X1, 129
#4618

UNIVERSITY OF HAWAII LIBRARY

MOTION ESTIMATION IN THE 3-D GABOR DOMAIN

A DISSERTATION SUBMITTED TO THE GRADUATE DIVISION OF THE
UNIVERSITY OF HAWAII IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

IN

ELECTRICAL ENGINEERING

AUGUST 2005

By
Mu Feng

Dissertation Committee:

Todd R. Reed, Chairperson
N.T. Gaarder
Anthony Kuh
Anders Host-Madsen
Nancy Reed

Copyright 2005

by

Mu Feng

iii

To my wife,

Jun Luo,

my father and mother,

Xiaokang Feng, Xiuqin Liu,

my father-in-law and mother-in-law,

Enbao Luo, Kezhen Zhao,

and my daughter,

Xingchen Feng,

for their love, support and encouragement.

Acknowledgements

I would like to express my deepest gratitude to my advisor, Professor Todd R. Reed. His enlightening advising, patient instruction and careful inspection are essential to the fulfillment of this dissertation. It is my great luck to have such an excellent mentor who dedicated so much time and energy to help me with the problems in the research. Without the guide of his insightful ideas and penetrating explanations, the accomplishment of this research would be impossible.

I also want to sincerely appreciate my committee members: N.T.Gaarder, Anthony Kuh, Anders Host-Madsen and Nancy Reed. They dedicated their precious time to examine the draft. The invaluable comments came from them are significant for the academic completeness of this dissertation.

The helps from my classmates and research group colleagues are also acknowledged. The discussions with Xin Chen, Jianhan Liu, Jianhua Lu, Juntan Zhang, Wenyi Jin, Zigui Yang and XianGang Li are very constructive for my steps ahead.

During these years, I am always supported by the love, understanding and encouragement from my wife, Jun Luo, my parents Xiaokang Feng, Xiuqin Liu and my parent-in-law, Enbao Luo, Kezhen Zhao. To dedicate some words here is far from expressing my gratitude. In fact, they are one of the main reasons that this dissertation is possible.

Finally, I want to take the chance to thank the University of Hawaii and all the faculties and staffs in the department of Electrical Engineering. As one of the students that have been surfing with her superior academic services, talent instructors and efficient managements, I am pleased and proud of that I can dedicate my dissertation to the ocean of her scholarly contributions.

ABSTRACT

This thesis presents a systematic study of motion estimation in the 3-D Gabor domain. The 3-D Gabor representation provides a joint spatiotemporal/spatiotemporal-frequency(st/stf) analysis framework to disclose the spatiotemporal localized frequency information in an image sequence in a manageable information structure. It offers the opportunity to take advantages of frequency domain approaches with spatiotemporal locality. The main contribution of this thesis is to realize an innovative motion estimation technique achieving high spatiotemporal resolution with the superior reliability of the model-based frequency domain approaches. Compared with the global nature of frequency domain approaches and the noise sensitive spatiotemporal approaches, the scheme enhances motion estimation performance both in accuracy and robustness. The mathematical forms of translational motion and rotational motion in the 3-D Gabor domain are derived. For the translational motion, an algorithm for piecewise uniform translational motion is designed. Various piecewise cases can be addressed by setting different Gabor representation parameters. For applications demanding very high spatiotemporal resolutions, a dense motion field estimation method is constructed in the 3-D Gabor domain. The performance evaluation based on artificial image sequences shows high spatiotemporal resolution motion estimation results with small errors and good noise resistance. In rotational motion studies, the 3-D Gabor representation also leads to an algorithm to identify and locate multiple rotating objects in an image sequence.

Contents

Acknowledgements	v
Abstract	vi
List of Tables	ix
List of Figures	x
1 Introduction	1
1.1 Motion Estimation, a Review	1
1.1.1 Motion estimation and Its Applications	1
1.1.2 Motion Representations	3
1.1.3 The Spatiotemporal Domain Approach	5
1.1.4 The Frequency Domain Approach	9
1.1.5 The Joint Spatiotemporal/Spatiotemporal-frequency(st/stf) Domain Approach	13
1.2 Organization	16
1.3 Notation	17
2 The 3-D Gabor Representation	19
2.1 Representation of Signals	19
2.2 Frame for Signal Representation	21
2.3 1-D Gabor Representation	23
2.3.1 The Gabor Basis Functions	23
2.3.2 The Continuous Gabor Representation	24
2.3.3 The Discrete Gabor Representation	26
2.3.4 Coefficient Calculation	26
2.4 The 3-D Gabor Representation	27

3	Motion Estimation in the 3-D Gabor Domain	35
3.1	The Translational Motion Model in the 3-D Gabor Domain	36
3.2	The Piecewise Uniform Translational Motion Estimation in the 3-D Gabor Domain	39
3.3	Performance Analysis of Motion Estimation in the 3-D Gabor Domain . . .	42
3.3.1	Image Sequences for Experiments	42
3.3.2	Performance Calibration	45
3.3.3	Performance Analysis for Experimental Results	47
3.4	Conclusion	59
4	Dense Motion Field Estimation in the 3-D Gabor Domain	62
4.1	Motivation	62
4.2	Partial Reconstruction	64
4.3	Calculation of the Dense Motion Field	65
4.4	Performance Analysis of Dense Motion Field Estimation Method	66
4.4.1	Image Sequences for Experiments	66
4.4.2	Performance Calibration	67
4.5	Performance Analysis for Experimental Results	69
4.6	Conclusion	72
5	Detection and Estimation of Rotational Motions in the 3-D Gabor Domain	76
5.1	Motivation	76
5.2	The 3-D Gabor Representation in the Polar Coordinates	77
5.3	Rotational Motion Model in the 3-D Gabor Domain	79
5.4	Detection and Estimation of the Localized Uniform Rotational Motion in the 3-D Gabor Domain	81
5.5	Conclusion	83
A	Derivation of the Translational Motion Model in the Discrete 3-D Gabor domain	88
B	Derivation of the Rotational Motion Model in the 3-D Gabor Domain	91
C	Source Code of the Algorithms	93
C.1	Generate the Over Sampling 3-D Gabor Auxiliary Function Matrix	93
C.2	The 3-D Gabor Expansion with Over Sampling	101
C.3	Motion Estimation in the 3-D Gabor Domain	106
C.3.1	Arrangment of Local Frequency Information	106
C.3.2	Motion Estimation and Motion Field Displaying (Matlab Version)	111
C.4	The Dense Motion Field Estimation	113
	Bibliography	125

List of Tables

3.1	Performance of Gabor scheme A for the “two-square” sequence.	50
3.2	Performance of Gabor scheme B for the “two-square” sequence	50
3.3	Performance of Gabor scheme B for the “wheel” sequence	54
3.4	Performance of Gabor scheme C for the “wheel” sequence	54
3.5	Performance of Gabor scheme C for the “three-wheel” sequence	59
4.1	Performance evaluation at different location	72
4.2	Performance evaluation with noise, $\sigma = 36$	72
4.3	Performance evaluation with noise, $\sigma = 50$	72
4.4	Performance evaluation with noise $\sigma = 64$	73

List of Figures

2.1	Example of Gabor basis functions with $\alpha = 0.0884$	32
2.2	Example of Gabor basis functions with $\alpha = 0.1788$	33
2.3	Example of inappropriate designing of the Gabor representation	34
3.1	A 1-D Gabor auxiliary function for the critical sampling case	38
3.2	A 1-D Gabor auxiliary function with over sampling	39
3.3	Two-square image sequence and its true motion Field	43
3.4	“Wheel” image sequence and its true motion field.	44
3.5	“Three-Wheel” sequence and its true motion field.	45
3.6	A : Gabor basis function coverage along one dimension of the spatial and spatial frequency domains under critical sampling with $\alpha = 0.125$, $\Delta l = 8$	47
3.7	B : Gabor basis function coverage along one dimension of the spatial and spatial frequency domains under critical sampling with $\alpha = 0.0625$, $\Delta l = 16$	48
3.8	Motion estimation results and errors for Gabor scheme A for the “two-square” sequence	49
3.9	Motion estimation results and errors on Gabor scheme B for the “two-square” sequence	49
3.10	Gaussian noise contaminated “two-square” image sequence.	51
3.11	Motion estimation results and errors for Gabor Scheme A for noised “two-squared” sequence	52
3.12	Motion estimation results and errors for Gabor Scheme B for noised “two-square” sequence	53
3.13	C : Gabor basis function coverage along one dimension of the spatial and spatial frequency domains under spatial over sampling with $\alpha = 0.0625$, $\Delta l = 8$	54
3.14	Motion estimation results and errors for Gabor Scheme B and the “wheel” sequence	55
3.15	Motion estimation results and errors for Gabor Scheme C and the “wheel” sequence	55
3.16	Gaussian noise contaminated “wheel” image sequence	56

3.17	Motion estimation results and errors for Gabor Scheme B and the noisy “wheel” sequence	57
3.18	Motion estimation results and errors for Gabor Scheme C and the noisy “wheel” sequence	58
3.19	Motion estimation results and errors for Gabor Scheme C and the “three-wheel” sequence.	59
3.20	Gaussian noise contaminated “three-wheel” image sequence.	60
3.21	Motion estimation results and errors for Gabor Scheme C and the noisy “three-wheel” sequence.	61
4.1	Dense motion field estimation for translational triangles sequence	70
4.2	Artificial Image Sequences of the Rotating Wheels.	74
4.3	Dense Motion Field Estimation of rotating wheels.	75
5.1	Magnitude of the Fourier transform of the auxiliary function in critical sampling and over sampling cases.	85
5.2	Temporal frequency distribution associated with two rotational objects.	86
5.3	Experiment image sequence and the detected rotational regions.	87
A.1	Waveform of a baseband Gabor auxiliary function in a span of one local temporal region	90

Chapter 1

Introduction

1.1 Motion Estimation, a Review

1.1.1 Motion estimation and Its Applications

One major effect that image sequences intend to present is motion. When objects in the scene and/or the camera are in 3-D motion, their relative movements are projected into 2-D motions by the camera projection system and recorded in a sequence of time ordered images. The image sequence can be expressed as a 3-D intensity function $I(x, y, t)$ where the (x, y) and t are spatial and temporal coordinates respectively. The 2-D motions, if significant, may dominate the varying patterns of the 3-D function along the temporal axis. The aim of motion estimation is to detect the motions from the noisy world of spatiotemporal variations and express in an explicit motion representation for the image sequences.

Motion estimation is a very active research field because of its correlation with human visual perceptions of the living world. As a fundamental technology demanded in most dynamic image signal processing tasks, motion estimation supports numerous applications[1].

In multimedia communications, video compression is necessary for efficient signal transmission and storage. Correct motion information is a very reliable reference for inter-frame prediction. By identifying “corresponding data blocks” along the video stream[2] or some other representation of video stream[3], motion estimation reveals the varying regularity of scenes, which is required for compact coding. Hence, motion estimation has been intergraded into H.261, H.263, and MPEG standards as a significant component. Moreover, people are also interested in the compact representation of video signals by segmentation and registration of moving objects, which leads to a relatively small number of parameters to describe the changes. That technology also relies heavily on motion estimation[4].

In computer vision, pattern recognition and edge detection are often needed for scene interpretation, dynamic segmentation and image registration. Integrated with motion estimation, these methods can be more efficient and accurate. For example, [5] introduces an iterative dynamic segmentation scheme using “highest confidence first” and “iterated condition mode” with motion information. In [6], an interactive processing between motion estimation and edge identification is presented. Under some circumstance, pattern recognition, segmentation or image registration can even be done directly in the motion field[7].

Motion estimation is also significant for many other video signal processing jobs. In noise suppression, motion estimation indicates intensity variations caused by moving objects whereas noise has no explicit motion model in the long term. Thus, the acquired motion field is a meaningful reference for denoising[8]. In this case, the motion estimation method should have dependable noise resistance ability itself. In video signal restoration such as deblurring and high quality frame-freezing, correct motion information is crucial in constructing the deblurring filter for good visual effect. In resolution enhancement tasks and frame rate conversion between different video systems, motion information helps the

spatiotemporal interpolation to keep or enhance the regularity of intensity changes, that prevent low-pass effects during up-sampling[9].

Given the rapidly growing demands in multimedia communication, entertainment, machine vision, and many other applications, motion estimation will continue to be a significant research field that enhances the ability to acquire, understand and manipulate dynamic visual information.

1.1.2 Motion Representations

Generally speaking, a motion representation is a vector-valued spatiotemporal function. The vector elements are a set of parameters to characterize certain motion models. Motions are classified into different models such as translational motion, rotational motion, and so on. Each model has a specific mathematical signature defined by a set of parameters. If the motion is not global, the whole image sequence is divided artificially or adaptively into many regions to isolate different motion models. The spatiotemporal division (spatiotemporal support of a representation unit) can be object-based, block-based or pixel-based. In each represented unit, some searching methods are implemented to match the signatures of the assumed motion models under some criteria.

The motion representation model is highly application oriented. For example, a detailed, object-based motion representation model provides valuable reference in dynamic segmentation or pattern recognition. Whereas its extra computational and coding burdens for segmentation and edges registration is far from economical for video compression tasks in which only the intensity correlation structure is needed. On the other hand, a compact, field-based representation model is demanded in video compression to indicate the redundancy along the motion trajectories, but may not fine enough for analysis purposes. For some special applications such as flow analysis, spatiotemporal interpolation, or resolution enhancement, the desired motion representation may be so fine that the motion of every

pixel or sub-pixel is displayed. Accordingly, different motion estimation stratagems are designed for different representation models.

The spatiotemporal partition and motion model together determine the motion estimation performance. Intuitively, the more degrees-of-freedom a motion model embodies, the higher accuracy and descriptive ability it may have for complex motions. However, its excessive degrees-of-freedom involve greatly increased computational burdens and coding complexity. In addition, the chances of multiple (conflicting) solutions in estimation also rise. On the other hand, simple motion models (such as uniform translational motion) can only approximate the “true” motion field in small regions. The corresponding spatiotemporal partition should be fine enough to ensure the approximation quality. In general, the smaller the local spatiotemporal regions, the simpler the motion can be used to estimate motion in the regions with a given degree of accuracy. However, dense partitions may also decrease representation efficiency and increase the complexity of the processing which follows. Object-based spatiotemporal partitions, as an adaptive approach, are also of interest. With some additional cost due to segmentation and edge registration[5], the motion information in an image sequence can be expressed in compact and functional ways. The designing of the motion representation models for specific applications is quite an engineering art and is of crucial importance for the estimation performance.

The motion estimation algorithm is not necessarily implemented in the same partition model as that of the desired motion representation. For example, motion estimation algorithms developed in hierarchical representations[10] realized a variable-size block partition to determine the motion representation in a “from coarse to fine” manner. The variable-size blocks promote further estimation only in important regions to enhance computational efficiency as well as spatiotemporal resolution. Another approach with even smaller calculation labor is to interpolate the motion vectors among estimated ones[11][12]. The interpolation methods are designed according to motion distribution, with high precision

around edges and coarse (hence compact) in background region. As the spatial correlation of the motion field is often very high, algorithms may boost the spatiotemporal resolution with only slightly increased computation.

The motion models, spatiotemporal localization and search methods are three significant factors for motion estimation. In practices, the three factors must operate in an analysis framework, namely, a domain. As a representation of a signal, the analysis framework exhibits the characteristics of the signal in a specific information structure. This structure determines the mathematical signature of a motion model, the possible spatiotemporal resolution and the style of the search algorithms. For different motion representation requirements, numerous approaches are under investigation[1]. In the following sections, a brief review is presented.

1.1.3 The Spatiotemporal Domain Approach

Since human perception of visual motion is naturally associated with space and time, collecting spatiotemporal characteristics such as the intensity gradient and feature structure for motion estimation has been extensively researched[13]. Two branches can be roughly concluded from these approaches.

One branch includes optical flow calculation[14], recursive region matching[15] and feature searching. The aim is to find out the intensity correspondence between two frames under some soft constraint between motions and the temporal intensity variation. Incorporated with mature matching methods, they produce results with good correlation to our visual experience. Although the motion model is simply uniform translation, the spatiotemporal resolution can be as high as pixel or subpixel[16]. High spatiotemporal resolution boosts the representation's ability to express complex motions and shapes. However, if the intensity changes are not mainly because of motion (due to noise, high spatial frequency or illumination and so on), some performance degradation will result.

Computing Optical Flow

The conception of optical flow came from the observation that apparent 2-D motion is due to the intensity changes between frames in an image sequence. Assuming the surface brightness of objects and the background illumination vary slowly, all the spatiotemporal intensity changes are caused by motion. Denote $I(x, y, t)$ is the intensity of a moving pixel at coordinates (x, y, t) in an image sequence. After a small time, it moves to $(x + dx, y + dy, t + dt)$. By Taylor expansion, we have:

$$I(x + dx, y + dy, t + dt) = I(x, y, t) + \frac{\partial I}{\partial x}dx + \frac{\partial I}{\partial y}dy + \frac{\partial I}{\partial t}dt + \dots \quad (1.1.1)$$

The dots are higher order terms. If the time interval dt and the spatial displacement (dx, dy) are small, they can be omitted. Since the intensity of the pixel is not changed by moving, we have:

$$I(x + dx, y + dy, t + dt) = I(x, y, t) \quad (1.1.2)$$

and

$$\frac{\partial I}{\partial x}dx + \frac{\partial I}{\partial y}dy + \frac{\partial I}{\partial t}dt = 0 \quad (1.1.3)$$

Divide the both side of the equation by dt , and denote $\frac{dx}{dt} = u$ and $\frac{dy}{dt} = v$ the speeds of the pixel in x and y directions respectively, the resulting partial derivative equation is established as the start point of almost all gradient based motion estimation methods:

$$\frac{\partial I}{\partial x}u + \frac{\partial I}{\partial y}v + \frac{\partial I}{\partial t} = 0 \quad (1.1.4)$$

The optical flow constraint equation introduces a spatiotemporal gradient domain which connects the speeds of intensity variations with the speeds of translational motions. Its spatiotemporal resolution is as high as for each pixel and at any instant.

However, because one equation is not enough to solve for the two unknowns u and v at a given t , other constrain functions are needed. In the pioneering work of Horn

and Schunk[14], a quadratic smoothness constraint is added. The basis of this constraint is the assumption that except for those pixels along object edges, the motion of adjacent pixel should be very similar. This “smoothness” velocity constraint can be evaluated as:

$$S = \int_x \int_y \left(\left(\frac{\partial u}{\partial x} \right)^2 + \left(\frac{\partial u}{\partial y} \right)^2 + \left(\frac{\partial v}{\partial x} \right)^2 + \left(\frac{\partial v}{\partial y} \right)^2 \right) dx dy \quad (1.1.5)$$

The smoother the velocity changes are, the smaller the S . Another famous constraint, also is the complement of the former one for the edge pixels, measures how the optical flow constraint is obeyed for all pixels:

$$R = \int_x \int_y \left(\frac{\partial I}{\partial x} u + \frac{\partial I}{\partial y} v + \frac{\partial I}{\partial t} \right)^2 dx dy \quad (1.1.6)$$

The better the optical flow constraint equation is followed, the smaller the R is.

The former two constraints can be combined in a Lagrangian multiplier $S + \lambda R$, whose minimization leads a unique solution of the optical flow equation (1.1.4). λ is a scalar to weight the R , the feasibility of the optical flow constraint. Heavy noise may make R hard to decrease, which implies the spatiotemporal intensity variations are more unlikely to follow equation (1.1.4). In this case, a small λ is used. A reasonable λ as well as the velocity (u, v) of any given pixel can be calculated in an iterative way.

Based on this approach, many other constraints accounting for different stationary conditions have been developed. To solve the edge blur problem due to the linear diffusive nature of the smoothness constraint, a curve evolution constraint was introduced to keep the flow-velocity across the edges[17]. If the illumination is not consistent, mathematic illumination models are introduced to balance the intensity change due to illumination[18]. For more complex cases in which the explicit illumination model is not available or keeps changing, the optical flow equation can be modified for spatial intensity gradient constancy instead of spatial intensity constancy. Furthermore, for color image sequences, the optical flow functions can be either employed on intensity alone or established in different primary color spaces. The components of these color spaces offer a degree of redundancy, which allows for the formation of new constraints.

Optical flow is a relatively mature branch in the motion estimation method family. If motion produces a sufficient intensity gradient in the image sequence, the optical flow is a good estimation for the motion field, and can be found with in very high spatiotemporal resolution. Moreover, application oriented constraints can be added according to different data conditions. Therefore, in high SNR conditions, the optical flow approach is quite practical. However, when noise contamination is not negligible, the local intensity change may be due to noise instead of motion. Thus, the optical flow equation no longer reflects true motion. In addition, the spatiotemporal gradient domain is very sensitive to noise, which reduces the reliability of the optical flow method dramatically in presence of noise.

