6-28-2010

Global optimization methods for full-reference and no-reference motion estimation with applications to atherosclerotic plaque motion and strain imaging

Sergio Murillo Amaya

Sergio Eduardo Murillo Amaya
_____
Candidate

Electrical and Computer Engineering
_____
Department

This dissertation is approved, and it is acceptable in quality
and form for publication:

Approved by the Dissertation Committee:

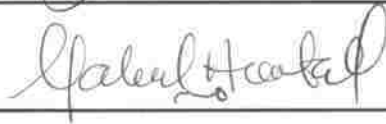_____, Chairperson

_____

_____

_____

_____

_____

_____

_____

i

GLOBAL OPTIMIZATION METHODS FOR FULL-REFERENCE AND NO-REFERENCE MOTION ESTIMATION WITH APPLICATIONS TO ATHEROSCLEROTIC PLAQUE MOTION AND STRAIN IMAGING

BY

SERGIO EDUARDO MURILLO AMAYA

B.S., Electrical Engineering, Pontificia Universidad Católica del Perú, 2004
M.S., Electrical Engineering, University of New Mexico, 2006

DISSERTATION

Submitted in Partial Fulfillment of the
Requirements for the Degree of

**Doctor of Philosophy
Engineering**

The University of New Mexico
Albuquerque, New Mexico

May, 2010

Dedication

To my wife, my parents, and my whole family. For giving me strength and resolve to pursue my dreams. This work is as much yours as it is mine.

To the memory of my late grandfathers. Eduardo and Hector. I am sure that where you are, you are enjoying this moment.

Pa' delante nomas! -Anonymous

"As far as the laws of mathematics refer to reality, they are not certain, as far as they are certain, they do not refer to reality" – Albert Einstein.

Acknowledgements

I would like to thank my advisor, Dr. Marios Pattichis. He is the best mentor I could ever have asked for. Thanks for pushing me to be better, for believing when I already lost hope, and for putting up with my cranky mood during the last stages of this dissertation. Without his wise advice, support, friendship, and teachings; I would have never finished.

I would like to thank Dr. Peter Soliz for his insightful comments, collaboration, and continued financial support through VisionQuest Biomedical.

I would like to thank the other members of my dissertation committee. Dr. Ramiro Jordan, Dr. Peter Dorato, and Dr. Gabriel Huerta. I was fortunate enough to have you as professors at some point during my graduate studies.

Thanks to Ana Edwards. She help me battle and defeat the evil in Microsoft Word.

Thanks to all my friends at the Istec Lab and UNM. Your friendship has always been a bright spot during the last five years and a half. Impossible to forget all the nights and drinks at the Peruvian embassy. Special thanks to Dr. Victor Murray and Dr. Alexandre Franco. Two extraordinary friends with whom I share countless memories of my graduate student years.

Thanks to my family in Perú. You are always in my thoughts. This dissertation is the combined effort of all of you. Without you, I would not be here today.

Lastly, but certainly not the least. I have to express my deepest gratitude and love for my wife, Hilda. You are the most precious gift God has giving me in this life. Every time I think of you, the sun shines in my heart and tears of joy flood my eyes. You are my definition of happiness. I have no words to thank you. Thanks for believing in what I could become, for always seeing through the forest, for sticking with me during this crazy road, and for the extraordinary life that we share and the years to come.

To all of you. Salud!

**GLOBAL OPTIMIZATION METHODS FOR FULL-REFERENCE
AND NO-REFERENCE MOTION ESTIMATION WITH
APPLICATIONS TO ATHEROSCLEROTIC PLAQUE MOTION
AND STRAIN IMAGING**

BY

SERGIO EDUARDO MURILLO AMAYA

ABSTRACT OF DISSERTATION

Submitted in Partial Fulfillment of the
Requirements for the Degree of

**Doctor of Philosophy
Engineering**

The University of New Mexico
Albuquerque, New Mexico

May, 2010

GLOBAL OPTIMIZATION METHODS FOR FULL-REFERENCE AND NO-REFERENCE MOTION ESTIMATION WITH APPLICATION TO ATHEROSCLEROTIC PLAQUE MOTION AND STRAIN IMAGING

by

Sergio Eduardo Murillo Amaya

B.S., Electrical Engineering, Pontificia Universidad Católica del Perú, 2004
M.S., Electrical Engineering, University of New Mexico, 2006
Ph.D., Engineering, University of New Mexico, USA, 2009

ABSTRACT

Pixel-based motion estimation using optical flow models has been extensively researched during the last two decades. The driving force of this research field is the amount of applications that can be developed with the motion estimates. Image segmentation, compression, activity detection, object tracking, pattern recognition, and more recently non-invasive biomedical applications like strain imaging make the estimation of accurate velocity fields necessary.

The majority of the research in this area is focused on improving the theoretical and numerical framework of the optical flow models. This effort has resulted in increased method complexity with an increasing number of motion parameters. The standard approach of heuristically setting the motion parameters has become a major source of estimation error.

This dissertation is focused in the development of reliable motion estimation based on global parameter optimization methods. Two strategies have been developed. In full-reference optimization, the assumption is that a video training set of realistic motion simulations (or ground truth) are available. Global optimization is used to calculate the

best motion parameters that can then be used on a separate set of testing videos. This approach helps provide bounds on what motion estimation methods can achieve. In no-reference optimization, the true displacement field is not available. By optimizing for the agreement between different motion estimation techniques, the no-reference approach closely approximates the best (optimal) motion parameters.

The results obtained with the newly developed global no-reference optimization approach agree closely with those produced with the full-reference approach. Moreover, the no-reference approach calculates velocity fields of superior quality than published results for benchmark video sequences. Unreliable velocity estimates are identified using new confidence maps that are associated with the disagreement between methods. Thus, the no-reference global optimization method can provide reliable motion estimation without the need for realistic simulations or access to ground truth.

The methods developed in this dissertation are applied to ultrasound videos of carotid artery plaques. The velocity estimates are used to analyze plaque motion and produce novel non-invasive elasticity maps that can help in the identification of vulnerable atherosclerotic plaques.

Table of contents

List of Figures

xxii

List of Tables

Chapter 1

Introduction

The general problem of recovering object motion from digital video using only image intensity information without any additional information regarding nature of the motion is an ill-posed problem. While extensive research has been dedicated to improve the accuracy of the vector flow estimates and many techniques [1-11] have been proposed; in all cases, there is a need for adding regularization terms and parameters to solve the ill-posed problem. In an effort to better define the problem; more constraints have been introduced in the form of energy functionals. As the number of regularization terms and parameters has increased, so has the complexity of the problem. Therefore, instead added constraints aiding in the solution to the problem, these same constraints have become one of the sources of inaccuracies [7, 12-15].

Current solutions to motion estimation, using optical flow models, estimate the displacement field that minimizes the value of a combined energy functional regardless of the optimality of this solution with respect to each individual functional or constraint.

This dissertation implements a global optimization strategy to calculate the regularization and any other motion parameters that associated with pixel-based motion estimation. In **full-reference** optimization, the assumption is the true displacement field is available. Global optimization is then used to help calculate the motion parameters that produce the best velocity estimates by minimizing the mean squared error between the estimates and

the ground truth velocities. In **no-reference** optimization, the true displacement field is not available and error metrics can not be minimized. The no-reference assumption is that two motion estimation techniques are able to provide reliable velocity estimates. By optimizing for the agreement between the methods, the no-reference approach calculates the best motion parameters that accurately estimate the unknown motion field. A confidence map of the motion estimates is computed based on the pixel-wise agreement between the methods.

1.1 Motivation for current work

Pixel-based motion estimation calculation using optical flow models has been extensively researched. The most common models are based on the conservation of image brightness intensities [6, 8, 10, 16, 17] along with some smoothing or regularization term that constraints the admissible motion field. While much attention has been paid to improving the theoretical and numerical framework, there has been only a small effort [7, 14] in applying optimization methods to current optical flow based techniques as well as optimization methods that find the optimal values of the model's regularization parameters [7, 12-15]. The complexity of the methods has grown so much in recent years that there is no guarantee that the motion estimates are optimal with respect to the energy functionals and regularization parameters that govern the weight of the constraints in the admissible motion field. Additionally, there is little information provided in the literature on how to optimize for the parameters, so the standard approach of heuristically choosing the weight of the regularization constraints has become another source of estimation error [7, 12-15].

The principal motivation of this dissertation is to develop global optimization methods that provide optimal parameters for motion estimation by developing a general platform where currently available techniques can be tested and optimized. Furthermore, this research will also be extended to video applications where there is no ground truth available to use for validating the velocity estimates. This is a problem present in all real video sequences that are used as input to motion estimation algorithms, and it is also an open problem in biomedical applications like motion analysis of ultrasound videos (US) of carotid artery (CA) plaques. Thus, the results of this dissertation will be used to obtain reliable US video motion of atherosclerosis plaques as an aid to clinical diagnosis.

1.2 Related Work on Motion Estimation Parameter Selection

Optical flow models for motion estimation use a smoothing constraint to uniquely determine the velocity field components of the model solution. The weight of the smoothness constraint is controlled by the value of the regularization parameter in order to find an appropriate trade off among the model energy functionals. With the exception of the approaches discussed in this section, the most commonly method for choosing the correct value of the regularization parameter is based on a heuristic trial and error approach.

1.2.1   The Data-driven Approach

Assuming that the image brightness intensity of a point that moves on a certain pattern does not change in time, the data-driven approach [15] starts by assuming the conservation of image brightness intensities:

$$\frac{d}{dt} I(x, y, t) = 0. \tag{1.1}$$

Applying the chain rule of differentiation to (1.1) gives:

$$\frac{d}{dt} I(x, y, t) = \frac{\delta I}{\delta x}\frac{dx}{dt} + \frac{\delta I}{\delta y}\frac{dy}{dt} + \frac{\delta I}{\delta t}\frac{dt}{dt} = 0, \tag{1.2}$$

and defining the optical flow field to be $\vec{v} = (dx/dt, dy/dt)^T$, equation (1.2) can be written in compact form as:

$$I_x u + I_y v + I_t = 0, \tag{1.3}$$

where $I_x, I_y,$ and I_t are the spatial and temporal image partial derivatives. The data-driven approach adds extra error terms to (1.3) that are modeled as independent Gaussian random variables yielding to:

$$-I_t = I_x u + I_y v + \delta, \tag{1.4}$$

where δ is $N(0, \sigma^2 \mathbf{I_n})$ with zero mean, variance noise of σ^2. A suitable performance measure needs to be defined to assess the accuracy of the estimates. Here, a weighted error norm or estimated risk is defined by:

$$R_\alpha = n^{-2} E \left[\left\| W \left(\hat{f} - f \right) \right\|^2 \right], \tag{1.5}$$

where

$$\begin{aligned} f &= [u, v]^T, \\ W &= \left[diag(I_x) \quad diag(I_y) \right], \end{aligned} \tag{1.6}$$

and $\hat{f}$ is the optical flow estimate obtained by minimizing the combined constant brightness model energy functional proposed by Horn [1] (also see table 2.1):

$$E_{CBM} = \iint_{\Omega} \left[\left(I_x u + I_y v + I_t \right)^2 + \alpha \left(\|\nabla u\|^2 + \|\nabla v\|^2 \right) \right] dxdy. \tag{1.7}$$

The unbiased estimator of R_α (1.5) is given by:

$$\hat{R}_\alpha = -\sigma^2 + \frac{1}{n^2} \|e\|^2 - \frac{2\sigma^2}{n^2} trace\left(WA^{-1}W^T \right), \tag{1.8}$$

where $A = n^2 \left(E_{CBM} \right)_{\tilde{f}\tilde{f}} / 2$, assuming square images n^2 is the number of pixels, and the matrix $\left(E_{CBM} \right)_{\tilde{f}\tilde{f}}$ can be computed by discretizing (1.7) and taking derivatives with respect to each component of $f = [u, v]^T$.

If σ^2 is known, then $\hat{R}_\alpha$ can be directly calculated and the optimal regularization parameter according to (1.8) is the α that minimizes $\hat{R}_\alpha$. If σ^2 is not known, then it can be replaced by an estimate [15].

The proposed noise model in (1.4) does not consider that finite differencing approximations to image derivatives and violations to the conservation of image brightness intensities which introduce a noise correlation component that is not modeled. However, the main disadvantage of the data driven approach is that it requires noise free spatial image derivatives [15] which is not a realistic assumption. Thus, their performance measure (1.8) is invalid because it explicitly depends on their noise model. Furthermore, a discretization of the energy functional (1.7) is required to calculate its derivatives with respect to each velocity component. This methodology now introduces errors in the minimization of (1.8) which arguably leads to a good estimate of the optimal regularization parameters. Finally, the cost of calculating $trace\left(WA^{-1}W \right)$ leads to a very time consuming algorithm even for small images as recognized in [15].

5

1.2.2 The Maximum Likelihood Regularization Parameter Estimator Method

Developed in [12, 13], this method uses a combined marginal maximum likelihood / maximum a posteriori (MML/MAP) estimator for estimating optical flow and simultaneously choosing the regularization parameter value in a Bayesian framework .

As with most differential approaches to motion estimation, the conservation of image intensities is the starting point to compute motion vectors leading to:

$$\vec{g}^T \vec{v}_h = 0, \tag{1.9}$$

where $\vec{g} = \left(I_x, I_y, I_t\right)^T$ and $\vec{v}_h = \left(u, v, 1\right)^T$. Since it is not possible to estimate $\vec{v}_h$ from a single equation with two unknowns, the necessary additional constraint is incorporated by a regularization term $\rho(\vec{v})$, where ρ is an operator acting on $\vec{v} = \left(u, v\right)^T$, that imposes a smoothing constraint on permissible motion fields. The optical flow field is estimated by minimizing:

$$E_{CBM} = \iint_{\Omega} \left(\psi\left(\vec{g}^T \vec{v}_h\right) + \lambda \rho(\vec{v})\right) dxdy, \tag{1.10}$$

where ψ is a real positive function and the regularization parameter λ controls the influence of the regularization term $\rho(\vec{v})$ relative to the data term $\psi\left(\vec{g}^T \vec{v}_h\right)$.

In the Bayesian formulation, the optical flow is estimated using a probability density function (pdf). In order to design the pdf, a regular grid in space-time is assumed where image intensities and optical flow vectors are only considered if they lie on the knots of the grid. Then, for N knots:

$$\vec{I} = \left(I(x_1), I(x_2), \ldots, I(x_N) \right),$$
$$\vec{u}_c = \left(u(x_1), u(x_2), \ldots, u(x_N) \right),$$
$$\vec{v}_c = \left(v(x_1), v(x_2), \ldots, v(x_N) \right), \tag{1.11}$$
$$\vec{v} = (u, v)^T.$$

In this framework, not only the measured image gradients $\vec{g} = \left(g(x_1), g(x_2), \ldots g(x_N) \right)$,

but also the estimated field $\vec{v}$ are considered random variables with pdfs $p(\vec{v})$ and

$p(\vec{g})$. Prior knowledge about $\vec{v}$ is incorporated into the estimation via the prior pdf

$p(\vec{v})$. The maximum a posteriori (MAP) estimator infers the optical flow field by

maximizing the posterior $p(\vec{v} \mid \vec{g})$ [13]. Using Bayes theorem:

$$\hat{\vec{v}} = \arg\max_{\vec{v}} \left\{ \frac{p(g \mid \vec{v}) p(\vec{v})}{p(g)} \right\},$$
$$\hat{\vec{v}} = \arg\min_{\vec{v}} \left\{ -\ln\left(p(\vec{g} \mid \vec{v}) \right) - \ln\left(p(\vec{v}) \right) \right\}. \tag{1.12}$$

The errors ε in the brightness constancy assumption are attributed only to the temporal

component of the image gradient by writing:

$$g_{sj}{}^T \vec{v}_{hj} = \varepsilon_{tj}, \tag{1.13}$$

where $\vec{g}_{sj} = \left(I_{xj}, I_{yj} \right)^T$ is the error free vector of spatial image gradient components.

Expressing each random variable ε_{tj} in the joint pdf $p(\vec{\varepsilon}) = \prod_{j=1}^{N} p(\varepsilon_{tj})$, the likelihood

function $p(\vec{g}_t \mid \vec{v}, \vec{g}_s)$ is given by:

$$p(\vec{g}_t \mid \vec{v}, \vec{g}_s) = \frac{1}{Z_L(\alpha)} \exp\left[-\alpha \sum_{j=1}^{N} \psi_1\left(\vec{g}_j \vec{v}_{hj} \right) \right]. \tag{1.14}$$

The prior $p(\vec{v})$ encodes the prior information/assumptions about the optical flow, i.e. the smoothness of the field, and then the *pdf* corresponding to the smoothness assumption is given by:

$$p(\vec{v}) = \frac{1}{Z_p(\vec{\beta})} \exp\left[-\sum_{j=1}^{N} \beta_x \psi_2\left(u_x^2 + u_y^2\right) + \beta_y \psi_2\left(v_x^2 + v_y^2\right)\right], \qquad (1.15)$$

where $\vec{\beta} = \left(\beta_x, \beta_y\right)^T$ in the more general anisotropic case.

After substituting (1.14) and (1.15) in (1.12), the objective function $E^{(N)}$ is given by:

$$E^{(N)} = E_L(\alpha) + E_p(\beta) + \ln\left(Z_L(\alpha) Z_p(\beta)\right). \qquad (1.16)$$

The regularization parameter $\lambda = \beta / \alpha$, in the isotropic case $\beta = \beta_x = \beta_y$, is expressed in terms of the hyper parameters α, and β which are the parameters of the prior distributions. Subtracting $\ln\left(Z_L(\alpha) Z_p(\beta)\right)$ from (1.16), taking the limit and dividing by α gives:

$$\begin{aligned}
E &= \lim_{N \to \infty}\left(E^{(N)} - \ln\left(Z_L(\alpha) Z_p(\beta)\right)\right) \\
&= \iint_{\Omega}\left(\alpha \psi_1\left(\vec{g}^T \vec{v}_h\right) + \beta \psi_2\left(\|\nabla u\|^2 + \|\nabla v\|^2\right)\right) dx dy \qquad (1.17) \\
&= \iint_{\Omega}\left(\psi_1\left(\vec{g}^T \vec{v}_h\right) + \lambda \psi_2\left(\|\nabla u\|^2 + \|\nabla v\|^2\right)\right) dx dy.
\end{aligned}$$

When the objective function is minimized, rather than choosing ad-hoc values for α, and β these are inferred using prior knowledge of the flow field, i.e ground truth is used to compute moments of functions of the motion field. Using the maximum entropy principle [18], the hyper parameters are calculated by constructing the prior $p(\vec{v})$ and incorporating all the constraints imposed on $\vec{v}$.

The advantage of this method is that the use of probabilistic models for motion estimation produces a method for estimating regularization parameters through velocity distributions. However, image motion requires specific spatio-temporal information that is not easily described through this approach. As an example, it is not clear how this model will be used to effectively describe a variable rate expansion field [19]. Furthermore, for more accurate estimation specific prior knowledge needs to be incorporated into the method. It is not clear how to produce such prior knowledge when dealing with video sequences without ground truth. This is a drawback to the scope of the MML/MAP estimator. Instead, this dissertation proposes to calculate a globally optimal solution to the estimation of motion parameters; where for different types of motion even if no ground truth is available, an approximate interval of parameters will lead to accurate estimations.

1.3 Regularization parameter choice for inverse problems

Inverse problems involve the estimation of unknown quantities of interest based on some related observations. Noise can lead to significant errors in the estimation, and the observations could not guarantee a unique solution if one exists. These types of systems are known as ill-posed. In order to deal with these systems, regularization, a mathematical operator, was developed to better constraint the problems and find a suitable solution.

In image and video processing, ill-posed inverse problems are common. Image restoration is arguably one of the most common examples. The goal is to recover a

desired unknown image $f(x,y)$ based on a distorted observation $g(x,y)$. The degradation process is modeled with a forward operator H and additive noise $\eta(x,y)$.

The observed and unknown images $g(x,y)$ and $f(x,y)$ are made up of individual pixels, g_i and f_i, and using matrix-vector notation the image restoration problem can be written as:

$$g = Hf + \eta \tag{1.18}$$

The purpose of regularization is to allow prior knowledge to be used in the inverse model and to avoid noise dominated reconstructions [20]. Thus, the regularized estimate is the solution to the following minimization problem:

$$\hat{f}(\alpha) = \arg\min \|g - Hf\|_2^2 + \alpha^2 \|Lf\|_2^2 \tag{1.19}$$

The first term, $J_1(f,g) = \|g - Hf\|_2^2$, ensures fidelity to the unknown image, and the second term, $J_2(f) = \|Lf\|_2^2$, incorporates prior knowledge about f penalizing departures from some expected behavior trough the regularization operator L. The smoothing parameter, α, controls the trade off between data fidelity and regularization terms.

Inverse problems like image restoration are very common in the literature [21], and methods to estimate the correct value of the regularization parameter have already been developed [22-24]. These interactions are similar to the ones found in motion estimation models. Thus, it is necessary to explore possible adaptations of the following methods to set the value of the regularization parameter in the solution of velocity vector fields using optical flow models.

1.3.1 The discrepancy principle

Knowledge of the noise variance λ_n allows the discrepancy principle to bound the residual norm to:

$$\left\| g - Hf \right\|_2 \leq \lambda_q. \tag{1.20}$$

When the regularization parameter α is set to a small value, the result is excessive noise amplification [20]. Thus, the idea is to make the value of α large enough to achieve the equality:

$$\left\| g - H\hat{f}\left(\alpha\right) \right\|_2 = \lambda_q. \tag{1.21}$$

Generally, the regularization parameter provided by the discrepancy principle leads to an over-regularized solution since the actual noise variance may be smaller than the used in (1.21).

1.3.2 The L-curve method

The L-curve method [25] involves the analysis of the trade off between the fidelity data term and the regularization term. A log-log scale plot of the behavior of these two terms as α varies is called the L-curve because it exhibits a characteristic "L" shape, and the optimal value of the regularization parameter, α^*, is considered to be at the corner of the curve (figure 1.1). This corner represents the transition between over and under-regularized solutions and it is achieved for α^*.

Possible definitions for the corner of the curve include the point of maximum curvature and a point of tangency with a slope of -1. This last definition leads to a α value that satisfies:

11

$$\alpha^2 = \frac{\left\| g - H\hat{f}(\alpha) \right\|_2^2}{\left\| L\hat{f}(\alpha) \right\|_2^2}. \tag{1.22}$$
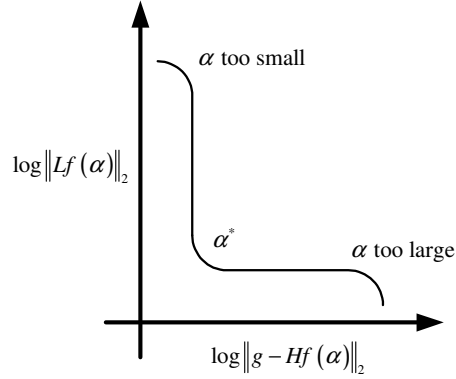
Figure 1.1. Shape of the L-curve

1.4 Thesis statement

This dissertation will demonstrate that global optimization methods can be used to provide accurate pixel motion estimates of video sequences. First, a full-reference global optimization method will be developed for estimating the best parameters in realistic motion simulations (training set). Second, a no-reference global optimization method will be developed for estimating the best motion parameters in synthetic benchmark videos, real video sequences, and clinical videos of ultrasound atherosclerotic plaques. For the benchmark and clinical video simulations, agreement between the full-reference and no-reference approaches will be demonstrated. Applications of the optimization approaches will be shown on motion estimation and strain imaging of atherosclerotic plaques.

1.5 List of Contributions

The contributions of this dissertation include:

12

- The development of a full-reference global optimization methodology for estimating the best regularization and other relevant motion parameters in the computation of motion vectors when ground truth is available.

- The development of a no-reference global optimization method when no ground truth is available. Unreliable velocity estimates are identified using new confidence maps that are associated with the disagreement between methods. Thus, the no-reference global optimization method can provide reliable motion estimation without the need for realistic simulations or access to ground truth

- The agreement between the full-reference and no-reference methods on standard benchmark videos and clinical ultrasound simulations.

- The application of the optimization approaches to motion estimation and strain imaging of ultrasound atherosclerotic plaques.

1.6 Dissertation Overview

This section contains a brief description of the all the work that is presented in this dissertation.

- Chapter 2 presents background on pixel based motion estimation methods using optical flow models. Global and local solutions using optimization theory are presented starting from the original work of Horn and Schunk [1], until more recently developed methods. The review will cover a period of three decades of active research in the area.

- Chapter 3 describes the full-reference global optimization approach and provides comparative results on standard benchmark videos.

- Chapter 4 describes the no-reference global optimization method and provides comparative results with the full-reference approach.

- Chapter 5 provides applications on motion and strain imaging of atherosclerotic ultrasound plaques.

- Chapter 6 gives concluding remarks.

Chapter 2

Optical Flow Model Solutions using Optimization Theory

This chapter presents an overview of optical flow based methods for motion estimation with special focus on optimization theory applied towards the minimization of energy functionals. Global and local solutions are studied using the calculus of variations and least-squares methods.

2.1 Overview of motion estimation and optical flow

The study of motion estimation from digital video will be restricted to the process of identifying pixel correspondences between different frames and assigning a velocity vector to each point in the image. However, note that 2-d motions are not always observable [26]. Instead, only the apparent motion of pixels is observed. Optical flow is the perceived motion of image intensity brightness patterns and is used to approximate the 2-d motion field of objects from a 3-d scene.

The work of Horn and Schunk [1] opened up the area of motion estimation using optical flow models with the classical assumption that the image intensity should be conserved under small displacements. At the same time, Lucas and Kanade [6] proposed a local solution to the conservation of image intensities using a weighted least squares fit. Five years later, Nagel [8] proposed an oriented smoothness constraint for estimating the

optical flow motion field. The conservation of the image intensities was applied to the image gradient by Uras et. al [10], and Fleet and Jepson [3] were the first to use image phase information in the estimation of the motion field.

At this point, Barron et. al [27] published a comprehensive survey of optical flow techniques. This work gave quantitative comparisons among differential methods (eg: [1], [6], [8], [10]), block matching [16], energy based [4], and phased based techniques ([3], [11]). For the comparisons, Barron et. al. [27] used synthetic images for which ground truth was available. Estimation performance was summarized in terms of the density of the flow field (percentage of the number of pixels with motion estimates as a fraction of all the pixels) and the angular error between ground truth and the estimated field.

Negahdaripour [9] modified the framework to allow changes in the pixel brightness intensities along their motion path. Then, local and global solutions to motion estimation using optical flow models were merged by Bruhn and finally. More recently, the intensity conservation assumption was assumed over multiple scales in conjunction with amplitude modulation frequency modulation (AM-FM) image models [7]. A complete list of each method assumptions and energy functionals are shown in table 2.1 and table 2.2.

Table 2.1. Energy Functionals for Optical Flow Techniques

Authors	Assumptions	Energy Functionals	Comments
Horn & Schunk 1981 [1]	• Intensity conservation. • Small displacements. • Field smoothness.	E_{OFC}, E_S	Adv.: Simple linear model. Disadv.: Intensity conservation often violated. Errors at object boundaries.
Lucas & Kanade 1981 [6]	• Weighted intensity conservation. • Small displacements. • Constant motion over blocks.	E_{WOFC}	Adv.: Noise Robustness Disadv.: Low field density.
Nagel H.H. 1986 [8]	• Intensity conservation. • Small displacements. • Oriented smoothness.	E_{OFC}, E_{OS}	Adv.: Better handling of Occlusion. Disadv.: Approximation of second order derivatives.
Uras et al. 1988 [10]	• Conservation of image gradient. • No rotation or dilation of image intensities. should be present	E_G	Adv.: One constraint that provides two equations per pixel. Disadv.: Low field density. Requires approximation of second order derivatives
Fleet & Jepson 1990 [3]	• Conservation of phase information. • Small displacements.	E_{Phase}	Adv.: Phase information is more robust than amplitude (intensity) information. Disadv.: Filter is tuned to only certain frequencies.
Negahdaripour 1998 [9]	• Non constant Brightness • Small displacements. • Field smoothness • Radiometric smoothness	E_{NCB} E_S E_{Srad}	Adv.: Realistic Model Disadv.: Underdetermined model that requires smoothness across velocity and scene variation parameters.
Bruhn et al. 2002 [2]	• Local-Global intensity conservation • Small displacements.	E_{CLG}	Adv.: Combines global and local Optical Flow approaches Disadv. Integration scale of local-global region is chosen heuristically.
Murray & Pattichis 2007 [7]	• Image intensity preservation translated into AM and FM constraints. • Assumes motion continuity	E_{AM} E_{FM}	Adv.: Three equations for each channel filter per pixel. Disadv.: Filterbank frequency should match the underlying motion to be estimated.

Table 2.2.Optical flow energy functionals used to constraint the motion present in a 2D scene.

Authors	Parameter
Horn & Schunk 1981 [1]	• Spread of Gaussian smoothing filter σ. • Weight of smoothness constraint α.
Lucas & Kanade 1981 [6]	• Spread of Gaussian smoothing filter σ.
Nagel H.H. 1986 [8]	• Spread of Gaussian smoothing filter σ. • Weight of smoothness constraint α.
Uras et al. 1988 [10]	• Spread of Gaussian smoothing filter σ.
Fleet & Jepson 1990 [3]	• Spread of Gaussian smoothing filter σ. • Threshold on local frequencies outside filter's tuning range.
Negahdaripour 1998 [9]	• Spread of Gaussian smoothing filter σ. • Weight of smoothness constraint λ • Weight of radiometric smoothness constraint β.
Bruhn et al. 2002 [2]	• Noise scale ρ.
Murray & Pattichis 2007 [7]	• Weight of AM contraint α_{AM} • Weight of FM constraint β_{FM}.

2.2 Constant brightness optical flow model for motion estimation

Let the image brightness intensity of a pixel with coordinates (x, y) at time t, be denoted as $I(x, y, t)$. Optical flow techniques assume that the image brightness intensities remain constant along their motion path under a sufficiently small time interval δt giving:

$$I(x + \delta x, y + \delta y, t + \delta t) = I(x, y, t). \qquad (2.1)$$

18

Expanding the left hand side of (2.1) into a first order Taylor series, and ignoring high order terms gives:

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

After subtracting $I(x,y,t)$ from both sides of (2.2), dividing by δt, and taking the limit as $\delta t \rightarrow 0$, equation (2.3) becomes the optical flow constraint (OFC):

$$I_x u+I_y v+I_t =0.$$ (2.3)

Global deviations from the OFC are measured using the following energy functional

$$E_{OFC} = \iint_{\Omega} \left(I_x u+I_y v+I_t\right)^2 d\Omega$$ (2.4)

which is minimized over the entire image domain Ω.

Alternatively, the OFC can be rewritten as:

$$\nabla \vec{I} \cdot \vec{v} = -I_t$$ (2.5)

where $\nabla I = (I_x, I_y)$ is the spatial intensity gradient, I_t is the first order temporal derivative, and $\vec{v}$ is the optical flow field with components given by $u = dx/dt$ and $v = dy/dt$.

The solution to equation (2.5) is the velocity component in the direction of the spatial intensity gradient given by:

$$\vec{v}_n = \frac{-I_t \nabla \vec{I}}{\|\nabla I\|_2^2}.$$ (2.6)

This is known as the aperture problem; only the velocity component in the direction of the spatial intensity gradient can be estimated with the OFC. The component orthogonal to this direction, which is the component along contours of constant brightness intensity,

can not be determined. The velocity in the direction of the spatial gradient (2.6) is known as the normal velocity i.e. the flow normal or orthogonal to local intensity structure, and only at locations where there is sufficient intensity structure full image velocity can be estimated [28]. The OFC is a single equation in two unknowns. The constraint can be expressed as a line in the velocity space and all the points (u, v) on that line are valid solutions. The normal velocity is the solution with the smallest magnitude on that line, and the problem of estimating full image velocity yields to introducing additional constraints.

2.3 Non constant brightness intensity optical flow model for motion estimation

Geometric transformations provide only one set of changes that result in optical flow changes. Brightness changes induced by non-uniform light source, light source changes, shading, and surface reflection are present in time-varying images and all induce optical flow changes that violate the brightness constancy assumption. To overcome the shortcomings of the image brightness constancy model, Negahdaripour [9] proposed a more general model for optical flow estimation that allows brightness intensity variations between successive time instants.

The brightness intensity change of a pixel after a small time interval can be modeled as

$$m(x, y, t) I(x, y, t) + c(x, y, t) = I(x + \delta x, y + \delta y, t + \delta t) \tag{2.7}$$

where the new terms $m(x, y, t)$ and $c(x, y, t)$ allow the approximation of brightness variations as a linear transformation of the intensity patterns using a multiplier $m(x, y, t)$ and offset $c(x, y, t)$ fields.

Expanding the right hand side of (2.7) in a first order Taylor series and neglecting high order terms, yields to

$$m(x,y,t)I(x,y,t)+c(x,y,t)=I(x,y,t)+\delta x\frac{\partial I}{\partial x}+\delta y\frac{\partial I}{\partial y}+\delta t\frac{\partial I}{\partial t} \qquad (2.8)$$

for small time intervals, δt, the multiplier field is expected to be close to one, and the offset field is expected to be close to zero. This allows to approximate the multiplier and offset fields using $m(x,y,t)=1+\delta m$ and $c(x,y,t)=0+\delta c$. Equation (2.8) is reduced to:

$$I\delta m+\delta c=I_x\delta x+I_y\delta y+I_t\delta t. \qquad (2.9)$$

After dividing by δt, and taking the limit as $\delta t\rightarrow 0$ the Generalized Dynamic Image Model (GDIM) is expressed as:

$$I_x u+I_y v+I_t-Im_t-c_t=0. \qquad (2.10)$$

Equation (2.10) reduces to the OFC when $m_t=c_t=0$.

Deviations from the GDIM are penalized using the following energy functional

$$E_{NCB}=\iint_\Omega\left(I_x u+I_y v+I_t-m_t I-c_t\right)^2 d\Omega. \qquad (2.11)$$

2.4 Global Solutions via functional optimization

Both the constant brightness model and the non constant brightness intensity model are underdetermined systems. Global methods assume that the motion field is continuous and differentiable in space and time [28] along with a global smoothness regularization term to compute dense optical flow estimates. If pixels in an image move independently, there

is little hope to recover the flow field [1]. In what follows, the assumption is that velocity field of the image varies smoothly over the spatial support of the image.

2.4.1 Constant brightness model (CBM) solution

The OFC (2.3) is not enough to fully estimate both components of the flow field (u, v) since it is a single equation with two unknowns per pixel. Additional constraints need to be introduced to the model formulation in order to calculate the optical flow field. Horn and Schunk [1] measured the smoothness of the field by minimizing the square of the magnitude of the gradient of the velocity vector over the whole image in the following energy functional:

$$E_S = \iint_\Omega \left(u_x^2 + u_y^2 + v_x^2 + v_y^2 \right) d\Omega. \tag{2.12}$$

The solution to the CBM is the velocity field that minimizes the sum of the errors in the following combined functional:

$$
\begin{aligned}
E_{CBM} &= E_{OFC} + \alpha^2 E_S, \\
E_{OFC} &= \iint_\Omega \left(I_x u + I_y v + I_t \right)^2 d\Omega, \\
E_S &= \iint_\Omega \left(u_x^2 + u_y^2 + v_x^2 + v_y^2 \right) d\Omega.
\end{aligned}
\tag{2.13}
$$

In (2.13), α is a parameter that weights the departure from the smoothness functional, E_S, relative to the optical flow constraint functional E_{OFC}. If brightness measures are accurate, this parameter should be small and large if they are noisy [17].

Quantization error and noise affect the calculation of image intensity derivatives. Thus, the OFC will not always equal to zero. Meanwhile, E_S only vanishes for constant velocities, i.e. $u_x = u_y = v_x = v_y = 0$. For non-constant velocities, the smoothness

22

functional is non-zero. The weight on the smoothness constraint is supposed to account for some of these errors. Clearly though, the choice of the correct regularization parameter is critical and optimizing for the best value is not a trivial task as shown in our previous work [7, 14, 29].

Minimizing a combined functional of the form of (2.13) is a problem for the calculus of variations [30, 31]. The associated Euler-Lagrange equations are:

$$
\begin{aligned}
I_x^2 u + I_x I_y v &= \alpha^2 \nabla^2 u - I_x I_t \\
I_x I_y u + I_y^2 v &= \alpha^2 \nabla^2 v - I_y I_t
\end{aligned}
\tag{2.14}
$$

where finite differencing is used to approximate the Laplacian of the velocities. System (2.14) depends on the regularization parameter α, and a suitable value needs to be chosen in order to find a solution. The most of common approach is a heuristic trial and error process that leads to inaccuracies in the estimated vector field [7, 13-15].

There are two equations per pixel and solving (2.14) using one of the standard methods like Gauss-Jordan elimination will be very costly. Thus, an iterative scheme [1] is used to calculate the new velocity values $\left(u^{k+1}, v^{k+1}\right)$ from local velocity neighborhood averages at the previous iteration $\left(\bar{u}^k, \bar{v}^k\right)$ and brightness intensity derivatives using:

$$
\begin{aligned}
u^{k+1} &= \bar{u}^k - \frac{I_x \left[I_x \bar{u}^k + I_y \bar{v}^k + I_t \right]}{\alpha^2 + I_x^2 + I_y^2}, \\
v^{k+1} &= \bar{v}^k - \frac{I_y \left[I_x \bar{u}^k + I_y \bar{v}^k + I_t \right]}{\alpha^2 + I_x^2 + I_y^2}.
\end{aligned}
\tag{2.15}
$$

Numerical differentiation is always a source of considerable error and authors in [27] applied a spatiotemporal Gaussian pre-smoothing filter to enforce continuity of the image brightness patterns and attenuate the effects of noise. However, too much smoothing

destroys local image structure, and some optimization will be needed in order to control the spread of the Gaussian filter.

2.4.2 Constant Brightness Model with Oriented Smoothness Constraint

In this model, as in the original approach by Horn and Schunk [1], the OFC is combined with a smoothness constraint. Here, the smoothness constraint was modified to limit the variation of the motion field [32]. The variation of the motion field is captured in the first derivatives of the two velocity components.

Nagel accomplished this by minimizing the flow variation that is orthogonal to the image gradient ∇I. The oriented smoothness constraint is formulated as:

$$E_{OS} = \iint \frac{\left[\left(u_x I_y - u_y I_x \right)^2 + \left(v_x I_y - v_y I_x \right)^2 + \delta \left(u_x^2 + u_y^2 + v_x^2 + v_y^2 \right) \right]}{\left\| \nabla I \right\|^2 + 2\delta} dxdy. \qquad (2.16)$$

The solution to the CBM with oriented smoothness constraint is the flow field (u, v) that minimizes the following combined functional:

$$
\begin{aligned}
E_{CBMOS} &= E_{OFC} + \alpha^2 E_{OS}, \\
E_{OFC} &= \iint_{\Omega} \left(I_x u + I_y v + I_t \right)^2 d\Omega, \qquad\qquad (2.17)\\
E_{OS} &= \iint_{\Omega} \frac{\left[\left(u_x I_y - u_y I_x \right)^2 + \left(v_x I_y - v_y I_x \right)^2 + \delta \left(u_x^2 + u_y^2 + v_x^2 + v_y^2 \right) \right]}{\left\| \nabla I \right\|^2 + 2\delta} dxdy.
\end{aligned}
$$

The parameter α weights the departure from the oriented smoothness constraint relative to the error in the OFC, and δ was introduced for numerical stability and equals to 1.0 [27]. Using the Gauss-Seidel iterations, the solution can be expressed as:

24

$$u^{k+1} = \xi\left(u^k\right) - \frac{I_x\left(I_x\xi\left(u^k\right) + I_y\xi\left(v^k\right) + I_t\right)}{I_x^2 + I_y^2 + \alpha^2},$$

$$v^{k+1} = \xi\left(v^k\right) - \frac{I_y\left(I_x\xi\left(u^k\right) + I_y\xi\left(v^k\right) + I_t\right)}{I_x^2 + I_y^2 + \alpha^2}.$$

(2.18)

In (2.18), k denotes the iteration number, Here, $\xi\left(u^k\right)$ and $\xi\left(v^k\right)$ are given by:

$$\xi\left(u^k\right) = \bar{u}^k - 2I_xI_y u_{xy}^{\ k} - q^T\left(\nabla u^k\right), \text{ and}$$

$$\xi\left(v^k\right) = \bar{v}^k - 2I_xI_y v_{xy}^{\ k} - q^T\left(\nabla v^k\right),$$

(2.19)

Where q and W are given by:

$$q = \frac{1}{I_x^2 + I_y^2 + 2\delta}\nabla I^T\left[\begin{pmatrix} I_{yy} & -I_{xy} \\ -I_{xy} & I_{xx} \end{pmatrix} + 2\begin{pmatrix} I_{xx} & I_{xy} \\ I_{xy} & I_{yy} \end{pmatrix}W\right], \text{ and}$$

(2.20)

$$W = \left(I_x^2 + I_y^2 + 2\delta\right)^{-1}\begin{pmatrix} I_y^2 + \delta & -I_xI_y \\ -I_xI_y & I_x^2 + \delta \end{pmatrix}.$$

(2.21)

Here, u_{xy}^k, v_{xy}^k denote estimates of the partial derivatives. Also, $\bar{u}^k$ and $\bar{v}^k$ denote local neighbourhood averages of u^k and v^k.

2.4.3 Generalized Dynamic Image Model (GDIM) Solution

Conservation of image intensity is the basis of most optical flow techniques but when intensities are not constant, true in most real video sequences, the OFC is severely violated yielding unreliable estimates. To overcome the shortcomings of the image brightness constancy model, a generalized brightness variation model was proposed by Negahdaripour [9] where not only image geometric transformations are considered but also transformations of image brightness patterns (2.11). Originally, Negahdaripour [9]

developed a local solution for the GDIM using a least square fit over block regions of 9×9 pixels. Such local solutions do not yield high density fields because of regions where the least square system is ill-conditioned due to lack of or weak texture. In this paper, a global solution for the GDIM is computed using the calculus of variations.

Errors in the GDIM are penalized using the non-constant brightness energy functional E_{NCB}:

$$E_{NCB} = \iint_\Omega \left(I_x u + I_y v + I_t - m_t I - c_t \right)^2 d\Omega. \tag{2.22}$$

Assuming that the unknowns (u, v, m_t, c_t) vary smoothly, the minimization problem can be formulated as one of estimating the functions $u, v, m_{t,} c_t$ that minimize the combined energy functional:

$$E_{GDIM} = \lambda E_{NCB} + \beta E_{Srad} + E_S. \text{ Where}$$
$$E_{NCB} = \iint_\Omega \left(I_x u + I_y v + I_t - m_t I - c_t \right)^2 d\Omega,$$
$$E_{Srad} = \iint_\Omega \left(m_{tx}^2 + m_{ty}^2 + c_{tx}^2 + c_{ty}^2 \right) d\Omega, \tag{2.23}$$
$$E_S = \iint_\Omega \left(u_x^2 + u_y^2 + v_x^2 + v_y^2 \right) d\Omega.$$

The system of associated Euler-Lagrange differential equations is given by:

$$\lambda \left(I_x u + I_y v + I_t - m_t I - c_t \right) I_x = \nabla^2 u,$$
$$\lambda \left(I_x u + I_y v + I_t - m_t I - c_t \right) I_y = \nabla^2 v,$$
$$-\lambda \left(I_x u + I_y v + I_t - m_t I - c_t \right) I = \beta \nabla^2 m_t, \text{ and} \tag{2.24}$$
$$-\lambda \left(I_x u + I_y v + I_t - m_t I - c_t \right) = \beta \nabla^2 c_t.$$

The Laplacian of the velocities is approximated using finite differencing. The final system to solve can be written as:

26

$$
\begin{bmatrix}
\lambda I_x^2 + 1 & \lambda I_x I_y & -\lambda I_x I & -\lambda I_x \\
\lambda I_x I_y & \lambda I_y^2 + 1 & -\lambda I_y I & -\lambda I_y \\
-\lambda I_x I_y & -\lambda I_y I & \lambda I^2 + \beta & \lambda I \\
-\lambda I_x & -\lambda I_y & \lambda I & \beta + \lambda
\end{bmatrix}
\begin{bmatrix}
u \\ v \\ m_t \\ c_t
\end{bmatrix}
=
\begin{bmatrix}
\bar{u} - \lambda I_x I_t \\
\bar{v} - \lambda I_y I_t \\
\beta \bar{m}_t + \lambda I_t I \\
\beta \bar{c}_t + \lambda I_t
\end{bmatrix}.
\tag{2.25}
$$

The Gauss-Seidel iteration scheme used by Horn and Schunk to solve system (2.14) is applied to system (2.25). This gives:

$$
\begin{aligned}
u^{k+1} &= \bar{u}^k - I_x \left[\frac{I_x \bar{u}^k + I_y \bar{v}^k + I_t - I\bar{m}_t^k - \bar{c}_t^k}{I_x^2 + I_y^2 + \beta'(I^2 + 1) + \lambda'} \right], \\
v^{k+1} &= \bar{v}^k - I_y \left[\frac{I_x \bar{u}^k + I_y \bar{v}^k + I_t - I\bar{m}_t^k - \bar{c}_t^k}{I_x^2 + I_y^2 + \beta'(I^2 + 1) + \lambda'} \right], \\
m_t^{k+1} &= \bar{m}_t^{k+1} - \beta' I \left[\frac{I_x \bar{u}^k + I_y \bar{v}^k + I_t - I\bar{m}_t^k - \bar{c}_t^k}{I_x^2 + I_y^2 + \beta'(I^2 + 1) + \lambda'} \right], \text{ and} \\
c_t^{k+1} &= \bar{c}_t^{k+1} - \beta' \left[\frac{I_x \bar{u}^k + I_y \bar{v}^k + I_t - I\bar{m}_t^k - \bar{c}_t^k}{I_x^2 + I_y^2 + \beta'(I^2 + 1) + \lambda'} \right].
\end{aligned}
\tag{2.26}
$$

Here, note that β' and λ' are used and they are defined to be $1/\beta$ and $1/\lambda$ respectively. Then, when β' is large, m_t and c_t equal to zero and the GDIM approximates to the CBM method proposed by Horn. The optimization methods developed in this dissertation proposal will be applied towards finding the optimal values for λ' and β'.

2.4.4   Amplitude Modulation-Frequency Modulation Motion Model

The Amplitude Modulation Frequency Modulation (AM-FM) motion estimation model was recently developed in [7]. The interest for this method arises from the fact that motion vectors can be estimated independently using AM constraints alone, FM

constraints alone, and the combination of both. The influence of the AM-FM constraints is controlled by weights on the energy functional formulation.

Amplitude Modulation Frequency Modulation is a model that represents images in terms of amplitude and phase functions using:

$$I(x, y, t) \approx \sum_{n=1}^{M} a_n(x, y, t) \cos \varphi_n(x, y, t). \tag{2.27}$$

The idea is to let the amplitude modulated (AM) components $a_n(x, y)$ to capture slow-changing spatial variability in the image intensity, as well as the energy attributed to each component. The frequency modulation components (FM) come from the cosine terms, are used to describe texture variations.

For a single AM-FM component approximation to the input video we have:

$$I(x, y, t) = a(x, y, t) \exp(j\varphi(x, y, t)). \tag{2.28}$$

Assuming the OFC, the authors in [7] used equation (2.28) to derive:

$$a_x u + a_y v + a_t = 0, \text{ and} \tag{2.29}$$

$$\varphi_x u + \varphi_y v + \varphi_t = 0. \tag{2.30}$$

Combining (2.29) and (2.30) with a smoothness constraint on the velocity estimates, the problem of estimating the optical flow field (u, v) reduces to minimizing the following combined energy functional over the entire image:

$$
\begin{aligned}
E_{AMFM} &= E_s + \alpha_{AM} E_{AM} + \beta_{FM} E_{FM}, \\
E_s &= \iint_{\Omega} \left(u_x^2 + u_y^2 + v_x^2 + v_y^2 \right) d\Omega, \\
E_{AM} &= \iint_{\Omega} \left(a_x u + a_y v + a_t \right)^2 d\Omega, \\
E_{FM} &= \iint_{\Omega} \left(\varphi_x u + \varphi_y v + \varphi_t \right)^2 d\Omega.
\end{aligned}
\tag{2.31}
$$

As in the case of the CBM and the GDIM models, the solution to (2.31) is computed after approximating derivatives with finite difference methods, and using an iterative scheme to get:

$$
\begin{aligned}
u^{k+1} = \bar{u}^k - \alpha_{AM} & \frac{\left(\varphi_x + \beta_{FM} a_y^2 \varphi_x - \beta_{FM} a_y a_x \varphi_y\right)\bar{u}^k + \varphi_x \bar{v}^k + \left(\beta a_y^2 + 1\right)\varphi_t - \beta_{FM} a_y \varphi_y a_t}{1 + \alpha_{AM}\left(\varphi_y^2 + \varphi_x^2\right) + \alpha_{AM}\beta_{FM}\left(a_y \varphi_x - a_x \varphi_y\right)^2 + \beta_{FM}\left(a_y^2 + a_x^2\right)} \varphi_x \\
-\beta_{FM} & \frac{\left[\alpha_{AM}\left(a_x \varphi_y - a_y \varphi_x\right)\varphi_y + a_x\right]\bar{u}^k + a_y \bar{v}^k - \alpha_{AM} a_y \varphi_y \varphi_t + \left(\alpha_{AM}\varphi_y^2 + 1\right)a_t}{1 + \alpha_{AM}\left(\varphi_y^2 + \varphi_x^2\right) + \alpha_{AM}\beta_{FM}\left(a_y \varphi_x - a_x \varphi_y\right)^2 + \beta_{FM}\left(a_y^2 + a_x^2\right)}
\end{aligned}
\tag{2.32}
$$

$$
\begin{aligned}
v^{k+1} = \bar{v}^k - \alpha_{AM} & \frac{\varphi_x \bar{u}^k + \left(\varphi_y + \beta_{FM} a_x^2 \varphi_y - \beta_{FM} a_x a_y \varphi_x\right)\bar{v}^k + \left(\beta_{FM} a_x^2 + 1\right)\varphi_t - \beta_{FM} a_x \varphi_x a_t}{1 + \alpha_{AM}\left(\varphi_y^2 + \varphi_x^2\right) + \alpha_{AM}\beta_{FM}\left(\varphi_x a_y - \varphi_y a_x\right)^2 + \beta_{FM}\left(a_x^2 + a_y^2\right)} \varphi_y \\
-\beta_{FM} & \frac{a_x \bar{u}^k + \left[\alpha_{AM}\left(a_y \varphi_x - a_x \varphi_y\right)\varphi_x + a_y\right]\bar{v}^k - \alpha_{AM} a_x \varphi_x \varphi_t + \left(\alpha_{AM}\varphi_x^2 + 1\right)a_t}{1 + \alpha_{AM}\left(\varphi_y^2 + \varphi_x^2\right) + \alpha_{AM}\beta_{FM}\left(\varphi_x a_y - \varphi_y a_x\right)^2 + \beta_{FM}\left(a_x^2 + a_y^2\right)} a_y.
\end{aligned}
\tag{2.33}
$$

2.5 Local solutions via least squares

Local solutions assume a constant velocity model and use least-squares (LS) fits over pixel neighborhoods. Local solutions have the advantage that are robust in the present of noise [2, 27] but suffer from low density field estimates wherever the system of equations to solve is ill-conditioned.

2.5.1   Lucas and Kanade least squares solution

Instead of adding further constraints to the CBM, Lucas and Kanade [6] chose to minimize a weighted version of the OFC assuming that nearby pixels share the same 2D velocity:

$$
E_{W_{OFC}} = \iint_{\Gamma} W(x,y)\left(I_x u + I_y v + I_t\right)^2 d\Gamma
\tag{2.34}
$$

29

where the weighted functional (2.34) is minimized over a local spatial neighborhood Γ. In discrete form, this becomes:

$$\sum_{x \in \Gamma} \left[W(x, y)(I_x u + I_y v + I_t)^2 \right]. \tag{2.35}$$

The associated Euler-Lagrange equations of (2.34) are:

$$W(x)(I_x^2 u + I_x I_y v + I_x I_t) = 0, \text{ and}$$
$$W(x)(I_x I_y u + I_y^2 v + I_y I_t) = 0. \tag{2.36}$$

The weight function $W(x, y)$ is assumed to be separable $W(x, y) = W_1(x) W_2(y)$ and a function of distance from the center. The idea is to give more emphasis to measurements at the centre of the window than at the periphery. The constraint errors are added over a 5×5 region, leading to

$$\sum_{x \in \Gamma} W(x, y)(I_x^2 u + I_x I_y v + I_x I_t) = 0,$$
$$\sum_{x \in \Gamma} W(x, y)(I_x I_y u + I_y^2 v + I_y I_t) = 0 \tag{2.37}$$

The least square estimator that minimizes the sum of the squared errors in (2.35) gives the solution to (2.37) as:

$$\vec{v} = \left[A^T W A \right]^{-1} A^T W b \tag{2.38}$$

where

$$A = \left[\nabla I(x_1), \ldots, \nabla I(x_n) \right]^T,$$
$$W = \bar{w} \times \bar{w}^T, \tag{2.39}$$
$$b = -\left[I_t(x_1), \ldots, I_t(x_n) \right]^T,$$

and $\vec{v}$ denotes the velocity estimates for n points $(x_i, y_i) \in \Gamma$ at a single time t.

There is a closed form solution to $\vec{v}$ when $A^T W A$ is non-singular given by:

30

$$A^T W A = \begin{bmatrix} \sum W(x,y) I_x^2 & \sum W(x,y) I_x I_y \\ \sum W(x,y) I_x I_y & \sum W(x,y) I_y^2 \end{bmatrix} \qquad (2.40)$$

with summations taken over the entire window. Here, matrix $A^T W A$ has to be full rank and depends on accurate image gradient calculations [33]. Pre-smoothing is applied to the image sequences to attenuate the effect of quantization noise and the amplitude of high order terms in (2.3). For high levels of noise and significant values of the higher order terms, sufficient smoothing will be needed to ensure the accuracy of the estimation. On the other hand, too much smoothing will also lead to inaccurate estimates since it destroys local image structure. The smoothing is controlled by adjusting the spread, σ, of a spatial Gaussian filter.

