ESTIMATION OF IMAGE MOTION IN SCENES CONTAINING MULTIPLE MOVING OBJECTS

Heyun Zheng

A thesis submitted to the Department of Electrical
Engineering in conformity with the requirements
for the degree of Doctor of Philosophy

Queen's University
Kingston, Ontario, Canada
April 1995

Acknowledgements

I wish to express my sincere thanks to Dr. S.D. Blostein for his guidance, support and patience throughout this work, and Dr. D.F. Fleet for inspiring discussions.

This research was partly supported by the Canadian Institute of Robotics and Intelligent Systems and by the Natural Sciences and Engineering Research Council of Canada (NSERC) Grant OGP0041731. Other financial assistance came from the School of Graduate Studies and Research at Queen's University and the Government of the People's Republic of China.

Abstract

This thesis is concerned primarily with the development of algorithms for estimating and segmenting image motion fields that contain discontinuities. An error-weighted regularization algorithm for image motion field estimation is proposed as a computationally attractive alternative to stochastic optimization based schemes. Block matching errors in the local motion measurement process are used in the regularization functional in order to avoid oversmoothing across motion boundaries. A second algorithm, anisotropic regularization, improves on the local measurement process, by employing alternative matching criteria and matching window organization. A selective confidence measure derived from anisotropic local measurements is used to further improve the error-weighted regularization.

For moving object estimation and segmentation needed in object-oriented video coding applications, a new optimality criterion based on the minimum description length (MDL) principle is developed. In the proposed MDL estimator, the cost to be minimized is the sum of the ideal coding lengths for the motion parameters, boundaries and motion-compensated predictive errors of all moving objects in a scene. An optimization procedure to obtain a sub-optimal MDL estimator is proposed based on a region-merging framework. A number of experimental comparisons has shown a significant ideal coding rate reduction of the object-oriented coding scheme using an MDL estimator over a standard block-oriented scheme.

Contents

Acknowledgements i

Abstract ii

List of Tables iv

List of Figures v

1 INTRODUCTION 1

1.1 MOTIVATION 1

1.1.1 Existing approaches and their drawbacks 2

1.1.2 The application-oriented viewpoint 3

1.2 Thesis outline 4

2 Image motion field estimation 7

2.1 Image motion field 7

2.2 Local image-motion field measurements 9

2.2.1 Differential approaches 9

2.2.2 Matching approaches 12

2.2.3 Spatiotemporal filtering 14

2.3 Image motion estimation by regularization 15

2.4 Regularization that includes discontinuities 18

3 ERROR-WEIGHTED REGULARIZATION **21**

3.1 Standard regularization and its drawbacks 22

3.2 Motivation for using block-matching 25

3.3 Properties of block-matching errors 27

3.4 Motion boundary types 29

3.5 Confidence measures for local motion 30

3.6 Error-weighted regularization 33

3.7 Experimental results 36

3.7.1 Sinusoidal input with pure translation 37

3.7.2 Rotating and zooming disc 39

3.7.3 Real images 40

4 ANISOTROPIC REGULARIZATION **47**

4.1 Piece-wise continuous functions 48

4.2 Anisotropic regularization 49

4.2.1 One dimensional regularization 49

4.2.2 Isotropic regularization functional 51

4.2.3 Anisotropic regularization functional 52

4.2.4 Two-dimensional anisotropic regularization 53

4.3 Constructing selective confidence measure 55

4.3.1 Motivating the SAD criterion 55

4.3.2 Multiple off-centered sub-window (MOW) matching 56

4.3.3 The selective confidence measure 58

4.4 Experimental results 59

4.4.1 Comparison between multiple and single window matching . . 59

4.4.2 Anisotropic smoothing experiments 63

4.4.3 Comparison between anisotropic and error-weighted regularizations 68

5 OBJECT-ORIENTED CODING AND MDL PRINCIPLE 72

5.1 Motivating the use of the MDL principle 73

5.2 Block- and object-oriented image coding 74

5.2.1 Block-oriented coder 76

5.2.2 Object-oriented coder 77

5.2.3 Moving object estimation 78

5.3 MDL principle 80

5.3.1 Relation between estimation and coding 81

5.3.2 Prior information and parameter coding 84

5.3.3 Data model structure 85

5.3.4 Relationship between regularization and MDL 87

5.3.5 Image segmentation by MDL 88

5.3.6 Summary of the advantages of MDL 89

6 OBJECT SEGMENTATION AND ESTIMATION BY MDL PRINCIPLE 91

6.1 Problem formulation 92

6.1.1 System modeling for single moving objects 92

6.1.2 System modeling for multiple objects 94

6.1.3 Modeling of image motion parameters and its coding length . 96

6.1.4 Object boundary modeling and its coding length 99

6.1.5 Prediction error modeling and its coding length 100

6.1.6 Total ideal coding length function for one frame 101

6.2 The minimization procedure 102

6.2.1 Initial segmentation and object merging scheme 103

6.2.2 Estimating the motion parameters 106

6.3 Experimental results 110

6.3.1 Experiment on synthetic images 111

6.3.2 Experiment on the "Femme et arbre" images 117
6.3.3 Experiment on the "Chair" sequence 118
6.4 Implementation issues 120
6.4.1 Computational complexity of the procedure 120
6.4.2 Efforts in reducing the computational cost of the procedure . . 124

7 SUMMARY AND CONCLUSION **127**

7.1 Image field estimation by regularization 127
7.1.1 Major results 127
7.1.2 Future research in image motion estimation 129
7.2 Moving object segmentation and estimation by MDL 130
7.2.1 Major results 130
7.2.2 Future research in the MDL estimator 132

Bibliography **133**

List of Tables

3.1 Motion field estimation for "Square" images 39

3.2 Motion field estimation for "Disc" images 40

3.3 Motion-compensated interpolation: MSE/pel for chair sequence . . . 45

4.1 SCW scheme: comparison between SSD and SAD for "Square". Upper: the left upper diagonal part of the images. Lower: the right lower diagonal part of the images. 61

4.2 MOW scheme: comparison between SSD and SAD for "Square". Upper: the left upper diagonal part of the images. Lower: the right lower diagonal part of the images. 62

4.3 SAD scheme: comparison between MOW and SCW schemes for "Square". Upper: the left upper diagonal part of the images. Lower: the right lower diagonal part of the images. 62

4.4 Comparison among SAD, SSD, MOW and SCW schemes for "Disc" . 63

4.5 SNR values of different motion fields for 1-d images 67

4.6 Comparison by SNR values for "Square" images 67

4.7 Comparison by SNR values for "Disc" images 68

4.8 Comparison between anisotropic and error-weighted regularization . . 70

6.1 Results for "Square" images 111

6.2 Results for "Disc" images 117

6.3 Results for "Femme et arbre" images 120

List of Figures

3.1 Translating square over textured and uniform background. (a) Local measurement. (b) matching error image. (c) Error-weighted regularization. (d) Distance-weighted regularization. 38

3.2 Rotating and zooming disc over translating background. (a) Local measurement. (b)matching error image. (c) Error-weighted regularization. (d) Distance-weighted regularization. 41

3.3 (a) Original Field 3 of "Femme et arbre" sequence. (b) Estimated motion field between Fields 1 and 5. (c) Recovered field 3 of "Femme et arbre"sequence. (d) Interpolation error image for Field 3. 43

3.4 (a) Original Field 55 of the chair sequence. (b) Interpolation error image for Field 55. 44

3.5 MSE/pel versus the field number for recovered "Chair" sequence . . . 46

4.1 An example of piece-wise continuous functions 49

4.2 Local motion field by SCW with SSD scheme 60

4.3 Local motion field by SCW with SAD scheme 60

4.4 Local motion field by MOW with SSD scheme 60

4.5 Local motion field by MOW with SAD scheme 60

4.6 Local motion field by SCW with SSD scheme 64

4.7 Local motion field by SCW with SAD scheme 64

4.8 Local motion field by MOW with SSD scheme 64

4.9 Local motion field by MOW with SAD scheme 64
4.10 A one dimensional test image pair 66
4.11 The experimental results for 1-d image pair 66
4.12 Smoothed motion field by standard regularization 69
4.13 Smoothed motion field by anisotropic regularization 69
4.14 Smoothed motion field by standard regularization 69
4.15 Smoothed motion field by anisotropic regularization 69

5.1 Motion-compensated predictive coder 75

6.1 A source coding model for the case of a single moving object 93
6.2 A example of initial adjacent graph 104
6.3 The adjacency graph in Figure 6.2 after merging two nodes 105
6.4 First frame of “Square” images 112
6.5 Initial segmentation of “Square” images 112
6.6 Final segmentation of “Square” images 112
6.7 flow field described by motion parameters 112
6.8 Coding length decrease during the merging process for ”Square” images 114
6.9 First frame of “Disc” images 115
6.10 Final segmentation by 8x8 block size 115
6.11 Final segmentation by 4x4 block size 115
6.12 Final segmentation by 2x2 block size 115
6.13 True motion field of “Disc” images 116
6.14 Motion field estimated by 8x8 block size 116
6.15 Motion field estimated by 4x4 block size 116
6.16 Motion field estimated by 2x2 block size 116
6.17 Coding length decrease during the merging process for ”Disc” images 118
6.18 First frame of “Femme et arbre” images 119
6.19 Initial segmentation of “Femme et arbre” images 119

6.20 Final segmentation of “Femme et arbre” images 119
6.21 Motion field described by motion parameters 119
6.22 Coding length decrease during the merging process for ”Femme et arbre” images 121
6.23 Field 54 of “Chair” sequence 122
6.24 Initial segmentation of Field 54 122
6.25 Final segmentation of Field 54 122
6.26 Motion field described by motion parameters 122
6.27 Ideal coding length comparison for ”Chair” sequence 123
6.28 Coding length decrease during the merging process for ”Disc” images. Dashed line: varying ε. Solid line: fixed ε. 126

NOTATION

A	2x2 parametric matrix of the affine motion model
$B(n)$	total number of boundary elements in $b(n)$
$\mathbf{b}(n)$	the set of boundaries of the nth object
c_{max}, c_{min}	confidence measures for local motion vector
D	translation vector in the affine motion model
$\mathbf{d}$	local motion vector
d_{max}	maximum displacement component in block-matching
$E^{\tau}(x,y)$	image intensity at position (x,y) and time τ
E_x, E_y, E_t	spatial and temporal image intensity derivatives
e^{τ}	motion-compensated prediction errors at time τ
$f(\cdot)$	image intensity mapping function
$L(\cdot)$	ideal coding length function
l_{ij}	the arc connecting node i to node j
$(-M_i, +M_i)$	range of uniform distributions for parameter θ_i
$\mathbf{m}(x,y)$	the partial derivative vector of pixel (x,y)
m	dimension of motion parameter vector θ
N	the number of objects
$P(\cdot)$	the probability mass function
p_l	relative frequencies of directions in chain coding
O_n	the nth object
$(\mathbf{O}, \mathbf{L})$	an adjacency graph
$S(x,y)$	neighborhood of the pixel at (x,y)
$\mathbf{x}$	spatial position for image pixel, image motion vector $\mathbf{x} = (x,y)$
λ	regularization constant
r_{ij}	coding length reduction by merging objects O_i and O_j
θ	motion parameter vector

σ_i^2	variances of θ_i
(V^X, V^Y, V^Z)	translational velocities of the object
$(\Omega^X, \Omega^Y, \Omega^Z)$	3-d rotational velocities of the object
(X, Y, Z)	camera centered Cartesian coordinate for the 3-D world
$bfu, (u, v)$	image motion vector
$\bar{\mathbf{u}}$	weighted local motion average
Φ	regularization functional
Δ	data fitting term
δ	the quantization constant for motion parameters
$\Delta E(x, y)$	motion-compensated prediction error
ε	minimum matching error and the quantization constant for the prediction errors
ε'	scaled minimum matching error
ρ^2	the variance of motion-compensated prediction error distribution
ξ	selective confidence measure

Chapter 1

INTRODUCTION

1.1 MOTIVATION

Moving objects in a scene cause temporally varying image intensity patterns in the image plane of a viewing camera. The analysis of motion in image sequences obtained from such a viewing camera has long been an important component of computer vision and image processing [2, 39]. Accurate image motion estimation or moving object estimation is crucial in many applications of image sequence processing. Applications such as object tracking in surveillance or navigation usually begins with the estimation of image-motion fields and follows by the detection of critical regions of a scene. Efficient image coding and motion-compensated temporal filtering can be realized by using image motion estimation [13, 43, 46, 61]. Image motion is also needed for frame-frequency conversion of television signals and temporal interpolation of missing or unknown images in the sequence [5, 47, 86]. Recently, object-oriented coding has promised a further coding rate reduction over traditional block-oriented motion compensated coding schemes and thus has drawn wide attention in the field of image compression [37, 50, 62, 60, 81].

This thesis addresses the problems of moving object estimation for image motion fields that may contain discontinuities. Motion discontinuities which occur in real

images are major error sources in image motion estimation. The quality of recovered images from motion-compensated temporal interpolation is directly related to the accuracy of image motion estimation. Moving object estimation and segmentation is a another way to deal with motion discontinuities since the task of moving object estimation is essentially the problem of motion-based scene segmentation and the motion boundaries can be explicitly estimated. The output of moving object estimation also provides necessary information for object-oriented coding schemes.

1.1.1 Existing approaches and their drawbacks

The estimation of image-motion fields poses significant theoretical and practical problems [2, 6, 19, 47]. One of these problems is the random sensor noise present in the real images. The local motion measurements are often corrupted by this random noise. However, even in the absence of noise, the image-motion fields cannot be uniquely determined by the local measurement process and the computation may turn out to be very ill-conditioned. This nonuniqueness is often referred to as the aperture problem, where only the velocity component parallel to the spatial image intensity gradients can be locally recovered. A solution to the aperture problem is to use second-order derivatives of pixel intensities, but the set of points where second-order derivatives can be reliably computed locally is sparse for most real images. This sparsity of motion measurements results in a smoothing or interpolation problem in the computation of a dense image-motion field. In order to compute image motion uniquely, constraints on the solution space are therefore required in the form of additional assumptions, such as smoothness of image-motion, or constancy about the physical world. Such constraints may be introduced using a regularization theory framework [3, 36, 58, 74, 75]. One drawback of regularization is its lack of consideration of the presence of motion discontinuities due to multiple moving objects in the image sequences. The standard regularization formulations often smooth over motion

boundaries and result in degraded image motion estimates [36].

In principle, to prevent smoothing over motion discontinuities, motion boundaries may be detected prior to motion-field estimation. However, motion boundary detection depends on a prior motion field estimate, and therefore requires a more sophisticated and computationally complex approach [77, 84].

Motivated by Geman and Geman's "line process" modeling technique [22], Konrad [47] proposes a displacement field model with discontinuities to avoid oversmoothing problem in a Bayesian estimation framework. The model combines the displacement field with a binary (on/off) stochastic line process. When a line element between adjacent pixels is turned on, the displacement vectors across the line element will not be smoothed. Conversely, when the displacement vectors across a line element site differ considerably, the line element at this site is turned on. A similar line process formulation is reported in [31, 32] and a deterministic version of a line process has also been proposed [41]. The problem with these algorithms is the high computational cost involved in the optimization process even if deterministic solution methods are used. Also, line process parameter estimation is data-dependent and difficult to justify in general.

1.1.2 The application-oriented viewpoint

The performance of an image-motion estimation technique should depend on the nature of the subsequent processes that will interpret or use the estimates. At present, most image-motion estimation algorithms are designed separately from the subsequent processes. The optimality criteria of the solution do not relate to later processes. For example, the mean-squared error criterion used in most regularization functionals is intuitively appealing but is hard to justify when image motion estimates are used for motion compensated image coders, since subjective qualities of the recovered images to humans are not measured by mean-squared error [50].

When the subsequent process is object-oriented image coding, a natural concern of moving object estimation is how efficient the encoding results will be in terms of either coding length to a given distortion or image distortion with given coding length. If the moving object estimator is designed only to minimize prediction errors, the coding cost for prediction errors may be small but the cost of encoding the moving object parameters may be high and an overall inefficient coding of the images might result. This thesis proposes to use the minimum description length (MDL) principle [69] as a new framework in which a combined moving object estimation and segmentation problem for image sequence coding is solved. The criterion established by the MDL principle serves the very purpose of image coding process. Specifically, the MDL principle has been applied in this thesis to object-oriented motion-compensated predictive image coding.

1.2 Thesis outline

This thesis addresses the problems of moving object estimation and segmentation for image motion fields that may contain discontinuities. An adaptive regularization technique which utilizes information provided in the local motion measurement process and the application of the MDL principle to the moving object estimation and segmentation problem are the two major contributions of this work.

In Chapter 2, relevant background on image motion field estimation is first presented, followed by a survey of existing techniques for measuring local image motion including differential, matching, and spatiotemporal filtering approaches. Then, regularization theory and its application to image-motion field estimation, specifically, regularization for discontinuous motion fields, are discussed.

In Chapter 3, an error-weighted regularization algorithm for image-motion field estimation from image sequences is presented. First, we discuss the essential features and drawbacks of a regularization framework for the problem of image-motion field

estimation. Then some arguments are given as to why a block-matching technique is chosen as the local measurement process. Following this is an examination of the properties of block-matching errors, motion boundary types and derived confidence measures associated with local motion vectors. The error-weighted regularization algorithm is then described in detail. The chapter concludes with experimental results which compare performance between the new algorithm and conventional regularization schemes.

Chapter 4 presents an alternative to the error-weighted regularization scheme presented in Chapter 3 based on a straight forward improvement to the local measurement process. First, a brief description of piece-wise continuous functions is presented. The concept of anisotropic regularization is then introduced with a detailed one-dimensional example. A selective confidence measure is introduced based on multiple off-centered sub-window matching scheme (MOW) and the sum of absolute difference (SAD) criterion. Finally, experimental results are provided which demonstrate the improved performance over existing schemes, as well as a comparison to error-weighted regularization.

Chapter 5 begins by motivating the use of the minimum length description (MDL) principle in moving object segmentation and estimation. An introductory discussion of *motion-compensated predictive coding* then follows, including brief overviews of block-oriented and object-oriented coders. Existing approaches to object-oriented coding and, more generally, moving object estimation and segmentation are reviewed. Section 5.3 describes the MDL principle in general. A philosophical comparison between regularization and the MDL estimation is included together with a summary of the advantages of the MDL principle for moving object segmentation and estimation problems. Existing applications of the MDL principle to the problem of single image segmentation based on intensity information are also discussed.

In Chapter 6, we first formulate the moving object estimation problem using the

MDL principle for scenes with single and multiple moving objects. The ideal coding length functions for motion parameters, object boundaries and motion-compensated predictive errors are derived. An optimization procedure to obtain a sub-optimal MDL estimator is then proposed based on a region-merging framework. The issue of motion parameter estimation within fixed object boundaries is also discussed. Section 6.3 describes experimental comparisons of the block-oriented and object oriented coding schemes, and assesses the coding rate reduction of the object-oriented coding scheme over a block-oriented scheme. The chapter ends with a discussion of the important implementation issues required for a practical MDL estimator.

Finally, Chapter 7 gives a summary of the results presented in this thesis. The contributions of the thesis and some open problems for future investigation are also included.

Chapter 2

Image motion field estimation

In this chapter, relevant background on image motion field is first presented, followed by a survey of existing techniques for measuring local image motion vectors which include differential approaches, matching approaches, and spatiotemporal filtering approaches. Then, regularization theory and its application to image-motion field estimation, including regularization with discontinuities, are discussed. The material presented in this chapter serves as background mainly for Chapters 3 and 4. The relevant background to moving object estimation and segmentation is deferred to Chapter 5.

2.1 Image motion field

When objects move in front of a camera, or when a camera moves through a fixed environment, there are corresponding changes in the image intensity pattern formed in the image plane of the viewing camera. These changes can be used to recover the relative motion between objects and camera as well as the shapes of the objects. As a first step towards these goals, the image-motion fields, which represents the perspective projection onto the image plane of the true 3-D velocity field of moving objects, are usually first estimated.

To define image-motion field explicitly, a camera centered Cartesian coordinate

system $(X, Y, Z)$for the 3-D real world is assumed in what follows. The Z axis is directed along the viewing direction. The image plane of the camera is normal to the Z axis at unit distance from the origin (here unit focal length is assumed). Then the image coordinate system (x,y) has its origin at (0,0,1). The x and y axes are parallel to the X and Y axes, respectively. In the perspective projection geometry, the image of a point (X,Y,Z) is formed by drawing a line from it to (0,0,0) which intersects the image plane at (x,y), therefore

$$x = X/Z \qquad y = Y/Z. \tag{2.1.1}$$

At a particular instant in time, suppose that a point p in the image corresponds to some point P on the surface of an object. The two are related by the projection equation (2.1.1) in the case of perspective projection. If a set of object points is moving in the scene, every corresponding image point will move according to the projection equation. In this way, a vector can be assigned to every image point. These vectors constitute the *image-motion field* of the moving object.