The Spatiotemporal Model Approach

The other branch in spatiotemporal methods uses multiple frames for certain spatiotemporal models (to determine the parameters of the motion models). The models are established according to some “typical” motions and “typical” surface conditions and act as hard constraints for matching intensity regularities. If sophisticated searching methods are affordable, determining parameters from multiple frames offers the opportunities to remove temporal interference and enhance robustness. Thus, if the motion representation does not require arbitrarily complex motion and shapes with very high resolution, the model based approaches provides more reliable results. In [19], a parametric spatiotemporal method is proposed. The trajectories of moving objects are modelled as polynomial or polynomial trigonometric time functions. To determine the polynomial coefficients, each frame of the original image sequence is transformed into the 2-D Fourier domain. Then, the “frequency image” sequences are processed by a cascaded operator in which the frames correlated with each polynomial coefficient are accumulated together in a “processing channel”. In each “processing channel”, the corresponding coefficients are estimated as parameters incorporated in the mathematic signature of the channel. With the aid of the Fourier transform,

the multi-frame information is assembled efficiently and relations with model parameters are easily established. This leads us naturally to ask the question: Can we establish motion models directly in the frequency domain?

1.1.4 The Frequency Domain Approach

Frequency domain approaches to motion estimation have been of interest for a number of years[20][21]. Using the 3-D Fourier transform, the motion information embodied in an image sequence is transferred into certain distribution models of Fourier coefficients[22]. For some motion models, the distribution can be very concise.

One of the important practical frequency motion models is the piecewise polynomial translational motion model presented in [23][24]. In the spatiotemporal sense, polynomial translational motion is defined as:

$$\mathbf{x}(t) = \mathbf{x}(0) + \sum_{m=1}^M \frac{\mathbf{a}_m t^m}{m!} \quad (1.1.7)$$

where $\mathbf{x}(0)$ contains the initial coordinates of the object, and $\mathbf{x}(t)$ the coordinates of the object at time t . The displacement is dependent with time t in an M order polynomial relation with coefficients $\mathbf{a}_m$. Denoting the intensity of the object as $I(\mathbf{x}, t)$ in the spatiotemporal domain, its 3-D Fourier transform is:

$$\mathbf{I}(\omega, f) = \mathbf{I}_0(\omega) \bigotimes_{m=1}^M \mathbf{F}[\exp(-j2\pi \frac{\omega^T \mathbf{a}_m t^m}{m!})](f) \quad (1.1.8)$$

where, ω is the 2-D spatial frequency variable, $\mathbf{I}_0(\omega)$ is the 2-D Fourier representation of the first frame and f is temporal frequency variable. The $\bigotimes_{m=1}^M$ means the convolution of M items together and the $\mathbf{F}$ denotes the temporal (1-D) Fourier transform. Equation (1.1.8) shows that the polynomial translation model in the frequency domain is the convolution of a function sequence. Each function in the sequence is the Fourier transform of a complex exponential whose exponent is one of the polynomial terms in model (1.1.7). This motion

model fits any trajectory that is M order determinative over the time interval $[0, t]$. Specially, if $M = 1$, the frequency model can be simplified as:

$$I(\omega, f) = I_0(\omega)\delta(f + \omega^T \mathbf{V}) \quad (1.1.9)$$

where $\mathbf{V}$ is the velocity vector corresponding to $\mathbf{a}_1$ in (1.1.7). It is the first order dependency of displacement with time. The equation (1.1.9) discloses that for an image sequence which contains only uniform translational motion, its 3-D Fourier coefficients are distributed only within a plane, and the slope of the plane is related to translational velocity directly. This is the well-known uniform translational motion model in the frequency domain, which is just a special case of (1.1.7).

Furthermore, if we define a sequence of time intervals $[t_i, t_{i+1}]$, $i = 0, 1, 2, \dots$ and a corresponding polynomial coefficients vector $\mathbf{a}^i = [a_1^i, a_2^i, \dots, a_M^i]$, the object can take translational motion in different polynomial trajectories in different temporal pieces. With this piecewise polynomial translational motion model, complex motions can be well approximated if the time interval is small enough. In this case, the equation (1.1.8) is applied over a rectangle time window for each temporal piece. In the frequency domain, the result is to convolute a sinc function with equation (1.1.8), and multiply with a phase shift term according to the position of current temporal piece[23]. The convolution of the sinc function also spread the “coefficient plane” into a plane cluster of coefficients in the frequency domain representation of the piecewise uniform translational motion.

Taking the piecewise uniform translational motion as an example, we can see many advantages for motion estimation in the frequency domain.

Noise resistance is one of the strengths of the frequency domain approach. In most applications, only the objects which undertake large scale movements, or significant motion, are of interested. The information about the intensity changes associated with these motions are carried mainly by low frequency coefficients in the Fourier domain. Since sudden temporal interference, even of high magnitude, is pushed into the high frequency

band, it cannot destroy the long term pattern established in the frequency domain. For the case of uniform translational motion, the motion estimation process is based on determining the slope of a plane cluster, which is resistant to temporal interference. As discussed in detail in chapter 3, it can be a very robust process. In the spatiotemporal domain, to obtain long-term resistance to temporal noise is very difficult.

Another benefit of operating in the frequency domain is that some human visual perception effects are more easily interpreted. For uniform translational motion, if the temporal sampling rate is too low, the object seems undergo motion in a direction opposite to the true motion. The aliasing effects in the frequency domain explains this effect accurately. The concept of “velocity bandwidth” introduced in [25] gives quantitative criteria on how to avoid this aliasing effect in practice. In [26], proper weighting stratagem on motion estimation method are suggested to decrease the aliasing effect in the frequency approach. Another example is stroboscopic apparent motion which is the basis for motion rendition in video and movies. The explanation based on the frequency domain also helps in modelling the motion perception by the visual system.

The third advantage of frequency domain methods is that the intensity changes due to spatial textures and motions are separated. Although the spatial frequency is significant in the 3-D frequency distribution, it does not affect the slope of the plane cluster in equation (1.1.9), which dominates the estimation of the uniform translational motion. This property, along with the former two advantages, is also seen in the case of rotational motion.

Rotational motion is periodic, which is suitable for frequency domain approaches. [27] presents the frequency domain model of polynomial rotational motion. The coordinates of any pixel within the rotating object can be expressed as:

$$r(t) = r \quad (1.1.10)$$

$$\theta(t) = \theta_0 + \sum_{m=1}^M \frac{\omega_m t^m}{m!} \quad (1.1.11)$$

The $r(t)$ is the radial position of a pixel, which is unchanged. The $\theta(0)$ and $\theta(t)$ are the instantaneous angles of the pixel at beginning and time t respectively. It is assumed that θ changes in a polynomial manner, specified by polynomial coefficients ω_m , $m = 1 \dots M$. Assuming that an object undertaking global rotational motion is presented in an image sequence, denote the intensity of any pixel at (r, θ, t) in the rotational object as $I(r, \theta, t)$. The 3-D Fourier transform of the image sequence is derived as:

$$I(r, \theta, f) = \sum_{n=-\infty}^{+\infty} L_n(\Omega_r, \Omega_\theta) \bigotimes_{m=1}^M F[\exp(-jn \frac{\omega_m t^m}{m!})](f) \quad (1.1.12)$$

where the $L_n(\Omega_r, \Omega_\theta)$ is a series expression of the 2-D Fourier transform of the first frame. In the special case that $M=1$ in (1.1.10), which stands for the uniform rotational motion, equation (1.1.12) is simplified to:

$$I(r, \theta, f) = \sum_{n=-\infty}^{n=+\infty} L_n(\Omega_r, \Omega_\theta) \delta(f + \frac{n\omega_1}{2\pi}) \quad (1.1.13)$$

That is to say, if a global rotational motion is presented in an image sequence, the projection of the Fourier distribution on the temporal frequency axis is a sequence of even spaced impulses, and the interval between the impulse is proportional to the rotational speed. The interval of the impulse sequence has little to do with the texture of each frame. It is a reflection of the periodic motion in the image sequence. This is an excellent example showing that the frequency domain representation converts a large scale regularity into a concise model. The uncorrelated information such as image texture and noise is suppressed to a great extent.

Due to these advantages, a great deal of activity has focussed on the frequency domain approach. For example, [24] gives solutions for a more complex polynomial translational motion model. In [28] and [29], the combination of translational and rotational motion is modelled in the frequency domain. As more motion models and detection algorithms are developed, the frequency domain approach is growing to be an important field of studying motion estimation.

However, there is an obvious drawback associated with this class of method. Since the Fourier transform has no spatiotemporal resolution, if two or more different motions are presented, their signatures in the frequency domain will be mixed together. Unfortunately, this situation happens in almost every practical application.

1.1.5 The Joint Spatiotemporal/Spatiotemporal-frequency(st/stf) Domain Approach

As image and video signals are highly non-stationary, examining the local frequency information at specific spatiotemporal locations discloses subtle structure of the signals and provides more flexibility in designing algorithms. Joint spatiotemporal-frequency representations, which represent spatial, temporal and frequency information simultaneously, offer opportunities to merge the advantages of frequency domain and spatiotemporal techniques. Therefore, extensive investigations have been carried out in several spaces in this representation family.

The Wavelet Domain Approach

The wavelet representation establishes a multi-resolution analysis space in which each subspace reveals the characteristics of a signal at a certain resolution level. The hierarchical information structure is a reasonable modelling of human visual perception behavior and enables the “from coarse to fine” mechanism to recognize the signal’s features in different scales. In signal processing terms, the wavelet coefficients are acquired by filtering signals with a set of bandpass filters under different scales. The filters are designed to have concentrated distributions both in the spatiotemporal and frequency domains. The wavelet coefficients obtained by filters with the same bandwidth (but different spatiotemporal shifts) disclose the joint characters of the signal at one resolution grade. By carefully

designing the filter banks, efficient and complete wavelet representation can be achieved with a multi-scale structure available for algorithm development.

Great achievements in the wavelet domain can be seen almost everywhere in image and video processing. In compression, the non-stationary image or video signal becomes much more predictable in the wavelet domain. The wavelet representation organizes the stationary components in the significant coefficients of low frequency bands, whereas the subtle and sudden changes (highly non-stationary components) are often organized into high frequency band levels in the wavelet domain. Hence, multi-stage video coding schemes can be implemented with higher efficiency. In computer vision tasks such as object edge detection, the multi-resolution property of the wavelet representation enables robust scene recognizing at large scales and detail detection with high spatiotemporal resolution. Since motion estimation is also an important technology in the former two applications, it is also investigated in the wavelet domain extensively.

One approach is to integrate a spatiotemporal domain algorithm into a multi-resolution scheme. In [30], an adaptive method is designed to choose the best scales to avoid the aperture problem in estimating the local motion field. In [31], a successive approximation scheme is proposed to compute optical flow in a Laplacian pyramid structure. Iterative gradient estimation is carried out in a coarse resolution subspace at first, and the result can be used as pre-knowledge, or a new constraint, for the space with higher spatiotemporal resolution until all local resolution requirements are met. The wavelet analysis space offers a flexible framework to balance robustness and accuracy with controllable spatiotemporal resolution. In [32], a complex valued 2-D wavelet transform is designed to implement intensity correspondence searching in a multi-scale manner. Beginning from the coarse stage, the local translational model is evaluated with a maximum likelihood estimator in a Bayesian approach. Each estimation output serves as reference for motion estimation in the finer stages. How best to use the reference knowledge from the previous

stage is still an interesting topic. The performance of this method is proven to be superior in accuracy and noise resistance.

Another class of algorithm seeks to find the motion models embodied in the wavelet coefficients. [33] defines a condition that the polynomial texture images are undertaking translational movement. By designing a separable wavelet filter bank matched to the image texture, the resulting 2-D wavelet representation bears the velocity information directly in the ratio between coefficients of two subimages. Although various image textures and wavelet designs are also studied, there is still a distance from practical applications.

The difficulty in determining motion models directly in the wavelet domain is due to the inconsistency of different resolution levels. The spatiotemporal correspondence between wavelet coefficients of different levels is hard to trace. In addition, the wavelet transform is spatiotemporal-variant. That is to say, a spatiotemporal shift will result in a different wavelet domain distribution. That also disturbs the inter-frame representation consistency, which is important in determining a concise model.

Nevertheless, [34] establishes an approach for detecting the frequency domain model of uniform translational motion in the wavelet domain. A 3-D wavelet transform is constructed with filter banks that are selective for different orientations, speeds and locations. This is accomplished by applying 1-D Hilbert transform along the dimension of interest to conventional wavelet basis functions. The “extended” wavelets couple the local frequency information at the diagonal positions of local frequency domain. By detecting the corresponding wavelet coefficients accordingly, motion vectors can be estimated for each spatiotemporal location. Utilizing a frequency domain motion model in a modified joint representation is a promising approach to keep the advantages of frequency domain analysis and enhance spatiotemporal resolution simultaneously. The 3-D Gabor domain approach, which is the main content of this dissertation, intends to achieve this goal with superior performance.

The Gabor Representation and Human Visual System

Because the Gabor representation is fundamental to the motion estimation methods developed in this work, chapter 2 will be dedicated to discussing it in detail. Compared with the wavelet representation, its joint st/stf analysis structure is more closely associated with the local frequency domain distribution. This makes it more feasible to derive corresponding motion models and construct algorithms.

Interest in the Gabor representation is due in part results in visual neural system research. In order to understand how the human visual perception system works, numerous studies have been undertaken to model visual information acquiring mechanisms such as position-dependent visual sensing, dynamic scene encoding, logarithmic frequency scaling and noise resistant motion tracking. Neurobiological experimental results show that the Gabor representation is a good simulation of the processes underlying vision in many aspects. For example, the cortical simple cells in the mammalian visual cortex exhibit properties such as “orientation selectivity”, “spatiotemporally localized sampling” and “quadrature phase relationship” between adjacent cells, which can be well approximated by certain types of Gabor representations[35]. In [36], a generalized Gabor scheme is suggested to modelling different vision physiology functions by changing parameters. The generalized the Gabor representation shows improved flexibility in explaining many visual perception phenomena and the experimental results using typical bandlimited images are quite promising.

1.2 Organization

The rest of the dissertation is arranged in 4 chapters.

Chapter 2 introduces the 3-D Gabor representation, the main analysis space for this research. Based on a systematic discussion of the representations of signals, the struc-

ture, properties and advantages of the 3-D Gabor domain are described to explain why it provides a suitable framework for motion estimation.

In Chapter 3, the mathematical form of the translational motion model in the 3-D Gabor domain is derived. A piecewise uniform translational motion model is then established. As the model form is rather concise, an efficient algorithm can be designed for motion estimation. The algorithm is applied experimentally, and its performance is analyzed.

In Chapter 4, the goal is to enhance spatiotemporal resolution while keeping the estimation accuracy and robustness. To accomplish this, a partial reconstruction technique is applied in the 3-D Gabor domain. Integrated with the algorithm constructed in chapter 3, the method achieves spatiotemporal resolution at the pixel level with the possibility of subpixel resolution. Performance evaluation of the new algorithm substantiates these significant resolution improvements.

Chapter 5 investigates rotational motion in the 3-D Gabor domain. The mathematical model is derived, and the detection algorithms are built accordingly. Although the algorithm is only for uniformly rotating objects and there is additional work to be done to make the method practical, the suitability of the 3-D Gabor domain for this kind of motion analysis is clearly demonstrated.

1.3 Notation

Lowercase letters such as x denote vectors. They are assumed to be column vectors if not indicated otherwise. Uppercase letters such as $A_{m \times n}$ denote matrix with m rows and n columns.

The $\langle x(t), y(t) \rangle$ indicates the inner product two functions:

$$\langle x(t), y(t) \rangle = \int_{-\infty}^{+\infty} x(t)y(t)dt \quad (1.3.1)$$

in the continuous case and

$$\langle x(t), y(t) \rangle = \sum_{t=-\infty}^{+\infty} x(t)y(t) \quad (1.3.2)$$

in the discrete case.

The norm $\|f(t)\|$ can be regarded as $\sqrt{\langle f(t), f(t) \rangle}$.

The Fourier transform of a time function $f(t)$ is presented as $\mathbf{F}[f(t)](\omega)$ where ω is the frequency variable. In a similar way, the Gabor transform of a function $f(t)$ is shown as $\mathbf{G}[f(t)](t_d, \omega_t)$ where t_d and ω_t are Gabor coefficient indices.

Chapter 2

The 3-D Gabor Representation

2.1 Representation of Signals

The information carried by a signal can be represented in innumerable ways. In each application, proper signal representations should be chosen to display the interesting aspects of information as explicitly as possible. For example, the function of time $f(t)$ is often used because the amplitude changing with time is easy to observe and utilize. Another popular representation is the Fourier transform, a function of frequency $F(\omega)$. It shows how frequently the signal changes its values by illustrating the different contributions of each frequency components it contains. Because of the different points of view in representing the signal, analysis and processing methods based on different representations have different performance potentials.

Therefore, establishing an appropriate representation (or domain) is of great importance in signal processing. If the desired representation is not the form in which the signal is acquired collected, a transform is needed to map the signal into the objective domain. For example, if the frequency domain processing is demanded but the current signal is stored as a time function $f(t)$, the $f(t)$ must be re-expressed as a frequency domain

function $F(\omega)$:

$$f(t) = \int_{-\infty}^{+\infty} F(\omega) \exp(j\omega t) d\omega \quad (2.1.1)$$

In the expansion, the integral can be regarded as linear combination of a set of complex exponential functions at every frequency point. Since each of them has a single and distinct frequency ω throughout the signals duration, the associated Fourier coefficient $F(\omega)$ reflects the amount of signal at each frequency point ω exclusively. That is: the frequency domain representation.

Finding the $F(\omega)$ for each basis function $\exp(j\omega t)$ is well known as the Fourier transform:

$$F(\omega) = \int_{-\infty}^{+\infty} f(t) \exp(-j\omega t) dt \quad (2.1.2)$$

It is the inner product of the time function $f(t)$ and a set of analysis functions, which are the complex conjugates of the transform basis functions. Each analysis function stands for the same frequency components as its conjugate part. In general, the analysis functions in a transform and the basis functions in the associated expansion are “dual” to each other. They set up a transform pair between the two representations under some conditions that will be mentioned later.

To design a set of suitable basis functions is a fundamental task in signal representation. The basis functions determine the information structure and the analyzing ability of a representation. For example, if the time resolution is the first priority, the time function $f(t)$ is selected for the finest time resolution of its basis function, $\delta(t)$. On the other hand, if frequency characteristics are of interest, the Fourier domain should be used because the frequency representation $F(\omega)$ provides the highest frequency resolution with basis functions at every frequency point. However, the limits of the two representations are as obvious as their advantages. The time function cannot distinguish different frequency components because the $\delta(t)$ is a constant throughout the frequency domain. On the other hand, if

the frequency distribution is time varying, the $F(\omega)$ tells nothing about when or where the frequency components exist. Since most of the practical systems in engineering are non-stationary in time or space, neither of the two representations are suitable to describe them dynamically. The Gabor representation to be introduced in this chapter, however, provides a representation platform on which the joint time-frequency information structure of a signal is revealed in a well organized way.

2.2 Frame for Signal Representation

In the following discussion, without losing generality, we assume the signal is previously expressed as a time function $f(t)$. The function sequence φ_j indexed by j are the elementary functions of the transform. Denote $T\{f(t)\} = \langle f(t), \varphi_j \rangle$ the representation of function $f(t)$ in the objective domain. be a rational transform, $T\{f(t)\}$ must satisfy two conditions:

- (1) One to one correspondence: if $f_1(t) = f_2(t)$, $T\{f_1(t)\} = T\{f_2(t)\}$.
- (2) Continuity in transform and inverse transform:

$$A\|f(t)\|^2 \leq \sum_j |\langle f(t), \varphi_j \rangle|^2 \leq B\|f(t)\|^2 \quad (2.2.1)$$

This can be understood as a function $f(t)$ with nonzero norm $\|f(t)\| \neq 0$ must have at least one projected component $\langle f(t), \varphi_j \rangle$ with nonzero norm; simultaneously, a function with finite norm $\|f(t)\| < \infty$ must have a finite norm summation $\sum_j |\langle f(t), \varphi_j \rangle|^2$ in the objective domain.

Under these two conditions, the function set $\varphi_j | j \in \mathbf{Z}$ spans a frame for signal representation. A signal $f(t)$ can be expressed as a linear combination of the φ_j where the combination coefficients c_j are the representation of $f(t)$ in the frame.

$$f(t) = \sum_j c_j \varphi_j \quad (2.2.2)$$

If a unique representation c_j exists for any function $f(t)$, the function set φ_j is called a basis which spans a function space.

For a set of basis functions of a frame, there is a dual function set $\hat{\varphi}_j | j \in \mathbf{Z}$ by which the expansion coefficients c_j can be calculated by inner product as:

$$c_j = \langle f(t), \hat{\varphi}_j^* \rangle = \int_{-\infty}^{\infty} f(t) \hat{\varphi}_j^*(t) dt \quad (2.2.3)$$

where the $*$ is complex conjugate.

In some special cases such as the Fourier transform, the basis function set is dual to itself: $\varphi_j = \hat{\varphi}_j$. They are called orthogonal basis function sets and compose orthogonal frames in which $A = B = 1$ in equation (2.2.1). In orthogonal function spaces, each basis function is orthogonal to the other basis functions in the function set and only couples the information of its own dimension.

