MOTION ESTIMATION USING COMPLEX DISCRETE WAVELET TRANSFORM

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
THE MIDDLE EAST TECHNICAL UNIVERSITY

BY

HÜSEYİN SARI

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE
IN
THE DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING

SEPTEMBER 2003

Approval of the Graduate School of Natural and Applied Sciences

_____

Prof. Dr. Canan Özgen
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

_____

Prof. Dr. Mübeccel Demirekler
Head of Department

This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.

_____

Prof. Dr. Mete Severcan
Supervisor

Examining Committee Members

Assoc. Prof. Dr. Gözde Bozdağı Akar _____

Prof. Dr. Mete Severcan _____

Assoc. Prof. Dr. Aydın Alatan _____

Prof. Dr. Gönül Turhan Sayan _____

Dr. Uğur Murat Leloğlu _____

ABSTRACT

MOTION ESTIMATION USING COMPLEX DISCRETE WAVELET TRANSFORM

Sarı, Hüseyin

M.Sc., Department of Electrical and Electronics Engineering

Supervisor: Prof. Dr. Mete Severcan

September 2003, 82 Pages

The estimation of optical flow has become a vital research field in image sequence analysis especially in past two decades, which found applications in many fields such as stereo optics, video compression, robotics and computer vision. In this thesis, the complex wavelet based algorithm for the estimation of optical flow developed by Magarey and Kingsbury is implemented and investigated.

The algorithm is based on a complex version of the discrete wavelet transform (CDWT), which analyzes an image through blocks of filtering with a set of Gabor-like kernels with different scales and orientations. The output is a hierarchy of scaled and subsampled orientation-tuned subimages. The motion estimation algorithm is based on the relationship between translations in image domain and phase shifts in CDWT domain, which is satisfied by the shiftability and interpolability property of CDWT. Optical flow is estimated by using this relationship at each scale, in a coarse-to-fine (hierarchical) manner, where information from finer scales is used to refine the estimates from coarser scales.

The performance of the motion estimation algorithm is investigated with various image sequences as input and the effects of the options in the algorithm like curvature-correction, interpolation kernel between levels and some parameter values like confidence threshold

maximum number of CDWT levels and minimum finest level of detail are also experimented and discussed.

The test results show that the method is superior to other well-known algorithms in estimation accuracy, especially under high illuminance variations and additive noise.

Key words: Stereo vision, phase based motion estimation, complex discrete wavelet transform, gabor filters, subband tree decomposition.

ÖZ

KARMAŞIK AYRIK DALGACIK DÖNÜŞÜMÜ KULLANARAK HAREKET KESTİRİMİ

Sarı, Hüseyin

Yüksek Lisans, Elektrik ve Elektronik Mühendisliği Bölümü

Tez Yöneticisi: Prof. Dr. Mete Severcan

Eylül 2003, 82 Sayfa

Optik akış kestirimi, özellikle son yirmi yılda, sıralı görüntü analizi konusunda çok önemli bir araştırma alanı haline gelmiş ve stereo optik, video sıkıştırması, robotlar ve bilgisayarla görme gibi konularda uygulama alanları bulmuştur. Bu tezde, Magarey ve Kingsbury'nin optik akış tahmini için geliştirdiği karmaşık ayrık dalgacık dönüşümü tabanlı algoritma araştırılmış ve uygulanmıştır.

Algoritma, resmi farklı ölçek ve yönelimlerdeki Gabor benzeri süzgeç kullanılan bloklardan geçirerek analiz eden, ayrık dalgacık dönüşümünün kompleks bir versiyonuna dayanır. Çıktı, ölçeklenmiş ve altörneklenmiş yönelim-akortlu altresimlerin sıradüzeninden oluşmuştur. Hareket kestirim algoritması, karmaşık ayrık dalgacık dönüşümünün aradeğerlendirilebilirlik ve kaydırılabilirlik özelliği tarafından sağlanan, resim alanındaki değişimler ve dönüşüm alanındaki faz kaymaları arasındaki bağıntıya dayanır. Optik akış her ölçekte bu bağıntıyı kullanarak, düşük çözünürlükten yüksek çözünürlüğe giden ve yüksek çözünürlükteki verilerin düşük çözünürlükteki verileri rafine etmek için kullanıldığı bir mantıkla tahmin edilir.

Hareket kestirim algoritmasının performansı farklı resimler girdi olarak kullanılarak araştırılmış ve eğrilik düzeltmesi, düzeyler arası aradeğerleme tipi gibi modifikasyonların ve güvenilirlik eşik değeri, maksimum seviye ve en detaylı çözünürlük gibi parametrelerin etkisi analiz edilmiştir.

Test sonuçları gösteriyor ki, metod iyi bilinen diğer algoritmalara oranla özellikle yüksek aydınlatma değişimleri ve gürültü eklenmesi koşulları altında çok daha iyi kestirim doğruluğu vermektedir.

Anahtar kelimeler: Çift kanallı görme, faza dayalı hareket kestirimi, karmaşık ayrık dalgacık dönüşümü, gabor süzgeçleri, altband ağaç ayrışımı.

ACKNOWLEDGEMENTS

I would like to thank my supervisor Mete Severcan for his unflagging support and expert guidance throughout the development and improvement of this thesis.

I, of course, would like to thank my parents for being and supporting me in educational life.

I am also grateful to Mahir Özdemir for his insightful advice, friendly support and providing support on Matlab.

I would also like to express my deep gratitude to all those who have supported and encouraged me in completing this thesis.

TABLE OF CONTENTS

LIST OF TABLES

TABLE

LIST OF FIGURES

FIGURE

LIST OF ABBREVIATIONS

CDWT: Complex Discrete Wavelet Transform

CSD: Cumulative Squared Difference

DFD: Displaced Frame Difference

DRD: Displaced Region Difference

DWT: Discrete Wavelet Transform

FIR: Finite Impulse Response

FT: Fourier Transform

MAD: Maximum Absolute Difference

ME: Motion Estimation

MSE: Mean Square Error

OFC: Optical Flow Constraint

QMF: Quadrature Mirror Filter

RI: Rotation Invariant

SSD: Subband Squared Difference

STFT: Short Time Fourier Transform

WT: Wavelet Transform

CHAPTER 1

INTRODUCTION

Humans and animals naturally develop the ability to discern objects, ascertain their motion, and navigate in three-dimensional space to interact with their surroundings. All of us are equipped with a pair of eyes and perform visual functions such as depth perception, object recognizing, tracking etc., through the use of this stereovision mechanism. Although this effective visual perception mechanism has not yet been fully explained or matched by any artificial vision system, the idea of incorporating such vision in machines gave rise to investigations and implementations in neurophysiological and computer vision areas especially in the past two decades.

Numerous studies in both areas have shown that motion is a basic data for visual functions. In fact, the analysis of a sequence of images provides information about the three-dimensional shape and structure of objects, the relative depth between different objects or their trajectories and direction of motion, which could not be obtained from a single image. Therefore, research work on motion estimation has already been carried out to various applications in which a compact representation of the changes between the successive frames in a digital image sequence is required. One application is the obstacle detection and navigation in robotics. Another application is the egomotion recovery, i.e. estimation of the observer motion. Also much work has also been generated in the field of video coding, since compression of image sequences using motion estimation reduces the bandwidth requirement for an efficient transmission. It also finds place in medicine and meteorology by processing related images. Another area of high importance is the detection and tracking of moving targets in military applications.

In the past decade, wavelets has become an important analysis technique for motion estimation. The success of wavelet analysis when applied to motion estimation problem has shown its suitability as an approach to the stereovision problem. The work presented here is

concerned with motion estimation based on wavelet analysis developed by J. Magarey and N. Kingsbury [1].

1.1 Stereo Vision

Stereo vision is related with the recovery of the three dimensional shape of a scene based on two images taken of that scene from slightly different viewpoints. Human vision system can be modeled as two cameras separated by a constant distance often referred to as a stereo head. (Figure 1.1)

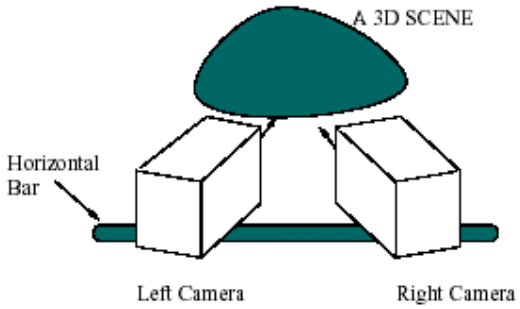


Figure 1.1 Model of human visual system

For the process of obtaining stereo image pairs, the common method is to use two cameras displaced from each other by a known distance. Another method is to use a moving camera. When the sensor itself is moving, motion detection alone is not enough. The sensor motion generates an apparent motion of the background, which needs to be compensated or segmented from the target motion, in order to allow detection of motion. In each case, the only detectable motion in an image sequence follows from the spatio-temporal changes of the gray level intensity function of the sequence, known as the optical flow. As expected, second image is not the purely shifted version of the first image. This is because the changes can be caused by the relative motion between the changing three dimensional scene and imaging device, but can also be produced by changes in the image formation process itself, such as illumination changes or noise in the electronics. Here, the work will be on stereo image pairs, which are taken from stationary cameras i.e. identifying true motion vectors. If desired, then this true motion vectors can be used to reconstruct depth information [10, 23] since the distance between two image sources is known.

1.2 The Motion Estimation Problem

The motion estimation problem deals with the analysis of motion in two-dimensional digital images, which are the snapshots of a three-dimensional moving scene taken at successive instants of time as shown in Figure 1.2.

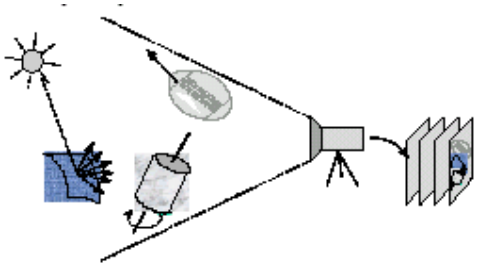


Figure 1.2 Model of obtaining 2-D projections of a 3D-scene

In the ideal case, any two-dimensional point in an image may correspond to another two-dimensional point in the following image. This can be formulated as,

$$u_{n-1}\left(\vec{x}+\hat{d}\vec{x}\right)=u_n\left(\vec{x}\right) \tag{1.1}$$

where $\vec{x}$ is the position vector of the pixel in the current image, $\hat{d}\vec{x}$ represents the displacement vector of the pixel with $\vec{x}+\hat{d}\vec{x}$ representing the location of the pixel in the reference image and $\{u_n, n\varepsilon Z\}$ is the scalar intensity function of the image number n. But in real life, there are a lot of factors effecting this one-to-one correspondence, i.e. addition of some means of noise to current image. So the image sequence is observed to be noise corrupted which can be formulated as

$$I_n\left(\vec{x}\right)=u_n\left(\vec{x}\right)+e_n\left(\vec{x}\right) \tag{1.2}$$

The frames are captured by a camera at a given frame rate and each frame can be viewed as a function of the light intensity information taken by the camera at different time instants. Since the captured sequence represents a projection of the three-dimensional scene to two dimensions, any image is not a purely shifted version of the previous image and the relationship between the observable changes in image intensities and the motion of three-dimensional objects in the scene is complicated. This complicated relationship is caused by the loss of information in the process of projection from three to two dimensions. This is

known as the projection ambiguity or the correspondence problem, which means that several points in the three-dimensional scene with potentially different motions may project to the same two-dimensional point in the image.

There are also other reasons for this complexity, which adds some kinds of noise to the motion estimation process. These are the effects like occlusion, illumination changes, aberrations and distortions in the image acquiring optics, etc. Another interesting factor is whether moving objects are rigid or change shape during the motion, for example, a smiling human face.

Because of the loss of information while projecting from three-dimensional space to two-dimensional space, the problem is ill-posed, i.e. a unique solution of the correspondence problem does not always exist.

Notice that a distinction is sometimes made between the true motion field, which is the 2D projection of the relative 3D motion between scene and camera, and the optical flow field, which describes the variation of the grayscale pattern over space and time. Ideally, they are identical if the image formation process is free of illumination changes, noise in electronics, object shape changes, etc.

1.3 Approaches To The Motion Estimation Problem

The motion estimation algorithms can, in general, be classified into two main categories. These are low-level and high-level computer vision algorithms.

High-level methods are based on extracting high-level features of the image, such as edges, corners, object boundaries or complete objects in order to solve the correspondence problem. The advantage of the feature-based approaches is that matching only on the basis of these features is relatively fast due to the small number of points extracted from each image. Also false matches are relatively less, because matching is only attempted for the points, which are easiest to match. However, the number of false matches for the feature based approaches increases with a scene containing wide featureless regions (smooth surfaces) or when features in an image are occluded in the other. Also, even there are no occlusions in the images, a highly featured area also increases the number of false matches.

Low-level approaches use low-level image descriptors (like intensity) of the image for motion estimation process. The low-level approaches are also referred as the optical flow estimation approaches, which can be categorized into gradient-based, block-matching and frequency domain techniques. This work, referred to as the CDWT motion estimation algorithm, will be on optical flow approaches and corresponds to the latter category.

1.4 The CDWT Motion Estimation Algorithm

The CDWT motion estimation algorithm is based on the complex discrete wavelet transform, developed by Magarey and Kingsbury [1, 2, 3, 7]. In order to compute the motion field, the algorithm uses the relation between the phase of the transform coefficients and translations in the image domain. The choice of the algorithm from a wide variety of approaches is because of three main factors.

- The algorithm uses a discrete wavelet transform implementation similar to a method developed by Mallat[11], which makes use of an efficient pyramidal algorithm based on convolutions with one-dimensional 4-tap filters.
- The basic filter pair used in implementation may be modeled as Gabor filters, which have a strong implication from a physiological/biological point of view. In fact, there are various investigations on modeling the mammalian vision system with a preprocessing stage in which Gabor filters constitute the main component.
- Motion estimation is based on the phase of the complex wavelet coefficients. The phase-based motion estimation is relatively robust and evolution of constant phase contours of the complex filter outputs gives a truer picture of the underlying spatiotemporal structure than that of constant intensity contours, which are used in gradient-based methods.

This work describes the investigation and application of the CDWT motion estimation algorithm. The algorithm is implemented in Matlab. Also, experiments are done and results are extracted in order to compare the performances of the modifications and parameter changes in the algorithm.

1.5 Thesis Outline

The organization of the thesis is as follows.

Chapter 2 presents a review of motion estimation techniques. It focuses on low-level motion estimation techniques and gives detailed information on them.

Chapter 3 describes the motion estimation algorithm developed by Magarey and Kingsbury [1], which is the basis of this work. The algorithm is based on a complex version of the discrete wavelet transform developed by Mallat [11]. This chapter is composed of two main parts. The first part describes the complex discrete wavelet transform, starting with a short summary of the wavelet transform in continuous and discrete domains, the

implementation using a quadrature mirror filter (QMF) bank, and finally the Gabor-like complex-valued FIR filters, which are the basis of the CDWT implementation. The second part is composed of the description of the motion estimation algorithm itself, which uses the CDWT coefficients at various resolutions in order to estimate motion in a coarse-to-fine manner. This is done by matching the phases of the CDWT coefficients, which are directly related to the displacements in the image domain.

Chapter 4 is composed of the simulation analysis and results, with synthetic tree sequences as inputs, of the implemented motion estimation algorithm. Also some options are introduced and their contributions to the performance of the proposed algorithm are analyzed.

CHAPTER 2

REVIEW OF MOTION ESTIMATION TECHNIQUES

As mentioned in the previous chapter, there are several methods that can be used to estimate optical flow. Since motion estimation is a research area for nearly two decades and much effort spent on the subject, giving a detailed review of motion estimation techniques would be hard and impractical. In this chapter, most popular methods will be introduced which show a closer relation with the work investigated here. These methods can be classified into three main categories, which are gradient-based, block-matching and frequency domain techniques. This work can be classified mostly in the last group. There are also other techniques such as pel-recursive and Bayesian algorithms. But, mostly three standard motion estimation techniques will be emphasized, since they show a closer relation to the work investigated here. However, a brief introduction about pel-recursive and Bayesian methods is given.

There are practical limitations in the above-mentioned methods, which can be overcome by use of multiresolution concept or hierarchical approach. This subject will be introduced in chapter 3.

2.1 Motion Estimation Techniques

Almost all motion estimation techniques are based on the assumption that any change in the image intensity is only due to the motion in the scene. This means that while an object is moving along a direction, the image intensity of any moving part of the object remains constant. Since image intensity is a function of time, intensity at any image point $\vec{x} = (x_1 \quad x_2)^T$ at time instant t, can be represented as $I(\vec{x}, t)$. Using constant intensity assumption, it can be said that any point in the image will appear with a displaced location in a later time.

$$I(\underset{\sim}{x} + d\underset{\sim}{x}, t + dt) = I(\underset{\sim}{x}, t) \tag{2.1}$$

If the left side of the equation is rewritten using a Taylor expansion,

$$I(\underset{\sim}{x} + d\underset{\sim}{x}, t + dt) = I(\underset{\sim}{x}, t) + \nabla I(\underset{\sim}{x}, t).d\underset{\sim}{x} + \frac{\partial I(\underset{\sim}{x}, t)}{\partial t} dt + \varepsilon T \tag{2.2}$$

where ∇ is the gradient operator with respect to the position parameter $\underset{\sim}{x}$, and εT is the error term representing higher order terms in Taylor series expansion. Neglecting higher order terms, substituting (2.2) in (2.1) and dividing both sides by dt results in a popular equation known as the optical flow constraint (OFC):

$$\nabla I(\underset{\sim}{x}, t).\underset{\sim}{v} + \frac{\partial I(\underset{\sim}{x}, t)}{\partial t} = 0 \tag{2.3}$$

where the vector $\underset{\sim}{v} = (v_1 \quad v_2)^T = \left(\frac{dx_1}{dt} \quad \frac{dx_2}{dt} \right)^T$ represents the velocity along the motion direction. The solution of the equation results in a straight line in (v_1, v_2) space as shown in Figure 2.1. Notice that presence of two variables for one equation makes optical flow estimation an ill-posed problem.

OFC also implies that, at any point, only the component of the motion parallel to the image spatial gradient, i.e. motion perpendicular to the image edges, is recoverable. This is known as the aperture problem, which means that the component of the motion along the direction of an edge cannot be determined unless the size of the aperture of the analyzing window is larger than the length of the edge. An illustration of the aperture problem is given in Figure 2.2.

