ \mathbf{g}_k, \mathbf{v}_1, \mathbf{v}_2, \mathbf{z})\} \tag{11.22}$$

where Q_1 is the partition function (a constant), and

$$U_1(\mathbf{g}_{k+1} \mid \mathbf{g}_k, \mathbf{v}_1, \mathbf{v}_2, \mathbf{z}) = \sum_{\mathbf{x}} [g_k(\mathbf{x}) - g_{k+1}(\mathbf{x} + \mathbf{v}(\mathbf{x})\Delta t)]^2 \tag{11.23}$$

is called the Gibbs potential. Here, the Gibbs potential corresponds to the norm-square of the displaced frame difference (DFD) between the frames $\mathbf{g}_k$ and $\mathbf{g}_{k+1}$. Thus, maximizationof (11.22) imposes the constraint that v(x) minimizes the DFD.

The second term in the numerator in (11.21) is the conditional pdf of the displacement field given the motion segmentation and the search image. It is modeled by a Gibbs distribution

$$p(\mathbf{v}_1, \mathbf{v}_2 \mid\ \mathbf{z}, \mathbf{g}_k) = p(\mathbf{v}_1, \mathbf{v}_{2\ \mathrm{I}}\ \mathbf{z}) = \frac{1}{Q_2} \exp\{-U_2(\mathbf{v}_1, \mathbf{v}_2 \mid \mathbf{z})\} \tag{11.24}$$

where Q_2 is a constant, and

$$U_2(\mathbf{v}_1, \mathbf{v}_2 \mid \mathbf{z}) = \alpha \sum_{\mathbf{x}} ||\mathbf{v}(\mathbf{x}) - \tilde{\mathbf{v}}(\mathbf{x})||^2 + \beta \sum_{\mathbf{x}_i} \sum_{\mathbf{x}_j \in \mathcal{N}_{\mathbf{x}}} ||\mathbf{v}(\mathbf{x}_i) - \mathbf{v}(\mathbf{x}_j)||^2\ \delta(z(\mathbf{x}_i) - z(\mathbf{x}_j)) \tag{11.25}$$

is the corresponding Gibbs potential, $||\ ||$ denotes the Euclidian distance, and $\mathcal{N}_{\mathbf{x}}$ is the set of neighbors of site x. The first term in (11.25) enforces a minimum norm estimate of the residual motion field $\mathbf{v}_r(\mathbf{x})$; that is, it aims to minimize the deviation of the motion field v(x) from the parametric motion field $\tilde{\mathbf{v}}(\mathbf{x})$ while minimizing the DFD. Note that the parametric motion field $\tilde{\mathbf{v}}(\mathbf{x})$ is calculated from the set of model parameters $\mathbf{a}_i$, i = 1, . . , K, which in turn is a function of v(x) and $z(\mathbf{x})$. The second term in (11.25) imposes a piecewise local smoothness constraint on the optical flow estimates without introducing any extra variables such as line fields. Observe that this term is active only for those pixels in the neighborhood $\mathcal{N}_{\mathbf{x}}$ which share the same segmentation label with the site x. Thus, spatial smoothness is enforced only on the flow vectors generated by a single object. The parameters α and β allow for relative scaling of the two terms.

The third term in (11.21) models the a priori probability of the segmentation field given by

$$p(\mathbf{z} \mid \mathbf{g}_k) = p(\mathbf{z}) = \frac{1}{Q_3} \sum_{\omega \in \Omega} \exp\{-U_3(\mathbf{z})\} \delta(\mathbf{z} - \omega) \tag{11.26}$$

where Ω denotes the sample space of the discrete-valued random vector z, Q_3 is given by Equation (11.17),

$$U_3(\mathbf{z}) = \sum_{\mathbf{x}_i} \sum_{\mathbf{x}_j \in \mathcal{N}_{\mathbf{x}_i}} V_C(z(\mathbf{x}_i), z(\mathbf{x}_j)) \tag{11.27}$$

$\mathcal{N}_{\mathbf{x}_i}$ denotes the neighborhood system for the label field, and

$$V_C(z(\mathbf{x}_i), z(\mathbf{x}_j)) = \begin{cases} -\gamma & \text{if } z(\mathbf{x}_i) = z(\mathbf{x}_j) \\ +\gamma & \text{otherwise} \end{cases} \tag{11.28}$$

The dependence of the labels on the image intensity is usually neglected, although region boundaries generally coincide with intensity edges.

11.3.3 The Algorithm

Maximizing the a posteriori pdf (11.21)is equivalent to minimizing the cost function,

$$E = U_1(\mathbf{g}_{k+1} \mid \mathbf{g}_k, \mathbf{v}_1, \mathbf{v}_2, \mathbf{z}) + U_2(\mathbf{v}_1, \mathbf{v}_2 \mid \mathbf{z}) + U_3(\mathbf{z}) \tag{11.29}$$

that is composed of the potential functions in Equations (11.22), (11.24), and (11.26). Direct minimization of (11.29) with respect to all unknowns is an exceedingly difficult problem, because the motion and segmentation fields constitute a large set of unknowns. To this effect, we perform the minimization of (11.29) through the following two-steps iterations [Cha 94]:

1. Given the best available estimates of the parameters $\mathbf{a}_i$, $i = 1, . , K$, and z, update the optical flow field $\mathbf{v}_1$, $\mathbf{v}_2$. This step involves the minimization of a modified cost function

$$E_1 = \sum_{\mathbf{x}} [g_k(\mathbf{x}) - g_{k+1}(\mathbf{x} + \mathbf{v}(\mathbf{x})\Delta t)]^2 + \alpha \sum_{\mathbf{x}} ||\mathbf{v}(\mathbf{x}) - \tilde{\mathbf{v}}(\mathbf{x})||^2 + \beta \sum_{\mathbf{x}_i} \sum_{\mathbf{x}_j \in \mathcal{N}_{\mathbf{x}}} ||\mathbf{v}(\mathbf{x}_i) - \mathbf{v}(\mathbf{x}_j)||^2 \, \delta(z(\mathbf{x}_i) - z(\mathbf{x}_j)) \tag{11.30}$$

which is composed of all the terms in (11.29) that contain v(x). While the first term indicates how well v(x) explains our observations, the second and third terms impose prior constraints on the motion estimates that they should conform with the parametric flow model, and that they should vary smoothly within each region. To minimize this energy function, we employ the HCF method recently proposed by Chou and Brown [Cho 90]. HCF is a deterministic method designed to efficiently handle the optimization of multivariable problems with neighborhood interactions.

2. Update the segmentation field z, assuming that, the optical flow field v(x) is known. This step involves the minimization of all the terms in (11.29) which contain z as well as $\tilde{\mathbf{v}}(\mathbf{x})$, given by

$$E_2 = \alpha \sum_{\mathbf{x}} ||\mathbf{v}(\mathbf{x}) - \tilde{\mathbf{v}}(\mathbf{x})||^2 + \beta \sum_{\mathbf{x}_i} \sum_{\mathbf{x}_j \in \mathcal{N}_{\mathbf{x}_i}} V_C(z(\mathbf{x}_i), z(\mathbf{x}_j)) \tag{11.31}$$

The first term in (11.31) quantifies the consistency of $\tilde{\mathbf{v}}(\mathbf{x})$ and v(x). The second term is related to the a priori probability of the present configuration of the segmentation labels. We use an ICM procedure to optimize E_2 [Cha 93]. The mapping parameters $\mathbf{a}_i$ are updated by least squares estimation within each region.

An initial estimate of the optical flow field can be found using the Bayesian approach with a global smoothness constraint. Given this estimate, the segmentation labels can be initialized by a procedure similar to Wang and Adelson's [Wan 94]. The determination of the free parameters a, β, and γ is a design problem. One strategy is to choose them to provide a dynamic range correction so that each term

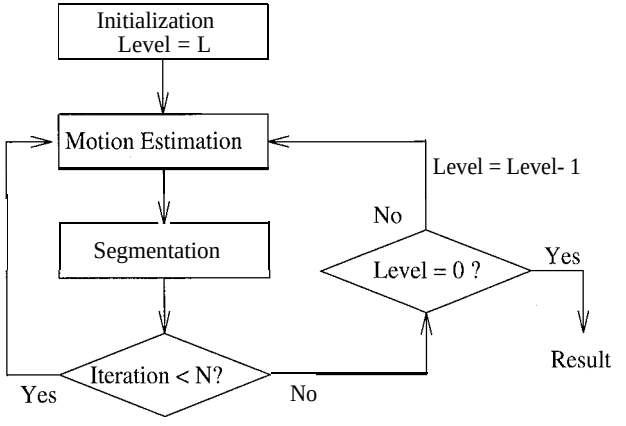


Figure 11.4: The block diagram of the simultaneous MAP algorithm

in the cost function (11.29) has equal emphasis. However, because the optimization is implemented in two steps, the ratio a/y also becomes of consequence. We recommend to select $1 \leq \alpha/\gamma \leq 5$, depending on how well the motion field can be represented by a piecewise-parametric model and whether we have a sufficient number of classes.

A hierarchical implementation of this algorithm is also possible by forming successive low-pass filtered versions of the images $\mathbf{g}_k$ and $\mathbf{g}_{k+1}$. Thus, the quantities $\mathbf{v}_1$, $\mathbf{v}_2$, and z can be estimated at different resolutions. The results of each hierarchy are used to initialize the next lower level. A block diagram of the hierarchical algorithm is depicted in Figure 11.4. Note that the Gibbsian model for the segmentation labels has been extended to include neighbors in scale by Kato *et* al. [Kat 93].

11.3.4 Relationship to Other Algorithms

It is important to recognize that this simultaneous estimation and segmentation framework not only enables 3-D motion and structure estimation in the presence of multiple moving objects, but also provides improved optical flow estimates. Several existing motion analysis algorithms can be formulated as special cases of this framework. If we retain only the first and the third terms in (11.29), and assume that all sites possess the same segmentation label, then we have Bayesian motion estimation with a global smoothness constraint. The motion estimation algorithm proposed by Iu [Iu 93] utilizes the same two terms, but replaces the $\delta(\cdot)$ function by a local outlier rejection function (Section 8.3.2).

The motion estimation and region labeling algorithm proposed by Stiller [Sti 94] (Section 8.3.3) involves all terms in (11.29), except the first term in (11.25). Fur-

thermore, the segmentation labels in Stiller's algorithm are used merely as tokens to allow for a piecewise smoothness constraint on the flow field, and do not attempt to enforce consistency of the flow vectors with a parametric component. We also note that the motion estimation algorithms of Konrad and Dubois [Kon 92, Dub 93] and Heitz and Bouthemy [Hei 93] whichuse line fields are fundamentally different in that they model discontinuities in the motion field, rather than modeling regions that correspond to different physical motions (Section 8.3.1).

