UMI Number: 9712640

UMI
300 North Zeeb Road
Ann Arbor, MI 48103

ABSTRACT

MOTION ESTIMATION AND VIDEO CODING

by
Xiaochun Xia

Over the last ten years. research on the analysis of visual motion has come to play a key role in the fields of data compression for visual communication as well as computer vision. Enormous efforts have been made on the design of various motion estimation algorithms.

One of the fundamental tasks in motion estimation is the accurate measurement of 2-D dense motion fields. For this purpose. we devise and present in this dissertation a multiattribute feedback computational framework. In this framework for each pixel in an image. instead of a single image intensity. multiple image attributes are computed as conservation information. To enhance the estimation accuracy. feedback technique is applied. Besides. the proposed algorithm needs less differentiation and thus is more robust to various noises. With these features. the estimation accuracy is improved considerably. Experiments have demonstrated that the proposed algorithm outperforms most of the existing techniques that compute 2-D dense motion fields in terms of accuracy.

The estimation of 2-D block motion vector fields has been dominant among techniques in exploiting the temporal redundancy in video coding owing to its straightforward implementation and reasonable performance. But block matching is still a computational burden in real time video compression. Hence. efficient block matching techniques remain in demand. Existing block matching methods including full search and multiresolution techniques treat every region in an image domain indiscriminately no matter whether the region contains complicated motion or not. Motivated from this observation. we have developed two thresholding techniques for block matching in video coding. in which regions experiencing relatively uniform motion are withheld from further processing via thresholding. thus saving computation drastically. One is a thresholding multiresolution block matching. Extensive experiments show that the proposed algorithm has a consistent performance for sequences with different motion complexities. It reduces the processing time ranging from 14% to 20% while maintaining almost the same quality of the reconstructed image (only about 0.1 dB loss in PSNR), compared with the fastest existing multiresolution technique. The other is a thresholding hierarchical block matching where no pyramid is actually formed. Experiments indicate that for sequences with less motion such as videoconferencing sequences. this algorithm works faster and has much less motion vectors than the thresholding multiresolution block matching method.

MOTION ESTIMATION AND VIDEO CODING

by
Xiaochun Xia

A Dissertation
Submitted to the Faculty of
New Jersey Institute of Technology
in Partial Fulfillment of the Requirements for the Degree of
Doctor of Philosophy

Department of Electrical and Computer Engineering

October 1996

APPROVAL PAGE

MOTION ESTIMATION AND VIDEO CODING

Xiaochun Xia

Dr. Yun-Qing Shi, Dissertation Advisor — Date
Associate Professor of Electrical and Computer Engineering, NJIT

Dr. Joseph Frank, Committee Member — Date
Associate Professor of Electrical and Computer Engineering, NJIT

Dr. Edwin Hou, Committee Member — Date
Assistant Professor of Electrical and Computer Engineering, NJIT

Dr. Zoran Siveski, Committee Member — Date
Assistant Professor of Electrical and Computer Engineering, NJIT

Dr. Douglas Hung, Committee Member — Date
Associate Professor of Computer and Information Science, NJIT

BIOGRAPHICAL SKETCH

Author: Xiaochun Xia

Degree: Doctor of Philosophy

Date: October 1996

Undergraduate and Graduate Education:

- Doctor of Philosophy in Electrical and Computer Engineering.
  New Jersey Institute of Technology. Newark. New Jersey. 1996
- Master of Science in Precision Instrumentation.
  Jiao Tong University. Shanghai. China. 1987
- Bachelor of Science in Precision Instrumentation.
  Jiao Tong University. Shanghai. China. 1984

Major: Electrical Engineering

Presentations and Publications:

X. Xia and Y.Q. Shi. "The determination of optical flow by a multiattribute feedback approach." (Preparing for journal).

Y. Q. Shi and X. Xia. "A thresholding multiresolution block matching algorithm." (Submitted to *IEEE Trans. on Circuits and Systems for Video Technology*).

X. Xia and Y.Q. Shi. "A thresholding hierarchical block matching algorithm for video coding." (Preapring for journal).

X. Xia and Y.Q. Shi. "A thresholding hierarchical block matching for motion estimation." *1996 IEEE International Symposium on Circuits and Systems.* Atlanta. GA. May 1996 (Accepted).

X. Xia and Y.Q. Shi. "Multiresolutional block matching in video compression by thresholding." *Proceedings of IEEE Ninth IMDSP Workshop jointly held by IEEE Signal Processing Society and the Society for Imaging Science and Technology.* Belize City. Belize. March 3-6. 1996. pp. 168 - 169.

X. Xia and Y.Q. Shi. "A new multiresolution block matching algorithm for motion estimation in video coding." *Proceedings of 29th Annual Conf. on Information and System.* John Hopkins University. Baltimore. March 1995. p. 599.

X. Xia and Y. Q. Shi. "A multiple attributes algorithm to compute optical flow." *Proceedings of 29th Annual Conf. on Information and System.* John Hopkins University. Baltimore. March 1995. p. 480.

X. Xia and Liangming Lin. "A study of experimental system on isolated-voice recognition." *Chinese Journal of Medical Instrumentation.* Vol. 14. No. 2. March. 1990. pp. 63-67.

This work is dedicated
to my family

ACKNOWLEDGMENT

It is a pleasure to be able to thank those who have contributed to this work in one way or another.

First of all. I would like to express my thanks to my dissertation advisor. Dr. Yun-Qing Shi. for his ideas. encouragement and guidance throughout the entire work.

Further. I would like to thank the other members of my dissertation committee. Dr. Joseph Frank. Dr. Edwin Hou. Dr. Zoran Siveski. Dr. Douglas Hung and Dr. Frank Shih. for their comments on the dissertation and their valuable insights.

I am grateful to the graduate research environment in the Electronics Imaging Center in Electrical and Computer Engineering Department at New Jersey Institute of Technology.

Finally. I wish to thank all members of my family for their support and constant patience which made it possible for me to concentrate fully on this work.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Figure Page

CHAPTER 1

INTRODUCTION

The motion estimation from image sequences is of crucial importance in image sequence processing [2]. Over the past two decades. enormous efforts have been made on the design of algorithms that extract motion information from a sequence of images [6] [34] [60].

One field in which motion estimation plays a dominant role is computer vision [48]. the ultimate goals of which are vision systems with the ability to discern objects. ascertain their motion. and navigate in 3-D space. Such vision systems are required in applications such as the automatic tracking and recognition of moving objects in traffic monitoring and defense research. the autonomous navigation of mobile vehicles. the inspection of moving objects in robotics. the interpretation and prediction of atmospheric process from satellite image sequences. The fundamental problem in these applications is the extraction of 3-D motion and structure information to predict the position and orientation of the moving object(s). In this process. the determination of 2-D motion field. a projection of the 3-D motion of objects onto the image plane. is considered to be an essential step.

The other field in which motion estimation is a vital issue is video coding [21]. the task of which is data compression for the transmission or storage of image sequences. The demand for image transmission and storage has greatly increased due to factors such as the increased availability of personal workstations. multimedia. and the information society requiring more communication. The application of visual communication ranges from the low bit-rate transmission of videophone. videoconference to the high bit-rate transmission of digital TV and HDTV. The application of digital image storage system involves the storage of medical images. satellite images and etc. The common problem of these applications is the transmission or storage of image sequences as efficient as possible at a certain accepted loss of image quality. The inclusion of motion models is one of the most recent important developments in video coding field [34]. Without motion compensation. it would be impossible to obtain a reasonable quality image at a low or very low transmission bitrates [26]. In the most recent international video compression standards [42]. motion compensation has been utilized as a powerful tool to reduce the temporal redundancy of video images.

In addition to the fields of computer vision and video coding. motion estimation technique has a variety of other applications in image sequence processing [28]. For instance. by estimating motion field. we can create a new image frame between two adjacent existing frames through interpolation. By motion estimation. we can estimate the motion field and identify regions in different frames when image intensities are expected to be the same or similar. Temporal filtering can then be performed in these regions for image restoration.

In the various areas outlined above. the common problem involved is the computation or estimation of motion from a sequence of images recorded from a 3-D scene. Because of the great importance of the motion estimation. various algorithms have been developed to compute 3-D and 2-D motion from image sequences for the wide range of applications since early 70s. and others continue to appear.

This dissertation is mainly concerned with the design of 2-D motion estimation algorithms with application in the areas of computer vision and video coding.

1.1 Motion Estimation and Computer Vision

The research on computer vision is motivated by a broad set of applications such as robotics. automous navigation. tracking of moving object and etc. For such applications. vision systems have to extract the motion and structural information about the 3-D scene. The results of this first step of processing are then used for higher

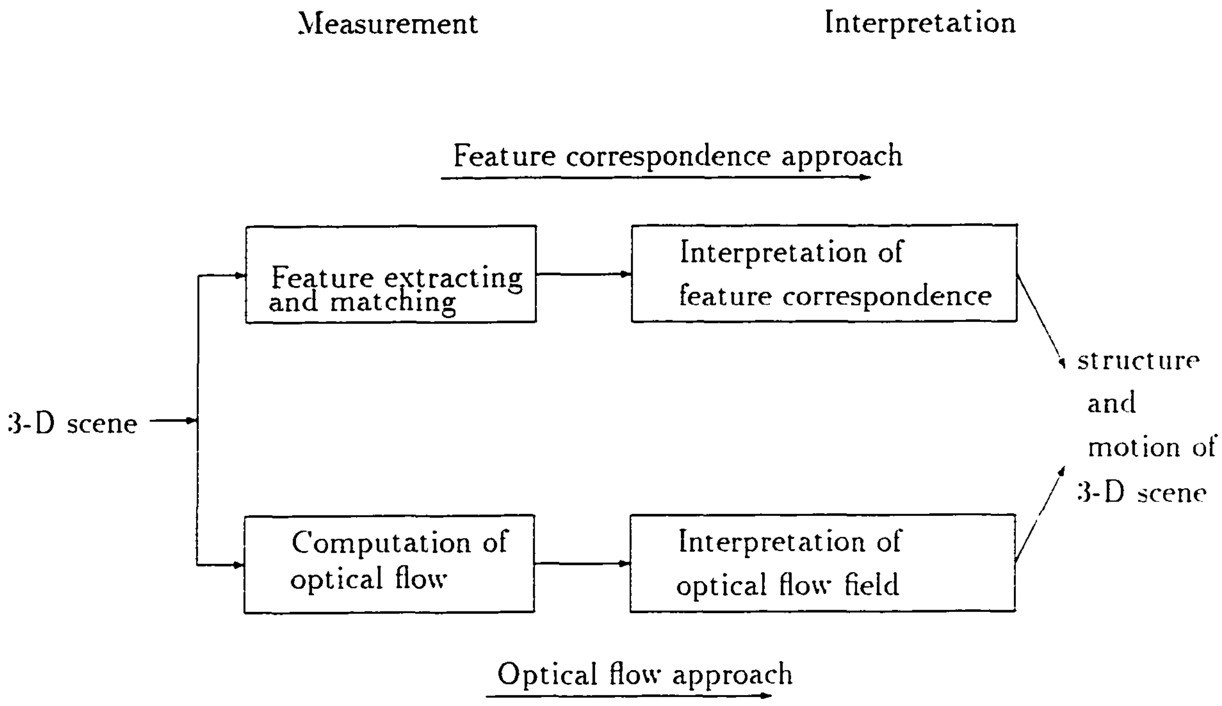

Measurement
Interpretation
Feature correspondence approach
Feature extracting and matching
Interpretation of feature correspondence
3-D scene
structure and motion of 3-D scene
Computation of optical flow
Interpretation of optical flow field
Optical flow approach

Figure 1.1 The recovery of 3-D structure and motion of scene

levels tasks such as navigation in the environment. manipulation of objects. and object recognition as well as scene interpretation. To obtain the structure and motion information of the 3-D scene. we resort to the 2-D images.

If an object is moving in the 3-D scene. its 3-D position and orientation will change in time. Due to the projection of the 3-D scene onto the image plane. these changes will be reflected in the image plane as well. This means that the relative motion between objects in a 3-D scene and a camera gives rise to motion of objects in a sequence of images. Hence. we usually derive the 3-D motion of the objects in the scene through the analysis of the motion information associated with objects in the sequence of images.

Figure 1.1 gives a functional description of the process of recovering 3-D motion and structure from image sequences [46]. As shown. there are two distinct approaches: feature correspondence approach and optical flow approach. In each one of these approaches. there are two stages: measurement and interpretation.

In feature correspondence approach. the measurement stage is responsible for identifying a set of distinctive 2-D features in two or more frames of an image sequence and matching them across the images. The output of the measurement stage gives the position of various features in a set of images. The interpretation stage uses this results to derive the 3-D position of all the points that correspond to the features and velocities of the rigid objects that contain these points. In essence. this approach provides 3-D information for only a sparse set of points.

In optical flow approach. the measurement stage involves constructing a 2-D optical flow field from an image sequence. Optical flow can be regarded as an approximation to a 2-D motion field that depicts the projection of the 3-D motion of the scene. The interpretation stage takes the optical flow field as its input to extract information about the depth and the velocity of every point in the 3-D scene.

Optical flow approach usually computes 2-D motion field along a pixel grid. There is no need for feature extraction and matching. However. they face following problems. First. motion estimation by the optical flow approach can be affected by aperture problem. The aperture problem means that while computing the motion for a given pixel. only the component of the motion vector normal to the underlying contour can be univocally determined by using information in a small neighborhood of the pixel. Generally. the aperture problem exists in regions of an image that have strongly oriented intensity gradients. say edges. Since the motion estimation by optical flow approach is usually carried out in a small spatial-temporal neighborhood of a pixel under consideration. the aperture problem is inherent in every optical flow techniques. To overcome the aperture problem. neighborhood information should be utilized. Second. to pursue a close approximation to the true 2-D motion field. optical flow approach utilizes very sophisticated mathematical tool and as a result. needs an enormous amount of computation. Third. in image acquisition and digitization. noises may be generated. This noise can affect the accuracy of optical flow computation. For instance. gradient-based technique can suffer from high noise sensitivity because of their dependence on spatial-temporal gradients. Fourth. since the interframe motion is restricted to be small. the estimated motion is limited within small range.

Feature correspondence approach allows either small or large motion. it does not suffer from the problem of varying image intensity since the distinct features is relatively more stable than intensity values. However. this approach also has its problems. The tasks of extracting features and establishing feature correspondence are nontrivial. so far only partial solutions suitable for simplistic situations have been developed [2].

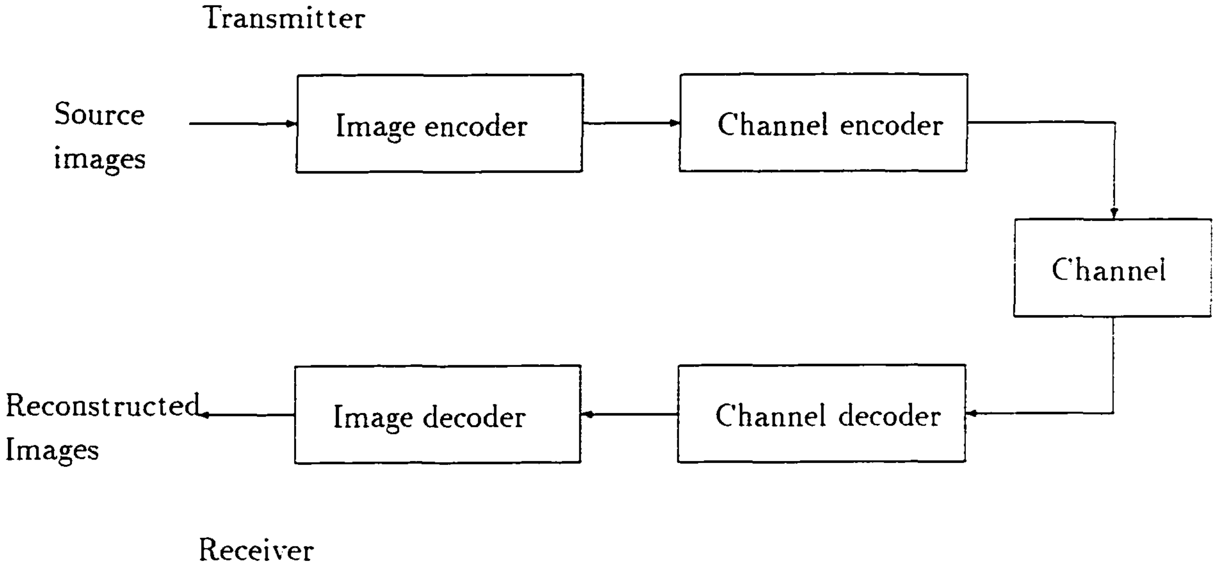


Figure 1.2 A typical system for image communication

The first part of this dissertation research is about the 2-D motion estimation by using the optical flow approach. In Figure 1.1. this work can be classified in the lower left box.

1.2 Motion Compensation and Image Coding

Recently. the need for image communication and image storage has been growing enormously. The key problem of these two applications is to minimize the amount of information necessary to adequately represent represent an image. Figure 1.2 shows a typical system for image communication [28]. The digital image is encoded by an image encoder. The output of the image encoder is a string of bits that represents the source image. The channel encoder transforms the string of bits to a form suitable for transmission over a communication channel through some form of modulation. The modulated signal is then transmitted over a communication channel. At the receiver. the received signal is demodulated and transformed back into a string of bits by a channel decoder. The image decoder reconstructs the image from the string of bits. In contrast to the communication application described in Figure 1.2. no communication channel is involved in application of image coding for storage purpose. In storage applications. the string of bits from the image encoder is stored in proper format on a recording medium. ready for future retrieval.

For both applications. the conventional coding scheme like predictive. transform and interpolative coding strategies can create an annoying flicker or jerkiness when the reconstructed frames are displayed as a video sequence. which is a sequence of still frames that are displayed in a rapid succession. In addition. blurring in the moving boundary may also appear. These distortions can be largely reduced by using mathematical models describing the motion of objects. The frame rate necessary to achieve proper motion rendition is usually high enough to ensure a great deal of temporal redundancy among adjacent frames. Much of the variation in intensity from one frame to the next is due to object motion. Considering the motion information extracted from the image sequence. we can improve the efficiency of the image sequence coding. Compression image sequence accounting for the presence of motion is referred to as motion-compensated (MC) image sequence coding. It should be noted that motion compensation consists of two steps. The first step involves a 2-D motion estimation technique which predicts the motion of an object between frames. The second step uses the estimated motion vector optimally to provide motion compensation at the decoder. with a minimum amount of data transmitted.

In motion-compensated predictive coding. the current frame is predicted from the previous frame by estimating the motion between the two frames and compensating for the motion. The difference between the current frame and the prediction of the current frame is called the motion-compensated prediction (MCP) error. To the extent that the intensity change between the current and previous frames is due to motion and that the motion can be estimated accurately. the error obtained by using motion compensation will have smaller magnitude than the intensities of original image. As a result. a smaller number of coding bits will be required with motion compensation than they would without motion compensation.

There are two kinds of schemes in implementing MC predictive coding as shown in Figures 1.3 and 1.4. that is. forward MC predictive scheme and backward MC predictive scheme [14]. In the forward MC predictive scheme in Figure 1.3. the motion estimation is performed on the current frame and the reconstructed previous frame. Since the current frame is not available at the decoder. the motion information has to be transmitted. This scheme can give an adequately estimated motion. but the lower bound of the bitrate is determined by the motion information to be transmitted. To reduce the amount of motion vectors transmitted. block-based motion compensation algorithms are used to compensate motion in this scheme. In backward MC predictive coding scheme in Figure 1.4. motion estimation is performed only on the reconstructed frames. Therefore. this coding scheme does not require the transmission of any motion information. Therefore. pel-based motion compensation is applied here. However this scheme has its own disadvantage. too. The decoder grows more complex since it has to contain a motion estimation unit. and the coding error also greatly influences the estimation accuracy of the motion field.

Although a critical evaluation of forward versus backward scheme using the same bitrate has not yet been reported. the forward MC scheme has dominated the motion compensation employed and has been recommended by many video compression standards.

The second part of this research deals with block-based motion estimation for forward MC predictive coding. which corresponds to the motion estimation box in Figure 1.3.