2.5.2   Uras's second order differential technique

Uras et al. [10] used the Hessian of the image brightness intensity to locally constraint the image velocity assuming the conservation of the image intensity gradient. Locally, they solved an over-determined system of equations over regions where the Hessian is non-singular. The conservation of image gradient energy functional is given by:

$$E_G = \iint_\Gamma \left\| \nabla \left(I_x u + I_y v + I_t \right) \right\|^2 d\Gamma. \qquad (2.41)$$

The associated Euler-Lagrange equations are:

$$\begin{aligned} I_{xx} u + I_{xy} v + I_{xt} &= 0, \text{ and} \\ I_{xy} u + I_{yy} v + I_{yt} &= 0, \end{aligned} \qquad (2.42)$$

and can be solved wherever the Hessian of the image brightness intensities is non singular. In practice, for robustness [27], the authors divide the image into 8×8 pixel regions. For each region only the eight estimates that best satisfies:

$$\left\| M^T \nabla I \right\| \ll \left\| \nabla I_t \right\|, \tag{2.43}$$

where M^T is the transpose of 2×2 matrix:

$$M = \begin{bmatrix} \dfrac{\partial u}{\partial x} & \dfrac{\partial u}{\partial y} \\[2mm] \dfrac{\partial v}{\partial x} & \dfrac{\partial v}{\partial y} \end{bmatrix}, \tag{2.44}$$

are considered to calculate their Hessian condition number, and the estimate with the smallest condition number is selected as the velocity field of the 8×8 region.

The main problem with the conservation of ∇I, $d\nabla I / dt = 0$, is that it is a far more restrictive assumption than the conservation of image intensities because first order deformations of image intensities, e.g rotations or dilations, should not be present in the permissible motion field. In addition, second order derivatives can not be approximated accurately.

2.5.3   Phase-based technique for motion estimation

The use of phase information for the computation of optical flow was first introduced in [3]. The input image is decomposed into band-pass channels using complex Gabor filters to extract amplitude and phase information. Each filter output can be written as:

$$R(\vec{x},t) = \rho(\vec{x},t) \exp\left[i\phi(\vec{x},t) \right] \tag{2.45}$$

and space-time surfaces of constant phase, $\phi(\vec{x},t) = c$, are used to estimate the velocity component normal to level phase contours. Differentiating these surfaces leads to the phase energy functional which is given by:

$$E_{Phase} = \iint_{\Gamma} \left(\phi_x u + \phi_y v + \phi_t \right) d\Gamma. \tag{2.46}$$

32

The velocity that minimizes (2.46) is called component velocity and is the velocity normal to level phase contours, $\vec{v}_n = s\vec{n}$ given by:

$$s = \frac{-\phi_t(\vec{x},t)}{\|\nabla \phi(\vec{x},t)\|}, \text{ and}$$
$$\vec{n} = \frac{\nabla \phi(\vec{x},t)}{\|\nabla \phi(\vec{x},t)\|}. \tag{2.47}$$

The computation of optical flow from the conservation of phase information is a differential technique applied to phase rather than image intensity, and it is motivated from the claim that phase information is more stable under changes in mean intensity and contrast than amplitude information [34]. However, phase instabilities do occur and the authors in [3] detect them using:

$$\|(k_0, w_0) - (k, w)\| \le \sigma_k \tau, \tag{2.48}$$

where (k_0, w_0) denotes the spatiotemporal frequency to which each filter is tuned, σ_k denotes the standard deviation of the filter's amplitude spectrum, and τ is a threshold that can be used to reject local frequencies that are far from the nominal tuning range of the filter [3]. Finally, using the measurements of component velocities, a linear velocity model:

$$\tilde{v}(\vec{x},t) = (\alpha_0 + \alpha_1 x + \alpha_2 y, \beta_0 + \beta_1 x + \beta_2 y)^T \tag{2.49}$$

is fitted to 5×5 regions yielding a linear system $Ra = \vec{v}_n$ that is solved using least squares. The estimated full 2d velocity was taken to be the constant parameters $\vec{v} = (\alpha_0, \beta_0)^T$, in (2.49).

33

2.6 Combining global and local optical flow solutions

The combination of global and local methods, known as the combined local-global method, has been studied in [35]. Here, the methods by Horn and Schunk [1] and Lucas and Kanade [6] are merged in a hybrid technique using a structural tensor at different integration scales creating the combined local-global functional which gives a robust solution on local neighbourhoods. The approach also yields high density fields.

Considerer the image sequence $I(x, y, t)$, and also the fact that it is common to smooth with a Gaussian $K_\sigma(x, y)$ of standard deviation σ:

$$f(x, y, t) = (K_\sigma * I)(x, y, t) \tag{2.50}$$

where σ can be assumed to correspond to the noise scale. Then, assuming the conservation of image brightness intensities for small displacements gives:

$$f_x u + f_y v + f_t = 0. \tag{2.51}$$

Now, recall that Lucas and Kanade assumed constant velocities within some neighborhood of size ρ. The Lucas and Kanade method is reformulated as one of using weighted least squares to solve:

$$E_{LK} = K_\rho * \left(\left(f_x u + f_y v + f_t \right)^2 \right). \tag{2.52}$$

In (2.52), ρ serves as an integration scale over which the influence of the weighted least squares is controlled [2]. In order to develop the combined local-global method, the following symbols are introduced:

$$w = (u, v, 1)^T,$$

$$|\nabla w|^2 = |\nabla u|^2 + |\nabla v|^2,$$

$$\nabla_3 f = (f_x, f_y, f_t)^T,$$

$$J_\rho(\nabla_3 f) = K_\rho * (\nabla_3 f \nabla_3 f^T).$$

(2.53)

Now, equation (2.52) can be expressed as:

$$E_{LK}(w) = w^T J_\rho(\nabla_3 f) w.$$

(2.54)

The constant brightness model combined energy functional (2.13) E_{CBM} becomes:

$$E_{CBM} = \iint_\Omega \left(w^T J_0(\nabla_3 f) w + \alpha^2 |\nabla w|^2 \right) dxdy.$$

(2.55)

By replacing the matrix $J_0(\nabla_3 f)$ in (2.55) with the structure tensor $J_\rho(\nabla_3 f)$ with some

integration scale $\rho > 0$. The combined local-global energy function can be written as:

$$E_{CLG} = \iint_\Omega \left(\vec{w}^T J_\rho(\nabla_3 f) \vec{w} + \alpha^2 |\nabla w|^2 \right) d\Omega,$$

(2.56)

and the associated Euler-Lagrange equations are:

$$\nabla^2 u - \frac{1}{\alpha^2} \left[K_\rho * (f_x^2) u + K_\rho * (f_x f_y) v + K_\rho * (f_x f_t) \right] = 0, \text{ and}$$

$$\nabla^2 v - \frac{1}{\alpha^2} \left[K_\rho * (f_x f_y) u + K_\rho * (f_y^2) v + K_\rho * (f_y f_t) \right] = 0.$$

(2.57)

The system (2.57) is solved iteratively using the Successive Over Relaxation method

(SOR) method with finite differences.

2.7 Summary of other Emerging Techniques

Motion estimation for video tracking under variation of image illumination was studied

by Hager and Belhumeur [36] using a parametric model. A target dependent basis for

illumination changes is computed *a priori* and used to project the image to a low

35

dimensional linear sub-space. The models are later incorporated into the tracking algorithm which can handle object deformations, changes in geometry, occlusion and illumination variations.

Following Nagahdaripour's work [9] and modeling brightness transformations with physical models, Haussecker and Fleet in [37] parameterized brightness changes in an attempt to model high-order brightness variations. They created models for changes due to light diffusion, exponential brightness decay, non-uniform moving illumination envelopes, and surface rotation under directional illumination. Using an anisotropic diffusion approach for modeling illumination changes, Alvarez et al. [38] created an energy functional that is invariant to linear brightness changes with the additional claim that their method can handle the estimation of large motion fields up to ten pixels per frame.

In order to improve numerical stability and robustness, many researches have focused on using different operators to improve the robustness of the solution of optical flow. Robust statistical estimators were introduced in the computation of optical flow [32, 39, 40] in order to better handle outliers, pixels that violate the assumptions, and reduce the estimation error due to discontinuity of the flow at object boundaries. Structural tensor based methods combined with diffusion approaches [41-45] proved to be a reliable tool for noise reduction, and the use of total variation TV-L[1] [46, 47] has been used to minimize the energy functionals by replacing quadratic norm penalties with a L[1] penalty norm that allows for discontinuities in the motion field.

Chapter 3

Full-reference Global Optimization for Motion Estimation

Pixel based motion estimation is a very active research area. Most of the innovations have come either as changes to the theoretical framework as new constraints and assumptions, or as improvements to the numerical solution methods. Much less attention has been focused on the selection of the model parameters. This chapter will demonstrate the importance of selecting the correct regularization parameter and any other motion related parameter so that accurate velocity vectors can be estimated. Optimization methods will be applied to minimize the error of the estimated motion field.

The term "full-reference" is used to refer to the fact that the ground truth is known. Therefore, the optimization approach described here is only applicable to simulated motions. A no-reference approach that can be applied to any given video with unknown motion field will be described in Chapter 4.

3.1 Regularization parameter selection based on inverse problems methods

The estimation of optical flow is an inverse problem since the models try to calculate the velocity vectors from measurements of image intensity values that are related to the unknown motion field. The following methods [21, 22, 24, 25, 48] have been proposed

for calculating the associated regularization parameter in general inverse problem applications. Some of the most important methods will be reviewed next.

3.1.1 The Discrepancy Principle

In the case of the Horn and Schunk technique, the optical flow variational formulation (2.13) leads to the solution of an associated system of Euler-Lagrange equations that depend on a regularization parameter given by:

$$\begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} -I_x I_t \\ -I_y I_t \end{bmatrix} + \alpha^2 \begin{bmatrix} \nabla^2 u \\ \nabla^2 v \end{bmatrix}. \tag{3.1}$$

System (3.1) needs to be solved for every pixel in the image. In general form, it can be re-arranged using a lexicographical ordering that leads to:

$$\mathbf{f}_\alpha = \arg\min \left\| K\mathbf{f} - \mathbf{b} \right\|_2^2 + \alpha^2 \left\| L\mathbf{f} \right\|_2^2, \tag{3.2}$$

where K is the forward operator composed of image intensity derivatives, the measurement vector is $\mathbf{b} = \begin{bmatrix} -I_x I_t & -I_y I_t \end{bmatrix}^T$, and L is the regularization operator that acts on the unknown velocity vector $\mathbf{f} = \begin{bmatrix} u & v \end{bmatrix}^T$. In system (3.1), the Laplacian of the velocities is the operator L, and can be approximated using finite differencing.

The discrepancy principle [23] makes use of available statistical information regarding the error or noise levels present in the data fidelity using the model:

$$\mathbf{b} = K\mathbf{f}_{true} + \eta, \tag{3.3}$$

where η is assumed to be an n-vector realization of white noise with variance λ^2, and $\mathbf{f}_{true}$ is the true solution. Thus, one seeks to find the largest regularization parameter α that satisfies:

$$\frac{1}{n}\left\|K\mathbf{f}_\alpha - \mathbf{b}\right\|^2 \le \lambda^2. \tag{3.4}$$

The dependency of the optical flow model on α is explicit from (3.1), but this is not the only parameter that needs to be calculated. Image intensity derivatives depend on the spread σ of a Gaussian pre-smoothing filter that controls the amount of averaging necessary to calculate intensity derivatives. The discrepancy principle only allows selecting an approximate value of the regularization parameter while the spread of the pre-smoothing filter has to be considered fixed. Thus for a given value of σ, the largest α that satisfies (3.4) will be calculated.

3.1.2 The Generalized Cross Validation

The generalized cross validation (GCV) method [24] does not require prior knowledge of the error in (3.3). The GCV estimate of the regularization parameter is the one that minimizes:

$$V(\alpha) = \frac{\dfrac{1}{n}\left\|K\mathbf{f}_\alpha - \mathbf{b}\right\|^2}{\left[\dfrac{1}{n}trace\left(I - A_\alpha\right)\right]^2} \tag{3.5}$$

where $A_\alpha = K^T\left(K^T K + \alpha L\right)^{-1} K$. The numerical implementation of the $trace\left(I - A_\alpha\right)$ is the main disadvantage of the GCV even for small systems. Practical applications of this operation rely on the singular value decomposition (SVD) of A_α [21]. In our case this is not an option since the computational complexity of this algorithm for video frames is prohibitive. Moreover, this method like the others presented so far, cannot estimate the spread of the Gaussian pre-smoothing filter.

3.2 Multi-Objective optimization

A multi-objective optimization problem has a number of conflicting objective functions that are to be minimized, and usually there is not a single solution that simultaneously minimizes all the objectives. Instead, there is a feasible set of solutions with conflicting values for each objective [49]. Along with the objective functions, there are also a number of constraints that any feasible set of solutions must satisfy.

Denote the general multi-objective optimization problem as:

$$
\begin{aligned}
\min \quad & E_m(\vec{x}), m = 1, 2, \ldots, M, \\
\text{subject to } & g_i(\vec{x}) \geq 0, j = 1, 2, \ldots, J, \\
& h_k(\vec{x}) = 0, k = 1, 2, \ldots, K, \\
& x_i^{(L)} \leq x_i \leq x_i^{(U)}, i = 1, 2, \ldots, n.
\end{aligned}
\tag{3.6}
$$

A solution $\vec{x} \in \mathbb{R}^n$ is the optimization vector variable, and associated with the problem there are constraints g_i, h_k, and bounds $x_i^{(L)}$, and $x_i^{(U)}$. If a solution $\vec{x}$ satisfies all the constraints and bounds, it is called a feasible solution, and the set of all feasible solutions is called the feasible region.

3.2.1   Optimal Point

In a multi-objective optimization problem, an optimal point x^* satisfies:

$$
E_m(x^*) \leq E_m(y), \ m = 1, 2, \ldots, M
\tag{3.7}
$$

for every feasible y. Therefore, x^* is simultaneously optimal for each of the objective functions. Most multi-objective optimization problems do not have an optimal point and optimal value, but it does occur in especial cases when the objectives are not competing against each other.

3.2.2 Pareto optimal points

Pareto points are feasible points for which improvement in one objective can only occur with the worsening of at least one other objective [50]. Instead of a unique optimal solution to (3.7), there is a set of Pareto optimal solutions that lies in the boundary of achievable objective values whose shape indicates the nature of the trade-off between the objectives [51] (figure 3.1).

Formally, a feasible point is a Pareto optimal point $\vec{x}^{po}$ if and only if there is no feasible $\vec{x}$ such that $E_m(\vec{x}) \leq E_m(\vec{x}^{po})$ with at least one strict inequality. In particular, if a point is not Pareto optimal, then there is at least one other point that is better [51].
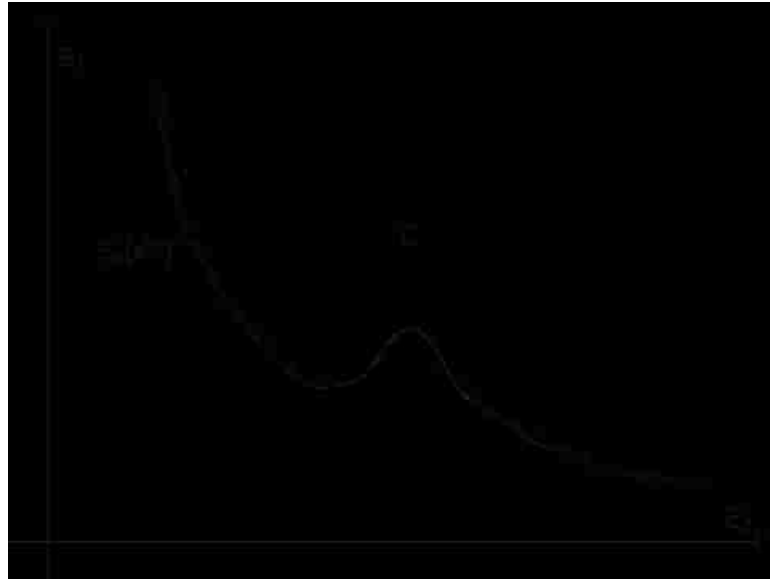
Figure 3.1. Set of achievable values for a multi-opjective optimization problem in R^2, see [51]. The depicted problem does not have a minimum value. Pareto optimal points are presented by the thicker line over the lower left boundary of the feasible solution region O. The point labeled $E_0(x^{po})$ is a Pareto optimal value, and $\vec{x}^{po}$ is a Pareto optimal point.

3.2.3 Trade off analysis

Suppose $\vec{x}$ and $\vec{y}$ are Pareto optimal points. Then, we have:

41

$$E_m(\vec{x}) < E_m(\vec{y}), m \in A$$
$$E_m(\vec{x}) = E_m(\vec{y}), m \in B \tag{3.8}$$
$$E_m(\vec{x}) > E_m(\vec{y}), m \in C$$

where A, B, C represents a partition of all the constraints with $A \cup B \cup C = \{1, 2, ..., M\}$.

Note here that $\vec{x}$ has achieved better objective values for $m \in A$ after trading off worse objective values for $m \in C$. Trade off analysis is the study of what set of objective values are achievable. The optimal trade-off surface (when $m > 2$) or curve ($m = 2$) is the set of Pareto optimal values for a given multi-objective optimization problem [51]

3.2.4 Weight sum method

The weight sum method is a technique to find Pareto optimal points that minimizes the following weighted sum objective:

$$E_0(\vec{x}) = \sum_{m=1}^{m=M} w_m E_m(\vec{x}) \tag{3.9}$$

when $\vec{x}$ is optimal, the solution to (3.9) is Pareto optimal if the weights are positive for all the objectives [50]. The weight of an objective is chosen in proportion to the objective's relative importance in the problem, and it can be thought of as quantifying the desire of making E_m small or large. This ad-hoc adjustment of the weights allows the exploration of optimal trade-off surfaces in order to find Pareto optimal points.

For motion estimation, the objectives E_m will come from discretized versions of the energy functional models and the weights will come from the values of the regularization parameters. Thus, optical flow models for motion estimation can be expressed in terms of (3.9).

3.2.5 The L-curve method

The L-curve is a logarithmic plot of the norm of the regularization term against the norm of the data fidelity term in (3.2) for different values of α. The method is called "L-curve" because the plotted relationship typically exhibits an "L" shape, and the optimal regularization parameter, α^*, is the one that corresponds to the "corner" of this curve which represents the transition between over and under- regularized solutions, see .

Possible definitions for the corner of the curve include the point of maximum curvature and a point of tangency with a slope of -1. This last definition leads to the α value that satisfies:

$$\alpha^2 = \frac{\left\| g - H\hat{f}(\alpha) \right\|_2^2}{\left\| L\hat{f}(\alpha) \right\|_2^2}. \tag{3.10}$$

In [22] it is shown that under certain assumptions the "L-curve" has the characteristic vertical and horizontal parts of an "L". However, note that problem (3.1) might not satisfy those conditions, and the "L" shape of the curve is not guaranteed unlike the case of standard Tikhonov regularization problems [48].

Figure 3.2. Shape of the L-curve. The optimal regularization parameter α^* is the one that corresponds to the corner of the curve.

3.3 Lipschitz optimization

Consider to following minimization problem:

$$
\begin{aligned}
\min \quad & f(x) \\
\text{s.t.} \quad & x \in D
\end{aligned}
\tag{3.11}
$$

When $D \in \mathbb{R}^n$ represents a compact set $D := \{x \in \mathbb{R}^n : a \le x \le b\}, a < b \in \mathbb{R}^n$, and $f(x)$ is a real valued Lipschitz function on a compact set $P \supset D$. The problem (3.11) is called a Lipschitz optimization problem [52].

3.3.1　Lipschitz functions

A real valued function f is called Lipschitzian or a Lipschitz function on a set $P \subset \mathbb{R}^n$ if there is a (Lipschitz) constant $L = L(f, P) > 0$ such that

$$
\left| f(x_2) - f(x_1) \right| \le L \|x_2 - x_1\| \text{ for all } x_1, x_2 \in P.
\tag{3.12}
$$

44

If f is Lipschitzian on P with constant L, then f is also Lipschitzian on P with all

constants $L^{'} > L$. When P is a real interval, L provides an upper bound for the absolute

value of the slope of any line joining two points on the graph of f [52].

Let $P \subset \mathbb{R}^n$ be convex, and let f be continuously differentiable on an open set containing

P with bounded gradient on P. Then f is Lipschitzian on P with constant

$$L = \sup\{\|\nabla f(x)\| : x \in P\}. \tag{3.13}$$

The supremum can be replaced by the maximum of $\|\nabla f(x)\|$ over P, when P is a

bounded closed set and $\nabla f(x)$ is continuous and bounded [52].

3.3.2   Branch and Bound

Branch and bound is a method in which the feasible set is partitioned into more and more

refined parts (branching), over which lower and upper bounds of the minimum objective

function value are determined (bounding). Parts of the feasible set with lower bounds

larger than the best upper bound found at a certain stage are deleted from further

consideration (pruning) because these parts of the domain cannot contain the optimum.

3.3.3   Lower bounds

Let P be an n-rectangle and let L be a Lipschitz constant of the function f on P. Then,

for $x, y \in P$ the following holds:

$$|f(x) - f(y)| \le L\|x - y\|. \tag{3.14}$$

It follows from (3.14) that, for all $x, y \in P$

$$f(x) \ge f(y) - L\|x - y\| \tag{3.15}$$

45

If $y \in P$ is fixed, then the concave function

$$F(x) = f(y) - L\|x - y\| \tag{3.16}$$

underestimates $f(x)$ on P.

Suppose that the diameter

$$\delta(P) := \sup\{\|x - y\| : x, y \in P\} \tag{3.17}$$

is known. Let $T \subset P$ denote a finite sample of points in P where the function values of f have been evaluated. Then

$$\mu(P) = \max_{y \in T} f(y) - L\delta(P) \tag{3.18}$$

is a lower bound for $\inf\{f(x) : x \in P\}$. When P is a rectangle or a simplex the set T often coincides with the vertex set $V(P)$ [52].

A worthwhile interior point to take into account is the midpoint $m = (a + b)/2$ of an n-rectangle $a \le x \le b$. This gives:

$$\mu(P) = f(m) - L\delta(P)/2. \tag{3.19}$$

3.3.4 Lipschitz optimization over rectangles

Consider the following realization of the branch and bound method for solving

$$\min \ f(\vec{x})$$
$$\text{s.t } \mathbf{a} \le \vec{x} \le \mathbf{b} \tag{3.20}$$

where $\mathbf{a} = (a_1, \ldots, a_n)^T$, $\mathbf{b} = (b_1, \ldots, b_n)^T \in \mathbb{R}^n, \mathbf{a} < \mathbf{b}$, and $f : R = \{x : \mathbf{a} \le \vec{x} \le \mathbf{b}\} \to \mathbb{R}$ is Lipschitzian on R with Lipschitz constant L. shows the algorithm [52] that performs branch and bounding with global Lipschitz optimization.

46

Initialization:

Set $x_R \leftarrow (a+b)/2, Q \leftarrow \{a,b,x_R\}, \gamma \leftarrow \min\{f(x): x \in Q\}$;

Choose $v \in Q$ satisfying $f(v) = \gamma$;

Set $\mu(R) \leftarrow \max\{\max\{f(a), f(b)\} - L\|b-a\|, f(x_R) - L\|b-a\|/2\}$;

Set $\mu \leftarrow \mu(R); M \leftarrow \{R\}; stop \leftarrow false, k \leftarrow 1$;

while stop = false **do**

    if $\gamma = \mu$ **then**

        $stop \leftarrow true$ (v is an optimal solution, and γ the optimal objective function value)

    else

        Compute $b_j - a_j = \max\{b_i - a_i : i = 1, \ldots, n\}$, and set

$$a^1 \leftarrow a, b^1 \leftarrow \left(b_1, \ldots, b_{j-1}, (b_j + a_j)/2, a_{j+1}, \ldots a_n\right)^T,$$

$$R_1 \leftarrow \{x : a^1 \le x \le b^1\}, R_2 \leftarrow \{x : a^2 \le x \le b^2\},$$

$$x_{R_1} \leftarrow (a^1 + b^1)/2, x_{R_2} \leftarrow (a^1 + b^1)/2;$$

        Set

$$\mu(R_1) \leftarrow \max\{\mu(R), \max\{f(a^1), f(b^1)\} - L\|b^1 - a^1\|, f(x_{R_1}) - L\|b^1 - a^1\|\},$$

$$\mu(R_2) \leftarrow \max\{\mu(R), \max\{f(a^2), f(b^2)\} - L\|b^2 - a^2\|, f(x_{R_2}) - L\|b^2 - a^2\|\};$$

        Set $Q \leftarrow \{v, b^1, x_{R_1}, a^2, x_{R_2}\}, \gamma \leftarrow \min\{f(x): x \in Q\}$;

        Update v satisfying $f(v) = \gamma$;

        Set

$$M \leftarrow (M \setminus \{R\}) \cup \{R_1, R_2\},$$

$$\Re \leftarrow \{R \in M : \mu(R) < \gamma\}.$$

$$\mu \leftarrow \begin{cases} \min\{\mu(R) : R \in \Re\}, & \text{if } \Re \ne \varnothing \\ \gamma, & \text{if } \Re = \varnothing \end{cases}$$

        Choose

$$R \in \Re \text{ satisfying } \mu(R) = \mu;$$

        Update a, b such that $R = \{x : a \le x \le b\}$.

    end if

    Set $M \leftarrow \Re, k \leftarrow k + 1$.

end while

Figure 3.3. Branch and bounding global Lipschitz optimization algorithm.

This algorithm uses combinations of the bounds (3.18) and (3.19) and bisection of rectangles at the midpoint of one of the longest edges. The proof of algorithm convergence is included in the appendix section for completeness and was originally published in [52].

3.3.5 Full-reference global optimization

Figure 3.4 shows the block diagram of the full-reference approach. The techniques of Horn [1] and Lucas [6] are trained on a set of benchmark video sequences in order to find the motion parameters that minimize the mean squared error (MSE) of the estimation. Once the optimal parameters are calculated, they are tested on natural video sequences (videos without ground truth).

Lipchitz optimization requires a real valued objective function to be Lipchitz continuous over a set, i.e. the slope of the objective functions needs to be bounded. For motion estimation, the objective function is set to be the MSE of the velocity magnitudes. The goal is to find the vector, $\vec{p}$, of motion estimation parameters that minimizes the MSE between the magnitudes of the ground truth velocities, *GTMag*, and estimated velocities, $ESTMag(\vec{p})$. The MSE is calculated for all pixels (i, j) in the image with $i = 1, \ldots, M$ and $j = 1, \ldots, N$.

$$GTMag = \left\| \vec{v}_{gt} \right\| = \sqrt{u_{gt}^2 + v_{gt}^2}$$

$$ESTMag(\vec{p}) = \left\| \vec{v}_{est}(\vec{p}) \right\| = \sqrt{u_{est}^2(\vec{p}) + v_{est}^2(\vec{p})} \qquad (3.21)$$

$$f(\vec{p}) = \frac{1}{MN} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} \left(ESTMag_{ij}(\vec{p}) - GTMag_{ij} \right)^2.$$
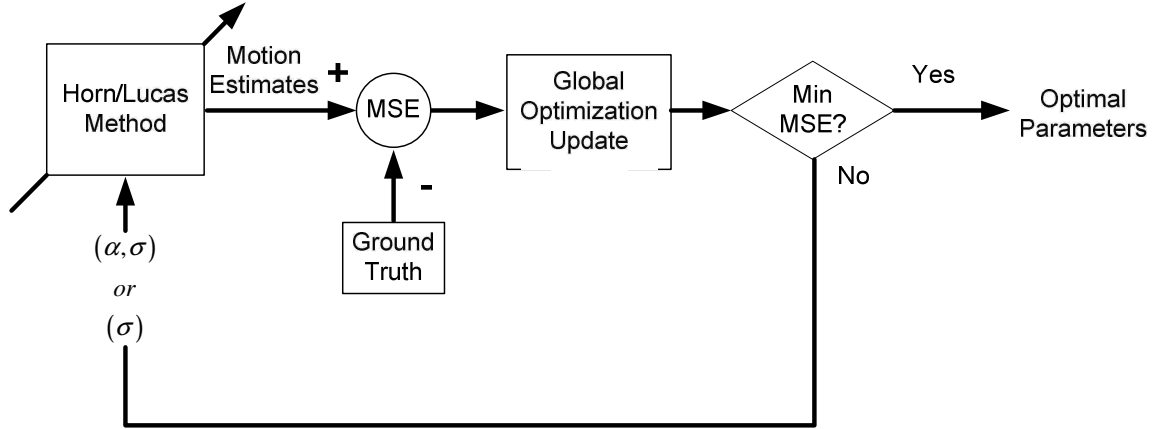
Figure 3.4. Block diagram of the full-reference global optimization approach.

The vector $\vec{p}$ is composed of the motion estimation parameters that govern the optical flow models and it varies for each method. In the case of the Horn and Schunk method [1], the parameter vector is $\vec{p} = (\alpha, \sigma)$, and optimization was carried out for the spread of the spatiotemporal Gaussian pre-smoothing filter, σ, and the value of the regularization parameter α that controls the influence of the smoothness constraint E_S in the optical flow model (2.13).

The parameter space P is defined to be large enough so that regularization parameter values reported in the literature [1, 17, 27] are included in the optimization process. The range of values of the Gaussian pre-smoothing filter, σ, is limited by the number of frames in the video sequence. For each of the ground truth sequences, the biggest σ is set to be the maximum possible value based on the number of video frames. A closed set $P = \left\{ \vec{p} \in \Re^2 : \vec{a} \le \vec{p} \le \vec{b} \right\}$ for $\vec{p} = (\alpha, \sigma)^T$ is defined. Table 3.1 shows the parameter space values for each video sequence.

Table 3.1. Motion estimation parameter space for Horn's method.

	Translating Tree		Divergence Tree		Yosemite	
	a	b	a	b	a	b
α	0.1	50	0.1	50	0.1	50
σ	0.5	4.0	0.5	1.75	0.25	1.25

The Lipschitz constant L could be calculated using equation (3.13), but the dependency of the MSE on the motion parameters is implicit only with respect to α in equation (3.22). Here, recall that the velocity estimates are given by:

$$u^{k+1} = \bar{u}^k - \frac{I_x \left[I_x \bar{u}^k + I_y \bar{v}^k + I_t \right]}{\alpha^2 + I_x^2 + I_y^2},$$

$$v^{k+1} = \bar{v}^k - \frac{I_y \left[I_x \bar{u}^k + I_y \bar{v}^k + I_t \right]}{\alpha^2 + I_x^2 + I_y^2}.$$

(3.22)

The dependency of the velocity estimates with respect to σ arises from the convolution operation of the image intensity values $I(x,y,t)$ with a separable Gaussian filter K_σ prior image differentiation. Here:

$$I_x = \frac{\partial}{\partial x} \left(K_\sigma(x) * I(x,y,t) \right)$$

(3.23)

$$I_y = \frac{\partial}{\partial y} \left(K_\sigma(y) * I(x,y,t) \right)$$

(3.24)

$$I_t = \frac{\partial}{\partial t} \left(K_\sigma(t) * I(x,y,t) \right).$$

(3.25)

Substituting equations (3.23)-(3.25) into (3.22) leads to an implicit expression that relates the velocities estimates to the motion parameters. Unfortunately, the gradient of the MSE with respect to (α, σ) is a very complicated expression for which a closed form solution

even though calculated, could not be used to set a value of L. Instead, the fact that the objective function, f, is Lipschitzian on P with constant L will be used to find another constant L' such that $L' > L$ also satisfies (3.12) [52].

Figure 3.5 shows the flowchart of the algorithm used to heuristically verify that an appropiate L' has been used. The algorithm starts with a large value of L' and a large number of iterations for which a minimum of the objective value is found. Alternatively, the algorithm always stops if the difference between the current minimum value of the objective function and the best lower bound is smaller than a given tolerance, $\varepsilon = 0.01$. This minimum is going to be used as a reference for successive runs where L' is decreased. When the algorithm stops after a small number of iterations returning a worse minimum of the objective function, then at this point L' cannot be decreased anymore because relationship (3.12) is no longer satisfied. In theory an infinite number of iterations are necessary to find the optimal value, but in practice starting with a large L' along with enough number of iterations works. This process, even though heuristic, appears to work very well.
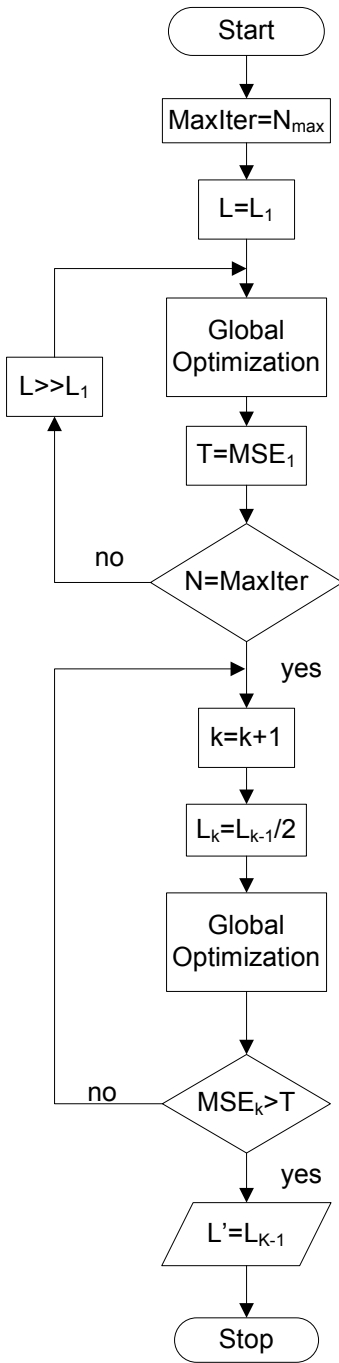
Figure 3.5. Flowchart of the algorithm used to verify that the correct Lipschitz constant, L, has been used to minimize the MSE.

3.4 Results

For comparison and evaluation of the optimization methods, the mean squared error (MSE) is the objective function. On the other hand, note that the average angular error (AAE), μ_{AE}, and the standard deviation angular error (STD), δ_{AE}, between the estimated field and the ground truth are the most used benchmarks [27]. Here, it is important to note that when the MSE equals zero, then there is convergence to the ground truth. The same cannot be said for the AAE and STD because they only measure the direction of the flow and not the magnitude of the velocity. Nevertheless, but they will also be used to evaluate the performance of optimization methods as a reference.

3.4.1   Synthetic Image sequences

Three standard test video sequences: the Divergence Tree, the Translating Tree, and the Yosemite videos are used to evaluate the performance of the optimization methods. The advantage of using synthetic video sequences is that the true displacement field or ground truth is available to quantify performance. These three sequences are widely used in the literature [2, 15, 27, 38, 53, 54] because they allow the estimation of translational motion, divergent motion, expanding motion fields, and a wide range of velocity values. The main characteristics of each one are:

- **Diverging Tree Sequence**: The camera moves closer to the textured surface along the camera's line of sight with the velocity field expanding about the center of the image. The speeds are 1.29 pixels/frame on the left side, and 1.86 pixels/frame on the right.

- **Translating Tree Sequence**: It simulates pure translational camera motion. All velocities are parallel to the image x-axis. The speeds are between 1.773 and 2.26 pixels/frame.

- **Yosemite Video Sequence**: By far the most difficult test case. The motion on the top part of the image, the cloud region, is supposed to be translational but violations to the brightness intensity conservation assumption occur. This is the reason why errors for this sequence are reported with the clouds and without them. The velocities of the lower left part of the image are around 4 pixel/frame. These values violate the small displacement assumption and are difficult to estimate.

3.4.2   Discrepancy principle results

The velocity MSE plots for the motion estimation parameters that satisfy the discrepancy principle are shown from figure 3.6 to figure 3.8. Values of the error variance for the Divergence Tree video sequence were $\lambda_{DT}^2 = 0.1,\ 0.25,$ and 0.5, noise variances for the Yosemite video sequence were $\lambda_Y^2 = 1,\ 10,\ 50,$ and 150, and for the Translating Tree sequence were $\lambda_{DT}^2 = 0.01,\ 0.25,$ and 0.5. The choice of the error variance value is done heuristically since it is not possible to a priori know how large this value should be. Results for the three sequences are summarized in table 3.2 to table 3.4 The density of the velocity estimation for all three sequences was 99%.

3.4.3   L-curve results

The L-curve was calculated for values of the regularization parameter, α, ranging from
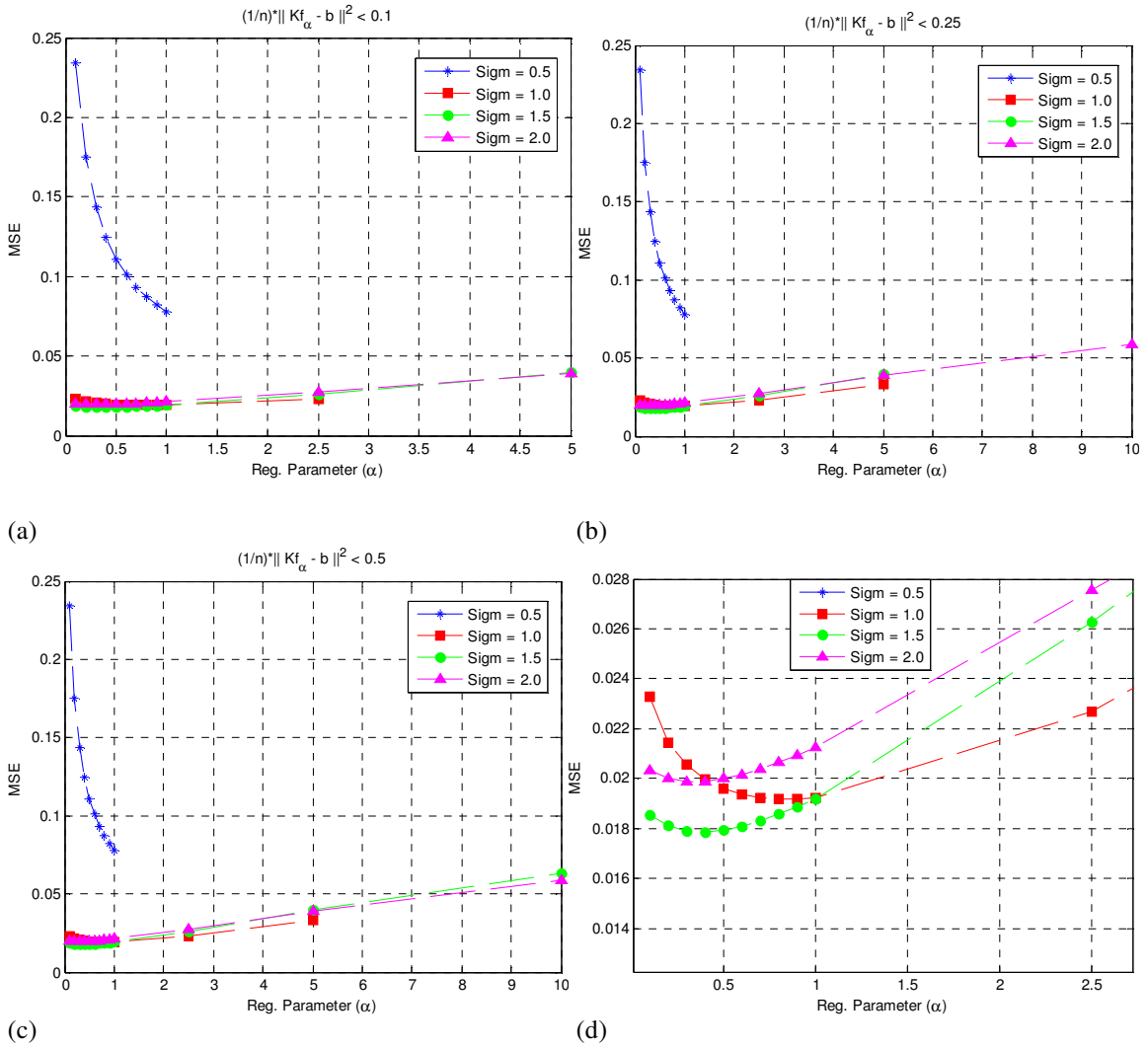
Figure 3.6. Velocity MSE values obtained by applying the discrepancy principle to the Divergence Tree video sequence for different error noise variances, λ_T. (a) Velocity MSE $\lambda_T = 0.1$. (b) Velocity MSE $\lambda_T = 0.25$. (c) Velocity MSE $\lambda_T = 0.5$. (d) The minimum MSE is obtained for $\sigma = 1.5$ and $\alpha = 0.4$.

$\alpha = 0.1$ to $\alpha = 150$. Figure 3.9 shows the L-curve results. Plots of MSE vs. the norm in logarithmic scale of the data fidelity and regularization terms are also provided in figure 3.10 for the Divergence Tree and Yosemite sequences. Translating Tree sequence results are shown in figure 3.11, and summarized in table 3.5 to table 3.7.

Table 3.2. Estimated motion parameters, MSE, and Av. Ang. error using the discrepancy principle on the Divergence Tree video sequence.

$\lambda_T^2 = 0.1$					$\lambda_T^2 = 0.25$					$\lambda_T^2 = 0.50$				
σ	α	MSE	μ_{AE}	δ_{AE}	σ	α	MSE	μ_{AE}	δ_{AE}	σ	α	MSE	μ_{AE}	δ_{AE}
0.5	1	0.078	11.3°	10.3°	0.5	1	0.078	11.3°	10.3°	0.5	1	0.078	11.3°	10.3°
1.0	**2.5**	**0.023**	**4.9°**	**2.5°**	1.0	5	0.033	5.6°	3.5°	1.0	5	0.033	5.6°	3.47°
1.5	5	0.040	6.5°	4.6°	1.5	5	0.040	6.5°	4.6°	1.5	10	0.063	8.1°	6.35°
2.0	5	0.039	6.7°	5.0°	2.0	10	0.059	8.2°	6.4°	2.0	10	0.059	8.2°	6.35°

Table 3.3 Estimated motion parameters, MSE, and Av. Ang. error using the discrepancy principle on the Yosemite video sequence

$\lambda_Y^2 = 1$					$\lambda_Y^2 = 10$					$\lambda_Y^2 = 50$				
σ	α	MSE	μ_{AE}	δ_{AE}	σ	α	MSE	μ_{AE}	δ_{AE}	σ	α	MSE	μ_{AE}	δ_{AE}
0.5	1	2.62	28.1°	25.6°	0.5	2.5	2.12	25.4°	22.6°	0.5	2.5	2.12	25.4°	22.6°
0.75	1	1.98	17.4°	19.2°	0.75	5	1.5	14.8°	14.4°	0.75	10	**1.4**	15.7°	**14.1°**
1.0	2.5	1.41	**12.3°**	14.4°	1.0	10	1.34	15.6°	14.4°	1.0	20	1.6	21.5°	16.9°

Table 3.4. Estimated motion parameters, MSE, AAE, and STD using the discrepancy principle on the Translating Tree video sequence.

$\lambda_{TT}^2 = 0.01$					$\lambda_{TT}^2 = 0.25$					$\lambda_{TT}^2 = 0.5$				
σ	α	MSE	μ_{AE}	δ_{AE}	σ	α	MSE	μ_{AE}	δ_{AE}	σ	α	MSE	μ_{AE}	δ_{AE}
0.5	0.3	1.24	28.5°	23.9°	0.5	0.7	1.03	26.7°	22.1°	0.5	0.9	0.98	25.9°	21.3°
1.0	**1.0**	**0.03**	**2.2°**	**2.1°**	1.0	2.5	0.07	3.2°	3.4	1.0	5	0.17	5.5°	5.7°
1.5	2.5	0.12	4.6°	5.1°	1.5	5	0.24	7.2°	7.3°	1.5	5	0.24	7.2°	7.3°
2.0	2.5	0.17	6.3°	6.3°	2.0	5	0.31	9.4°	8.5°	2.0	5	0.31	9.4°	8.5°
2.5	2.5	0.24	8.0°	8.1°	2.5	5	0.38	11.2°	9.9°	2.5	5	0.38	11.2°	9.9°
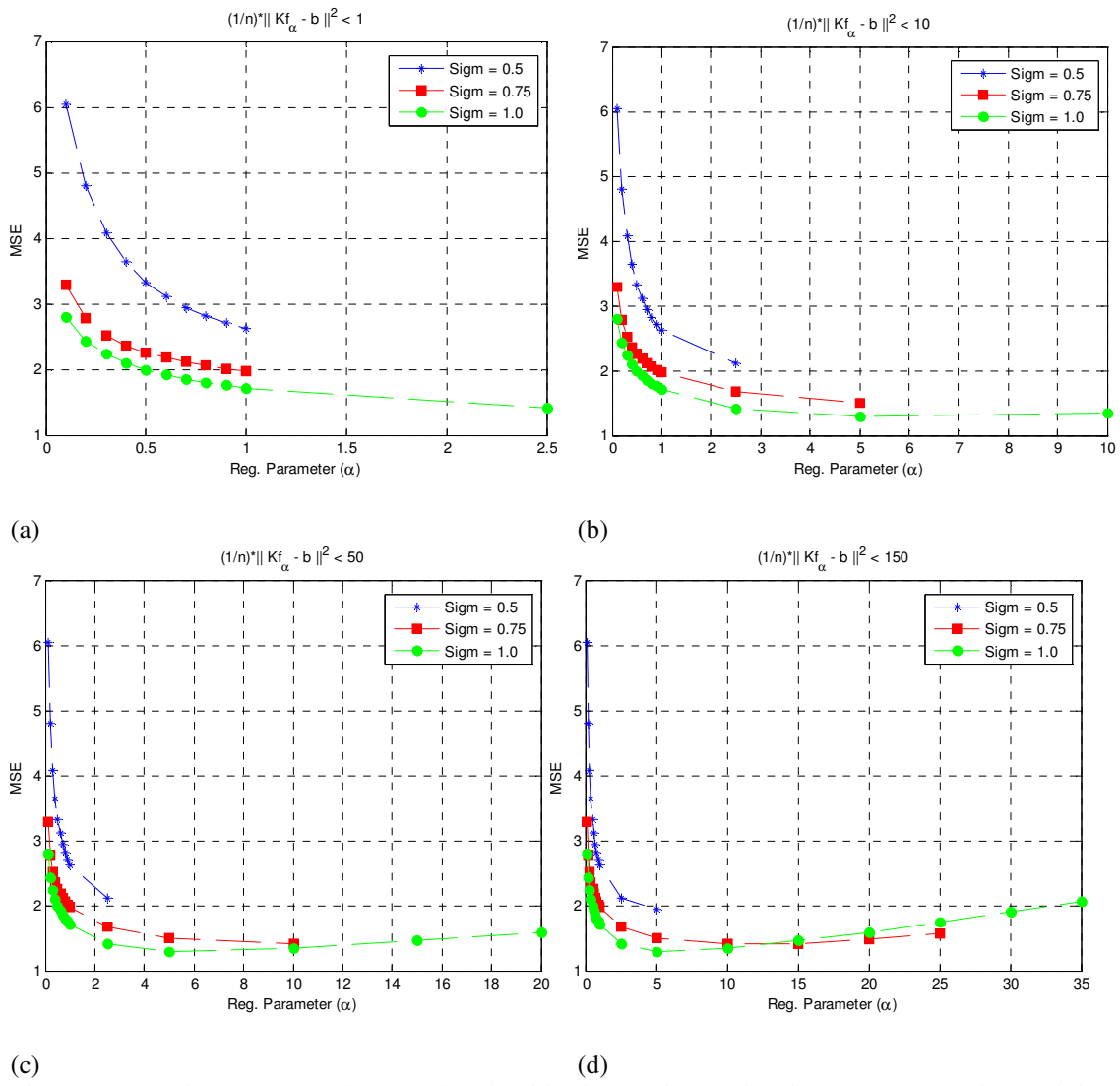
(a)

(b)

(c)

(d)

Figure 3.7. Velocity MSE plots produced with the motion estimation parameters that satisfy the discrepancy principle for different values of error variance, λ_γ, in the Yosemite video sequence. (a) Velocity MSE for $\lambda_\gamma = 1$. (b) Velocity MSE for $\lambda_\gamma = 10$. (c) Velocity MSE for $\lambda_\gamma = 50$. (d) Velocity MSE for $\lambda_\gamma = 150$.

