A Data-driven Learning Approach to Image Registration

Mohammad Abdur R. Mustafa

Thesis submitted to the University of Nottingham for the degree of Doctor of Philosophy

February 2016

Abstract

Handling large displacement optical flow is a remarkably arduous task. For instance, standard coarse-to-fine techniques often struggle to adequately deal with moving objects whose motion exceeds their size. Here we propose a learning approach to the estimation of large displacement between two non-consecutive images in a sequence on the basis of a learning set of optical flows estimated *a priori* between different consecutive images in the same sequence. Our method refines an initial estimate of the flow field by replacing each displacement vector by a linear combination of displacement vectors at the center of similar patches taken from a codebook built from the learning set. The key idea is to use the accurate flows estimated *a priori* between consecutive images to help improve the potentially less accurate flows estimated *online* between images further apart. Experimental results suggest the ability of a purely data-driven learning approach to handle fine scale structures with large displacements.

Acknowledgements

This thesis would have never seen the light of day if it had relied on my efforts alone. Instead, it was the result of the collective efforts of a number of extraordinary people who supported and assisted me in many ways during this long and challenging journey.

First and foremost, I wish to express my sincere appreciation and gratitude to my supervisor Dr. Alain Pitiot, a man of wisdom and warmth to whom I owe a great intellectual debt. I appreciate all his contributions of time, ideas, and funding to make my PhD experience productive and stimulating.

I will forever be thankful to my other supervisor Dr. Timor Kadir who provided helpful advice on many occasions. He was and remains my best role model for a scientist and mentor. I still think fondly of my time as a research student in Mirada Medical, Oxford. This thesis has also benefited from discussions with Dr. Jonathan Peirce and Dr. Denis Schluppeck. Their useful suggestions helped to guide the project along a smooth path.

My time at Nottingham was made enjoyable in large part due to the many friends and groups that became a part of my life. I am grateful for time spent with friends in my research lab, especially Matthew Buckley, Stephanie McGarity and Natalia Kominiarczuk.

I would like to thank my family for all their love and encouragement, for my parents who raised me with love and supported me in all my pursuits. For my loving, supportive, encouraging, and patient wife Ifat whose support during all stages of this PhD is so appreciated. Most of all, I am ever so grateful to my 6 years old son, Muhammad who used to inspire me every morning by telling stories of how goodies are defeating baddies and making the world a place without worry.

Finally, I greatly thank the Creator, Allah for giving me the physical and intellectual ability to finish this work. With His will and mercy, I have been guided to all those great people who helped me to finish this work.

Contents

List of Figures

Nomenclature

AAE Average Angular Error

AAE Average Angular Error

AEPE Average Endpoint Error

AEPE Average Endpoint Error

ANOVA Analysis of variance

CT Computed Tomography

FLIRT FMRIB's Linear Image Registration Tool

FMRIB Oxford centre for Functional MRI of the Brain

FNIRT FMRIB's Linear Image Registration Tool

FoE Field of Experts

HOG Histogram of Oriented Gradients

HS The Horn–Schunck method of estimating optical flow

LDOF Large Displacement Optical Flow

LK The Lucas-Kanade method for optical flow estimation

LLE Locally Linear Embedding

LTSA Local Tangent Space Alignment

MRF Markov Random Fields

MRI Magnetic Resonance Image

MSE Mean Squared Error

NLM Non Local Means

PCA Principal Component Analysis

RP Representative Patches

SIFT Scale Invariant Feature Transform

SMD Statistical Model of Deformation

SMF Sun's Median Filtering Method

SMF Sun's Median Filtering Method

SPSA Simultaneous Perturbation Stochastic Approximation

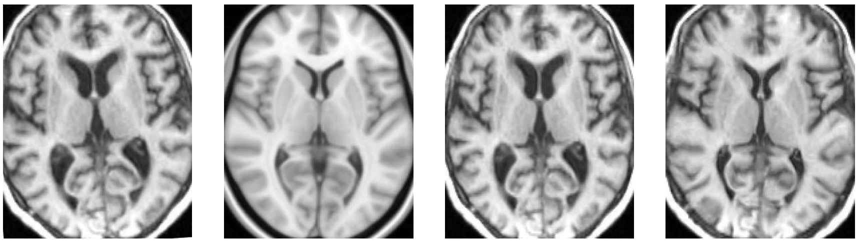
SRF Steerable Random Field

Chapter 1

Introduction

1.1 Motivation

Image registration is the process of establishing correspondences between two or more images (Crum et al., 2014). It is motivated by the hope that better or more information can be extracted from an adequate merging of the images than from analyzing them independently. It consists in transforming the input images until the relevant image structures or features are correctly aligned. Based on the geometric flexibility of the transformation, the image registration process can be divided into two categories, linear and non-linear. An image registration is linear when only rotation, scaling, translation and shearing are allowed. Linear image registration can be applied, for instance, to register images of the same patient taken at different points in time for tumor monitoring or functional imaging. This kind of registration is global in nature and cannot model local geometric differences between images. On the other hand, nonlinear image registration can locally transform the source image to align it more accurately with the target image. For an example, it can be used to register the Magnetic Resonance Image (MRI) of a patient's scan to an anatomical atlas. In Figure 1.1, both linear and non-linear image registrations are performed using FMRIB's FLIRT and FNIRT respectively to transform a participant's brain (the source image) to a T1-weighted average structural template image (the target image). The linear transformation manages to correctly orient the source image but is not very accurate overall and especially around the ventricles and at the sulcus level. The non-linear model does a much better job at aligning the images.



(a) (b) (c) (d)

Figure 1.1: Image transformations; (a) a source image; (b) the template image; (c) linearly transformed source image; (d) non-linearly transformed source image.

Multiple images of subjects can be obtained at multiple times from multiple imaging devices. These imaging modalities can be divided into two global categories: anatomical and functional (Maintz and Viergever, 1998). Anatomical modalities mainly show morphological information. Examples of this category include X-ray, CT (Computed Tomography), MRI (Magnetic Resonance Imaging) , Ultrasound, CTA (Computed Tomography Angiography) etc. Functional modalities show principally the information of metabolism. SPECT (Single Photon Emission Computed tomography), PET (Positron Emission Tomography), fMRI (functional MRI) are some of the examples of this modality. It may be beneficial to accurately combine images of these two categories to get more potential information. For instance, the registration of a pre-operative CT image to an intra-operative X-ray image can be very helpful to guide treatment.

Medical image registration technique is used to accurately align, and thus to combine, multiple images. One of the earliest attempts to deal with non-rigid image registration was made by Horn and Schunck (1980). Their method uses optical flow to estimate the motion of intensity values across two images acquired with the same modality. Optical flow is the spatial distribution of apparent velocities of movement of brightness patterns in an image (Beauchemin and Barron, 1995). It assumes that when a pixel flows from one image to another, its intensity (or color) does not change. The optical flow method calculates the motion between two image frames which are taken at times t and $t + \delta t$ at every voxel position. If a voxel at location (x, y, t) with intensity $I\,(x, y, t)$ moves after time δt from one image to other image by δx and δy then the image constraint equation can be written as:

$$I\,(x, y, t) = I\,(x + \delta x, y + \delta y, t + \delta t) \tag{1.1}$$

If the displacement is very small, we can apply Taylor series to expand this equation and get:

$$I(x+\delta x, y+\delta y, t+\delta t) = I(x,y,t) + \frac{\partial I}{\partial x}\delta x + \frac{\partial I}{\partial y}\delta y + \frac{\partial I}{\partial t}\delta t + O^2 \tag{1.2}$$

where O^2 is the second or higher order terms. From these equations, it follows that:

$$\frac{\partial I}{\partial x}\delta x + \frac{\partial I}{\partial y}\delta y + \frac{\partial I}{\partial t}\delta t = 0 \tag{1.3}$$

or

$$\frac{\partial I}{\partial x}\frac{\delta x}{\delta t} + \frac{\partial I}{\partial y}\frac{\delta y}{\delta t} + \frac{\partial I}{\partial t}\frac{\delta t}{\delta t} = 0 \tag{1.4}$$

i.e.,

$$\frac{\partial I}{\partial x}V_x + \frac{\partial I}{\partial y}V_y + \frac{\partial I}{\partial t} = 0 \tag{1.5}$$

where V_x, V_y are the x and y components of the optical flow of $I(x,y,t)$ and $\frac{\partial I}{\partial x}$, $\frac{\partial I}{\partial y}$ and $\frac{\partial I}{\partial t}$ are the derivatives of images (or image gradients) at (x,y,t) in the corresponding directions. if I_x, I_y and I_t are the first order partial derivatives of $I\,(x,y,t)$, the optical flow equation becomes:

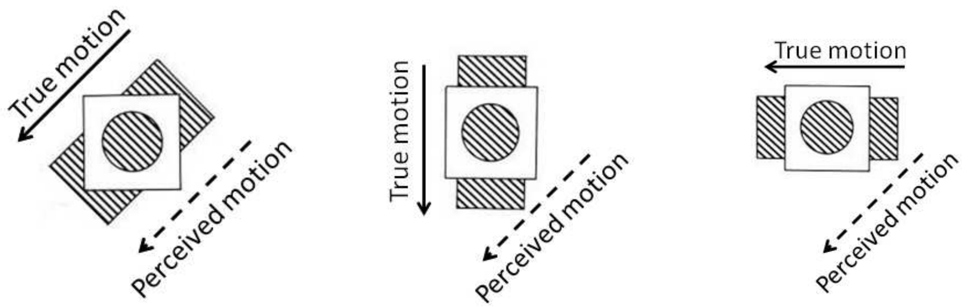
$$I_xV_x + I_yV_y + I_t = 0 \tag{1.6}$$

or ,

$$I_xV_x + I_yV_y = -I_t \tag{1.7}$$

where I_x, I_y and I_t are the image gradient with respect to positions x, y and time t (for detail see Fleet and Weiss (2006)). This is one equation with two unknowns, which cannot be solved as such.

Equation 1.7 has two unknowns which cannot be solved without another equation or constraint. This is known as the aperture problem of the optical flow algorithms (Movshon et al., 1985).



(a) (b) (c)

Figure 1.2: Aperture problem; three gratings (a), (b) and (c) are moving in three directions (i.e. true motion). When viewed through a small circular aperture, all three gratings appear to move in the same direction (i.e. perceived motion) which is perpendicular to the orientation of the lines in the gratings.

Figure 1.2 illustrates the aperture problem. Three gratings (Figure 1.2a, b and c) have same pattern of parallel lines. These gratings are moving in three different directions. If we see through a small aperture, all three gratings appear to have a motion in the same direction, which is perpendicular to the orientation of the lines in the gratings (Barron et al., 1994). Given the constrain in equation 1.7, where we have two unknowns, we can only determine the flow perpendicular to the orientation of the lines in the gratings, but we cannot determine the motion on other directions. This failure to accurately detect the true direction of motion is called the aperture problem.

To find the optical flow from Equation 1.7 another set of equations is needed, given by some additional constraint. Most optical flow methods introduce additional conditions for estimating the actual flow. For instance, Horn and Schunck introduced a global constraint of smoothness to solve this equation (for detail see 2.1.2). It assumes smoothness in the optical flow over the whole image. It tries to minimize distortions in flow and prefers solutions which show more smoothness.

Most of the attention devoted to optical flow has been dedicated to addressing the shortcomings of the initial HS formulation such as outliers (Black and Anandan, 1996; Brox et al., 2004; Lempitsky et al., 2008; Wedel et al., 2009a); lighting changes (Brox et al., 2004; Lempitsky et al., 2008; Zimmer et al., 2009; Wedel et al., 2008); over-smoothing (Xiao et al., 2006; Black and Jepson, 1996; Zitnick et al., 2005; Xu et al., 2008) and non-convex energy minimization (Black and Anandan, 1996; Boykov et al., 2001; Trobin et al., 2008)). Despite the variety of methods, there are many occasions where most of these techniques may not estimate an accurate dense correspondence field between images. Examples of such cases are motions of untextured areas, aliasing, occlusions etc. (see Butler et al. (2012)) or, crucially, flows with large displacements, which is the focus of this thesis.

1.1.1 Handling large displacements

The original Horn and Schunck (HS) framework can only handle small motions as the linearization of the data term only holds for small magnitude velocities. In the presence of large displacements, this method may not estimate motion correctly as image gradient is not smooth enough. For large displacements, matching pixels further apart rapidly becomes a computationally intensive problem as the number of potential matches grows quadratically with the magnitude of the largest recoverable displacement. Figure 1.3 illustrates how HS method struggles with large displacements. In this example we use a pair of images from the Temple3 sequence in the MPI-Sintel dataset (rendered scenes from the Durian Open Source Movie Project; for detail see section 2.3.3) where the wings of a dragon undergo large displacements (see Figure 1.3a and b). The basic HS method cannot estimate a correct flow as matching pixels are too far apart (see Figure 1.3d). Figure 1.3g shows the color code map where the color represents the orientation of the vector and brightness stands for its magnitude.

A popular way to address this intrinsic limitation is to adopt a coarse-to-fine (or pyramidal) strategy (Anandan, 1989; Enkelmann, 1988; Fleet and Weiss, 2006). Under such scheme, the flow can be estimated at the coarsest scale of a Gaussian pyramid first. As the images are downsampled, the velocity decreases. Thus, derivatives can be used to estimate the residual velocity progressively at the finer scales. The flow is iteratively refined on the downsampled images with the underlying assumption that the residual motion field at each scale satisfies the linearization constraint since the motion field between the images is scaled together with the images.

A synthetic example is given in Figure 1.4, which illustrates the coarse-to-fine approach for large displacement. The images consist of a gray moving disc. As the images are downsampled (each time image size is reduced by half), the distance between the discs decreases, which eventually makes it possible to estimate a flow (at the coarsest scale). Figure 1.3e shows that compared to the original HS method, HS with coarse-to-fine approach estimates a more accurate flow.

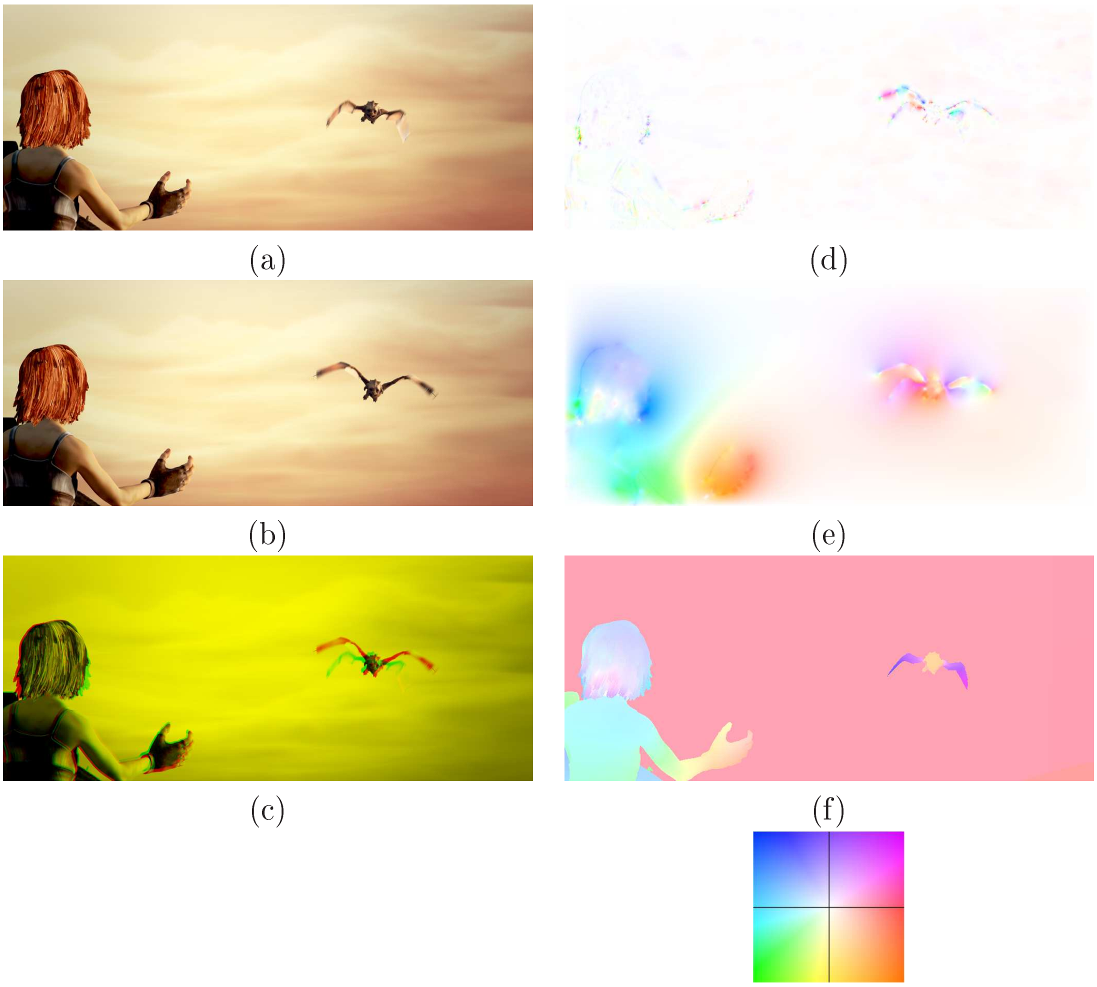

(a) (d) (b) (e) (c) (f) (g)

Figure 1.3: An example from the Temple3 sequence in the MPI-Sintel dataset where standard optical flow methods struggle to accurately estimate large displacements of small structures: (a) source frame; (b) target frame; (c) source (in green) superimposed on target image (in red); (d) flow estimated with basic Horn-Schunck method without coarse-to-fine approach; (e) Horn-Schunck method with coarse-to-fine approach; (f) ground truth flow; (g) flow field color coding: the color represents the orientation of the vector and brightness stands for its magnitude.

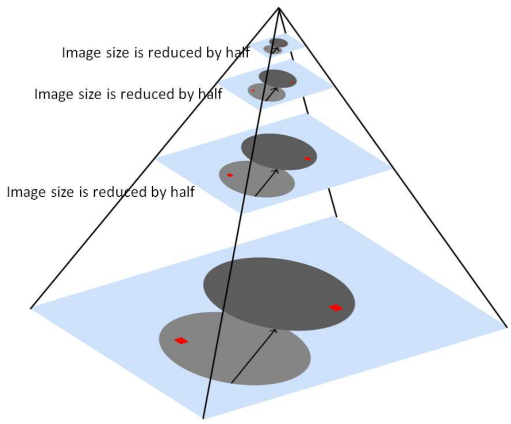
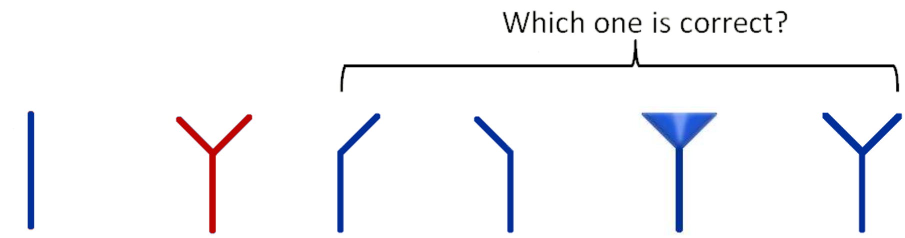


Figure 1.4: A synthetic example demonstrates the coarse-to-fine (or pyramidal) approach to large displacement. Two images of a gray moving disc are super-imposed. Images are progressively downsampled. Each time image size is reduced by half. Coarser images are shown in the higher level of the pyramid. Displacement between source and target (indicated by the arrow) becomes smaller in the downsampled images.

However, since fine scale image structures may become invisible at coarser scales, coarse-to-fine approaches cannot reliably estimate the flow of structures whose motion is larger than their size (Wulff et al., 2012). That is the case with the red rectangle in Figure 1.4, which rapidly disappear from the progressively downsampled images.

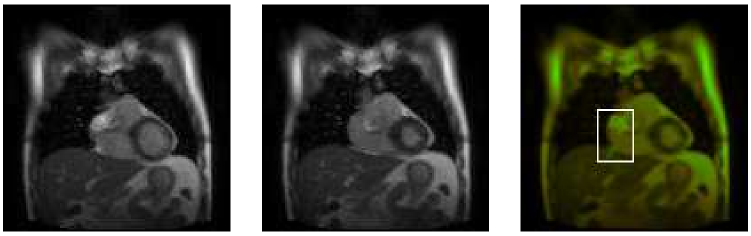
This realisation led to the development of new methods capable of handling large displacements of fine structures. For instance, Alvarez et al. (2000) used a linear scale-space focusing strategy to increase the robustness to local minima. Steinbrucker et al. (2009) avoided coarse-to-fine warping and linearization altogether by decoupling the data and regularisation terms and alternatively optimising them. Brox and Malik (2011) complemented the standard continuation method with rich local descriptors (such as SIFT or HOG). Along similar lines, Xu et al. (2012) also used SIFT to generate candidate flows which were then fused together and integrated to the ordinary optical flow.

But descriptor based approaches have limitations too. Regions with weak texture often do not produce reliable keypoints. Also highly repetitive structures can create ambiguity for the descriptor matching. The synthetic example in Figure 1.5 illustrates a situation where a descriptor matching based method cannot decide by



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

Figure 1.5: An ambiguous registration situation; (a) source image; (b) target image; (c), (d), (e) and (f) four possible registered source images. Which one is correct?



(a) (b) (c)

Figure 1.6: Dynamic MRI scans: (a) source frame; (b) target frame; (c) target frame (in red) on top of source frame transformed using LDOF flow (in green).

itself what the correct transformation is. Figure 1.6 displays an example with a 2-D dynamic Magnetic Resonance Imaging (dMRI) scan of a healthy volunteer breathing normally in a Siemens 1.5T scanner. To estimate the transformation, we used the Large Displacement Optical Flow (LDOF) method (Brox and Malik, 2011) which matches local descriptors. Figure 1.6c displays the target frame (in red) on top of source frame transformed using LDOF flow (in green). LDOF managed to recover most of the motion between frames (both vertical breathing motion and the heart contraction), but it produced poorly regularized flow due to ambiguity of local descriptor (see the region highlighted with a white rectangle in Figure 1.6c).

1.1.2 Learning Optical Flow

Whilst considerable efforts have been expanded to improve flow estimation using techniques like coarse-to-fine approaches and feature descriptors, learning approaches have only attracted limited attention to date. Simoncelli et al. (1991) first introduced a probabilistic framework to estimate the deviation of the estimated flow from the true flow. Black et al. (1997) extracted orthogonal basis flow fields using principal component analysis to learn parameterized models of image motion. Xue et al. (2006) coupled PCA with wavelet-based decomposition to learn dense deformation fields when a limited number of training images are available. Freeman et al. (2000) learned the parameters of a Markov network from a training set and used a Bayesian belief propagation approach to estimate the flow of amorphous blobs. Roth and Black's (Roth and Black, 2007a) Field of Experts also relied on Markov random fields to model the spatial statistics of optical flow fields. Sun et al. (2008) modelled both spatial and brightness statistics using a steerable random field. Li and Huttenlocher (2008a,b) developed a proposed Markov Random Field model of optical flow and proposed a technique that learns parameters of the model. Their method minimizes training loss that occurs due to unseen or unmatched data, for instance data that appears due to occlusion. To overcome similar problem with occlusion, Mac Aodha et al. (2013) introduced a supervised learning approach that learns confidence measure for optical flow.

To the best of our knowledge, none of these methods were designed to handle large displacements. Here we propose a data-driven, learning approach to motion estimation capable of dealing with those. We focus on the computation of the optical flow between two non-consecutive images in a sequence on the basis of a learning set of optical flows carefully estimated *a priori* between different consecutive images in the same sequence. Rather than learning a statistical model of the flow, we propose to refine an initial estimate of the flow field by replacing each displacement vector by a linear combination of displacement vectors at the centre of similar patches taken from an *a priori* code-book. The key idea is to use the accurate flows estimated *a priori* between consecutive images to help improve the potentially less accurate flows estimated *online* between images further apart. In common with recent developments, our approach does not require a coarse-to-fine or warping strategy, which makes it possible to handle fine scale structures with large displacements.

The contributions of this thesis are (1) to analyze current flow models and methods to understand how they deal with large displacements; (2) to formulate an algorithm that estimates transformations using a learning set of optical flows taken from carefully estimated *a priori* between different consecutive images; (3) to compare the performances of the proposed algorithm against those of state-of-the-art methods.

There were not online codes available for LK 3-D and HS 3-D methods. The codes developed for the experiments in thesis are available at Matlabcentral (mathworks.co.uk/ matlabcentral/ fileexchange/ authors/ 257136). There are four set of codes; they are:

1. Horn-Schunck optical flow method for 3-D images
2. Lucas-Kanade optical flow method for 3-D images
3. Lucas-Kanade optical flow method with pyramidal approach for 3-D images
4. Lucas-Kanade optical flow method with weighted window approach for 3-D images

1.2 Thesis Overview

In chapter 2, we present a general overview of image registration techniques. We implemented Horn-Schunck and Lucas-Kanade and experimented with several established registration techniques (Median Filtering, LDOF, DeepFlow) as a means to familiarize ourselves with their advantages and drawbacks. In particular, we explore the practical issues related to optical flows for large displacement. Then we introduce the datasets that we used in our experiments. At the end of this chapter, we discuss the experimental setup and how to estimate registration accuracy.

In chapter 3, we present our learning approach to optical flow estimation. We detail its various components and explore their contributions and drawbacks. In particular, we introduce and compare different patch similarity measures, explore the issue of oversmoothing and propose vector composition of pairwise deformation field as a means to overcome it.

In chapter 4, we assess the performance of the codebook of patches. We also discuss various features related to codebook of patches, and in particular the performances of several manifold learning techniques and different clustering algorithms.

In chapter 5, we discuss about patch similarity measures of spatial distance and intensity and assess their performance.

Finally, in chapter 6, we discuss the overall strengths and limitations of our approach. We conclude by scrutinizing what questions have been answered and what questions are still open. In doing so, we point to a number of future directions for our work.

Chapter 2

Background

In this chapter, we present a general overview of optical flow methods and explore the practical issues related to large displacement. We review previous learning approaches to optical flow estimations. We also describe the datasets that we are going to use in our experiments. Finally, we discuss how to compare the performances of flow estimation methods.

2.1 Estimating Optical Flow

We first discuss two fundamental optical flow techniques, Horn-Schunck and Lucas-Kanade, and analyze the way they deal with large displacements. We then review more modern approaches such as Median Filtering method, LDOF and DeepFlow, which form the current state-of-the-art.

2.1.1 Lucas-Kanade (LK) Optical Flow

The Lucas-Kanade method (Lucas and Kanade, 1981) assumes that the flow is essentially constant in a local neighbourhood of the pixel under consideration. It uses a least squares criterion to solve the basic optical flow equations for all the pixels in that neighborhood. Therefore, the local optical flow vector (V_x, V_y) in that neighborhood must satisfy:

$$
\begin{gathered}
I_x(p_1)\,V_x + I_y(p_1)\,V_y = -I_t(p_1) \\
I_x(p_2)\,V_x + I_y(p_2)\,V_y = -I_t(p_2) \\
\vdots \\
I_x(p_n)\,V_x + I_y(p_n)\,V_y = -I_t(p_n)
\end{gathered}
\tag{2.1}
$$

where $p_1, p_2, ..., p_n$ are the pixels inside the block, and $I_x(p_i), I_y(p_i), I_t(p_i)$ are the image gradient of the image I with respect to position x, y and time t evaluated at the point p_i and at the current time. Here, we have more equations than unknowns and thus the system is over-determined, which typically creates an aperture problem (Beauchemin and Barron, 1995) (for detail see appendix A). To overcome this problem, Lucas and Kanade estimate an approximate solution using a least squares approach:

$$\begin{bmatrix} V_x \\ V_y \end{bmatrix} = \begin{bmatrix} \sum_i I_x(p_i)^2 & \sum_i I_x(p_i) I_y(p_i) \\ \sum_i I_x(p_i) I_y(p_i) & \sum_i I_y(p_i)^2 \end{bmatrix}^{-1} \begin{bmatrix} -\sum_i I_x(p_i) I_t(p_i) \\ -\sum_i I_y(p_i) I_t(p_i) \end{bmatrix} \tag{2.2}$$

where $i = 1$ to n.

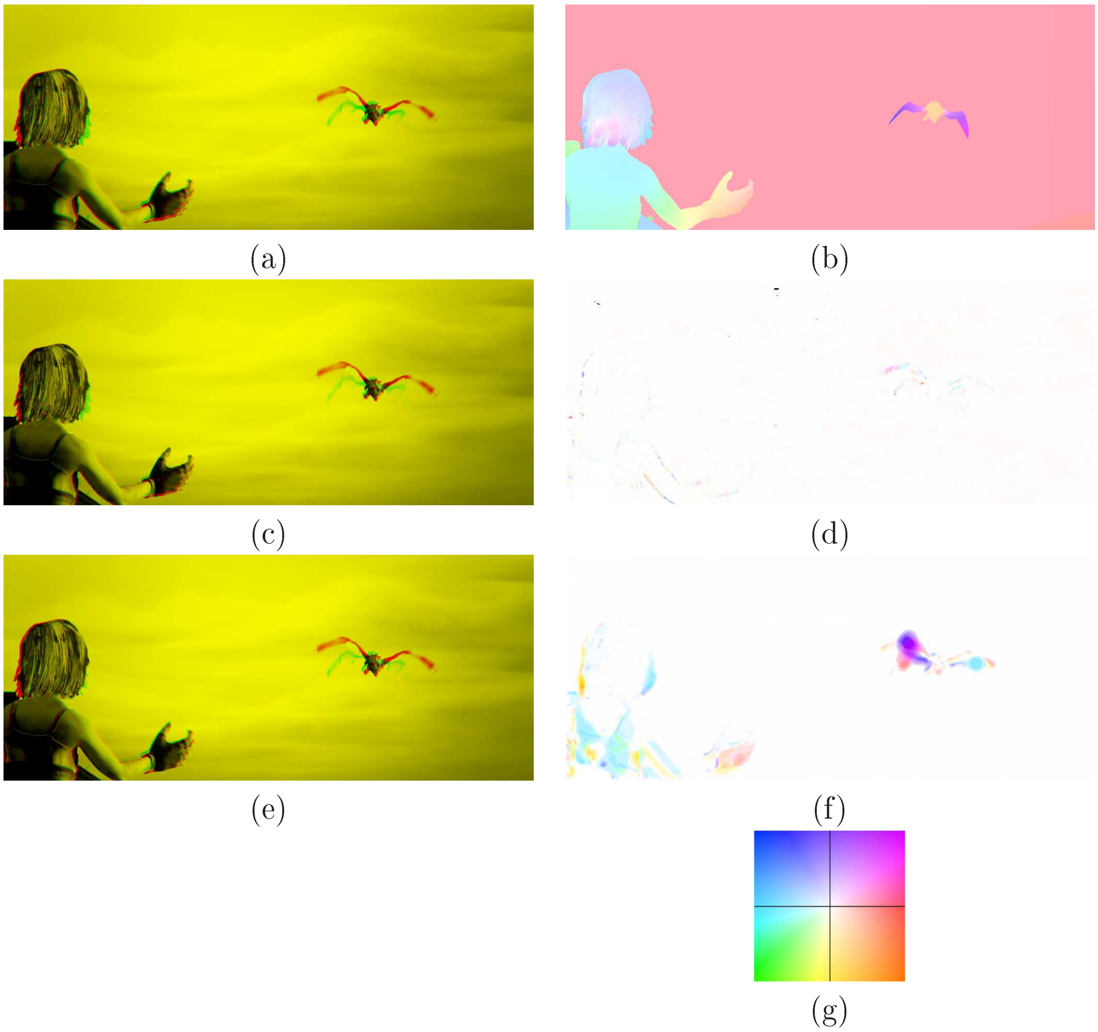
Note that LK cannot provide the information of optical flow inside blocks.

Figure 2.1 shows an example where LK is used to estimate motion between two frames from the Temple3 sequence in the MPI-Sintel dataset. Figure 2.1a shows the source image (in green) super-imposed on target image (in red). The ground truth motion is given in Figure 2.1b. Figure 2.1c shows the registered source image (in green) super-imposed on the target image (in red) and Figure 2.1d shows the color coded LK flow field. That field is not very accurate as image flow vectors (V_x, V_y) of motions of the wings of the dragon between the two frames are larger than the local neighborhood window and thus violate the assumption that the flow is constant in a local neighborhood (see section 2.1.3 for detail).

2.1.2 Horn-Schunck (HS) Optical Flow Method

The Horn-Schunck framework solves the aperture problem by introducing a global constraint of smoothness (Horn and Schunck, 1980). This approach tries to minimize the irregularities in the optical flow by favouring solutions that maximize smoothness. The optical flow is formulated as a global energy functional, which is minimized. For a pair of two-dimensional images, this function is given as:

$$E(V_x, V_y) = \int\int \left[(I_x V_x + I_y V_y + I_t)^2 + \alpha^2 \left(\|\nabla V_x\|^2 + \|\nabla V_y\|^2 \right) \right] dxdy \tag{2.3}$$



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

Figure 2.1: MPI-Sintel Temple3: (a) source image (in green) super-imposed on target image (in red); (b) ground truth motion; (c) source image registered using basic LK (in green) super-imposed on target image (in red); (d) flow estimated by basic LK method without coarse-to-fine approach; in each level of pyramid image sizes are reduced by half; (e) source image registered using LK with coarse-to-fine approach (in green) super-imposed target image (in red); in each level of pyramid image sizes are reduced by half; (f) flow estimated by the LK method with coarse-to-fine approach; (g) flow field color coding: the color represents the orientation of the vector and brightness stands for its magnitude.

where I_x and I_y are the derivatives of the image intensity values along the x and y coordinates respectively. I_t is the derivative along time dimensions, $\overrightarrow{V} = [V_x, V_y]^T$ is the optical flow vector, and α is the regularization constant. Larger values of α yield smoother flows.

If the flow information is missing in inner parts of homogeneous objects, HS can fill in that part from the flow at the boundaries as it is a global optical flow method. However, it is more susceptible to sudden change of motion direction than local optical flow methods because the smoothness term does not allow for sharp discontinuities in the motion field (Bruhn et al., 2005). In Figure 2.2, HS is used to estimate the motion between the same two frames of the previous example from the Temple3 sequence. Figure 2.2b shows the color coded ground truth motion. The estimated flow is shown in Figure 2.2d. The registered source image (in green) is super-imposed on the target image (see and Figure 2.2c). Unlike LK (Figure 2.1c), the HS flow field is smoother and it fills in the inner parts of homogeneous objects such as the body of Sintel, the female protagonist in Figure 2.2c. Even then, HS is not very accurate as motions of the wings of the dragon are still very large for the optical flow assumption to hold.