The orthogonal frame is widely used and appreciated because it not only saves work in finding dual functions, but also is easy to manipulate due to the independence of representation coefficients. Yet there exists a much larger family of basis functions that possess powerful signal expression abilities, but with the dual parts having a biorthogonal relationship:

Substitute (2.2.2) to (2.2.3) we can see the relation between φ_j and $\hat{\varphi}_j$:

$$\langle \varphi_j(t), \hat{\varphi}_k(t) \rangle = \delta_{j,k}(t) \quad (2.2.4)$$

To solve the biorthogonal equation for the dual functions is not a trivial task, even for some function spaces whose basis functions look very nice. The Gabor representation is one example.

2.3 1-D Gabor Representation

2.3.1 The Gabor Basis Functions

The Gabor representation was proposed by Dr. Dennis Gabor in 1946. In an attempt to analyze time and frequency information simultaneously, he suggested a set of basis functions having concentrated shapes both in time and frequency domains.

The basis functions are composed of Gaussian windows with different time shifts and modulated to different frequency points,

$$g_{n,m}(t) = g(t - nT) \exp(j2\pi m\Omega(t - nT)) \quad (2.3.1)$$

Where the Gaussian window $g(t)$ has the normalized energy $\int_t \|g(t)\|^2 dt = 1$, and has concentrated distributions both in the time and frequency domains. For example:

$$g(t) = \left(\frac{\alpha}{\pi}\right)^{\frac{1}{4}} \exp(-\alpha t^2) \quad (2.3.2)$$

where α is a parameter (like σ of Gaussian function) controlling the degree of concentration in the time and frequency domains. The larger the α is, the higher resolution it achieves in time. At the same time, the frequency resolution decreases under the constraint of the Heisenberg uncertainty. As the Gaussian window reach the lower bound of the Heisenberg uncertainty, the representation is widely regarded as an optimum one in the sense of joint resolution. The T and the Ω in equation (2.3.1) are the steps in time and frequency shifts respectively. The Gabor basis function indexed by $[n, m]$ is jointly localized around time-frequency point $(nT, m\Omega)$. It is not difficult to see that T and Ω are the sampling intervals of adjacent time-frequency points. Their product $T \times \Omega$ indicates the density of sampling. Also, it represents the region controlled by a Gabor basis function. If the representation is to be complete, the region cannot be too large. In uniform sampling case, which is implied from now on, it is necessary to ensure $T \times \Omega \leq 2\pi$. If $T \times \Omega = 2\pi$, we refer to it as critical sampling. Sampling density less than this threshold may lose information. The

case of $T \times \Omega < 2\pi$ is over sampling, which is recommended for reasons that will be explained later. Fig(2.1) and Fig(2.2) displays some basis functions of two Gabor spaces with different α 's.

2.3.2 The Continuous Gabor Representation

When a continuous signal $f(t)$ is expanded as a linear combination of a set of continuous basis functions $g_{n,m}(t)$ shown in equation (2.3.1), the coefficients $C_{n,m}$ associated with each basis functions uniquely define the signal in its continuous Gabor representation:

$$\begin{aligned} f(t) &= \sum_n \sum_m C_{n,m} g_{n,m}(t) \\ &= \sum_n \sum_m C_{n,m} \left(\frac{\alpha}{\pi}\right)^{\frac{1}{4}} \exp(-\alpha(t - nT)^2) \exp(j2\pi m\Omega(t - nT)) \end{aligned} \quad (2.3.3)$$

Because of the joint time-frequency locality property of the Gabor basis functions, the time varying frequency distribution is distinguished by different Gabor coefficients. This joint time-frequency analysis ability makes the Gabor representation a powerful tool in processing non-stationary signals and analyzing dynamic systems.

Designing a Gabor Representation Framework for special Applications

The three parameters, α , T and Ω in equation (2.3.3) must be determined before creating a Gabor framework for an application. Although the α of the Gaussian window plays a significant part in the time and frequency resolution, experiments have shown that a denser sampling rate along the dimension of interest also helps to enhance resolution to some degree. The T and Ω , besides be selected to ensure completeness, must be carefully planned to meet the resolution requirements of the application as well as keeping moderate overlaps between the adjacent basis functions. If the overlap is too small (sampling too sparsely for the selected α), large signal components in a poorly covered region may result in very large coefficients for the neighboring basis functions. Fig(2.3) shows an example

where the shifts of the Gaussian windows provide efficient coverage in the time domain. It also illustrates a case of improper parameter settings for α and the sampling rate. If the overlap is too much (sampling too dense for α), the representation becomes too redundant to be computationally efficient. Sometimes, moderate additional overlap beyond necessary is helpful in achieving better performance. We will discuss this further in the following chapters. To avoid difficulties in implementation, throughout this thesis we consider only the case that the three parameters do not change spatially or temporally in a constructed Gabor domain. The more complex Gabor representation with attention-based variable parameters [36] may be a topic of further work. In conclusion, the design of a Gabor representation is an engineering art, with the goal of covering the entire time-frequency plane efficiently, and with the required local time-frequency resolution. In the following chapter, we will show the different performance for different parameter settings based on the same motion estimation algorithm in the Gabor domain.

The Auxiliary Functions of the Gabor Basis

Unfortunately, the Gabor representation is not an orthogonal frame. In theory, the analysis functions for computing the Gabor coefficients must be solved from the biorthogonal equation (2.2.4) for each Gabor basis function set. Although researchers have derived some analysis solutions [37], there are many constraints on the feasible Gabor expansion, and the auxiliary function is neither time nor frequency localized. A method for determining the dual function of an arbitrary Gabor basis set does not yet exist. Fortunately, since most of the signal processing applications in engineering deal with discrete data, the discrete dual functions for a given discrete Gabor basis function set can be solved numerically from linear equation groups. Thus the discrete Gabor representation for any given digital signal can be found numerically.

2.3.3 The Discrete Gabor Representation

The discrete Gabor representation was developed for numerical implementation on digital signals [38]. Suppose we have discrete signal $f(l)$ where $l \in [0, L - 1]$, the discrete Gabor expansion is defined as:

$$f(l) = \sum_{k=0}^{\hat{K}-1} \sum_{w=0}^{\hat{W}-1} C_{k,w} g_{k,w}(l) \quad (2.3.4)$$

where

$$g_{k,w}(l) = g(l - k\Delta K) \exp(j2w\Delta W((l - \Delta K))) \quad (2.3.5)$$

where $\Delta K = \frac{L}{\hat{K}}$ and $\Delta W = \frac{2\pi}{\hat{W}}$. The $g(l)$ is a discrete Gaussian-type function such as: $g(l) = \exp(-\pi\alpha l^2)$. The α has the same effect as in function (2.3.2).

To ensure the invertibility of the discrete Gabor representation, the basis functions must be arranged over a critical or over sampling lattice such as: $\Delta K \times \Delta W \leq 2\pi$. In this case, the number of the Gabor coefficients should be equal to or larger than the signal length. Otherwise, the original data cannot be recovered from the expansion.

2.3.4 Coefficient Calculation

One of the general method to find coefficients of any Gabor representation for any signal is to solve a set of linear equations that embody the biorthogonal relationship between the Gabor basis functions and their auxiliary functions. Denote as $P = \hat{K} \times \hat{W}$ the number of Gabor basis functions in the representation and L the number of signal pixels. From the discrete 3-D Gabor representation in equation (2.3.4) we have a matrix form :

$$\mathbf{f}_{L \times 1} = \mathbf{G}_{P \times L} \mathbf{C}_{P \times 1} \quad (2.3.6)$$

where

$$\mathbf{f}_{L \times 1} = [f(1), f(2), \dots, f(L)]^T$$

is the column vector of all signal points.

$$\mathbf{G}_{P \times L} = \begin{bmatrix} \mathbf{g}_1(1) & \mathbf{g}_1(2) & \dots & \mathbf{g}_1(L) \\ \mathbf{g}_2(1) & \mathbf{g}_2(2) & \dots & \mathbf{g}_2(L) \\ \dots & \dots & \dots & \dots \\ \mathbf{g}_P(1) & \mathbf{g}_P(2) & \dots & \mathbf{g}_P(L) \end{bmatrix} \quad (2.3.7)$$

is the matrix of the Gabor basis functions. Each row is the set of discrete values in a given Gabor basis function.

$$\mathbf{C}_{P \times 1} = [C_1, C_2, \dots, C_P]^\top$$

is the corresponding set of Gabor coefficients. To ensure numerical stability as well as allowing for the over sampling case in which $P > L$, The least mean square (LMS) criteria can be applied, by which we choose our solution $\mathbf{C}$ to minimize the error $(\mathbf{G}\mathbf{C} - \mathbf{f})^\top (\mathbf{G}\mathbf{C} - \mathbf{f})$. In the LMS sense, the solution is:

$$\hat{\mathbf{C}} = (\mathbf{G}^\top \mathbf{G})^{-1} \mathbf{G}^\top \mathbf{f} \quad (2.3.8)$$

2.4 The 3-D Gabor Representation

To apply the Gabor decomposition to image sequences, it can be extended to the 3-D Gabor representation[26][39]:

$$f(x, y, t) = \sum_{i,j,k,u,v,w} C_{i,j,k,u,v,w} \mathbf{g}_{i,j,k,u,v,w}(x, y, t) \quad (2.4.1)$$

The basis functions,

$$\begin{aligned} \mathbf{g}_{i,j,k,u,v,w}(x, y, t) &= g(x - i\Delta_X, y - j\Delta_Y, t - k\Delta_T) \\ &\quad \exp(j2\pi(u\omega_x(x - i\Delta_X) + v\omega_y(y - j\Delta_Y) + \\ &\quad w\omega_t(t - k\Delta_T))) \end{aligned} \quad (2.4.2)$$

have six parameters (i, j, k, u, v, w) to specify their different spatial, temporal and three dimensional frequency positions with spatiotemporal sampling intervals of $\Delta_X, \Delta_Y, \Delta_T$ and frequency sampling intervals of $\omega_x, \omega_y, \omega_z$. The 3-D extension of the Gabor representation establishes an analysis framework that can capture the dynamic joint spatiotemporal-frequency information carried by moving objects.

The 3-D Discrete Gabor Representation

The 3-D discrete Gabor representation for digital image sequences can be extended from the 1-D discrete Gabor expansion in equation (2.3.4). Suppose we have an image sequence with L frames of $M \times N$ images, whose gray values are denoted as $f(m, n, l)$ where $x \in [0, M - 1], y \in [0, N - 1], t \in [0, L - 1]$. The 3-D discrete Gabor representation is defined as:

$$f(m, n, l) = \sum_{i=0}^{\hat{I}-1} \sum_{j=0}^{\hat{J}-1} \sum_{k=0}^{\hat{K}-1} \sum_{u=0}^{\hat{U}-1} \sum_{v=0}^{\hat{V}-1} \sum_{w=0}^{\hat{W}-1} C_{i,j,k,u,v,w} \mathbf{g}_{i,j,k,u,v,w}(m, n, l) \quad (2.4.3)$$

The basis functions utilize a Gaussian envelope shifted spatiotemporally by index $[i, j, k]$ and frequency modulated by index $[u, v, w]$:

$$\begin{aligned} \mathbf{g}_{i,j,k,u,v,w}(m, n, l) = & \frac{1}{(2\pi)^{\frac{3}{2}} \sigma_m \sigma_n \sigma_l} \exp(-[(\frac{m-i\Delta_m}{\sigma_m})^2 + (\frac{n-j\Delta_n}{\sigma_n})^2 + (\frac{l-k\Delta_l}{\sigma_l})^2]) \\ & / 2 \exp(j2\pi(u\Delta f_m(m-i\Delta_m)) + v\Delta f_n(n-j\Delta_n)) + \\ & w\Delta f_l(l-k\Delta_l)) \end{aligned} \quad (2.4.4)$$

where $\Delta_m = \frac{M}{\hat{I}}, \Delta_n = \frac{N}{\hat{J}}, \Delta_l = \frac{L}{\hat{K}}, \Delta f_m = \frac{1}{\hat{U}}, \Delta f_n = \frac{1}{\hat{V}}, \Delta f_l = \frac{1}{\hat{W}}$.

To ensure the invertibility of the discrete 3-D Gabor representation, the basis functions must be arranged over a critical or over sampling lattice as: $\Delta_m \times \Delta f_m \leq 2\pi, \Delta_n \times \Delta f_n \leq 2\pi, \Delta_l \times \Delta f_l \leq 2\pi$. In this case, the number of the Gabor coefficients in each dimension should be equal to or larger than the number of pixels in that dimension. Otherwise, the original data cannot be recovered from the expansion. Although critical

sampling is enough to retain all the information in an image sequences, we will explain latter that the over sampling lattice has better performance.

The Calculation of the 3-D Gabor Coefficients

In theory, the 3-D Gabor representation can be calculated by inner products between the signal and the 3-D auxiliary functions,

$$\begin{aligned} C_{i,j,k,u,v,w} &= \sum_{m,n,l} f(m,n,l) \Upsilon_{i,j,k,u,v,w}(m,n,l) \\ &= \sum_{m,n,l} f(m,n,l) \gamma(m-i\Delta_m, n-j\Delta_n, l-k\Delta_l) \exp(j2\pi \\ &\quad (u\Delta f_m(m-i\Delta_m)) + v\Delta f_n(n-j\Delta_n) + w\Delta f_l(l-k\Delta_l)) \end{aligned} \quad (2.4.5)$$

where the Δ_m , Δ_n , Δ_l , Δf_m , Δf_n , and Δf_l have the same meaning as in equation (2.4.3). The 3-D auxiliary functions $\gamma(m,n,l)$ are 3-D biorthogonal with the Gabor basis functions in the base band. In practice, we can use a method similar to the one presented in the 1-D case. The only difference is that the matrix is filled with the 3-D data sequentially. Denote as $M = \hat{I} \times \hat{J} \times \hat{K} \times \hat{U} \times \hat{V} \times \hat{W}$ the number of 3-D Gabor basis functions in the representation and $N = I \times J \times K$ the number of signal pixels. From the discrete 3-D Gabor representation we have:

$$\mathbf{f}_{N \times 1} = \mathbf{G}_{M \times N} \mathbf{C}_{M \times 1} \quad (2.4.6)$$

where

$$\mathbf{f}_{N \times 1} = [f(1, 1, 1), f(2, 1, 1), \dots, f(I, J, K)]^\top$$

is the column matrix of the gray values of all pixels in the image sequence;

$$\mathbf{G}_{M \times N} = \begin{bmatrix} \mathbf{g}_1(1, 1, 1) & \mathbf{g}_1(2, 1, 1) & \dots & \mathbf{g}_1(I, J, K) \\ \mathbf{g}_2(1, 1, 1) & \mathbf{g}_2(2, 1, 1) & \dots & \mathbf{g}_2(I, J, K) \\ \dots & \dots & \dots & \dots \\ \mathbf{g}_M(1, 1, 1) & \mathbf{g}_M(2, 1, 1) & \dots & \mathbf{g}_M(I, J, K) \end{bmatrix} \quad (2.4.7)$$

is the matrix of the Gabor basis functions. Each row is the set of discrete values in a given Gabor basis function.

$$\mathbf{C}_{M \times 1} = [C_1, C_2, \dots, C_M]^\top$$

is the corresponding set of Gabor coefficients. The solution in the LMS sense is exactly the same as equation(2.3.8).

Compared with the auxiliary function inner product method, $(\mathbf{G}^\top \mathbf{G})^{-1} \mathbf{G}^\top$ is the matrix of discrete auxiliary functions with the same general form as the $\mathbf{G}$ matrix. In the over sampling case, the matrix $\mathbf{G}^\top \mathbf{G}$ may be poorly conditioned because there may be too much similarity between the basis functions. In this case, suitable solutions can sometimes be found by singular value decomposition. For further discussion on the calculation of the discrete Gabor transform, see [39].

The Design of 3-D Gabor Representations for Specific Applications

Compared with the design of 1-D frameworks, although the 3-D Gabor analysis framework provides more flexibility in setting parameters because of more degrees of freedom, the same constraints on completeness and numerical stability of the 1-D case must be followed along each dimension. For example, the product of the number of temporal samples and the number of temporal frequency bands must be equal to or larger than the number of frames in the image sequence. Otherwise, temporal information may be omitted in the 3-D Gabor domain.

Instead of the three main parameters in the 1-D case, there are nine parameters to be determined for a 3-D Gabor representation. The three spatiotemporal variances of the 3-D Gaussian window mainly contribute to the joint analysis resolution of the basis functions. The six spatiotemporal/spatiotemporal-frequency sampling intervals trade the efficient coverage of the 6-D coefficient space under the constraints of completeness and the resolution performance in the dimensions of interest. For example: if the spatiotempo-

ral/frequency domain information is important for detecting a frequency motion model, the sampling rate along the three frequency axes should be selected based on the complexity of the model. In practice, the nine parameters are interact with each other to meet the theoretical and application requirements. In the following chapter, we will discuss different 3-D Gabor domain configurations and their performance for given applications.

Although the critical sampling scheme is much more computationally efficient, the over sampling framework is recommended under some circumstances because of its better spatiotemporal/ spatiotemporal-frequency localization properties. In the critical sampling case, the auxiliary functions are poorly localized in the spatiotemporal domain and in the frequency domain[36][40]. In the over sampling case, the auxiliary functions yield much better localization and are more similar to the Gabor basis functions[41] in both domains. An orthogonal-like analysis framework can be formed by sufficient over sampling. In the following chapters, the motion estimation method constructed in the 3-D Gabor domain will be presented. The advantages of over sampling schemes are also discussed and demonstrated by the experiments.

(a) Temporal domain of the Gabor Basis functions.

(b) The Frequency domain of the Gabor Basis functions.

Figure 2.1: Example of Gabor basis functions with $\alpha = 0.0884$.

(a) Temporal domain of the Gabor Basis functions

when $\alpha = 0.0884$.

(b) The frequency domain of
the Gabor Basis functions.

Figure 2.2: Example of Gabor basis functions with $\alpha = 0.1788$.

(a) Proper design of the Gabor basis envelopes in the temporal domain.

(b) Improper design of the Gabor basis envelopes in the temporal domain.

Figure 2.3: Example of inappropriate designing of the Gabor representation

Chapter 3

Motion Estimation in the 3-D Gabor Domain

Applying the Gabor representation in image and image sequence processing has been a topic of research for a number of years. As reviewed in chapter one, this was motivated in part by the results of neurobiological experiments seeking to model the biological visual system. Experimental results based on the visual neural networks of mammals indicate physical structures implementing tunable spatial/spatial-frequency localization, orientation selectivity, attention-based flexible sampling rates and quadrature phase relations among cortical simple cells. These effects can be well modelled by the Gabor basis functions[36][35]. The 3-D Gabor representation, with all properties of the Gabor domain and temporal parameters, was developed as a model to be suitable for dynamic visual perception mechanisms, particularly motion estimation.

The 3-D Gabor representation has a unique advantage for motion estimation. For those Gabor coefficients with the same spatiotemporal index set $[i, j, k]$ but different frequency indices $[u, v, w]$ in equation (2.4.3), they contain the frequency information of the spatiotemporally localized region. This local frequency information can be organized in the same structure as the 3-D Fourier domain. In this way, the information structure of the

3-D Gabor domain can be arranged in a manner analogous to a set of local 3-D Fourier domains centered at different spatiotemporal positions. This information structure enables the implementation of frequency domain methods with spatiotemporal resolution. However, since the 3-D Gabor coefficients are inner products between the image sequence and the 3-D Gabor auxiliary functions instead of Fourier basis functions, their frequency information distribution is not the same as in the 3-D Fourier domain. It is therefore instructive to find the spatiotemporal/frequency model of the corresponding motion in the 3-D Gabor domain.

3.1 The Translational Motion Model in the 3-D Gabor Domain

Translational motion is frequently encountered in practice. The general model of translational motion of a rigid object can be described as:

$$\mathbf{x} = \mathbf{x}_0 + D(l) \quad (3.1.1)$$

where the vector $\mathbf{x}$ represents the spatial coordinates (m, n) of the object in equation (2.4.3), at discrete time point l , $\mathbf{x}_0$ represents its spatial coordinates in the first frame and $D(l)$ is the time dependent displacement of the object. Denote the 2-D intensity function of the first frame as $I_0(\mathbf{x}_0)$. After time l , the image moves to $I(\mathbf{x})$. Assuming the intensity of the object does not change, we have

$$I(\mathbf{x}) = I_0(\mathbf{x}_0) = I_0(\mathbf{x} - D(l)) \quad (3.1.2)$$

The discrete 2-D Gabor transform for the object's first frame is:

$$C_{i,j,u,v}^0 = C_{\mathbf{d},\mathbf{w}}^0 = \sum_{\mathbf{x}_0} I_0(\mathbf{x}_0) \gamma_1(\mathbf{x}_0 - \mathbf{d}) \exp(-j2\pi \mathbf{w}^T (\mathbf{x}_0 - \mathbf{d})) \quad (3.1.3)$$

where the subscript 0 means the 2-D representation of the first frame. As (i, j) and (u, v) are the index numbers of the Gabor basis function in spatial and frequency domain respectively, the $\mathbf{d} = [i\Delta_m, j\Delta_n]$ and $\mathbf{w} = [u\Delta f_m, v\Delta f_n]$ are the spatial and spatial frequency locations of the corresponding Gabor basis functions. The Δ_m and the Δ_n are spatial distances between the adjacent Gabor basis functions. The Δf_m and Δf_n are distances between adjacent Gabor basis functions on the two spatial frequency axes. Since the 3-D auxiliary function $\gamma(m, n, l)$ in equation (2.4.5) is sparable, we express its spatial part as $\gamma_1(\mathbf{x})$, and the temporal part as $\gamma_2(l)$.