As an example of a closed-form expression for the image-motion field in the case where the object is undergoing a rigid body motion with a rotational velocity $\mathbf{\Omega} = (\Omega^X, \Omega^Y, \Omega^Z)^T$ and a translational velocity $\mathbf{V} = (V^X, V^Y, V^Z)^T$, the instantaneous velocity $(\dot{X}, \dot{Y}, \dot{Z})^T$ of a point $P = (X, Y, Z)^T$ is given by [2]

$$\begin{aligned} \dot{X} &= -V^X - \Omega^Y Z + \Omega^Z Y \\ \dot{Y} &= -V^Y - \Omega^Z X + \Omega^X Z \\ \dot{Z} &= -V^Z - \Omega^X Y + \Omega^Y X. \end{aligned} \tag{2.1.2}$$

The corresponding point p in the image plane then moves with a velocity $\mathbf{u} = (u, v)$ given by

$$\begin{aligned} u &= (x\tfrac{V^Z}{Z} - \tfrac{V^X}{Z}) + (xy\Omega^X - (1 - x^2)\Omega^Y + y\Omega^Z \\ v &= (y\tfrac{V^Z}{Z} - \tfrac{V^Y}{Z}) + ((1 + y^2)\Omega^X - xy\Omega^Y - x\Omega^Z. \end{aligned} \tag{2.1.3}$$

It is clear from (2.1.3) that the image motion vector $\mathbf{u}$ depends on the surface depth Z of the moving object. When moving objects are far enough from the camera or

the surface is smooth, the induced motion field in the image plane will be continuous. This property of image motion fields is exploited as an additional constraint on image motion estimators. But when the scene consists of multiple moving objects with differing velocities and positions, discontinuities will occur in the image motion field along the object boundaries.

Another term *displacement field* is more often used in the context of video sequence processing. The displacement field describes the displacement of each image pixel between two successive frames. The image-motion field and displacement field are equal if one assumes constant image-motion velocity and a unit temporal interval between frames.

2.2 Local image-motion field measurements

An important problem in image-motion field estimation concerns the type of local motion measurements that should be used. The existing techniques for measuring local image motion vectors roughly fall into three categories:

1. differential approaches,

2. matching, or area correlation approaches and

3. spatiotemporal filtering approaches.

These three approaches will be reviewed in the following subsections.

2.2.1 Differential approaches

Differential techniques typically are based on the assumption that the inter-frame motion is small and the intensity function is smooth and well behaved. These techniques rely on the following relationship between the spatial and temporal image intensity gradients:

Let $E(x, y, t)$ represent the intensity at time t at the pixel (x, y). Let $u(x, y)$ and $v(x, y)$ denote the x and y components, respectively, of the image motion vector at pixel (x, y). By assuming that the spatial structure of image intensity is constant during the time interval $(t, t + \delta t)$, we have [2]

$$E(x + u\delta t, y + v\delta t, t + \delta t) = E(x, y, t) \tag{2.2.1}$$

where $\delta x = u\delta t$, $\delta y = v\delta t$, and δt represents the length of a small time interval. If intensity function $E(x, y, t)$ varies smoothly within a small neighborhood of (x, y, t), we can expand the left-hand side of (2.2.1) into a Taylor series, and (2.2.1) thus becomes

$$E(x, y, t) + \delta x \frac{\partial E}{\partial x} + \delta y \frac{\partial E}{\partial y} + \delta t \frac{\partial E}{\partial t} + e = E(x, y, t) \tag{2.2.2}$$

where e contains second- and higher-order terms in $\delta x, \delta y$, and δt. Canceling $E(x, y, t)$ from both sides of Equation (2.2.2), dividing through by δt, and taking the limit as $\delta t \to 0$, we obtain

$$\frac{\partial E}{\partial x}\frac{dx}{dt} + \frac{\partial E}{\partial y}\frac{dy}{dt} + \frac{\partial E}{\partial t} = 0, \tag{2.2.3}$$

which is actually just the expansion of the equation

$$\frac{dE}{dt} = 0 \tag{2.2.4}$$

in the total derivative of E with respect to time. Using the abbreviations

$$u = \frac{dx}{dt}, \qquad v = \frac{dy}{dt}, \tag{2.2.5}$$

$$E_x = \frac{\partial E}{\partial x}, \quad E_y = \frac{\partial E}{\partial y}, \quad E_t = \frac{\partial E}{\partial t}, \tag{2.2.6}$$

we obtain

$$E_x u + E_y v + E_t = 0. \tag{2.2.7}$$

The derivatives E_x, E_y, and E_t are estimated from the image data. Equation (2.2.7) is often called the image-motion field constraint equation, since it expresses one constraint on the components u and v of the image-motion field.

Equation (2.2.7) alone is insufficient to determine the two components of the vector (u, v). Haralick and Lee [27] use the image-motion field constraint equation in conjunction with the requirement that the first derivatives of image intensity pattern that have been displaced in the image due to object motion must remain the same. This yields a system of four equations in two unknowns:

$$\begin{aligned} E_x u + E_y v + E_t &= 0 \\ E_{xx} u + E_{xy} v + E_{xt} &= 0 \\ E_{yx} u + E_{yy} v + E_{yt} &= 0 \\ E_{tx} u + E_{ty} v + E_{tt} &= 0. \end{aligned} \tag{2.2.8}$$

The above formulation is usually referred to as the multi-constraint method. Algorithms based on (2.2.8) in general have not yielded promising results because high order derivatives of the brightness function are difficult to compute accurately. However Mitiche *et al* [55] and Barron *et al* [4] have reported some success in the computation of image-motion fields using the multi-constraint method.

By noticing that the image-motion field constraint equation is a first-order approximation of Equation (2.2.1) which is poor along intensity edges and corners, Snyder *et al* [76] instead use first- and second-order derivatives of the Taylor series expansion of (2.2.1) to obtain a single nonlinear equation in the two unknowns u, v. Nagel [58, 59] has posed specific conditions on local intensity distributions and presented a corner detector that detects locations in the image that satisfy these conditions. Using second-order terms, Nagel also obtains a closed-form solution for the image motion vectors at the image locations detected by the corner-detector.

A major problem faced by differential approaches is the difficulties in computing derivatives of images. Image pre-filtering can alleviate these difficulties to a certain extent, but new problems may be created in localization and discontinuity identification. Also, Since the inter-frame image motion is restricted to be small, the velocities computed can be easily overridden by pixel-level perturbations.

2.2.2 Matching approaches

Matching, or area correlation techniques have been designed to solve the short- and long-range feature correspondence problem, which associates certain intensity patterns in one image with corresponding intensity patterns in subsequent images. Various intensity patterns , image-dependent or image-independent, can be used. First, these patterns are identified in a reference image. Then, an organized search for the corresponding patterns is performed in the following image.

A simple intensity pattern which can be used to solve the correspondence problem is a fixed size block of pixels. The basic assumption inherent in block matching algorithm is that motion is locally translational and slowly varying within the block. The motion vectors are either computed over a dense grid or are assumed constant over a block if only a single vector is estimated for each block. Then an optimization problem based on some objective criteria, is formulated in the form of

$$Min_{\{u,v\}} I(x,y;u,v) = \sum_{i,j} \phi(E^{\tau}(x+i,y+j) - E^{\tau+1}(x+u+i,y+v+j)) \quad (2.2.9)$$

where $\phi(\cdot)$ is a objective function, and (u,v) is restricted to a search space defined by a pre-determined maximum velocity. The summation is carried out over all pixels within the block. The local image motion values obtained by matching techniques are usually integer-valued. When sub-pixel precision is required, intensity interpolation is needed.

The optimization problem (2.2.9) can be solved by performing a very simple exhaustive search: computation of the objective function for every possible vector in the given search space, choosing the vector which offers the minimum value of the objective function. To speed-up the search procedure, several methods have been proposed:

1. 2D-logarithmic search [43],

2. three-step search [46], and

3. modified conjugate direction search [78].

These techniques assume monotonicity of the objective function which in general is not always valid. For small search space, however, this function may frequently turn out to be unimodel.

An interesting matching algorithm has been proposed by Anandan [3]. He uses a Laplacian pyramid image structure [10] and a coarse-to-fine matching strategy. Anandan applies local analysis of the matching error surfaces using principle axis decomposition. This results in a heuristic confidence measure associated with each local image motion vector. Singh [74] defines the local image motion vector as a weighted average of a set of candidate matching vectors. Thus, the resulting motion vectors have sub-pixel precision. A covariance-matrix is also associated with each local image motion vector. The reciprocals of the eigenvalues of the covariance-matrix serve as confidence-measures with the directions given by the corresponding eigenvectors.

The simple translational motion model assumption made in matching approaches is often not adequate for images which contain complex motion. In [17], an improved algorithm for block matching is proposed, which allows the blocks to undergo affine shape deformation. The parameters for this affine model are found via a least-squares algorithm. The cost of this new model is the increased computation time for each local image motion vector.

Compared to differential approaches, matching approaches are usually more computationally demanding. However, since the matching operation is identical for all pixels in an image, it can be efficiently implemented as a parallel convolution operation using special-purpose hardware.

2.2.3 Spatiotemporal filtering

Estimation of image motion fields based on spatiotemporal filtering relies on three-dimensional filtering of image sequences[19, 20, 29]. Intensity is represented as a

three-dimensional function of spatial variables x and y and temporal variable t. A bank of three-dimensional filters in (x, y, t) space is used to determine the image-motion field vector for each pixel. Each three-dimensional filter in the bank is tuned to a specific velocity magnitude and direction.

Spatiotemporal filtering approaches are also thought of as frequency-domain based owing to the common design of velocity-tuned filters in the Fourier domain. However, these filters are still based on the assumption that image intensity is constant over time as expressed in Equation (2.2.1). The Fourier transform of Equation (2.2.1) is

$$\hat{E}(\mathbf{k}, \omega) = \hat{E}_0(\mathbf{k})\delta(\omega + (u, v)^T\mathbf{k}) \tag{2.2.10}$$

where $\hat{E}_0(\mathbf{k})$ is the Fourier transform of $E(\mathbf{x}, 0)$, $\delta(k)$ is a Dirac delta function, ω denotes temporal frequency and $\mathbf{k} = (k_x, k_y)$ denotes spatial frequency. This equation shows that all nonzero power associated with a translating pattern lies on a plane passing through the origin in frequency space. Heeger [29] used 3-D Gabor filters tuned to different spatiotemporal-frequency bands and describes a method for combining the outputs of the filters to compute local velocity vectors.

Techniques based on spatiotemporal filtering are usually used to explain and model the human visual system. With regards to image motion estimation at motion boundaries, the same problems exist as with other approaches. The image motion vector at a point (x, y) on frame t is determined by the outputs of a set of three-dimensional filters. Each of these filters has a finite region of support in both spatial and temporal domains. When the image motion vector at a pixel is based on the outputs of filters applied to a finite three-dimensional neighborhood of the pixel, it is implicitly assumed that all pixels in the associated three-dimensional neighborhood move with the same velocity as that pixel itself. However, this is an assumption which is violated at motion boundaries. As a result, image motion estimation at motion boundaries are often not reliable and motion boundaries are easily blurred. The extent of blurring is directly proportional to the regions of support of the filters in the spatial and

temporal domains.

2.3 Image motion estimation by regularization

The estimation of image-motion field is often ill-posed in the original sense of Hadamard [6]: the solution may not be unique, it may not exist, or it may not depend continuously on the data. Even when a problem is not ill-posed, if constraints relating data to the real world variables of interest are noisy, the unique solution to the noisy constraints is not particularly meaningful. For image-motion field estimation, the major difficulty is that the local motion measurements obtained by the approaches described in the previous section are often corrupted by image noise and may not yield a unique solution.

To deal with ill-posed problems, two branches of mathematical analysis have been developed: the theory of *generalized inverses* and *regularization* theory. Assume that functional spaces X and Y, as well as a linear, continuous operator L from X to Y are given. The task of an inverse problem is to find, for some given $d \in Y$, a function $w \in X$ such that

$$d = Lw. \tag{2.3.1}$$

The theory of generalized inverses attempts to solve the problem by minimizing the norm of a certain function derived from (2.3.1). Inverses can be classified according to the choice of that function as follows:

1. *Least squares inverses*: the following variational problem is solved

$$min_{\{w \in X\}} \|Lw - d\|_Y \tag{2.3.2}$$

where $\|\cdot\|_Y$ denotes the norm in space Y. This problem results in the linear system of equations $L^*Lw = L^*d$ (L^* is the adjoint operator of L) for which the existence and uniqueness of the solution depend on the rank of L^*L.

2. *Generalized inverse*: the solution (2.3.2) is sought such that it is also of minimum norm:

$$min_{\{w \in X\}} \|w\|_X \tag{2.3.3}$$

3. *C-Generalized inverse*: the solution (2.3.2) is sought such that it is also minimum in a constraint space:

$$min_{\{w \in X\}} \|Cw\|_Z \tag{2.3.4}$$

where C is a linear operator from X into the constraint(functional) space Z.

An alternative to the above formulations based generalized inverses are regularization methods. The most investigated regularization method is to form the following optimization problem:

$$min_{\{w \in X\}} \lambda \|Lw - d\|_Y^2 + \|Cw\|_Z^2 \tag{2.3.5}$$

where C is a linear operator from X into the constraint space Z. The parameter λ is called the regularization parameter, and $\|C \cdot\|$ is the regularization functional, which usually expresses some desired property expected from the solution (e.g., smoothness, directionality). Regularization parameter $\lambda > 0$ weights the compromise between data approximation and model fitting (expressed by the regularization functional).

There are numerous examples of regularization in the field of image processing [6]. Numerical differentiation in image edge detection has been formulated in the framework of regularization [35, 66], where the regularization functional uses the second derivative of the approximating function. Other problems which are approached using regularization theory are shape from shading [42], and surface interpolation [8]. Most image-motion field estimation techniques also employ regularization [23, 34]. Horn and Schunck [36] use regularization to minimize the departure of smoothness in the flow field as measured by the squared magnitude of the gradient of the flow field summed over both components which has the form

$$(u_x^2 + u_y^2 + v_x^2 + v_y^2) \tag{2.3.6}$$

with the local constraint over the space, D, of admissible image-motion field vectors $\mathbf{u} = (u, v)$. The estimate of the image-motion field is the solution to the following optimization problem:

$$Min_{(\mathbf{u})} \int\int_D \lambda(E_x u + E_y v + E_t)^2 + (u_x^2 + u_y^2 + v_x^2 + v_y^2). \tag{2.3.7}$$

Regularization theory has a simple statistical interpretation. Minimizing "energy" $= \lambda\|Lw - d\|_Y^2 + \|Cw\|_Z^2$ is the same as maximizing $e^{-k\text{"}energy\text{"}}$. The expression $e^{-k\text{"}energy\text{"}}$ can be regarded as a probability density function if the normalization constant k is chosen correctly. Standard regularization theory maximizes the likelihood of the solution w if the error vector $(Lw - d)$ is assumed to be Gaussian. The spaces X, Y and Z in this case refer to the the Hilbert spaces of random variables. Therefore, when a priori knowledge of statistical properties of the signal and noise are available, probabilistic versions of regularization methods can also be developed. In this probabilistic formulation, the underlying process and/or its relationship with observations are considered as samples of some random processes.

Konrad and Dubois [48, 47] present a probabilistic formulation for image-motion field estimation and a stochastic algorithm for minimization of the associated objective function. In their formulation, the observation model, relating the underlying displacement field and the observed images, expressed as additive Gaussian noise, is combined with the structural model assuming constant image intensity along motion trajectories. A vector Markov Random Field (MRF) is used for the displacement field model. Such a model is able to capture various image-motion field properties (e.g. smoothness) in terms of spatial interactions which can be controlled by certain parameters. Gibbs distributions are then used to uniquely characterize the spatial properties of this vector Markov Random Field. Combined via Bayes' rule, these distributions provide a cost functional to be minimized. The minimization problem, involving several thousands of unknowns, has been solved using simulated annealing. A stochastic relaxation algorithm, the Gibbs sampler, originally proposed in [22], has

been used to generate MRF samples according to the *a posteriori* probability.

2.4 Regularization that includes discontinuities

An outstanding problem for the computation of image-motion fields by using regularization is the presence of discontinuities in the image-motion field. The smoothness assumption disambiguates the aperture problem and smoothes the local measurement noise. But the global smoothing may also smear motion boundaries and produce inaccurate estimates. In a recent experimental study [4], it is reported that the techniques based on global smoothing tend to have lower accuracy of motion field estimation than those based on local smoothness constraints. Also, it difficult to distinguish the moving objects from background or other moving objects when the motion boundaries are smoothed over. It is generally agreed that motion discontinuities convey useful information in many applications, because they indicate where one object ends and another one begins. Motion as well as intensity discontinuities are also vital for solving the critical object segmentation problem.

Classical regularization theory does not address the problem of the presence of motion discontinuities in image motion fields. On the contrary, discontinuities contradict the basic smoothness assumption in the theory. A significant challenge is thus to extend regularization theory to deal with discontinuities. Lee and Pavlidis [52] have proposed an extension for the one dimensional case. The two-dimensional extension is significantly more difficult. Blake uses a weak continuity constraint in the problems of surface reconstruction and edge detection. Though his formulation is not explicitly framed in the context of classical regularization theory, it represents some promising initial steps in this direction [8].

Geman and Geman [22] proposed a successful strategy for dealing with discontinuities in image restoration and intensity-based segmentation. They exploited an analogy between statistical mechanics and digital images, where the intensity values

at each pixel and the presence of discontinuities are viewed as states of particles on a lattice. The smoothness assumption is formulated in terms of a Markov Random Field (MRF) model. In the MRF, the conditional probability that a given variable at location (i, j) has a particular value f_{ij} depends only on the values of f in a neighborhood of (i, j). Two random processes are used in their formulation: one is the intensity random field and another random process, a line process, is introduced to model the intensity discontinuities. Line elements are located on a regular lattice consisting of sites placed between each adjacent pair of pixels. A line element, l, can occupy one of two states: "on" ($l = 1$) or "off" ($l = 0$). The decision to turn on a particular line element state is combined within the global model and estimation of the intensity and line processes are simultaneous.

Motivated by this idea, Konrad and Dubois [47, 49] proposed a displacement field model incorporating a line processes to model motion field discontinuities. When a line element is turn on, any motion vectors that cross the line element will not be smoothed. Conversely, when displacement vectors on each side of a line element site differ considerably, the line element at this site is turn on. To prevent a line process from forming everywhere and to incorporate additional knowledge of motion discontinuities into the line process, a structural model is constructed for the line process. Structural considerations are used to prevent the formation of parallel line elements, multiple line intersections and isolated discontinuities, while at the same time, are designed to favor the formation of motion discontinuities along extended contours. A deterministic version of the line process has also been used in image-motion field estimation via nonstochastic methods [41].

A formulation similar to [47] which combines gradient-based and feature-based motion estimation schemes is proposed by Heitz and Bouthemy [31]. In the observation model, two complementary motion measurement equations are used. The first is based on the image-motion field constraint equation, and the second is derived from

the output of a moving edge estimator. The model employs a simple intensity edge detector to provide binary information about intensity discontinuities which is used as partial evidence supporting the presence of motion discontinuities. The method of Iterated Conditional Modes [7], a low cost alternative to simulated annealing, is used to minimize the cost functional. In this deterministic relaxation scheme, the final result depends heavily on state initialization and site visiting order.

The line process described above models motion discontinuities in an explicit form: values of line elements are binaried and cause computational problems in the above formulations since the resulting optimization functional is not convex. Blake and Zisserman [8] show that the line process can be eliminated from the regularization functional, resulting in a cost functional which is solely dependent on the actual surface function under consideration. In Chapter 3, we develop an error-weighted regularization algorithm which uses the local matching errors to guide the smoothing process instead of the modeling discontinuities explicitly. It is shown that the high computational cost associated with the line process is greatly reduced without a corresponding degradation in the quality of motion estimation.

Chapter 3

ERROR-WEIGHTED REGULARIZATION

In this chapter, an error-weighted regularization algorithm for image-motion field estimation from image sequences is presented. The main goal of this new algorithm is to obtain reliable image-motion field estimates when there are motion discontinuities and image occlusions due to multiple moving objects present in the image sequences. The algorithm is based a general regularization framework that includes a new form of regularization functional and takes the motion discontinuities into consideration. The new algorithm's regularization functional differs from existing regularization functionals in that block-matching errors are used to control the field-smoothing process. The large block-matching errors along motion boundaries act as a motion vector propagation barrier at the global smoothing stage. As a result, the motion measurement errors in occluded areas do not spread out to other regions. The local measurements have confidence measures associated with them based on the distribution of block-matching errors. Thus the problem of nonuniqueness of local measurements is automatically taken into account in the formulation of the cost functional. The regularization constant is also specified by the confidence measures as suggested by Anandan [3]. No measurement or motion information filling is attempted on the regions with uniform intensity. This enhances the algorithm's robustness to image noise and reduces the

possibility of motion boundary smearing. A later stage of filling and segmenting the image-motion field can be accomplished by using *a priori* knowledge or assumptions about the scene being viewed. Besides improved image-motion estimation performance near motion discontinuities, the algorithm is computationally similar to the standard regularization approach as used by Horn and Schunck [36] except that fewer iterations are required for convergence. The algorithm is also amenable to implementation in special purpose hardware due to the algorithm's inherent parallel nature at both the local and global processing stages [41].