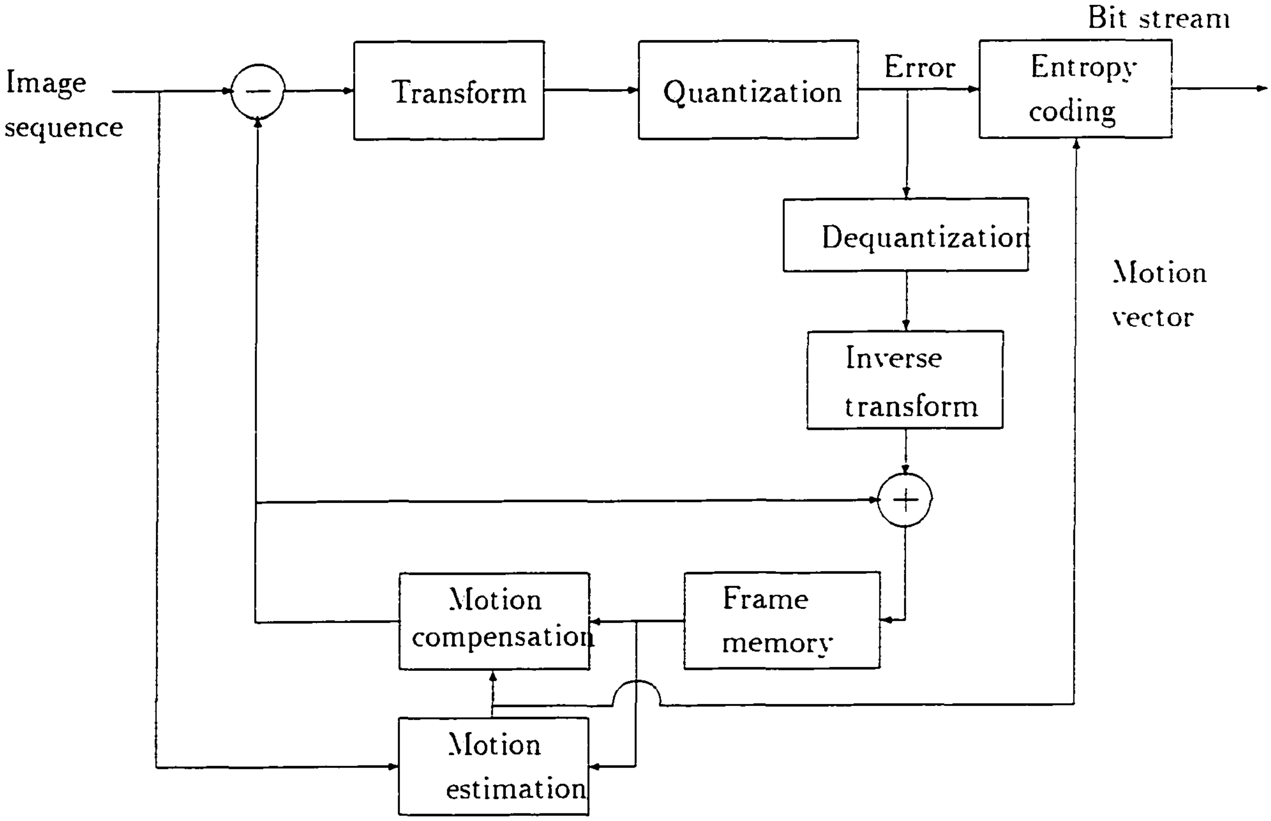


Figure 1.3 The forward motion-compensated predictive coding scheme

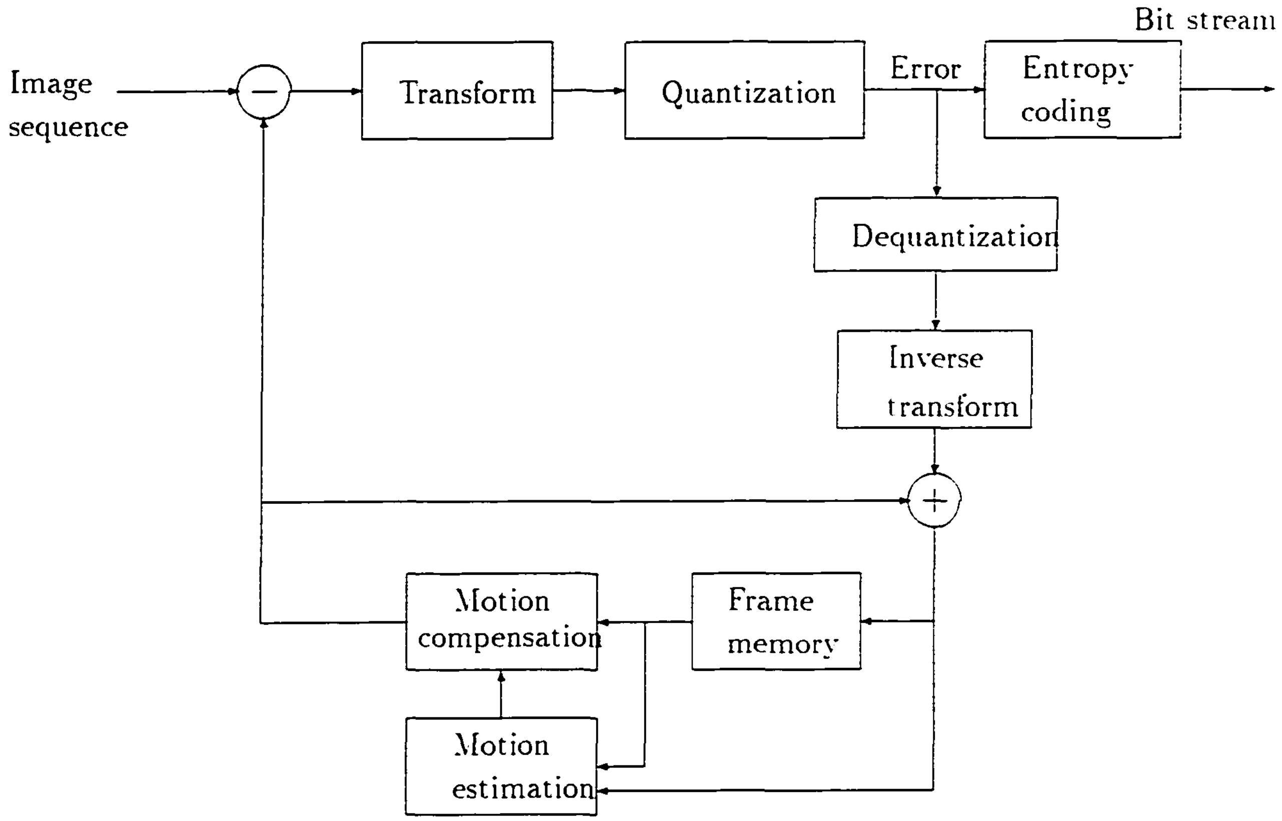


Figure 1.4 The backward motion-compensated predictive coding scheme

1.3 Dissertation Overview

This chapter (Chapter 1) introduces general background of my research work and the rest parts of the dissertation are organized as follows.

In Chapter 2. we give a brief overview of some representative motion estimation techniques and evaluate their performances. To meet the needs of diversified requirements in different application areas. numerous motion estimation algorithms have been presented in the literature. For computer vision application. many tasks require that the computed motion field be accurate and dense. providing a close approximation to the true 2-D motion field. While for image coding application. many visual applications require that coding algorithms be implemented in real time. The computational complexity is an important factor in the derivation of motion estimation algorithms. In this chapter. we first review major related optical flow estimation techniques from the past research. i.e.. gradient-based technique. correlation-based technique and multiple attribute technique. Then. two main classes of motion estimation algorithms in video coding area. namely block-based approach and pel-recursive approach. are summarized.

In Chapter 3. based on an analysis of both advantages and disadvantages of two newly developed algorithms by Weng et al. [52] and Pan et al. [40]. we present a multiattribute feedback approach to determine optical flow field. In this approach for each pixel in an image plane. instead of a single image attribute (image intensity). multiple motion invariant image attributes are computed as conserved information. The estimation of optical flow is carried out in two steps. i.e.. conservation step and propagation step. Feedback technique is utilized to enhance the estimation accuracy. Finally. we present experiments performed on three testing sequences to show the superiority of the presented algorithm over the most of the existing algorithms in computing optical flow field in terms of estimation accuracy.

The next two chapters are devoted to two new thresholding techniques for block matching utilized in video coding. Motion complexity for different regions within an image is usually different. Regions experiencing complex motion deserve more computational resources than those with slow motion. Based on this observation. we withhold regions containing relatively uniform motion from further processing via thresholding. thus saving computation.

In Chapter 4. we present a new thresholding multiresolution block matching algorithm. in which thresholding technique is applied to multiresolution block matching. In extensive experiments with quite different motion complexities. the developed algorithm outperforms the fastest existing multiresolution block matching algorithm while maintaining almost the same quality of reconstructed images.

In Chapter 5. we show details of our newly developed thresholding hierarchical block matching algorithm. In this algorithm. all levels in a hierarchy have the same resolution. No multiresolution pyramid is formed. Experiments have demonstrated that for videoconferencing sequences. this algorithm outperforms the fastest existing multiresolution block matching algorithm while maintaining almost the same quality of reconstructed images.

In Chapter 6. we summarize the results of the research presented in this dissertation and outline the further questions which can be considered as the subjects of our future research.

CHAPTER 2

MAJOR RELATED 2-D MOTION ESTIMATION APPROACHES

The strong interest in estimation of motion from image sequences is motivated by its various applications. These broad applications have different requirements on the motion estimation algorithms. which include the predefined estimation accuracy. computational complexity. economical feasibility. In this chapter. we will give a detailed descriptions to some motion estimation algorithms that are related to this research.

In Section 2.1. motion estimation by optical flow is discussed. and three specific algorithms upon which the first part of the thesis work is based are introduced. In Section 2.2. two groups of motion estimation approaches developed for video coding are presented. the most important algorithms of these two groups are dealt with in Sections 2.2.1 and 2.2.2. respectively.

2.1 Motion Estimation - Optical Flow Determination

The optical flow approach usually computes 2-D motion field on an image grid. There is no need for explicit feature extraction and matching. It can potentially derive dense depth maps for tasks such as 3-D structure and motion analysis. Much of the current work in the computation of optical flow can be classified as one of the three categories: gradient-based techniques [20] [31][35][50]. correlation-based techniques [3][46][40] and spatio-temporal energy based techniques [18][51].

2.1.1 Gradient-based Techniques

The gradient-based techniques are based on the assumption that for a given 3-D scene point. the intensity I at the corresponding point image point remains constant over time. That is. if a 3-D scene point projects onto the image point $(x.y)$ at time t and onto the image point $(x + \Delta x, y + \Delta y)$ at time $(t + \Delta t)$, we can write

$$I(x, y, t) = I(x + \Delta x, y + \Delta y, t + \Delta t) \tag{2.1}$$

where $I(x, y, t)$ is the image intensity at the point (x, y) in the image at time t. This equation is called intensity constant equation.

Expanding the right-hand side by a Taylor series about (x, y, t) and ignoring the second and higher order terms, we obtain

$$I(x + \Delta x, y + \Delta y, t + \Delta t) = I(x, y, t) + \Delta x\frac{\partial I}{\partial x} + \Delta y\frac{\partial I}{\partial y} + \Delta t\frac{\partial I}{\partial t} \tag{2.2}$$

Combining the two equations results in the following expression:

$$\Delta x\frac{\partial I}{\partial x} + \Delta y\frac{\partial I}{\partial y} + \Delta t\frac{\partial I}{\partial t} = 0 \tag{2.3}$$

Dividing throughout by Δt, taking the limit as $\Delta t \rightarrow 0$ and denoting the partial derivatives of I by I_x, I_y, and I_t, we get the intensity constant constraint

$$I_x u + I_y v + I_t = 0 \tag{2.4}$$

where $u = \frac{dx}{dt}$ and $v = \frac{dy}{dt}$. $U = (u, v)$ is the flow vector associated with the point under consideration. The collection of flow vectors $U = (u, v)$ for the entire image constitute the optical flow field for the image.

Equation (2.4) embodies two unknowns u and v, and is not sufficient by itself to specify the optical flow uniquely. But, it does constrain the solution. To compute optical flow for images using this constraint equation, some additional assumptions must be made.

2.1.1.1 Horn and Schunck's Approach Horn and Schunck [20] assumed that the optical flow field varies smoothly across an image plane. A smoothness error , denoted by E_{sm}, is defined as

$$E_{sm} = \int\int(|\nabla u|^2 + |\nabla v|^2)dxdy \tag{2.5}$$

where ∇ stands for gradient operation. From the smoothness assumption. the smoothness error E_{sm} should be small.

An intensity error. E_{int}. is defined as

$$E_{int} = \int\int(\nabla I \cdot U + I_t)^2 dxdy \tag{2.6}$$

From the intensity constant constraint in Equation (2.4). the intensity error should be small. too.

The problem of computing a dense optical flow field is defined as that of minimizing a weighted sum of the two errors.

$$\int\int[(\nabla I \cdot U + I_t)^2 + \alpha^2(|\nabla u|^2 + |\nabla v|^2)]dxdy \tag{2.7}$$

where α^2 determines the relative contributions of the two errors. Horn and Schunck derived an iterative method to calculate the optical flow by

$$\begin{aligned} u^{k+1} &= \bar{u}^k - \frac{I_x[I_x\bar{u}^k + I_y\bar{v}^k + I_t]}{I_x^2 + I_y^2} \\ v^{k+1} &= \bar{v}^k - \frac{I_y[I_x\bar{u}^k + I_y\bar{v}^k + I_t]}{I_x^2 + I_y^2} \end{aligned} \tag{2.8}$$

where k denotes the iteration number. u^0 and v^0 denote initial velocity estimate which is set to zero. and $\bar{u}^k$ and $\bar{v}^k$ denote neighborhood averages of u^k and v^k.

Horn and Schunck were among the first to estimate a 2-D dense motion field by using the optical flow approach. This algorithm has a very fast convergence speed. The primary difficulties with this algorithm are:

1. It is only suitable when the displacements are small respect to the scale of the image intensity variations. In addition. any change in illumination or contrast between frames can cause the intensity constancy assumption to be violated.

2. This method heavily depends on the gradient calculation. as indicated in Equation (2.8). Due to noises in digitized images. it is impossible to accurately

the partial derivatives (I_x. I_y. and I_t) and hence. this approach is sensitive to various noises

3. Horn and Schunck's smoothness constraint may not be valid at image motion boundaries.

2.1.1.2 Nagel's Approach The same as Horn and Schunck's algorithm. Nagel [35] also formulated the problem of computing an optical flow as that of minimizing the sum of an intensity error E_{int} and a smoothness error E_{sm}. He observed that smoothness error E_{sm} used by Horn and Schunck smooths out the flow field omnidirectionally in all directions. The term corresponding to smoothness error in Horn and Schunck's formulation (Equation(2.5)) can be rewritten as

$$E_{sm} = \int\int trace((\nabla U)^T(\nabla U)dxdy \qquad (2.9)$$

The ∇ term expresses the partial derivatives of the two components of velocity in a matrix form

$$\nabla U = \begin{pmatrix} \frac{\partial u}{\partial x} & \frac{\partial v}{\partial x} \\ \frac{\partial u}{\partial y} & \frac{\partial v}{\partial y} \end{pmatrix} \qquad (2.10)$$

Nagel modified the smoothness error as

$$E_{sm} = \int\int trace((\nabla U)^T W(\nabla U)dxdy \qquad (2.11)$$

where W is a 2x2 positive-definite matrix defined as follows:

$$W = \frac{F}{trace(F)} \qquad (2.12)$$

with

$$F = \begin{pmatrix} I_y^2 + \beta^2(I_{xy}^2 + I_{yy}^2) & -I_xI_y - \beta^2 I_{xy}(I_{xx} + I_{yy}) \\ -I_xI_y - \beta^2 I_{xy}(I_{xx} + I_{yy}) & I_x^2 + \beta^2(I_{xx}^2 + I_{xy}^2) \end{pmatrix} \qquad (2.13)$$

The change in smoothness error - that is. setting $W = F/trace(F)$ instead of an identity matrix used by Horn and Schunck - has the following effect. In regions with strong second-order intensity variations. say corners. the smoothness constraint is enforced very weakly and the flow field is allowed to be nonsmooth. Further. in the vicinity of edges. the smoothness constraint is enforced strongly along the direction of the underlying contour and weakly across the contour. Because of these features. Nagel termed his smoothness constraint an oriented smoothness constraint.

Nagel's approach overcomes the drawbacks of Horn and Schunck's smoothness constraint. That is. it does not blur the flow field at motion boundaries and it gives the flow field everywhere. not just at the contours. However. it has practical limitations. It is based on the second-order spatial partial derivatives of the image intensity. Noise and quantization errors associated with digitized images make the computation of second-order derivative error prone.

2.1.2 Correlation-based Techniques

In contrast to the gradient-based techniques where the calculation of partial derivatives of image intensity is required. in correlation-based techniques. there is no need for numerical differentiation. To determine a velocity associated with a pixel under consideration. correlation-based techniques consider a small region around the pixel in an image frame and search for the "best match" among all possible regions in an adjacent frame. The relative position of these two corresponding regions gives a flow estimate. Thus. correlation-based techniques are less sensitive to noises. Since a small region instead of a single image point is used while performing matching. the computation of the correlation-based techniques is very time consuming.

In correlation-based approaches. the velocity $U = (u, v)$ is estimated by minimizing a matching error which is given by

$$E = \sum \sum_{(x,y)\in R} C[I(x,y,t), I(x+u\Delta t, y+v\Delta t, t+\Delta t)] \qquad (2.14)$$

where $C[\cdot,\cdot]$ is a correlation measure that indicates the amount of dissimilarity between two arguments. R is the local spatial region used to estimate (u, v). It is assumed that (u, v) is constant over the region R.

The size of R is dictated by several considerations. If it is chosen too large, the assumption that (u, v) is approximately constant over the region R may not be valid and evaluation of the error expression require more computations. If it is chosen too small, the estimates may become very sensitive to noise. Reasonable choices based on these consideration are a 5x5 or 3x3 pixel region.

There are many possible choices for the correlation measure $C[\cdot,\cdot]$. The most commonly used correlation measures are listed below.

1. Direct correlation, in which the image intensity values of the corresponding pixels in the regions R are multiplied and summed.

2. Mean normalized correlation, in which the average intensity of each region is subtracted from the intensity values of each pixel in that region before multiplication and summing.

3. Variance normalized correlation, in which the correlation sum is divided by the product of the variances of the intensities in each region.

4. Sum of squared differences, in which the sum of the square of the differences between the intensities at corresponding pixel is calculated.

5. Sum of absolute difference, which is similar to sum of squared differences, except that the absolute values of the differences are used instead of their squares.

2.1.2.1 Singh's Approach In Singh's approach [46], the optical flow field is estimated in two steps: i.e., conservation step and propagation step.

In the conservation step, flow vectors are estimated based on the assumption of conservation of local intensity distribution. For each pixel $p(x,y)$ at location (x,y) in the first image I_1, a correlation window W_c of size $(2n+1)\times(2n+1)$ is formed around the pixel. The search window W_s of size $(2N+1)\times(2N+1)$ is established around the pixel at location (x,y) in the second image I_2. For each candidate in W_s, $(2N+1)\times(2N+1)$ samples of error distribution are computed using the sum-of-squared difference (SSD) as

$$E(\Delta x,\Delta y)=\sum_{i=-n}^{n}\sum_{j=-n}^{n}\left(I_1(x+i,y+j)-I_2(x+i+\Delta x,y+j+\Delta y)\right)^2 \qquad (2.15)$$

where $-N \leq \Delta x,\Delta y \leq N$.

Then the $(2N+1)\times(2N+1)$ samples of response distribution are computed as follows:

$$R_c(\Delta x,\Delta y)=e^{-kE(\Delta x,\Delta y)} \qquad (2.16)$$

where $-N \leq \Delta x,\Delta y \leq N$. k is chosen so as to make $R_c(\cdot,\cdot)$ vary between zero and unity over the entire range of error.

An estimate of flow vector at the conservation step, denoted by $U_c=(u_c,v_c)$ where c stands for conservation, is obtained by using a so-called weighted least square estimation technique [46] as follows:

$$u_c=\frac{\sum_{\Delta x}\sum_{\Delta y}R(\Delta x,\Delta y)\Delta x}{\sum_{\Delta x}\sum_{\Delta y}R(\Delta x,\Delta y)}$$
$$v_c=\frac{\sum_{\Delta x}\sum_{\Delta y}R(\Delta x,\Delta y)\Delta y}{\sum_{\Delta x}\sum_{\Delta y}R(\Delta x,\Delta y)} \qquad (2.17)$$

where the summation is carried out over $-N \leq \Delta x,\Delta y \leq N$. Each estimate given above is associated with a covariance matrix

$$S_c=\begin{pmatrix}\frac{\sum_{\Delta x}\sum_{\Delta y}R_c(\Delta x,\Delta y)(\Delta x-u_c)^2}{\sum_{\Delta x}\sum_{\Delta y}R_c(\Delta x,\Delta y)} & \frac{\sum_{\Delta x}\sum_{\Delta y}R_c(\Delta x,\Delta y)(\Delta x-u_c)(\Delta y-v_c)}{\sum_{\Delta x}\sum_{\Delta y}R_c(\Delta x,\Delta y)}\\ \frac{\sum_{\Delta x}\sum_{\Delta y}R_c(\Delta x,\Delta y)(\Delta x-u_c)(\Delta y-v_c)}{\sum_{\Delta x}\sum_{\Delta y}R_c(\Delta x,\Delta y)} & \frac{\sum_{\Delta x}\sum_{\Delta y}R_c(\Delta x,\Delta y)(\Delta y-v_c)^2}{\sum_{\delta x}\sum_{\delta y}R_c(\Delta x,\Delta y)}\end{pmatrix} \qquad (2.18)$$

The covariance matrix S_c measures the deviation of the estimate U_c from the true velocity and is used as a confidence measure for the estimate U_c.

In the propagation step, the estimate from the conservation step is propagated by using the neighborhood information. The velocity estimate at this step, denoted by $V_n = (u_n, v_n)$ where n stands for neighborhood, can be derived from velocities $V_i = (u_i, v_i)$ in its local $(2w+1) \times (2w+1)$ neighborhood as follows:

$$\begin{aligned} u_n &= \frac{\sum_i R_n(u_i, v_i) u_i}{\sum_i R_n(u_i, v_i)} \\ v_n &= \frac{\sum_i R_n(u_i, v_i) v_i}{\sum_i R_n(u_i, v_i)} \end{aligned} \qquad (2.19)$$

where both $R_n(u_i, v_i)$ and $V_i = (u_i, v_i)$ are assumed to be known in advance from an independent source. The corresponding covariance matrix is

$$S_n = \begin{pmatrix} \frac{\sum_i R_n(u_i, v_i)(u_i - u_n)^2}{\sum_i R_n(u_i, v_i)} & \frac{\sum_i R_n(u_i, v_i)(u_i - u_n)(u_i - v_n)}{\sum_i R_n(u_i, v_i)} \\ \frac{\sum_i R_n(u_i, v_i)(u_i - u_n)(v_i - v_n)}{\sum_i R_n(u_i, v_i)} & \frac{\sum_i R_n(u_i, v_i)(v_i - v_n)^2}{\sum_i R_n(u_i, v_i)} \end{pmatrix} \qquad (2.20)$$

Also, the covariance matrix S_n gives a measure of the deviation of the estimate V_n from the true velocity and is used as a confidence measure for the estimate V_n.

It is obvious that the velocity estimates V_c and V_n are erroneous. The error at the conservation step is given by

$$(V - V_c)^T S_c^{-1} (V - V_c) \qquad (2.21)$$

while the error at the propagation step is given by

$$(V - V_n)^T S_n^{-1} (V - V_n) \qquad (2.22)$$

where V is the true velocity of the pixel under consideration.

The final velocity estimate, $V = (u, v)$, is determined by minimizing the sum of errors of the conservation step and the propagation step as

$$\int\int [(V - V_n)^T S_n^{-1} (V - V_n) + (V - V_c)^T S_c^{-1} (V - V_c)] dx dy \rightarrow Min \qquad (2.23)$$

Singh derived the estimate from this equation by using a calculus of variations iteratively as

$$\begin{aligned} V^{k+1} &= [S_c^{-1} + S_n^{-1}]^{-1} [S_c^{-1} V_c + S_n^{-1} V_n^k] \\ V^0 &= V_c \end{aligned} \qquad (2.24)$$

where U_c and S_c are known from the conservation step (and fixed) for each pixel. On the other hand, U_n and S_n are derived from the assumption that the velocity of each pixel in the neighborhood is known in advance from an independent source. In practice, this assumption is invalid. In his implementation, Singh obtains S_n and U_n from the neighborhood velocity from the previous iteration.

A significant contribution of Singh's work is an estimation-theoretic framework to compute optical flow. Observing that estimated optical flow cannot be exactly the same as true 2-D motion field, Singh treated the problem of optical flow recovery as that of parameter estimation, where the estimated parameter is a flow estimate accompanied by a covariance matrix for each pixel in the image. The second major contribution is that Singh gave a unified perspective for all existing optical flow techniques. He showed that all of the optical flow techniques consist of two distinct step: conservation step and propagation step. The conservation step sets up constraints based on conservation of image properties. The propagation step uses conservation constraints along with some neighborhood information to recover true optical flow field.

1. Like all of the correlation-based techniques, in Singh's algorithm, the assumption that the intensity distribution within the correlation window remains unchanged over time can be invalid at motion boundaries and causes great errors.

2. As indicated in Equation (2.24), the final estimate consists of two parts. The U_c and S_c are estimated flow vector and its associated covariance matrix which are derived from the original image intensity at the conservation step and are fixed for each point, while U_n and S_n are derived from the previous iteration by using neighborhood information. Thus, in Singh's algorithm, the information in original images is utilized only once at the conservation step. At the propagation step, the estimate is refined repeatedly by exploiting the neighborhood information.

3. In this method. subpixel problem is not addressed.

2.1.2.2 Pan, Shi and Shu's Approach

Motivated from the above discussion about the drawbacks of Singh's algorithm. Pan et al. [40] developed a correlation-feedback approach to compute the optical flow which enhances the estimation accuracy of optical flow by fully exploiting the information of the original images.

Pan et al.'s algorithm consists of two stages: correlation stage and propagation stage.