From a discrete point of view, the images of the 3D scene taken at equally spaced time instants can be considered as a sequence of pictures. By this, the time variable of the intensity function of any point in the image can be dropped and a discrete version of equation (2.1) can be written as

$$I_{k+1}(\underset{\sim}{h} + d(\underset{\sim}{h})) = I_k(\underset{\sim}{h}) \tag{2.4}$$

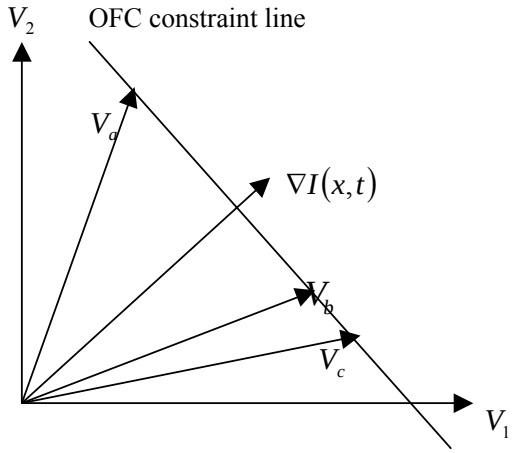


Figure 2.1. OFC constraint line with possible solutions v_a, v_b and v_c

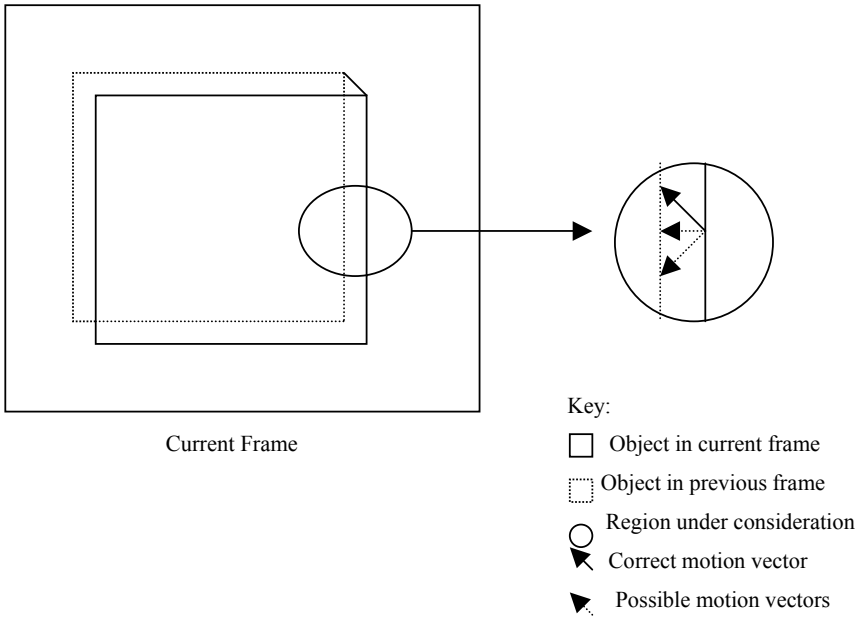


Figure 2.2 The aperture problem

where $I_k(\vec{h})$ represents the intensity value of a pixel at location $\vec{h} = (n_1 \quad n_2)^T$ in frame k, and $d(\vec{h})$ is the corresponding displacement vector. However, since this displacement is the quantity to search, the equation may be rewritten as an error function of the displacement vector i.e. a similarity measure can be written between pixels in adjacent frames as a function of estimated displacement vector,

$$\varepsilon_{DFD}(\vec{h}, d(\vec{h})) = I_{k+1}(\vec{h} + d(\vec{h})) - I_k(\vec{h}) \tag{2.5}$$

where ε_{DFD} is known as the displaced frame difference (DFD). When $d(\vec{h})$ is truly estimated, the DFD will be zero under constant intensity flow and translational motion conditions. Therefore, the displacement vector estimation can be done by minimizing ε_{DFD} with respect to $d(\vec{h})$. Notice that equation (2.4) should fail in the case of motion due to rotations, dilations or occlusions in the image. However, if these deformations are small, the corresponding motions may be approximated by translations, and equation (2.4) may become a reasonable assumption. In such a case, we may write $I_{k+1}(\vec{h}+d(\vec{h}))$ using a Taylor series expansion

$$I_{k+1}(\vec{h}+d(\vec{h})) = I_{k+1}(\vec{h}) + \nabla I_{k+1}(\vec{h}).d(\vec{h}) + \varepsilon \tag{2.6}$$

which can be substituted in equation (2.5) yielding

$$\varepsilon_{DFD}(\vec{h}, d(\vec{h})) = I_{k+1}(\vec{h}) - I_k(\vec{h}) + \nabla I_{k+1}(\vec{h}).d(\vec{h}) \tag{2.7}$$

ignoring higher order terms.

If we divide the above expression by the time interval Δt between frames and take the limit as $\Delta t \to 0$, we obtain again the OFC in time domain.

The motion estimation techniques that will be described next use one of these viewpoints, i.e. they either use the OFC directly by measuring the spatiotemporal rate of change of the image intensity or they minimize the DFD over a set of local regions by searching over a set of motion vectors.

2.1.1 Gradient Based Techniques

Gradient-based techniques provide an estimate of the optical flow field in terms of spatiotemporal intensity gradients. Usually the OFC is used in conjunction with an appropriate spatiotemporal smoothness constraint, which requires that the velocity vector varies slowly over a neighborhood.

Optical flow constraint was first used to estimate motion by Cafforio and Rocca [21]. They attempted to segment the image into a fixed background and a moving object region using dynamic programming. They assumed velocity to be constant within the region of movement and minimized the quantity

$$\varepsilon_{OFC} = \nabla I(\vec{x},t).\vec{v} + \frac{\partial I(\vec{x},t)}{\partial t} \tag{2.8}$$

Horn and Schunck [12] imposed a global smoothness constraint on the optical flow field from which a second quantity to be minimized comes. This quantity is the sum of squares of the spatial gradients of the velocity components

$$\varepsilon_{SC}^2 = \left\|\nabla v_1(\vec{x},t)\right\|^2 + \left\|\nabla v_2(\vec{x},t)\right\|^2 \tag{2.9}$$

Combining these two quantities result in a total error to be minimized,

$$\varepsilon_{global}^2(\vec{v}) = \int_{image} \left(\varepsilon_{OFC}^2 + \alpha^2 \varepsilon_{SC}^2\right) d\vec{x} \tag{2.10}$$

where α is the relative weighting factor between the two error terms and it indicates the importance of global smoothness constraint relative to the OFC. Equation (2.8) can be analytically solved. But solution of equation (2.10) requires a series of Gauss-Seidel iterations for convergence to the minimizing solution, $v(\vec{x})$, over the whole image.

Instead of a global smoothness constraint, Lucas and Kanade [13] proposed a local smoothness constraint, by assuming that the motion vector is the same for a particular image region $\Re$. This can be expressed as the minimization of the squared DFD with respect to the displacement vector $\vec{d}$,

$$\varepsilon_{local}^2(\vec{d}) = \sum_{x\varepsilon\Re} w(\vec{x})\varepsilon_{DFD}^2 \tag{2.11}$$

where $w(\vec{x})$ is a window function, giving more weight to the central part of the region. Using equation (2.7) and ignoring the higher order term, the minimization of (2.11) with respect to the displacement vector $\vec{d}$ can be expressed as

$$\sum_{x\varepsilon\Re} w(\vec{x})\left[I_{k+1}(\vec{x}) - I_k(\vec{x}) + \nabla I_{k+1}(\vec{x}).d(\vec{x})\right]\nabla I_{k+1}(\vec{x}) = 0 \tag{2.12}$$

which represents an overdetermined system of linear equations on the two displacement components, and its solution may be found using standard least squares techniques.

An alternative to the global and local smoothness constraints is to use a constraint on second order derivatives of the image intensity function for the velocity vector. The constraint is the conservation of the spatial intensity gradient $\nabla I(\underset{\rightharpoonup}{x},t)$, i.e.

$$\frac{d[\nabla I(\underset{\rightharpoonup}{x},t)]}{dt} = 0 \tag{2.13}$$

which gives

$$\begin{bmatrix} \frac{\partial^2 I}{\partial x_1^2} & \frac{\partial^2 I}{\partial x_2 \partial x_1} \\ \frac{\partial^2 I}{\partial x_1 \partial x_2} & \frac{\partial^2 I}{\partial x_2^2} \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} + \begin{bmatrix} \frac{\partial^2 I}{\partial t \partial x_1} \\ \frac{\partial^2 I}{\partial t \partial x_2} \end{bmatrix} = 0 \tag{2.14}$$

But for this method to be successful, there must not be first-order deformations of intensity in the image such as dilation or rotation, which is commonly present in real image sequences. The problem with this assumption is that it does not allow for first-order deformations of intensity, such as dilation or rotation, commonly present in image sequences. Because of this, second-order methods often produce less accurate estimates than other methods.

Both local and global gradient-based methods rely on the assumption of smoothness of the optical flow field. The local methods present difficulties in regions where the spatial gradients change slowly. The local information is insufficient and therefore constraint equations arising from different (neighboring) points provide essentially the same constraint on the optical flow. The global methods reduce these problems, since the information is propagated over the image. However, the errors are also propagated. In this case, the main problem is that the smoothness constraint does not hold across motion boundaries i.e. it blurs motion edges.

Gradient methods also suffer from the fact that the OFC is often violated. Since the Taylor expansion used in order to derive this constraint is truncated i.e. higher order terms are neglected, the OFC breaks down for large motion vectors. A subclass of gradient-based methods, known as pel-recursive (iterative gradient) algorithms, represents an attempt to overcome this problem. In this algorithm, an initial displacement field $\underset{\rightharpoonup}{d}^{(i)}$ is computed. This is then updated recursively using the Taylor expansion of $I(\underset{\rightharpoonup}{x} + \underset{\rightharpoonup}{d}^{(i)})$ around $\underset{\rightharpoonup}{d}^{(i)}$,

producing a refined estimate $\vec{d}^{(i+1)}$. The main drawback of these algorithms is that the iterative procedure may converge to a local minimum rather than to the global one.

Bayesian techniques [24], which may be considered as a generalization of gradient-based techniques, make use of probabilistic smoothness constraints in addition to the OFC in order to estimate the motion field. Since optical flow fields include image noise, lighting changes, low contrast regions, aperture problem, and multiple motions in a single localized region, a probabilistic framework would allow these uncertainties to be represented in the computations, and passed along to the next stage of computation. For this purpose, a conditional probability function of the velocity based on the image intensity gradient may be used.

Other techniques developed to deal with larger motions are hierarchical gradient-based methods. These are usually based on a Gaussian pyramid, which is a multiresolution structure composed of successively smoother and subsampled versions of the image. These image versions are obtained through iterative filtering of the image with a Gaussian kernel followed by subsampling. The motion estimation is performed in a coarse-to-fine way, i.e. estimation starts at the coarsest resolution, and these estimates are used as initial points to motion estimation at the next finer resolution.

2.1.2 Block-Matching Techniques

Block-matching techniques can be considered as the first and most popular methods for practical motion estimation because of their less hardware complexity requirement and implementation ease. In block-matching techniques, a regional direct search procedure on image is used instead of using a Taylor series expansion to estimate $\vec{d}$ in terms of the intensity derivatives. The starting point of block-matching techniques is the DFD extended over a region, named as displaced region difference (DRD),

$$DRD_m(\Re, \vec{d}) = \sum_{\vec{x} \in \Re} z\left(I_{m+1}\left(\vec{x} + \vec{d}\right) - I_m\left(\vec{x}\right)\right) \tag{2.15}$$

where z is a distortion function and m represents the number of picture in frame sequence. The most common distortion functions are mean square error (MSE) and maximum absolute difference (MAD). The aim of block-matching methods is to minimize the DRD and find the motion estimate $\vec{d}$ for which the DRD is minimum, i.e.

$$\hat{d}(\Re) = \arg\min_{\vec{d}} \left\{ DRD_m\left(\Re, \vec{d}\right) \right\} \tag{2.16}$$

Block-matching algorithms work by partitioning the image into blocks of the same size, and then assigning the same motion vector to all the pixels included by the block. For each block B, the motion vector is evaluated by matching the information content of the block in frame k with that of blocks of the same size within a search area S, placed in the following frame k+1 as shown in Figure 2.3.

In order to find an absolute minimum for the matching criterion (2.15), an exhaustive search of a series of discrete candidate displacements within a maximum displacement range must be performed. This technique is called full-search block matching. Despite the heavy computation it requires, it is widely used in video coding, due to its simplicity and ease of hardware implementation.

Instead of minimizing the squared DFD, a similar approach is to maximize the cross-correlation between the blocks in the two frames. From the definition of the DFD (2.5) and equation (2.16), the expression to be minimized for generally used distortion function MSE is

$$\sum_{n \varepsilon B} \left[I_{k+1}^2\left(\vec{h} + d(\vec{h})\right) + I_k^2(\vec{h}) - 2I_{k+1}\left(\vec{h} + d(\vec{h})\right) I_k(\vec{h}) \right] \tag{2.17}$$

From this expression, it can be seen that finding the displacement $\vec{d}$ that minimizes (2.17) is equivalent to finding $\vec{d}$ that maximizes

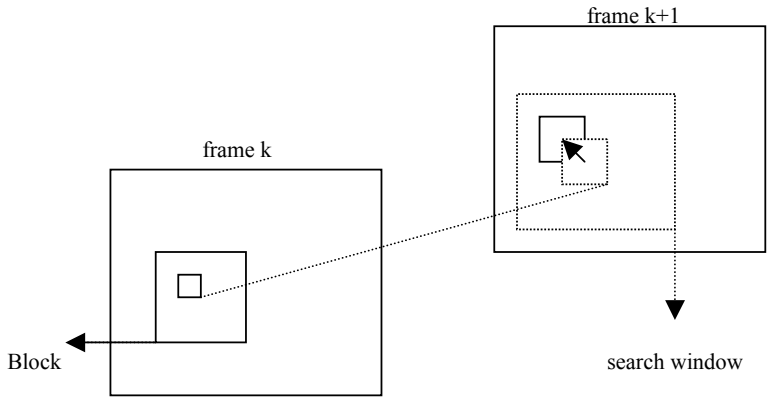


Figure 2.3 Block-matching search algorithm

$$C_B(\vec{d}) = \sum_{n \in B} [I_{k+1}(\vec{h} + d(\vec{h}))I_k(\vec{h})] \tag{2.18}$$

which is the expression for the cross-correlation defined over a block B of the intensity functions of frames k and k+1.

A problem with block-matching techniques is the high computational expense. For example, for a block containing N^2 pels, a full search over a region $\Re$ comprising r candidates requires $3N^2 r$ addition and multiplication operations. However, fast search algorithms have been proposed in order to decrease the computational expenses of the full-search algorithm. But they decreases the accuracy of the method. The most popular search algorithms are 3-step search, 2-D search, conjugate direction search and hierarchical search.

Another problem with block-matching techniques is that they produce integer estimates, since the search is performed over a discrete grid. In order to produce sub-pixel accuracy, the image intensity has to be interpolated at fractional pixel locations.

The block size is one of the most important parameters in any block-based motion estimation algorithm. Selection of the block size needs an optimization because of two facts. The window must be large enough in order to be able to estimate large displacements vectors (Otherwise, number of false matches with large displacement vectors increases). On the other hand, it should be small enough so that the displacement vector remains constant within the window. These two requirements can usually be addressed by hierarchical methods. For instance, Anandan proposed a block-matching coarse-to-fine algorithm based on a Laplacian pyramid [15], which is a multiresolution structure such as the Gaussian pyramid, but the pyramid images represent bandpass versions of the original one, rather than lowpass. Instead Dufaux and Moscheni [16] used an adaptive multigrid structure, in which the image is initially divided into a few large blocks, and a motion vector is estimated for each block. Then the corresponding block is equally divided into four children blocks, which inherit a motion estimate, which is a combination of the closest four parent blocks. This estimate is then refined using a block-matching technique at that grid, and the process is repeated until the finest grid.

2.1.3 Frequency Domain Techniques

Frequency domain techniques estimate motion by transforming the image sequence to the frequency domain, and then minimizing some disparity measure (or maximizing a correlation measure) based on the transform coefficients rather than image intensities [9]. Approaches have been developed where the relevant measure is a function of either the

magnitude or the phase of the transform coefficients, given rise respectively to energy-based and phase-based techniques.

The cross-correlation expression given by equation (2.18) can be computed more efficiently in the frequency domain by using the property of the Fourier transform that convolution in the space domain is equivalent to multiplication in the frequency domain.

$$C_B\left(\vec{d}\right) = F^{-1}\left\{\hat{I}_{B,k+1}\left(\vec{\Omega}\right) \quad \hat{I}_{B,k}\left(\vec{\Omega}\right)\right\} \tag{2.19}$$

where $\hat{I}_{B,k}\left(\vec{\Omega}\right)$ is the Fourier transform of $I_k\left(\vec{h}\right)$ over the block B in frame k, $\Omega = \begin{pmatrix} w_1 & w_2 \end{pmatrix}^T$ is the spatial-frequency vector, and F^{-1} indicates the inverse Fourier transform. Therefore the displacement vector $\vec{d}$ may be estimated by locating the peak of the cross-correlation function given by (2.19). This method is equivalent to performing a local cross-correlation, where the window function is a rectangular window.

Other approaches use phase correlation rather than the magnitude correlation to estimate motion. The idea of phase correlation-based motion estimation is originated from the shifting property of the Fourier transform.

$$\begin{aligned} f(t) &\Leftrightarrow F(w) \\ f(t+\tau) &\Leftrightarrow e^{iw\tau} F(w) \end{aligned} \tag{2.20}$$

If equation (2.4) is written in frequency domain using equation (2.20), we get

$$\hat{I}_{k+1}\left(\vec{\Omega}\right) e^{i\vec{\Omega}^T \vec{d}} = \hat{I}_k\left(\vec{\Omega}\right) \tag{2.21}$$

where the displacement can be obtained by dividing the Fourier phase difference by angular frequency ω. However, the Fourier transform describes behavior over all time and shifts which are localized to a particular time interval can not be estimated by this method. To estimate local shifts we need a description which is responsive to local changes in the signal. For this purpose, Short-Time Fourier Transform (STFT) is proposed, which windows the Fourier basis functions to provide a time-frequency description of the signal,

$$STFT_x(\tau, w) = \int_{\tau-T}^{\tau+T} x(t) w(t-\tau) e^{-jwt}\, dt \tag{2.22}$$

where $w(t)$ is a real-valued symmetric function defined over a finite interval $[-T,T]$ (i.e. a window). This transform can be considered as an analysis of a signal by filters of finite extent in time and frequency. The product of the time and frequency resolutions depends on the form of the window w. The Gabor transform, for example, uses a Gaussian window, which achieves the best possible combination of frequency and time resolution [18].

Fleet and Jepson [17] used the phase output of several Gabor filters, tuned to different spatiotemporal frequencies, in order to estimate the motion. They assumed that the phase of the filters' outputs is constant along motion trajectories, ending up with

$$\nabla\phi(\vec{x},t).\vec{v} + \frac{\partial\phi(\vec{x},t)}{\partial t} = 0 \tag{2.23}$$

i.e. an optical flow constraint on the phase ϕ of each filter output, rather than on intensity. They used this equation to estimate normal velocities (i.e. along the phase spatial gradient $\nabla\phi(\vec{x},t)$), and combined the estimates obtained from different filters over local regions to produce two-dimensional motion estimates. To support their use of the phase rather than of the amplitude of the Gabor filters, they made simulations to show that the evolution of constant phase contours was closer to the 2-D projection of the 3-D motion of the scene than the evolution of constant amplitude contours. Also, they showed that phase information is robust with respect to smooth illumination changes and small deformations caused by the perspective projection of moving three-dimensional objects, which cause deviation from the simple translation motion model used in most optical flow techniques.