2.1.3 The Large Displacement Challenge

Recall that the flow equation is:

$$I_x V_x + I_y V_y = -I_t \tag{2.4}$$

where V_x, V_y are the x and y components of the optical flow of $I(x, y, t)$, and I_x, I_y and I_t are image gradient with respect to positions x, y and time t, i.e. directional changes in the intensity (or colour). Image gradients can be computed using many different operators such as Sobel, Prewitt, Central Difference gradient, Intermediate Difference gradient or Roberts gradient amongst others. Most of these operators use intensity values in a small neighborhood around each pixel. For instance, the Sobel and Prewitt operators use intensity values in a 3×3 region around each image point to approximate the corresponding image gradient. The flow equation can handle motions as long as the linearization of the data term holds, i.e. for small magnitude velocities. Indeed, when the displacements become larger, the gradients are not smooth any longer and an accurate estimate of the motion may not be found. Figure 2.3b shows

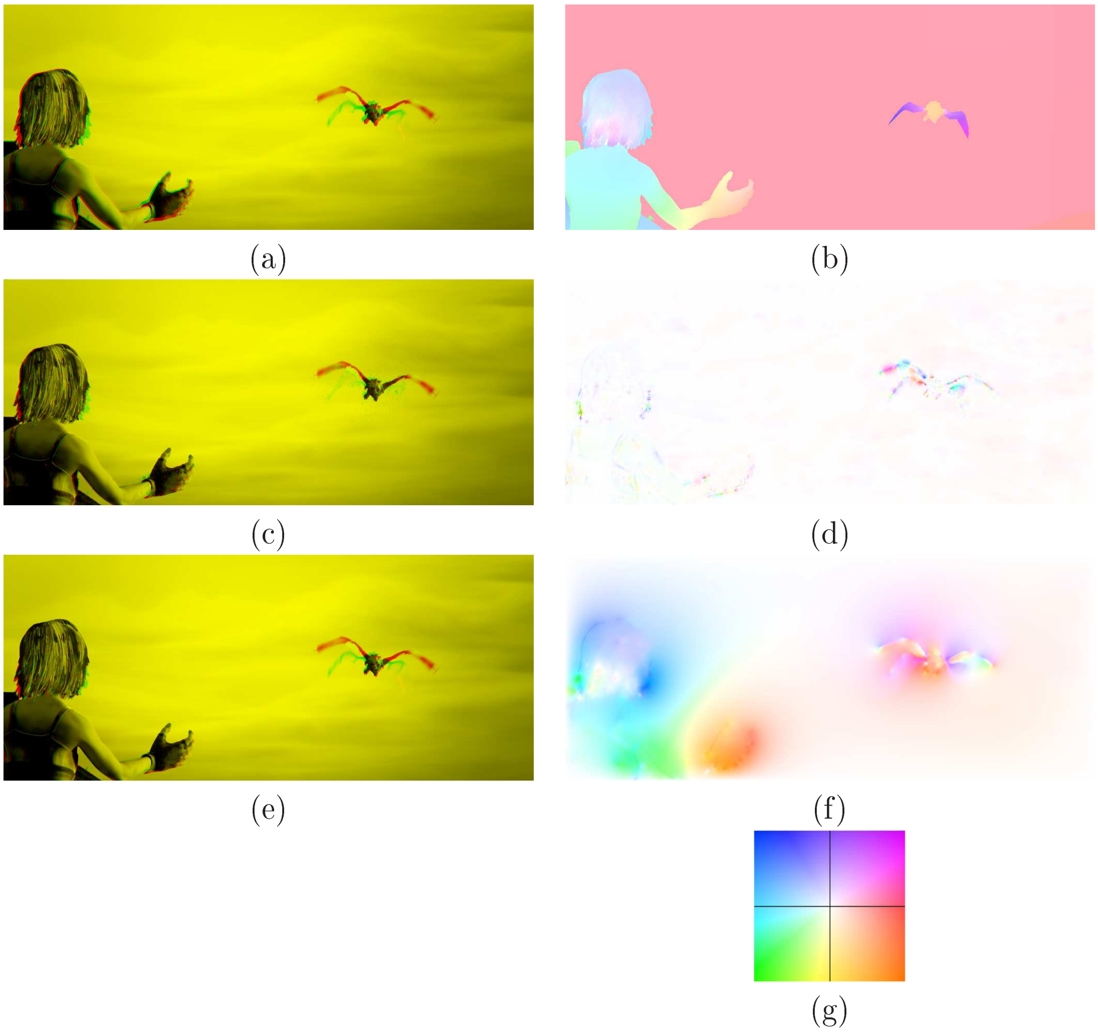


Figure 2.2: The HS approach: (a) source image (in green) super-imposed on target image (in red); (b) ground truth motion; (c) source image registered using HS (in green) super-imposed on target image (in red); (d) flow estimated with basic HS method without coarse-to-fine approach; in each level of pyramid image sizes are reduced by half; (e) source image registered using HS with coarse-to-fine approach (in green) super-imposed target image (in red); in each level of pyramid image sizes are reduced by half; (f) flow estimated the HS method with coarse-to-fine approach; (g) flow field color coding: the color represents the orientation of the vector and brightness stands for its magnitude.

(a) (b)

Figure 2.3: Image gradients: (a) source image (in green) super-imposed on the target frame (in red); (b) magnitudes of the image gradients of source and target frames estimated using the Intermediate Difference method.

the image gradients of frames #7 and #8 from the Temple3 sequence of MPI-Sintel dataset. We compute the image gradients using the Intermediate Difference operator and super-impose them on each other. Clearly, the wings of the flying dragon have relatively large motion (at least 6 pixels between two frames), which is not small enough for the optical flow assumption to hold.

One standard approach to solve this problem is to apply a coarse-to-fine (or pyramidal) strategy where images are downsampled iteratively so that the large displacements become smaller with respect to the displacements in the original image before downsampling, and to apply the flow equation on the downsampled images (for detail see section 1.1.1). Estimated transformations are upscaled and refined accordingly. In Figure 2.4 the pair of frames (#7 and #8) from Temple3 are downsampled twice. Each time image sizes are reduced by half. Consequently, the displacements become smaller with respect to the displacements in the image before downsampling. The magnified images show that after downsampling twice, the distance between the body of the flying dragon in the two consecutive frames has reduced enough for the flow equation to be used to adequately estimate the motion. By further downsampling the frames we can to deal with even larger displacements. Figure 2.1d and 2.2d show the results of using a multilevel coarse-to-fine approach with LK and HS, which produce better estimations compared to the results without coarse-to-fine approach.

2.1.4 Sun's Median Filtering Method (SMF)

Sun's Median Filtering Method (SMF) is a good illustration of a modern optical flow technique. After an extensive review of the recent developments of optical flow meth-

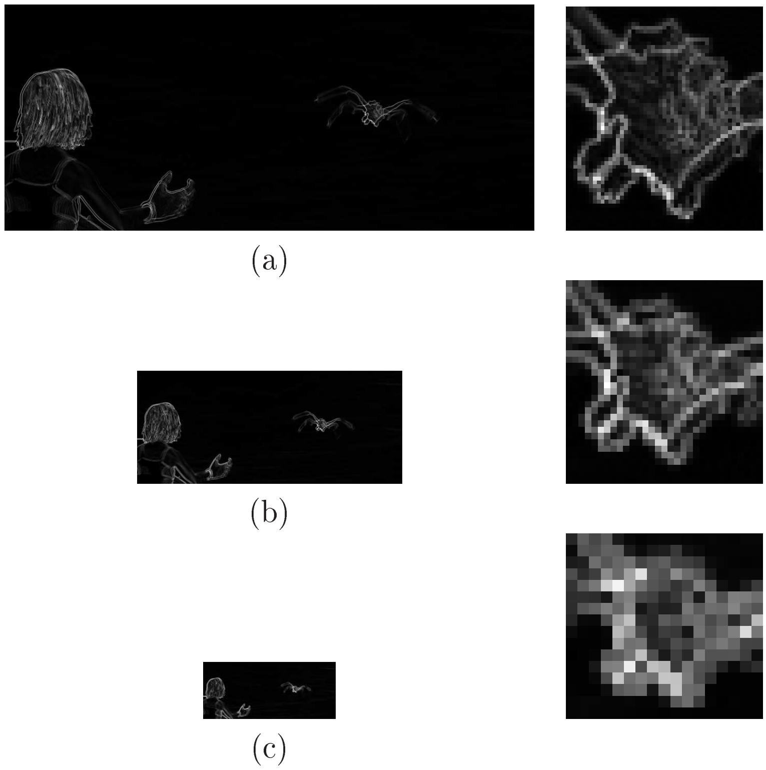


Figure 2.4: Coarse to fine approach: (a) super-imposed image gradients of frame #7 and #8 with magnified view of the dragon on the right; (b) downsampled by half; (c) further downsampled by half.

ods, Sun et al. (2013) proposed to extend the original work of Horn and Schunck (1980) with an asymmetric pyramidal approach where images are downsampled horizontally and vertically in different proportions.

Recall that, with optical flow, we make the assumption that every object has the same brightness or intensity before and after the displacement when computing the apparent velocities of movement. Unfortunately, the brightness of a same object can be different in different images. For instance, Figure 2.5 shows frame #5 and #10 from the Market2 sequence in MPI-Sintel dataset. The red rectangles show patches where the brightness of a walking man changes due to shading. In the shade, the intensity of the walking man became similar like other objects around it. Therefore, in those patches, the image gradient finds no difference of intensity. In other words, the use of image gradients is therefore not helping here (see Figure 2.5d). Consequently, the HS flow field, with a coarse-to-fine approach (in each level of pyramid image sizes are reduced by half), is not very accurate (see Figure 2.5f).

To deal with this issue, the SMF techniques first pre-processes the input images following the method proposed by Wedel et al. (2009b). They use an image decomposition method, which linearly combines the texture (i.e. fine scale-details) and structure components (i.e. the main large objects in the image). This captures the intensity value artifacts generated by shading reflections and shadows. Figure 2.6 illustrates such a structure-texture decomposition. The expectation is that shadows show up only in the structural part, i.e. the main large objects. The hope is then that the computation of optical flow using the textural part of the image is not perturbed by shadow and shading reflection artifacts.

In terms of the regularization parameter, SMF uses a Lorentzian penalty function $\rho(x) = \log\left(1 + \frac{x^2}{2\sigma^2}\right)$, which was originally proposed by Black and Anandan (1996). Consequently the objective function of equation 2.3, in its discrete form, becomes:

$$E(u,v) = \sum_{i,j} \{ \rho_D \left(I_1(i,j) - I_2(i + u_{i,j}, j + v_{i,j}) \right)$$

$$+ \lambda \left[\rho_S (u_{i,j} - u_{i+1,j}) + \rho_S (u_{i,j} - u_{i,j+1}) + \rho_S (v_{i,j} - v_{i+1,j}) + \rho_S (v_{i,j} - v_{i,j+1}) \right]$$

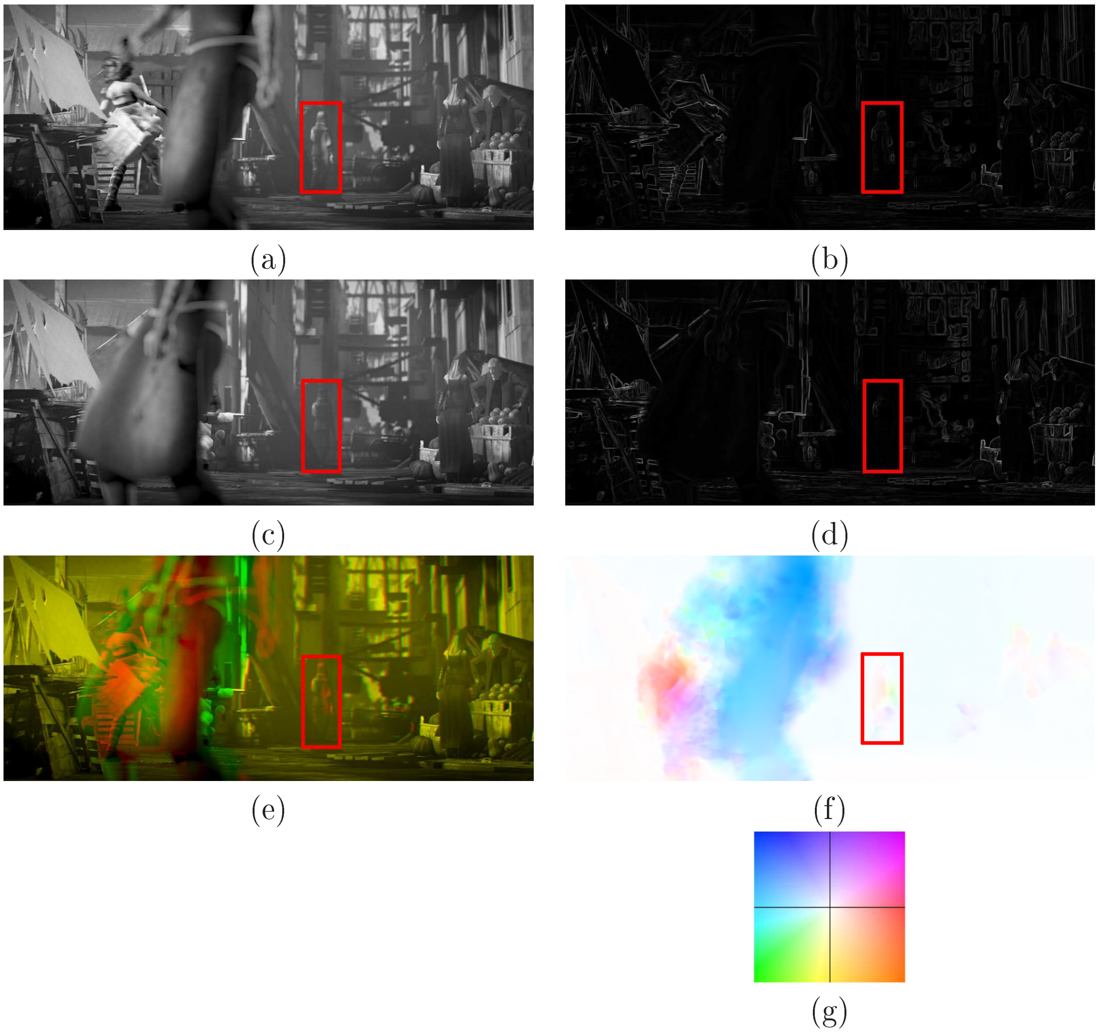


Figure 2.5: Large displacement and HS: (a) frame #5 from sequence Market_2 from MPI-Sintel; (b) image gradient of frame #5; (c) frame #10; (d) image gradient of frame #10; (e) frame #5 (in green) super-imposed on frame #10 (in red); (f) estimated motion using HS with coarse-to-fine approach; in each level of pyramid image sizes are reduced by half. Red rectangles shows the patches where the brightness of a man changed due to shading, consequently image gradient can not differentiate the man and HS flow field is not very accurate; (g) flow field color coding: the color represents the orientation of the vector and brightness stands for its magnitude.

(a) Original image (b) Structure part (c) Texture part

Figure 2.6: The Mequon sequence from Middlebury dataset; the original image is decomposed into a structural part and a textural part.

$$+\lambda_N \sum_{i,j} \sum_{(i',j') \epsilon \aleph_{i,j}} (\mid u_{i,j} - u_{i',j'} \mid + \mid v_{i,j} - v_{i',j'} \mid)\} \tag{2.5}$$

where u and v are the horizontal and vertical components of the optical flow field to be estimated from the source and target images I_1 and I_2; i, j indexes a particular image pixel location, $u_{i,j}$ and $v_{i,j}$ are elements of u and v respectively, λ is a regularization parameter, and ρ_D and ρ_S are the data and spatial penalty functions, $\aleph_{i,j}$ is the set of neighbors of pixel (i, j) in a possibly large area and λ_N is a scalar weight.

This method applies a median filter to intermediate flow values during incremental estimation and warping, which, according to Wedel et al. (2009b), successfully discard the outliers from the intermediate flow values. The method also adds a non-local term to compensate for the increase of energy in the objective function due to median filtering.

However, the SMF coarse-to-fine framework has limitations. As mentioned above, small structures may disappear at the coarser level. Therefore it may not estimate accurately their displacements. Moreover, since SMF uses median filter centred on a thin structure, the flow field gets dominated by the surrounding intensity values and suffers from over-smoothing. Figure 2.7c shows that small structures, such as the different parts of the flying dragon, disappear in the downsampled images, which yield a not very accurate estimated flow (see Figure 2.7d).

2.1.5 Brox and Malik's Large Displacement Optical Flow (LDOF) Method

To deal with the large flows, Brox and Malik (2011) proposed to incorporate information from image descriptors about shapes, colors or texture into a coarse-to-fine optical flow framework. Their method, Large Displacement Optical Flow (LDOF), uses Scale-Invariant Feature Transformation (SIFT) (Lowe, 2004), which detects local features in images and Histogram Oriented Gradients (HOG) (Dalal and Triggs, 2005), which also describes local image features like SIFT, but computed on a dense grid of uniformly spaced cells and uses overlapping local contrast normalization for improved accuracy. After adding information derived from image descriptors to the coarse-to-fine approach of Horn and Schunck (1980), the objective function (see equation 2.3) becomes:

$$
\begin{aligned}
E\left(w(x)\right) = \textstyle\int \Psi\left(\left|I_2(x+w(x)) - I_1\left(x\right)\right|^2\right) dx \\
+\gamma \textstyle\int \Psi\left(\left|\nabla I_2\left(x+w\left(x\right)\right) - \nabla I_1\left(x\right)\right|^2\right) dx \\
+\beta \sum \textstyle\int \rho_j\left(x\right) \Psi\left(\left(u\left(x\right) - \underset{j=1}{\overset{5}{u_j\left(x\right)}}\right)^2 + \left(v(x) - v_j\left(x\right)\right)^2\right) dx \\
+\alpha \int \Psi\left(\left|\nabla u\left(x\right)\right|^2 + \left|\nabla v\left(x\right)\right|^2 + g\left(x\right)^2\right) dx
\end{aligned}
\tag{2.6}
$$

where $w(x) = (u, v)$ is the displacement field between the source, I_1, and target image, I_2, $x = (x, y)$ denotes a point in the image. $\left(u_i, v_j\right)(x)$is one of the motion vectors derived at position x by region matching (j matching nearest 5 nearest neighbors). $\alpha = 100$, $\beta = 25$ and $\gamma = 5$ are tuning parameters (as suggested by Brox and Malik (2011)), they steer the importance of smoothness, region correspondences, and gradient constancy, respectively. If there is no correspondence at this position, $\rho_j\left(x\right) = 0$. Otherwise, $\rho_j\left(x\right) = c_j$ where,

$$
c_j\left(i\right) := \begin{cases} \frac{\bar{d}^2(i) - d^2(i,j)}{d^2(i,j)} & \bar{d}^2\left(i\right) > 0 \\ 0 & else \end{cases}
\tag{2.7}
$$

Here, $d^2\left(i, j\right)$ is the Euclidean distance between the two patches after deformation correction and $\bar{d}^2\left(i\right)$is the average Euclidean distance among the 10 nearest neighbors. $\Psi\left(s^2\right) = \sqrt{s^2 + 10^{-6}}$ in order to deal with outliers in the data as well as in the smoothness assumption as suggested by (Brox et al., 2004) and $g\left(x\right)$ is boundary map as recommended by (Arbelaez et al., 2009).

Figure 2.8d shows the estimated motion between frame #7 and #8 from the Temple3 sequence using LDOF. For the motion between this pair of images, LDOF shows better estimation than SMF. However, it does not adequately estimate the motions of small structures like fingers and hair due to ambiguous and false matching

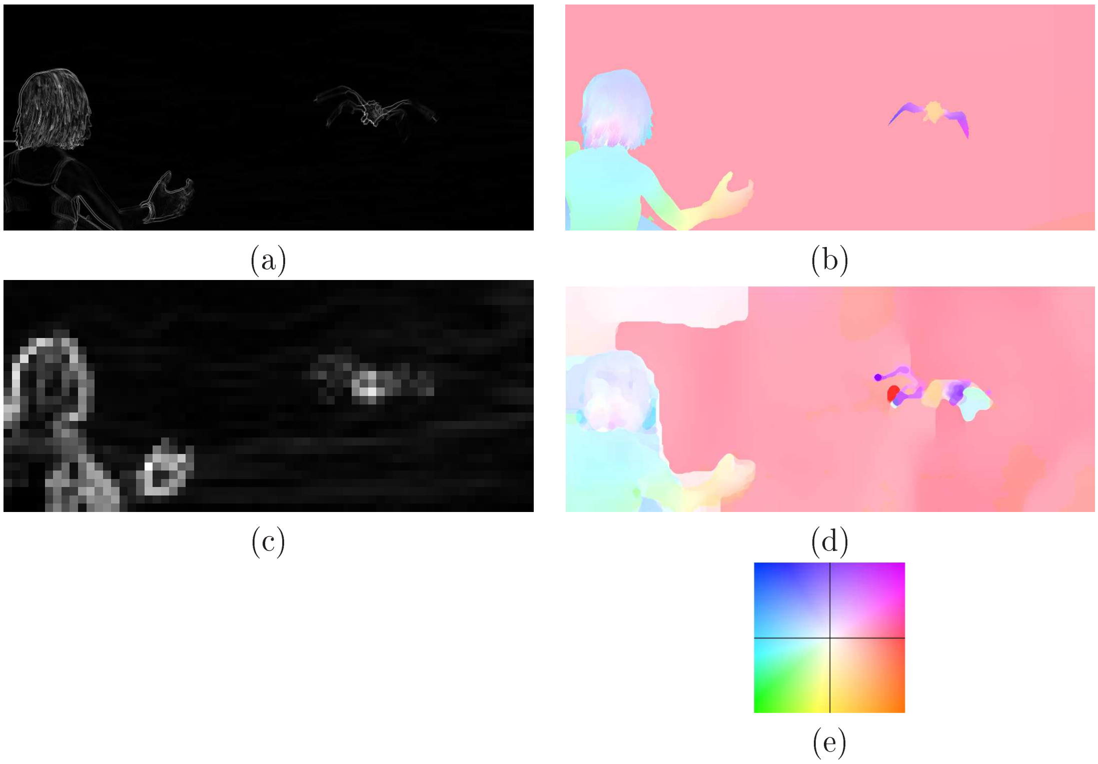


Figure 2.7: The SMF approach: (a) super-imposed image gradients of frame #7 and #8; (b) ground truth flow; (c) frames are downsampled to $\frac{1}{16}$ of their sizes and zoomed; (d) flow estimated using SMF; (e) flow field color coding: the color represents the orientation of the vector and brightness stands for its magnitude.

(see Figure 2.7d). Further analysis of the performance of LDOF is discussed in section 5.5

2.1.6 DeepFlow

Weinzaepfel et al. (2013) proposed to improve LDOF by using a deep matching algorithm based on a multistage architecture, uses inter-leaving convolutions and max-pooling. Unlike LDOF, DeepFlow does not use HOG which a rigid descriptor. Rather, it normalizes the data term to downweight the impact of locations with high spatial image derivatives. In the coarse-to-fine approach, it uses different weights at each level to downweight at finer scales based on feature matches as proposed by Stoll et al. (2013). This method considers the matching energy of the baseline method that allows to reliably identify locations where feature matches can be particularly useful and thus enables to sort out unreliable matches before the integration. Figure

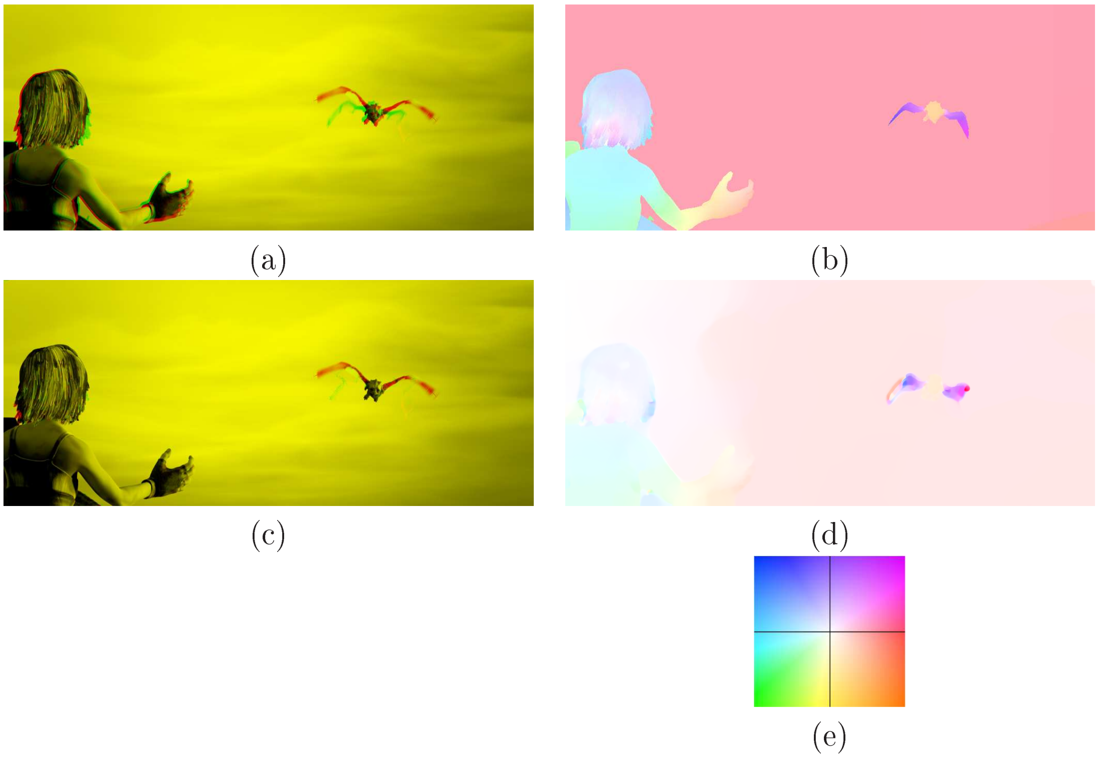


Figure 2.8: The LDOF approach: (a) frame #7 (in green) super-imposed on frame #8 (in red); (b) ground truth flow; (c) registered image using LDOF (in green) super-imposed on target image (in red); (d) estimated flow; (e) flow field color coding: the color represents the orientation of the vector and brightness stands for its magnitude.

2.9d shows the estimated motion between frame #7 and #8 using DeepFlow. Unlike LDOF, DeepFlow can identify image features of the fingers and estimates motion of the fingers better than LDOF.

Descriptor matching algorithms often suffer from ambiguity. In particular, images with highly repetitive structures may create several matches in a local neighborhood and the algorithm may not be able decide the correct transformations. Also, regions with weak texture often do not yield reliable image features and so their correspondence is ambiguous. Figure 2.9d illustrates such a case where repetitive small structures like the hair of Sintel, the female protagonist, create ambiguity for DeepFlow. Consequently, it cannot estimate a very accurate motion.

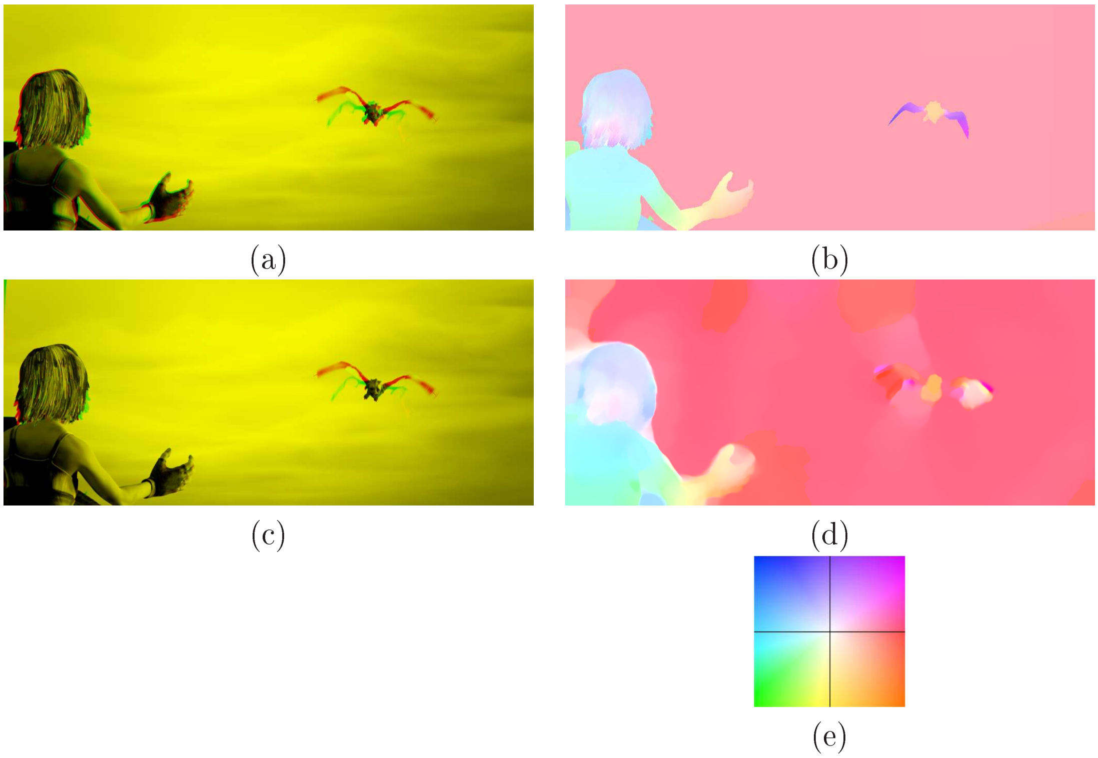
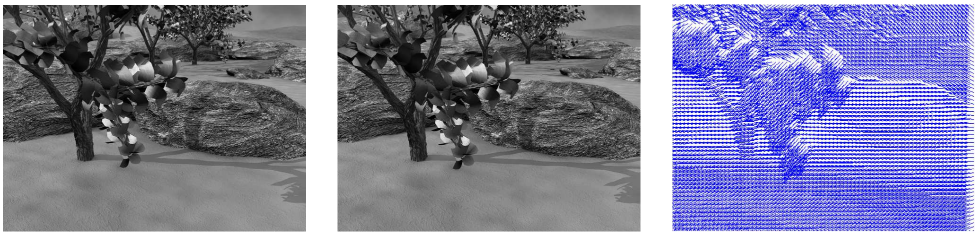


Figure 2.9: The DeepFlow approach: (a) frame #7 (in green) super-imposed on frame #8 (in red); (b) estimated motion using DF; (c) registered image using DF (in green) super-imposed on target image (in red); (d) ground truth motion; (e) flow field color coding: the color represents the orientation of the vector and brightness stands for its magnitude.



(a) (b) (c)

Figure 2.10: The optical flow assumption: (a) frame #10; (b) frame #11; (c) ground truth deformation field. Note how the intensity of small structures stays constant across the two consecutive frames and how regular the vectors in a small neighborhood are in the ground truth deformation field.

2.2 Learning The Optical Flow

To improve optical flow estimation, most current methods use mainly two constraints on image motion: data conservation and spatial coherence. The data conservation constraint is based on the idea that the surfaces of objects generally persist in time and the intensity structure of a small region in one image remains constant over time, although its position may change (Black and Anandan, 1996). The spatial coherence constraint is derived from the observation that surfaces have spatial extent and neighboring pixels in an image are likely to belong to the same surface. Since the motion of a neighborhood on a smooth surface transforms gradually in most of the cases, smoothness can be enforced on the motion of neighboring points in the image plane (Horn and Schunck, 1980). Figure 2.10 shows the ground truth deformation field of two consecutive frames (#10 and #11) from Grove2 sequence of Middlebury dataset. Small intensity structures in the frames remain almost constant over time and vectors in small neighborhoods are very regular.

A great deal of work in the field of optical flow has been devoted to improving both the data conservation and spatial coherence. However, learning approaches have only attracted limited attention to date. Simoncelli et al. (1991) first introduced a probabilistic framework to compute image gradients. Their model produces flow vector confidence information and uses it to address the problem of inherent uncertainty of optical flow. It uses a Gaussian noise model to compute the maximum likelihood of the estimated motion.

To address the problem of motion discontinuity, Black et al. (1997) proposed a framework for learning parameterized models of image motion. This method uses

Principal Components Analysis (PCA) to compute a low-dimensional model for spatial structure of the flow fields. Motion discontinuity is recovered with a linear combination of a small number of the basis motions taken from a training set.

Xue et al. (2006) proposed a statistical model of deformation (SMD) that uses an *a priori* statistics of high-dimensional displacement fields to improve accuracy of image registration. It utilizes wavelet-based decompositions coupled with PCA in each wavelet band to compute probability density function of high dimensional deformation fields. This method can improve accuracy of image registration even with a relatively small number of training samples.

Freeman et al. (2000) introduced a learning-based method for estimating motion of low-level vision. This method models relationships between neighboring image patches using a Markov network by assigning each patch as a node connected by lines. It solves the Markov network with a learning phase where the parameters of the network connections are learned from training data. These learned parameters are used in a Bayesian belief propagation approach that estimates the flow.

Roth and Black (2007a) also proposed to use Markov random fields (MRF) to model the spatial statistics of optical flow fields. Their method incorporates the Field of Experts (FoE) of flow priors into standard optical flow algorithm and obtains statistically significant accuracy improvements (FoE is a model of prior probability of images and optical flow fields (Roth and Black, 2005)). In contrast to other MRF models, it uses larger cliques of pixels (a subset of vertices of an undirected graph where every two vertices in the subset are connected by an edge). This method learns the appropriate clique potentials from training data. The learned cliques are used as a spatial regularizer for final flow estimation.

Sun et al. (2008) introduced a method that models both spatial and brightness statistics using a Steerable Random Field (SRF) (Roth and Black, 2007b). By using naturalistic training sequences with ground truth flow it develops a learning framework of optical flow that captures the spatial statistics of the flow field, the statistics of brightness inconstancy and the relation of flow boundaries to the image intensity structure. Image intensity boundaries are used to improve the accuracy of optical flow near motion boundaries and SRF are utilized to model the conditional statistical relationship between the flow and the image sequence. It also incorporates a statistical model of the data term by extending the Field-of-Experts formulation (Roth and Black, 2005) to the spatio-temporal domain to model temporal changes in image features.

Li and Huttenlocher (2008a,b) presented a continuous-state MRF model of optical flow (proposed by Szeliski (1990)) by minimizing the training loss for a set of ground-truth images using Simultaneous Perturbation Stochastic Approximation (SPSA) (Spall, 1992). It uses SPSA to minimizes the training loss that occurs due to unseen or unmatched data, for instance data that appears due to occlusion. This method does not require approximations to address the intractable nature of maximum-likelihood estimation.

To overcome the same problem of unmatched data with occlusion, Mac Aodha et al. (2013) introduced a supervised learning approach that learns confidence measures for optical flow. Their method estimates a per-pixel confidence for optical flow vectors that are comprised of multiple different measures, incorporating a broad range of motion and appearance cues and the photoconstancy residual.

To the best of our knowledge, none of these methods used learning approach to handle large displacements. In this thesis, we introduce a novel learning approach to optical flow capable of dealing with large displacements. We propose to refine an initial estimate of the flow field by replacing each displacement vector by a linear combination of displacement vectors at the centre of similar patches taken from an *a priori* code-book. This approach does not require a coarse-to-fine or warping strategy, which makes it possible to handle fine scale structures with large displacements.

2.3 Datasets Used in Our Experiments

As our plan is to learn motions from a learning set of *a priori* registered images, we chose datasets consisting of image streams with motion; for instance, frames from a video sequence with moving objects or thoracic images with periodic cardiac motion. We used two datasets exhibiting thoracic and cardiac motions: a 2-D dynamic MRI scan and a 3-D gated CT scan. Unfortunately, these datasets do not offer ground truths of pair-wise motions. Since ground truth motion is invaluable in quantifying the accuracy of the estimated transformations (see section 2.4 for detail), we also used the synthetic MPI-Sintel dataset (http://sintel.is.tue.mpg.de/) with publicly available ground truths.

2.3.1 2-D Dynamic Magnetic Resonance Imaging Dataset

The 2-D dMRI consists of a series of 300 sequential MR images of the thorax of a volunteer breathing normally in a Siemens 1.5T scanner. These images exhibit both respiratory and cardiac motions. All images are 64×64 pixels in size. This anonymised dataset has been provided by Mirada Medical (www.mirada-medical.com). Mirada Medical is one of the collaborators of this PhD.

Learning set

We prepared 30 trials from this dataset (see Table 2.1). Column 1 shows the trial index, learning sets are given in column 2, column 3 and 4 mention the nature of respiratory and cardiac motions occurred in the learning set respectively. Unless otherwise stated, we used our own implementation of the original Lucas-Kanade optical flow method with coarse-to-fine approach (in each level of pyramid image sizes are reduced by half) to register the consecutive images and generate pair-wise deformation fields for the learning set. We specifically selected frames in those periodic sequences which exhibit the largest displacements, taking into account the period indeed, to make sure we did not test images which were similar even though they were far apart in the sequence.

Test Cases

Column 4 shows the test cases which are disjoint set from the learning set. In the test cases, we chose source and target images that had large displacements in the dynamic MR scans.