The remainder of this chapter is organized as follows. In the next section, we overview the standard regularization framework and mention its drawbacks with respect to the image-motion field estimation problem. In Section 3.2, several arguments are provided to motivate the block-matching technique for the local measurement process. In Sections 3.3, 3.4, and 3.5, the properties of block-matching errors, the different types of motion boundaries, and the confidence measures derived for local motion vectors are examined, respectively. In Section 3.6, we describe the new matching-error weighted regularization algorithm and in the final section experimental results are shown that verify the improved performance of the new algorithm.

3.1 Standard regularization and its drawbacks

Due to the noise corruption and "aperture problem" embedded in the local image-motion measurement process, the method of regularization has been employed in a number of image-motion field estimation algorithms [3, 36, 41, 57, 75]. In the regularization framework, image-motion field estimation can be formulated as the solution to an optimization problem of the form:

$$min_{\{\mathbf{u}\}}\{\lambda\Delta(\mathbf{u},\mathbf{d}) + \Phi(\mathbf{u})\}, \tag{3.1.1}$$

where $\mathbf{u}$ is the image-motion vector field, Δ is a measure of the lack of fidelity to local measurement data $\mathbf{d}$, Φ is a regularization functional specifying the smoothness of the motion field, and λ is a positive scalar regularization parameter that weights the compromise between Δ and Φ. The solution to (3.1.1) will produce an image motion estimate which is both faithful to local measurements, and has the desired smoothness properties of Φ as well.

Equation (3.1.1) has the same structure as Equation (2.3.5) in Chapter 2. The data $\mathbf{d}$ and estimate $\mathbf{u}$ lie in two dimensional vector spaces and the norm used in Equation (3.1.1) is the Euclidean norm. In Section 3.6 of this Chapter, a new regularization functional Φ is defined in terms of the Euclidean norm.

The cost functional (3.1.1) is expressed in very general terms, and specification of a particular algorithm entails choosing Δ, Φ and λ. The choice of the Δ term is usually made on the basis of the measured data error distribution. For most cases, this distribution is assumed to be Gaussian. For vector data, confidence measures are often used to reflect the reliabilities of each component measurement. The choice of the Φ term is based on prior knowledge about true image motion. Due to the complexity of real motion in image sequences, only general properties are used to specify Φ . Since one conventionally expects the image-motion field to be smooth and without abrupt changes globally, this Φ term is often called the smoothness constraint. As far as the degree of image-motion field smoothness is concerned, the choice of λ becomes critical.

The previous discussion leads to three different problems in formulating the regularization functional: 1) how well the estimate should fit the measured data, 2) the choice of smoothness constraint used in the regularization functional Φ, and 3) the strength of the smoothness requirements. If the image motion fields are not continuous everywhere, the choice of the Φ term will be very important in order to avoid smoothing across the motion boundaries.

For the data fitting term Δ, Horn and Schunk [36] first use the optical flow constraint equation (2.2.7) in which only the component of local motion vectors normal to the intensity gradients is constrained.

Anandan [3] and Singh [75] use local motion vectors obtained from the block-matching for the data term Δ. The confidence measures derived from the block-matching error surfaces are used to provide space-varying directional weights to local motion vectors.

The optimal choice of λ has proved to be a very difficult problem. Methods based on the properties of the residuals and on generalized cross-validation have been proposed for estimating the regularization parameter [6]. Two alternative criteria, weighted least squares and sum of squared weighted residuals, based on cross-validation have been used by Galatsanos and Katsaggelos [21]. The problem with these methods is that the task of estimating the regularization parameter is more computationally demanding than the original estimation problem. As a result, the application of cross-validation methods have not been readily accepted by most researchers. As an alternative, Horn and Schunk [36] have suggested that λ should be roughly equal to the expected noise in the intensity derivatives. Anandan and Singh [3, 75] implicitly choose λ by using the confidence measures associated with the local motion vectors.

The smoothing functional, Φ, usually takes the form of the squared first- or second-order derivatives of image-motion field estimates, such as in (2.3.6). Singh [74] uses the L_2 norm of the differences between a central estimate and a neighborhood average estimate as the smoothing functional.

In principle, in order to prevent smoothing across motion boundaries by the smoothing functional, one can first detect the motion boundaries [77, 84], and limit the smoothing functional to lie within these boundaries. However, explicit motion boundary detection is a nontrivial task in its own sake and often depends on a prior

motion field estimate. The resulting joint boundary detection and motion estimation problem requires a more sophisticated approach such as described in Section 2.4.

The new algorithm to be introduced in this chapter provides significant improvements in constructing smoothness constraints. A new regularization functional, Φ, is proposed by using matching-error-weighted local motion averaging. The matching errors are readily available from the block-matching process. No auxiliary computations need be performed on the intensity images. Moreover, the increasing availability of special-purpose hardware for block-matching is an important practical consideration [64].

3.2 Motivation for using block-matching

As discussed in Chapter 2, there are three major techniques for the local measurement of dense image-motion fields from image intensity functions:

1. Differential approaches,

2. Matching approaches, and

3. Spatiotemporal filtering approaches.

Block-matching, which belongs to the second category, is used for the local image-motion measurement process in the new algorithm. This choice is based on the following considerations:

- **Applicable to a wide range of displacements**

  In most video image sequences, the amount of image motion between image frames will often be several pixels. Block-matching can measure either small or moderate inter-frame displacements. If intensity interpolation is used, block-matching may also handle sub-pixel displacements.

- **No image pre-filtering is needed**

  Block-matching can be applied to different image intensity functions. In contrast to differential-based techniques, no image pre-filtering is necessary for block-matching [53]. Differential techniques, on the other hand, require a certain amount of image pre-filtering to obtain accurate numerical approximation of intensity derivatives which may be difficult in regions where the intensity function is not continuous, such as at edges. Since the pre-filtering will mix the intensity patterns of different moving surfaces, the motion field discontinuities will often be poorly estimated and large measurement errors may result such as those occurring at the lower levels when using a multiple resolution representation [3].

- **Amenable to hardware implementation**

  For block-matching, the local measurement stage is homogeneous over the whole image, so is amenable to implementation in special purpose hardware. In fact, new integrated circuit chips for block matching have already been announced by LSI Logic Corp. [64].

- **Confidence measures derivable from block-matching assist global smoothing**

  Image motion estimation using block-matching can be divided into two stages: local measurement and global smoothing. An important advantage of block-matching is that by-products from the local matching stage can be utilized in the global smoothing stage. One such by-product is the confidence measure for the local measurement data which is derived from the matching error surfaces [3]. As will be discussed in the next sections, the new algorithm exploits such information from the local measurements in the global smoothing process to avoid smoothing across the motion boundaries.

One of the disadvantages of block-matching is that it cannot deal with transparent motion phenomena. Also, the motion vectors obtained by block-matching are often integer valued and thus the estimates are poor when motion field contains small velocities with a significant dilational component [4].

3.3 Properties of block-matching errors

Block-matching approaches assume a local 2-d translational motion model for all pixels within a small local window. This model is adequate for most real image sequences as long as the interframe image-plane motion is not very large. The matching algorithm is described as follows: at each pixel in the image, under each integer displacement, the windowed images in the two frames are compared and a measure of the match quality between pixels in the window is computed, and summed over the window. This can be interpreted as matching small patches from the first image to small patches in the second image. There are different matching criteria, such as the maximization of cross-correlation or the minimization of a Euclidean distance metric. A more detailed study of different matching criteria is deferred to Chapter 4. In the following, the mean sum-of-squared difference (SSD) is used as the matching-distance metric due to its simplicity and wide-spread use[3].

Let $E^{\tau}(x,y)$, $E^{\tau+1}(x,y)$ represent the image intensity functions at the times, τ and $\tau+1$, respectively. The SSD matching error between two image windows with a displacement $\mathbf{d}=(u,v)$ is defined as

$$\varepsilon_{x,y}(u,v)=\frac{1}{(2N+1)^2}\sum_{i,j=-N}^{i,j=N}\left(E^{\tau}(x+i,y+j)-E^{\tau+1}(x+i+u,y+j+v)\right)^2 \quad (3.3.1)$$

$$-d_{max}\leq u,v\leq +d_{max}$$

where $2N+1$ is the window size, and d_{max} is the predetermined maximum displacement component. The search space of the matching is denoted by $\mathbf{D}$, and is of size $(2d_{max}+1)^2$. An error surface is defined over $\mathbf{D}$ with height at $\mathbf{d}=(u,v)$ equal to

$\varepsilon_{x,y}(u,v)$. The displacement vector $\mathbf{d}$ with the minimum matching error is chosen as the local motion estimate at pixel (x,y).

Let $\mu_{x,y}$ be the mean height of the matching-error surface over $\mathbf{D}$. The matching-error variance is computed by

$$\sigma^2(x,y) = \frac{1}{(2d_{max}+1)^2} \sum_{(u,v)\in\mathbf{D}} (\varepsilon_{x,y}(u,v) - \mu_{x,y})^2. \tag{3.3.2}$$

For real video image sequences, the existence of a perfect match cannot be guaranteed. There are several factors which preclude a perfect match:

1. **Random noise in the images.**

2. **Illumination change over time and photometric effects of the object surfaces.**

3. **The actual image motion is not purely translational.** For example, object rotation or camera zooming will render a non-translational image-motion field. In this case, the matching window undergoes an area deformation, which violates the assumption of the local translational model used in block-matching.

4. **The matching errors are also related to the intensity variation in the matching windows.** For the image regions with small intensity variation, a large mismatching in displacement may not necessarily result in a large matching error.

5. **The matching windows contain at least two moving surface with different velocities.** In this case, the matching window straddles motion boundaries, the intensity patterns within the window vary between two frames. This problem is more serious when the motion discontinuities cause image occlusion.

Among all the factors listed above, the matching errors will be abnormally large only in case 5, where the motion discontinuities are present in the matching window.

This has been verified experimentally in Section 3.7. If one can remove the effect of image intensity variation on the matching errors, the moving surfaces then can be well demarcated by large local matching errors. To counteract the effects of intensity variation, the minimum matching error is scaled by the error surface variance $\sigma^2(x, y)$ as defined in (3.3.2) and this scaled matching error is denoted as $\varepsilon'_{x,y}$. This scaling is based on the observation that the variances of the matching error surfaces are proportional to the intensity variations.

3.4 Motion boundary types

To cope with motion discontinuities and the image occlusion problem, we need to study matching error effects more closely. To illustrate the concepts, we classify three extreme types of motion boundaries and image occlusions according to the type of bordering image texture patterns. It is worth reminding the reader that real motion boundaries do not always fall into these three extreme types.

1. **No-texture/no-texture**: In this motion boundary type, both sides of motion boundary are uniform intensity patterns and only edges are present. Thus, the disoccluded or occluded image regions do not produce any mismatches. In this case, the image-motion discontinuities do not cause the matching error to increase.

2. **No-texture/texture**: Here one side of the boundary is a uniform intensity pattern and the other side is a well-textured intensity pattern. An example of this type of motion boundary is shown on the top part of the image in Figure 3.1(a). If the textured intensity pattern is the occluding surface, the image occlusion will have no effect on the local measurements made on the textured side since the occluded pixels have false matches. These can be seen from the matching error image shown in Figure 3.1(b) where small matching errors are

present along the motion boundaries on the top part of the image. However, if the textured intensity pattern is disoccluded or occluded in the second frame, large matching errors will occur in the local motion measurement process.

3. **Texture/texture**: Here both sides of motion boundaries have well-textured intensity patterns. In this case, the image occlusions on either side will cause large matching errors since both occluded and disoccluded pixels will not have correspondences either in the first frame or in the second frame. The motion boundaries of this type are shown in Figure 3.1(a) in the lower part of the image. The matching errors along those boundaries have abnormally large values as shown in Figure 3.1(b).

From the above illustrations, it can be concluded that matching errors will abruptly increase in the case of the **texture/texture** boundary or the **no-texture/texture** boundary when the textured intensity pattern is disoccluded or occluded in the second frame. Therefore, the only type of motion boundaries that can be reliably indicated by matching errors is the **texture/texture** boundary. On the other hand, the other two types of motion boundary can be easily identified by detecting uniform intensity surfaces and eliminating them from the local measurement and global smoothing processes. In other words, the uniform intensity regions do not contain any motion information and do not contribute anything to image-motion field estimation. The detection of uniform intensity regions can be achieved by a threshold test of image intensity variance with the threshold determined using prior information on the image noise-level. This amounts to a test of variance for simple binary hypotheses [67].

3.5 Confidence measures for local motion

In general, there will be many areas of the image with insufficient information for a complete and reliable local determination of the image motion vectors. The different

directional components of the image motion vectors may be locally computed with different degrees of reliability. For instance, it is obvious that in a uniform intensity region, no component of the image motion vector can be estimated. At a pixel location along a line (or edge), the component perpendicular to the line will have higher reliability than the component parallel to the line. Finally, at a pixel of high curvature along an image contour it may be possible to completely and reliably determine the image motion vector on the basis of local information. Therefore confidence measures are needed to reflect local measurement reliability for the later global smoothing stage.

In Anandan's paper [3], a computational framework of image motion measurement is introduced that associates a direction-dependent confidence measure with every measured local image-motion vector. This confidence measure is based on the variation of the matching errors over the search space. It combines information in the minimum matching error with the matching error distribution around the minimum matching error location. The confidence measure consists of two directions, $\hat{e}_{max}$ and $\hat{e}_{min}$ which denote the principal axes of the matching error surface, and two scale factors, c_{max} and c_{min}, as given by

$$c_{max} = \frac{C_{max}}{k_1 + k_2\varepsilon_{min} + k_3 C_{max}} \tag{3.5.1}$$

and

$$c_{min} = \frac{C_{min}}{k_1 + k_2\varepsilon_{min} + k_3 C_{min}} \tag{3.5.2}$$

where C_{max} and C_{min} are the two principal curvatures of the matching error surface associated with $\hat{e}_{max}$ and $\hat{e}_{min}$, respectively, k_1, k_2 and k_3 are three normalization parameters, and ε_{min} is the matching error corresponding to the best match. In this formulation, the local image motion vector is decomposed into components along $\hat{e}_{max}$ and $\hat{e}_{min}$, respectively. Each component's reliability is computed based on the minimum matching error and corresponding curvature in its direction.

Singh employs a covariance-matrix as a confidence measure associated with each local image motion vector [74]. In Singh's algorithm, the matching errors $\varepsilon(u, v)$ are

first mapped onto the unit interval by an exponential function. That is,

$$R(u,v) = e^{-k\varepsilon(u,v)} \tag{3.5.3}$$

where k is a normalization factor chosen such that the maximum of $R(u,v)$ is close to unity. The local image motion measurements are obtained by a matching error weighted-least-squares estimator which actually is a weighted mean of $R(u,v)$ over the search space. This estimate is given by

$$u_{cc} = \frac{\sum_u \sum_v R(u,v)u}{\sum_u \sum_v R(u,v)}, \tag{3.5.4}$$

$$v_{cc} = \frac{\sum_u \sum_v R(u,v)v}{\sum_u \sum_v R(u,v)}. \tag{3.5.5}$$

Under the assumptions of additive, zero mean and statistically independent errors, the covariance-matrix associated with u_{cc} and v_{cc} is of the form:

$$S_{cc} = \begin{pmatrix} \frac{\sum_u \sum_v R(u,v)(u-u_{cc})^2}{\sum_u \sum_v R(u,v)} & \frac{\sum_u \sum_v R(u,v)(u-u_{cc})(v-v_{cc})}{\sum_u \sum_v R(u,v)} \\ \frac{\sum_u \sum_v R(u,v)(u-u_{cc})(v-v_{cc})}{\sum_u \sum_v R(u,v)} & \frac{\sum_u \sum_v R(u,v)(v-v_{cc})^2}{\sum_u \sum_v R(u,v)} \end{pmatrix} \tag{3.5.6}$$

where the summation is carried out over the search space D. The reciprocals of the eigenvalues of this covariance-matrix are taken as the confidence measures associated with the estimates u_{cc} and v_{cc}, along directions given by the corresponding eigenvectors.

It should be pointed out that these confidence measures only reflect the relative reliabilities between the two components of the local image motion vectors. A separate regularization parameter is still needed in the global smoothing process. In [3, 74], the confidence measures also serve as regularization parameters. Such a treatment might be interpreted as a practical *ad hoc* solution to the problem of choosing regularization parameter λ.

3.6 Error-weighted regularization

Since the matching errors can indicate the presence of motion boundaries as discussed in Section 3.3, they could instead be used in the global smoothing stage to avoid smoothing across motion boundaries. For this purpose, we introduce a new regularization functional Φ which utilizes information contained in the matching error surfaces.

For a pixel located at (x, y) which is on a motion boundary, there will two sets of neighborhood pixels around that pixel. Let set A belong to the same moving object as the pixel and let set B belong to a different moving object. Using a new regularization functional, we regularize the local motion vector $d(x, y)$ by the motion information in set A. Thus the motion information from two moving objects will not be mixed and no oversmoothing across motion discontinuities will occur. To do this, a local average displacement vector $\bar{\mathbf{u}}(x, y)$ of $\mathbf{u}(x, y)$ for each pixel (x, y) is first defined as

$$\bar{\mathbf{u}}(x, y) = \frac{\sum_{(i,j)\in S(x,y)} w(i, j)\mathbf{u}(i, j)}{\sum_{(i,j)\in S(x,y)} w(i, j)} \tag{3.6.1}$$

and

$$w(i, j) = \frac{1}{\varepsilon'_{i,j}}$$

where $S(x, y)$ is a neighborhood of the pixel at (x, y) and $\varepsilon'_{i,j}$ is the scaled minimum matching error defined by Equation (3.3.1). We have experimentally compared the results of an eight-pixel and a four-pixel neighborhoods and there is no significant difference. Therefore a four-pixel neighborhood $S(x, y)$ is used which makes the new algorithm computationally comparable to standard regularization [36].

The contribution of neighborhood motion information to the local average motion vector in Equation (3.6.1) is weighted by the inverse of the scaled matching-error, $\varepsilon'_{i,j}$ of each neighborhood motion vector. The pixels with larger $\varepsilon'_{i,j}$ contribute less

to $\bar{\mathbf{u}}(x,y)$. The scaled minimum matching errors are used to direct motion information propagation. Each pixel obtains global motion information only from neighborhood pixels which have comparatively reliable measurements. Therefore, the value of $\bar{\mathbf{u}}(x,y)$ is primarily determined by displacement vectors of pixels on one side of the boundary when there is a motion discontinuity adjacent to the pixel (x,y) and the matching window size is small enough.

An alternative formulation would be to derive the above matching error weights based on the inverse covariance matrix of the matching error surface. Such an approach would generalize Singh's Φ functional [75] from a distance-based measure to an adaptive matching error based measure. With a certain idealized statistical interpretation, it can be argued that such a formulation would be theoretically preferred to that above. However, the matching error covariance matrix requires significant added computational complexity.

The new regularization functional Φ is defined as the Euclidean norm of the difference of $\mathbf{u}(x,y)$ and $\bar{\mathbf{u}}(x,y)$, i.e.,

$$\Phi_{x,y}(\mathbf{u}) = \|\mathbf{u}(x,y) - \bar{\mathbf{u}}(x,y)\|^2. \tag{3.6.2}$$

From the construction of the local motion average vector $\bar{\mathbf{u}}(x,y)$, minimizing the Φ term in the cost function will smooth each image motion vector selectively towards its consistent neighboring motion vectors.