The CDWT motion estimation algorithm investigated here is also related with the frequency domain class of motion techniques. As in Fleet and Jepson's method, motion is estimated from the phase output of Gabor-like filters. However, these filters are implemented efficiently through a scheme derived from the discrete wavelet transform of Mallat [11]. The method is described with more details in the next chapter.

2.2 Summary

This chapter has reviewed the most widely used strategies to the problem of motion estimation, in particular the ones belonging to the classes of gradient-based, block-matching and frequency domain techniques.

Gradient-based methods produce real-valued motion fields but have limited measurement range and are susceptible to noise because of their use of spatiotemporal derivatives. Block-based matching methods have arbitrary range and good immunity to noise but are limited in precision and are computationally expensive. Frequency domain methods have (in principle) infinite precision but greater computational cost than gradient-based methods because of the preprocessing stage for transformation. In particular, phase-based methods have demonstrated robustness to the failures of the translating-region assumption in real sequences. However, they often lack robustness to noise.

Hierarchical or coarse-to-fine estimation is a computationally efficient means of increasing measurement range and also improving noise immunity. Any estimation algorithm may be implemented in a hierarchical framework using a multiresolution representation of the input image such as a Gaussian or Laplacian pyramid.

In the next chapter, a hierarchical phase-based motion estimation algorithm, based on complex discrete wavelet transform, is introduced which combines efficiency, precision and robustness.

CHAPTER 3

MOTION ESTIMATION BASED ON A COMPLEX DISCRETE WAVELET TRANSFORM

The wavelet transform has become a popular technique especially in the past two decades, which found applications in various fields such as signal and image processing, pattern recognition and computer vision, astronomy, acoustics and geophysics. There are two main factors for this popularity.

The first reason is due to the intrinsic nature of the wavelet transform, which has a good time-frequency localization property. In order to understand this, standard Fourier techniques must be analyzed first. These techniques analyze the total frequency content of a signal, using infinite exponential waves. Since the Fourier transform (FT) is suited to the analysis of periodic signals, if the frequency content of a signal at a particular time is required, e.g. if the instant at which a spike occurred in the signal is required, the FT will be unable to provide the information. The lack of FT is that it provides good frequency localization, but no time localization.

To overcome this problem, the short-time Fourier transform (STFT) was developed, which analyzes the signal at specific time locations by using windowed exponential waves. This is done through multiplication of the exponential waves by a suitable window function, centered at the required time instant. The Gabor transform, used in this work, is a particular case of the STFT, which uses a Gaussian kernel as the window function. Gabor [18] proved that with this window the STFT achieved the best joint time-frequency localization.

Besides the good joint time-frequency localization property, STFT still presents a problem of scale dependency caused by the window function. To understand this, think again of seeking the time location of a spike in the signal. If the width of the spike is much smaller or larger than the width of the analysis window, there will be still a problem of unability to specify the time instant of spike occurrence. To overcome this problem, WT which analyzes

a signal at different time locations with kernels of varying sizes is used where the kernels are formed by translations and dilations of a prototype function called the mother wavelet.

The second factor for the popularity of the wavelet transform is due to Mallat [11]. He introduced the idea of multiresolution decomposition and showed that the wavelet coefficients were suitable to represent the information difference between two different resolution versions of the same signal. He also pointed out the similarity between the wavelet transform and quadrature mirror filter (QMF) banks. This enables for an efficient implementation of a discrete version of the WT (DWT).

Combining the scale independence and time-frequency localization properties of the wavelet transform with the relation of local translations in the image with FT phase gives a good idea of an approach to ME problem. In this chapter, we describe the complex discrete wavelet transform based ME approach of Magarey and Kingsbury [1, 19]. This transform combines the efficiency of a QMF implementation, with the fact that the corresponding filters may be modeled as scaled Gabor functions, providing in this way the desired Fourier/wavelet combination.

3.1 THE COMPLEX DISCRETE WAVELET TRANSFORM

In this section, we briefly review the continuous and discrete wavelet transforms in one and two dimensions as a basis to the more detailed description of the complex discrete wavelet transform of Magarey and Kingsbury [1, 19].

3.1.1 ONE-DIMENSIONAL WAVELET TRANSFORM

3.1.1.1 Continuous time

The continuous wavelet transform of a one-dimensional signal $I(x)$ is defined as [20, 22]

$$WT\{I(x)\} = D(\tau,\alpha) = \int_{-\infty}^{\infty} I(x)\psi_{\tau\alpha}(x)dx \qquad (3.1)$$

where $\psi_{\tau\alpha}(x)$ represents a set of wavelets, generated from the mother wavelet $\psi(x)$ by a change of scale (the scale of $\psi(x)$ is conventionally 1, and in $\psi_{\tau\alpha}(x)$ the scale parameter $\alpha\rangle 0$) and a translation in time (the function $\psi(x)$ is conventionally centered around 0, and

$\psi_{\tau\alpha}(x)$ is then centered around τ). The set of wavelets generated from the mother wavelet is formulated as follows.

$$\psi_{\tau\alpha}(x) = \frac{1}{\sqrt{\alpha}}\psi\left(\frac{x-\tau}{\alpha}\right) \tag{3.2}$$

where $\alpha, \tau \varepsilon \Re$. Notice for the scale factor α that, very large scales mean global views (stretched wavelets), while very small scales mean detailed views(shrunk wavelets).

It is obvious that, the continuous wavelet transform is highly redundant for representation of a signal. This redundancy may be reduced by sampling the scale and translation parameters, resulting in the discrete wavelet transform. Daubechies [22] used a dyadic sampling for α and a linear (proportional to the scale) sampling for τ , i.e., $\alpha = 2^j \alpha_0$ and $\tau = k\alpha,\ j, k \varepsilon Z$. The transform can then be written as a set of integer-indexed coefficients

$$D_{jk} = \int_{-\infty}^{\infty} \psi_{jk}(x) I(x) dx \tag{3.3}$$

where

$$\psi_{jk}(x) = 2^{-j/2}\psi\left(2^{-j}x - k\right) \tag{3.4}$$

3.1.1.2 Discrete Time

In most signal processing applications, the signal is sampled in time $\left(I = I(n), n \varepsilon Z\right)$ and a discrete version of the continuous wavelet transform (DWT) can be writen as,

$$D^{(j)}(n) = \sum_k I(k)\psi^{(j)}\left(2^j n - k\right) \tag{3.5}$$

or

$$D^{(j)}(n) = \left(I * \psi^{(j)}\right)(n) \downarrow 2^j,\ j = 1,2,... \tag{3.6}$$

where $\psi^{(j)}(n)$ are discrete-time wavelet filters and the operator $\downarrow 2^j$ represents downsampling of j times.

The main difference between this transform and its equivalent in continuous time is that the filters $\psi^{(j)}(n)$ are not perfectly scaled versions of one another[20], i.e. they do not satisfy the relation

$$\psi^{(j)}(n) = 2^{-\frac{j}{2}} \psi\left(-\frac{x}{2^j}\right)\Big|_{x=n} \tag{3.7}$$

However, this relation tends to hold as $j \to \infty$.

3.1.1.3 Wavelet Transform As a Filter Bank

In [11], Mallat introduced the multiresolution analysis concept and showed the suitability of wavelet bases to represent the difference in information between approximations of a signal at different resolutions. In turn, there is a close connection between a multiresolution representation and a subband filtering scheme, thus making it possible to implement the DWT defined by (3.5) using this scheme.

In a subband filtering scheme, the input signal $I(n)$ is split into two and simultaneously analyzed by a pair of halfband filters, a lowpass filter h_0 and a highpass filter h_1, followed by downsampling by a factor of 2 as shown in Figure 3.1. The highpass filter provides the first level of detail after downsampling by 2, while the downsampled lowpass output becomes a coarse approximation to the input signal, each of which have half the resolution but double the scale. The output of the lowpass branch is then input to the next stage, and analyzed in the same way as the original signal. This structure is repeated as required, resulting in a subband decomposition tree shown in Figure 3.2. Note that due to the downsampling operation, the number of output samples equals the number of input ones, therefore the output signal is critically sampled i.e. has no redundancy.

The relation between the DWT and the subband filter bank is that the output of the highpass branches may be interpreted as the result of the convolution of the signal $I(n)$ with the corresponding discrete wavelet filters $\psi^{(j)}$ of equation (3.5). Since the filter pair

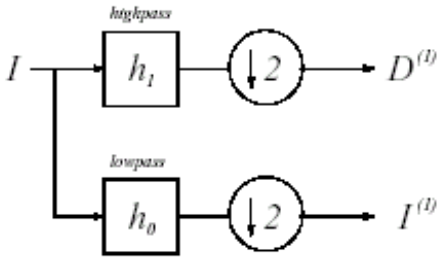


Figure 3.1 Two-band building block for dyadic DWT.

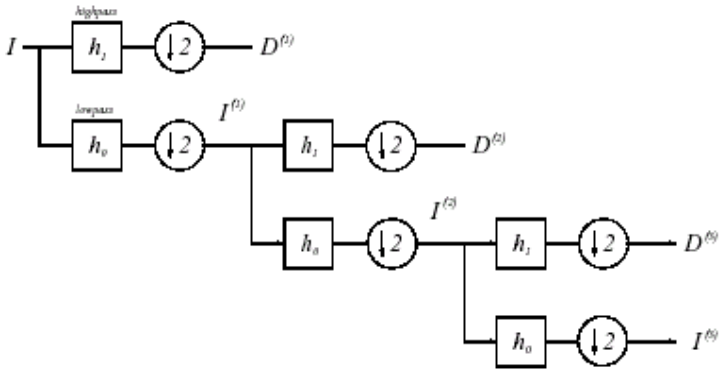


Figure 3.2 Subband decomposition tree (3 levels)

$\{h_0, h_1\}$ bears a direct relation with the wavelet filters $\psi^{(j)}$, using the Z-transform description of digital filters, this relation may be written as

$$\Psi^{(j)}(z) = H_1\left(z^{2^{j-1}}\right)\prod_{k=0}^{j-2} H_0\left(z^{2^k}\right) \tag{3.8}$$

where $H_0, H_1 and \Psi^{(j)}$ stand respectively for the Fourier transforms of $h_0, h_1 and \psi^{(j)}$, and $z = e^{iw}$.

Equation (3.8) shows that in order to obtain the j^{th} level coefficients of the transform, the input signal needs to be lowpassed and downsampled (j-1) times, and subsequently highpassed and downsampled once. The corresponding residual of this tree(i.e., the output of the j^{th} lowpass filter) may be obtained by a similar expression to (3.5)

$$I^{(j)}(n) = \sum_k I(k)\phi^{(j)}\left(2^j n - k\right) \tag{3.9}$$

or

$$I^{(j)}(n) = \left(I * \phi^{(j)}\right)(n) \downarrow 2^j \tag{3.10}$$

where $\phi^{(j)}$ is the equivalent scaling filter defined by the expression

$$\Phi^{(j)}(z) = \prod_{k=0}^{j-1} H_0\left(z^{2^k}\right) \tag{3.11}$$

(note that $\Phi^{(j)}$ is the Fourier transform of $\phi^{(j)}$). As $j \to \infty$, filtering by $\phi^{(j)}$ tends to give the average value, or DC component, of the input signal.

3.1.2 Two-Dimensional Discrete Wavelet Transform

As Mallat [11] shows, the wavelet transform in two dimensions may be implemented most efficiently by using a two-dimensional separable filter. This corresponds to writing two-dimensional mother wavelet as a multiplication of two one-dimensional functions and processing the columns and then rows of the image $I(n)$, as in the structure shown in Figure 3.3. The result is four subimages, each having one quarter size of the original image, so the separable 2-d DWT has no redundancy as in the 1-d case.

$I^{(1)}$ is a coarse version of the input image I. It has been lowpass filtered by h_0 and downsampled in each direction and therefore represents a half-resolution and double-scale version of the original image. Then, $I^{(1)}$ is used as the input to the next level of the subband tree. The remaining bandpass subimages(outcoming from the application of h_0h_1, h_1h_0 and h_1h_1 followed by corresponding downsampling in both directions) correspond to the two-dimensional wavelet coefficients where $\left\{D^{(1,j)}, D^{(2,j)}, D^{(3,j)}\right\}$ contain detail in the horizontal, vertical and diagonal directions respectively. Mathematically, the lowpass subimages' coefficients at scale j are

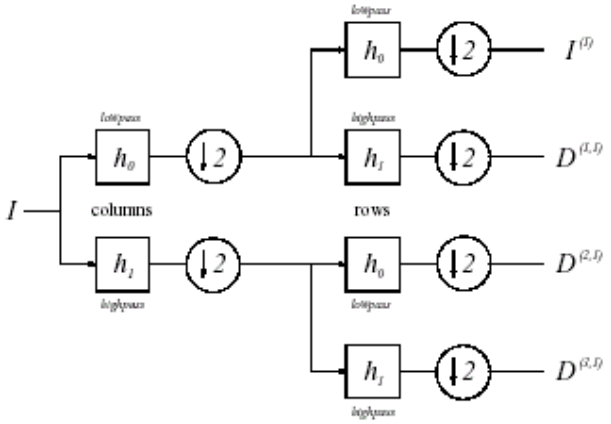


Figure 3.3 Building block for separable DWT on image I.

$$I^{(j)}(\underset{\rightharpoonup}{n}) = \sum_{\underset{\rightharpoonup}{k}} I(\underset{\rightharpoonup}{k})\phi^{(j)}\left(2^j \underset{\rightharpoonup}{n} - \underset{\rightharpoonup}{k}\right) \tag{3.12}$$

and the bandpass subimages' coefficients at scale j and subband s are

$$D^{(s,j)}(\underset{\rightharpoonup}{n}) = \sum_{\underset{\rightharpoonup}{k}} I(\underset{\rightharpoonup}{k})\psi^{(s,j)}\left(2^j \underset{\rightharpoonup}{n} - \underset{\rightharpoonup}{k}\right) \tag{3.13}$$

where $\phi^{(j)}$ and $\left\{\psi^{(s,j)}, s = 1,2,3\right\}$ are the corresponding two-dimensional scaling and wavelet filters, given by

$$\phi^{(j)}(\underset{\rightharpoonup}{n}) = \phi^{(j)}(n_1)\phi^{(j)}(n_2) \tag{3.14}$$

$$\psi^{(1,j)}(\underset{\rightharpoonup}{n}) = \psi^{(j)}(n_1)\phi^{(j)}(n_2) \tag{3.15}$$

$$\psi^{(2,j)}(\underset{\rightharpoonup}{n}) = \phi^{(j)}(n_1)\psi^{(j)}(n_2) \tag{3.16}$$

$$\psi^{(3,j)}(\underset{\rightharpoonup}{n}) = \psi^{(j)}(n_1)\psi^{(j)}(n_2) \tag{3.17}$$

Figure 3.4 shows the partition of the unit (normalised) frequency cell by a single-level DWT.

Note that at a given scale, each of the above defined wavelet filters emphasises features in the input image positioned at different orientations. Therefore the filter $\psi^{(1,j)}$, which is a

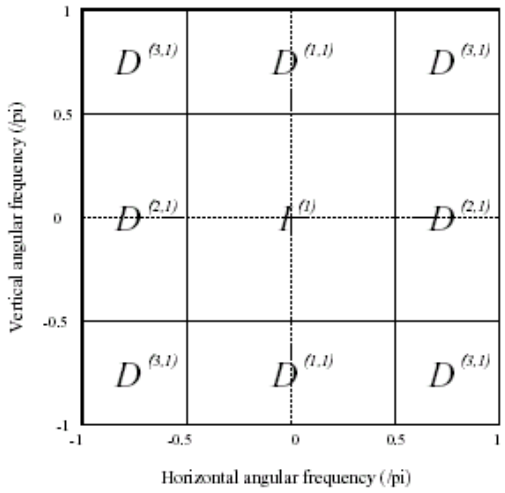


Figure 3.4 Partition of the two-dimensional unit frequency cell by a single-level separable DWT

combination of a highpass filter in the vertical direction and a lowpass filter in the horizontal direction, emphasises horizontal edges, while the filter $\psi^{(2,j)}$, which is the opposite combination, emphasises vertical edges. Finally, the filter $\psi^{(3,j)}$ emphasises diagonal edges, but note that it does not make distinction between diagonal edges oriented at $\pi/4$ or $3\pi/4$ radians. A solution to the distinction problem of diagonal edges will be introduced in section 3.1.3.3 through the introduction of mirror filters.

To summarise, the separable 2-d DWT decomposes an image into a hierarchy of subimages of different scales and orientations by using a set of distinctly scaled and oriented spatial filters.

3.1.3 COMPLEX DWT

In this section, the complex discrete wavelet transform (CDWT) developed by Magarey and Kingsbury [1, 2, 19] will be introduced. This transform uses a complex-valued filter pair $\{h_0, h_1\}$, instead of the real-valued pair used in Mallat's DWT subband tree described before. The CDWT was developed to be used in the phase-based motion estimator described before.

3.1.3.1 One-Dimensional CDWT

Up to now, we implemented DWT with real-valued filters. Since the ME algorithm uses the phase of a complex-valued transform, the $\{h_0, h_1\}$ pair must be complex-valued and they may be modeled as Gabor filters [1, 19]

$$h_0(n) \approx a_0 e^{-\frac{(n-n_0)^2}{2\sigma_0^2}} e^{iw_0(n-n_0)} \tag{3.18}$$

$$h_1(n) \approx a_1 e^{-\frac{(n-n_0)^2}{2\sigma_1^2}} e^{iw_1(n-n_0)} \tag{3.19}$$

for n=-L,...,L-1,

where 2L is the length of the filters, n_0 is set to $-1/2$ to position the Gaussian window symmetrically in the interval [-L, L-1], and w_0 and w_1 are the center frequencies. For proper choices of $\sigma_0 and \sigma_1$, Fourier transform of equation (3.18) and (3.19) can be writen as,

$$H_0(w) = a_0\sigma_0\sqrt{2\pi}e^{-\frac{(w-w_0)^2}{2/\sigma_0^2}}e^{iw_0(n-n_0)} \tag{3.20}$$

$$H_1(w) = a_1\sigma_1\sqrt{2\pi}e^{-\frac{(w-w_1)^2}{2/\sigma_1^2}}e^{iw_1(n-n_0)} \tag{3.21}$$

for $w\varepsilon[0,2\pi]$

For a particular choice of the parameters $a_0, a_1, \sigma_0, \sigma_1, w_0$ and w_1 (section 3.2.3), the equivalent wavelet and scaling filters (equations (3.8) and (3.11)) may also be approximated as Gabor functions [1, 19]

$$\psi^{(j)}(n) \approx a_j e^{-\frac{(n-n_j)^2}{2\sigma_j^2}}e^{iw_j(n-n_j)} \tag{3.22}$$

$$\phi^{(j)}(n) \approx \hat{a}_j e^{-\frac{(n-n_j)^2}{2\hat{\sigma}_j^2}}e^{i\hat{w}_j(n-n_j)} \tag{3.23}$$

for n=-(2^j-1)L,..., (2^j-1) (L-1), with $n_j = 2^{-j}n_0$. The parameters $a_j, \hat{a}_j, \sigma_j, \hat{\sigma}_j, w_j$ and $\hat{w}_j$ can be calculated from the parameters of h_0 and h_1 using equations (3.8) and (3.11).