Table 2.1: List of trials

Trials	learning set	Respiration	Cardiac motion	Test case
1	Images 41 to 46	Inhalation 41 to 46	Systole phases 41, 43, 45; Diastole phases 42, 44, 46	Source: 82 Target: 83
2	Same as above	Same as above	Same as above	Source: 83 Target: 84
3	Images 46 to 51	Exhalation 46 to 51	Systole phase 47, 49, 51; Diastole phase 46, 48, 50	Source: 72 Target: 73
4	Images 51 to 56	Exhalation 51 to 53; Inhalation 54 to 56	Systole phases 51, 53, 55; Diastole phases: 52, 54, 56	Source: 92 Target: 93
5	Same as above	Same as above	Same as above	Source: 161 Target: 162
6	Images 60 to 65	Exhalation 60 to 65	Systole phases 61, 63, 65; Diastole phases: 60, 62, 64	Source: 53 Target: 54
7	Same as above	Same as above	Same as above	Source: 78 Target: 79
8	Same as above	Same as above	Same as above	Source: 79 Target: 80
9	Same as above	Same as above	Same as above	Source: 100 Target: 101

10	Same as above	Same as above	Same as above	Source: 102 Target: 103
11	Images 77 to 82	Exhalation 77 to 80 Inhalation 80 to 82	Systole phase 77, 79, 81; Diastole phase 78, 80, 82	Source: 108 Target: 109
12	Images 90 to 95	Exhalation 90 to 93 Inhalation 94 to 95	Systole phase 91, 93, 95; Diastole phase 90, 92, 94	Source: 76 Target: 77
13	Same as above	Same as above	Same as above	Source: 106 Target: 107
14	Same as above	Same as above	Same as above	Source: 235 Target: 236
15	Images 90 to 96	Exhalation 90 to 93 Inhalation 94 to 965	Systole phase 91, 93, 95; Diastole phase 90, 92, 94, 96	Source: 248 Target: 249
16	Images 151 to 156	Inhalation 151 to 156	Systole phase 151, 153, 155; Diastole phase 152, 154, 156	Source: 285 Target: 286
17	Images 211 to 216	Inhalation 211 to 213 Exhalation 213 to 216	Systole phase 212, 214, 216; Diastole phase 211, 213, 215	Source: 114 Target: 115
18	Images 231 to 236	Exhalation 231 to 236	Systole phase 232, 234, 236; Diastole phase 231, 233, 235	Source: 274 Target: 275
19	Images 231 to 238	Exhalation 231 to 236 Inhalation 237 to 238	Systole phase 232, 234, 236, 238; Diastole phase 231, 233, 235, 237	Source: 276 Target: 277
20	Images 231 to 239	Exhalation 231 to 236 Inhalation 237 to 239	Systole phase 232, 234, 236, 238 ; Diastole phase 231, 233, 235, 237, 239	Source: 277 Target: 278
21	Images 246 to 251	Exhalation 246 to 249 Inhalation 250 to 251	Systole phase 246, 248, 250; Diastole phase 247, 249, 251	Source: 277 Target: 278
22	Images 246 to 253	Exhalation 246 to 249 Inhalation 250 to 253	Systole phase 246, 248, 250, 252; Diastole phase 247, 249, 251, 253	Source: 278 Target: 279
23	Images 251 to 256	Inhalation 251 to 256	Systole phase 252, 254, 256; Diastole phase 251, 253, 255	Source: 96 Target: 97
24	Images 258 to 263	Exhalation 258 to 263	Systole phase 258, 260, 262; Diastole phase 259, 261, 263	Source: 232 Target: 233
25	Same as above	Same as above	Same as above	Source: 277 Target: 278
26	Same as above	Same as above	Same as above	Source: 278 Target: 279
27	Images 270 to 275	Inhalation 270 to 275	Systole phase 270, 272, 274; Diastole phase 271, 273, 275	Source: 57 Target: 58
28	Images 273 to 278	Inhalation 273 to 276 Exhalation 277 to 278	Systole phase 274, 276, 278; Diastole phase 273, 275, 277	Source: 231 Target: 232
29	Images 273 to 279	Inhalation 273 to 276 Exhalation 277 to 279	Systole phase 274, 276, 278; Diastole phase 273, 275, 277, 279	Source: 233 Target: 234
30	Images 274 to 279	Inhalation 274 to 276 Exhalation 277 to 279	Systole phase 274, 276, 278; Diastole phase 275, 277, 279	Source: 235 Target: 236

As an illustration, Figure 2.11 shows the MR images used in trial #4. Figure 2.11a and 2.11b shows the learning set and test case respectively.

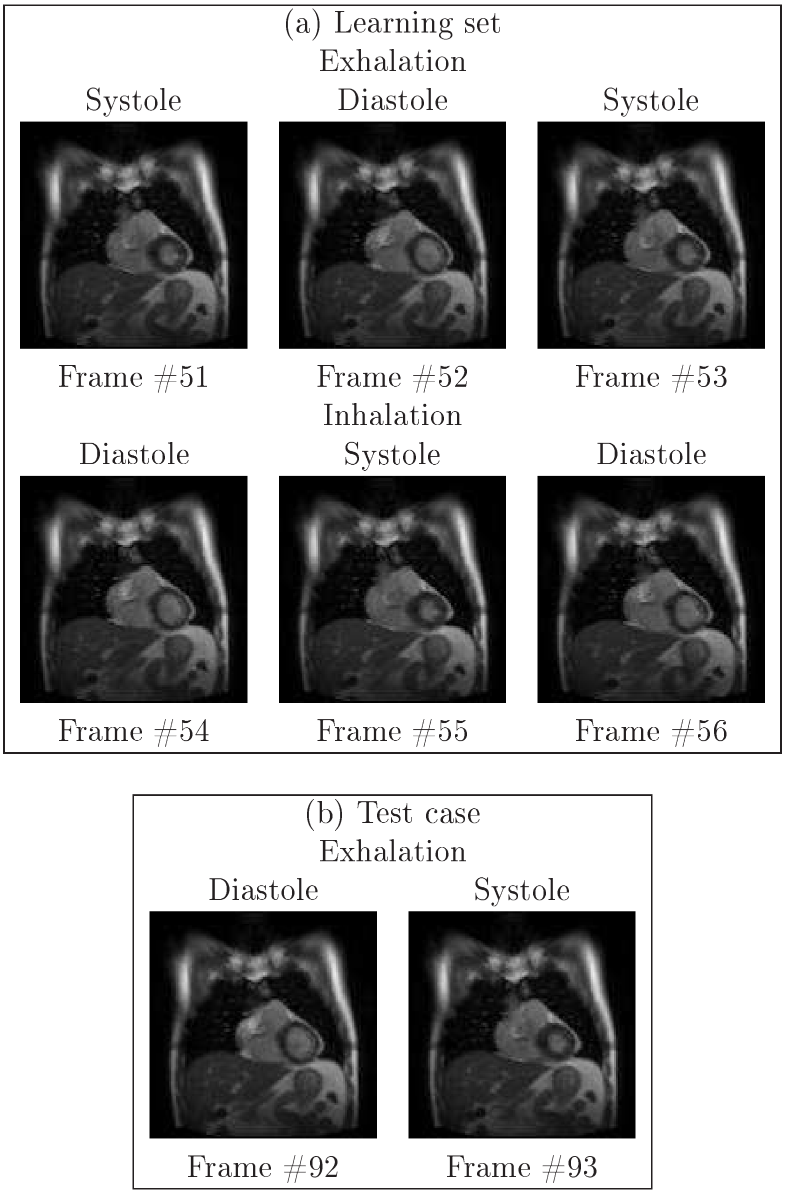


Figure 2.11: Trial #4 from 2D MRI dataset; (a) learning set; (b) test case.

2.3.2 3-D Gated CT Dataset

This dataset consists of nine sequential gated-CT 3-D images of the thorax of a volunteer. It is also provided by Mirada Medical (www.mirada-medical.com). All images were $512 \times 512 \times 100$ pixels in size. There were many complex motions in these images, including respiratory and cardiac motions. The inhalation process starts at the first image and ends in the fifth image. The exhalation process starts and ends in the ninth image. Images 1, 3, 5, 7 and 9 correspond to the systolic phase and images 2, 4, 6 and 8 to the diastolic phase.

Learning Set

Unless otherwise stated, we also used LK to perform pairwise registrations between the 3rd image and the 7th image. As a result, we have 4 pairwise deformation fields of size $512 \times 512 \times 100$ in our learning set.

Test Cases

For the test cases, we arbitrarily selected 400 sub-images (each of size $21 \times 21 \times 21$ pixels) from the 2nd image and registered them to corresponding sub-images in the 5th image.

2.3.3 MPI-Sintel Dataset

Butler et al. (2012) used data from the Durian Open Source Movie Project (Roosendaal, 2010) to render scenes under conditions of varying complexity. It is publicly available to download in http:// sintel.is.tue.mpg.de. The MPI-Sintel dataset consists of much longer image sequences, ground truth flow is available for all frames, it exhibits large non-rigid motions and much more complexity (blur, atmosphere, specular surfaces, etc.). 35 clips were selected from the full movie by the authors. Apart from six shorter action sequences, each sequence is 50 frames long, giving 49 pair-wise flow fields per clip. Ground truth flows were estimated using Classic Non Local Fast method (Wang et al., 2006; Sun et al., 2010). At the time of creating the MPI Sintel dataset, Classic Non Local Fast performed the best on estimating small motions. This is why Butler et al. (2012) chose Classic Non Local Fast method to compute ground truth flows. For two images that are further apart in a sequence, the ground truth motion can be computed by performing vector composition of the pair wise ground truth flows in between those two images. We selected 93 trials from 8 sequences (Alley1, Ambush7,

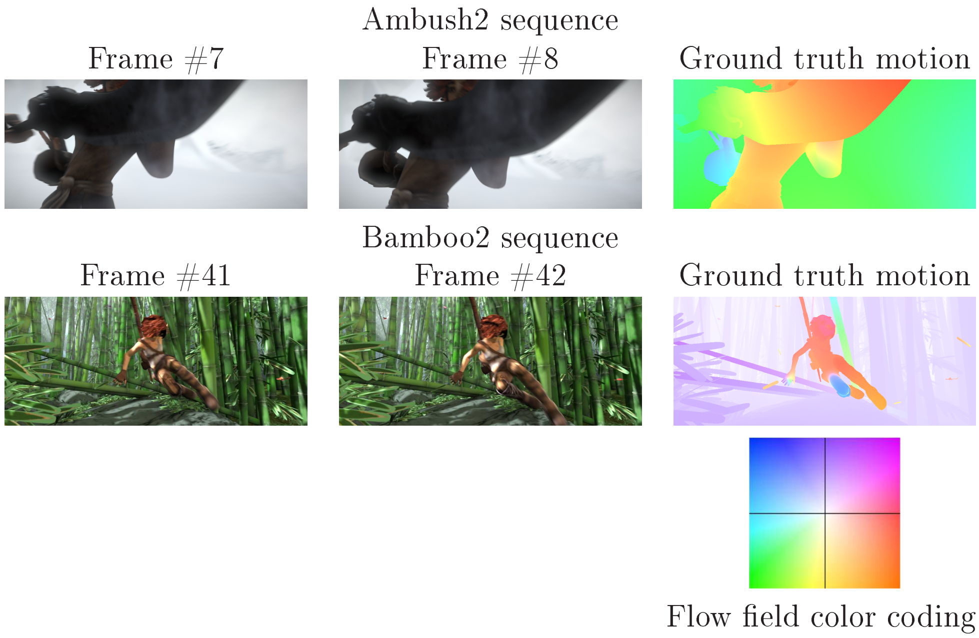


Figure 2.12: Scenes from the Ambush2 and Bamboo2 sequences in the MPI-Sintel dataset; the colors in the flow field color coding represents the orientation of the vector and brightness stands for its magnitude.

Bamboo1, Bamboo2, Market2, Market6, Shaman2 and Shaman3) based on their exhibiting large motions of small structures, since it is the focus of this thesis. Figure 2.12 shows two pairs of scenes and their corresponding ground truth motion.

2.3.4 Rationale Behind Prefering Medical Images in the Development of the Proposed Method

MPI-Sintel dataset is a synthetic dataset that represents naturalistic scenes. Unlike medical images, MPI Sintel dataset have some natural scene issues like occlusion, appearance of new structures and shadows. Estimated motions of MPI-Sintel dataset are highly effected by these issues (Butler et al., 2012). On the other hand, the medical image datasets that we used in our experiment do not have such issues. Thus, in each step of the development of the proposed method, we use medical image datasets to check its performance in estimating large displacements (which is our main focus) where the results are not effected by aforementioned critical issues of the naturalistic images. However, once the whole method is developed, we also use the MPI-Sintel dataset to evaluate the performance of the proposed method with other state-of-the-art methods (for detail, see section 5.5).

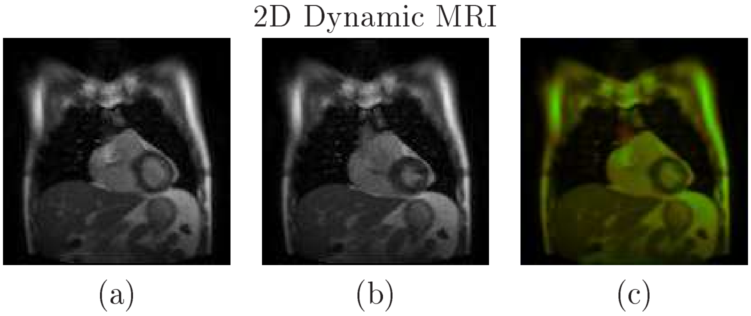


Figure 2.13: 2D MRI dataset; (a) frame #82; (b) frame #85; (c) Motion is estimated using DeepFlow method and then transformed image (in green) super-imposed on target image (in red).

2.4 Estimating the Accuracy of the Flow

In this section, we discuss the methods that we will be using in our experiments to evaluate both qualitatively and quantitatively the accuracy of the produced optical flows

2.4.1 Qualitative Performance: Super-Imposition of Registered Image

After estimating the motion between two images, we can transform the source image and super-impose the transformed image on top of the target image. We can use two different colors for the target and transformed image. Figure 2.13 shows an example where the motion between frame #82 (source) and #85 (target) from the 2-D dynamic MRI dataset is estimated using the DeepFlow method. The source image is transformed using the estimated motion and super-imposed in green on top of the target frame in red. We can see inaccurate alignments either in red or in green while the accurate transformations can be seen in yellow.

However, it is less convenient to compare performances of different algorithms using this method since it does not generate quantitative values of accuracy of the estimated motion.

2.4.2 Quantitative Performance:

2.4.2.1 Mean Squared Error (MSE) Based on Intensity

We can compute the Squared Differences (SD) of intensity values at each pixel between target and transformed images and calculate a Mean Squared Error (MSE). This method generates quantitative values of accuracy of estimated transformation without ground truth motion. However, MSEs can be highly biased by extreme values and changes in lighting conditions.

2.4.2.2 Average Angular Error (AAE)

In cases where ground truth motions are available, the most commonly used measure of performance for optical flow is the angular error (AE) which was proposed by Barron et al. (1994). The AE between a flow vector (u, v) and the ground-truth flow (u_{GT}, v_{GT}) is the angle in 3D space between $(u, v, 1.0)$ and $(u_{GT}, v_{GT}, 1.0)$. The AE can be computed by taking the dot product of the vectors, dividing by the product of their lengths, and then taking the inverse cosine:

$$AE = \cos^{-1}\left(\frac{1.0 + u \times u_{GT} + v \times v_{GT}}{\sqrt{1.0 + u^2 + v^2}\sqrt{1.0 + u_{GT}^2 + v_{GT}^2}}\right) \tag{2.8}$$

The goal of the AE is to provide a relative measure of performance that avoids the divide by zero problem for zero flows.

2.4.2.3 Average Endpoint Error (AEPE) of Deformation Field

With the AAE approach, errors in regions of zero motion are penalized more than errors in a region of smooth non-zero motion. The AAE also contains an arbitrary scaling constant (1.0) to convert the units from pixels to degrees. Therefore, Otte and Nagel (1994) proposed the error in flow endpoint (EPE) which is, according to Baker et al. (2011), more accurate than AE. Endpoint Error is defined by

$$EE = \sqrt{(u - u_{GT})^2 + (v - v_{GT})^2} \tag{2.9}$$

MPI-Sintel datasets use both AAE and $AEPE$ as accuracy measures.

2.4.3 Statistical Analysis

We use repeated measures ANOVA (analysis of variance) to determine whether there are any significant differences across registration methods in terms of MSE, AAE or AEPE. This choice was motivated by the fact that the distribution of those measures was generally normal, with relatively little skew and kurtosis. For instance, the skews and kurtoses of the MSEs for the first experiment (see Table B.1) are (0.85, 1.66) for LK, (0.78, 1.06) for degraded LK, (0.36, 0.06) for the learning approach with vector difference, (0.24, 0.44) for for learning approach with angular difference and (0.73, 0.87) for the learning approach with magnitude difference.

Chapter 3

A Learning Approach to Optical Flow

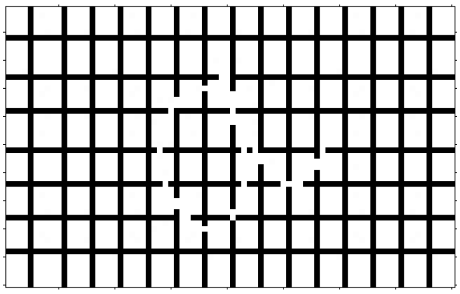
3.1 Introduction

The previous chapter discussed the problems posed by large motions when estimating optical flows with a focus on the estimation of the motion of small structures with large displacements. This chapter introduces a novel learning approach to optical flow capable of dealing with large displacements. The learning algorithm estimates the flow between two non-consecutive images in a sequence on the basis of a learning set of flows estimated *a priori* between different consecutive images in the same sequence. The key idea is to use the accurate flows estimated *a priori* between consecutive images to help improve the potentially less accurate flows estimated *online* between images further apart. Our approach is inspired by non-local means filtering.

3.2 Non-Local Means Filtering

Buades et al. (Buades et al., 2005) introduced the non local means filter as an image denoising method. This algorithm takes advantage of the high degree of redundancy of images. Given a patch in an image, the non local means (NLM) algorithm replaces it with a weighted average of neighborhood patches that are similar to it. In other words, this filtering algorithm takes a mean of all pixels in the neighbourhood of a pixel, weighted by how similar these pixels are to the target pixel.

Suppose we are considering pixel i for denoising. Now if we find another pixel j in the same image that has a neighborhood around it similar to the neighborhood around i then we can use the value of j for predicting the value of i. Let $i\,(x, y)$ be a pixel in a noisy image f on a bounded domain $R\epsilon\mathbb{R}^2$:


(a) Noisy image

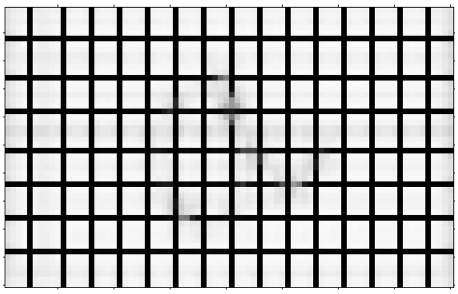

(b) Filtered image

Figure 3.1: An example of image denoising using non-local means filter.

$$NLM\left(f\right)\left(i\right)=\frac{1}{C\left(i\right)}\int_{R}\exp^{-\frac{\int_{\mathbb{R}^2}G_a(t)|f(i+t)-f(j+t)|^2dt}{h^2}}f\left(j\right)dj \tag{3.1}$$

where G_a is a Gaussian kernel with standard deviation a, h is a filtering parameter and $C\left(x\right)=\int_{R}\exp^{-\frac{\int_{\mathbb{R}^2}G_a(t)|f(i+t)-f(z+t)|^2dt}{h^2}}dz$ is a normalization factor.

An example of non-local means filter is given in Figure 3.1. We used 5×5 pixels for the size of the similarity window, 11×11 pixels for the size of the neighborhood and 0.05 for the standard deviation a, the filtering parameter $h=0.1$. Notice how the white noises in Figure 3.1a are filled up by NLM, which took advantage of the regularity of the grid and then incorporated that information into the filtered image.

Figure 3.2 shows another example where NLM is used to denoise a natural image (Lena). Here, the size of the image is 500×500 pixels. We used 5×5 pixels for the size of the similarity window, 15×15 pixels for the size of the neighborhood and 0.01 for the standard deviation a and the filtering parameter $h=0.1$. The NLM filtered the white noises in the Figure 3.2c. The NLM filtering method has reduced the MSE of intensity from 971.81 (between Figure 3.2a and Figure 3.2b) to 782.49 (between Figure 3.2a and Figure 3.2c).

3.3 A Learning Approach to Optical Flow

Motivated by non local means, we propose a learning algorithm that learns from redundant patterns of vectors in a learning set of deformation fields. The datasets that

(a) (b) (c)

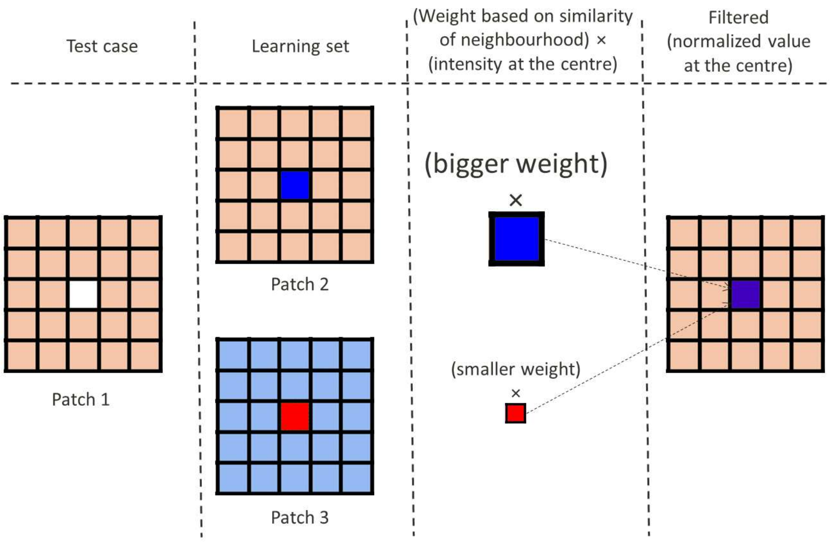
Figure 3.2: Denoising Lena with NLM : (a) original image, (b) with added white noise, (c) filtered with NLM..

we are using in our experiments have sequential images with regular motions such as cardiac and respiratory motions in medical images, or the flapping wings of dragon in MPI-Sintel dataset etc (for detail see section 2.3). Given two non-consecutive images in the sequence, we aim to refine the flow $(\mathbf{u}^0, \mathbf{v}^0)$ computed between them using a given optical flow algorithm (e.g. the LK). At every pixel (m,n) , we consider the patch $\{(m,n), \mathbf{q}_{m,n}\}$ around it, with $\mathbf{q}_{m,n}$ the matrix of displacement vectors around (m,n), and look in a learning set for similar patches. Here, we take advantage of the high degree of redundancy of vector patches across the images from the sequence to filter the initial flow by replacing the displacement vector $(u_{m,n}, v_{m,n})$ at pixel (m,n) by a Gaussian weighted average of the displacement vectors at the center of the most similar patches, with the aim to make $(\mathbf{u}^0, \mathbf{v}^0)$ more similar to the flows in the learning set.

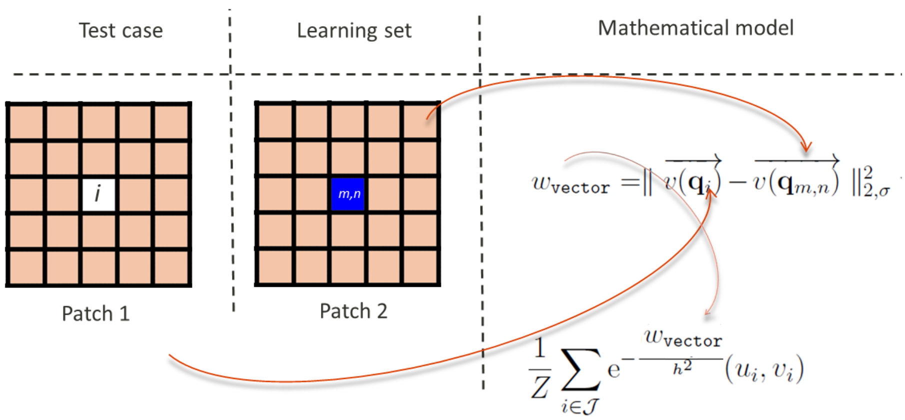
In a similar fashion to the non-local means approach (Buades et al., 2005), the filtering phase replaces the displacement vector $(u_{m,n}, v_{m,n})$ at pixel (m,n) by:

$$(u_{m,n}, v_{m,n}) = \frac{1}{Z} \sum_{i \in \aleph} \mathrm{e}^{-\frac{w_{\mathtt{vector}}}{h^2}} (u_i, v_i) \tag{3.2}$$

where (u_i, v_i) is the displacement vector at the center of $\mathbf{q}_i$; $w_{\mathtt{vector}}$ is the weight that depend on the similarity between the patch at (m,n) and the selected ones from the learning set; h controls the decay of the weights, $\aleph$ is the size of the neighborhood and Z is an overall normalization factor. The pseudo-code of the implementation of equation 3.2 is given in Algorithm 3.1.



(a)



(b)

Figure 3.3: Block diagram illustrating the effect of similarity measure used between the initial flow and the learning set. (a) Patch 1 is more similar to Patch 2 compared to Patch 3. Thus, larger weight is given with the value at the centre of Patch 1. (b) Patches with stronger similarity contributes more in the equation 3.2.

Algorithm 3.1 Pseudo Code of Learning Algorithm

```
function [ learned_flow_field ] = learningAlgorithm( initial_flow, learning_set,
threshold)
% initial_flow : the initially estimated flow field computed ...
%       using an standard optic flow algorithm such as HS
% learning_set : the set of all deformation fields in a priori
% the similarity measure similarity_vector

loop_1 : for each index (i,j) in the initial_flow
  {
  % initialization of the weight of similarity measures for each patch
  sum_of_weight=0;
  loop_2 : for each patch in the neighbourhood
    {
    compute the similarity_vector for (i,j) using the equation 3.2

    sum_of_weight=sum_of_weight + similarity_vector

    weighted_vector=weighted_vector + ...
      similarity_vector × vector_at_the_center_of_the_patch
    }
    end_of_loop_2

  % normalization
  learned_flow_field(i,j)=weighted_vector / sum_of_weight;
  }
end_of_loop_1

regularize the learned_flow_field using Gaussian filter
```

3.4 Selecting a Patch Similarity Measures

Given a patch in the initial flow$(\mathbf{u}^0, \mathbf{v}^0)$, we look for similar patches in the learning set. Figure 3.3 illustrates how patches with stronger similarity contribute more in equation 3.2.

With the polar representation of vectors in mind (magnitude and angle), we consider three similarity measures:

1. Weighing sum based on vector difference (difference of both magnitude and angle): $w_{\texttt{vector}} = \sum \parallel \overrightarrow{v(\mathbf{q}_i)} - \overrightarrow{v(\mathbf{q}_{m,n})} \parallel_\sigma^2$ is the Gaussian weighted sum of squared differences of vectors of patches with standard deviation σ.

2. Weighing sum based on magnitude difference: $w_{\texttt{vector}} = \sum \parallel \ |v(\mathbf{q}_i)| - |v(\mathbf{q}_{m,n})| \ \parallel_\sigma^2$ is the Gaussain weighted sum of squared differences of magnitudes of vector patches and with standard deviation σ.

3. Weighing sum based on angular difference: $w_{\texttt{vector}}(\mathbf{q}_i, \mathbf{q}_{m,n}) = \sum \parallel \theta(\mathbf{q}_i) - \theta(\mathbf{q}_{m,n}) \parallel_\sigma^2$ is the Gaussian weighted sum of squared differences of polar angles of vectors in vector patches with standard deviation σ.

Here, $\overrightarrow{v(\mathbf{q}_i)}$ is the patch in the initial flow and $\overrightarrow{v(\mathbf{q}_{m,n})}$ is a patch in the learning set. All of the weighing sums are normalized using their standard deviation σ.

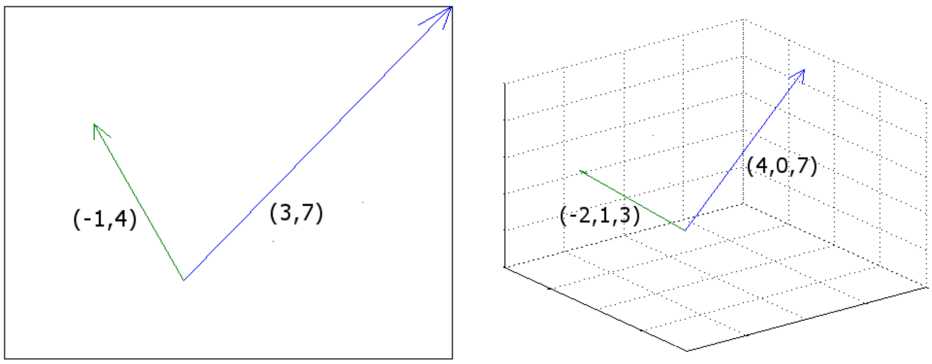
We can compute an angle between two vectors using either of the three equations 3.3, 3.4 or 3.5.

$$\theta = \cos^{-1}\left(\frac{\overrightarrow{v(\mathbf{q}_i)} \cdot \overrightarrow{v(\mathbf{q}_{m,n})}}{\|\overrightarrow{v(\mathbf{q}_i)}\|\|\overrightarrow{v(\mathbf{q}_{m,n})}\|}\right) \tag{3.3}$$

$$\theta = \sin^{-1}\left(\frac{\|\overrightarrow{v(\mathbf{q}_i)} \times \overrightarrow{v(\mathbf{q}_{m,n})}\|}{\|\overrightarrow{v(\mathbf{q}_i)}\|\|\overrightarrow{v(\mathbf{q}_{m,n})}\|}\right) \tag{3.4}$$

$$\theta = \tan^{-1}\left(\frac{\|\overrightarrow{v(\mathbf{q}_i)} \times \overrightarrow{v(\mathbf{q}_{m,n})}\|}{\overrightarrow{v(\mathbf{q}_i)} \cdot \overrightarrow{v(\mathbf{q}_{m,n})}}\right) \tag{3.5}$$

For orthogonal vectors (angle = 90 or 270 degrees) cosine gives poor accuracy as $\cos(90) = \cos(270) = 0$. Similarly, for parallel vectors (angle = 0 or 180 degrees) sine gives poor accuracy as $\sin(0) = \sin(180) = 0$. Both equation 3.3 and 3.4 can suffer from divide by zero error. Whereas equation 3.5 gives accurate results for angles 0, 90, 180 and 270. Moreover, it can tackle divide by zeros error as $\tanh(\inf) = 90^o$. Therefore, we select equation 3.5 to compute an angle between two vectors.



(a) (b)

Figure 3.4: Finding angle between (a) two 2D vectors; (b) two 3D vectors.

3.4.1 An example of finding angle between two 2D vectors

Suppose we have two 2D vectors $(3,7)$ and $(-1,4)$ (see figure 3.4a). So the angle between these vectors is,

$$\theta = \tan^{-1}\left(\frac{\|\overrightarrow{(3,7)}\times\overrightarrow{(-1,4)}\|}{\overrightarrow{(3,7)}\cdot\overrightarrow{(-1,4)}}\right)$$

Now, $\|\overrightarrow{(3,7)} \times \overrightarrow{(-1,4)}\| = (3\times 4) - ((-1)\times 7) = 19$

So, $\theta = \tan^{-1}\left(\frac{19}{(3\times(-1))+(4\times 7)}\right)$

$= \tan^{-1}\left(\frac{19}{-3+28)}\right)$

$= \tan^{-1}\left(\frac{19}{25}\right)$

$= 0.6499 radian$ or $37.2348 degree$

3.4.2 An example of finding angle between two 3D vectors

Suppose we have two 3D vectors $(4,0,7)$ and $(-2,1,3)$ (see figure 3.4b). The angle between these vector is,

$$\theta = \tan^{-1}\left(\frac{\|\overrightarrow{(4,0,7)}\times\overrightarrow{(-2,1,3)}\|}{\overrightarrow{(4,0,7)}\cdot\overrightarrow{(-2,1,3)}}\right)$$

$= \tan^{-1}\left(\frac{\|\overrightarrow{(4,0,7)}\times\overrightarrow{(-2,1,3)}\|}{\overrightarrow{(4,0,7)}\cdot\overrightarrow{(-2,1,3)}}\right)$

Now, $\|\overrightarrow{(4,0,7)} \times \overrightarrow{(-2,1,3)}\| = \left\| \begin{matrix} \hat{x} & \hat{y} & \hat{z} \\ 4 & 0 & 7 \\ -2 & 1 & 3 \end{matrix} \right\| = 27.2213$

So, $\theta = \tan^{-1}\left(\frac{27.2213}{4\times(-2)+0\times1+7\times3}\right)$

$= \tan^{-1}\left(\frac{27.2213}{-8+0+21}\right)$

$= \tan^{-1}\left(\frac{27.2213}{13}\right)$

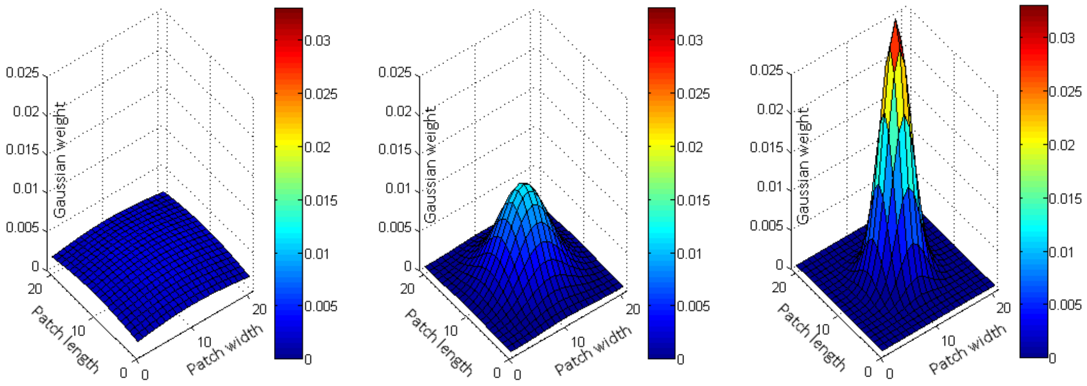
$= 1.1253 radian$ or, $64.4724 degree$

3.4.3 Determining the filtering parameter h decay of weight in Gaussian kernel

The Filtering Parameter σ in equation 5.3 defines the decay of weights in the Gaussian Kernel. It is dependent of the size of the patch. The two objectives of using Gaussian weighted kernel are:

1. put more weights at the center of each patch
2. incorporate the information around the boundary.

The decay of weights should be distributed in a way so that it fulfills both of the objectives. We typically pick $\frac{\texttt{patch radius}+1}{3}$ for h. Figure 3.5 illustrates the reason behind that choice. Here, the patch size is equal to $(\,(2 \times patch\ radius) + 1\,)^2$. If the h has a large value compared to the patch radius, then the weights in the Gaussian kernel tend be equally distributed (see Figure 3.5a). This goes against the objective of using a Gaussian kernel in the first place since we want to put more weights at the centre of the kernel. On the other hand, if h has a small value, weights at the edges become almost zero. Thus the information at the edges of the patch will remain unused (see Figure 3.5c). By choosing $h = \frac{\texttt{patch radius}+1}{3}$, we generate a Gaussian kernel that fulfills both objectives of (a) putting higher weights at the centre and (b) considering all the information available in the patch (see Figure 3.5b).



(a) (b) (c)

Figure 3.5: Relationship between standard deviation and patch radius; `patch radius` = 10; color bars on the right size show the correspondence of color with values in the kernel; (a) $h = \texttt{patch radius} + 1$; (b) $h = \frac{\texttt{patch radius}+1}{3}$; (c) $h = \frac{\texttt{patch radius}+1}{5}$.

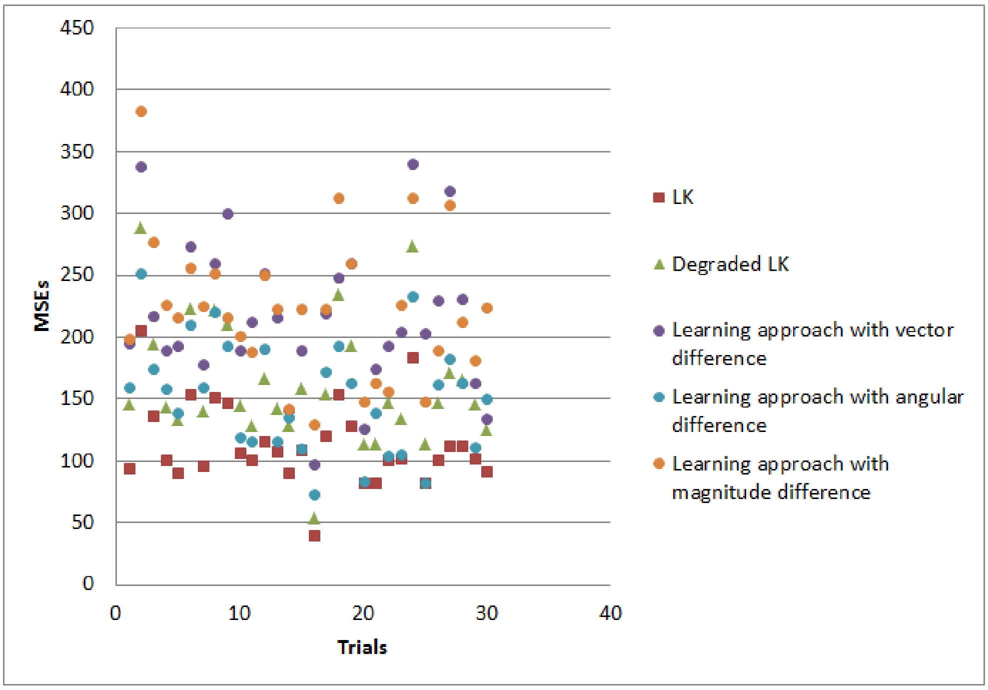


Figure 3.6: MSEs of images registered with LK, degraded LK and learning algorithm using different similarity measures (experiment described in section 3.4.4).