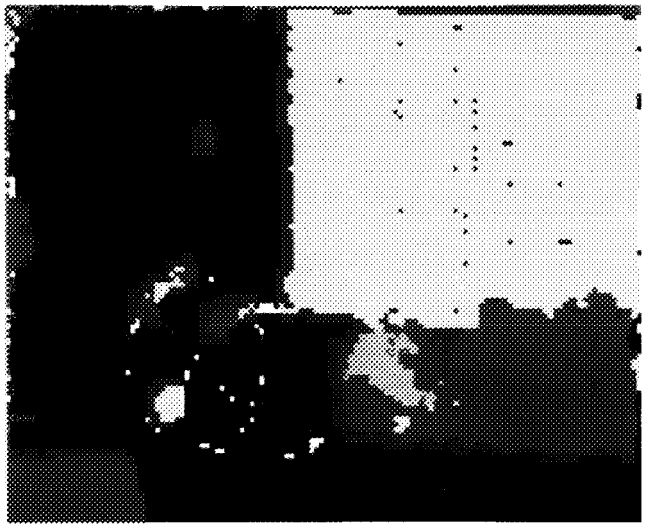
On the other hand, the motion segmentation algorithm of Murray and Buxton [Mur 87] (Section 11.2.2) employs only the second term in (11.25) and third term in (11.29) to model the conditional and prior pdf, respectively. Wang and Adelson [Wan 94] relies on the first term in (11.25) to compute the motion segmentation (Section 11.2.3). However, they also take the DFD of the parametric motion vectors into consideration when the closest match between the estimated and parametric motion vectors, represented by the second term, exceeds a threshold.

11.4 Examples

Examples are provided for optical flow segmentation using the Wang-Adelson (W-A) and Murray-Buxton (M-B) methods, as well as for simultaneous motion estimation and segmentation using the proposed MAP algorithm with the same two frames of the Mobile and Calendar sequence shown in Figure 5.9 (a) and (b). This is a challenging sequence, since there are several objects with distinct motions as depicted in Figure 11.1.

The W-A and M-B algorithms use the optical flow estimated by the Horn-Schunck algorithm, shown in Figure 5.11 (b), as input. We have set the number of regions to four. In order to find the representative affine motion parameters in the W-A algorithm, 8 x 8 and 16 x 16 seed blocks have been selected, and the affine parameters estimated for each of these blocks are clustered using the K-means algorithm. It is important that the size of the blocks be large enough to identify rotational motion. The result of the W-A segmentation is depicted in Figure 11.5 (a). We have initialized the M-B algorithm with the result of the W-A algorithm, and set the initial temperature equal to 1. The resulting segmentation after 50 iterations is shown in Figure 11.5 (b). Observe that the M-B algorithm eliminates small isolated regions on the calendar, and grows the region representing the rotating ball in the right direction by virtue of the probabilistic smoothness constraint that it employs.

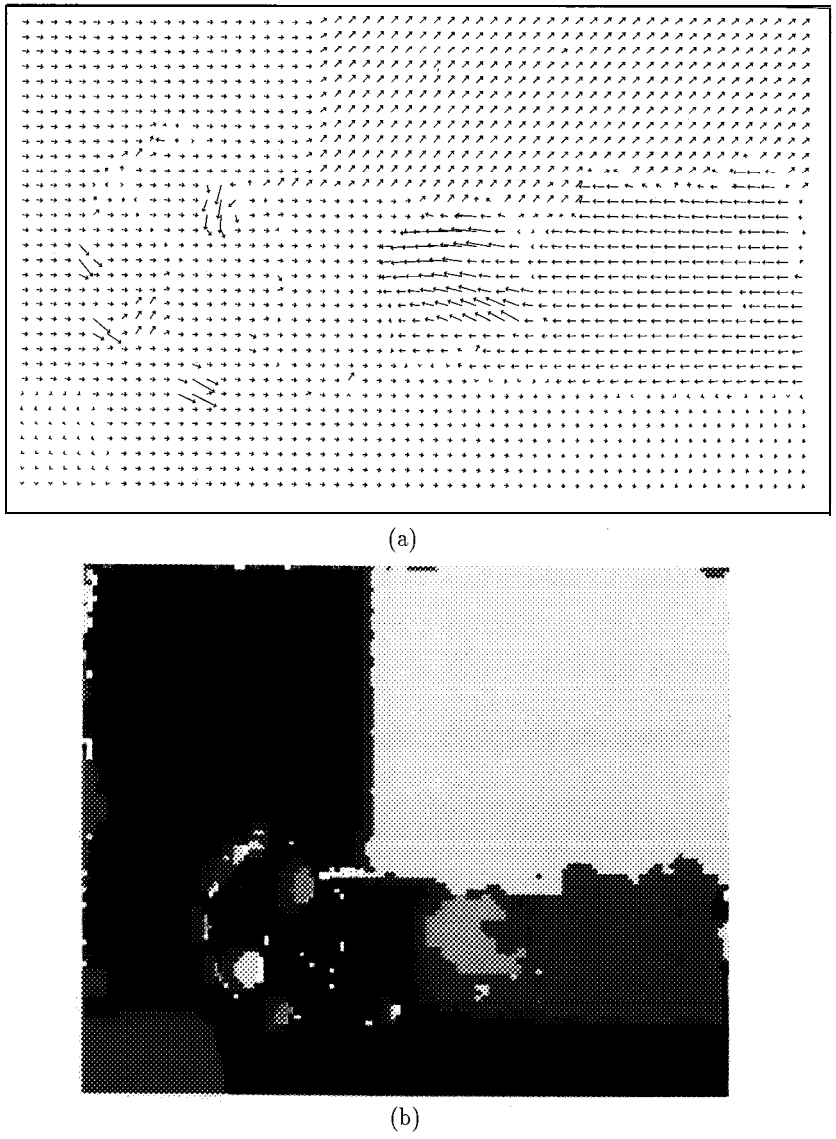
Next, we initialized the simultaneous MAP estimation and segmentation method, also by the result of the W-A algorithm. We have set $\alpha = \beta = 10$ and $\alpha/\gamma = 5$, since the motion field can be well-represented by a piecewise parametric model. The estimated optical flow and segmentation label fields are shown in Figure 11.6 (a) and (b), respectively. Note that the depicted motion field corresponds to the lower right portion of the segmentation field. The results show some improvement over the M-B method.



(a)

(b)

Figure 11.5: The segmentation field obtained by the a) Wang-Adelson method and b) Murray-Buxton method. (Courtesy Michael Chang)



(a)

(b)

Figure 11.6: a) The optical flow field and b) segmentation field estimated by the simultaneous MAP method after 100 iterations. (Courtesy Michael Chang)

11.5 Exercises

1. Show that the MAP segmentation reduces to the K-means algorithm if we assume the conditional pdf is Gaussian and no a *priori* information is available.
2. How would you apply the optimum threshold selection method discussed in Appendix B.l.l to change detection?
3. How would you modify (11.15) considering that (v_1, v_2) is the measured normal flow rather than the projected flow?
4. Do you prefer to model the flow discontinuities through a segmentation field or through line fields? Why?
5. Verify the relationships claimed in Section 11.3.3
6. Discuss how to choose the scale factors α, λ, γ, and ψ in (11.30) and (11.31).

Bibliography

[Adi 85] G. Adiv, "Determining three-dimensional motion and structure from optical flow generated by several moving objects," *IEEE Trans. Pattern* Anal. *Mach. Intell.*, vol. 7, pp. 384-401, 1985.

[Ber 92] J. R. Bergen, P. J. Burt, R. Hingorani, and S. Peleg, "A three-frame algorithm for estimating two-component image motion," *IEEE* Trans. *Putt.* Anal. *Mach. Intel.,* vol. 14, pp. 8866896, Sep. 1992.

[Cha 93] M. M. Chang, A. M. Tekalp, and M. I. Sezan, "Motion field segmentation using an adaptive MAP criterion," *Proc. Int. Conf. ASSP,* Minneapolis, MN, April 1993.

[Cha 94] M. M. Chang, M. I. Sezan, and A. M. Tekalp, "An algorithm for simultaneous motion estimation and scene segmentation," *Proc. Int. Conf. ASSP,* Adelaide, Australia, April 1994.

[Cho 90] P. B. Chou and C. M. Brown, "The theory and practice of Bayesian image labeling," *Int. J. Comp. Vision,* vol. 4, pp. 1855210, 1990.

[Die 91] N. Diehl, "Object-oriented motion estimation and segmentation in image sequences," Signal *Processing:* Image *Comm.,* vol. 3, pp. 23-56, 1991.

[Dub 93] E. Dubois and J. Konrad, "Estimation of 2-D motion fields from image sequences with application to motion-compensated processing," in *Motion* Analysis and *Image Sequence Processing,* M. I. Sezan and R. L. Lagendijk, eds., Norwell, MA: Kluwer, 1993.

[Hoe 88] M. Hoetter and R. Thoma, "Image segmentation based on object oriented mapping parameter estimation," Signal *Pm.*, vol. 15, pp. 315–334, 1988.

[Hsu 94] S. Hsu, P. Anandan, and S. Peleg, "Accurate computation of optical flow by using layered motion representations," *Proc. Int. Conf. Patt. Recog.*, Jerusalem, Israel, pp. 743-746, Oct. 1994.

[Iu 93] S.-L. Iu, "Robust estimation of motion vector fields with discontinuity and occlusion using local outliers rejection," *SPIE*, vol. 2094, pp. 588–599, 1993.

[Kat 93] Z. Kato, M. Berthod, and J. Zerubia, "Parallel image classification using multiscale Markov random fields," *Proc. IEEE Id. Conf. ASSP*, Minneapolis, MN, pp. V137–140, April 1993.

[Mur 87] D. W. Murray and B. F. Buxton, "Scene segmentation from visual motion using global optimization," *IEEE* Trans. Pall. Anal. *Mach. Intel.*, vol. 9, no. 2, pp. 220-228, Mar. 1987.

[Sti 93] C. Stiller, "A statistical image model for motion estimation," *Proc. Int. Conf. ASSP*, Minneapolis, MN, pp. V193–196, April 1993.

[Sti 94] C. Stiller, "Object-oriented video coding employing dense motion fields," *Proc. Int. Conf. ASSP*, Adelaide, Australia, April 1994.

[Tho 80] W. B. Thompson, "Combining motion and contrast for segmentation," *IEEE Trans. Pattern* Anal. Mach. *Intel.*, vol. 2, pp. 543-549, 1980.

[Wan 94] J. Y. A. Wang and E. Adelson, "Representing moving images with layers," *IEEE* Trans. on Image *Proc.*, vol. 3, pp. 625-638, Sep. 1994.

[Wu 93] S. F. Wu and J. Kittler, "A gradient-based method for general motion estimation and segmentation," J. *Vis.* Comm. Image *Rep.*, vol. 4, no. 1, pp. 25-38, Mar. 1993.

Chapter 12

STEREO AND MOTION TRACKING

Because 3-D motion and structure estimation from two monocular views has been found to be highly noise-sensitive, this chapter presents two new approaches, stereo imaging and motion tracking, that offer more robust estimates. They are discussed in Sections 12.1 and 12.2, respectively. In stereo imaging, we acquire a pair of right and left images, at each instant. More robust motion and structure estimation from two stereo pairs is possible, because structure parameters can be estimated using stereo triangulation, hence decoupling 3-D motion and structure estimation. Moreover, the mutual relationship between stereo disparity and 3-D motion parameters can be utilized for stereo-motion fusion, thereby improving the accuracy of both stereo disparity and 3-D motion estimation. In motion tracking, it is assumed that a long sequence of monocular or stereo video is available. In this case, more robust estimation is achieved by assuming a temporal dynamics; that is, a model describing the temporal evolution of the motion. Motion between any pair of frames is expected to obey this model. Then batch or recursive filtering techniques can be employed to track the 3-D motion parameters in time. At present, stereo-motion fusion and motion tracking using Kalman filtering are active research topics of significant interest.

12.1 Motion and Structure from Stereo

It is well known that 3-D motion and structure estimation from monocular video is an ill-posed problem, especially when the object is relatively far away from the camera. Several researchers have employed stereo sequences for robust 3-D motion and structure estimation [Dho 89, Hua 94]. Stereo imaging also alleviates the scale ambiguity (between depth and translation) inherent in monocular imaging, since a properly registered stereo pair contains information about the scene structure.