The wavelet and scaling filters given by (3.22) and (3.23) have the same characteristics as the real-valued ones, in terms of not being perfectly scaled versions of one another. However, as perfect scaling is a desirable property for the application in mind, a prefilter f is applied to the input signal before the first level of the tree. This prefilter is defined by the equation

$$(h_0 * f)(2n) = \lambda . f(n) \tag{3.24}$$

where $\lambda = |H_0(0)|/2$ and H_0 is the Fourier transform of h_0. The purpose of the prefilter f is to simulate an infinitely large DWT tree. Consequently the resulting f-modified equivalent wavelet and scaling filters satisfy the relation (3.7). The perfectly scaled one-dimensional CDWT can be implemented as a standard DWT (Figure 3.2), except that the filters for the first level are $h_{0f} = h_0 * f$ and $h_{1f} = h_1 * f$, instead of h_0 and h_1.

Notice that in a perfectly scaled 1-d wavelet decomposition with its prefilter f has equivalent wavelet and scaling filters f-modified as follows,

$$\psi_f^{(j)}(n) = \left(\psi^{(j)} * f\right)(n) \tag{3.25}$$

$$\phi_f^{(j)}(n) = \left(\phi^{(j)} * f\right)(n) \tag{3.26}$$

In turn, the f-modified wavelet and scaling filters $\psi_f^{(m)}$ and $\phi_f^{(m)}$ can still be approximated as Gabor filters:

$$\psi_f^{(j)}(n) \approx a_{mf} e^{-\frac{(n-n_{mf})^2}{2\sigma_{mf}^2}} e^{iw_{mf}(n-n_{mf})} \tag{3.27}$$

$$\phi_f^{(j)}(n) \approx \hat{a}_{mf} e^{-\frac{(n-n_{mf})^2}{2\hat{\sigma}_{mf}^2}} e^{i\hat{w}_{mf}(n-n_{mf})} \tag{3.28}$$

where again the parameters can be calculated from those of h_0 and h_1 where,

$$w_{jf} = w_{1f} / 2^{j-1} \tag{3.29}$$

$$\hat{w}_{jf} = w_{0f} / 2^{j-1} \tag{3.30}$$

showing that the ratio $w_{jf} / \hat{w}_{jf}$ is independent of m, as desired.

Note: From now on, we assume the presence of the perfect-scaling prefilter f and therefore drop all f subscripts on filter names and parameters.

3.1.3.2 Two-Dimensional CDWT

In case of two dimensions, the transform is implemented as previously described in 2-d for real-valued filters i.e. in a separable way, processing first the columns and then the rows of the input image. As previously mentioned, the resulting two-dimensional wavelet and scaling filters may also be modeled as two-dimensional Gabor functions, which are given by the multiplication of the corresponding one-dimensional filters as in equations (3.14) to (3.17).

In practise, filtering an image with this set of two-dimensional Gabor functions results in an orientation analysis, i.e., each filter emphasises features within the input image which are oriented in the preferred direction of the filter. These directions are perpendicular to the corresponding two-dimensional center frequency vectors.

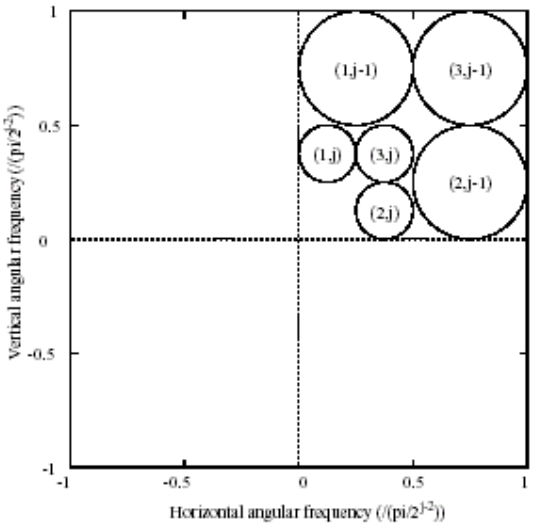


Figure 3.5 Elliptical (quasi-circular) contours of the magnitude responses of the two-dimensional CDWT wavelet filters at levels j and j-1.

$$\begin{aligned} \Omega^{(1,j)} &= \begin{pmatrix} w_j & \hat{w}_j \end{pmatrix}^T \\ \Omega^{(2,j)} &= \begin{pmatrix} \hat{w}_j & w_j \end{pmatrix}^T \\ \Omega^{(3,j)} &= \begin{pmatrix} w_j & w_j \end{pmatrix}^T \end{aligned} \tag{3.31}$$

3.1.3.3 Mirror filters

The 1-d CDWT wavelet filters have significant magnitude response only in the range $[0,\pi]$. Since the 2-d wavelet filters are products of the 1-d wavelet and scaling filters, they also have significant magnitude responses in the same range. This implies that the output of the corresponding two-dimensional wavelet filters is mainly localised in the first quadrant of the unit frequency cell, as shown in Figure 3.5. In terms of the orientation selectivity just mentioned, this is equivalent to analysing only orientations ranging from 0 to $\pi/2$. However, real-valued images contain significant information in the first and second quadrants of the unit frequency cell (the third and fourth quadrants are conjugated versions of the first and second quadrants).

In order not to loose any information when moving to the CDWT domain, the second quadrant (negative horizontal frequency, positive vertical frequency) of the unit frequency cell (i.e. orientations within the interval $[\pi/2, \pi)$) must be covered. For this purpose, a set of mirror filters is added to the filter bank. This is achieved by using the complex conjugates

h_0^* and h_1^* in parallel with h_0 and h_1 when filtering the rows of the image (Figure 3.6). Conjugating h_0 and h_1 reflects their magnitude frequency responses about w=0, so the conjugated filters cover the frequency range $[-\pi,0]$ Consequently, the output of the first level of the CDWT tree is eight complex subimages, two being lowpass versions of the original image and six corresponding to the two-dimensional wavelet coefficients. The subimages coming from the conjugate filters branch contain information related to the second quadrant of the unit frequency cell (see Figure 3.7). In this way, the CDWT distinguishes between orthogonal diagonal edges, which is not true for the standard real-valued DWT.

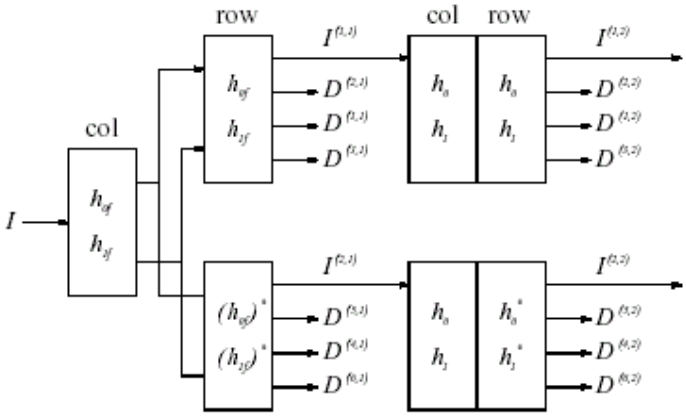


Figure 3.6 Two-dimensional CDWT (2 levels)

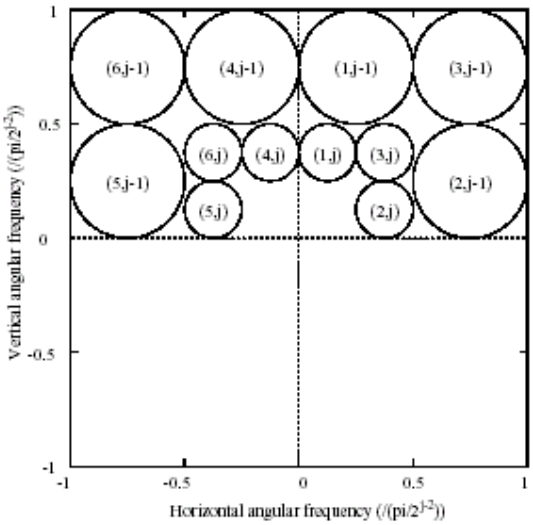


Figure 3.7 The magnitude responses of the mirror filters covering the second quadrant of the unit frequency cell.

The subsequent levels of the tree are computed by applying h_0 and h_1 as usual to the first quadrant lowpass subimage, and the mirror filters to the second quadrant lowpass subimage. Overall, this implies a redundancy of 4 to 1 for the CDWT, independent of the depth of the tree. Hence the complex version of the DWT is not critically sampled, however, this redundancy is crucial for the success of the motion estimation method (Section 3.2).

The two-dimensional mirror scaling and wavelet filters are

$$\phi^{(2,j)}(\vec{n}) = \phi^{(j)}(n_1)\phi^{(j)*}(n_2) \tag{3.32}$$

$$\psi^{(4,j)}(\vec{n}) = \psi^{(j)}(n_1)\phi^{(j)*}(n_2) \tag{3.33}$$

$$\psi^{(5,j)}(\vec{n}) = \phi^{(j)}(n_1)\psi^{(j)*}(n_2) \tag{3.34}$$

$$\psi^{(6,j)}(\vec{n}) = \psi^{(j)}(n_1)\psi^{(j)*}(n_2) \tag{3.35}$$

and the corresponding center frequencies of the mirror wavelet filters are given by

$$\begin{aligned} \Omega^{(4,j)} &= \begin{pmatrix} w_j & -\hat{w}_j \end{pmatrix}^T \\ \Omega^{(5,j)} &= \begin{pmatrix} \hat{w}_j & -w_j \end{pmatrix}^T \\ \Omega^{(6,j)} &= \begin{pmatrix} w_j & -w_j \end{pmatrix}^T \end{aligned} \tag{3.36}$$

3.1.3.4 Contributing Regions

At each scale of the transform, a pixel within a subband, or a subpel, contains information coming from a neighborhood surrounding the original pixel. The size of the contributing neighborhood at each level j (in relation to the pixel density of the original image) may be estimated by considering an approximately circular region with radius corresponding to twice the variance σ_j (see equation (3.22))

3.1.4 Summary

In this section, 2-d CDWT, which is built separably by using complex-valued filters(modeled as Gabor Kernels) is introduced. This transform analyses an image by decomposing it into a set of six orientation-tuned subbands at different resolutions. The inclusion of the mirror filters prevents information loss on transform by including the second quadrant of the unit frequency cell and provides emphasizing of diagonal orientations with different subbands. Figure 3.8 shows an example runtarget image and its CDWT

decomposition (5 levels). The upper right quadrant shows the real and imaginary parts of the finest vertical subband (top left and right of this quadrant) plus the real and imaginary parts of the mirror subband (bottom left and right). In the same way, the lower right quadrant shows the diagonal subbands, and the lower left quadrant shows the horizontal subbands. The upper left quadrant shows the same structure for the next coarser resolution, an so on. The smallest images in the upper left corner correspond to the lowpass residuals. Notice that, runtarget picture is used because of its wide-spectrum.

3.2 Motion Estimation Algorithm

Figure 3.9 shows the hierarchical structure of our algorithm. I_1 and I_2 are the reference and current frames respectively, which are transformed by CDWT decomposition from top to bottom, producing six complex bandpass subimages at each stage, and then input to the algorithm. The CDWT bandpass subimages at each level j are inputs to the motion estimator,

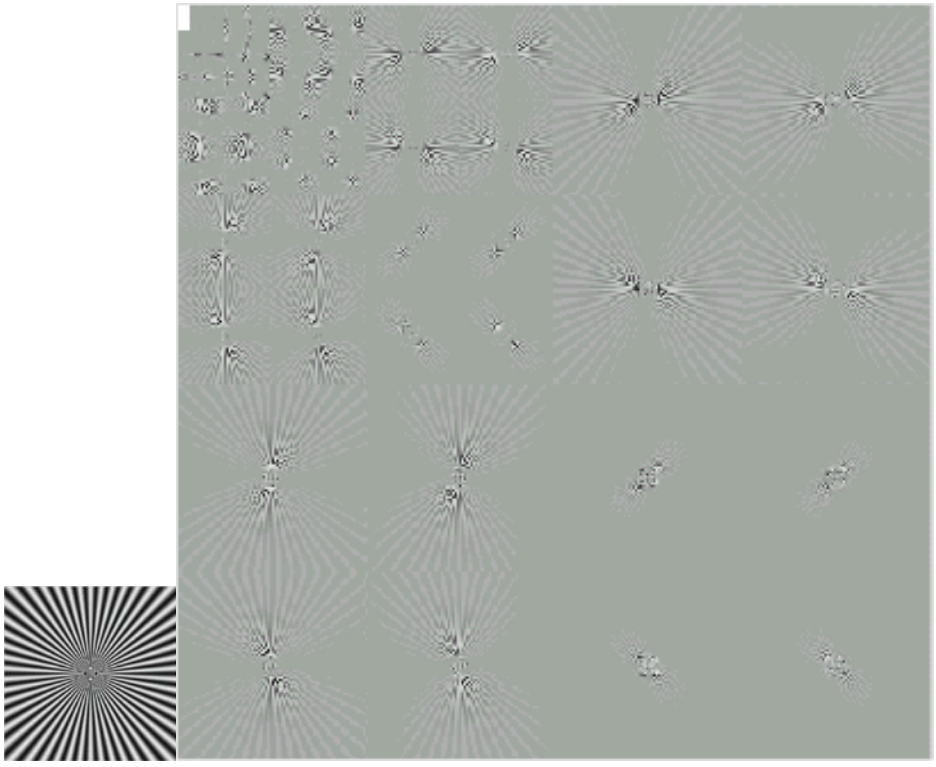


Figure 3.8 Frequency responses of subband filters in 2-d CDWT decomposition.

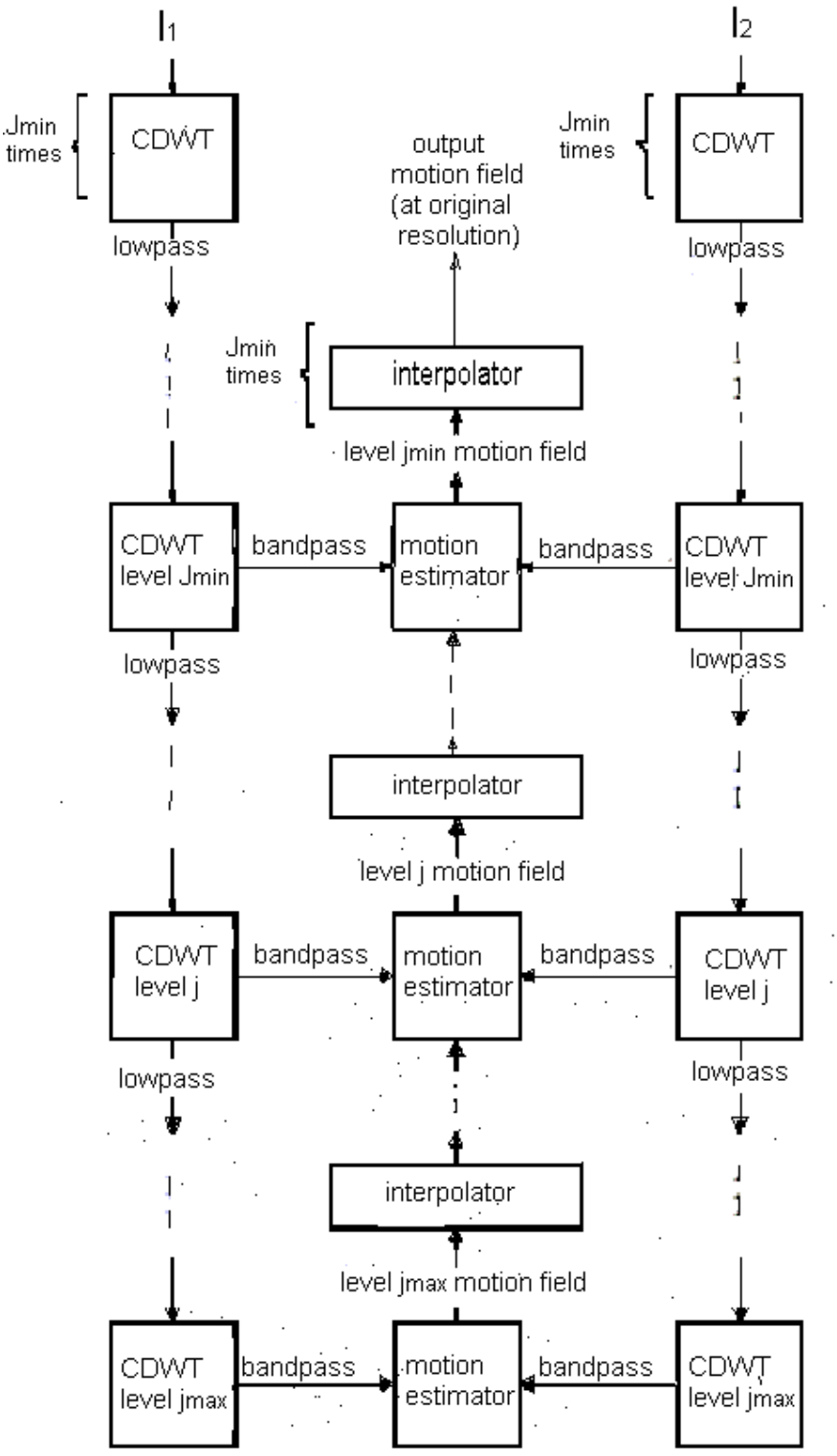


Figure 3.9 Hierarchical structure of CDWT-based motion estimation algorithm

whereas the lowpass subimages are inputs to the next decomposition stage. After CDWT decomposition is completed, motion estimation starts by matching the transform coefficients at the coarsest level $j_{\max}$, producing a motion vector for each subpel at this resolution. The algorithm works then from coarse to fine, using at each level the corresponding transform coefficients plus the estimates of the previous coarser level to produce a denser and more accurate motion field. Our algorithm produces a single motion estimate at each subpel at a given level. Because we use a dyadic DWT, each level-j subpel directly subtends four level j-1 subpels. The output field of each estimator must be quadrupled in size (interpolated by 2 in each direction) and scaled up by 2 before it can be used as an input to the next stage. The motion estimates are used as starting points for the next finer level of estimation by means of a warping or subband interpolation step. This process is repeated until a pre-set finest resolution $j_{\min}$ is attained. The algorithm halts at level $j_{\min}$, where the motion field has a density of $2^{-2j_{\min}}$, i.e. the field contains one motion estimate per block of $2^{j_{\min}}$ by $2^{j_{\min}}$ pels $(j_{\min} \geq 1)$. To obtain a full resolution motion field(i.e. one motion vector for every pixel in the original image), the motion field at level $j_{\min}$ is upsampled and interpolated $j_{\min}$ times.