Table 3.1: Descriptive Statistics (experiment described in section 3.4.4)

Method	Dependent Variable	Mean	Std. Deviation	N
1	LK	113.47	33.10	30
2	Degraded LK	161.80	49.92	30
3	Vector Difference	216.34	59.36	30
4	Angular difference	152.43	45.42	30
5	Magnitude difference	222.38	57.45	30

3.4.4 Results with the 2-D Dynamic MRI Dataset

We first compared the performances of these three similarity measures using our 2-D dMRI dataset. The source images are registered to corresponding target images with both the LK optical flow algorithm and a degraded version of LK. LK is degraded by restricting the maximum downsampling factor, i.e. the number of pyramidal levels, to 2, such that it can handle smaller displacements but struggles with large displacements. This is to simulate a challenging large-displacement scenarios. Our objective is to improve the estimated motion using the proposed learning algorithm.

Table B.1 shows the results. Column 1 shows the trial numbers (detail of trials can be found in Table 2.1 and Figure 3.6 shows MSEs of images registered with LK, degraded LK and learning algorithm using different similarity measures). The MSEs of LK and degraded LK are given in column 2 and 3 respectively; they are computed on the basis of the squared differences of intensities between target and registered images. The MSEs of our learning approach with each of the three similarity measures are given in columns 4, 5 and 6. We used the following parameters:

- patch radius: $r = 5$,
- parameter for decay of weight: $h = 2$ (as described in section 3.4.3, $h = (r + 1)/3$), and
- size of the neighborhood, $\aleph$ is the whole image.

Descriptive statistics of this experiment is given in Table 3.1. Figure 3.7 shows the Box and Whisker plot of MSEs of LK, degraded LK and different similarity measures. We performed a repeated measures ANOVA to compare the performances

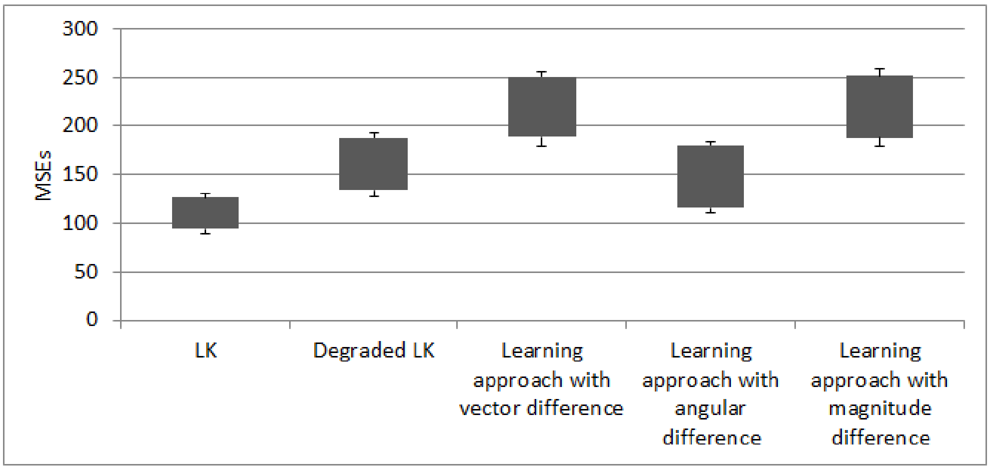


Figure 3.7: The Box and Whisker plot of MSEs of LK, degraded LK and different similarity measures (experiment described in section 3.4.4).

Table 3.2: Mauchly's Test of Sphericity (experiment described in section 3.4.4)

Measure:MSE

Within Subjects Effect	Mauchly's W	Approx. Chi-Square	df	Sig.	Epsilon		
					Greenhouse-Geisser	Huynh-Feldt	Lower-bound
Method	.45	21.93	9	$9.21 \times 10^{-3**}$	.78	.89	.25

Sig. (* for $<.05$ and ** for $<.01$)

of the different approaches (rationale for using repeated measures ANOVA is discussed in section 2.4.3):

- Method 1: LK
- Method 2: Degraded LK
- Method 3: Vector Difference
- Method 4: Angular difference
- Method 5: Magnitude difference

Mauchly's test (see Table 3.2) indicates that the assumption of sphericity has been violated ($\chi^2\,(9) = 21.93$, $p < 0.05$) therefore Greenhouse-Geisser corrected tests are reported ($\varepsilon = .78$). The results show that the MSEs are significantly affected by the choice of method ($V = 0.93$, $F\,(4, 26) = 90.78$, $p < 0.05$, see Table 3.3).

Table 3.3: Multivariate Tests (experiment described in section 3.4.4)

Effect		Value	F	Hypothesis df	Error df	Sig.
Method	Pillai's Trace	.93	90.78	4	26	6.95×10^{-15}**
	Wilks' Lambda	.07	90.78	4	26	6.95×10^{-15}**
	Hotelling's Trace	13.97	90.78	4	26	6.95×10^{-15}**
	Roy's Largest Root	13.97	90.78	4	26	6.95×10^{-15}**

Sig. (* for $<.05$ and ** for $<.01$)

Table 3.4: Tests of Within-Subjects Effects (experiment described in section 3.4.4)

Measure:MSE

Source		Type III Sum of Squares	df	Mean Square	F	Sig.
Method	Sphericity Assumed	252258.74	4	63064.69	133.51	1.88×10^{-42}**
	Greenhouse-Geisser	252258.74	3.12	80834.06	133.51	1.17×10^{-33}**
	Huynh-Feldt	252258.74	3.54	71253.12	133.51	7.41×10^{-38}**
	Lower-bound	252258.74	1.00	252258.74	133.51	2.26×10^{-12}**
Error (Method)	Sphericity Assumed	54792.32	116	472.35		
	Greenhouse-Geisser	54792.32	90.50	605.44		
	Huynh-Feldt	54792.32	102.67	533.68		
	Lower-bound	54792.32	29.00	1889.39		

Sig. (* for $<.05$ and ** for $<.01$)

Table 3.5: Pairwise Comparisons (experiment described in section 3.4.4)

Method	1	2	3	4	5
1		5.13×10^{-15}**	7.83×10^{-16}**	1.01×10^{-9}**	5.81×10^{-17}**
2	5.13×10^{-15}**		7.02×10^{-11}**	4.40×10^{-2}*	4.93×10^{-12}**
3	7.83×10^{-16}**	7.02×10^{-11}**		9.15×10^{-11}**	.381
4	1.01×10^{-9}**	4.40×10^{-2}*	9.15×10^{-11}**		2.25×10^{-12}**
5	5.81×10^{-17}**	4.93×10^{-12}**	.381	2.25×10^{-12}**	

Sig. (* for <.05 and ** for <.01)

Table 3.6: Descriptive Statistics (experiment described in section 3.4.5)

Method	Dependent Variable	Mean	Std. Deviation	N
1	LK	198.34	689.61	400
2	Degraded LK	303.54	1054.97	400
3	Vector difference	167.66	561.86	400
4	Angular difference	152.58	475.21	400
5	Magnitude difference	165.41	533.62	400

From the post hoc test we can conclude that (see Table 3.4 and Table 3.5):

- The learning Algorithm using *angular difference* as a similarity measure generates significantly lower MSEs than both LK ($p < .05$) and degraded LK ($p < .05$).
- The use of *angular difference* as a similarity measure generates significantly lower MSEs than that of *vector difference* or *magnitude difference* ($p < .05$).

3.4.5 Results with the 3-D Gated CT Dataset

We repeated the experiment with the 3-D Gated dataset.

As above, we registered the source images to the corresponding target images with both LK, degraded LK and our learning approach, with the same parameters. Results are reported in the table in Appendix C.1. Column 1 shows the trial numbers; detail

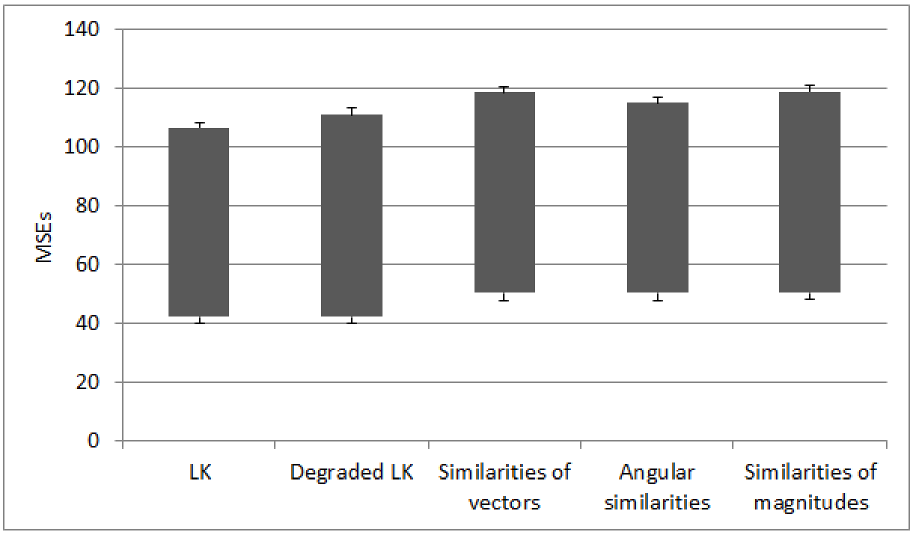


Figure 3.8: The Box and Whisker plot of MSEs of LK, degraded LK and different similarity measures (experiment described in section 3.4.5).

of trials can be found in the table in Appendix A.1. The MSEs of LK and degraded LK are given in column 2 and 3 respectively. The MSEs of our learning approach with each of the three similarity measures are given in columns 4, 5 and 6. We used the following parameters:

Descriptive statistics of this experiment is given in Table 3.6. Figure 3.8 shows the Box and Whisker plot of MSEs of LK, degraded LK and different similarity measures. Here as well, Mauchly's test indicates that the assumption of sphericity has been violated ($\chi^2(9) = 3930.823$, $p < 0.01$) therefore Greenhouse-Geisser corrected tests are reported ($\varepsilon = .282$). The results show that MSEs are significantly affected by the choice of method ($V = 0.06$, $F(4, 396) = 6.734$, $p < 0.01$.

From the post hoc test we can conclude that (see Table 3.7 and Table 3.8):

- The learning Algorithm using *angular difference* as a similarity measure generates significantly lower MSEs than both LK ($p < .01$) and degraded LK ($p < .01$).
- The use of *angular difference* as a similarity measure generates significantly lower MSEs than that of *vector difference* or *magnitude difference* ($p < .01$).

3.4.6 Discussion

These results suggest that the *angular difference* is the best performing measure. This is not surprising given that a substantial number of vectors in the training patches have larger magnitudes than those in the test patches since the former are from de-

Table 3.7: Tests of Within-Subjects Effects (experiment described in section 3.4.5)

Measure:MSE

Source		Type III Sum of Squares	df	Mean Square	F	Sig.
Method	Sphericity Assumed	1.88E+06	1	1.88E+06	16.32	6.43E-05**
	Greenhouse-Geisser	1.17E+05	1	1.17E+05	10.54	1.27E-03**
	Huynh-Feldt	2.89E+06	1	2.89E+06	19.99	1.01E-05**
	Lower-bound	1.18E+06	1	1.18E+06	15.38	1.03E-04**
Error (Method)	Sphericity Assumed	4.60E+07	399	1.15E+05		
	Greenhouse-Geisser	4.42E+06	399	1.11E+04		
	Huynh-Feldt	5.78E+07	399	1.45E+05		
	Lower-bound	3.07E+07	399	7.68E+04		

Sig. (* for $<.05$ and ** for $<.01$)

formation fields estimated between consecutive images whereas the latter are from deformation fields estimated between images further apart (since we are specifically dealing with large displacement scenarios). When we use *vector difference* or *magnitude difference* as similarity measures, those vectors with larger magnitudes yield lower similarity with the test patch and contribute less to the weighted sum in equation 3.2. When we use *angular difference* however, the learning algorithm puts larger weight to a training patch where vectors have similar directions to the test vectors, irrespective of whether or not the training vectors may have larger (or indeed different) magnitudes.

Figure 3.8 may appear to demonstrate that the three similarity measures perform similarly. But, the descriptive statistics (Table 3.6) and the pairwise comparisons (Table 3.8) show that even though the three approaches yield similar results, they are indeed statistically significantly different.

Figure 3.9 shows a synthetic example that illustrates why *angular difference* yields better performance. We have two training patches of vectors (a and b) and a patch of vectors as a test case (c). The test patch has one vector at its centre which is different from the others (i.e. probably incorrect):

- Training patch 1 (Figure 3.9a) consists of vectors with larger magnitudes and similar angular values compared to those of the test patch. The vector at its centre is $(0.5, 0.5)$.

Table 3.8: Pairwise Comparisons (experiment described in section 3.4.5)

Measure: MSE

	1	2	3	4	5
1		3.00E-05**	2.24E-03**	8.50E-04**	2.72E-03**
2	3.00E-05**		2.99E-05**	1.43E-05**	2.76E-05**
3	2.24E-03**	2.99E-05**		5.95E-03**	2.99E-01
4	8.50E-04**	1.43E-05**	5.95E-03**		4.32E-04**
5	2.72E-03**	2.76E-05**	2.99E-01	4.32E-04**	

Sig. (* for $<.05$ and ** for $<.01$)

- Training patch 2 (Figure 3.9b) has vectors that have different angular values from those in the test case. The vector at its centre is $(-0.5, 0.5)$.

We applied our learning algorithm to the vector at the centre of the test patch using the three different similarity measures. Results are given in Figure 3.9 d, e, and f:

- The learned vector using *vector difference* (Figure 3.9d) is $(0.0935, 0.5)$ and the learned vector using *magnitude difference* (Figure 3.9e) is $(0, 0.5)$. Both similarity measures failed to produce a sufficiently large displacement vector.
- The learned vector using algorithm with *angular difference* (Figure 3.9f) is $(0.5, 0.5)$ i.e. the correct magnitude and a similar direction to the vectors in neighbourhood in the test patch.

3.5 Vector Composition of Pairwise Deformation Fields

We saw in section 2.1.3 that standard methods may not be able to accurately estimate large displacements. However, in the context where a series of consecutive images is available, we may accurately estimate the larger displacements which exist between images that are further apart in the series by performing vector composition of the smaller pairwise displacements. Figure 3.10 illustrates this approach. In the first row of Figure 3.10, we see a large motion split in four smaller motions. The actual motion can be constructed by computing the vector composition of these discrete vectors.

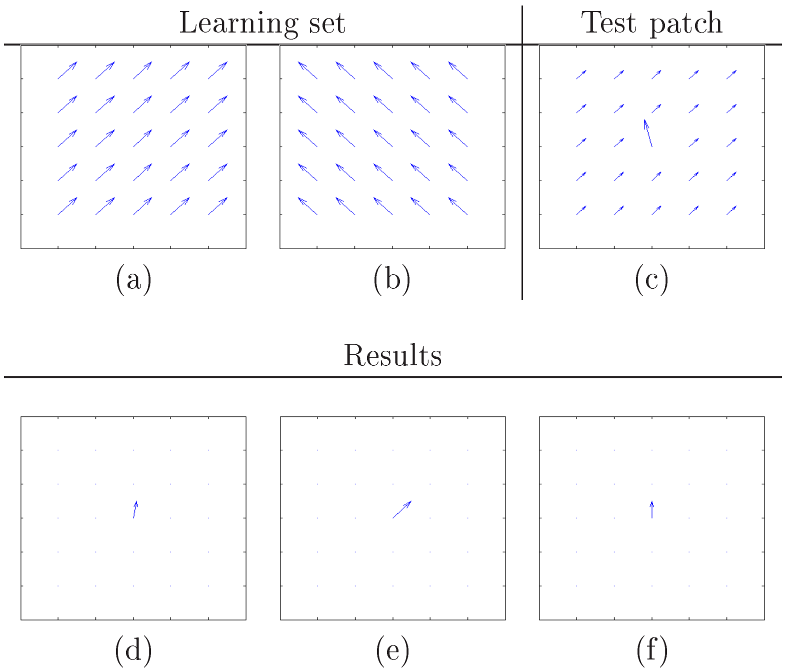


Figure 3.9: A synthetic example showcasing the use of angular difference as a similarity measure; (a) training patch 1; (b) training patch 2; (c) test patch; (d) result with *vector difference*; (e) result with *angular difference*; (f) result with *magnitude difference*

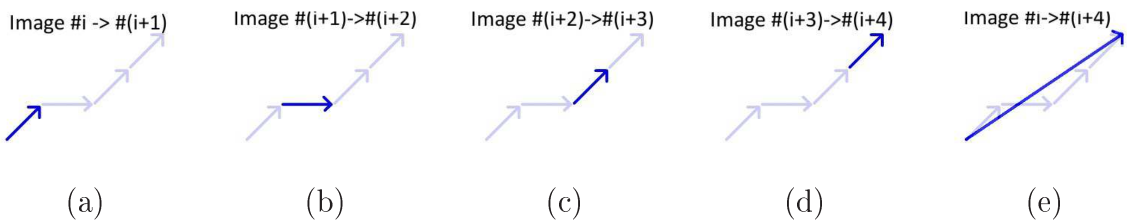


Figure 3.10: An example of a large vector split in elements in a learning set : (a), (b), (c) and (d) are four parts of the large vector; (e) the actual displacement.

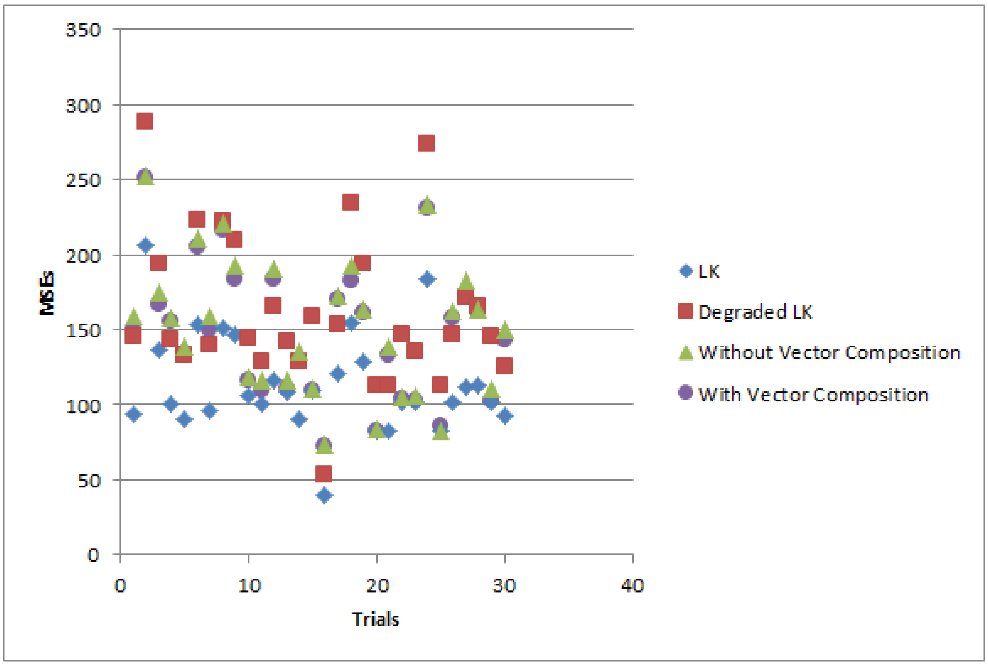


Figure 3.11: MSEs of images registered using LK, degraded LK, learning algorithm with and without vector composition.

In a similar fashion, we can add to the learning set the composed deformation fields obtained by composing consecutive deformation fields, so as to better model large displacements. We use a multi-threaded function for this composition task. In each thread, we apply a bottom-up memoization technique. Memoization (Michie, 1968) is an optimization technique used primarily to speed up programs by having function calls that avoid repeating the calculation of results for previously processed inputs. A memoized function remembers results of some inputs and returns the remembered result rather than recalculating it. If we have n sequential displacement fields in our learning set, then we can divide our problem into n independent sub-problems where each element from the learning set performs vector composition disjointly. Consequently, multiple threads that compose vectors can be run independently in parallel starting sequentially from each element in the learning set ending at the last element.

In the end, we get $\frac{n\times(n+1)}{2}$ displacement fields consisting all possible vector compositions.

Table 3.9: Descriptive Statistics

Method	Dependent Variable	Mean	Std. Deviation	N
1	LK	113.47	33.10	30
2	Degraded LK	161.80	49.92	30
3	Without vector composition	152.43	45.42	30
4	With vector composition	147.89	44.57	30

3.5.1 Results with the 2-D Dynamic MRI Dataset

We compared the proposed method using learning sets with and without composed displacement fields. As before, we registered the sources image to the corresponding target images with LK, degraded LK and our learning approach, using the same parameters. The results are in Table D.1. Column 1 shows the trial numbers; detail of trials can be found in Table 2.1. The MSEs of LK and degraded LK are given in column 2 and 3 respectively, and the MSEs of the learning approach without and with vector composition in columns 5 and 6. The Figure 3.11 shows the MSEs of images registered using LK, degraded LK, learning algorithm with and without vector composition. We used the following parameters:

- patch radius: $r = 5$,
- parameter for decay of weight: $h = 2$ (as described in section 3.4.3, $h = (r + 1)/3$),
- size of the neighborhood, $\aleph$ is the whole image, and
- the *angular difference* is used as a similarity measure

We performed a repeated measures ANOVA to compare the performances of the various approaches (rationale for using repeated measures ANOVA is discussed in section 2.4.3):

- Method 1: LK
- Method 2: Degraded LK
- Method 3: learning approach without vector composition
- Method 4: learning approach with vector composition

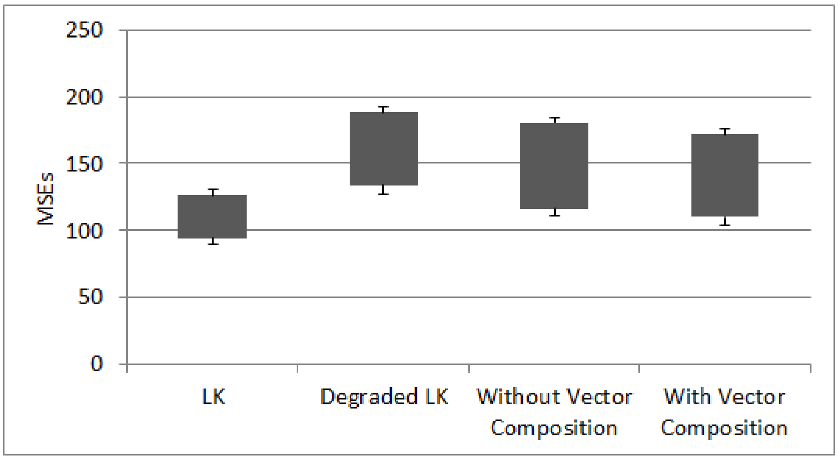


Figure 3.12: The Box and Whisker plot of MSEs of LK, degraded LK, learning algorithms with and without vector composition (experiment described in section 3.5.1).

Descriptive statistics of this experiment is given in Table 3.9. Figure 3.12 shows the Box and Whisker plot of MSEs of LK, degraded LK, learning algorithms with and without vector composition. Mauchly's test indicates that the assumption of sphericity has been violated ($\chi^2\,(5) = 89.13$, $p < 0.05$) therefore Greenhouse-Geisser corrected tests are reported ($\varepsilon = .543$). The results show that MSEs are significantly affected by the choice of method ($V = 0.894$, $F\,(3, 27) = 75.91$, $p < 0.05$.

From the post hoc test we can conclude that (see Table 3.10 and Table 3.11):

- The learning algorithm with vector composition generates significantly lower MSEs than both LK ($p < .05$) and degraded LK ($p < .05$).
- The learning algorithm with vector composition generates significantly lower MSEs than the learning algorithm without vector composition ($p < .05$).

3.5.2 Results with the 3-D Gated CT Dataset

We repeated the experiment with the 3-D Gated dataset.

The table in Appendix E.1 shows the results of the same experiment with the 3D gated CT dataset. Column 1 shows the trial numbers (detail of trials can be

Table 3.10: Tests of Within-Subjects Effects (experiment described in section 3.5.1)

Measure:MSE

Source		Type III Sum of Squares	df	Mean Square	F	Sig. (* for <.05 and ** for <.01)
Method	Sphericity Assumed	40059.14	3	13353.05	62.03	1.53×10^{-21}**
	Greenhouse-Geisser	40059.14	1.63	24589.01	62.03	9.51×10^{-13}**
	Huynh-Feldt	40059.14	1.71	23391.27	62.03	2.76×10^{-13}**
	Lower-bound	40059.14	1.00	40059.14	62.03	1.10×10^{-8}**
Error (Method)	Sphericity Assumed	18728.88	87	215.27		
	Greenhouse-Geisser	18728.88	47.25	396.42		
	Huynh-Feldt	18728.88	49.66	377.11		
	Lower-bound	18728.88	29.00	645.82		

Sig. (* for <.05 and ** for <.01)

Table 3.11: Pairwise Comparisons (experiment described in section 3.5.1)

Method	1	2	3	4
1		5.13×10^{-15}**	1.01×10^{-09}**	4.11×10^{-9}**
2	5.13×10^{-15}**		4.40×10^{-2}*	2.95×10^{-3}**
3	1.01×10^{-09}**	4.40×10^{-2}*		7.82×10^{-08}**
4	4.11×10^{-9}**	2.95×10^{-3}**	7.82×10^{-08}**	

Sig. (* for <.05 and ** for <.01)

Table 3.12: Descriptive Statistics (experiment described in section 3.5.2)

Method	Dependent Variable	Mean	Std. Deviation	N
1	LK	198.34	689.61	400
2	Degraded LK	303.54	1054.97	400
3	Without vector composition	152.58	475.21	400
4	With vector composition	150.47	475.07	400

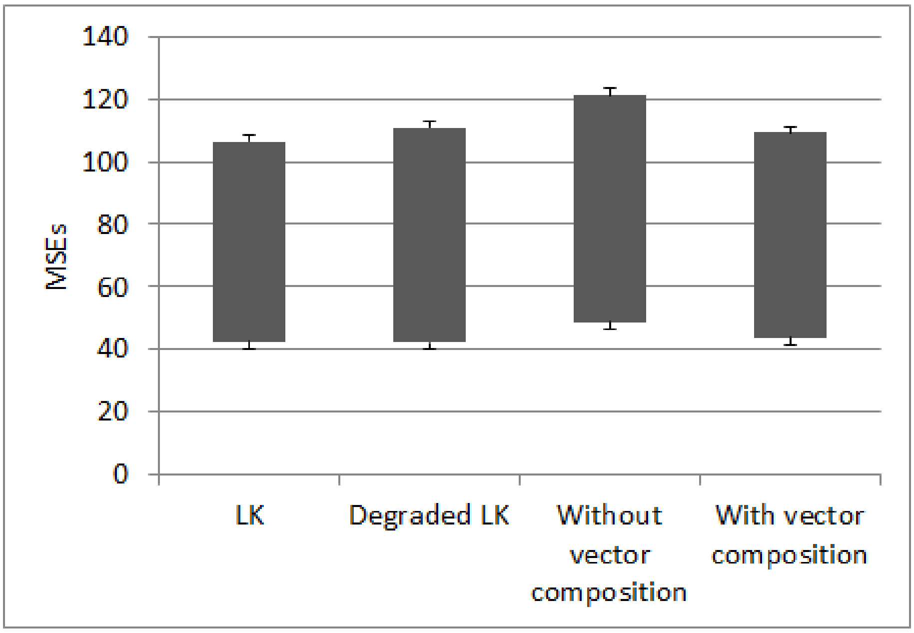


Figure 3.13: The Box and Whisker plot of MSEs of LK, degraded LK, learning algorithms with and without vector composition (experiment described in section 3.5.2).

found in the table in Appendix A.1). The MSEs of LK and degraded LK are given in column 2 and 3 respectively. We compare learning algorithm without and with vector composition in the learning set; their MSEs are given in columns 5 and 6.

Descriptive statistics of this experiment is given in Table 3.12. Figure 3.13 shows the Box and Whisker plot of MSEs of LK, degraded LK, learning algorithms with and without vector composition. Here also, Mauchly's test indicates that the assumption of sphericity has been violated ($\chi^2(5) = 5062.204$, $p < 0.01$) therefore Greenhouse-Geisser corrected tests are reported ($\varepsilon = .376$). The results show that MSEs are significantly affected by the choice of method ($V = 0.78$, $F(3, 397) = 456.09$, $p < 0.01$.

Table 3.13: Tests of Within-Subjects Effects (experiment described in section 3.5.2)

Measure:MSE

Source		Type III Sum of Squares	df	Mean Square	F	Sig.
Method	Sphericity Assumed	6.17E+06	3	2.06E+06	18.39	1.14E-11**
	Greenhouse-Geisser	6.17E+06	1.13	5.47E+06	18.39	8.83E-06**
	Huynh-Feldt	6.17E+06	1.13	5.46E+06	18.39	8.76E-06**
	Lower-bound	6.17E+06	1.00	6.17E+06	18.39	2.26E-05**
Error (Method)	Sphericity Assumed	1.34E+08	1197	1.12E+05		
	Greenhouse-Geisser	1.34E+08	450.21	2.97E+05		
	Huynh-Feldt	1.34E+08	450.61	2.97E+05		
	Lower-bound	1.34E+08	399.00	3.35E+05		

Sig. (* for <.05 and ** for <.01)

Table 3.14: Pairwise Comparisons (experiment described in section 3.5.2)

Measure: MSE

	1	2	3	4
1		3.00E-05**	8.50E-04**	4.90E-04**
2	3.00E-05**		1.43E-05**	1.10E-05**
3	8.50E-04**	1.43E-05**		1.29E-130**
4	4.90E-04**	1.10E-05**	1.29E-130**	

Sig. (* for <.05 and ** for <.01)

From post hoc tests we can conclude that (see Table 3.13 and Table 3.14):

- The learning algorithm with vector composition generates significantly lower MSEs than both LK ($p < .01$) and degraded LK ($p < .01$).
- The learning algorithm with vector composition generates significantly lower MSEs than without composition ($p < .01$).

3.5.3 Discussion

These results suggest that the learning algorithm with vector composition shows better performance than without vector composition. Figure 3.14 illustrates the reason why including vector composition improves the performance to the learning algorithm. Here, we have two vector patches in the learning set (Figure 3.14a and Figure 3.14b).

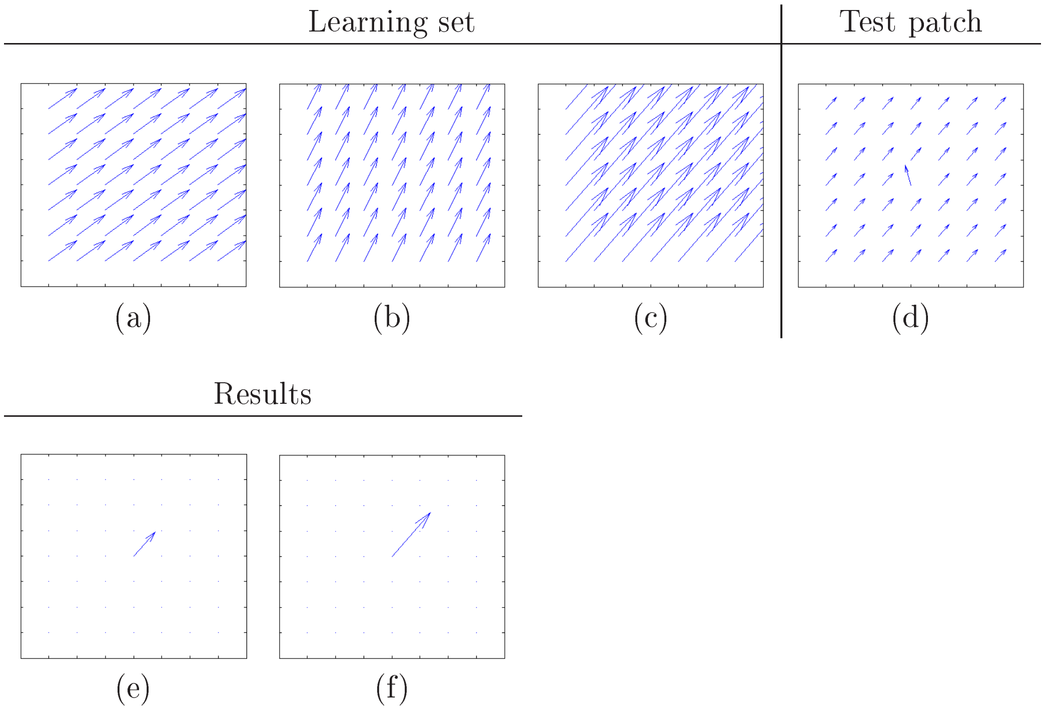


Figure 3.14: A synthetic example explaining the role of vector composition in learning set; (a) training patch 1; (b) training patch 2; (c) vector composition in the training patches; (d) vector patch as a test case; (e) learned vector without vector composition; (f) learned vector with vector composition.

The composed vector field between these two patches is given in Figure 3.14c. The test patch is shown in Figure 3.14d. The learned vector without vector composition is $(0.8, 0.9)$ (see Figure 3.14e) and the learned vector with vector composition is $(1.4, 1.7)$ (see Figure 3.14f). The learned vector with vector composition (Figure 3.14f) is larger than the learned vector without vector composition (Figure 3.14e). Therefore, if we include the vector composition of larger motion from pair-wise deformation fields the learning set, we will be better able to learn from larger motions.

3.6 Conclusion

In this chapter, we introduced our learning approach to the estimation of optical flow. We apply the learning algorithm on MPI-Sintel dataset and compared its performance quantitatively with HS, SMF, LDOF and DeepFlow. The learning algorithm shows better performance than both the HS and the SMF. Results suggest that our learning approach applied to the straight-forward HS shows similar performances than both of the very sophisticated techniques, LDOF and DeepFlow. Results suggest that *angular difference* is the preferred choice of patch similarity measure. From the second experiment we conclude that the addition of composed displacement fields improves the performance of the algorithm. However, adding more fields to the learning set substantially increases its size, which is not without issue, as we will explore in the next chapter.

Chapter 4

Building a Codebook of Patches

4.1 Introduction

In the previous chapter, we discussed how adding composed deformation fields to the learning set can improve performances. However, this substantially increases the size of the learning set. Looking for similarities in a larger set of patches during the filtering phase is not only computationally more expensive but may also result in an over-smoothed filtered flow as the $\mathcal{J}$ in the equation 3.2 becomes larger with larger learning set and thus computes Gaussian weighted average of a larger number of patches. Therefore the filtered flow becomes over-smoothed. Figure 4.1 illustrates this issue. Image 221 from the 2-D dMRI dataset was registered to image 222 using our learning algorithm without using a codebook of patches approach. In this example, the learning set contains the pairwise deformation fields between image 263 and 268 and their vector compositions, i.e. 15 fields (5 pairwise fields and 10 composed fields). Due to the large learning set, over-smoothing occurred in the filtered flow, consequently the registration was not accurate (see red rectangles in Figure 4.1).

4.2 Clustering the Patches

To alleviate the issue of over-smoothing, we structure the learning set, $\mathcal{C}$, into clusters of similar patches. The clustering approach is a way to reduce the number of patches in a dynamic, image-specific fashion: rather than arbitrarily decreasing the overall number of patches a priori, we use clustering and representative patches to select the

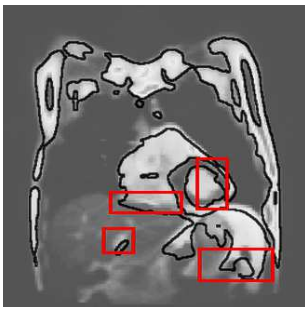


Figure 4.1: Over-smoothing as a result of too large a learning set. The edges of the target image are super-imposed on the registered image; the red rectangles highlight incorrect registration.

most appropriate ones given the images to be registered so that is more powerful. We considered two clustering approaches: K-means and hierarchical clustering.

4.2.1 K-means Clustering

K-means clustering splits observations into k clusters in which each observation belongs to the cluster with the nearest mean, which serves as a prototype (MacQueen, 1967). It therefore requires a distance metric to be specified. K-means also requires that the number of clusters, k, be specified a priori. Deciding upon an optimal k is not trivial, and a number of approaches have been suggested, e.g. the F-test (Fisher, 1922) and the elbow method (Thorndike, 1953). In F-test, the test statistics has an F-distribution under the null hypothesis. It is sensitive to non-normality (Box, 1953; Markowski and Markowski, 1990) . The elbow method computes the percentage of variance for k number of clusters and gradually increases the number of clusters. Initially, the percentage of variance will decrease as k increases. For some value of k, the marginal gain will drop, which should correspond to an "optimal" number of clusters. However, this optimal number of clusters cannot always be unambiguously identified.