In the following, we first briefly present the principles of still-frame stereo imaging. Subsequently, motion and structure estimation from a sequence of stereo pairs will be discussed.

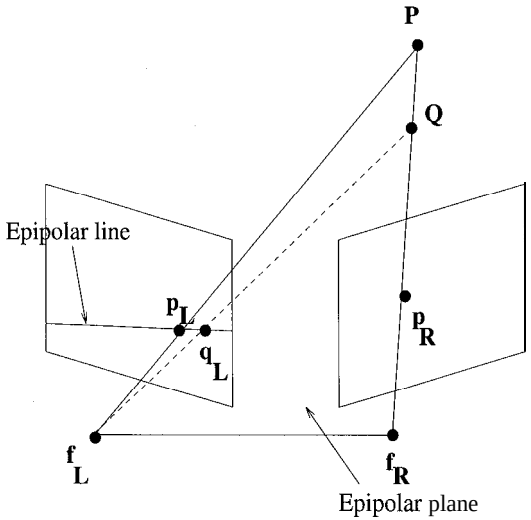


Figure 12.1: The epipolar plane and lines.

12.1.1 Still-Frame Stereo Imaging

Let $\mathbf{p}_L = (x_{1L}, x_{2L})$ and $\mathbf{p}_R = (x_{1R}, x_{2R})$ denote the perspective projections of a point $\mathbf{P} = (X_1, X_2, \text{Xs})$ onto the left and right image planes, respectively. In general, the point $\mathbf{P}$ and the focal points of the left and right cameras, $\mathbf{f}_L$ and $\mathbf{f}_R$, define the so-called epipolar plane, as depicted in Figure 12.1. The intersection of the epipolar plane with the left and right image planes is called the epipolar lines. It follows that the perspective projection of a point anywhere on the epipolar plane falls on the epipolar lines.

A cross-section of the imaging geometry, as seen in the $X_1 - X_3$ plane, is depicted in Figure 12.2. Here, $\mathbf{C}_W$, $\mathbf{C}_R$, and $\mathbf{C}_L$ denote the world, right camera, and left camera coordinate systems, respectively, and the focal lengths of both cameras are assumed to be equal, $f_R = f_L = f$. Let $\mathbf{P}_R = (X_{1R}, X_{2R}, X_{3R})$ and $\mathbf{P}_L = (X_{1L}, X_{2L}, X_{3L})$ denote the representation of the point $\mathbf{P} = (X_1, X_2, X_3)$ in $\mathbf{C}_R$ and $\mathbf{C}_L$, respectively. Then the coordinates of P in the right and left coordinate systems are given by

$$\mathbf{P}_R = \mathbf{R}_R\mathbf{P} + \mathbf{T}_R \tag{12.1}$$

$$\mathbf{P}_L = \mathbf{R}_L\mathbf{P} + \mathbf{T}_L \tag{12.2}$$

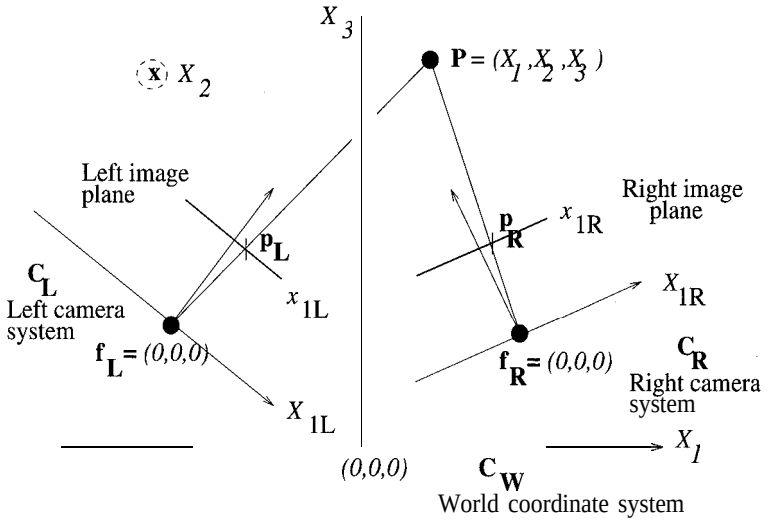


Figure 12.2: Stereo image formation.

where $\mathbf{R}_R$ and $\mathbf{T}_R$, and $\mathbf{R}_L$ and $\mathbf{T}_L$ are the extrinsic parameters of the camera model indicating relative positions of $\mathbf{C}_R$ and $\mathbf{C}_L$ with respect to $\mathbf{C}_W$, respectively. Combining equations (12.1) and (12.2), we have

$$\begin{aligned}\mathbf{P}_L &= \mathbf{R}_L\mathbf{R}_R^{-1}\mathbf{P}_R - \mathbf{R}_L\mathbf{R}_R^{-1}\mathbf{T}_R + \mathbf{T}_L \\ &\doteq \mathbf{M}\mathbf{P}_R + \mathbf{B}\end{aligned} \tag{12.3}$$

where M and B are known as the relative configuration parameters [Wen 92]. Then, based on similar triangles, the perspective projection of the point $\mathbf{P}$ into the left and right image planes. can be expressed as

$$\begin{aligned} x_{1L} &= f\frac{X_{1L}}{X_{3L}}, & x_{2L} &= f\frac{X_{2L}}{X_{3L}} \\ x_{1R} &= f\frac{X_{1R}}{X_{3R}}, & x_{2R} &= f\frac{X_{2R}}{X_{3R}} \end{aligned} \tag{12.4}$$

respectively. Substituting (12.4) into (12.3), we have

$$\frac{X_{3L}}{f}\begin{bmatrix} x_{1L} \\ x_{2L} \\ f \end{bmatrix} = \frac{X_{3R}}{f}\mathbf{M}\begin{bmatrix} x_{1R} \\ x_{2R} \\ f \end{bmatrix} + \mathbf{B} \tag{12.5}$$

The structure from stereo problem refers to estimating the coordinates (X_1, X_2, X_3) of a 3-D point **P,** given the corresponding points (x_{1L}, x_{2L}) and

(x_{1R}, x_{2R}) in the left and right image planes, and the extrinsic camera calibration parameters. We first solve for X_{3L} and X_{3R} from (12.5). It can be shown that only two of the three equations in (12.5) are linearly independent in the event of noise-free point correspondences. In practice, (12.5) should be solved in the least squares sense to find X_{3L} and X_{3R}. Next, the 3-D coordinates of the point **P** in the left and right camera coordinate systems can be computed from (12.4). Finally, a least squares estimate of **P** in the world coordinate system can be obtained from (12.1) and (12.2). A closed-form expression for **P** in the world coordinate system can be written in the special case when the left and right camera coordinates are parallel and aligned with the $X_1 - X_2$ plane of the world coordinates. Then

$$X_1 = \frac{b\,(x_{1L} + x_{1R})}{x_{1L} - x_{1R}}, \quad X_2 = \frac{2\,b\,x_{2L}}{x_{1L} - x_{1R}}, \quad X_3 = \frac{2\,f\,b}{x_{1L} - x_{1R}} \tag{12.6}$$

where b refers to half of the distance between the two cameras (assuming that they are symmetrically placed on both sides of the origin).

Finding corresponding pairs of image points is known as the stereo matching problem [Mar 78, Dho 89, Bar 93]. We define

$$\mathbf{w}(\mathbf{x}_R) \doteq (w_1(\mathbf{x}_R), w_2(\mathbf{x}_R)) = (x_{1R} - x_{1L}, x_{2R} - x_{2L}) \tag{12.7}$$

as the stereo disparity (taking the right image as the reference). Observe that the matching problem always involves a 1-D search along one of the epipolar lines. Suppose we start with the point $\mathbf{p}_R$. The object point **P** must lie along the line combining $\mathbf{p}_R$ and $\mathbf{f}_R$. Note that the loci of the projection of all points Q along this line onto the left image plane define the epipolar line for the left image plane (see Figure 12.1). Thus, it suffices to search for the matching point $\mathbf{p}_L$ along the left epipolar line.

12.1.2 3-D Feature Matching for Motion Estimation

The simplest method for motion and structure estimation from stereo would be to decouple the structure estimation and motion estimation steps by first estimating the depth at selected image points from the respective stereo pairs at times t and t' using 2-D feature matching between the left and right pairs, and temporal matching in one of the left or right channels, independently. Three-D rigid motion parameters can then be estimated by 3-D feature matching between the frames t and t' [Aru 87, Hua 89c]. However, this scheme neglects the obvious relationship between the estimated disparity fields at times t and t'. Alternatively, more sophisticated stereo-motion fusion schemes, which aim to enforce the mutual consistency of the resulting motion and disparity values, have also been proposed [Wax 86]. Both approaches are introduced in the following.

There are, in general, two approaches for the estimation of the 3-D motion parameters **R** and **T**, given two stereo pairs at times t and t': 3-D to 3-D feature matching and 3-D to 2-D feature matching. If we start with an arbitrary

point (x_{1R}, x_{2R}) in the right image at time t, we can first find the matching point (x_{1L}, x_{2L}) in the left image by still-frame stereo matching, which determines a 3-D feature point **P** = (Xi, X_2, X_3) in the world coordinate system at time t, as discussed in the previous section. Next, we locate the corresponding 2-D point (x'_{1R}, x'_{2R}) at time t' that matches (x_{1R}, x_{2R}) by 2-D motion estimation. At this moment, we can estimate the 3-D motion parameters using a 3-D to 2-D point-matching algorithm based on a set of matching (x'_{1R}, x'_{2R}) and **P.** Alternatively, we can determine (x'_{1L}, x'_{2L}) at time t' again by still-frame stereo matching or by 2-D motion estimation using the left images at time t and t', and then use a 3-D to 3-D point-matching algorithm based on a set of matching **P'** and **P.**

3-D to 3-D Methods

Given N 3-D point correspondences $(\mathbf{P}_i, \mathbf{P}'_i)$ (expressed in the world coordinate system) at two different times, obtained by still-frame stereo or other range-finding techniques, which lie on the same rigid object; the rotation matrix **R** (with respect to the world coordinate system) and the translation vector **T** can be found from

$$\mathbf{P}'_i = \mathbf{R}\mathbf{P}_i + \mathbf{T}, \quad i = 1, \ldots, N \tag{12.8}$$

It is well known that, in general, three noncollinear point correspondences are necessary and sufficient to determine **R** and **T** uniquely [Hua 94]. In practice, point correspondences are subject to error, and therefore, one prefers to work with more than the minimum number of point correspondences. In this case, **R** and **T** can be found by minimizing

$$\sum_{i=1}^{N} ||\mathbf{P}'_i - (\mathbf{R}\mathbf{P}_i + \mathbf{T})||^2 \tag{12.9}$$

subject to the constraints that **R** is a valid rotation matrix. Robust estimation procedures can be employed to eliminate outliers from the set of feature correspondences to improve results. Observe that if the small angle assumption is applicable, the problem reduces to solving a set of linear equations.

Establishing temporal relationships between 3-D lines and points from a sequence of depth maps computed independently from successive still-frame pairs to estimate 3-D motion was also proposed [Kim 87].