3.2.1 SINGLE LEVEL ESTIMATION

3.2.1.1 Subband Squared Differences

The algorithm is based on a matching criterion at subpel $\vec{h}$ and subband (s, j) given by

$$SD^{(s,j)}(\vec{h}, \vec{f}) = \left| D_1^{(s,j)}(\vec{h} - \vec{f}) - D_2^{(s,j)}(\vec{h}) \right|^2 \tag{3.37}$$

where $\vec{f}$ is the offset vector. Since at least two distinct orientational subbands are needed to obtain unique estimate and all six orientational subbands to obtain a robust estimate, summing six subbands with a weighting factor gives the resultant matching criterion called the subband squared difference(SSD), given by

$$SD^{(j)}(\vec{h}, \vec{f}) = \frac{\sum_{s=1}^{6} \left| D_1^{(s,j)}(\vec{h} - f(\vec{h})) - D_2^{(s,j)}(\vec{h}) \right|^2}{\varepsilon^{(j)} P^{(s,j)}} \tag{3.38}$$

where $D_1^{(s,j)}$ and $D_2^{(s,j)}$ are the CDWT coefficients at level j and subband s of the reference and current frames, respectively. f($\underset{\sim}{h}$) is the displacement of subpel $\underset{\sim}{h}$ at scale j, which amounts to a displacement of $2^j \times f(\underset{\sim}{h})$ at the original resolution of the input images. $P^{(s,j)}$ is a weighting factor corresponding to the energy of the wavelet filter at subband s and scale j given by

$$P^{(s,j)} = \frac{1}{(2\pi)^2} \int_0^{2\pi} \int_0^{2\pi} \left| \Psi^{(s,j)}(\Omega) \right|^2 d\Omega \tag{3.39}$$

where $\Psi^{(s,j)}$ is the Fourier transform of $\psi^{(s,j)}$. The idea behind this factor is to eliminate the filter dependency when combining the different subbands. In turn $\varepsilon^{(j)}$ is a factor introduced to avoid the scale dependency of $P^{(s,j)}$ when combining the SSDs over different scales[1].

The SSD is analogous to the squared DFD in the intensity domain which was described in chapter 2. But there are two significant differences between those quantities. The first one is, in the case of the SSD, the average is performed over the set of oriented subbands with the filter energy weighting factor rather than over a set of neighboring pixels in the image as in DFD case. The second one is that the computed displacement $\underset{\sim}{f}$ for the SSD is real-valued rather than integer-valued.

3.2.1.2 CDWT Coefficients Interpolation and Position Shiftability

The reason why this real-valued displacement vector may be computed follows from the interpolability property of the CDWT, i.e. the ability of the CDWT to produce real-indexed coefficients from the known integer-indexed ones. In turn this is possible due to another property, termed shiftability which CDWT approximately satisfies.[19]. Therefore we can write

$$D^{(s,j)}(\underset{\sim}{h} - \underset{\sim}{f}) \approx \sum_{\underset{\sim}{k}} W_{\underset{\sim}{f}}^{(s,j)}(\underset{\sim}{k}) D^{(s,j)}(\underset{\sim}{h} - \underset{\sim}{k}) \tag{3.40}$$

where $W_f^{(s,j)}$ is a lowpass kernel modulated to the center frequency of the equivalent wavelet filter, i.e.

$$W_{\vec{f}}^{(s,j)}(\vec{k}) = H_{\vec{f}}(\vec{k})e^{i2^j(\Omega^{(s,j)})^T(\vec{k}-\vec{f})} \tag{3.41}$$

where H_f is a separable lowpass kernel given as,

$$H_f(\vec{k}) = h_{f1}(k1)h_{f2}(k2) \tag{3.42}$$

The 1-d interpolator function $h_f(k)$ can be either Windowed Sinc or the Lagrange Interpolator [19]. Notice that the range of k depends on the range of $\vec{f}$ values required. In practise only a unit range (in each direction) needs to be considered, because the CDWT coefficients may always be integer-shifted to cope with values that fall outside that range. For example,

$$D^{(j)}\left(\vec{n}+\vec{f}\right) = D^{(j)}\left(\vec{n}+\lfloor\vec{f}\rfloor+\left(\vec{f}-\lfloor\vec{f}\rfloor\right)\right) = D^{(j)}\left(\vec{n}'+\vec{f}'\right) \tag{3.43}$$

where $\vec{n}' = \vec{n}+\lfloor\vec{f}\rfloor \varepsilon Z$

and $\vec{f}' = \vec{f}-\lfloor\vec{f}\rfloor \varepsilon [0,1]$

Here, the symbol $\lfloor\vec{f}\rfloor$ indicates the floor operator on $\vec{f}$, i.e. the integer immediately to the left of $\vec{f}$ on the number line.

The simplest choice for the one-dimensional lowpass kernel $h_f(k)$ is the staircase interpolator which is the trivial Lagrange interpolator (L=1),

$$h_f(k) = \begin{cases} 1 & if \quad k = 0 \\ 0 & otherwise \end{cases} \tag{3.44}$$

$$\text{for f } \varepsilon \text{ [-0.5,0.5]} \tag{3.45}$$

In this case, equation (3.40) becomes

$$D^{(s,j)}(\vec{n}-\vec{f}) \approx D^{(s,j)}(\vec{n})e^{-i2^j(\Omega^{(s,j)})^T\vec{f}} \tag{3.46}$$

$$\text{for } \vec{f}\varepsilon[-0.5,0.5]\times[-0.5,0.5] \tag{3.47}$$

The above expression assumes that, for small f shifts, the magnitude of the CDWT coefficients is approximately constant, with only the phase changing according to the centre frequency of the corresponding wavelet filter. In his thesis [19] Magarey reports that this model "holds up well over the unit interval provided the original image has no strong spectral components in the passband of the filter, but is reasonably spectrally flat". Therefore, as the filter bandwidth decreases as j increases, this property is more readily satisfied at the lower resolutions, where the greatest accuracy is required.

3.2.1.3 Quadratic Surfaces

Consider the subband squared difference for one particular subband s,

$$SD^{(s,j)}(\underset{\rho}{h}, \underset{\rho}{f}) = \left| D_1^{(s,j)}(\underset{\rho}{h} - f(\underset{\rho}{h})) - D_2^{(s,j)}(\underset{\rho}{h}) \right|^2 \tag{3.48}$$

This equation may be further expanded into

$$SD^{(s,j)}(\underset{\rho}{h}, \underset{\rho}{f}) = \left| D_1^{(s,j)}(\underset{\rho}{h} - \underset{\rho}{f}) \right|^2 + \left| D_2^{(s,j)}(\underset{\rho}{h}) \right|^2 - 2\Re e\left\{ D_1^{(s,j)}(\underset{\rho}{h} - \underset{\rho}{f}) D_2^{*(s,j)}(\underset{\rho}{h}) \right\} \tag{3.49}$$

and the SSD can be further approximated as,

$$SD^{(s,j)}\left(\underset{\rho}{h}, \underset{\rho}{f}\right) \approx \left| D_1^{(s,j)}\left(\underset{\rho}{h}\right) \right|^2 + \left| D_2^{(s,j)}\left(\underset{\rho}{h}\right) \right|^2 - 2\left| D_1^{(s,j)}\left(\underset{\rho}{h}\right) D_2^{(s,j)}\left(\underset{\rho}{h}\right) \right| . \cos\left(\theta_1^{(s,j)}\left(\underset{\rho}{h} - \underset{\rho}{f}\right) - \theta_2^{(s,j)}\left(\underset{\rho}{h}\right)\right)$$
$$\text{for } \underset{\nu}{f} \varepsilon [-0.5, 0.5] \times [-0.5, 0.5] \tag{3.50}$$

where $\theta_1^{(s,j)}\left(\underset{\rho}{h} - \underset{\nu}{f}\right)$ and $\theta_2^{(s,j)}\left(\underset{\rho}{h}\right)$ are the phases of $D_1^{(s,j)}\left(\underset{\rho}{h} - \underset{\nu}{f}\right)$ and $D_2^{(s,j)}\left(\underset{\rho}{h}\right)$ respectively. This shows that minimizing $SD^{(s,j)}$ is very likely maximising the phase correlation between two subpels. We can characterise the $SD^{(s,j)}$ surface around its minimum line by using the approximation $\cos x \approx 1 - \frac{x^2}{2}$ and the staircase model provided by equation (3.46) we find,

$$SD^{(s,j)}(\underset{\rho}{h}, \underset{\nu}{f}) = (\left| D_1^{(s,j)}(\underset{\rho}{h}) \right| - \left| D_2^{(s,j)}(\underset{\rho}{h}) \right|)^2 + \left| D_1^{(s,j)}(\underset{\rho}{h}) \right| \left| D_2^{(s,j)}(\underset{\rho}{h}) \right| (2^j (\Omega^{(s,j)})^T \underset{\nu}{f} - \theta_{12}^{(s,j)}(\underset{\rho}{h}))^2$$
(3.51)

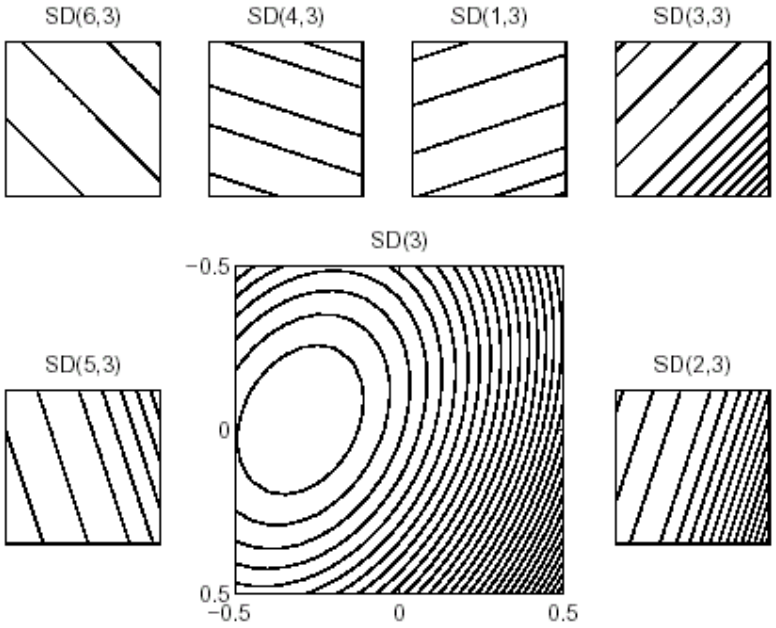


Figure 3.10 Contours of surfaces $SD^{(s,3)}$, s=1,...,6 and their sum $SD^{(3)}$ (centre bottom) for a typical displacement estimation.

where $\theta_{12}^{(s,j)}(\underline{n})$ is the phase difference between $D_1^{(s,j)}(\underline{n})$ and $D_2^{(s,j)}(\underline{n})$ defined by equation (3.52)

$$\theta_{12}^{(s,j)}(\underline{n}) = \angle\left[\frac{D_2^{(s,j)}(\underline{n})}{D_1^{(s,j)}(\underline{n})}\right] = \theta_2^{(s,j)}(\underline{n}) - \theta_1^{(s,j)}(\underline{n}) \tag{3.52}$$

Expression (3.51) represents a quadratic surface, whose minimum is a straight line [1] i.e. $SD^{(s,j)}$ is a quadratic surface with parabolic cross-section and contours parallel to its minimum line as shown in Figure 3.10. If the minimum lines corresponding to each subband do not lie too far apart, the sum (equation (3.51)) is also a quadratic surface, and it can be written in the form

$$SD^{(j)}(\underline{f}) = Af_1^2 + Bf_2^2 + Cf_1 f_2 + Df_1 + Ef_2 + G \tag{3.53}$$

or equivalently

$$SD^{(j)}(\underline{f}) = \alpha(f_1 - f_{10})^2 + \beta(f_2 - f_{20})^2 + \gamma(f_1 - f_{10})(f_2 - f_{20}) + \delta \tag{3.54}$$

where f_1 and f_2 are respectively the vertical and horizontal components of the real offset $\underset{\rho}{f}$.

The parameters A, B, C, D, E, G are functions of the coefficients $D_1^{(s,j)}(\underset{\rho}{h})$ and $D_2^{(s,j)}(\underset{\rho}{h})$, of the centre frequencies $\Omega^{(s,j)}$, and of the phase difference $\theta^{(s,j)}(\underset{\rho}{h})$, and can be directly derived from equations (3.38), (3.39), (3.48), (3.53) given by the expressions,

$$A = \sum_{n=1}^{6} \left| D_1^{(s,j)}(\underset{\rho}{h}) D_2^{(s,j)}(\underset{\rho}{h}) \right| \left(\Omega_1^{(s,j)}\right)^2 \tag{3.55}$$

$$B = \sum_{n=1}^{6} \left| D_1^{(s,j)}(\underset{\rho}{h}) D_2^{(s,j)}(\underset{\rho}{h}) \right| \left(\Omega_2^{(s,j)}\right)^2 \tag{3.56}$$

$$C = \sum_{n=1}^{6} \left| D_1^{(s,j)}(\underset{\rho}{h}) D_2^{(s,j)}(\underset{\rho}{h}) \right| .2\Omega_1^{(s,j)}\Omega_2^{(s,j)} \tag{3.57}$$

$$D = \sum_{n=1}^{6} \left| D_1^{(s,j)}(\underset{\rho}{h}) D_2^{(s,j)}(\underset{\rho}{h}) \right| .(-2)\Omega_1^{(s,j)}\theta^{(s,j)}(\underset{\rho}{h}) \tag{3.58}$$

$$E = \sum_{n=1}^{6} \left| D_1^{(s,j)}(\underset{\rho}{h}) D_2^{(s,j)}(\underset{\rho}{h}) \right| .(-2)\Omega_2^{(s,j)}\theta^{(s,j)}(\underset{\rho}{h}) \tag{3.59}$$

$$G = \sum_{n=1}^{6} \left(\left| D_1^{(s,j)}(\underset{\rho}{h}) \right| - \left| D_2^{(s,j)}(\underset{\rho}{h}) \right| \right)^2 + \sum_{n=1}^{6} \left| D_1^{(s,j)}(\underset{\rho}{h}) D_2^{(s,j)}(\underset{\rho}{h}) \right| \left(\theta^{(s,j)}(\underset{\rho}{h})\right)^2 \tag{3.60}$$

where

$$2^m \underset{\rho}{\Omega}^{(s,j)} = \left(\Omega_1^{(s)}, \Omega_2^{(s)}\right)^T \tag{3.61}$$

By using the above derived parameters, the surface minimum location is given by the expression

$$\underset{\rho}{f}_0 = \begin{pmatrix} f_{10} \\ f_{20} \end{pmatrix} = \frac{1}{(C^2 - 4AB)} \begin{pmatrix} 2BD - CE \\ 2AE - CD \end{pmatrix} \tag{3.62}$$

where the curvature parameters α,β,γ = A, B, C, and the surface minimum value is given by

$$\delta = G - Af_{10}^2 - Bf_{20}^2 - Cf_{10}f_{20} \tag{3.63}$$

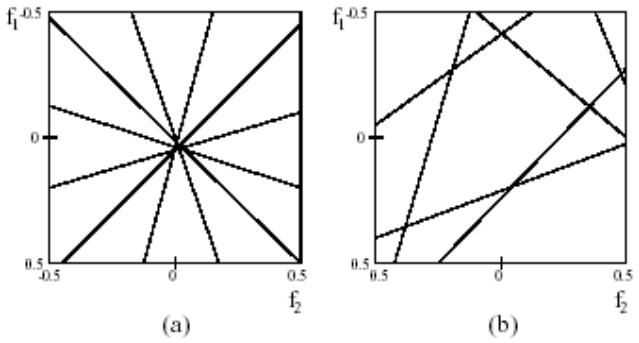


Figure 3.11 Minimum lines of quadratic surfaces given by expression (3.51) (a) when the motion is well-defined, the minimum lines corresponding to the different oriented subbands closely intercept each other (b) when the motion is not well-defined

which indicates the closeness of the match between $D_1^{(s,j)}(\vec{n} - \vec{f}_0)$ and $D_2^{(s,j)}(\vec{n})$ over $s = 1,...,6$.

It is interesting to note properties of these surfaces. The parameters $\{\alpha, \beta, \gamma\}$ specify the orientation and eccentricity of the ellipses. In general, the larger they are, the steeper the surface is at the minimum point, hence the closer lie the intercepts with each other of the minimum lines arising from the individual subbands(e.g., as in Figure 3.11a) This means that the corresponding motion estimate (minimum coordinates) is more precise (high confidence). Inversely, if the surface is flat in a particular direction(small curvature parameters), it indicates unreliability(low confidence) in the component of the motion in that direction(e.g. as in Figure 3.11b). This has a closer relation with the aperture problem, mentioned in Chapter 2. The component of the motion along the direction of an edge can not be determined unless the size of the "aperture" of the analysing device is larger than the length of the edge. We will refer again to these properties when discussing curvature correction (Section 3.2.2.4).

There are also some special cases for finding the minimum location. Equation 3.61 will fail in finding the minimum location where $C^2 = 4AB$. There are two possible causes for this:

- The null case, A=0. Since A is non-negative (see Equation 3.54), this must be the zero-activity case, in which all other parameters are zero as well. In such cases, set $\vec{f}_0 = (0,0)$.

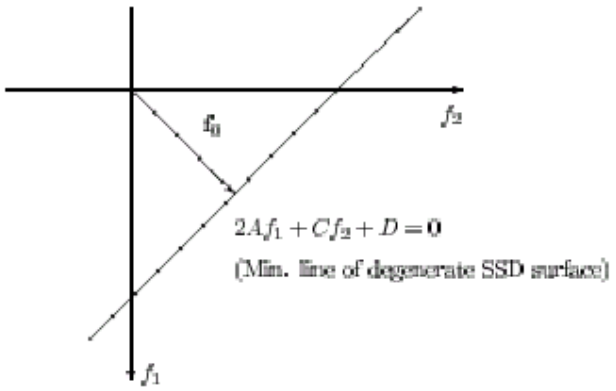


Figure 3.12 Choice of minimum location closest to the origin in case when $SD^{(j)}$ is degenerate.

- The degenerate case, $C^2 - 4AB = 0$. This situation is the manifestation of the aperture problem in the CDWT domain. Note that each individual (non-null) surface $SD^{(s,j)}$ falls into this category, so any signal that excites only one of the six subbands will produce a degenerate $SD^{(j)}$. In this case a unique minimum can not be found. Instead there is a minimum line specified by

$$2Af_1 + Cf_2 + D = 0 \tag{3.64}$$

where $\underset{\rho}{f}_0$ is set to the value on the line with the smallest perpendicular distance from the origin as shown in Figure 3.12, given by the expression,

$$\underset{\rho}{f}_0 = \begin{bmatrix} -2AD \\ -C \end{bmatrix} \Big/ \left(C^2 + 4A^2\right) \tag{3.65}$$

Besides this special cases, it is obvious that when a component of $\underset{\rho}{f}_0$ outside the range $[-0.5,0.5]$ is produced, null values are also assigned. Because that range is the absolute limit of validity of the staircase approximation (equation 3.46) on which ME algorithm is based.