4.2.2 Hierarchical Clustering

Hierarchical clustering builds a sequence of partitions in which each partition is nested into the next partition in the sequence (Ward, 1963). Like K-means, it requires the specification of a distance metric for the patches. Additionally, a linkage criterion must also be selected. At the beginning of this process, each element is in a cluster of its own. The clusters are then sequentially combined into larger clusters until all elements end up being in the same cluster. At each step, the two clusters separated by the shortest distance are combined.

In complete-linkage clustering, the link between two clusters contains all element pairs, and the distance between clusters equals the distance between those two elements (one in each cluster) that are farthest away from each other. Whereas, in single-linkage clustering, the link between two clusters is made by a single element pair, namely those two elements (one in each cluster) that are closest to each other. The shortest of these links that remains at any step causes the fusion of the two clusters whose elements are involved.

We evaluated a number of strategies and found that agglomerative clustering, a bottom up approach, with single linkage clustering performed best. This approach first assign each data point to its own singleton group. Then, pairs of clusters are merged as one and move up the hierarchy until all the data are merged into a single cluster. We found that inconsistency coefficient threshold of 1.15 as the value of the cutoff argument perform best for our datasets.

Hierarchical clustering is generally considered to be a better approach, though it is more computationally expensive than K-means (Steinbach et al., 2000). Preliminary experiments confirmed these considerations and we selected it in our experiments.

4.2.3 Manifold Embedding

The choice of distance metric will greatly influence the shape of the clusters, and, in turn, the overall performance of our approach. The metric should be linked to the characteristics of the patches. Consequently, we first extract feature vectors from each patch. We focused on orientation-specific and distribution-specific descriptors since these are the strongest features of the vector fields we are using. We selected the following 7 features for 2D motion: variance and Gaussian weighted average of the x and of the y components of the displacement vectors, variance of the vector polar angles, angular velocity perpendicular to the flow and divergence (volume density of the outward flux). Compared to 2D motion, 3D motion has one more dimension. Therefore, the higher manifold for 3D motion has 9 features: variance and Gaussian weighted average of the x, y and z components of the displacement vectors, variance of the vector polar angles, angular velocity perpendicular to the flow and divergence (volume density of the outward flux).

Clustering patches in a non-linear, high dimensional space is a non trivial task, whose complexity can be greatly alleviated by reducing the dimensionality of the problem using manifold embedding. Many approaches have been proposed in the literature (Pearson, 1901; Tenenbaum et al., 2000; Roweis and Saul, 2000; Belkin and Niyogi, 2003; Zhang and Zha, 2004). However, the non-inhomogeneity and non-convexity introduced by the addition of the composed displacement vectors would call for a robust non-linear approach capable of preserving the local properties of the manifold. For instance, ISOMAP (Tenenbaum et al., 2000) defines the connectivity of each data point in the neighbourhood graph as its nearest k Euclidean neighbours in the high-dimensional space. This step is vulnerable to short-circuit errors if k is too large with respect to the manifold structure. Indeed, even a single short-circuit error can alter many entries in the geodesic distance matrix, which in turn can lead to a drastically different (and incorrect) low-dimensional embedding (Balasubramanian and Schwartz, 2002).

We selected four dimensionality reduction/embedding techniques based on (a) their appropriateness in light of the nature of our dataset and (b) their computational complexity as the number of patches in our learning set is very large. They are:

1. Principal Component Analysis (PCA) (Pearson, 1901) : It is a dimensionality reduction method in which a covariance analysis between factors takes place. The original data is remapped into a new coordinate system based on the variance within the data. PCA applies a mathematical procedure for transforming a number of correlated variables into a smaller number of uncorrelated principal components. The first principal component accounts for as much of the variability in the data as possible, and each succeeding component accounts for as much of the remaining variability as possible.

2. Locally Linear Embedding (LLE) (Roweis and Saul, 2000) : It begins by finding a set of the nearest neighbors of each point. It then computes a set of weights for each point that best describe the point as a linear combination of its neighbors. Finally, it uses an eigenvector-based optimization technique to find the low-dimensional embedding of points, such that each point is still described with the same linear combination of its neighbors.

3. Laplacian Eigenmaps (Belkin and Niyogi, 2003) : It builds a graph from neighborhood information of the data set. Each data point serves as a node on the graph and connectivity between nodes is governed by the proximity of neighboring points. The graph thus generated can be considered as a discrete approx-

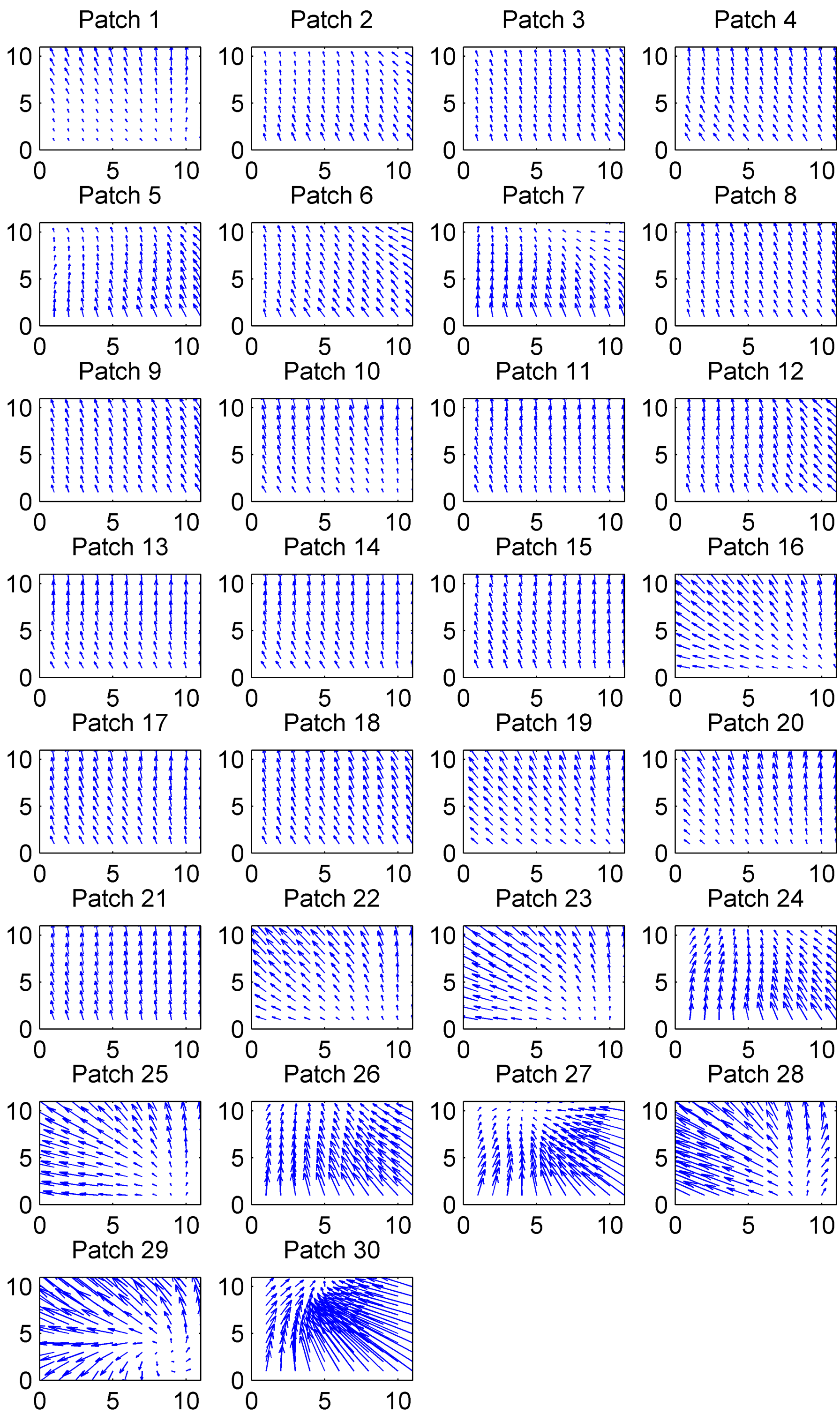
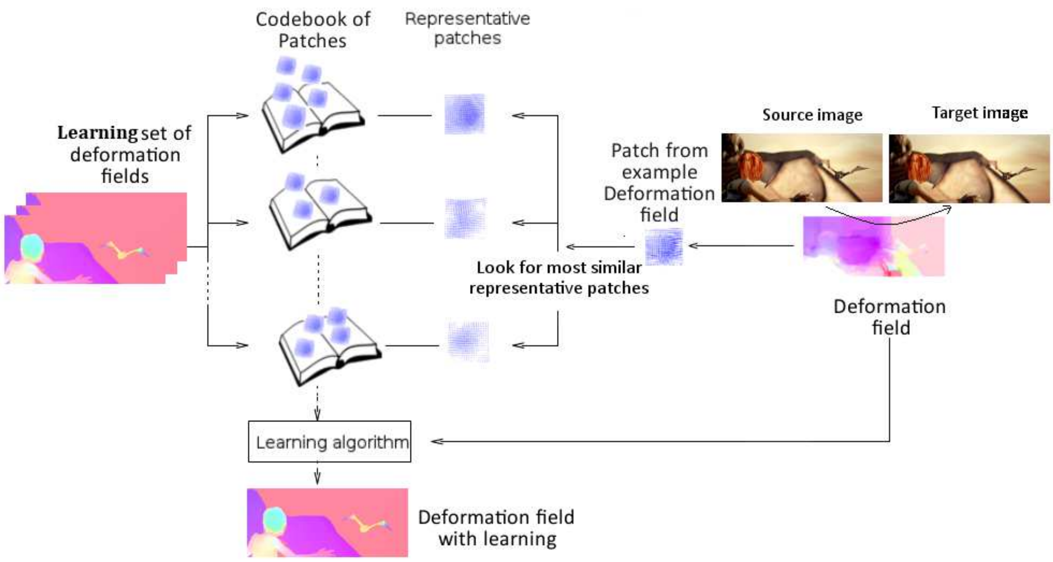
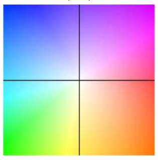


Figure 4.2: From 2D dMRI dataset; 3 flows between frame #41 to #44 are in the learning set. A codebook with patch size 11×11 is created using the learning algorithm. This figure shows 30 representative patches representing 30 clusters of patches. Representative patches are sorted in ascending order of the sums of magnitudes of the vectors in the patches.



(a)



(b)

Figure 4.3: (a) Block diagram of codebook of patches; (b) flow field color coding of deformation fields, the color represents the orientation of the vector and brightness stands for its magnitude.

imation of the low-dimensional manifold in the high-dimensional space. Minimization of a cost function based on the graph ensures that points close to each other on the manifold are mapped close to each other in the low-dimensional space, preserving local distances.

4. Local Tangent Space Alignment (LTSA) (Zhang and Zha, 2004) : it is based on the intuition that when a manifold is correctly unfolded, all of the tangent hyperplanes to the manifold will become aligned. It begins by computing the k-nearest neighbors of every point. It computes the tangent space at every point by computing the d-first principal components in each local neighborhood. It then optimizes to find an embedding that aligns the tangent spaces.

Except for PCA, all other methods are nonlinear.

Once embedded in a two dimensional manifold, the feature vectors are clustered using hierarchical clustering, with the optimal number of clusters determined using the single linkage clustering method. Finally, we compute representative patches for each cluster by averaging all the patches in that cluster.

Figure 4.2 displays the 30 11×11 representative patches computed from 3 pairwise flows estimated using HS around the heart region in 4 consecutive frames (#41 to #44) from the dMRI scans of a healthy volunteer breathing normally in a Siemens 1.5T scanner. Representative patches are sorted in ascending order of the sums of magnitudes of the vectors in the patches. All leaves at or below a node with height less than c are grouped into a cluster. The patches successfully capture both the regular vertical translations due to respiration and the contraction and dilation movements of the heart.

4.3 Learning with a code-book of patches

Recall that our approach consists of a training phase where we build a learning set of intensity and displacement vector patches, $\mathcal{L} = \{\mathbf{I}_l\}_a^b$ from consecutive images in a sequence. For each pair of consecutive images $(\mathbf{I}_l, \mathbf{I}_{l+1})_{l \in [a,b]}$ in the learning set $\mathcal{L}$, we first estimate the optical flow $(\mathbf{u}_l, \mathbf{v}_l)$ between them with a standard optical flow method. We used HS in our experiments. We then compose the flows using bicubic interpolation (for 2D images) or tri-cubic interpolation (for 3D images) to estimates the flows across all pairs of images $(\mathbf{I}_l, \mathbf{I}_{m,m>l})$, consecutive or not. For each pixel in each image $\mathbf{I}_l$, we get a set of patches, which capture the correspondences between the area of $\mathbf{I}_l$ around that pixel and the corresponding areas in all subsequent images $\mathbf{I}_{m,m>l}$.

Given a patch in the initial flow we then restrict ourselves to the patches in those clusters whose representative patches best match the one under consideration, rather than considering all the patches in the learning set. This match is estimated using as a similarity measure the same Gaussian weighted sum of squared differences between the vector polar angles in q_i and in the representative patches $\mathbf{q}_{m,n}$ with standard deviation σ that we use for the filtering: $w_{\texttt{vector}}(\mathbf{q}_i, \mathbf{q}_{m,n}) = \| \theta(\mathbf{q}_i) - \theta(\mathbf{q}_{m,n}) \|_{2,\sigma}^2$. We then discard those clusters for which the similarity falls below a hand tuned, experiment-dependent threshold (see Figure 4.3 for an overview).

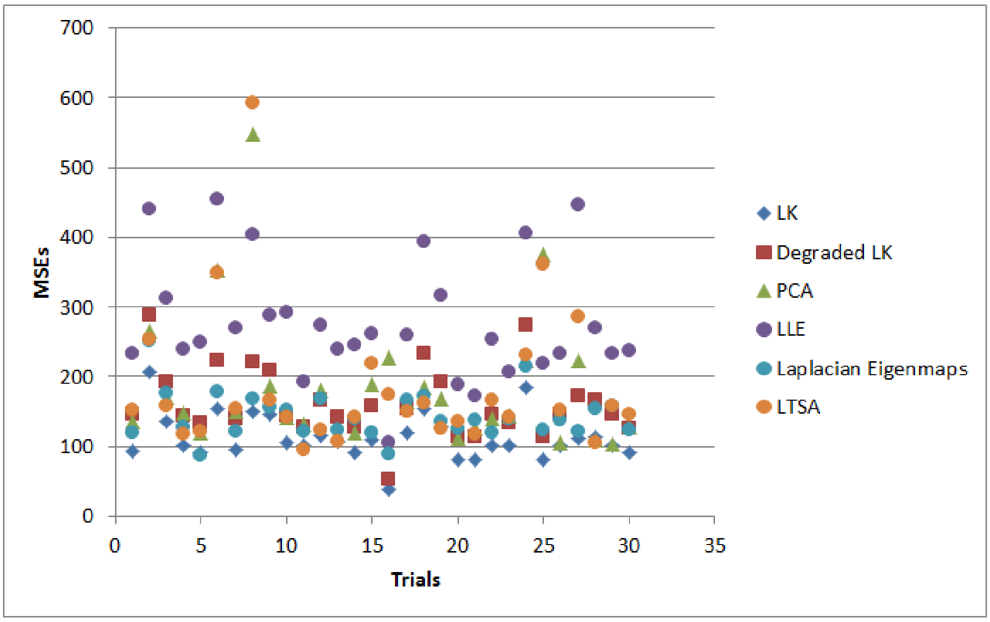


Figure 4.4: MSEs of images registered with LK, degraded LK, learning algorithm with different data embedding methods.

Table 4.1: Descriptive Statistics (experiment described in section 4.3.1)

Method	Dependent Variable	Mean	Std. Deviation	N
1	LK	113.47	33.10	30
2	Degraded LK	161.80	49.92	30
3	PCA	183.17	95.48	30
4	LLE	278.21	85.92	30
5	Laplacian Eigenmaps	144.59	33.74	30
6	LTSA	184.01	101.42	30

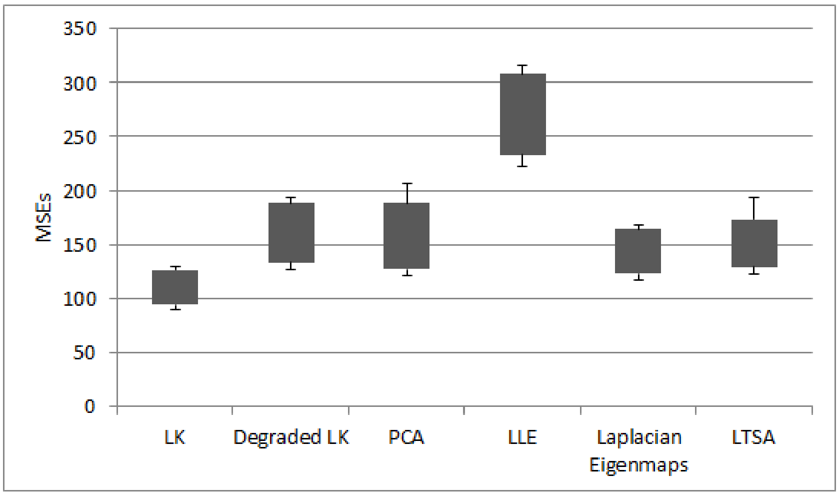


Figure 4.5: The Box and Whisker plot of MSEs of LK, degraded LK, learning algorithms with different data embedding methods (experiment described in section 4.3.1).

4.3.1 Results with the 2-D dMRI Dataset

We applied the learning algorithm with hierarchical clustering and a variety of embedding methods on the 2-D dMRI dataset. The parameters were as follows:

- patch radius: $r = 5$,
- parameter for decay of weight: $h = 2$ (as described in section 3.4.3, $h = (r + 1)/3$),
- the *angular difference* is used as a similarity measure
- all the composed fields were included in learning set,
- inconsistency coefficient threshold of 1.15 as the value of the cutoff argument in hierarchical clustering,
- threshold for the selection of representative patches: maximum 5 degree of angular difference for each vector in a patch.

Results are shown in Table F.1. Column 1 shows the trial numbers; detail of trials can be found in Table 2.1. The MSEs of LK and degraded LK optical flow algorithms are given in column 2 and 3 respectively. The MSEs for different data embeddings are given in columns 4 to 8. The Figure 4.4 shows the MSEs of images registered with LK, degraded LK, learning algorithm with different data embedding methods.

Table 4.2: Tests of Within-Subjects Effects (experiment described in section 4.3.1)
Measure:MSE

Source		Type III Sum of Squares	df	Mean Square	F	Sig.
Method	Sphericity Assumed	469384.89	5	93876.98	35.08	2.13E-23**
	Greenhouse-Geisser	469384.89	1.64	285963.51	35.08	3.49E-09**
	Huynh-Feldt	469384.89	1.73	271825.52	35.08	1.51E-09**
	Lower-bound	469384.89	1.00	469384.89	35.08	1.97E-06**
Error (Method)	Sphericity Assumed	387996.37	145	2675.84		
	Greenhouse-Geisser	387996.37	47.60	8151.01		
	Huynh-Feldt	387996.37	50.08	7748.02		
	Lower-bound	387996.37	29.00	13379.19		

Sig. (* for $<$.05 and ** for $<$.01)

We performed a repeated measures ANOVA to compare the following methods (rationale for using repeated measures ANOVA is discussed in section 2.4.3):

- Method 1: LK
- Method 2: Degraded LK
- Method 3: learning approach with PCA as embedding
- Method 4: learning approach with LLE as embedding
- Method 5: learning approach with Laplacian Eigenmaps as embedding
- Method 6: learning approach with LTSA as embedding

Descriptive statistics of this experiment is given in Table 4.1. Figure 4.5 shows the Box and Whisker plot of MSEs of LK, degraded LK, learning algorithms with different data embedding methods. Mauchly's test indicates that the assumption of sphericity has been violated ($\chi^2\,(14) = 197.11$, $p < 0.05$) therefore Greenhouse-Geisser corrected tests are reported ($\varepsilon = .33$). The results show that MSEs are significantly affected by the choice of method ($V = 0.95$, $F\,(5, 25) = 99.48$, $p < 0.05$.

From the post hoc test we can conclude that (see Table 4.2 and Table 4.3):

- The learning algorithm with Laplacian Eigenmaps embedding generates significantly lower MSEs than both LK ($p < .05$) and degraded LK ($p < .05$).

Table 4.3: Pairwise Comparisons (experiment described in section 4.3.1)

Measure:MSE

	1	2	3	4	5	6
1		5.13E-15**	1.60E-04**	3.58E-15**	1.76E-11**	3.19E-04**
2	5.13E-15**		1.96E-01	7.56E-14**	2.02E-03**	2.08E-01*
3	1.60E-04**	1.96E-01		2.88E-06**	2.69E-02*	8.81E-01
4	3.58E-15**	7.56E-14**	2.88E-06**		1.13E-11**	3.72E-06**
5	1.76E-11**	2.02E-03**	2.69E-02*	1.13E-11**		3.48E-02*
6	3.19E-04**	2.08E-01*	8.81E-01	3.72E-06**	3.48E-02*	

Sig. (* for <.05 and ** for <.01)

Table 4.4: Descriptive Statistics (experiment described in section 4.3.2)

Method	Dependent Variable	Mean	Std. Deviation	N
1	LK	198.34	689.61	400
2	Degraded LK	303.54	1054.97	400
3	PCA	150.50	490.17	400
4	LLE	223.08	739.39	400
5	Laplacian Eigenmaps	147.37	475.26	400
6	LTSA	171.93	566.39	400

- The learning algorithm with Laplacian Eigenmaps generates significantly lower MSEs than the learning algorithm with either PCA ($p < .05$), LLE ($p < .05$) or LTSA ($p < .05$).

4.3.2 Results with the 3-D Gated Dataset

We repeated the experiment with the 3-D Gated dataset.

Descriptive statistics of this experiment is given in Table 4.1. Figure 4.6 shows the Box and Whisker plot of MSEs of LK, degraded LK, learning algorithms with different data embedding methods. Mauchly's test indicates that the assumption of sphericity has been violated ($\chi^2(14) = 5960.31$, $p < 0.01$) therefore Greenhouse-Geisser corrected tests are reported ($\varepsilon = .260$). The results show that MSEs are significantly affected by the choice of methods, $V = 0.09$, $F(5, 395) = 7.71$, $p < 0.01$.

From the post hoc test we can conclude that (see Table 4.5 and Table 4.6):

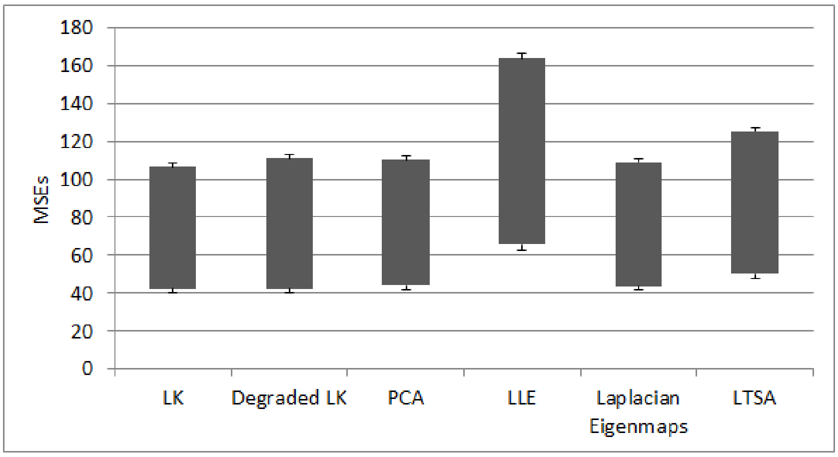


Figure 4.6: The Box and Whisker plot of MSEs of LK, degraded LK, learning algorithms with different data embedding methods (experiment described in section 4.3.2).

Table 4.5: Tests of Within-Subjects Effects (experiment described in section 4.3.2)

Measure:MSE

Source		Type III Sum of Squares	df	Mean Square	F	Sig.
Method	Sphericity Assumed	6.90E+06	5	1.38E+06	17.785	2.78E-17**
	Greenhouse-Geisser	6.90E+06	1.30	5.30E+06	17.785	3.67E-06**
	Huynh-Feldt	6.90E+06	1.30	5.29E+06	17.785	3.60E-06**
	Lower-bound	6.90E+06	1.00	6.90E+06	17.785	3.06E-05**
Error (Method)	Sphericity Assumed	1.55E+08	1995	7.76E+04		
	Greenhouse-Geisser	1.55E+08	519.35	2.98E+05		
	Huynh-Feldt	1.55E+08	520.35	2.98E+05		
	Lower-bound	1.55E+08	399.00	3.88E+05		

Sig. (* for <.05 and ** for <.01)

- The learning algorithm with Laplacian Eigenmaps embedding generates significantly lower MSEs than both LK ($p < .01$) and degraded LK ($p < .01$).
- The learning algorithm with Laplacian Eigenmaps generates significantly lower MSEs than the learning algorithm with PCA ($p < .05$), LLE ($p < .01$) and LTSA ($p < .01$).

Table 4.6: Pairwise Comparisons (experiment described in section 4.3.2)
Measure: MSE

	1	2	3	4	5	6
1		3.00E-05**	2.48E-04**	1.97E-02*	2.05E-04**	1.59E-02*
2	3.00E-05**		8.86E-06**	4.90E-03**	7.30E-06**	4.81E-05**
3	2.48E-04**	8.86E-06**		1.26E-08**	1.09E-02*	6.23E-08**
4	1.97E-02*	4.90E-03**	1.26E-08**		2.54E-08**	7.41E-09**
5	2.05E-04**	7.30E-06**	1.09E-02*	2.54E-08**		6.27E-07**
6	1.59E-02*	4.81E-05**	6.23E-08**	7.41E-09**	6.27E-07**	

Sig. (* for $<$.05 and ** for $<$.01)

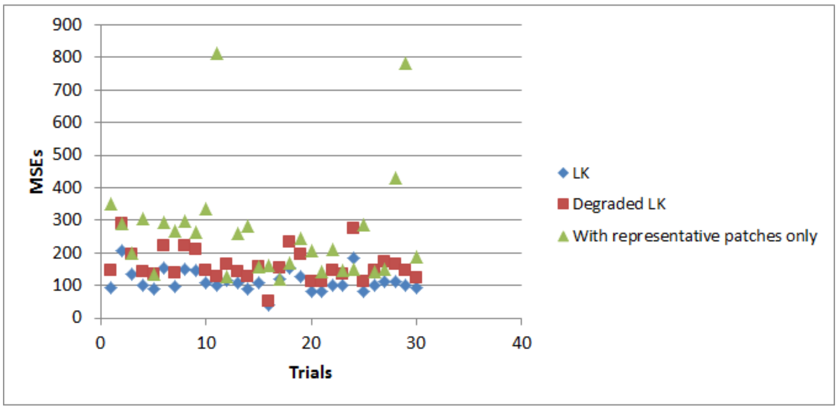


Figure 4.7: MSEs of images registered with learning algorithm with LK, degraded LK, and learning set of representative patches only (experiment described in section 4.3.3.1).

4.3.3 Learning Set of Representative Patches Only

Representative patches are generated by computing the averages of all patches in each cluster. Therefore, using a learning set with *representative patches only* rather than all the patches in the learning set would make for a much faster approach.

Table 4.7: Descriptive Statistics (experiment described in section 4.3.3.1)

Method	Dependent Variable	Mean	Std. Deviation	N
1	LK	113.47	33.10	30
2	Degraded LK	161.80	49.92	30
3	With only RP	263.78	164.65	30

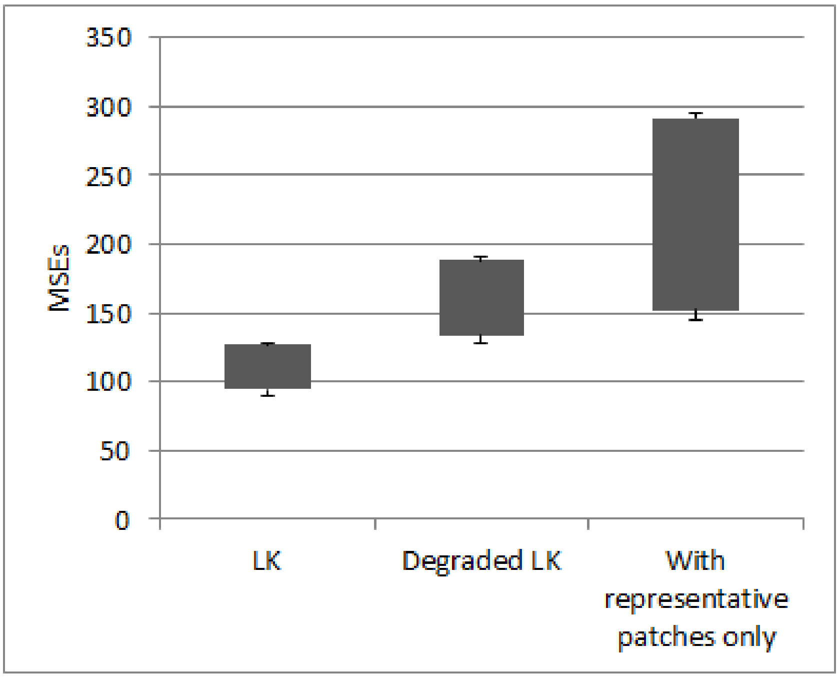


Figure 4.8: The Box and Whisker plot of MSEs of LK, degraded LK and the learning algorithms with a learning set of representative patches only ; rather than improving performance, the inaccuracy increased highly (experiment described in section 4.3.3.1).

4.3.3.1 Results with the 2-D dMRI Dataset

Table H.1 shows the results. Column 1 shows the trial numbers; detail of trials can be found in Table 2.1. The MSEs of LK and degraded LK optical flow algorithms are given in column 2 and 3 respectively. MSEs of learning algorithm with learning set of *representative patches only* are given in column 4. We used the learning algorithm with code-book approach. The Figure 4.7 shows the MSEs of images registered with LK, degraded LK and learning algorithm with a learning set of representative patches only. We used the following parameters:

- patch radius: $r = 5$,
- parameter for decay of weight: $h = 2$ (as described in section 3.4.3, $h = (r + 1)/3$),
- the *angular difference* is used as a similarity measure
- all the composed fields were included in learning set,
- code-book of patches: 2-D Laplacian eigenmaps with a neighborhood of size 6 for the embedding of the feature vector and hierarchical clustering

- inconsistency coefficient threshold of 1.15 as the value of the cutoff argument in hierarchical clustering,
- threshold for the selection of representative patches: maximum 5 degree of angular difference for each vector in a patch.

We performed a repeated measures ANOVA to compare the different methods (rationale for using repeated measures ANOVA is discussed in section 2.4.3):

- Method 1: LK
- Method 2: Degraded LK
- Method 3: With only RP

Descriptive statistics of this experiment is given in Table 4.7. Figure 4.8 shows the Box and Whisker plot of MSEs of LK, degraded LK and the learning algorithms with a learning set of representative patches only . Mauchly's test indicates that the assumption of sphericity has been violated ($\chi^2(2) = 99.39$, $p < 0.05$) therefore Greenhouse-Geisser corrected tests are reported ($\varepsilon = .51$). The results show that MSEs are significantly affected by learning set with representative patches, $V = 0.91$, $F(2, 28) = 147.55$, $p < 0.05$.

From the post hoc test (see Table 4.8 and Table 4.9) we can conclude that the learning algorithm with learning set of *representative patches only* generates significantly higher MSEs than both LK ($p < .05$) and degraded LK ($p < .05$). In fact, the results are so bad that we did not compare it with the learning algorithm.

4.3.3.2 Results with 3-D Gated CT Dataset

We repeated the experiment with the 3-D gated CT dataset. Results are in the table in Appendix I.1 shows the results.

Table 4.8: Tests of Within-Subjects Effects (experiment described in section 4.3.3.1)

Measure:MSE

Source		Type III Sum of Squares	df	Mean Square	F	Sig.
Method	Sphericity Assumed	3.53E+05	2	1.77E+05	17.81	9.34E-07**
	Greenhouse-Geisser	3.53E+05	1.01	3.48E+05	17.81	2.03E-04**
	Huynh-Feldt	3.53E+05	1.02	3.48E+05	17.81	2.01E-04**
	Lower-bound	3.53E+05	1.00	3.53E+05	17.81	2.19E-04**
Error (Method)	Sphericity Assumed	5.75E+05	58	9.92E+03		
	Greenhouse-Geisser	5.75E+05	29.42	1.96E+04		
	Huynh-Feldt	5.75E+05	29.47	1.95E+04		
	Lower-bound	5.75E+05	29.00	1.98E+04		

Sig. (* for <.05 and ** for <.01)

Table 4.9: Pairwise Comparisons (experiment described in section 4.3.3.1)

Measure:MSE

	1	2	3
1		5.13E-15**	3.65E-05**
2	5.13E-15**		3.38E-03**
3	3.65E-05**	3.38E-03**	

Sig. (* for <.05 and ** for <.01)

Table 4.10: Descriptive Statistics (experiment described in section 4.3.3.2)

Method	Dependent Variable	Mean	Std. Deviation	N
1	LK	198.34	689.61	400
2	Degraded LK	303.54	1054.97	400
3	With only RP	357.29	1008.51	400

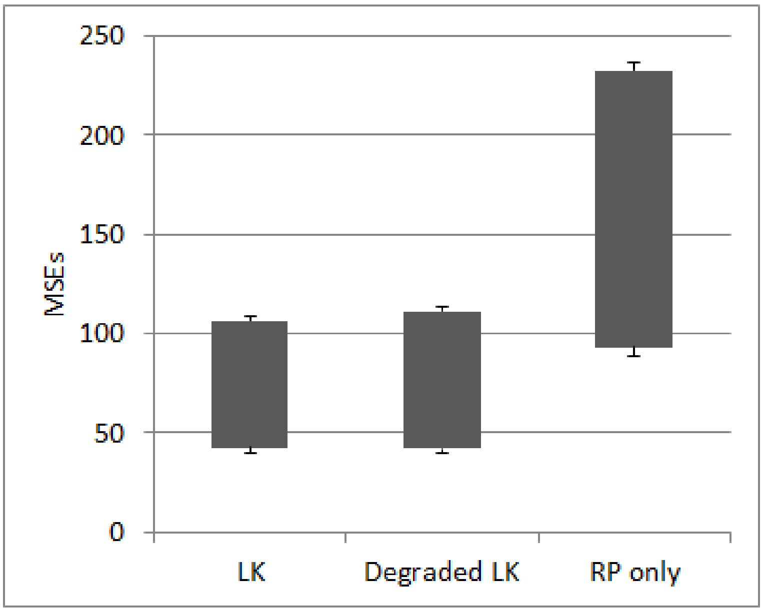


Figure 4.9: The Box and Whisker plot of MSEs of LK, degraded LK and the learning algorithms with a learning set of representative patches only (experiment described in section 4.3.3.2).

Descriptive statistics of this experiment is given in Table 4.10. The Figure 4.9 shows the Box and Whisker plot of MSEs of LK, degraded LK and the learning algorithms with a learning set of representative patches only . Mauchly's test indicates that the assumption of sphericity has been violated ($\chi^2(2) = 42.36$, $p < 0.01$) therefore Greenhouse-Geisser corrected tests are reported ($\varepsilon = .90$). The results show that MSEs are significantly affected by learning set with representative patches, $V = 0.14$, $F(2, 398) = 32.46$, $p < 0.01$.

From the post hoc test we can conclude that (see Table 4.11 and Table 4.12):

- The learning algorithm with learning set of *representative patches only* generates significantly higher MSEs than LK ($p < .01$).
- The learning algorithm with learning set of *representative patches only* generates higher MSEs compared to degraded LK, but not significantly so ($p < .05$).

4.3.4 Discussion

The learning set of displacement fields generates a highly non-convex high-dimensional feature space. From a theoretical standpoint, linear embedding methods such as PCA,

Table 4.11: Tests of Within-Subjects Effects (experiment described in section 4.3.3.2)

Measure:MSE

Source		Type III Sum of Squares	df	Mean Square	F	Sig.
Method	Sphericity Assumed	5.23E+06	2	2.61E+06	22.20	4.13E-10**
	Greenhouse-Geisser	5.23E+06	1.82	2.88E+06	22.20	2.14E-09**
	Huynh-Feldt	5.23E+06	1.82	2.87E+06	22.20	2.00E-09**
	Lower-bound	5.23E+06	1.00	5.23E+06	22.20	3.39E-06**
Error (Method)	Sphericity Assumed	9.40E+07	798	1.18E+05		
	Greenhouse-Geisser	9.40E+07	724.82	1.30E+05		
	Huynh-Feldt	9.40E+07	727.95	1.29E+05		
	Lower-bound	9.40E+07	399.00	2.36E+05		

Sig. (* for $<.05$ and ** for $<.01$)

do not preserve the local properties of the high-dimensional manifold. Unsurprisingly, the learning algorithm with PCA as an embedding technique did not then deliver better performance than that of degraded LK.