3-D to 2-D Methods

From (9.1) and (12.1), we can express the equation of 3-D motion with respect to the right camera coordinate system as

$$\mathbf{X}'_R = \mathbf{R}_R\mathbf{X}_R + \mathbf{T}_R \tag{12.10}$$

where

$$\begin{aligned} \mathbf{R}_R &= \mathbf{R}_R\mathbf{R}\mathbf{R}_R^{-1} \\ \mathbf{T}_R &= \mathbf{R}_R\mathbf{T} + \mathbf{T}_R - \mathbf{R}_R\mathbf{T}_R \end{aligned}$$

denote the rotation matrix and the translation vector, respectively, with respect to the right camera coordinate system. Then, substituting (12.10) into (12.4), we have a computed correspondence in the right image given by

$$\tilde{x}'_{1R} = f\frac{X'_{1R}}{X'_{3R}} \quad \text{and} \quad \tilde{x}'_{2R} = f\frac{X'_{2R}}{X'_{3R}} \tag{12.11}$$

Given N 3-D to 2-D point correspondence measurements, $(\mathbf{P}_{iR}, \mathbf{p}'_{iR})$, where $\mathbf{P}_R = (X_{1R}, X_{2R}, X_{3R})$ refers to a 3-D point with respect to the right camera coordinate system at time t, and $\mathbf{p}'_R = (x'_{1R}, x'_{2R})$ denotes the corresponding right image point at time t'; we can estimate $\hat{\mathbf{R}}_R$ and $\hat{\mathbf{T}}_R$ by minimizing

$$\sum_{i=1}^{N} ||\mathbf{p}'_{iR} - \tilde{\mathbf{p}}'_{iR}||^2 \tag{12.12}$$

where $\tilde{\mathbf{p}}'_{iR}$ refers to the corresponding image point computed by (12.11) as a function of $\hat{\mathbf{R}}_R$ and $\hat{\mathbf{T}}_R$. The minimization of (12.12) can be posed as a nonlinear least squares problem in the six unknown 3-D motion parameters. Once again, a linear formulation is possible if the small angle of rotation approximation is applicable. Assuming that the extrinsic parameters of the camera system $\mathbf{R}_R$ and $\mathbf{T}_R$ are known, the rotation and translation parameters $\mathbf{R}$ and $\mathbf{T}$ in the world coordinates can be easily recovered from $\mathbf{R}_R$ and $\mathbf{T}_R$ based on (12.10).

The difficulty with both 3-D to 3-D and 2-D to 3-D approaches is in the process of establishing the matching 3-D and/or 2-D features. Stereo disparity estimation and estimation of feature correspondences in the temporal direction are, individually, both ill-posed problems, complicated by the aperture and occlusion problems; hence, the motivation to treat these problems simultaneously.

12.1.3 Stereo-Motion Fusion

Theoretically, in a time-varying stereo pair, the stereo matching need be performed only for the initial pair of frames. For subsequent pairs, stereo correspondences can be predicted by means of the left and right optical flow vectors. Stereo matching would only be required for those features that newly enter the field of view. However, because both optical flow estimation and disparity estimation are individually ill-posed problems, several studies have recently been devoted to fusion of stereo disparity and 3-D motion estimation in a mutually beneficial way. Stereo-motion fusion methods impose the constraint that the loop formed by the disparity and motion correspondence estimates, in Figure 12.3, must be a closed loop. Richards [Ric 85] and Waxman *et* al. [Wax 86] derived an analytical expression for the temporal rate of change of the disparity to disparity ratio as a function of the 3-D motion and structure parameters. They then proposed a simultaneous stereo and motion estimation method based on this relation. Aloimonos *et al.* [Alo 90] proposed an algebraic solution which requires no point-to-point correspondence estimation for the

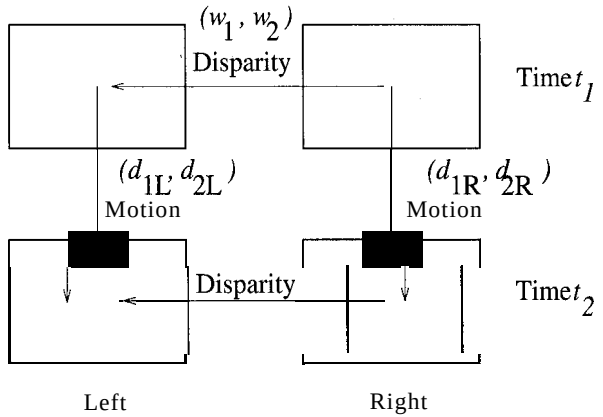


Figure 12.3: Stereo-motion fusion.

special case of 3-D planar objects. Methods to integrate stereo matching and optical flow estimation using multiresolution edge matching and dynamic programming [Liu 93] and constrained optimization [Tam 91] have also been proposed.

In the following, we first develop a maximum a *posteriori* probability (MAP) estimation framework (see Chapter 8) for stereo-motion fusion, based on dense displacement and disparity field models, in the case of a single rigid object in motion. Note that stereo-motion fusion with a set of isolated feature points (instead of dense fields) can be posed as a maximum-likelihood problem, which is a special case of this MAP framework. The MAP framework is extended to the case of multiple moving objects in the next section.

Dense Fields: Let $\mathbf{d}_R(\mathbf{x}_R) = (d_{1R}(\mathbf{x}_R), d_{2R}(\mathbf{x}_R))$ and $\mathbf{d}_L(\mathbf{x}_L) = (d_{1L}(\mathbf{x}_L), d_{2L}(\mathbf{x}_L))$ denote the 2-D displacement fields computed at each pixel $\mathbf{x}_R = (x_{1R}, x_{2R})$ and $\mathbf{x}_L = (x_{1L}, x_{2L})$ of the right and left images, respectively, and $\mathbf{w}(\mathbf{x}_R)$ denote the disparity field at each pixel $\mathbf{x}_R$ of the right image. Let $\mathbf{d}_{1R}$, $\mathbf{d}_{2R}$, $\mathbf{d}_{1L}$, $\mathbf{d}_{2L}$, $\mathbf{w}_1$, and $\mathbf{w}_2$ denote vectors formed by lexicographic ordering of the scalars $d_{1R}(\mathbf{x}_R)$, $d_{2R}(\mathbf{x}_R)$, $d_{1L}(\mathbf{x}_L)$, $d_{2L}(\mathbf{x}_L)$, $w_1(\mathbf{x}_R)$, and $w_2(\mathbf{x}_R)$ at all pixels, respectively. We wish to estimate $\mathbf{d}_{1R}$, $\mathbf{d}_{2R}$, $\mathbf{d}_{1L}$, $\mathbf{d}_{2L}$, $\mathbf{w}_1$, and $\mathbf{w}_2$, given $\mathbf{I}_L$, $\mathbf{I}_R$, $\mathbf{I}'_L$, $\mathbf{I}'_R$, the left and right images at times t and t', respectively, in order to maximize the joint a *posteriori* probability density function (pdf)

$$\begin{aligned} p(\mathbf{d}_{1R}, \mathbf{d}_{2R}, \mathbf{d}_{1L}, \mathbf{d}_{2L}, \mathbf{w}_1, \mathbf{w}_2 | \mathbf{I}'_L, \mathbf{I}'_R, \mathbf{I}_L, \mathbf{I}_R) \propto \\ p(\mathbf{I}'_L, \mathbf{I}'_R, \mathbf{I}_L | \mathbf{d}_{1R}, \mathbf{d}_{2R}, \mathbf{d}_{1L}, \mathbf{d}_{2L}, \mathbf{w}_1, \mathbf{w}_2, \mathbf{I}_R) \\ p(\mathbf{d}_{1R}, \mathbf{d}_{2R}, \mathbf{d}_{1L}, \mathbf{d}_{2L} | \mathbf{w}_1, \mathbf{w}_2, \mathbf{I}_R) p(\mathbf{w}_1, \mathbf{w}_2 | \mathbf{I}_R) \end{aligned} \tag{12.13}$$

where $p(\mathbf{I}'_L, \mathbf{I}'_R, \mathbf{I}_L | \mathbf{d}_{1R}, \mathbf{d}_{2R}, \mathbf{d}_{1L}, \mathbf{d}_{2L}, \mathbf{w}_1, \mathbf{w}_2, \mathbf{I}_R)$ provides a measure of how well

the present displacement and disparity estimates conform with the observed frames, $p(\mathbf{d}_{1R}, \mathbf{d}_{2R}, \mathbf{d}_{1L}, \mathbf{d}_{2L}|\mathbf{w}_1, \mathbf{w}_2, \mathbf{I}_R)$ quantifies the consistency of the 2-D motion field with the 3-D motion and structure parameters and imposes a local smoothness constraint on it, and $p(\mathbf{w}_1, \mathbf{w}_2|\mathbf{I}_R)$ enforces a local smoothness constraint on the disparity field. Given the MAP estimates of $\mathbf{d}_{1R}$, $\mathbf{d}_{2R}$, $\mathbf{d}_{1L}$, $\mathbf{d}_{2L}$, $\mathbf{w}_1$, and $\mathbf{w}_2$, the 3-D rotation matrix **R,** the 3-D translation vector **T,** and the scene depth X_3 at each pixel can be estimated by one of the 3-D to 3-D or 3-D to 2-D matching techniques discussed in Section 12.1.2.

The three pdfs on the right hand side of (12.13) are assumed to be Gibbsian with the potential functions $U_1(\cdot)$, $U_2(\cdot)$, and Us(.), respectively, that are given by

$$U_1(\mathbf{I}'_L, \mathbf{I}'_R, \mathbf{I}_L|\mathbf{d}_{1R}, \mathbf{d}_{2R}, \mathbf{d}_{1L}, \mathbf{d}_{2L}, \mathbf{w}_1, \mathbf{w}_2, \mathbf{I}_R) = \tag{12.14}$$
$$\sum_{\mathbf{x}_R \in I_R} \big[(I_L(\mathbf{x}_R + \mathbf{w}(\mathbf{x}_R)) - I_R(\mathbf{x}_R))^2 + (I'_R(\mathbf{x}_R + \mathbf{d}_R(\mathbf{x}_R)) - I_R(\mathbf{x}_R))^2 + (I'_L(\mathbf{x}_L + \mathbf{d}_L(\mathbf{x}_L)) - I_L(\mathbf{x}_L))^2 \big]$$

$$U_2(\mathbf{d}_{1R}, \mathbf{d}_{2R}, \mathbf{d}_{1L}, \mathbf{d}_{2L}|\mathbf{w}_1, \mathbf{w}_2, \mathbf{I}_R) = \tag{12.15}$$
$$\sum_{\mathbf{x}_R \in I_R} \big(||\mathbf{d}_R(\mathbf{x}_R) - \tilde{\mathbf{d}}_R(\mathbf{x}_R)||^2 + ||\mathbf{d}_L(\mathbf{x}_L) - \tilde{\mathbf{d}}_L(\mathbf{x}_L)||^2 \big)$$
$$+ \alpha \sum_{\mathbf{x}_{Ri}} \left(\sum_{\mathbf{x}_{Rj} \in \mathcal{N}_{\mathbf{x}_{Ri}}} ||\mathbf{d}_R(\mathbf{x}_{Ri}) - \mathbf{d}_R(\mathbf{x}_{Rj})||^2 + \sum_{\mathbf{x}_{Lj} \in \mathcal{N}_{\mathbf{x}_{Li}}} ||\mathbf{d}_L(\mathbf{x}_{Li}) - \mathbf{d}_L(\mathbf{x}_{Lj})||^2 \right)$$

where $\mathbf{x}_L = \mathbf{x}_R + \mathbf{w}(\mathbf{x}_R)$ is the corresponding point on the left image,

$$\mathbf{d}_R(\mathbf{x}_R) \doteq \tilde{\mathbf{x}}'_R - \mathbf{x}_R \qquad \text{and} \qquad \mathbf{d}_L(\mathbf{x}_L) \doteq \tilde{\mathbf{x}}'_L - \mathbf{x}_L \tag{12.16}$$

denote the projected 3-D displacement field onto the right and left image planes, respectively, in which $\tilde{\mathbf{x}}'_R$ and $\tilde{\mathbf{x}}'_L$ are computed as functions of the 3-D motion and disparity parameters (step 1 of the algorithm below), α is a constant, $\mathcal{N}_{\mathbf{x}_{Ri}}$ and $\mathcal{N}_{\mathbf{x}_{Li}}$ denote neighborhoods of the sites $\mathbf{x}_{Ri}$ and $\mathbf{x}_{Li} = \mathbf{x}_{Ri} + \mathbf{w}(\mathbf{x}_{Ri})$, respectively, and

$$U_3(\mathbf{w}_1, \mathbf{w}_2|\mathbf{I}_R) = \sum_{\mathbf{x}_{Ri}} \sum_{\mathbf{x}_{Rj} \in \mathcal{N}_{\mathbf{x}_{Ri}}} ||\mathbf{w}(\mathbf{x}_{Ri}) - \mathbf{w}(\mathbf{x}_{Rj})||^2 \tag{12.17}$$