3.2.2 Hierarchical Estimation

3.2.2.1 Interpolation and Scaling of Quadratic Surfaces

As previously mentioned, the hierarchical approach to ME is to use coarse level estimates as starting points or initial guesses. Therefore the first step is to compute the coarsest level $j_{\max}$ motion estimate, which is output in the form of a field of SD parameters, $SD^{(j_{\max})} = \{f_{10}, f_{20}, \alpha, \beta, \gamma, \delta\}$ or $\{A, B, C, D, E, G\}$. In other words, the parameters $f_{10}, f_{20}, \alpha, \beta, \gamma, \delta$ are calculated for every subpel at that level. Before the $SD^{(j_{\max})}$ is input to the next finer stage $j_{\max} - 1$ of the motion estimation, it must be interpolated to attain the required pixel density, and scaled to account for the change in coordinate system $\underset{f}{\rho} \to 2\underset{f}{\rho}$, producing a new field $SD'^{(j_{\max})}$.

The interpolation is done separately in the columns and then rows, by upsampling each column(row) followed by filtering with an interpolation kernel. The simplest interpolation is staircase, in which each parameter is simply copied to its four finer level neighbours. Another choice is the bilinear kernel given by $[1331]/4$. The scaling factors are

$$\alpha', A' \to \alpha/4, A/4 \tag{3.66}$$

$$\beta', B' \to \beta/4, B/4 \tag{3.67}$$

$$\gamma', C' \to \gamma/4, C/4 \tag{3.68}$$

$$D', E' \to D/2, E/2 \tag{3.69}$$

$$f_{10}, f_{20} \to 2f_{10}, 2f_{20} \tag{3.70}$$

3.2.2.2 Cumulative Squared Differences

Once obtained the field $SD'^{(j_{\max})}$, motion is estimated at level $(j_{\max} - 1)$ following the same procedure (i.e., using equations (3.51) (3.53) (3.54) (3.38)). The resulting field of parameters $SD^{(j_{\max}-1)}$ is added to $SD'^{(j_{\max})}$ forming the cumulative squared difference (CSD) at level $(j_{\max} - 1)$. This process (interpolate, find next estimate , add) is repeated iteratively, i.e.

$$CSD^{(j)}(\underset{h}{\rho}, \underset{f}{\rho}) = \begin{cases} CSD'^{(j+1)}(\underset{h}{\rho}, \underset{f}{\rho}) + SD^{(j)}(\underset{h}{\rho}, \underset{f}{\rho}) & j_{\min} \le j < j_{\max} \\ SD^{(j)}(\underset{h}{\rho}, \underset{f}{\rho}) & j = j_{\max} \end{cases} \tag{3.71}$$

until the desired resolution $j_{\min}$ is attained. In this way, information from all resolutions is incorporated into one single quadratic surface to form the final motion estimate. This combination of information from all levels is a means to tackle the aperture problem, since surfaces originating from coarser levels(large support regions) will be steeper and therefore contribute more to the cumulative surface than surfaces from finer levels(small support regions).

In addition, as the scale dependency of the SSDs was eliminated through the introduction of the $\varepsilon^{(j)}$ factor in equation (3.38), this means that in fact the finer levels have greater weight, allowing for increasing adaptivity to local variations in the motion field.

Notice that, in some given references, there is no $\varepsilon^{(j)}$ factor in equation (3.38), instead $CSD'^{(j+1)}(\underline{h}, \underline{f})$ term in equation (3.71) is multiplied by a weighting factor of $4\lambda^4$. This term is chosen to stop level dependency of $P^{(s,j)}$ factor given by equation (3.39). The effect is therefore to give the finer level information a greater weight in the final motion estimate.

3.2.2.3 Coarse-to-Fine Approaches

When combining information from two subsequent levels, one of two approaches may be used.

The first approach is the non-refining strategy where the finer level estimate $SD^{(j)}$ is computed independently of the previous level estimate $CSD'^{(j+1)}$.

The second approach is the refining strategy, where the previous level estimate is used to warp the CDWT coefficients of the reference image at the next finer level, using equation (3.40) to produce new coefficients

$$D_{1,\underline{f}_0}^{(s,j)}(\underline{h}) \approx D_1^{(s,j)}(\underline{h} - \underline{f}_0) \tag{3.72}$$

Therefore the field $SD^{(j)}$ is computed using the warped coefficients $D_{1,\underline{f}_0}^{(s,j)}(\underline{h})$ in place of the original $D_1^{(s,j)}(\underline{h})$. In addition, due to this warping a translation of the origin back to (0,0) is necessary before addition to the previous estimate $CSD'^{(j+1)}$. For a better understanding, the refining strategy can be stated as follows,

- Estimate coefficients $\left\{D_1^{(s,j)}\left(\underline{h} - \underline{f}_0'\right), n = 1,...,6\right\}$ using equations (3.40) to (3.42).
- Form $SD^{(j)}\left(\underline{h}, \underline{f} + \underline{f}_0'\right)$ by using the interpolated coefficients instead of the integer indexed ones

- Find parameters of valid $SD^{(j)}\left(\vec{h}, \vec{f}\right)$ from those of $SD^{(j)}\left(\vec{h}, \vec{f} + \vec{f}_0'\right)$ (i.e. translate origin back to (0,0))
- Add valid $SD^{(j)}$ to (scaled) $CSD'^{(j+1)}$ to form $CSD^{(j)}$ (equation 3.71)
- Transform $CSD^{(m)}$ to $\left\{\vec{f}, \alpha, \beta, ...\right\}$ form.

Instead of the staircase kernel used in equation (3.46), the interpolation in (3.72) is performed using a four-tap windowed-sinc kernel

$$h_f(k) = \left(\frac{\cos\frac{\pi}{2}(k-f)}{1-(k-f)^2}\right)\left(\frac{\sin\pi(k-f)}{\pi(k-f)}\right) \qquad (3.73)$$

k = -2,...,1

which Magarey[19] showed to provide better accuracy.

The main advantage of the refining strategy is that uniform motion of large-scale features can be found with high accuracy. However, because estimation can only refine valid previous level estimates, any motion of smaller scale features independent of large-scale motion will not be detected. The non-refining strategy allows scale-dependent motion to be estimated, while retaining the ability of coarse scale estimates to propagate down the pyramid in the absence of clear finer scale information. Consequently, in cases where the motion of small features may be assumed to broadly follow that of the larger, such as stereo matching, the refining strategy is likely to be suitable. However in video coding one may not make such an assumption, and the non-refining strategy may be preferable.

Another advantage of the refining strategy is about the maximum disparity value to be estimated with accuracy. Due to the limit on the range of $\vec{f}$ values imposed by the use of the staircase model when computing the SSDs, the maximum range of estimated $\vec{f}_0$ values is restricted to $0.5 \times 2^{j_{\max}}$ in each direction for the non-refining strategy, and $0.5 \times \left(2^{j_{\max}} + 2^{j_{\max}-1} + ... + 2^{j_{\min}}\right)$ for the refining strategy.

Consequently, refining strategy increases maximum disparity value to be estimated with accuracy. But any motion of smaller scale features independent of large-scale motion can not be detected.

3.2.2.4 Curvature Correction

It has been stated previously that flat surfaces indicate low confidence in the motion estimate, while steep surfaces indicate high confidence. In addition, coarser levels estimates are more aperture-free than finer levels due to their large region of support. In a hierarchical approach, coarser level estimates, hopefully aperture free, must be passed down to aperture-affected estimates at lower levels, where aperture-free estimates aimed to determine the parallel component of the motion. The curvature of the $SD^{(j)}$ surface parallel to the edge determines how much the component of motion in that direction being passed down from the coarser level is changed. To allow the parallel component of the coarser level estimate to propagate unchanged down the pyramid, aperture-affected ellipses at a fine level should be infinitely eccentric(i.e. ratio of major to minor axes is infinite) and parallel to the underlying edge, indicating zero curvature in that direction.

As can be seen from Figure 3.10, each $SD^{(s,j)}$ has infinite eccentricity. Consequently, if an edge is present, one of the six subbands (each having different orientation) will be excited and the sum, $SD^{(j)}$ will have infinite eccentricity. But in practice, the CDWT wavelet filters have non-zero bandwidth and any edge will induce activity in more than one subband, producing elliptical contours and therefore finite eccentricity.

The variation of eccentricity with edge angle is dependent on the properties of the original basis pair $\{h_0, h_1\}$ - bandwidth and center frequencies. If a Gabor filter pair is chosen which responds to edges of all orientations with ellipses of near-constant eccentricity, it will be possible to correct the curvatures of edge-dominated surfaces to give them very large eccentricity, while leaving all other surfaces unchanged. In this way the parallel component of motion at fine-level aperture-affected subpels can be weakened, allowing the (more certain) parallel component of coarse-level estimates to propagate unchanged.

For this purpose, a circular bowl with the same minimum location $\vec{f}_0$ of $SD^{(j)}$, is subtracted from all $SD^{(j)}$. The modified quadratic surface is given by,

$$SD'^{(j)}\left(\vec{h}, \vec{f}\right) = SD^{(j)}\left(\vec{h}, \vec{f}\right) - \rho\left(f_1 - f_{10}\right)^2 - \rho\left(f_2 - f_{20}\right)^2 \qquad (3.74)$$

where ρ [1, 19] is the radius of the circular bowl. The location and height δ of the minimum of $SD^{(j)}$ are unaffected by curvature correction. The radius ρ is chosen so that all surfaces with eccentricity $e \geq e_t$, for some e_t (In [19] Magarey showed that 3.5 is a

suitable value for e_t), have very large eccentricity after correction. The derivation of ρ is given in [19] as,

$$\rho = \min\left\{\left(\frac{\alpha+\beta}{e_t^2+1}\right), 0.98\lambda_2\right\} \tag{3.75}$$

where λ_2 is defined as

$$\lambda_2 = \alpha + \beta - \sqrt{(\alpha-\beta)^2 + \gamma^2} \tag{3.76}$$

Consequently, curvature correction is applied to the entire field of surfaces $SD^{(j)}$ before combining with previous level surfaces $CSD'^{(j+1)}$.

3.2.2.5 Confidence Measure

The algorithm will produce estimates of motion, except some special cases (see section 3.2.1.3), regardless of testing the confidence of estimation. It is obvious that passing unconfident estimates (due to the break down of the translation model) to lower levels in a hierarchical manner will cause false matches. So, a confidence measure must be defined and the estimates, which does not satisfy the confidence criteria, must not be passed to finer levels.

The confidence measure adopted by Magarey et al. is based on the weighted sum of perpendicular distances from the six minimum lines given by each of the $SD^{(s,j)}$ at the minimum point $\underset{\rho}{f}_0$, i.e. the residual

$$r^{(j)}(\underset{\rho}{h}) = \sum_{s=1}^{6} \frac{\left|D_1^{(s,j)}(\underset{\rho}{h}) D_2^{(s,j)}(\underset{\rho}{h})\right|}{P^{(s,j)}} \left(\left(\nabla\theta^{(s,j)}(\underset{\rho}{h})\right)^T \underset{\rho}{f}_0 - \theta_{12}^{(s,j)}(\underset{\rho}{h})\right)^2 \tag{3.77}$$

where $\nabla\theta^{(s,j)}(\underset{\rho}{h})$ is the phase gradient. In the ideal translational model case, all six constraint lines coincide and the residual is zero. On the other hand, if the six subbands consist of unrelated signals, the six lines are dispersed, and the residual is non-zero. This residual may be computed from a quantity called the discrepancy

$$\Delta^{(j)}(\underset{\rho}{h}) = \sum_{s=1}^{6} \frac{\left(\left|D_1^{(s,j)}(\underset{\rho}{h})\right| - \left|D_2^{(s,j)}(\underset{\rho}{h})\right|\right)^2}{P^{(s,j)}} \tag{3.78}$$

and from the surface minimum value $\delta(\underset{\mu}{h})$ (given by equation (3.63)) as

$$r^{(j)}(\underset{\rho}{h}) = \frac{\delta - \Delta^{(j)}(\underset{\rho}{h})}{E^{(j)}(\underset{\rho}{h})} \tag{3.79}$$

where

$$E^{(j)}(\underset{\rho}{h}) = \sum_{s=1}^{6} \frac{\left|D_1^{(s,j)}(\underset{\rho}{h}) D_2^{(s,j)}(\underset{\rho}{h})\right|}{P^{(s,j)}} \tag{3.80}$$

Finally, the confidence measure is defined as

$$C^{(j)}(\underset{\rho}{h}) = 1 - \frac{r^{(j)}(\underset{\rho}{h})}{2} \tag{3.81}$$

The reason for this definition is so that the highest values of $C^{(j)}(\underset{\rho}{h})$ indicate the most certain estimates. The confidence measure always lies in the range $[-\infty,1]$ due to the condition that residual is always positive.

3.2.3 Filters' Coefficients and Corresponding Gabor Parameters

One of the rotation invariant (RI) filter pair $\{h_0, h_1\}$ utilised in this work corresponds to two 4-tap complex filters with coefficients,

$$h_0 = [1-i \quad 4-i \quad 4+i \quad 1+i]/10 \tag{3.82}$$

$$h_1 = [-1-2i \quad 5+2i \quad -5+2i \quad 1-2i]/14 \tag{3.83}$$

and corresponding Gabor parameters are $D = 2$, $n_0 = -0.5$, $w_0 = \pi/6$, $w_1 = 0.76\pi$, $\sigma_0 = 0.97$, $\sigma_1 = 1.07$, $a_0 = 0.47$, and $a_1 = 0.43i$.

The second Gabor filter pair corresponds to two 8-tap complex filters with coefficients,

$$h_0 = [-0.2 - 1i \quad 1 - 5i \quad 14 - 14i \quad 35 - 9i \quad 35 + 9i \quad 14 + 14i \quad 1 + 5i \quad -0.2 + 1i]/100$$

(3.84)

$$h_0 = [0.2 - 1i \quad 1 + 5i \quad -14 - 14i \quad 35 + 9i \quad -35 + 9i \quad 14 - 14i \quad -1 + 5i \quad -0.2 - 1i]/100$$

(3.85)

and corresponding Gabor parameters are $D = 4$, $n_0 = -0.5$, $w_0 = \pi/6$, $w_1 = 0.83\pi$, $\sigma_0 = 1.27$, $\sigma_1 = 1.27$, $a_0 = 0.39$, and $a_1 = 0.39$.

The prefilter f discussed before is a 3-tap complex filter with coefficients

$$f = [-i \quad 5 \quad i]/5 \tag{3.86}$$

3.2.4 Summary

This chapter has described the CDWT-based motion estimation algorithm of Magarey and Kingsbury[1, 2, 3, 7, 19]. The algorithm is based on a phase-matching criteria implemented in a hierarchical manner, i.e. it works by matching the transform phase at each level. This is achieved through the definition of a quantity called the subband square difference(SSD) at each subpel for all orientations (subbands). The six oriented SSD surfaces are added to produce a single matching surface whose minimum location is the motion estimate at that subpel. For the accuracy of estimates at each subpel, some improvement methods to the original method is introduced, such as confidence thresholding, curvature correction and refining strategy. In the next chapter, the algorithm will be analyzed and experimented.

CHAPTER 4

SIMULATIONS

In previous chapter, the ME algorithm based on complex discrete wavelet transform is analyzed and implemented. The algorithm uses the relation between spatial translations of image intensity with phase shifts in Fourier domain in order to estimate motion in a hierarchical manner.

The ME algorithm has several choices of parameters and options for obtaining more accurate results for various stereo image pairs. In this chapter, the effect of parameter choices and option inclusions on accuracy of ME algorithm will be analyzed and experimented. The wide range of options and parameters that will be analyzed is given below:

- CDWT Analysis Stage Options and Parameters
  Usage of 8-tap Gabor filters or 4-tap Gabor filters.
- Options and Parameters In A Level
  Confidence Threshold
  Curvature Correction Parameter e_t
- Options and Parameters Between Levels
  Number of Pyramid Levels $j_{\max}$
  Choice of Finest Level $j_{\min}$
  Coarse-to-fine Interpolation Type

4.1 Test Data

In all motion estimation tests, synthetic image sequences are used in order to quantify the accuracy of the proposed algorithm with respect to true motion fields. Since, synthetic image sequences are obtained by a camera motion relative to a static scene, true motion vectors of the image sequence is readily known.

There is a constraint on input images coming from CDWT decomposition tree. The dimensions of the input images must be a multiple of $2^{j_{max}}$. Because of this constraint, the original images may be extended to the nearest multiple of $2^{j_{max}}$ or a portion of the original image with dimensions a multiple of $2^{j_{max}}$ may be used. But in the second technique, there might be significant information loss. So, first technique is commonly used.

A brief description of synthetic image sequences that will be used in testing the algorithm is given below. Note that the choice of the tree sequence is due to the property that motion can be well defined since the picture is not highly or weakly featured and the choice of Yosemite sequence is to test algorithm against discontinuous motion field. (Synthetic images can be obtained from ftp address ftp.csd.uwo.ca/pub/vision)

Translating Tree(T.Tree): This and the following sequences are obtained by simulating camera motion relative to a scene featuring a tree on a smooth background. The current frame, shown in Figure 4.1(a), is the 160x160 extended version of the 20th translating tree image, which is a horizontally displaced version of the 18th translating tree sequence. The true motion field is composed of vectors with magnitudes ranging from $3.46 \le d_{hor} \le 4.52$. The aim to use this sequence is to test ME algorithm for horizontal translational motion.

Rotating Tree(R.Tree): This sequence is obtained by rotating the 20th translating tree image and rotating 1 degrees in counter-clock wise direction. The size of the images are extended to 160x160. The aim of this sequence is to test ME algorithm against 2d-translational motion i.e. horizontal and vertical translational motion.

Diverging Tree(D.Tree): The sequence simulates camera motion towards the scene i.e. an analog zooming operation. The current and reference frames are 144x144 extended version of the 20th and 19th diverging tree images. The true motion field is composed of vectors ranging from $(0,0)$ in the center to $(2,2)$ pixels on edges. The aim to use this sequence is to test ME algorithm for divergent motion.

Yosemite(Y.Seq.): The sequence is a simulated "fly-through" of Yosemite National Park, produced by Lynn Quam at Stanford Research Institute which is shown in Figure 4.1(b). The current and reference frames are 256x320 extended version of the 20th and 19th Yosemite

Figure 4.1 (a) Current frames of synthetic tree sequences (b) Current frame of Yosemite sequence

sequence. The motion is of two kinds: divergent, as the camera moves along the line of the valley, and translating, as the clouds translate from left to right at 2 pels/frame. The maximum displacement is (3.86,-3.88) pels at the lower left corner. The aim to use this sequence is to test ME algorithm for discontinuous motion field.

Note that, by these four sequences, ME algorithm will be tested for motion in all possible dimensions and discontinuous motion fields. In Figure 4.2, the corresponding true motion vector fields for tree and Yosemite sequences are given, where knowledge of true motion vectors is necessary to quantify the accuracy of the estimates produced by the algorithm.

4.2 Error Measurement

Since synthetic sequences are used for testing the algorithm, true motion vectors of the images are known. This enables error measurement of estimated optical flow fields with respect to known true motion fields.