By including all possible vector compositions in our learning set, we greatly increase the number of vector fields in the higher dimensional manifold (7 dimensions for 2D motion and 9 dimensions for 3D motion). Consequently, in some trials, LLE could not successfully construct an efficient lower dimensional manifold. LLE was then associated with the highest MSEs among the embedding methods.

Like LLE, LTSA also performs inefficiently against the learning set with composed vector fields. Moreover, in our learning set, we have different patterns of motions (like cardiac motion, respiratory motion etc.). So, high dimensional manifolds may consist of several disjoint components where several of the smallest eigenvalues are about the same magnitude. Therefore, low dimensional manifolds constructed by LTSA may not be accurate.

Laplacian Eigenmaps shows the best result. We know that Laplacian Eigenmaps embedding is non-linear and preserves local properties of the manifold. It can also handle non-convex features, curvatures and corners of a manifold. In that sense, it is more robust than the other three embedding methods.

In all cases, learning algorithm with only representative patches in the learning set cannot improve accuracy of registration. This is because the representative patches are actually the average of all patches in a cluster. Many fine patterns of motion are missing in these representative patches.

Table 4.12: Pairwise Comparisons (experiment described in section 4.3.3.2)

Measure MSE

	1	2	3
1		3.00E-05**	4.77E-14**
2	3.00E-05**		4.77E-02*
3	4.77E-14**	4.77E-02*	

Sig. (* for <.05 and ** for <.01)

4.4 Conclusion

This chapter described the concept of code-book of patches, and the methods we selected for its generation. In particular, we discussed and compared the performances of a variety of manifold embedding approaches. The following chapter will discuss a number of refinements.

Chapter 5

Improving the Flow

5.1 Introduction

Recall that in the fashion of the non local means, our objective is to take advantage of the high degree of redundancy of motion across images in a sequence. To improve an initial flow, we replace the displacement vectors at the center of patches by a Gaussian weighted average of the displacement vectors at the center of similar patches in a learning set.

Given a patch in the initial flow at position (m, n), rather than consider the set of all patches in the learning set, $\mathcal{C} = \left\{ \{(x_i, y_i), \mathbf{q}_i\} \right\}_{i \in \mathcal{I}}$, we restrict ourselves to patches in those clusters whose representative patches best match the one under consideration. Let $\mathcal{J} \subset \mathcal{I}$ be the indices of the selected patches.

So far, we have used as a similarity measure the Gaussian weighted sum of squared differences between the vector polar angles in q_i and in $\mathbf{q}_{m,n}$ with standard deviation σ: $w_{\texttt{vector}}\left(\mathbf{q}_i, \mathbf{q}_{m,n}\right) = \parallel \theta(\mathbf{q}_i) - \theta(\mathbf{q}_{m,n}) \parallel_{\sigma}^{2}$.

In this chapter we discuss the effects of two similarity distances in the learning algorithm, based on the spatial distance between patches (section 5.2) and on intensity differences (5.3). We also explore the effect of iteratively applying the algorithm. In the previous chapters, we performed the experiments on 2D and 3D medical images, for which we did not have ground truth flows. Here, we use the MPI-Sintel dataset, for which ground truth flows are available, and compare the performance of our approach, both qualitatively and quantitatively, against those of state-of-the-art optical flow methods.

5.2 Spatial Distance as an Additional Similarity Measure

We note that patches in the vicinity of each other are more likely to belong to the same moving object and, consequently, to exhibit the same motion, than patches further apart. For instance, patches in the neighbourhood of the heart exhibit a repetitive motion in sync with cardiac motion, one very different, in terms of phase, frequency and direction, from those patches in the neighbourhood of the diaphragm. This motivated us to introduce spatial distance as an additional similarity measure between patches in equation 3.2:

$$(u_{m,n}, v_{m,n}) = \frac{1}{Z} \sum_{i \in \mathcal{J}} \mathrm{e}^{-\frac{\left(\frac{w_{\mathtt{vector}}}{\sigma_{\mathtt{vector}}}\right)^{\gamma_{\mathtt{vector}}} \times \left(\frac{w_{\mathtt{distance}}}{\sigma_{\mathtt{distance}}}\right)^{\gamma_{\mathtt{distance}}}}{h^2}} (u_i, v_i) \tag{5.1}$$

Where $w_{\mathtt{vector}}\left(\mathbf{q}_i, \mathbf{q}_{m,n}\right) = \| \theta(\mathbf{q}_i) - \theta(\mathbf{q}_{m,n}) \|^2_{2,\sigma}$ is the similarity measure the same Gaussian weighted sum of squared differences between the vector polar angles in q_i and in the representative patches $\mathbf{q}_{m,n}$ with standard deviation σ. $w_{\mathtt{distance}}\big((x_i, y_i), (m, n)\big)$ is the Euclidean distance between the centers of both patches. $\gamma_{\mathtt{vector}}$and$\gamma_{\mathtt{distance}}$ control the relative contributions of the weights.

We evaluate below the performance of the learning algorithm without and with spatial distance as a similarity measure.

5.2.1 Results with the 2-D dMRI Dataset

We first evaluated the influence of spatial distance on our 2-D MRI dataset. For our learning approach, we used the following parameters:

- patch radius: $r = 5$,
- parameter for decay of weight: $h = 2$ (as described in section 3.4.3, $h = (r + 1)/3$), $\gamma_{\mathtt{vector}} = 1$ and $\gamma_{\mathtt{distance}} = 1$,
- the *angular difference* is used as a similarity measure
- all the composed fields were included in learning set,

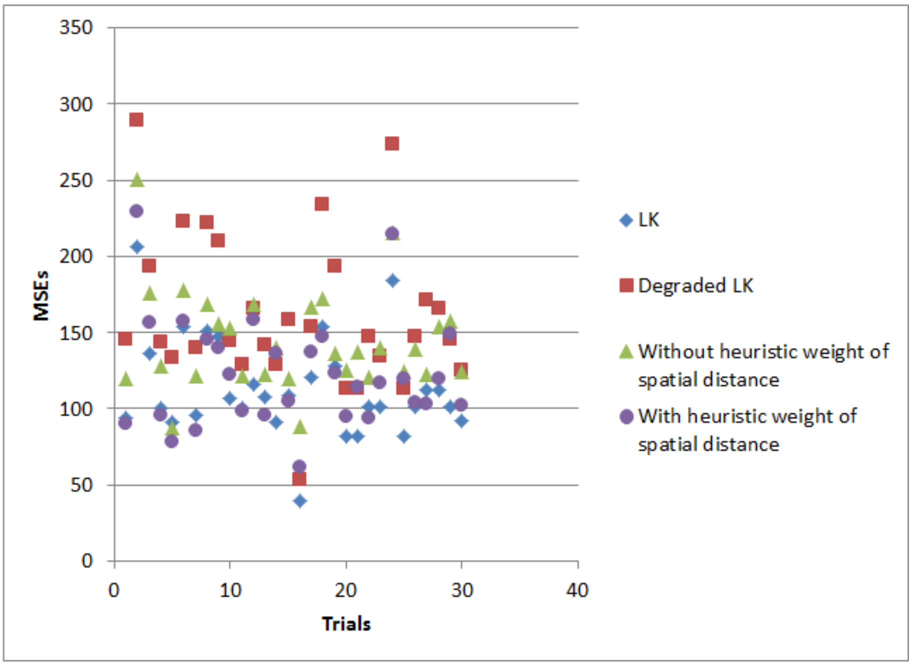


Figure 5.1: MSEs of images registered using LK, degraded LK and learning algorithm with and without heuristic weight of spatial distance (experiment described in section 5.2.1).

Table 5.1: Descriptive Statistics (experiment described in section 5.2.1)

Method	Dependent Variable	Mean	Std. Deviation	N
1	LK	113.47	33.10	30
2	Degraded LK	161.80	49.92	30
3	Without weight of spatial distance	144.59	33.74	30
4	With weight of spatial distance	123.41	36.82	30

- code-book of patches: 2-D Laplacian eigenmaps with a neighborhood of size 6 for the embedding of the feature vector and hierarchical clustering
- inconsistency coefficient threshold of 1.15 as the value of the cutoff argument in hierarchical clustering,
- threshold for the selection of representative patches: maximum 5 degree of angular difference for each vector in a patch.

Results are shown in Table J.1. As per the previous chapters, column 1 shows the trial numbers with detail of trials in Table 2.1. Figure 5.1 shows the MSEs of

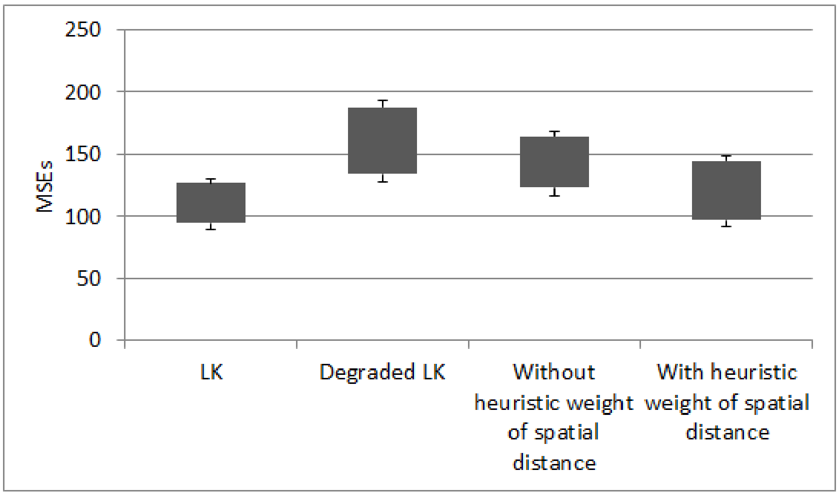


Figure 5.2: The Box and Whisker plot of MSEs of LK, degraded LK and learning algorithms with and without heuristic weight of spatial distance (experiment described in section 5.2.1).

images registered using LK, degraded LK and learning algorithm with and without heuristic weight of spatial distance. The MSEs are computed based on the sum of squared differences between the intensities of the target and registered images. The MSEs of LK and degraded LK optical flow algorithms are given in column 2 and 3 respectively. The MSEs of the learning algorithm without and with spatial distance are given in columns 4 and 5 respectively.

We performed a repeated measures ANOVA to compare the performances of the different approaches (rationale for using repeated measures ANOVA is discussed in section 2.4.3):

- Method 1: LK
- Method 2: Degraded LK
- Method 3: Learning algorithm without weight of spatial distance
- Method 4: Learning algorithm without weight of spatial distance

Descriptive statistics of this experiment is given in Table 5.1. Figure 5.2 shows the Box and Whisker plot of MSEs of LK, degraded LK and learning algorithms with and without heuristic weight of spatial distance. Mauchly's test indicates that the assumption of sphericity has been violated ($\chi^2(5) = 49.69$, $p < 0.05$) therefore we report Greenhouse-Geisser corrected tests ($\varepsilon = .50$). The results show that the MSEs are significantly affected by the choice of method ($V = 0.96$, $F(3, 27) = 243.63$, $p < 0.05$.

Table 5.2: Tests of Within-Subjects Effects (experiment described in section 5.2.1)

Measure:MSE

Source		Type III Sum of Squares	df	Mean Square	F	Sig.
Method	Sphericity Assumed	4.22E+04	3	1.41E+04	65.95	2.46E-22**
	Greenhouse-Geisser	4.22E+04	1.51	2.80E+04	65.95	2.25E-12**
	Huynh-Feldt	4.22E+04	1.57	2.68E+04	65.95	8.27E-13**
	Lower-bound	4.22E+04	1.00	4.22E+04	65.95	5.90E-09**
Error (Method)	Sphericity Assumed	1.85E+04	87	213.10		
	Greenhouse-Geisser	1.85E+04	43.74	423.82		
	Huynh-Feldt	1.85E+04	45.63	406.34		
	Lower-bound	1.85E+04	29.00	639.29		

Sig. (* for $<$.05 and ** for $<$.01)

Table 5.3: Pairwise Comparisons (experiment described in section 5.2.1)

Measure: MSE

	1	2	3	4
1		5.13E-15**	1.76E-11**	6.35E-03**
2	5.13E-15**		2.02E-03**	2.31E-08**
3	1.76E-11**	2.02E-03**		8.71E-13**
4	6.35E-03**	2.31E-08**	8.71E-13**	

Sig. (* for $<$.05 and ** for $<$.01)

From the post hoc test we conclude that (see Table 5.2 and Table 5.3):

- The learning algorithm with spatial distance as an additional similarity measure generated significantly lower MSEs than LK ($p < .05$) and degraded LK ($p < .05$).
- The use of spatial distance significantly improved the performance of the learning algorithm in terms of MSEs ($p < .05$).

5.2.2 Results with the 3-D Gated CT Dataset

We also evaluated the influence of spatial distance on our 3-D gated CT dataset, with the same parameters as above. The table in Appendix K.1 shows the results.

Table 5.4: Descriptive Statistics (experiment described in section 5.2.2)

Method	Dependent Variable	Mean	Std. Deviation	N
1	LK	198.34	689.61	400
2	Degraded LK	303.54	1054.97	400
3	Without weight of spatial distance	147.37	475.26	400
4	With weight of spatial distance	145.33	475.28	400

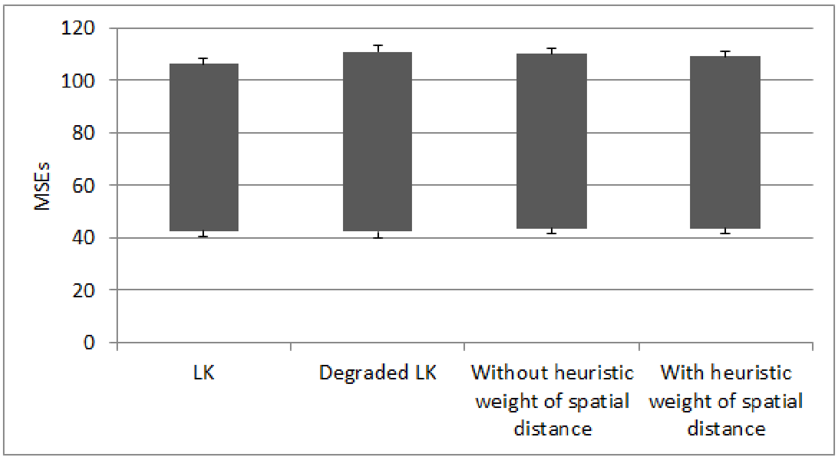


Figure 5.3: The Box and Whisker plot of MSEs of LK, degraded LK and learning algorithms with and without heuristic weight of spatial distance (experiment described in section 5.2.2).

Table 5.5: Tests of Within-Subjects Effects (experiment described in section 5.2.2)
Measure:MSE

Source		Type III Sum of Squares	df	Mean Square	F	Sig.
Method	Sphericity Assumed	6.59E+06	3	2.20E+06	19.66	1.92E-12**
	Greenhouse-Geisser	6.59E+06	1.13	5.84E+06	19.66	4.35E-06**
	Huynh-Feldt	6.59E+06	1.13	5.84E+06	19.66	4.32E-06**
	Lower-bound	6.59E+06	1.00	6.59E+06	19.66	1.20E-05**
Error (Method)	Sphericity Assumed	1.34E+08	1197	1.12E+05		
	Greenhouse-Geisser	1.34E+08	450.09	2.97E+05		
	Huynh-Feldt	1.34E+08	450.49	2.97E+05		
	Lower-bound	1.34E+08	399.00	3.35E+05		

Sig. (* for <.05 and ** for <.01)

We performed a repeated measures ANOVA to compare the performances of the same approaches as those described in the previous section (rationale for using repeated measures ANOVA is discussed in section 2.4.3).

Descriptive statistics of this experiment is given in Table 5.4. Figure 5.3 shows the Box and Whisker plot of MSEs of LK, degraded LK and learning algorithms with and without heuristic weight of spatial distance. Mauchly's test indicates that the assumption of sphericity has been violated ($\chi^2(5) = 5078.87$, $p < 0.01$) therefore Greenhouse-Geisser corrected tests are reported ($\varepsilon = 2.85E - 06$). The results show that MSEs are significantly affected by the choice of method ($V = 0.78$, $F(3, 397) = 459.29$, $p < 0.01$.

From the post hoc test we conclude that (see Table 5.5 and Table 5.6):

- The learning algorithm with spatial distance as an additional similarity measure generated significantly lower MSEs than LK ($p < .01$) and degraded LK ($p < .01$).
- The use of spatial distance significantly improved the performance of the learning algorithm in terms of MSEs ($p < .01$).

Table 5.6: Pairwise Comparisons (experiment described in section 5.2.2)

Measure: MSE

	1	2	3	4
1		3.00E-05**	2.05E-04**	1.14E-04**
2	3.00E-05**		7.30E-06**	5.56E-06**
3	2.05E-04**	7.30E-06**		9.91E-129**
4	1.14E-04**	5.56E-06**	9.91E-129**	

Sig. (* for <.05 and ** for <.01)

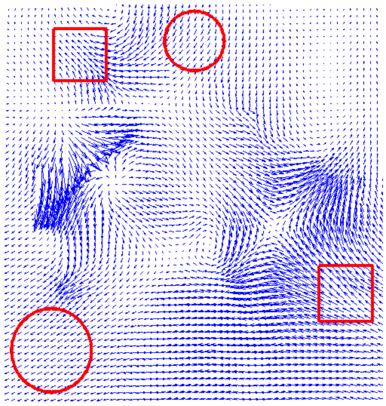


Figure 5.4: Similar motion in different spatial parts of a deformation field.

5.2.3 Discussion

These results suggest that the addition of spatial distance improves performances. Indeed, similar motions tend to occur in the same spatial area. For instance, in our 2-D dMRI dataset, respiratory motion around the diaphragm produces patches with very similar vectors in the lower part of each deformation field whereas cardiac motion produces similar patches around the middle part of those same fields (see regions highlighted in red on Figure 5.4). Obviously, we do not want to consider similar patches indiscriminately. For an example, we may not want to combine similar patches of vectors for motion of diaphragm with cardiac motion. When we use spatial distance, we actually prioritize vectors that are in a closer neighborhood.

5.3 Intensity Difference as an Additional Similarity Measure

We further note that patches with similar intensity patterns are more likely to correspond to the same moving objects than patches with different intensity patterns, and that a moving object in a sequence is likely to exhibit similar motion across consecutive images. This motivated us to introduce intensity difference as an additional similarity measure between patches in equation 5.1:

$$(u_{m,n}, v_{m,n}) = \frac{1}{Z} \sum_{i \in \mathcal{J}} \mathrm{e}^{-\frac{\left(\frac{w_{\texttt{vector}}}{\sigma_{\texttt{vector}}}\right)^{\gamma_{\texttt{vector}}} \times \left(\frac{w_{\texttt{distance}}}{\sigma_{\texttt{distance}}}\right)^{\gamma_{\texttt{distance}}} \times \left(\frac{w_{\texttt{intensity}}}{\sigma_{\texttt{intensity}}}\right)^{\gamma_{\texttt{intensity}}}}{h^2}} (u_i, v_i) \tag{5.2}$$

Where $w_{\texttt{vector}}\left(\mathbf{q}_i, \mathbf{q}_{m,n}\right) = \parallel \theta(\mathbf{q}_i) - \theta(\mathbf{q}_{m,n}) \parallel_{2,\sigma}^2$ is the similarity measure the same Gaussian weighted sum of squared differences between the vector polar angles in q_i and in the representative patches $\mathbf{q}_{m,n}$ with standard deviation σ . We use for $w_{\texttt{intensity}}\left(\mathbf{p}_i, \mathbf{p}_{m,n}\right)$ the Gaussian weighted sum of squared differences between both matrices of intensities. $w_{\texttt{distance}}\big((x_i, y_i), (m, n)\big)$ is the Euclidean distance between the centers of both patches. $\gamma_{\texttt{vector}}$, $\gamma_{\texttt{distance}}$ and $\gamma_{\texttt{intensity}}$ control the relative contributions of the weights.

We evaluate below the performance of the learning algorithm without and with intensity distance as an similarity measure.

5.3.1 Results with the 2-D dMRI Dataset

We evaluated the influence of intensity difference on our 2-D dMRI dataset and the same parameters as in Section 5.2 above. Results are shown in Table L.1. Column 1 shows the trial numbers; detail of trials can be found in Table 2.1. Figure 5.5 shows the MSEs of images registered using LK, degraded LK and learning algorithm with and without heuristic weight of intensity. The MSEs of LK and degraded LK optical flow algorithms are given in column 2 and 3 respectively. The MSEs of the learning algorithm without and with intensity difference are given in columns 4 and 5 respectively. For our learning approach, we used the following parameters:

- patch radius: $r = 5$,

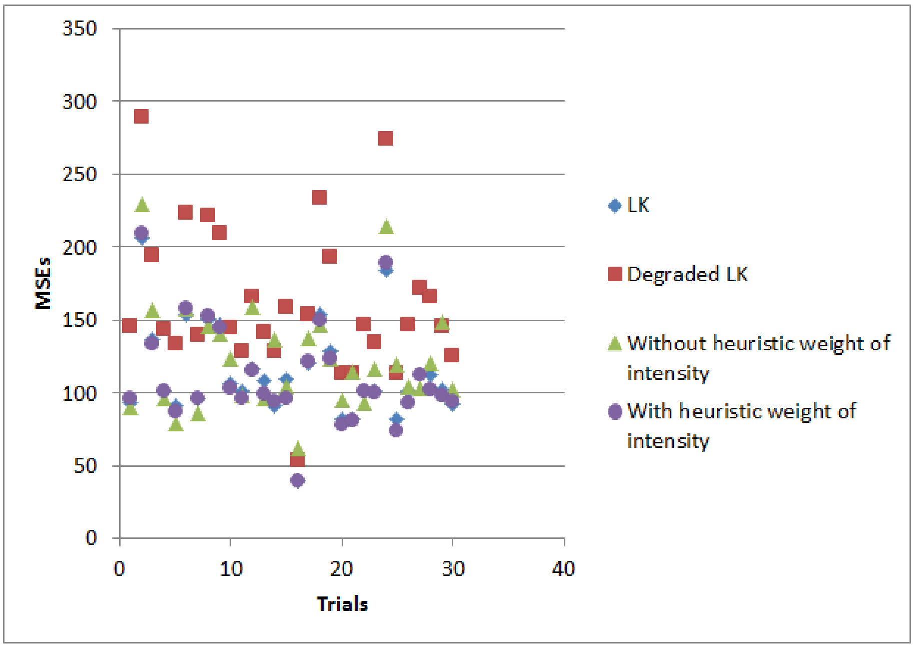


Figure 5.5: MSEs of images registered using LK, degraded LK and learning algorithm with and without heuristic weight of intensity (experiment described in section 5.3).

- parameter for decay of weight: $h = 2$ (as described in section 3.4.3, $h = (r + 1)/3$), $\gamma_{\texttt{vector}} = 1$, $\gamma_{\texttt{distance}} = 1$ and $\gamma_{\texttt{intensity}} = 1$
- the *angular difference* and *heuristic weight of spatial distance* are used as a similarity measure
- all the composed fields were included in learning set,
- code-book of patches: 2-D Laplacian eigenmaps with a neighborhood of size 6 for the embedding of the feature vector and hierarchical clustering
- inconsistency coefficient threshold of 1.15 as the value of the cutoff argument in hierarchical clustering,
- threshold for the selection of representative patches: maximum 5 degree of angular difference for each vector in a patch.

Table 5.7: Descriptive Statistics (experiment described in section 5.3)

Method	Dependent Variable	Mean	Std. Deviation	N
1	LK	113.47	33.10	30
2	Degraded LK	161.80	49.92	30
3	Without weight of intensity	111.32	34.59	30
4	With weight of intensity	123.41	36.82	30

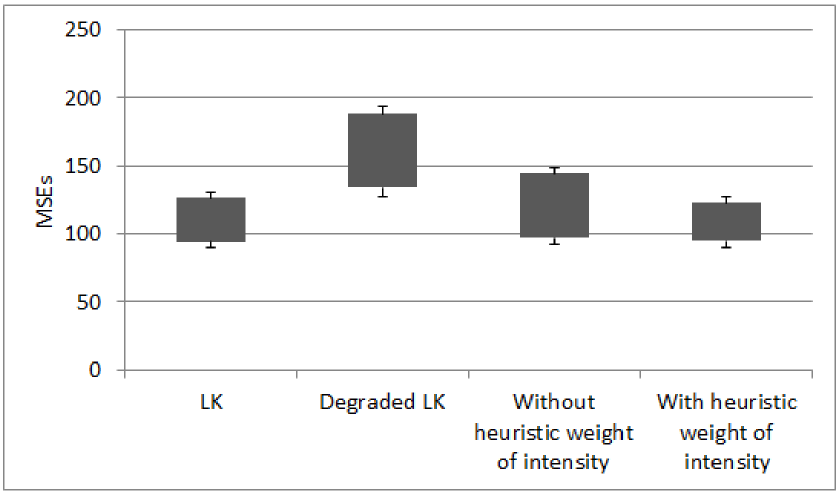


Figure 5.6: The Box and Whisker plot of MSEs of LK, degraded LK and learning algorithms with and without heuristic weight of intensity (experiment described in section 5.3).

We performed a repeated measures ANOVA to compare the performances of the different approaches (rationale for using repeated measures ANOVA is discussed in section 2.4.3):

- Method 1: LK
- Method 2: Degraded LK
- Method 3: Learning algorithm without intensity difference as an similarity measure
- Method 4: Learning algorithm with intensity difference as an additional similarity measure

Descriptive statistics of this experiment is given in Table 5.7. Figure 5.6 shows the Box and Whisker plot of MSEs of LK, degraded LK and learning algorithms with and without heuristic weight of intensity. Mauchly's test indicates that the assumption of sphericity has been violated ($\chi^2(5) = 66.88$, $p < 0.05$) therefore Greenhouse-Geisser corrected tests are reported ($\varepsilon = .55$). The results show that the MSEs are significantly affected by the choice of method ($V = 0.92$, $F(3, 27) = 97.53$, $p < 0.05$.

From the post hoc test we conclude that (see Table 5.8 and Table 5.9):

- The learning algorithm with intensity difference as an additional similarity measure generated significantly lower MSEs than LK ($p < .05$) and degraded LK ($p < .05$).
- The use of intensity difference significantly improved the performance of the learning algorithm in terms of MSEs ($p < .05$).

5.3.2 Discussion

These results suggest that the learning algorithm with heuristic weight of intensity shows better performance than learning algorithm without heuristic weight of intensity. In particular, it improves the flow estimation of motion around the boundary of structures in the images. A moving object may have different intensity compared to the environment around it. But, the estimated flow around the boundary of that

Table 5.8: Tests of Within-Subjects Effects (experiment described in section 5.3)

Measure:MSE

Source		Type III Sum of Squares	df	Mean Square	F	Sig.
Method	Sphericity Assumed	4.96E+04	3	1.65E+04	95.23	2.16E-27**
	Greenhouse-Geisser	4.96E+04	1.65	3.00E+04	95.23	3.81E-16**
	Huynh-Feldt	4.96E+04	1.74	2.85E+04	95.23	7.11E-17**
	Lower-bound	4.96E+04	1.00	4.96E+04	95.23	1.15E-10**
Error (Method)	Sphericity Assumed	1.51E+04	87	173.48		
	Greenhouse-Geisser	1.51E+04	47.93	314.88		
	Huynh-Feldt	1.51E+04	50.46	299.10		
	Lower-bound	1.51E+04	29.00	520.43		

Sig. (* for $<$.05 and ** for $<$.01)

Table 5.9: Pairwise Comparisons (experiment described in section 5.3)

Measure: MSE

	1	2	3	4
1		5.13E-15**	1.35E-02*	6.35E-03**
2	5.13E-15**		7.14E-16**	2.31E-08**
3	1.35E-02*	7.14E-16**		9.93E-04**
4	6.35E-03**	2.31E-08**	9.93E-04**	

Sig. (* for $<$.05 and ** for $<$.01)

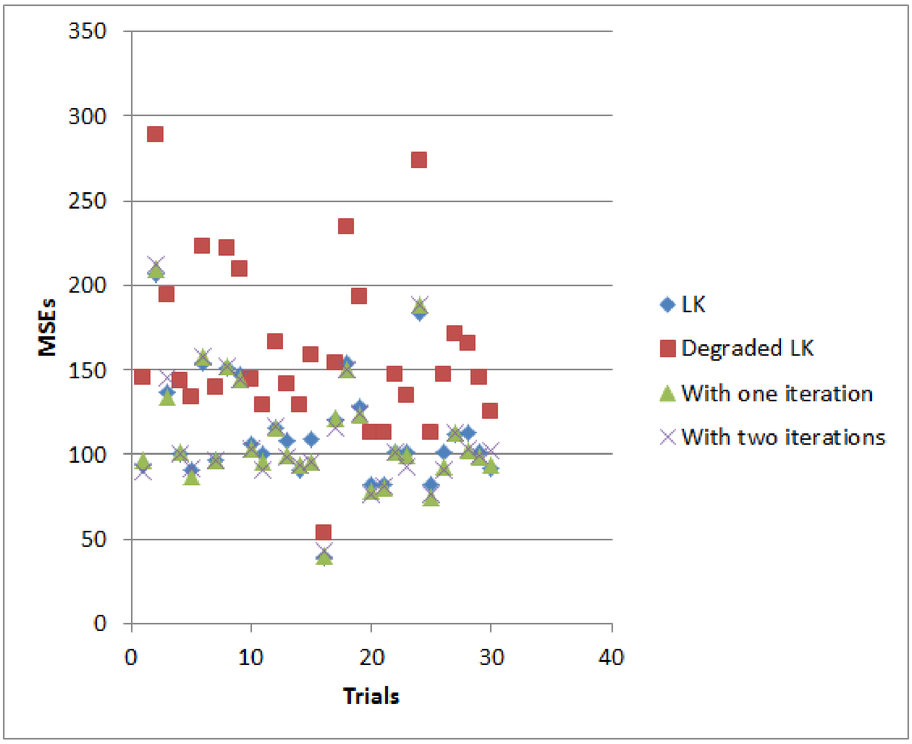


Figure 5.7: MSEs of images registered using LK, degraded LK and learning algorithm with one iteration and two iterations (experiment described in section 5.4.1).

moving object may become less accurate due to the regularization of flow. With this heuristic weight, the learning algorithm decreases weight of the filtered flow when it finds higher intensity difference at the boundary. Consequently, the filtered flow becomes more accurate.

5.4 Iterating the Learning Algorithm

We observe that many registration approaches are iterative in nature, with the estimated flow being improved at each step. Consequently, we wanted to check whether successively applying the learning algorithm would also improve the overall performance. The idea is to use the learned flow field (the one estimated by the learning algorithm) from the first iteration as an input for the second iteration.

Table 5.10: Descriptive Statistics (experiment described in section 5.4.1)

Method	Dependent Variable	Mean	Std. Deviation	N
1	LK	113.47	33.10	30
2	Degraded LK	161.80	49.92	30
3	With one iteration	111.32	34.59	30
4	With two iterations	111.65	34.95	30

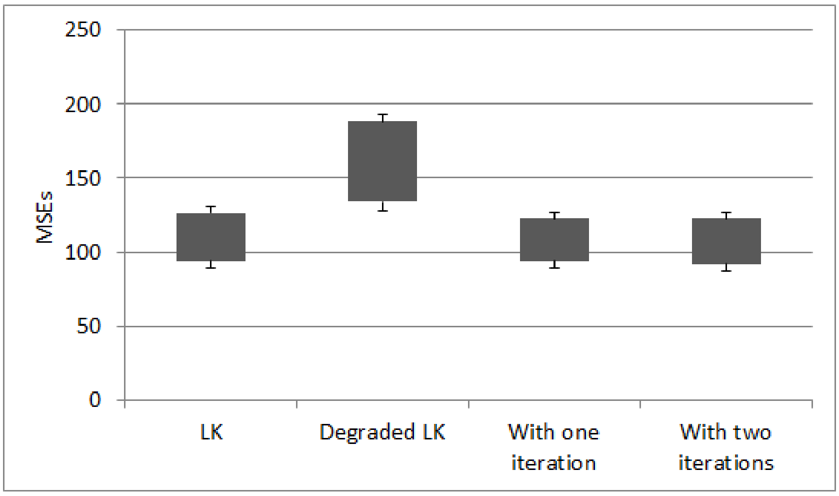


Figure 5.8: The Box and Whisker plot of MSEs of LK, degraded LK and learning algorithms with and without iteration (experiment described in section 5.4.1).

5.4.1 Results with the 2-D dMRI Dataset

For the learning algorithm, we used all three patch similarity measures: vector angular difference, spatial distance and intensity difference, with the same parameters as in the previous experiments.

Table M.1 shows the results. Column 1 shows the trial numbers; detail of trials can be found in Table 2.1. Figure 5.7 MSEs of images registered using LK, degraded LK and learning algorithm with one iteration and two iterations. The MSEs of LK and degraded LK optical flow algorithms are given in column 2 and 3 respectively. MSEs of learning algorithm with one and two iterations are given in columns 4 and 5 respectively. For our learning approach, we used the following parameters:

- patch radius: $r = 5$,
- parameter for decay of weight: $h = 2$ (as described in section 3.4.3, $h = (r + 1)/3$), $\gamma_{\texttt{vector}} = 1$, $\gamma_{\texttt{distance}} = 1$ and $\gamma_{\texttt{intensity}} = 1$
- the *angular difference, heuristic weight of spatial distance* and *heuristic weight of intensity* re used as a similarity measure
- all the composed fields were included in learning set,
- code-book of patches: 2-D Laplacian eigenmaps with a neighborhood of size 6 for the embedding of the feature vector and hierarchical clustering
- inconsistency coefficient threshold of 1.15 as the value of the cutoff argument in hierarchical clustering,
- threshold for the selection of representative patches: maximum 5 degree of angular difference for each vector in a patch.

We performed a repeated measures ANOVA to compare the performances of the different approaches (rationale for using repeated measures ANOVA is discussed in section 2.4.3):

- Method 1: LK
- Method 2: Degraded LK
- Method 3: Learning algorithm with one iteration
- Method 4: Learning algorithm with two iterations

Table 5.11: Tests of Within-Subjects Effects (experiment described in section 5.4.1)

Measure:MSE

Source		Type III Sum of Squares	df	Mean Square	F	Sig.
Method	Sphericity Assumed	5.56E+04	3	1.85E+04	224.43	7.88E-41**
	Greenhouse-Geisser	5.56E+04	1.20	4.64E+04	224.43	9.69E-18**
	Huynh-Feldt	5.56E+04	1.22	4.55E+04	224.43	4.93E-18**
	Lower-bound	5.56E+04	1.00	5.56E+04	224.43	3.47E-15**
Error (Method)	Sphericity Assumed	7.18E+03	87	82.52		
	Greenhouse-Geisser	7.18E+03	34.75	206.56		
	Huynh-Feldt	7.18E+03	35.42	202.70		
	Lower-bound	7.18E+03	29.00	247.55		

Sig. (* for <.05 and ** for <.01)

Table 5.12: Pairwise Comparisons (experiment described in section 5.4.1)

Measure: MSE

	1	2	3	4
1		5.13E-15**	1.35E-02*	9.38E-02
2	5.13E-15**		7.14E-16**	7.87E-16**
3	1.35E-02*	7.14E-16**		6.33E-01
4	9.38E-02	7.87E-16**	6.33E-01	

Sig. (* for <.05 and ** for <.01)

Descriptive statistics of this experiment is given in Table 5.10. Figure 5.8 shows the Box and Whisker plot of MSEs of LK, degraded LK and learning algorithms with and without iteration. Mauchly's test indicates that the assumption of sphericity has been violated ($\chi^2\,(5) = 91.65$, $p < 0.05$) therefore Greenhouse-Geisser corrected tests are reported ($\varepsilon = .4$). The results show that MSEs are significantly affected by the choice of method ($V = 0.90$, $F\,(3, 27) = 81.59$, $p < 0.05$.

From the post hoc test we conclude that (see Table 5.11 and Table 5.12):

- The learning algorithm with heuristic weight of intensity generates significantly lower MSEs than both LK ($p < .05$) and degraded LK ($p < .05$).

- The learning algorithm with two iterations generates more MSEs than the learning algorithm with one iteration, but statistically it is not significant ($p > .05$).