The maximization of (12.13) can, then, be performed by the following two-step iteration process:

1. Given the present estimates of $\mathbf{d}_{1R}$, $\mathbf{d}_{2R}$, $\mathbf{d}_{1L}$, $\mathbf{d}_{2L}$, $\mathbf{w}_1$, and $\mathbf{w}_2$:

i) reconstruct a set of 3-D points (Xi, X_2, X_3) as described in Section 12.1.1,

ii) estimate R and T by means of 3-D to 2-D point matching or 3-D to 3-D point matching, and

iii) calculate the projected displacement fields $\tilde{\mathbf{d}}_R(\mathbf{x}_R)$ and $\tilde{\mathbf{d}}_L(\mathbf{x}_L)$ from (12.1), (12.2), and (12.4).

2. Perturb the displacement $\mathbf{d}_{1R}$, $\mathbf{d}_{2R}$, $\mathbf{d}_{1L}$, $\mathbf{d}_{2L}$ and the disparity $\mathbf{w}_1$, $\mathbf{w}_2$ fields in order to minimize

$$E = U_1(\mathbf{I}'_L, \mathbf{I}'_R, \mathbf{I}_L|\mathbf{d}_{1R}, \mathbf{d}_{2R}, \mathbf{d}_{1L}, \mathbf{d}_{2L}, \mathbf{w}_1, \mathbf{w}_2, \mathbf{I}_R) + \gamma_1 U_2(\mathbf{d}_{1R}, \mathbf{d}_{2R}, \mathbf{d}_{1L}, \mathbf{d}_{2L}|\mathbf{w}_1, \mathbf{w}_2, \mathbf{I}_R) + \gamma_2 U_3(\mathbf{w}_1, \mathbf{w}_2|\mathbf{I}_R)$$

where γ_1 and γ_2 are some constants.

The algorithm is initialized with motion and disparity estimates obtained by existing decoupled methods. A few iterations using the ICM algorithm are usually sufficient. Observe that $U_1(\cdot)$ requires that the 2-D displacement and disparity estimates be consistent with all four frames, while the first term of $U_2(\cdot)$ enforces consistency of the 2-D and 3-D motion estimates. The second term of $U_2(\cdot)$ and $U_3(\cdot)$ impose smoothness constraints on the motion and disparity fields, respectively.

Isolated Feature *Points:* Because stereo-motion fusion with dense motion and disparity fields may be computationally demanding, the proposed MAP formulation can be simplified into a maximum-likelihood (ML) estimation problem by using selected feature points. This is achieved by turning off the smoothness constraints imposed by the potential (12.17) and the second term of (12.15), which does not apply in the case of isolated feature points [Alt 95]. Results using 14 feature points are shown in Section 12.3. The methods cited here are mere examples of many possible approaches. Much work remains for future research in stereo-motion fusion.

12.1.4 Extension to Multiple Motion

The simultaneous MAP estimation and segmentation framework presented in Section 11.3 can be easily extended to stereo-motion fusion in the presence of multiple moving objects. Here, pairs of disparity $\mathbf{w}(\mathbf{x}_R)$ and displacement vectors $\mathbf{d}_R(\mathbf{x}_R)$ are segmented into K regions, where within each region the displacement vectors are expected to be consistent with a single set of 3-D motion parameters, and the disparity is allowed to vary smoothly. The regions are identified by the label field $\mathbf{z}$.

We seek the disparity, motion, and segmentation field configuration that would maximize the *a posteriori* probability density function

$$p(\mathbf{d}_{1R}, \mathbf{d}_{2R}, \mathbf{d}_{1L}, \mathbf{d}_{2L}, \mathbf{w}_1, \mathbf{w}_2, \mathbf{z}|\mathbf{I}'_L, \mathbf{I}'_R, \mathbf{I}_L, \mathbf{I}_{R,}) \propto p(\mathbf{I}'_L, \mathbf{I}'_R, \mathbf{I}_L|\mathbf{d}_{1R}, \mathbf{d}_{2R}, \mathbf{d}_{1L}, \mathbf{d}_{2L}, \mathbf{w}_1, \mathbf{w}_2, \mathbf{z}, \mathbf{I}_R) p(\mathbf{d}_{1R}, \mathbf{d}_{2R}, \mathbf{d}_{1L}, \mathbf{d}_{2L}|\mathbf{w}_1, \mathbf{w}_2, \mathbf{z}, \mathbf{I}_R)\, p(\mathbf{w}_1, \mathbf{w}_2|\mathbf{z}, \mathbf{I}_R)\, p(\mathbf{z}|\mathbf{I}_R) \tag{12.18}$$

given $\mathbf{I}_L$, $\mathbf{I}_R$, $\mathbf{I}'_L$, $\mathbf{I}'_R$. We assume that all pdfs on the right-hand side of (12.18) are Gibbsian with the potential functions

$$U_1(\mathbf{I}'_L, \mathbf{I}'_R, \mathbf{I}_L|\mathbf{d}_{1R}, \mathbf{d}_{2R}, \mathbf{d}_{1L}, \mathbf{d}_{2L}, \mathbf{w}_1, \mathbf{w}_2, \mathbf{z}, \mathbf{I}_R) =$$
$$\sum_{\mathbf{x}_R \in \mathbf{I}_R} \big[(I_L(\mathbf{x}_R + \mathbf{w}(\mathbf{x}_R)) - I_R(\mathbf{x}_R))^2 + (I'_R(\mathbf{x}_R + \mathbf{d}_R(\mathbf{x}_R)) - I_R(\mathbf{x}_R))^2 + (I'_L(\mathbf{x}_L + \mathbf{d}_L(\mathbf{x}_L)) - I_L(\mathbf{x}_L))^2 \big] \tag{12.19}$$

$$U_2(\mathbf{d}_{1R}, \mathbf{d}_{2R}, \mathbf{d}_{1L}, \mathbf{d}_{2L} | \mathbf{w}_1, \mathbf{w}_2, \mathbf{z}) =$$
$$\sum_{\mathbf{x}_R \in \mathbf{I}_R} \left(||\mathbf{d}_R(\mathbf{x}_R) - \tilde{\mathbf{d}}_R(\mathbf{x}_R)||^2 + ||\mathbf{d}_L(\mathbf{x}_L) - \tilde{\mathbf{d}}_L(\mathbf{x}_L)||^2 \right)$$
$$+ \sum_{\mathbf{x}_{Ri}} \left(\sum_{\mathbf{x}_{Rj} \in \mathcal{N}_{\mathbf{x}_{Ri}}} ||\mathbf{d}_R(\mathbf{x}_{Ri}) - \mathbf{d}_R(\mathbf{x}_{Rj})||^2 \right.$$
$$\left. + \sum_{\mathbf{x}_{Lj} \in \mathcal{N}_{\mathbf{x}_{Li}}} ||\mathbf{d}_L(\mathbf{x}_{Li}) - \mathbf{d}_L(\mathbf{x}_{Lj})||^2 \right) \delta(z(\mathbf{x}_{Ri}) - z(\mathbf{x}_{Rj})) \quad (12.20)$$

where $\tilde{\mathbf{d}}_R$ is the projected 3-D motion given by (12.16), and $\mathbf{x}_L$, $\mathcal{N}_{\mathbf{x}_{Ri}}$, and $\mathcal{N}_{\mathbf{x}_{Li}}$ are as defined in the previous section,

$$U_3(\mathbf{w}_1, \mathbf{w}_2 | \mathbf{z}) = \sum_{\mathbf{x}_{Ri}} \sum_{\mathbf{x}_{Rj} \in \mathcal{N}_{\mathbf{x}_{Ri}}} ||\mathbf{w}(\mathbf{x}_{Ri}) - \mathbf{w}(\mathbf{x}_{Rj})||^2 \, \delta(z(\mathbf{x}_{Ri}) - z(\mathbf{x}_{Rj}))$$

imposes a piecewise smoothness constraint on the disparity field, and

$$U_4(\mathbf{z}) = \sum_{\mathbf{x}_{Ri}} \sum_{\mathbf{x}_{Rj} \in \mathcal{N}_{\mathbf{x}_{Ri}}} V_C(z(\mathbf{x}_{Ri}), \; z(\mathbf{x}_{Rj})) \quad (12.21)$$

where $V_C(z(\mathbf{x}_{Ri}), z(\mathbf{x}_{Rj}))$ is defined by (11.28). Observe that, in the case of multiple motion, the smoothness constraints are turned on for motion and disparity vectors which possess the same segmentation label.

The maximization of (12.18) can be performed in an iterative manner. Each iteration consists of the following steps:

1. Given the initial estimates of the optical flow, $\mathbf{d}_{1R}$, $\mathbf{d}_{2R}$, $\mathbf{d}_{1L}$, $\mathbf{d}_{2L}$ the disparity $\mathbf{w}_1$, $\mathbf{w}_2$, and the segmentation z, estimate the 3-D motion parameters **R** and **T** for each segment. (Similar to step 1 of the algorithm in the previous subsection.)

2. Update the optical flow $\mathbf{d}_{1R}$, $\mathbf{d}_{2R}$, $\mathbf{d}_{1L}$, and $\mathbf{d}_{2L}$ assuming that the disparity field and the segmentation labels are given by minimizing

$$E_1 = U_1(\mathbf{I}'_L, \mathbf{I}'_R, \mathbf{I}_L | \mathbf{d}_{1R}, \mathbf{d}_{2R}, \mathbf{d}_{1L}, \mathbf{d}_{2L}, \mathbf{w}_1, \mathbf{w}_2, \mathbf{z}, \mathbf{I}_R) + U_2(\mathbf{d}_{1R}, \mathbf{d}_{2R}, \mathbf{d}_{1L}, \mathbf{d}_{2L} | \mathbf{w}_1, \mathbf{w}_2, \mathbf{z})$$

3. Update the disparity field, $\mathbf{w}_1$, $\mathbf{w}_2$ assuming that the motion field and the segmentation labels are given. This step involves minimization of

$$E_2 = U_1(\mathbf{I}'_L, \mathbf{I}'_R, \mathbf{I}_L | \mathbf{d}_{1R}, \mathbf{d}_{2R}, \mathbf{d}_{1L}, \mathbf{d}_{2L}, \mathbf{w}_1, \mathbf{w}_2, \mathbf{z}, \mathbf{I}_R) + U_3(\mathbf{w}_1, \mathbf{w}_2 | \mathbf{z})$$

4. Update the segmentation labels z, assuming that disparity field and motion estimates are given, This step involves the minimization of all terms that contain z, given by

$$E_3 = U_1(\mathbf{I}'_L, \mathbf{I}'_R, \mathbf{I}_L | \mathbf{d}_{1R}, \mathbf{d}_{2R}, \mathbf{d}_{1L}, \mathbf{d}_{2L}, \mathbf{w}_1, \mathbf{w}_2, \mathbf{z}, \mathbf{I}_R) + U_2(\mathbf{d}_{1R}, \mathbf{d}_{2R}, \mathbf{d}_{1L}, \mathbf{d}_{2L} | \mathbf{w}_1, \mathbf{w}_2, \mathbf{z}) + U_3(\mathbf{w}_1, \mathbf{w}_2 | \mathbf{z}) + U_4(\mathbf{z})$$

The initial estimates of the optical flow, disparity, and segmentation fields can be obtained from existing still-image disparity estimation and monocular motion estimation and segmentation (see Chapter 11) algorithms, respectively.