If image sequences are thought of a set of slides in a projector, a 3d displacement estimate $\underline{\vec{s}} = (s_1, s_2, 1)$ with third dimension is set to 1 connects each pel to its corresponding pel in the next slide. Using this 3d displacement estimates, an error measure can be defined. One of the common error measurement choices is the relative error defined by equation (4.1).

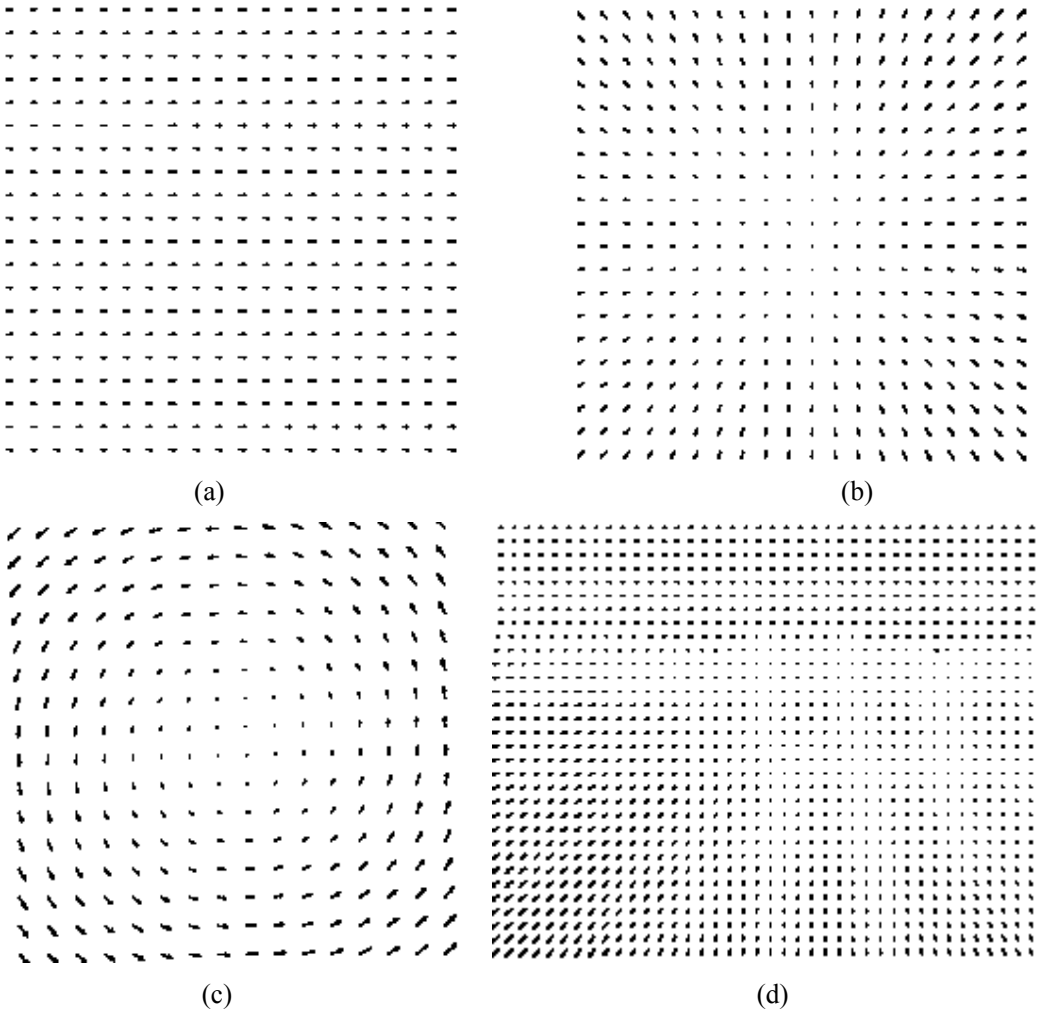


Figure 4.2 True motion fields of synthetic sequences (a) Translating Tree (b) Diverging Tree (c) Rotating Tree (d) Yosemite

$$RE(\vec{s}) = \frac{\left\| \vec{d} - \vec{s} \right\|}{\left\| \vec{d} \right\|} \tag{4.1}$$

where $\vec{d}$ is the corresponding true motion vector.

Instead of relative error measure, mean error angle measure defined by equation (4.2) will be used due to the property that it handles large and small displacements without the amplification inherent in a relative measure error.

$$\varphi_E(\vec{s}) = \cos^{-1}\left(\frac{\vec{s}.\vec{d}}{\|\vec{s}\|\|\vec{d}\|}\right) \tag{4.2}$$

By using equation (4.2), error is calculated through averaging error angle of an inner portion. This is because of the fact that edge effected measurement results must be excluded from average value. For comparison with other algorithms, MSE values of all motion field estimates are also calculated using the final algorithm.

4.3 Testing The Algorithm

In this section, various options and parameters of the ME algorithm will be analyzed and an optimum algorithm is investigated.

4.3.1 Confidence Threshold

Confidence threshold is one of the most important parameters in popular ME algorithms. For this purpose, a scalar confidence measure (section 3.2.2.5) is defined which expresses the certainity of the motion estimate at given pel. This value can be in the range $(-\infty,1)$ reaching $-\infty$ for random noise data, and 1 for a well-defined motion data.

In this test, the aim is to obtain an optimum value for confidence threshold. The criteria is to set confidence threshold to a high value which is enough to filter unconfident "possibly wrong" estimates and to a low value which does not reject confident "possibly true" estimates. In order to find an optimum value, a test is done on diverging tree sequence as shown in Figure 4.3, where the results are also given in Table 4.1. In test, both the field density and the mean error angle is calculated at level 2 for various confidence thresholds, where density is the number of confident pels over field size. Note that mean error angle is calculated for only confident pels.

Discussion

In Figure 4.3, it can be seen that increasing confidence threshold decreases both mean error angle and field density. This means that as threshold increases, accuracy of the estimates also increases with decreasing number of confidently calculated pels.

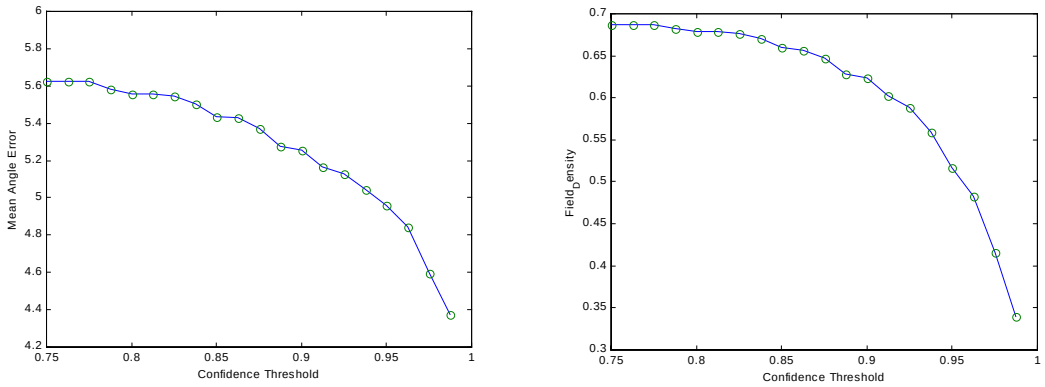


Figure 4.3 Mean error angle and field density plots for diverging tree image at level 2.

Confidence Threshold	Mean Error Angle	Field Density
0.7500	5.625	0.687
0.7625	5.625	0.687
0.7750	5.625	0.687
0.7875	5.583	0.682
0.8000	5.559	0.679
0.8125	5.559	0.679
0.8250	5.544	0.677
0.8375	5.506	0.670
0.8500	5.436	0.660
0.8625	5.429	0.656
0.8750	5.371	0.648
0.8875	5.277	0.629
0.9000	5.255	0.623
0.9125	5.163	0.602
0.9250	5.130	0.589
0.9375	5.047	0.559
0.9500	4.959	0.517
0.9625	4.845	0.483
0.9750	4.596	0.417
0.9875	4.377	0.340
1.0	3.924	0.102

Table 4.1. Mean error angle and field density calculated for various confidence thresholds.

Notice that when confidence threshold is 0.95, 11.9 percent of improvement is achieved in mean error angle with a 17 percent rejection of pel estimates. After this point, the accuracy increases more with a highly reduced number of confident estimates. After confidence threshold value of 0.9625 more increase in accuracy is achieved with a dramatically decrease in density of estimates. Consequently, an optimum confidence threshold can be chosen between 0.95 and 0.9625. From now on, 0.95 for confidence threshold will be used as default.

4.3.2 Number of Pyramid Levels $j_{\max}$

There are two limitations on maximum number of levels $j_{\max}$. The first limitation is induced by image size. If we factorize the image dimensions as,

$$No\_of\_rows = r \times 2^{l_r} \tag{4.3}$$

$$No\_of\_cols = c \times 2^{l_c} \tag{4.4}$$

where r and c are odd, we must have,

$$j_{\max} \le \min(l_r, l_c) \tag{4.5}$$

This limitation is due to the fact that an image must have at least 4 samples for a 2-d DWT to be possible. To increase the possible range of $j_{\max}$, input images are boundary replicated to nearest multiple of $2^{j'_{\max}}$ where $j'_{\max}$ is the desired maximum number of levels for subband tree decomposition.

The second limitation is the measurement range. Recall from chapter3 that, the maximum range for displacement estimation is $0.5 \times 2^{j_{\max}}$ i.e. the algorithm can not see motions bigger than $0.5 \times 2^{j_{\max}}$ pels at original image resolution. In order to find an optimum value for $j_{\max}$, these two criterias will constitute upper and lower limits. To examine the effect of $j_{\max}$ on accuracy, $j_{\min}$ is fixed at 3 and a confidence threshold of 0.95 is used as found in section 4.3.1. The results for the tree sequences are as given in Table 4.2.

Discussion

The most important point observed from the results is that there is dramatically high improvement as the maximum number of levels increases from 3 to 4. This is because of the

Jmax	Jmin	T.Tree	D.Tree	R.Tree
3	3	11.49	9.51	11.61
4	3	6.82	8.53	9.45
5	3	6.74	8.10	8.99
6	3	6.50	7.89	8.74

Table 4.2. Mean error angles for tree sequences, $j_{\min}$ fixed to 3, using bilinear interpolation.

fact that $j_{\max}$ is equal to $j_{\min}$ where the multiresolution hierarchical approach is discarded i.e. image is processed in only one resolution resulting in high aperture affects. In this test, this condition causes obtaining relatively less number of confident estimates and detecting motions only smaller than 0.5×2^3. Consequently, this results in a relatively big mean error angle.

As the maximum number of levels, $j_{\max}$, is increased from 4 to 6, the accuracy also increases with decreasing improvements. Since we are dealing with continuous motion fields, the optimum value of $j_{\max}$ is the biggest possible value we can set, according to constraint in equation (4.5). This upper limit can be extended to any desired value with boundary value replication with increased edge effects. After overcoming this limitation, $j_{\max}$ can be set to a high value so that the coarsest stage produces a field of resolution at least 4 by 4. But, it must be taken into account that boundary value replication increases edge effects in the image. Also the complexity of the algorithm increases as $j_{\max}$ increases. Consequently, according to the results and algorithm complexity conditions, 5 seems to be an optimum value for $j_{\max}$.

Note that, the main benefit of setting $j_{\max}$ to a high value comes in addressing the aperture problem, and more generally in the propagation of high-confidence vectors from the edges of moving objects into their low-activity interiors. There is also the benefit of greater robustness resulting from larger contributing regions at the coarsest level.

4.3.3 Choice Of Finest Level $j_{\min}$

After analyzing the optimum value of $j_{\max}$ by setting $j_{\min}$ to a constant value, we will now find an optimum value for $j_{\min}$ by setting $j_{\max}$ to a suitable value. The algorithm halts at level $j_{\min}$. If $j_{\min} \rangle 1$, the algorithm does not process motion estimation for finer levels and the output estimate will be the $j_{\min}$ times scaled and interpolated version of the estimate at level $j_{\min}$. By this way, the finest level of detail is effectively excluded from the estimation process. The number of subpels quadruples at every level, so the number of operations required for estimation increases geometrically as $j_{\min}$ decreases. The aim of this test is to determine whether the computation required to incorporate successively more detail provides sufficient benefit to accept this complexity.

In the test, confidence threshold is set to 0.95 and bilinear interpolation is used between levels. Then, $j_{\max}$ is set to 5 and an optimum value of $j_{\min}$ is investigated through tree sequences for values ranging from 4 to 1. The results are listed in Table 4.3.

Discussion

Analyzing Table 4.3, it can be seen that decreasing $j_{\min}$ down to 2 gives a high improvement in all cases. Setting $j_{\min}$ to 1 gives little improvement in rotating tree sequence. But there is a decrease in accuracy for diverging and rotating tree sequences.

Since the complexity of the algorithm is nearly quadrupled by the inclusion of finest level of detail, a reduced version of the hierarchical algorithm may be used which uses the reduced CDWT tree and skips over the level 1 processing. It is obvious from the results that using a reduced version of the hierarchical algorithm will give the best tradeoff between complexity and accuracy.

4.3.4 Field Interpolation Type

Up to now, we have analyzed optimal values for confidence threshold, maximum number of levels and finest level of detail. In this test, the effect of using staircase and bilinear interpolation kernels on accuracy will be analyzed using the results obtained before. For this purpose an algorith is used with a confidence threshold of 0.95, a $j_{\max}$ of 5 and a $j_{\min}$ of 2. The results are given in Table 4.4.

Jmin	Jmax	T.Tree	D.Tree	R.Tree
4	5	6.83	9.70	9.73
3	5	6.74	8.10	8.99
2	5	6.74	7.44	8.87
1	5	6.73	7.61	8.96

Table 4.3. Mean error angles for tree sequences, $j_{\max}$ fixed to 5, using bilinear interpolation with confidence threshold set to 0.95.

Interpolation Type	T.Tree	D.Tree	R.Tree
Staircase	7.15	9.80	11.19
Bilinear	6.74	7.44	8.87

Table 4.4. Mean error angles of tree sequences using staircase and bilinear interpolation.

Discussion

According to the results in Table 4.4, it is obvious that bilinear interpolation is much more superior to staircase interpolation. This condition is most apparent in diverging and rotating tree sequences where true motion field is not uniform. In translating tree sequence, this condition is less apparent due to uniform true motion field. From now on, bilinear interpolation will be used in following tests.

4.3.5 Type Of Gabor Filters

In this test, use of 4.tap RI pair and 8-tap Gabor filters will be analyzed. For this purpose, the algorithm developed with options and parameters up to here is used. This means that an algorithm with $j_{\max}$=5, $j_{\min}$=2, confidence threshold=0.95 is used with bilinear interpolation. The results obtained for 4-tap RI pair and 8-tap Gabor filters are given in Table 4.5.

Discussion

In all tree sequences, using 8-tap Gabor filters shows high improvements with respect to use of 4-tap RI pair. The main reason for this is the number of coefficients in 8-tap Gabor filters. Since they have more coefficients, they are good approximations to ideal Gabor

filters. Consequently, using 8-tap Gabor filters will highly increase accuracy of the optimal algorithm.

Filter Pair	T.Tree	D.Tree	R.Tree
4-tap RI	6.74	7.44	8.87
8-tap Gabor	5.91	6.67	7.92

Table 4.5. Mean error angles for 4-tap RI pair and 8-tap Gabor filters with $j_{\max}$=5, $j_{\min}$=2, confidence threshold = 0.95, using bilinear interpolation.

4.3.6 Curvature Correction

The aim of this test is to analyze curvature correction option (section 3.2.2.4) on accuracy of ME algorithm. In order to analyze effect of curvature correction, algorithm is processed with a wide range of e_t values. Recall from section 3.2.2.4 that, e_t is the eccentricity threshold above which ellipses of confidence become greatly elongated after correction is applied. The confidence ellipses of edge-based estimates of all orientations should all be so transformed, while those of estimates should be left relatively unchanged. The algorithm options and parameters developed in tests up to now is again used. The results are listed in Table 4.6 and mean error angles are plotted against e_t in Figure 4.4

Note that, since 8-tap Gabor filters' output eccentricities differs with edge orientation [19], applying curvature correction to these filters is meaningless and gives worse results as e_t value is decreased from $+\infty$. Figure 4.5 illustrates application of curvature correction to translating tree sequence for 8-tap Gabor filters.

Discussion

From the test results obtained in Table 4.6, it is clearly seen that curvature correction has great effect on increasing accuracy. In Figure 4.4, as the eccentricity threshold decreases from $+\infty$ to 3.5, accuracy of the ME algorithm increases in all tree sequences. But if eccentricity threshold continues to decrease, the mean error angle dramatically increases. This is due to changing the curvature of the aperture free estimates with aperture-affected ones. In all tests, it has been found that 3.5 is the optimum value for eccentricity threshold e_t using 4-tap RI pair. From Figure 4.5, it can be seen that applying curvature correction for 8-tap Gabor pair reduces accuracy as expected from the theory.

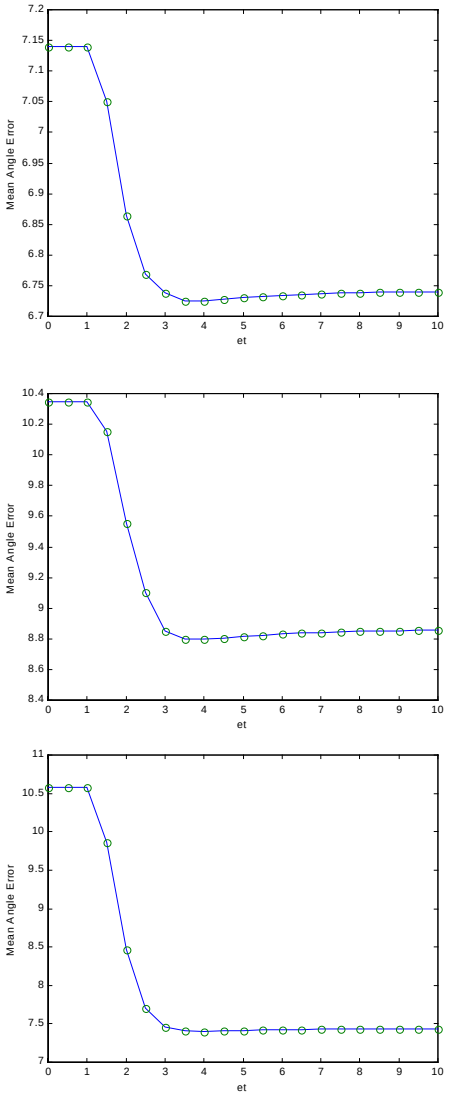


Figure 4.4 Mean error angle plots of tree sequences for various e_t values.
(a) Translating tree (b) Diverging Tree
(c) Rotating tree

et	T.Tree	D.Tree	R.Tree
0	7.141	10.584	10.346
0.5	7.141	10.584	10.346
1	7.141	10.584	10.346
1.5	7.050	9.867	10.152
2	6.864	8.468	9.556
2.5	6.769	7.706	9.104
3	6.737	7.458	8.849
3.5	6.725	7.411	8.801
4	6.725	7.404	8.799
4.5	6.728	7.409	8.807
5	6.731	7.415	8.816
5.5	6.733	7.420	8.824
6	6.734	7.423	8.832
6.5	6.736	7.426	8.837
7	6.737	7.428	8.842
7.5	6.738	7.430	8.846
8	6.739	7.432	8.849
8.5	6.740	7.433	8.852
9	6.740	7.434	8.854
9.5	6.741	7.435	8.856
10	6.741	7.436	8.858

Table 4.6. Mean error angles of tree sequences for various e_t values.