.
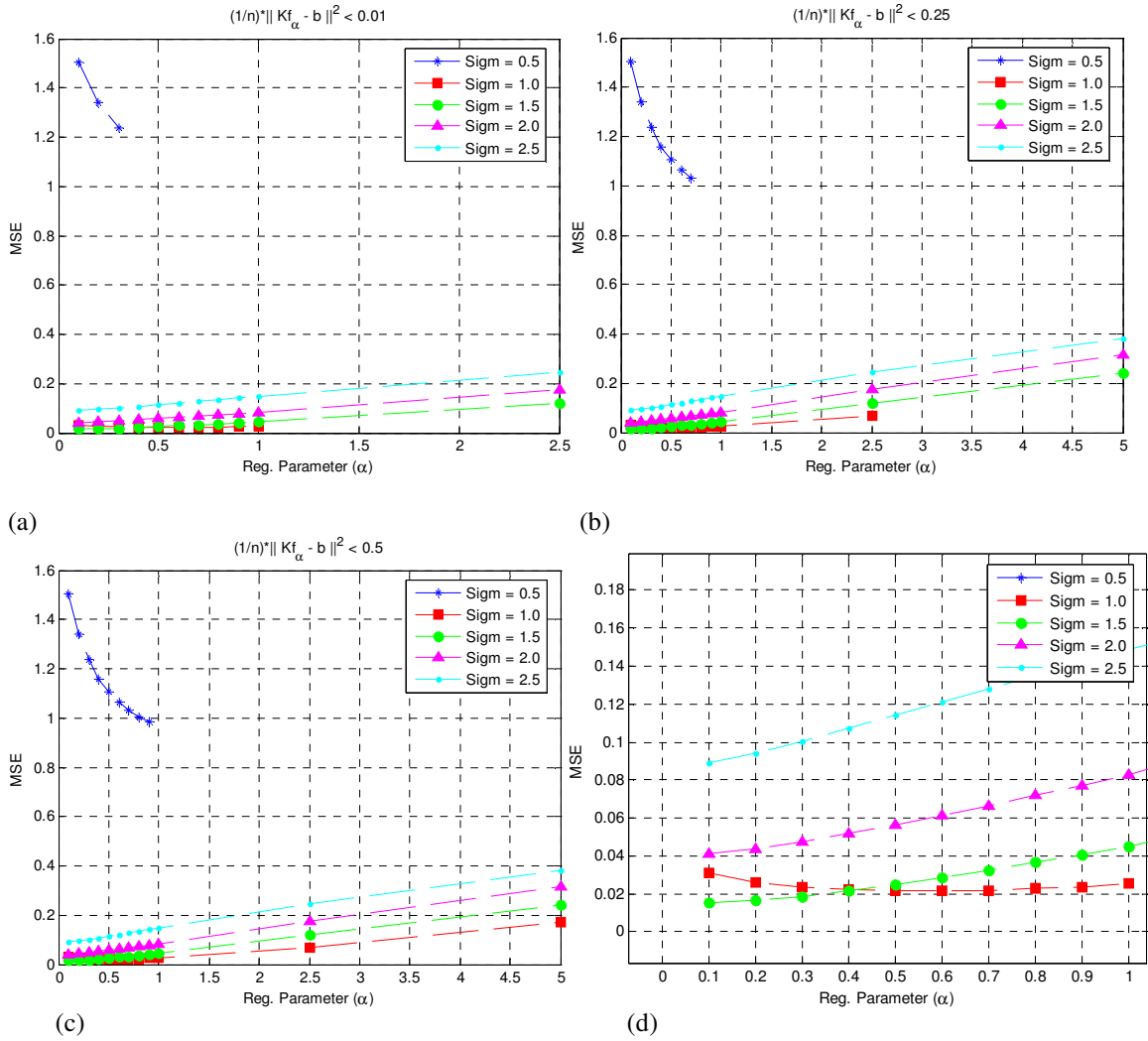
Figure 3.8. Velocity MSE plots produced with the motion estimation parameters that satisfy the discrepancy principle for different values of error variance, λ_{TT}, in the Translating Tree video sequence. (a) Velocity MSE for $\lambda_{TT} = 0.01$. (b) Velocity MSE for $\lambda_{TT} = 0.25$. (c) Velocity MSE for $\lambda_{TT} = 0.5$. (d) The minimum MSE is obtained for $\sigma = 1.5$ and $\alpha = 0.1$.
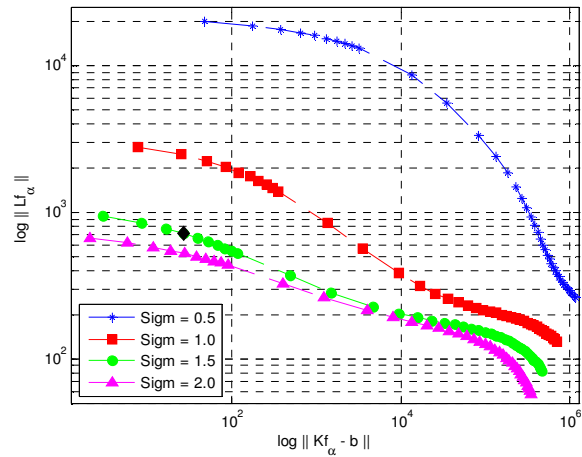
(a)
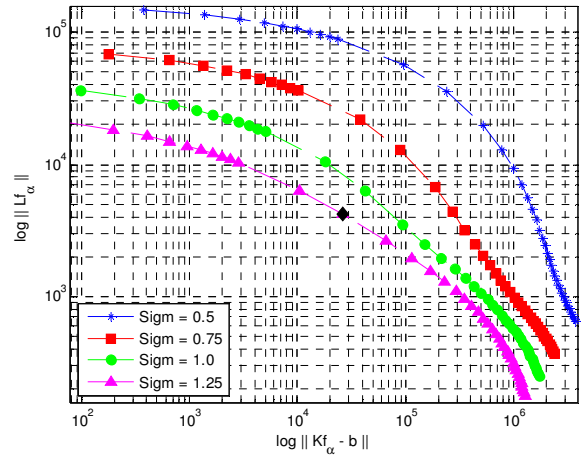
(b)
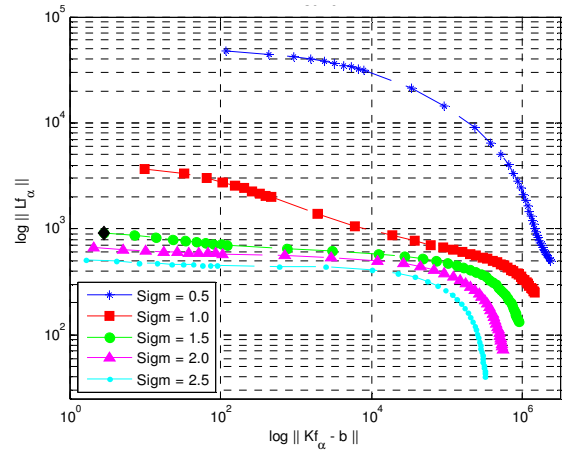
(c)

Figure 3.9. L-curve plots for (a) Divergence Tree, (b) Yosemite, and (c) Translating Tree sequences. The point that achieves the minimum MSE is marked with a dark diamond. Note that in (a)-(c) the optimal point is not associated with the lowest curve. The lowest curve provides lower values for the energy functionals.
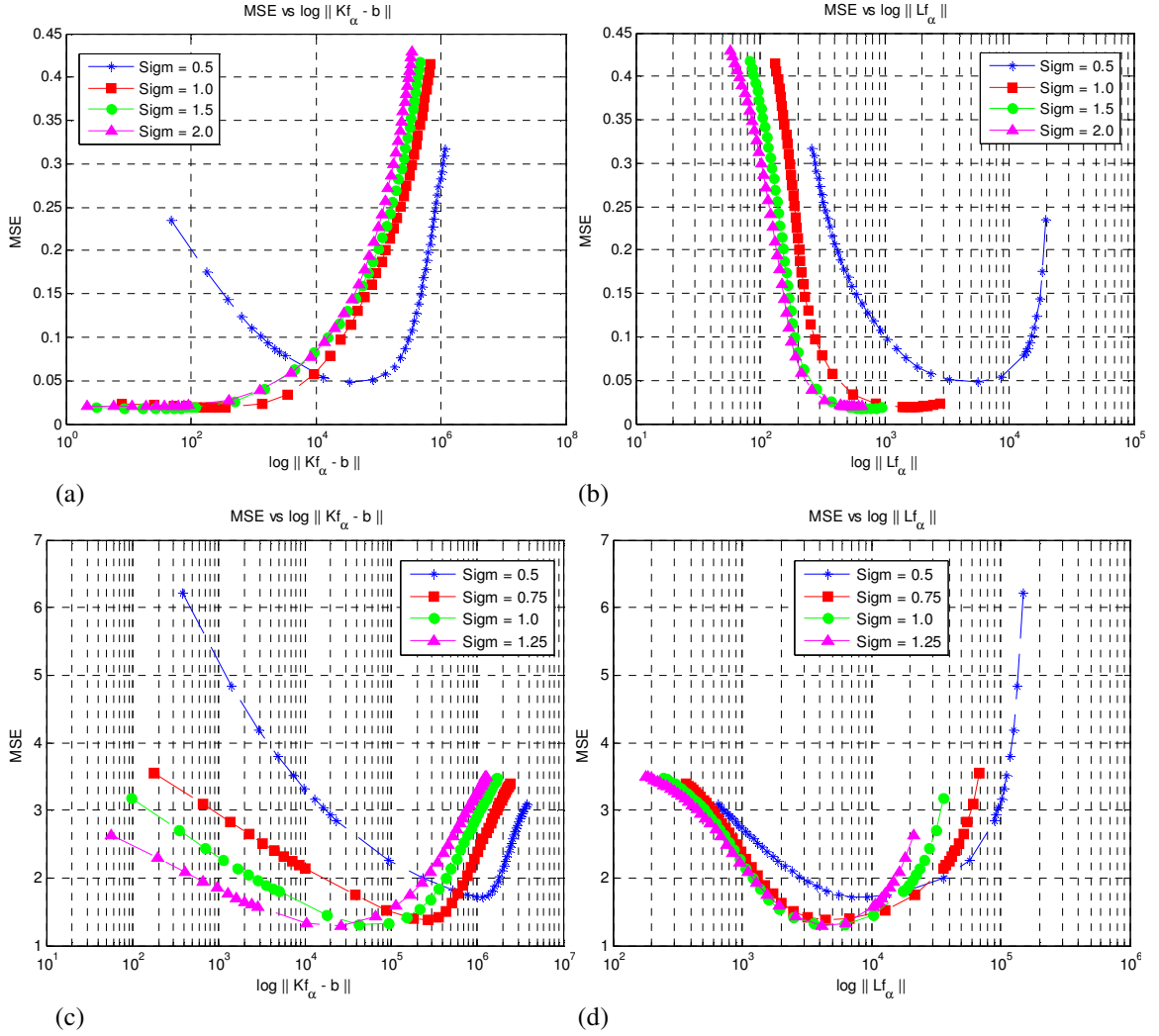
Figure 3.10. Plots of the MSE vs. the norm in logarithmic scale of the data fidelity and regularization terms. (a)-(b) Divergence Tree sequence. (c)-(d) Yosemite sequence.

Table 3.5. L-curve parameter values for the point that achieves the minimum MSE for the Divergence Tree sequence.

σ	α	$\left\|Kf_\alpha - b\right\|$	$\left\|Lf_\alpha\right\|$	MSE	μ_{AE}	δ_{AE}
0.5	5	3.45×10^4	5.58×10^3	0.048	7.8°	5.5°
1.0	0.9	309.4	1.45×10^3	0.019	4.8°	2.5°
1.5	**0.4**	**27.3**	**711.57**	**0.018**	**4.7°**	**2.48°**
2.0	0.3	11.7	575.79	0.020	5.1°	3.14°

60

Table 3.6. L-curve parameter values for the point that achieves the minimum MSE on the Yosemite sequence.

σ	α	$\left\|K\mathbf{f}_\alpha - \mathbf{b}\right\|$	$\left\|L\mathbf{f}_\alpha\right\|$	MSE	μ_{AE}	δ_{AE}
0.5	20	9.77×10^5	9.17×10^5	1.75	21.7°	17.3°
0.75	10	1.83×10^5	6.65×10^3	1.40	15.7°	14.1°
1.0	5	4.22×10^4	6.28×10^3	1.31	12.9°	13.6°
1.25	**5**	$\mathbf{2.63\times10^4}$	$\mathbf{4.18\times10^3}$	**1.29**	**13.5°**	**14.8°**
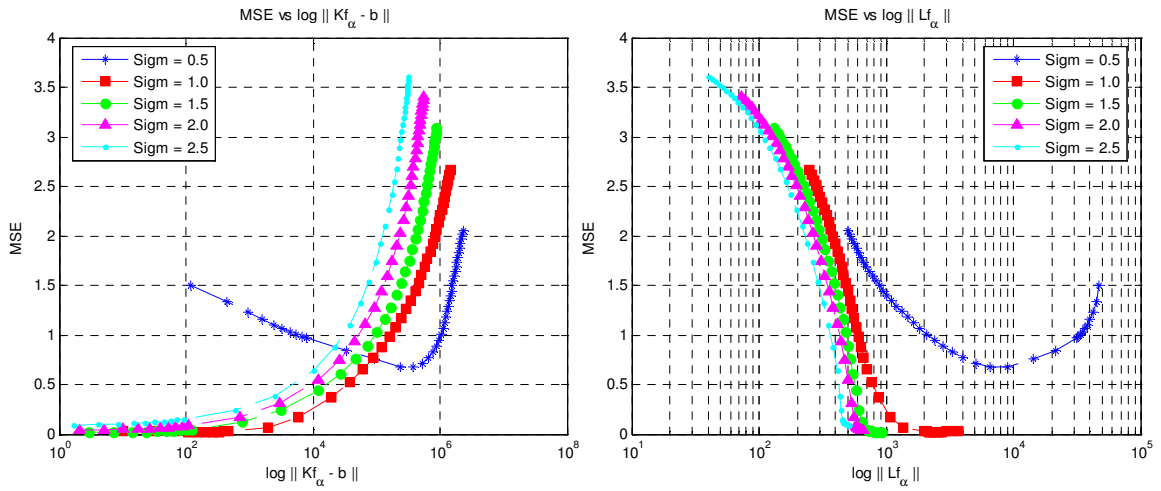
(a)   (b)

Figure 3.11 Plots of the MSE vs. the norm in logarithmic scale of the data fidelity (a) and regularization terms (b) for the Translating Tree sequence.

Table 3.7. L-curve parameter values for the point that achieves the minimum MSE on the Translating Tree sequence.

σ	α	$\left\|K\mathbf{f}_\alpha - \mathbf{b}\right\|$	$\left\|L\mathbf{f}_\alpha\right\|$	MSE	μ_{AE}	δ_{AE}
0.5	15	3.8×10^5	6.5×10^3	0.67	15.95°	17.3°
1.0	0.6	208.62	2.42×10^3	0.022	2.17°	2.11°
1.5	**0.1**	**2.9**	**924.96**	**0.015**	**1.70°**	**1.93°**
2.0	0.1	2.07	664.12	0.04	2.68°	3.38°
2.5	0.1	1.66	514.92	0.09	4.18°	5.21°

3.4.4 Results for full-reference global optimization

Figure 3.12 shows the contour plots of the MSE of the velocity magnitude for the diverging tree. The minimum MSE value is $0.018 \left(pixels / frame \right)^2$ for $\alpha = 0.39$, and $\sigma = 1.52$ pixels. Figure 3.13 presents the velocity magnitude error in logarithmic scale, and figure 3.14 shows the ground truth field and the optimal estimated field. The optimal parameters along with the performance metrics are shown in table 3.8.

Results for the Yosemite sequence were computed in the same manner as those previously published in the literature. Since this is a very challenging sequence, there are results with the clouds and without them because this is a region where the image intensities are not constant. Thus, global optimization is applied to the video with and without the clouds. Results are presented in figure 3.15 and figure 3.16. The minimum MSE is 1.27 for $\alpha = 4.5$ and $\sigma = 1.2$. Results for the Translating Tree sequence are shown in figure 3.17 and figure 3.18. Table 3.8 shows the optimal parameters along with the performance metrics for all three test videos.

Table 3.8. Global optimization results for Horn's method on the three test video sequence.

	Div. Tree	Trans. Tree	Yosemite w/o clouds	Yosemite
σ	1.51	1.26	1.25	1.19
α	0.39	0.2	0.68	4.53
MSE	0.0180	0.014	0.158	1.28
μ_{AE}	$4.73°$	$1.65°$	$5.23°$	$12.94°$
δ_{AE}	$2.52°$	$1.96°$	$7.07°$	14.67
Density	99.7%	99.7%	77.1%	99.7%

For Lucas' technique, optimization is performed to find the value of the spread of Gaussian pre-smoothing filter. The closed set values $P = \{\sigma \in \Re : a \leq \sigma \leq b\}$ for both test video sequences is shown in table 3.9.

Table 3.9. Motion estimation parameter space. Lucas' method.

	Divergence Tree		Translating Tree		Yosemite	
	a	b	a	b	a	b
σ	0.25	1.75	0.25	4.0	0.25	1.25

Global optimization results for this motion estimation method are shown in figure 3.19 and figure 3.20 for the divergence tree sequence. The filter's spread that produces the minimum MSE error of 0.013 (pixels/frame)^2 is $\sigma = 1.53$ pixels. Figure 3.21 and figure 3.22 show results for the Yosemite sequence. The minimum attained MSE is 0.71 (pixels/frame)^2 for a filter spread $\sigma = 1.25$ pixels. The minimum MSE of the Translating Tree sequence is 0.017 (pixels/frame)^2 for a filter spread $\sigma = 1.75$ pixels.

Table 3.10. Optimal results obtained with Lucas's technique.

	Div. Tree	Trans. Tree	Yosemite w/o clouds.	Yosemite
σ	1.53	2.85	1.25	1.25
MSE	0.013	0.0012	0.111	0.71
μ_{AE}	4.35°	0.62°	4.11°	8.68°
δ_{AE}	1.9°	0.53°	5.89°	14.9°
Density	84.1%	66.6%	63.6°	73.7%

Optimization for the average angular error was also performed on the three test videos. The results are presented in table 3.11 for Horn's method, and table 3.14 for Luca's technique. Additionally, error plots of the angular error are also provided.
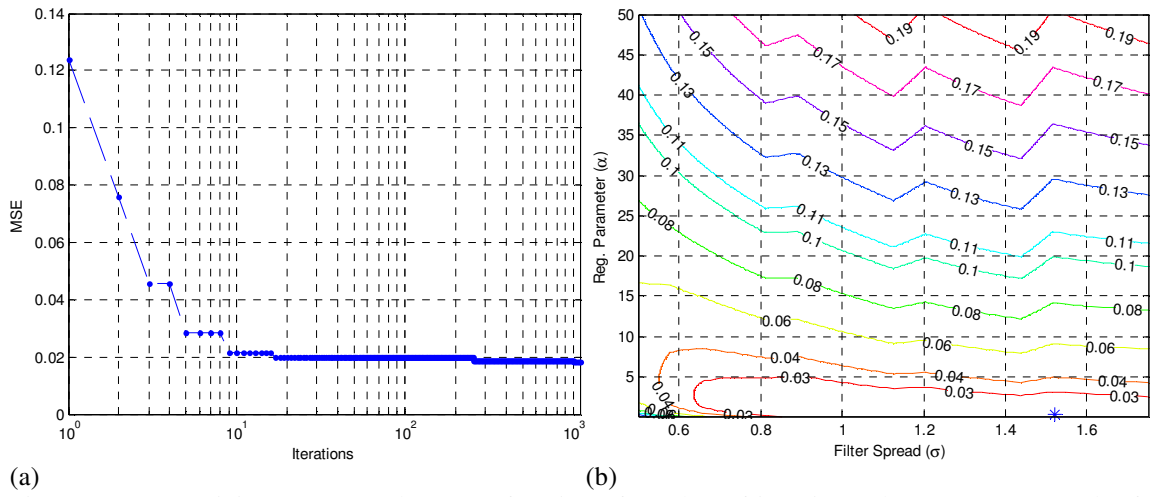
Figure 3.12. (a) Minimum MSE values as a function of number of iterations. (b) MSE contours plot for Horn's motion estimation technique on Divergence Tree video sequence. Optimal point is marked with a blue '*'.
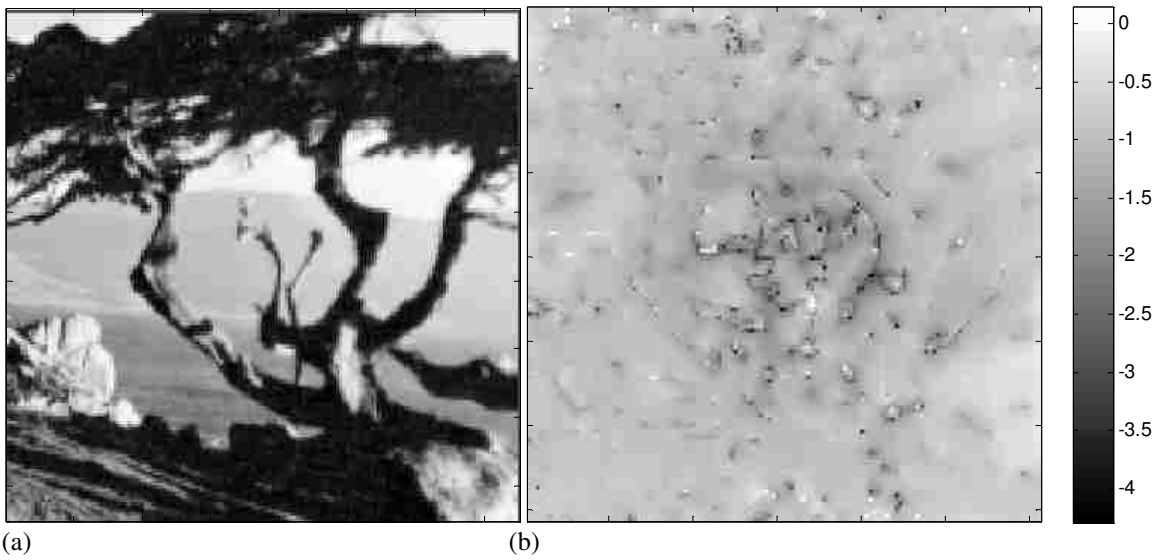
Figure 3.13. (a) Divergence Tree video frame. (b) Logarithmic scale plot of the velocity magnitude error calculated with the optimal motion estimation parameters of Horn's technique.

(a)
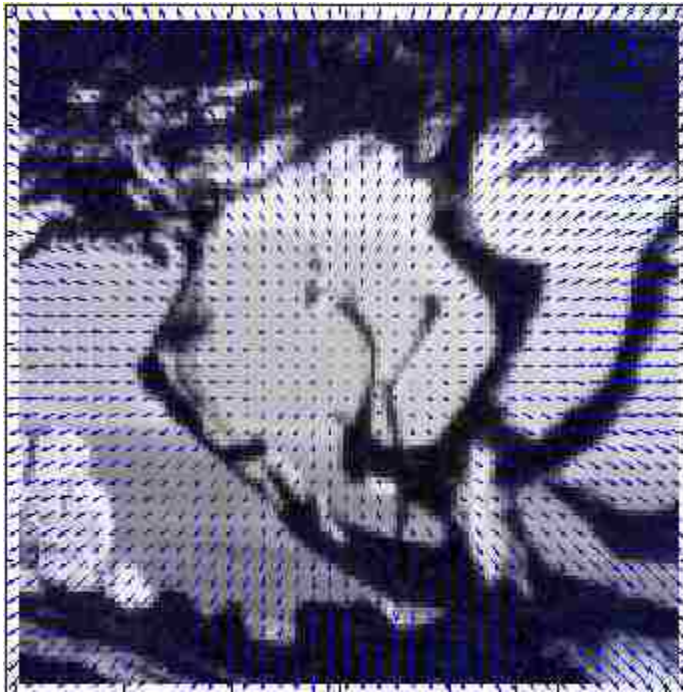
(b)

Figure 3.14. (a) Ground truth velocity field and (b) estimated field over a frame of the Divergence Tree sequence. The difference or error in the estimation is shown in figure 3.13.

(a)

(b)
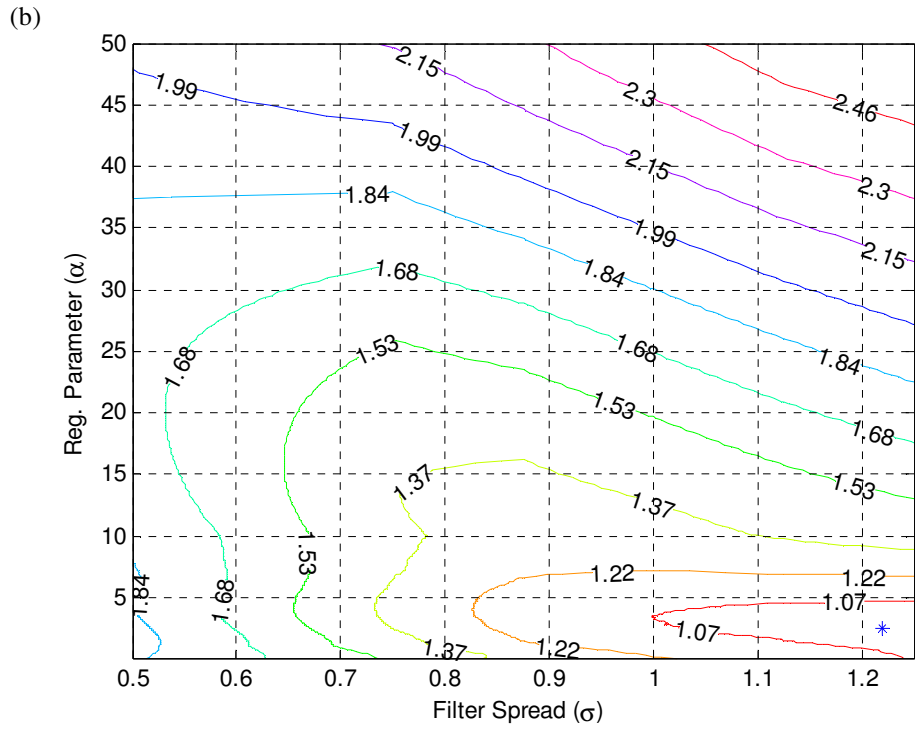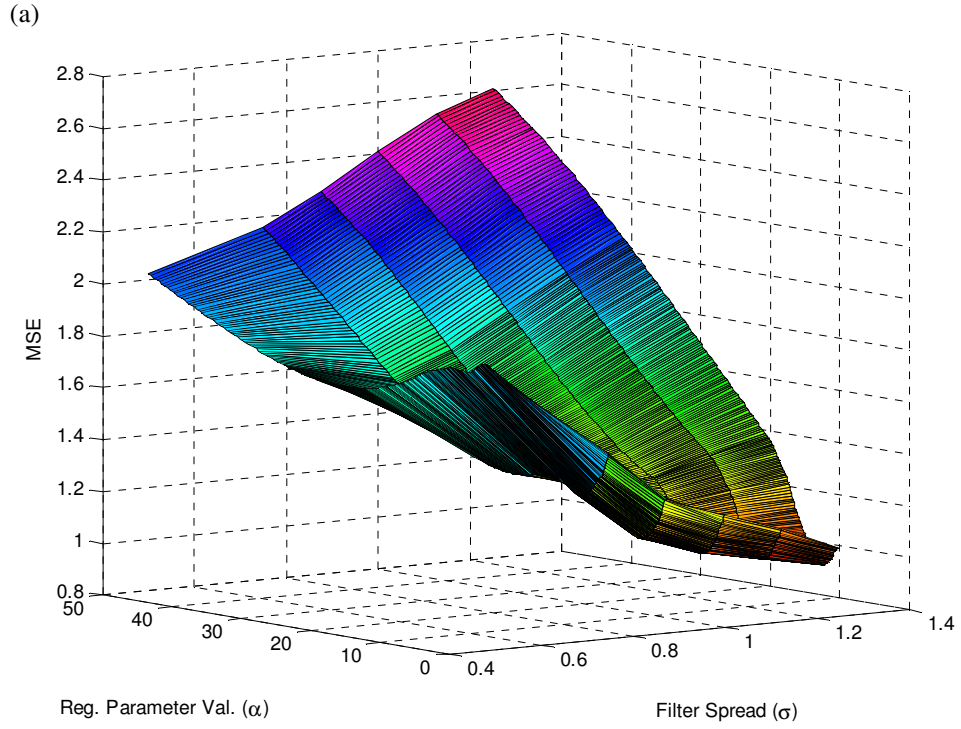
Figure 3.15. (a) MSE surface and (b) contour plots for the Yosemite video sequence. The minimum MSE is marked with a blue '*'.
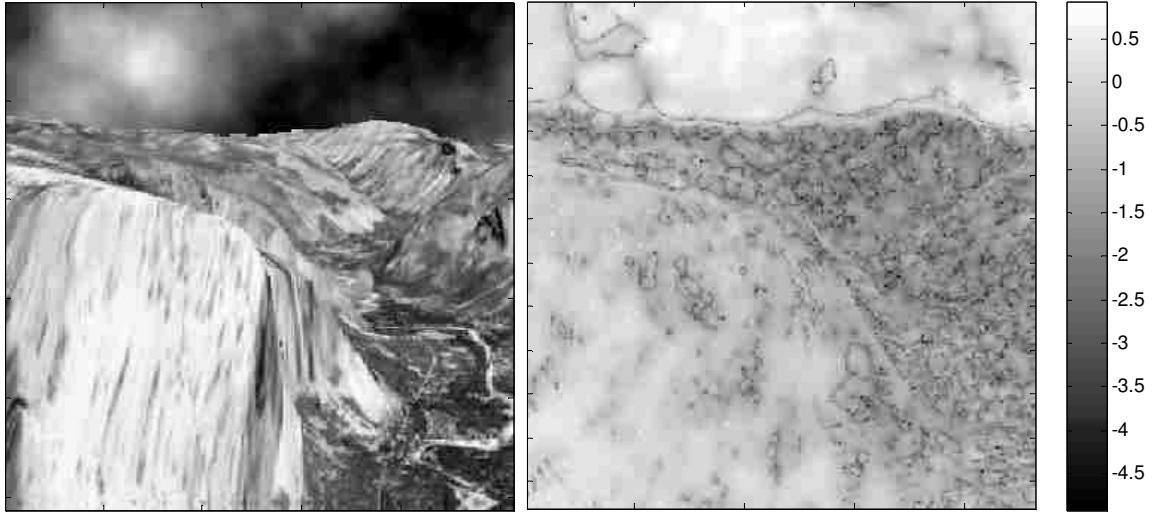
Figure 3.16. (a) Yosemite video frame. (b) Logarithmic plot of the velocity magnitude error using Horn's technique.
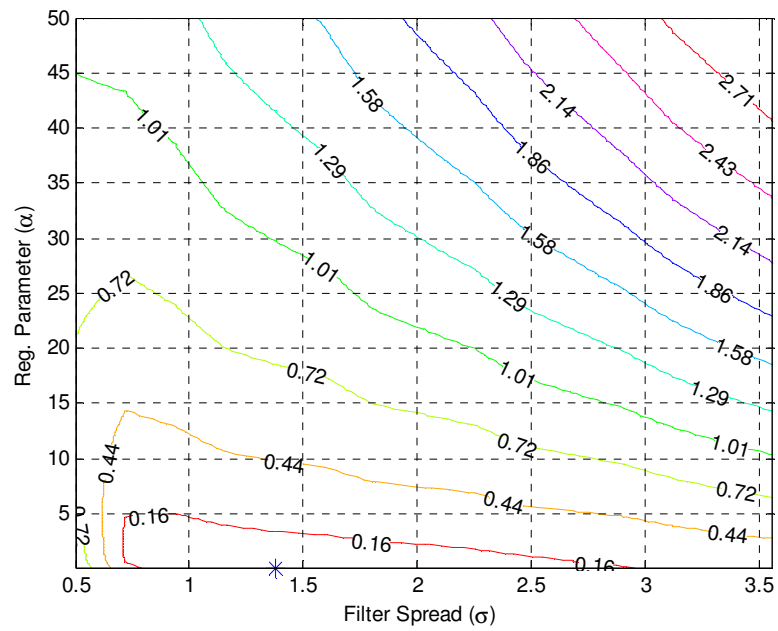
Figure 3.17. MSE contour for the Translating Tree video sequence. The minimum MSE is marked with a blue '*'.
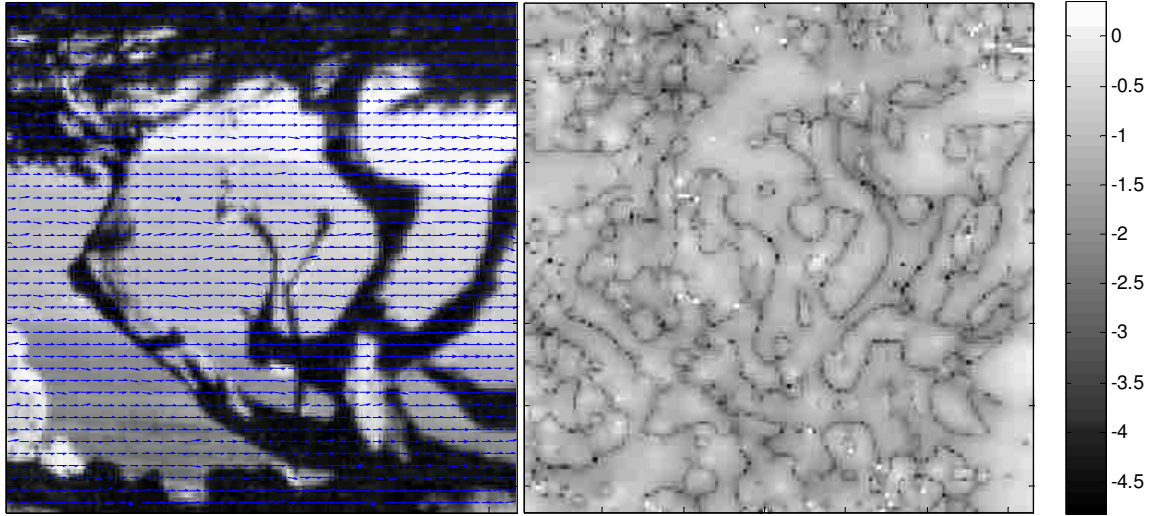
(a)   (b)

Figure 3.18. (a) Estimated velocity field and (b) logarithmic plot of the velocity magnitude error of the Translating Tree sequence. Horn's method.
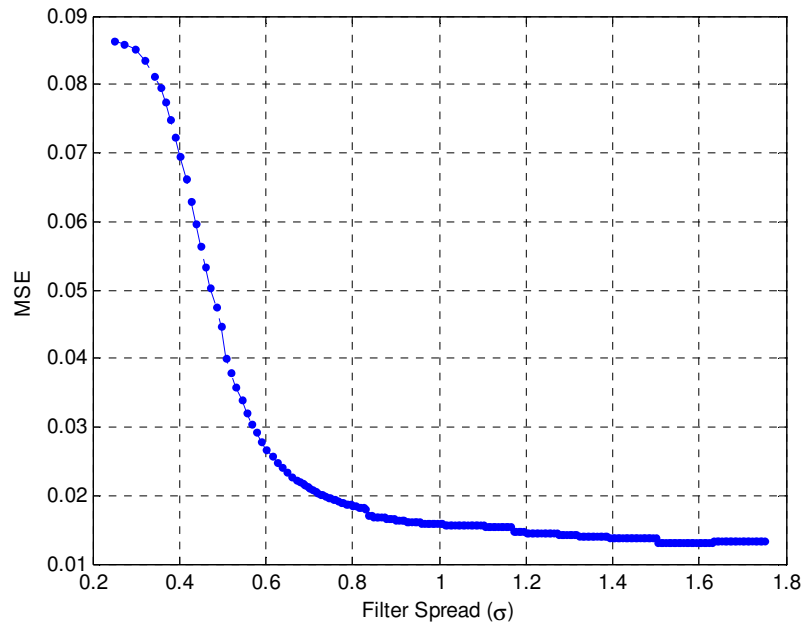
Figure 3.19. MSE as a function of the filter spread. Luca's method on Divergence Tree sequence.
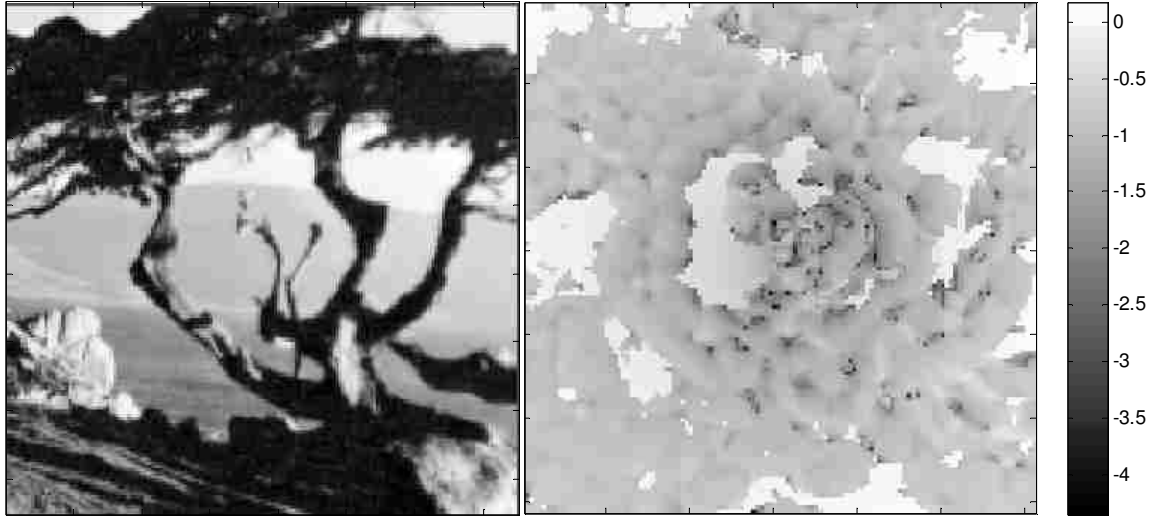
Figure 3.20. (a) Divergence tree video frame. (b) Logarithmic scale plot of the velocity magnitude error using Lucas's technique.
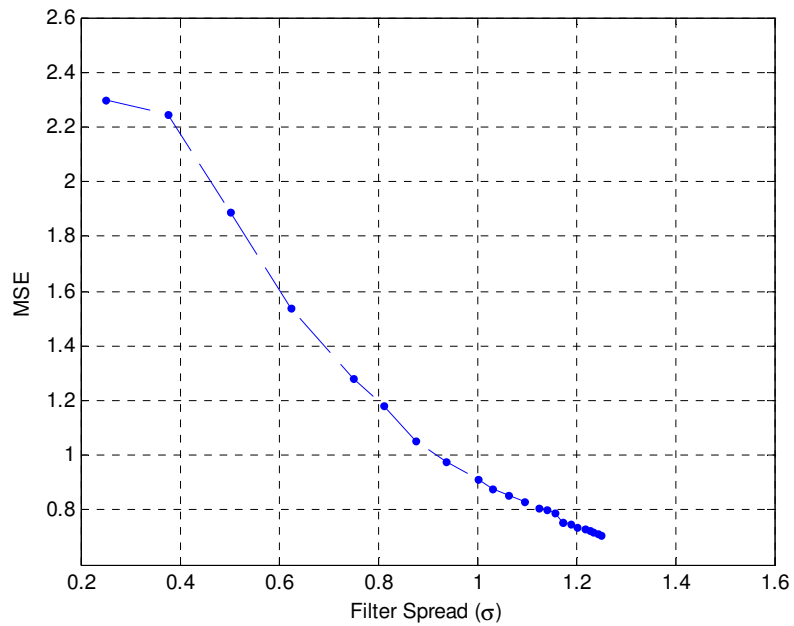
Figure 3.21. MSE as a function of the filter spread. Lucas's method on Yosemite sequence. Note that the maximum value of $\sigma = 1.25$ pixels is due to the limited number of frames. See section 3.3.5.
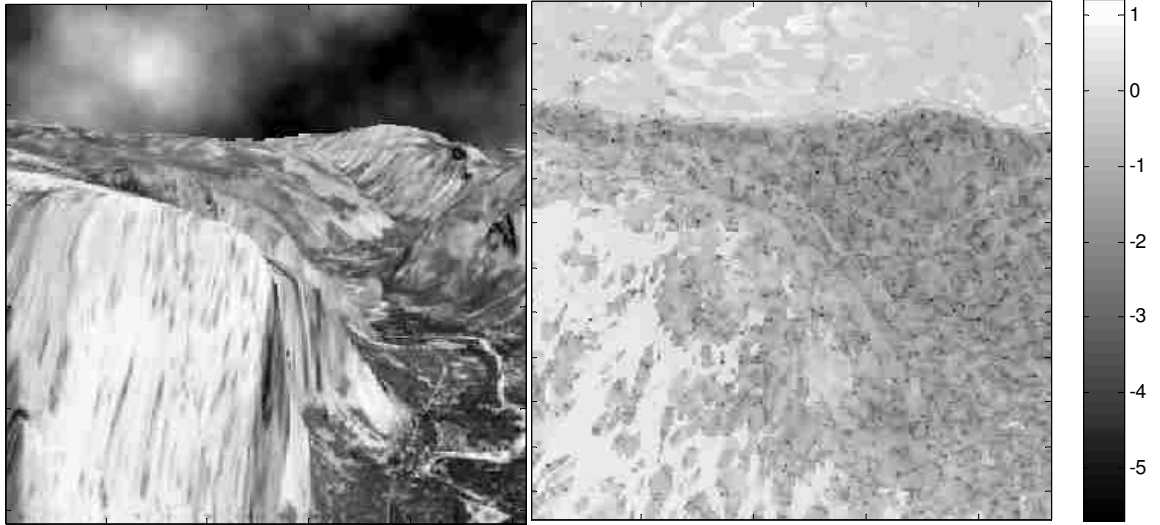
(a)   (b)

Figure 3.22. (a) Yosemite video frame. (b) Logarithmic scale plot of the velocity magnitude error using Lucas's technique.
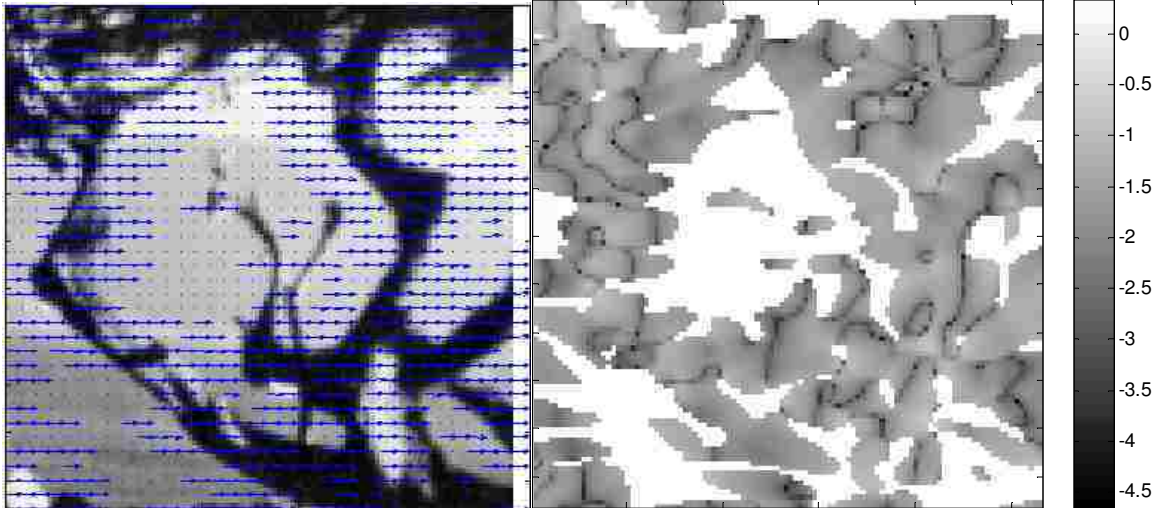
(a)   (b)

Figure 3.23. Lucas's method results on the Translating Tree sequence. (a) Estimated velocity field and (b) logarithmic plot of the velocity magnitude error

Table 3.11. Angular optimization results obtained with Horn's technique.

	Div. Tree	Trans. Tree	Yosemite w/o clouds.	Yosemite
α	1.11	0.1	1.07	1.66
σ	1.28	1.38	1.25	1.25
MSE	0.0188	0.0148	0.1645	1.42
μ_{AE}	4.63°	1.67°	5.13°	11.67°
δ_{AE}	2.22°	2.14°	6.75°	15.8°
Density	99.8%	99.6%	77.1%	99.7%

Table 3.12. Angular optimization results obtained with Lucas's technique.

	Div. Tree	Trans. Tree	Yosemite w/o clouds.	Yosemite
σ	1.51	2.82	1.25	1.25
MSE	0.0132	0.0013	0.1114	1.16
μ_{AE}	4.35°	0.62°	4.11°	9.04°
δ_{AE}	1.9°	0.53°	5.89°	15.38°
Density	84.4%	66.17%	63.6°	74.42%

(a)   (b)
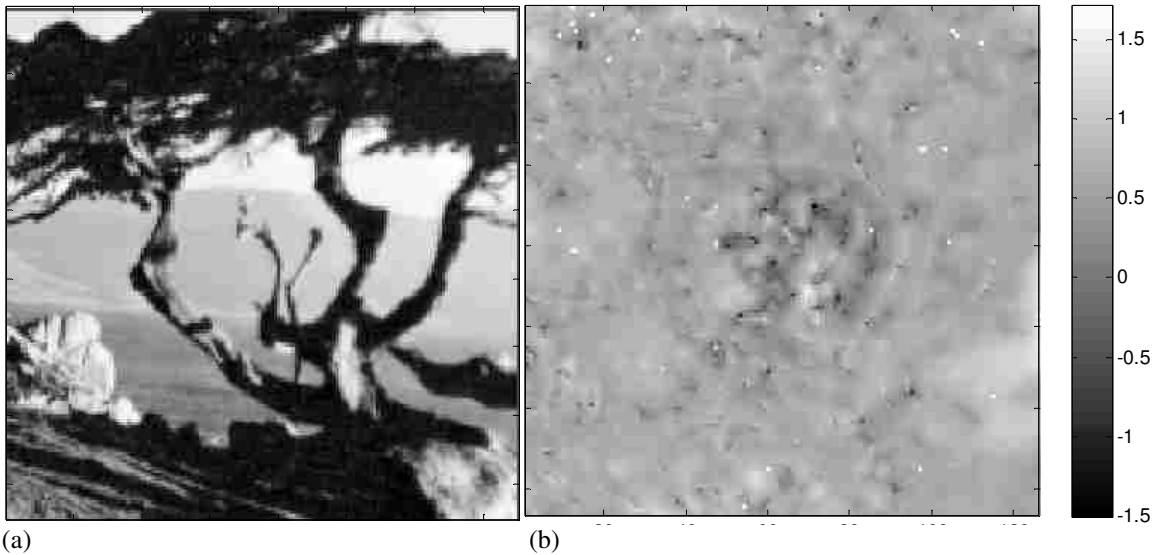
Figure 3.24. (a) Divergence Tree video frame. (b) Horn's angle error results in logarithmic scale.

71
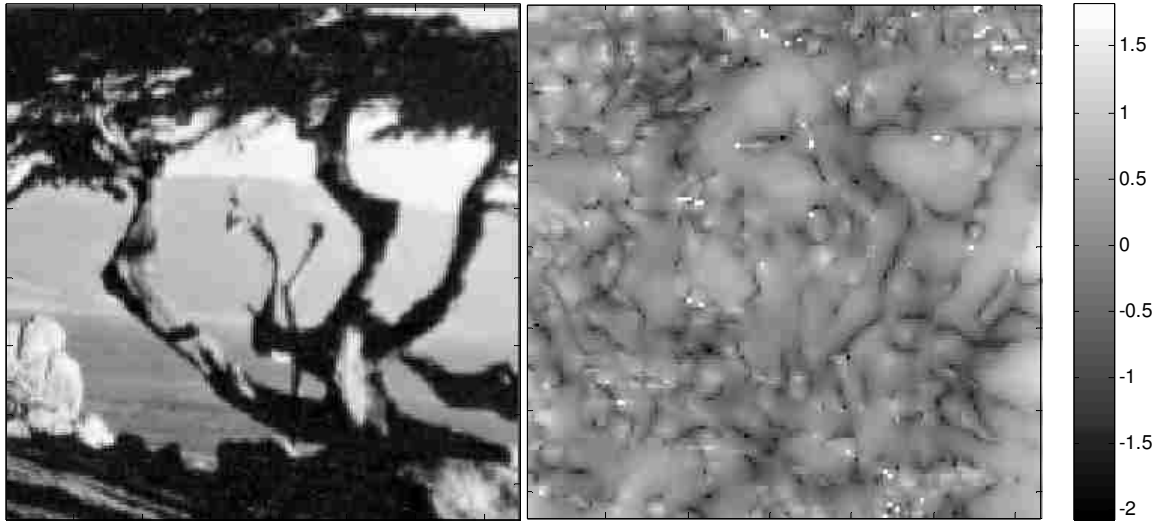
(a)   (b)

Figure 3.25. (a) Translating Tree video frame. (b) Horn's angle error results in logarithmic scale.
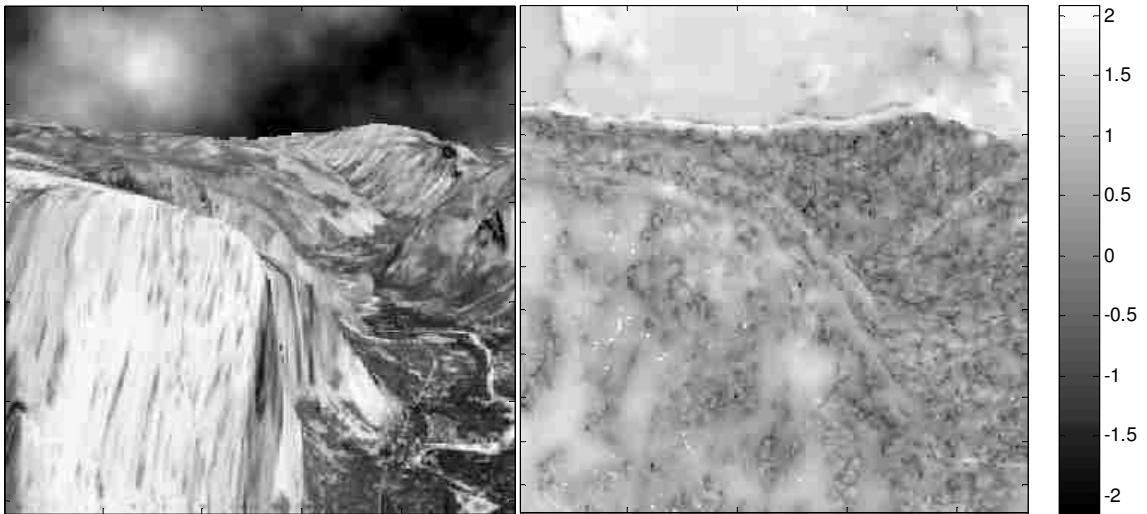
(a)   (b)

Figure 3.26. (a) Yosemite video frame. (b) Horn's angle error results in logarithmic scale.
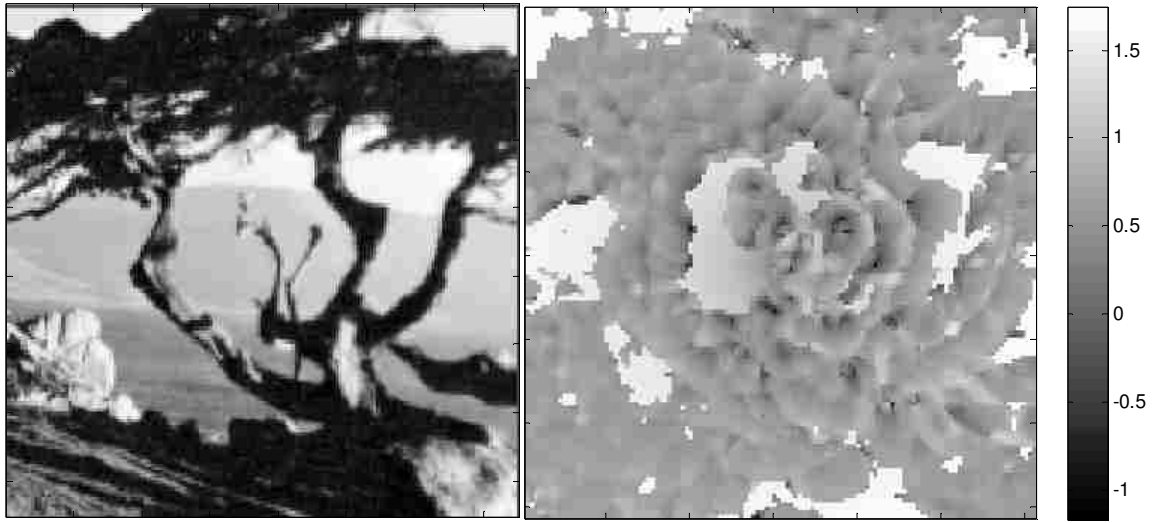
(a)   (b)

Figure 3.27. (a) Divergence Tree video frame. (b) Lucas's angle error results in logarithmic scale.
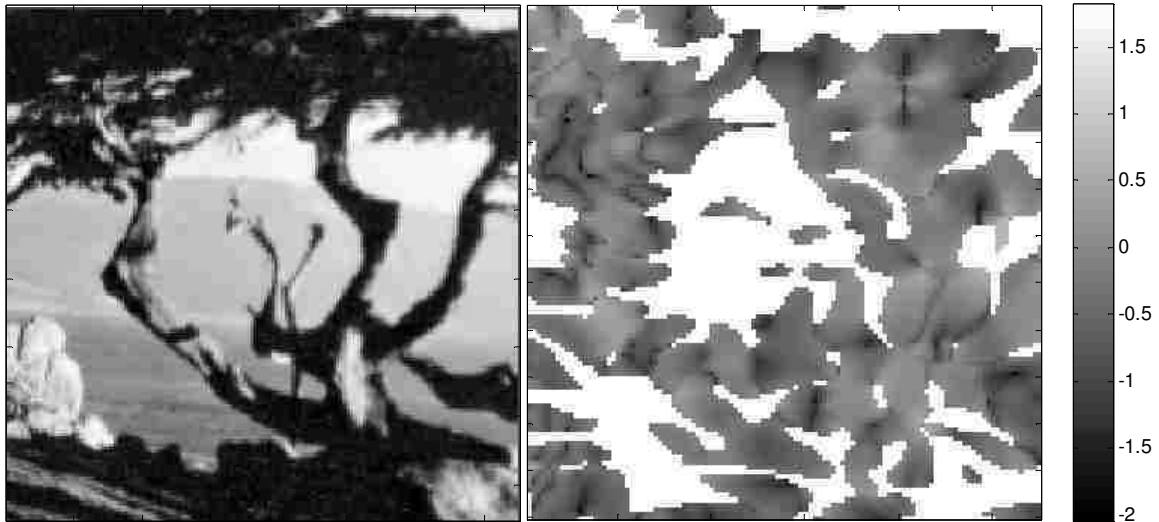
(a)   (b)

Figure 3.28. (a) Translating Tree video frame. (b) Lucas's angle error results in logarithmic scale.

(a)   (b)
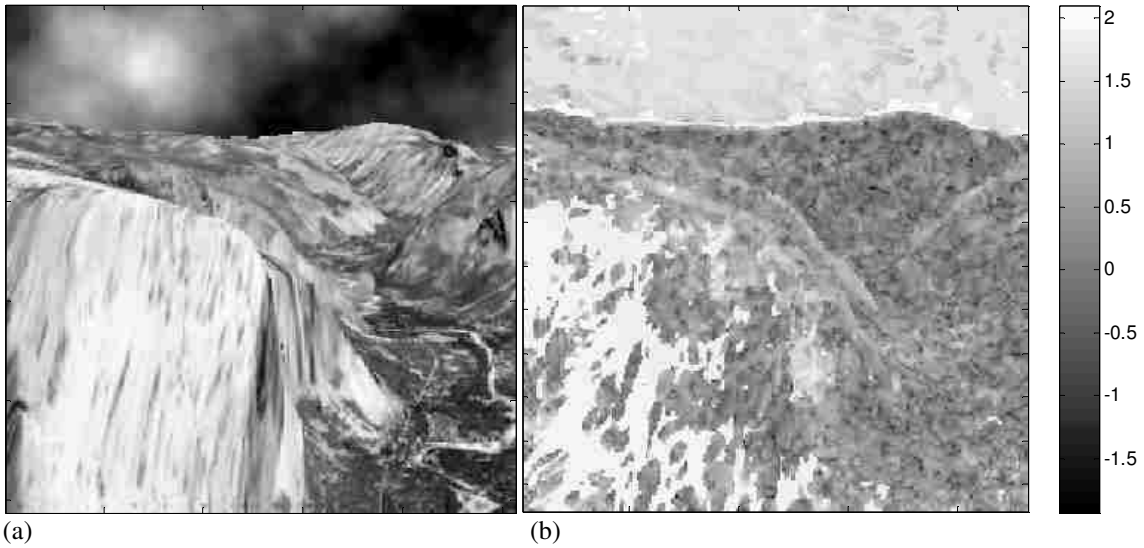
Figure 3.29. (a) Yosemite video frame. (b) Luca's angle error results in logarithmic scale.

3.5 Discussion

This section analyzes the results obtained with the methods presented in this chapter and compares them with previously published results. The motion estimation parameter values for the Divergence Tree, Translating Tree, and Yosemite sequences are taken from [27] were $\sigma = 1.5$ for both Tree sequences, and $\sigma = 1.25$ for Yosemite. The regularization parameter of Horn's technique was set to $\alpha = 0.5$ for all three videos. Table 3.13 and table 3.14 show the error metrics produced with those parameter values for these video sequences.

Table 3.13. Published error metrics for Horn's method.

Video	MSE	μ_{AE}	δ_{AE}	%
Divergence Tree	0.0179	4.69°	2.41°	100
Translating Tree	0.0249	2.02	2.27	100
Yosemite w.c.	1.85	12.05°	16.8°	100
Yosemite w/o c.	0.16	5.35°	7.35°	77.1

74

Table 3.14. Published error metrics for Lucas's method.

Video	MSE	μ_{AE}	δ_{AE}	%
Divergence Tree	0.0131	4.35°	1.9°	84.5
Translating Tree	0.0025	0.81°	0.93°	74.5
Yosemite w.c.	1.16	9.0°	15.38°	74.4
Yosemite w/o c.	0.11	4.11°	5.89°	63.6

3.5.1 Discussion for inverse problem approaches to select the regularization parameter

The discrepancy principle relies on knowledge about the size of the residual norm, $\left\| K\mathbf{f}_\alpha - \mathbf{b} \right\|^2$, which changes considerable for different motion patters as shown in figure 3.6 and figure 3.7.

Note that according to equation (3.4), the calculated regularization parameters in table 3.2 and table 3.3 do not produce the smaller MSE nor beats previously published results, see table 3.13 and table 3.14. In order to achieve this optimal value, the variance of the error in the discrepancy principle should be set to smaller values, but since it is always safer to over-estimate this parameter, then the optimal value is most likely to be missed. Prior knowledge about the error is only based on subjective expectations.