On the correlation stage. velocity is estimated based on the correlation information between the neighboring images. Let I_1 and I_2 denote the image at moment 1 and 2. respectively. Let $U^n = (u^n.v^n)$ denote the estimated flow vector at the nth iteration. Then at the $n+1$th iteration. for each pixel $p(x.y)$ at location $(x.y)$ in the image I_2. a correlation window W_c of size $(2n+1) \times (2n+1)$ is formed around the pixel $p(x.y)$. The search window W_s established at the image I_1 is of variable size. The position of the candidates within W_s. denoted by $(\Delta x.\Delta y)$. satisfies the following

$$\begin{aligned} \Delta x &\in (u^n - u^n/2.\, u^n - u^n/4.\, u^n.\, u^n + u^n/4.\, u^n + u^n/2) \\ \Delta y &\in (v^n - v^n/2.\, v^n - v^n/4.\, v^n.\, v^n + v^n/4.\, v^n + v^n/2) \end{aligned} \qquad (2.25)$$

There are total 5×5 candidates within W_s. For each candidate. the matching error is computed using SSD as

$$E(\Delta x.\Delta y) = \sum_{x=-n}^{n} \sum_{y=-n}^{n} (I_1(i+x.j+y) - f(i+x+\Delta x.j+y+\Delta y))^2 \qquad (2.26)$$

where $\Delta x.\Delta y$ satisfy Equation (2.25). $f(\cdot.\cdot)$ is an interpolated image and is given by

$$\begin{aligned} f(i+x+\Delta x.j+y+\Delta y) &= (1-a)[(1-b)I_2(i.j) + b \times I_2(i.j+1)] \\ &\quad + a[(1-b)I_2(i+1.j) + b \times I_2(i+1.j+1)] \end{aligned} \qquad (2.27)$$

where $i = [i + x + \Delta x]$:$j = \lfloor j + y + \Delta y \rfloor$: $a = i + x + \Delta x - i$:$b = j + y + \Delta y - j$:$\lfloor x \rfloor$ means that only the integer part of x is retained. By using the interpolated image. the velocity estimate will not be restricted to be integers and the error caused by subpixel problem can be reduced.

The same as Equation (2.16). the 5×5 samples of matching errors are converted into a probability distribution using

$$R(\Delta x. \Delta y) = e^{-kE(\Delta x. \Delta y)} \tag{2.28}$$

where k is chosen so as to make R be a number close to unity.

Then the velocity estimate on the correlation stage. denoted by $U' = (u'. v')$. is given by

$$\begin{aligned} u' &= \frac{\sum_{\Delta x} \sum_{\Delta y} R(\Delta x. \Delta y) \Delta x}{\sum_{\Delta x} \sum_{\Delta y} R(\Delta x. \Delta y)} \\ v' &= \frac{\sum_{\Delta x} \sum_{\Delta y} R(\Delta x. \Delta y) \Delta x}{\sum_{\Delta x} \sum_{\Delta y} R(\Delta x. \Delta y)} \end{aligned} \tag{2.29}$$

On the propagation stage. the estimates from the correlation stage are further improved by using the neighborhood information. based on the assumption that velocity at the local neighborhood should be similar. The estimate at the $n + 1$th iteration. denoted by $U^{n+1} = (u^{n+1}. v^{n+1})$. is given by

$$\begin{aligned} u^{n+1} &= \sum_{x=-N}^{N} \sum_{y=-N}^{N} w(x. y) * u'(i + x. j + y) \\ v^{n+1} &= \sum_{x=-N}^{N} \sum_{y=-N}^{N} w(x. y) * v'(i + x. j + y) \end{aligned} \tag{2.30}$$

where $w(x. y)$ is a 3×3 Gaussian mask as shown in Figure 2.1.

Compared with Singh's algorithm. Pan et al.'s algorithm has the following characteristics. First. the refinement of estimated flow field is based on the original images. In the algorithm. the flow estimate from the last iteration is fedback to the algorithm. This flow vector together with its perturbed values (refer to Equation (2.25)) is utilized as matching candidate for the next iteration. The larger the

	-1	0	1
-1	0.25*0.25	0.5*0.25	0.25*0.25
0	0.25*0.5	0.5*0.5	0.25*0.5
1	0.25*0.25	0.5*0.25	0.25*0.5

Figure 2.1 Gaussian mask

matching error. the smaller the contribution of the matching candidate to the flow estimate. Thus the estimate is repeatedly refined by fully exploiting the information of the original images. Second. the incorporation of the bilinear interpolation technique largely reduces the error caused by subpixel problem. However. this algorithm has some problems. too.

1. The same as Singh's algorithm. this algorithm assumes the conservation of intensity distribution within a correlation window. At motion boundaries. this assumption can be violated and cause great errors.

2. In applying the motion smoothness constraint. this algorithm does not consider motion discontinuities. As in Equation (2.30). the flow estimate on the propagation stage is a weighted mean of those over a small neighborhood. The weight is simply a Gaussian function and does not take into account discontinuities.

2.1.3 Multiple Attributes Technique

In all above-mentioned methods. only the image intensity is used as the conservation information. Based on an analysis indicating that using image intensity as a single attribute is not enough in accurate matching for image points. Weng. Ahuja and Huang [52] have proposed a multiple image attribute technique to image point matching. Although this approach performs image matching instead of computing the optical flow explicitly. the performed image matching amounts to optical flow field computation since it calculates a displacement field for each point in the image plane which is essentially a flow field if the time interval between two different images is known.

In the algorithm. for each image point. multiple image attributes are defined. which include image intensity. edgeness. negative cornerness and positive cornerness. These image attributes are motion invariant and computed as conserved information. They are briefly introduced as follows.

Intensity

The image intensity at a point $\mathbf{p}$ in an image I. denoted by $i(\mathbf{p})$. is equal to $I(\mathbf{p})$. i.e.. $i(\mathbf{p}) = I(\mathbf{p})$.

Edgeness

The edgeness at a point $\mathbf{p}$. denoted by $\epsilon(\mathbf{p})$. is given by

$$\epsilon(\mathbf{p}) = \left\| \frac{\partial I(\mathbf{p})}{\partial \mathbf{p}} \right\| \tag{2.31}$$

i.e.. the edgeness is defined as the magnitude of the gradient of image intensity.

Cornerness

The positive cornerness and negative cornerness at a point $\mathbf{p}$. denoted by $c_p(\mathbf{p})$ and $c_n(\mathbf{p})$. respectively. are defined by

$$c_p(\mathbf{p}) = \begin{cases} \epsilon(\mathbf{p})\{1 - |1 - angle(\mathbf{a}.\mathbf{b}) * \{2/\pi\}|\} & 0 \le angle(\mathbf{a}.\mathbf{b}) \le \pi \\ 0 & \text{otherwise} \end{cases}$$
$$c_n(\mathbf{p}) = \begin{cases} \epsilon(\mathbf{p})\{1 - |1 + angle(\mathbf{a}.\mathbf{b}) * \{2/\pi\}|\} & -\pi \le angle(\mathbf{a}.\mathbf{b}) \le 0 \\ 0 & \text{otherwise} \end{cases} \tag{2.32}$$

where $\mathbf{a}$ and $\mathbf{b}$ are intensity gradients at points $\mathbf{p} + \mathbf{r}_a$. and $\mathbf{p} + \mathbf{r}_b$. respectively.

$$\mathbf{a}^T = \left.\frac{\partial I(\mathbf{s})}{\partial \mathbf{s}}\right|_{\mathbf{s}=\mathbf{p}+\mathbf{r}_a}$$

$$\mathbf{b}^T = \left.\frac{\partial I(\mathbf{s})}{\partial \mathbf{s}}\right|_{\mathbf{s}=\mathbf{p}+\mathbf{r}_b}$$

and $\|\mathbf{r}_a\| = \|\mathbf{r}_b\| = r$. $\mathbf{r}_a$ and $\mathbf{r}_b$ are such that

$$\left.\frac{\partial I(\mathbf{v})}{\partial \mathbf{v}}\right|_{\mathbf{v}=\mathbf{p}+\mathbf{r}_a} * \mathbf{r}_a^{\perp} = \min_{\|\mathbf{r}\|=r} \left.\frac{\partial I(\mathbf{v})}{\partial \mathbf{v}}\right|_{\mathbf{v}=\mathbf{p}+\mathbf{r}} * \mathbf{r}^{\perp}$$

$$\left.\frac{\partial I(\mathbf{v})}{\partial \mathbf{v}}\right|_{\mathbf{v}=\mathbf{p}+\mathbf{r}_b} * \mathbf{r}_b^{\perp} = \max_{\|\mathbf{r}\|=r} \left.\frac{\partial I(\mathbf{v})}{\partial \mathbf{v}}\right|_{\mathbf{v}=\mathbf{p}+\mathbf{r}} * \mathbf{r}^{\perp}$$

The superscript $\perp$ denotes the corresponding perpendicular vector. i.e.. if $\mathbf{r} = (r_u.r_v)^T$. then $\mathbf{r}^{\perp} = (-r_v.r_u)$.

In order to take those regions into consideration where no significant intensity variation occurs. additional local smoothness constraints are also imposed. Namely. both amplitude and orientation of the displacement vector of the point under consideration should be similar to those displacement vectors in the vicinity of this point. Let $\bar{\mathbf{d}}(\mathbf{p}_0)$ denote the displacement vector field in the vicinity of the point $\mathbf{p}_0$. Then. $\bar{\mathbf{d}}(\mathbf{p}_0)$ is computed as

$$\bar{\mathbf{d}}(\mathbf{p}_0) = \int\int_{0<\|\mathbf{p}-\mathbf{p}_0\|<r} w(\eta_i)\mathbf{d}(\mathbf{p})d\mathbf{p} \tag{2.33}$$

where $0 < \|\mathbf{p} - \mathbf{p}_0\| < r$ denotes a region around $\mathbf{p}_0$. $w(\cdot)$ is a weighting function and is given by

$$w(\eta_i) = \frac{c}{\epsilon + |\eta_i|} \tag{2.34}$$

where $\eta_i = |I(\mathbf{p}) - I(\mathbf{p}_0)|$. ϵ is a small positive number to prevent the denominator from zeroing. and c is a normalization constant that makes the summation of weights equal to 1. Obviously. the weight is inversely proportional to the intensity difference between the point $\mathbf{p}_0$ and the surrounding points $\mathbf{p}$. The larger the difference in intensity is. the more likely the two points come from different regions. and the smaller the weight will be. Hence. the weight implicitly takes into account motion discontinuities. Consequently. to a certain extent. the local smoothness constraint preserves discontinuities in the displacement field.

To measure the similarity of attributes between the corresponding points in two images. a set of residual functions are defined. The residual of intensity at a point $\mathbf{p}$ with a displacement vector $\mathbf{d}(\mathbf{p})$ is defined by

$$r_i(\mathbf{p}.\mathbf{d}) = i_2(\mathbf{p}+\mathbf{d}) - i_1(\mathbf{p}) \tag{2.35}$$

In the same way. the residual of edgeness $r_e(\mathbf{p}.\mathbf{d})$. that of positive cornerness $r_{c_p}(\mathbf{p}.\mathbf{d})$. and that of negative cornerness $r_{c_n}(\mathbf{p}.\mathbf{d})$ are determined. A measure of similarity between the displacement vector at a point $\mathbf{p}$ and the vector $\bar{\mathbf{d}}(\mathbf{p})$ is given by orientation residual and amplitude residual. which are defined as

$$r_o(\mathbf{p}.\mathbf{d}) = \frac{||\mathbf{d}(\mathbf{p}) \times \bar{d}(\mathbf{p})||}{||\bar{\mathbf{d}}(\mathbf{p})||} \tag{2.36}$$

and

$$r_d(\mathbf{p}.\mathbf{d}) = ||\mathbf{d}(\mathbf{p}) - \bar{\mathbf{d}}(\mathbf{p})|| \tag{2.37}$$

Let s denote a weighted sum of squares of residuals at the point $\mathbf{p}$. i.e.. $\mathbf{s}(\mathbf{d}) = \sum\{r_i^2 + \lambda_e r_e^2 + \lambda_p r_{c_p}^2 + \lambda_n r_{c_n}^2 + \lambda_o r_o^2 + \lambda_d r_d^2\}$. then the displacement estimate at the point $\mathbf{p}$ is determined such that the following is minimized

$$\mathbf{s}(\mathbf{d}) \rightarrow \min \tag{2.38}$$

where $\lambda_e. \lambda_p. \lambda_n. \lambda_o. \lambda_d$ are weighting parameters.

To solve for $\mathbf{d}(\mathbf{p})$. this method resorts to numerical differentiation. The estimate can be recursively obtained by

$$\delta_{\mathbf{d}} = -(J^T \Lambda^2 J)^{-1} J^T \Lambda^2 \mathbf{s}(\mathbf{d}) \tag{2.39}$$

where

$$\Lambda = diag(1.1.\lambda_e.\lambda_{c_p}.\lambda_{c_n}.\lambda_o.\lambda_d) \tag{2.40}$$

and

$$\begin{aligned} J &= \frac{\partial \mathbf{s}(\mathbf{d})}{\partial \mathbf{d}} \\ &= \begin{bmatrix} \frac{\partial i_2}{\partial x}\frac{\partial i_2}{\partial y} \\ \frac{\partial e_2}{\partial x}\frac{\partial e_2}{\partial y} \\ \frac{\partial p_2}{\partial x}\frac{\partial p_2}{\partial y} \\ \frac{\partial n_2}{\partial x}\frac{\partial n_2}{\partial y} \\ -\bar{d}_x/||\bar{\mathbf{d}}||\bar{d}_y/||\bar{\mathbf{d}}|| \\ 10 \\ 01 \end{bmatrix} \end{aligned}$$

where $(\bar{d}_x.\bar{d}_y)^T = \bar{\mathbf{d}}$. and the partial derivative $\frac{\partial i_2}{\partial x}$ denotes the partial derivative of $i_2(x.y)$ with respect to x at the point $\mathbf{p} + \mathbf{d}$. and so on.

The algorithm by Weng et al. takes advantages of both optical flow approach and feature correspondence approach. First. compared with optical flow approach where a single image attribute (image intensity) is utilized. multiple attributes associated with each image point are employed to determine the displacement field on a point grid and makes image point matching more robust. Second. in contrast to the feature correspondence approach. this algorithm has essentially avoided the problems of feature extraction and matching by considering dense motion vector field. Third. this method can deal with large motion. However. this algorithm has some problems. too.

1. The image attributes. edgeness and cornerness defined in Equations (2.31) and (2.32) need the computation of the spatial gradient of the original image intensity. Due to various noises. the computed attribute images can be noisy.

2. In solving for the displacement field. this method resorts to numerical differentiation again. The estimated displacement vectors are updated based on the

computation of the partial derivatives of the noisy attribute images (refer to Equation (2.39)).

3. This algorithm does not address the subpixel problem.

2.2 Motion Estimation for Image Coding

For image coding application. the extracted motion information is utilized for improving the bandwidth reduction of image sequences. In addition to the quality of prediction. the computational complexity is an important factor in the derivation of motion estimation algorithms due to the real-time implementation requirement of the MC coding scheme. Most motion estimation algorithms designed for image coding estimate only a special motion that is 2-D translation.

The motion estimation algorithms developed for image coding can basically be classified into two groups which are known as block-based approach and pel-recursive approach. In this section. the most important block-based approach is discussed in section 2.2.1. while the pel-recursive approach is briefly summarized in section 2.2.2.

2.2.1 Block-based Approach

In block-based motion estimation technique. the present frame of an image sequence is divided into rectangular or square blocks of pixels. It is assumed that all pixels within one block are of the same motion vector. Hence each block has only one motion vector. For a block in the current frame. we look for the block of pixels in the previous frame that gives the best match in terms of a predefined criterion. This best matched block is then used as a predictor for the present block. The relative position of these two blocks defines a motion vector associated with the present block. The collection of all motion vectors defines a motion field and is sent to the receiver. Compared with the correlation-based techniques in section 1.1.2 which estimate dense motion vector fields. the block-based approach here estimates block motion vector fields.

Among the various criteria for block matching. the mean square error (MSE) and the mean absolute difference (MAD) are mostly used [34]. Let $I_n(i.j)$ denote the frame of an image sequence at present moment n. We refer to a block of $b_x \times b_y$ pixels by the coordinate $(i.j)$ of its upper left corner. The motion vector of the block $(i.j)$ is denoted by $V(i.j) = (u.v)$. The MSE and MAD between the block $(i.j)$ of the current frame and the block $(i + u.j + v)$ of the previous frame can be defined as follows. respectively.

$$MSE_{(i.j)}(u.v) = \frac{1}{b_x \times b_y} \sum_{k=0}^{b_x-1} \sum_{l=0}^{b_y-1} (I_n(i+k.j+l) - I_{n-1}(i+k+u.j+l+v))^2 \quad (2.41)$$

and

$$MAD_{(i.j)}(u.v) = \frac{1}{b_x \times b_y} \sum_{k=0}^{b_x-1} \sum_{l=0}^{b_y-1} |I_n(i+k.j+l) - I_{n-1}(i+k+u.j+l+v)| \quad (2.42)$$

The motion vector $V(i.j)$ of the block $(i.j)$ is given by

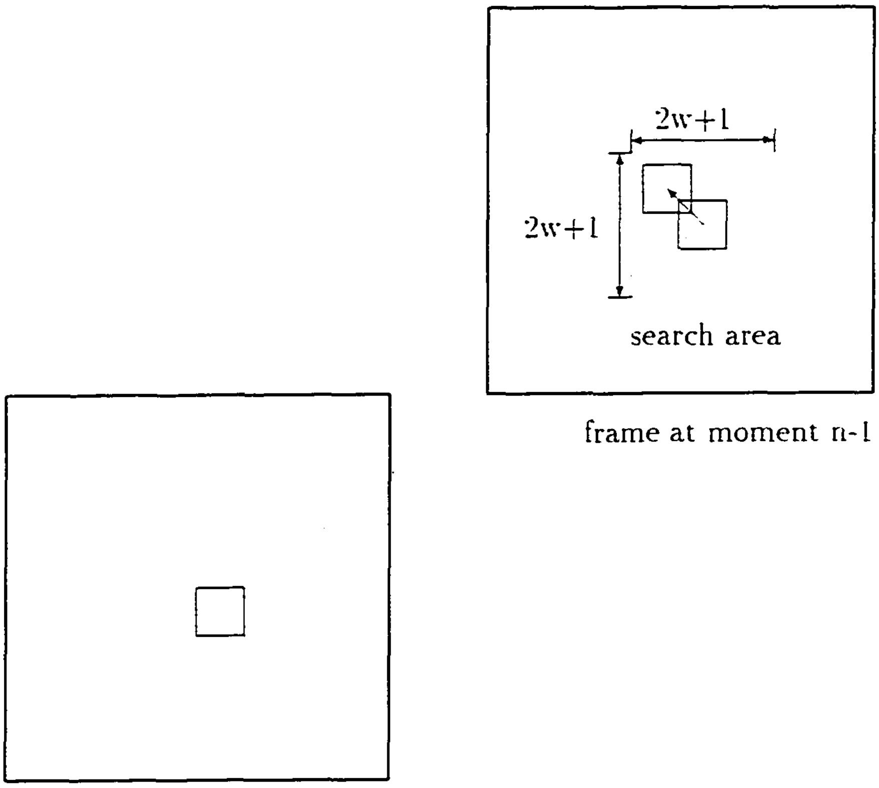
$$V(i.j) = \arg min MSE_{(i.j)}(u.v) \quad (2.43)$$

or

$$V(i.j) = \arg min MAD_{(i.j)}(u.v) \quad (2.44)$$

For each location in the previous frame to be tested. the calculation of the MSE requires $2b_xb_y$ additions and b_xb_y multiplications while the calculation of MAD requires only $2b_xb_y$ additions. Since the MAD requires no multiplication and gives similar performance as the MSE. the MAD is favored in most block motion estimation algorithms.

Figure 2.2 gives an illustration of the technique of block-based motion estimation.



frame at moment n-1

frame at moment n

Figure 2.2 Basics of block-based approach

2.2.1.1 Full Search Block Matching Full search block matching (FBM) is also called exhaustive block matching and has been recommended by various video compression standards. This technique computes the MAD (or MSE) at all the possible locations within the search area to find the optimal motion vector. Let w denote the maximum displacement which can be estimated in both horizontal and vertical directions. there are $(2w+1)^2$ locations for the FBM to search for the best match to the current block $(i.j)$. To perform motion estimation in real time. the number of operations required by the FBM is often too high. To reduce the computational complexity. a number of fast search algorithms have been proposed.

2.2.1.2 Three-step Search Based upon the assumption that within a predetermined search area. the MAD has a single peak and the average matching distortion increases monotonically in directions other than the actual displacement. the three-step algorithm [24] simplified the search procedure for motion estimation.

As shown in Figure 2.3. three different sets of parameters are used in the motion estimation. In the first step. eight coarsely spaced points are tested except for the central point (x=i.y=j). Searching the minimum of the MAD function. The first approximation of the displacement vector is obtained which is the point A(i+3. j+3) in Figure 2.3. In a second step. again eight search point are spaced. but less coarsely around the point which was chosen in the first step. Again using the MAD criterion. the point B(i+3. j+5) is found to be the best match. The second step is repeated until the required accuracy is achieved. In this case (dm=6). the third step gives the final approximation of the displacement vector C(i+2. j+6).

This simplified search algorithm works well if the assumptions hold. Unfortunately. however. these assumptions are not always true. especially in the situations where image contains highly detailed texture and complicated motion. Most often a non-optimal or even error estimation can be the matching results and that will have a serious effect on the coding quality and efficiency.