12.2 Motion Tracking

Physical objects generally exhibit temporally smooth motion. In motion estimation from two views, whether monocular or stereo, it is not possible to make use of this important cue to resolve ambiguities and/or obtain more robust estimates. For example, acceleration information and the actual location of the center of rotation cannot be determined from two views. To this effect, here we present some recent results on motion estimation (tracking) from long sequences of monocular and stereo video based on an assumed kinematic model of the motion. We first discuss basic principles of motion tracking. Subsequently, some examples of 2-D and 3-D motion models and algorithms to track the parameters of these models are presented.

12.2.1 Basic Principles

Fundamental to motion tracking are a set of feature (token) matches or optical flow estimates over several frames, and a dynamic model describing the evolution of the motion in time. The feature matches or optical flow estimates, which serve as observations for the tracking algorithm, are either assumed available or estimated from pairs of frames by other means. The tracking algorithm in essence determines the parameters of the assumed motion model that best fit (usually in the least squares or minimum mean square error sense) the entire set of observations. In the following, we provide an overview of the main components of a tracking system.

Motion Model

Motion models vary in complexity, ranging from constant velocity models to more sophisticated local constant angular momentum models [Wen 87] depending upon the application. The performance of a tracking algorithm is strongly dependent on the accuracy of the dynamical model it employs. We may classify temporal motion models as 2-D motion models, to represent image-plane trajectory of 3-D points, and 3-D motion models, to represent the kinematics of physical motion.

- 2-D Trajectory Models: Temporal trajectory of pixels in the image plane can be approximated by affine, perspective, or polynomial spatial transformations (see Sections 6.1.2 and 9.1), where the parameters a_i, $i = 1, \ldots, 6$ or 8, become functions of time (or the frame index k). Such trajectory models can be employed to track the motion of individual tokens or group of pixels (regions) (see Section 12.2.2 for examples). The temporal evolution of the transformation parameters can be modeled by either relating them to some 2-D rotation, translation, and/or dilation dynamics, or by a low-order Taylor series expansion of an unknown dynamics. Examples of the

former approach include assuming a purely translational constant acceleration trajectory (see the token-tracking example in Section 12.2.2), and assuming a spatially local simplified affine model given by

$$\begin{aligned} x_1(k+1) &= x_1(k)\cos\alpha(k) - x_2(k)\sin\alpha(k) + t_1(k) \\ x_2(k+1) &= x_1(k)\sin\alpha(k) + x_2(k)\cos\alpha(k) + t_2(k) \end{aligned}$$

with some 2-D rotation (denoted by the angle $\alpha(k)$) and translation (represented by $t_1(k)$ and $t_2(k)$) dynamics. An example of the latter approach (see region tracking) is presented in Section 12.2.2 based on the work of Meyer *et* al. [Mey 94].

. *3-D* Rigid Motion *Models:* There are some important differences between 3-D rigid motion modeling for two-view problems and for tracking problems. The models used in two-view problems, discussed in Chapters 2, 9, and 10, do not include acceleration and precession, since they cannot be estimated from two views. Another difference is in choosing the center of rotation. There are, in general, two alternatives:

i) Rotation is defined with respect to a fixed world coordinate system, where it is assumed that the center of rotation coincides with the origin of the world coordinate system. This approach, which has been adopted in Chapter 2, is generally unsuitable for tracking problems. To see why, suppose we wish to track a rolling wheel. The rotation parameters computed with respect to the origin of the world coordinates are different for each frame (due to the translation of the center of the wheel), although the wheel rotates with a constant angular velocity about its center [Sha 90]. Thus, it is unnecessarily difficult to model the kinematics of the rotation in this case.

ii) Rotation is defined about an axis passing through the actual center of rotation. This approach almost always leads to simpler kinematic models for tracking applications. The coordinates of the center of rotation, which are initially unknown and translate in time, can only be determined after solving the equations of motion. It has been shown that the center of rotation can be uniquely determined if there exists precessional motion [You 90]. In the case of rotation with constant angular velocity, only an axis of rotation can be determined. It follows that one can estimate at best the axis of rotation in two-view problems.

2-D and 3-D motion models can each be further classified as rigid versus deformable motion models. The use of active contour models [Ley 93], such as snakes, and deformable templates [Ker 94] for tracking 2-D deformable motion, and superquadrics [Met 93] for tracking 3-D deformable motion have recently been proposed. Tracking of 2-D and 3-D deformable motion are active research topics of current interest.

Observation Model

All tracking algorithms require the knowledge of a set of 2-D or 3-D feature correspondences or optical flow data as observations. In practice, these feature correspondences or optical flow data are not directly available, and need to be estimated from the observed spatio-temporal image intensity. Feature matching and optical flow estimation are ill-posed problems themselves (see Chapter 5), because of ambiguities such as multiple matches (similar features) within the search area or no matches (occlusion). Several approaches have been proposed for estimating feature correspondences, including multifeature image matching [Wen 93] and probabilistic data association techniques [Cox 93]. These techniques involve a search within a finite window centered about the location predicted by the tracking algorithm. It has been shown that the nearest match (to the center of the window) may not always give the correct correspondence. To this effect, probabilistic criteria have been proposed to determine the most likely correspondence within the search window [Cox 93]. In optical-flow-based tracking, a multiresolution iterative refinement algorithm has been proposed to determine the observations [Mey 94]. In stereo imagery, stereo-motion fusion using dynamic programing [Liu 93] or constrained optimization [Tam 91] have been proposed for more robust correspondence estimation.

Batch vs. Recursive Estimation

Once a dynamical model and a number of feature correspondences over multiple frames have been determined, the best motion parameters consistent with the model and the observations can be computed using either batch or recursive estimation techniques. Batch estimators, such as the nonlinear least squares estimator, process the entire data record at once after all data have been collected. On the other hand, recursive estimators, such as Kalman filters or extended Kalman filters (see Appendix C), process each observation as it becomes available to update the motion parameters. It can be easily shown that both batch and recursive estimators are mathematically equivalent when the observation model is linear in the unknowns (state variables). Relative advantages and disadvantages of batch versus recursive estimators are:

1) Batch estimators tend to be numerically more robust than recursive estimators. Furthermore, when the state-transition or the observation equation is nonlinear, the performance of batch methods is generally superior to that of recursive methods (e.g., extended Kalman filtering).
2) Batch methods would require processing of the entire data record every time a new observation becomes available. Hence, recursive methods are computationally more attractive when estimates are needed in real-time (or "almost" real-time).

To combine the benefits of both methods, that is, computational efficiency and robustness of the results, hybrid methods called recursive-batch estimators have been proposed [Wen 93].

12.2.2 2-D Motion Tracking

Two-dimensional motion tracking refers to the ability to follow a set of tokens or regions over multiple frames based on a polynomial approximation of the image-plane trajectory of individual points or the temporal dynamics of a collection of flow vectors. Two examples are presented in the following.

Token Tracking

Several possibilities exist for 2-D tokens, such as points, line segments, and corners. Here, an example is provided, where 2-D lines, given by

$$x_2 = m\, x_1 + 6 \quad \text{or} \quad x_1 = m\, x_2 + b$$

where m is the slope and b is the intercept, are chosen as tokens. Each such line segment can be represented by a 4-D feature vector $\mathbf{p} = [\mathbf{p}_1\ \mathbf{p}_2]^T$ consisting of the two end points, $\mathbf{p}_1$ and $\mathbf{p}_2$.

Let the 2-D trajectory of each of the endpoints be approximated by a second-order polynomial model, given by

$$\begin{aligned} \mathrm{x(k)} &= \mathbf{x}(k-1) + \mathbf{v}(k-1)\Delta t + \frac{1}{2}\mathbf{a}(k-1)(\Delta t)^2 \\ \mathbf{v(k)} &= \mathbf{a}(k-1)\Delta t \quad \text{and} \quad \mathbf{a}(k) = \mathbf{a}(k-1) \end{aligned} \tag{12.22}$$

where $\mathbf{x}(k)$, $\mathbf{v}(k)$, and **a(h)** denote the position, velocity, and acceleration of the pixel at time k, respectively. Note that (12.22) models an image plane motion with constant acceleration.

Assuming that the tracking will be performed by a Kalman filter, we define the 12-dimensional state of the line segment as

$$\mathbf{z}(k) = \begin{bmatrix} \mathbf{p}(k) \\ \dot{\mathbf{p}}(k) \\ \ddot{\mathbf{p}}(k) \end{bmatrix} \tag{12.23}$$

where $\dot{\mathbf{p}}(k)$ and $\ddot{\mathbf{p}}(k)$ denote the velocity and the acceleration of the coordinates, respectively. Then the state propagation and observation equations can be expressed as

$$\mathrm{z(k)} = \mathbf{\Phi}(k, k-1)\mathbf{z}(k-1) + \mathbf{w}(k), \quad k = 1, .., \mathrm{N} \tag{12.24}$$

where

$$\mathbf{\Phi}(k, k-1) = \begin{bmatrix} \mathbf{I}_4 & \mathbf{I}_4\Delta t & \frac{1}{2}\mathbf{I}_4(\Delta t)^2 \\ \mathbf{0}_4 & \mathbf{I}_4 & \mathbf{I}_4\Delta t \\ \mathbf{0}_4 & \mathbf{0}_4 & \mathbf{I}_4 \end{bmatrix} \tag{12.25}$$

is the state-transition matrix, $\mathbf{I}_4$ and $\mathbf{0}_4$ are 4 x 4 identity and zero matrices; respectively, $\mathbf{w}(k)$ is a zero-mean, white random sequence, with the covariance

matrix $\mathbf{Q}(k)$ representing the state model error, and

$$\mathbf{y(k)} = \mathbf{p}(k) + \mathbf{v}(k), \quad k = 1, .., \mathrm{N} \tag{12.26}$$

respectively. Note that the observation equation (12.26) simply states that the noisy coordinates of the end points of the line segment at each frame can be observed. It is assumed that the observations can be estimated from pairs of frames using some token-matching algorithm. The application of Kalman filtering follows straightforwardly using (C.5)-(C.9) given the state (12.24) and the observation (12.26) models.

Region Tracking

Tracking regions rather than discrete tokens may yield more robust results and provide new means to detect occlusion. Meyer et al. [Mey 94] recently proposed a region-tracking algorithm that is based on motion segmentation and region boundary propagation using affine modeling of a dense flow field within each region. Their method employs two Kalman filters, a motion filter that tracks the affine flow model parameters and a geometric filter that tracks boundaries of the regions. A summary of the state-space formulation for both Kalman filters is provided in the following.