Results for Yosemite sequence provide a good example of how unfounded expectations about the behavior of the regularized residual lead to inaccurate velocity estimations. Notice that variance values of 10 and 50 (really high values) had to be used in order to obtain a low MSE.

The optimal regularization parameter value is linked to the underlying video motion. In order to pick up a good value, there has to be some prior knowledge of the underlying motion. Such knowledge is very difficult to obtain or assume even when working with

ground truth sequences. Moreover, there is no clear method to estimate the bound on the residual norm for non-ground truth based sequences that guaranties accurate solutions. If one chooses to only minimize the residual norm, the solution will be over-fitted to the data fidelity term, $\|K\mathbf{f}_\alpha - \mathbf{b}\|^2$, which depends exclusively on the ill-posed system of equations $K\mathbf{f} = \mathbf{b}$. Therefore, it is logical to not bound the residual norm to small values, so that the regularization term can influence the solution. However, in this case over-regularized estimates are most likely to be calculated since knowledge of λ is difficult to quantize.

In either case, the optimal parameter values are not found. The MSE as shown in figure 3.6 and figure 3.7 is further reduced by the correct choice of the pre-smoothing spread. The discrepancy principle does not allow the estimation of this parameter. Its application to video sequences without ground truth will be considerably limited because guessing a bound for the residual norm will lead to a heuristic approach that will certainly be the source of considerable inaccuracies. This method does not solve the problem of calculating the optimal regularization parameter, and it is not different from current arbitrary choices. At the end, the heuristic choice of α is replaced by another heuristic. One that chooses the value of λ.

In the case of the L-curve method, the idea is to find some correlation between the regularization parameter that produces the smallest MSE and the corner of the L-curve. The logarithmic scale plots do not show a clear "L" shape, but there is an approximate corner for the divergence tree sequence, figure 3.9(a). The values of $\log\|K\mathbf{f}_\alpha - \mathbf{b}\|$ and $\log\|L\mathbf{f}_\alpha\|$ for which the minimum MSE is attained are marked with a dark diamond in the

76

"L-curve" plots, see figure 3.9, and located on the left side of the curves where the regularization parameter is not too big (avoiding over-regularization).

The approximate corners of the curves correspond to over-regularized solutions that produce large data fidelity norm values. The plots of the MSE vs. the norm of the data fidelity and regularization terms confirm that under- and over-regularization greatly affect the estimation error, and that picking the correct α value is critical in order to achieve accurate velocity estimates. As in the case of the discrepancy principle, the "L-curve" method does not address how to pick the right amount of smoothing controlled by σ. Moreover, the corner of the curve, if it exists, does not estimate the optimal regularization value that minimizes the MSE or the other error metrics. The "L-curve" plots can be produced for videos without ground truth but from the results, there is no evidence that suggests that α values at the corner of the curve will produce accurate velocity estimates. Note that the approximate corner does not produce the smallest MSE in the simulations.

3.5.2   Discussion for calculating the Lipschitz constant

The objective of the algorithm described in figure 3.5 is to find a Lipschitz constant, $L^{'}$, that allows the minimum of the objective function to be found. Unfortunately, for motion estimation, relationship (3.13) cannot be directly evaluated. The devised alternative approach is based on a large number of iterations and setting the initial value of $L^{'}$ to a big value. Here, note that to apply Lipschitz optimization, the only condition is that the graph of the objective function, f, should not go to infinite. Thus, all jumps among points on the graph of f must be bounded [52], and this bound equals L. Starting with

77

an L' such that $L' > L$, does not stop the algorithm from minimizing f. For successive runs of the algorithm, L' is decreased until relationship (3.12) is no longer satisfied. At this point, the algorithm finds a worse minimum than the ones previously found and stops after few iterations.

Table 3.15 provides the values returned by the algorithm for five runs starting with $L' = 10$ and *MaxIter* $= 1100$ when estimating the minimum MSE for the Translating Tree sequence using Horn's algorithm. The bold faced columns indicate the found L' value and its associated error metrics. The columns in italic are extra runs of the algorithm.

Table 3.15. Optimal point and minimum MSE of the Translating Tree sequence for different values of L'.

	$L' = 10$	$L' = 5$	**L' = 2.5**	$L' = 1.25$	*L' = 0.625*
Iterations	1100	1100	**1100**	989	*185*
MSE	0.014	0.014	**0.014**	0.0144	*0.0148*
α	0.2	0.2	**0.2**	0.3	*0.1*
σ	1.26	1.26	**1.26**	1.26	*1.375*

The algorithm is also applied to the other two test video sequences and the results are presented in table 3.16 and table 3.17.

Table 3.16. Optimal point and minimum MSE of the Divergence Tree sequence for different values of L'.

	$L' = 10$	$L' = 5$	$L' = 2.5$	$L' = 1.25$	$L' = 0.625$	**L' = 0.3125**	*L' = 0.1563*
Iterations	1100	1100	1100	1100	1100	**392**	*96*
MSE	0.018	0.018	0.018	0.018	0.018	**0.018**	*0.0184*
α	0.39	0.39	0.39	0.39	0.39	**0.39**	*0.29*
σ	1.51	1.51	1.51	1.51	1.51	**1.51**	*1.59*

78

Since the three video sequences simulate different types of motion and range of velocities, it is safe to use their optimization results as reference to set the value of L for future motion estimation on other video sequences. The largest L' among all test video sequences that still satisfies condition (3.12) is chosen. Thus, $L' = 2.5$ is set.

Table 3.17. Optimal point and minimum MSE of the Yosemite sequence for different values of L'.

	$L' = 10$	$L' = 5$	$L' = 2.5$	$L' = 1.25$	$L' = 0.625$
Iterations	2000	2000	2000	828	43
MSE	1.28	1.28	1.28	1.30	1.37
α	4.53	4.53	4.53	7.26	13.74
σ	1.19	1.19	1.19	0.97	0.75

3.5.3 Discussion for Lipschitz global optimization results

Using the branch and bound strategy, Lipschitz optimization starts by partitioning the set P in rectangles for which lower bounds of the velocity magnitude MSE are calculated. The region P_i with the smallest lower bound is selected for further branching and bounding; and regions with lower bounds bigger than the current minimum value of the objective function are removed from further branch and bounding. This procedure continues until a maximum number of iterations is reached or when the difference between the current minimum value of the objective function and the lower bound is less than a given tolerance $\varepsilon = 0.01$. This tolerance value provides a reasonable approximation of the lower bound to the current minimum value of the objective function. Additionally, a tolerance in the order of 10^{-2} is a convenient value to pick.

The flat regions where there is no error reduction in figure 3.12 are due to the fact that the algorithm explores subsets of the parameter space where no better minimum value is found, and the optimization proceeds with the current minimum.

When optimizing for the correct value of the spread of the Gaussian pre-smoothing filter and regularization parameter, global optimization results for Horn's technique using the divergence tree sequence accomplish an order of magnitude in MSE reduction, see figure 3.12, The estimated flow shown in figure 3.14 presents higher errors at the periphery of the image, see figure 3.13, because at these regions the velocity field is expanding at a maximum speed of 2.7 pixels/frame. The Yosemite video sequence is far more challenging than the divergence tree sequence. Bigger errors are expected especially over the cloud regions where the intensity conservation assumption is violated and over the image borders where the velocity reaches speeds of 4.5 pixels/frame. The estimation of the correct motion parameters helps to reduce the error as shown in figure 3.15, specially for complicated video sequences because they yield higher errors.

Lucas's technique produces similar MSE values than those produced with Horn's technique but at the cost of lees pixels with velocity estimates, density of the results, because the local solution to (2.35) fails to produce velocity estimates for image regions that lacked texture. Results for the Yosemite sequence are greatly affected by the lack of conservation of image intensity in the cloud region and the large velocity values of the mountains. The problem of estimating large motions arises because of the linear approximation to (2.1) using a Taylor series expansion when the spatial displacements are not small.

The optimal parameters found with the proposed Lipschitz optimization strategy produce error metrics that are comparable with published results. In the case of the Divergence Tree sequence, the minimum MSE is almost the same value as the one produced with the parameters published in [27]. The difference is in the order of 10^{-3}. The angular error metrics are not as good as those of [27] but the differences are not significant.

For the Translating Tree sequence, the error metrics produced with the optimal parameters outperform previously published results for both Horn's and Luca's techniques, see table 3.8, table 3.10, table 3.13, and table 3.14, not only on MSE error but also on the angular metrics.

The real advantage of the implemented optimization algorithm can be seen when the motion of the Yosemite sequence is estimated. Given that this video exhibits a complex motion, the selection of the motion estimation parameters becomes more relevant than in the previous case. The optimal parameters of table 3.8 for Horn's technique produce a minimum MSE value that is thirty percent smaller than the one calculated with the parameters of [27]. Smaller values of the angular errors are also obtained with the optimized parameters. It is important to note that these results are obtained at the same density value. When the cloud region is not considered for calculating error metrics, the optimized parameters also outperform previous published values.

Optimization for the average angular error also produces smaller values than those published in [27] at the same density of estimates. For Horn's technique the optimal errors are smaller for all three sequences, while for Lucas's method only for the Translating Tree sequence optimization yields better results. For the other two sequence the optimal values are almost the same as those of [27].

81

3.6 Concluding Remarks

The importance of optimizing for the correct motion estimation parameter values has been established with the results presented in section 3.4. The contributions of this chapter include:

- The identification of the motion parameters that affect the accuracy of the velocity estimates.

- The implementation of a full-reference global optimization algorithm for motion estimation that successfully calculates the optimal sets of motion parameters.

- The application of the discrepancy principle and L-curve methods to calculate the regularization parameter of motion estimation techniques.

- An algorithm that finds the Lipschitz constant when the maximum of the gradient of the objective function cannot be calculated.

It is important to note that the accuracy of the velocity estimation also depends on the complexity of the motion that is being tried to estimate. When ground truth is available, there are a few options to pick these values even thought some are better than others. Performance measures can be evaluated and decisions can be made regarding the motion parameters.

The problem of selecting the correct parameters increases in complexity when there is no ground truth or reference to evaluate the performance measures. This latter case occurs in reality when video sequences are taken with digital recording devices. The true displacement field is unknown, and accurate velocity estimates are required.

The reminder of this dissertation deals with the estimation of motion parameters when ground truth is not available. This process is analog to walking in the darkness. There is

no light or reference to guide our algorithms to the most accurate velocity estimates. That is why a blind platform will be introduced that calculates the motion parameters that estimate accurate velocity fields when there is no ground truth.

Chapter 4

No-Reference Global Optimization for Motion Estimation

4.1 Motivation

The accuracy of motion estimation techniques when no ground truth is available cannot be qualitatively measured. This problem is present when velocity estimation is performed on any real video sequence because the true motion field is unknown. Performance metrics cannot be evaluated, and the error in the estimation cannot be measured. This is a serious limitation of the motion estimation techniques. So far, for real image sequences, the evaluation of the velocity estimates is drawn from subjective evaluations [27].

4.2 Introduction

The no-reference global optimization platform proposed in this dissertation aims to find the optimal motion estimation parameters when there is no reference or ground truth to guide us to the correct answer. The motion estimation algorithms are blind to the true displacement field, but with the application of the proposed methodology, they will still be able to estimate accurate velocities with certain confidence.

The main contribution of this chapter is to extend the global optimization platform for the application of motion estimation algorithms to real video sequences. The concept behind

the blind validation approach is to optimize for agreement among the estimations. The fact that the true displacement field is unknown does not mean its estimation is not possible. The correct velocity field exists and it is unique. If different motion estimation techniques estimate similar values, then it is expected that the estimates are close to the correct answer. Of course, absolute certainty is not possible.

4.3 Assumptions

This section lays out the theoretical framework for the no-reference global optimization methodology to work. There are only two assumptions:

- With proper parameter optimization, the motion estimation techniques should be able to provide an estimate of the unknown displacement field.
- The estimation methods are sufficiently different so that their agreement will likely correspond to the estimation of the globally optimal motion.

The first assumption also requires that each of the methods can produce a close estimate of the unknown motion field. Since the approach is based on agreement in the estimation, one technique cannot outperform the other. Otherwise, the underachieving method will drive the better method to the wrong estimation. As an example, in chapter three, the motion estimation results on both tree sequences were more accurate than the results on the Yosemite sequence. This is due to the large displacements present in the Yosemite sequence. Velocity values close to 5 pixels/frame are difficult to estimate with motion estimation techniques that rely on finite differencing and small displacement assumptions. Thus, if one of the methods is able to successfully estimate large displacements and the other does not, then when agreement is maximized, the final

estimated field will not be close to the true unknown velocities because one of the methods is not able to estimate the correct answer. The performance of the techniques has to be similar. When the true displacement field is within the bounds of what the techniques can estimate, then there is a combination of motion parameters that make the algorithms estimate the unknown motion field sufficiently well.

The second assumption is perhaps the most important one. The idea of the proposed method relies on maximizing agreement between the techniques. Some disagreement is unavoidable, but what it is important is to minimize the amount by which the methods disagree. So when the methods converge to the same or almost the same answer, then the estimated field is close to the true velocity field.

4.4 No-Reference Global Optimization

Without loss of generality, the techniques of Horn and Schunk [1] and Lucas and Kande [6] are used to test and develop the no-reference approach. Note that from section 4.3 the only requirement is to use methods capable of independently estimating the true displacement field.

When a set of motion parameters is used with each technique, the results are compared in order to measure their agreement. One way to compare the results and decide whether or not they agree is to apply the Bland and Altman [55, 56] statistical measures of analysis. Figure 4.1 shows a block diagram of the no-reference global optimization approach.
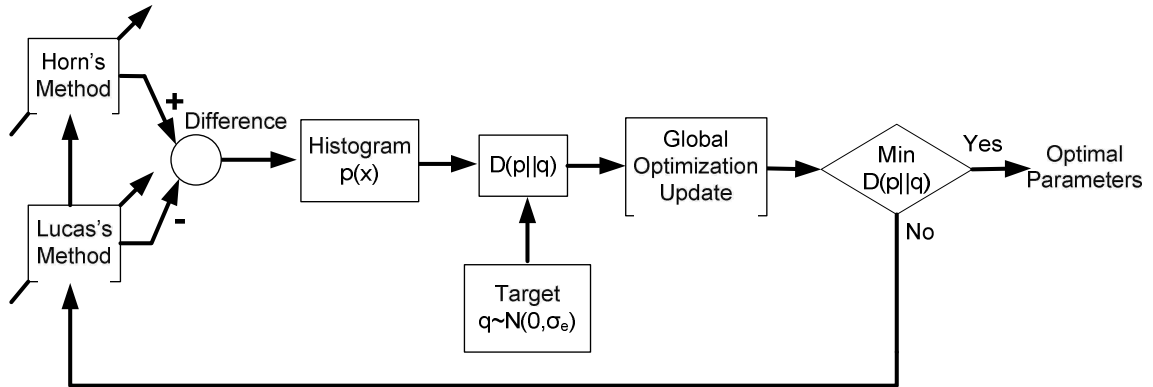
Figure 4.1. Diagram block of the no-reference global optimization approach.

4.4.1 Measuring agreement between methods

Agreement is measured [55, 56] by quantifying the variation in between-method differences. If two motion estimation techniques produce exactly the same results, then when the velocity estimates are plotted against each other, all points must lie in the line of equality. Figure 4.2 shows an example of perfect estimation.

(a)   (b)
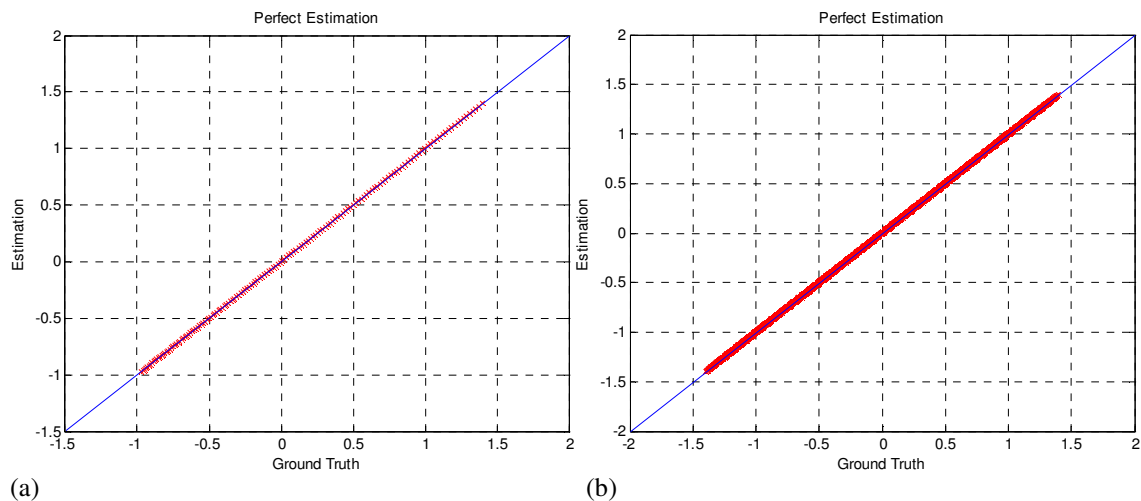
Figure 4.2 Equality plots for both motion components. Perfect velocity estimation of the (a) horizontal and (b) vertical velocities. Note that all points line on the equality line, i.e. y = x.

The second tool used to measure agreement in the estimation is the plot of the difference of the methods against their mean. This plot allows us to quantify the magnitude of the disagreement, spot outliers, and indentify if there is any bias. The bias is given by the

mean of the difference, when the centroid of the difference is away from zero. For example if bias = 1 pixel/frame, then one method could be estimating a value of 0.5 pixels/frame and the other 1.5 pixels/frame. Figure 4.3 shows this plot when perfect velocity estimation is achieved.

The plot of the estimates of one method against the other when there is disagreement will not have all the points lying on the identity or equality line. An example of dispersion can be seen in figure 4.4. Figure 4.5 shows the plot of the difference against the mean of the estimates. Here, note that there are several outliers that are more than two standard deviations away from the mean.
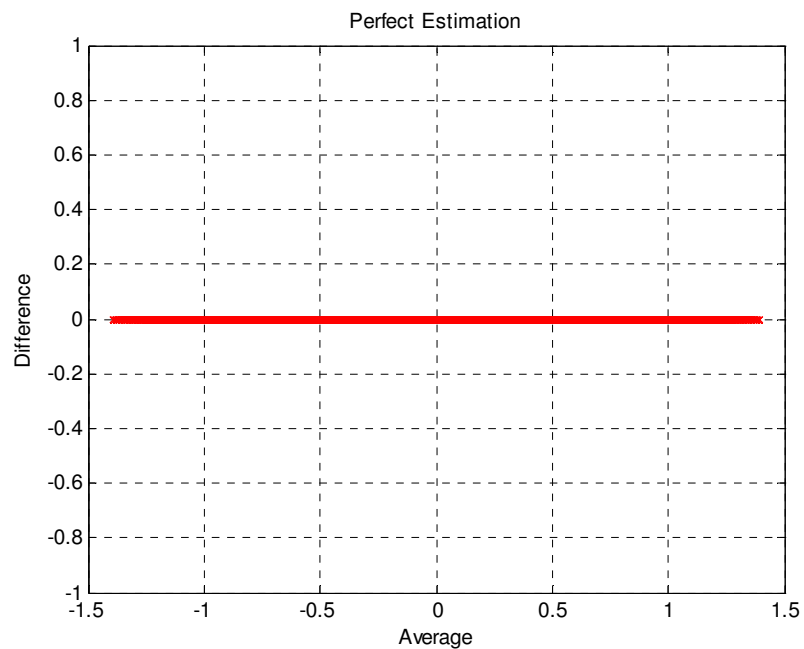
Figure 4.3 Difference against the average of the estimation. Case of perfect velocity estimation. The difference equals zero, no bias, and the average of the estimates can be considered as the best estimate of the actual velocity values.
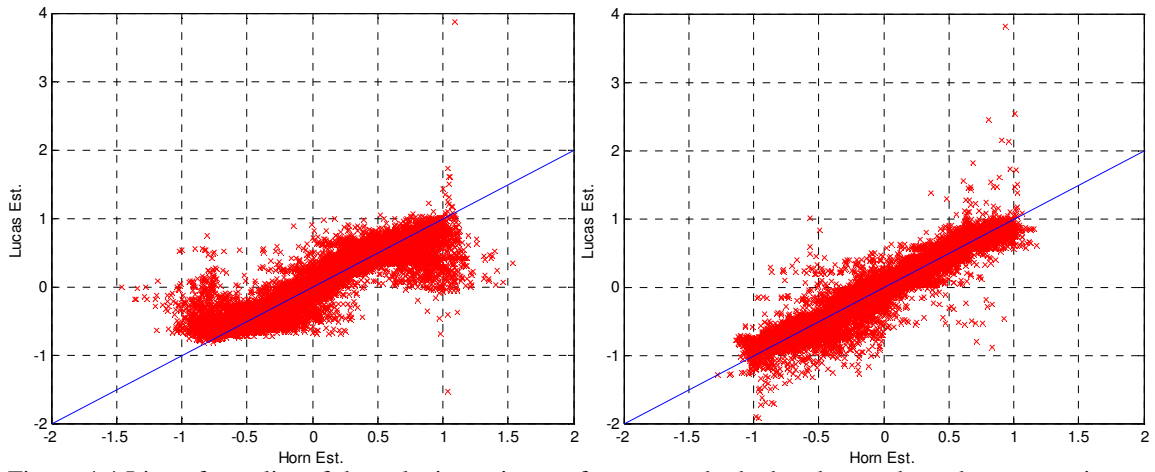
Figure 4.4 Line of equality of the velocity estimates for two methods that do not share the same estimates. (a) Horizontal motion component. (b) Vertical motion component.
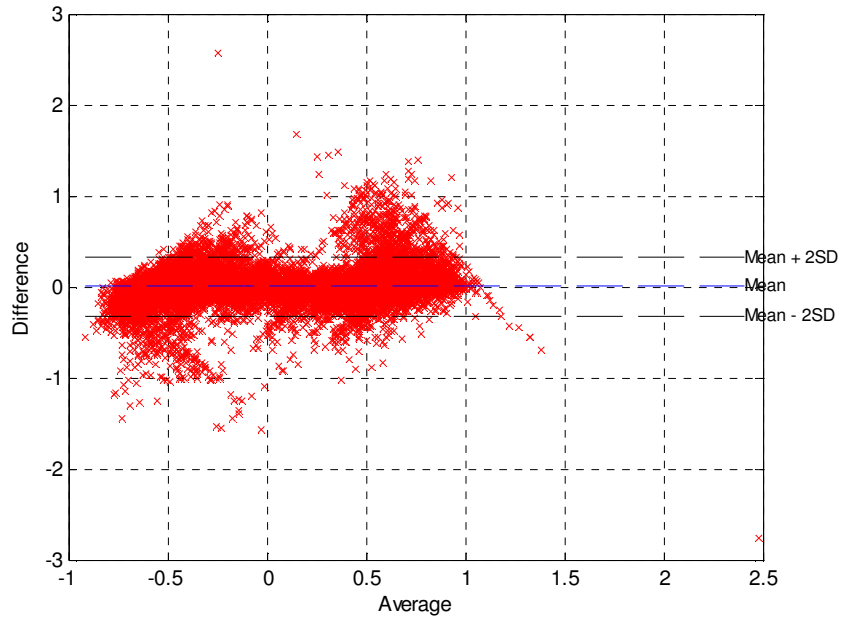
(a)
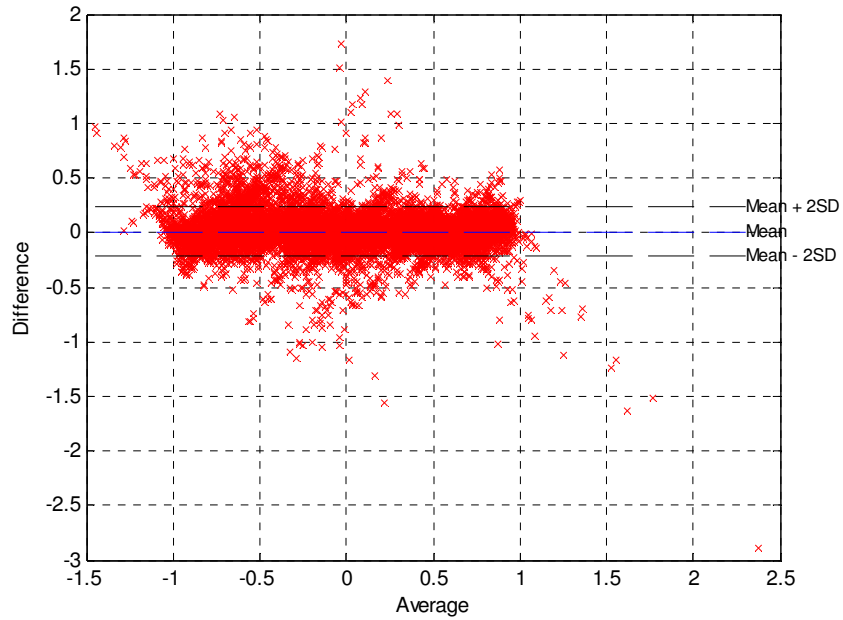
(b)

Figure 4.5 Difference against the average of the estimation. Outliers are points that lie more than two standard deviations away from the mean of the difference. (a) Horizontal motion component. (b) Vertical motion component.

The objective of the no-reference global optimization approach is to make most of the points lie on the line of equality, and make difference points lie no more than two standard deviations away from their mean.

4.4.2 Maximizing agreement in the estimation

Here, we note that agreement or convergence of the estimation is not equivalent to calculating the true displacement field nor should it be understood that the estimated field is the same as ground truth motion. The no-reference approach only seeks to find a good estimate of the motion using the results of two independent algorithms that have been optimized to produce accurate estimates as long as the assumptions of section 4.3 are met. It is important to stress that in the absence of ground truth motion, optimizing for the agreement of two independent methods is most likely to produce accurate velocity estimates. Thus, the no-reference approach provides a good alternative to the current state of the art of heuristically choosing the motion parameters of each technique.

Maximizing the agreement in the estimation is equivalent to minimize the disagreement. Therefore, the disagreement of the methods, measured by the difference in the estimation, can be treated as the error that needs to be minimized. In order to minimize the error, the difference will be assumed to follow a zero mean normal distribution. The amount by which the methods disagree can be controlled by fixing the standard deviation of the distribution. This value is the approximate precision of the estimation. Since about 95% of the values drawn from a normal distribution are within two standard deviations away from the mean, the outliers or pixels for which agreement on the estimation was not

possible can be easily spotted. Two standard deviations away from the mean is the value set for the limits of agreement.

The global optimization algorithm of section 3.3 will be applied to minimize the relative entropy, $D(p\|q)$, between two discrete probability mass functions $p(x)$ and $q(x)$, and it is defined as:

$$D(p\|q) = \sum_{x \in X} p(x) \log \frac{p(x)}{q(x)}. \tag{4.1}$$

The relative entropy measures the inefficiency of assuming that the true distribution is $q(x)$ when the true distribution is $p(x)$ [57]. The relative entropy is always non-negative and is zero if and only if $p(x) = q(x)$ [57].

The true continuous distribution of the error, $p(x)$, is approximated by the histogram of the measured differences. The discrete target error distribution $q(x)$ is generated by drawing numbers with $q(x) \sim N(0, 0.02)$. The normal distribution has to be zero mean in order to remove any bias in the estimation. Also, the value of the distribution standard deviation controls the accuracy of the estimation. Keeping it to a small value guarantees that large velocity estimation errors will be avoided. Thus, the selected value of 0.02 pixels per frame for σ_n will help in the calculation of very accurate results.

The optimization returns the set of motion estimation parameters that generate an error with distribution $p(x)$ closest to $q(x)$. Here, note that the histogram approach can be easily modified to accommodate any other distribution of the error.

4.4.3 Parameter space for no-reference optimization

The motion estimation parameter space is set to be the same as the reference optimization parameter space for sequences with ground truth, see table 3.1 for Horn's method and table 3.9 for Lucas's technique. For real video sequences, the range of values is given in table 4.1 and

table 4.2. The values of the regularization parameter are selected to cover values used in the literature for ground truth sequences. The spread of the pre-smoothing filter for both techniques is set so that the temporal support of the frames is not exceeded.

Table 4.1. Motion estimation parameter space for Horn's method.

	Taxi Sequence		Rubik's Cube Sequence	
	a	b	a	b
α	0.1	0.5	0.1	0.5
σ	0.5	2.25	0.5	2.5

Table 4.2. Motion estimation parameter space. Lucas' method.

	Taxi Sequence		Rubik's Cube Sequence	
	a	b	a	b
σ	0.25	2.25	0.25	2.25

4.4.4 Confidence maps

The mean and standard deviation of the measured difference probability are used to produce confidence maps of the velocity estimates. The limits of agreement between the estimations are set to two standard deviations away from their mean. Thus, a map of unreliable velocity estimates can be generated by locating all the points that lie outside these limits.

The limits of agreement for the target distribution have values of +/-0.04 pixels per frame. Realistic values expected in the measured distribution of the error would be

ranging from 0.1 to 0.15 pixels per frame. This means that if the measured error perfectly follows a normal distribution, then 95% of the estimates are within the limits of agreement. Differences of +/-0.1 pixels per frame in the estimation translate to very accurate velocity values.

4.5 Results

The methodology is first tested on videos with ground truth. The results obtained with the "blind parameters" are compared against the ones found in section 3.4. Then, results for natural video sequences will be computed.

4.5.1 Natural video sequences

Two standard natural video sequences were used. Even though the true displacement field remains unknown, general guidelines for these sequences were found in [27]. The description of each one is as follows:

- **Rotating Rubik Cube:** This video shows a counter-clockwise rotating Rubik's cub on a turntable. The cube is rotating with velocities less than 2 pixels/frame while velocities of the turntable range from 1.2 to 1.4 pixels/frame.

- **Hamburg Taxi Sequence:** This is a street video sequence that shows four moving objects: a taxi turning the corner, a car in the lower left (driving from left to right), a van in the lower right (driving from right to left), and a pedestrian in the upper left. Image speeds are approximately 1.0, 3.0, 3.0 and 0.3 pixel/frame respectively.

94

4.5.2 Results on synthetic videos

Results for the three test sequences are compared with those obtained with the global optimization algorithm of section 3.4.1. Additionally, the error histograms along with the equality and difference against average plots will be provided. The "blind parameters" and their error metrics are presented in table 4.3 for Horn's technique, and table 4.4 for Lucas's method.

Table 4.3. Blind validation results for Horn's method on the three test video sequence.

	Div. Tree	Trans. Tree	Yosemite w/o clouds	Yosemite
σ	1.44	1.38	1.25	1.25
α	0.78	0.1	0.88	0.88
MSE	0.0207	0.0148	0.1598	1.61
μ_{AE}	4.76°	1.67°	5.16°	11.77°
δ_{AE}	2.58°	2.14°	6.87°	16.4°
Density	99.7%	99.7%	77.1%	99.7%

Table 4.4. Blind validation results for Lucas's method on the three test video sequence.

	Div. Tree	Trans. Tree	Yosemite w/o clouds	Yosemite
σ	1.56	1.65	1.25	1.25
MSE	0.0132	0.0019	0.11	1.16
μ_{AE}	4.4°	0.72°	4.11°	9.0°
δ_{AE}	1.9°	0.82°	5.98°	15.4°
Density	83.8%	76.3%	63.6%	74.4%

(a)   (b)

(c)   (d)

Figure 4.6. Equality plots of the estimated velocities against ground truth on horizontally Translating Tree video sequence. (a) Horizontal motion component for Horn's method. (b) Vertical motion component for Horn's method. (c) Horizontal motion component for Lucas's method. (d) Vertical motion component for Lucas's method.

The equality plots of the estimated velocities on the Translating Tree sequence against the ground true velocities are shown in figure 4.6. These plots are provided as a reference to evaluate how well the estimated velocities agree with the true displacement field. Note that despite the evident dispersion for some velocity values, the overall estimated velocities follow the equality line. Large deviations are observed for the estimation of the vertical velocity component. To see if this artifact affects only the velocities produced

with the blind parameters, the equality plots against ground truth of the optimal velocities (the ones computed with the optimal parameters of section 3.4.4) are also provided in figure 4.7. The same artifact is present in these velocities which can be attributed to a limitation of the motion estimation techniques rather than the velocities produces with the blind parameters. Finally, equality plots of the estimated velocities against each method are presented in figure 4.8.

Figure 4.9 shows the difference against the average of the estimation plot. The limits of agreement include estimations for which their difference is within two standard deviations away from the mean. These are convenient limits since approximately 95% of the samples drawn from a normal distribution fall inside these limits. Estimations outside this range are considered outliers or pixels for which the confidence in the motion estimation is low. Outliers account for 6.7% of the horizontal velocities, and 8.3% of the vertical velocities.

The histogram of the total difference in estimation, from the difference of each velocity component a single one is created, is shown in figure 4.10. This difference is produced with the blind parameters, and it is the closest distribution to the theoretical normal used for the optimization.

Equality plots of the estimated velocities for the Divergence Tree sequence are shown in figure 4.11. Most of the velocity estimates lie on the equality line, showing good agreement between the estimates of both methods. To measure how close the estimated velocities are to the ground truth velocities, equality plots against the true displacement field are provided in figure 4.12. These plots show a tendency of the estimated velocities

97

to underestimate the actual value. This artifact especially affects pixels with larger

velocity values and shows on both techniques as well as velocity components. As in the

(a)

(b)

(c)

(d)
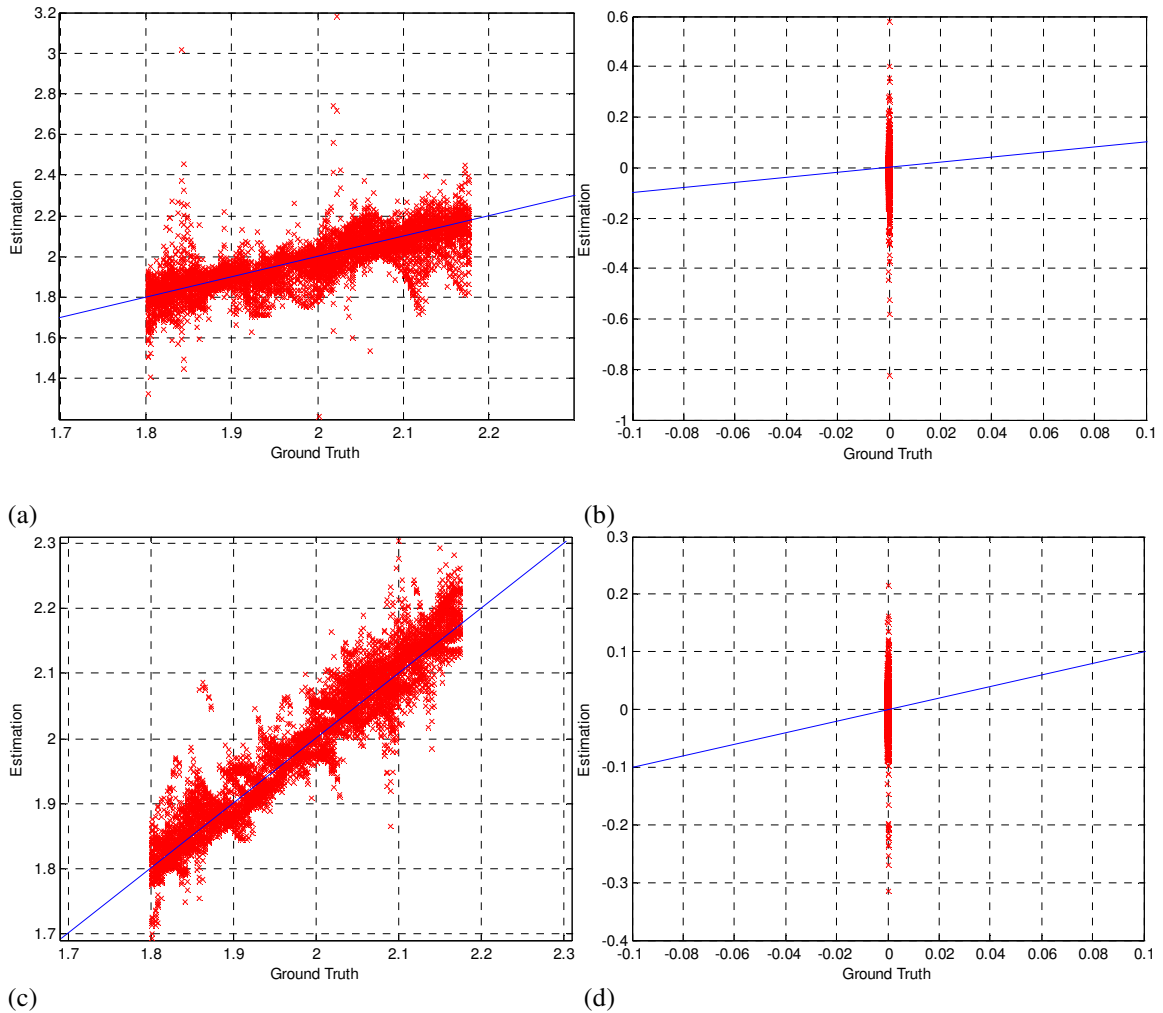
Figure 4.7. Equality plots of the optimal velocities against ground truth on horizontally Translating Tree video sequence. (a) Horizontal motion component for Horn's method. (b) Vertical motion component for Horn's method. (c) Horizontal motion component for Lucas's method. (d) Vertical motion component for Lucas's method.

the case of the Translating Tree video, equality plots of the optimal velocities against the

true displacement field are provided to identify whether this is due to some limitation of

the optical flow techniques or to not finding the optimal parameters.

98
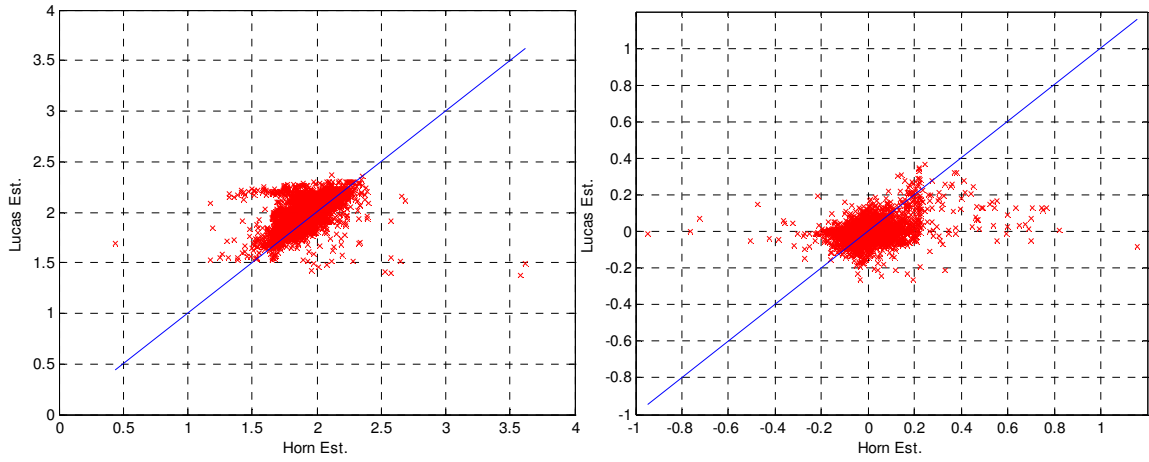
(a)   (b)

Figure 4.8. Equality plots of the estimated velocities on horizontally Translating Tree video sequence. (a) Horizontal velocity component. (b) Vertical velocity component.
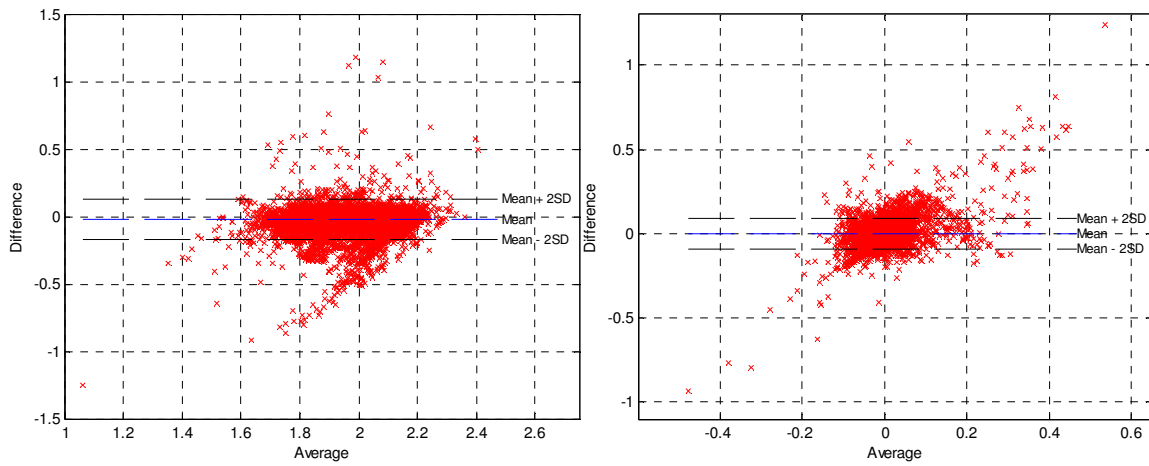
(a)   (b)

Figure 4.9. Difference-average plot of the estimated velocities on Translating Tree video sequence for the (a) horizontal velocity component, and (b) vertical velocity component.
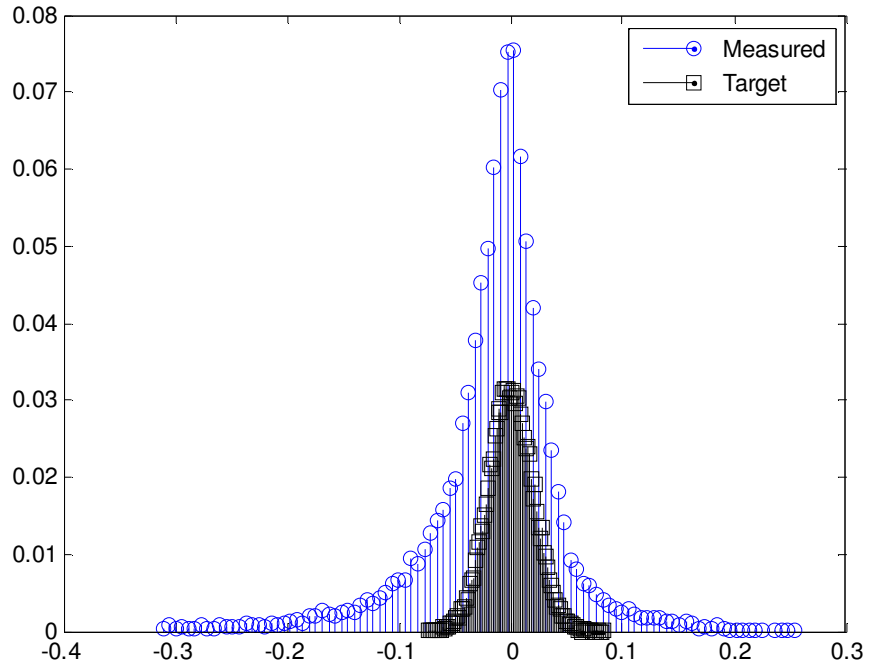
Figure 4.10. Histogram of the total difference in estimation for the horizontally Translating Tree sequence. The target distribution is set to lower errors than the measured ones. The measured distribution is the closest distribution to the target from all the distributions calculated with the no-reference optimization algorithm.

(a)   (b)

Figure 4.11. Equality plots of the estimated velocities for the Divergence Tree video sequence. (a) Horizontal velocity component. (b) Vertical velocity component.

Figure 4.13 shows that the underestimation problem is also present in the velocities produced with the globally optimal parameters of section 3.4.4. The difference between the estimates by the two methods against their average is shown in figure 4.14 for both velocity components. The limits of agreement include 92.2% of the horizontal velocities and 91.8% of the vertical velocities. The distribution of the total difference of the estimates is shown in figure 4.15 along with the target one. The true distribution is produced with the blind parameters.

The equality plots of the velocity estimates for the Yosemite sequence are shown in figure 4.16. The estimates lie on the equality line, but the estimated values are well outside what motion estimation techniques can calculate. The large displacements present in this sequence make accurate estimation almost impossible. This is an example of poor estimation but not because the no-reference strategy fails to produce accurate results. The failure is on the motion estimation techniques themselves because they are not able to estimate the velocity field. For this particular example, the first assumption of the no-reference strategy does not hold.

Figure 4.17 shows the lack of accuracy in the estimation. Most of the errors are concentrated for the velocities of the cloud regions (big spikes in the equality line). The non-conservation of the image intensities really drives down the performance of the motion estimation algorithms. Large dispersion is also noticeable for velocity values bigger than two pixels per frame. Here, note that the motion estimation techniques assume small displacements along the motion path. The difference versus average plot, figure 4.18, shows the limits of agreement of the estimated velocities.
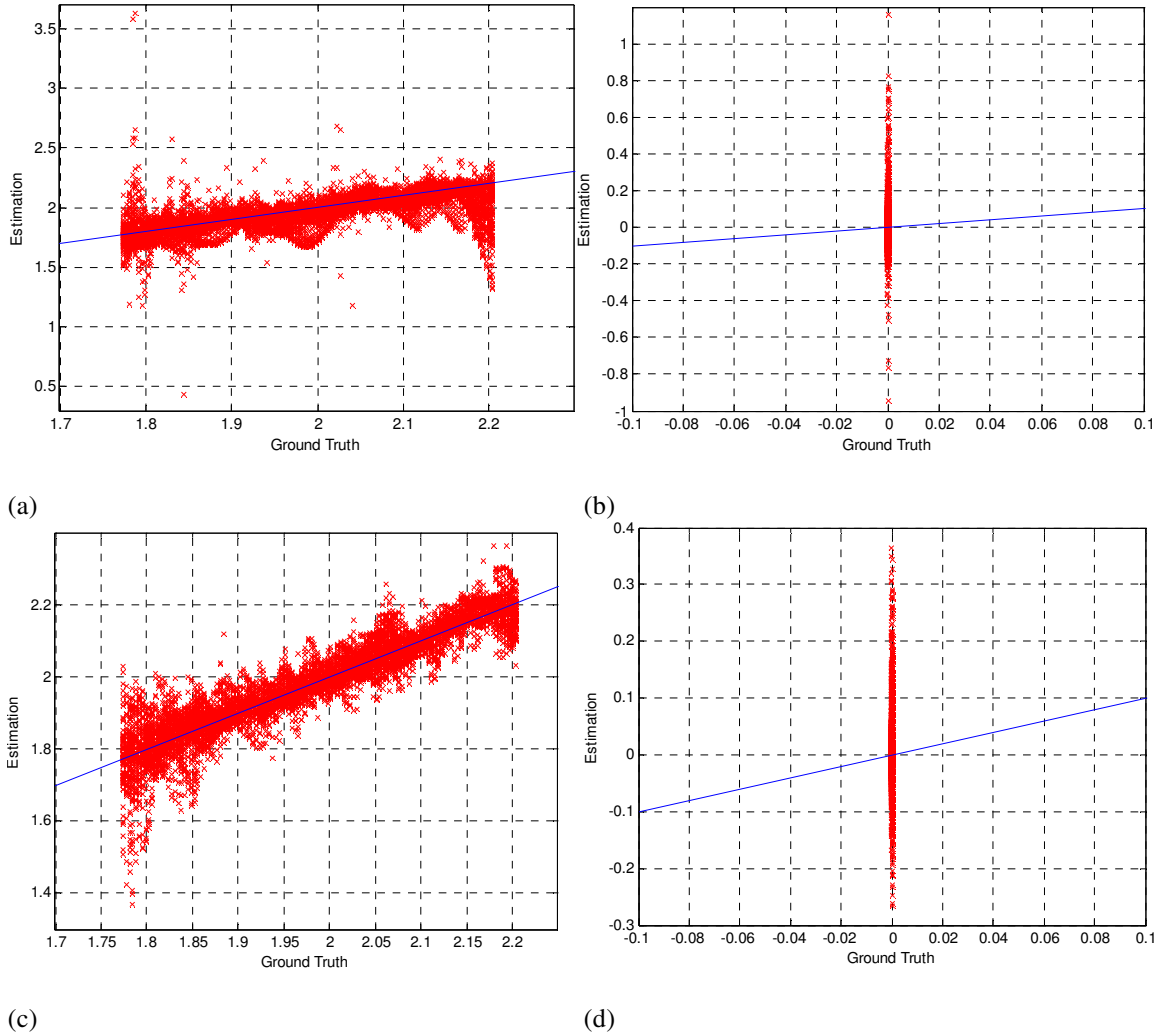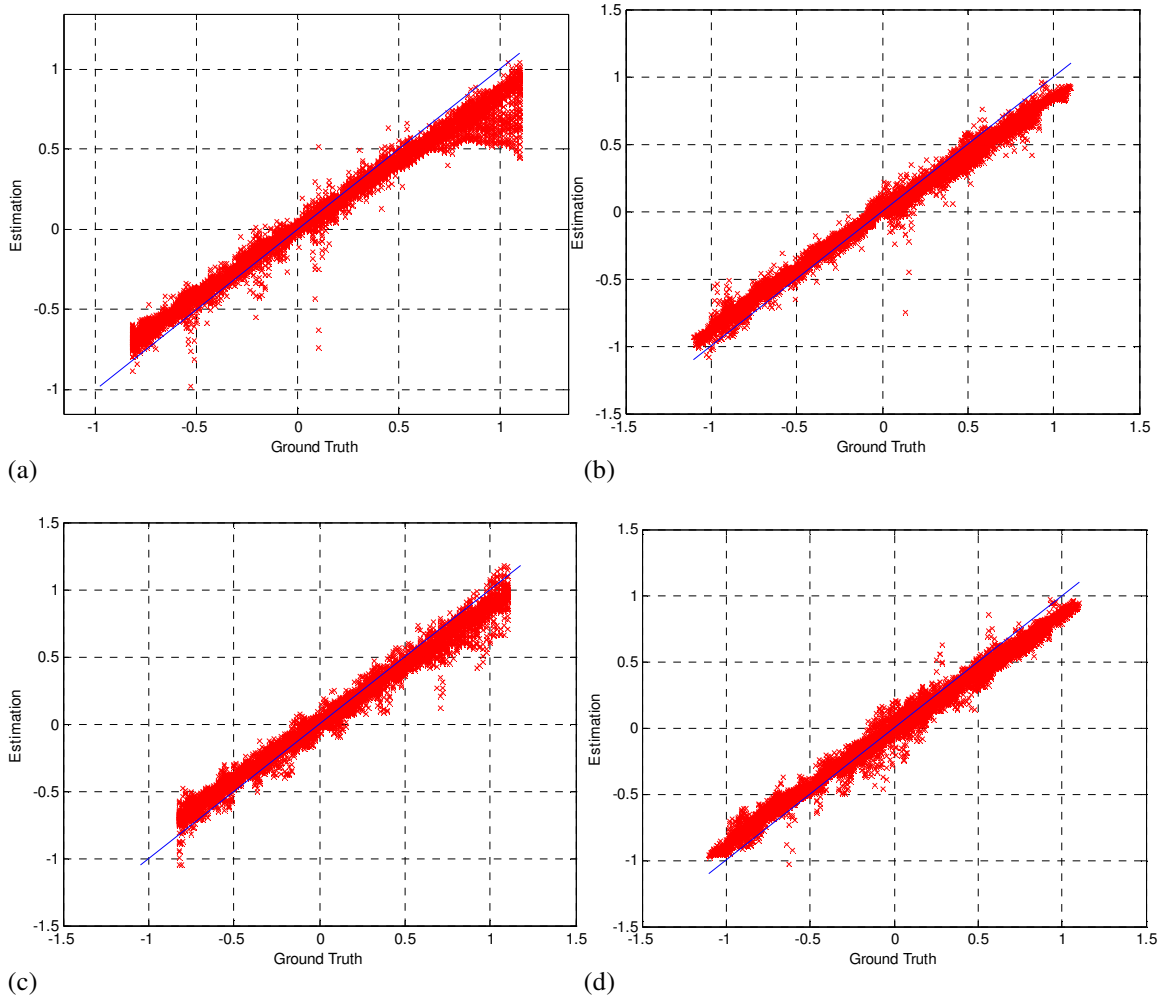
Figure 4.12. Equality plots of the estimated velocities against ground truth on Divergence Tree video sequence. (a) Horizontal motion component for Horn's method. (b) Vertical motion component for Horn's method. (c) Horizontal motion component for Lucas's method. (d) Vertical motion component for Lucas's method.