The data fidelity terms, Δ and λ in Equation (3.1.1), are designed as in [3]. Let $\{\mathbf{d}\}$ be the set of local image motion measurements obtained by the matching process, which can be represented using the local orthogonal basis $(\hat{\mathbf{e}}_{max}(x,y),\hat{\mathbf{e}}_{min}(x,y))$, which denote the principal axes of the matching error surface. In this data term, one minimizes the error between the image-motion field estimate $\{\mathbf{u}\}$ and $\{\mathbf{d}\}$. The error is a weighted sum of the squared deviations of the components of $\mathbf{u}(x,y)$ along corresponding components of the local matching motion vector $\mathbf{d}(x,y)$. The weights

are the confidences $c_{max}(x,y)$ and $c_{min}(x,y)$, i.e.,

$$\lambda\Delta(\mathbf{u},\mathbf{d}) = \sum_{x,y}[c_{max}(\mathbf{u}\cdot\hat{\mathbf{e}}_{max} - \mathbf{d}\cdot\hat{\mathbf{e}}_{max})^2 + c_{min}(\mathbf{u}\cdot\hat{\mathbf{e}}_{min} - \mathbf{d}\cdot\hat{\mathbf{e}}_{min})^2]. \quad (3.6.3)$$

In the above and following equations, the indices (x,y) have been omitted from each term for notational compactness. Using (3.1.1), (3.6.2) and (3.6.3), the cost functional to be minimized for the error-weighted regularization is

$$I(\mathbf{u}) = \sum_{x,y}[c_{max}(\mathbf{u}\cdot\hat{\mathbf{e}}_{max} - \mathbf{d}\cdot\hat{\mathbf{e}}_{max})^2 + c_{min}(\mathbf{u}\cdot\hat{\mathbf{e}}_{min} - \mathbf{d}\cdot\hat{\mathbf{e}}_{min})^2 + \|\mathbf{u}-\bar{\mathbf{u}}\|^2]. \quad (3.6.4)$$

There are well-developed techniques for minimizing the cost functional given by Equation (3.6.4). Analogous to [74], at each spatial location a discrete normal equation is first derived by viewing this functional minimization problem as a vector parameter estimation problem and the term $\bar{\mathbf{u}}$ as constant vectors. The motivation for constant $\bar{\mathbf{u}}$ is to make the structure of the resulting normal equations similar to that from the standard regularization functional which is computationally efficient for implementation. Setting the derivatives of $I(\mathbf{u})$ over each $\mathbf{u}(x,y)$ to zero, and assuming $\bar{\mathbf{u}}$ as constant vectors, the following system of coupled linear equations is obtained:

$$(\mathbf{u}-\bar{\mathbf{u}}) + c_{max}(\mathbf{u}\cdot\hat{\mathbf{e}}_{max} - \mathbf{d}\cdot\hat{\mathbf{e}}_{max})\hat{\mathbf{e}}_{max} + c_{min}(\mathbf{u}\cdot\hat{\mathbf{e}}_{min} - \mathbf{d}\cdot\hat{\mathbf{e}}_{min})\hat{\mathbf{e}}_{min} = 0. \quad (3.6.5)$$

Numerical methods exist for solving the system of couple linear equations (3.6.5). One of the simplest methods is the Gauss-Seidel relaxation algorithm [33]. This is an iterative process, where during each iteration the value of $\mathbf{u}$ at each pixel in the image is solved for in terms of the values of its neighbors and its local motion measurement. The iterative update equation at the $k+1$st iteration is

$$\mathbf{u}^{k+1} = \bar{\mathbf{u}}^k + \frac{c_{max}}{c_{max}+1}((\mathbf{d}-\bar{\mathbf{u}}^k)\cdot\hat{\mathbf{e}}_{max})\hat{\mathbf{e}}_{max} + \frac{c_{min}}{c_{min}+1}((\mathbf{d}-\bar{\mathbf{u}}^k)\cdot\hat{\mathbf{e}}_{min})\hat{\mathbf{e}}_{min} \quad (3.6.6)$$

where $\bar{\mathbf{u}}^k$ is defined as

$$\bar{\mathbf{u}}^k = \frac{\sum_{(i,j)\in S(x,y)} w(i,j)\mathbf{u}^k(i,j)}{\sum_{(i,j)\in S(x,y)} w(i,j)}. \quad (3.6.7)$$

Since the normal system of Equations (3.6.5) is linear in $\mathbf{u}$, the stability of the iterative algorithm given by Equations (3.6.6) and (3.6.7) can be shown using a similar argument to that found in [73].

We remark that computationally, this algorithm is similar to standard regularization as used in [36], and can therefore be run on similar special-purpose hardware. The only difference is that this algorithm uses spatially-varying FIR filter coefficients for neighborhood-averaging. It has been observed experimentally that the new algorithm needs fewer iterations than standard regularization for the same stopping criteria.

3.7 Experimental results

This section describes the results by applying the ideas described above to a set of test images. The stopping criterion used in the iteration process (3.6.6) is

$$\frac{\sum \|\mathbf{u}_{k+1} - \mathbf{u}_k\|^2}{\sum \|\mathbf{u}_k\|^2} \leq 10^{-4} \tag{3.7.1}$$

In the implementation, the matching window size, N, is set to 2. The parameters involved in the computation of the confidence measures are the normalization constants k_1,k_2 and k_3 in (3.5.1) and (3.5.2). For these experiments, the choices of $k_1 = 50$, $k_2 = 1$ and $k_3 = 0$ are used based on guidelines discussed in [3]. A 3x3 set of matching errors around the best match is taken as the error surface for the computation of the local confidence measure for each pixel. This involves computing weighted sums of the 3x3 values to obtain first- and second-order derivatives of the error surface and then a singular-value decomposition (SVD) algorithm [68] is used to calculate the principal curvatures and principal axes. Although a larger error surface size may yield more reliable confidence measure estimates, the computational cost is higher.

The image-motion field estimation error, measured by mean squared error (MSE),

is defined by:

$$\frac{1}{I}\sum \|\mathbf{d}-\mathbf{u}\|^2 \tag{3.7.2}$$

where I is the total number of pixels in one image, and $\mathbf{d}$ is the local image-motion measurement and $\mathbf{u}$ is the true motion vector. The signal-to-noise ratio (SNR) of the estimates is defined as

$$SNR = 10log_{10}\frac{\|\mathbf{u}\|^2}{\|\mathbf{d}-\mathbf{u}\|^2}dB. \tag{3.7.3}$$

For purpose of comparison, the algorithm is also tested with all the error-weights set to unity. The resulting algorithm is called distance-weighted regularization since the weights will be roughly same for the pixels in a 3x3 window by using a Gaussian distance mask which is comparable to the approach taken by Singh [75].

Synthetic image sequences were first used to test the algorithm. Synthetic sequences have the advantage that the true image motion fields are known, and the results from the motion estimation algorithms can be quantitatively compared to the true motion field using (3.7.2) and (3.7.3). For image interpolation, real image sequences can then be used since the images to be interpolated are known.

3.7.1 Sinusoidal input with pure translation

The first pair of synthetic images (64x64) represents a diagonal translation of the square sinusoidal intensity pattern towards the lower-right corner (two pixels horizontally and four pixels vertically). The first frame of the pair is shown in Figure 3.1(a). Gaussian grey-level noise with zero mean and variance of 2 is added to this sequence. The sinusoids on the lower part of the stationary background have spatial wavelengths of 10 pixels and the sinusoids belonging to the moving foreground pattern have spatial wavelengths of 15 pixels. The motion boundaries in the top part are the **no-texture/texture** type as described in Section 3.4 due to the uniform intensity pattern in the upper half of the image. These motion boundaries are processed by detecting the uniform background and excluded from the local motion

Figure 3.1: Translating square over textured and uniform background. (a) Local measurement. (b) matching error image. (c) Error-weighted regularization. (d) Distance-weighted regularization.

measurement and global smoothing stages. The uniform background is detected by thresholding the intensity variance. The threshold value used is 8 in the experiments. This threshold value is obtained by subjectively viewing the detection results. The image-motion field obtained from local block-matching superimposed in the first image is shown in Figure 3.1(a). Clearly, the motion fields have large noises along motion boundaries. The minimum matching error image is shown in Figure 3.1(b). Large matching errors appear along the motion boundaries in the lower part of the image as we discussed in Section 3.5. The image-motion fields smoothed by error-weighted and distance-weighted regularizations are shown in Figures 3.1(c) and 3.1(d), respectively. The motion boundaries in the image-motion field are well-preserved for the error-weighted regularization algorithm and the motion field estimates are more accurate than those produced by distance-weighted regularization. The motion boundaries in the image-motion field obtained by distance-weighted regularization algorithm are mostly oversmoothed. Table 3.1 lists the MSE/vector and SNR values for the local measured motion field and smoothed motion fields obtained by the two regularization algorithms. Note that error-weighted regularization takes fewer iterations than distance-weighted regularization for the same stopping criterion.

Algorithms	MSE	SNR	number of iterations
Local measurement only	1.56	5.06 dB	0
Distance-weighted regularization	0.96	7.16 dB	11
Error-weighted regularization	0.78	8.04 dB	5

Table 3.1: Motion field estimation for "Square" images

3.7.2 Rotating and zooming disc

The second image pair represents a more complicated motion. The first frame is shown in Figure 3.2.(a). This pair of images is designed to test the overall abilities of error-weighted regularization in more realistic conditions. Here, the background pattern translates two pixels horizontally, and the circular foreground pattern zooms by a factor of 0.04 units in pixel distance while rotating by 4 degrees. The resulting maximum displacement is 6 pixels. For this example, the matching errors are caused by image noise, local motion model error and motion discontinuities as discussed in Section 3.3.

The local motion measurements for the image pair is shown in Figure 3.2(a). Clearly, the measured motion field has large errors along motion boundaries. Figure 3.2(b) reveals significant matching errors along the motion boundaries and also matching errors appearing within the surface of the circular disc. The latter errors are caused by a failure of the local motion translational model. The motion field obtained by the error-weighted regularization algorithm is shown in Figure 3.2(c). One can see that motion vectors do not propagate across motion boundaries where the matching error values are relatively high. At the same time, the motion field is adequately smoothed within the motion boundaries. The motion estimation errors, calculated for both distance-weighted and error-weighted smoothing, are listed in Table 3.2.

Algorithms	MSE	SNR	number of iterations
Local measurement only	1.39	8.21 dB	0
Distance-weighted regularization	0.53	12.43 dB	9
Error-weighted regularization	0.24	17.30 dB	4

Table 3.2: Motion field estimation for "Disc" images

Figure 3.2: Rotating and zooming disc over translating background. (a) Local measurement. (b)matching error image. (c) Error-weighted regularization. (d) Distance-weighted regularization.

3.7.3 Real images

In subsequent examples, we apply the new algorithm to real video images and the image-motion fields obtained are then applied to motion-compensated image sequence interpolation. In this application, the task is to interpolate missing images temporally located between two given two image fields based on motion information. In the following, the intensity values $E(\mathbf{x}(t))$ for all pixels along the motion trajectory $\mathbf{x}(t) = \mathbf{x}(t_0) + \tau\mathbf{u}(\mathbf{x}, t_0)$ is given by [14]

$$E(\mathbf{x}(t)) = (1-\tau)E(\mathbf{x}, t_0) + \tau E(\mathbf{x} + \mathbf{u}(\mathbf{x}, t_0), t_1) \tag{3.7.4}$$

where $\tau = \frac{t-t_0}{t_1-t_0}$ and t_0, t_1 are the temporal indices of first and second image frames respectively. First, the experiment is performed on Fields 1 and 5 of "Femme et arbre", by Konrad and Dubois [47, 49]. This sequence contains natural motion (a woman decorating a Christmas tree). Field 3 of the sequence is shown in Figure 3.3(a), and the resulting motion field by error-weighted regularization is shown in Figure 3.3(b). As shown, the motion boundary between the right hand and mouth is accurately estimated. This motion field is then used to interpolate images between fields. The motion-compensated interpolation error for Field 3 is shown in Figure 3.3(d) and the MSE/pel is 12.87 which is somewhat higher than reported in [47] (MSE/pel of 8.77) where a *line process* is introduced to model motion discontinuities. However the computational requirements of the new algorithm are much less than the stochastic solution using a line process and a very good image interpolation quality is still obtained, as shown in Figure 3.3(c).

The algorithm is also applied to a locally digitized image sequence of a translating and spinning chair which consists of 124 interlaced fields with 256 by 256 pixel spatial resolution. Field 55 of the chair sequence is shown in Figure 3.4(a). The motion field between Fields 53 and 57 of the sequence is estimated by the error-weighted regularization algorithm. The mean interpolation errors (MSE/pel) for fields 54, 55 and 56 recovered by motion-compensated temporal image interpolation are summarized

Figure 3.3: (a) Original Field 3 of "Femme et arbre" sequence. (b) Estimated motion field between Fields 1 and 5. (c) Recovered field 3 of "Femme et arbre"sequence. (d) Interpolation error image for Field 3.

Figure 3.4: (a) Original Field 55 of the chair sequence. (b) Interpolation error image for Field 55.

in Table 3.3, where the motion-compensated interpolation error performance of the new algorithm is compared with other schemes. The interpolation error image for Field 55 is shown in Figure 3.4(b).

Motion field estimated by	field 54	field 55	field 56
Local motion measurement only	17.160	12.489	16.267
Distance-weighted regularization	15.614	9.572	16.298
Error-weighted regularization	12.815	6.165	12.940

Table 3.3: Motion-compensated interpolation: MSE/pel for chair sequence

The entire 124-field sequence is also uniformly temporally subsampled by a factor of four, representing a maximum inter-field displacement of around 8 pixels. The subsampled sequence of 31 odd fields was then used to interpolate the in-between fields using the image-motion fields obtained by the error-weighted regularization algorithm. The results are shown in Figure 3.5. A sample of interpolation error image is shown in Figure 3.4(b) for Field 55. Evaluating the subjective quality of the recovered and original video sequences on a TV monitor reveals only minor differences in perceived quality. Since the interpolation is performed on the odd fields, significant excess MSE is introduced in the recovery of even fields due to spatial intensity interpolation: here the MSE per pixel ranged from about 4-12 for odd fields, and 13-20 for even fields. This suggests that the interpolation error in interlaced video images is significant and warrants further investigation.

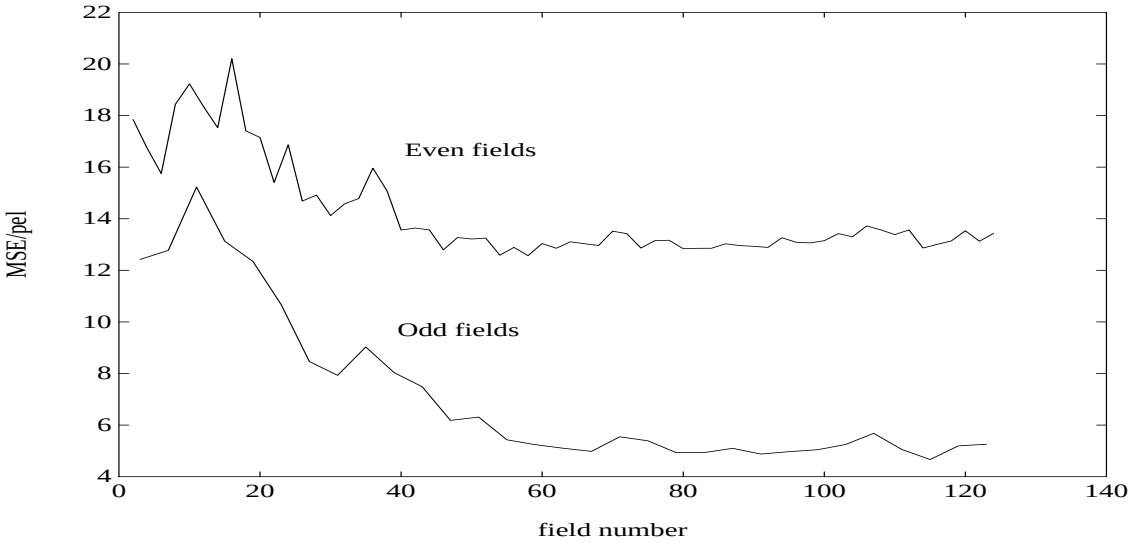

Figure 3.5: MSE/pel versus the field number for recovered "Chair" sequence

Chapter 4

ANISOTROPIC REGULARIZATION

In this chapter, an alternative to the error-weighted regularization scheme presented in Chapter 3 is presented based on anisotropic regularization. In this regularization scheme, a selective confidence measure is proposed. This selective confidence measure is derived from local measurement errors as in Anandan's work [3]. However, instead of being used for judging the reliability of the local motion estimates, the new measure is designed for use in the regularization functional to select the consistent neighboring motion information. The resulting smoothness constraint is no longer isotropic. Also a local matching scheme called *multiple off-centered sub-window matching* (MOW) with the sum of absolute difference (SAD) criterion is designed for more accurate local motion estimation. The matching errors from this local matching scheme are then used to calculate the selective confidence measures. As will be discussed, image-motion boundaries can be remarkably well-preserved by anisotropic regularization and the accuracy of image-motion field estimation can also be significantly improved.

The chapter begins with a brief description of piece-wise continuous functions. The concept of isotropic and anisotropic regularizations is introduced in Section 4.2 using a detailed one-dimensional example. Section 4.3 defines the selective confidence measure with the introduction of the multiple off-centered sub-window matching scheme and SAD criterion. Finally, experimental results are provided which show improved

performance over both isotropic schemes as well as the error-weighted regularization algorithm proposed in Chapter 3.

4.1 Piece-wise continuous functions

By definition, a function $f(x)$ has a discontinuity of degree k at x_0, k=0,1,...., if the kth order left and right derivatives at x_0 differ [52], i.e.,

$$f^{(k)}(x_0+) \neq f^{(k)}(x_0-).$$

In image motion fields, the discontinuities due to different moving objects in the scene are usually zero-order. The zero-order discontinuities may be also caused by depth discontinuities of a single object. Other types of discontinuity are often caused by the surface textures of the moving objects. Continuous functions commonly refer to functions with continuous zero-order derivatives while smooth functions usually refer to those with higher-order continuous derivatives. The term, discontinuity , used here refers to zero degree discontinuity. A piece-wise continuous function is defined as a function which has all orders of derivatives almost everywhere except at a finite number locations of zero degree discontinuity. A very simple example of a piece-wise continuous function is depicted in Figure 4.1. The function consists of two pieces,

$$f(x) = \begin{cases} 21 - 0.02x & 0 \leq x < 50 \\ 22 - 0.12x & 50 \leq x \leq 140. \end{cases} \tag{4.1.1}$$

This function has a domain $D = [0, 140]$ and an internal boundary pixel at $x = 50$. It has well-defined derivatives of all orders except on the boundary points.

In the two dimensional case, the continuity of a function $f(x, y)$ is defined in terms of its directional partial derivatives. A line which intersects the point (x_0, y_0) has the form

$$\begin{aligned} y &= a(x - x_0) + y_0 \text{ , or} \\ x &= b(y - y_0) + x_0 \end{aligned} \tag{4.1.2}$$

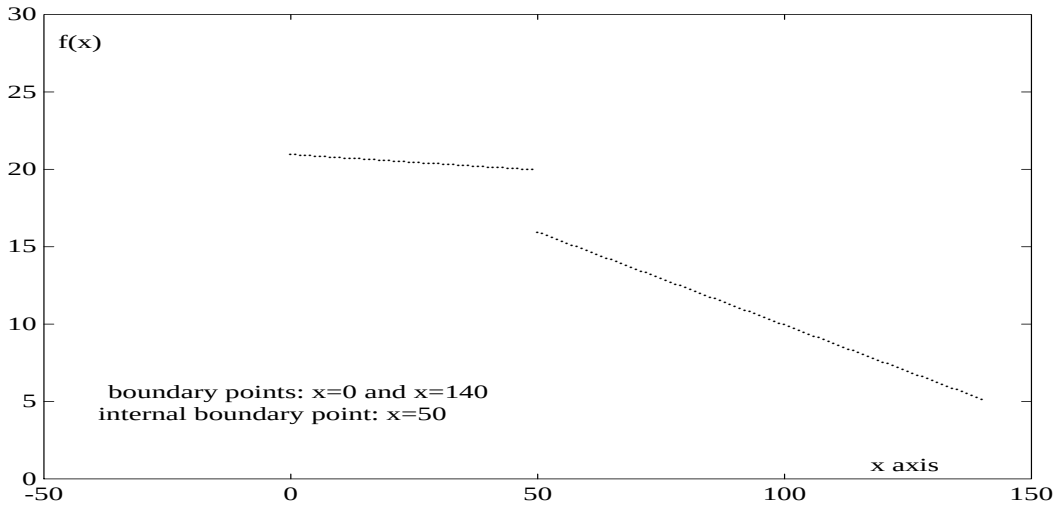

Figure 4.1: An example of piece-wise continuous functions

where a and b are real constants. If the set of one dimensional functions $f(x, a(x - x_0) + y_0)$ or $f(b(y - y_0) + x_0, y)$ is continuous at point (x_0, y_0) for any constants a or b, then the function $f(x, y)$ is said to be continuous at that point. A vector function of $\vec{f}(x, y) = (f_1(x, y), f_2(x, y), ..., f_n(x, y))$ is continuous at point (x_0, y_0) if all of its component functions are continuous at that point. The image motion field is a vector function of the form $\mathbf{u}(x, y) = (u(x, y), v(x, y))$.

4.2 Anisotropic regularization

4.2.1 One dimensional regularization

To introduce anisotropic regularization, we first examine the simpler problem of one dimensional regularization with discontinuities.

Given the measurements $g(x)$ with measurement errors being modeled by a noise term n_ε,

$$g(x) = f(x) + n_\varepsilon \tag{4.2.1}$$

for x in a domain D, we consider the problem of recovering the function $f(x)$ from data $g(x)$. The function $f(x)$ is assumed to be piece-wise continuous on D. The precise locations of the discontinuities are unknown. If the problem is seriously ill-conditioned due to the noise or incomplete measurements, regularization techniques are usually employed. As done in Chapter 3, a cost functional $I(f)$ is constructed and minimized for a given parameter λ,

$$I(f) = \int_D \Phi(f) + \lambda \Delta(f, g) dx \tag{4.2.2}$$

The first term is usual regularization functional or global smoothness constraint and often takes the form

$$\Phi(f) = (f^{(x)})^2$$

where $f^{(x)}$ is the first-order derivative of the function $f(x)$. The minimization of this term will result in a continuous approximation of function $f(x)$.

One disadvantage of this formulation is that the smoothness condition is applied throughout the whole function domain D by the global smoothness assumption of the solution. Also from the formulation in (4.2.2) , it is not clear that how the derivatives $f^{(x)}$ along the boundaries of D are defined, since by the definition, we have that

$$f^{(x)} \text{ exists if and only if } \begin{cases} f^{(x)}(x+) \ \& \ f^{(x)}(x-) \text{ exist, and} \\ f^{(x)}(x+) = f^{(x)}(x-). \end{cases}$$

For image data, the boundaries of D are the image frame borders. In Horn and Schunck's work [36], the derivatives along the image frame borders are simply copied from adjacent pixels further in from the image borders. For the same reason, the derivatives along motion boundaries have the same problem . In essence, it is assumed that the function to be estimated belongs to the class of continuous functions in the standard regularization scheme. The consequence of this is oversmoothing over motion boundaries.