2.2.1.3 Multiresolution Block Matching Multiresolution block matching technique reduces the computation of motion estimation by taking advantage of pyramid structure. In a so-called top-down multiresolution technique [49]. pyramids are formed before matching. The bottom level L of the pyramid contains the input image. The image at any level $l = L - 1. \cdots .0$ is generated by applying a low-pass filter to the image at level $l + 1$ and subsampling the filtered image. The low-pass filtering is achieved through convolution with a separable filter with 5×5 point impulse response $h(m.n)$ given by

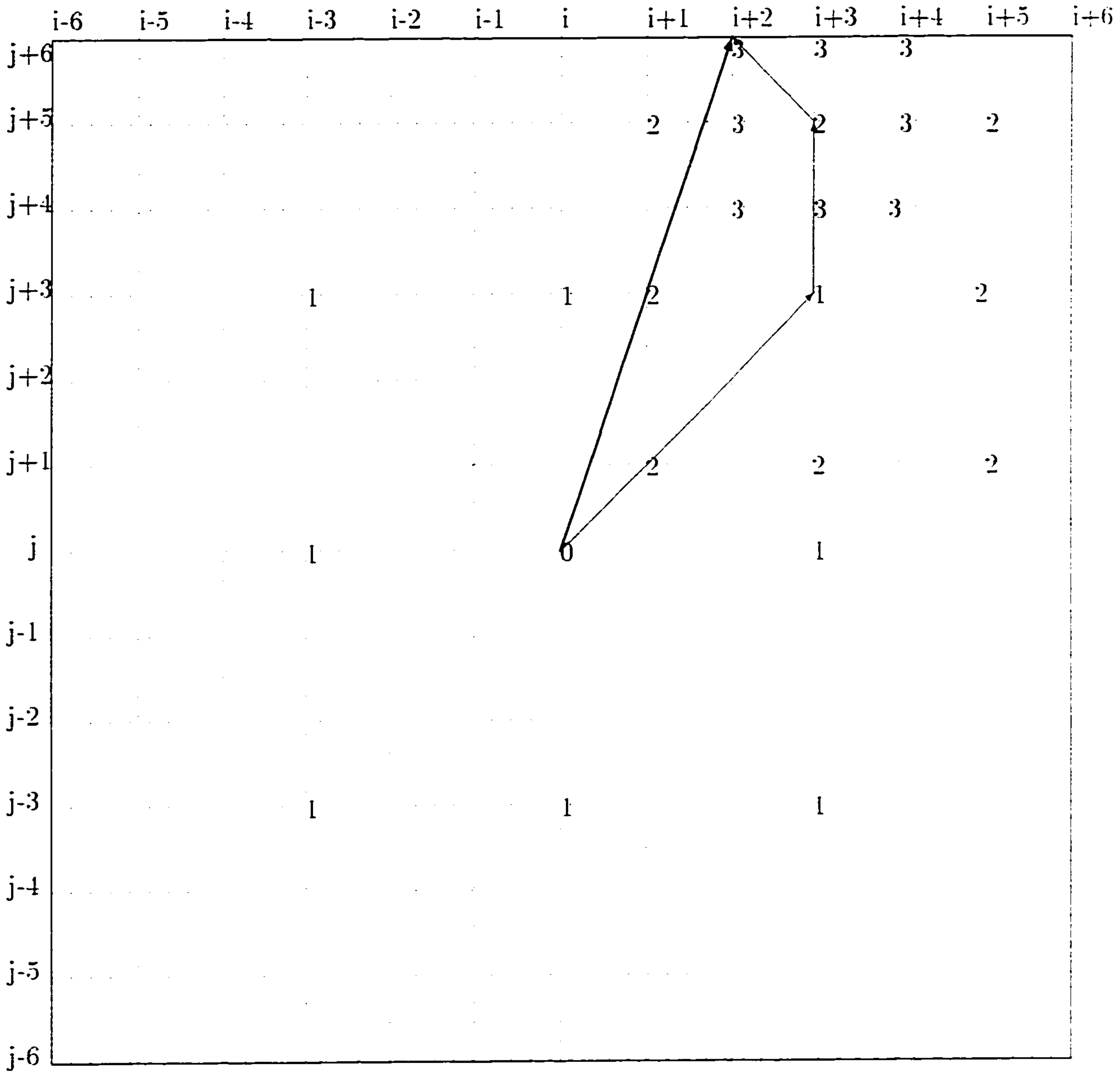
$$h(m.n) = h(m)h(n) \tag{2.45}$$

where $h(0) = \alpha. h(10 = h(-1) = \frac{1}{4}. h(2) = h(-2) = \frac{1}{4} - \frac{\alpha}{2}$. The constant α is a free parameter and is chosen typically between 0.3 and 0.6.

The sampling is done simply by selecting every 2nd pixel in both horizontal and vertical directions. Matching is first conducted at the top level of the pyramid to obtain an initial estimation of the motion field: the computed motion filed is then propagated to the next pyramid level. In the variable block size method (refer to Method 1 in [49]). the size of the block varies with the pyramid levels. That is. if the block size at level l is $b_x \times b_y$. at level $l + 1$ it becomes $2b_x \times 2b_y$. Therefore. if $V^l(i.j)$ is the computed motion vector for the block $(i.j)$ at level l of the pyramid. the propagated motion vector for the same block at level $l + 1$ is $V^{l+1}(2i.2j)$ and is given by

$$V^{l+1}(2i.2j) = 2V^l(i.j) \tag{2.46}$$

At level $l + 1$. this propagated motion vector is corrected and again propagated to the next level until the bottom level of the pyramid is reached. Figure 2.4 shows a



1: search point in the first step
2: search point in the second step
3: search point in the third step

Figure 2.3 Three-step search

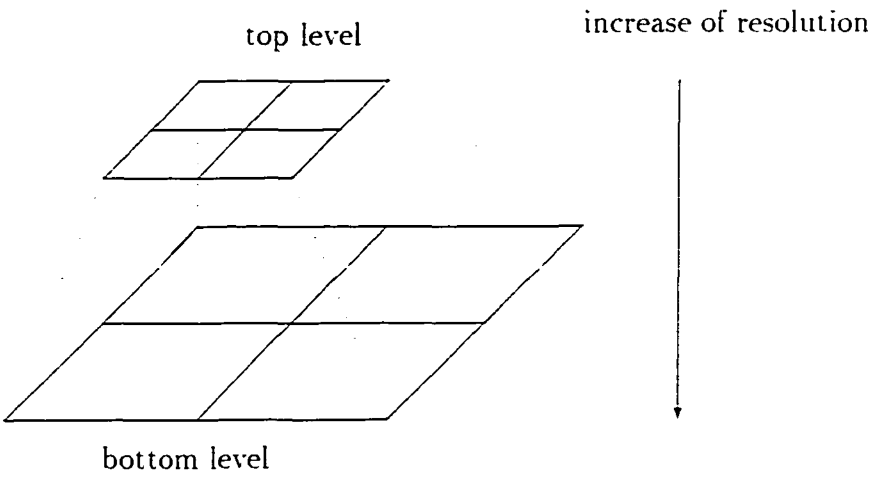


Figure 2.4 2-level pyramid structure

Table 2.1 The typical set of parameters for 2-level multiresolution block matching

Parameters at level	Top level	Bottom level
Search range	4x4	1x1
Block size	4x4	8x8
Motion estimation accuracy	1	0.5

2-level pyramid structure for block matching used in Method 1 and Table 2.1 lists the corresponding set of parameters.

2.2.1.4 Hierarchical Block Matching The reliability of motion estimation depends upon the chosen size of blocks and the amount of motion. Large blocks give more reliable motion estimation in the case of large displacement while small blocks are more suitable for small displacement. Based upon this observation, a hierarchical block matching algorithm was proposed by Bierling [7].

In this method, the motion estimation starts with large blocks at the highest level of a hierarchy. From one level to the next, the size of blocks is decreased. The motion estimate is obtained recursively, i.e., at each level of the hierarchy, the

Table 2.2 The typical set of parameters for hierarchical block matching

Parameters at level	1	2	3
Maximum displacement	± 7	± 3	± 1
Block size	64x64	28x28	12x12

resulting estimate serves as an initial guess for the next lower level. The first hierarchical level serves to estimate the large part of motion. whereas the last level serves to estimate the remaining part of the motion. The final estimated motion vector is the sum of the estimates from all hierarchical levels. Figure shows the principle of hierarchical block matching with 3 levels. The associated set of parameters is listed in Table. In this example. the first level is to estimate the major part of the displacement of maximum ±7 pixels using large blocks of size 64 x 64 pixels. At the second level. an additional displacement of maximum ±3 pixels can be estimated using a block size of 28 x 28 pixels. and the second hierarchical level starts motion estimation using the result of the first level. At the third level. the maximum displacement is ±1 pixel and the block size is 12 x 12 pixels. The maximum displacement which can be estimated in total is ±11 pixels.

2.2.2 Pel-recursive Approach

The pel-recursive approach estimates the motion between consecutive frames on a pixel-by-pixel basis. Let $I(x.y.t)$ denote the image intensity at a point $(x.y)$ in the image at time t. We seek to find the corresponding pixel in the previous frame at a displacement $D = (d_x.d_y)$. In the pel-recursive approach. the displacement is estimated recursively. Let $\hat{D}^k = (\hat{d}_x^k.\hat{d}_y^k)$ denote the estimate of the displacement D after the kth iteration . the estimate of D after the $k+1$th iteration. $\hat{D}^{k+1} = (\hat{d}_x^{k+1}.\hat{d}_y^{k+1})$. is obtained by

$$\hat{D}^{k+1} = \hat{D}^k + U^{k+1} \qquad (2.47)$$

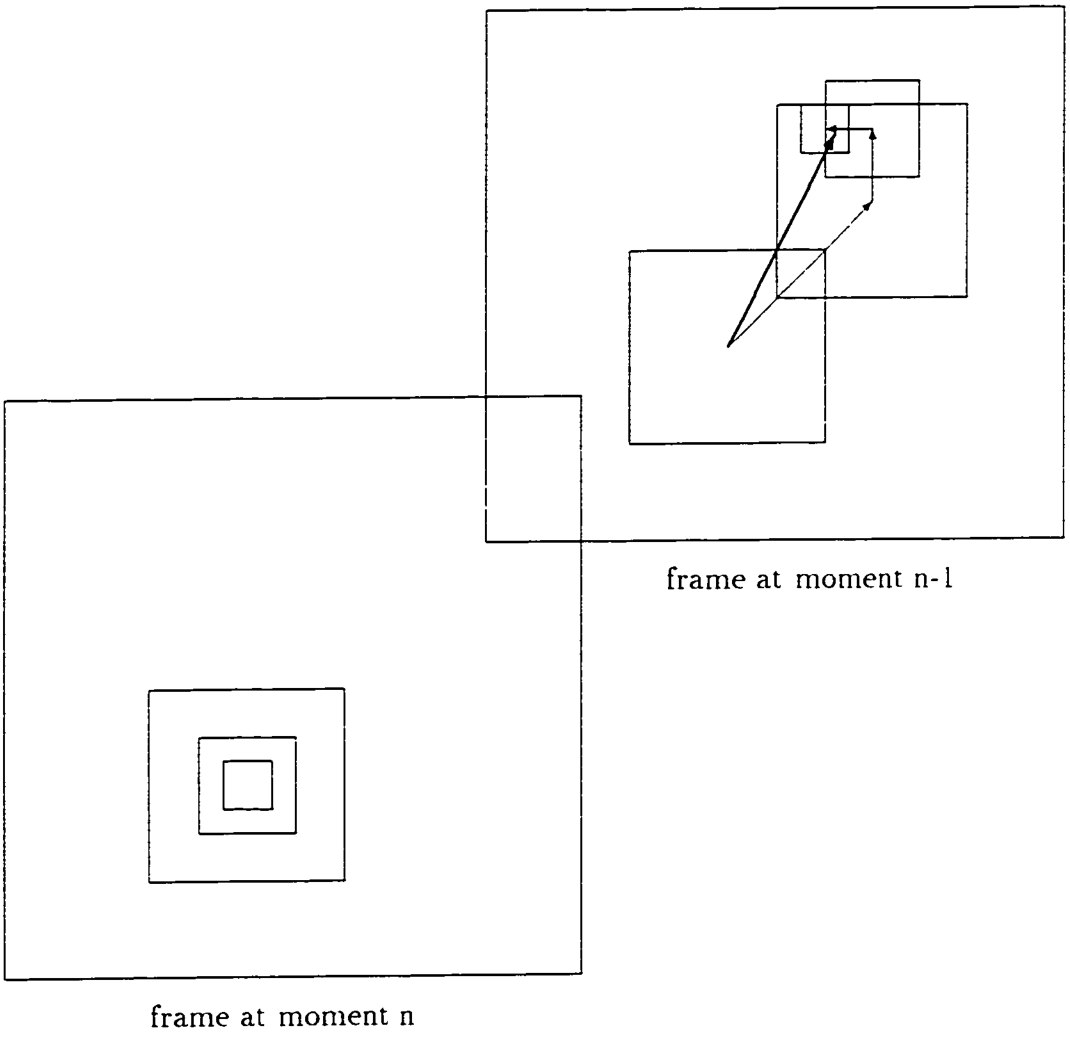


Figure 2.5 Principle of hierarchical block matching

where $\hat{D}^k$ is an initial estimate of $\hat{D}^{k+1}$ and U^{k+1} is the update of $\hat{D}^k$ to make it more accurate.

The criterion used to estimate D is the displacement frame difference (DFD). which is defined as

$$DFD(i.j.\hat{D}^k) = I(x.y.t) - I(x - \hat{d}_x^k.y - \hat{d}_y^k.t - 1) \tag{2.48}$$

By minimizing the DFD. we can maximize the accuracy of the displacement estimate $\hat{D}^k$.

Netravali and Robbins [37] were the first to develop a pel-recursive motion estimation algorithm for image sequence coding. They proposed that the estimated displacement $\hat{D}^k$ be the one which minimizes the square value of the DFD. That is.

$$|DFD(x.y.\hat{D}^k)|^2 \to min \tag{2.49}$$

To solve for the estimate $\hat{D}^k$ recursively. they used the steepest descent method

$$\hat{D}^{k+1} = \hat{D}^k - \frac{1}{2}\epsilon\nabla_D[DFD(x.y.\hat{D}^k)]^2 \tag{2.50}$$

where ϵ is a positive constant. and ∇ is the gradient with respect to the displacement D. The gradient ∇_D can be calculated using the definition of DFD in Equation (2.48) and noting that

$$\nabla_D(DFD(x.y.\hat{D}^k)) = \nabla I(x - \hat{d}_x^k.y - \hat{d}_y^k.t - 1) \tag{2.51}$$

where ∇ is the gradient with respect to x and y. This leads to

$$\hat{D}^{k+1} = \hat{D}^k - \epsilon DFD(x.y.\hat{D}^k)\nabla I(x - \hat{d}_x^k.y - \hat{d}_y^k.t - 1) \tag{2.52}$$

where DFD and ∇I are evaluated by interpolation for nonintegral $\hat{D}^k$.

The choice of the positive constant ϵ requires a compromise. A large value of ϵ yields a quick convergence but a noisy estimate. whereas a small ϵ yields more accurate displacement estimate but takes more processing time. In [37]. ϵ is chosen to be $\frac{1}{1024}$.

The same authors presented an extension of Equation (2.52). in which displacement is estimated by considering DFD at a small neighborhood in the vicinity of the pixel under consideration:

$$\hat{D}^{k+1} = \hat{D}^k - \frac{1}{2}\epsilon\nabla_D \sum_{j\in M} W_j[(DFD(x.y.\hat{D}^k))]^2 \tag{2.53}$$

where W_j are not negative weights and $\sum_{j\in M} W_j = 1$. Although this iteration formula is more complex. it significantly improves the performance of the displacement estimation in those regions where the displacement is spatially uniform.

2.2.3 Discussions

The pel-recursive approach updates its displacement estimation at every pixel and in principle. such algorithms overcome. to a large extent. the problems of multiple moving objects. as well as different parts of an object undergoing different displacement. Practically. however. the partial derivatives involved makes this approach very sensitive to the presence of noise or the fine details in an image. On the other hand. the block-based approach assumes that all pixels within a block have the same motion vector and considers a block of pixels while performing matching. Hence. this approach is less sensitive to various noises. But block matching is very time consuming due to the fact that a block of pixels are involved in matching. Although critical evaluation of block-based approach versus pel-recursive approach has not yet been reported. the block-based approach has dominated the motion estimation employed in image coding area and has been recommended by many video coding standards [26].

CHAPTER 3

MULTIATTRIBUTE FEEDBACK APPROACH

In the correlation-feedback approaches. the estimate of optical flow is repeatedly fedback to the algorithm to compensate the uncertainty of the estimated flow vector. the accuracy of estimation is improved (refer to Equations (2.25) and (2.26)). The utilization of bilinear interpolation reduces the errors caused by the subpixel problem. In this computational framework. less differentiation is required. Hence. this approach is reliable in the presence of noise due to the image acquisition and digitization. However. it has drawbacks. First. it is window oriented. The assumption that the local intensity distribution does not change under motion will be violated if the image area undergoes significant rotation. expansion or if it contains motion boundaries. Second. in applying the motion smoothness constraint. it does not consider the motion discontinuities.

The approach by Weng et al. computes the displacement field by taking multiple image attributes as conservation information. These image attributes are point-based local properties. this approach is point oriented. When imposing local smoothness constraints. this method considers motion discontinuities. However. this approach has some problems. too. First. the image attributes used are intensity. edgeness. positive cornerness and negative cornerness. Among them. edgeness and cornerness need the computation of the spatial gradient of the original image intensity (refer to Equations (2.31) and (2.32)). Due to various noises. it is difficult to estimate the gradients accurately. Hence. the computed attribute images can be noisy. Second. in solving for the displacement field . this approach resorts to differential operation again. Specifically. the estimated displacement vectors are updated based upon the computation of the partial derivatives of the noisy attribute images (refer to Equation (2.39)). Hence. the computational framework is heavily depended on the numerical differentiation. which is considered to be impractical for accurate computation [6]. Third. the algorithm does not address the subpixel problem.

Based upon the comparison between Pan et al.'s and Weng et al.'s approaches. we observe that both advantages and disadvantages exist in two approaches. Furthermore. the advantage in one approach might be utilized to enhance the performance of the other approach. For instance. the incorporation of multiple image attributes and point oriented processing of Weng et al.'s algorithm into Pan et al.'s algorithm may improve the estimation accuracy. while the utilization of the feedback and computational framework of Pan et al.'s algorithm can make Weng et al.'s algorithm less sensitive to various noises. Motivated from this observation. in this paper. we present a new way to compute optical flow field that takes advantages of the approaches by both Pan et al. and Weng et al.. This algorithm has the following characteristics.

1. It is point oriented. Multiple image attributes are computed as conservation information. We use two types of image attributes. One describes the structural information of the point under consideration: the other reflects the textural information of its local neighborhood. These attributes need less derivative operations and are not sensitive to various noises.

2. No differentiation is involved in the computational framework.

3. Feedback techniques is utilized to enhance the estimation accuracy. The estimation is carried out in two steps. In the first step. for each point under consideration. its matching candidates in the second image are determined by the estimated flow vector from the last iteration and its perturbed values. Multiple image attributes are fully employed to determine the optical flow by using the weighted least squared estimation. In the second step. the flow

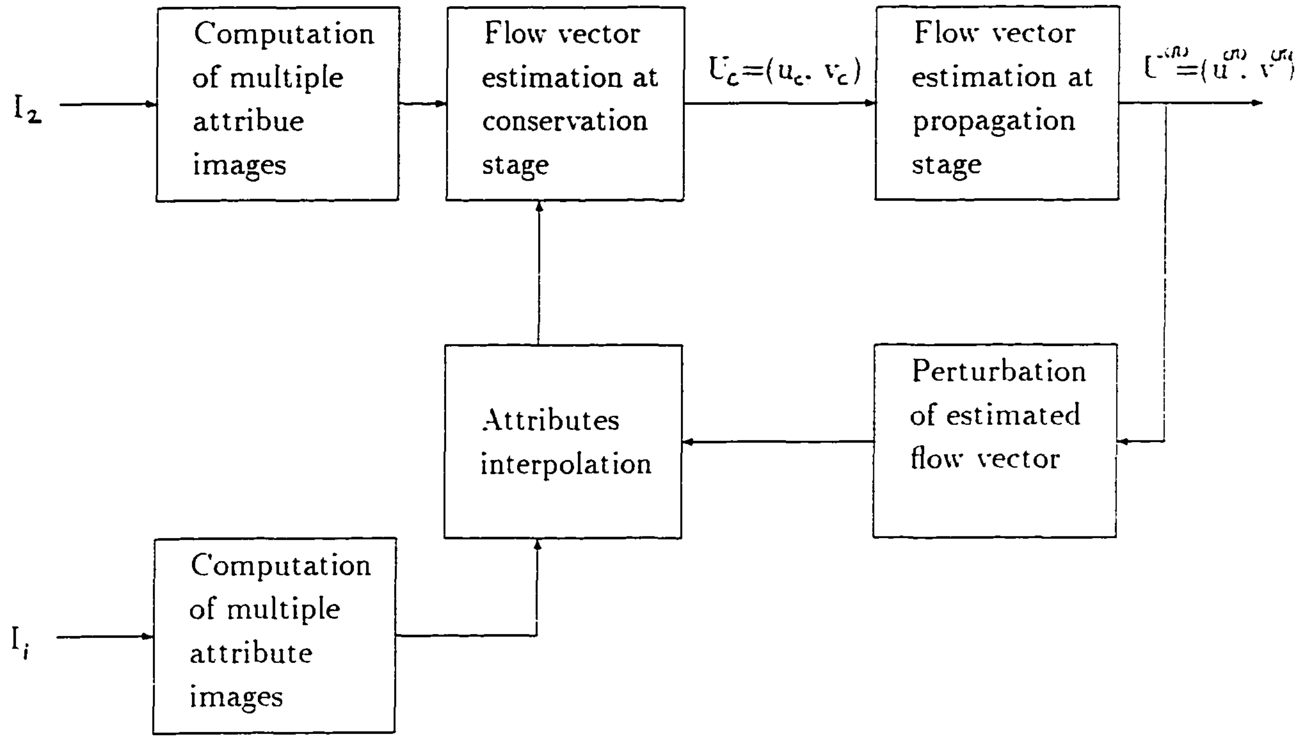

I₂
Computation of multiple attribue images
Flow vector estimation at conservation stage
Uc=(uc, vc)
Flow vector estimation at propagation stage
U(n)=(u(n), v(n))
Attributes interpolation
Perturbation of estimated flow vector
Ii
Computation of multiple attribute images

Figure 3.1 An schematic illustration of the computational framework

estimates are computed as a weighted sum of those over a small neighborhood. The weight considers discontinuities in the flow field.

4. The subpixel problem is considered by using the bilinear interpolation technique.

In the following. the multiconstraint feedback approach is discussed in detail.

3.1 Proposed Framework

Figure 3.1 shows an overview of the proposed computational framework. Let I_2 and I_1 denote two images at moments 2 and 1. respectively. For each image. a set of attribute images will be computed as conservation information. In the framework. the computation of optical flow field is performed iteratively. Each iteration consists of a conservation stage and a propagation stage. At the conservation stage. for each point in the current image I_2. the corresponding matching candidates are determined by the estimated flow vector from the last iteration and its perturbed values. The matching error is calculated by the sum of squared difference. The estimate of flow vector at this stage. denoted by $U_c = (u_c. v_c)$. is computed by weighted least square estimation technique. At the propagation stage. the flow estimate U_c is further improved by using the neighborhood information. At the nth iteration. the output of the propagation stage is denoted by $U^{(n)} = (u^{(n)}. v^{(n)})$. The iteration process will continue until either the predefined iteration number or the predefined accuracy threshold is reached. Besides. an interpolation mechanism is incorporated into the algorithm to reduce the error caused by the subpixel problem.

A full description of every block in Figure 3.1 is given below.

3.1.1 Multiple Motion Invariant Image Attributes

To compute the optical flow field from the two frames of an image sequence. motion invariant attributes are required since the conservation of such attributes can be used as a criterion for matching process. Under the assumption that for a given scene point. the intensity at the corresponding image points remains constant over time. the intensity is motion invariant. However. if the matching is based on intensity only. a point in current image can be matched to any point with the same or similar intensity in the previous image. To reduce the ambiguity in matching. we have to resort to more motion invariant image attributes.

In addition to the intensity. the image attributes used in this work are horizontal edgeness. vertical edgeness. contrast and entropy. The edgeness gives the structural information for matching and is used in [52] too. The two other attributes. contrast and entropy. reflect the textural information about the local neighborhood of the point under consideration [17].

The following are attributes used in our new algorithm.

Intensity

The image intensity at a point (x, y) in an image I, denoted by $A_i(x, y)$, is given by $A_i(x, y) = I(x, y)$.

Horizontal edgeness

The horizontal edgeness at a point (x, y) in an image I, denoted by $A_h(x, y)$, is defined as

$$A_h(x, y) = |\frac{\partial I(x, y)}{\partial x}| \tag{3.1}$$

i.e., the horizontal edgeness is defined as the magnitude of the horizontal component of the gradient of intensity.