Denote as l_d and f_l the time and temporal frequency shifts of the Gabor basis functions. In the appendix, we show that the discrete 3-D Gabor representation of the object during the interval $[0, l]$ is:

$$C_{\mathbf{d}, l_d, \mathbf{w}, f_l} \approx C_{\mathbf{d}, \mathbf{w}}^0 \sum_l \exp(j2\pi \mathbf{w}^T D(l)) \gamma_2(l - l_d) \exp(-j2\pi f_l(l - l_d)) \quad (3.1.4)$$

The $\mathbf{d}$ and $\mathbf{w}$ are the same as that of equation (3.1.3). The $\gamma_2(l)$ is the temporal part of the 3 D auxiliary function.

The summation can be regard as the 1-D Gabor transform of $\exp(j2\pi \mathbf{w}^T D(l))$. Denoting by $\mathbf{G}$ the 1-D Gabor transform, equation (3.1.4) can be written as:

$$C_{\mathbf{d}, l_d, \mathbf{w}, f_l} \approx C_{\mathbf{d}, \mathbf{w}}^0 \mathbf{G}[\exp(j2\pi \mathbf{w}^T D(l))](l_d, f_l) \quad (3.1.5)$$

This is the model of translational motion in the 3-D Gabor domain.

In the case of uniform translational motion, the time dependent displacement term is just $\mathbf{V}l$ where $\mathbf{V}$ is the velocity vector. In this case, the summation in equation (3.1.4) can be regarded as the Fourier transform of the temporal auxiliary function $\gamma_2(l)$ with a frequency shift $\mathbf{w}^T \mathbf{V}$. Therefore, for an object undertaking uniform translational motion, if we denote by $\mathbf{F}$ the Fourier transform, the discrete 3-D Gabor coefficients can be expressed in the form of:

$$C_{\mathbf{d}, l_d, \mathbf{w}, f_l} \approx C_{\mathbf{d}, \mathbf{w}}^0 \mathbf{F}[\gamma_2(l)](f_l + \mathbf{w}^T \mathbf{V}) \quad (3.1.6)$$

In the critical sampling case, the auxiliary function γ_2 is not st/stf localized and the waveform is very different from the corresponding Gabor basis function. As the sampling rate rises, the auxiliary function will approach the Gabor basis function both in waveform and in st/stf localization. In the over sampling case, the envelope of the auxiliary functions is similar to a Gaussian envelope, which is also concentrated in the frequency domain. The magnitude and spectrum of $\gamma_2(l)$ for critical sampling and over sampling by a rate of 2 are shown in Fig(3.1) and Fig(3.2). We can see that if the variance of the Gaussian envelope is not small (which means the 3-D basis function has a large temporal span), the spectrum is like a δ function spread around the original frequency point. In this case, we can expect that after the frequency shift of $w^T V$, the coefficients with significant energy are distributed around the plane: $f_l + w^T V = 0$. If we can determine the “coefficient plane” in the 3-D Gabor space, the velocity parameter can be directly obtained from the plane equation.

(a) The 1-D Gabor auxiliary function
in the temporal domain

(b) The Gabor auxiliary function
in the frequency domain

Figure 3.1: A 1-D Gabor auxiliary function for the critical sampling case

(a) The 1-D Gabor auxiliary function
in the temporal domain

(b) The Gabor auxiliary function
in the frequency domain

Figure 3.2: A 1-D Gabor auxiliary function with over sampling

3.2 The Piecewise Uniform Translational Motion Estimation in the 3-D Gabor Domain

The piecewise uniform translation model is the simplest practical motion model. The motion trajectory is approximated as a sequence of uniform translational motions. The motion of an object is described as a set of time interval $[t_i, t_{i+1}]$, $i = 0, 1, 2, \dots$ and corresponding translation vectors $\mathbf{V} = [V_x^{(i)}, V_y^{(i)}]$ as the average velocity in the interval. If considered over infinite time duration, the frequency signature of the uniform translation model is well known as a pure plane. The slope of the plane is determined by the velocity of the translation. If the object's velocity is far from uniform throughout the image sequence, such as polynomial time dependent or sinusoidal time dependent, its trajectory can be approximated as piecewise uniform translations in a sequence of small spatiotemporal span. The discrete 3-D Gabor representation provides a suitable information structure to estimate piecewise uniform translational motion. Using the model presented in equation (3.1.6), the velocity information at each local spatiotemporal position can be found directly

from the 3-D Gabor coefficients belonging to that local region. That is: the slope of the plane around which the most significant Gabor coefficients are distributed.

In practice, we use the magnitude of the coefficients as a measurement of significance. Weighted LMS (Least Mean Square error) estimation is used to find the plane that best fits the cloud of coefficients.

The 3-D Gabor coefficients corresponding to a local spatiotemporal region are arranged by their center frequencies. Denote by u, v , and w the horizontal, vertical, and temporal frequency respectively. Suppose there are n nonzero coefficients at frequencies points, $[u_1, v_1, w_1], [u_2, v_2, w_2], \dots, [u_n, v_n, w_n]$, and that they have corresponding magnitudes in the vector $\mathbf{m} = [m_1, m_2, \dots, m_n]$. In the ideal case, if uniform translational motion is exhibited at a spatiotemporal location, most of those coefficients should be clustered around frequency points in the plane defined by the equation:

$$w = -r_x u - r_y v \quad (3.2.1)$$

where $[r_x, r_y]$ is the velocity of the uniform translational motion and $[u, v, w]$ are the coordinates of the point belonging to the plane. With $\mathbf{u} = [u_1, u_2, \dots, u_n]^T$, $\mathbf{v} = [v_1, v_2, \dots, v_n]^T$, and, $\mathbf{w} = [w_1, w_2, \dots, w_n]^T$, the plane equation can be written as:

$$\mathbf{w} = \mathbf{A}\mathbf{r} \quad (3.2.2)$$

where $\mathbf{A} = [\mathbf{u}\mathbf{v}]$ and $\mathbf{r} = [r_x, r_y]^T$.

However, since the motion model in the 3-D Gabor domain is a spread δ function, there are some coefficients with small magnitudes standing far from the plane. In addition, some coefficients due to noise may be distributed randomly out of the plane. To reduce these negative effects in estimating the plane, an $n \times n$ reliability information matrix $\mathbf{M}$ is introduced based on the magnitude vector $\mathbf{m}$. The Gabor coefficients with higher magnitude are more likely due to the object and its motion and should be weighted more in finding the slope. Each element in $\mathbf{m}$ is mapped to a reliability measurement $f(m_i), (i = 1, 2, \dots, n)$,

which is a positive monotonically increasing function of m_i . Here we choose $f(m_i) = m_i^4$ by experiment. The reliability matrix is: $M = \text{diag}[m_1^4, m_2^4, \dots, m_n^4]$. With this reliability measurement, the plane can be estimated in the well-known weighted least mean square error sense: as:

$$\hat{\mathbf{r}} = -(\mathbf{A}^\top \mathbf{M} \mathbf{A})^{-1} \mathbf{A}^\top \mathbf{M} \mathbf{w} \quad (3.2.3)$$

Therefore, the procedure of estimating piecewise uniform translational motion in the 3-D Gabor domain can be summarized as:

(1) According to any pre-knowledge of the characteristics of the image sequence and resolution requirements of the application, design a proper 3-D Gabor representation. That is, determine the 3-D Gabor basis functions as well as the six st/stf sampling rates. The analysis framework is built by filling the basis function matrix $\mathbf{G}$ in equation (2.3.8).

(2) Input the signal vector $\mathbf{f}$ in equation (2.3.8), perform the 3-D Gabor transform to calculate the Gabor coefficients. Group the 3-D Gabor coefficients based on their spatiotemporal locations. That is: put all the coefficients with the same spatiotemporal index into a distinct group so that the coefficients in the group contain the local frequency information of that spatiotemporal region. Global statistics such as average magnitude or median magnitude of all coefficients can also be computed in this step.

(3) In each local spatiotemporal group, perform uniform translational motion estimation as in equation (3.2.3). To enhance computational efficiency, we first examine the magnitudes of the 3-D Gabor coefficients, removing those coefficients whose magnitudes are much smaller than the average before filling the frequency location matrix $\mathbf{A}$ and $\mathbf{w}$.

(4) Repeat (3) for every spatiotemporal coefficient group.

3.3 Performance Analysis of Motion Estimation in the 3-D Gabor Domain

Because the performance of image processing methods are highly application oriented and associated with human visual perception (which is not fully understood), it is very hard to develop a theoretical performance analysis systematically. One practical approaching is to construct some typical image sequences in which the facets of performance of interest are emphasized. By numerically comparing these ground truth motion fields with the estimation results, we can grade the quality of the estimation method and predict its behavior to some extent.

3.3.1 Image Sequences for Experiments

For the purpose of this research, three image sequences are constructed.

The first image sequence has 48 frames with 256×256 pixels each and with grey levels in the range 0 to 255. It contains two squares with sinusoidal texture. They are undergoing uniform translational motion on a sinusoidal texture background. Fig(3.3) shows two frames from the sequence together with their true motion fields. The upper square (80×80 pixels in size) is moving from right to left with velocity 2 pixels/frame. The lower square (also 80×80 pixels in size) is moving towards the upper left corner at speed of 1 pixel/frame in the y direction and 2 pixels/frame in the x direction. The sinusoidal texture of the background and the squares have frequencies of $\frac{\pi}{4}$ and $\frac{\pi}{2}$ respectively. This image sequence not only satisfies the uniform translational motion assumption, but also has explicit spatial frequency components to calculate the plane slope. We use it to test the performance of our algorithm under the best possible condition.

The second image sequence is a wheel with isotropic sinusoidal texture undergoing uniform rotational motion on a black background. The size of the image sequence is

(a) One frame of the two-square image sequence

(b) True motion vectors on 32×32 spatial centers

(c) True motion vectors on 16×16 spatial centers

Figure 3.3: Two-square image sequence and its true motion Field

$256 \times 256 \times 48$ with grey levels from 0 to 255 . Fig(3.4) shows one of the frames from the sequence together with its true motion field. The center of the wheel is fixed at (120, 120) and the radius is also fixed at 100 pixels. The intensity of pixels along the 100-pixel radius is distributed in sinusoidal wave with 32 cycles. The wheel is rotating at an angular speed of $\frac{\pi}{128}$ /frame. This image sequence includes pixels moving in all directions. The velocity linearly increases from the center of the wheel to the border. The motion in this image sequence is well approximated by piece-wised translational motion under low rotating speed.

The estimation results also demonstrate whether or not the method is isotropic for different directions of velocity.

(a) One frame of the wheel image sequence.

(b) True motion vectors on 16×16 spatial centers.

(c) True motion vectors on 32×32 spatial centers.

Figure 3.4: “Wheel” image sequence and its true motion field.

The third sequence has three wheels with isotropic sinusoidal texture undergoing uniform rotational motion on a black background. The size of the image sequence is also $256 \times 256 \times 48$. Fig(3.5) shows one of the frames together with the true motion field. The center of the wheels are fixed at $(60, 128)$, $(150, 70)$ and $(150, 170)$. All the radii are 45 pixels. The intensity of pixels along the radius is distributed in a sinusoidal wave of 16 cycles. The wheels are rotating at an angular speed of $\frac{\pi}{100}$ /frame. This image sequence not

only includes pixels moving in different directions with different speeds, but also tests the localization property of the 3-D Gabor representation by showing the influence among the different neighboring motions in the border areas of adjacent wheels.

(a) One frame of the three-wheel image sequence.

(b) True motion vectors on 32×32 spatial centers.

Figure 3.5: “Three-Wheel” sequence and its true motion field.

In all the artificial image sequences, although the true motion of every pixel is known, the true motion field is displayed with the same spatiotemporal resolution as the 3-D Gabor framework under investigation. In each spatiotemporal region, the motion vector of the center pixel is displayed.

3.3.2 Performance Calibration

The aspect of the algorithm being evaluated in these tests include: the spatiotemporal resolution, the errors between estimated motion fields and the true motion fields, and the noise resistance ability.

Estimation Accuracy

The accuracy of estimation is measured by the errors between the motion vectors of the true motion field and the estimated motion field. Three metrics are utilized to quantify these errors: overall similarity is gauged by calculating an “inner product” field; angle error measures the absolute value of the angle difference between vectors; and relative magnitude error evaluates the error in estimating vector magnitude. From now on, we refer to the corresponding true motion vector and estimated motion vector as “vector pairs”.

The similarity of a vector pair is calculated in the x and y direction separately. Before computing the inner product, the two vectors are normalized by the larger one. If the true motion vector is (x_t, y_t) and the estimated motion vector is (x_e, y_e) , the normalized motion vectors are $(\frac{x_t}{\max(x_t, x_e)}, \frac{y_t}{\max(y_t, y_e)})$ and $(\frac{x_e}{\max(x_t, x_e)}, \frac{y_e}{\max(y_t, y_e)})$ for the true and estimated motion vector respectively. The similarity of the two vectors is calculated by computing the inner product of the normalized vectors. The result is within the range $[0, 1]$ from “totally different(orthogonal)” to “the same”. Overall similarity is estimated by the average of all vector pair similarities.

The angle error is simply the difference between the directions of the vectors. If the true motion has angle θ_t , the estimated motion has angle θ_e , the angle error is $|\theta_t - \theta_e|$. The overall angle error is the average over all the vector pairs. The value is within the range $[0, \pi]$. The smaller the value the better the performance.

Magnitude error must be a relative metric to provide a valid comparison for vector fields of different average magnitudes. Before calculating the error, the magnitudes of all the vectors in the true motion field are computed. The median V_{med} is chosen for normalization. If the true motion vector is (x_t, y_t) and estimated motion vector is (x_e, y_e) , the relative magnitude difference for the vector pair is defined as :

$$Error_{mag} = \frac{|\sqrt{x_t^2 + y_t^2} - \sqrt{x_e^2 + y_e^2}|}{V_{med}}. \quad (3.3.1)$$

The overall magnitude error is the average of all vector pairs.

Spatiotemporal Resolution

Based on the above accuracy measurements, performance in terms of spatiotemporal resolution can also be evaluated. By increasing the spatiotemporal resolution of the 3-D Gabor framework while examining the accuracy of the estimation results, we can find the spatiotemporal resolution of the method under “almost ideal” conditions: the “two-square” image sequence in the noise free case.

Noise Resistance Ability

Noise resistance can be evaluated by comparing motion field accuracy under different amounts of additive noise.

3.3.3 Performance Analysis for Experimental Results

(a)spatial domain critical sampling

with $\Delta l = 8$.

(b)Frequency domain critical

sampling with $\Delta\omega = \frac{\pi}{4}$.

Figure 3.6: A: Gabor basis function coverage along one dimension of the spatial and spatial frequency domains under critical sampling with $\alpha = 0.125$, $\Delta l = 8$.

(a) Spatial domain critical sampling
with $\Delta l = 16$.

(b) Frequency domain critical
sampling with $\Delta\omega = \frac{\pi}{8}$.

Figure 3.7: **B**: Gabor basis function coverage along one dimension of the spatial and spatial frequency domains under critical sampling with $\alpha = 0.0625$, $\Delta l = 16$.

The Two-square Image Sequence

Two 3-D Gabor representation schemes are compared on the two-square image sequence. The first one (scheme **A**) leads to estimation result shown in Fig(3.8). Its basis functions, with Gaussian envelope and $\alpha = 0.125$, have $32 \times 32 \times 6$ different spatiotemporal locations and $8 \times 8 \times 8$ different frequency bands in a 3-D critical sampling framework. Fig(3.6) shows the coverage of the Gaussian envelopes in a spatial dimension and in the corresponding spatial frequency domain. The 3-D Gabor coefficients are arranged by $32 \times 32 \times 8$ spatiotemporal groups within which the $8 \times 8 \times 8$ coefficients (with the same spatiotemporal index) are associated with different frequency bands. The other 3-D Gabor framework (scheme **B**) yields the result in Fig(3.9). The basis function set is indexed by $16 \times 16 \times 3$ spatiotemporal shifts and $16 \times 16 \times 16$ frequency bands. As the $\alpha = 0.0625$ is only half of the first case, the wider Gabor basis function in spatiotemporal domain provides proper coverage with more sparse spatiotemporal sampling. On the other hand, the frequency coverage is halved, which enhances the frequency resolution to $16 \times 16 \times 16$ bands in the local frequency domain. Fig(3.7) displays the coverage status in the 1-D case. The estimation result is calculated from the local frequency model of uniform translational

motion in every local $16 \times 16 \times 16$ Gabor coefficient group. The first 3-D Gabor scheme has better spatiotemporal resolution whereas the second one is stronger in distinguishing different frequency components.

(a) Motion estimation result.

(b) Corresponding error vectors.

Figure 3.8: Motion estimation results and errors for Gabor scheme A for the “two-square” sequence

(a) Motion estimation result.

(b) Corresponding error vectors.

Figure 3.9: Motion estimation results and errors on Gabor scheme B for the “two-square” sequence

For this image sequence, high performance is achieved in all three accuracy measurements. In the first column of Tables(3.1) and (3.2), we can see the overall similarity is higher than 99% and the angle and relative magnitude errors are negligibly small. The experiments validate the fundamental principle of the motion estimation methods in the 3-D Gabor domain. Under these almost ideal conditions, the local uniform translational

Table 3.1: Performance of Gabor scheme **A** for the “two-square” sequence.

blocks	Noise Free	$\sigma = 36$	$\sigma = 64$
overall similarity	0.9921	0.9821	0.9580
angle error	0.0015	0.0041	0.0131
magnitude error	0.0099	0.0247	0.0661

Table 3.2: Performance of Gabor scheme **B** for the “two-square” sequence

blocks	Noise Free	$\sigma = 36$	$\sigma = 64$
overall similarity	0.9852	0.9836	0.9790
angle error	0.0042	0.0043	0.0069
magnitude error	0.0176	0.0196	0.0254

motion is detected and estimated with high accuracy. In addition, the fact that both of the results (acquired from two different Gabor schemes) achieve high performance proves good consistency of the 3-D Gabor representation and the estimation method.

Comparing Fig(3.8) and Fig(3.9), the first scheme is more impressive because of its higher spatiotemporal resolution. It seems that the spatiotemporal resolution is safely enhanced by arranging more spatiotemporal centers and designing a narrower Gaussian envelope. Nevertheless, further experiments will show that $8 \times 8 \times 8$ spatiotemporal frequency bands are not sufficient to estimate more complex motion fields. The potential of the 3-D Gabor layout in scheme **A** is in fact quite limited.

Fig(3.10) shows examples of the noise contaminated image frames. The corresponding motion estimation results are illustrated in Fig(3.11) and Fig(3.12). The noise is Gaussian white noise with mean of 128 and variance of 36 and 64 respectively. The noise-free image sequence has 256 grey levels. After noise is added, the image sequence is normalized and scaled into 256 ([0,255]) gray levels again. As can be seen in Tables (3.1) and (3.2), the performance degrades quite slowly upon increasing noise.

From the data in the second and third columns of the tables we can see that the robustness of the motion estimation method is also excellent under the tested noise conditions. It is also noticeable that the second Gabor framework performs better than the first one. This is not only because of its higher frequency resolution, but also because the wider basis function envelope provides more robustness against short term interference. Together with the need for higher frequency resolution for more complex motion as mentioned previously, although scheme A has better spatiotemporal resolution, it is not used in the following experiments.

(a) $\sigma = 36$

(c) $\sigma = 64$

Figure 3.10: Gaussian noise contaminated “two-square” image sequence.

The Rotating Wheel Image Sequence

In this test, the critical sampling and over sampling approaches are compared. With higher sampling density, over sampling is more computationally expensive. However, its superior spatiotemporal localization and more orthogonal-like representation are useful features. Therefore, besides scheme B (the second critical sampling plan in the former experiment), an over sampling layout C is implemented. With respect to the 3-D Gabor representation designed in B, the number of basis functions along each spatial dimension

Figure 3.11: Motion estimation results and errors for Gabor Scheme A for noised “two-squared” sequence

is doubled by halving the distance between adjacent spatial centers. Fig(3.13) illustrates the coverage manner for the 1-D spatial and spatial frequency cases. One may notice that scheme C has substantially greater overlap in the spatiotemporal domain. In fact, the spatiotemporal resolution is enhanced because of more “sensors” for spatiotemporally locating the signal energy. Also, the 3-D Gabor domain in the over sampling case is an orthogonal-like framework. The Gabor auxiliary function found via the biorthogonal equation are similar to the basis functions, which ensures better st/stf localization properties.