Table 5.13: Descriptive Statistics (experiment described in section 5.4.2)

Method	Dependent Variable	Mean	Std. Deviation	N
1	LK	198.34	689.61	400
2	Degraded LK	303.54	1054.97	400
3	Learning algorithm with one iteration	143.88	475.23	400
4	Learning algorithm with two iterations	143.75	474.67	400

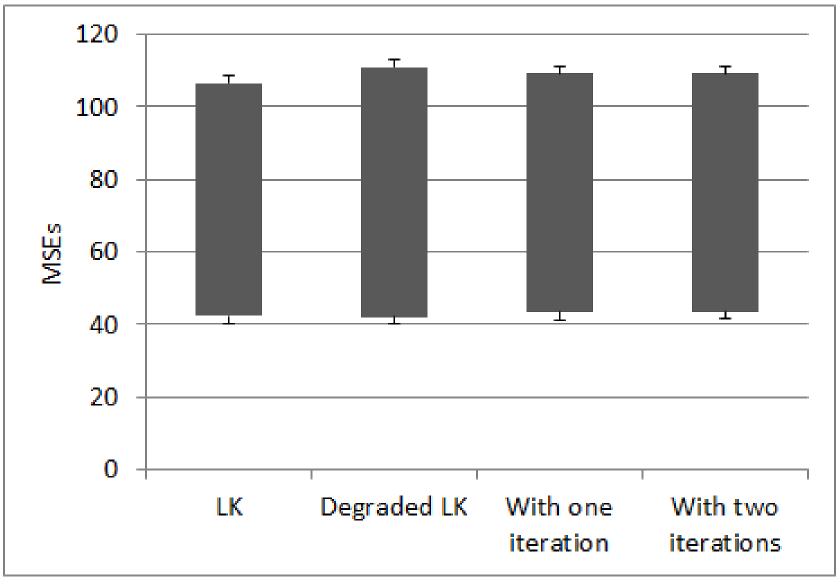


Figure 5.9: The Box and Whisker plot of MSEs of LK, degraded LK and learning algorithms with and without iteration (experiment described in section 5.4.2).

5.4.2 Results with the 3-D Gated CT Dataset

We also evaluated the influence of iterating the learning algorithm on our 3-D gated CT dataset, with the same parameters as above. The table in Appendix N.1 shows the results. Column 1 shows the trial numbers; detail of trials can be found in the table in Appendix A.1.

We performed a repeated measures ANOVA to compare the same methods (rationale for using repeated measures ANOVA is discussed in section 2.4.3).

Descriptive statistics of this experiment is given in Table 5.15. Figure 5.9 shows the Box and Whisker plot of MSEs of LK, degraded LK and learning algorithms with and without iteration. Mauchly's test indicates that the assumption of sphericity has

Table 5.14: Tests of Within-Subjects Effects (experiment described in section 5.4.2)

Measure:MSE

Source		Type III Sum of Squares	df	Mean Square	F	Sig.
Method	Sphericity Assumed	6.8E6	3	2267870	20.248	<.001**
	Greenhouse-Geisser	6.8E6	1.129	6025002	20.248	<.001**
	Huynh-Feldt	6.8E6	1.130	6019584	20.248	<.001**
	Lower-bound	6.8E6	1.000	6803609	20.248	<.001**
Error (Method)	Sphericity Assumed	1.34E8	1197	112003		
	Greenhouse-Geisser	1.34E8	450.56	297555		
	Huynh-Feldt	1.34E8	450.97	297288		
	Lower-bound	1.34E8	399.00	336008		

Sig. (* for <.05 and ** for <.01)

Table 5.15: Pairwise Comparisons (experiment described in section 5.4.2)

Measure MSE

	1	2	3	4
1		<.001**	<.001**	<.001**
2	<.001**		<.001**	<.001**
3	<.001**	<.001**		.695
4	<.001**	<.001**	.695	

Sig. (* for <.05 and ** for <.01)

been violated ($\chi^2(5) = 3697$, $p < 0.01$) therefore Greenhouse-Geisser corrected tests are reported ($\varepsilon = .376$). The results show that the MSEs are significantly affected by the choice of method ($V = 0.053$, $F(3, 397) = 7.379$, $p < 0.01$.

From the post hoc test we conclude that (see Table 5.14 and Table 5.15):

- The learning algorithm with iteration generates significantly lower MSEs than both LK ($p < .01$) and degraded LK ($p < .01$).

- The learning algorithm with two iterations has lower MSEs than the learning algorithm with one iteration approach, but statistically it is not significant ($p > .05$).

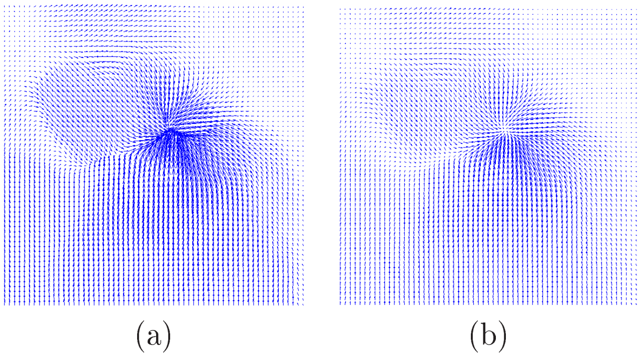


Figure 5.10: Over-regularization of flow field due to iteration; (a) initial learned flow field before iteration; (b) over-smoothed flow after iteration.

5.4.3 Discussion

These results suggest that iterating the learning algorithm does not significantly improve performances. This might be because we apply the learning algorithm on the same learning set in the second iteration which results in an over-regularization of the flow fields. Figure 5.10 (a) shows a patch around the heart of the initial learned flow field between frame #105 and #106. Figure 5.10 (b) show the same patch after second iteration. Clear we can observe over-smoothing compared to the initial learned flow field.

5.5 The Full Monty

Informed by the results from the previous experiments, we formulated the best learning approach, which combines all previously introduced similarity measures, the use of a code-book/representative patches system and only one iteration. Equation 3.2 becomes:

$$(u_{m,n}, v_{m,n}) = \frac{1}{Z}\sum_{i\in\mathcal{J}} \mathrm{e}^{-\frac{\left(\frac{w_{\mathtt{vector}}}{\sigma_{\mathtt{vector}}}\right)^{\gamma_{\mathtt{vector}}} \times \left(\frac{w_{\mathtt{distance}}}{\sigma_{\mathtt{distance}}}\right)^{\gamma_{\mathtt{distance}}} \times \left(\frac{w_{\mathtt{intensity}}}{\sigma_{\mathtt{intensity}}}\right)^{\gamma_{\mathtt{intensity}}}}{h^2}} (u_i, v_i) \tag{5.3}$$

Where $w_{\mathtt{vector}}(\mathbf{q}_i, \mathbf{q}_{m,n}) = \| \theta(\mathbf{q}_i) - \theta(\mathbf{q}_{m,n}) \|^2_{2,\sigma}$ is the similarity measure the same Gaussian weighted sum of squared differences between the vector polar angles in q_i and in the representative patches $\mathbf{q}_{m,n}$ with standard deviation σ . $w_{\mathtt{distance}}\big((x_i, y_i), (m, n)\big)$ is the Euclidean distance between the centers of both patches. We use for $w_{\mathtt{intensity}}(\mathbf{p}_i, \mathbf{p}_{m,n})$

the Gaussian weighted sum of squared differences between both matrices of intensities. $\gamma_{\texttt{vector}}$,$\gamma_{\texttt{distance}}$ and $\gamma_{\texttt{intensity}}$ control the relative contributions of the weights.

The pseudo-code of the implementation of equation 5.3 is given in Algorithm 5.1.

We evaluated performances on sequences from the MPI-Sintel dataset (see section 2.3.3 for details). This dataset exhibits both larger motions and increased complexity (such as motion and defocus blur, atmospheric effects or specular surfaces), which makes it a lot more challenging. We compared the proposed approach against the classic, coarse-to-fine Horn and Schunck technique (HS) and three recent, state-of-the-art approaches: Sun et al.'s median filtering algorithm (SMF) (for detail see section 2.1.4), Large Displacement Optical Flow (LDOF) (for detail see section 2.1.5) and DeepFlow (for detail see section 2.1.6). We used publicly available implementations for all of them, with default settings.

We selected 93 trials from 8 sequences (Alley1, Ambush7, Bamboo1, Bamboo2, Market2, Market6, Shaman2 and Shaman3) based on their exhibiting large motions of small structures, since it is the focus of this thesis. In order to ensure large displacements, we picked non-consecutive frames for the test cases with the immediately preceding three frames in the learning set, i.e. two proceeding pairwise flow fields and their vector composition. The complete trial table is given in section O.1.

Since ground truth flows are available for MPI-Sintel, we used AAE and AEPE to measure performances (see section 2.4 for details). We computed the vector composition of the ground truth flows since we are using non-consecutive frames in our test cases. Interestingly we could not have used MSEs here. This is for essentially two reasons:

1. The MPI-Sintel dataset consists of complex, dynamic sequences where new objects may suddenly appear, such as large building structures, new characters etc. Consequently, the target image in a trial may be substantially different from the source image. Therefore, computing the MSEs of intensity differences between the registered source image and the target image may yield artificially high values, which would not be so indicative of the quality of the estimated flow.

2. The intensity of objects may be altered by shading, which would, again, induce artificially high MSE values.

Algorithm 5.1 Pseudo Code of Learning Algorithm with Improved Flow

function [*learned_flow_field*] = learningAlgorithm(*initial_flow*, *learning_set*, *threshold*)
% *initial_flow* : the initially estimated flow field computed ...
%   using an standard optic flow algorithm such as HS
% *learning_set* : the set of all deformation fields in *a priori*
% *threshold* : used for selecting the representative patches
% there are three similarity measures used in this algorithm; they are ...
%   *similarity_vector, similarity_spatial_distance* and *similarity_intensity*

% use Hierarchical clustering algorithm; output *idx* is the set of indices of the clusters
idx=cluser(*learning_set*);

compute *representativePatches* by computing the average patch of each cluster

loop_1 : for each index *(i,j)* in the initial_flow
{
% initialization of the weight of similarity measures for each patch
sum_of_weight=0;
compute the *similarity_vector* for *(i,j)*
loop_2 : for each patch in each cluster in *idx* ...
that satisfy the *threshold* on *similarity_measure_vector*
{
compute the *similarity_spatial_distance*
compute the *similarity_intensity*
compute the *overall_weight_of_similarity_of_this_patch* ...
using the equation 5.3

sum_of_weight=*sum_of_weight* + ...
overall_weight_of_similarity_of_this_patch

weighted_vector=*weighted_vector* + ...
overall_weight_of_similarity_of_this_patch × ...
vector_at_the_center_of_the_patch
}
end_of_loop_2

% normalization
learned_flow_field(i,j)=*weighted_vector* / *sum_of_weight*;
}
end_of_loop_1

regularize the *learned_flow_field* using Gaussian filter

Table 5.16: Descriptive Statistics (experiment described in section 5.5.1)

Method	Dependent Variable	Mean	Std. Deviation	N
1	HS	11.744	7.515	93
2	SMF	6.574	3.516	93
3	LDOF	7.460	7.864	93
4	DeepFlow	5.644	2.803	93
5	Learning Algorithm	5.799	2.593	93

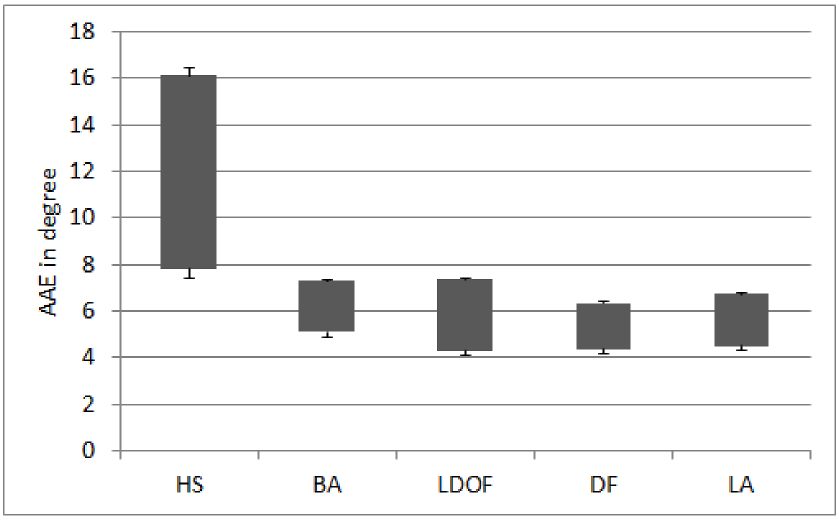


Figure 5.11: The Box and Whisker plot of AAEs of different registration methods (experiment described in section 5.5.1).

Pairwise optical flows between consecutive frames in the learning set were estimated using HS rather LK since HS yields a higher density of flow vectors. We used the same parameters as before.

For our learning approach, we used the following parameters:

- code-book of patches: 2-D Laplacian eigenmaps with a neighborhood of size 6 for the embedding of the feature vector and hierarchical clustering
- patch radius: $r = 5$, sigma $\sigma = 2$, filtering: $h = 0.5$, $\gamma_{\mathtt{distance}} = 1$, $\gamma_{\mathtt{intensity}} = 1$ and $\gamma_{\mathtt{vector}} = 1$,
- inconsistency coefficient threshold of 1.15 as the value of the cutoff argument in hierarchical clustering,
- all the composed fields were included in learning set,
- threshold for the selection of representative patches: maximum 5 degree of angular difference for each vector in a patch..

5.5.1 Average Angular Error (AAE)

We performed a repeated measures ANOVA to compare the performances of the different approaches (rationale for using repeated measures ANOVA is discussed in section 2.4.3):

- Method 1: HS
- Method 2: SMF
- Method 3: LDOF
- Method 4: DeepFlow
- Method 5: Learning Algorithm

Descriptive statistics of this experiment is given in Table 5.16. Figure 5.11 shows the Box and Whisker plot of AAEs of different registration methods. Mauchly's test indicates that the assumption of sphericity has been violated ($\chi^2\,(9) = 331.783$, $p < 0.01$) therefore Greenhouse-Geisser corrected tests are reported ($\varepsilon = .442$). The results show that AAEs are significantly affected by the choice of method ($V = 0.637$, $F\,(4, 89) = 39.112$, $p < 0.01$.

Table 5.17: Tests of Within-Subjects Effects (experiment described in section 5.5.1)

Measure:MSE

Source		Type III Sum of Squares	df	Mean Square	F	Sig.
Method	Sphericity Assumed	2342.846	4	585.712	50.704	5.53E-34**
	Greenhouse-Geisser	2342.846	1.767	1325.516	50.704	1.97E-16**
	Huynh-Feldt	2342.846	1.800	1301.912	50.704	1.10E-16**
	Lower-bound	2342.846	1.000	2342.846	50.704	2.32E-10**
Error (Method)	Sphericity Assumed	4251.011	368	11.552		
	Greenhouse-Geisser	4251.011	162.610	26.142		
	Huynh-Feldt	4251.011	165.558	25.677		
	Lower-bound	4251.011	92.000	46.207		

Sig. (* for <.05 and ** for <.01)

Table 5.18: Pairwise Comparisons (experiment described in section 5.5.1)

Measure MSE

	1	2	3	4	5
1		2.09E-16**	2.52E-13**	3.59E-17**	2.98E-18**
2	2.09E-16**		1.41E-01	1.80E-06**	4.82E-05**
3	2.52E-13**	1.41E-01		9.88E-03**	1.04E-01
4	3.59E-17**	1.80E-06**	9.88E-03**		3.17E-01
5	2.98E-18**	4.82E-05**	1.04E-01	3.17E-01	

Sig. (* for <.05 and ** for <.01)

From the post hoc test we conclude that (see Table 5.17 and Table 5.18):

- The learning algorithm generated significantly lower AAEs than both HS ($p < .05$) and SMF ($p < .05$).
- The AAEs of the learning algorithm is not significantly difference from those of LDOF ($p <> .05$) and DeepFlow ($p > .05$).

5.5.2 Average Endpoint Error (AEPE)

Descriptive statistics of this experiment is given in Table 5.19. We performed a repeated measures ANOVA to compare the same methods (rationale for using repeated measures ANOVA is discussed in section 2.4.3).

Table 5.19: Descriptive Statistics (experiment described in section 5.5.2)

Method	Dependent Variable	Mean	Std. Deviation	N
1	HS	3.3877	6.22225	93
2	SMF	2.3245	4.80320	93
3	LDOF	2.3619	4.80422	93
4	DeepFlow	2.1576	4.77292	93
5	Learning Algorithm	1.8390	3.92781	93

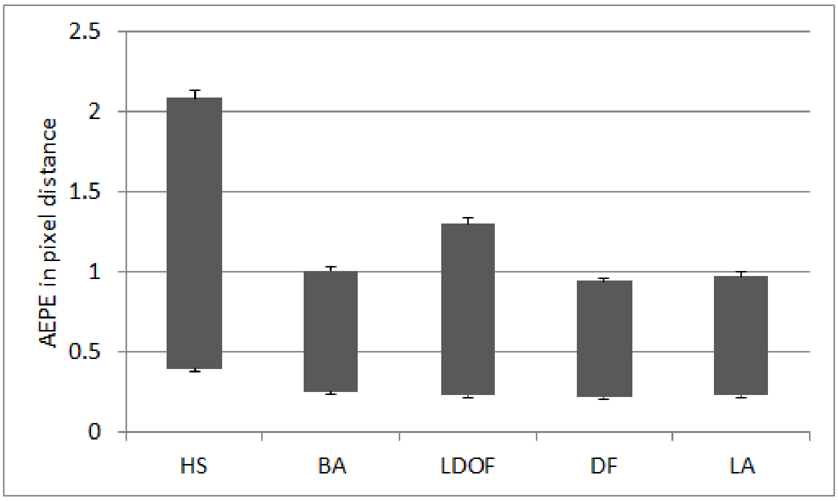


Figure 5.12: The Box and Whisker plot of AEPEs of different registration methods (experiment described in section 5.5.2).

Figure 5.12 shows the Box and Whisker plot of AEPEs of different registration methods. Mauchly's test indicates that the assumption of sphericity has been violated ($\chi^2\,(9) = 165.477$, $p < 0.001$) therefore Greenhouse-Geisser corrected tests are reported ($\varepsilon = .501$). The results show that AEPEs are significantly affected by the choice of method ($V = 0.296$, $F\,(4, 89) = 9.351$, $p < 0.001$.

From the post hoc test we conclude that (see Table 5.20 and Table 5.21):

- The learning algorithm generates significantly lower AEPEs than HS ($p < .05$) , SMF ($p < .05$) and LDOF ($p < .05$).
- The AEPEs of the learning algorithm are not significantly different from those of DeepFlow ($p > .05$).

Table 5.20: Tests of Within-Subjects Effects (experiment described in section 5.5.2)

Measure:MSE

Source		Type III Sum of Squares	df	Mean Square	F	Sig.
Method	Sphericity Assumed	126.029	4	31.507	23.006	5.52E-17**
	Greenhouse-Geisser	126.029	2.003	62.912	23.006	1.18E-09**
	Huynh-Feldt	126.029	2.048	61.541	23.006	8.05E-10**
	Lower-bound	126.029	1.000	126.029	23.006	6.20E-06**
Error (Method)	Sphericity Assumed	503.991	368	1.370		
	Greenhouse-Geisser	503.991	184.301	2.735		
	Huynh-Feldt	503.991	188.407	2.675		
	Lower-bound	503.991	92.000	5.478		

Sig. (* for <.05 and ** for <.01)

Table 5.21: Pairwise Comparisons (experiment described in section 5.5.2)

Measure MSE

	1	2	3	4	5
1		5.70E-07**	1.48E-05**	9.74E-08**	9.53E-08**
2	5.70E-07**		7.53E-01	1.11E-01	4.47E-04**
3	1.48E-05**	7.53E-01		3.22E-02*	7.66E-04**
4	9.74E-08**	1.11E-01	3.22E-02*		9.86E-02
5	9.53E-08**	4.47E-04**	7.66E-04**	9.86E-02	

Sig. (* for <.05 and ** for <.01)

5.5.3 Discussion

These results suggest that not only does our learning approach perform better than HS and SMF, it also delivers performances in line with two very sophisticated techniques, LDOF and DeepFlow, on the basis of initial and learning flows estimated from the humble HS.

Figure 5.13.d and 5.14.d illustrate that the HS has no particular difficulty registering the bamboo stems, the pillars or the barrels since these structures are large enough with respect to the magnitude of their movement for a coarse-to-fine approach to deal with them adequately. However it is not able to correctly register the much smaller bamboo leaves and fast moving chicken. We use as a learning set of consecutive frames different from the test frames (frames #12, #13 and #14 in Market6, frames #1, #2 and #3 in Bamboo2). The accurate flows which HS could compute

between the consecutive frames in the learning set made it possible for the learning algorithm to estimate more accurate flows between non-consecutive frames in the test set. Interestingly, it specifically improves the accuracy of the flow in the regions of interest without degrading the accuracy elsewhere (see Figure 5.13.l and 5.14.l).

Figure 5.13.e-f and 5.14.e-f show the registration results for SMF, Figure 5.13.g-h and 5.14.g-h show the registration results for LDOF, Figure 5.13.i-j and 5.14.i-j show the registration results for DeepFlow applied directly to the test frames. Judged from the super-imposed registered source and target images, the performance of these very sophisticated techniques is actually in line with that of our learning approach applied to the straight-forward HS. As a matter of fact, neither of SMF, LDOF and DeepFlow could adequately handle the running chicken (see Figure 5.14.h,j). Their estimated flows are inevitably more regular though not necessarily better everywhere.

5.6 Conclusion

In this chapter, we introduced two additional similarity measures in the learning algorithm based on spatial distance and intensity. Both of these improved the performance. The use of a second iteration did result in over-regularization of the learned flow field and thus did not improve the performance. We apply the learning algorithm on MPI-Sintel dataset and compared its performance quantitatively with HS, SMF, LDOF and DeepFlow. The learning algorithm shows better performance than both the HS and the SMF. Results suggest that our learning approach applied to the straight-forward HS shows similar performances than both of the very sophisticated techniques, LDOF and DeepFlow.

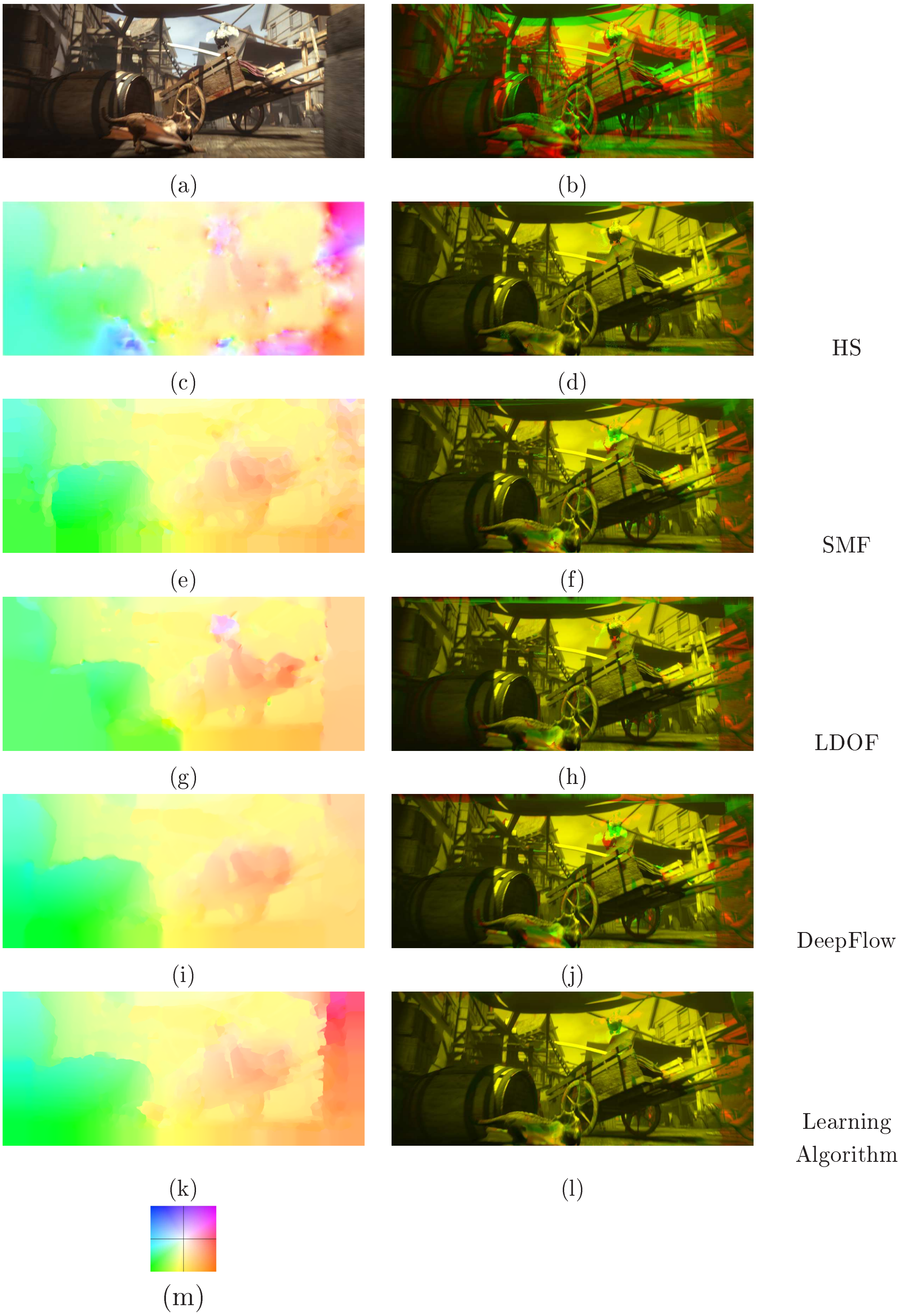


Figure 5.13: MPI-Sintel's Market6 sequence: (a) source frame #14; (b) target frame #16 (in red) super-imposed on top of source frame (in green); (c) flow estimated with HS; (d) target frame (in red) on top of source frame transformed using HS flow (in green); (e-f) idem for learning SMF; (g-h) idem for LDOF; (i-j) idem for DeepFlow; (k-l) idem for learning algorithm; (m) flow field color coding: the color represents the orientation of the vector and brightness stands for its magnitude.

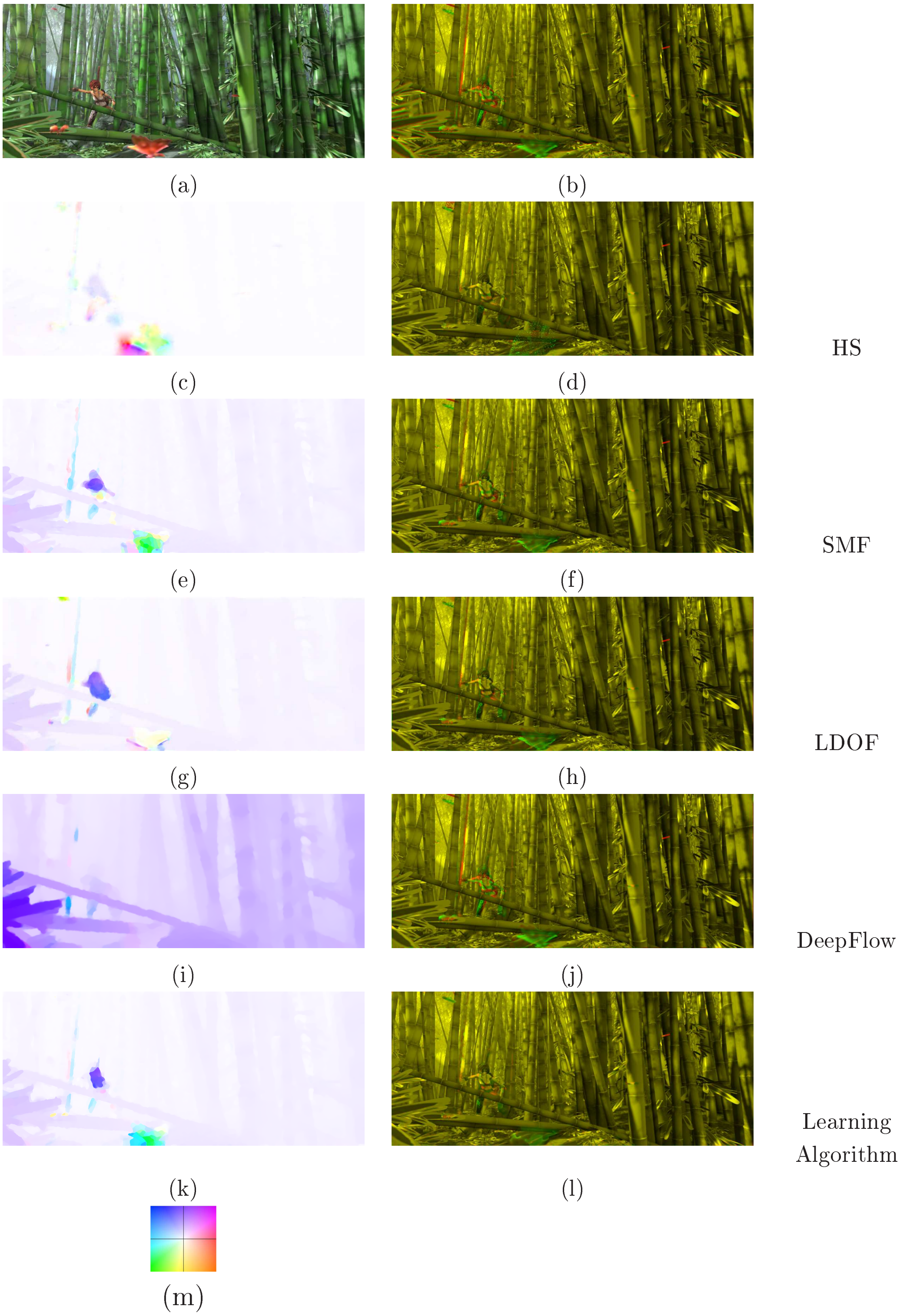


Figure 5.14: MPI-Sintel's Bamboo2 sequence: (a) source frame #3; (b) target frame #5 (in red) super-imposed on top of source frame (in green); (c) flow estimated with HS; (d) target frame (in red) on top of source frame transformed using HS flow (in green); (e-f) idem for the SMF; (g-h) idem for the LDOF; (i-j) idem for the DeepFlow; (k-l) idem for the learning algorithm; (m) flow field color coding: the color represents the orientation of the vector and brightness stands for its magnitude.

Chapter 6

Conclusion

6.1 Summary

Accurately estimating the optical flow of small structures animated with large motion remains a core challenge in computer vision. Most state-of-the-art approaches rely on the additional information provided by image descriptors to improve the flow (e.g. LDOF, DeepFlow). In this thesis, we proposed a different, data-driven, learning approach to motion estimation. We focused on the computation of the optical flow between two non-consecutive images in a sequence on the basis of a learning set of optical flows carefully estimated *a priori* between different consecutive images in the same sequence. Rather than learning a statistical model of the flow, our approach refines an initial estimate of the flow field by replacing each displacement vector by a linear combination of displacement vectors at the centre of similar patches taken from an *a priori*, structured code-book.

We evaluated the use of a variety of similarity measures, of a number of embedding techniques to help with the structuring (clustering) of the code-book, and of numerous refinements (such as vector composition of deformation fields and iterative application of the learning algorithm). Experimental results suggest that with careful selection of the learning set, the proposed approach shows better performance than many advanced method such as SMF, LDOF or DeepFlow.

We also had the opportunity to contribute software to the research community when the need arose for a Matlab implementation of the classic Horn-Schunck and Lucas-Kanade optical flow algorithms for 3D images. Our codes are available at Matlabcentral (mathworks.co.uk/ matlabcentral/ fileexchange/ authors/ 257136) where it attracted maximum rating and is regularly downloaded (116 times in the past 30 days).

6.2 Future Directions

Here is a selection of potential research directions, aimed at improving upon the proposed learning framework.

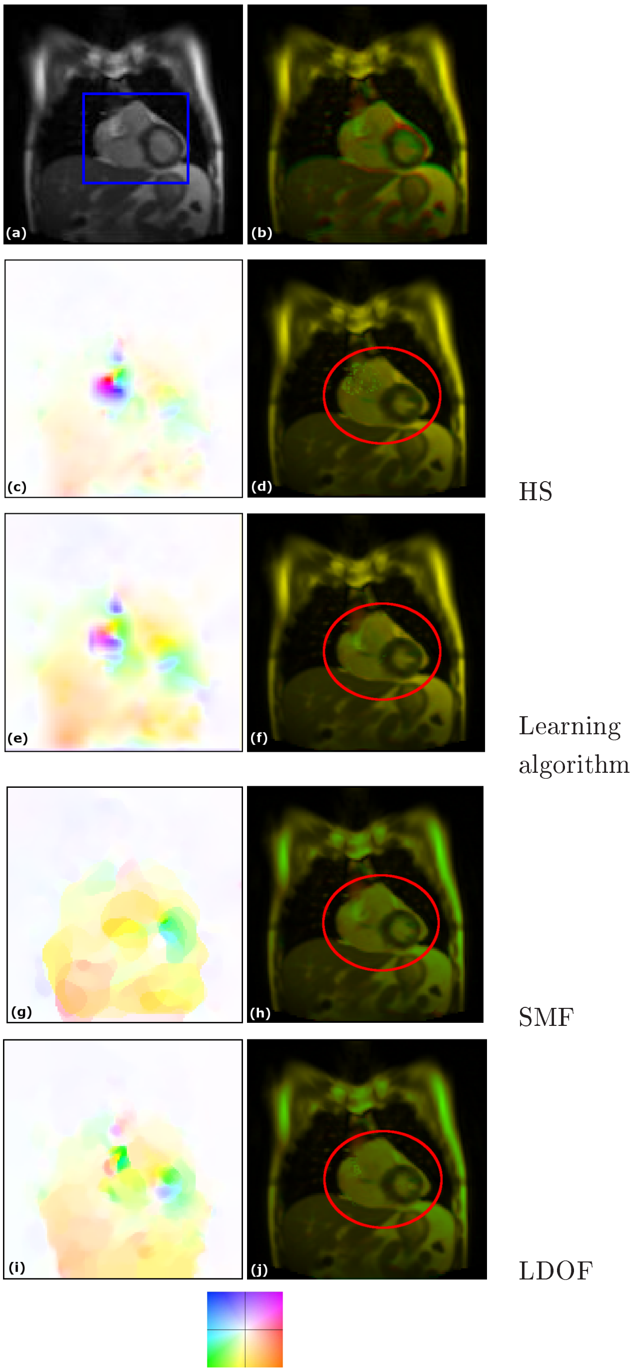
6.2.1 Piece-wise learning set

One of the limitations of our approach is that its performance is highly dependent on the selection of an appropriate learning set. If the learning set does not contain motions similar enough to those in the test case, the proposed algorithm will over-smooth the estimated flow fields. Therefore, it might be preferable to consider a composite, or piece-wise, learning set. For instance, in 2D and 3D images of thorax, if we exclusively include motions around heart and diaphragm in the learning set, the proposed approach may perform better.

We conducted a preliminary experiment, which is illustrated on Figure 6.1. Here, HS is used to estimate the initial flow between two non-consecutive frames, #82 and #85 in the 2D MRI dataset. The learning consists of the flows from from a region around the heart between frames #41 and #44 (see Figure 6.1.a). Whilst HS manages to recover most of the motion between the test frames (both the vertical breathing motion and the heart contractions), it produces a poorly regularized flow (Figure 6.1.c). In contrast, the learning algorithm matches the heart across both frames in a very satisfactory fashion, without introducing errors elsewhere in the frame. The learning algorithm also performs better transformation than both SMF (Figure 6.1.h) and LDOF (Figure 6.1.j).

6.2.2 Similarity measures

We considered a number of similarity measures over the course of the thesis and retained three: angular difference, distance between patches and intensity difference. Many other statistical similarity measures could be considered, such as intensity correlation, mutual information or Earth Mover's distance amongst many others. It would be interesting to conduct a systematic and broad study of their respective merit and how they could be combined. Our preliminary experiment suggests that the SSD of intensity used in the learning algorithm outperforms intensity correlation as a similarity measure.



(k)

Figure 6.1: Dynamic MRI scan: (a) source frame #82; (b) target frame #85 (in red) superimposed on top of source frame (in green); (c) flow estimated with HS; (d) target frame (in red) on top of source frame transformed using HS flow (in green); (e-f) idem for learning algorithm; (g-h) idem for SMF; (i-j) idem for LDOF; (k) flow field color coding: the color represents the orientation of the vector and brightness stands for its magnitude.

6.2.3 Cross-patient optical flow estimation

Whilst we focused on improving the flow within a sequence of images on the basis of a learning set consisting of images form the same sequence, it would of course be extremely interesting to learn from one sequence and estimate flows from another, in particular when dealing with several patients. The motions of several internal organs such as heart, lungs etc. are fundamentally very similar even in different patients. It is very likely that patterns of motion of different organs of one patient may help to improve estimation of those motions of another patient.