For this particular case, the information conveyed by the graph is not very informative due to the poor performance of the motion estimation techniques. The values of the limits of agreement do provide some information regarding the quality of the estimates. The particular high values for these limits are an indication that unreliable velocities are estimated.
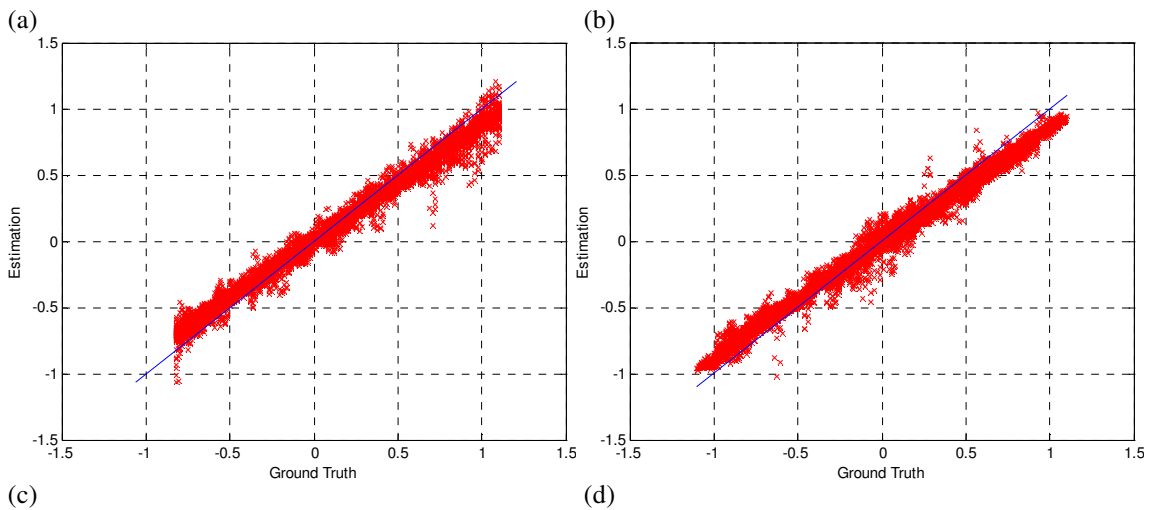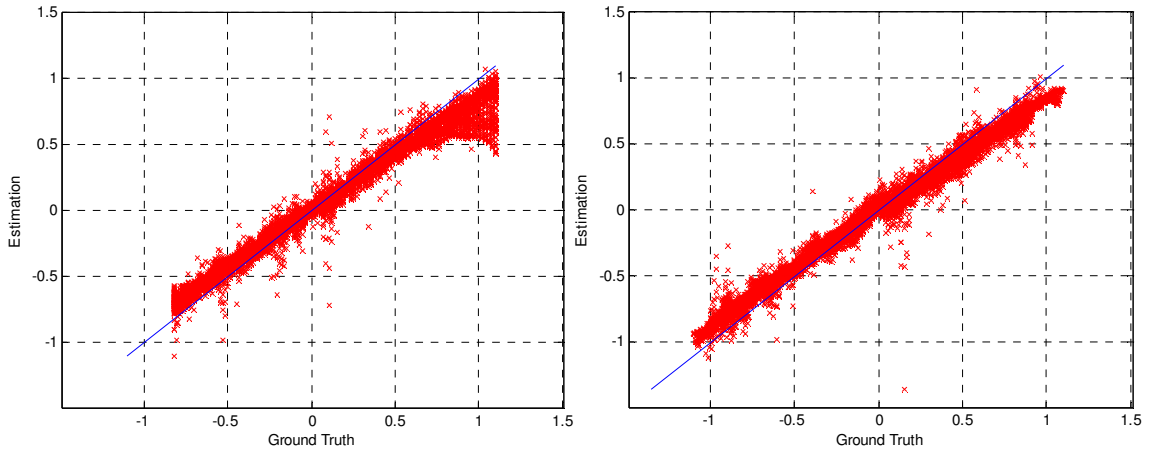
Figure 4.13. Equality plots of the optimal velocities against ground truth on Translating Tree sequence. (a) Horizontal motion component for Horn's method. (b) Vertical motion component for Horn's method. (c) Horizontal motion component for Lucas's method. (d) Vertical motion component for Lucas's method.
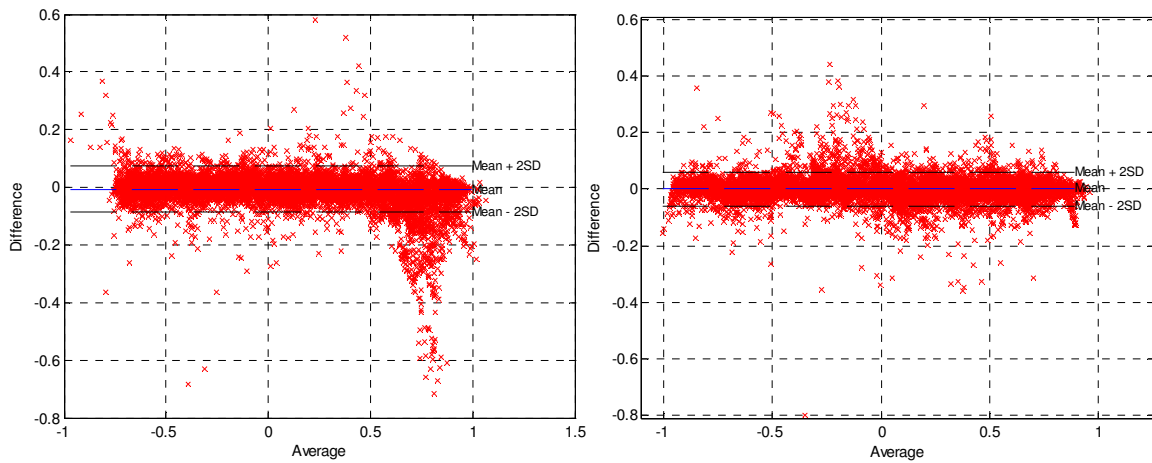
Figure 4.14. Difference-average plot of the estimated velocities on the Divergence Tree sequence for the (a) horizontal velocity component, and (b) vertical velocity component.
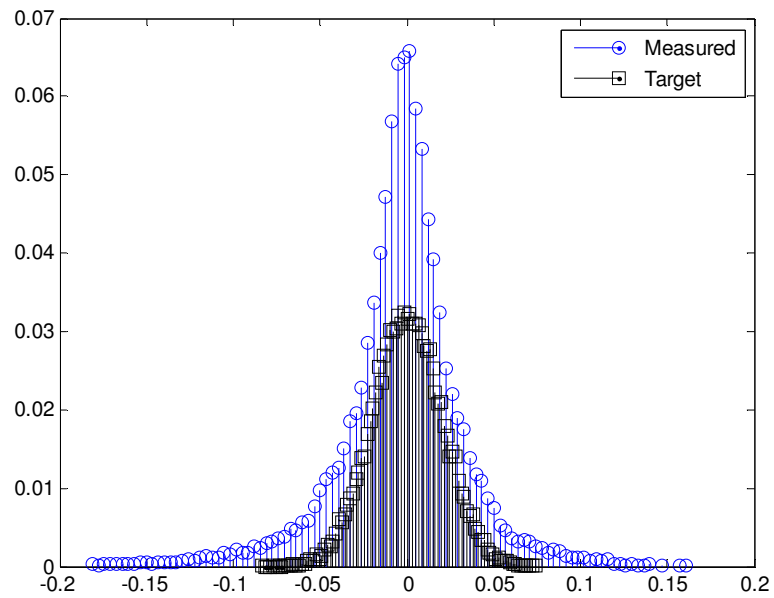
Figure 4.15. Histogram of the total difference in estimation for the Divergence Tree sequence. The smaller distribution is the theoretical difference distribution. The observed or true difference distribution is the bigger one, and is the closest distribution to the theoretical from all the distributions calculated with the blind optimization algorithm.
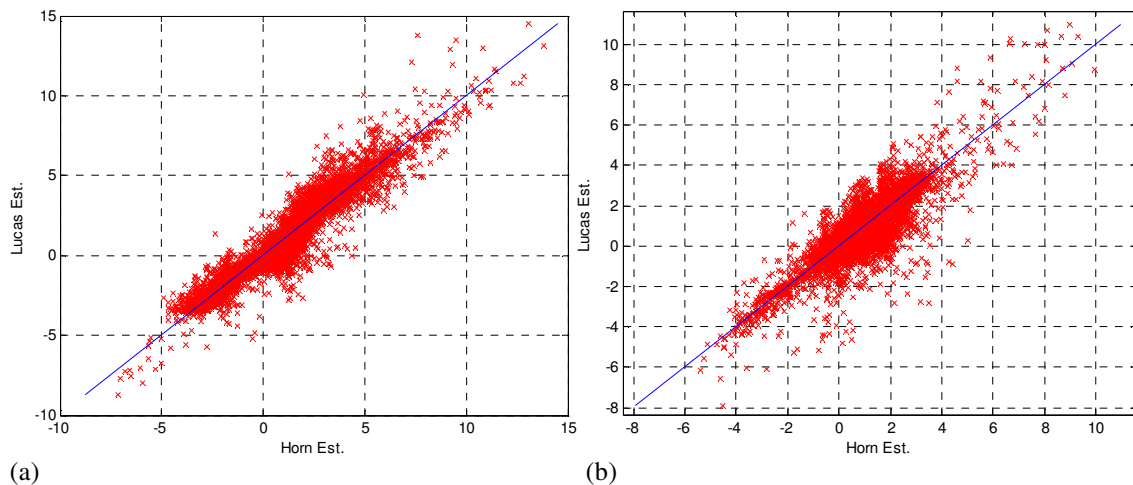
(a)   (b)

Figure 4.16. Equality plots of the estimated velocities for the Yosemite sequence. (a) Horizontal velocity component. (b) Vertical velocity component.
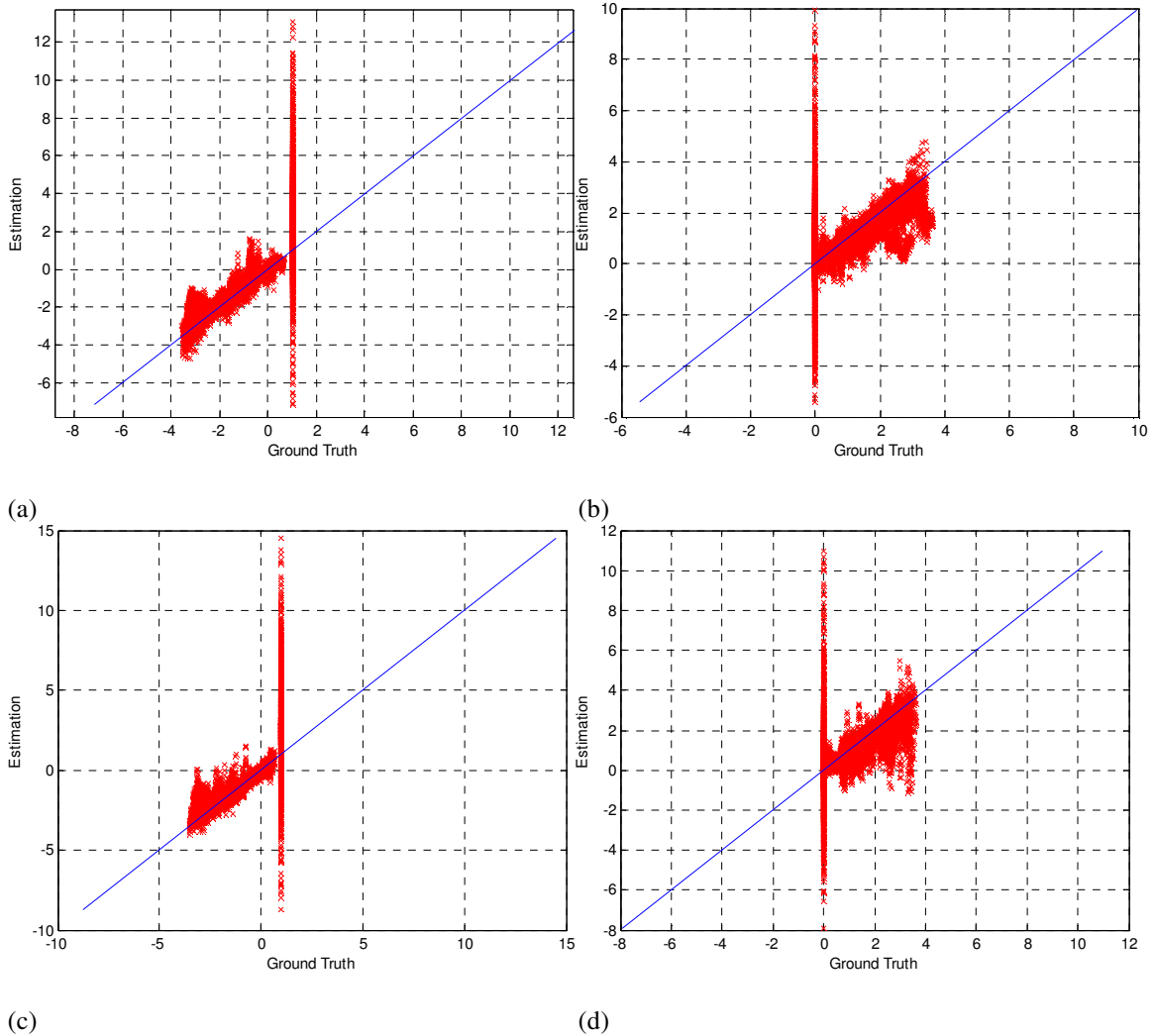
Figure 4.17. Equality plots of the estimated velocities against ground truth for the Yosemite sequence. (a) Horizontal motion component for Horn's method. (b) Vertical motion component for Horn's method. (c) Horizontal motion component for Lucas's method. (d) Vertical motion component for Lucas's method.

To provide representative results on the Yosemite sequence, the cloud region is removed from the no-reference optimization. This region is where most of the error is located because the image intensities are not constant during the motion path. The parameters found with the no-reference approach produce an error that is almost the same as the one found with the global optimization algorithm of section 3.4.1 when the cloud region is removed. The equality plots, figure 4.19, of the estimation against the ground truth velocities still reveal errors for the largest velocity values.
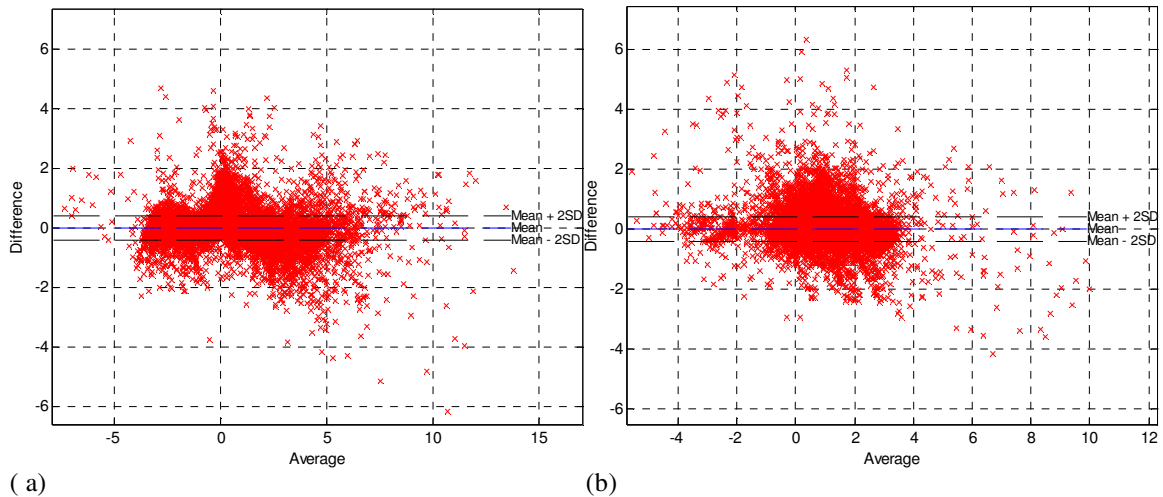
Figure 4.18. Difference-average plot of the estimated velocities for the Yosemite sequence. (a) Horizontal velocity component, and (b) vertical velocity component.

The equality plots of the estimation between the two methods, figure 4.20, show good agreement with most of the points on the equality line despite the evident errors due to the large displacements. Smaller values but still above optimal ones of the limits of agreements can be seen in figure 4.21 and figure 4.22. The ideal values would be between 0.1 and 0.15 like those of the tree sequences. The limits are important features to spot outliers or unreliable velocity estimates. In this case, given the challenging nature of the motion, it is expected that locations that undergo large displacements will lie outside the limits of agreement or even within them due to their big values, +/- 0.2 and +/-0.25 pixels per frame. Thus, instead of using two standard deviations away from the mean, only locations with differences smaller than one standard deviation away from the mean can be considered as reliable estimates. Using this information, a map of the locations outside these limits can be generated. Figure 4.23 shows the confidence map of the estimates for the Yosemite sequence without the clouds. Locations with high disagreement are

considered unreliable and they are located where the large displacements occur along with the cloud region which was removed from the estimation.
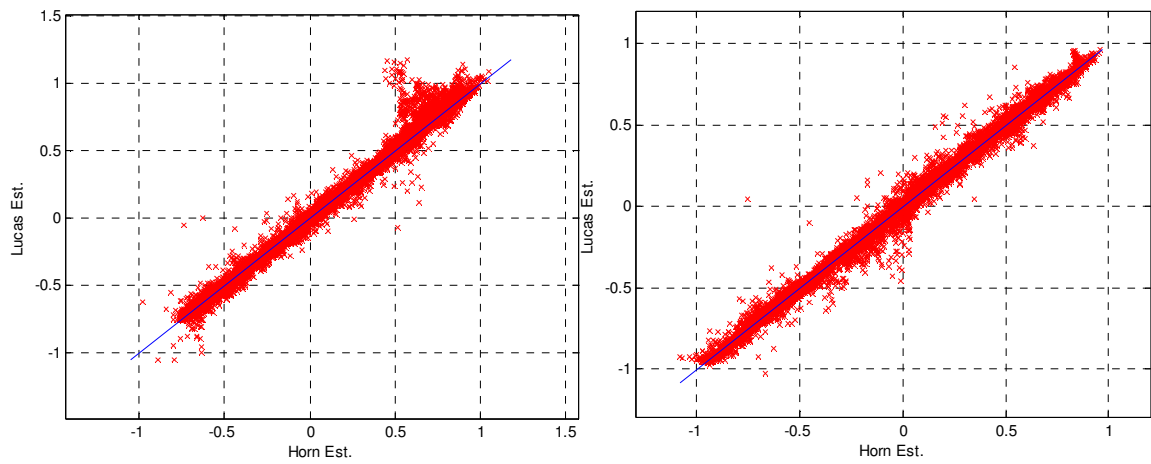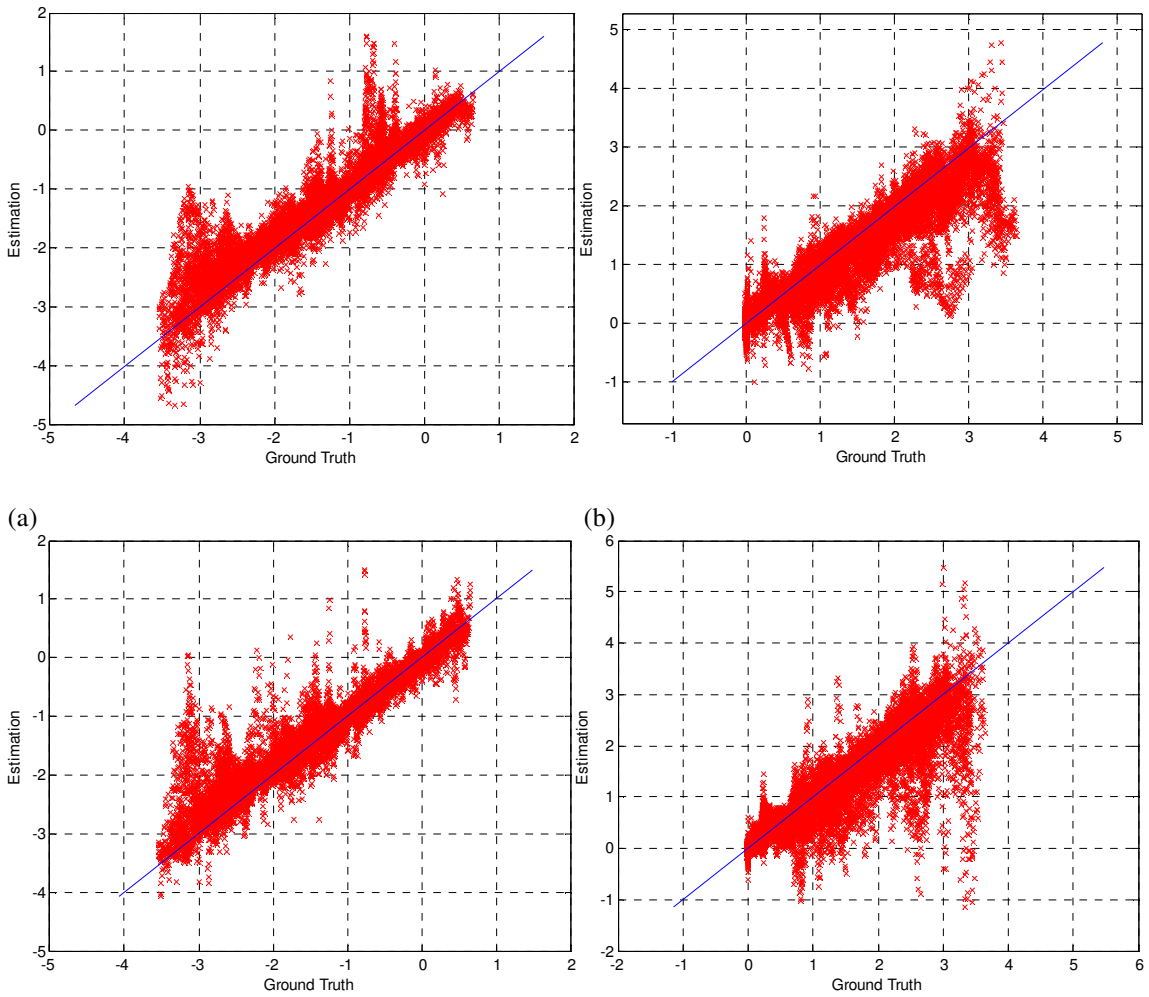

Figure 4.19. Equality plots of the estimated velocities against ground truth for the Yosemite sequence without the cloud region. (a) Horizontal motion component for Horn's method. (b) Vertical motion component for Horn's method. (c) Horizontal motion component for Lucas's method. (d) Vertical motion component for Lucas's method.

(a)   (b)
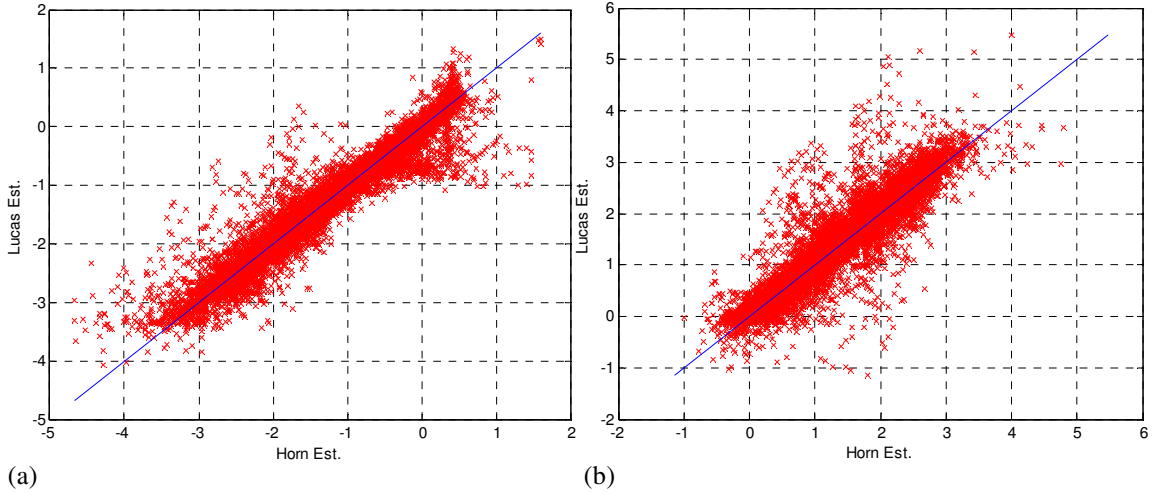
Figure 4.20. Equality plots of the estimated velocities for the Yosemite sequence without clouds. (a) Horizontal velocity component. (b) Vertical velocity component.
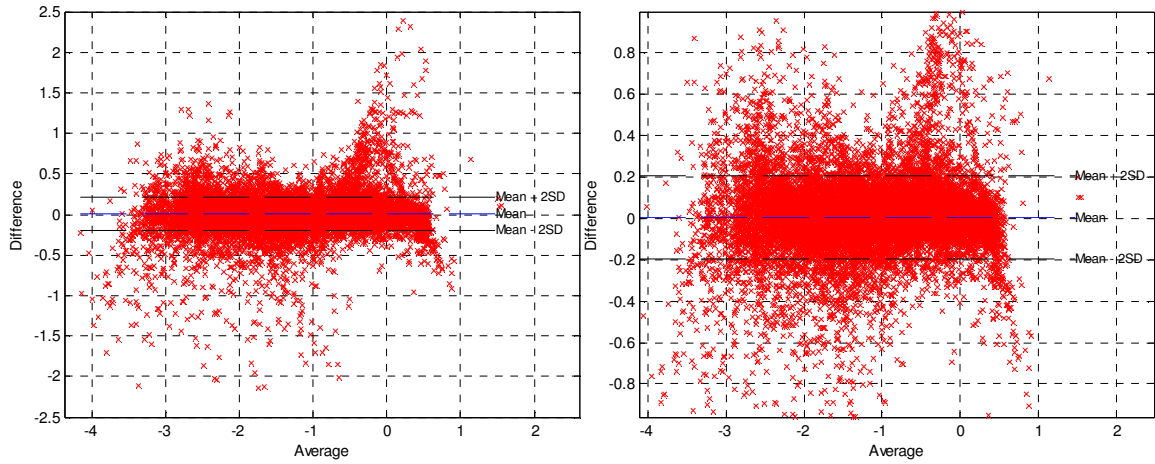
Figure 4.21. Difference-Average plot of the estimated horizontal velocities for the Yosemite sequence without the cloud region. (b) The value of the limits of agreements is approximately +/-0.2 pixels/frame which is still high.
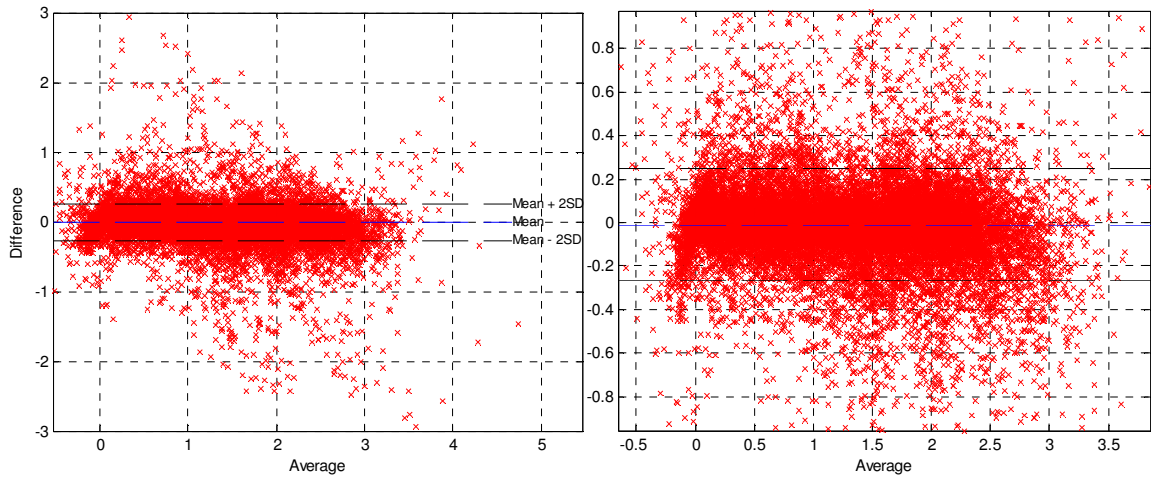
Figure 4.22. Difference-Average plot of the estimated vertical velocities for the Yosemite sequence without the cloud region. (b) The value of the limits of agreements is approximately +/-0.25 which is still high.
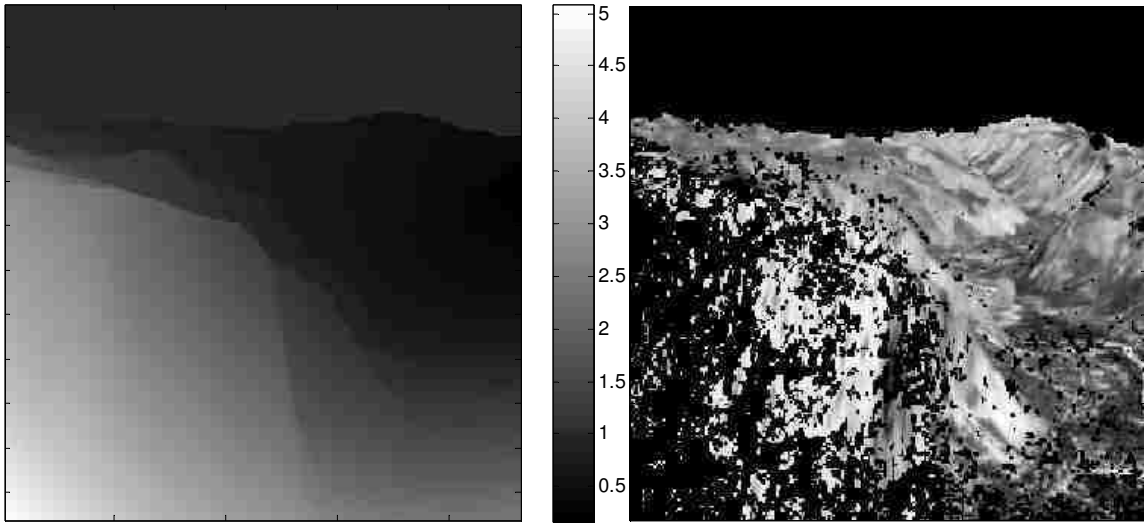
(a)   (b)

Figure 4.23. Confidence map of the velocity estimates (a) Magnitude of the ground truth velocity field. (b) Dark regions are locations where the difference in the estimation by the two methods is bigger than one standard deviation away from the mean. Notice that most of these areas have velocity magnitudes bigger than 2.5 pixels/ frame in the bottom left part of the image and the cloud region which was removed from the estimation.

(a)   (b)
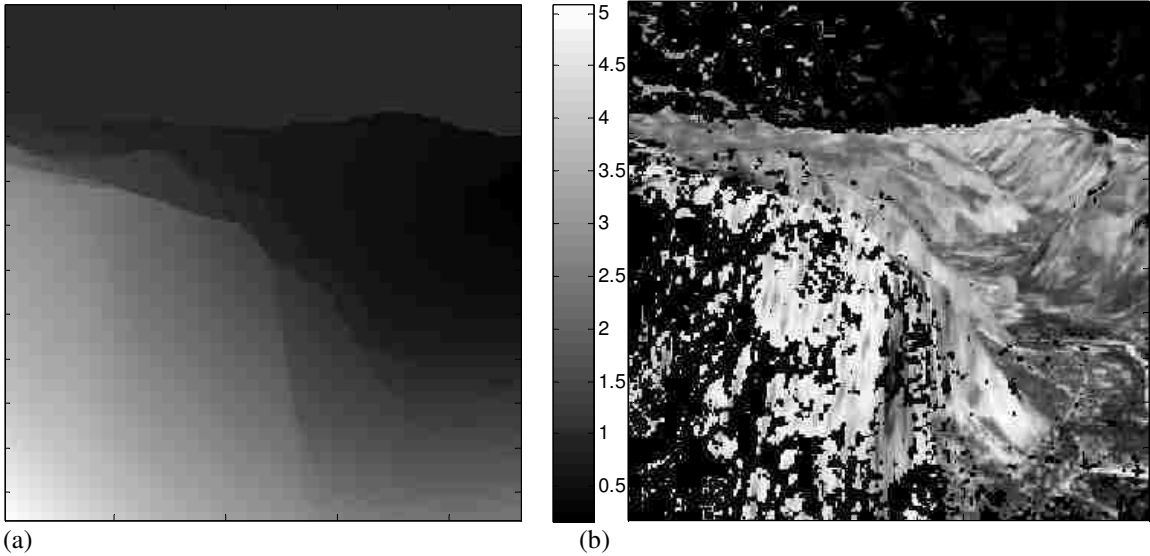
Figure 4.24. Confidence map of the velocity estimates including the cloud region(a) Magnitude of the ground truth velocity field. (b) Dark regions are locations where the difference in the estimation by the two methods is bigger than one standard deviation away from the mean. In this case the limits of agreement were +/-0.2 pixels per frame. Notice that most of these areas have correspond to the largest velocities and the cloud region.

A confidence map that includes velocity estimates of the cloud region is also presented in figure 4.24. Notice how regions with large displacements as well as and the top part of the image, cloud region, are identified as unreliable velocity estimates.

Table 4.5 shows the mean of the differences and their corresponding limits of agreement for the three test sequences. If the mean of the estimates is different than zero, then it means that there is some bias in the estimation. Small or nearly zero bias is the preferable result.

Table 4.5. Mean of the difference in estimation by the two method and the agreement limit values.

	Translating Tree	Divergence Tree	Yosemite w/o clouds	Yosemite
Diff. Mean U	-0.025	-0.006	0.0047	-0.0020
Upper Limit U	0.124	0.0722	0.2068	0.4066
Lower Limit U	-0.1734	-0.0847	-0.1975	-0.4106
Diff. Mean V	-0.002	0.0003	-0.0096	0.0044
Upper Limit V	0.0885	0.0619	0.2458	0.4144
Lower Limit V	-0.0926	-0.0612	-0.2650	-0.4056

4.5.3 Results on natural video sequences

Results for the real videos include equality plots and the difference against average plot between methods. Figure 4.25 shows the equality plots of the taxi sequence. Most of the points lie in this line, and it can be observed form figure 4.25(a) that most of the values of the horizontal motion component are concentrated between -1 to 1 pixels/frame, but there are clusters close to 3 and -3 pixels per frame. Values for the vertical velocity are concentrated between -1 to 1 pixels/frame.

The difference-average plots, figure 4.26 and figure 4.27, also reveal these values for the velocities. The limits of agreement include 91% of the estimated horizontal velocities and 91.8% of the estimated vertical velocities.

Equality plots for the Rubik's cube sequence, figure 4.28, show lack of agreement between both methods particularly in the horizontal component of the motion. There is underestimation for velocity value between 0.5 and 1.5 pixels per frame. The equality plot of the vertical velocities has most of the values lying on the equality line but still there are problems at the ends of the range of the estimated velocity values.

The plots of the difference against the average of the estimates (figure 4.29 and figure 4.30) reveal that there is more agreement in the estimation of the vertical velocities than the horizontal ones. Thus, estimation of one motion component is more reliable than the other since better agreement was found for the vertical velocities. Also, the scatter plot of figure 4.29 reveals that the error is not following a normal distribution and there is even some correlation between the magnitude of the differences and the value being tried to estimate. These findings re-enforce the notion that the estimation of one component of the motion is unreliable. A solution is to perform another run of the no-reference

approach but only for the horizontal component in order to find a different set of motion parameters that produce more accurate results. Additionally, note that the difference-average plot is not suppose to be a test for normality. Instead, the correct procedure is the use of a *q-q* plot. The difference-average plot is visualization tool that helps in the evaluation of the estimated velocities.

Finally, the mean of the differences and the values of the limits of agreement for the two natural video sequences are shown in table 4.6.
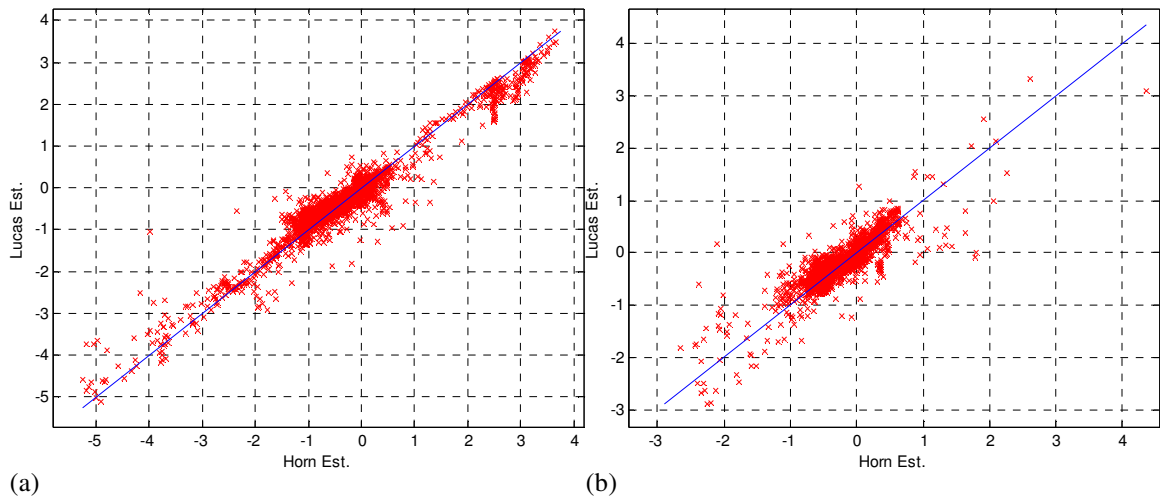
(a)   (b)

Figure 4.25. Equality plots of the estimated velocities for the Taxi sequence. (a) Horizontal velocity component. (b) Vertical velocity component.
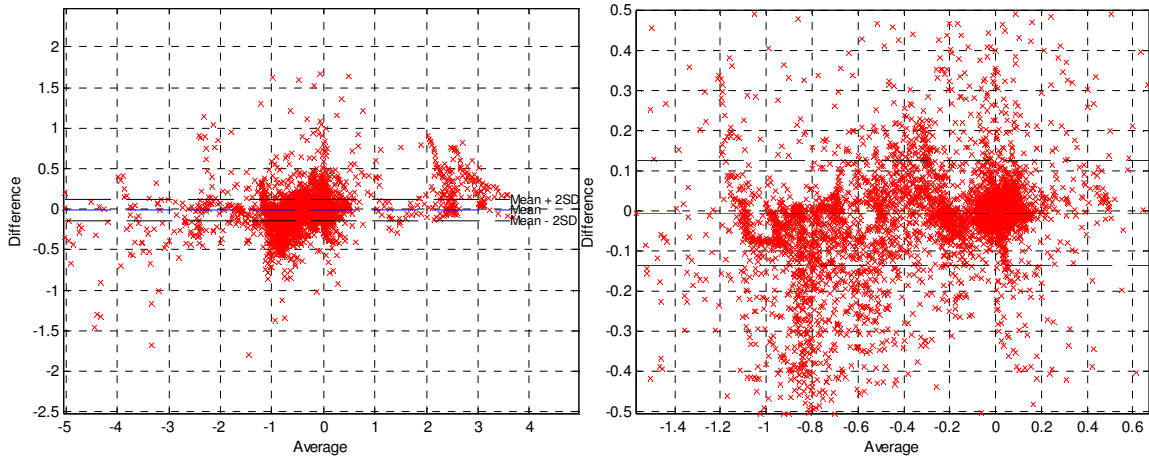
(a)   (b)

Figure 4.26. (a) Difference in horizontal velocity estimation by the two methods against their average for the Taxi sequence. (b) Closer look at the middle cluster reveals that most of the values lie within the limits of agreement.
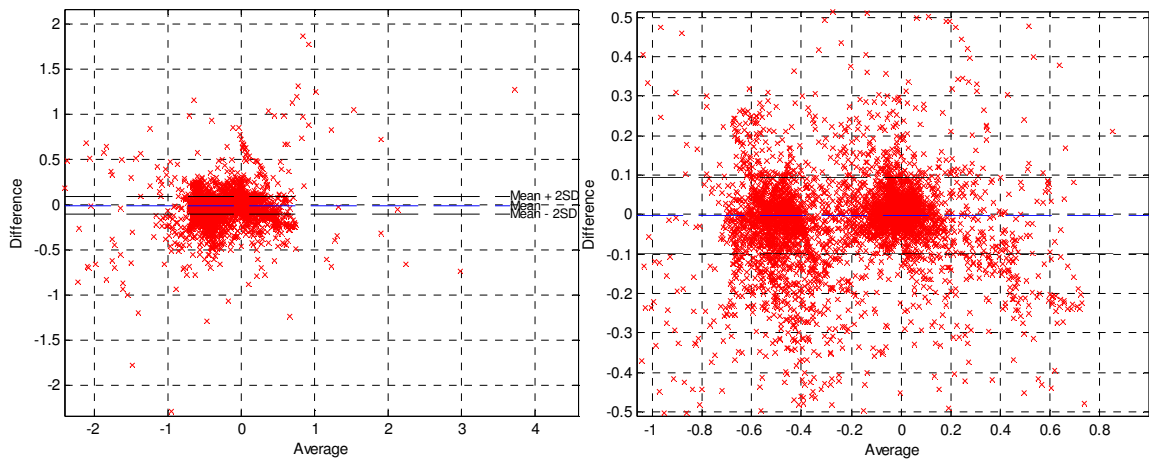
(a)   (b)

Figure 4.27. (a) Difference in vertical velocity estimation by the two methods against their average for the Taxi sequence. (b) Closer look at the middle cluster reveals that most of the values lie within the limits of agreement.
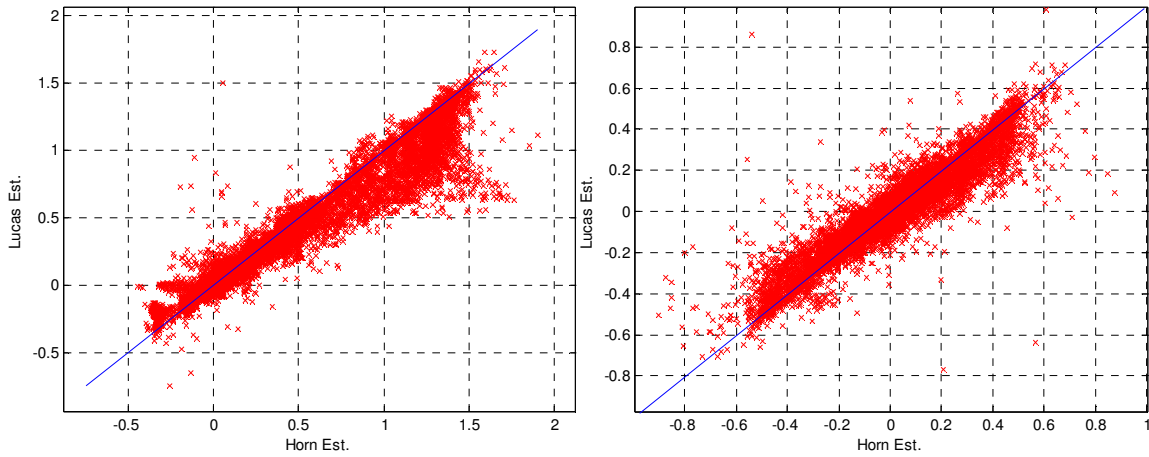
(a)   (b)

Figure 4.28. Equality plots of the estimated velocities for the Rubik's cube sequence. (a) Equality plot of the horizontal component. (b) Equality plot of the vertical component.
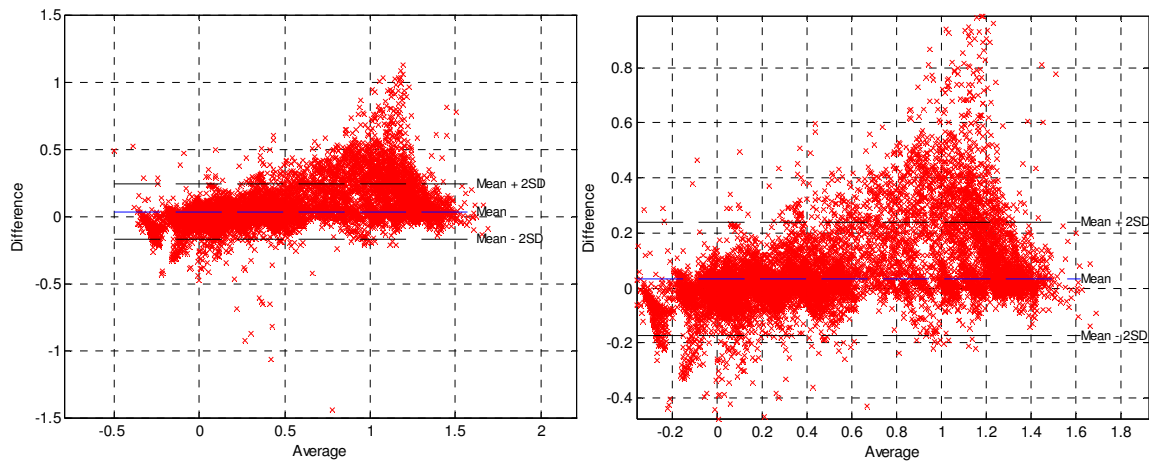
(a)   (b)

Figure 4.29. (a) Difference in horizontal velocity estimation by the two methods against their average for the Rubik's sequence. (b) Closer look at the middle section of the plot reveals a slight bias and high values of the limits of agreement.

114

(a)   (b)
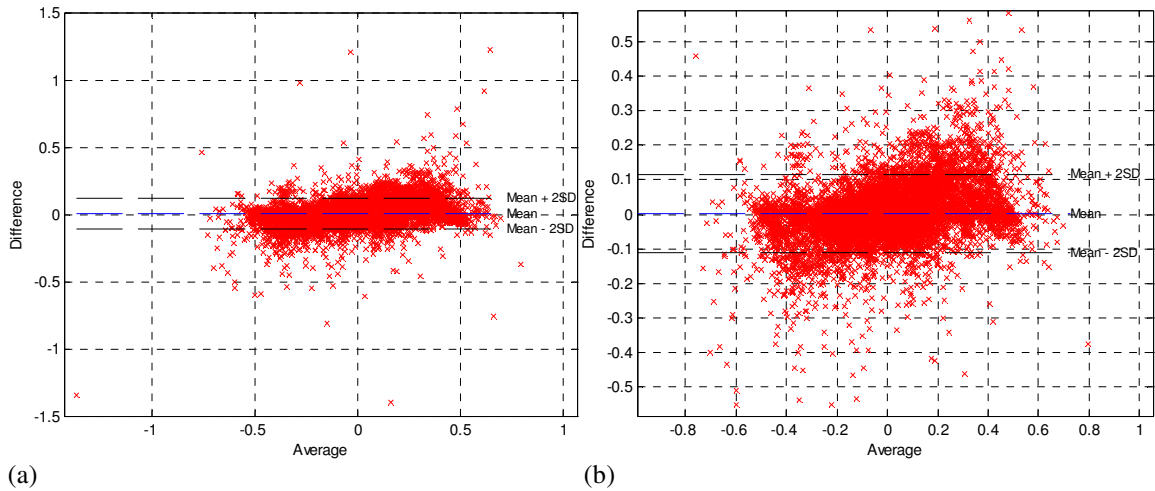
Figure 4.30. (a) Difference in vertical velocity estimation by the two methods against their average for the Rubik's sequence. (b) Closer look at the middle section of the plot indicates almost zero bias and better limits of agreement values than the horizontal component results.

Table 4.6. Mean of the difference in estimation by the two methods and the agreement limit values for real sequences.

	Taxi	Rubik
Diff. Mean U	-0.0053	0.0328
Upper Limit U	0.1245	0.2378
Lower Limit U	-0.1351	-0.1721
Diff. Mean V	-0.003	0.0032
Upper Limit V	0.0933	0.1166
Lower Limit V	-0.0993	-0.1102

(a)   (b)
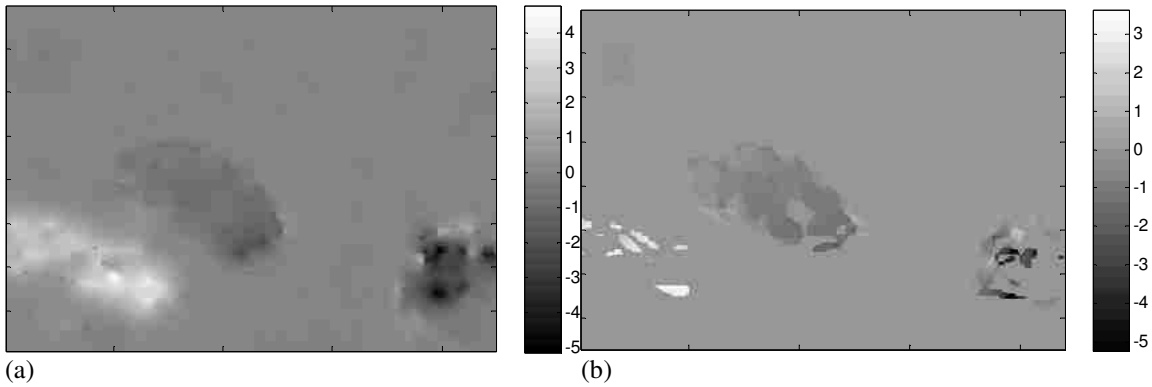
Figure 4.31. Horizontal velocity estimates for the Taxi sequence. (a) Horn's estimates. (b) Lucas's estimates.

(a)   (b)
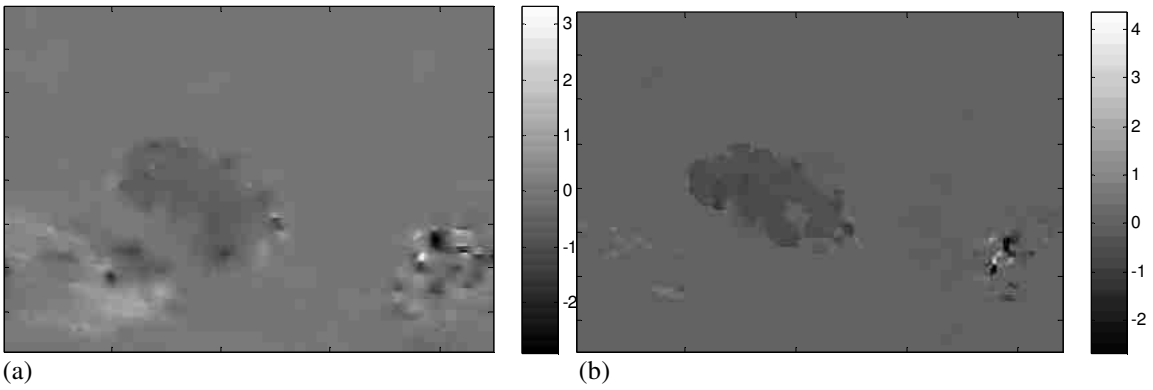
Figure 4.32. Vertical velocity estimates for the Taxi sequence. (a) Horn's estimates. (b) Lucas's estimates.

Figure 4.33. Confidence map of the velocity estimates. Dark areas with low confidence are located on the objects with the largest velocities close to 3 pixels per frame.

(a)   (b)
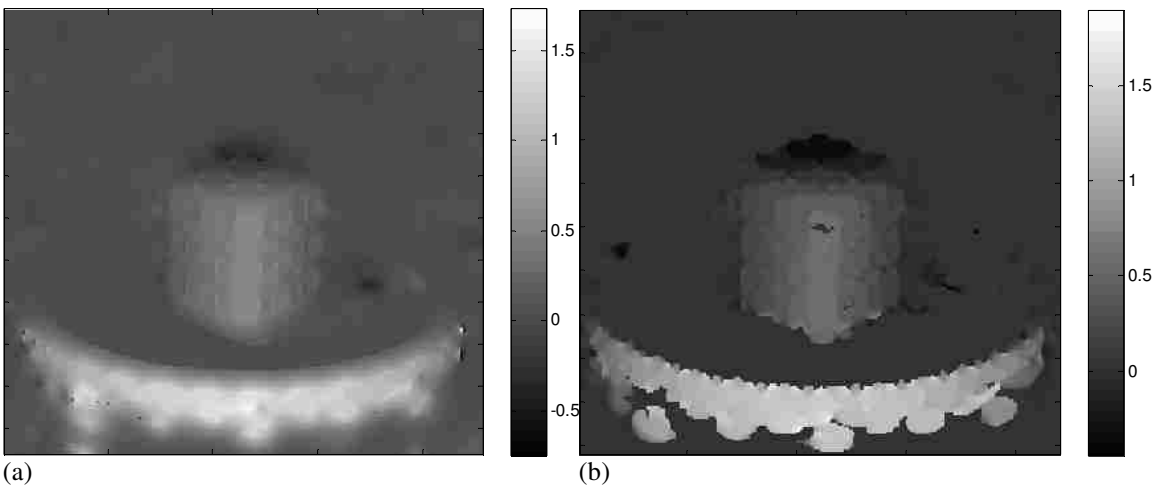
Figure 4.34. Horizontal velocity estimates for the Rubik's cube sequence. (a) Horn's estimates. (b) Lucas's estimates.

(a)   (b)
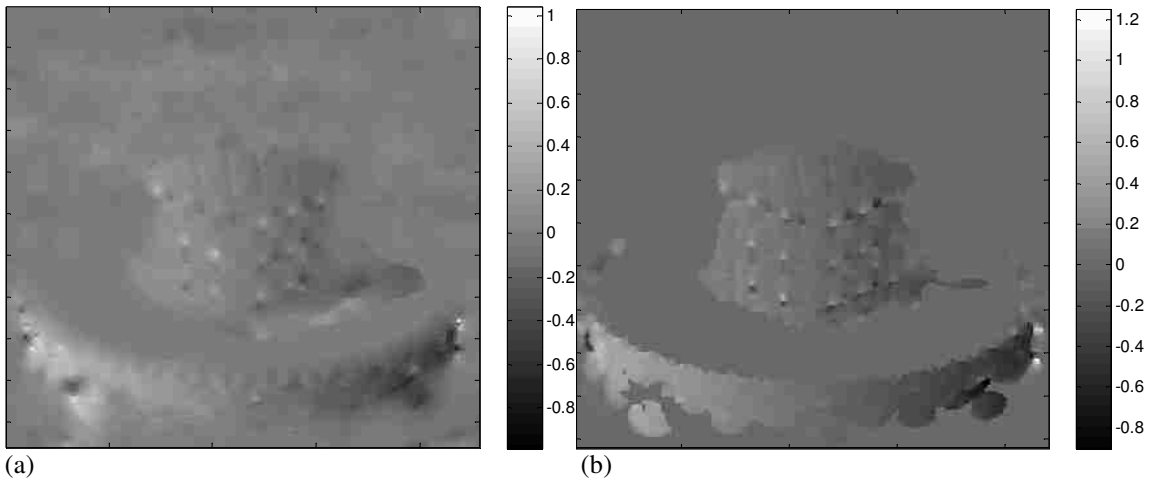
Figure 4.35. Vertical velocity estimates for the Rubik's cube sequence. (a) Horn's estimates. (b) Lucas's estimates.
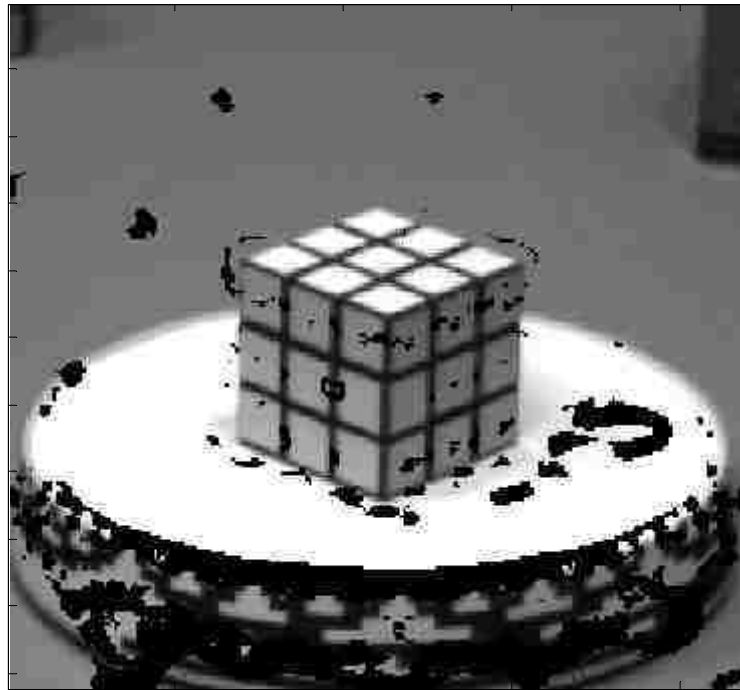
Figure 4.36. Confidence map of the velocity estimates for the Rubik's sequence. Dark areas with low confidence are located on the borders of the turntable.