(a) Motion estimation result under
Gaussian noise $\sigma = 36$.

(b) Corresponding error
vectors for (a).

(c) Motion estimation result under
Gaussian noise $\sigma = 64$.

(d) Corresponding error
vectors for (c).

Figure 3.12: Motion estimation results and errors for Gabor Scheme **B** for noised “two-square” sequence

Fig(3.14) and Fig(3.15) are the motion estimation results for critical sampling and over sampling respectively. The corresponding performance analysis data are shown in Table(3.3) and Table(3.4). Instead of the ideal uniform translational motion, the wheel image sequence contains velocities varying in magnitude and direction, for which only approximate piecewise uniform translational motion can be found. Since the velocity of the motion changes for each pixel in the wheel, pixels undertaking different motions contribute to the 3-D Gabor coefficients with the same spatiotemporal index. The estimate in each spatiotemporal region is the weighted average of the pixel motions. Consequently, it is not surprising that the errors are not so small for the “two-square” image sequence. Yet,

(a) Spatial domain over sampling

with $\Delta l = 8$.

(b) Frequency domain critical

sampling with $\Delta\omega = \frac{\pi}{8}$.

Figure 3.13: C: Gabor basis function coverage along one dimension of the spatial and spatial frequency domains under spatial over sampling with $\alpha = 0.0625$, $\Delta l = 8$

Table 3.3: Performance of Gabor scheme **B** for the “wheel” sequence

blocks	Noise Free	$\sigma = 36$	$\sigma = 64$
overall similarity	0.8833	0.8151	0.7900
angle error	0.0058	0.0076	0.0715
magnitude error	0.2260	0.2937	0.2951

the estimation results are quite promising. Although the relative magnitude error is not insignificant, we can see that the error is concentrated at the outer radius of the wheel. In fact, these spatiotemporal local regions contain both the moving and still pixels, which violate the uniform translational assumption of our algorithm. In this case, higher spatiotemporal resolution is required. We will develop algorithms for this issue in the next chapter. The second aspect revealed by these results is the superiority of the over sampling design. Qual-

Table 3.4: Performance of Gabor scheme **C** for the “wheel” sequence

blocks	Noise Free	$\sigma = 36$	$\sigma = 64$
overall similarity	0.9255	0.9239	0.9039
angle error	0.0079	0.0080	0.0103
magnitude error	0.1140	0.1171	0.1202

(a) Motion estimation result.

(b) Corresponding error vectors.

Figure 3.14: Motion estimation results and errors for Gabor Scheme **B** and the “wheel” sequence

(a) Motion estimation result.

(b) Corresponding error vectors.

Figure 3.15: Motion estimation results and errors for Gabor Scheme **C** and the “wheel” sequence

itatively and quantitatively, it outperforms the critical sampling case both in spatiotemporal resolution and noise resistance ability. Fig(3.16) shows the noise contaminated frames for the same noise conditions as in the last experiment. The estimation results are exhibited in Fig(3.17) and Fig(3.18) for $\sigma = 36$ and $\sigma = 64$ respectively. Even in the heavy noise case, the motion estimation method based on the over sampled 3-D Gabor representation achieves overall similarity of 90% and angle error slightly over 0.01 rad.

This experiment verifies that the piecewise uniform translation model can be applied in the 3-D Gabor domain. It also shows that the over sampled 3-D Gabor representation performance is superior to critical sampling in spatiotemporal resolution and noise

(a) $\sigma = 36$

(c) $\sigma = 64$

Figure 3.16: Gaussian noise contaminated “wheel” image sequence

resistance. Another conclusion drawn from this test is that with a frequency resolution of $16 \times 16 \times 16$, the method is isotropic for motion vector of various directions within acceptable errors. Finally, the consistency between the two Gabor schemes confirms the that our motion estimation algorithm is reasonably robust.

The Three Rotating Wheels Image Sequence

According to previous experiments, the 3-D Gabor representation scheme C was proven to be a wise choice with respect to spatiotemporal resolution, accuracy and noise resistance. The “Three-Wheel” image sequence is more challenging: the directions and

Figure 3.17: Motion estimation results and errors for Gabor Scheme **B** and the noisy “wheel” sequence

speeds of neighbor pixels change more quickly in spatiotemporal domain and opposing motions are close to each other along the circumferences of the wheels. The estimation procedure is the same as before and the results are displayed in Fig(3.19) and Table(3.5). Unexpectedly, the performance is even better than with the “rotating wheel” image sequence. Although relatively large errors appear along the border of wheels, the overall similarity, angle error and velocity magnitude error are better than obtained with the “rotating wheel” sequence. we may explain this by their relative slower absolute motion speed due to shorter radius. As we have mentioned before, since the major error comes from the violation of uniform motion models along circumference region, the higher speed of

Figure 3.18: Motion estimation results and errors for Gabor Scheme C and the noisy “wheel” sequence

the “rotating wheel” results in more “contrast” between motion and still regions, which leads to more errors. On the other hand, the better result confirms the good localization property of the over sampled 3-D Gabor representation. In terms of noise resistance, noise contaminated frames and estimation results are illustrated in Fig(3.20) and Fig(3.21). The degradation in performance increases only slightly with increasing noise, which substantiates the superior noise resistance of motion estimation methods developed in the 3-D Gabor domain.

Table 3.5: Performance of Gabor scheme C for the “three-wheel” sequence

blocks	Noise Free	$\sigma = 36$	$\sigma = 64$
overall similarity	0.9642	0.9589	0.9411
angle error	0.0054	0.0088	0.0066
magnitude error	0.0812	0.097	0.104

(a) Motion estimation result.

(b) Corresponding error vectors.

Figure 3.19: Motion estimation results and errors for Gabor Scheme C and the “three-wheel” sequence.

3.4 Conclusion

The investigations in this chapter show that the 3-D Gabor representation is well suited as a basis for motion estimation. As an analysis framework displaying joint st/stf information dynamically, the 3-D Gabor representation enables the exploration of frequency domain information with spatiotemporal resolution. We have shown that by determining the parameters of the frequency domain model of uniform translational motion, the piecewise uniform translational motion can be estimated. This approach to motion estimation has considerable spatiotemporal resolution, and is superior in noise resistance.

(a) $\sigma = 36$.

(c) $\sigma = 64$.

Figure 3.20: Gaussian noise contaminated “three-wheel” image sequence.

(a) Motion estimation result under
Gaussian noise $\sigma = 36$.

(b) Corresponding error
vectors of (a).

(c) Motion estimation result under
Gaussian noise $\sigma = 64$.

(d) Corresponding error
vectors of (c).

Figure 3.21: Motion estimation results and errors for Gabor Scheme C and the noisy “three-wheel” sequence.

Chapter 4

Dense Motion Field Estimation in the 3-D Gabor Domain

4.1 Motivation

In chapter 3, the piecewise uniform translational motion model was utilized to approximate more complex motion. Image sequences are often divided into a number of spatiotemporal regions so that relatively simple motion models can be applied within a local region. The goal is to avoid the computational complexity associated with more complex motion models. The most widely used scheme is the block-based partition, with disjoint blocks covering the entire spatiotemporal span of the image sequence. A disadvantage of this approach, however, is that multiple motions may exist within a given block. This can lead to erroneous motion estimation. In addition, as the required spatiotemporal resolution is application dependent and sometimes varies within an application, a pre-defined partition is not always appropriate. Some multi-resolution schemes address these issues to some extent[32]. For applications such as dynamic segmentation, flow analysis, and video signal restoration (in which the motion information of very high spatiotemporal resolution is desired), dense motion estimation is a better choice[16]. For digital image sequences, dense

motion field usually represents the velocity vector of every pixel (although sub-pixel resolution is sometimes required). Compared with block-based schemes, a dense motion field estimation provides more accurate border information for moving objects and a smoother velocity representation, which enhances the utility of motion information for further applications.

The aim of designing motion estimation methods based on the 3-D Gabor representation is to achieve higher spatiotemporal resolution, less estimation error and more robustness against noise contamination. For the algorithms discussed in chapter three, robustness is improved by exploiting motion models in the spatiotemporal-frequency domain. If the motion in a region is well approximated as uniform translation motion, a highly accurate motion estimation can be achieved if the underlying representation has sufficient frequency resolution. I.e, the number of spatiotemporal-frequency bands must be high enough to locate the slope of motion induced plane with sufficient accuracy. Furthermore, the bandwidth of the 3-D Gabor basis functions must also be small enough to avoid too much overlaps between adjacent frequency regions. Under these circumstances, because of the trade off between the resolution of spatiotemporal and spatiotemporal-frequency domains, spatiotemporal resolution may be insufficient. Motion estimates over large spatiotemporal regions also invalidate the assumption of uniform translational motion. Although over sampling can improve the situation to some degree, there are practical limitations to the over sampling rate. The Gabor basis function vectors become highly dependent, so that calculating the 3-D Gabor coefficients by solving linear equations becomes numerically unstable. Therefore, the motion estimation algorithm established in chapter three is limited in its spatiotemporal resolution. The objective of this chapter is to overcome this limitation while preserving the advantages of the 3-D Gabor domain approach.

Although there is no analysis framework that can overcome the uncertainty constraint, this does not imply that information on frequency content is available only at spa-

tiotemporal locations corresponding to basis function centers. The frequency information at the positions neighboring a particular spatiotemporal position can be decoded from the interrelations among the 3-D Gabor coefficients near its location. Makeing good use of this "hidden" information can enhance motion estimation performance greatly. For example, selecting a specific frequency index (u, v, w) in equation (2.4.3) and computing the partial sum over all (i, j, k) yields an image sequence of dimension $M \times N \times L$ corresponding to the selected frequency. Following this idea, we will construct a novel algorithm that keeps the same frequency resolution as before while achieving dense motion field estimation. The starting point of this development is the concept of partial reconstruction.

4.2 Partial Reconstruction

Notice that the discrete 3-D Gabor representation equation (2.4.3) can be arranged as:

$$f(m, n, l) = \sum_{u,v,w} \left\{ \sum_{i,j,k} C_{i,j,k,u,v,w} g_{i,j,k,u,v,w}(m, n, l) \right\} \quad (4.2.1)$$

where u, v , and w are indices of frequency bands and i, j and k are indices of spatial and temporal position. We can see that the summation inside the curly braces includes all the Gabor coefficients with different spatiotemporal locations but the same frequency indices. This can be thought of as the partial reconstruction of the image sequence in that frequency band. Each partial reconstruction result is a sub-band image sequence having the same dimensions as the original. For each pixel in the sub-band image sequence, the phase and magnitude information of the 3-D Gabor coefficients around that pixel are combined to form a specific frequency band. For any given pixel, collecting values at its position from every sub-band image sequence forms the "local-instant" frequency domain for that pixel.

4.3 Calculation of the Dense Motion Field

For any pixel at spatiotemporal coordinates (m, n, l) , the “local instant” frequency information at frequency band $\mathbf{w} = (u, v, w)$ is:

$$F_{m,n,l}(u, v, w) = \sum_{i,j,k} C_{\mathbf{d},\mathbf{w}} g_{\mathbf{d},\mathbf{w}}(m, n, l) \quad (4.3.1)$$

where $\mathbf{d} = [i\Delta_m, j\Delta_n]$ and $\mathbf{w} = [u\Delta f_m, v\Delta f_n]$ are the spatial and spatial frequency locations of the corresponding Gabor basis functions. Applying this on every combination of (u, v, w) , a complete frequency domain can be formed for the pixel at (m, n, l) . In practice, to combine the contribution of all Gabor basis functions for a given pixel is too computationally expensive. Due to the locality of the Gabor basis functions, however, the influences of basis functions far away from a pixel are very small. Therefore, we only include the contributions of the 27 nearest spatiotemporal neighbor coefficients in the partial reconstruction.

The process can be briefly described as:

- 1) Find the 3D Gabor representation of the original image sequence.
- 2) For the selected pixel and frequency band, select the coefficients of the same band at the 27 nearest spatiotemporal neighbors. Compute the partial reconstruction as:

$$F_{m,n,l}(u, v, w) = \sum_{i=-1,j=-1,k=-1}^{i+1,j+1,k+1} C_{\mathbf{d},\mathbf{w}} g_{\mathbf{d},\mathbf{w}}(m, n, l) \quad (4.3.2)$$

where (i, j, k) is the index of a Gabor basis function whose spatiotemporal neighborhood contains the point (m, n, l) . The $\mathbf{d}$ and $\mathbf{w}$ are the same as in equation (4.3.1).

- 3) Repeat 2) for every 3D frequency band.
- 4) Conduct frequency domain motion estimation using a piecewise translational model and repeat 2) to 4) for every pixel in the image sequence.

4.4 Performance Analysis of Dense Motion Field Estimation Method

Performance evaluation for dense motion field estimation follows the same procedure as in chapter 3. Some image sequences that can be used to demonstrate motion estimation of high spatiotemporal resolution are constructed. By numerically comparing the ground truth motion field with the estimation results pixel by pixel, we can estimate the algorithm's performance in terms of accuracy, robustness and spatiotemporal resolution.

4.4.1 Image Sequences for Experiments

For the purpose of this research, two image sequences were constructed:

The first image sequence has dimensions of $256 \times 256 \times 48$ and grey values within the range $[0,255]$. It is composed of two triangular regions undergoing uniform translational motion (Fig 4.1-a). The sinusoidal texture is $\omega = \frac{\pi}{4}$ in both x and y directions. (Fig4.1-b) is the true motion field for the 8×8 pixel block in the center of Fig(4.1-a) (at the boundary between the two triangles). The upper-left region moves at a speed of 1 pixel/frame in both the x and y directions, whereas the lower-right region moves in the opposite direction at the same speed. This sequence has a well defined discontinuity in motion, so that the spatiotemporal resolution of the motion estimate can be examined.

The second sequence also has dimensions of $256 \times 256 \times 48$ and grey values within the range $[0,255]$. There are three rotating wheels undergoing uniform rotational motion (Fig 4.2). The wheels are centered at $(60, 128)$, $(150, 70)$ and $(150, 170)$ respectively and have the same radius of 45 pixels. The intensity of pixels is distributed along the radius in a sinusoidal wave of 16 cycles. All the wheels are rotating at an angular speed of $\frac{\pi}{100}$ /frame. This image sequence includes motions in all directions with the speed increasing from

the center of each wheel. It is used to test the resolution, accuracy and robustness of the algorithm for different orientations, speeds and neighborhood conditions.

4.4.2 Performance Calibration

The performance evaluation of the dense motion field estimation algorithm is conducted at the pixel level. Since the true instantaneous velocity of every pixel is known in these artificial image sequences, the evaluation of the estimation quality consists of a numerical comparison between the reference dense motion field and the estimation results, pixel by pixel. Aspects of the estimate that will be evaluated include: the spatiotemporal resolution, the accuracy of motion vector fields, and noise resistance ability.

Spatiotemporal Resolution

Although the motions are estimated at pixel resolution, the true resolution depends on how well the neighboring motions are distinguished. The true spatiotemporal resolution of the algorithm is evaluated by the width of the ambiguous region (indicated by the number of pixels with large error) between the two moving texture fields in Fig(4.1-a). The narrower the ambiguous region, the better the spatiotemporal resolution of the algorithm.

Estimation Accuracy

Similar to the approach we defined in chapter 3, the estimation accuracy is measured by comparing the motion vectors of the true motion field and the estimated motion field. The same three metrics are measured pixel by pixel in the regions of interest. As the dense motion field of an image sequence is highly non-stationary and contains large amount of data, evaluating performance on the whole spatiotemporal domain is neither efficient nor constructive. In these experiments, the “overall similarity”, “average angle

error” and “average magnitude error” are calculated only in selected local regions, such as at the boundaries between regions with different motions, the centers of the wheels, or regions exhibiting uniform translational motion. As before, we refer to the true motion vector and the corresponding estimated motion vector as a “vector pair”.

The first metric comes from the normalized inner product of every vector pair in a given region. This evaluates the overall similarity of the vector pair. As in chapter 3, before computing the inner product, the two vectors are normalized by the larger one. That is, if the true motion vector is (x_t, y_t) and the estimated motion vector is (x_e, y_e) , the normalized motion vectors are $(\frac{x_t}{\max(x_t, x_e)}, \frac{y_t}{\max(y_t, y_e)})$ and $(\frac{x_e}{\max(x_t, x_e)}, \frac{y_e}{\max(y_t, y_e)})$ for the true and estimated motion vector respectively. Then, inner products of the normalized vectors are computed in the x and y directions separately. The result is within the range $[0, 1]$ from “totally different (orthogonal)” to “the same”. Overall similarity is estimated by the average of all vector pair similarities.

The angle error is simply the difference between the directions of the vectors. If the true motion has angle θ_t , the estimated motion has angle θ_e , and the angle error is $|\theta_t - \theta_e|$. The overall angle error is the average over all the vector pairs. The value is within the range $[0, \pi]$.

Magnitude error must be a relative metric to provide a valid comparison for vector fields of different average magnitudes. Before calculating the error, the magnitudes of all the vectors in the true motion field are computed. The median V_{med} is chosen for normalization. If the true motion vector is (x_t, y_t) and estimated motion vector is (x_e, y_e) , the relative magnitude difference for the vector pair is defined as :

$$Error_{mag} = \frac{|\sqrt{x_t^2 + y_t^2} - \sqrt{x_e^2 + y_e^2}|}{V_{med}}. \quad (4.4.1)$$

The overall magnitude error is the average of all vector pairs.

Noise Resistance

Noise resistance is evaluated by comparing motion field accuracy corresponding to different amounts of additive noise.

4.5 Performance Analysis for Experimental Results

Experiments on the “Triangle” Image Sequence

Two schemes of 3-D Gabor representation were implemented for testing with the “translational triangle” sequence. The basis functions of the first 3-D Gabor framework have Gaussian envelopes spanning approximately 8 pixels in the spatiotemporal domain. Under critical sampling, there are $8 \times 8 \times 8$ local frequency coefficients at each of the $32 \times 32 \times 6$ spatiotemporal locations. The second layout uses 3-D Gabor basis functions whose spatiotemporal span is double of the previous one. As the result, their bandwidth in the frequency domain is halved. $16 \times 16 \times 16$ frequency divisions can be allocated without too much overlap between the functions. The spatiotemporal domain is sampled at $32 \times 32 \times 3$ spatiotemporal positions. Since the number of Gabor coefficients is twice as that of the original signal, it is an over sampled.

Fig(4.1-c) and Fig(4.1-d) are the estimation results of the same region as Fig(4.1-b) for the 3-D Gabor critical sampling and over sampling schemes respectively. We can see that both of them achieve very high spatiotemporal resolutions. In the critical sampling case, the ambiguous region is 4 pixels in width, and the over sampling case has a region of approximately 6 pixels wide. The first design has Gabor basis functions with narrow span in the spatiotemporal domain, whereas the second design has superior frequency localization properties. We have seen experimentally that both factors are important in achieving high spatiotemporal resolution. However, since the $8 \times 8 \times 8$ frequency domain (although it works

well in this simple case) is not enough to distinguish motions of different orientations, the spatial over sampling framework is used in the following tests.

(a) A typical frame from the translating triangle sequence.

(b) True dense motion field at the center of the frame.

(c) Estimated motion field in the critical sampling case.

(d) Estimated motion field in the over sampling case.

Figure 4.1: Dense motion field estimation for translational triangles sequence

The high accuracy and good consistency of the results obtained in the experiments demonstrates the feasibility of dense motion field estimation based on the 3-D Gabor representation. Compared with the algorithm in chapter 3, it provides much higher spatiotemporal resolution in motion estimation, and with high accuracy. To the author's knowledge, such resolution for regions exhibiting opposite motions have not been reached by any published spatiotemporal domain method.

Experiments on the “Three-Wheel” Image Sequences

The “three-wheel” image sequence exhibits a wide range of motions of different orientations and speeds. It also involves adjacent regions with different motions. A sample frame from the sequence (with additive noise) and the true motion field are shown in Fig (4.2-a). Three local spatiotemporal regions are used for evaluating the results with this sequence. One is the center of a wheel, one close to the center and one far away from the center. The Gabor transform uses $32 \times 32 \times 3$ spatiotemporal over sampling with $16 \times 16 \times 16$ local frequency coefficients for each spatiotemporal local region. The estimation results can be seen in Fig(4.3-b,d,f) compared to the true dense motion field in Fig(4.3-a,c,e). Table (4.1) shows the corresponding performance analysis results. Considering that the motion vectors in the figures are associated with each pixel, the dense motion field estimation algorithm does boost the spatiotemporal resolution significantly. It can be observed that: the farther away the region of interest is from the center of the circle, the better the performance. This is as expected, since the regions close to the center exhibit a broader range of conflicting velocities, leading to less accurate estimates.