Error-weighted regularization proposed in the preceding chapter can alleviate over-smoothing across motion discontinuities if the needed weights can be derived from

the local motion measurement process. The efficiency of error-weighted regularization scheme can be further improved by introducing a new notion of anisotropic smoothing or anisotropic regularization as will be discussed in Section 4.2.3.

4.2.2 Isotropic regularization functional

Upon to now, the best-known smoothness constraints or regularization functionals for image processing in the literature are isotropic in the sense that the neighboring information used to smooth a given pixel comes from all the neighboring pixels. When a pixel is close to a motion boundary, the neighboring pixels may come from different moving surfaces. In other words, smoothing across motion boundaries is unavoidable by using such regularization functionals. Nagel [58] has used the notion of "oriented smoothness " with a regularization functional Φ of the form

$$\Phi(\mathbf{u}) = trace((\nabla \mathbf{u})^T W^{-1} (\nabla \mathbf{u})) \tag{4.2.3}$$

where the matrix W is a function of first-order of intensity derivatives which can be viewed as projection operators that project any vector onto its component parallel or perpendicular to the local image gradient, and $(\nabla \mathbf{u})$ represents the first-order derivatives of image-motion field $\mathbf{u}$. In this formulation, the smoothness requirement would be retained only for the displacement vector component perpendicular to edges along significant gray value transitions. The smoothness requirement for the displacement vector component in the direction of the pixel intensity change would be suppressed. It is hoped that this component could be reliably determined from the pixel intensity change itself. Thus, Nagel's notion of oriented smoothness is weaker than the smoothness requirement of Horn and Schunk [36]. However Nagel's approach will not solve the problem of smoothing across the motion boundaries since neighboring information used to smooth the image-motion component may come from different moving surfaces.

4.2.3 Anisotropic regularization functional

The underlying assumption of anisotropic regularization is that each pixel of a piece-wise continuous function always belongs to a continuous surface and no isolated points exist for the function. If the function to be estimated is piece-wise continuous, then there always exists a consistent neighboring subset for each point on the whole function domain D.

For a one-dimensional piece-wise continuous function $f(x)$, at least, one of the one-sided derivatives $f^{(x)}(x-)$ or $f^{(x)}(x+)$ is always well defined for every $x \in D$. If $f^{(x)}(x-)$ or $f^{(x)}(x+)$ exists, the point x will be then defined on the continuous interval $(x-A, x)$ or $(x, x+A)$, respectively. Here A is a small positive constant. We say that all the points in the interval are consistent with point x. Therefore, we can define a new regularization functional Φ as

$$\Phi(f) = \begin{cases} (f^{(x)}(x-))^2 & \text{if } f^{(x)}(x-) \text{ exists,} \\ (f^{(x)}(x+))^2 & \text{otherwise.} \end{cases} \tag{4.2.4}$$

For piece-wise continuous functions, if both one-sided derivatives exist they must equal to each other. Thus, the new regularization functional (4.2.4) is well-defined everywhere on D.

To determine which one-side derivative exists in (4.2.4), we can assign a numerical value $\xi(x+)$ or $\xi(x-)$ to each point (x) corresponding to $f^{(x)}(x+)$ or $f^{(x)}(x-)$. These two numerical values can be calculated based on the information within a small interval $(x - A \leq x \leq x + A)$ and are called selective confidence measures since those values can provide information for the regularization functional to select the functional $\Phi(f)$ of (4.2.4). We will discuss this property in detail in Section 4.3. The confidence measure will always give higher values to the side where the function is smoother within that finite interval. That is, if $\xi(x-) > \xi(x+)$, then we assume that $f^{(x)}(x-)$ exists. Otherwise, $f^{(x)}(x+)$ will be assumed to exist. Therefore, $\Phi(f)$ in

(4.2.4) can be rewritten as

$$\Phi(f) = \begin{cases} (f^{(x)}(x-))^2 & \xi(x-) > \xi(x+) \\ (f^{(x)}(x+))^2 & \text{otherwise} \end{cases} \tag{4.2.5}$$

The anisotropic regularization functional defined in (4.2.5) differs from Nagel's "oriented smoothness" defined in (4.2.3) in that anisotropic regularization smooths each pixel by adaptively incorporating its neighboring pixels. It is clear that the smoothness requirement of anisotropic regularization can hold for piece-wise continuous functions but an isotropic smoothing requirement may not. Therefore, for each measurement $g(x)$ at location x, we can always smooth that local measurement towards either $(x - A, x)$ or $(x, x + A)$. If it happens that the point x is close to a boundary point, the smoothing action will not take place across the boundary.

A problem still remains with the functional (4.2.5). For the pixels within continuous surfaces, only part of the neighboring information is used in the smoothing. Therefore, one continuous surface might suboptimally be treated as several smaller continuous surfaces. The global information, therefore, may not be propagated effectively within continuous surfaces. A better choice of $\Phi(f)$ would be

$$\Phi(f) = \xi(x-)(f^{(x)}(x-))^2 + \xi(x+)(f^{(x)}(x+))^2. \tag{4.2.6}$$

That is, $\Phi(f)$ is chosen as a weighted sum of two squared one-sided derivatives. The weights $\xi(x\pm)$ are the selective confidence measures. When a point is on the boundary or close to a boundary, the two selective confidence measures will differ significantly and thus the functional $\Phi(f)$ will behave as defined in (4.2.5). Otherwise, $\Phi(f)$ will be similar to an isotropic functional since the two weights $\xi(x\pm)$ will be roughly similar.

4.2.4 Two-dimensional anisotropic regularization

In the two-dimensional case, such as in the case of image-motion field estimation, it is more convenient to formulate the anisotropic regularization functional by averaging

function values of neighboring subsets of a function on each pixel since the number of directional derivatives is not finite. To do this, the neighborhood of each pixel is first divided into M subsets denoted by $Q_m, m = 1, 2, ..., M$. For a piece-wise continuous motion field, the motion vector at a pixel is always consistent with the motion vectors of pixels belonging to one or more of $Q_1, ..., Q_M$. Thus each motion vector can be regularized by the motion vectors of consistent neighboring subsets if consistent neighboring subsets can be determined. A selective confidence measure is used to indicate the degree to which a candidate subset Q_i is a consistent subset. Define a new regularization functional $\Phi(\mathbf{u})$ as

$$\Phi(\mathbf{u}) = \|\mathbf{u} - \sum_{m=1}^{M} \xi_m \bar{\mathbf{u}}_m\|^2 \tag{4.2.7}$$

where $\bar{\mathbf{u}}_m$ is the local average motion vector within Q_m and ξ_m is the selective confidence measure of Q_m. A large selective confidence measure will smooth the image-motion vector $\mathbf{u}$ at the pixel (x, y) towards the average vector $\bar{\mathbf{u}}_m$ in minimizing $\Phi(\mathbf{u})$. For the pixels close to motion boundaries, the selective confidence measures for different subsets Q_i will differ significantly from each other, and the resulting smoothing will proceed away from the boundaries. Otherwise, the selective confidence measures will be roughly the same for each subset and the smoothing will be isotropic.

If we use the same data term $\Delta(\cdot)$ of (3.6.3) in Chapter 3, the cost functional to be minimized in the new scheme is

$$I(\mathbf{u}) = \sum_{x,y} [c_{max}(\mathbf{u} \cdot \hat{\mathbf{e}}_{max} - \mathbf{d} \cdot \hat{\mathbf{e}}_{max})^2 + c_{min}(\mathbf{u} \cdot \hat{\mathbf{e}}_{min} - \mathbf{d} \cdot \hat{\mathbf{e}}_{min})^2 + \|\mathbf{u} - \sum_{m=1}^{m=M} \xi_m \bar{\mathbf{u}}_m\|^2]. \tag{4.2.8}$$

The minimization of this cost functional can proceed as in Chapter 3. The resulting iteration equation of the solution is

$$\mathbf{u}^{k+1} = \bar{\mathbf{u}}_S^k + \frac{c_{max}}{c_{max} + 1}((\mathbf{d} - \bar{\mathbf{u}}_S^k) \cdot \hat{\mathbf{e}}_{max})\hat{\mathbf{e}}_{max} + \frac{c_{min}}{c_{min} + 1}((\mathbf{d} - \bar{\mathbf{u}}_S^k) \cdot \hat{\mathbf{e}}_{min})\hat{\mathbf{e}}_{min} \tag{4.2.9}$$

where $\bar{\mathbf{u}}_S^k$ is defined as

$$\bar{\mathbf{u}}_S^k = \sum_{m=1}^{M} \xi_m \bar{\mathbf{u}}_m^k. \tag{4.2.10}$$

4.3 Constructing selective confidence measure

The central issue in anisotropic regularization is how to derive an effective selective confidence measure for the functional Φ of (4.2.7). Here we will arrive at such a measure from a new matching approach, multiple off-centered sub-window matching (MOW) approach combined with the sum of absolute difference (SAD) criterion. This new local matching scheme not only provides a convenient selective confidence measure function but also can significantly improve the local measurement accuracy.

4.3.1 Motivating the SAD criterion

In image motion estimation schemes using a regularization framework, any local motion measurement errors will propagate into the global smoothing. It is therefore desirable to have accurate local motion measurements for a better final image motion field estimate.

There are essentially two sources of local measurement errors: image noise and local motion model error. The first error source is from imaging sensors, which we do not have control over. Our major emphasis for improvements to local motion measurements is therefore the second error source. Below we will discuss how the sum of absolute difference (SAD) matching criterion affects the local motion measurements.

Let $S(x, y)$ denote the set of pixels within a local matching window. In block-matching, it is implicitly assumed that all pixels in $S(x, y)$ are moving with the same translational velocity. This assumption is generally invalid for real images with other than purely translational motion. In the absence of motion discontinuities, the motion measurement errors caused by motion modeling error can be described statistically by Gaussian noise over a small local windows. In this case, we say that the pixels within the set $S(x, y)$ are consistent in the sense that they come from the same moving surface. At motion boundaries, the error distribution can no longer be adequately described by a Gaussian distribution since there will be a subset of pixels

$Q(x,y) \subset S(x,y)$ whose velocity vectors are not consistent with the velocity of the pixel (x,y) itself. This subset $Q(x,y)$ can be viewed as statistical outliers which, if not rejected or down-weighted, will cause incorrect measurements and motion boundary blurring. Usually, the extent of blurring is proportional to the sizes of both the outlying subset of pixels and the correlation window used in the matching.

The sum-of-squared difference (SSD) matching criterion, a quadratic objective function $\phi(\cdot) = (\cdot)^2$, implicitly is optimum when noise in the matching is statistically modeled as Gaussian and thus significantly penalizes large matching errors. But in the presence of multiple motions, this Gaussian model is no longer valid and the SSD criterion disproportionately weights the outliers and results in measurement errors. Robustness in this case can be achieved by adopting a more appropriate model of the error by absolute differences, i.e., $\phi(\cdot) = |(\cdot)|$ [40]. The matching criterion derived from this model is the sum of absolute difference (SAD) measure. Compared with SSD, the SAD error measure performs better in dealing with multiple motions within a matching window. When two motions are present in the matching window, less weights is given to the inconsistent subset $Q(x,y)$, and thus, allows a larger population of outlier pixels within a matching window.

4.3.2 Multiple off-centered sub-window (MOW) matching

The SAD criterion down-weights outliers relative to SSD but does not totally discard them. Thus there will still some local measurement errors by using SAD criterion in local motion measurements. A better option to improve local measurements is to discard those outlying pixels. A low-complexity scheme for realizing this is through using a multiple off-center sub-window (MOW) matching scheme which also exploits the properties of piece-wise continuous functions discussed in Section 4.1. Since each pixel on the domain D of a piece-wise continuous function is not isolated, there is always a subset of neighboring pixels which is consistent with that pixel.

In MOW, $S(x,y)$ is divided into M possibly overlapping subsets $Q_m(x,y)$, $m = 1,2,....M$. For each subset $Q_m(x,y)$, a correlation operation is conducted with the candidate motion vector that has the smallest matching error. Mathematically, we can write the matching error as a function of both displacement vector and Q_m,

$$\varepsilon^m_{x,y}(u,v) = \sum_{(i,j)\in Q_m(x,y)} \phi(E(x+i,y+j) - E(x+i+u,y+j+v)) \qquad (4.3.1)$$

where ϕ is an objective function corresponding the matching criterion used. The candidate vector which has the smallest matching error among all subsets is taken as the local motion measurement for the pixel (x,y).

We note that at least one of these subsets will be consistent with pixel (x,y) for all pixels in the image, even for the pixels on the motion boundaries.

The division of $S(x,y)$ into subsets $Q_m(x,y)$ can be accomplished by considering the geometric shapes of motion boundaries. One possible choice is to use off-centered sub-windows of the original centered window. For the experiments presented in the next section, we set the number of sub-windows, $M = 4$, which correspond to the upper half, lower half, right half and left half of the original centered windows. In principle, the MOW scheme can have different sub-window organizations.

As an illustration, we consider a matching problem for a one-dimensional image sequence. In this case, the number of subsets of the pixel x_0 in the image $E(x)$ is two, that is,

$$\begin{aligned} Q_{x_0-} &= \{E(x) : x \in (x_0 - A, x_0)\} \text{ and} \\ Q_{x_0+} &= \{E(x) : x \in (x_0, x_0 + A)\} \end{aligned} \qquad (4.3.2)$$

where A is the matching window size. Therefore, the local image motion measurement is obtained by performing two block-matchings for each pixel, i.e., $min\{\varepsilon_{x+}, \varepsilon_{x-}\}$, where

$$\varepsilon_{x\pm}(u) = \sum_{(i)\in Q_{x_0\pm}} \phi(E(i) - E(i+u)). \qquad (4.3.3)$$

The displacement with smaller matching error of the two matchings is taken as the local motion estimate of that pixel.

4.3.3 The selective confidence measure

The local motion measurements can be improved by combining MOW with the SAD criterion. This will be verified through experiments presented in the next section. In addition, the minimum matching error of each subwindow can be exploited to construct a selective confidence measure to be used in the regularization functional.

The selective confidence measure for anisotropic regularization is defined as a reciprocal of the minimum matching error corresponding to each subset match using multiple off-centered subwindows. From the matching error properties discussed in Section 3.3, some useful properties of selective confidence measure can be easily identified. For pixels close to boundaries, the minimum matching errors of different subsets will differ from one another. The matching error for the subset which contains the motion boundary will be significant and a small selective confidence measure will then result. For other subsets which are on the same object surface as the central pixels, a large selective confidence measure can be expected. To reduce the effect of image noise on the selective confidence measures, the maximum difference of the minimum matching errors over all sub-window matchings is also used in the computation of the selective confidence measures . Let

$$\delta = max_{i,j \in \{1,2,...,M\}} |\varepsilon^i - \varepsilon^j|, \qquad i \neq j$$

be the maximum absolute difference of any two minimum matching errors of the subwindow matchings. The selective confidence measure for each subwindow is defined as

$$\xi_m = \frac{\frac{1}{\varepsilon^m + \frac{c}{\delta}}}{\sum_{i=1}^{M} \frac{1}{\varepsilon^i + \frac{c}{\delta}}}, \qquad m = 1, 2, ...M \tag{4.3.4}$$

where c is a simple scaling factor. The denominator in (4.3.4) is used for normalization purposes. The term $\frac{c}{\delta}$ is used to enhance the selectivity when the maximum difference of the minimum matching errors is large and to weaken the selectivity otherwise. This can be explained as follows: when δ is large, the the term $\frac{c}{\delta}$ will be small. The

selective confidence measure of each subwindow will mainly be determined by its minimum matching error and thus a large difference among M selective confidence measures will result. This is the case when the pixel is close or at motion boundaries. On the other hand, when δ is small, the selective confidence measures will mainly be determined by $\frac{c}{\delta}$ and thus all ξ_m will be nearly the same. The selectivity will be less strong in this case, where a pixel is away from motion boundaries.

4.4 Experimental results

4.4.1 Comparison between multiple and single window matching

It is now shown that the multiple off-centered sub-window matching (MOW) together with the SAD criterion has better performance than a standard single centered window (SCW) local measurement scheme since the former takes motion boundaries into account. This is verified through two synthetic image pairs. In the implementation, the window radius N of the single centered window is set to 2. Thus there are 25 pixels in a single centered window and 15 pixels in an off-centered sub-window if we take the upper half, lower half, right half and left half of the original centered window as four sub-windows. In the figures shown, all the resulting flow fields are superimposed on the first image. The mean-squared error (MSE) and signal-to-noise ratio (SNR) used below are defined by Equations (3.7.2) and (3.7.3), respectively.

The first synthetic image pair (64X64) tested is the "Square " images shown in Figure 4.2, representing a diagonal translation of the square sinusoidal intensity pattern towards the lower-right corner (two pixels horizontally and vertically). By using pure translation, we exclude the effects of motion modeling error in the block-matching. Both image disocclusion and occlusion occur along the motion boundaries. The resulting image motion fields obtained by the standard SCW with both SSD and

Figure 4.2: Local motion field by SCW with SSD scheme

Figure 4.3: Local motion field by SCW with SAD scheme

Figure 4.4: Local motion field by MOW with SSD scheme

Figure 4.5: Local motion field by MOW with SAD scheme

SAD criteria are shown in Figure 4.2 and Figure 4.3, respectively. The quantitative comparison between two criteria is made in Table 4.1. From the results, we see that the SAD criterion obtains better local measurements than SSD.

Criteria	MSE(Upper)	MSE(Lower)	SNR(Upper)	SNR(Lower)
SSD	0.256	0.343	6.28 dB	4.21 dB
SAD	0.202	0.310	7.30 dB	4.65 dB
Improving	-0.054	-0.033	+1.02 dB	+ 0.44 dB

Table 4.1: SCW scheme: comparison between SSD and SAD for "Square". Upper: the left upper diagonal part of the images. Lower: the right lower diagonal part of the images.

An interesting discovery from observing the results is that the improvement in measurements mainly comes from disocclusion boundaries. For the occluded area, SAD does not perform much better than the SSD. This due to the fact that the erroneous pixels in the matching window represent more than half of the total pixels in the window when the considered pixel is occluded on the second image. In Table 4.1, the MSE and SNR values on the left upper diagonal part and right lower diagonal part of the image are separately listed. Note the disocclusion boundaries are on the upper part and the occlusion boundaries are on the lower part. The SNR improvement on the left upper diagonal part which contains the disocclusion boundaries is 1.02 dB and the SNR improvement for the right lower diagonal part is only 0.44 dB. This difference can also observed from the flow fields shown in Figures 4.2 and 4.3. Hence we can conclude that neither SSD nor SAD criteria can obtain reliable motion estimates along the occlusion regions by using the SCW scheme.

The quantitative comparisons for motion fields obtained by the MOW and SCW schemes are shown in Tables 4.2 and 4.3. The corresponding motion fields are shown

in Figures 4.4 and 4.5. Here N is still set to 2.

Criteria	MSE	MSE(Upper)	MSE(Lower)	SNR	SNR(Upper)	SNR(Lower)
SSD	0.144	0.010	0.134	11.41 dB	20.27 dB	8.30 dB
SAD	0.150	0.007	0.143	11.25 dB	21.73 dB	8.03 dB
Improving	+0.006	-0.003	+0.009	-0.16 dB	+1.46 dB	-0.27 dB

Table 4.2: MOW scheme: comparison between SSD and SAD for "Square". Upper: the left upper diagonal part of the images. Lower: the right lower diagonal part of the images.

From Table 4.2, we see that there are no significant differences in the performance of SSD and SAD criteria for the particular pair of images using the MOW scheme. Both offer significant improvement along both disocclusion and occlusion boundaries. Carefully observing the resulting fields, one can find that SAD does perform better than SSD on corner motion boundaries, since the support pixels for the 4 sub-windows may come from different moving surfaces.

Schemes	MSE	MSE(Upper)	MSE(Lower)	SNR	SNR(Upper)	SNR(Lower)
SCW	0.513	0.202	0.310	5.90 dB	7.30 dB	4.65 dB
MOW	0.150	0.007	0.143	11.25 dB	21.73 dB	8.03 dB
Improving	-0.363	-0.195	-0.167	+5.35 dB	+14.43 dB	3.38 dB

Table 4.3: SAD scheme: comparison between MOW and SCW schemes for "Square". Upper: the left upper diagonal part of the images. Lower: the right lower diagonal part of the images.

The comparison between two different schemes with the SAD criterion is made in Table 4.3. From Table 4.3, it is immediately evident that the MOW scheme provides

a significant improvement in the accuracy of the estimated motion field. The SNR value of the motion field obtained by MOW scheme nearly doubles the SNR value obtained by the SCW scheme.