4.6 Discussion

4.6.1 Discussion for synthetic sequences

The results on the three test video sequences provide good examples of the performance of the no-reference global optimization strategy. Results for the Tree sequences are far

better than the results of the Yosemite video. The blind parameters produce error values comparable to those produced with the optimal parameters. The difference of MSE for the Translating Tree sequence is in the order of 8×10^{-4} which for all practical purposes is an insignificant amount. Differences for the angular errors are also very small. Here, note that the blind parameters produce smaller error metrics than those published in [27]. In the case of the Divergence Tree sequence, the difference in MSE between the blind strategy and the global optimization is 2.7×10^{-3}, and the differences in angular error is also very small.

Results for the Yosemite sequence are far from satisfactory. This is not a failure of the blind approach rather it can be attributed to the limitations of the motion estimation algorithms. Note that the first assumption of the blind approach states that the optical flow methods should be able to estimate the unknown velocity field. Thus, there should exist a set of motion parameters that produce accurate estimates. For this example, this is clearly not the case. This is a realistic and really important assumption in order to have success when estimating the parameters blindly.

The equality plots provide a good graphical tool to asses the agreement in the estimation. The blind approach relies heavily on maximizing agreement, and most of the estimates should lie on the equality line. More revealing are the equality plots of the velocity estimates against the ground truth. Most of the errors can be observed for the larger velocity values. This is a limitation of the techniques that assume small displacements and use finite differencing to compute image and velocity derivatives. Theoretically, the error should be independent of the value being tried to estimate, but from the examples this is not the case. As long as the majority of velocity values are not large, there should

not be a problem applying the motion estimation techniques. In any case, the blind strategy can be applied to any technique. If there is a motion estimation implementation that overcomes the small displacement limitation, then it can be used with an implementation of similar performance to produce accurate velocity estimates.

Another possible solution is to identify clusters of velocities and optimize for those values independently. Thus, different parameters can be used to accurately estimate certain velocity values. The final velocity estimates can be produced by combining the individual results. Right now, one set of parameters is used to estimate all the velocity values of the motion field, but the evidence suggest that improved performance could be achieved by changing the parameters and combine the estimates in one motion field. The same idea could be applied to each velocity component by optimizing separately for each component. Then an improved velocity field could be produced with the combination of each optimized component. This strategy makes sense if the values of the horizontal and vertical velocity components are very different from each other.

The limits of agreement can be used as a confidence metric on the estimation provided that their values are small. For example, these values for both tree sequences are acceptable but not those of the Yosemite sequence. In this latter case, the limits have values close to half a pixel per frame. Considering that the goal was to produce a difference distribution close to zero mean normal with 0.02 of standard deviation, agreement values ten times bigger than the goal is not acceptable. Limits around 0.1 to 0.15 can still be considered acceptable. Let's keep in mind that if the differences were to follow a normal distribution, then 95% of the estimated velocities would lie within these limits.

119

4.6.2 Discussion for natural sequences

The estimated velocity values for the Taxi sequence agree with the guidelines obtained for this sequence, section 4.5.1. The limits of agreement provide a confidence map of reliable velocities. It is expected that most of the disagreement or unreliable velocities are located on the objects moving at 3 pixels per frame as shown in figure 4.33. The velocity estimation for the pedestrian and taxi show good agreement and most of the reliable velocities belong to these objects.

The velocity values for each method, figure 4.31 and figure 4.32, show that despite having objects moving at different velocities, the "blind parameters" are able to produce velocities that are within the range of the expected velocity values for these objects. The results of Lucas's technique seem to be closer to the guideline values, but suffer from low density. Meanwhile, Horn's method is able to estimate velocities for most of the image, higher density, but it seems to struggle to estimate values close to 3 pixels per frame. The difference in density is due to the local and global solutions of the methods, see sections 2.4 and 2.5.

The confidence of the Rubik's cube sequence shows that reliable estimates are located on the cube with velocity values within the range of the expected velocity values. Most of the disagreement occurs for the estimation of the turntable which according to the guidelines range from 1.2 to 1.4 pixels per frame, but both methods are estimating velocities in the range of 1.5 pixels per frame, figure 4.34 and figure 4.35. From the confidence map, figure 4.33, it can be seen that most of the error is concentrated at the borders of the turntable. Velocities for the borders of the objects are always difficult to estimate due the change of the spatial intensity gradient. This effect is more noticeable

when the objects are close to the edges of the image like in this case. Notice that there is high confidence in the estimates of the middle section of the turntable with values within the expected velocities for this object.

It is interesting to note that there is little disagreement in the estimates of the vertical velocities suggesting that different parameters are needed for vertical and horizontal velocities.

4.7 Concluding remarks

The proposed no-reference optimization strategy shows that it is possible to obtain accurate or reliable velocity estimates despite not having access to the unknown motion field. Throughout the different results presented in this chapter, it has been shown that maximizing agreement between the estimation of different motion estimation techniques leads to more accurate velocity estimate than those produced with the standard approach of heuristically choosing the motion parameters. Moreover, a confidence map of the estimation can be created which allows the identification of areas where velocity estimation is not performing well. This is a very important contribution because the confidence in the estimation comes from the estimated velocity value depending if it is close or not to the true unknown velocity. Most of the confidences metrics in the literature depend on heuristics like thresholds on the magnitude of the intensity gradient [1, 17, 27].

The no-reference optimization shows that it is possible to select motion parameters that produce reliable velocity estimates for real video sequences. This contribution is one of the most important made in this dissertation because it extends the application of motion

estimation techniques to real video sequences producing accurate velocity estimates despite not having access to the true displacement field.

Chapter 5

Applications to Motion, Strain, and Pixel Trajectory Estimation of Carotid Artery Plaques

This chapter presents the application of the full reference and no-reference optimization techniques to the study of carotid artery plaque motion. Optimization is used in calculating the motion parameters able to produce accurate pixel velocity estimates of ultrasound videos of carotid artery plaques.

The pixel velocities are used with Kalman filters to provide motion trajectories throughout the cardiac cycle. K-means clustering is used in identifying different motion patterns present in the video. Strain maps of atherosclerotic plaques are produced with a 2-D strain tensor that identifies the deformation directions.

5.1 Motivation

Biomedical applications are of great interest for computer aided diagnosis systems. In the case of ultrasound videos of carotid plaque artery, the main motivation is to develop verifiable methods for the estimation of plaque motion and reconstruction of plaque trajectories with the expectation that plaque and wall motion analysis will provide additional information about plaque instability. Plaque build-up is a characteristic of

atherosclerosis, and a consequence of progressive intimal accumulation of lipid, protein, and cholesterol esters in the blood vessel wall [58] which results in a significant reduction of blood flow. Atherosclerosis is the primary cause of strokes and the third cause of death in the United States [59] with almost twice as many people dying from cardiovascular diseases than from the combined deaths due to cancer.

Motion estimation of ultrasound videos of carotid artery plaques provides important information regarding plaque deformation and feature extraction to distinguish between stable and unstable plaques or symptomatic and asymptomatic plaques [60].

5.2 Introduction

Previous studies [2, 28, 54] have assessed the accuracy of optical flow based techniques by generating ground truth video sequences that capture standard video motion characteristics like translational constant motion, non-deformable body motion, wide range of velocity values, diverging velocity fields, and occluding edges using a uniform light source. Despite this generalized use, there are not, to the best of my knowledge, biomedical motion models to simulate the displacement of plaques and artery walls on carotid ultrasound videos.

Atherosclerotic plaque motion has been previously studied. Block matching using cross correlation was the most common technique found [61-63]. However, block matching does not provide motion estimation resolution at the pixel level. Since it provides a single motion vector for an entire block of pixels, and is not accurate enough for video image analysis applications

To provide high-resolution motion estimates, it is standard practice to use optical flow methods. Motion estimation using optical flow models has been proven to be a useful non-invasive tool for diagnosis of symptomatic and asymptomatic atherosclerotic plaques [60]. Results using this increased resolution will yield better understanding of the mechanics and deformations of atherosclerotic plaques during the cardiac cycle, and can be potentially used to predict their rupture.

5.3 Methodology

Prior research [62] does not address some of the unique characteristics associated with ultrasound imaging of atherosclerotic plaque motion. Thus, for the full-reference global optimization approach, a realistic motion simulator was developed based on clinical expectations using ultrasound images of atherosclerotic plaques.

The simulations cover a variety of periodic motions due to the cardiac cycle along with their discontinuities that mimic sudden plaque and artery wall movements during systole and diastole. These simulations are used to estimate globally optimal parameters that would be applicable to a wide variety of realistic conditions. Once the optimal parameters are found, they are used to calculate the motion of clinical ultrasound videos with special focus on the plaque which is segmented using an automated algorithm [64].

The developed methodology considers that speckle is a major source of inaccuracies when estimating ultrasound video motion due to loss of coherence that causes changes in image intensities. Here, it is important to note that prior research did not address the need to account for significant levels of ultrasound speckle.

The simulated examples were corrupted with speckle noise following the model proposed in [65]. This model assumes that after logarithmic compression of the envelop detected signal, speckle can be modeled as independent and identically distributed (i.i.d) additive white Gaussian noise (AWG). Thus, proper optimization is carried out to find the limits of the motion estimation techniques by corrupting the simulations with different levels of speckle signal to noise ratios (SNR). This is used to calculate the set of optimal input parameters that produce the most accurate velocity estimates at a given SNR.

The no-reference approach will also be used on clinical ultrasound videos in order to find the set of motion parameters that can accurately estimate video motion. Moreover, the confidence maps will identify regions with reliable velocity estimates for better analysis of plaque and artery motion.

The estimated velocities calculated with the set of optimal parameters (full-reference and no-reference) are fed to a Kalman filter [66] that reconstructs pixel trajectories, and these trajectories are used to analyze plaque motion patterns using Principal Component Analysis (PCA).

Pixel velocity estimates also allow the study of plaque elastic behavior in a new non-invasive approach by computing a 2D strain tensor that quantifies the deformation of the different plaque components with the amount of deformation depending on the tpoissue's mechanical properties. These deformations could lead to the differentiation of stable and unstable plaques.

A block diagram of the developed plaque motion and deformation analysis system is presented in figure 5.1.
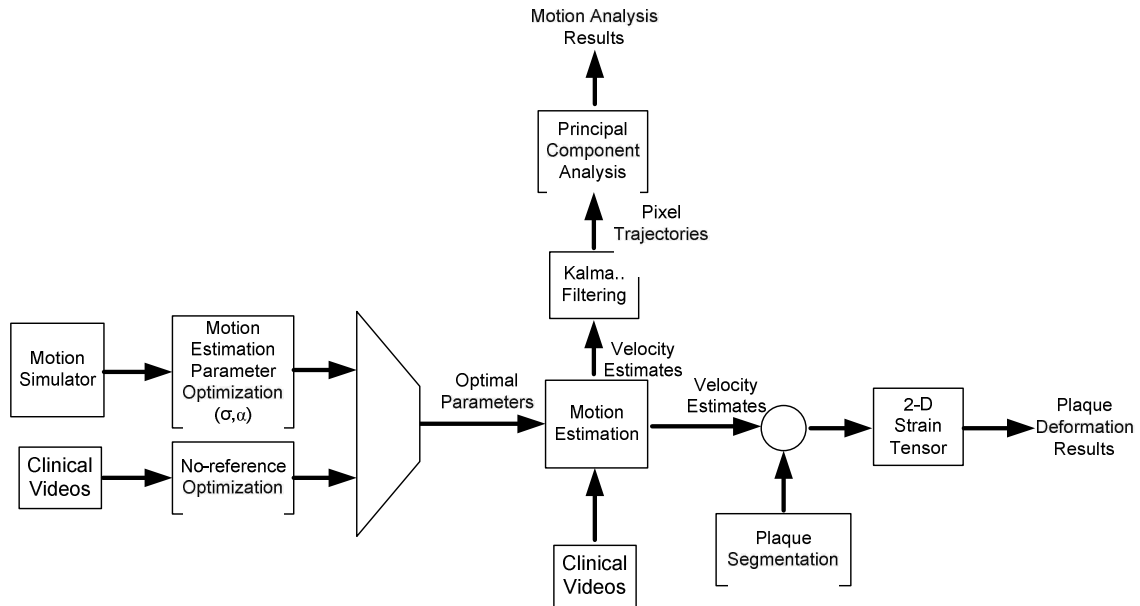
Figure 5.1. Block diagram of the developed plaque motion and deformation analysis system.

5.3.1 Motion model for ultrasound videos of carotid artery plaque

Periodic motion in ultrasound videos of carotid plaque artery can be observed, and it comes from the fact that the artery follows the cardiac cycle. Clinical verification on atherosclerotic plaque videos was given in previous work published in [29][14][67] and also reported by Lever et al. [68], Golemati et al. [61], and Stoitsis et al. [62] where a cyclical pattern along with its discontinuities was observed. In this dissertation, these periodic motions (including their discontinuities) are simulated using a Fourier Series expansion. The motion period can be determined from the videos, and motion discontinuities are expected during the cardiac cycle. These discontinuities produce Fourier harmonic coefficients that are inversely proportional to the fundamental frequency [69].

A realistic motion model is created by taking the first frame of the video sequence and applying a set of coordinate transformation equations in order to displace the pixel

127

intensity values to new coordinates. The coordinate transformation creates image pixel

displacements governed by the following set of equations:

$$y(t) = A_a \sin\left(\frac{2\pi}{N} f_a t\right) + \frac{A_a}{2} \sin\left(\frac{2\pi}{N}(2f_a t)\right) + \frac{A_a}{3} \sin\left(\frac{2\pi}{N}(3f_a t)\right) + \eta_a(x,y,t) \quad (5.1)$$

$$x(t) = A_l \sin\left(\frac{2\pi}{N} f_l t\right) + \frac{A_l}{2} \sin\left(\frac{2\pi}{N}(2f_l t)\right) + \frac{A_l}{3} \sin\left(\frac{2\pi}{N}(3f_l t)\right) + \eta_l(x,y,t), \quad (5.2)$$

where the amplitude of the axial motion component (5.1) is given by A_a while the lateral

motion amplitude is represented by A_l (see figure 5.2 for definitions of the axial, $y(t)$,

and lateral, $x(t)$, directions). Similarly, lateral and axial motion frequencies are referred

by f_l and f_a. In (5.1) and (5.2) N is the total number of frames in the simulation and t

is the index time parameter. Note that amplitudes are decaying at a rate that is inversely

proportional to the fundamental frequency simulating motion discontinuities that can be

induced by random events like for example lung motions [7, 14].

The simulator works under the assumption that plaques with low degree of stenosis

(stenosis is the narrowing of the carotid artery due to plaque build up) move at larger

amplitudes than plaques with high stenosis since the narrowing of the artery increases

with the stenosis degree. The simulation parameters are given in table 5.1, and figure 5.2

shows an ultrasound video frame with the axial and lateral directions.

Table 5.1. Synthetic motion simulation parameters. Amplitudes, frequencies, and degree of stenosis for each simulation case. These parameters are set to cover different motion patterns observed in clinical ultrasound videos. The resolution of the videos is 0.15 mm per pixel.

Synthetic Motion Parameters				
Video Number	#1	#2	#3	#4
Stenosis (%)	73%	60%	52%	26%
Axial Frequency (cycles per video length)	5	3	5	5
Lateral Frequency (cycles per video length)	6	2.5	4	4.5
Axial Motion Amplitude (pixels)	1.5	1	3.5	5.5
Lateral Amplitude (pixels)	2	2	2.5	3.0
Frame Dimensions (pixels x pixels)	100x240	221x251	125x250	125x250
Number of frames	300	200	300	300
Maximum Velocity Magnitude (pixels/frame)	0.5	0.3	1.1	1.8
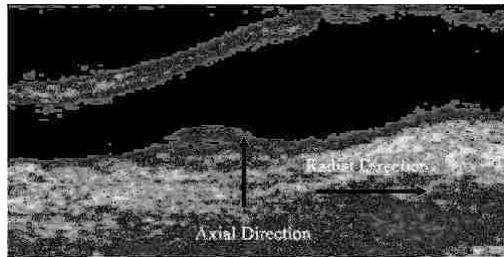
Figure 5.2. Single frame of a carotid ultrasound image. The figure illustrates the axial and lateral directions.

For generating synthetic velocities, an automated segmentation method [64] was used to separate the plaque from the rest of the video image. Both the plaque and the non-plaque regions were originally set using two separate Fourier Series expansions. Since plaques are extensions of the carotid artery, convolution was applied on the boundary to make the motion smoother.

5.3.2   Speckle noise simulations

Speckle generally modeled as a form of multiplicative noise corrupts ultrasound images and makes visual inspection difficult even for the most trained specialists [70]. Speckle also degrades the accuracy of the motion estimation algorithms.

When the envelope signal is captured at the output of the beamformer before logarithmic compression, speckle can be modeled as a multiplicative form of noise following a Rayleigh distribution [70]. Then, logarithmic compression is applied to the signal in order to fit it to the display range. The effects of logarithmic compression have been studied in [71][72], where it was shown that it affects the tails of the Rayleigh distribution in a manner that speckle can now be expressed as additive white Gaussian noise (WGN). Therefore, the multiplicative noise model transforms into a classical additive one [65]:

$$g_{ij} = f_{ij} + n_{ij}. \tag{5.3}$$

The term g_{ij} is the ultrasound displayed pixel after logarithmic compression, and the terms f_{ij} and n_{ij} are the noise-free pixel and noise component after logarithmic compression. The spatial coordinates in the 2D image plane are given by i and j.

Using (5.3), independent and identically distributed (i.i.d) GWN of different variances were added to the simulated video in order to generate corrupted videos of 10 db, 20 dB, 30 dB, 40 dB, and 50 dB signal to noise ratios (SNR), and they are used to calculate the set of optimal input parameters that produce the most accurate velocity estimates at a given SNR.

5.3.3 Full-reference global optimization for biomedical simulations

The model (5.1)-(5.2) is used to create five synthetic carotid artery plaque motions according to the parameters in table 5.1. The goal for each simulation case and motion estimation technique is to find the set of parameters that minimize the MSE between the estimated velocities and ground truth. The full-reference optimization algorithm of section 3.3 is used to find the optimal parameters that minimize the MSE of the velocity

130

estimation. Intersecting the optimal interval results of all the simulated cases leads to a common optimal point that will be used as the starting value of the motion parameters when estimating motion of clinical ultrasound videos.

5.3.4   No-reference global optimization

The no-reference approach is used to find the motion parameters that produce accurate motion estimates of clinical ultrasound videos. The velocity estimates are used to calculate strain maps of atherosclerotic plaques.

5.3.5   Trajectory reconstruction

Once the optimal parameters are calculated, they are used in clinical ultrasound videos in order to estimate the corresponding velocity field. For tracking the motions, Kalman filters were used for every pixel. The goal is to predict the position of a pixel given its initial coordinates and velocity estimates throughout the video sequence. The linear models for trajectory reconstruction applied independently to the lateral and axial directions are given by:

$$\begin{bmatrix} x(k+1) \\ u(k+1) \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x(k) \\ u(k) \end{bmatrix} + \xi_l(k)$$
$$\begin{bmatrix} u(k) \end{bmatrix} = \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} x(k) \\ u_t(k) \end{bmatrix} + \eta_l(k) \tag{5.4}$$

$$\begin{bmatrix} y(k+1) \\ v(k+1) \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} y(k) \\ v(k) \end{bmatrix} + \xi_a(k)$$
$$\begin{bmatrix} v(k) \end{bmatrix} = \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} y(k) \\ v_t(k) \end{bmatrix} + \eta_a(k) \tag{5.5}$$

The state vector is composed of position and velocity, and the index k indicates the video frame numbers. The velocity estimates, $u(k)$ and $v(k)$, are the measurements and can be treated as observations in the Kalman filter framework. The vector $\begin{bmatrix} u_t(k) & v_t(k) \end{bmatrix}^T$ represents the motion vector. Process noise and observation noise are modeled as zero-mean, uncorrelated, white-noise $\xi(k)$, and $\eta(k)$ processes respectively. The covariance matrix of the process noise and variance of the observation noise are given by:

$$E\left[\xi(k)\xi^T(k)\right] = Q(n) = \sigma_\xi^2 \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \text{ with } \sigma_\xi^2 = 0.5 \; pixels^2, \text{ and}$$
$$\sigma_\eta^2 = 0.1 \; (pixels/frame)^2.$$

(5.6)

5.3.6 Principal Component Analysis (PCA)

PCA [73] is applied to the plaque trajectories in order to find patterns from which plaque motion can be characterized. The goal is to differentiate among plaque regions that undergo different motions and identify these patterns along the cardiac cycle.

The motion of a point consists of n samples in each of the two directions (axial and lateral), and can be represented by the vector:

$$q = (x_1, \dots x_n, y_1, \dots, y_n)^T.$$

(5.7)

Before creating the data matrix, location dependency is removed from each trajectory. Then, the trajectories are placed as measurements in the $2n \times p$ data matrix of standardized trajectories given by:

$$
\mathbf{Z} = \begin{bmatrix}
\dfrac{q_{11}-\bar{q}_1}{\sqrt{s_{11}}} & \dfrac{q_{12}-\bar{q}_1}{\sqrt{s_{11}}} & \cdots & \dfrac{q_{1p}-\bar{q}_1}{\sqrt{s_{11}}} \\[2mm]
\dfrac{q_{21}-\bar{q}_2}{\sqrt{s_{22}}} & \dfrac{q_{22}-\bar{q}_2}{\sqrt{s_{22}}} & \cdots & \dfrac{q_{2p}-\bar{q}_2}{\sqrt{s_{22}}} \\[2mm]
\vdots & \vdots & \ddots & \vdots \\[2mm]
\dfrac{q_{N1}-\bar{q}_N}{\sqrt{s_{NN}}} & \dfrac{q_{N1}-\bar{q}_N}{\sqrt{s_{NN}}} & \cdots & \dfrac{q_{Np}-\bar{q}_N}{\sqrt{s_{NN}}}
\end{bmatrix}, \tag{5.8}
$$

where $N=2n$, $\bar{q}_i = \dfrac{1}{p}\sum_{j=1}^{p} q_{ij}$, and $s_{ii} = \dfrac{1}{p}\sum_{j=1}^{p}\left(q_{ij}-\bar{q}_i\right)^2$ with $i=1,\ldots,N$. The principal

components (PC) are obtained from the covariance matrix $\mathrm{Cov}(\mathbf{Z})$ with eigen-value-

eigenvector pairs $(\lambda_1,\mathbf{e}_1),(\lambda_2,\mathbf{e}_2),\ldots,(\lambda_p,\mathbf{e}_p)$. Only the k PC that account for 90% of

total variance are retained.

The matrix whose columns are the of eigenvectors $\Phi_k = (\mathbf{e}_1,\ldots,\mathbf{e}_k)$ is used to

approximate any standardized trajectory z_i by:

$$
z_i \approx \bar{z} + \Phi_k \mathbf{b_i}, \tag{5.9}
$$

and the k-dimensional vector $\mathbf{b_i}$ is calculated as:

$$
\mathbf{b_i} = \Phi_k^T \left(z_i - \bar{z}\right). \tag{5.10}
$$

The PCs define a k-dimensional plane, and the elements of the k-dimensional vector $\mathbf{b_i}$

are the coordinates of the projection of each trajectory onto this low dimensional sub-

space [73]. The $\mathbf{b_i}$ vectors are inputted to a *k-means* clustering algorithm [73] that is used

to distinguish between different motion patterns.

Finally, the same motion analysis is also applied independently to the axial and lateral

directions in order to gain more insight about the movement of the plaque. The 2D model

presented in (5.7)-(5.8) is equivalent to the 1D models when the motion in the axial and lateral directions are independent.

5.3.7 Strain calculation and principal axis of deformation

Pixel velocity estimates allow the study of plaque elastic behavior by computing a 2D strain tensor that quantifies the deformation of the different plaque components. For every plaque pixel, a strain tensor that measures the elastic behavior of the underlying tissue is estimated.

When an object deforms, the distance between points changes, and the amount of deformation depends on the tissue's mechanical properties. These deformations along with other indicators of stroke can help to differentiate between stable and unstable plaques.

In 2-D object deformation, there are four strain components of a 2×2 strain tensor matrix. The diagonal entries ε_{xx} and ε_{yy} give the extensions or contractions, deformations normal to the object borders, and the off-diagonal entries ε_{xy} and ε_{yx} give the shear, deformations parallel to the object border. The 2-D tensor, E, is given by:

$$E = \begin{bmatrix} \varepsilon_{xx} & \varepsilon_{xy} \\ \varepsilon_{yx} & \varepsilon_{yy} \end{bmatrix}. \tag{5.11}$$

The total deformation of a point is measured by the trace of E. The components of the tensor are computed from the gradient of the velocity field. The strain is the spatial derivate of the tissue velocity and each component in (5.11) is calculated according to:

$$E = \begin{bmatrix} \dfrac{\partial u}{\partial x} & \dfrac{\partial u}{\partial y} \\[2ex] \dfrac{\partial v}{\partial x} & \dfrac{\partial v}{\partial y} \end{bmatrix}. \tag{5.12}$$

The gradient of the velocity field is calculated using a 2-D Gaussian derivative filter. The spread of the Gaussian will need to be adjusted according to characteristic of the motion field. Here, higher spreads are used to suppress noise (inaccurate estimates), and smaller spreads are used when the velocity estimation is accurate, see section 5.3.8 and results in section 5.4.6.

The directions of pure extensions and contractions are the principal axes of deformation (no shear occur along these axes) with associated vectors that point into the directions of these deformations. The principal axes are given by eigenvectors of the 2-D strain tensor (5.11). The sign of the associated eigenvalues distinguishes between positive and negative deformations (stretching or shortening) in the direction of their corresponding eigenvectors. Thus, for each pixel with an associated strain tensor the principal deformation axes can be visualized as an ellipsoid with axis length proportional to the magnitude of its eigenvalue. The shape of the ellipsoid, governed by the ratio of the eigenvalues, determines how an infinitesimal circular area is deforming. When the deformation is concentrated along a single direction, the thinner and line-like the corresponding ellipsoid is. On the other hand, when the deformation is equally distributed along both axes, the ellipsoid resembles a circle.

5.3.8 Differentiation of velocity fields

In order to select the best derivative filter, a deterministic motion field was created using chirp functions that span a wide range of frequencies. Independent identically distributed (i.i.d) zero mean White Gaussian Noise (WGN) of different noise levels was added to the simulated velocities.

Three filters were used to calculate the velocity derivatives from the noisy estimates. One was a 9-tap derivative filter developed by Farid and Simoncelli [74], and the other two were: a four point difference mask with a Gaussian pre-smoother, and a two dimensional Gaussian derivative filter. In both Gaussian cases, optimization was carried out to find the best value of the filter spread that can suppress noise and accurately calculate the velocity field derivatives for different frequencies.

The chirp field linearly sweeps low and high frequencies with the following equation:

$$u(x, y) = \sin \varphi(x, y)$$
$$\varphi(x, y) = 2\pi \left[\alpha x + \beta \frac{x^2}{2} + \alpha y + \beta \frac{y^2}{2} \right] \tag{5.13}$$
$$\nabla \varphi(x, y) = 2\pi (\alpha + \beta x, \alpha + \beta y)$$

The vertical field is the horizontal and vertical mirrored version of the horizontal field generated using:

$$v(x, y) = fliplr\left(flipud\left(u(x, y) \right) \right) \tag{5.14}$$

The gradient of the field is given by:

$$\nabla u = \begin{bmatrix} u_x & u_y \end{bmatrix}^T$$
$$u_x(x, y) = 2\pi A(\alpha + \beta x) \cos \varphi(x, y) \tag{5.15}$$
$$u_y(x, y) = 2\pi A(\alpha + \beta y) \cos \varphi(x, y),$$

and:

136

$$\nabla v = \begin{bmatrix} v_x & v_y \end{bmatrix}^T$$
$$v_x(x, y) = -2\pi A(\alpha + \beta x)\cos\varphi(x, y) \qquad (5.16)$$
$$v_y(x, y) = -2\pi A(\alpha + \beta y)\cos\varphi(x, y).$$

For (5.13), table 5.2 summarizes the simulation parameters.

Table 5.2. Chirp velocity field parameters. $N = 150$ pixels.

	f_{min}	f_{max}	α	β
Low Freq. Simulation (cycles/pixel)	0.001	0.05	$(f_{min} + f_{max})/2$	$(2 f_{max} - \alpha)/N$
High Freq. Simulation (cycles/pixel)	0.05	0.1	$(f_{min} + f_{max})/2$	$(2 f_{max} - \alpha)/N$

The Signal to Noise Ratio (SNR) of the noisy velocities where simulated at levels of

$5\ dB\ (\sigma = 0.8105)$, $10\ dB\ (\sigma = 0.4558)$, $20\ dB\ (\sigma = 0.0456)$, and $30\ dB\ (\sigma = 0.014)$.

5.4 Results

This section presents all the results obtained with the methods presented in section 5.3. The results will be broken in two sections starting with global optimization results on simulated ultrasound videos (full-reference) and followed by results on clinical videos (no-reference).

5.4.1 Full-reference motion estimation parameter optimization

Horn's technique depends on two parameters: the spread of the Gaussian pre-smoothing filter σ, and the regularization parameter α in model (2.13). Lucas's technique depends only on the Gaussian pre-smoothing spread. The spread of the Gaussian pre-smoothing filter was allowed to vary $\sigma = 0.5 - 4$ pixels for both techniques, and the values of the regularization parameter α varied $\alpha = 1 - 50$ for Horn's technique.

Smooth velocity fields of small amplitudes were simulated for synthetic motions with high stenosis values (see table 5.1). Low stenosis values for simulations #3 and #4 allow the simulation of non-smooth velocity fields of higher amplitudes and frequencies. High velocity fields are difficult to estimate using optical flow models due to the fact that finite differencing errors and large motions do not satisfy the small displacement assumption. The estimation error is measured in terms of the maximum velocity magnitude: *Relative Error = Magnitude Error / Max. Magnitude*. Results of Horn's technique are presented in figure 5.3(a)-(d), and from table 5.3 to table 5.6. The contour plots along with the tables indicate the relative error for a combination of input parameters. For example, in table 5.4, for $\alpha = 1.0$ and $\sigma = 1.0$, the error is 0.1. This means that on average, the error is 10% of the maximum ground truth velocity. Figure 5.4 shows the relative error as a function of the Gaussian pre-smoothing step for Lucas technique.

The information of the contour plots and relative error tables is used to select a set of parameter values that produces the smallest error among all the simulated cases. For Horn's technique, the optimal parameter values are: $\alpha = 1.0$, $\sigma = 1.25$. In the case of Lucas' method, the optimal spread of the smoothing filter can vary in the range $\sigma = 0.5 - 2.25$. The optimal parameters are used as a starting point to estimate pixel velocities on clinical ultrasound videos.

5.4.2 Speckle simulation results

Speckle degrades the accuracy of the ME algorithms. Thus, following [65] the model in (5.3) adds independent and identically distributed (i.i.d) GWN of different variances to the simulations in order to generate degraded videos of 10 dB, 20 dB, 30 dB, 40 dB, and

50 dB signal to noise ratios (SNR). The studied example is based on simulation # 3 (see table 5.1).

The goal is to use the global optimization algorithm to find a set of input parameters able to calculate the best possible velocity estimates. Contour levels of the relative error are shown in figures for the different SNRs. As expected, the estimation error is bigger for videos with low SNR than for videos with higher SNR values. Compared to the noise free simulations, higher regularization parameter and filter spread values are necessary to obtain reliable velocity estimates.

The regularization parameter varies from $\alpha = 2.5 - 15$, and the filter spread can take values from $\sigma = 1.0 - 2.5$ pixels.

5.4.3   Trajectory reconstruction of simulated motion of ultrasound carotid artery plaques

Figure 5.6 shows a frame of simulation #3 (52% stenosis) with three selected points for which lateral and axial trajectories are reconstructed. These trajectories are shown in figure 5.7 (a)-(b). The MSE error for trajectory reconstruction is shown in table 5.7 with a maximum value of $0.033\ pixels^2$ in the lateral direction, and $0.093\ pixels^2$ in the axial direction.

Trajectory reconstruction results for simulation #1 using Lucas's technique are shown in figure 5.8, figure 5.9, and table 5.8. Maximum MSE values of $0.002\ pixels^2$ in the lateral direction and $0.005\ pixels^2$ in the axial direction were achieved.

(a)

(b)
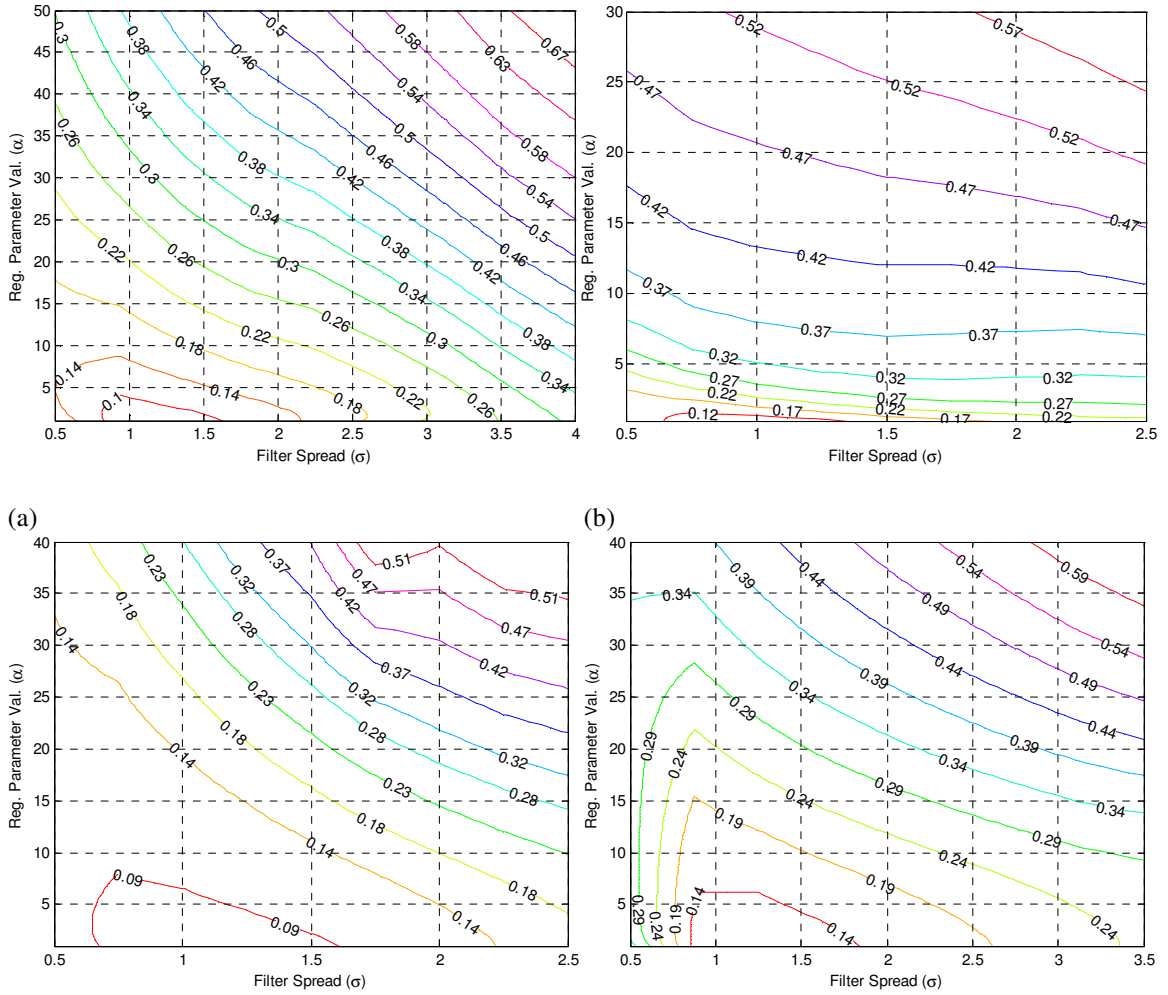
(c)

(d)

Figure 5.3. Contour plots of relative error for full-reference Horn's technique. (a) Relative error for simulation #1. (b) Relative error for simulation #2. (c) Relative error for simulation #3. (d) Relative error for simulation #4.

140

Figure 5.4. Full-reference relative error of Lucas' technique. Errors smaller than 0.1 (10%) are acceptable.

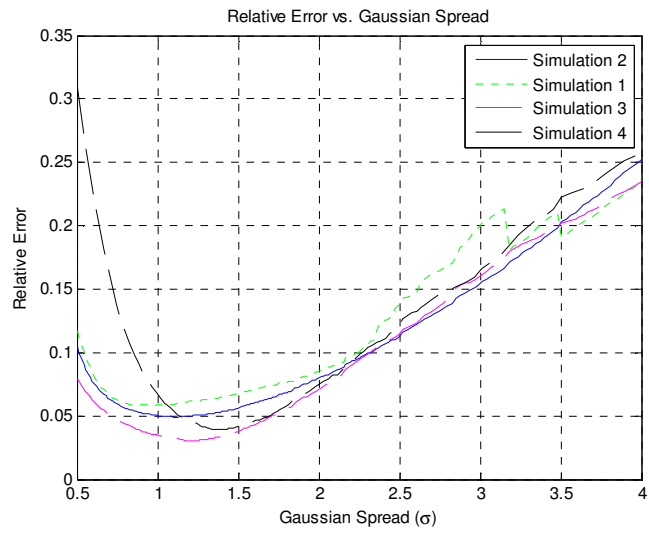Table 5.3. Full-reference values for Horn's technique of the velocity estimation relative error for simulation #1 (73% stenosis). $v_{GT(\max)} = 0.5\ pix / frame$.

Reg. Par. (α)	Smoothing Filter Spread (σ)								
	0.5	0.93	1.37	1.81	2.25	2.68	3.12	3.56	4.0
1.0	0.17	0.08	0.08	0.11	0.15	0.19	0.23	0.26	0.31
1.7	0.15	0.02	0.09	0.11	0.15	0.18	0.23	0.27	0.31
2.5	0.15	0.08	0.10	0.12	0.15	0.19	0.23	0.27	0.32
3.3	0.86	0.91	0.89	0.87	0.84	0.81	0.77	0.72	0.67
4.0	0.86	0.90	0.88	0.86	0.83	0.80	0.76	0.71	0.67
5.5	0.86	0.89	0.87	0.84	0.82	0.79	0.75	0.70	0.65
7.1	0.86	0.87	0.85	0.83	0.81	0.77	0.74	0.68	0.64
8.6	0.85	0.86	0.84	0.81	0.79	0.76	0.72	0.67	0.62

Table 5.4. Full-reference values for Horn's technique of velocity estimation relative error for simulation #2 (60% stenosis). $v_{GT(\max)} = 0.3\ pix / frame$.

Reg. Par. (α)	Spread Smoothing Filter (σ)								
	0.5	0.75	1.0	1.25	1.5	1.75	2.0	2.25	2.5
1.0	0.14	0.10	0.10	0.21	0.13	0.16	0.18	0.19	0.21
1.4	0.14	0.12	0.13	0.15	0.18	0.21	0.22	0.23	0.24
2.1	0.14	0.15	0.18	0.21	0.24	0.26	0.26	0.26	0.27
2.5	0.15	0.18	0.21	0.25	0.27	0.28	0.28	0.28	0.28
2.8	0.16	0.19	0.23	0.26	0.28	0.29	0.29	0.29	0.29
3.0	0.17	0.21	0.24	0.27	0.31	0.30	0.30	0.29	0.30

Table 5.5 Full-reference values for Horn's technique of velocity estimation relative error for simulation #3 (52% stenosis). $v_{GT\,(\max)} = 1.1\,pix\,/\,frame$.

Reg. Par. (α)	Spread Smoothing Filter (σ)								
	0.5	0.75	1.0	1.25	1.5	1.75	2.0	2.25	2.5
1.0	0.87	0.92	0.93	0.93	0.91	0.90	0.88	0.85	0.83
2.5	0.89	0.92	0.92	0.92	0.91	0.89	0.87	0.85	0.82
4.0	0.89	0.92	0.92	0.91	0.90	0.88	0.86	0.84	0.82
5.5	0.89	0.91	0.91	0.90	0.89	0.87	0.85	0.83	0.80
8.6	0.89	0.91	0.90	0.89	0.87	0.85	0.83	0.80	0.78
10.1	0.89	0.90	0.89	0.88	0.86	0.84	0.81	0.79	0.76
11.6	0.89	0.90	0.89	0.87	0.85	0.83	0.80	0.77	0.74
13.1	0.89	0.89	0.88	0.87	0.84	0.81	0.78	0.75	0.72
14.5	0.88	0.89	0.88	0.86	0.83	0.80	0.76	0.73	0.71

Table 5.6. Full-reference values for Horn's technique of velocity estimation relative error for simulation #4 (26% stenosis). $v_{GT\,(\max)} = 1.8\,pix\,/\,frame$.

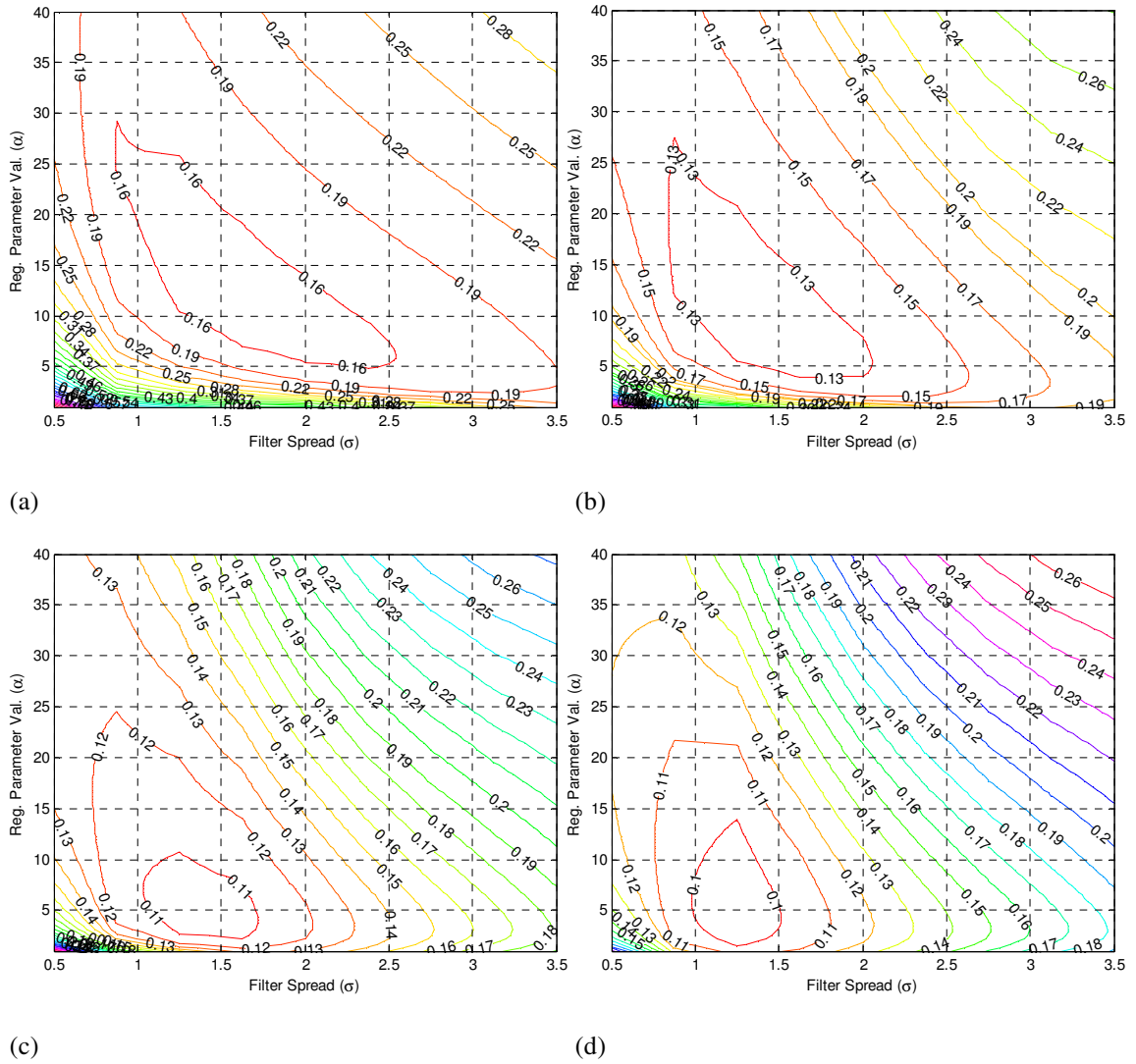| Reg. Par. (α)	Spread Smoothing Filter (σ)								
	0.5	0.87	1.25	1.62	2.0	2.37	2.75	3.12	3.5
1.0	0.28	0.13	0.12	0.13	0.16	0.18	0.21	0.23	0.26
3.7	0.23	0.14	0.13	0.15	0.17	0.20	0.22	0.25	0.27
6.4	0.23	0.15	0.15	0.17	0.19	0.22	0.24	0.26	0.28
9.2	0.23	0.16	0.17	0.20	0.22	0.25	0.27	0.29	0.30
11.9	0.23	0.17	0.20	0.22	0.25	0.28	0.30	0.32	0.33
14.7	0.23	0.19	0.22	0.25	0.28	0.31	0.33	0.35	0.37
17.4	0.23	0.21	0.25	0.28	0.31	0.34	0.36	0.39	0.41
20.1	0.23	0.24	0.28	0.31	0.34	0.37	0.40	0.43	0.45

(a)

(b)

(c)

(d)

Figure 5.5. Contour plots of the relative error for speckle simulations of SNR =10dB (a), 20 dB (b), 30 dB (c), and 40 dB (d). Compared to noise free cases, higher regularization parameters and filter spreads are necessary to obtain reliable velocity estimates.
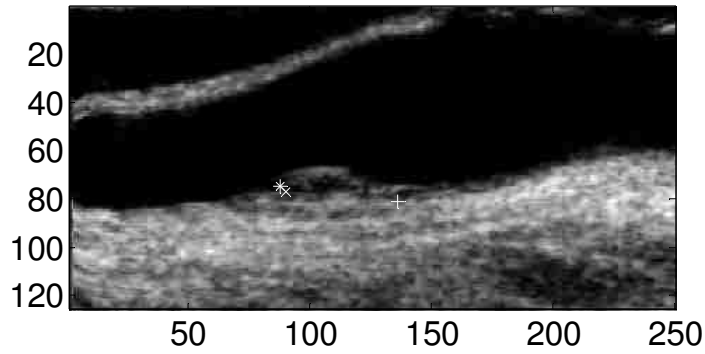
144

Figure 5.6. Three selected points of simulation #3 (52% stenosis) for trajectory reconstruction.

(a)   (b)

Figure 5.7. Lateral (a) and axial (b) trajectory reconstructions of three selected points of Figure 5.6. Ground truth is shown in solid black line. Point 'x' is shown on a blue '. line', point '+' shown on a red '-. line', and point '*' shown a magneta '-- line'.

Table 5.7. MSE table of lateral and axial trajectory reconstruction for simulation #3 (52% stenosis) for Horn's method.

	MSE	
Point	Lateral	Axial
'x'	0.028	0.093
'+'	0.032	0.063
'*'	0.033	0.046

Figure 5.8. Three selected points of simulation #1 for trajectory reconstruction.

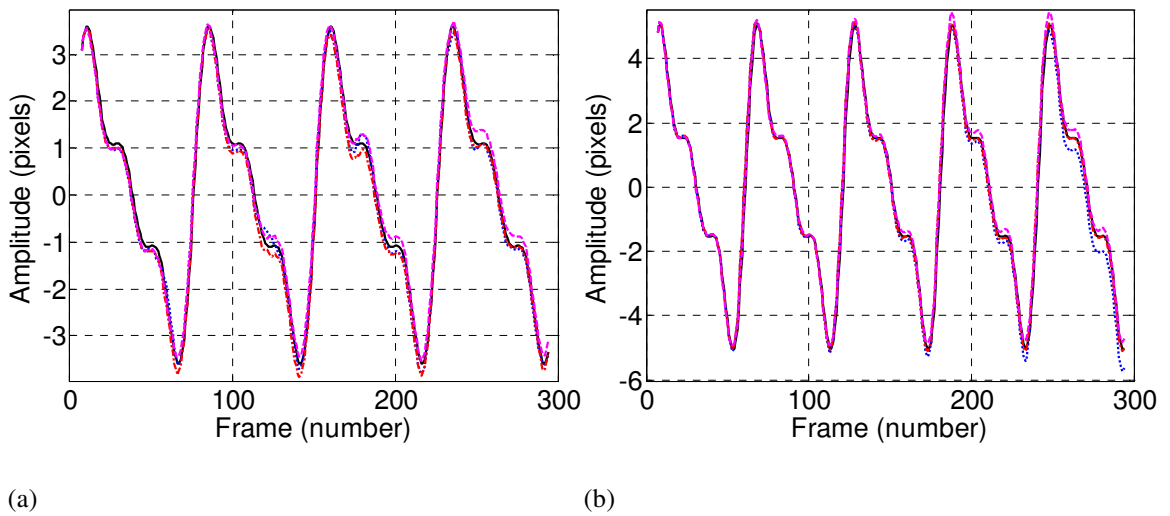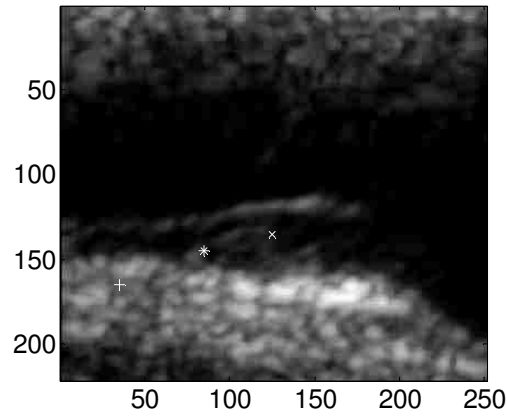(a)   (b)
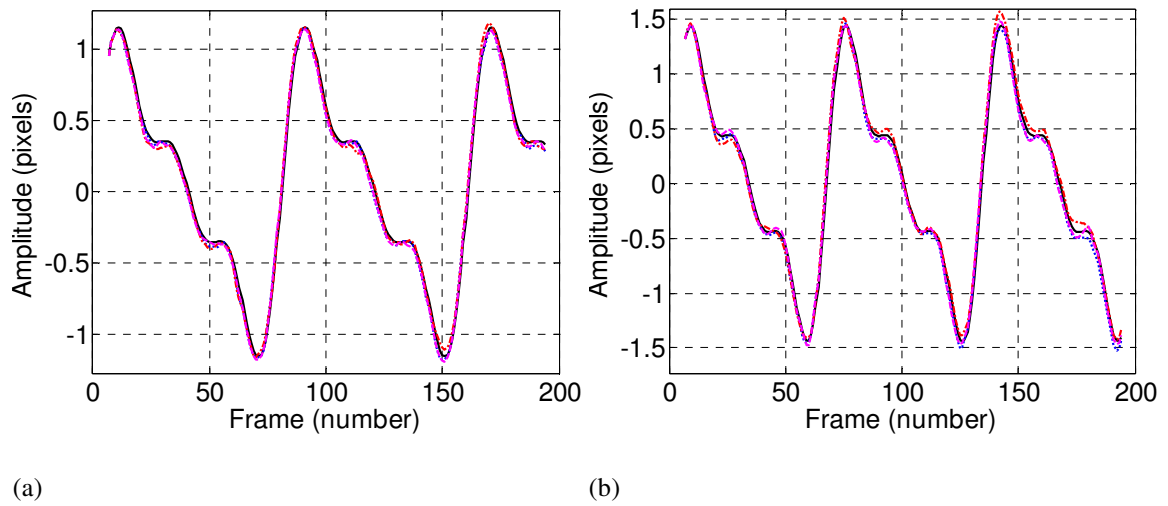
Figure 5.9. Lateral (a) and axial (b) trajectory reconstructions of three selected points of Figure 5.8. Ground truth is shown in solid black line. Point 'x' is shown on a blue '. line', point '+' shown on a red '-. line', and point '*' shown a magneta '-- line'.