1. *Motion Filter:* Let the affine flow field within each region be expressed as

$$\begin{bmatrix} v_1(k) \\ v_2(k) \end{bmatrix} = \begin{bmatrix} \dot{x}_1(k) \\ \dot{x}_2(k) \end{bmatrix} = \mathbf{A}(k) \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} + \mathrm{b}(k) \tag{12.27}$$

where the matrix **A** and the vector **b** are composed of the six affine parameters a_i, $i = 1, .. 6$.

The temporal dynamics of the affine parameters, a_i, $i = 1, .. 6$, have been modeled by a second-order Taylor series expansion of unknown temporal dynamics, resulting in the state-space model

$$\begin{bmatrix} a_i(k) \\ \dot{a}_i(k) \end{bmatrix} = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} a_i(k-1) \\ \dot{a}_i(k-1) \end{bmatrix} + \begin{bmatrix} \epsilon_{i1}(k) \\ \epsilon_{i2}(k) \end{bmatrix}, \quad i = 1, \ldots, 6 \tag{12.28}$$

where $[\epsilon_{i1}(k)\ \epsilon_{i2}(k)]^T$, $i = 1, , 6$ are uncorrelated, identically distributed sequences of zero-mean Gaussian random vectors with the covariance matrix

$$\mathbf{P} = \sigma_P^2 \begin{bmatrix} \frac{\Delta t^3}{3} & \frac{\Delta t^2}{2} \\ \frac{\Delta t^2}{2} & \Delta t \end{bmatrix}$$

Observe that the state-space model (12.28) models the dynamics of each of the affine parameters separately, assuming that the temporal evolution of the six parameters are independent. Then, each parameter can be tracked by a different Kalman filter with a 2-D state vector.

The measurements for the motion filters, $\tilde{a}_i$, $i = 1, .6$ are modeled as noisy observations of the true parameters, given by

$$\tilde{a}_i(k) = a_i(k) + \beta_i(k) \tag{12.29}$$

where $\beta_i(k)$ is a sequence of zero-mean, white Gaussian random variables with the variance σ_β^2. As mentioned before, the measurements $\tilde{a}_i$, i = 1, .6 are estimated from the observed image sequence, two views at a time. Combining (12.27) with the well-known optical flow equation (5.5), a multiresolution estimation approach has been proposed by Meyer et al. [Mey 94] using least-squares fitting and iterative refinement within each region.

2. ***Geometric Filter:*** The geometric filter is concerned with tracking each region process that is identified by a motion segmentation algorithm [Bou 93]. The convex hull (the smallest convex polygon covering the region) of a polygonal approximation of each region is taken as the region descriptor. The region descriptor vector consists of 2N components, consisting of the (x_1, x_2) coordinates of the N vertices of the convex hull.

   A state-space model for tracking the evolution of the region descriptor vector over multiple frames can be written as

$$\begin{pmatrix} x_{11}(k) \\ x_{12}(k) \\ \vdots \\ x_{N1}(k) \\ x_{N2}(k) \end{pmatrix} = \begin{bmatrix} \mathbf{\Phi}(k,k-1) & 0 & \cdots & 0 \\ \vdots & & & \\ 0 & & 0 & \mathbf{\Phi}(k,k-1) \end{bmatrix} \begin{pmatrix} x_{11}(k-1) \\ x_{12}(k-1) \\ \vdots \\ x_{N1}(k-1) \\ x_{N2}(k-1) \end{pmatrix} + \begin{bmatrix} \mathbf{I}_2 \\ \vdots \\ \mathbf{I}_2 \end{bmatrix} \mathbf{u}(k) + \mathbf{w}(k) \tag{12.30}$$

   where

$$\mathbf{\Phi}(k, k-1) = \mathbf{I}_2 + \Delta t \mathbf{A}(k)$$

   which can be seen by considering a first-order Taylor expansion of $[x_1(k)\ x_2(k)]^T$ given by

$$\begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} = \begin{bmatrix} x_1(k-1) \\ x_2(k-1) \end{bmatrix} + \Delta t \begin{bmatrix} \dot{x}_1(k) \\ \dot{x}_2(k) \end{bmatrix}$$

   about the time k – 1, and substituting the flow field model (12.27) for $[\dot{x}_1(k)\ \dot{x}_2(k)]^{\mathbf{T}}$,

$$u(k) = \text{At b(k)}$$

   is a deterministic input vector which follows from the above derivation, and *w(k)* is a white-noise sequence with the covariance matrix *Q(k).*

The measurement equation can be expressed as

$$\begin{pmatrix} \tilde{x}_{11}(k) \\ \tilde{x}_{12}(k) \\ \\ \tilde{x}_{N1}(k) \\ \tilde{x}_{N2}(k) \end{pmatrix} = \begin{pmatrix} x_{11}(k) \\ x_{12}(k) \\ \\ x_{N1}(k) \\ x_{N2}(k) \end{pmatrix} + \mathbf{n}(k) \tag{12.31}$$

where n(k) is a sequence of zero-mean, white Gaussian random vectors. An observation of the region descriptor vector can be estimated by two-view optical flow analysis as described in [Bou 93].

All motion-tracking algorithms should contain provisions for detecting occlusion and de-occlusion, as moving objects may exit the field of view, or new objects may appear or reappear in the picture. An occlusion and de-occlusion detection algorithm based on the divergence of the motion field has been proposed in [Mey 94]. The main idea of the detector is to monitor the difference in the area of the regions from frame to frame, taking into account any global zooming. The descriptor is updated after each frame using occlusion/de-occlusion information. The region tracking algorithm presented in this section has recently been extended to include active contour models for representation of complex primitives with deformable B-splines [Bas 94].

12.2.3 3-D Rigid Motion Tracking

Three-dimensional motion tracking refers to the ability to predict and monitor 3-D motion and structure of moving objects in a scene from long sequences of monocular or stereo video. In what follows, we present examples of 3-D rigid motion tracking from monocular and stereo video based on the works of Broida and Chellappa [Bro 91] and Young and Chellappa [You 90].

From Monocular Video

Suppose we wish to track the 3-D motion of M feature points on a rigid object. We define two coordinate systems, depicted in Figure 12.4: C,, the object coordinate system whose origin $\mathbf{X}_o = [X_{o1}(k)\ X_{o2}(k)\ X_{o3}(k)]^T$, with respect to the camera and world coordinate systems, coincides with the center of rotation (which is unknown); and $\mathbf{C}_s$, the structure coordinate system, whose origin is located at a known point on the object. It is assumed that $\mathbf{C}_s$ and $\mathbf{C}_o$ are related by an unknown translation $\mathbf{T}$,. Let $\mathbf{X}_i = [\tilde{X_{i1}}\ \tilde{X_{i2}}\ \tilde{X}_{i3}]^T$ denote the known coordinates of a feature point i with respect to the structure coordinate system. Then the coordinates of the feature point with respect to the world and camera coordinate systems are given by

$$Xi(k) = X,(k) + \mathbf{R}(k)(\tilde{\mathbf{X}}_i - \text{T,}), \quad k = 0,.\ .,\ N \tag{12.32}$$

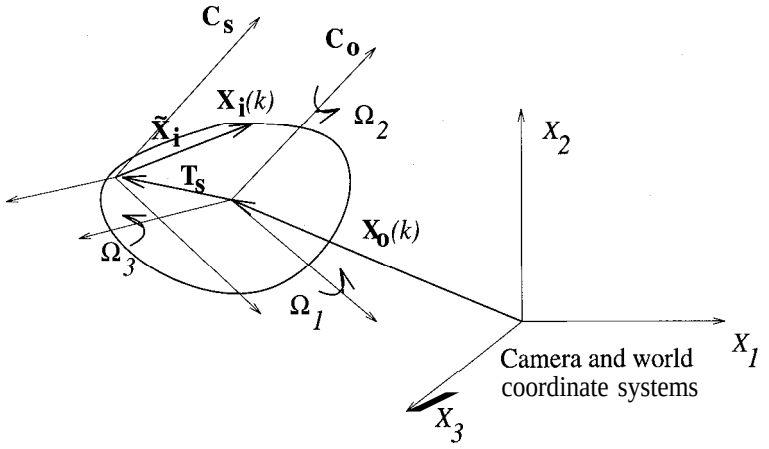


Figure 12.4: Motion tracking with monocular video.

where N is the number of frames, $\mathbf{R}(k)$ is the rotation matrix defined with respect to the center of rotation (object reference frame), R(0) is taken as the 3×3 identity matrix, the origin of the object coordinate system translates in time with respect to the camera and world coordinate systems, and the object is assumed rigid; that is, the coordinates of the feature points $\tilde{\mathbf{X}}_i$ with respect to the object coordinate system remains fixed in time.

The translational component of the motion will be represented by a constant acceleration model given by

$$\text{X,(b)} = \mathbf{X}_o(k-1) + \mathbf{V}_o(k-1)\Delta t + \frac{1}{2}\mathbf{A}_o(k-1)(\Delta t)^2 \tag{12.33}$$

$$\mathbf{V}_o(k) = \mathbf{V}_o(k-1) + \mathbf{A}_o(k-1)\Delta t \tag{12.34}$$

$$\mathbf{A}_o(k) = \mathbf{A}_o(k-1) \tag{12.35}$$

where V,(k) and $\mathbf{A}_o(k)$ denote 3-D translational velocity and acceleration vectors.

The rotational component of the motion can be represented by a constant precession (rate of change of angular velocity) model, where $\mathbf{\Omega}(k) = [\Omega_1(k)\, \Omega_2(k)\, \Omega_3(k)]^T$ and P = $[P_1\ P_2\ P_3]^T$ denote the angular velocity and precession vectors, respectively. Assuming that the rotation matrix R is expressed in terms of the unit quaternion $\mathbf{q}(k) = [q_0(k)\ q_1(k)\ q_2(k)\ q_3(k)]^T$, given by (2.10), the evolution of the rotation matrix in time can be expressed by that of the unit quaternion $\mathbf{q}(k)$. It has been shown that the temporal dynamics of the unit quaternion, in this case, can be expressed in closed form as [You 90]

$$\mathbf{q}(k) = \Phi[\mathbf{\Omega}(k-1), \text{P}; \Delta t]\mathbf{q}(k-1), \quad k = 1,\ldots, \text{N} \tag{12.36}$$

where q(0) = $[0\ 0\ 0\ 1]^T$,

$$\Phi[\mathbf{\Omega}(k-1), \text{P; At}] = \Lambda[\mathbf{P}; \Delta t]\Lambda[\mathbf{\Omega}(k-1) - \text{P; At}] \tag{12.37}$$

$$\Lambda[\mathbf{G}; \text{At}] = \begin{cases} \mathbf{I}_4 \cos\frac{|\mathbf{G}|\Delta t}{2} + \frac{2}{|\mathbf{G}|}\mathbf{\Gamma}[\mathbf{G}] \sin\frac{|\mathbf{G}|\Delta t}{2} & \text{if } \mathbf{G} \neq \mathbf{0} \\ \mathbf{I}_4 & \text{if } \mathbf{G} = \mathbf{0} \end{cases} \tag{12.38}$$

and

$$\mathbf{\Gamma}[\mathbf{G}] = \frac{1}{2}\begin{bmatrix} 0 & -G_3 & G_2 & -G_1 \\ G_3 & 0 & -G_1 & -G_2 \\ -G_2 & G_1 & 0 & -G_3 \\ G_1 & G_2 & G_3 & 0 \end{bmatrix} \tag{12.39}$$

In Equation (12.38), G stands for a 3×1 vector, which takes the values of P or $\mathbf{\Omega}(k-1) - \text{P}$. Observe that this model is also valid for the special case P = 0.