To see the performance differences between SSD , SAD, MOW and SCW in more complex scenes, another pair of synthetic images (256X256) called "Disc" is used to test each scheme. The first image of the pair is displayed in Figure 4.6. The background pattern of the images translates two pixels horizontally to the left, and the circular foreground pattern zooms-in by 0.04 unit in pixel distance while rotating by 4 degrees. The resulting maximum displacement is 6 pixels. The actual image-motion field for the center pattern is not translational and motion boundaries are not at all straight. The estimated motion fields are displayed in Figures 4.6, 4.7, 4.8 and 4.9. The MSE and SNR error measures are listed in Table 4.4. The SAD criterion

schemes	SAD (MSE, SNR)	SSD (MSE, SNR)
MOW	0.326, 14.53 dB	0.593, 11.92 dB
SCW	0.959, 9.84 dB	1.394, 8.22 dB

Table 4.4: Comparison among SAD, SSD, MOW and SCW schemes for "Disc"

with the MOW scheme gives the best local motion measurements on the occlusion boundaries and other image regions. Particularly, when using the SCW scheme, the SAD criterion results in a 1.622 dB over the SSD criterion. In the MOW scheme, the SAD criterion results in a 1.016 dB improvement over the SSD criterion.

4.4.2 Anisotropic smoothing experiments

In the following examples, the stopping criterion for iteration of Equation (4.2.9) in the minimization is taken as

$$\frac{\sum \|\mathbf{u}_{k+1} - \mathbf{u}_k\|^2}{\sum \|\mathbf{u}_k\|^2} \leq 10^{-6}. \tag{4.4.1}$$

Figure 4.6: Local motion field by SCW with SSD scheme

Figure 4.7: Local motion field by SCW with SAD scheme

Figure 4.8: Local motion field by MOW with SSD scheme

Figure 4.9: Local motion field by MOW with SAD scheme

A pair of 1-d images is first tested. Two images $E_1(x)$ and $E_2(x)$ are shown in Figure 4.10. The images consist of two sinusoids: a stationary background with a wavelength of 30 pixels and a central moving part with a wavelength of 40 pixels, which translates to the right by 6 units. Note that occlusion occurs at the boundaries of the background and foreground waves. Gaussian noise with a zero mean and standard deviation of 2 is added to the images.

The cost functional to be minimized in this case is taken as

$$I(u) = \int_D \xi_-(u'_-)^2 + \xi_+(u'_+)^2 + \lambda(u-d)^2 \tag{4.4.2}$$

where d is the local motion vector, u'_-, u'_+ are the derivatives from left and right sides, and λ is a regularization constant which is set to 5. The discrete problem of (4.4.2) is

$$\begin{aligned} I(u) = \sum \xi_{i-}(u(i) - u(i-1))^2 + \\ \xi_{i+}(u(i+1) - u(i))^2 + \lambda(u(i) - d(i))^2 \end{aligned} \tag{4.4.3}$$

The minimization of this function leads to an iteration equation

$$u^{k+1}(i) = \frac{\lambda d + \xi_{i-}u^k(i-1) + \xi_{i+}u^k(i+1)}{1+\lambda}. \tag{4.4.4}$$

Note that u^{k+1} is a combination of local measurements and a weighted average of the neighboring values at the kth stage.

The local measurements of image motion are depicted in the Figure 4.11(a) which have large measurement errors along the image occlusion areas. The performance comparison between anisotropic regularization and isotropic regularization (where all the selective confidence measures are set equally) for image-motion estimation is shown, respectively, in Figures 4.11 (b) and (c). From the figures, one can see that anisotropic regularization performs remarkably well in the presence of the motion discontinuities. Table 4.5 lists the SNR values for local measurements as well as the motion fields estimated by anisotropic regularization and isotropic regularization

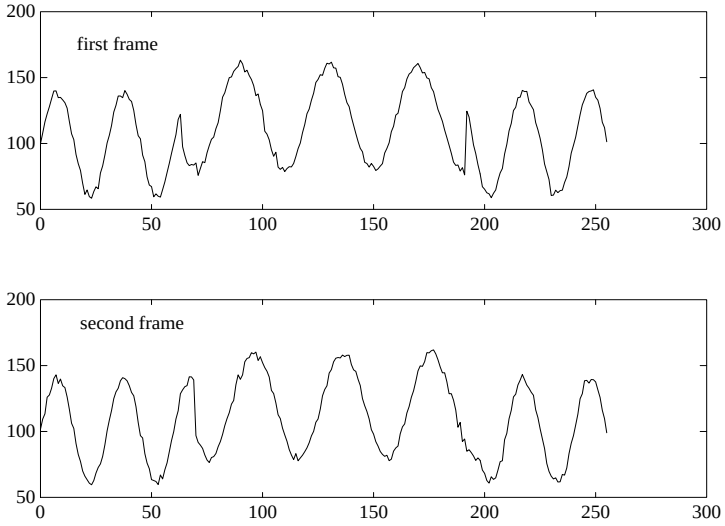

Figure 4.10: A one dimensional test image pair

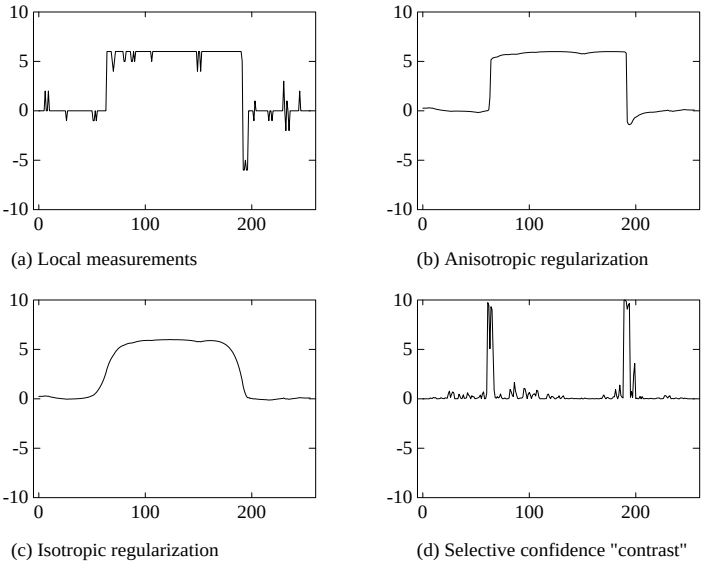

Figure 4.11: The experimental results for 1-d image pair

scheme	SNR	iteration number
Local measurement	13.05 dB	0
Isotropic regularization	14.78 dB	75
Anisotropic regularization	23.25 dB	41

Table 4.5: SNR values of different motion fields for 1-d images

schemes. From the table, we see that anisotropic regularization can provide more accurate image motion field using fewer iterations for the same stopping rule.

The Figure 4.11(d) shows the selective confidence "contrast" which is defined as $|\xi_{i+} - \xi_{i-}|$, the absolute difference of two selective confidence measures. As seen from Equation (4.4.4), the large "contrast" in motion boundary areas smooths the value of u^{k+1} toward the side with the higher selective confidence measure. Smoothing across motion boundaries is thus avoided.

In the two dimensional case, the implementation begins with the local motion measurement together with the selective confidence measure computation using (4.3.4). The estimation is then obtained by iterating (4.2.9) with the stopping rule (4.4.1). The performance comparison between anisotropic regularization and isotropic regularization for "Square" and "Disc" image pairs is shown in Tables 4.6 and 4.7, respectively. The isotropically and anisotropically smoothed motion fields are shown

Scheme	after smoothing	iteration number
Local measurements	11.25 dB	0
Isotropic regularization	8.25 dB	16
Anisotropic regularization	14.58 dB	4

Table 4.6: Comparison by SNR values for "Square" images

in Figures 4.12, 4.13, 4.14 and 4.15, respectively. In both cases, the motion boundaries obtained by isotropic scheme are oversmoothed, and thus the SNR values of the fields drop by 3 and 1.01 dB, respectively, for the "Square" and "Disc" image pairs. On the other hand, anisotropic regularization does not suffer from the problem of oversmoothing at motion discontinuities. The SNR values of the image-motion fields obtained by anisotropic regularization have gained 3.33dB and 4.75 dB, respectively, for the two cases over local measurements. Also note that anisotropic regularization uses fewer iterations than isotropic regularization for the same stopping rule.

Scheme	after smoothing	iteration number
Local measurements	14.53 dB	0
Isotropic regularization	13.52 dB	9
Anisotropic regularization	19.28 dB	4

Table 4.7: Comparison by SNR values for "Disc" images

4.4.3 Comparison between anisotropic and error-weighted regularizations

It is interesting to compare anisotropic regularization proposed in this chapter with the error-weighted regularization proposed in Chapter 3. Computationally, they have a similar global smoothing process except for the choice of weights used. For local motion measurement, the multiple window matching involved in the MOW scheme has twice as much computation as that in the SCW scheme if the MOW scheme uses four subwindows, since the number of pixels in a subwindow, $(2N+1)(N+1)$, is more than half that in the center window $(2N+1)^2$. The computation involved in computing the confidence measures is similar for both procedures.

The comparison results on the "Square" and "Disc" images are shown in Table

Figure 4.12: Smoothed motion field by standard regularization

Figure 4.13: Smoothed motion field by anisotropic regularization

Figure 4.14: Smoothed motion field by standard regularization

Figure 4.15: Smoothed motion field by anisotropic regularization

4.8. The local measurements performed by two algorithms differ significantly due to the different local matching schemes used. The global smoothing by error-weighted regularization has produced a larger gain than by anisotropic algorithm. But overall the anisotropic algorithm performs better than the error-weighed algorithm with twice as much computation as that used by error-weighted regularization in the local measurement.

In our serial processor implementation, for each pixel, the MOW scheme will have $(2d_{max}+1)(2N+1)^2$ multiplications more than that in the local matching used in error-weighted regularization. On the other hand, there are only 16 multiplications used in each iteration of Equation (3.6.6). When d_{max} and N are both set to 2 as in the example of the "Square" images, error-weighted regularization will take less computation than the MOW scheme as long as the number of global smoothing iterations is less than 8. In our "Square" example, only 4 iterations are used for error-weighted regularization. For larger d_{max}, the computation in the MOW scheme will cost even more.

Things would be different if fast and low-cost special-purpose hardware is available for block-matching. In this case one may consider using only local measurements by allowing some performance degradation. The global smoothing, however, can be performed in parallel-array machines with SIMD architectures very efficiently. As a

Scheme	local measurement	after smoothing	iteration number
Error-weighted (Square)	5.22 dB	12.81 dB	4
Anisotropic (Square)	11.25 dB	14.58 dB	4
Error-weighted (Disc)	8.21 dB	17.30 dB	4
Anisotropic (Disc)	14.53 dB	19.28 dB	4

Table 4.8: Comparison between anisotropic and error-weighted regularization

result, the error-weighted regularization algorithm may still require less overall computation than the MOW scheme. Therefore, the choice of regularization algorithms is dependent on the processor architecture used.

Chapter 5

OBJECT-ORIENTED CODING AND MDL PRINCIPLE

Based on the application-oriented viewpoint discussed in Chapter 1, a new formulation for estimating moving objects from the image sequence is presented with application to object-oriented image coding in the following two chapters. This chapter begins by motivating the use of *minimum description length* (MDL) principle to be used in a new image motion segmentation and estimation framework. A brief review of *motion-compensated predictive coding* then follows. Block-oriented and object-oriented coders are described, including several existing algorithms for moving object segmentation and estimation as they are essential to object-oriented image coding. Section 5.3 describes the MDL principle in detail and discusses the relationship between regularization and MDL estimation. Some previous applications of the MDL principle to intensity-based single image segmentation are also reviewed in this section. In the end of this chapter, the advantages of the MDL principle for moving object segmentation and estimation are summarized. Using the material presented in this chapter as background, a procedure for moving object segmentation and estimation is presented in Chapter 6.

5.1 Motivating the use of the MDL principle

In Chapters 3 and 4, we have developed two new regularization algorithms for the estimation of image-motion fields. From the experiments, we have demonstrated that these techniques can properly handle the motion discontinuities when multiple moving objects are present in the image sequences. As was demonstrated, the image motion fields estimated from the algorithms in Chapters 3 and 4 are more accurate than those from the standard regularization algorithms. However, two issues remain that are important in certain applications:

1. **The problem of choosing the optimal regularization parameter λ is still unsolved.** It is important to choose the regularization parameter optimally so that the regularized image motion fields are close to the true motion field. Since the regularization parameter is used to balance fitting the data and *a priori* knowledge about the solution, it will be very difficult to choose an optimal λ if the properties of the data and solution are not exactly known.

2. **The motion discontinuities are only preserved and not explicitly determined.** For certain applications such as moving object tracking or object-oriented coding, a further step of segmenting the image motion field [1, 56, 62] is often needed. In general, it is a computationally complex task to obtain such a segmentation from the estimated image motion field. For example, an algorithm that uses simulated annealing to perform scene segmentation from image motion field is presented in [56].

In addition to the above issues, the problems of integrating the requirements of a particular application of image motion field estimation into the optimization criterion is also addressed in this chapter. In Chapters 5 and 6 of this thesis, our application of image motion estimation is object-oriented image sequence coding. The minimum length description (MDL) principle is used as the optimization criterion

for the segmentation and estimation of moving objects in video image sequences. The criterion produced by the MDL principle, as we shall see below, is related to *maximum-likelihood* (ML) and *maximum a posteriori* (MAP) criteria, but is a more natural criterion when prior probabilities are not well-defined or the model structures (the number and order of models) are not constrained. By the using MDL principle, the data fitting and *a priori* knowledge can be more reasonably balanced since these two factors are closely related to a single quantity: the ideal coding length. This will be clearly shown in the following chapter where the motion parameters and boundaries of moving objects are estimated using an MDL framework that uses the ideal coding length as a cost function.

5.2 Block- and object-oriented image coding

In many applications of digital image processing, it is necessary to describe an image sequence in a compressed form. A typical example is the transmission of images. In these applications one can exploit correlation in space for still images and in both space and time for moving image sequences together with a certain degradation of image quality to achieve a low bit rate. Here we only consider inter-frame coding techniques which utilize the existing redundancy between image frames.

An important class of inter-frame coding schemes is motion-compensated predictive coding. For a large number of image transmission and storage applications, such as teleconferencing, videophone, television and satellite image transmission, etc., very high compression could be achieved if the trajectories of the moving objects are known. To achieve compression, one could simply code the initial frame together with the trajectory information of each set of pixels belonging to a moving object. Image coding schemes that predict the next frame using motion estimation are called motion compensated predictive coding algorithms.

The basic motion-compensated predictive coder consists of the following elements:

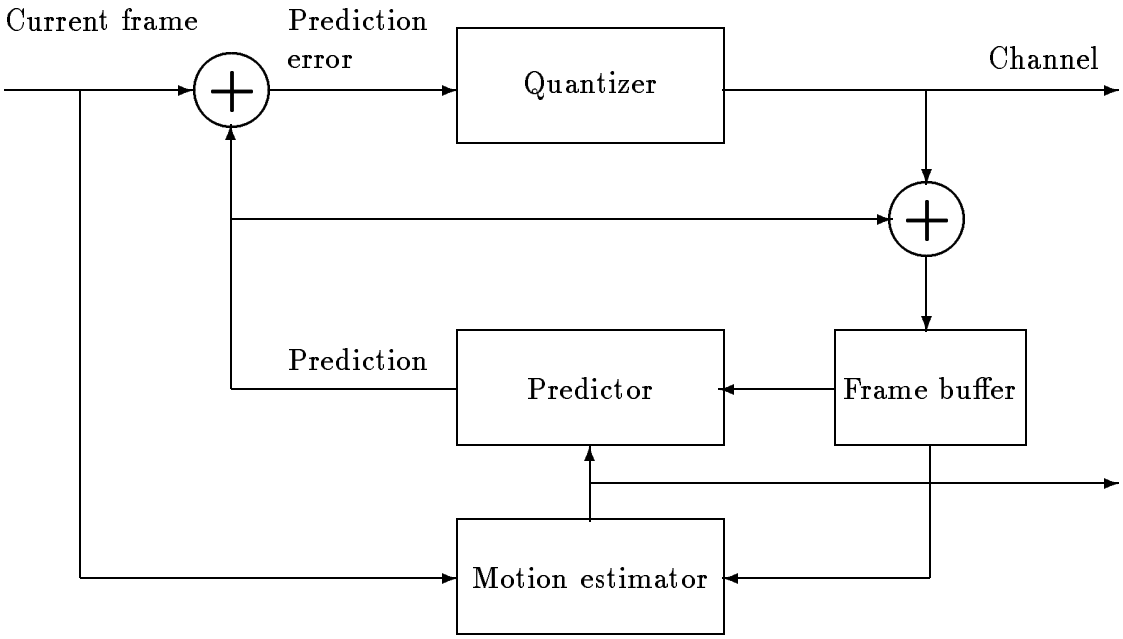

Figure 5.1: Motion-compensated predictive coder

1. Estimation of motion information of the moving objects from the image sequence,

2. Use of motion information to generate motion-compensated prediction error images,

3. A coding scheme for the prediction error and side information (motion estimates and/or segmentation).

A block diagram of a general motion-compensated predictive coder is shown in Figure 5.1. Motion estimator, predictor and quantizer blocks are the three basic elements of the coder. A key factor contributing to the success of motion-compensated predictive coding is motion extraction from the image sequences. The input to the motion estimator is the current frame and reconstructed previous frame which is stored in a frame buffer. The output of the motion estimator consists of motion vectors and segmentation information which are fed into the predictor and sent to the receiver.

The predictor produces a motion-compensated prediction of the current frame from the reconstructed frame based on the output of the motion estimator. The difference between the prediction image and the original image is then quantized and coded before being sent to the receiver.

5.2.1 Block-oriented coder

Traditionally, block-oriented motion estimation has been widely investigated due to its simplicity and effectiveness [9, 28, 64]. The method divides the image into fixed-size rectangular blocks and assumes that each block is undergoing independent uniform translation. For each block in the current frame a correlation of all possible blocks is performed within a search area in the previous frame. The best match is then found by minimizing a distortion measurement such as the sum of squared difference (SSD). In this scheme, the motion estimator in Figure 5.1 usually works on a 8-by-8 block basis, that is, each 8-by-8 block in the current frame is matched within a search window in the frame buffer. The motion vector that represents the offset between the current block and a block in the prior reconstructed frame that forms the best match is coded and sent to the receiver. In the block-oriented coder, the output of the motion estimator provides no motion segmentation information to the receiver other than the predetermined blocks.

The block-oriented motion-compensated predictive coder works quite well for typical videophone scenes, if the amount of motion is not large. In fact, it has been employed as an industry standard and special VLSI chips have been developed [64]. Nevertheless, block-oriented coding has some disadvantages. Since it only uses statistical dependencies, it does not explicitly consider the real (semantic) contents of the images. Thus, it does not try to understand or model the contents of the scene. The block boundaries and the boundaries of the objects in the scene normally do not coincide, because the blocks are not adapted to the image contents. This can lead to

visible distortion known as blocking and mosquito effects in low bit-rate coders [37]. For instance, if the boundary between two differently moving objects is in the middle of one block, the motion estimation will be unreliable and the predicted image will be of low quality. Furthermore, if there is a large region with homogeneous motion, this region might be split up into different blocks. For each block, motion information is transmitted separately which leads to information redundancy. Also, more complicated motion that includes rotation or zoom cannot be described correctly if only pure translation is considered. Another disadvantage is the fact that all blocks are treated equally. There is no distinction between more or less important parts of the scene. It would be desirable if important regions (e.g. the face of a person) could be extracted and then transmitted with a higher quality than less important parts (e.g. stationary background).

5.2.2 Object-oriented coder

To overcome the disadvantages of the block-oriented coder mentioned above, an object-oriented technique has recently attracted considerable attention in the field of image compression [37, 50, 62, 60, 81].

By object-oriented coding, moving objects are first extracted through the image sequence using more powerful algorithms than simple block matching techniques. The parameters describing the objects' motion and shape are encoded and transmitted along with motion-compensated prediction error (MCPE) images. Since the images are segmented into moving objects, fewer segments will result than in block-oriented schemes for most natural images. Furthermore, a segment's boundaries will coincide with the object's boundaries and artifacts in the reconstructed image will be greatly reduced. It is also easy to distinguish among all the moving objects extracted from the sequence and to assign different coding rates based on their importance to the overall quality of the images. These advantages make object-oriented coding a promising

approach to realize ultra-low bit rates and high quality image transmission.

The structure of a practical object-oriented coder is similar to Figure 5.1, that is, it still belongs to the class of hybrid coders. The output of the motion estimator will consist of both motion parameters and segmented contour information. These outputs are fed to the predictor and sent to the receiver. In the predictor, the image motion field between the current frame and reconstructed frame is first recovered from the motion parameters for all the moving objects and then this image motion field is used to predict the current frame. The difference between the predicted and the original frame is also quantized and coded.

Hotter [37] proposed an object-oriented colour image coder which encodes arbitrarily shaped objects instead of rectangular blocks. The objects are described by three parameter sets defining their motion, shape and colour. The colour parameters denote the luminance and chrominance values of the object surface. The parameters of each object are obtained by image analysis based on source models of moving 2-D objects and coded by a parameter coder with a mode control unit. The mode control unit decides whether or not the motion information is adequate to describe the objects. If the objects cannot be described by the models, only shape and colour information are transmitted. In his work, Hotter reported that the quality of the recovered image is drastically increased in image background areas which border on moving objects. Annoying mosquito effects are eliminated by the new coder concept.