Tables (4.2),(4.3), and (4.4) demonstrate the performance of this method on the rotating wheels sequence under Gaussian white noise of different variances. The means of the noise signals are fixed at 128 and the variances are 36, 50 and 64 respectively. After noise is added, the image sequence is normalized and scaled into 256 ([0,255]) gray levels. For the region rather far from the center, noise resistance is very good. This is because the noise manifests in the local frequency domain as a pseudo uniform background of small magnitude coefficients. Since only 10% of the largest Gabor coefficients are used in the weighted LMS estimation, interference of noise is greatly reduced. However, in the center portion of each wheel, as the pixel moves very slowly, the relative SNR for motion estimation is quite poor. This can lead to increased degradation upon noise contamination.

Table 4.1: Performance evaluation at different location

blocks	Fig 4.3-b	Fig 4.3-d	Fig 4.3-f
overall similarity	0.7724	0.8863	0.9987
angle error	0.1665	0.0537	0.0382
magnitude error	0.2354	0.0154	0.0307

Table 4.2: Performance evaluation with noise, $\sigma = 36$.

blocks	Fig 4.3-b	Fig 4.3-d	Fig 4.3-f
overall similarity	0.6082	0.8935	0.9972
angle error	0.3761	0.0513	0.0458
magnitude error	0.2580	0.0947	0.0291

4.6 Conclusion

Dense motion field estimation developed using the 3-D Gabor representation achieves high spatiotemporal resolution with accurate motion estimates and good noise resistance. The algorithm calculates the local-instant frequency domain of any pixel by partial reconstruction, a process which combines the information in the neighboring Gabor coefficients for each pixel. The spatiotemporal resolution of this method is controlled mainly by the spatiotemporal span of the Gabor basis functions and the localization property of the Gabor representation. The noise resistance is quite good for a method with pixel-level spatiotemporal resolution.

Table 4.3: Performance evaluation with noise, $\sigma = 50$.

blocks	Fig 4.3-b	Fig 4.3-d	Fig 4.3-f
overall similarity	0.5647	0.8304	0.9871
angle error	1.0383	0.0418	0.04580
magnitude error	0.4358	0.1630	0.0598

Table 4.4: Performance evaluation with noise $\sigma = 64$.

blocks	Fig 4.3-b	Fig 4.3-d	Fig 4.3-f
overall similarity	0.5450	0.8527	0.9936
angle error	1.4488	0.0603	0.0603
magnitude error	0.5164	0.1353	0.0550

(a) noise contaminated
rotating wheels.

(b) True motion field
of the Rotating Wheels.

Figure 4.2: Artificial Image Sequences of the Rotating Wheels.

(a) True dense motion field
of the center part.

(b) Estimated motion field
of the center part.

(c) True dense motion field
of the near center part.

(d) Estimated motion field
of the near center part.

(e) True dense motion field
of the part far from center.

(f) Estimated motion field
of the part far from center.

Figure 4.3: Dense Motion Field Estimation of rotating wheels.

Chapter 5

Detection and Estimation of Rotational Motions in the 3-D Gabor Domain

5.1 Motivation

Rotational motion is frequently encountered in practice. The ability to detect rotating objects and estimate their motion parameters such as rotation center and angular speed can greatly facilitate scene interpretation and other tasks. Compared with spatiotemporal domain approaches such as those based on affine motion models[42], frequency domain approaches have many advantages for global rotational motion. As a motion with a large scale of regularity, global rotation fits the global nature of the Fourier transform and has concise models in the frequency domain. This is especially the case for uniform rotational motion. It is well known that for an image sequence with objects rotating uniformly around the spatial center, the projection of its 3-D frequency distribution on the temporal frequency axis is an evenly spaced impulse sequence. The interval between the adjacent impulses relates to the angular speed of the rotation[27]. Another attractive advantage of the frequency domain methods is noise resistance. The global rotation with intensity varying on a large scale is distributed mainly in the low spatiotemporal-frequency band, which

is not very susceptible to temporal and local interference. In the spatiotemporal domain, however, we cannot expect an analogous property. As a result, a number of methods for detecting and estimating global rotational motion have been developed in the Fourier domain. For example, in [28][29], estimation methods are developed in the 3-D Fourier domain for global rotational and rotational-translational motions. However, if the rotational motion is not global or more than one rotating object is present in an image sequence, global frequency analysis mixes their motion signatures and causes difficulties in estimation.

In the 3-D Gabor domain, the joint st/stf characteristics of signals are made explicit. This makes it possible to utilize local frequency information for rotational motion estimation. If we can find a feasible rotational motion model in the 3-D Gabor domain. The existing frequency domain estimation methods can be reformulated and utilized for localized rotational motion, and for different rotational objects in an image sequence. Although the 3-D Gabor coefficients carrying local frequency information can be organized into the same structure as the Fourier domain, there are differences in the energy distributions expected for a given motion. It is therefore useful to explore the mathematic form of rotational motion in the 3-D Gabor domain.

5.2 The 3-D Gabor Representation in the Polar Coordinates

Because the rotational motion is easier to manipulate in polar coordinates, our study is based on the 3-D Gabor representation in the “spatial polar and temporal Cartesian (SPTC)” coordinates. Under this coordinates system, an image sequence can be expressed as $f(r, \theta, t)$, in which r is the radius to the origin, θ is the angle from an angular reference and t is the temporal position. The 3-D Gabor representation with basis functions in the

same coordinate system is established as:

$$f(r, \theta, t) = \sum_{r_d, \theta_d, t_d, \omega_r, \omega_\theta, \omega_t} C_{r_d, \theta_d, t_d, \omega_r, \omega_\theta, \omega_t} \mathbf{g}_{r_d, \theta_d, t_d, \omega_r, \omega_\theta, \omega_t}(r, \theta, t) \quad (5.2.1)$$

The image sequence $f(r, \theta, t)$ is expressed as the linear combination of a set of Gabor basis functions:

$$\mathbf{g}_{r_d, \theta_d, t_d, \omega_r, \omega_\theta, \omega_t}(r, \theta, t) = \hat{g}(r - r_d, \theta - \theta_d, t - t_d) e^{j(r\omega_r + \theta\omega_\theta + t\omega_t)} \quad (5.2.2)$$

where the ω_r and ω_θ are polar spatial frequency shifts of the basis functions, and the $\hat{g}(r, \theta, t)$ is a 3-D Gaussian window in SPTC coordinates. In Cartesian coordinates, this window is well known as:

$$\hat{g}(x, y, t) = \frac{1}{(2\pi)^{\frac{3}{2}} \sigma_x \sigma_y \sigma_t} \exp\left(-\frac{\left(\frac{x}{\sigma_x}\right)^2 + \left(\frac{y}{\sigma_y}\right)^2 + \left(\frac{t}{\sigma_t}\right)^2}{2}\right) \quad (5.2.3)$$

where σ_x , σ_y and σ_t are the variance parameters along the spatiotemporal dimensions. Although the expression in spatial polar coordinates can be found from the spatial Cartesian coordinate version by substituting $r \sin \theta$ for x and $r \cos \theta$ for y , in general the conversion of σ_x and σ_y to r and θ coordinates is not straightforward. Here, we add the constraint that $\sigma_x = \sigma_y$, which makes the 3-D Gaussian envelope have a round cross section in the $x - y$ plane. Thus the σ_x and σ_y can be replaced by σ_r so that:

$$\begin{aligned} \hat{g}(r - r_d, \theta - \theta_d, t - t_d) = & \frac{1}{(2\pi)^{\frac{3}{2}} \sigma_r^2 \sigma_t} \exp\left(-\left[\left(\frac{r \sin \theta - r_d \sin \theta_d}{\sigma_r}\right)^2 \right. \right. \\ & \left. \left. + \left(\frac{r \cos \theta - r_d \cos \theta_d}{\sigma_r}\right)^2 + \left(\frac{t - t_d}{\sigma_t}\right)^2\right]/2\right) \end{aligned} \quad (5.2.4)$$

The Gaussian window is localized both in the spatial and frequency domains. As a result, the basis obtained by shifting and modulating (r_d, θ_d, t_d) and $(\omega_r, \omega_\theta, \omega_t)$ (the spatial and spatial frequency polar coordinates and temporal Cartesian coordinates of the st/stf centers) can capture the dynamic joint st/stf information carried by individual moving objects. It is obvious that this basis function set is just the polar coordinates expression of the Cartesian

one. The st/stf shifts of the 3-D Gabor basis functions are not distributed along radii and across angles. The reason we keep the “compatibility” of the representation in the two coordinate systems will be explained in the next paragraph. In biological visual modelling, the polar Gabor representation is suggested in pure polar coordinates with σ and polar sampling space changes along radii. The model fits the attention-based visual perception and visual coding behaviors quite well. However, it is not easy to apply it in motion estimation because of implementing the Gabor expansion polar coordinates and of establishing polar motion models.

The 3-D polar Gabor expansion can be calculated by computing the inner product between the signal and the auxiliary functions $\gamma(r, \theta, t)$, which are biorthogonal to the Gabor basis functions.

$$C_{r_d, \theta_d, t_d, \omega_r, \omega_\theta, \omega_t} = \int_r \int_\theta \int_t f(r, \theta, t) \gamma_{r_d, \theta_d, t_d}(r, \theta, t) e^{j(r\omega_r + \theta\omega_\theta + t\omega_t)} dr d\theta dt \quad (5.2.5)$$

An analysis method to find the auxiliary functions for any given Gabor basis function set does not yet exist. In chapter3, a numerical solution was found for discrete signals by solving linear equations in Cartesian coordinates. In this chapter, we show that under some conditions, it is possible to detect and estimate the parameters of rotational motion models deduced in the SPTC coordinates, with the existing 3-D Gabor expansion in Cartesian coordinates.

5.3 Rotational Motion Model in the 3-D Gabor Domain

Rotational motion is defined in the spatiotemporal domain as:

$$r(t) = r \quad (5.3.1)$$

$$\theta(t) = \theta_0 + D(t) \quad (5.3.2)$$

$r(t)$ and $\theta(t)$ are the polar coordinates of the pixels belonging to the rotational object. θ_0 is the phase angle of a pixel in the first frame, $\theta(t)$ is the phase angle of the pixel at frame t , and $D(t)$ is the time dependent phase angle displacement.

In this way, if the intensity distribution of the rotational object in the first frame is denoted as $I_0(r, \theta_0)$, after time t it is rotated to $I(r, \theta(t))$. Assuming the intensity of the object does not change with time and substituting from equation(5.3.1), we have:

$$I(r, \theta(t)) = I_0(r, \theta_0) = I_0(r, \theta(t) - D(t)) \quad (5.3.3)$$

For convenience, we replace $\theta(t)$ with θ . According to the definition of the polar coordinate Gabor representation, the 2 D Gabor representation of the rotational object in the first frame is:

$$C_0(r_d, \theta_d, \omega_r, \omega_\theta) = \int_0^{+\infty} \int_{-\pi}^{+\pi} I_0(r, \theta) \gamma_{r_d, \theta_d}(r, \theta) \exp j(r\omega_r + \theta\omega_\theta) dr d\theta \quad (5.3.4)$$

The subscript 0 refers to the first frame. (r_d, θ_d) and $(\omega_r, \omega_\theta)$ are the spatial and spatial frequency centers of the Gabor auxiliary functions. Since the integral kernel $I_0(r, \theta) \gamma_{r_d, \theta_d}(r, \theta)$ is a periodic function of θ , it can be expressed as a Fourier series:

$$I_0(r, \theta) \gamma_{r_d, \theta_d}(r, \theta) = \sum_{n=-\infty}^{\infty} P_n(r) e^{jn\theta} \quad (5.3.5)$$

where

$$P_n(r) = \frac{1}{2\pi} \int_{-\pi}^{+\pi} I_0(r, \theta) \gamma_{r_d, \theta_d}(r, \theta) e^{-jn\theta} d\theta \quad (5.3.6)$$

Thus, equation(5.3.4) can be written as:

$$\begin{aligned} C_0(r_d, \theta_d, \omega_r, \omega_\theta) &= \int_0^{+\infty} \int_{-\pi}^{+\pi} \sum_{n=0}^{\infty} P_n(r) e^{jn\theta} e^{j(r\omega_r + \theta\omega_\theta)} dr d\theta \\ &= \sum_{n=-\infty}^{\infty} \int_0^{+\infty} \int_{-\pi}^{+\pi} P_n(r) e^{jn\theta} e^{j(r\omega_r + \theta\omega_\theta)} dr d\theta \\ &= \sum_{n=-\infty}^{\infty} L_n(r_d, \theta_d, \omega_r, \omega_\theta) \end{aligned} \quad (5.3.7)$$

We express the $\int_0^{+\infty} \int_{-\pi}^{+\pi} P_n(r) e^{jn\theta} e^{j(r\omega_r + \theta\omega_\theta)} dr d\theta$ as $L_n(r_d, \theta_d, \omega_r, \omega_\theta)$ for convenience.

In appendix B, we show that in the 3-D Gabor domain based on the SPTC coordinates, the mathematical signature of rotational motion is:

$$C(r_d, \theta_d, t_d, \omega_r, \omega_\theta, \omega_t) = \sum_{n=-\infty}^{+\infty} L_n(r_d, \theta_d, \omega_r, \omega_\theta) \mathbf{G}_t[e^{-jnD(t)}](t_d, \omega_t) \quad (5.3.8)$$

where the $\mathbf{G}_t$ is temporal Gabor transform.

5.4 Detection and Estimation of the Localized Uniform Rotational Motion in the 3-D Gabor Domain

For an object undergoing uniform rotational motion, its time dependent displacement is $D(t) = at$ with a the rotational speed in *rad/second*. The 3-D Gabor expansion of the image sequence depicting this object is:

$$\begin{aligned} C(r_d, \theta_d, t_d, \omega_r, \omega_\theta, \omega_t) &= \sum_{n=-\infty}^{+\infty} L_n(r_d, \theta_d, \omega_r, \omega_\theta) \mathbf{G}_t[e^{-jnD(t)}](t_d, \omega_t) \\ &= \sum_{n=-\infty}^{+\infty} L_n(r_d, \theta_d, \omega_r, \omega_\theta) \mathbf{F}_t[\gamma_{t_d}(t) e^{-jnD(t)}](\omega_t); \end{aligned} \quad (5.4.1)$$

$\mathbf{G}_t$ and the $\mathbf{F}_t$ are the temporal Gabor transform and the Fourier transform respectively. In the critically sampled case, the auxiliary function γ_{t_d} is not st/stf localized and the waveform is very different from the corresponding Gabor basis function. As the sampling rate rises, the auxiliary function will approach the Gabor basis function both in waveform and in st/stf localization. The Fourier transform of $\gamma_{t_d}(t)$ for critical sampling and over sampling by a rate of 2 are shown in Fig(5.1). In the over sampling case, the envelopes of the auxiliary functions are similar to Gaussian envelopes, which are also concentrated in the frequency domain. If the variance of the Gaussian envelope is not small (which means the 3-D basis function has a large temporal span), the spectrum is like a δ function spread

around the original frequency point. As the modulation of e^{-jnat} effects a frequency shift, the model of uniform rotational motion is:

$$C(r_d, \theta_d, t_d, \omega_r, \omega_\theta, \omega_t) \approx \sum_{n=-\infty}^{+\infty} L_n(r_d, \theta_d, \omega_r, \omega_\theta) \delta(\omega_t + na) \quad (5.4.2)$$

Equation(5.4.2) indicates that if an object is undergoing uniform rotational motion, its 3-D Gabor coefficients occupy a sequence of evenly spaced positions on the temporal frequency axis. The interval between the adjacent positions is the rotational speed. On the digital frequency axis, the interval is normalized by the temporal sampling frequency. Fig(5.2) shows the spatially localized temporal frequency spectrum of two objects rotating at speeds of 8π rad/second and 4π rad/second respectively. The temporal sampling rate is 32 frames/second. The rotational motion is clearly displayed as impulse sequences with intervals $\frac{\pi}{4}$ and $\frac{\pi}{8}$.

To detect such motion in an image sequence, we construct an algorithm to evaluate the periodicity along the temporal frequency axis of each spatiotemporal region in the 3-D Gabor domain. The rotational speed can be estimated by finding the period in equation(5.4.2). The process can be briefly described as:

(1) Find the 3-D Gabor representation of the image sequence, and group the 3-D Gabor coefficients according to their spatiotemporal centers. From equation(5.4.2) and Fig (5.2) we can see that the motion model is a projection onto the temporal frequency axis and is independent of the spatial organization of the Gabor basis functions. Therefore, we use spatial Cartesian coordinates for computational convenience.

(2) Project the coefficients of each spatiotemporal region onto the temporal frequency axis.

(3) Evaluate the periodicity of the projection by examining the standard deviation of the intervals.

(4) For spatiotemporal locations in which the standard deviation is small enough, estimate the rotational speed by computing the mean of the intervals.

Fig(5.3) shows a $256 \times 256 \times 32$ image sequence with two rotating wheels at the upper left and right corners, with rotational speeds of $\frac{\pi}{4}$ rad/frame and $\frac{\pi}{8}$ rad/frame respectively. A square undergoing translational motion starts from the lower right corner and moves to the left. The algorithm detects the rotating objects and shows their rotational speeds in different intensities. The rotational speeds are estimated from the intervals between the impulses on the temporal frequency axis (shown in Fig(5.2)). As there are 32 sampling points in temporal frequency axis covering $2/\pi$, the interval of 2 refers to $\frac{\pi}{8}$ rad/frame ($\frac{1}{16}$ of $2/\pi$), and the interval of 4 refers to $\frac{\pi}{4}$ rad/frame ($\frac{1}{8}$ of $2/\pi$). However, we do not display the motion field here because the information about the direction of the rotation is contained in the phase of the 3-D Gabor coefficients. An algorithm for determining direction from phase is yet to be investigated. Note that since the algorithm is designed to detect the local rotational motion and estimate its rotational speeds the translational motion is invisible to it.

5.5 Conclusion

The 3-D Gabor representation has been presented as an analysis framework for the detection and estimation of localized multiple rotational motions in image sequences. Its joint st/stf analysis ability enables taking advantage of frequency domain approaches, with spatiotemporal localization. The method has no special requirements on spatial characteristics or textures of the objects.

In this approach, There are a number of areas for future work. The frequency resolution is not so fine as in global frequency analysis because of the trade-off between st/stf resolutions. In the digital frequency domain, low frequency resolution means a low sampling rate on the frequency axis. Therefore, rotations with a frequency signature falling between frequency samples may not be matched and detected. This implies the need for

a method to increase sampling density. For rotational motions with non-uniform speed, motion models in the 3-D Gabor domain are still to be found.

(a) The magnitude of the Fourier transform of an auxiliary function in the critical sampling case.

(b) The magnitude of the Fourier transform of an auxiliary function in the over sampling case.

Figure 5.1: Magnitude of the Fourier transform of the auxiliary function in critical sampling and over sampling cases.

(a) The projection of the 3-D Gabor transform on the temporal frequency axis for a rotation of $\frac{\pi}{4}$ /frame.

(b) The projection of the 3-D Gabor transform on the temporal frequency axis for a rotation of $\frac{\pi}{8}$ /frame

Figure 5.2: Temporal frequency distribution associated with two rotational objects.

(a) Frame No.1 of the image sequence

(b)The grey regions indicate the velocities of the rotating objects.

Figure 5.3: Experiment image sequence and the detected rotational regions.