A multiframe batch algorithm, proposed by Broida and Chellappa [Bro 91], when both translational and rotational motion are with constant velocity is briefly summarized below. Assuming that all M feature points have the same 3-D motion parameters, the unknown parameters are

$$\mathbf{u} = \begin{bmatrix} \frac{X_{o1}}{X_{o3}}(0) \\ \frac{X_{o2}}{X_{o3}}(0) \\ \frac{V_1}{X_{o3}}(0) \\ \frac{V_2}{X_{o3}}(0) \\ \frac{V_3}{X_{o3}}(0) \\ \Omega_1(0) \\ \Omega_2(0) \\ \Omega_3(0) \\ \frac{X_{11}}{X_{o3}}(0) \\ \frac{X_{12}}{X_{o3}}(0) \\ \frac{X_{21}}{X_{o3}}(0) \\ \frac{X_{22}}{X_{o3}}(0) \\ \frac{X_{23}}{X_{o3}}(0) \\ \\ \frac{X_{M1}}{X_{o3}}(0) \\ \frac{X_{M2}}{X_{o3}}(0) \\ \frac{X_{M3}}{X_{o3}}(0) \end{bmatrix} \tag{12.40}$$

Observe that all position and translational motion parameters are scaled by X_{o3}, which is equivalent to setting $X_{o3} = 1$ to account for the scale ambiguity inherent in monocular imaging. Furthermore, the X_3 component of the first feature point is left as a free variable, because the origin of the object-coordinate frame (center of

rotation) cannot be uniquely determined (only the axis of rotation can be computed) in the absence of precessional motion.

Suppose we have point correspondences measured for these M points over N frames, given by

$$x_{i1}(k) = \frac{X_{i1}(k)}{X_{i3}(k)} + n_{i1}(k) \quad (12.41)$$

$$x_{i2}(k) = \frac{X_{i2}(k)}{X_{i3}(k)} + n_{i2}(k) \quad (12.42)$$

where $X_{i1}(k)$, $X_{i2}(k)$, and $X_{i3}(k)$ can be computed in terms of the unknown parameters u using (12.32), (12.35), (12.36), and (2.10). Assuming the noise terms $n_{i1}(k)$ and $n_{i2}(k)$ are uncorrelated, identically distributed zero-mean Gaussian random variables, the maximum-likelihood estimate of the parameter vector u can be computed by minimizing the summed square residuals

$$E(\mathbf{u}) = \sum_{k=1}^{N}\sum_{i=1}^{M}[x_{i1}(k) - \frac{X_{i1}(k)}{X_{i3}(k)}]^2 + [x_{i2}(k) - \frac{X_{i2}(k)}{X_{i3}(k)}]^2 \quad (12.43)$$

The minimization can be performed by the conjugate-gradient descent or Levenberg-Marquardt methods. Alternatively, defining the value of the parameter vector $\mathbf{u}$ at each time k, we can define a recursive estimator. However, because of the nonlinearity of the rotational motion model and the observation equation in the unknowns, this would result in an extended Kalman filter.

From Stereo Video

In stereo imaging, the camera and the world coordinate systems no longer coincide, as depicted in Figure 12.5. Hence, it is assumed that feature correspondences both between the right and left images and in the temporal direction are available at each time k as observations. They can be obtained either separately or simultaneously using stereo-motion fusion techniques with two stereo-pairs [Tam 91, Liu 93].

Given these feature correspondence data and the kinematics of the 3-D motion modeled by (12.32), (12.35), (12.36), and (2.10), batch or recursive estimation algorithms can be derived using either 2-D or 3-D measurement equations [You 90]. In the case of 2-D measurements, the corresponding points in the right and left images are separately expressed in terms of the state vector at each time and included in the measurement vector. In the case of 3-D measurements, stereo triangularization is employed at each time k to reconstruct a time sequence of 3-D feature matches, which are then expressed in terms of the state vector. A comparison of both recursive and batch estimators has been reported in [Wu 94]. It has been concluded that, in the case of recursive estimation, experiments with 2-D measurements have reached steady-state faster. More recently, simultaneous stereo-motion fusion and 3-D motion tracking have been proposed [Alt 95].

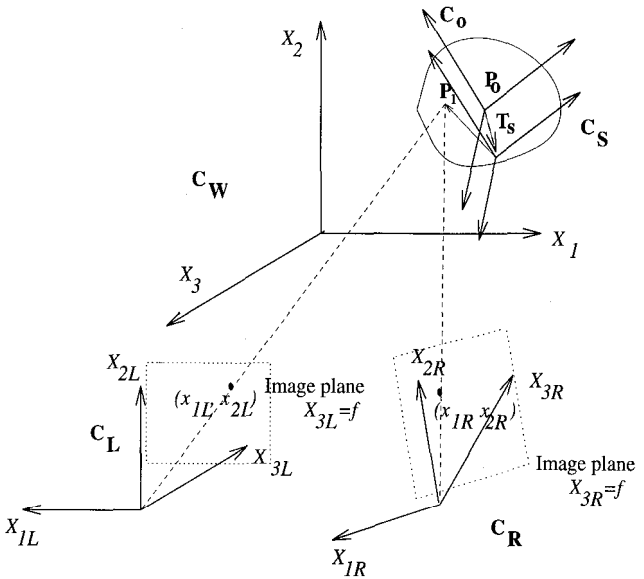


Figure 12.5: Motion tracking with stereo video.

12.3 Examples

We have experimented with some of the algorithms presented in this section using a stereo image sequence, known as a static indoor scene, produced at the Computer Vision Laboratory of Ecole Polytechnique de Montreal. The sequence consists of 10 frame pairs, where each left and right image is 512 pixels x 480 lines with 8 bits/pixel. The calibration parameters of the two cameras were known [Wen 92]. The first and second frame pairs of the sequence are shown in Figure 12.6.

We have interactively marked 14 feature points on the right image of the first frame (at time t_1), which are depicted by white circles in Figure 12.6 (b). The corresponding points on the other three images are computed by the maximum likelihood stereo-motion fusion algorithm described in Section 12.1.3. The initial estimates of the displacement $\mathbf{d}_{1R}$, $\mathbf{d}_{2R}$, $\mathbf{d}_{1L}$, $\mathbf{d}_{2L}$ and the disparity $\mathbf{w}_1$, $\mathbf{w}_2$ are computed by three-level hierarchical block matching. Next, we have tracked the motion of one of these feature points over the 10 frames using a batch algorithm. Observe that the selected feature point leaves the field of view after nine frames on the right image and eight frames on the left, as shown in Figure 12.7. We note that tracking using a temporal trajectory not only improves 3-D motion estimation, but also disparity and hence depth estimation.

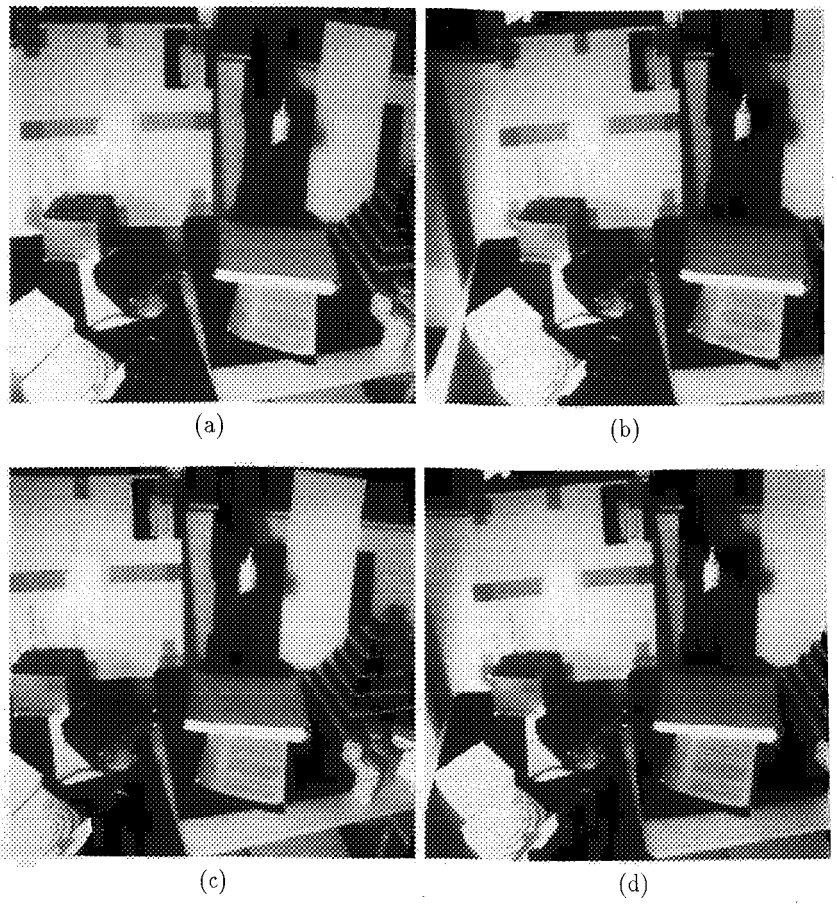


Figure 12.6: The first and second frame pairs of a stereo indoor scene produced at the Computer Vision Laboratory of Ecole Polytechnique de Montreal: a) left image at time t_1, b) right image at time t_1, c) left image at time t_2, and d) right image at time t_2. Fourteen feature points (marked by white circles) have been selected interactively on the right image at time t_1. Point correspondences on the other images are found by the maximum likelihood stereo-motion fusion algorithm. (Courtesy Yucel Altunbasak)

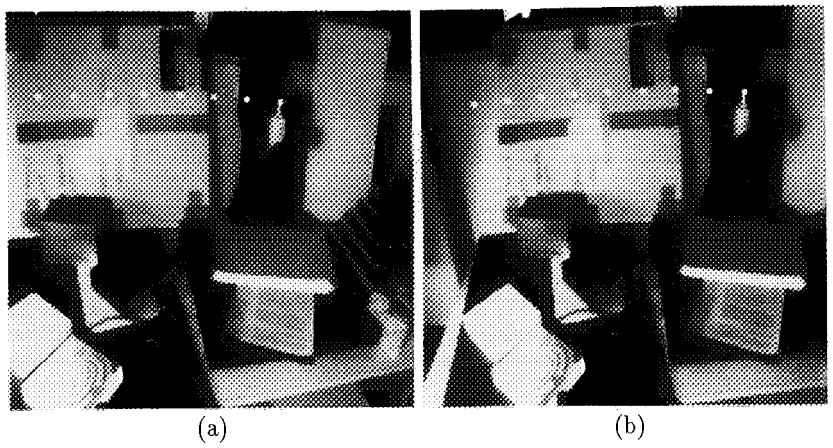


Figure 12.7: Temporal trajectory of a single point marked on the a) left and b) right images of the first frame. The white marks indicate the position of the feature point in the successive frames. (Courtesy Yucel Altunbasak)

12.4 Exercises

1. Show that only two of the three equations in (12.5) are linearly independent if we have exact left-right correspondences.

2. Derive (12.6) given that the left and right camera coordinates are parallel and aligned with the $X_1 - X_2$ plane of the world coordinate system.

3. Derive (12.10).

4. Write the Kalman filter equations for the 2-D token-matching problem discussed in Section 12.2.2 given the state-transition model (12.24) and the observation model (12.26).

5. Write the equations of a recursive estimator for the 3-D motion-tracking problem discussed in Section 12.2.3. Discuss the relationship between this recursive estimator and the batch solution that was provided.