5.2.3 Moving object estimation

Moving object segmentation and motion estimation is a key step to object-oriented coding. A better image analysis will result in a smaller encoding cost. Much research on this topic exists in the literature [12, 38, 62, 81]. There are two important components in the estimation of moving objects: one is the models used to describe the moving objects and other is the criterion used to estimate those model parameters.

In [60], a 3-d facial model is used to describe scenes in the object-oriented coding of videophone images. The transmitter and receiver have the same copy of a 3-d facial model. The motion parameters of the facial model are estimated by determining feature points on the face. The feature points are extracted manually in the leading image of the sequence and roughly tracked by block matching in succeeding frames. Templates of the eyebrows, eyes, and the mouth of the past frames are used for this block matching. Then, after thresholding the extracted blocks, the ends of the eyebrows, eyes and mouth are searched. This analysis strategy correctly traces feature points only under limited conditions. An optimal threshold needs to be determined interactively to match the illumination conditions. Since the extracted location of the feature points are unstable, the head tends to vibrate in the recovered image sequence. A time domain nonlinear filter is utilized to filter out this vibration.

Although 3-d object models are effective for reducing bit rates, at present it is still quite impractical to successfully estimate those 3-d models if the input image sequences are not confined to a particular type of scene. As a result, 2-d models have widely been used due to their simplicity and effectiveness.

Nicolas and Labit [62] have used four parameters in their 2-d motion model of moving objects. These four parameters are 2-d components of apparent translation, divergence ratio, and rotation angle. Their moving object estimation algorithm starts with an initial dense image motion field. A merge segmentation procedure, initialized by a spatial segmentation, is applied using a homogeneity criterion based on intensity and the initial image motion field. Two objects are merged if the combined object meets the homogeneity criterion. The motion parameters are refreshed for each merged object. The merging process stops when no hypothesized object meets the homogeneity criterion. Because the initial field is noisy, all the motion parameters from the merging process are refined by a steepest descent gradient method with regard to a cost function based on prediction errors. Promising results for motion

compensated predictive coding are obtained on TV image sequences.

Diehl [12] also proposed an objected-oriented motion segmentation and estimation scheme based on 2-d object model. He uses a hierarchical scene description approach. The image is first divided into changed and unchanged regions. The unchanged region is taken as a single stationary object. For the changed region, motion parameters are estimated using a modified Newton and quasi-Newton algorithm on the hypothesis that the changed region is a single moving object. After that, the changed region is split into two objects in the next lower level: one object that is consistent with the newly estimated motion parameters and the other object that has large motion-compensated predictive errors. The objects in the next lower level will be segmented by repeating the same procedures. Therefore, this algorithm is a type of top-down splitting algorithm. Also, the segmentation in each level heavily depends on some thresholding constants for intensity-edge based consistency checks.

5.3 MDL principle

The common drawback of present moving object estimation algorithms is that the choice of the optimality criterion does not directly match the requirements of the image sequence coding process. A better approach towards moving object estimation is to integrate the coding requirements into the criterion used for moving object estimation. The minimum description length (MDL) principle minimizes the coding cost for the data to be compressed, and therefore seems to be a natural choice to combine moving object estimation with image sequence coding.

The MDL principle was originally proposed by Rissanen [69, 70, 71]. MDL provides a framework for estimating both integer-valued structure parameters and real-valued parameters which specify a model for the data source. The principle is to use the least number of bits necessary to encode an observed data sequence $\mathbf{x}$ generated

by a stochastically modeled source. This principle leads to an optimal parameter estimator with minimum coding length for the observed data as the optimality criterion. The coding length obtained by such estimator corresponds to a notion of information in the data $\mathbf{x}$ relative to the class of models [70]. This notion of information consists of two terms: 1) Shannon's probabilistic notion of information, which describes the observations $\mathbf{x}$ generated by the stochastically modeled source and 2) Kolmogorov's algorithmic notion of information [79], which describes the nonrandom selection of the models or parameters. It is Kolmogorov's algorithmic notion of information that extends the classical maximum likelihood criterion and permits estimation of the number of parameters without a separate hypothesis test. This mixture model of information provides the common measure of complexity that can be assigned to both the data models and parameters.

5.3.1 Relation between estimation and coding

In the coding problem, we are given a string of observed data points x_t, $t = 1,...,n$, each truncated to some finite precision, and the objective is to redescribe the data with a suitably designed code as efficiently as possible, i.e., with a short coding length. In estimation, which is a fundamental problem in signal processing and related fields, we seek an explanation of the observations, or, rather, of the underlying mechanism, which we believe has generated the observed data. More precisely, we select a parametrically defined statistical model described by a probability mass function, $P_\theta(\mathbf{x})$, for the data $\mathbf{x}$, and try to estimate the vector parameter $\theta = (\theta_1,...,\theta_m)$ from the observations, where m is an integer variable to be estimated. The use of the probability mass functions is motivated by the fact that each observed realization x_i is always expressed in finite precision, with, say, q fractional binary digits.

By representing the number x_i in binary notation, we see that the entire sequence $\mathbf{x}$ can be written down using nq bits. But such a trivial coding or description of

the observed sequence does not take into account the possible correlations that exist between the numbers x_i nor the relative frequency with which each observation occurs. If such dependencies were taken advantage of, we might be able to reduce the total number of binary digits in the description of $\mathbf{x}$. The dependencies between data can often best described by a parametric model, and the coding length $L(\mathbf{x})$ of the data $\mathbf{x}$ will be a function of those model parameters. The shortest coding length should result if the true parameters are used in the code design.

Rissanen [69] has shown that a one-to-one relation exists between the coding length function $L(\mathbf{x})$ and the negative base-two logarithm of the probability mass function $P_\theta(\mathbf{x})$ used to describe the data model, i.e.,

$$L(\mathbf{x}) = -logP_\theta(\mathbf{x}) \tag{5.3.1}$$

where θ is a parameter vector which specifies a whole class of probability mass functions. If we pick just any "model" $P_\theta(\mathbf{x})$ in the class and encode the data $\mathbf{x}$ using $-logP_\theta(\mathbf{x})$ bits, then the mean coding length $-\sum P_{\theta^0}(\mathbf{x})logP_\theta(\mathbf{x})$, where the sum is over all data sequences $\mathbf{x}$ of length n and θ^0 denotes the "true" parameter, cannot be smaller than the entropy, which is defined as $-\sum P_{\theta^0}(\mathbf{x})logP_{\theta^0}(\mathbf{x})$. Moreover, equality is achieved only when $\theta = \theta^0$. Therefore, if the observed data sequence has probability $P_\theta(\mathbf{x})$ with θ regarded as fixed, then the minimum coding length for the observed data is $-logP_\theta(\mathbf{x})$ bits. This coding length is called the *ideal coding length.* If θ is variable, and we wish to design the shortest code, we clearly have to estimate θ so as to minimize the ideal coding length $-logP_\theta(\mathbf{x})$. This is an alternative interpretation of the familiar Maximum Likelihood (ML) estimator.

We have not yet considered the problem of obtaining a compact description of θ. Without any cost assigned to encoding the parameters we could, in principle, bring the the mean coding length $-\sum P_{\theta^0}(\mathbf{x})logP_\theta(\mathbf{x})$ as near to the entropy $-\sum P_{\theta^0}(\mathbf{x})logP_{\theta^0}(\mathbf{x})$ as we like by increasing the complexity of the model, i.e., the

dimension of θ. This is one reason why the correct model structure cannot be determined by the ML estimator. This problem can be solved by including the number of bits spent on encoding the parameters into the ideal coding length function. The interpretation of this solution can be identified with Maximum A Posteriori (MAP) estimation [51] as discussed below.

When we include the ideal coding length for the parameters into the total coding length function, we have

$$L(\mathbf{x}) = -logP_\theta(\mathbf{x}) + L(\theta) \tag{5.3.2}$$

where $L(\theta)$ denotes the ideal coding length for the parameters. The problem of efficiently encoding the model parameters, θ, is quite different from encoding the random observations $\mathbf{x}$ because θ can not be readily modeled by probability distributions. By similar arguments as used for the data model term, Rissanen has shown in [69] that $2^{-L(\theta)}$ defines a prior distribution function for the parameters under certain conditions. That is

$$P(\theta) = 2^{-L(\theta)}. \tag{5.3.3}$$

As examples, if the parameter is a constant vector known to the decoder, we will not need to encode it at the transmitter, so $L(\theta) = 0$ and $P(\theta) = 1$. If the parameter is known to range uniformly over a finite set of M values, then we will need $log(M)$ bits to encode it and with $P(\theta) = 1/M$.

Using (5.3.2) and (5.3.3), we can write the total coding length function as

$$L(\mathbf{x}) = -logP_\theta(\mathbf{x}) - logP(\theta) \tag{5.3.4}$$

or in a more familiar form

$$\begin{aligned} L(\mathbf{x}) &= -log[P_\theta(\mathbf{x})P(\theta)] \\ &= -log[P(\mathbf{x}|\theta)P(\theta)] \end{aligned} \tag{5.3.5}$$

On the other hand, the MAP criterion chooses the parameter vector θ that maximizes the conditional probability of the model, given the data: $P(\theta|\mathbf{x})$. An application of

Bayes's rule yields

$$P(\theta|\mathbf{x}) = \frac{P(\mathbf{x}|\theta)P(\theta)}{P(\mathbf{x})}. \tag{5.3.6}$$

Since $P(\mathbf{x})$ is constant with respect to θ, the MAP strategy is to choose θ that maximizes

$$P(\mathbf{x}|\theta)P(\theta).$$

From (5.3.5) and (5.3.6) , we see that the strategy of finding the minimum coding length by choosing particular model parameters is equivalent to the MAP strategy of maximizing the conditional probability for the model, given the data: $P(\theta|\mathbf{x})$.

5.3.2 Prior information and parameter coding

There are two sources of information in the estimation problem. The first consists of observed data $\mathbf{x}$, and the second, called prior information, consists of everything else, based on earlier observations that are no longer available to us or based on known properties of the data source. Prior information plays as crucial a role in the MDL criterion as in MAP estimation. We first need to know how the data are generated, that is, the prior information that is used to define an observation data model. Usually this is done by selecting a parametric class of probability mass functions, $P_\theta(\mathbf{x})$, and assigning a probability to every possible observed data $\mathbf{x}$. If the observations consist of both an "input" sequence $\mathbf{y}$ and an "output" sequence $\mathbf{x}$, then the appropriate probability mass function is $P_\theta(\mathbf{x}|\mathbf{y})$. Secondly, the prior information must be taken into account to derive the ideal coding length functions for model parameters.

Of particular interest is the case where the model parameters are integers. Suppose k is an integer to be coded and one knows that the number of bits in the binary representation of the integer equal to n. Then the coding length for k is simply n. That is, integer k has a uniform distribution over a finite range $[0, 2^n]$, i.e., $P(k) = 1/2^n$.

If one does not know the number of bits in the binary representation of this integer,

we can encode it by a simple but inefficient method which uses a sequence 01 as a "comma". This is done by repeating every bit of the binary expansion of k twice and then ending the description with a sequence 01 so that the decoder knows that the end of the code has come. For example, the number $k = 5$ (binary 101) would be encoded as 11001101. This code requires $2\lceil logk \rceil + 2$ bits.

A more efficient method for encoding k is through the following recursive procedure: at first, the number ($logk$) of bits in the binary representation of k is specified, followed by the actual bits of k. To specify $logk$, the length of the binary representation of k, we can use $loglogk$ bits. Continuing this recursively, we can encode k in $logk + loglogk + logloglogk + \cdots$ bits, summing until the last positive term. This sum of iterated logarithms is sometimes written as log^*k. The associated probability $P(k) = 2^{-log^*k}$ is known as a universal proper prior for the integers.

For encoding a real-valued vector parameter θ, without prior knowledge, we first truncate each component of θ,θ_i, to an integer number of bits and then encode the integer as above. The truncation performed is by writing each component θ_i of θ to precision $\pm\delta_i/2$. This allows for the precision of each component to be adjusted according to its contribution to the total coding length of the data $\mathbf{x}$.

Distributions of parameters other than uniform may also be considered. For instance, since asymptotically efficient estimators in general have a near Gaussian distribution, θ_i could be modeled as Gaussian. Also, we could assume that the observed data points come in batches of, say, N points each and we could specify a conditional probability $P(\theta^k|\theta^{k-1})$ of the parameter vector θ^k for the kth batch given the previous parameter θ^{k-1} for the $(k-1)$th batch in terms of prior knowledge of temporal correlations of the model parameters.

5.3.3 Data model structure

In many situations, the observed data to be encoded are generated by several underlying models rather than just a single model. In the modeling process, the observed data may not be adequately described even if a complicated single model is used. A simple example is the data string generated by the piece-wise continuous function defined by Equation (4.1.1) in Chapter 4.

In such situations universal modeling of the encoder is needed. In broad terms, the modeling of the observed data involves a determination of local structure within the entire data and its contexts. Thus we can regard the *model* as consisting of two parts: 1) the local structure which specifies the set of events and their contexts, and 2) the parameters which define the probabilities assigned to the local events.

The local structure captures the global redundancies while the parameters are tailored to each individual local structure. If we can estimate the local structure of the data and use a shorter model for each subset of the data, then a shorter coding length would be obtained even though additional bits are used to describe the local data structures. Generically, the process of finding local structures of the data is a segmentation problem for observed data. The number of local segments and its boundaries are the integer-valued structure parameters to be estimated.

The original MDL formulation did not consider this segmentation problem. We now extend MDL to the multiple model case by posing a combined segmentation and estimation problem. Let $O_n, n = 1, ..., N$ be a collection of disjoint subsets that partition the data. Let each subset be generated by a parametric model $P_{\theta(n)}(x|x \in O_n)$. If the prior probability distribution of the parameters for O_n is $P(\theta(n))$, then our combined segmentation and estimation problem under MDL is to estimate the number of subsets N, the points of b_n, representing boundaries, in subset O_n, and N parameter vectors $\theta(n), n = 1, ..., N$ together with their order number m_i such that

the coding length for $\mathbf{x}$, as expressed below, is minimized

$$\mathbf{L}(\mathbf{x}) = \sum_{n=1}^{N}[-logP_{\theta(n)}(x|x \in O_n) - logP(\theta(n)) + log^*m_n] + log^*N + \sum_{n=1}^{N} log^*b_n. \quad (5.3.7)$$

The last term in the above expression denotes the coding length for the boundary contour of the ith segment if the data $\mathbf{x}$ is in the form of a two-dimensional array.

5.3.4 Relationship between regularization and MDL

It is interesting to consider regularization theory discussed in Chapter 3 in the context of the MDL principle. First, regularization theory is usually used to estimate an unknown function from noisy data and the solution is obtained by minimizing a functional of the unknown function. The MDL principle treats the unknown functions as a class of parametrized functions. The cost functional in regularization theory is thus expressed in terms of a function of unknown parameters in the MDL estimation context. Therefore, the MDL estimator is a parameter estimator. Once the parameters for the unknown function are estimated, the unknown function can be easily reconstructed.

Regularization theory deals with *ill-posed* problems by adding a measure of the solution's *smoothness* requirement. The cost functional to be minimized thus consists of two terms: one term measures the fit to the data and the other term measures the smoothness of the estimates. Since these two measures are fundamentally different, an optimal combination of the two is obviously difficult to obtain. On the other hand, the two terms in the criterion produced by the MDL principle are similar in nature: the number of bits needed to encode both the prediction errors and the parameters which specify the unknown function. Therefore, the MDL principle can automatically balance between the smoothness and data fitting capabilities of estimators by using coding length as a common measure. From a coding point of view, the number of bits needed to encode the parameters will be smaller if the estimates become smoother. Therefore, regularization with smoothness constraints is roughly equivalent

to MDL estimation except that it lacks the automatic balancing capability between smoothness and and data fitting.

As an example, consider the classical curve-fitting problem, in which one is presented with an ordered set of numerical observations that can be described as points along some mathematically defined curve, such as a polynomial. A smoother fitting will result if a low order polynomial is used. By allowing the order of the polynomial to be variable, the smoothness and accuracy of curve fitting can be balanced by minimizing the coding length for describing the fitting errors and the parameters of the polynomial model. If the observations come from several curve segments, the MDL estimator can get a segmented fitting which has a minimal coding length among all other possible fittings.

5.3.5 Image segmentation by MDL

The general MDL principle discussed above has been proposed for some time [69, 70, 71], but it is only recently that researchers have found the MDL principle to be a powerful tool for image analysis. In [45, 51, 11], the MDL principle has been applied to intensity-based segmentations of still images.

Leclerc [51] uses the MDL criterion to segment single images. Using a low-order polynomial description of the intensity variation within each local region and a chain-code representation of the region boundaries, a global cost function is constructed in terms of coding length for the image. A local minimum of that cost function is obtained by an iterative descent algorithm after linearization of a system of normal equations. Successful image segmentations have obtained for two simple synthetic images. However, the algorithm fails to give a truly region-based description for the real images and behaves as an edge detector. The reason of this failure is that the system is essentially a form of surface reconstruction with "line process". The minimization of the cost function proceeds locally and no region labeling occurs.

In contrast to Leclerc's image analysis system, a region-based segmentation algorithm is proposed by Darrell [11]. The algorithm incorporates two mechanisms: 1) a cooperative estimation process, that produces a large set of hypotheses about the scene's local structure and 2) a global optimization process that searches for the subset of these hypotheses which constitutes the simplest and most likely global description of the image. In the cooperative estimation, an array of robust M-estimators is applied to the image. In the global optimization stage, the redundant estimates are eliminated by identifying hypotheses which overlap and offer little or no encoding reductions. Simulations are performed on simple synthetic 2-d and 3-d images. A region-based segmentation is obtained for each case. One problem of this algorithm is that it is not apparent as how to specify the initial region support for each estimator. Also the problem of estimating the number of objects is not properly addressed.

5.3.6 Summary of the advantages of MDL

From above discussion, it is clear that the notion of an ideal coding length function, (5.3.2), is a coding theoretic equivalent of a random process and the resultant estimator is equivalent to the MAP estimator. Although equivalent, the coding length interpretation is preferable due to the following reasons.

1. **The coding length interpretation is valid even when the objects to be coded are "deterministic" parameters, admitting no traditional probabilistic interpretation**. One of the advantages of the MDL approach is the uniform manner in which one can combine purely stochastic models (such as white noise) with deterministic models (such as the polynomials).

2. **The coding length interpretation conveniently handles both integer-valued structure parameters and real-valued model parameters**. Typically, integer-valued structure parameters can denote model order. In addition, integer-valued parameters describe the number of underlying models and local

data structures. This advantage has made the MDL a natural framework for the integration of parameter estimation and data segmentation.

3. **Using MDL, we can estimate the least number of bits that are needed to encode the observed data with regard to a particular data model.** When a particular coding scheme is specified, there is a natural trade-off between bits spent on model parameters and bits spent on data from that model. This feature is intuitively appealing when the purpose of the estimation problem is to encode the observed data.

4. By combining motion segmentation with motion parameter estimation, we can **estimate motion discontinuities more appropriately** in the context of the application at hand.

Based on the advantages mentioned above, The MDL principle is applied to the moving object segmentation and motion estimation in the next chapter. A computational procedure for the MDL estimator based on region merging is developed. The great potential of the MDL principle in source coding bit-rate reduction is demonstrated.

Chapter 6

SUMMARY AND CONCLUSION

This thesis addresses the problem of moving object estimation for image motion fields that contain discontinuities. We have approached the problem in two ways: adaptive regularization, which includes error-weighted and anisotropic regularization where segmentation is implicit, and the MDL principle, where the motion-based segmentation is explicit.

6.1 Image field estimation by regularization

6.1.1 Major results

The first contribution of the thesis, presented in Chapter 3 and 4, is the use of the matching errors in the regularization smoothing functional to adaptively smooth the motion fields.

Block-matching is used in the proposed regularization algorithms since the matching errors can be used for measuring the reliability of local measurements and for guiding the global smoothing process as discussed in Sections 3.2 and 3.5. To support this concept, Sections 3.3 and 3.4 discussed the exploitable connections between image-motion discontinuities and the matching errors among the different types of motion boundaries. Matching errors have also been used by Anandan and Singh but

only for measuring the reliability of local measurements. The usage of local matching information in the global smoothing process is novel, to the author's knowledge.

The key formulation of error-weighted regularization presented in Chapter 3 is the construction of the regularization functional Φ. As discussed in Section 3.6, Φ is taken as the L_2 norm of the difference between the motion vector $\mathbf{u}$ and the error-weighted average, $\bar{\mathbf{u}}$, of its neighboring motion vectors. The weights used here are the the inverses of the scaled matching-errors. By using this functional, global motion information will be propagated to each $\mathbf{u}$ only from neighboring points which have comparatively reliable measurements. As a result, oversmoothing across the motion boundaries is avoided.

By exploiting the properties of piece-wise continuous functions as discussed in Section 4.1 and noting the disadvantages of the isotropic smoothing functional as discussed in Section 4.2.2, the new concept of anisotropic regularization is proposed in this thesis.

In anisotropic regularization, multiple, spatially offset windows (MOW) are used for the local measurement process. The cost functional formulation of anisotropic regularization is similar to that of error-weighted regularization except for the new selective confidence measures defined in Equation (4.3.4) which are based on matching errors from MOW. The selective confidence measures are designed for the regularization functional to select the consistent neighboring motion information so that anisotropic regularization smooths each pixel adaptively.