Table 5.8. MSE table of lateral and axial trajectory reconstruction of simulation #1 (73% stenosis). Lucas' method.

	MSE	
Point	Lateral	Axial
'x'	0.001	0.003
'+'	0.002	0.005
'*'	0.002	0.003

5.4.4   Motion estimation of clinical ultrasound videos

Velocity estimation of clinical ultrasound videos was performed on a set of twelve videos.

First, the full-reference optimal parameters found in section 5.4.1 are used to estimate video motion. The evaluation of the estimated velocities is done visually. For visual validation, the velocities are first fed to Kalman filters to reconstruct pixel trajectories. The clinical video is then played back with the tracked points overlaid over the plaque. Visual inspection of the tracked points can help detect if the tracked points agree with our expectations. When a large majority of the pixel motions appear to be tracked correctly, there is quality motion estimation. Else, the motion estimation is found to be of poor quality. This process identifies if the current motion parameters are doing a good job at estimating the true displacement field. When the trajectories appear to lose track of the plaque, a new set of input parameters need to be used. Significant levels of speckle is the cause that the velocity estimates produced with the optimal parameters cannot track certain plaque regions. For these cases, the motion parameters are changed to the ones found in the full-reference speckle simulations, section 5.3.2. The full-reference motion estimation parameter values for each clinical video are given in table 5.9. Out of the twelve videos, quality motion estimation was possible on nine. On the remaining three videos the amount of speckle de-correlation, plaque deformation, large displacements, and change of pixel intensity values did not allow for accurate motion estimation. For the remaining nine videos, the velocity field is estimated for every frame of the video and pixel trajectories are reconstructed for the plaque.

Figure 5.10 shows a frame of video #4. The velocity field is overlaid on top of the frame with the velocity vectors pointing in the direction of the estimated displacements. Trajectories for video #7 are shown in figure 5.11. This figure shows the motions of points located over the plaque and near the plaque-artery wall boundary. For these examples, evidence of periodic motion, one of the main assumptions of the motion simulator, can be observed. The periodicity of the motion trajectories is evident in figure 5.12 where two dominant harmonics can be observed.

Table 5.9. Parameter values used to estimate clinical ultrasound video motion for Horn's method.

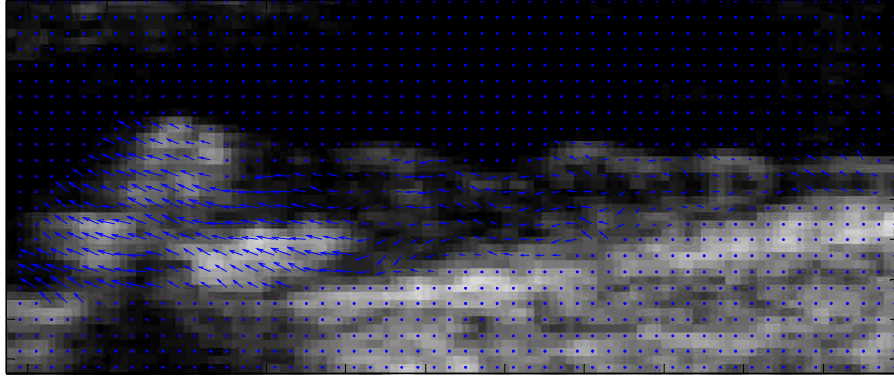|	Parameter Values	
Video	α	σ
1	10	1.5
2	1	1.25
3	15	1.25
4	15	1.25
5	10	1.5
6	10	1.5
7	15	1.25
8	15	1.25
9	15	1.25
10	N/A	N/A
11	N/A	N/A
12	N/A	N/A

Figure 5.10. Estimated velocity field of clinical video #4. The arrows point in the direction of the displacement
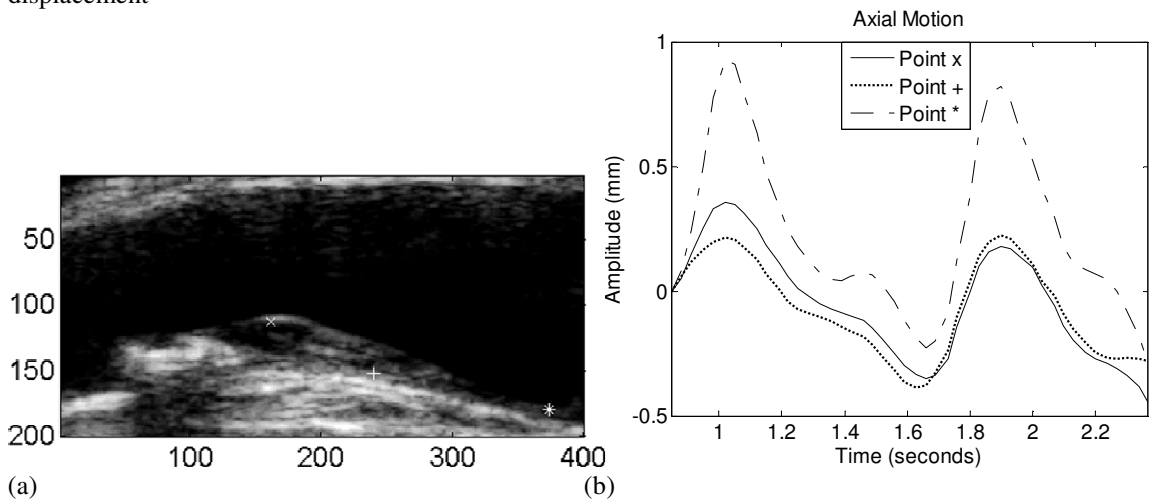

(a)   (b)

Figure 5.11. Estimated trajectories of a clinical ultrasound video. (a) Frame of video #7 with three selected points. (b) Reconstructed trajectories for the three points shown in Figure 5.11a. Evidence of periodic motion is found in the reconstructed trajectories.

Consistency in the reconstructed trajectories for clinical video #2 can be observed in figure 5.13. This is evidence that when two different motion estimation algorithms produce almost the same estimates, they are most likely close to estimating the true video motion. When the incorrect input parameters are used on two different methods, it is easy for the methods to produce different and wrong estimations, but when the correct parameters are used both methods will most likely converge close to the true velocity field.
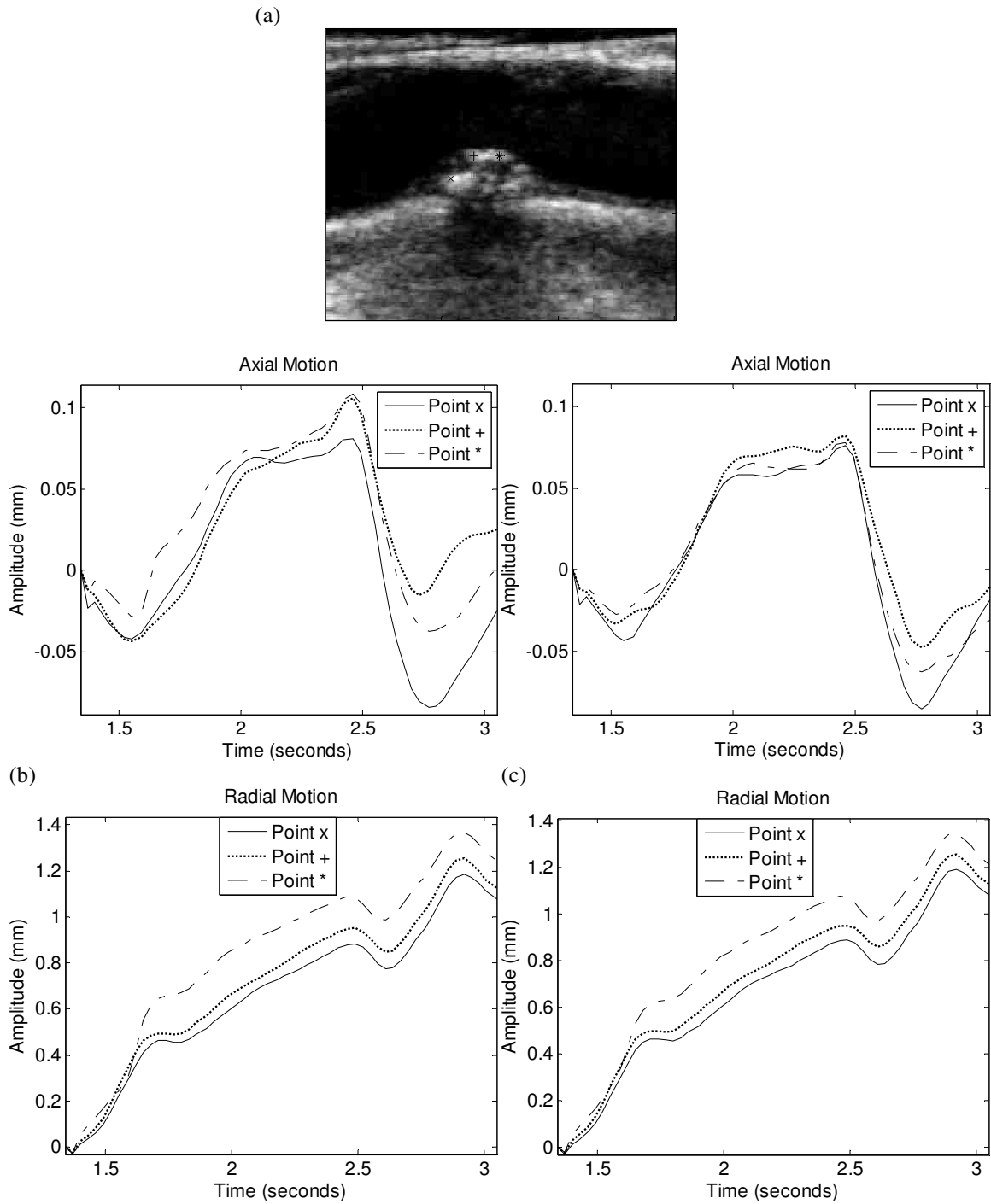
149
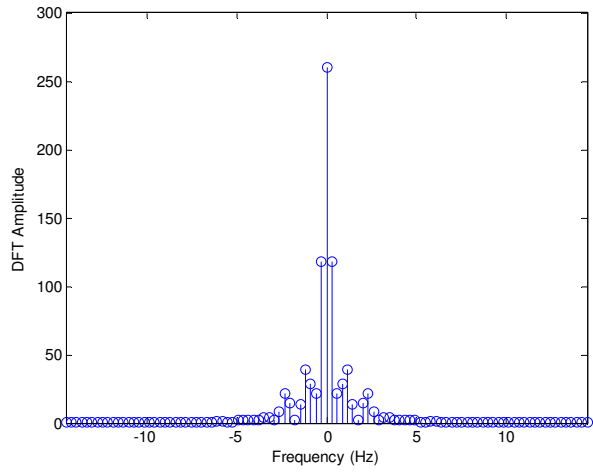
(a)
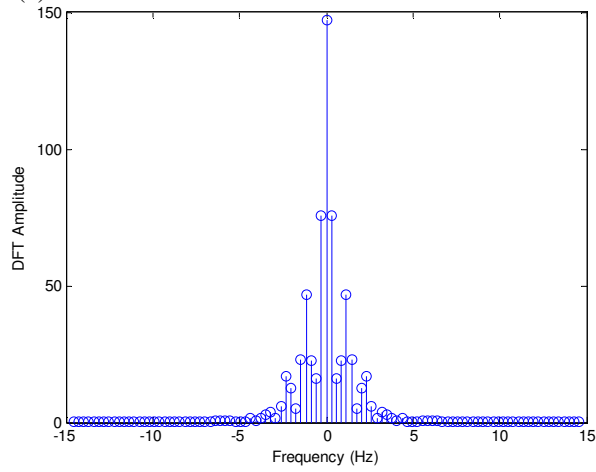
(b)
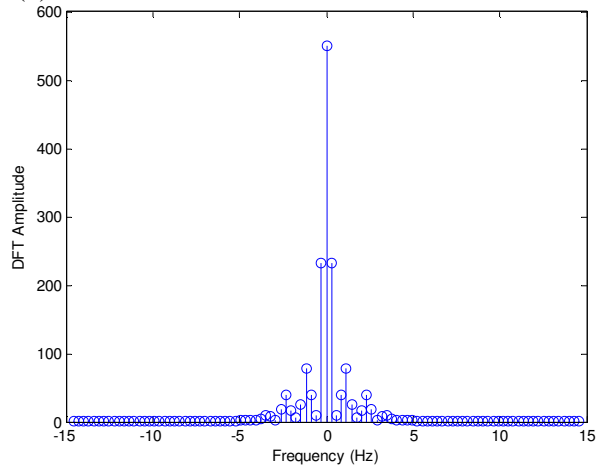
(c)

Figure 5.12. Windowed DFT amplitudes of the three selected points on Figure 5.11a. (a) Windowed DFT amplitude of point 'x'. (b) Windowed DFT amplitude of point '+'. (c) Windowed DFT amplitude of point '*'. A hamming window is used for windowing the trajectory amplitude.

150

(a)

(b) Axial Motion

(c) Axial Motion

(d) Radial Motion

(e) Radial Motion

Figure 5.13. Reconstructed trajectories of three plaque points using Horn's and Lucas's techniques. (a) Plaque points. (b) Horn axial motion. (c) Lucas axial motion. (d) Horn radial motion. (e) Lucas radial motion. Similar trajectories produced with two different methods indicate that the correct pixel velocities are being calculated.
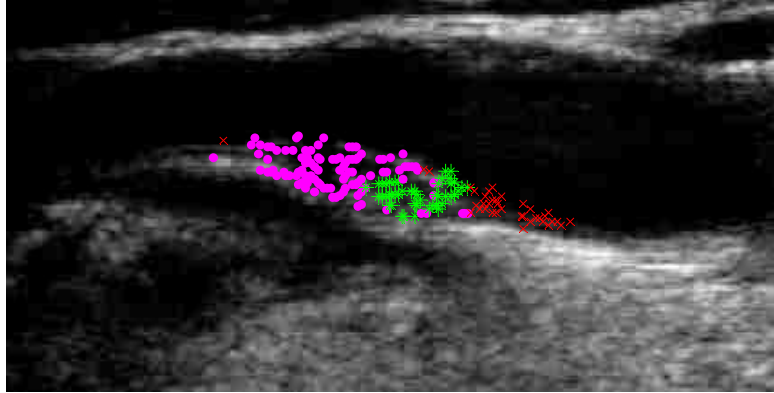
Figure 5.14. Regions of clinical video #2 that exhibit different motion patterns. Identification of these patterns is done with PCA and a k-means clustering algorithm.

PCA is also performed on the trajectories to identify motion patterns over different plaque regions which can help to better understanding of how the plaque moves during the cardiac cycle especially at systole and diastole. For video # 2, PCA retained four principal components that account for 90% of the total variance of the trajectories. The coefficient vector $\mathbf{b_i}$ was calculated according to equation (5.10), and the k-means algorithm clustered the motions in three different clusters. Figure 5.14 shows a frame of this video with the plaque region separated into clusters according to their motion patterns. The motion of this video shows three noticeable patterns. The left part of the plaque is mainly dominated by axial motion. The right part is dominated by movement in the lateral direction. The motion of the middle section is affected by speckle noise and changes in image intensities created a distinctive motion that was recognized by the clustering algorithm.

5.4.5   No-reference motion estimation results

The no-reference approach was run first on the simulated videos and then on clinical cases. For the simulations, the range of parameters are $\alpha = 1.2 - 1.7$ and

$\sigma = 1.0 - 1.25$ pixels for Horn's technique, and $\sigma = 1.25$ pixels for Lucas's technique. These values are within the range found with the full-reference optimization, section 5.4.1. Figure 5.15 shows the distribution of the error for two simulations. The smallest error is for the simulation with low motion amplitudes, and the bigger error belongs to the non-smooth velocity field with low stenosis.

Figure 5.16 shows the difference-average plots for clinical video # 1. Notice the limits of agreement are inside the preferred value of 0.15 pixels. The agreement in the estimation of the axial motion component, figure 5.16(b), is higher than the agreement of the lateral component. This suggests that axial velocities are more reliable than lateral ones. Here note that the ultrasound beam travels in the axial direction. Thus, higher accuracy is expected when estimating this component. This fact has been noticed in other ultrasound motion studies [75, 76]. A solution to improve the agreement in the lateral motion component and potentially its accuracy is to only optimize for this component. In the no-reference results section, it was noted that the same motion parameters did not produce the same level of accuracy for both components.

The confidence map, figure 5.17, shows that there is high confidence for the estimates of the plaque region.

The parameters returned by the no-reference optimization algorithm for video # 1 were $\alpha = 9.2$ and $\sigma = 0.75$ pixels for Horn's technique. Table 5.10 summarizes the parameter values found in five clinical videos. These values are close to the ones used when estimating the clinical motion using the full-reference approach, see table 5.9, especially the value of the regularization parameter.
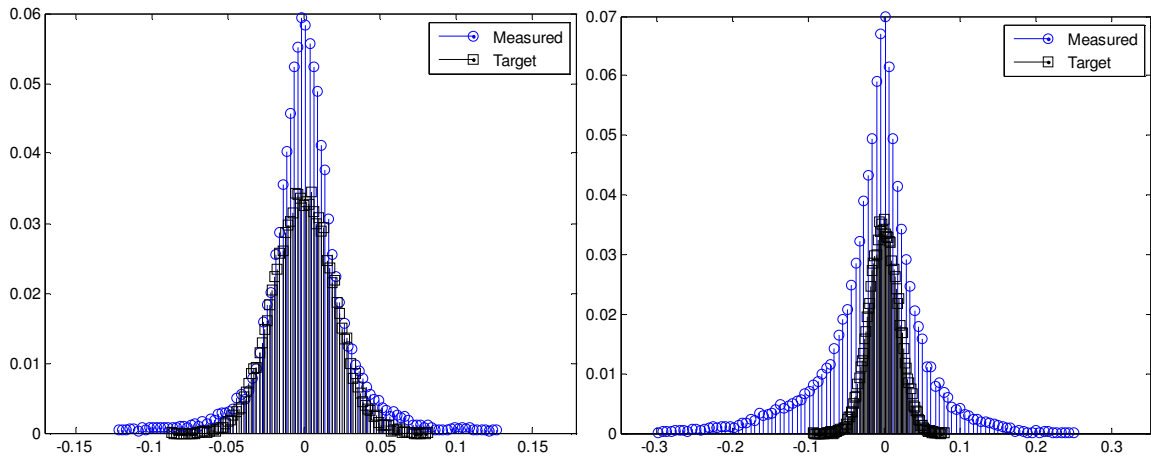
Figure 5.15. Measured difference distributions. (a) Simulation # 1. (b) Simulation # 3.
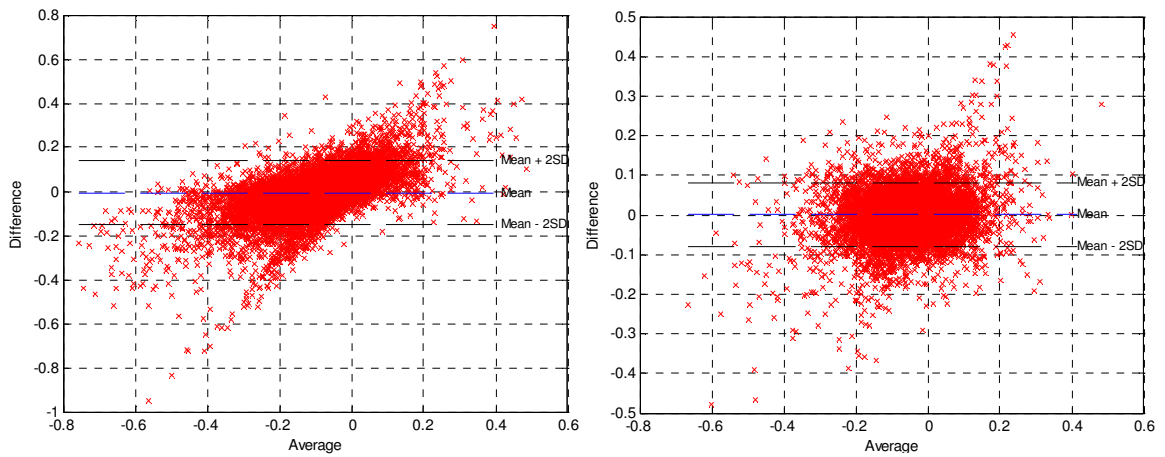

Figure 5.16. Difference-average plots for clinical video #1. (a) Horizontal motion component. (b) Vertical motion component.
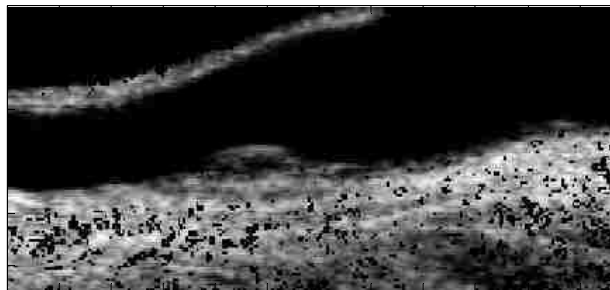

Figure 5.17. Confidence map of clinical video # 1. There is high confidence in the estimation of plaque velocities.

Table 5.10. No-reference parameter values used to estimate clinical ultrasound video motion for Horn's method.

	Parameter Values	
Video	α	σ
1	9.23	0.75
2	1.57	2.5
3	1.1	2.38
4	1.38	2.38
7	1.38	2.5

5.4.6 Differentiation filter results

Mean square errors of the velocity gradient field estimation are calculated for three different derivative filters. The simulated deterministic velocity field is a chirp signal that linearly sweeps from low to high frequencies (see table 5.2). The goal is to find the filter that produces the smallest MSE of the gradient field estimation. For the Gaussian filters, optimization was performed in order to find the best value of the filter spread that produces the most accurate estimates while suppressing noise levels of different SNR.

Low frequency results for the four point finite differencing filter with 1-D Gaussian pre-smoother are shown in figure 5.18. Results for the high frequency range are presented in figure 5.19.

MSE plots for the 2D Gaussian derivative filter are shown in figure 5.20 for the low frequencies, and figure 5.21 for the high frequencies. Table 5.11 shows the optimal filter spreads of both Gaussian filters at different noise levels.

The optimal filter spreads of the Gaussian filters were used to estimate the total MSE in the calculation of the velocity gradient field. The performance of the nine-tap Simoncelli derivative filter was also compared with the errors of the Gaussian filters in order to select the best derivative filter. The results for the low frequency derivative calculation at different noise levels for the three filters are shown from figure 5.22 to figure 5.26. High frequency derivative calculation results are shown from figure 5.27 to figure 5.31.

The filter that produced the smallest error in the calculation of the velocity gradient field among all noise levels was the 2D-Gaussian filter. When calculating the plaque strain tensor, the spread of the 2D Gaussian filter needs to set according to the accuracy of the velocity estimates. The noise levels simulated in this section, accounted for inaccurate velocity estimates. A filter spread of 1 pixel provides a good compromise between noisy and accurate velocities. Thus, this value is used in the calculation of the strain tensor.

Table 5.11. Optimal spreads of Gaussian filters for different noise levels.

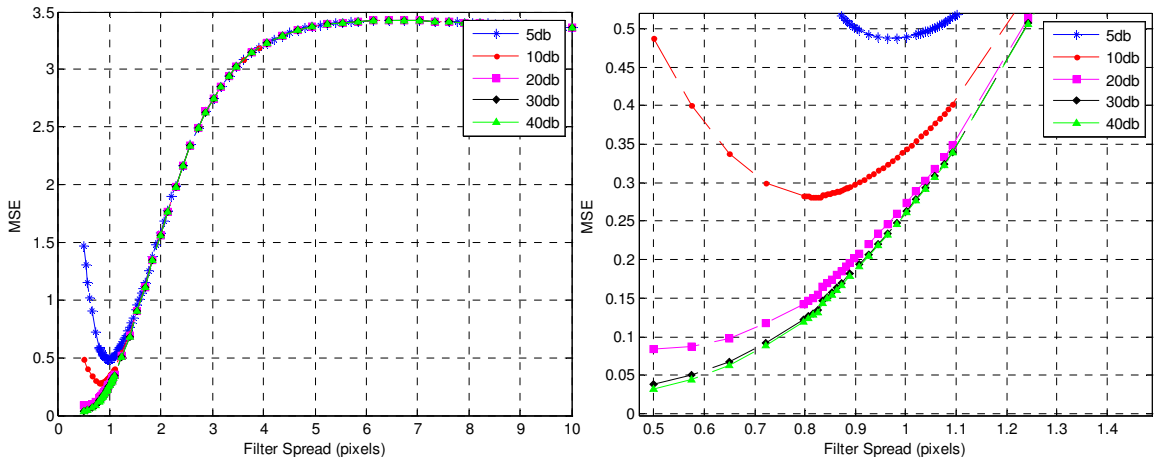|	Low Frequencies		High Frequencies	
	4-pts. 1D-G	2D-G	4-pts. 1D-G	2D-G
5db	1.85 pixels	1.88 pixels	0.91 pixels	0.96 pixels
10db	1.57 pixels	1.60 pixels	0.72 pixels	0.83 pixels
20db	1.17 pixels	1.16 pixels	0.5 pixels	0.5 pixels
30db	0.83 pixels	0.84 pixels	0.5 pixels	0.5 pixels
40db	0.54 pixels	0.57 pixels	0.5 pixels	0.5 pixels
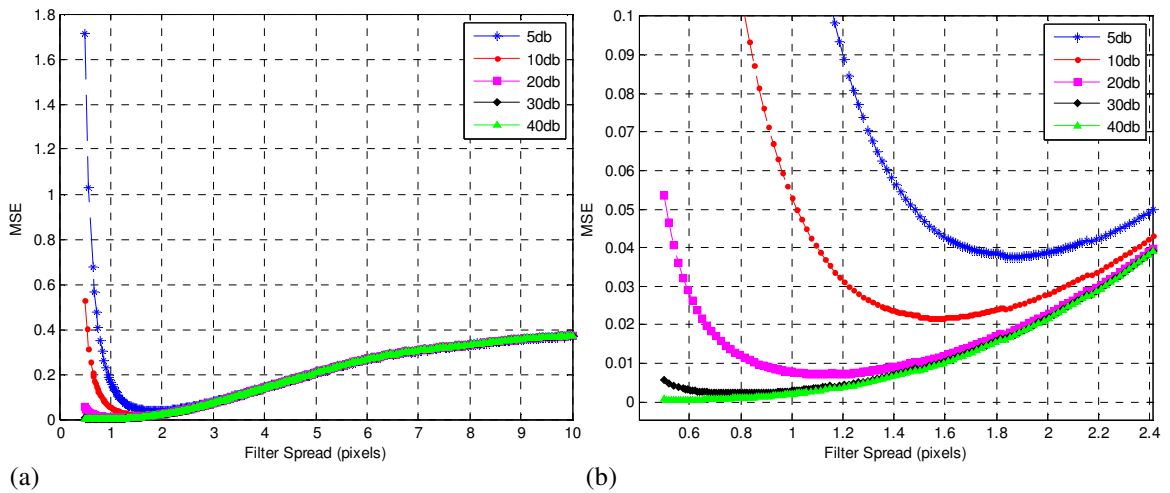
(a)   (b)

Figure 5.18. Mean Square Error (MSE) of the derivative estimation at low frequencies using a four point finite difference mask with a 1-D Gaussian pre-smoother filter. (a) MSE plot. The filter spread varied from 0.5-10 pixels. (b) Closer view of the MSE at the region where the minimum MSE was obtained.
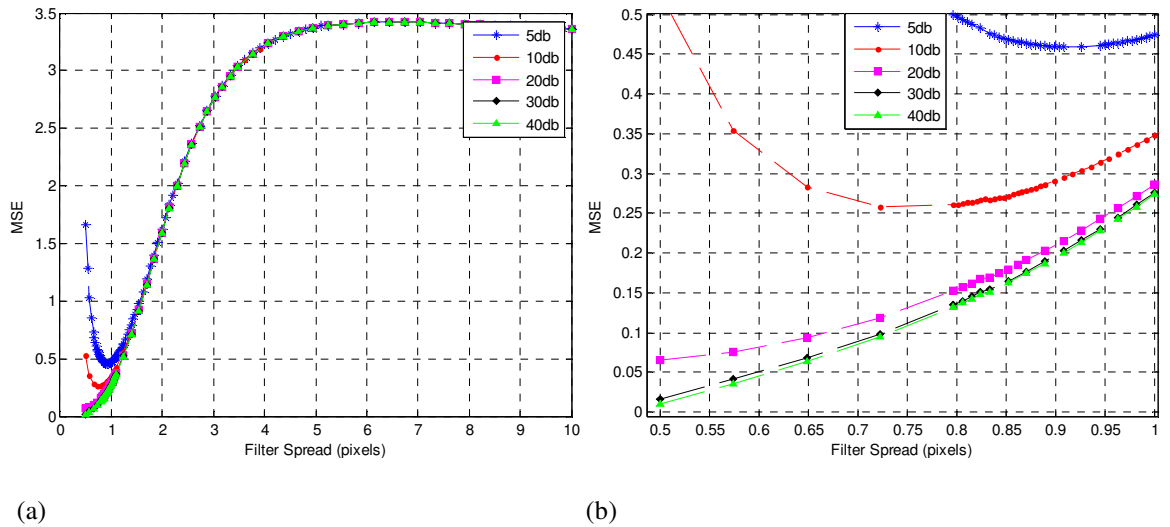

(a)   (b)

Figure 5.19. Mean Square Error (MSE) of the derivative estimation for high frequencies using a four point finite difference mask with a 1-D Gaussian pre-smoother filter. (a) MSE plot. The filter spread varied from 0.5-10 pixels. (b) Closer view of the MSE at the region where the minimum MSE was obtained.
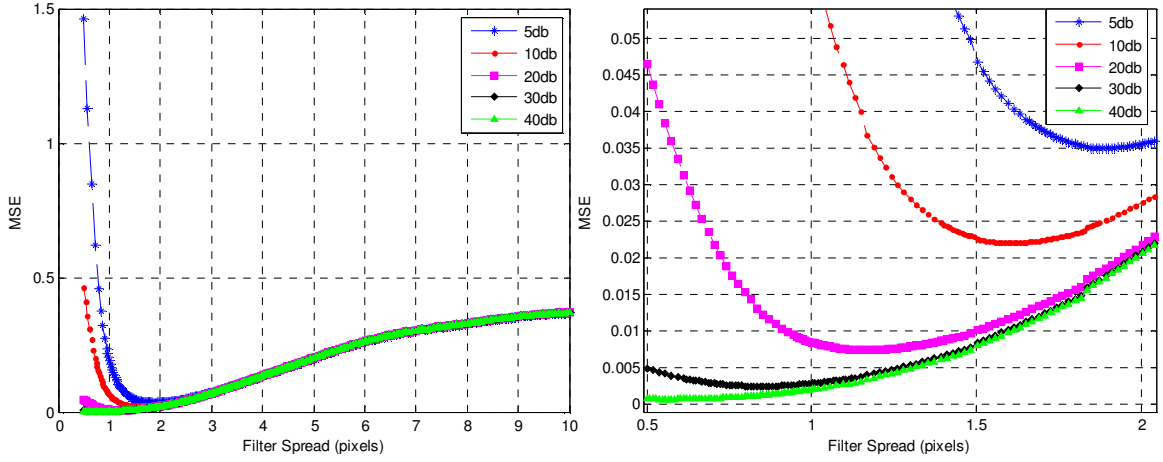
(a)   (b)

Figure 5.20. Mean Square Error (MSE) of the derivative estimation for low frequencies using a 2-D Gaussian derivative filter. (a) MSE plot. The filter spread varied from 0.5-10 pixels in both directions. (b) Closer view of the MSE at the region where the minimum MSE was obtained.

(a)   (b)

Figure 5.21. Mean Square Error (MSE) of the derivative estimation for high frequencies using a 2-D Gaussian derivative filter. (a) MSE plot. The filter spread varied from 0.5-10 pixels in both directions. (b) Closer view of the MSE at the region where the minimum MSE was obtained.
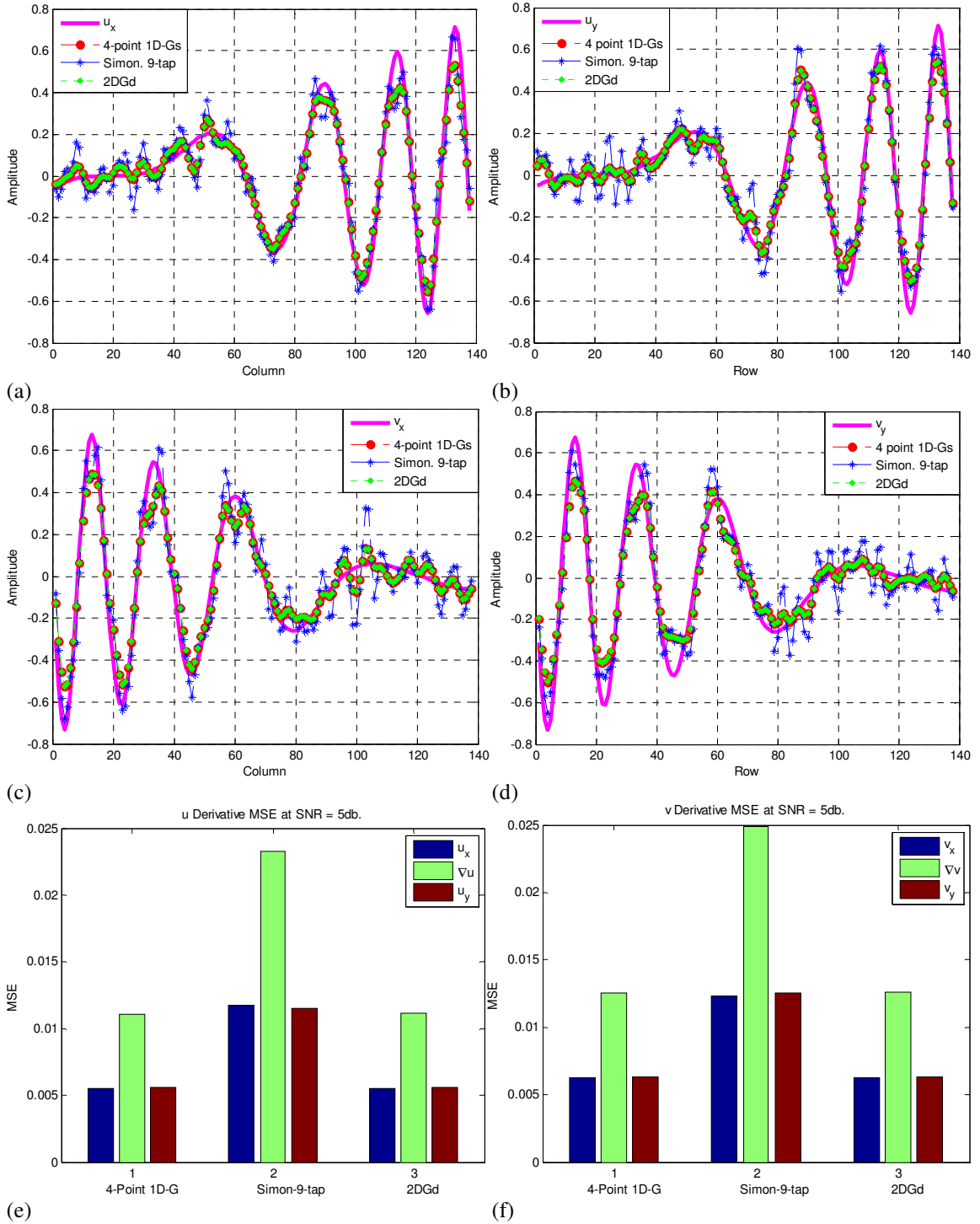
Figure 5.22. Low frequency derivative estimation at SNR = 5dB. (a)-(b) Derivatives of the horizontal component of the velocity field. (c)-(d) Derivatives of the vertical component of the velocity field. (e)-(f) MSE of the velocity magnitude error for derivative estimation.

Figure 5.23. Low frequency derivative estimation at SNR = 10dB. (a)-(b) Derivatives of the horizontal component of the velocity field. (c)-(d) Derivatives of the vertical component of the velocity field. (e)-(f) MSE of the velocity magnitude error for derivative estimation.

160

Figure 5.24. Low frequency derivative estimation at SNR = 20dB. (a)-(b) Derivatives of the horizontal component of the velocity field. (c)-(d) Derivatives of the vertical component of the velocity field. (e)-(f) MSE of the velocity magnitude error for derivative estimation.
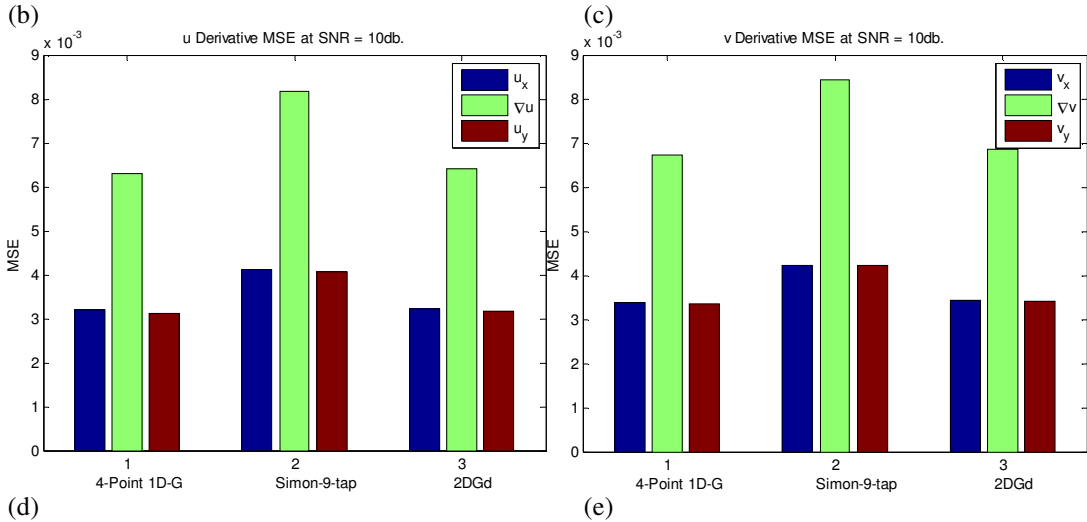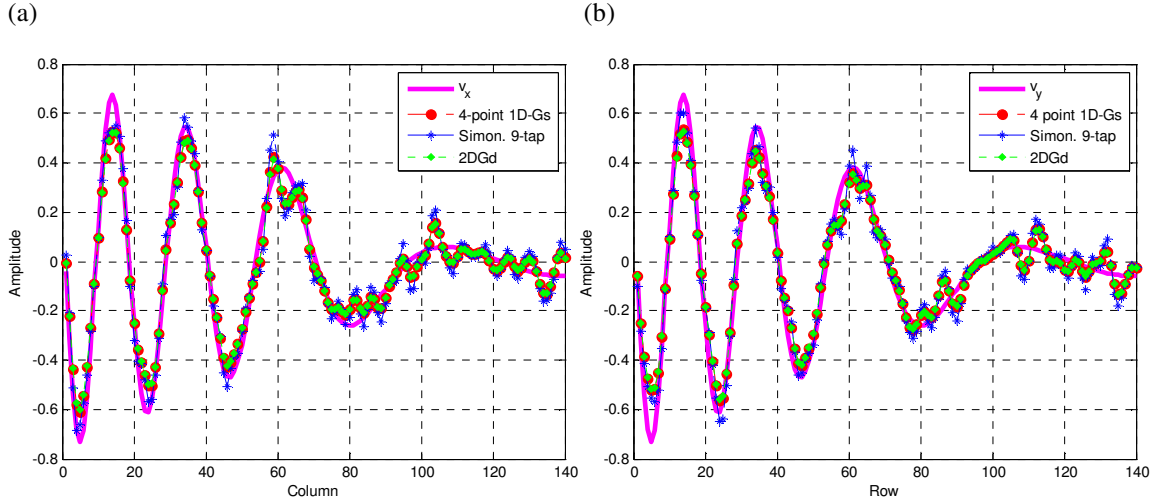
161

(a)

(b)

(c)

(d)

(d)

(e)

Figure 5.25. Low frequency derivative estimation at SNR = 30dB. (a)-(b) Derivatives of the horizontal component of the velocity field. (c)-(d) Derivatives of the vertical component of the velocity field. (e)-(f) MSE of the velocity magnitude error for derivative estimation.
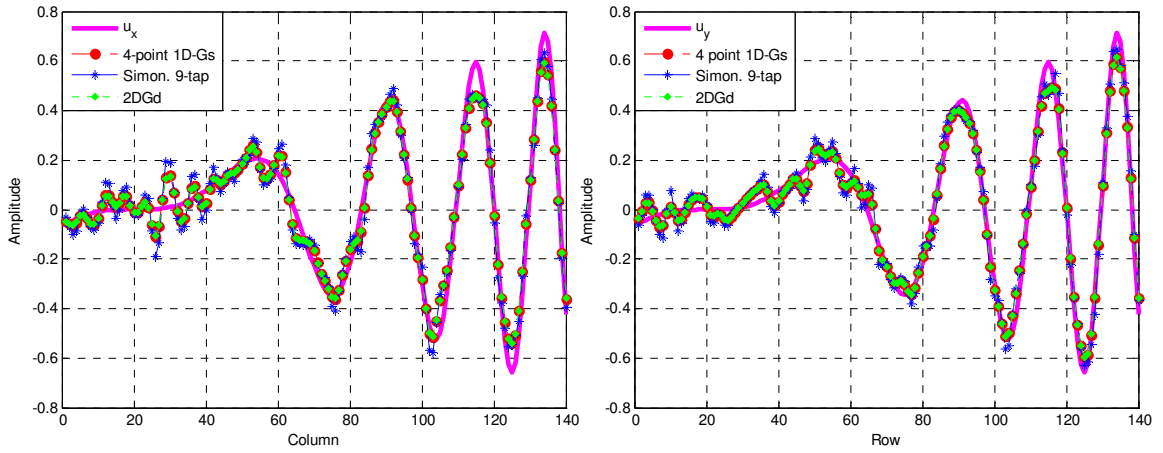
162

Figure 5.26. Low frequency derivative estimation at SNR = 40dB. (a)-(b) Derivatives of the horizontal component of the velocity field. (c)-(d) Derivatives of the vertical component of the velocity field. (e)-(f) MSE of the velocity magnitude error for derivative estimation.
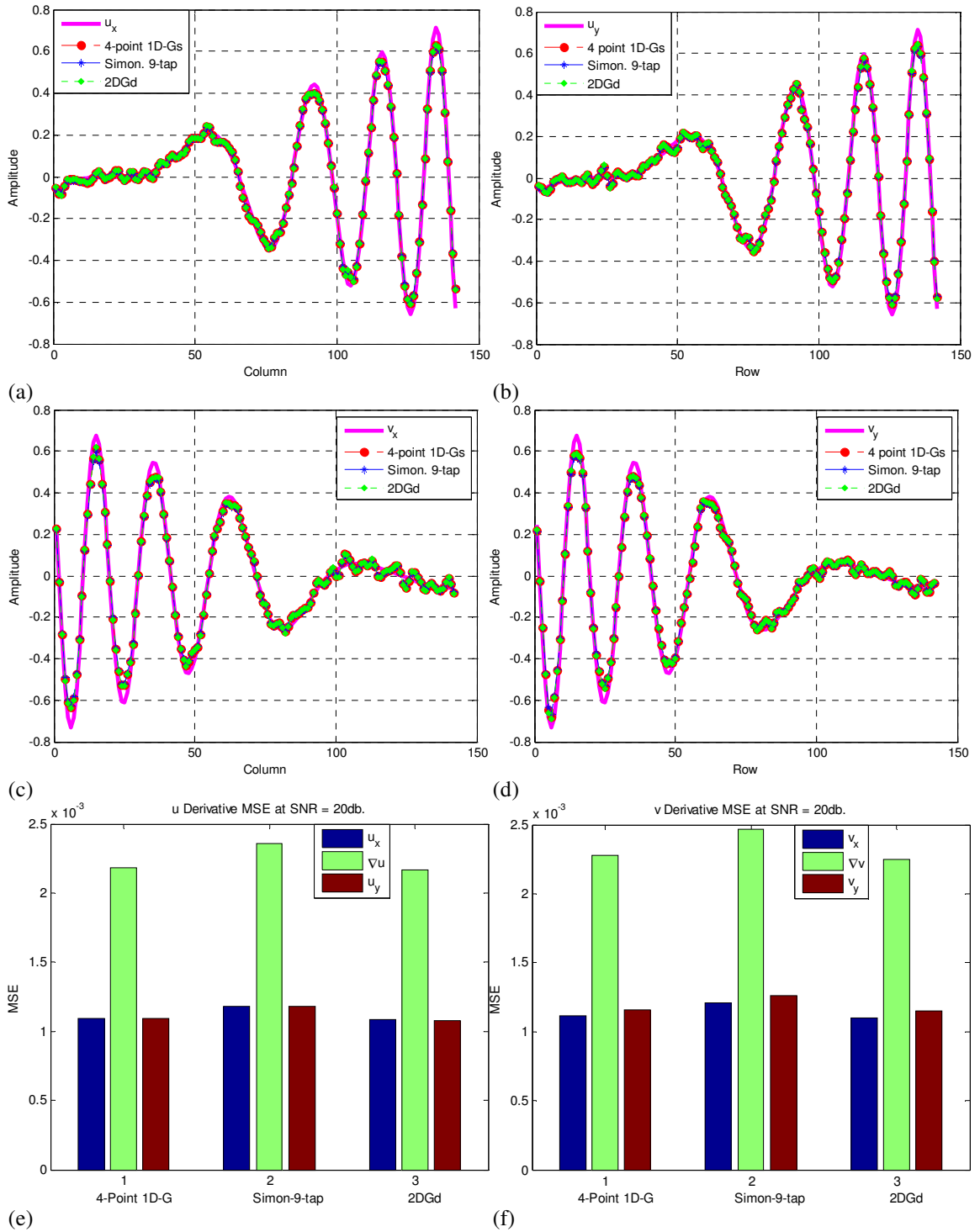
163
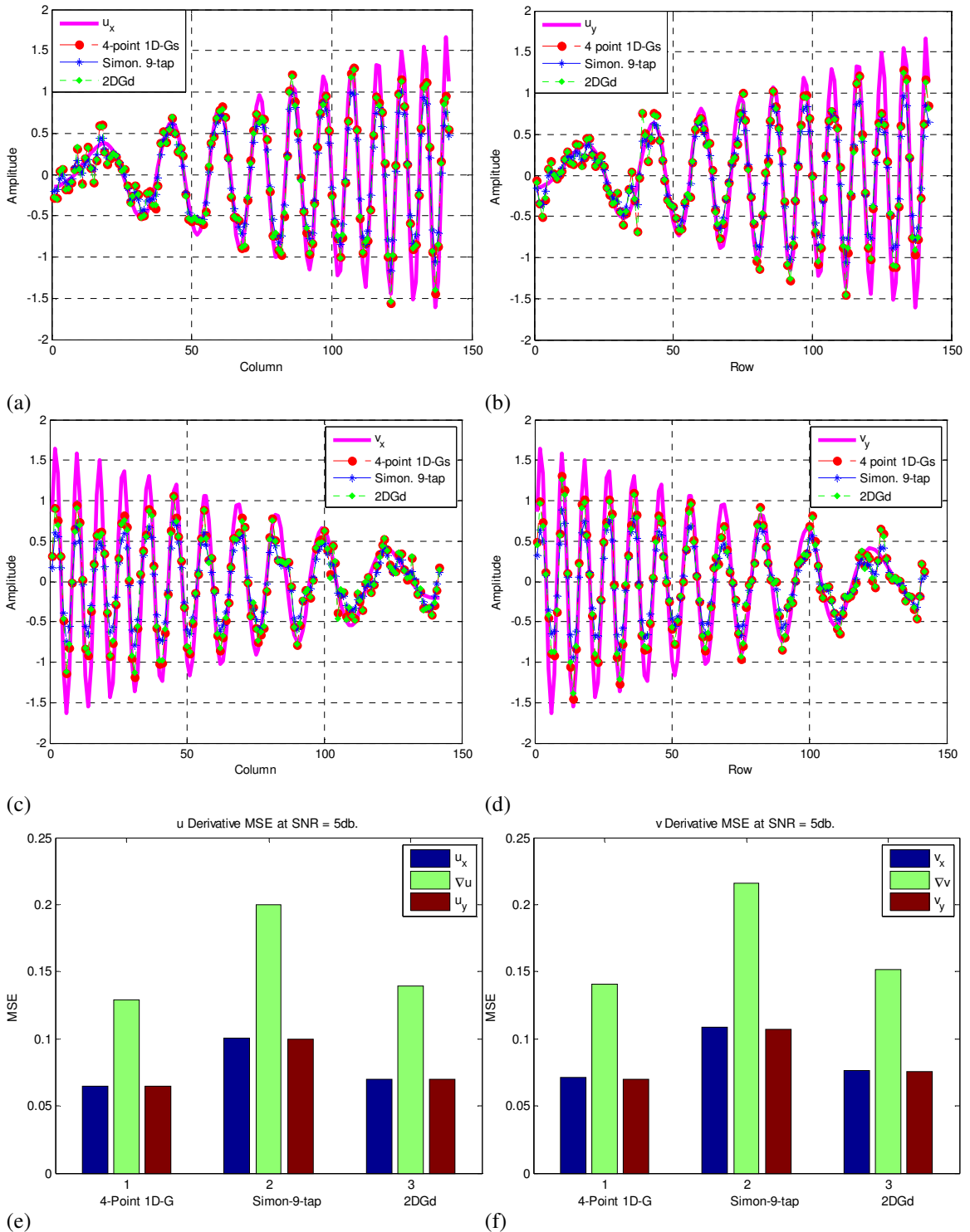
Figure 5.27. High frequency derivative estimation at SNR = 5dB. (a)-(b) Derivatives of the horizontal component of the velocity field. (c)-(d) Derivatives of the vertical component of the velocity field. (e)-(f) MSE of the velocity magnitude error for derivative estimation.
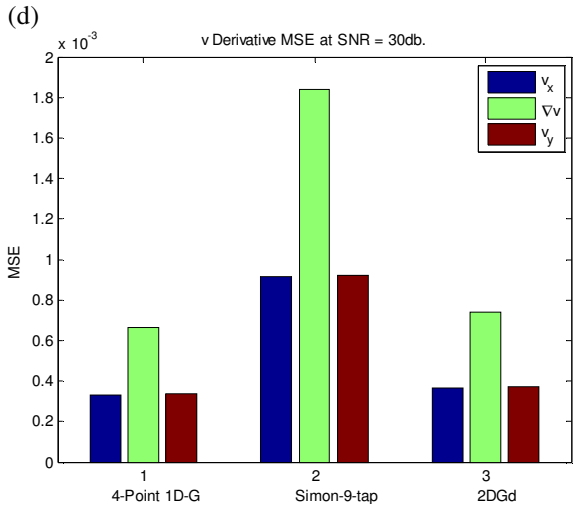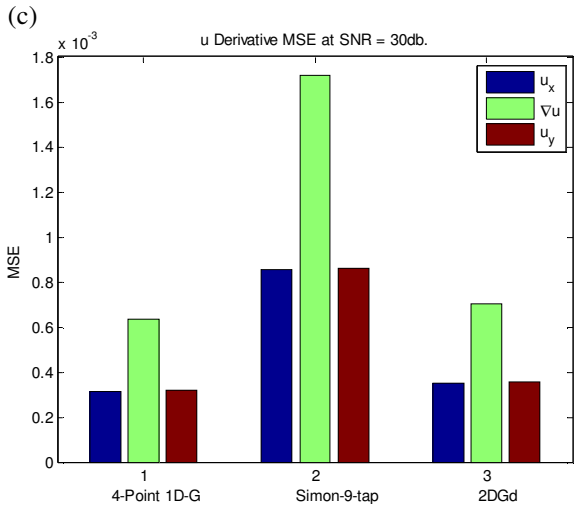
164
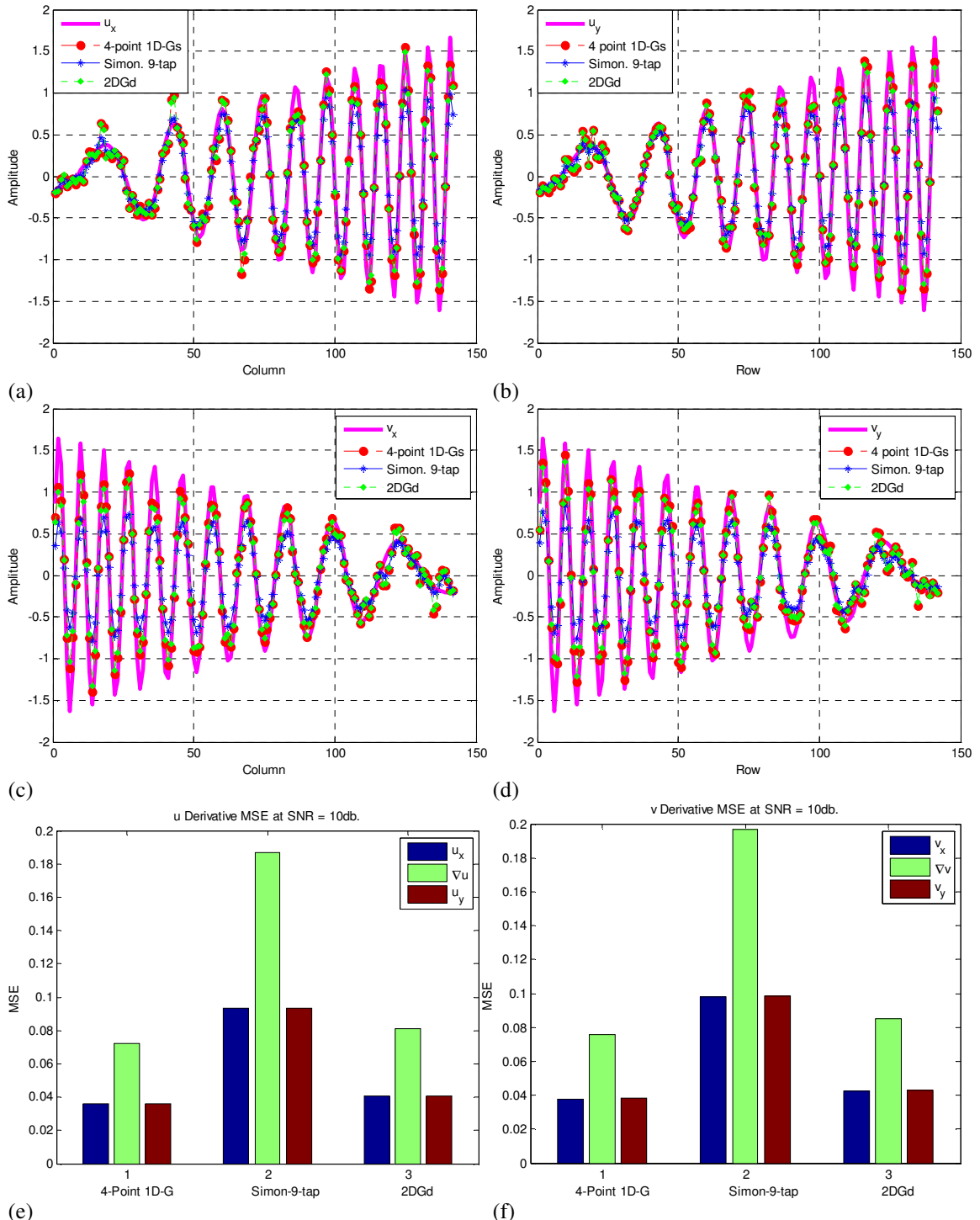
Figure 5.28. High frequency derivative estimation at SNR = 10dB. (a)-(b) Derivatives of the horizontal component of the velocity field. (c)-(d) Derivatives of the vertical component of the velocity field. (e)-(f) MSE of the velocity magnitude error for derivative estimation.
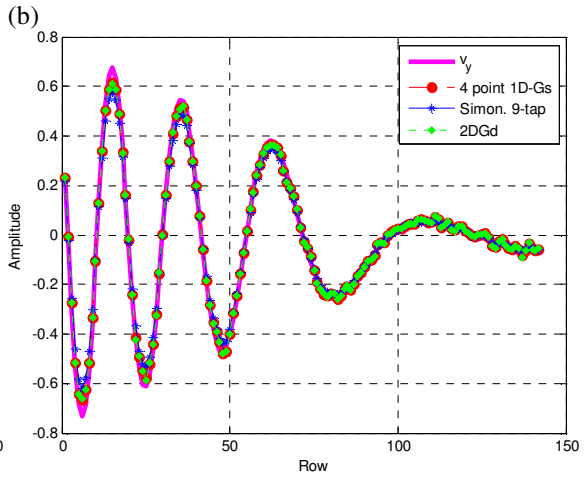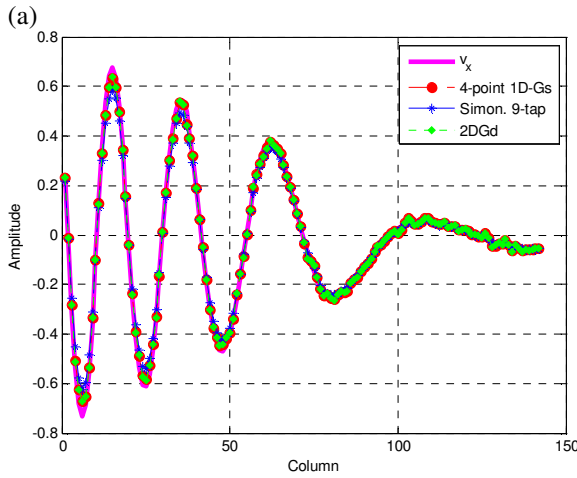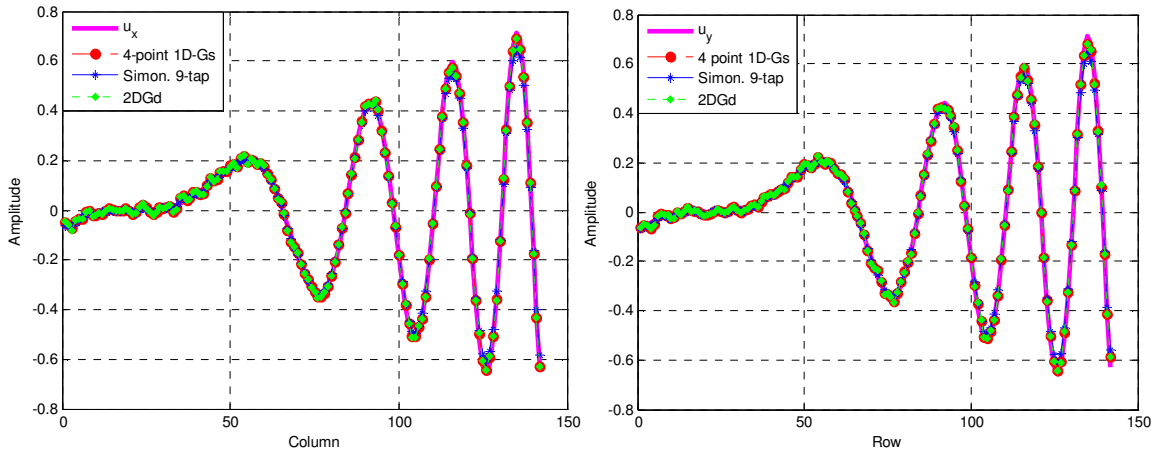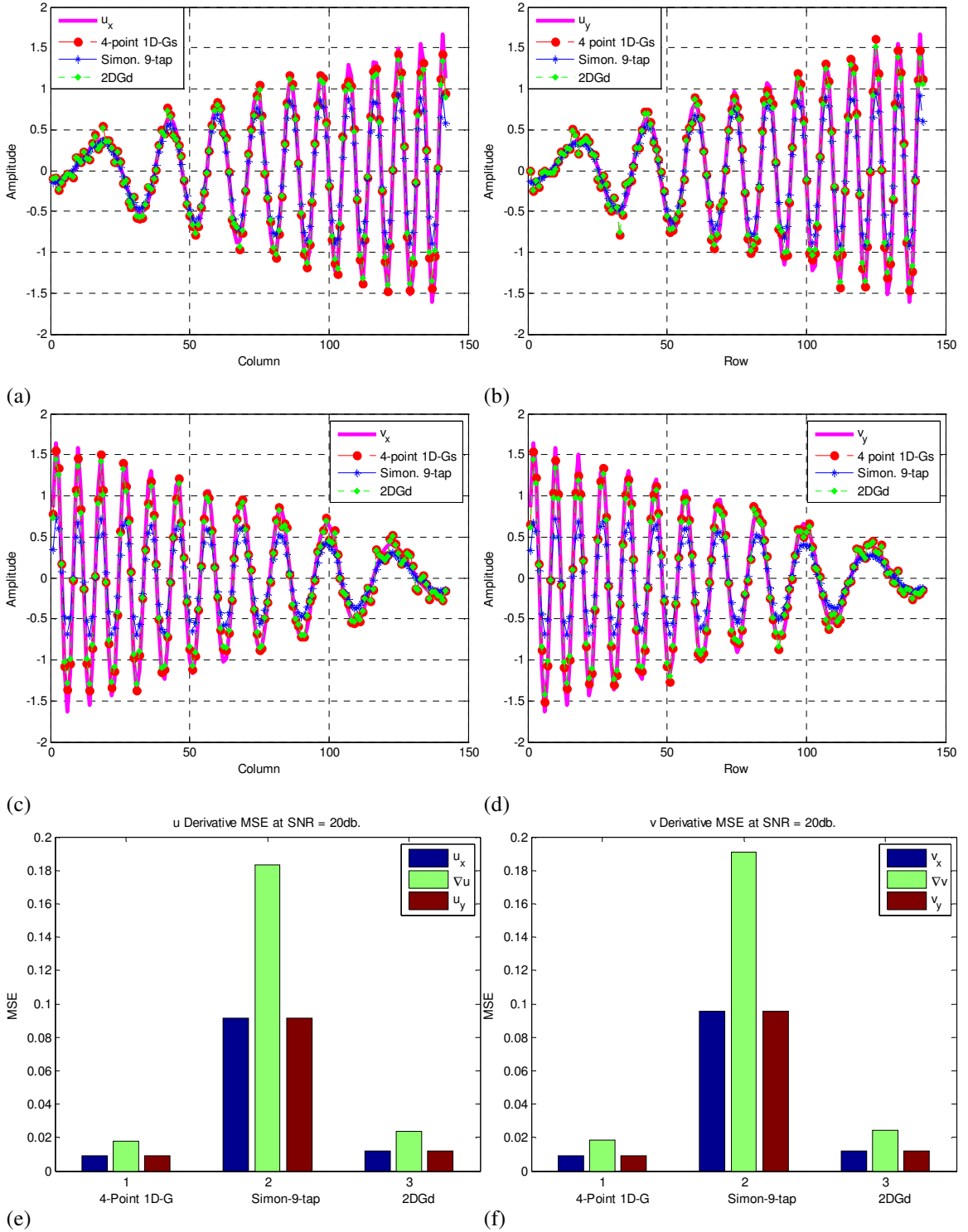
Figure 5.29. High frequency derivative estimation at SNR = 20dB. (a)-(b) Derivatives of the horizontal component of the velocity field. (c)-(d) Derivatives of the vertical component of the velocity field. (e)-(f) MSE of the velocity magnitude error for derivative estimation.