Vertical edgeness

The vertical edgeness at a point (x, y) in an image I, denoted by $A_v(x, y)$, is defined as

$$A_v(x, y) = |\frac{\partial I(x, y)}{\partial y}| \tag{3.2}$$

Contrast

The local contrast at a point (x, y), denoted by $A_c(x, y)$, is defined as

$$A_c(x, y) = \sum_{i,j \in s} (i - j)^2 c_{ij} \tag{3.3}$$

where s is a set of distinct gray levels within a 3×3 window centered at the point (x, y), $c_{i,j}$ specifies a relative frequency with which two neighboring points separated horizontally by a distance 1 occur on the 3×3 window, one with gray level i and the other with gray level j.

Entropy

The local entropy at a point (x, y), denoted by $A_e(x, y)$, is given by

$$A_e(x, y) = -\sum_{i \in s} p_i \log p_i \tag{3.4}$$

where s is a set of distinct intensity values within a 3×3 window around the point $(x.y)$. p_i is the probability (relative frequency) of occurrence of the gray level i in the window.

In above defined image attributes. the intensity and edgeness are used in Weng et al.'s algorithm as well. Compared with the negative cornerness and positive cornerness used in Weng et al.'s algorithm. the local contrast and entropy defined by Equations (3.3) and (3.4) need no differentiation at all and therefore are less sensitive to noises in the original image. Besides. these two attributes is inexpensive in computation.

3.1.2 Conservation Stage

Let p or $(x.y)$ denote a point in an image I_2. Let $U'(p) = (u.v)$ denote the flow estimate at the point p. Then the point $(x.y)$ can be viewed as a shifted result of the point $(x - u.y - v)$ in an image I_1. assuming that the times interval between the two moments is a unit. Hence. if the estimated flow vector $U'(p)$ at the point p is accurate. the attributes associated with the two corresponding points should be similar.

To measure the similarity between the attributes of the corresponding points. we use a set of residual functions. The residual function of intensity at the point p. denoted by $r_i(p.U')$. is given by

$$r_i(p.U') = I_2(p) - f(p - U') \qquad (3.5)$$

where $f(\cdot)$ is given by

$$\begin{aligned} f(p - U') &= f(x - u.y - v) \\ &= (1-a)[(1-b) \times I_1(k.l) + b \times I_1(k.l+1)] \\ &\quad +a[(1-b) \times I_1(k+1.l) + b \times I_1(k+1.l+1)] \qquad (3.6) \end{aligned}$$

where $k = int(x - u)$: $l = int(y - v)$: $a = x - u - k$: $b = y - v - l$: $int(\cdot)$ means the integer part of the variable in $(\cdot)$. I_2 and I_1 are the intensity image in the two images. respectively. The residual of horizontal edgeness $r_h(p, U^*)$. that of vertical edgeness $r_v(p, U^*)$. that of contrast $r_c(p, U^*)$. and that of entropy $r_e(p, U^*)$ can be defined. similarly.

In the conservation step. for each point p in the image I_2. we consider a set of points in image I_1. denoted by $p - U^*$. where $U^* = (u, v)$ is given by

$$\begin{aligned} u &\in (u^{(n)} - u^{n)}/2, u^{(n)} - u^{(n)}/4, u^{(n)}, u^{(n)} + u^{(n)}/4, u^{(n)} + u^{(n)}/2) \\ v &\in (v^{(n)} - v^{(n)}/2, v^{(n)} - v^{(n)}/4, v^{(n)}, v^{(n)} + v^{(n)}/4, v^{(n)} + v^{(n)}/2) \end{aligned} \tag{3.7}$$

There are total 5 x 5 candidate points. For each candidate point. the matching error at the point p in the image I_2. denoted by $E(p, U^*)$. is computed as

$$E(p, U^*) = \sum [r_i^2(p, U^*) + r_h^2(p, U^*) + r_v^2(p, U^*) + r_c^2(p, U^*) + r_e^2(p, U^*)] \tag{3.8}$$

where $U^* = (u, v)$ satisfies Equation (3.7).

The same as Equation (2.16). the resulting 5 x 5 matching errors are converted into a response distribution using

$$R(p, U^*) = e^{-kE(p, U^*)} \tag{3.9}$$

where k is chosen so as to make R be a number close to unity.

Then the estimated flow vector at this step. denoted by $U_c^* = (u_c, v_c)$. is calculated as follows according to the weighted least-squares estimation

$$\begin{aligned} u_c &= \frac{\sum_u \sum_v R(p, U^*) u}{\sum_u \sum_v R(p, U^*)} \\ v_c &= \frac{\sum_u \sum_v R(p, U^*) v}{\sum_u \sum_v R(p, U^*)} \end{aligned} \tag{3.10}$$

3.1.3 Propagation Stage

For the true optical flow field. the flow vectors within the same object should be similar. and that belonging to different moving objects should differ from each other.

Based on this fact, the flow information of the local neighboring points within the same region can be used to further improve the estimated flow vector of the point under consideration. We form a window W_l of size $(2M+1) \times (2M+1)$ around the point (x, y) in the image I_2. The flow estimate at the point (x, y) in this step, denoted by $U^{(n+1)} = (u^{(n+1)}, v^{(n+1)})$, is computed as the weighted sum of the flow vectors of the points within the window W_l and is given by

$$\begin{aligned} u^{(n+1)} &= \sum_{s=-M}^{M} \sum_{t=-M}^{M} w(I_2(x, y), I_2(x+s, y+t)) * u_c(s+x, t+y) \\ v^{(n+1)} &= \sum_{s=-M}^{M} \sum_{t=-M}^{M} w(I_2(x, y), I_2(x+s, y+t)) * v_c(s+x, t+y) \end{aligned} \tag{3.11}$$

where $w(\cdot, \cdot)$ is a weighting function. For each point in the window W_l, a weight will be assigned by the weighting function. Let $p'(x+s, y+t)$ denote a point in the vicinity of the point $p(x, y)$, the weight for the point p' is given by

$$w(I_2(x, y), I_2(x+s, y+t)) = \frac{c}{\epsilon + |I_2(x, y) - I_2(x+s, y+t)|} \tag{3.12}$$

where ϵ is a small positive number to prevent the denominator from zeroing, and c is a normalization constant that makes the summation of the weights by Equation (3.12) equal to 1. The weight is determined based on the intensity difference between the point under consideration and its neighboring point. The larger the difference in intensity, the more likely the two points belong to different regions. Therefore, the weight will be small in this case. On the other hand, the flow vector in the same region will be similar since the corresponding weight is large. Thus the weighting function implicitly takes flow discontinuities into account.

3.1.4 Summary of the Algorithm

The following summarizes the procedures of the proposed new algorithm.

1. Perform low-pass filtering on each image to remove noise.

2. Generate attribute images: intensity. horizontal edgeness. vertical edgeness. contrast. and entropy.

3. Set the initial flow field to zero. Set the maximum iteration number and estimation accuracy.

4. For each point under consideration. compute $U = (u, v)$ according to Equation (3.7). Compute the matching error for each matching candidate using Equation (3.8) and transform them to the corresponding response distribution R. using Equation (3.9). Compute the estimate U_c from the probability distribution R using Equation (3.10).

5. Form a $(2M + 1) \times (2M + 1)$ window around the point under consideration. Compute the weight for each point within this window using Equation (3.12). Update the flow vector using Equation (3.11).

6. Decrease the preset iteration number by one: If the iteration number is zero. the algorithm returns with the resulting optical flow field: otherwise. goto Step 7.

7. If the change in flow vector over two successive iterations is less than the predefined threshold. the algorithm returns with the estimated optical flow field: otherwise. goto Step 4.

3.2 Experiments

Recently. a very comprehensive study of various optical flow techniques and comparison of their performance mainly in terms of accuracy have been conducted in [6]. In order to test the performance of our algorithm and compare with other techniques in a more objective. quantitative manner. we choose to work on the same testing sequences and then report the results with the same criterion as used in [6].

Three experimental works are reported here. The image sequences used are Translating Tree 2-D. Diverging Tree 2-D. and Yosemite as shown in Figures 3.2. 3.3 and 3.4. respectively. The first two sequences simulate translating camera motion with respect to a textured planar surface. In the Translating Tree 2-D sequence. the camera moves normal to its line of sight along its x axis. while in the Diverging Tree 2-D sequence. the camera moves along its line of sight. The Yosemite sequence is considered as the most challenging in [6] because of the range of velocities and the occluding edges between the mountains and at the horizon. The motion in the upper right is mainly divergent. the clouds translate to the right with a speed of 1 pixels/frame . while velocities in the lower left are about 4 pixel/frame.

As the same as in [6]. the angular error between the true optical flow vector and an estimated flow vector is used as an error measure. Let $(u_t. v_t)$ and $(u_e. v_e)$ denote the true optical flow vector and the estimated one. respectively. The angular error between the true optical flow $(u_t. v_t)$ and an estimate $(u_e. v_e)$ is given by

$$\eta = \arccos(\hat{V}_t \cdot \hat{V}_e) \tag{3.13}$$

where $\hat{V}_t = \frac{1}{u_t^2+v_t^2+1}(u_t. v_t. 1)$. $\hat{V}_e = \frac{1}{u_e^2+v_e^2+1}(u_e. v_e. 1)$.

To save the computation. all images are compressed by subsampling before computing flow field. Yosemite sequence is subsampled by a factor of 4 in both horizontal and vertical directions and compressed from 316 x 252 to 79 x 63. That is. pixels in every 4x4 region are averaged and become one pixel in the compressed image. The images of the other two sequences are subsampled by a factor of 2 in both directions and are compressed from 150 x 150 to 75 x 75. The same averaging procedure is involved. Obviously. this type of compression is low-pass filtering in nature. Tables 3.1. 3.2 and 3.3 give the experimental results by some typical techniques surveyed in [6] and our algorithm. All methods that compute optical flow field with 100% density are listed in Tables. The reported errors are averaged results over the optical flow field. Besides. the corresponding standard deviations (where

Figure 3.2 The 10th frame of Translating Tree 2-D

sample mean is used to replace the statistical mean) of the measurements are also listed in the Tables. We note that the multiconstraint feedback approach performs sensibly better than all other algorithms listed in the Tables. Specifically. in the first (Translating Tree 2-D) and the third (Yosemite) experiments. our algorithm performs the best. In the second experiment (Diverging Tree 2-D). our algorithm performs the second best. In all three cases. our algorithm outperforms both algorithms by Pan et al. and Weng et al.. based on which our approach is developed.

3.3 Summary and Discussions

The proposed approach is mainly motivated from two newly developed optical flow determination techniques. i.e. Weng et al.'s and Pan et al.'s methods. This approach combines the merits from both algorithms and avoids the disadvantages existing in the two methods.

Compared with Weng et al.'s algorithm. our new method has the following distinctions. First. the multiple image attributes used are different. The image attributes used in our algorithm are image intensity. horizontal edgeness. vertical

Figure 3.3 The 10th frame of Diverging Tree 2-D

Figure 3.4 The 10th frame of Yosemite

Table 3.1 Experimental results on "Translating Tree 2-D" sequence

Techniques	Average Error	Standard Deviation	Density
Horn and Schunk (original)	38.72	27.67	100%
Horn and Schunk(modified)	2.02	2.27	100%
Uras et al.	0.62	0.52	100%
Anandan	4.52	3.10	100%
Singh(step 1, n=2,w=2)	1.64	2.44	100%
Singh(step2,n=2,w=2)	1.25	3.29	100%
Correlation feedback(n=1,w=1)	1.07	0.48	100%
Weng's appraoch	1.81	2.03	100%
New approach	0.55	0.52	100%

Table 3.2 Experimental results on "Diverging Tree 2-D" sequence

Techniques	Average Error	Standard Deviation	Density
Horn and Schunk (original)	12.02	11.72	100%
Horn and Schunk (modified)	2.55	3.67	100%
Uras et al.	4.64	3.48	100%
Anandan(frame 19 and 21)	7.64	4.96	100%
Singh(step 2,n=2,w=2,N=4)	8.60	5.60	100%
Correlation feedback	5.12	2.16	100%
Weng'a approach	8.01	9.71	100%
New approach	4.04	3.82	100%

Table 3.3 Experimental results on "Yosemite" sequence

Techniques	Average Error	Standard Deviation	Density
Horn and Schunk(original)	32.43	30.28	100%
Horn and Schunk(modified)	11.26	16.41	100%
Uras et al.	10.44	15.00	100%
Anadan	15.84	13.46	100%
Singh(step2.n=2.w=2.N=4)	13.16	12.07	100%
correlation feedback	7.93	6.72	100%
Weng's approach	8.41	8.22	100%
New approach	7.54	6.61	100%

edgeness. contrast and entropy. The first three are used in [52] as well. The last two attributes give the textural information about the local neighborhood of the point under consideration. Compared with the negative cornerness and positive cornerness used in [52]. the contrast and entropy need no spatial gradient calculation and are less sensitive to the noises in the original images. Second. the computational framework is quite different. In our algorithm. flow vector is estimated by using the weighted least squared estimation. There is no differential calculation required. In Weng et al.'s algorithm. the estimates of displacement vector are updated based upon the calculation of derivatives of the attribute images. which makes the estimated displacement field sensitive to various noises in the original images. Third. in our new method. feedback technique is utilized to enhance the estimation accuracy.

Our new approach is also quite different from Pan et al.'s algorithm. First. in our algorithm. multiple image attributes are computed as conservation information. while in Pan et al.'s algorithm only the image intensity is assumed to be conserved. Second. our approach is point oriented. while Pan et al.'s algorithm is window oriented. Third. in both Pan et al.'s and our algorithms. the flow estimate at the propagation stage is computed as a weighted sum of the flow vectors of the neighboring points. However. the weight in our algorithm relates to the intensity difference between the point under consideration and the surrounding points and therefore takes motion discontinuities into consideration. while the weight in Pan et al.'s algorithm is simply a Gaussian mask and the motion discontinuities are ignored.

The experimental results show that our proposed approach outperforms in general both Pan et al.'s and Weng et al.'s algorithm in terms of accuracy of optical flow determined.

Computationally speaking. our algorithm is less expensive than Pan et al.'s algorithm. To determine the estimated flow vector in the conservation step. both algorithms have to compute the matching error $E(\cdot,\cdot)$ (refer to Equation (3.8)). In Pan et al.'s algorithm. the $E(\cdot,\cdot)$ is computed as the sum of squared difference between 3×3 correlation windows in two images. while in our algorithm. the $E(\cdot,\cdot)$ is calculated as the sum of squared residual function of five attributes. Hence. our algorithm needs only almost half of the computation required by Pan et al.'s algorithm. We spend an extra amount of computation on multiple attribute images. but achieve a big saving in computing the matching error $E(\cdot,\cdot)$. It is noted that multiple attributes need to be computed only once. while the $E(\cdot,\cdot)$ has to be computed in each iteration.

Compared with Weng et al.'s algorithm. the computation of our method is a little bit more expensive. Among the five image attributes used in our method. i.e.. intensity. horizontal edgeness. vertical edgeness. contrast and entropy. the first three are used in Weng et al.'s algorithm as well. Compared with the negative cornerness and positive cornerness used in [52]. the local contrast and entropy used here need less computation. But the computation of flow vector by using the weighted least squared estimation in our algorithm takes much more time than that by using numerical iteration in [52]. That is. although we achieve some savings in attribute computation. we spend more on the calculation of flow vector.

CHAPTER 4

THRESHOLDING MULTIRESOLUTION BLOCK MATCHING

Motion estimation is of great importance in video coding applications for the exploitation of the high correlation between neighboring frames of an image sequence. Among two basic approaches: block matching and pel recursion (refer to Section 2.2), the block matching approach is more popular and has been adopted by several video compression standards.

The multiresolution technique has been regarded as one of the most efficient methods in block matching [49]. In a so-called top-down multiresolution technique (refer to Section 2.2.1.3), a typical Gaussian pyramid is formed first. Motion search ranges are allocated among different pyramid levels. Matching is initiated at the lowest resolution pyramid level to obtain an initial estimation of motion vectors. The computed motion vectors are then propagated to the next higher resolution level, where it is corrected and again propagated to the next level until the highest resolution level is reached. As a result, a large amount of computation can be saved.

In the multiresolution technique, however, the computed motion vectors at any intermediate pyramid level are all projected to the next higher resolution level. In reality, some computed motion vectors at the lower resolution level may be poor interms of accuracy and have to be further refined, while others are relatively accurate and able to give a satisfactory motion compensation for the corresponding block. From saving computation point of view, it may not be worth for the latter class of motion vectors to be propagated top the next higher resolution level for further processing.

Motivated by the above consideration, we devise and present in the following a new multiresolution block matching method in which a thresholding technique is applied to withhold those blocks whose estimated motion vectors give a satisfactory motion compensation from further processing, thus saving lots of computation.

4.1 The Framework

In this section, the proposed thresholding multiresolution block matching algorithm is discussed in detail.

4.1.1 General Description

Let $I_n(i,j)$ be the frame of an image sequence at current moment n. First, we form two Gaussian pyramids, pyramids n and $n-1$, from image frames $I_n(i,j)$ and $I_{n-1}(i,j)$, respectively. Let the levels of the pyramids be denoted by l, $l = 0, 1, ..., L$, where 0 is the lowest resolution level (top level) and L the full resolution level (bottom level). If (i,j) is the coordinates of the upper left corner of a block at the level l of pyramid n, the block is referred to as block $(i,j)_n^l$. The horizontal and vertical dimensions of a block at the level l are denoted by b_x^l and b_y^l, respectively. Similar to the Method 1 in [49], the size of the block in this work varies with the pyramid levels. That is, if the size of a block at level l is $b_x^l \times b_y^l$, then at level $l+1$ the block size becomes $2b_x^l \times 2b_y^l$. The reason we use the variable block size is that the variable block size method gives more efficient motion estimation than the fixed block size method. The matching criterion used for motion estimation here is the MAD because it requires no multiplication and gives similar performance as the mean square error (MSE) does. The MAD between the block $(i,j)_n^l$ of the current frame and the block $(i+v_x, j+v_y)_{n-1}^l$ of the previous frame at the level l can be calculated as

$$MAD_{(i,j)_n^l}(v_x^l, v_y^l) = \frac{1}{b_x^l \times b_y^l} \sum_{k=0}^{b_x^l-1} \sum_{l=0}^{b_y^l-1} |I_n^l(i+k, j+l) - I_{n-1}^l(i+k+v_x^l, j+l+v_y^l)| \quad (4.1)$$

where $V^l = (v_x^l, v_y^l)$ is one of the candidates of the motion vector of the block $(i,j)_n^l$.

The block matching is initiated at the top pyramid level, and going down towards the bottom level. At each level of the pyramid, a full-search block matching is performed to search for the best match in a predefined search range. To threshold those blocks which have relatively accurate estimated motion vectors, an accuracy threshold is predefined according to the required accuracy for reconstructed image.

If the accuracy threshold is satisfied. the motion estimation for this block will be stopped. Otherwise. the computed motion vector will be propagated into the next higher resolution pyramid level for refinement. In summary. the motion estimation process for a block will be stopped if either the motion estimation satisfies the accuracy threshold or the block reaches the full resolution pyramid level. whichever occurs first.

Figure 4.1 illustrates the data flow and computional procedures of the proposed framework.

4.1.2 Threshold Determination

The threshold used in this work is the MAD for the sake of saving computation. The thresholding value has a direct impact on the performance of the proposed algorithm. A small thresholding value can improve the reconstructed image quality at the expense of increased computational effort. On the other hand. a large thresholding value can reduce the computational complexity. but the quality of the reconstructed image may be degraded.

One possible way to determine the thresholding value. which is used in our many experiments. is as follows. The peak signal-to-noise (PSNR) gives an objective measure of the quality of the motion compensated image. It is defined as

$$PSNR = 10 \log_{10} \frac{255^2}{MSE} \tag{4.2}$$

From the given PSNR. one can find out needed MSE value. A square root of this MSE value can be chosen as a thresholding value. We apply this estimated threshold to the first two images from the sequence. If the resulting PSNR and needed processing time are satisfactory. we use it for the rest of the sequence. Otherwise. we can adjust the threshold a little accordingly and apply it to the second and third images to check the PSNR and processing time. In our experiments this adjusted parameter has been good enough and there is no need for further adjustment. That is. it can

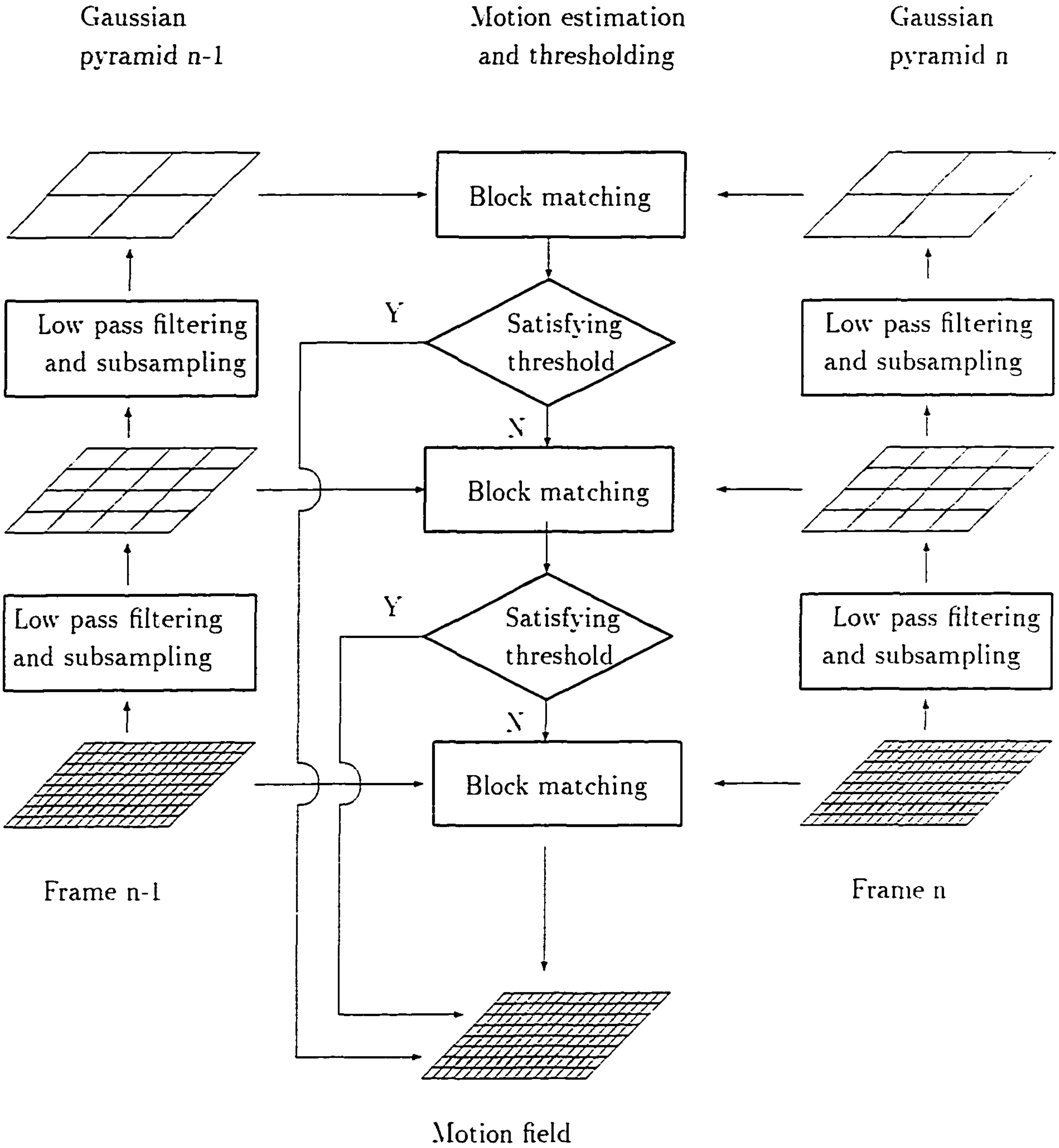


Figure 4.1 Data flow and computional structure

be used for the rest of sequence. As shown in Table 4.1 (refer to Section 4.2), the thresholding value used for "Miss America," "Train," and "Football" sequences are 2, 3, and 4, respectively. It is noted that they are all determined in this fashion and give satisfactory performance, as shown in those three rows marked with "New Method (TH=2)," "New Method (TH=3)" and "New Method (TH=4)," respectively, in Table 4.2, that is, the PSNR experiences only about 0.1 dB loss and the processing time reduces drastically. In our experiments, we also tried the threshold of 3, i.e., the average value of 2, 3, and 4. Refer to those three rows marked with "New Method (TH=3)" in Table 4.2. It is noted that this average threshold 3 has already given satisfactory performance for all of three sequences. Specifically, for "Miss America" sequence, since the threshold increases from 2 to 3, the PSNR loss increases from 0.12 dB to 0.48 dB and the processing time reduction increases from 20% to 38%. For "Football" sequence, since the threshold decreases from 4 to 3, the PSNR loss decreases from 0.08 dB to 0.05 dB and the processing time reduction decreases from 14% to 9%. Obviously, for "Train" sequence, the threshold as well as performance remain the same. One can therefore conclude that the threshold determination may not require much computation at all.