Up to now, we have two different optimum algorithms for ME. The first algorithm uses 4-tap RI pair with curvature correction and the second algorithm uses 8-tap Gabor filters. 4-tap RI pair requires less than half the number of operations required by the 8-tap pair to implement the CDWT. The actual computational saving is even greater than this because the

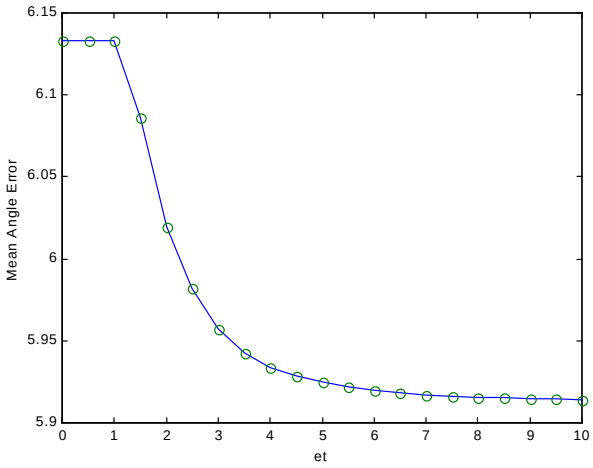


Figure 4.5 Curvature correction to translating tree sequence using 8-tap Gabor filters

RI pair has rational coefficients, so the entire tree may be implemented using integer operations. On the other side, 8-tap pair gives better results in terms of accuracy. From now on, one have to consider the accuracy gain achieved and complexity increased. Our choice is the first algorithm, which uses 4-tap RI pair with curvature correction.

Figure 4.6 illustrates estimated motion fields of tree and Yosemite sequences at 8-pel resolution where the final algorithm is used. Also, mean error angle and MSE are calculated for four sequences using the final algorithm and corresponding softwares for pel-recursive and half-pel BM algorithms [25]. The results are given in Table 4.7 and Table 4.8.

4.3.7 Illuminance Variation

In this test, the current frame of the translating tree sequence was subjected to two separate uniform perturbations of intensity: addition of an offset and scaling. The aim is to simulate simple cases of illuminance variation between the frames of a sequence. The offset test is done for offset values ranging from 0 to 8 and the scaling test is done for scale values ranging from 0.95 to 1.04. Figure 4.7 illustrates results for translating tree sequence. Also, the results for pel-recursive and half-pel BM algorithms that are obtained by using the corresponding softwares [25] are discussed in discussion section.

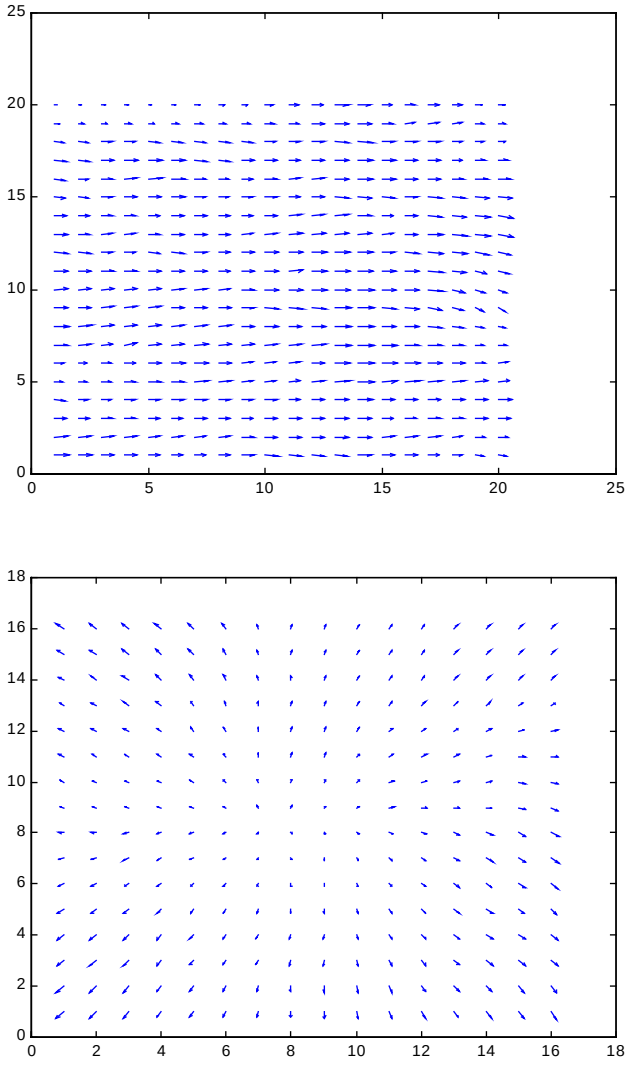
Discussion

In each test, it can be clearly seen that CDWT based ME algorithm is perfectly immune. This is because of the fact that phase of the CDWT coefficients is independent from both kinds of perturbation. Considering results of pel-recursive and half-pel BM algorithms

obtained from corresponding softwares, a strong robustness of CDWT based ME algorithm can be clearly seen. Pel-recursive and half-pel BM algorithms have mean error angles of 10.12 and 11.71 respectively for translating tree sequence and have mean error angle deviations against illuminance variations up to 1218 and 1711 percent respectively. But CDWT based ME algorithm is invariant to simple illumination changes.

4.3.8 Noise Immunity

In this test, white Gaussian noise of varying variances is added to diverging tree sequence in order to analyze robustness of the algorithm against noise. Also, the results for Pel-recursive and half-pel BM algorithms that are obtained by using the corresponding softwares [25] are discussed in discussion section. Figure 4.8 shows the results.



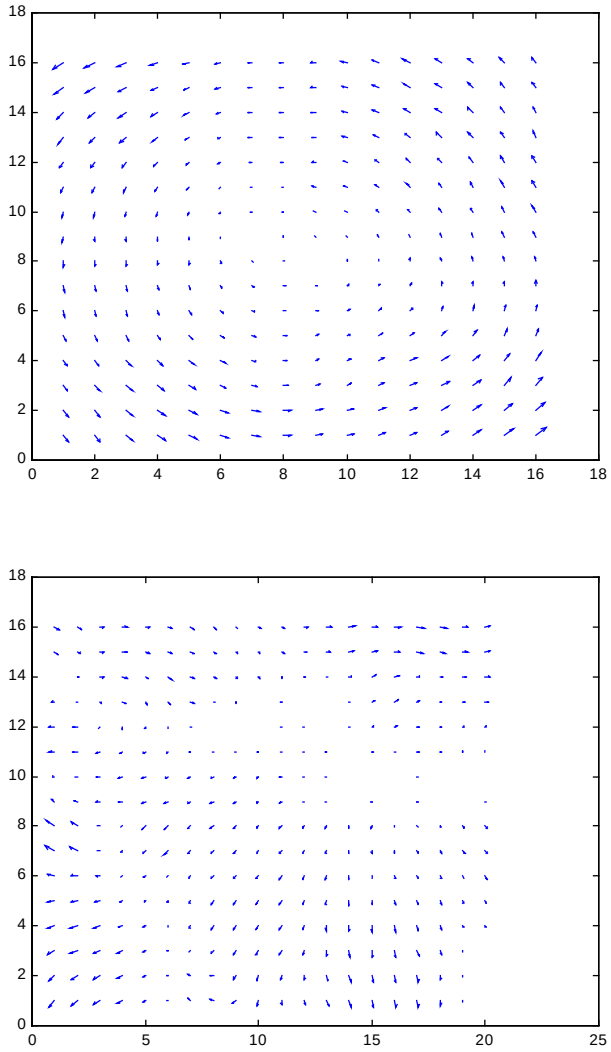


Figure 4.6 Estimated motion fields of tree and Yosemite sequences at 8-pel resolution with $ConfidenceThreshold = 0.95$, $j_{\max} = 5$, $j_{\min} = 2$, using 4-tap RI pair, bilinear interpolation with curvature correction

Algorithm	T.Tree	D.Tree	R.Tree	Yosemite
CDWT	6.725	7.411	8.801	12.612
Half-pel BM	9.923	11.122	14.712	15.614
Pel-recursive	10.678	10.983	14.118	16.219

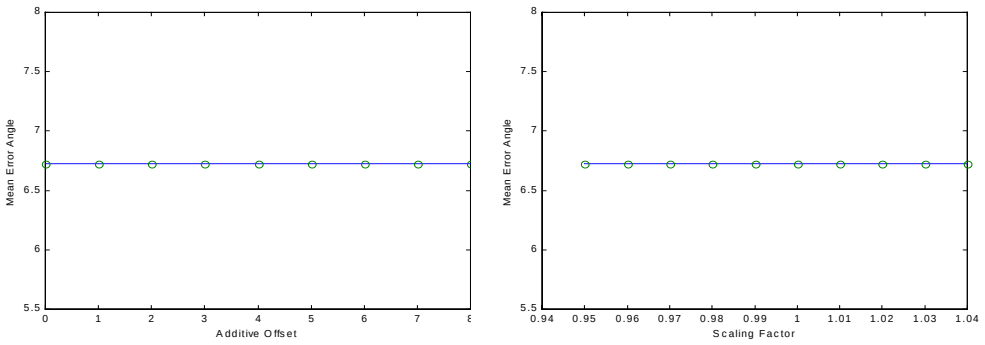
Table 4.7. Mean error angle of tree and Yosemite sequences

Algorithm	T.Tree	D.Tree	R.Tree	Yosemite
CDWT	55.18	67.83	87.32	141.67
Half-pel BM	101.14	118.49	135.14	158.14
Pel-recursive	111.91	119.90	132.21	144.19

Table 4.8. MSE of tree and Yosemite sequences

Discussion

The CDWT algorithm is more robust to additive noise when compared with pel-recursive and half-pel BM algorithms. Note that when variance is increased to 0.08, the increase in mean error angle is about 59 percent where using pel-recursive and half-pel BM algorithms with corresponding softwares [25] give results about 310 and 550 percent respectively.



(a) (b)

Figure 4.7 Mean error angles against uniform intensity perturbations (a)Additive offset (b) Scaling

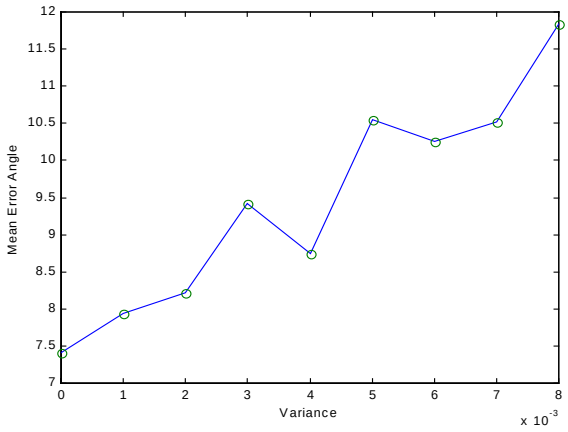


Figure 4.8 Mean error angle under addition of white Gaussian noise

4.4 Summary

This chapter has described various tests applied to CDWT ME algorithm in order to find effect of options and parameters on accuracy. Tests are carried out on synthetic sequences with known true motion fields. Use of these sequences enables measurement of accuracy in terms of a defined error measure. To measure errors, mean error angle measure is used instead of relative error due to obtaining better results on relatively small valued estimates.

The starting point of the tests is finding an optimum value for confidence threshold. A value of 0.95 is found, where for a lower value, the number of false estimates increases and for a higher value, the number of rejected confident estimates increases. After finding confidence threshold, the number of maximum levels $j_{\max}$ is found to be as big as possible for a better accuracy. Then, the optimum value of finest level of detail $j_{\min}$ is found to be 2, where a value of 1 does not provide desired accuracy in all sequences when compared with added algorithm complexity. After finding the main parameters of the ME algorithm, the effect of staircase and bilinear interpolation between levels is analyzed and found that bilinear interpolation is superior to staircase interpolation. Then, for a better tree decomposition, the effect of using 8-tap Gabor filters is analyzed and found that 8-tap Gabor filters give better results with respect to 4-tap RI pair. In the last test, curvature correction option is analyzed and an optimum value of 3.5 is found for eccentricity threshold e_t using 4-tap RI pair. Then two algorithms, which use 4-tap RI pair with curvature correction and 8-tap pair, are discussed in terms of accuracy and algorithm complexity. Consequently, the algorithm using 4-tap RI pair with curvature correction is chosen. Moreover, the CDWT based motion estimation algorithm is compared with two popular motion estimation algorithms in terms of MSE and mean error angle measures. Finally, the robustness of algorithm against illumination changes and noise is analyzed.

CHAPTER 5

CONCLUSIONS & FUTURE WORK

As can be seen from the test results in previous chapter, CDWT based motion estimation algorithm gives better results according to other most challenging algorithms. In this chapter, a comparison of the proposed algorithm according to other popular algorithms and a list of future works will be given.

5.1 CONCLUSIONS

From the test results in previous chapter, CDWT based ME algorithm is constructed with parameters and options as confidence threshold = 0.95, $j_{\max} = 5$, $j_{\min} = 2$, using 4-tap RI pair with bilinear interpolation and curvature correction with an eccentricity threshold of $e_t = 3.5$. From the results given in Table 4.7 and Table 4.8, it can be clearly seen that CDWT based motion estimation gives relatively more robust estimates for tree sequences according to half-pel BM and pel-recursive algorithms. But this robustness is weak in Yosemite sequence, where motion field is discontinuous. In order to get robust estimates for stereo image pairs with discontinuous motion fields, further investigations can be done.

In illuminance variation and noise immunity tests, it is clearly seen that using the phase information gives robustness to CDWT based ME algorithm especially in simple illuminance variation conditions. The algorithm gives much better results according to other popular algorithms in case of additive noise and is perfectly immune against simple illuminance variation conditions. This property is important especially for real life conditions where illuminance variance is a commonly encountered condition for real digital image sequences.

The proposed algorithm's computational complexity is also much more less according to other popular algorithms. The algorithm with our choice of options and parameters has a computational complexity of 234 flops per image pel where half-pel BM and pel-recursive algorithms have 584 and 609 flops per image pel, respectively. With this relatively less computational complexity, the proposed algorithm is suitable for real time motion estimation applications.

Consequently, the CDWT based motion estimation algorithm is superior to other challenging algorithms in terms of accuracy, robustness and computational complexity. But further improvements can be done in order to have more robust estimates for discontinuous motion fields.

5.2 FUTURE WORK

For future work, some extra options that increase accuracy under various conditions may be listed. The first is the use of iterative approaches. Iterative refinement strategy includes warping of current level wavelet coefficients according to the estimate done previously in that level. This procedure can be iteratively applied for desired times to increase accuracy. Another iterative approach is the refining strategy, where previous level estimates are used to refine current level estimates in terms of warping current level coefficients. Both iterative options increase accuracy under various input image sequences. But they may give worse accurate results in some cases where motion at coarsest level (starting point estimate) is not estimated as accurate as required. Another option is using variable local frequency instead of predefined constant center frequencies for all levels. The last option to increase accuracy is the inclusion of an external regularization field to all level estimates with use of the directional confidence measure. This approach increases accuracy by smoothing untextured regions while preserving sharp features.

Moreover, this algorithm can be modified to process colored images and it can be used to extract 3d structures in scenes.

REFERENCES

[1] J. Magarey and N. Kingsbury. Motion estimation using a complex valued wavelet transform. IEEE Transactions on Signal Processing, Vol. 46, No. 4, April 1998

[2] N. Kingsbury and J. Magarey. Wavelets in image analysis: Motion and displacement estimation. In Proceedings of the Irish Digital Signal Processing and Control Conference, Dublin, Eire, June 1996.

[3] J. Magarey and N. Kingsbury. Robust motion estimation using complex wavelets. Signal Processing and Communications Laboratory Cambridge University Engineering Department Trumpington Street, Cambridge U.K. CB2 1PZ, 1995.

[4] G. Castellano, J. Boyce and M. Sandler. Moving target detection in infrared imagery using a regularized CDWT optical flow. King's College London Strand, London WC2R 2LS, U.K. 1999

[5] M.K.Mandal, E. Chan, X. Wang and S. Panchanathan. Multiresolution motion estimation techniques for video compression. Visual Computing and Comm. Lab. Department of EE University of Ottawa, Ottawa Ontario, Canada – K1N 6N5, January 1996.

[6] J.Magarey and A. Dick. Multiresolution stereo image matching using complex wavelets. Cooperative research centre for sensor signal and information processing SPRI Building, Technology Park Adelaide, SA 5095, Australia, 1998.

[7] J. Magarey and N. Kingsbury. An improved motion estimation algorithm using complex wavelets. Signal processing and comm. Lab. Cambridge university engineering department, Cambridge, U.K., 1996.

[8] S.S. Beauchemin and J.L. Barron. The computation of optical flow. Dept. Of computer science, University of Western Ontario, London, Ontario, Canada N6A 5B7, 1995.

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

[10] A. Dick. Multiresolution stereo image matching using complex wavelets. Computer Science University of Adelaide, November 1997.

[11] S.Mallat. A theory for multiresolution signal decomposition: the wavelet representation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 11(7):674-693, July 1989.

[12] B. K. P. Horn and B. G. Schunck. Determining optical flow. Artificial Intelligence, 17:185-203, 1981.

[13] B. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In Proceedings of the Image Understanding Workshop, DARPA, pages 121-130, 1981.

[14] P. Nesi, A. Bimbo, and D. Ben-Tzvi. A robust algorithm for optical flow estimation. Computer Vision and Image Understanding, 62(1):59-68, July 1995.

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

[16] F. Dufaux and F. Moscheni. Motion estimation techniques for digital TV: a review and a new contribution. Proceedings of the IEEE 83(6):858-876, 1995.

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

[18] D.Gabor. Theory of communication. Journal of the IEE, 93(26):429-457, November 1946.

[19] J. Magarey. Motion estimation using complex wavelets. PhD thesis, Department of Engineering, University of Cambridge, February 1997.

[20] O. Rioul and M. Vetterli. Wavelets and signal processing. IEEE Signal Processing Magazine, pages 14-38, October 1991.

[21] C. Cafforio and F. Rocca. Methods for measuring small displacements of television images. IEEE Trans. on Information Theory, IT-22(5):573-579 September 1976.

[22] L. Daubechies. Ten lectures on wavelets. Society for Industrial and Applied Mathematics, 1992.

[23]R.Jain, R.Kasturi, B.G.Schunck. Machine vision. Book, McGraw-Hill ISBN:0070320187, 1995.

[24] A.K. Tekalp. Digital video processing. Prentica-Hall, 1995.

[25] http://peipa.essex.ac.uk/ipa/src/vision/optical-flow