Appendix A

Derivation of the Translational Motion Model in the Discrete 3-D Gabor domain

Denote by $\mathbf{d} = (m_d, n_d)$ and l_d the spatial and temporal centers of the Gabor basis functions, and $\mathbf{w} = (f_m, f_n)$ and f_l their spatial and temporal frequency centers. The corresponding discrete 3-D Gabor coefficient $C_{\mathbf{d}, l_d, \mathbf{w}, f_l}$ for an image sequence $I(\mathbf{x}, l)$ depicting translational motion during the interval $[0, l]$ is:

$$C_{\mathbf{d}, l_d, \mathbf{w}, f_l} = \sum_{\mathbf{x}} \sum_l I(\mathbf{x}, l) \gamma_1(\mathbf{x} - \mathbf{d}) \exp(-j2\pi \mathbf{w}^\top (\mathbf{x} - \mathbf{d})) \gamma_2(l - l_d) \exp(-j2\pi f_l (l - l_d)) \quad (\text{A.0.1})$$

The γ_1 and γ_2 are the spatial and temporal parts of the separable 3-D auxiliary function. Substituting from equation (3.1.2) we have:

$$C_{\mathbf{d}, l_d, \mathbf{w}, f_l} = \sum_{\mathbf{x}} \sum_l I_0(\mathbf{x} - D(l)) \gamma_1(\mathbf{x} - \mathbf{d}) \exp(-j2\pi \mathbf{w}^\top (\mathbf{x} - \mathbf{d})) \gamma_2(l - l_d) \exp(-j2\pi f_l (l - l_d)) \quad (\text{A.0.2})$$

Then, substituting $\hat{\mathbf{x}}$ for $\mathbf{x} - D(l)$ we can see that:

$$C_{\mathbf{d}, l_d, \mathbf{w}, f_l} = \sum_{\mathbf{x}} I_0(\hat{\mathbf{x}}) \gamma_1(\hat{\mathbf{x}} - (\mathbf{d} - D(l))) \exp(-j2\pi \mathbf{w}^\top (\hat{\mathbf{x}} - (\mathbf{d} - D(l))))$$

$$\sum_l \gamma_2(l - l_d) \exp(-j2\pi f(l - l_d)) \quad (\text{A.0.3})$$

$$= \sum_{\mathbf{x}} I_0(\hat{\mathbf{x}}) \gamma_1(\hat{\mathbf{x}} - (\mathbf{d} - D(l))) \exp(-j2\pi \mathbf{w}^\top (\hat{\mathbf{x}} - \mathbf{d})) \\ \sum_l \exp(j2\pi \mathbf{w}^\top D(l)) \gamma_2(l - l_d) \exp(-j2\pi f(l - l_d)) \quad (\text{A.0.4})$$

Compared with equation (3.1.3), the first summation has an extra shift $D(l)$ in the spatial auxiliary function γ_1 . Because the envelope of $\gamma_1(\hat{\mathbf{x}})$ is very similar to a Gaussian envelope (shown in Fig(A.1)), if we observe the signal in a small time interval in which the displacement $D(l)$ is small, we can argue that:

$$\gamma_1(\hat{\mathbf{x}} - (\mathbf{d} - D(l))) \approx \gamma_1(\hat{\mathbf{x}} - \mathbf{d}) \quad (\text{A.0.5})$$

Therefore, we have:

$$C_{\mathbf{d}, l_d, \mathbf{w}, f_l} \approx C_{\mathbf{d}, \mathbf{w}}^0 \sum_l \exp(j2\pi \mathbf{w}^\top D(l)) \gamma_2(l - l_d) \exp(-j2\pi f(l - l_d)) \quad (\text{A.0.6})$$

The summation can be regarded as the 1-D Gabor transform of $\exp(j2\pi \mathbf{w}^\top D(l))$. Denoting by $\mathbf{G}$ the 1-D Gabor transform, equation A.0.6 can be written as:

$$C_{\mathbf{d}, l_d, \mathbf{w}, f_l} \approx C_{\mathbf{d}, \mathbf{w}}^0 \mathbf{G}[\exp(j2\pi \mathbf{w}^\top D(l))] \quad (\text{A.0.7})$$

This is the model of translational motion in the 3-D Gabor domain. In addition, even if equation A.0.5 is questionable, which makes the first item in equation A.0.6 different from $C_{\mathbf{d}, \mathbf{w}}^0$, the translational motion model is still valid.

Figure A.1: Waveform of a baseband Gabor auxiliary function in a span of one local temporal region

Appendix B

Derivation of the Rotational Motion Model in the 3-D Gabor Domain

Denote the intensity of an object undertaking rotational motion as $I(r, \theta(t))$. In the SPTC coordinate system, its 3-D Gabor representation during the time interval $[0, t]$ is:

$$C(r_d, \theta_d, t_d, \omega_r, \omega_\theta, \omega_t) = \int_{-\infty}^{+\infty} \int_0^{+\infty} \int_{-\pi}^{+\pi} I(r, \theta(t)) \gamma_{r_d, \theta_d}(r, \theta) e^{j(r\omega_r + \theta\omega_\theta)} \gamma_{t_d}(t) e^{j(t\omega_t)} dr d\theta dt \quad (\text{B.0.8})$$

the (r_d, θ_d, t_d) and $(\omega_r, \omega_\theta, \omega_t)$ are the spatial and frequency centers of the 3-D Gabor basis functions. The corresponding auxiliary function $\gamma_{r_d, \theta_d, t_d}(r, \theta, t) e^{j(r\omega_r + \theta\omega_\theta + t\omega_t)}$ is separated into a spatial part $\gamma_{r_d, \theta_d}(r, \theta) e^{j(r\omega_r + \theta\omega_\theta)}$ and a temporal part $\gamma_{t_d}(t) e^{j(t\omega_t)}$. Substituting equation (5.3.3) into equation (B.0.8), we have:

$$C(r_d, \theta_d, t_d, \omega_r, \omega_\theta, \omega_t) = \int_{-\infty}^{+\infty} \int_0^{+\infty} \int_{-\pi}^{+\pi} I_0(r, \theta(t) - D(t)) \gamma_{r_d, \theta_d}(r, \theta) e^{j(r\omega_r + \theta\omega_\theta)} \gamma_{t_d}(t) e^{j(t\omega_t)} dr d\theta dt \quad (\text{B.0.9})$$

As the envelope of the 2 D auxiliary function $\gamma(r, \theta)$ is isotropic, the magnitude does not change with θ . Therefore, equation (B.0.9) can be written as:

$$\begin{aligned}
C(r_d, \theta_d, t_d, \omega_r, \omega_\theta, \omega_t) = & \int_{-\infty}^{+\infty} \int_0^{+\infty} \int_{-\pi}^{+\pi} I_0(r, \theta(t) - D(t)) \\
& \gamma_{r_d, \theta_d}(r, \theta - D(t)) e^{j(r\omega_r + \theta\omega_\theta)} \\
& \gamma_{t_d}(t) e^{j(t\omega_t)} dr d\theta dt
\end{aligned} \tag{B.0.10}$$

substituting equation (5.3.5) and equation (5.3.7) into equation (B.0.10) we have:

$$\begin{aligned}
C(r_d, \theta_d, t_d, \omega_r, \omega_\theta, \omega_t) = & \int_{-\infty}^{+\infty} \int_0^{+\infty} \int_{-\pi}^{+\pi} \sum_{n=-\infty}^{\infty} P_n(r) e^{jn\theta} e^{-jnD(t)} \\
& e^{j(r\omega_r + \theta\omega_\theta)} \gamma_{t_d}(t) \\
& e^{j(t\omega_t)} dr d\theta dt
\end{aligned} \tag{B.0.11}$$

By assembling all spatial components into a function sequence $L_n(r_d, \theta_d, \omega_r, \omega_\theta)$ we have:

$$C(r_d, \theta_d, t_d, \omega_r, \omega_\theta, \omega_t) = \sum_{n=-\infty}^{+\infty} L_n(r_d, \theta_d, \omega_r, \omega_\theta) \int_{-\infty}^{+\infty} e^{-jnD(t)} \gamma_{t_d}(t) e^{jt\omega_t} dt \tag{B.0.12}$$

The trailing integral in the equation is the temporal Gabor transform of $\exp(-jnD(t))$, that is:

$$C(r_d, \theta_d, t_d, \omega_r, \omega_\theta, \omega_t) = \sum_{n=-\infty}^{+\infty} L_n(r_d, \theta_d, \omega_r, \omega_\theta) \mathbf{G}_t[e^{-jnD(t)}](t_d, \omega_t) \tag{B.0.13}$$

where $\mathbf{G}_t$ is temporal Gabor transform. This is the model of the rotational motion.

Appendix C

Source Code of the Algorithms

C.1 Generate the Over Sampling 3-D Gabor Auxiliary Function Matrix

This module is to create the 3-D Gabor auxiliary function matrix. The 3-D Gabor coefficients can be calculated by computing the inner product of the digital image sequence and the auxiliary function matrix. First, the 3-D Gabor basis function matrix is formed according to the parameter settings. Equation (2.3.8) is then applied to obtain numerical values of the auxiliary functions at the sampling points. The α and the st/stf sampling rate can be set to realize different 3-D Gabor representations.

```
program gag-over-samp
integer dim,D,m1
parameter (dim=256,m1=16,D=32)
c    m1 is the number of frequency band
c    D is the number of spatial center
double complex gabor,g(dim,2*dim),gtginv(2*dim,2*dim),
+ a(dim,2*dim),gtg(2*dim,2*dim)
real b(dim)
double precision alpha
```

```

integer x,m,r,row,col
alpha=1/(16.0)

c
c   Create g. Note: it may be advantageous to reorder this matrix, to
c   exploit sparse matrix methods.
c
do 30 r=0,m1-1
c   r is the index of frequency band
do 20 m=0,D-1
c   m is the index of the spatial band
do 10 x=0,dim-1
g(x+1,m+r*D+1)=gabor(x,m,D,m1,r,alpha,dim)
10   continue
20   continue
30   continue
c   g stores Gabor basis functions
c
c   Output g.
c
open(unit=1,file='Gabor Basis function file',status='unknown')
write(1,*) g
close(unit=1)

c
c   Create gtg.
c
do 60 col=1,2*dim

```

```

do 50 row=1,2*dim
    gtg(row,col)=dcmplx(0.0d0,0.0d0)
    do 40 i=1,dim
        gtg(row,col)=gtg(row,col)+g(i,row)*g(i,col)
40    continue
50    continue
60    continue
c
c    Invert gtg.
c
call cinvert(gtg,gtginv,2*dim,2*dim)
c
c    Create a.
c
do 90 col=1,2*dim
    do 80 row=1,dim
        a(row,col)=dcmplx(0.0d0,0.0d0)
        do 70 i=1,2*dim
            a(row,col)=a(row,col)+g(row,i)*gtginv(i,col)
70    continue
80    continue
90    continue
c
c    Output a.
c
open(unit=1,file='Gabor auxiliary function file',status='unknown')

```

```

write(1,*) a
end

c
c
c

double complex function gabor(x,m,D,m1,r,alpha,dim)
integer x,m,m1,r,D,dim
double precision alpha,pi,gauss
c   double complex gabor
parameter (pi=3.14159265358979323846264338328)
gauss=dexp(-pi*alpha*alpha*((real(x)-real(m))*real(dim/D)-
+ (real(dim/D)-1.0d0)/2.0d0)**2.0d0))

c
c   NOTE SIGN CHANGE and PHASE CHANGE!!!!
c

gabor=gauss*cdexp(dcmplx(0.0d0,2.0d0*pi*real(r)*
+ (real(x)-real(m))*real(dim/D)-(real(dim/D)-1.0d0)/2.0d0)
+ /real(m1)))
return
end

c
c
c

subroutine cinvert(a,y,n,np)
double complex a(np,np),y(np,np)
dimension indx(512)

```

```

c
c      Set up identity matrix.
c
      do 20 i=1,n
        do 10 j=1,n
          y(i,j)=dcmplx(0.0d0,0.0d0)
10      continue
          y(i,i)=dcmplx(1.0d0,0.0d0)
20      continue
      call cludcmp(a,n,np,indx,d)
      do 30 j=1,n
        call clubksb(a,n,np,indx,y(1,j))
30      continue
      return
      end

c
c
c
      subroutine cludcmp(a,n,np,indx,d)
      double complex a(np,np),sum,cdum
      parameter (nmax=1000)
      double precision aamax,vv(nmax),d,tiny,dum
      parameter (tiny=1.0d-20)
      dimension indx(n)
      d=1.0d0
      do 12 i=1,n
        aamax=0.0d0

```

```

do 11 j=1,n
    if (cdabs(a(i,j)).gt.aamax) aamax=cdabs(a(i,j))
11    continue
    if (aamax.lt.1.0e-6) pause 'singular matrix.'
    vv(i)=1.0d0/aamax
12    continue
do 19 j=1,n
    if (j.gt.1) then
        do 14 i=1,j-1
            sum=a(i,j)
            if (i.gt.1)then
                do 13 k=1,i-1
                    sum=sum-a(i,k)*a(k,j)
13                continue
                a(i,j)=sum
            endif
14        continue
    endif
    aamax=0.0d0
do 16 i=j,n
    sum=a(i,j)
    if (j.gt.1)then
        do 15 k=1,j-1
            sum=sum-a(i,k)*a(k,j)
15        continue
        a(i,j)=sum

```



```

endif
dum=vv(i)*cdabs(sum)
if (dum.ge.aamax) then
    imax=i
    aamax=dum
endif
16    continue
    if (j.ne.imax)then
        do 17 k=1,n
            cdum=a(imax,k)
            a(imax,k)=a(j,k)
            a(j,k)=cdum
17        continue
            d=-d
            vv(imax)=vv(j)
        endif
        indx(j)=imax
        if(j.ne.n)then
            if(a(j,j).eq.dcmplx(0.0d0,0.0d0))a(j,j)=dcmplx(tiny,tiny)
            cdum=dcmplx(1.0d0,0.0d0)/a(j,j)
            do 18 i=j+1,n
                a(i,j)=a(i,j)*cdum
18            continue
        endif
19    continue
    if(a(n,n).eq.dcmplx(0.0d0,0.0d0))a(n,n)=dcmplx(tiny,tiny)

```

```

return
end
c
c
c
subroutine clubksb(a,n,np,indx,b)
double complex a(np,np),b(n),sum
dimension indx(n)
ii=0
do 12 i=1,n
    ll=indx(i)
    sum=b(ll)
    b(ll)=b(i)
    if (ii.ne.0)then
        do 11 j=ii,i-1
            sum=sum-a(i,j)*b(j)
11        continue
        else if (sum.ne.dcmplx(0.0d0,0.0d0)) then
            ii=i
        endif
        b(i)=sum
12    continue
do 14 i=n,1,-1
    sum=b(i)
    if(i.lt.n)then
        do 13 j=i+1,n
            sum=sum-a(i,j)*b(j)

```

```

13         continue

        endif

        b(i)=sum/a(i,i)
14     continue

    return

end

```

C.2 The 3-D Gabor Expansion with Over Sampling

This module realizes the 3-D Gabor expansion. The inner product of the 3-D image sequence and the 3-D auxiliary function matrix. The kernel functions are separatable.

```

program dec3d-over-samp

integer dim,ddim,fnum,ffnum

parameter (dim=256,ddim=512,fnum=48,ffnum=48)

integer image(dim,dim),seq(dim,dim,fnum),row,col,frame,fstn,scdn

character fname(fnum)*15,uname*15

double complex a(dim,ddim),atime(fnum,ffnum),temp(ddim,dim),
+ tcoeffs(ddim,ddim,fnum),coeffs(ddim,ddim,ffnum)

c
c   Prompt for filenames
c

uname='Image sequence name'

do 5 frame=1,9

    fname(frame)(1:5)=uname

    fname(frame)(6:6)=char(frame+48)

```

```

        fname(frame)(7:)='.raw'
5 continue
do 8 frame=10,fnum
    fname(frame)(1:5)=uniname
    fstn=aint(frame/10.0)
    scdn=mod(frame,10)
    fname(frame)(6:6)=char(fstn+48)
    fname(frame)(7:7)=char(scdn+48)
    fname(frame)(8:)='.raw'
8 continue

c
c    Read the sequence (raw format).
c

        do 20 frame=1,fnum
call read256(image,dim,fname(frame),10)
        do 15 col=1,dim
            do 10 row=1,dim
                seq(row,col,frame)=image(row,col)
10                continue
15            continue
20        continue

c
c    Read a.
c

open(unit=1,file='Gabor auxiliary function file',status='old')

```

```

read(1,*) a
close(unit=1)

c
c      Calculate 2-D (spatial) coefficients (at*seq*a).
c
do 70 frame=1,fnum
  do 50 col=1,dim
    do 45 row=1,ddim
      temp(row,col)=dcmplx(0.0d0,0.0d0)
      do 40 i=1,dim
        temp(row,col)=temp(row,col)+a(i,row)*
          + dcmplx(real(seq(i,col,frame)),0.0d0)
40      continue
45      continue
50      continue
    do 65 col=1,ddim
      do 60 row=1,ddim
        tcoeffs(row,col,frame)=dcmplx(0.0d0,0.0d0)
        do 55 i=1,dim
          tcoeffs(row,col,frame)=tcoeffs(row,col,frame)+
            + temp(row,i)*a(i,col)
55      continue
60      continue
65      continue
70      continue
c

```

```

c      Read atime.
c
      open(unit=1,file='Gabor temporal auxiliary function',status='old')
      read(1,*) atime
      close(unit=1)
c
c      Calculate final coefficients (along time dimension).
c
      do 90 col=1,ddim
        do 85 row=1,ddim
          do 80 frame=1,ffnum
            coeffs(row,col,frame)=dcmplx(0.0d0,0.0d0)
            do 75 i=1,fnum
              coeffs(row,col,frame)=coeffs(row,col,frame)+
                + atime(i,frame)*tcoeffs(row,col,i)
75          continue
80        continue
85      continue
90    continue
c
c      Output final (3D) coefficients as coeffs3d.
c
      open(unit=1,file='the 3-D Gabor coefficients file',status='unknown')
      write(1,*) coeffs
      end
c

```

```

subroutine read256(iarray,idim,fname,lunit)
c
c   read 256x256 image file into integer array
c
integer ix,iy,iyy,ival,idim
dimension iarray(idim,idim)
character*1 bytee(512)
character*512 buffer
character*(*) fname
equivalence (buffer,bytee(1))
open(unit=lunit,file=fname,access='direct',
1 form='formatted',recl=512,status='old')
do 30 iy=1,128
    read (lunit,rec=iy,fmt=5) buffer
5    format(a512)
    iyy=iy*2-1
    do 10 ix=1,256
        ival=ichar(bytee(ix))
        if (ival.lt.0) ival=ival+256
        iarray(iyy,ix)=ival
10    continue
    iyy=iyy+1
    do 20 ix=1,256
        ival=ichar(bytee(ix+256))
        if (ival.lt.0) ival=ival+256
        iarray(iyy,ix)=ival

```

```

20         continue
30         continue
        close(unit=lunit)
        return
    end

```

C.3 Motion Estimation in the 3-D Gabor Domain

C.3.1 Arrangment of Local Frequency Information

This module is to group the 3-D Gabor coefficients according to their spatiotemporal positions.

```

program intreov
integer dim,fnum,m1,D,mt
parameter (dim=512,fnum=48,m1=16,D=32,mt=16)
integer row,col,frame,mag(dim,dim,fnum),newmag(dim,dim,fnum)
c
c    Read the coefficients.
c
    open(unit=1,file='the normalized 3-D Gabor coefficients file',status='old')
    read(1,*) mag
    close(unit=1)
c
c    Output coeffs at each spatiotemporal location.
c
    do 60 col=1,D
        do 50 row=1,D

```



```

do 40 frame=1,fnum/mt
do 30 i=0,m1-1
do 20 j=0,m1-1
do 10 k=0,mt-1
newmag(j+1+(row-1)*m1,i+1+(col-1)*m1,k+1+(frame-1)*mt)=
+ mag(row+j*D,col+i*D,frame+k*fnum/mt)
10 continue
20 continue
30 continue
40 continue
50 continue
60 continue
c
c   Output final (3D) coefficients as newmagc3d.
c
open(unit=1,file='newmagc3d',status='unknown')
write(1,*) newmag
end
c
c   Arrange the 3-D Gabor coefficients within each spatiotemporal group
c   make them distributed the same as the 3-D Fourier domain c
program intreo2ov
integer dim,fnum,m1,D,mt
parameter (dim=512,fnum=48,m1=16,D=32,mt=16)
integer row,col,frame,mag(dim,dim,fnum),
+ newmag(dim,dim,fnum)

```

```

c
c      Read the coefficients.
c
open(unit=1,file='newmagc3d',status='old')
read(1,*) mag
close(unit=1)

c
c      Reorganize coeffs at each spatiotemporal location
c      to center the spectrum at (4,4,4).
c
do 60 col=1,D
  do 50 row=1,D
    do 40 frame=1,fnum/mt
      do 30 i=1,m1
        do 20 j=1,m1
          do 10 k=1,mt
            newmag((row-1)*m1+mod(j+(m1/2-2),m1)+1,(col-1)*m1+mod(i+
              + (m1/2-2),m1)+1,(frame-1)*mt+mod(k+(mt/2-2),mt)+1)=
              + mag(j+(row-1)*m1,i+(col-1)*m1,k+(frame-1)*mt)
10              continue
20              continue
30              continue
40              continue
50              continue
60              continue
c
c      Output final (3D) coefficients as cmagc3d.