4.1.3 Thresholding

Motion vectors estimated at each pyramid level will be checked to see if they give a satisfactory motion compensation. Let $V^l(i,j) = (v_x^l, v_y^l)$ denote the estimated motion vector for the block $(i,j)_n^l$ at l level of the pyramid n. For thresholding, $V^l(i,j)$ should be directly projected to the bottom level L. The corresponding motion vector for the same block at the bottom level of the pyramid n will be $V^L(2^{(L-l)}i, 2^{(L-l)}j)$ and is given as

$$V^L(2^{(L-l)}i, 2^{(L-l)}j) = 2^{(L-l)}V^l(i,j) \qquad (4.3)$$

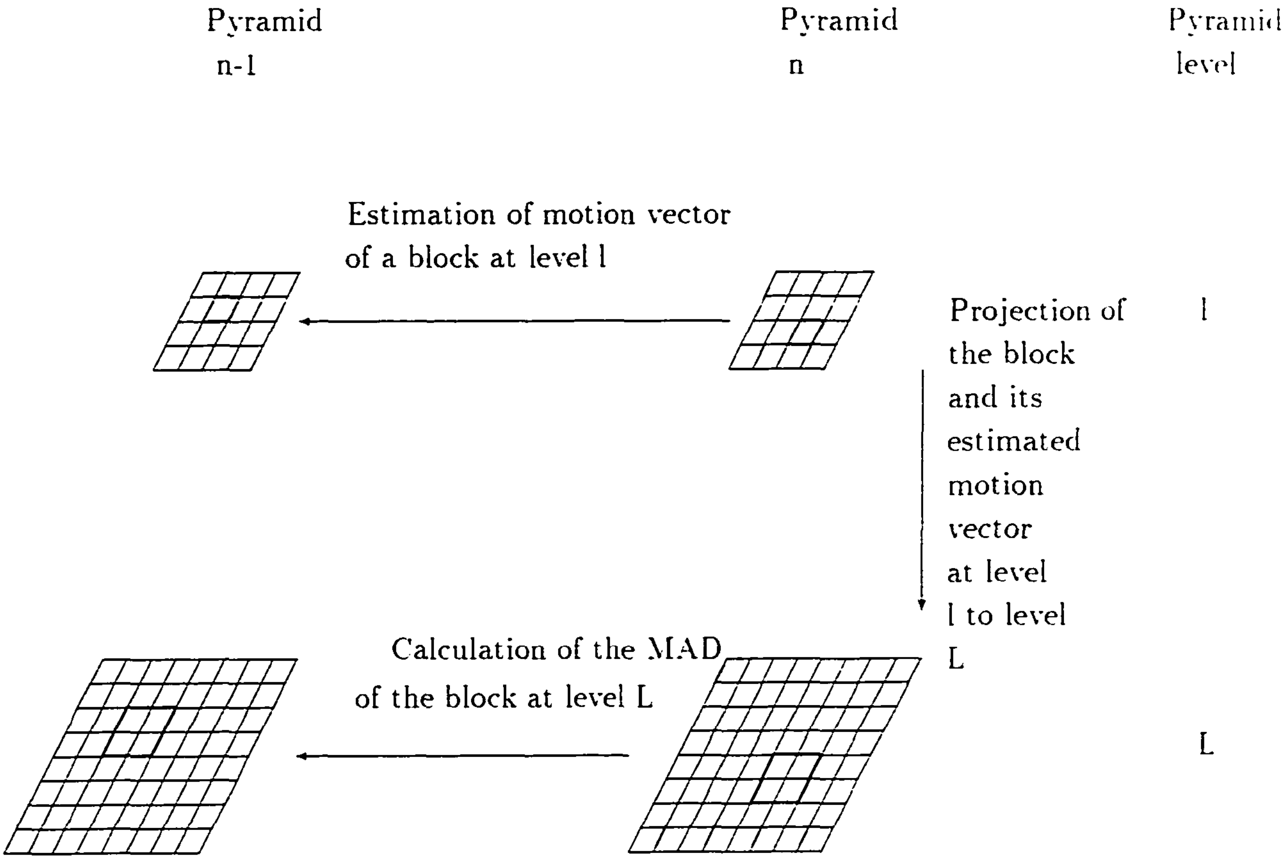


Figure 4.2 Thresholding process

The MAD between the block at the bottom pyramid level of the current frame and its counterpart in the previous frame can be determined according to Equation (4.1), where the motion vector is $V^L = V^L(2^{(L-l)}i, 2^{(L-l)}j)$. This computed MAD value can be compared with the predefined threshold. If this MAD value is less than the threshold, the computed motion vector $V^L(2^{(L-l)}i, 2^{(L-l)}j)$ will be assigned to the block $(2^{(L-l)}i, 2^{(L-l)}j)_n^L$ at the level L in the current frame and motion estimation for this block will be stopped. If not, the estimated motion vector $V^l(i, j)$ at the level l will be propagated to the level $l + 1$ for further refinement.

Figure 4.2 gives an illustration of the above thresholding process.

4.2 Experiments

To verify the effectiveness of the proposed algorithm, extensive experiments have been performed. The performance of the new algorithm is evaluated and compared with that of Method 1 [49], one of the most efficient multiresolution block matching methods, in terms of PSNR, error entropy, motion vector entropy, the number of blocks stopped at the top level versus total number of blocks, and processing time.

These performance indexs are stated below.

(a) The peak-to-peak signal-to-noise ratio (PSNR)

This term, defined by Equation (4.2), gives an objective measure of the accuracy of the reconstructed image.

(b) The error entropy

The error image entropy is the entropy of the difference between the original image and the reconstructed image and is given by

$$H_b = - \sum_{b=-S}^{S} p(b) \log_2 p(b) \qquad (4.4)$$

where b is a distinctive value of the error image and S is its maximum value. The error image entropy gives a lower bound of number of bits per pixel needed in transmision.

(c) The motion vector entropy

This term is given by

$$H_{mv} = - \sum_{mv=-MV}^{MV} p(mv) \log_2 p(mv) \qquad (4.5)$$

where mv is a numbering index of a distinctive motion vector and MV is its maximum value, which can be determined from the total search range and the motion vector accuracy (e.g. pixel or half pixel accuracy) with respect to the full resolution image.

(d) The number of blocks stopped at the top level versus the total number of blocks

This term indicates how many blocks at the top level are withheld from further processing.

(e) The processing time

The processing time is the sum of total number of additions for the evaluation of the MAD involved. and the thresholding operation.

In the experiments. two-level pyramids are used since pyramids of two levels give a better performance for motion estimation purpose [49]. The algorithms are tested on three video sequences with different motion complexities. i.e.. "Miss America." "Train" and "Football." as shown in Figures 4.3. 4.4 and 4.5. "Miss America" sequence has a speaker imposed on a static background and contains less motion. "Train" sequence has more detail and contains a fast moving object (train) [49]. The sequence "Football" contains most complicated motion. Table 4.1 is the list of implementing parameters used in the experiments. Table 4.2 gives the performance of the proposed algorithm. compared with Method 1. Here the motion estimation has half pixel accuracy. All performance measures listed there are averaged for the first 25 frames of the testing sequences.

Each frame of the sequence "Miss America" is of size 360×288 pixels. For convenience. only the central portion of 320×256 pixels is processed. The total number of blocks is 1280. With the operational parameters listed in Table 4.1 (TH=2). 53% of the total blocks (679 blocks) at the top level satisfy the predefined threshold and are not propagated to the bottom level. The processing time needed by the proposed algorithm is 20% less than Method 1 while the PSNR. the error image entropy and the vector entropy are almost the same. Compared with Method 1. we spend an extra amount of computation (around 0.16×10^6 additions) on thresholding operation. but achieve a big savings of computation (around 2.16×10^6 additions) by withholding those blocks. whose MAD values at the full resolution level is less than the predefined accuracy. from further processing. The computational savings comes from here.

The frames of the "Train" sequence are of size 720×288 pixels. and only the central portion of 640×256 pixels is processed. The total number of blocks is 2816. Refer to Table 4.2 (TH=3). about 52%of the total blocks (1465 blocks) are stopped

Figure 4.3 The 10th frame of "Miss America" sequence

at the top level. The processing time has been reduced about 17% by the new algorithm, compared with Method 1. The PSNR, the error entropy and the vector entropy are almost the same.

The frames of sequence "Football" are of size 720×480 pixels, and only the central portion of 640×448 pixels is processed. The total number of blocks is 4928. With the operational parameters listed in Table 4.2 (TH=4), about 38% of the total blocks (1873 blocks) are stopped at the top level. The processing time is about 14% less than that required by Method 1. The PSNR, the error entropy and the vector entropy are almost the same.

In addition to the objective measures, we also compare the reconstructed images by Method 1 and New Method subjectively. For three testing sequences, i.e., "Miss America," "Train," and "Football,", we virtually cannot see any difference between the reconstructed images obtained by Method 1 and New Method.

In summary, it is clear that in all three different testing sequences, our algorithm works faster than the existing top-down multiresolution block matching algorithm while achieving almost the same quality of the reconstructed image.

Figure 4.4 The 10th frame of "Train" sequence

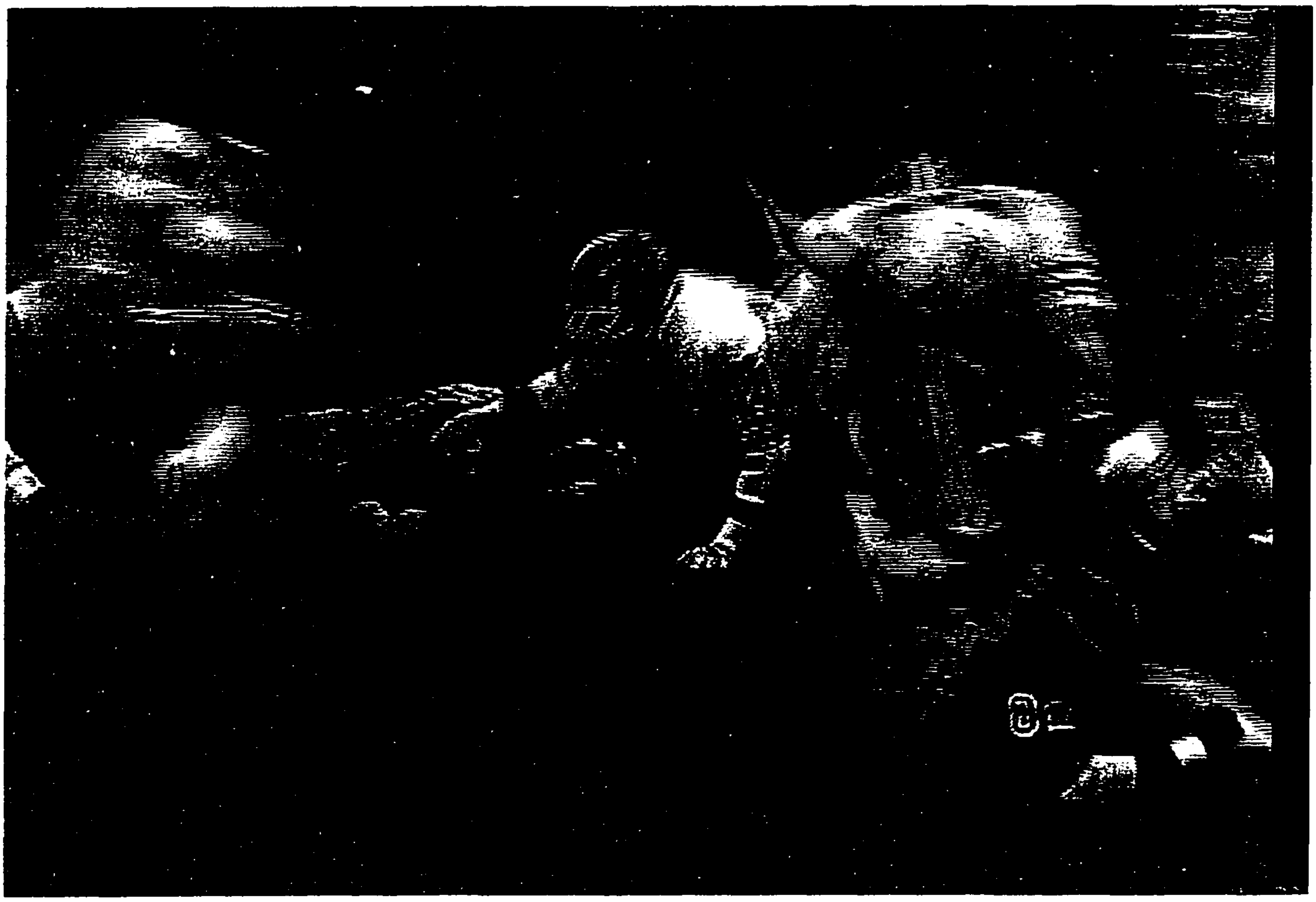


Figure 4.5 The 10th frame of "Football" sequence

Table 4.1 Parameters for testing sequences

Parameters at level	Low resolution level	Full resolution level
" Miss America "		
Search range	3x3	1x1
Block size	4x4	8x8
Thresholding value	2	None (Not applicable)
" Train "		
Search range	4x4	1x1
Block size	4x4	8x8
Thresholding value	3	None (Not applicable)
" Football "		
Search range	4x4	1x1
Block size	4x4	8x8
Thresholding value	4	None (Not applicable)

Table 4.2 Experimental results on testing sequences

	PSNR (dB)	Error entropy (bits/pixel)	Vector entropy (bits/vector)	No. of blocks stopped at top level/ No. of total blocks	Processing times (No. of additions. 10^6)
" Miss America " sequence					
Method 1 [49]	38.91	3.311	6.02	0/1280	10.02
New Method (TH=2)	38.79	3.319	5.65	679/1280	8.02
New Method (TH=3)	38.43	3.340	5.45	487/1280	6.17
" Train " sequence					
Method 1 [49]	27.37	4.692	6.04	0/2816	22.58
New Method (TH=3)	27.27	4.788	5.65	1465/2816	18.68
" Football " sequence					
Method 1 [49]	24.26	5.379	7.68	0/4928	30.06
New Method (TH=4)	24.18	5.483	7.58	1873/4928	25.9
New Method (TH=3)	24.21	5.483	7.57	1456/4928	27.1

4.3 Conclusions

The existing multiresolution block matching technique such as the top-down pyramid technique propagates all the motion vectors estimated at a lower resolution level to its next higher resolution level for refinement no matter whether the computed motion vector gives a satisfactory motion compensation or not. Based on this observation, we present in this paper a new thresholding multiresolution block matching algorithm so that motion vectors computed at lower resolution level will be treated differently. According to the motion compensation performance. those blocks satisfying the predefined accuracy criterion are withheld from further processing. and a large amount of computation is saved. Three experiments with different motion complexities have shown that the proposed algorithm works well. It largely reduce the processing time ranging from 14% to 20%. compared to the fastest existing multiresolution technique. while maintaining almost the same quality of the reconstructed image.

CHAPTER 5

THRESHOLDING HIERARCHICAL BLOCK MATCHING

Block-based motion estimation approach has donimated various video codecs due to its simplicity and easy implementation. Full search block matching is known to give an optimal estimation at the expense of enormous amount of computation. To improve the computational complexity. many efficient search techniques have been proposed such as the three-step search. multiresolution block matching and so on. Among them. the multiresolution technique is considered to be the very efficient.

It is noted that all the above mentioned block matching techniques. i.e.. full search block matching. three-step search. and multiresolution block matching. treat every region in an image domain indiscriminately no matter whether the region under consideration contains complicated motion or not. The complexity of motion for different regions within an image is usually different. Some regions may contain complex motion. while others may have relatively static background or slow motions. For instance. a typical videoconferencing scene is an image of a speaker imposed on a static background. Except a limited amount of complex motion such as facial expression changes. other motions are relatively slow. the background regions even do not have any motion. To use the computing resources efficiently. different computation efforts should be made for regions containing different amount of motion. Regions experiencing complex motion deserve more computational time. but for regions with slow motion. the search procedure can be simplified to expedite the motion estimation process.

Motivated by the above consideration. we devise and present in this chapter a new hierarchical block matching algorithm in which a thresholding technique is applied to withhold those regions containing less amount of motions from further processing. thus saving computation drastically. In the following. Section 5.1 gives a general description of the new framework and algorithm. and then several issues in its implementation are discussed in detail. Section 5.2 demonstrates an extensive experiments to verify the algorithm. Section 5.3 presents conclusions and some discussions.

5.1 New Approach

In this section. the new hierarchical block matching algorithm using thresholding is presented. First. we give a general description of the algorithm. and then several issues in its implementation are discussed in detail.

5.1.1 General Strategy

In order to expedite the motion estimation process and save computation by simplifying the search procedure for regions containing less motion. we resort to the hierarchical structure in this new algorithm.

In the following. let $I_n(i.j)$ denote the frame of an image sequence at present moment n. We first form a block hierarchy. Let the different levels of the hierarchy be denoted by l. $l = 0. 1. \cdots . L$. where 0 is the top level and L the bottom level. If $(i.j)$ is the coordinate of the upper left corner of a block at the level l in the hierarchy. the block is referred to as block $(i.j)^l$. If the horizontal and vertical dimensions of the blocks at the level l are denoted by b^l_x and b^l_y. respectively. at the level $l+1$ the dimensions of the blocks will be

$$\begin{aligned} b_x^{l+1} &= \frac{b_x^l}{2} \\ b_y^{l+1} &= \frac{b_x^l}{2} \end{aligned} \tag{5.1}$$

That is. from the top to the bottom level in the hierarchy. the block sizes are decreased. Figure 5.1 gives an illustration of the hierarchical structure used in our work. As shown. a block at one level corresponds to four blocks at the next level and all levels are of the same resolution. This is quite different from a pyramid structure.

Motion estimation is performed from the top to the bottom level of the hierarchy. At each level. a separate search procedure with different sets of parameters

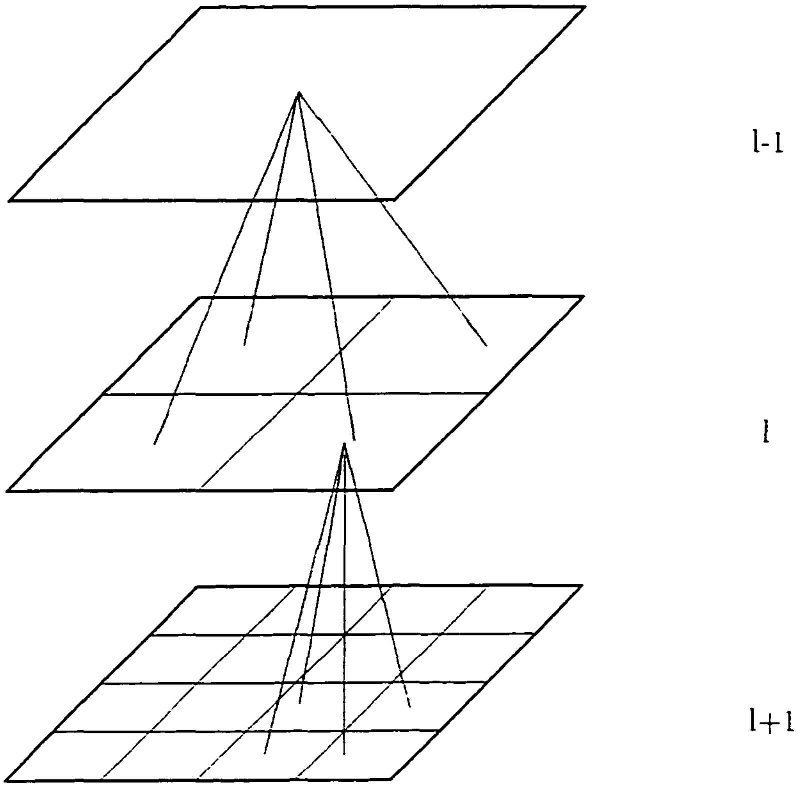


Figure 5.1 An illustration of the hierarchical structure.

is carried out. For a block $(i,j)^l$ at level l in the current frame. We look for a block of pixels in the previous frame that gives the best match in terms of the mean absolute difference (MAD) (refer to Section 2.2.1).

It has been noted that for some blocks. the motion vectors estimated at the intermediate levels of the hierarchy give a satisfactory motion compensation. Therefore. it is inefficient to put these blocks into the next level for further processing. Based on this consideration. in this work a set of accuracy thresholds is predefined according to the required accuracy for reconstructed images. The computed motion vector $V^l(i,j)$ (refer to Section 4.1.3) will be checked to see if it satisfies the predefined threshold. That is. the MAD value associated with the computed motion vector $V^l(i,j)$ is compared with the threshold. If the MAD value is less than the threshold. the estimated motion vector $V^l(i,j)$ will be assigned to the block $(i,j)^l$. and the motion estimation for the block will be stopped.

On the other hand. if the accuracy threshold is not satisfied. the block $(i,j)^l$ will be propagated into level $l+1$ in the hierarchy. According to the Equation (5.1). the block $(i,j)^l$ corresponds to four blocks at the level $l+1$. The computed motion vector $V^l(i,j)$ will be assigned to those four blocks as follows:

$$\begin{aligned} V^{l+1}(i,j) &= V^l(i,j) \\ V^{l+1}(i+\tfrac{b_x^l}{2},j) &= V^l(i,j) \\ V^{l+1}(i,j+\tfrac{b_y^l}{2}) &= V^l(i,j) \\ V^{l+1}(i+\tfrac{b_x^l}{2},j+\tfrac{b_y^l}{2}) &= V^l(i,j) \end{aligned} \tag{5.2}$$

At level $l+1$. the motion estimation for four blocks will be carried out with the above assigned motion vector as an initial guess.