166

(a)   (b)
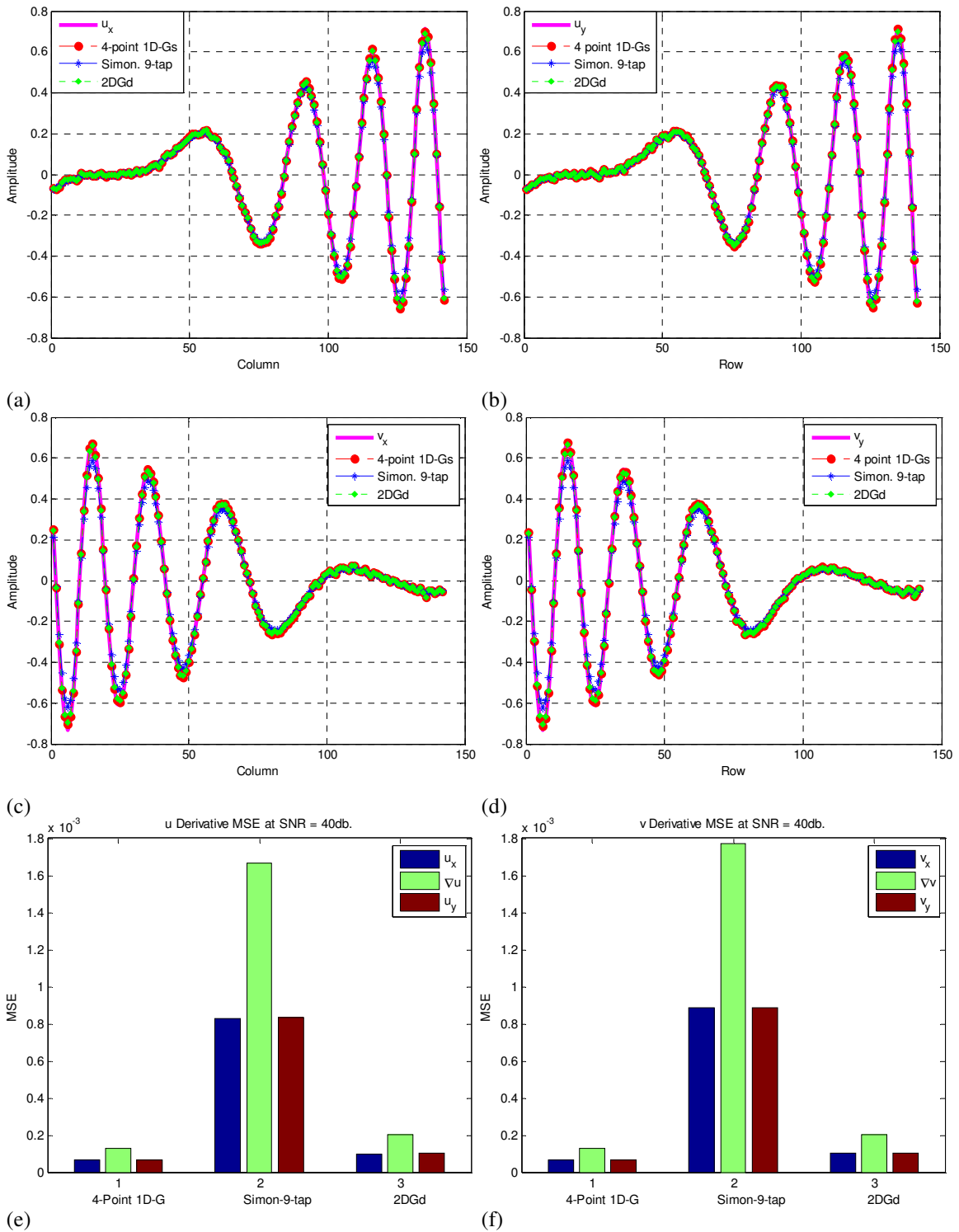
(c)   (d)
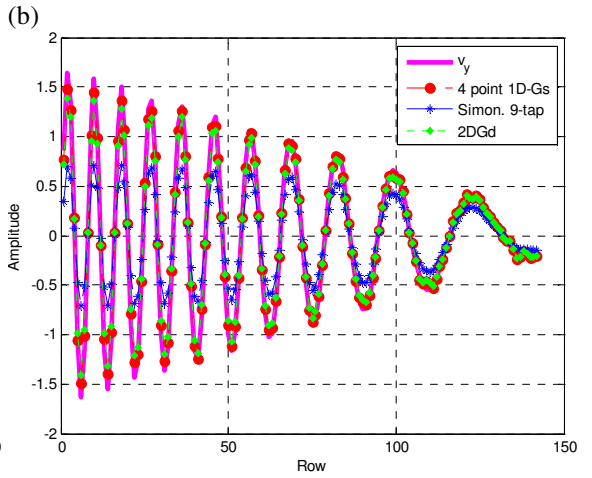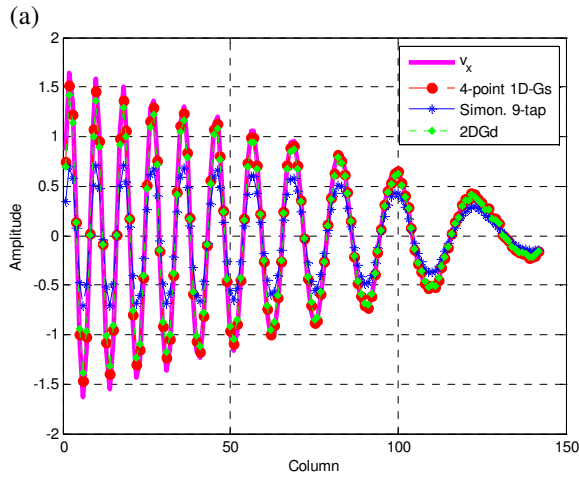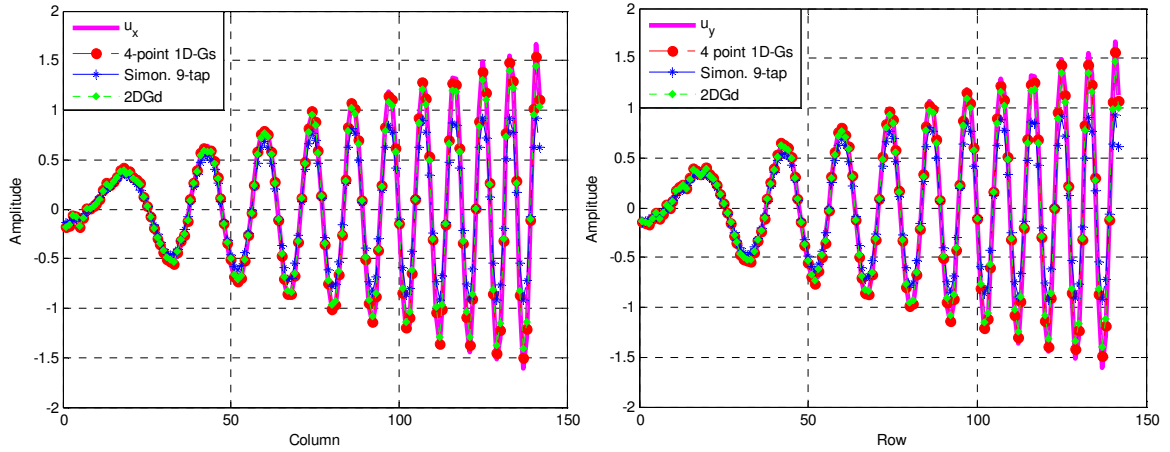
(e)   (f)

Figure 5.30. High frequency derivative estimation at SNR = 30dB. (a)-(b) Derivatives of the horizontal component of the velocity field. (c)-(d) Derivatives of the vertical component of the velocity field. (e)-(f) MSE of the velocity magnitude error for derivative estimation.

167
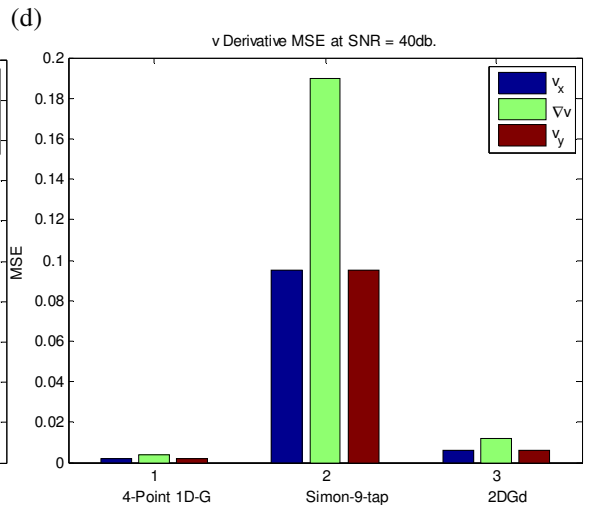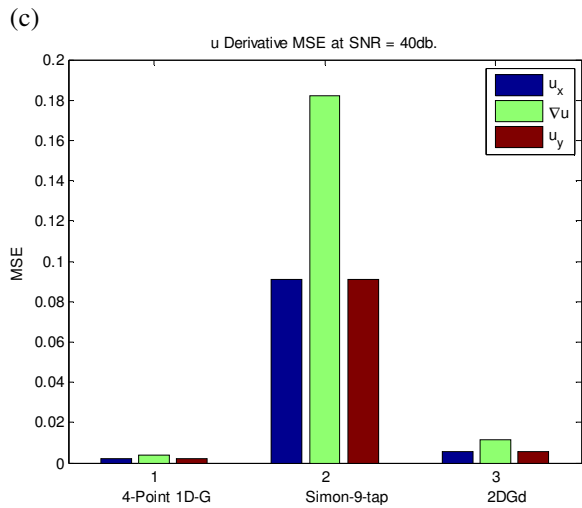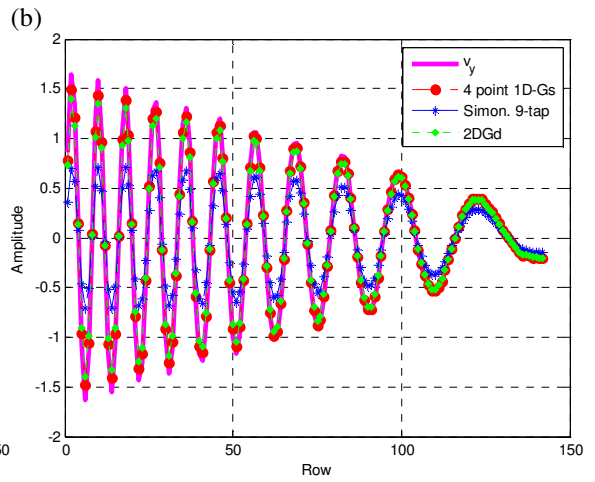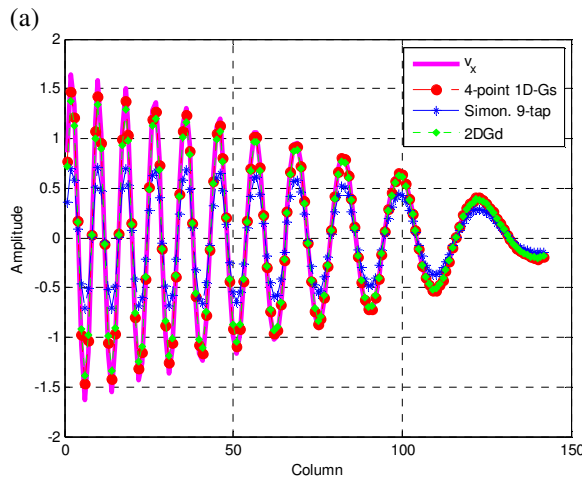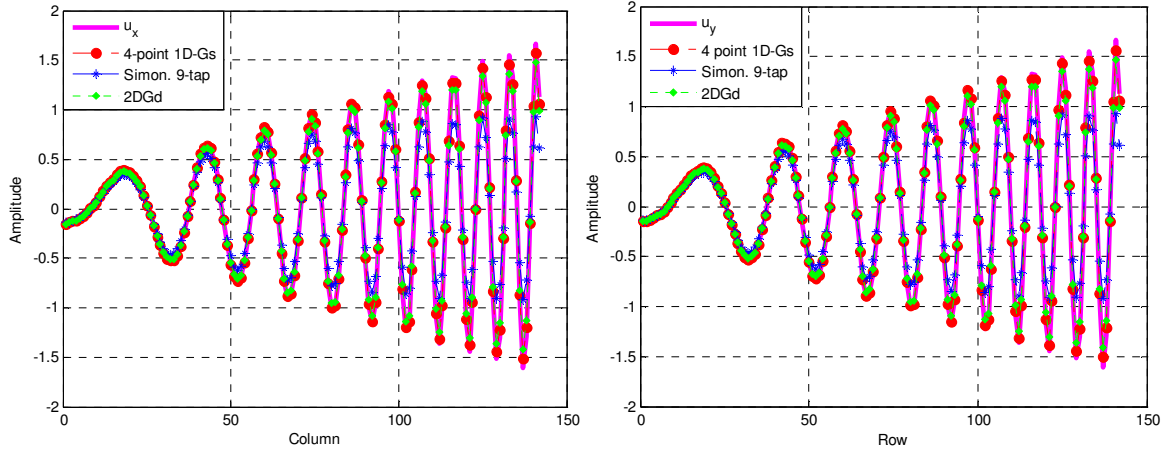
Figure 5.31. High frequency derivative estimation at SNR = 40dB. (a)-(b) Derivatives of the horizontal component of the velocity field. (c)-(d) Derivatives of the vertical component of the velocity field. (e)-(f) MSE of the velocity magnitude error for derivative estimation.
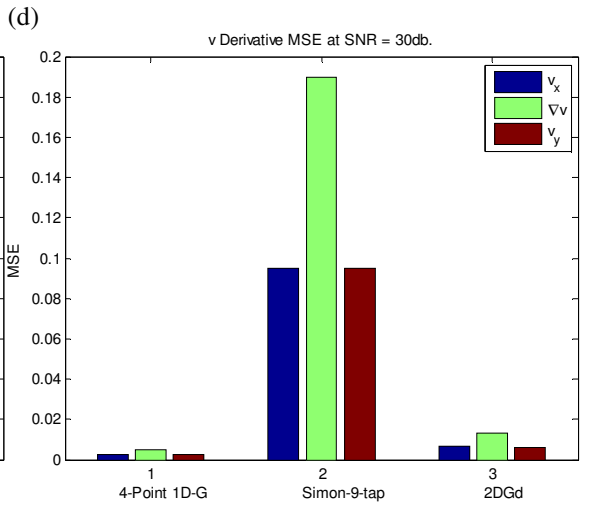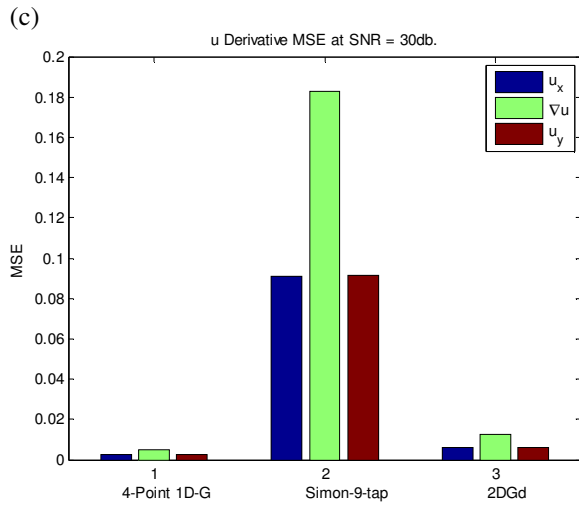
168

5.4.7 Strain measurements on clinical ultrasound videos

Accurate velocity estimates provide not only information regarding the motion but also can be used to analyze plaque deformation. The derivatives of the pixel velocities create a 2D tensor that enables the study of the elastic behavior of the underlying tissue. The tensor can be potentially used to identify plaque materials like lipid deposits or calcifications based on their mechanical properties. The total strain of a point is measured by the trace of tensor with positive values indicating stretching and negative ones indicating contraction.

The principal axes of deformation will be shown over the plaque. They are given by the eigenvectors of the strain tensor, and point in the directions of pure extensions or contractions. Thus, for each pixel with an associated strain tensor, the principal deformation axes can be visualized as an ellipsoid with axis length proportional to the magnitude of the eigenvalues. The shape of the ellipsoid, governed by the ratio of the eigenvalues, determines how a point is deforming. When the deformation is concentrated along a single direction, the thinner and line-like the corresponding ellipsoid is. On the other hand, when the deformation is equally distributed along both axes, the ellipsoid resembles a circle.

The positive bright areas of figure 5.32(b) indicate stretching, and the dark negative values compression. The strain map indicates that the middle section of this plaque is expanding while there is compression on the top plaque boundaries.

The axes of deformation for video # 7 are shown in figure 5.33. The zoomed in section analyzes the left-bottom region of the plaque.
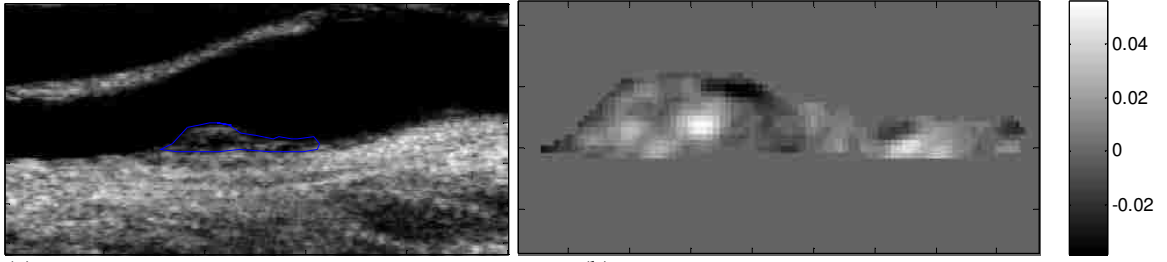
(a) (b)

Figure 5.32. Total strain of an atherosclerotic plaque. (a) Plaque delineation. (b) The total strain is given by the trace of the strain tensor.
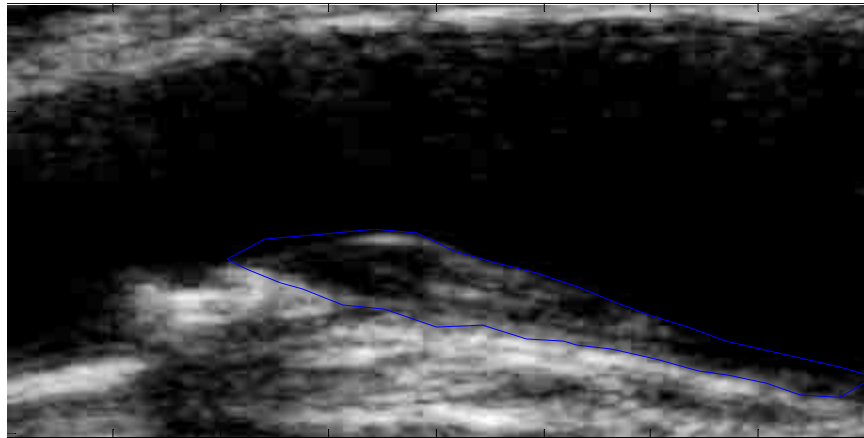
(a)
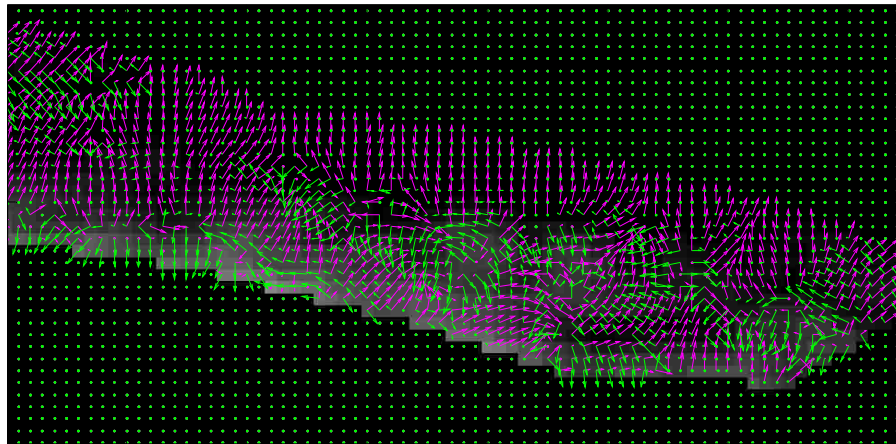
(b)

Figure 5.33. Plaque principal axis of deformation. (a) Plaque delineation. (b) The vectors point in the direction of the deformation.

170

Note that mostly, there is only one axis. This indicates that the deformation is concentrated in one direction only. When there are two arrows, the deformation is distributed on both directions.

Strain measurements were calculated for video # 1 using the no-reference parameters. The total strain of the plaque in video #1 is shown in figure 5.34. The inner part of the plaque is expanding while the boundaries show compression. These results are in line with those obtained with the full-reference parameters.
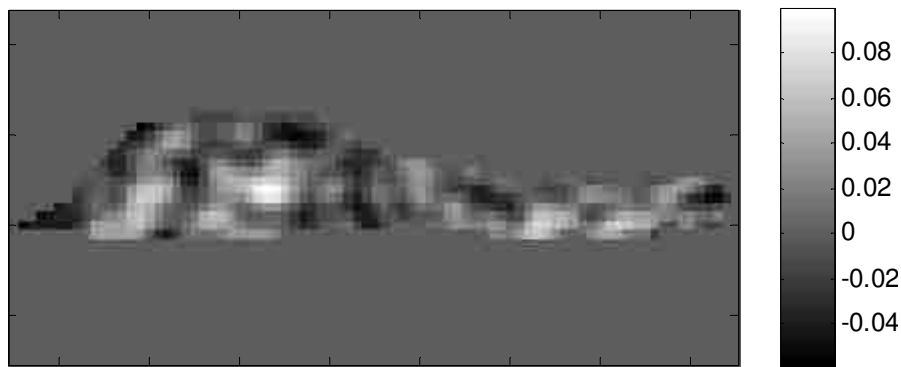
Figure 5.34. Total strain of the atherosclerotic plaque on clinical video #1.

5.5 Concluding Remarks

Motion analysis of carotid artery plaques provides us with a tool to help differentiate between stable and vulnerable plaques that could lead to stroke. The accuracy of the pixel estimates are related to the selection of the motion parameters.

The developed motion simulator is a first step to help find the range of parameters that can be used on clinical videos. The simulated motion needs to resemble the motion present in the clinical videos, so that the parameter values can produce accurate estimates.

The speckle simulation provides insight regarding how the parameters can to be adjusted when considerable levels of speckle affect the videos.

When estimating velocities of clinical videos it is important to obtain high density fields because the more dense the estimated field, the more information can be assessed from the video. Thus, global techniques are preferred over local methods because they produce higher density fields. Local methods can be used to measure the agreement of the estimation in the no-reference approach. This is an extra layer of validation that provides confidence in the estimation of pixel velocities.

Chapter 6

Conclusions and Future Work

6.1 Conclusions

The methods presented in this dissertation have demonstrated that the developed optimization methods lead to improved performance of motion estimation techniques. The full-reference and no-reference approaches find motion parameters that produce more accurate velocity estimates than the fields estimated with current heuristic choices of the parameters.

The agreement between the full-reference and no-reference optimization results demonstrates that accurate velocity estimation is possible when ground truth is not available. Thus, the no-reference global optimization approach can be applied to any video sequence expecting estimates that will be of accuracy that is comparative to full-reference optimization.

For real video sequences with unknown velocity fields, the detection of unreliable velocity estimates can be done with a novel confidence map that grades the estimates based on their expected accuracy rather than on ad-hoc metrics that do not correlate with the properties of the velocity field.

The no-reference optimization approach has been shown to provide with reliable velocity estimates that can be obtained by maximizing the agreement in the estimation of different motion estimation techniques. The developed optimization methods are independent of

currently available motion estimation techniques. The work of this dissertation can be applied to any existing motion estimation technique or any technique that will be developed in the future. The flexibility of the implemented optimization strategy is such that new methods that handle large displacements will benefit the most achieving superior performance than currently available techniques.

For ultrasound videos of carotid artery plaques, it has been shown that the developed optimization methods produce velocity estimates that enable the study of motion and deformation patterns that can potentially identify vulnerable plaques that lead to stroke.

6.2 Future work

Even though the global optimization algorithm used in the full-reference and no-reference approaches obtains the optimal motion parameters and minimizes the error of the velocity estimates, an expression that explicitly relates the parameters to the value of this constant, L, was not fully developed. Improved convergence and reduced number of function evaluations can be attained by avoiding the problem of significantly over-estimating L.

For carotid artery plaques, a full reference and a no-reference approach for strain estimation will need to be developed. This work will add another layer of confidence in the elastography maps to identify vulnerable plaques.

The medical validation of the motion analysis and strain estimation of carotid artery plaques still needs to be performed. A team of doctors will need to grade the cases as symptomatic and asymptomatic. After this, a classification algorithm that assigns the atherosclerotic plaques to those two categories will need to be developed. Once correct

classification is achieved, the ultimate goal is to identify the asymptomatic cases that could become symptomatic.

Appendix A

A.1 Convergence Proof of Lipschitz Global Optimization Algorithm

In order to proof convergence of the algorithm the conditions of Theorem A.1 must be fulfilled using Lemma A.2 and Proposition A.3.

Theorem A.1

If for any inifinite sequence $\{P_{k_q}\}$, $P_{K_q} \supset P_{k_{q+1}}$ $(q = 1, 2, \ldots)$, of successively refined partition sets the bounds at iterations k_q satisfy:

$$\lim_{q \to \infty} \left(\gamma_{k_q} - \mu \left(P_{k_q} \right) \right) = 0, \tag{A.1}$$

then,

$$u = \lim_{k \to \infty} \mu_k = \lim_{k \to \infty} f\left(v_k\right) = \lim_{k \to \infty} \gamma_k = \gamma \tag{A.2}$$

and every accumulation point v^* of the sequence $\{v_k\}$ is an optimal solution of $\min\{f(x) : x \in D\}$.

Lemma A.2 Successive bisection of n-rectangles at the midpoint of one of its longest edges is exhaustive.

Proof: Let $\min\{f(x) : x \in D\}$ and $\{R_q\}$, $q \geq 1$, be a decreasing sequence of rectangles, where R_{q+1} is obtained from R_q by bisection at the midpoint of one of the longest edges of R_q. Let $c_i = b_i - a_i$ $(i = 1, \ldots, n)$ be ordered such that $c_1 \geq c_i (i = 2, \ldots, n)$. Let $c = (c_1, \ldots, c_1)^T \in \mathbb{R}^n$, and consider the n-cube $C_1 = \{x : a \leq x \leq a + c\}$ and the sequence

$\{C_q\}$, $q \geq 1$ of rectangles, where C_{q+1} is obtained from C_q by bisection at the midpoint

of one of the longest edges of C_q. Clearly:

$$\delta(R_q) \leq \delta(C_q) \quad \forall q \in \mathbb{N}. \tag{A.3}$$

Moreover, after n bisections, the cube C_{n+1} with edges of length $c_1 / 2$, i.e.

$$\delta(R_{n+1}) \leq \delta(C_{n+1}) = \frac{1}{2}\delta(C_1). \tag{A.4}$$

It follows that $\delta(C_q) \underset{q \to \infty}{\to} 0$ and hence $\delta(R_q) \underset{q \to \infty}{\to} 0$ q.e.d.

Let R_k, Q_k, u_k, γ_k, and v_k stand for R, Q, u, γ, and v respectively at the beginning of

iteration k.

Proposition A.3 If algorithm 3.1 is infinite, then

$$\mu = \lim_{k \to \infty} u_k = \lim_{x \to \infty} f(v_k) = \lim_{k \to \infty} \gamma_k = \gamma, \tag{A.5}$$

and every accumulation point v^* of the sequence $\{v_k\}$ is an optimal solution of

$\min\{f(x) : x \in R\}$.

Proof: According to theorem A.1 it suffies to show that for every decreasing sequence

$R_{k_q}, R_{k_q} \supset R_{k_{q+1}}$, of successively refined rectangles the corresponding bounds satisfy

$$\lim_{q \to \infty}\left(\gamma_{k_q} - \mu\left(R_{k_q}\right)\right) = 0. \tag{A.6}$$

Let

$$R_{k_q} = \left\{x : a_{k_q} \leq x \leq b_{k_q}\right\}, \quad x_{R_{k_q}} = \left(b_{k_q} + a_{k_q}\right)/2, \tag{A.7}$$

and

$$\mu'\left(R_{k_q}\right) = \max\left\{\max\left\{f\left(a_{k_q}\right), f\left(b_{k_q}\right)\right\} - L\left\|b_{k_q} - a_{k_q}\right\|, f\left(x_{R_q} - L\left\|b_{k_q} - a_{k_q}\right\|/2\right)\right\}. \quad \text{(A.8)}$$

Moreover, let

$$\gamma\left(R_{k_q}\right) = \min\left\{f\left(x\right) : x \in Q_{k_q} \cap R_{k_q}\right\}. \quad \text{(A.9)}$$

It follows from Lemma 3.2 that $\lim\limits_{q\to\infty} R_{k_q} = \bigcap\limits_{q=1}^{\infty} R_{k_q} = \{s\}$ for some point s, and hence

$$\lim_{q\to\infty} \gamma\left(R_{k_q}\right) = \lim_{q\to\infty} \mu'\left(R_{k_q}\right) = f\left(s\right), \quad \text{(A.10)}$$

since f is continuous and $\delta\left(R_{k_q}\right) = \left\|b_{k_q} - a_{k_q}\right\| \underset{q\to\infty}{\to} 0$. Using $\gamma\left(R_{k_q}\right) \geq \gamma_{k_q}$ and

$u'\left(R_{k_q}\right) \leq u\left(R_{k_q}\right)$, then $\gamma_{k_q} - \mu\left(R_{k_q}\right) \underset{q\to\infty}{\to} 0$ holds q.e.d.

References

[1]   B. Horn and B. Schunck, "Determining Optical Flow," *Artificial Intelligence,* vol. 17, pp. 185-204, 1981.

[2]   A. Bruhn, J. Weickert, and C. Schnorr, "Combining the Advantages of Local and Global Optic Flow Methods," *Lecture notes in computer science,* vol. 2449/2002, pp. 454-462, 2002.

[3]   D. J. Fleet and A. D. Jepson, "Computation of component image velocity from local phase information," *International Journal of Computer Vision,* vol. 5, pp. 77-104, 1990.

[4]   D. J. Heeger, "Model for the Extraction of Image Flow," *Optical Society of America,* vol. 4, pp. 1455-71, August 1987.

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

[6]   B. D. Lucas and T. Kanade, "An Iterative Image Registration Technique with an Application to Stereo Vision," in *International Joint Conf. on Artificial Intelligence*, 1981, pp. 674-679.

[7]   V. Murray, S. Murillo, M. Pattichis, C. P. Loizou, C. Pattichis, A. Kyriacou, and A. Nicolaides, "An AM-FM model for Motion Estimation in Atherosclerotic Plaque Videos," in *Forty-First Asilomar Conference on Signals, Systems and Computers*, 2007, pp. 746-750.

[8]   H. H. Nagel and W. Enkelmann, "An investigation of smoothness constraints for the estimation of displacement vector fields from image sequences," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 8, pp. 565-593, 1986.

[9]   S. Negahdaripour, "Revised definition of optical flow: integration of radiometric and geometric cues for dynamic scene analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 20, pp. 961-979, 1998.

[10]   S. Uras, F. Girosi, A. Verri, and V. Torre, "A computational approach to motion perception," *Biological Cybernetics,* vol. 60, pp. 79-87, 1988.

[11]   A. M. Waxman, J. Wu, and F. Bergholm, "Convected activation profiles and the measurement of visual motion," in *Computer Society Conference on Computer Vision and Pattern Recognition*, 1988, pp. 717-723.

[12]   K. Krajsek and R. Mester, "On the equivalence of variational and statistical differential motion estimation," in *IEEE Southwest Symposium on Image Analysis and Interpretation*, 2006, pp. 11-15.

[13]   K. Krajsek and R. Mester, "A Maximum Likelihood Estimator for Choosing the Regularization Parameters in Global Optical Flow Methods," in *IEEE International Conference on Image Processing* 2006, pp. 1081-1084.

[14]   S. Murillo, M. Pattichis, C. P. Loizou, C. Pattichis, E. Kyriacou, A. Constantinides, and A. Nicolaides, "Atherosclerotic Plaque Motion Analysis from Ultrasound Videos," in *Fortieth Asilomar Conference on Signals, Systems and Computers* 2006, pp. 836-840.

[15]   L. Ng and V. Solo, "A data-driven method for choosing smoothing parameters in optical flow problems," in *International Conference on Image Processing*, 1997, pp. 360-363.

[16]   P. Anandan, "A Computational Framework and an Algorithm for the Measurement of Visual Motion," *International Journal of Computer Vision,* vol. 2, pp. 283-310, 1989.

[17]   B. Horn, *Robot Vision*: McGraw-Hill, 1986.

[18]   C. Thomas, *Elements of Information Theory*: John Wiley & Sons, 1991.

[19]   V. Willert, M. Toussaint, J. Eggert, and E. Korner, "Uncertainty optimization for robust dynamic optical flow estimation," in *Sixth International Conference on Machine Learning and Applications ICMLA*, 2007, pp. 450-457.

[20]   W. C. Karl, "Regularization in Image Restoration and Reconstruction," in *Handbook of Image and Video Processing*, A. Bovik, Ed.: Elsevier Academic Press, 2005, pp. 183-202.

[21]   C. R. Vogel, *Computational methods for inverse problems*: Society for Industrial and Applied Mathematics SIAM, 2002.

[22]   P. C. Hansen, "The use of the L-curve in the regularization of discrete ill-posed problems," *SIAM Journal of Scientific Computing,* vol. 14, pp. 1487-1503, 1993.

[23]   V. A. Morozov, "On the solution of functional equations by the method of regularization," *Soviet Mathematics Doklady,* vol. 7, pp. 414-417, 1966.

[24]   G. Wahba, "Practical approximate solutions to linear oprator equations when the data are noisy," *SIAM Journal on Numerical Analysis,* vol. 14, pp. 651-677, 1977.

[25]   P. C. Hansen, *Rank-Deficient and Discrete Ill-Posed Problems*. Philadelphia: Siam, 1998.

[26]   A. M. Tekalp, *Digital Video Processing*, 1995.

[27]   J. L. Barron, D. J. Fleet, and S. S. Beauchemin, "Performance of Optical Flow Techniques," *International Journal of Computer Vision,* vol. 12, pp. 43-77, 1994.

[28]   S. S. Beauchemin and J. L. Barron, "The computation of optical flow," *ACM Computer Surveys,* vol. 27, pp. 433-466, 1995.

[29]   S. Murillo, M. Pattichis, C. P. Loizou, C. Pattichis, E. Kyriacou, A. Constantinides, and A. Nicolaides, "Atherosclerotic Plaque Motion Trajectory Analysis from Ultrasound Videos," in *IEEE Int. Special Topic Conf. on Inform. Technology in Biomedicine*, Ioannina, Greece, 2006, pp. 1-5.

[30]   G. Strang, *Introduction to Applied Mathematics*: Wellesley-Cambridge Press 1986.

[31]   R. Courant and D. Hilbert, *Methods for Mathematical Physics* vol. I: Wiley-Interscience, 1989.

[32] C.-H. Teng, S.-H. Lai, Y.-S. Chen, and W.-H. Hsu, "Accurate Optical Flow Computation under non-uniform Brightness Variations," *Computer Vision and Image Understanding,* vol. 97, pp. 315-346, 2005.

[33] D. J. Fleet and Y. Wiess, "Optical Flow Estimation," in *Handbook of Mathematical Models in Computer Vision*, N. Paragios, Y. Chen, and O. Faugeras, Eds.: Springer, 2005, pp. 239-258.

[34] D. J. Fleet and A. D. Jepson, "Stability of phase information," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 15, pp. 1253-1268, 1993.

[35] N. Bauer, P. Pathirana, and P. Hodgson, "Robust Optical Flow with Combined Lucas-Kanade/Horn-Schunck and Automatic Neighborhood Selection," in *International Conference on Information and Automation*, 2006, pp. 378-383.

[36] G. D. Hager and P. N. Belhumeur, "Efficient Region Tracking with Parametric Models of Geometry and Illumination," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 20, pp. 1025-1039, 1998.

[37] H. W. Haussecker and D. J. Fleet, "Computing Optical Flow with Physical Models of Brightness Variation," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 23, pp. 661-673, 2001.

[38] L. Alvarez, "Reliable Estimation of Dense Optical Flow Fields with Large Displacements," *International Journal of Computer Vision,* vol. 39, pp. 41-56, 2000.

[39] Y.-H. Kim, A. M. Martínez, and A. C. Kak, "Robust Motion Estimation under Varying Illumination," *Image and Vision Computing,* vol. 23, pp. 365-375, 2005.

[40] Y.-H. Kim, A. M. Martinez, and A. C. Kak, "A Local Approach for Robust Optical Flow Estimation under Varying Illumination," in *Brithish Vision Conference*, 2004.

[41] H. Spies and H. Scharr, "Accurate optical flow in noisy image sequences," in *Proceedings on Eighth IEEE International Conference on Computer Vision*, 2001, pp. 587-592 vol.1.

[42]   A. Beghdadi, M.-f. Auclair-fortier, and J. Monteil, "Tracking of image intensities based on optical flow: An evaluation of nonlinear diffusion process," in *In Second IEEE International Symposium on Signal Processing and Information Technology*, 2002, pp. 691-696.

[43]   T. Brox and J. Weickert, *Nonlinear Matrix Diffusion for Optic Flow Estimation* vol. 2449/2002: Springer Berlin, 2002.

[44]   J. Xiao, H. Cheng, H. Sawhney, C. Rao, and M. Isnardi, "Bilateral Filtering-Based Optical Flow Estimation with Occlusion Detection " *Lecture notes in computer science,* vol. 3951/2006, pp. 211-224, 2006.

[45]   H.-Y. Wang and K.-K. Ma, "Motion field discontinuity classification for tensor-based optical flow estimation," in *IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings. (ICASSP '03)* 2003, pp. III-665-8 vol.3.

[46]   T. Pock, M. Urschler, C. Zach, R. Beichel, and H. Bischof, *A Duality Based Algorithm for TV- L1-Optical-Flow Image Registration* vol. 4792/2007: Springer Berlin, 2007.

[47]   C. Zach, T. Pock, and H. Bischof, *A Duality Based Approach for Realtime TV- L1 Optical Flow* vol. 4713/2007: Springer Berlin, 2007.

[48]   A. N. Tikhonov, *Solutions of ill-posed problems*: V.H. Winstons & Sons, 1997.

[49]   Y. Sawaragi, H. Nakayama, and T. Tanino, *Theory of Multiobjective Optimization*: Academic Press, 1985.

[50]   K. Deb, *Multi-Objective Optimization Using Evolutionary Algorithms*: John Wiley and Sons, 2001.

[51]   S. Boyd, *Convex Optimization*: Cambridge University Press, 2004.

[52]   R. Horst, P. M. Pardalos, and N. V. Thoai, *Introduction to Global Optimization*: Kluwer Academic Publishers, 2000.

[53]   D. Gibson and M. Spann, "Robust optical flow estimation based on a sparse motion trajectory set," *IEEE Transactions on Image Processing,* vol. 12, pp. 431-445, 2003.

[54]   A. Bruhn and J. Weickert, "Lucas/Kanade Meets Horn/Schunk: Combining Local and Global Optic Flow Methods," *International Journal of Computer Vision,* vol. 61, pp. 211-231, 2005.

[55]   J. M. Bland and D. G. Altman, "Measuring agreement in method comparison studies," *Statistical Methods in Medical Research,* pp. 135-160, 1999.

[56]   D. G. Altman and J. M. Bland, "Measurement in medicine: the analysis of method comparison studies," *The Statistician,* vol. 32, pp. 307-317, 1983.

[57]   T. Cover, *Elements of Information Theory*: Wiley Interscience, 1991.

[58]   C. K. Zarins, C. Xu, and S. Glagov, "Atherosclerotic enlargement of the human abdominal aorta," *Atherosclerosis,* vol. 155, pp. 157-164, 2001.

[59]   A. H. Association, "Heart Disease and Stroke Statistics-2008," Dallas.

[60]   S. Meairs and M. Hennerici, "Four-Dimensional Ultrasonographic Characterization of Plaque Surface Motion in Patients with Symptomatic and Asymptomatic Carotid Artery Stenosis," *Stroke,* pp. 1807-1813, 1999.

[61]   S. Golemati, A. Sassano, J. M. Lever, and A. A. Bharath, "Carotid artery wall motion estimated from B-mode ultrasound using region tracking and block matching.," *Ultrasound Med. Biol.,* vol. 29, pp. 387-399, 2003.

[62]   J. Stoitsis, S. Golemati, A. K. Dimopoulos, and K. S. Nikita, "Analysis and quantification of arterial wall motion from B-mode ultrasound images - comparison of block-matching and optical flow," in *Proc. 27th Annual International Conference of the Engineering in Medicine and Biology Society IEEE-EMBS 2005*, 2005, pp. 4469-4472.

[63]   J. Stoitsis, S. Golemati, D. Perakis, A. Alexandridi, C. Davos, and K. S. Nikita, "Carotid Artery Motion Estimation from Sequences of B-mode Ultrasound Images: Effect of Dynamic Range and Persistence," *Proc. IEEE International*

Workshop on [Imagining read Imaging] Imagining Systems and Techniques IST 2006, pp. 131-136, 2006.

[64] C. Loizou, C. Pattichis, M. Pantziaris, and A. Nicolaides, "An Integrated System for the Segmentation of Atherosclerotic Carotid Plaque," *IEEE Transactions on Information Technology in Biomedicine,* vol. 11, pp. 661-667, 2007.

[65] C. P. Loizou, C. S. Pattichis, C. I. Christodoulou, R. S. H. Istepanian, M. Pantziaris, and A. Nicolaides, "Comparative Evaluation of Despeckle Filtering In Ultrasound Imaging of the Carotid Artery," *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control,* vol. 52, pp. 1653-1668, October 2005.

[66] S. Haykin, *Adaptive Filter Theory*: Pearson Education, 2002.

[67] S. E. Murillo, M. S. Pattichis, P. Soliz, S. Barriga, C. P. Loizou, and C. Pattichis, "Global optimization for motion estimation with applications to ultrasound videos of carotid artery plaques," in *SPIE Medical Imaging 2010: Ultrasonic Imaging, Tomography, and Therapy* San Diego, California, 2010.

[68] J. M. Lever, M. Dimaki, A. A. Bharath, M. Otawara, and H. Fujiwara, "Investigation of human carotid artery wall and atherosclerotic plaque mechanics using b-mode ultrasound," in *Summer Bioengineering*, 2003.

[69] E. Kreyszig, *Advance Engineering Mathematics*. New York: Wiley, 1988.

[70] R. F. Wagner, S. W. Smith, J. M. Sandrik, and H. Lopez, "Statistics of Speckle in Ultrasound B-Scans," *Sonics and Ultrasonics, IEEE Transactions on,* vol. 30, pp. 156-163, 1983.

[71] K. Z. Abd-Elmoniem, A. B. M. Youssef, and Y. M. Kadah, "Real-time speckle reduction and coherence enhancement in ultrasound imaging via nonlinear anisotropic diffusion," *Biomedical Engineering, IEEE Transactions on,* vol. 49, pp. 997-1014, 2002.

[72] H. Xiaohui, G. Shangkai, and G. Xiaorong, "A novel multiscale nonlinear thresholding method for ultrasonic speckle suppressing," *Medical Imaging, IEEE Transactions on,* vol. 18, pp. 787-794, 1999.

185

[73]   R. A. Johnson and D. W. Wichern, *Applied multivariative statistical analysis*, Third ed.: Pretince Hall, 1992.

[74]   H. Farid and E. P. Simoncelli, "Differentiation of discrete multidimensional signals," *Image Processing, IEEE Transactions on,* vol. 13, pp. 496-508, 2004.

[75]   R. L. Maurice and M. Bertrand, "Lagrangian speckle model and tissue-motion estimation-theory [ultrasonography]," vol. 18, pp. 593-603, 1999.

[76]   R. L. Maurice, J. Ohayon, Y. Fretigny, M. Bertrand, G. Soulez, and G. Cloutier, "Noninvasive vascular elastography: theoretical framework," vol. 23, pp. 164-180, 2004.