```

c

```
open(unit=1,file='cmagc3d',status='unknown')  
write(1,*) newmag  
end
```

c

c Collect information in each local spatiotemporal region

c Prepare data for motion estimation

c

```
program listgenmov  
implicit none  
integer dim,fnum,m1,D,mt  
parameter (dim=512,fnum=48,m1=16,D=32,mt=16)  
integer count,thresh,records,row,col,frame,mag(dim,dim,fnum),  
+list(m1*m1*m1,4),i,j,k,l,magtemp
```

c

c Read the coefficients.

c

```
open(unit=1,file='cmagc3d',status='old')  
read(1,*) mag  
close(unit=1)
```

c c Set the number of records (the total number

c of spatiotemporal locations in the list).

c

```
records=(dim*dim*fnum)/(m1*m1*mt)
```

c

c Open the output file and write the number of records.

c

```
open(unit=1,file='file to write local frequency information',status='unknown')
```

```
write(1,*) records
```

c

c Output the shifted coordinates of all coeffs with squared

c magnitude exceeding thresh, at each spatiotemporal location.

c

```
do 60 frame=1,fnum/mt
```

```
do 50 row=1,D
```

```
do 40 col=1,D
```

```
count=0
```

```
do 30 j=1,m1
```

```
do 20 i=1,m1
```

```
do 10 k=1,mt
```

```
magtemp=mag(j+(row-1)*m1,i+(col-1)*m1,
```

```
+ k+(frame-1)*mt)
```

```
if(magtemp.gt.5)then
```

```
count=count+1
```

```
list(count,1)=i-m1/2
```

```
list(count,2)=j-m1/2
```

```
list(count,3)=k-mt/2
```

```
list(count,4)=magtemp
```

```
endif
```

10 continue

20 continue

30 continue

```
write(1,*) count
```

```

do 35 l=1,count
    write(1,*) list(1,1),list(1,2),list(1,3),list(1,4)
35    continue
40    continue
50    continue
60    continue
close(unit=1)
end

```

C.3.2 Motion Estimation and Motion Field Displaying (Matlab Version)

```

fid1=fopen('Output file in the previous step');
mm1=16;%number of the spatial frequency bands
mmt=16;%number of the temporal frequency bands
%optical2 is to estimate the solution of  $A(1:2)m=A(3)$  by weighted LS estimation
method
% Weighted by the confidence of each coefficients: magnitude.
records=fscanf(fid1,'%d',[1,1])
for i=1:records;
    N=fscanf(fid1,'%d',1);
    AA=fscanf(fid1,'%d',[4,N]);
    if N < 5;
        A=AA(1:3,:);
        A=A';
        B=A(:,1:2);% coefficients matrix
        z=(1.0*mm1/mmt)*A(:,3);          wt=zeros(N,N);          for j=1:N;

```

```

        wt(j,j)=AA(4,j)*AA(4,j)*AA(4,j)*AA(4,j)*AA(4,j);
    end;
    M=pinv(B'*wt*B)*B'*wt*z;
    m1(i)=-M(1);
    m2(i)=-M(2);
else
    m1(i)=0;
    m2(i)=0;
end;
end;
status=fclose(fid1);
dimx=512;
dimy=512;
dimz=3;

    for h=1:dimz;
        dx=zeros(dimx/mm1,dimy/mm1);
        dy=zeros(dimx/mm1,dimy/mm1);

        for i=2:dimx/mm1-1;
            for j=2:dimy/mm1-1;
                dx(i,j)=m1((h-1)*dimx*dimy/(mm1*mm1)+(i-1)*dimy/mm1+j);
                dy(i,j)=m2((h-1)*dimx*dimy/(mm1*mm1)+(i-1)*dimy/mm1+j);
            end;
        end;
    end;
    figure(h)
    quiver(dx,dy)

```

```

axis('ij')
axis equal;
end;

```

C.4 The Dense Motion Field Estimation

```

program estpxlmov
c    m1 is the number divisions in freq domain, Ddiv is the size of each location block
C    grpszt is the number of location centers, bandiv is the size of each freq band
integer dim,fnum,m1,mt,D,Dt,rato,totalnum,thresh,Ddiv
parameter (dim=512,fnum=48,m1=16,mt=16,D=32,Dt=3,Ddiv=8,Dtdiv=16)
integer myrow,mycol,myframe,subrow,subcol,subframe,rowlocal,
+ collocal,framelocal,nebrow,nebcoll,nebframe,count
integer int_sub_pxl(m1,m1,mt),list(m1*m1*mt,4),histogram(256)
character listname(m1,m1)*15,uniname*15;
double complex coeffs(dim,dim,fnum)
real mag(dim,dim,fnum),magmin,magmax,rsub_pixel(m1,m1,mt)
real gate,small_pixel(m1,m1,mt)
double precision alpha,alphat,scalediv,realthresh
double complex gabor
double complex sub_pixel(m1,m1,mt),local_freq_coef(m1,m1,mt)
c    alpha=1.0/16.0
    alphat=1.0/16.0
c c    threthold gives an int number within (0 to 255) that the probability of a Gabor
c    coefficients more than this number is gate
c    gate=0.02

```

```

thresh=0
totalnum=dim*dim*fnum
c
c   Read the coefficients.
c
open(unit=1,file='the 3-D Gabor coefficients file',status='old')
read(1,*) coeffs
close(unit=1)

c
c   initialize the histogram
c
do 5 qnum=1,256
    histogram(qnum)=0
5 continue

c
c   Calculate the magnitudes of the coefficients.
c
do 35 col=1,dim
    do 25 row=1,dim
        do 15 frame=1,fnum
            mag(row,col,frame)=cdabs(coeffs(row,col,frame))
        15 continue
    25 continue
35 continue

c

```



```

c      Scale the magnitude to fall in the range 0. to 255.
c
      call rminmax3d(mag,dim,fnum,magmin,magmax)
      scalediv=(magmax-magmin)/256.0d0
      call scale3d(mag,dim,fnum,magmin,magmax)
c
c      Quantize (round to the nearest integer).
c
      do 61 col=1,dim
        do 51 row=1,dim
          do 41 frame=1,fnum
            intemp=nint(mag(row,col,frame))
            histogram(intemp+1)=histogram(intemp+1)+1
41          continue
51        continue
61      continue
c
c      determine the threshold
c
      call freqgt3d(histogram,gate,thresh,totalnum)
c
c      get global min and max value
c
      myframe=2
      do 23 mycol=(begion col),(end col)

```

```
do 27 myrow=(begin row),(end row)
```

```
    uniname='dirctory to store local frequency domain of every pixel'
```

```
do 115 subcol=1,Ddiv
```

```
    do 225 subrow=1,Ddiv
```

```
        subframe=6
```

```
        rowlocal=(myrow-1)*Ddiv+subrow
```

```
        collocal=(mycol-1)*Ddiv+subcol
```

```
        framelocal=(myframe-1)*Dtdiv+subframe
```

```
        listname(subrow,subcol)(1:4)=uniname
```

```
        if (myrow.le.9) then
```

```
            listname(subrow,subcol)(5:5)=char(48)
```

```
            listname(subrow,subcol)(6:6)=char(48+myrow)
```

```
        elseif (myrow.le.19) then
```

```
            listname(subrow,subcol)(5:5)=char(48+1)
```

```
            listname(subrow,subcol)(6:6)=char(48+myrow-10)
```

```
        else
```

```
            listname(subrow,subcol)(5:5)=char(48+2)
```

```
            listname(subrow,subcol)(6:6)=char(48+myrow-20)
```

```
        endif
```

```
        if (mycol.le.9) then
```

```

listname(subrow,subcol)(7:7)=char(48)
listname(subrow,subcol)(8:8)=char(48+mycol)
elseif (mycol.le.19) then
listname(subrow,subcol)(7:7)=char(48+1)
listname(subrow,subcol)(8:8)=char(48+mycol-10)
else
listname(subrow,subcol)(7:7)=char(48+2)
listname(subrow,subcol)(8:8)=char(48+mycol-20)
endif

      if (subrow.le.9) then
listname(subrow,subcol)(9:9)=char(48)
listname(subrow,subcol)(10:10)=char(48+subrow)
elseif (subrow.le.19) then
listname(subrow,subcol)(9:9)=char(48+1)
listname(subrow,subcol)(10:10)=char(48+subrow-10)
else
listname(subrow,subcol)(9:9)=char(48+2)
listname(subrow,subcol)(10:10)=char(48+subrow-20)
endif
endif

if (subcol.le.9) then
listname(subrow,subcol)(11:11)=char(48)
listname(subrow,subcol)(12:12)=char(48+subcol)
elseif (subcol.le.19) then
listname(subrow,subcol)(11:11)=char(48+1)
listname(subrow,subcol)(12:12)=char(48+subcol-10)

```

```

else
    listname(subrow,subcol)(11:11)=char(48+2)
    listname(subrow,subcol)(12:12)=char(48+subcol-20)
endif
write(*,*) listname(subrow,subcol)

c
c      Output coeffs at each spatiotemporal location.
c
do 10 k=1,mt
    do 20 j=1,m1
        do 30 i=1,m1
            sub_pixel(i,j,k)=0.0
30          continue
20          continue
10         continue
c
c      myrow,mycol,myframe is the current spatial-temporal location
c      it can not be 1 or dim/m1!!!
c

```

```

do 500 nebframe=-1,1
do 400 nebcoll=-1,1
do 300 nebrow=-1,1
    rato=1
do 550 k=1,mt

```

```

do 440 j=1,m1
do 330 i=1,m1
local_freq_coef(i,j,k)=coeffs(myrow+nebrow+(i-1)*D,
+ mycol+nebcoll+(j-1)*D,myframe+nebframe+(k-1)*Dt)
330          continue
440          continue
550          continue
c
c      Calculate the subband pixel map.
c
do 60 k=1,mt
do 50 j=1,m1
do 40 i=1,m1
sub_pixel(i,j,k)=sub_pixel(i,j,k)+
+(gabor(rowlocal-1,myrow+nebrow-1,D,m1,i-1,alpha,dim/2)*
+gabor(collocal-1,mycol+nebcoll-1,D,m1,j-1,alpha,dim/2)*
+gabor(framelocal-1,myframe+nebframe-1,Dt,mt,k-1,alphan,fnum)*
+local_freq_coef(i,j,k)*rato)
40          continue
50          continue
60          continue

300          continue

400          continue
500          continue
do 11 k=1,mt

```

```

do 22 j=1,m1
  do 33 i=1,m1
    rsub_pixel(i,j,k)=cdabs(sub_pixel(i,j,k))
    small_pixel(i,j,k)=1
    if (rsub_pixel(i,j,k).lt.realthresh) then
      small_pixel(i,j,k)=-1
    end if
33      continue
22      continue
11      continue
c
c    Scale the magnitudes to fall in the range 0. to 255.
c
    call rminmax3d(rsub_pixel,m1,mt,magmin,magmax)
    call scale3d(rsub_pixel,m1,mt,magmin,magmax)
c
c    Quantize (round to the nearest integer).
c
    do 90 z=1,mt
      do 80 y=1,m1
        do 70 x=1,m1
          if (small_pixel(x,y,z) .gt. 0)then
            int_sub_pxl(x,y,z)=nint(rsub_pixel(x,y,z))
          else
            int_sub_pxl(x,y,z)=0

```

```

                                end if
70                                continue
80                                continue
90                                continue
c
c    Open the output file and write the number of records.
c
                                open(unit=1,file=listname(subrow,subcol),status='unknown')
                                write(1,*) magmax
c
c    Output the shifted coordinates of all coeffs with squared
c    magnitude exceeding thresh, at each spatiotemporal location.
c
                                count=0
                                do 120 i=1,m1
                                    do 110 j=1,m1
                                        do 100 k=1,mt
                                            if(int_sub_pxl(i,j,k).gt.thresh)then
                                                count=count+1
                                                list(count,1)=mod(j+(m1/2-2),m1)+1-m1/2
                                                list(count,2)=mod(i+(m1/2-2),m1)+1-m1/2
                                                list(count,3)=mod(k+(mt/2-2),m1)+1-mt/2
                                                list(count,4)=int_sub_pxl(i,j,k)
                                            end if
100                                continue
110                                continue

```

```

120          continue
          write(1,*) count
          do 135 l=1,count
              write(1,*) list(1,1),list(1,2),list(1,3),list(1,4)
135          continue
          close(unit=1)
225          continue
115          continue
27          continue
23          continue
          end

c
c
c

subroutine scale3d(array,dim,fnum,minval,maxval)
integer dim,fnum,row,col,frame
real minval,maxval,scale,array(dim,dim,fnum)

c
c    This subroutine scales a real 3-D array so that all entries
c    fall between 0. and 255.
c
c    dim = array dimension (spatial or spatial-frequency)
c    fnum = array dimension (temporal or temporal-frequency)
c    scale = scale factor
c    maxval = maximum value in the array
c    minval = minimum value in the array

```



```

        scale=(maxval-minval)/255.0d0
c
c      Scale the array.
c
      do 30 frame=1,fnum
        do 20 col=1,dim
          do 10 row=1,dim
            array(row,col,frame)=(array(row,col,frame)-minval)/scale
10          continue
20          continue
30          continue
      return
    end
c
c
c
subroutine rminmax3d(array,dim,fnum,minval,maxval)
integer dim,fnum,row,col,frame
real minval,maxval,array(dim,dim,fnum)

      minval=array(1,1,1)
maxval=array(1,1,1)
      do 300 frame=1,fnum
        do 200 col=1,dim
          do 100 row=1,dim
            if(array(row,col,frame).gt.maxval)then

```

```

        maxval=array(row,col,frame)
    endif
    if(array(row,col,frame).lt.minval)then
        minval=array(row,col,frame)
    endif
100    continue
200    continue
300    continue
    return
end
c
subroutine freqgt3d(histo,gate,threthold,totalnum)
integer histo(256),threthold,totalnum,intgate,sum
real gate
intgate=int(gate*totalnum)
sum=0
threthold=256
do 35 while (sum<intgate)
    sum=sum+histo(threthold)
    threthold=threthold-1
35    continue

    return
end

```

Bibliography

- [1] Todd R. Reed, editor. *Digital Image Sequence Processing, Compression, and Analysis*. CRC Press, 2004.
- [2] M. Foodeei and E. Dubois. Coding image sequence intensities along motion trajectories using ec-celp quantization. In *Proc. IEEE International Conference on Image Processing*, pages 1720 – 1724, Nov. 1994.
- [3] Hyun-Wook Park and Kim Hyung-Sun. Motion estimation using low-band-shift method for wavelet-based moving-picture coding. *IEEE Trans. on Image Processing*, 9(4):577–587, April 2000.
- [4] Harpreet S. Sawhney and Serge Ayer. Compact representations of videos through dominant and multiple motion estimation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 18(8):814–830, August 1996.
- [5] A. Chang, M. Tekalp and M. Sezan. Simultaneous motion estimation and segmentation. *IEEE Trans. Image Processing*, 6:1326–1333, Sept. 1997.
- [6] S. Wu and J. Kittler. A gradient-based method for general motion estimation and segmentation. *Journal of Vision Communications. Image Representation*, 4:25–38, March 1993.
- [7] F. Dufaux, I. Moccagatta, F. Moscheni, and H. Nicolas. Vector quantization-based motion field segmentation under the entropy criterion. *Journal of Vision Communications. Image Representation*, 5:356–369, December 1994.

- [8] J. Brailean, R.P. Kleihorst, S.N. Efstratiadis, and A.K. Katsaggelos. Noise reduction filters for dynamic image sequences: a review. *Proc. IEEE*, 83(9):1272–1292, Sept. 1995.
- [9] Michel Chahine and Janusz Konrad. Estimation and compensation of accelerated motion for temporal sequence interpolation. *Signal Processing: Image Communication*, 7:503–527, 1995.
- [10] M. Chan, Y. Yu, and A. Constantinides. Variable size block matching motion compensation with applications to video coding. *IEE Proc. I, Communication of Speech and Vision*, 37:205–212, Aug 1990.
- [11] Y. Wang and O. Lee. Active mesh: A feature seeking and tracking image sequence representation scheme. *IEEE Trans. Image Processing*, 3:610–624, Sept 1994.
- [12] A. Tekalp, editor. *Digital Video Processing*. Prentice Hall PTR, 1995.
- [13] C. Stiller and J. Konrad. Estimating motion in image sequences - a tutorial on modeling and computation of 2D motion. *IEEE Signal Processing Magazine*, pages 70–91, July 1999.
- [14] B. K. P. Horn and B. G. Schunk. Determining optical flow. *Artificial Intelligence*, 17(1-3):185–203, 1981.
- [15] J. Konrad and E Dubois. Bayesian estimation of motion vector fields. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 14:910–927, 1992.
- [16] Christoph Stiller. Object-based estimation of dense motion fields. *IEEE Trans. on Image Processing*, 6(2):234 – 250, February 1997.
- [17] Aurn Kumar, Allen R. Tannenbaum, and Gary J. Balas. Optical flow: A curve evolution approach. *IEEE Trans. on Image Processing*, 5(4):598–610, April 1996.

- [18] C. Moloney and E. Dubois. Estimation of motion fields from image sequences with illumination variation. In *Proceedings of IEEE International Conference in Acoustics, Speech and Signal Processing*, pages 2425–2428, May 1991.
- [19] Wei ge Chen, G.B. Giannakis, and N. Nandhakumar. Spatiotemporal approach for time-varying global image motion estimation. *IEEE Trans. on Image Processing*, 5(10):1448 – 1461, October 1996.
- [20] A. Kojima, N. Sakurai, and J. Kishigami. Motion detection using 3D-FFT spectrum. In *Proc. of IEEE Int'l Conf. on Acoustics, Speech and Signal Processing*, volume V, pages 213–216, Minneapolis, Minnesota, April 27-30 1993.
- [21] J. Chamorro-Martinez, J. Fdez-Valdivia, J.A. Garcia, and J. Martinez-Baena. A frequency-domain approach for the extraction of motion patterns. In *Proceedings of the 2003 International Conference on Acoustics, Speech, and Signal Processing*, volume 3, pages 165–168, 6-10 April 2003.
- [22] D. J. Fleet and A. D. Jepson. Computation of component image velocity from local phase information. *Int'l Journal of Computer Vision*, 5(1):77–104, 1990.
- [23] G. Cortelazzo and M. Balanza. Frequency domain analysis of translations with piecewise cubic trajectories. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 15(4):411–416, 1993.
- [24] G. Cortelazzo and G. Nalesso. A differential equation approach to the computation of the Fourier transform of the images of translating objects. *IEEE Trans. on Information Theory*, 40(6):2049–2058, 1994.
- [25] Regis J Crinon and Wojciech J Kolodziej. Motion estimation: the concept of velocity bandwidth. In *Proc. SPIE Visual Communication and Image Processing, 1992*, volume 1818, pages 62 – 73, 1992.

- [26] T. R. Reed. A spatiotemporal/spatiotemporal-frequency interpretation of apparent motion reversal. In *Proc. of the Sixteenth Int'l Joint Conf. on Artificial Intelligence*, volume 2, pages 1140–1145, Stockholm, Sweden, July 31 - August 6 1999.
- [27] G. Cortelazzo and C.M. Monti. Frequency domain analysis of rotational motion. *Multidimensional Systems and Signal Processing*, (4):203–225, 1993.
- [28] J.L. Lucchese and G. M. Cortelazzo. A noise-robust frequency domain technique for estimating planar roto-translations. *IEEE Trans. on Signal Processing*, 48(6):1769–1786, June 2000.
- [29] L. Lucchese, G. Cortelazzo, and C. Monti. A frequency domain technique for estimating rigid planar rotations. In *Proc. of ISCAS'96*, pages 774–777, Atlanta, Georgia, May 1996.
- [30] C. P. Bernard. Discrete wavelet analysis: a new framework for fast optic flow computation. In *Proc. of the 5th European Conference on Computer Vision*, volume 2, pages 354–368, Freiburg, Germany, June 2-6 1998.
- [31] A. Corghi and R. Leonardi. Determining the optical flow using wavelet coefficients. In *Proc. of Visual Communications and Image Processing, 1997*, pages 315–327, Feb. 1997.
- [32] J. Magarey and N. Kingsbury. Motion estimation using a complex-valued wavelet transform. *IEEE Trans. on Signal Processing*, 46(4):1069–1084, April 1998.
- [33] Enrique A. Barria and Jogikall M. Jagadeesh. Wavelet functions to estimate velocity in spatiotemporal signals. *IEEE Trans. Signal Processing*, 46(4):1105–1118, April 1998.

- [34] T. J. Burns, S. K. Rogers, M. E. Oxley, and D. W. Ruck. Discrete, spatiotemporal, wavelet multiresolution analysis method for computing optical flow. *Optical Engineering*, 33(7):2236–2247, 1994.
- [35] John G. Daugman. Complete discrete 2-d gabor transforms by neural networks for image analysis and compression. *IEEE Trans. Acoustics, Speech and Signal Processing*, 36(7):1169–1179, July. 1988.
- [36] M. Porat and Y. Y. Zeevi. The generalized Gabor scheme of image representation in biological and machine vision. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 10(4):452–468, July 1988.
- [37] M. Bastiaans. Gabor’s expansion of a signal into gaussian elementary signals. *Proc. IEEE*, 68:538 – 539, 1980.
- [38] J. Wexler and S. Raz. Discrete gabor expansions. *Signal Processing*, 21(3):207–221, November 1990.
- [39] T. R. Reed. On the computation of optical flow using the 3-D Gabor transform. *Multidimensional Systems and Signal Processing*, 9(4):115–120, 1998.
- [40] Troy T. Chinen and T. R. Reed. A performance analysis of fast gabor transform methods. *Graphical Models and Image Processing*, 59(3):117–127, 1997.
- [41] Shie Qian and Dapang Chen. Discrete gabor transform. *IEEE Trans. Signal Processing*, 41(7):2429–2438, July. 1993.
- [42] A. Girotra, S. Johar, D. Ghosh, and I. Chakrabarti. An architecture for affine motion estimation in real-time video coding. In *Proc. of the 9th Asia-Pacific Conference on Communications, 2003.*, volume 1, pages 103 – 107, Bangalore, India, Sept. 21-24 2003.