The motion estimation is initiated at the top level of the hierarchy and is going down towards the bottom level. By thresholding. motion estimation for blocks with less motion will be terminated at some intermediate levels. thus saving computation. Blocks stopped at upper levels are of sizes larger than those stopped at lower levels. Hence. the final motion field consists of blocks of different sizes. The motion estimation for the whole image will be terminated if motion estimation of each block either satisfies the accuracy threshold or reaches the bottom level of the hierarchy.

5.1.2 Block Size in the Hierarchy

In the extreme. the block size can be as large as the full image at the top level of the hierarchy and as small as a single pixel at the bottom level. But this kind of hierarchy structure makes the matching inefficient. On one hand. taking the whole image as a single block and searching for a corresponding block in the previous frame is almost certainly a waste of effort. unless the sequence is static at that particular instant. On the other hand. a block of very small size such as a single pixel does not necessarily result in a better motion estimation. The smaller the block size. the higher the probability that there are blocks in the previous frame having a very similar or identical pattern of the small block. This can cause mismatching. Besides. noise will affect matching more when the block size is too small.

In practice. it is desirable to make the block size at the top level of the hierarchy smaller than the full image and the block size at the bottom level larger than a single pixel. As listed in Tables 5.1. 5.3 and 5.5. the actual block sizes in a 4-level hierarchy used in our extensive experiments are 64x64. 32x32. 16x16. 8x8.

5.1.3 Thresholding

The threshold used in this work is the MAD for the sake of saving computation. It has a direct impact on the performance of the proposed algorithm. Blocks at the upper levels of the hierarchy are of large sizes. Large blocks deserve more chances to contain nonuniform motions. The use of small thresholding value helps to detect complex motions embeded in large blocks and splits these blocks for further processing. Thus compared with large thresholding value. the use of small one makes the motion estimation for large blocks relatively accurate. On the other hand. however. the sizes of blocks at the lower levels of the hierarchy are relatively small. There are many circumstances under which blocks containing complex motion cannot find a good match in the previous frame such as occlusion and disocclusion. In these cases. forcing blocks into further processing by the use of small thresholding value cannot result in a better motion estimation but more computation. To take both the quality of the reconstructed image and computational complexity into consideration. we use variable thresholding technique here. That is. the thresholding values vary with the hierarchical levels. For blocks at the upper levels. we use a small thresholding value while for blocks at the lower levels. we use a large one. Specifically. let T^l denote the thresholding value at level l of the hierarchy. $l = 0. 1. \cdots. L$. Then T^l will be given by

$$T^l = c \cdot 2^l \tag{5.3}$$

where c is a parameter. As seen. the thresholding value at one level is twice as large as that at its immediate upper level.

One possible way to determine the parameter c. which is used in our many experiments. is as follows. The peak signal-to-noise (PSNR) gives an objective measure of the quality of the motion compensated image. It has been defined in Section 4.1.2 and is rewritten below

$$PSNR = 10 \log_{10} \frac{255^2}{MSE} \tag{5.1}$$

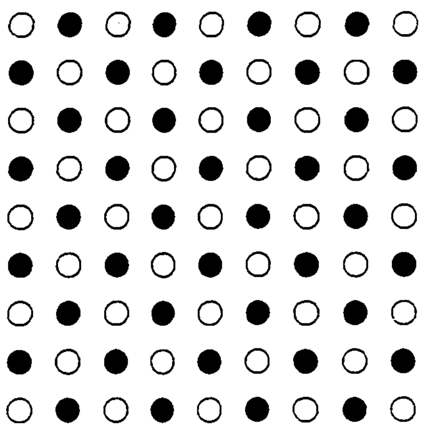
From the given PSNR. one can find out needed MSE value. A square root of this MSE value can be chosen as an initial value of c. We apply this parameter c to the top level of the first two images from the sequence. If the resulting PSNR and needed processing time are satisfactory. we use it for the rest of the sequence. Otherwise. we can adjust the parameter c a little accordingly and apply it to the second and third images to check the PSNR and processing time. In our experiments this adjusted parameter has been good enough and there is no need for further adjustment. That is. it can be used for the rest of sequence. It is noted that the same procedure has been used in Section 4.1.2.

The typical thresholding values used in our extensive experiments have been listed in Tables 5.1. 5.3 and 5.5.

5.1.4 Allocation of Search Range

It is clear that in the proposed algorithm. motion estimation is conducted at different hierarchical levels. At each level. a separate search procedure is performed.

Let D_{max} denote the maximum displacement which can be estimated. It would be very time-consuming to search for motion vectors within the search range $\pm D_{max}$ at each level and the search range should be allocated among different levels. On the other hand. the larger the block size. the less possible the block contains large motion. To take the above two factors into consideration. we assign the search range nonuniformly to each level. We make the search range decrease from the top to the



● — points involved in the computation of the MAD

Figure 5.2 Subsampling procedure

bottom level of the hierarchy while the sum of the search ranges for all levels remains equal to the maximum displacement for the image.

Let D^l_{max} denote the maximum displacement at level l, $l = 0, 1, \cdots, L$, then, we have the following:

$$D^0_{max} \leq D^1_{max} \leq \cdots \leq D^l_{max} \leq \cdots \leq D^L_{max} \tag{5.5}$$

and

$$D^0_{max} + D^1_{max} + \cdots + D^l_{max} + \cdots + D^L_{max} = D_{max} \tag{5.6}$$

Typical search ranges used in our implementation have been given in Tables 5.1, 5.3, and 5.5.

5.1.5 Subsampling

In the evaluation of the matching criterion MAD, all pixels within the block are involved. In order to further reduce the computational effort, a subsampling inside the block is performed. As shown in Figure 5.2, every other pixel (horizontally and vertically) inside the block is taken into account for the evaluation of the matching criterion. When the subsampling procedure is applied. instead of Equation (2.42). the MAD value can be calculated as

$$MAD_{(i.j)}(v_x^l. v_y^l) = \frac{1}{\frac{b_x^l}{2} \times \frac{b_y^l}{2}} \sum_{k=0}^{\frac{b_x^l}{2}-1} \sum_{l=0}^{\frac{b_y^l}{2}-1} |I_n(i+2k. j+2l) - I_{n-1}(i+2k+v_x^l. j+2l+v_y^l)| \tag{5.7}$$

Obviously. by using the subsampling technique. the computation is reduced by a factor of 4. However. since 3/4 of the pixels in the block are not involved in the matching computation. the use of such subsampling procedure may affect the accuracy of the motion vectors. especially for blocks of small size. In the algorithm. this subsampling procedure only applies to blocks at the top two levels in the hierarchy where the block size is large enough such that the matching accuracy will not be seriously affected.

5.1.6 Summary of the Algorithm

The algorithm is summarized below.

1. Apply Gaussian filter to original images to remove various noises.

2. Define a hierarchical structure as illustrated in Figure 5.1. Label the levels of the hierarchy as level l. $l = 0. 1. \ldots. L$: level 0 is the top level and level L the bottom level.

3. Allocate the search range among the different hierarchical levels. as discussed in Section 5.1.4.

4. Set the level to the top. i.e.. $l = 0$. and set the block motion vectors at level 0 to zero.

5. For blocks at level l in the current frame. search for a best match within a predefined search range at the same level in the previous frame by the full-search block matching. The matching criterion is the MAD.

6. If $l = L$. the procedure returns with the resulting motion field: otherwise. go to step 7.

7. Threshold the block motion vectors computed at the level l. If the MAD value given by the estimated motion vector is less than the threshold. motion estimation for this block will be stopped: otherwise. go to the next step.

8. Split the block into four subblocks with equal size. That is. propagate the block into the next lower level in the hierarchy. Assign the computed motion vector of the block to its corresponding subblocks at level $l + 1$. This motion vector gives an approximate estimation of the motion vector at level $l + 1$ and serve as an initial guess for the motion estimation. The block matching is conducted at level $l + 1$. Go to step 6.

5.2 Experiments

To verify the effectiveness of the proposed new algorithm. extensive experiments have been performed. The new algorithm is implemented in two different ways: one (New Method 1) does not employ the subsampling procedure: the other (New Method 2) does. The performance of New Methods 1 and 2 are evaluated and compared with the full-search block matching (FBM) and Method 1 in [49]. in terms of PSNR. error image entropy. motion vector entropy. the number of motion vectors. and processing time. The definition of the first three terms have been given in Chapter 4. The number of motion vectors is the total number of motion vectors in the final motion field. The processing time here is the total number of additions needed for evaluation of the MAD. The amount of thresholding operation is negligible compared with that needed by the MAD evaluation and is hence not counted in.

In the experiments. the algorithms are tested on three groups of video sequences containing different motion complexities.

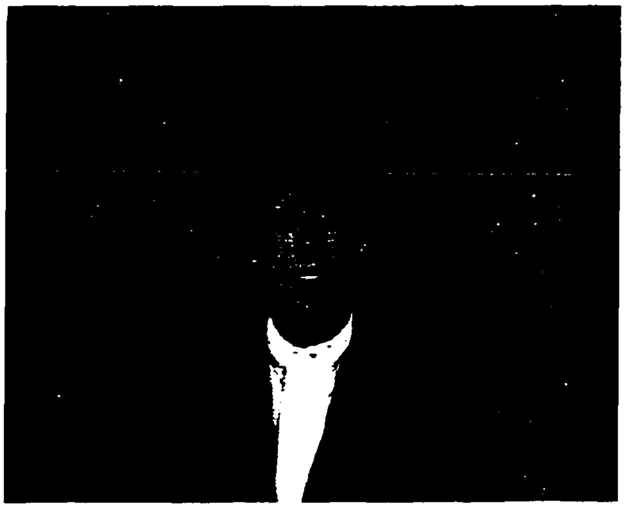


Figure 5.3 The 10th frame of "Claire" sequence

Figure 5.4 The 10th frame of "Salesman" sequence

Group 1

Group 1 of the testing sequence contains typical videoconferencing sequences. i.e.. "Claire." "Miss America." and "Salesman." as shown in Figures 5.3. 4.3 and 5.4. Each frame of the sequences is of size 360×288 pixels. For convenience. only the central portion of 320×256 pixels is processed. The motion estimation has one pixel accuracy. Table 5.1 lists the implementation parameters used in this experiment. Table 5.2 gives the performance of New Methods 1 and 2. compared to the FBM and Method 1 in [49]. averaged for the first 25 frames of the testing sequences. It is noted that. great saving has been achieved in both processing time and the number of resulting motion vectors by our algorithms. The superiority over the FBM is obvious. Compared with Method 1 in [49]. the processing time needed by New Method 1 (New Method 2) is 62% (83%) less in the case of "Claire". 26% (80%) less in the case of "Miss America". and 9% (47%) less in the case of "Salesman" while maintaining almost the same quality of reconstructed images (for New Method 1. loss in PSNR is about 0.2 dB and for New Method 2. loss in PSNR is about 0.7 dB). Furthermore. the number of final resulting motion vectors is substantially less than Method 1 [49] (ranging from 88% to 99% less for the three sequences). The performance of New Methods 1 and 2 is slightly different. Due to the subsampling. New Method 2 is faster than New Method 1 as expected. and gives 0.5-0.7 dB decrease in the PSNR for the three testing sequences.

As a whole. it is clear that our algorithm outperforms both full-search block matching and top-down pyramid technique in this group of experiments as discussed above.

Group 2

The experiments are also conducted on the other video sequences which contain more complicated image details and motion. The sequence of "Train" (see Figure 4.4) is of detailed images with a fast moving object (train). The frames of "Train" sequence are of size 720 × 288 pixels. and only the central portion of 704 × 256 pixels is processed. The motion estimation has a half pixel accuracy.

Table 5.3 lists the parameters used in applying the proposed algorithm to the "Train" sequence. Table 5.4 illustrates the results achieved in the experiments. As seen. for the "Train" sequence. compared with the FBM and top-dowm pyramid technique. a saving in both the number of motion vectors and processing time by the proposed algorithm has been achieved. But the saving is not as much as obtained in the case of the videoconferencing sequences as reported in Group 1. Besides. the quality of the reconstructed images deteriorates. specifically. the PSNR decreases by 1.02 dB and 2.35 dB for New Methods 1 and 2. respectively. Subjectively. this deterioration is obvious in the reconstructed images shown in 5.5 and 5.6. Figure 5.5 is the motion compensated image by Method 1 while Figure 5.5 is the motion compensated image by New Method 1. Compared with the orignal image Figure 4.4. there exist some distortions in both Figures 5.5 and 5.6. In Figure 5.5. the head of the train is slightly blurred while in Figure 5.6. it is almost disappeared. There is a serious distortion in Figure 5.6.

Group 3

The "Football" sequence in this group contains most complicated motion among all sequences tested. The frames of the sequence "Football" (see Figure 4.5) are of size 720 × 480 pixels. and only the central portion of 704 × 448 pixels is processed. The motion estimation has a half pixel accuracy. Tables 5.5 and 5.6 is the implementation parameters and experimental results by the proposed algorithm. respectively. For the "Football" sequence. our algorithm does not seem as effective as in the other experiments. Figures 5.7 and 5.8 are the motion compensated image by Method 1 and motion compensated image by New Method 1. respectively. Compared with the original image in Figure 4.5. one can tell that the distortions in Figure 5.8 are a little more severe than that in Figure 5.7.

Table 5.1 Parameters for videoconferencing sequences

Parameters at level	1	2	3	4
Search range	1x1	1x1	2x2	3x3
Block size	64x64	32x32	16x16	8x8
Thresholding value	2	4	8	16

Figure 5.5 The motion compensated 10th frame of "Train" by Method 1

As a conclusion, for sequences having less motion such as videoconferencing sequences, our proposed algorithm works more efficient than the FBM and top-down pyramid technique, while for sequences containing complex motion, the new algorithm loses the superiority.

5.3 Conclusions and Discussions

The existing block matching techniques such as full-search block matching and top-down pyramid treat every region in an image domain indiscriminately no matter whether the region under consideration contains complicated motion or not. Motivated from this observation, we present in this chapter a new thresholding hierarchical block matching algorithm so that different computational efforts may be

Table 5.2 Experiment results on videoconferencing sequences

	PSNR (dB)	Error entropy (bits/pixel)	Vector entropy (bits/vector)	Number of final blocks	Processing times (No. of operations. 10^6)
" Claire "					
FBM	41.13	2.32	1.82	1024	41.5
Method 1 [49]	40.92	2.33	1.83	1024	7.44
New Method 1	41.07	2.32	1.62	13	2.79
New Method 2	40.58	2.35	1.59	12	1.28
" Miss America "					
FBM	37.71	3.64	3.45	1024	41.5
Method 1 [49]	37.51	3.65	3.61	1024	7.44
New Method 1	37.32	3.69	3.0	81	5.51
New Method 2	36.92	3.73	2.37	38	1.44
" Salesman "					
FBM	35.81	3.82	3.63	1024	41.5
Method 1 [49]	35.62	3.84	3.81	1024	7.44
New Method 1	35.36	3.91	2.89	113	6.75
New Method 2	34.76	4.13	2.61	77	3.79

Table 5.3 Parameters for "Train" sequence

Parameters at level	1	2	3	4
Search range	1x1	2x2	2x2	4x4
Block size	64x64	32x32	16x16	8x8
Thresholding value	3	6	12	24

Table 5.4 Experiment result on "Train" sequence

	PSNR (dB)	Error entropy (bits/pixel)	Vector entropy (bits/vector)	Number of final blocks	Processing times (No. of operations. 10^6)
FBM	27.75	4.69	6.02	4096	442.9
Method 1 [49]	27.51	4.71	6.33	4096	19.9
New Method 1	26.49	4.82	5.54	3567	18.21
New Method 2	25.16	4.97	4.78	2416	16.12

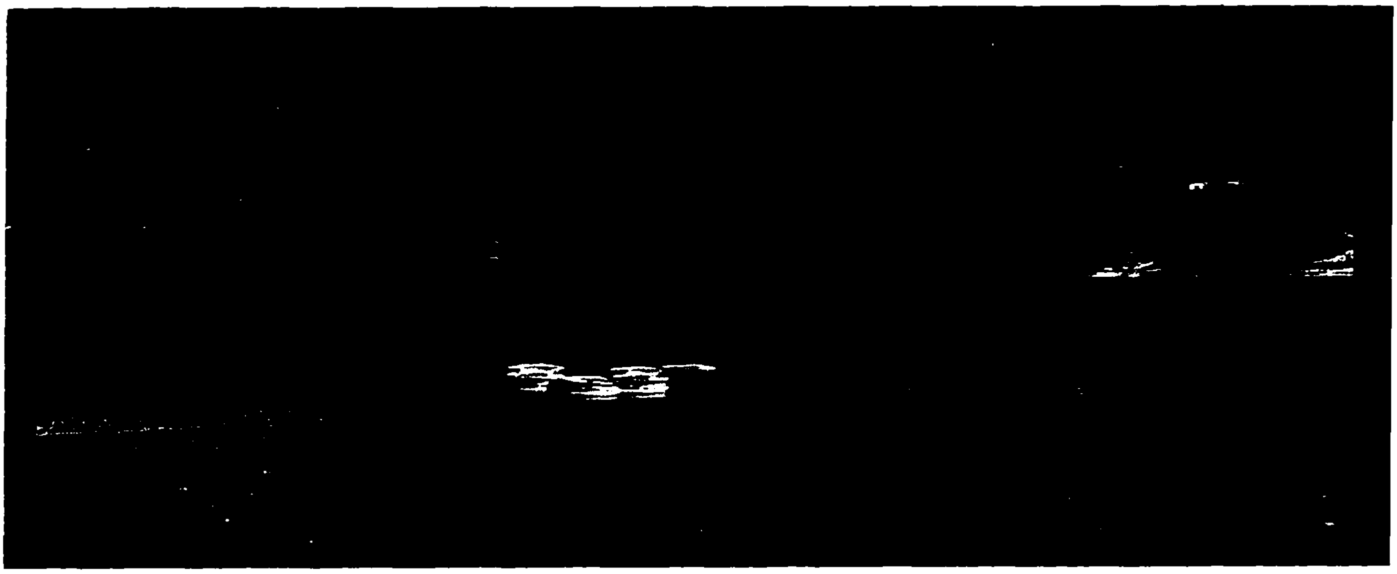


Figure 5.6 The motion compensated 10th frame of "Train" by New Method 1

Table 5.5 Parameters for "Football" sequence

Parameters at level	1	2	3	4
Search range	1x1	2x2	2x2	4x4
Block size	64x64	32x32	16x16	8x8
Thresholding value	4	8	16	32

Table 5.6 Experimental results on "Football" sequence

	PSNR (dB)	Error entropy (bits/pixel)	Vector entropy (bits/vector)	Number of final blocks	Processing times (No. of operations. 10^6)
FBM	24.61	5.46	7.54	4096	891.5
Method 1 [49]	24.29	5.48	7.86	4096	30.1
New Method 1	22.33	5.71	7.22	3889	29.24
New Method 2	20.41	5.92	6.13	3216	26.12