Anisotropic regularization has improved motion estimates over error-weighted regularization as compared in Section 4.4.3. But much more computation is used in the MOW scheme. Therefore, the choice of regularization algorithms is dependent on the implementation architecture used.

Compared to stochastic optimization based schemes where a line process is used,

the major advantage of the two new regularization algorithms is that they are computationally similar to standard regularization as used by Horn and Schunck [36] and achieve improved performance as evidenced by the experimental results provided in Sections 3.7 and 4.4.

6.1.2 Future research in image motion estimation

Regularization parameters

The regularization parameters in anisotropic and error-weighted regularization algorithms have been chosen using Anandan's algorithm. In this method, regularization parameters are implicitly specified in terms of confidence measures. Even though the regularization parameters are spatially varying, the optimality of such a formulation has not been proven. A recent study of regularization parameter optimization in [80] in the context of image restoration may provide some insights into this problem. However, as noted in Section 3.1, a tradeoff exists between parameter optimization and computational considerations

Hierarchical approach

Multiple resolution image representation are widely used in image processing to speed up computation and allow for progressive transmission [3, 16, 24]. But the low pass filtering operation in resolution reduction often spreads occlusion and disocclusion regions and mixes the intensity information among different moving surfaces. It would be desirable to consider multiple resolution representations that properly avoid smoothing over motion boundaries.

Interpolation of interlaced video images

In Section 3.7.3, it has been found from the experiments that the motion-compensated interpolation error in interlaced video images is significant for even fields if the motion information is estimated from odd fields. This problem warrants further investigation in order to obtain the recovered images with a higher quality.

6.2 Moving object segmentation and estimation by MDL

6.2.1 Major results

A significant contribution of the thesis, presented in Chapters 5 and 6, is the application of the minimum description length (MDL) principle to motion segmentation and estimation on scenes with multiple moving objects. The importance of this contribution is evident by increasing demands for ultra low bit-rates for video image transmission. The recent concept of object-oriented image sequence coding is reviewed in Section 5.2. This thesis has suggested a new direction in the area of moving object estimation for applications in object-oriented image sequence coding.

As discussed in Section 5.1, the MDL criterion is motivated by its unified treatment of the data fitting and parameter model terms found in a regularization framework based on the ideal coding length and the explicit motion boundary representation. More importantly, the MDL criterion is well-suited to object-oriented image sequence coding, since the ideal coding length is minimized for a given distortion.

The aim of the MDL principle originally proposed by Rissanen is to derive a criterion estimating both integer-valued structure parameters and real-valued parameters of a stochastic model. The principle is to use the least number of bits necessary to encode an observed data sequence generated by a stochastically modeled source. In Sections 5.3, the MDL principle is described, and together with its relation to estimation and coding, prior information, parameter coding and local data structures. To use the MDL principle for moving object segmentation and estimation, this thesis has straightforwardly extended the MDL principle to multiple data models as described in Section 5.3.3. Existing applications of the MDL principle to intensity-based single image segmentation are reviewed in Section 5.3.5. The present work is an important extension of those early applications.

The formulation of the new MDL estimator, presented in Section 6.1, uses a motion-compensated coding system model for scenes with multiple moving objects as a basis, and systematically establishes the ideal coding length functions for motion parameters, motion boundaries and motion-compensated prediction errors. In the implementation, the motion models are either linear affine or translational. Each motion parameter is assumed to have a uniform distribution and is encoded by six bits. A chain-coding scheme is assumed for representing object boundaries due to its simplicity. A Gaussian distribution is assumed for the motion-compensated prediction errors. The quantization constant introduced in this Gaussian distribution determines the visual quality of recovered images by a decoder.

The minimization of the MDL estimator turns out to be difficult. This thesis has proposed a solution to the problem. The importance of this computational procedure is that it provides a means to systematically quantify the great potential of object-oriented image coding over block-oriented schemes. As discussed in Section 6.2, the proposed computational procedure is based on a region-merging scheme. The image is first divided into disjoint blocks. The local motion vectors from block matching then are then used to obtain a coarse segmentation. Based on this initial segmentation, coding length reduction is used to direct the merging process using an adjacency graph. The motion parameters within object boundaries are estimated based on solving a linear system of equations containing spatial and temporal intensity derivatives. The computational complexity of the merging procedure is proportional to the object size, the square of the number of objects in the sequence, and the square of the motion model order.

Although the solution of the MDL estimator is not globally optimal, several experimental comparisons presented in Section 6.3 between the block-oriented and object oriented coding schemes verify the further coding rate reduction ability of object-oriented coding schemes using this computational procedure. As opposed to fixed

block-oriented coding, the new procedure's performance improves with decreasing block-sizes. We have also shown experimentally that the affine motion model may be more suitable for complex scene motions than the purely translational model.

6.2.2 Future research in the MDL estimator

Motion model order detemination in MDL estimator

In the present MDL estimator, we have only considered a fixed-order motion model. A variable-order motion model can be readily accommodated using the MDL principle. Moving objects undergoing complex motion can therefore be described by more parameters. An efficient computational algorithm should be found to incorporate this formulation other than by simply hypothesizing all possible motion model orders for each moving object.

Computation of MDL estimator

The present merging scheme for the MDL estimator cannot be computed quickly enough for processing video image sequences on-line. Optimizing the graph data structures used in the merging process will reduce the computation required somewhat. To speed up the computation further, we might sacrifice, to some degree, the optimality of the MDL estimator's solution and use parallel merging schemes designed with special-purpose hardware such as used in [82].

Recursive estimation

The present MDL estimator only utilizes two frames at a time. Recursive estimation in the temporal direction will make more efficient use of information contained in the image sequence. The current segmentation and estimation of moving objects may potentially be performed much more efficiently by incorporating information in previous frames. However, the straightforward approach described in Section 6.4.2 was not as successful as expected. Therefore, recursive estimation is still an open problem.

Bibliography

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

[2] J.K. Aggarwal and N. Nandhakumar. " On the computation of motion from sequences of images - a review" *Proc IEEE,* Vol. 76, No. 8, pp. 917-935. 1988.

[3] P. Anandan. "A computational framework and an algorithm for the measurement of visual motion." *International journal of computer vision.* Vol. 2, pp. 283–310, 1989.

[4] J.L. Barron, D.J. Fleet, S.S. Beauchemin and T.A. Burkitt. "Performance of optical flow techniques" *Proceedings of CVPR'92* pp. 236-242, 1992.

[5] C. Bergeron and E. Dubois. "Gradient-based algorithms for block-oriented map estimation of motion and application to motion-compensated interpolation." *IEEE Trans. on Circuits and Systems for Video Technology*, 1(1), pp. 72–85, 1991.

[6] M. Bertero, T. Poggio and V. Torre. "Ill-posed problems in early vision," *Proc. IEEE,* Vol. 76, pp. 869-889, 1988.

[7] J. Besag. "On the statistical analysis of dirty pictures, " *J. Royal Statist. Soc.,* Vol. 48, No. 3, pp. 259-302, 1986.

[8] A. Blake and A. Zisserman. **Visual Reconstruction.** Cambridge, MA: MIT Press, 1987.

[9] S. Brofferio and F. Rocca. "Interframe redundancy reduction of video signals generated by translating objects" *IEEE Trans. Commun.,* Vol. COM-25. pp. 448-455, 1977.

[10] P.J. Burt, E. Adelson. "The Laplacian pyramid as a compact image code." *IEEE Trans. Commun.,* COMM-31, pp. 532-382, 1983.

[11] T. Darrell, S. Sclaroff and A. Pentland. "Segmentation by Minimal Description" *Third International Conference on Computer Vision*, pp, 121-125, 1990.

[12] N. Diehl. "Object-oriented motion estimation and segmentation in image sequences." *Signal Processing: Image Communication.* Vol. 3, pp. 23-56, 1991.

[13] E. Dubois and S. Sabri. "Noise reduction in image sequences using motion-compensated temporal filtering." *IEEE Trans. Commun.*, 32(7), pp. 826–831, 1984.

[14] E. Dubois. "The sampling and reconstruction of time-varying imagery with application in video systems." *Proc. of the IEEE*, 73(4), pp. 502–522, 1985.

[15] M. Eden and M. Kocher. "On the performance of a contour coding algorithm in the context of image coding. Part I: Contour segment coding", *Signal processing*, Vol. 8, No. 4, pp. 381-386, 1985.

[16] W. Enkelmann. "Investigations of multigrid algorithms from the estimation of optical flow fields in image sequences." *Proc. IEEE Computer Society Workshop on Motion: Representation and Analysis*, pp. 81-87, 1986.

[17] C. S. Fuh and P. Maragos. "Affine models for image matching and motion detection" *Proceedings of ICASSP 91* Vol. 4, pp.2409-2412, 1991.

[18] P. W. Fung, G. Grebbin and Y. Attikiouzel. "Model-based region growing segmentation of textured images *Proc. ICASSP*, pp. 2313–2316, 1990.

[19] D.J. Fleet. **Measurement of Image velocity** Ph.D. Dissertation, Department of Computer Science, University of Toronto, October, 1990.

[20] D.J. Fleet and A.D. Jepson. "Velocity extraction without form interpretation" *Proceedings of the Third Workshop on Computer Vision: Representation and Control*, (Bellaire,MI), pp. 179-185, 1985.

[21] N.P. Galatsanos and A.K. Katsaggelos. "Cross-validation and other criteria for estimating the regularizing parameter" *Proc. ICASSP* , pp. 3021-3024, 1990.

[22] S. Geman and D. Geman. "Stochastic Relaxation, Gibbs Distribution, and the Bayesian Restoration of images," *IEEE Trans. Pattern Anal. Machine Intell.,* Vol. PAMI-6, pp. 721-741. 1984.

[23] F. Girosi, A. Verri and V. Torre. " Constraints for the computation of optical flow". *Proc. Workshop on Visual Motion* pp.116-124, 1989.

[24] F. Glazer. **Hierarchical motion detection** Ph.D. Dissertation, COINS Department, University of Massachusetts, Amherst, MA, Feb., 1987.

[25] R. C. Gonzalez and P. Wintz. **Digital Image Processing** Addison-wesley publishing company, Second Edition, 1987

[26] N.M. Grzywacz, J.A. Smith and A.l. Yuille. "A common theoretical framework for visual motion's spatial and temporal coherence." *Proc. Workshop on Visual Motion* pp. 148-155, 1989.

[27] R.M. Haralick and J.S. Lee. " The facet approach to optical flow." In *Proc of Image Understanding workshop.* pp. 84-93. 1983.

[28] B.G. Haskell. "Entropy measurements for non-adaptive and adaptive frame-to-frame linear predictive coding of video telephone signals." Bell Syst. Tech. J. Vol. 54, pp. 1147-1155, 1975.

[29] D.J. Heeger. "Optical flow using spatiotemporal filters" *International Journal of Computer Vision* Vol. 1, pp. 279-302, 1988.

[30] D.J. Heeger and A. Jepson. " Simple method for computing 3D motion and depth", *Third International conference on computer vision* pp. 96-100, 1990.

[31] F. Heitz and P. Bouthemy. "Multimodal motion estimation and segmentation using Markov Random Fields," *Proc. Int. Conf. of Pattern Recognition.* pp. 378-383. 1990.

[32] F. Heitz and P. Bouthemy. "Motion estimation and segmentation using a global Bayesian approach." *Proc. ICASSP* , pp. 2305–2308, 1990.

[33] F.B. Hildebrand. **Introduction to Numerical Analysis** McGraw-Hill, New Tork, 1974.

[34] E.C. Hildreth. **The Measurement of Visual Motion. Cambridge**: MIT Press, 1984.

[35] E.C. Hildreth. "Edge detection," MIT Artificial Intelligence Laboratory A.I. Memo 858, 1985.

[36] B. K. Horn and B.G. Schunck. "Determining optical flow." *Artificial Intelligence*, pp. 185–203, 1981.

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

[38] M. Hötter and R. Thoma. "Image segmentation based on object oriented mapping parameter estimation". *Signal Processing* Vol. 15, No. 3, pp. 315-334, 1988.

[39] T.S. Huang editor. **Image sequence analysis** Springer-Verlag, 1981.

[40] P. Huber. **Robust Statistics** Wiley, 1981.

[41] J. Hutchinson, C. Koch, J. Luo, and C. Mead. "Computing motion using analog and binary resistive networks." *Computer*, Vol. 21, pp. 52–63, 1988.

[42] K. Ikeuchi and B.K.P. Horn, "Numerical shape from shading and occluding boundaries," *Artificial Intelligence*, Vol. 17, pp. 141-184, 1981.

[43] J. R. Jain and A. K. Jain. "Displacement measurement and its application in interframe image coding " *IEEE Trans. on Communications.* Vol. com-29, No. 12, pp. 1799-1808, 1981.

[44] D. S. Kalivas, A. A. Sawchuk and R. Chellappa. " Segmentation and 2-d motion estimation of noisy image sequences" *Proceedings of ICASSP 88* Vol. 2, pp. 1076-1079. 1988.

[45] D. Keren, R. Marcus, M. Werman and S. Peleg. " Segmentation by minimum length encoding" *10th ICPR* NJ, Vol. 1, pp. 681-683, 1990.

[46] T. Koga, K. Linuma, A. Hirano, Y. lijima and T. Ishiguro. " Motion-compensated interframe coding for video conferencing" in *Conf. Rec., Nat. Telecomm. Conf.,* pp. G5.3.1-G5.3.5, 1981.

[47] J. Konrad. **Bayesian estimation of motion fields from image sequences**. PhD thesis, McGill University, Montreal, Canada, 1989.

[48] J. Konrad and E. Dubois. " Estimation of image motion fields: Bayesian formulation and stochastic solution," *Proc. IEEE Int. Conf. Computer Vision ICCV'88,* pp. 354-362. 1988.

[49] J. Konrad and E. Dubois. "Bayesian estimation of vector motion fields." *IEEE Transactions on PAMI*, vol. 14, No. 9, 1992.

[50] M. Kunt, A. Ikonomopoulos and M. Kocher. " Second generation image-coding techniques." *Proc. IEEE,* Vol. 73, No. 4, pp. 549-574, 1985.

[51] Y. G. Leclerc. " Constructing simple stable description for image partitioning " *International journal of computer vision.* Vol. 3, pp. 73-102, 1989.

[52] D. Lee and T. Pavlidis. "One dimensional regularization with discontinuities," *IEEE Trans. PAMI,* Vol. 10(6), pp. 822-829, 1988.

[53] J.J. Little and A. Verri. "Analysis of differential and matching methods for optical flow", *Proc. Workshop on Visual Motion* pp. 173-180, 1989.

[54] S. Liu and M. Hayes. "Segmentation-based coding of motion difference and motion field images for low bit-rate video compression". *Proc. ICASSP*, Vol. 2, pp. 525–528, 1992.

[55] A. Mitiche, Y.E. Wang and J.K. Aggarwal. " Experiments in computing optical flow with the gradient-based multiconstraint method," *Pattern Recognition.* Vol. 20, No,2, pp. 173-179, 1987.

[56] D. W. Murray and B. F. Buxton. "Scene segmentation from visual motion using global optimization" *IEEE Trans. on PAMI.* Vol. PAMI-9, No. 2, pp. 220-228. 1987.

[57] H. Nagel and W. Enkelmann. " An investigation of smoothness constraints for the estimation of displacement vector fields from image sequences", *IEEE Transactions on PAMI*, Vol. 8, No. 5, 1986. pp.565-593.

[58] H. Nagel. "On the estimation of optical flow: Relations between different approaches and some new results." *Artif. intell.* Vol. 33, pp. 299-324, 1987.

[59] H. Nagel. " Displacement vectors derived from second-order intensity variations in image sequences." *Computer Vision, Graphics and Image Processing.* Vol. 21, pp. 85-117, 1983.

[60] Y. Nakaya, Y.C. Chuah and H. Harashima. "Model-based/waveform hybrid coding for videotelephone images" *Proc. ICASSP*, pp. 2741–2744, 1991.

[61] A. N. Netravali A.N. and J. D. Robbins. "Motion-compensated television coding: Part 1." *Bell System Technical Journal*, 58(3), pp. 631-670, 1979.

[62] H. Nicolas and C. Labit. "Global motion identification for image sequence analysis and coding" *Proceedings of ICASSP 91* Vol. 4, pp. 2825-2828, 1991.

[63] J. Pavlidis and S.L. Horowitz. "Picture segmentation by a directed split and merge procedure," in**Computer Methods in Image Analysis** Duda, Rosenfield and Aggarwal. Eds. New York: IEEE, pp. 101-110, 1977.

[64] H. Peng, A. Peter, and D. Auld. "Video compression makes big gains" *IEEE Spectrum*, October, 1991.

[65] E. Persoon and K.S. Fu. "Shape discrimination using Fourier descriptors" *IEEE Trans. Systems Man Cybernet.,* Vol. SMC-7, pp. 170-179, 1977.

[66] T. Poggio, H. Voorhees and A. Yuille. " A regularized solution to edge detection," MIT Artificial Intelligence Laboratory AI. Memo 773, 1984.

[67] H.V. Poor. **An introduction to signal detection and estimation** Springer-Verlag, 1988.

[68] W.H. Press *at al.* **Numerical recipes in C:** the art of scientific computing Cambridge University Press, 1988.

[69] J. Rissanen. " Minimum-description-length principle". *Encyclopedia of Statistical Sciences*, Vol. 5, pp. 523-527, 1987.

[70] J. Rissanen. "Universal coding, information, prediction, and estimation" *IEEE Trans. on Inform. Theory,* Vol. It-30, No. 4, 1984.

[71] J. Rissanen. "Modeling by shortest data description" *Automatica,* Vol. 14, pp. 465-471, 1978.

[72] D.L. Russell. **Calculus of variations and control theory** Academic Press, New York, 1976.

[73] P. Saint-Marc, J.S. Chen, and G. Medioni. "Adaptive smoothing: a general tool for early vision." *IEEE Trans. PAMI*, 13(6), pp. 514-529, 1991.

[74] A. Singh. "An estimation-theoretic framework for image-flow computation", *Third international conference on computer vision* pp. 168-177, 1990.

[75] A. Singh. "Incremental Estimation of Image Flow Using a Kalman Filter", *Journal of Visual Communication and Image Representation*, Vol. 3, No. 1, pp. 39-57, 1992.

[76] W.E. Snyder, S.a. Rajala and G. Hirzinger. "Image modeling, the continuity assumption and tracking." In *Proc. Int. Conf. of Pattern Recognition.* pp. 1111-1114, 1980.

[77] A. Spoerri A. and S. Ullman. "The early detection of motion boundaries." *First International Conference On Computer Vision*, pp. 209-218, 1987.

[78] R. Srinivasan and K.R. Rao. "Predictive coding based on efficient motion estimation" in *Conf. Rec., Int. Conf. Commun.,* pp. 521-526, 1984.

[79] M. Thomas and J. Thomas. **Elements of Information Theory** A Wiley-Interscience publication *John Wiley & Sons, Inc*, 1991.

[80] A.M. Thompson *at al.* "A study of methods of choosing the smoothing parameter in image restoration by regularization" *IEEE Trans. on PAMI, Vol 13*, No. 14, 1991

[81] Y. T. Tse and R. L. Blaker. "Global zoom/pan estimation and compensation for video compression". *Proceedings of ICASSP 91* Vol. 4, pp. 2725-2728, 1991.

[82] A. Tyagi and M.A. Bayoumi. "Image segmentation on a 2-D array by a directed split and merge procedure" *IEEE Trans. Signal Processing*, Vol. 40, No. 11, 1992

[83] S. Ullman. *The Interpretation of Visual Motion.* The MIT Press, 1979.

[84] K. Wohn and A.M. Waxman. "The analytic structure of image flows, deformation and segmentation" *Computer Vision, Graphics, and Image Processing* , Vol. 49, pp. 127-151, 1990.

[85] M. Yamamoto. "A general aperture problem for direct estimation of 3-D motion parameters" *IEEE Trans. PAMI*, 11(5), pp. 528-536, 1989.

[86] H. Yamaguchi et al. "Movement-compensated frame-frequency conversion of television signals." *IEEE Trans. Commun.*, 35(10), pp. 1069-1082, 1987.

VITA

Heyun Zheng

EDUCATION:

- M. Eng in Electrical Engineering
  Southwestern JiaoTong University
  Sichuan, P.R.C.
  September, 1982 to March, 1985.

- B. Eng in Electrical Engineering
  Hunan University, Hunan, P.R.C.
  September, 1978 to March, 1982

EXPERIENCE:

- Research assistant, Department of Electrical Engineering
  Southwestern JiaoTong University, Sichuan, P.R.C.
  September, 1985 to October, 1988

PUBLICATIONS:

- Heyun Zheng and Steven D. Blostein "Anisotropic Regularization for Image-Motion Field Estimation" *International Conference on Signal Processing '93/Beijing* October, 1993. Beijing, P.R. China

- Heyun Zheng and Steven D. Blostein "An Error-Weighted Regularization Algorithm for Image Motion Field Estimation" *IEEE Trans. on Image Processing* April, 1993, pp. 246-252.

- Heyun Zheng and Steven D. Blostein "Adaptive regularization for motion field estimation" *Sixth European Signal Processing Conference* Brussels, Belgium, August, 1992, pp. 1327-1330.