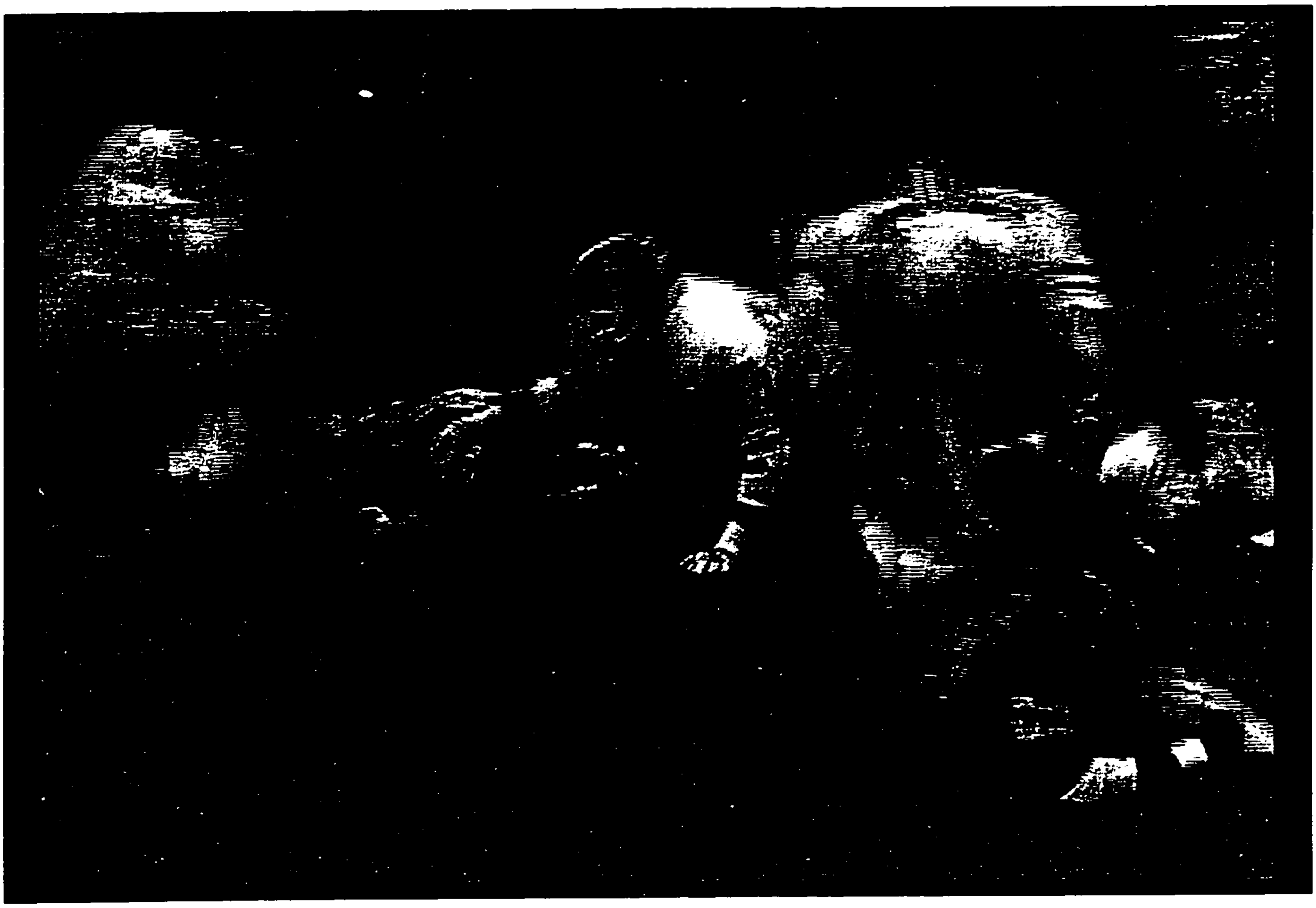


Figure 5.7 The motion compensated 10th frame of "Football" by Method 1

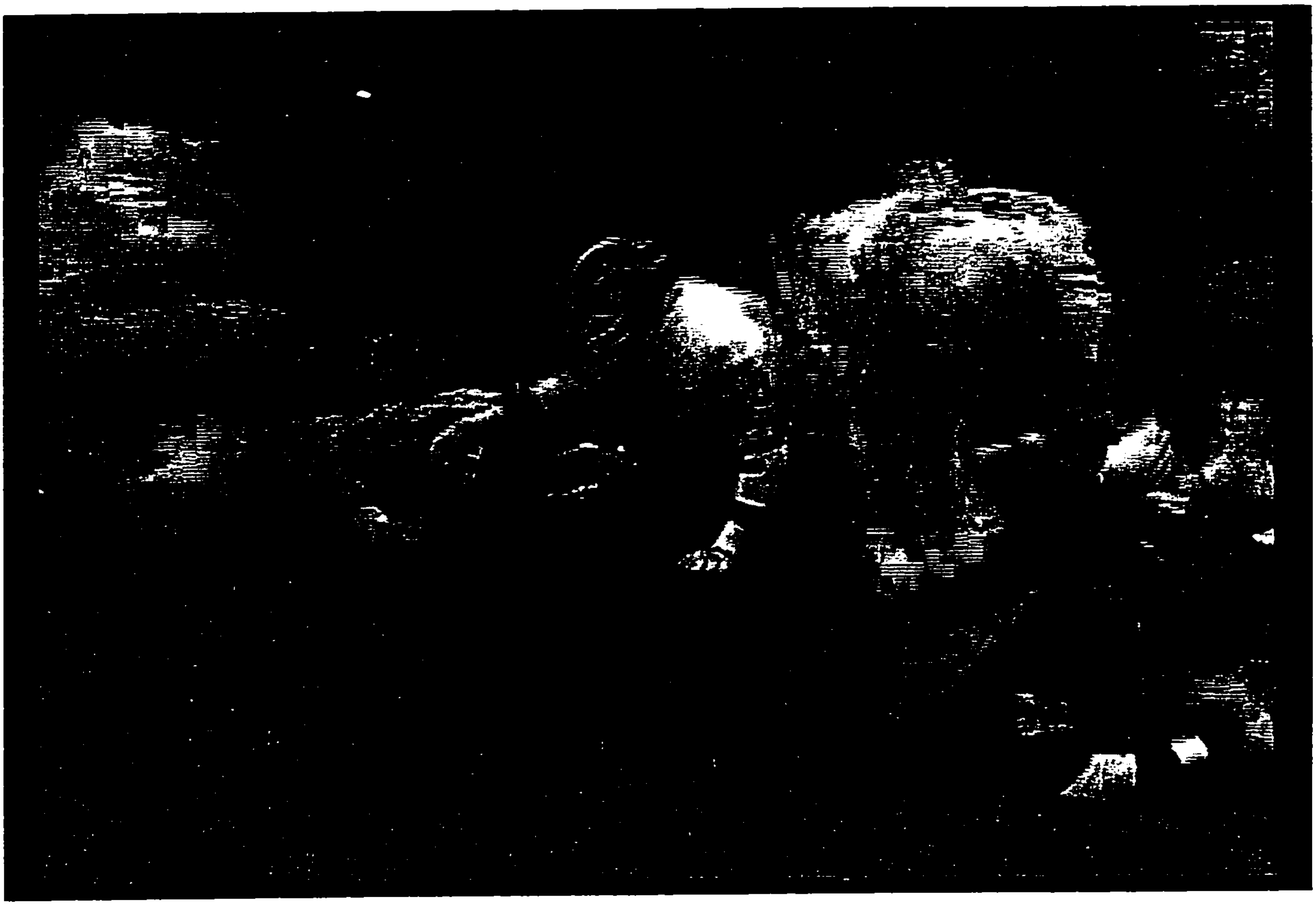


Figure 5.8 The motion compensated 10th frame of "Football" by New Method 1

made for regions having different complexity of motion. Extensive experiments have shown that for sequences with less motion such as videoconferencing sequences. the proposed algorithm reduces both the processing time and the number of final motion vectors drastically. while maintaining almost the same quality of the reconstructed image. For videoconferencing sequences it works even better than thresholding multiresolution block matching discussed in Chapter 4. For sequences containing complicated motion. such as sequences "Train" and "Football". the thresholding hierarchical block matching may not be suitable.

Our proposed algorithm is quite different from the existing hierarchical block matching algorithm [7]. Based upon the observation that large measurement windows give more reliable motion estimation in the case of large displacement while small measurement windows are more suitable for small displacement. in the existing hierarchical method. the motion estimation of each block in the current frame is performed recursively at three hierarchical levels. At each level. a separate motion estimation is performed. The finally estimated motion vector is the sum of the estimates from three hierarchical levels. On the other hand. the purpose of our algorithm is to reduce the computation of motion estimation. The strategy of our algorithm is also different. In the proposed algorithm. blocks are treated differently according to the accuracy of the estimated motion vectors. If the estimate of motion vector does not satisfying the predetermined threshold. the block will be splitted into four subblocks for further processing: otherwise. the motion estimation for this block will be terminated. thus saving computation. Besides. contrary to the algorithm in [7]. in our algorithm the search range of each level is increased from the top to the bottom level of a hierarchy.

CHAPTER 6

SUMMARY OF THE DISSERTATION

This chapter contains a summary of our major contributions and possible avenues for future research.

6.1 Summary

This dissertation has focused on motion estimation for applications in the field of video coding and computer vision. Here. we briefly summarize what have been accomplished in the research presented in this dissertation.

In the dissertation. we first present a multiattribute feedback approach to determining 2-D dense motion field. This approach has the following features. First. for each point in an image. multiple image attributes are computed as conserved information and make image point matching more robust. Specifically. we use two types of image attributes. One describes the structure information of the point under consideration: the other reflects the textural information of its local neighborhood. These attributes need less derivative operation and are hence less sensitive to various noises. Second. feedback technique is utilized to enhance the estimation accuracy. For each point under consideration. the estimated motion vector from the last iteration and its perturbed values lead to possible matching candidates in the second image. Third. except horizontal and vertical edgeness. no other differentiation is involved in the proposed computational framework. The estimation is carried out in two steps. In the conservation step. matching error is calculated by the sum of squared difference between the point under consideration in one image and its possible matching candidate in the other image. Estimation of motion is determined by using the weighted least squared estimation. In the propagation step. the estimates are computed as a weighted sum of those over a small neighborhood.

The proposed approach is mainly motivated from two newly developed motion estimation techniques. i.e.. Weng et al.'s and Pan et al.'s methods. This approach combines the merits and avoids the disadvantages of both existing algorithms. Compared with Weng et al.'s algorithm. our new method has the following distinctions. First. the multiple image attributes used are different and less sensitive to noises in images. Second. the computational framework needs much less differential calculation and it is therefore more robust to various noises. Third. in our new method. feedback technique is utilized to enhance the estimation accuracy. Our new approach is also quite different from Pan et al.'s algorithm. First. instead of using image intensity as a single attribute. multiple image attributes are computed as conserved information. Second. instead of window-wise. our approach is point oriented. Third. we consider the motion boundary as applying motion smoothness constraint. Experimental results show that our proposed approach outperforms in general most of the existing techniques computing 2-D dense motion field in terms of accuracy.

Block-based motion estimation has been successful in image sequence coding application. Block matching may be realized using full search technique or faster. more efficient search techniques. The well-known full search block matching is very time-consuming as a result of exhaustive searching where all the possible motion vectors are considered. Multiresolution block matching algorithms reduce the computation by taking advantage of pyramid structure. However. both approaches treat every region in an image domain indiscriminately no matter whether the region under consideration contains complicated motion or not. Motivated from this observation. we have developed two thresholding block matching algorithms.

One is the thresholding multiresolution block matching. In this method. we form multiresolution pyramid first. The motion estimation is initiated at the top pyramid level. and going down towards the bottom level. At each level of the pyramid. a full search block matching is conducted to search for the best matching in a predefined search range. If the accuracy is satisfied. the motion estimation for this block will be stopped. Otherwise. the computed motion vectors will be propagated into the next higher resolution level for refinement. Experiments with different motion complexities have shown that the proposed algorithm has a consistent performance. It reduces the processing time ranging from 14% to 20% while maintaining almost the same quality of the reconstructed image (only about 0.1 dB loss in PSNR). compared with the fastest existing multiresolution block matching.

The other is the thresholding hierarchical block matching. In the algorithm. a block at one level in the hierarchy corresponds to four blocks at the next level and all levels are of the same resolution. Motion estimation is performed from the top to the bottom level of the hierarchy. At each level. a separate search procedure with different sets of parameters is carried put. By thresholding. motion estimation for blocks with less motion will be terminated at some intermediate levels. thus saving computation. Extensive experiments indicate that for sequences with less motion such as video conferencing sequences. the proposed algorithm gives a better performance than the thresholding multiresolution block matching. It reduces both the processing time and the number of final motion vectors drastically. while maintaining almost the same quality of the reconstructed image.

The threshold has a major influence on the performance of the proposed thresholding schemes. The threshold used here is MAD and is determined by using two or three frames in a sequence before motion estimation. Once determined. the threshold can be used for the whole sequence of images.

In both above thresholding frameworks. block matching is initiated at the top level and going down towards the bottom level. and to save the computation. thresholding techniques are applied. However. there exist distinct difference between two algorithms. In the multiresolution approach. multiresolution pyramid is formed before matching. From the top to the bottom level of the pyramid. block size is increased and search range is decreased. Experiments show that this approach gives a consistent performance for image sequences with different motion complexities. On the other hand. in the hierarchical approach. each level in the hierarchy has the same resolution and no pyramid is formed. From the top to the bottom level. the block size is decreased and search range is increased. This approach needs less processing time and less number of motion vectors for image sequences containing less motion. But for video sequences having complex motion. its performance is degraded. As a conclusion. the hierarchical algorithm is more suitable for video conferencing applications. while the multiresolution algorithm can be used for sequences with more complex motion.

6.2 Future Research

In connection with the motion estimation approaches proposed in this thesis. there remain several questions which are worth investigating. Here we present some facts which can be considered to be the subject of future research.

-One of the possible research directions is to incorporate an effective confidence measure into the proposed multiconstraint feedback approach. In a computed optical flow field. not all flow vectors have high accuracy. Those vectors having high accuracy are more reliable than the others. It is important to search for a best confidence measure to determine the reliability and accuracy of the estimated flow vectors. For computer vision application. we can enhance the accuracy of 3-D motion analysis by extracting those flow vectors with high confidence measure. For video coding application. in regions where the computed motion vectors are less reliable. other coding mode than motion compensation can be applied.

-We can address the problem of choice of motion-invariant image attributes. To a large extent. both the accuracy and convergence speed of image point matching in the multiconstraint feedback approach depend on the image attributes used in the computation. The more effective the image attributes. the more accurate and faster the image point matching. In the current computational framework. we use two sets of motion invariant image attributes: structural attributes and textural attributes. The problem remains open to search for more effective motion invariant image attributes to enhance the performance of the proposed approach.

-In both thresholding multiresolution block matching and thresholding hierarchical block matching approaches presented in Chapters 4 and 5. respectively. thresholding technique is utilized to reduce the computation of block matching. Currently. the threshold is computed based on the first two or three frames from a sequence. Once determined. the threshold is used for the rest of the sequence. Frames in a video sequence can have different image details and contain different amount of motion. A larger threshold is appropriate for a frame having less detailed and small amount of motion while a smaller one is needed for an image with more details and complex motion. Hence. it can be one of the important subjects of future research to study a mechanism to automatically determine the threshold on a frame by frame basis.

-In the thresholding hierarchical block matching algorithm presented in Chapter 5. the size of blocks are decreased from the top to the bottom level in a hierarchy. Hence. blocks stopped at upper levels are of sizes larger than those stopped at lower levels. Consequently. the final motion field consists of blocks of different sizes. That is. motion vectors sent to a decoder are with blocks of different sizes. To reconstruct images. the variable block size information should be transmitted to the decoder . too. Therefore. it is worth investigating an effective way to transmit the variable block size information.

REFERENCES

1. E.H. Adelson and J.R. Bergen. "The extraction of spatialtemporal energy in human and machine vision." *Proc. IEEE Workshop on Visual Motion.* Charleston. 1986. pp. 151 - 156.

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

3. P. Anandan."A computational framework and an algorithm for the measurement of visual motion."*Int.J.Comp.Vision 2.* pp.283-310. 1989.

4. G. Adiv. "Determining three-dimensional motion and structure from optical flow generated by several motion objects." *IEEE Transactions on Pattern Analysis and Machine Intelligence.* Vol. 7. No. 4. July 1985. pp. 384 - 401.

5. S.T. Barnard and W.B. Thompson. "Disparity analysis of images." *IEEE Transactions on Computer.* Vol. 27. No. 4. 1978. pp. 359 - 366.

6. J. L. Barron. D. J. Fleet and S. S. Beauchemin. "Performance of optical flow techniques." *Technical Report.* No.299. Department of Computer Science. University of Western Ontario. London. Ontario. Canada.

7. M. Bierling and R. Thoma."Motion compensation field interpolation using a hierarchically structured displacement estimator."*Signal Processing.* Vol.11. No. 4. December 1986. pp.387-404. "Displacement estimation by hierarchical block matching." *SPIE Visual Communications and Image Processing.* Vol. 1001. 1988. pp. 942-951.

8. P.J. Burt. "The pyramid as a structure for efficient computation." in *Multi Resolution Image Processing and Analysis.* Springer Verlag. Berlin. Germany. 1984. pp. 6 - 37.

9. P.J. Burt. C. Yen. and X. Xu. "Multi-resolution flow-through motion estimation." *IEEE CVPR Conference Proceedings.* 1983. pp. 246 - 252.

10. B.F. Buxton and H. Buxton. "Computation of optical flow from the motion of edge features in image sequences." *Image and Vision Computing.* Vol. 2. 1984. p. 59 - p. 75.

11. Yui-Lam Chan and Wan-Chi Siu. "New adaptive pixel decimation for block motion vector estimation." *IEEE Transactions on Circuits and Systems for Video Technology.* Vol. 6. No. 1 February 1996. p. 113 - p. 122.

12. Woo Young Choi and Rae-Hong Park. "Motion vector coding with conditional transmission." *Signal Processing.* Vol. 8. No. 3. November 1989. p. 259 - p. 267.

13. Keith Hung-Kei Chow and Ming L. Liou. "Genetic motion search algorithm for video compression." *IEEE Transactions on Circuits and Systems for Video Technology*. Vol. 3. No. 6. December 1993. p. 440 - p. 445.

14. J.N. Driessen."Motion estimation for digital video."*Ph.D. Dissertation*. Department of Electrical Engineering. Delft University of Technology. Netherlands. September. 1992.

15. F. Dufaux and M. Kunt. "Multigrid based motion estimation for interframe image sequence coding." *Proc. EUSIPCO-92*. Brussels. Belgium. August 1992. pp. 1323-1326.

16. F. Glazer. *Hierarchical Motion Detection*. Ph.D. Dissertation. Department of Computer Science. Carnegie-Mellon University. Pittsburgh. PA. 1989.

bibitemhann Gerard Hann. Paul W.A.C. Biezen. Henk Huijgen. and Olukayole A. Ojo. "True-motion estimation with 3-D recursive search block matching." *IEEE Transactions on Circuits and Systems for Video Technology*. Vol. 3. No. 5. October 1993. p. 368 - p. 379.

17. R.M. Haralick. "Statistical and structural approaches to texture." *Proc. of the IEEE*. Vol. 67. No. 5. pp. 786-804. May 1979.

18. D.J. Heeger."Optical flow using spatialtemporal filters." *Int.J.Comp.Vision 1*. pp.279-302. 1988.

19. E.C. Hildreth. *The Measurement of Visual Motion*. Ph.D. Dissertation. Department of Electrical Engineering and Computer Science. MIT. Cambridge. MA. 1983.

20. B.K.P. Horn and B.G. Schunck."Determining optical flow."*Artificial Intelligence*. 17(1981). pp. 185-203.

21. J.R. Jain and A.K. Jain."Displacement measurement and its application in interframe image coding."*IEEE Trans. Commun.*. Vol.Com-29. No. 12. December 1981. pp. 1799-1806.

22. J.K. Kearney. W.B. Thompson and D.L. Boley. "Optical flow estimation: an error analysis of gradient-based methods with local optimization." *IEEE Transactions on Pattern Analysis and Machine Intelligence*. PAMI-9. 1987. pp. 229 - 244.

23. Gertjan Keesman et al.. "Study of the subjective performance of a range of MPEG-2 encoders." *Proceedings of ICIP-95*. Hyatt Regency Crystal City. Washington. D.C.. Oct. 23-26. 1995.

24. T.Koga. K. Linoma. A. Hirano. Y. Iijima and T. Ishiguro. "Motion compensated interframe coding for video conferencing."Proceedings of NTC'81. New Orleans. LA. December 1981. pp. G5.3.1-G5.3.5.

25. T. Koga and M. Ohta. "Entropy coding for a hybrid scheme with motion compensation in subprimary rate video transmission." *IEEE Journal on Selected Areas in Communications*. Vol. SAC-5. No. 7. August 1987. p. 1166 - p. 1173.

26. H. Li. A. Lundmark. and R. Forchheimer."Image sequence coding at very low bitrates: a review."*IEEE Trans. Image Processing*. Vol.3. No.5. September 1994. pp. 589-609.

27. Renxiang Li, Bing Zeng. and Ming L. Liou. "A new three-step search algorithm for block motion estimation." *IEEE Transactions on Circuits and Systems for Video Technology*. Vol. 4. No. 4. August 1994. p. 438 - p. 442.

28. Jae S. Lim. *Two-dimensional Signal and Image Processing*. Prentice-Hall. Englewood Cliffs. New Jersey. 1990.

29. Bede Liu and Andre Zaccarin. "New fast algorithms for the estimation of block motion vectors." *IEEE Transactions on Circuits and Systems for Video Technology*. Vol. 3. No. 2. April 1993. p. 148 - p. 157.

30. B.D. Lucas and T. Kanade. "An iterative image registration technique with an application to stereo vision." *Proc. DARPA IU Workshop*. pp. 121-130. 1981.

31. B.D. Lucas. *Generalized Image Matching by the Method of Difference*. Ph.D. Dissertation. Department of Computer Science. Carnegie-Mellon University. Pittsburgh. PA. 1984.

32. D. Marr and T. Poggio. "A computational theory of human stereo vision." *Proc. Royal Society of London*. Ser. B204. 1979. pp. 301 - 308.

33. L. Matthies. R. Szeliski and T. Kanade. "Kalman filter-based algorithms for estimating depth from image sequences." *International Journal of Computer Vision*. Vol. 3. 1989. pp. 209 - 236.

34. H.G. Musmann and P. Pirsch. "Advances in picture coding."*Proc. of the IEEE*. Vol.73. No.4. April 1985. pp.523-548.

35. H.H. Nagel."On a constraint equation for the estimation of displacement rates in image sequences."*IEEE Trans. PAMI 11*. pp. 13-30.

36. Kwon Moon Nam. Joon-Seek Kim. Rae-Hong Park. Young Serk Shim. "A fast hierarchical motion vector estimation algorithm using mean pyramid." *IEEE Transactions on Circuits and Systems for Video Technology*. Vol. 5. No. 4. August 1995. p. 344 - p. 351.

37. A.N. Netravali and J.D. Robbins. "Motion compensated Television coding: Part I." The B.S.T.J.. Vol.58.No.3. March 1979. pp.631-670.

38. Michael T. Orchard and Gary J. Sullivan. "Overlapped block motion compensation: an estimation-theoretic approach." *IEEE Transactions on Image Processing.* Vol. 3. No. 5. September 1994. p. 693 - p. 699.

39. Michael T. Orchard. "Predictive motion-field segmentation for image sequence coding." *IEEE Transactions on Circuits and Systems for Video Technology.* Vol. 3. No. 1. February 1993. p. 54 - p. 70.

40. J. N. Pan. Y. Q. Shi and C. Q. Shu. "Correlation-feedback approach to computation of optical flow." *Proceedings of IEEE 1994 International Symposium on Circuits and Systems.* vol. 3. pp. 33-36. May 1994. London. "Feedback Technique in Optical Flow Determination." *IEEE Transactions on Image Processing.* (Submitted).

41. J. N. Pan. "Motion estimation using optical flow field." *Ph.D. Dissertation.* Electrical and Computer Engineering. New Jersey Institute of Technology. Newark. NJ. April. 1994.

42. Ralf Schafer and Thomas Sikora. "Digital video coding standards and their role in video communications." *Proc. of the IEEE.* Vol. 83. No. 6. June 1995. pp. 907-924.

43. H. Schiller and B.B. Chaudhuri. "Efficient coding of side information in a low bitrate hybrid image coder." *Signal Processing.* Vol. 19. No. 1. January 1990. p. 61 - p. 73.

44. C.Q. Shu and Y.Q. Shi. "Computation of motion from stereo image sequence using the unified optical flow field." *SPIE's 1990 International Symposium on Optical and Optoelectronic Applied Science and Engineering.* San Diego. CA. July 1990.

45. C.Q. Shu and Y.Q. Shi. "On unified optical flow field." *Pattern Recognition.* Vol. 24. No. 6. June. 1991. pp. 579-586.

46. A. Singh. "Optical flow computation: a unified perspective." *IEEE Computer Society Press.* 1991.

47. A. Singh. "Incremental estimation of image-flow using a Kalman filter." *Proceedings of IEEE Workshop on Visual Motion.* Princeton. October 7-9 1991. pp. 36 - 43.

48. W.B. Thompson. "Introduction to the special issue on visual motion." *IEEE Trans. PAMI.* Vol.11. No.5. pp. 449-450. May 1989.

49. Dimitrios Tzovaras. M.G. Strintzis. H. Sahinolou. "Evaluation of multiresolution block matching techniques for motion and disparity estimation."*Signal Processing: Image communication.* 6(1994) . pp.56-67.

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

51. A.M. Waxman. J.Wu. and F. Bergholm."Convected activation profiles and receptive fields for real time measurement of short range visual motion." *Proc. IEEE CVPR*. Ann Arbor. pp.717-723. 1988.

52. J. Weng. N. Ahuja and T. S. Huang. "matching two perspective views." *IEEE Transactions on PMAI*. Vol. 14. no. 8. pp. 806-825. August 1992.

53. J. Weng. T.S. Huang and N. Ahuja. "Motion and structure from two perspective views: algorithm. error analysis and error estimation." *IEEE Transactions on Pattern Analysis and Machine Intelligence*. PAMI-11. 1989. pp. 451 - 476.

54. Yiwan Wong. "An efficient heuristic-based motion estimation algorithm." *Proc. of ICIP-95*. Hyatt Regency Crystal City. Washington. D.C.. October 23-26. 1995.

55. X. Xia and Y.Q. Shi. " A multiple attributes algorithm to compute optical flow." *Proceedings of 29th Annual Conference on Information Sciences and Systems*. John Hopkins University. Baltimore. MD. March. 1995. p. 480.

56. X. Xia and Y.Q. Shi. " A new multiresolution block matching algorithm for motion estimation in video coding." *Proceedings of 29th Annual Conference on Information Sciences and Systems*. John Hopkins University. Baltimore. MD. March. 1995. p. 599.

57. X. Xia and Y.Q. Shi. "A thresholding hierarchical block matching for motion estimation." *1996 IEEE International Symposium on Circuits and Systems*. Atlanta. GA. May 1996 (Accepted).

58. X. Xia and Y.Q. Shi. "Multiresolutional block matching in video compression by thresholding." *Proceedings of IEEE Ninth IMDSP Workshop*. Belize City. Belize. March 3-6. 1996. pp. 168 - 169.

59. Kan Xie. Luc Van Eycken and Andre Ooslerlinck. "A new block-based motion estimation algorithm." *Signal Processing: Image Coomunication*. Vol. 4. No. 6. November 1992. p. 507 - p. 517.

60. Y.-Q. Zhang. W. Li. and M.L. Liou. Special Issue on Advances in Image and Video Compression. *Proc. of the IEEE*. February. 1995.