References

Alvarez, L., Weickert, J., and Sánchez, J. (2000). Reliable estimation of dense optical flow fields with large displacements. *International Journal of Computer Vision*, 39(1):41–56.

Anandan, P. (1989). A computational framework and an algorithm for the measurement of visual motion. *International Journal of Computer Vision*, 2(3):283–310.

Arbelaez, P., Maire, M., Fowlkes, C., and Malik, J. (2009). From contours to regions: An empirical evaluation. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 2294–2301. IEEE.

Baker, S., Scharstein, D., Lewis, J., Roth, S., Black, M. J., and Szeliski, R. (2011). A database and evaluation methodology for optical flow. *International Journal of Computer Vision*, 92(1):1–31.

Balasubramanian, M. and Schwartz, E. L. (2002). The isomap algorithm and topological stability. *Science*, 295(5552):7–7.

Barron, J., Fleet, D., and Beauchemin, S. (1994). Performance of optical flow techniques. *International Journal of Computer Vision*, 12(1):43–77.

Beauchemin, S. S. and Barron, J. L. (1995). The computation of optical flow. *ACM Comput. Surv.*, 27:433–466.

Belkin, M. and Niyogi, P. (2003). Laplacian eigenmaps for dimensionality reduction and data representation. *Neural computation*, 15(6).

Black, M. J. and Anandan, P. (1996). The robust estimation of multiple motions: Parametric and piecewise-smooth flow fields. *Computer Vision and Image Understanding*, 63(1):75–104.

Black, M. J. and Jepson, A. D. (1996). Estimating optical flow in segmented images using variable-order parametric models with local deformations. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 18(10):972–986.

Black, M. J., Yacoob, Y., Jepson, A. D., and Fleet, D. J. (1997). Learning parameterized models of image motion. In *Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on*, pages 561–567. IEEE.

Box, G. E. P. (1953). Non-normality and tests on variances. *Biometrika*, 40(3/4):pp. 318–335.

Boykov, Y., Veksler, O., and Zabih, R. (2001). Fast approximate energy minimization via graph cuts. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 23(11):1222–1239.

Brox, T., Bruhn, A., Papenberg, N., and Weickert, J. (2004). High accuracy optical flow estimation based on a theory for warping. In *Computer Vision-ECCV 2004*, pages 25–36. Springer Berlin Heidelberg.

Brox, T. and Malik, J. (2011). Large displacement optical flow: descriptor matching in variational motion estimation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(3):500–513.

Bruhn, A., Weickert, J., and Schnorr, C. (2005). Lucas/kanade meets horn/schunck: Combining local and global optic flow methods. *International Journal of Computer Vision*, 61(3):211–231.

Buades, A., Coll, B., and Morel, J. M. (2005). A Review of Image Denoising Algorithms, with a New One. *Multiscale Modeling & Simulation*, 4(2):490–530.

Butler, D. J., Wulff, J., Stanley, G. B., and Black, M. J. (2012). A naturalistic open source movie for optical flow evaluation. In *Computer Vision–ECCV 2012*, pages 611–625. Springer Berlin Heidelberg.

Crum, W. R., Hartkens, T., and Hill, D. (2014). Non-rigid image registration: theory and practice.

Dalal, N. and Triggs, B. (2005). Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886–893. IEEE.

Enkelmann, W. (1988). Investigations of multigrid algorithms for the estimation of optical flow fields in image sequences. *Computer Vision, Graphics, and Image Processing*, 43(2):150 – 177.

Fisher, Ronald A., S. (1922). On the dominance ratio. *Proceedings of the Royal Society of Edinburgh*, 42:321–341.

Fleet, D. and Weiss, Y. (2006). Optical flow estimation. In Paragios, N., Chen, Y., and Faugeras, O., editors, *Handbook of Mathematical Models in Computer Vision*, pages 237–257. Springer US.

Freeman, W. T., Pasztor, E. C., and Carmichael, O. T. (2000). Learning low-level vision. *International journal of computer vision*, 40(1):25–47.

Horn, B. K. and Schunck, B. G. (1980). Determining optical flow. Technical report, Cambridge, MA, USA.

Lempitsky, V., Roth, S., and Rother, C. (2008). Fusionflow: Discrete-continuous optimization for optical flow estimation. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE.

Li, Y. and Huttenlocher, D. P. (2008a). Learning for optical flow using stochastic optimization. In *Computer Vision–ECCV 2008*, pages 379–391. Springer Berlin Heidelberg.

Li, Y. and Huttenlocher, D. P. (2008b). Learning for stereo vision using the structured support vector machine. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE.

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110.

Lucas, B. D. and Kanade, T. (1981). An iterative image registration technique with an application to stereo vision. In *Proceedings of the 7th international joint conference on Artificial intelligence - Volume 2*, pages 674–679, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Mac Aodha, O., Humayun, A., Pollefeys, M., and Brostow, G. J. (2013). Learning a confidence measure for optical flow. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(5):1107–1120.

MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. In *Proc. Fifth Berkeley Sympos. Math. Statist. and Probability (Berkeley, Calif., 1965/66)*, pages Vol. I: Statistics, pp. 281–297. Univ. California Press, Berkeley, Calif.

Maintz, J. B. A. and Viergever, M. A. (1998). A survey of medical image registration. *Medical Image Analysis*, 2:1–36.

Markowski, C. A. and Markowski, E. P. (1990). Conditions for the effectiveness of a preliminary test of variance. *The American Statistician*, 44(4):pp. 322–326.

Michie, D. (1968). Memo functions and machine learning. *Nature*, 218:19–22.

Movshon, J. A., Adelson, E. H., Gizzi, M. S., and Newsome, W. T. (1985). The analysis of moving visual patterns. *Pattern recognition mechanisms*, 54:117–151.

Otte, M. and Nagel, H.-H. (1994). Optical flow estimation: Advances and comparisons. In *Computer Vision - ECCV '94*, volume 800 of *Lecture Notes in Computer Science*, pages 49–60. Springer Berlin Heidelberg.

Pearson, K. (1901). On lines and planes of closest fit to systems of points in space. *Philosophical Magazine Series 6*, 2(11):559–572.

Roosendaal, T. P. (2010). Sintel. blender foundation, durian open movie project. *www.sintel.org*.

Roth, S. and Black, M. J. (2005). Fields of experts: A framework for learning image priors. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 2, pages 860–867. IEEE.

Roth, S. and Black, M. J. (2007a). On the spatial statistics of optical flow. *International Journal of Computer Vision*, 74(1):33–50.

Roth, S. and Black, M. J. (2007b). Steerable random fields. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8. IEEE.

Roweis, S. T. and Saul, L. K. (2000). Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500).

Simoncelli, E. P., Adelson, E. H., and Heeger, D. J. (1991). Probability distributions of optical flow. In *Computer Vision and Pattern Recognition, 1991. Proceedings CVPR'91., IEEE Computer Society Conference on*, pages 310–315. IEEE.

Spall, J. (1992). Multivariate stochastic approximation using a simultaneous perturbation gradient approximation. *Automatic Control, IEEE Transactions on*, 37(3):332–341.

Steinbach, M., Karypis, G., Kumar, V., et al. (2000). A comparison of document clustering techniques. *KDD workshop on text mining*, 400(1).

Steinbrucker, F., Pock, T., and Cremers, D. (2009). Large displacement optical flow computation withoutwarping. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 1609–1614.

Stoll, M., Volz, S., and Bruhn, A. (2013). Adaptive integration of feature matches into variational optical flow methods. In *Asian Confernece on Computer Vision (ACCV)*. Springer Heidelberg.

Sun, D., Roth, S., and Black, M. (2010). Secrets of optical flow estimation and their principles. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 2432–2439.

Sun, D., Roth, S., and Black, M. J. (2013). A quantitative analysis of current practices in optical flow estimation and the principles behind them. Technical Report CS-10-03, Brown University, Department of Computer Science.

Sun, D., Roth, S., Lewis, J., and Black, M. J. (2008). Learning optical flow. In *Computer Vision–ECCV 2008*, pages 83–97. Springer Berlin Heidelberg.

Szeliski, R. (1990). Bayesian modeling of uncertainty in low-level vision. *International Journal of Computer Vision*, 5(3):271–301.

Tenenbaum, J. B., de Silva, V., and Langford, J. C. (2000). A Global Geometric Framework for Nonlinear Dimensionality Reduction. *Science*, 290(5500):2319–2323.

Thorndike, R. (1953). Who belongs in the family? *Psychometrika*, 18(4):267–276.

Trobin, W., Pock, T., Cremers, D., and Bischof, H. (2008). Continuous energy minimization via repeated binary fusion. In *Computer Vision–ECCV 2008*, pages 677–690. Springer Berlin Heidelberg.

Wang, J., Guo, Y., Ying, Y., Liu, Y., and Peng, Q. (2006). Fast non-local algorithm for image denoising. In *Image Processing, 2006 IEEE International Conference on*, pages 1429–1432. IEEE.

Ward, Joe H., J. (1963). Hierarchical grouping to optimize an objective function. *Journal of the American Statistical Association*, 58(301):pp. 236–244.

Wedel, A., Cremers, D., Pock, T., and Bischof, H. (2009a). Structure- and motion-adaptive regularization for high accuracy optic flow. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 1663–1668.

Wedel, A., Pock, T., Braun, J., Franke, U., and Cremers, D. (2008). Duality tv-l1 flow with fundamental matrix prior. In *Image and Vision Computing New Zealand, 2008. IVCNZ 2008. 23rd International Conference*, pages 1–6.

Wedel, A., Pock, T., Zach, C., Bischof, H., and Cremers, D. (2009b). An improved algorithm for tv-la optical flow. In *Statistical and Geometrical Approaches to Visual Motion Analysis*, volume 5604 of *Lecture Notes in Computer Science*, pages 23–45. Springer Berlin Heidelberg.

Weinzaepfel, P., Revaud, J., Harchaoui, Z., and Schmid, C. (2013). DeepFlow: Large displacement optical flow with deep matching. In *IEEE Intenational Conference on Computer Vision (ICCV)*.

Wulff, J., Butler, D., Stanley, G., and Black, M. (2012). Lessons and insights from creating a synthetic optical flow benchmark. In Fusiello, A., Murino, V., and Cucchiara, R., editors, *Computer Vision - ECCV 2012. Workshops and Demonstrations*, volume 7584 of *Lecture Notes in Computer Science*, pages 168–177. Springer Berlin Heidelberg.

Xiao, J., Cheng, H., Sawhney, H., Rao, C., and Isnardi, M. (2006). Bilateral filtering-based optical flow estimation with occlusion detection. In *Computer Vision–ECCV 2006*, pages 211–224. Springer Berlin Heidelberg.

Xu, L., Chen, J., and Jia, J. (2008). A segmentation based variational model for accurate optical flow estimation. In *European Conference on Computer Vision (ECCV), 2008*, volume 1, pages 671–684.

Xu, L., Jia, J., and Matsushita, Y. (2012). Motion detail preserving optical flow estimation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(9):1744–1757.

Xue, Z., Shen, D., and Davatzikos, C. (2006). Statistical representation of high-dimensional deformation fields with application to statistically constrained 3d warping. *Medical Image Analysis*, 10(5):740–751.

Zhang, Z.-y. and Zha, H.-y. (2004). Principal manifolds and nonlinear dimensionality reduction via tangent space alignment. *Journal of Shanghai University*, 8(4):406–424.

Zimmer, H., Bruhn, A., Weickert, J., Valgaerts, L., Salgado, A., Rosenhahn, B., and Seidel, H.-P. (2009). Complementary optic flow. In *Energy minimization methods in computer vision and pattern recognition*, pages 207–220. Springer Berlin Heidelberg.

Zitnick, C., Jojic, N., and Kang, S. B. (2005). Consistent segmentation for optical flow estimation. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, volume 2, pages 1308–1315. IEEE.

Appendix A

Test Cases of 3D Gated CT Dataset

A.1 Test Cases

Table A.1 shows the list of 400 trials; column 1 shows the number of trials, column 2, 3 and 4 show starting point (coordinates X, Y and Z respectively) of test ptaches from 3D gated CT images each of size $512 \times 512 \times 100$ pixels.

List of trials

Trial no.	X	Y	Z	Trial no.	X	Y	Z	Trial no.	X	Y	Z	Trial no.	X	Y	Z
1	1	190	22	101	106	463	22	201	232	400	22	301	358	358	43
2	1	190	43	102	106	463	64	202	232	400	43	302	358	358	64
3	1	190	64	103	127	316	22	203	232	400	64	303	358	379	22
4	1	211	22	104	127	316	43	204	232	421	22	304	358	379	43
5	1	211	43	105	127	316	64	205	232	421	43	305	358	379	64
6	1	211	64	106	127	337	22	206	232	421	64	306	358	400	22
7	1	232	1	107	127	337	43	207	232	442	22	307	358	400	43
8	1	232	22	108	127	337	64	208	232	442	43	308	358	400	64
9	1	232	43	109	127	358	64	209	232	442	64	309	358	421	22
10	1	232	64	110	127	379	22	210	232	463	22	310	358	421	43
11	1	253	1	111	127	379	43	211	232	463	43	311	358	421	64
12	1	253	22	112	127	379	64	212	232	463	64	312	358	442	22
13	1	253	43	113	127	400	22	213	232	484	22	313	358	442	43
14	1	253	64	114	127	400	43	214	232	484	43	314	358	442	64
15	1	274	1	115	127	400	64	215	232	484	64	315	358	463	22
16	1	274	22	116	127	421	22	216	253	379	43	316	358	463	43
17	1	274	43	117	127	421	43	217	253	400	22	317	358	463	64
18	1	274	64	118	127	421	64	218	253	400	43	318	358	484	64
19	1	295	1	119	127	442	22	219	253	400	64	319	379	316	43
20	1	295	22	120	127	442	43	220	253	421	22	320	379	316	64
21	1	295	43	121	127	442	64	221	253	421	43	321	379	337	22
22	1	295	64	122	127	463	22	222	253	421	64	322	379	337	64
23	1	316	43	123	127	463	43	223	253	442	22	323	379	358	22
24	1	316	64	124	127	463	64	224	253	442	43	324	379	358	43
25	22	316	22	125	148	316	64	225	253	442	64	325	379	358	64
26	22	316	43	126	148	337	43	226	253	463	22	326	379	379	22
27	22	316	64	127	148	337	64	227	253	463	43	327	379	379	64
28	22	337	22	128	148	358	22	228	253	463	64	328	379	400	43
29	22	337	43	129	148	358	43	229	253	484	22	329	379	400	64
30	22	337	64	130	148	358	64	230	253	484	43	330	379	421	22
31	22	358	22	131	148	379	43	231	253	484	64	331	379	421	43
32	22	358	43	132	148	379	64	232	274	358	22	332	379	421	64
33	22	358	64	133	148	400	43	233	274	400	22	333	379	442	22
34	22	379	22	134	148	400	64	234	274	400	43	334	379	442	43
35	43	316	22	135	148	421	43	235	274	400	64	335	379	442	64

36	43	316	43		136	148	421	64		236	274	421	22		336	379	463	22	
37	43	316	64		137	148	442	22		237	274	421	64		337	379	463	43	
38	43	337	22		138	148	442	43		238	274	442	22		338	379	463	64	
39	43	337	43		139	148	442	64		239	274	442	43		339	400	316	43	
40	43	337	64		140	148	463	22		240	274	442	64		340	400	337	43	
41	43	358	22		141	148	463	43		241	274	463	22		341	400	337	64	
42	43	358	43		142	148	463	64		242	274	463	43		342	400	358	22	
43	43	358	64		143	148	484	22		243	274	463	64		343	400	358	43	
44	43	379	22		144	148	484	43		244	274	484	22		344	400	358	64	
45	43	379	43		145	148	484	64		245	274	484	43		345	400	379	22	
46	43	379	64		146	169	358	64		246	274	484	64		346	400	379	43	
47	43	400	22		147	169	379	43		247	295	358	64		347	400	379	64	
48	43	400	43		148	169	379	64		248	295	379	64		348	400	400	22	
49	43	400	64		149	169	400	22		249	295	400	22		349	400	400	43	
50	64	316	22		150	169	400	43		250	295	400	43		350	400	400	64	
51	64	316	64		151	169	400	64		251	295	400	64		351	400	421	22	
52	64	337	22		152	169	421	22		252	295	421	22		352	400	421	43	
53	64	337	43		153	169	421	43		253	295	421	43		353	400	421	64	
54	64	337	64		154	169	421	64		254	295	421	64		354	400	442	22	
55	64	358	43		155	169	442	22		255	295	442	22		355	400	442	43	
56	64	358	64		156	169	442	43		256	295	442	43		356	400	442	64	
57	64	379	22		157	169	442	64		257	295	442	64		357	421	316	43	
58	64	379	43		158	169	463	22		258	295	463	22		358	421	316	64	
59	64	379	64		159	169	463	43		259	295	463	43		359	421	337	64	
60	64	400	22		160	169	463	64		260	295	463	64		360	421	358	43	
61	64	400	43		161	169	484	22		261	295	484	22		361	421	358	64	
62	64	400	64		162	169	484	43		262	295	484	43		362	421	379	22	
63	64	421	22		163	169	484	64		263	295	484	64		363	421	379	43	
64	64	421	43		164	190	358	64		264	316	358	64		364	421	379	64	
65	64	421	64		165	190	379	43		265	316	379	64		365	421	400	22	
66	85	316	22		166	190	400	22		266	316	400	43		366	421	400	43	
67	85	316	43		167	190	400	43		267	316	421	22		367	421	400	64	
68	85	316	64		168	190	400	64		268	316	421	43		368	421	421	22	
69	85	337	64		169	190	421	22		269	316	421	64		369	421	421	43	
70	85	358	43		170	190	421	43		270	316	442	22		370	421	421	64	
71	85	358	64		171	190	421	64		271	316	442	43		371	442	316	22	
72	85	379	43		172	190	442	22		272	316	442	64		372	442	316	43	
73	85	379	64		173	190	442	43		273	316	463	22		373	442	316	64	
74	85	400	22		174	190	442	64		274	316	463	43		374	442	337	43	
75	85	400	43		175	190	463	22		275	316	463	64		375	442	337	64	
76	85	400	64		176	190	463	43		276	316	484	22		376	442	358	22	
77	85	421	22		177	190	463	64		277	316	484	43		377	442	358	43	
78	85	421	43		178	190	484	22		278	316	484	64		378	442	358	64	
79	85	421	64		179	190	484	43		279	337	337	64		379	442	379	22	
80	85	442	22		180	190	484	64		280	337	358	64		380	442	379	43	
81	85	442	43		181	211	316	64		281	337	379	22		381	442	379	64	
82	85	442	64		182	211	358	64		282	337	379	64		382	442	400	22	
83	106	316	22		183	211	379	22		283	337	400	22		383	442	400	43	
84	106	316	43		184	211	379	43		284	337	400	43		384	442	400	64	
85	106	316	64		185	211	400	43		285	337	400	64		385	463	316	43	
86	106	337	43		186	211	400	64		286	337	421	22		386	463	316	64	
87	106	337	64		187	211	421	22		287	337	421	43		387	463	337	43	
88	106	358	43		188	211	421	43		288	337	421	64		388	463	337	64	
89	106	358	64		189	211	421	64		289	337	442	22		389	463	358	22	
90	106	379	22		190	211	442	22		290	337	442	43		390	463	358	43	
91	106	379	43		191	211	442	43		291	337	442	64		391	463	358	64	
92	106	379	64		192	211	442	64		292	337	463	22		392	463	379	22	
93	106	400	22		193	211	463	22		293	337	463	43		393	463	379	43	
94	106	400	43		194	211	463	43		294	337	463	64		394	463	379	64	
95	106	400	64		195	211	463	64		295	337	484	22		395	484	316	22	
96	106	421	22		196	211	484	22		296	337	484	43		396	484	316	43	
97	106	421	43		197	211	484	43		297	337	484	64		397	484	316	64	
98	106	421	64		198	211	484	64		298	358	316	64		398	484	337	22	
99	106	442	43		199	232	379	22		299	358	337	64		399	484	337	64	
100	106	442	64		200	232	379	64		300	358	358	22		400	484	358	22	

Appendix B

Trials of Different Similarity Measures for 2D MRI Dataset

B.1 Different Similarity Measures

MSEs of Images Registered with Learning Algorithm Using Different Similarity Measures (experiment described in section 3.4.4)

Trial no.	LK	Degraded LK	Learning approach with vector difference	Learning approach with angular difference	Learning approach with magnitude difference
1	93.55	145.52	195.46	159.54	199.01
2	206.17	289.01	337.81	252.30	383.09
3	136.65	193.89	216.85	174.74	277.18
4	100.84	143.45	189.09	158.42	226.18
5	90.89	133.58	192.87	138.53	215.53
6	153.87	223.29	273.68	210.43	256.45
7	96.30	139.58	177.59	159.24	224.91
8	151.23	221.61	260.17	220.94	252.07
9	146.99	209.90	300.74	193.23	215.68
10	106.67	144.48	189.75	118.77	201.43
11	100.85	128.76	212.43	116.23	188.15
12	115.93	166.25	251.87	190.78	250.80
13	107.85	141.88	216.58	115.91	222.81
14	90.83	128.97	141.50	135.58	142.03
15	109.11	158.78	189.56	110.13	222.45
16	39.66	53.29	96.91	73.69	129.52
17	120.54	153.63	219.51	172.10	222.49
18	154.17	233.98	248.37	192.52	312.80
19	128.31	193.54	259.91	163.06	259.48
20	82.14	113.10	126.61	83.18	148.26
21	82.14	113.10	174.42	138.68	162.57
22	101.25	147.03	193.29	104.62	156.26

23	101.60	134.88	204.37	105.82	226.53
24	184.06	273.92	340.15	233.10	312.72
25	82.14	113.10	204.00	82.54	148.16
26	101.25	147.03	229.51	162.44	189.13
27	112.29	171.59	318.38	182.26	307.17
28	112.78	165.85	231.41	163.42	212.68
29	101.75	145.75	163.20	110.92	182.09
30	92.25	125.34	134.12	149.90	223.84

Appendix C

Trials of Different Similarity Measures for 3D Gated CT Dataset

C.1 Different Similarity Measures

MSEs of Registered Images Using Learning Algorithm with Different Similarity Measures

Trial no.	LK	Degraded LK	Vector Difference	Angular Difference	Magnitude Difference		Trial no.	LK	Degraded LK	Vector Difference	Angular Difference	Magnitude Difference
1	51.78	51.87	68.09	61.69	66.49		201	401.28	636.92	330.06	326.82	339.22
2	32.89	33.06	51.3	35.36	47.55		202	79.63	78.25	83.86	96.26	100.13
3	42.79	42.23	61.18	43.34	51.85		203	42.35	42.15	58.27	52.27	54.17
4	88.98	89.91	115.47	104.17	104.07		204	127.95	127.24	146.75	137.01	141.5
5	40.68	41.00	47.45	46.04	51.64		205	67.51	67.47	74.6	75.84	80.41
6	55.56	54.10	71.39	62.59	59.23		206	42.67	42.14	58.72	50.1	57
7	85.23	82.26	94.91	94.96	106.7		207	87.59	87.07	104.2	106.89	102.41
8	82.61	80.91	90.89	90.35	91.11		208	53.63	53.49	71.45	71.35	57.12
9	42.32	41.81	49.48	51.73	63.66		209	32.87	32.45	45.51	42.95	48.68
10	60.79	61.13	66.94	73.72	77.89		210	73.90	73.61	96.67	94.55	90.08
11	75.71	74.70	88.35	80.6	96.23		211	54.00	53.95	66.02	64.72	65.57
12	89.72	87.70	102.38	107.19	109.28		212	35.12	33.79	44.55	43.88	39.22
13	39.08	39.44	58	48.92	47.56		213	56.94	56.87	70.09	63.04	68.32
14	51.79	51.87	70.47	63.35	62.89		214	41.91	42.68	50.88	47.82	50.68
15	93.04	93.63	121.92	102.94	118.98		215	24.88	24.86	43.29	39.45	30.29
16	102.30	100.55	121.66	112.5	132.41		216	513.30	886.18	173.51	173.13	180.52
17	41.53	41.88	58.85	55.2	53.52		217	857.31	1238.20	309.55	309.22	317.36
18	59.79	58.93	80.89	66.73	78.51		218	86.59	87.01	95.15	97.97	89.61
19	72.64	72.92	94.01	90.52	86.22		219	48.98	48.64	70.33	63.64	64.43
20	88.52	86.83	103.46	110.05	108.38		220	111.92	112.94	126.39	124.1	134.14
21	43.21	43.22	63.08	55.99	53.6		221	54.80	54.64	60.18	59.16	60.91
22	65.41	64.22	72.26	80.11	84.98		222	45.18	44.66	64.53	57.1	65.64
23	30.69	30.61	40.74	48.56	42.09		223	92.46	91.41	114.89	103.62	110.79
24	33.60	34.01	52.78	46.81	37.1		224	50.17	49.57	59.59	51.56	60.08
25	97.50	95.54	109.29	111.9	111.5		225	35.09	35.63	42.71	48.95	42.75
26	60.11	59.22	68.54	77.4	76.1		226	82.42	81.16	95.99	98.96	106.45
27	65.92	64.44	73.48	68.98	73.67		227	38.40	37.71	50.42	43	42.72
28	93.74	89.67	123.13	106.72	114.37		228	36.34	36.23	45.89	44.12	56.05
29	66.03	65.10	76.19	87.97	84.35		229	66.54	65.57	86.65	83.79	76.35
30	59.38	57.41	70.36	75.99	80.48		230	41.27	41.36	46.1	52.41	50.27
31	110.41	108.75	133.82	119.73	123.49		231	28.99	29.27	46.02	31.61	43.16
32	57.57	57.21	72.04	72.05	70.78		232	4144.00	5467.10	1983.2	1778.28	2053.82
33	44.97	44.58	60.54	54.92	65.49		233	316.89	336.40	305.2	293.19	311.26
34	13.77	14.00	25.16	20.43	27.27		234	102.93	113.99	101.54	90.88	100.86
35	129.92	129.40	155.24	141.28	148.78		235	58.42	58.45	66.37	68.31	79.72
36	70.24	70.78	81.29	83.97	94.35		236	94.17	92.20	117.36	104.23	119.93

37	76.86	74.40	93.24	81.92	89.55		237	45.11	44.81	58.43	57.75	62.11
38	140.83	134.86	157.67	160.61	163.58		238	87.76	85.79	93.56	88.08	98.7
39	74.72	73.99	98.45	86.01	83.2		239	43.15	43.31	44.25	44.99	48.55
40	56.16	56.12	67.05	74.79	67.19		240	52.81	52.59	71.29	71.31	67.3
41	126.67	124.03	158.6	145.34	153.31		241	63.57	64.05	79.03	80.76	69.94
42	79.04	77.25	88.57	95.22	89.25		242	41.19	40.72	51.14	43.36	59.18
43	66.17	63.65	90.5	83.98	86.41		243	32.92	32.39	38.77	44.67	50.02
44	108.06	103.11	131.82	119.79	119.8		244	58.51	57.62	74.77	76.14	78.32
45	43.95	43.83	46.95	58.49	63.68		245	34.00	33.93	39.09	38.73	50.48
46	43.70	42.35	48.99	63.01	64.24		246	29.63	29.51	46.28	45.74	41.74
47	63.86	63.57	71.42	76.69	78.15		247	1587.80	3113.90	547.9	536.91	551.03
48	30.73	30.64	45.56	47.39	42.24		248	187.29	893.97	144.84	145.75	153.28
49	23.70	23.34	36.41	37.06	35.86		249	454.13	848.79	469.15	429.99	473.57
50	133.19	129.16	147.09	149.26	155.73		250	94.12	122.74	91.29	83.3	94.86
51	72.59	67.25	94.21	93.28	84.8		251	70.13	69.64	90.91	87.18	89.64
52	149.82	144.35	170.47	169.34	180.37		252	109.62	108.05	119.24	124.48	132.55
53	81.31	81.26	89.34	101.41	92		253	77.57	77.41	84.65	85.17	93.61
54	59.82	60.17	65.72	73.68	80.97		254	48.77	48.01	66.6	61.91	66.97
55	72.94	71.87	81.93	83.29	80.42		255	86.82	85.46	108.8	95.71	105.79
56	71.75	68.10	100.64	90.71	92.31		256	50.30	50.03	68.4	61.97	70.14
57	128.65	126.12	132.05	151.86	143.74		257	36.55	36.09	47.33	53.61	39.47
58	44.72	44.69	52.39	48.16	61.08		258	52.61	51.80	63.98	69.27	74.51
59	54.80	54.68	65.35	61.04	63.87		259	37.32	37.17	51.77	44.48	42.76
60	134.30	130.77	150.95	153.66	166.16		260	30.51	30.05	48.53	43.34	43.07
61	46.57	46.90	66.37	58.54	61.25		261	48.03	47.07	63.88	55.78	59.34
62	39.71	39.96	55.48	42.38	48.01		262	29.85	29.75	41.16	47.9	34.25
63	70.62	69.01	95.04	86.15	94.99		263	36.17	35.28	45.56	40.06	53.38
64	37.64	38.21	52.73	53.49	58.77		264	765.44	2593.70	218.7	202.01	217.7
65	32.27	32.25	43.28	34.47	50.29		265	242.50	1016.10	138.12	139.64	146.11
66	172.73	170.31	244.86	187.44	265.19		266	89.38	97.77	88.01	77.07	90.11
67	90.56	90.63	106.85	106.93	115.57		267	113.47	111.80	139.11	129.3	138.53
68	74.34	76.14	84.67	95.53	89.05		268	70.16	70.28	89.15	86.18	86.14
69	74.26	72.24	82.57	87.48	82.17		269	40.34	39.25	61.27	56.03	62.62
70	86.29	86.46	98.1	98.41	112.22		270	95.43	95.04	118.76	100.11	117.67
71	78.25	76.02	105.51	86.68	90.19		271	58.69	58.75	77.99	67.94	72.45
72	68.89	69.01	98.08	88.31	88.1		272	35.79	35.61	38.79	43.32	52.91
73	66.79	67.44	75.66	85.92	82.35		273	65.89	65.27	71.46	76.42	80.52
74	86.83	88.03	98.41	106.42	108.99		274	43.12	42.57	47.55	54.83	61.38
75	50.81	51.41	67.67	68.71	57.92		275	31.97	31.69	39.26	35.9	40.3
76	40.09	39.66	49.24	46.35	59.51		276	51.49	50.85	55.29	68.22	72.62
77	96.43	94.17	118.66	111.8	110.08		277	34.80	34.47	44.5	47.01	42.82
78	43.70	43.18	55.51	59.16	50.67		278	24.65	24.58	32.01	27.83	27.54
79	35.59	35.01	46.58	39.29	48.18		279	540.89	428.74	619.76	556.36	583.03
80	62.27	62.50	72.22	78.68	82.42		280	436.53	1206.80	178.03	170.9	177.4
81	26.95	26.71	42.37	42.95	32.42		281	472.66	1530.60	361.89	356.85	375.81
82	18.31	18.38	35.57	22.33	23.4		282	131.94	251.03	122.52	123.15	122.66
83	231.27	279.15	234.1	211.87	238.91		283	156.59	156.46	180.35	166.61	180.69
84	324.01	407.17	212.12	203.59	216.02		284	88.45	86.36	106.01	100.47	110.08
85	271.61	318.32	99.66	94.45	89.56		285	62.71	62.11	70.74	74.94	76.79
86	152.17	162.12	150.52	147.49	164.12		286	118.12	117.71	135.49	125.4	141.7
87	84.00	85.21	98.18	92.53	105.8		287	69.26	70.17	89.42	76.32	91.57
88	112.62	113.49	151.71	124.34	143.86		288	36.28	36.61	47.72	55.1	42.16
89	73.38	72.51	83.26	78.05	79.2		289	81.86	81.29	98.73	99.24	100.68
90	139.06	136.15	154.81	148.57	166.36		290	55.70	55.44	68.25	70.08	62.83
91	72.56	72.83	94.71	90.92	94.18		291	33.33	34.02	43.65	50.4	43
92	54.58	54.22	60.43	69.3	62.61		292	69.54	67.28	86.7	78.05	83.97
93	118.80	119.92	136.54	134.74	133.99		293	42.57	42.30	61.08	51.39	59.78
94	63.37	64.49	77.57	80.35	80.39		294	23.07	23.20	26.39	34.44	43.22
95	49.32	49.12	60	55	67.77		295	35.99	35.36	47.91	56.3	55.49
96	105.68	104.35	131.41	124.94	132.86		296	18.72	18.80	27.69	30.14	32.35
97	41.80	42.09	58.92	55.49	45.68		297	15.99	16.17	27.96	23.13	34.89
98	36.52	35.98	50.03	45.02	53.51		298	178.87	178.73	204.25	192.53	200.77
99	41.18	40.37	55.65	50.33	62.19		299	175.88	178.61	197.7	194.65	199.35
100	31.33	31.73	47.74	41.09	49.29		300	210.43	210.09	234.21	226.65	233.6
101	27.54	27.38	33.17	42.91	31.12		301	376.91	410.02	319.41	293.25	317.83
102	7.77	7.88	19.11	21.42	14.66		302	373.82	1336.30	125.67	127.64	123.89
103	267.45	267.65	292.39	278.91	298.2		303	853.49	1778.80	272.5	246.89	264.84
104	115.52	114.68	130.58	120.26	131.39		304	6978.70	12552.00	6501.92	6194.36	6501.94
105	840.16	1450.60	715.83	642.48	675.28		305	115.10	115.47	135.2	123.55	135.9
106	778.31	1518.40	429.87	450.8	448.88		306	116.17	115.85	131.95	125.1	134.65
107	547.22	735.77	279.81	266.33	285.25		307	102.02	100.11	117.06	122.08	129.92
108	631.96	1570.30	104.03	98.43	113.39		308	60.70	59.25	69.02	69.16	71.5
109	71.90	76.03	93.99	89.45	96.84		309	91.67	90.59	103.99	115.3	109.1
110	153.31	148.83	178.39	167.85	172.02		310	66.04	64.84	86.37	80.19	81.42
111	80.27	78.04	93.6	96.86	86.24		311	39.96	39.55	47.58	48.89	52.18
112	53.21	54.22	76.9	72.78	72.31		312	95.79	94.32	121.3	106.53	111.26
113	150.50	150.41	155.34	164.8	174.18		313	60.66	60.89	77.98	66.68	76.07
114	60.23	60.64	65.6	73.52	67.91		314	31.67	31.59	48.46	45.04	48.68
115	37.33	36.89	47	52.2	43.95		315	69.98	69.01	96.65	77.92	81.98
116	112.28	111.70	122.54	133.13	134.92		316	47.42	47.67	53.07	60.25	58.35
117	47.52	47.42	57.26	50.27	60.08		317	21.65	21.30	30.78	33.17	25.17
118	35.27	35.02	44.63	43.42	54.48		318	1.51	1.69	19.44	14.4	11.93
119	102.65	102.59	125.6	108.68	128.53		319	118.41	119.74	136.88	137.75	135.88

120	49.88	49.70	55.91	66.83	57.97		320	258.16	221.37	252.71	232.35	252.17
121	31.79	32.03	40.6	40.93	37.89		321	317.89	350.14	320.95	302.68	315.62
122	62.29	61.95	70.92	71.87	68.24		322	550.76	2080.50	102.3	112.22	112.91
123	41.93	41.78	51.76	48.98	57.31		323	1078.90	2094.20	393.29	379.38	392.22
124	25.12	25.06	30.2	36.37	42.19		324	4707.40	5967.20	3262.63	1968.44	2851.97
125	185.20	186.99	209.86	195.7	209.24		325	100.01	118.53	96.86	85.8	100.1
126	373.72	300.85	253.88	239.79	251.66		326	123.56	122.08	143.38	135.38	136.55
127	136.21	170.00	141.99	134.92	157.36		327	82.89	83.67	101.13	91.11	97.23
128	401.80	812.35	355.36	338.99	370.27		328	95.77	95.63	115.8	109.23	110.17
129	196.91	236.89	173.73	175.22	178.31		329	69.64	68.79	82.43	76.72	92.32
130	473.21	1453.00	198.28	183.11	192.29		330	98.20	96.68	114.85	110.37	115.24
131	251.65	600.16	239.88	225.91	239.71		331	75.79	76.20	91.81	92.85	87.59
132	66.67	65.54	72.68	84.63	82.97		332	37.60	37.92	40.55	48.05	47.43
133	69.49	69.53	89.17	76.08	86.81		333	63.36	63.71	82.25	83.21	86.56
134	40.28	40.22	45.7	44.37	47.22		334	47.18	48.44	69.06	52.91	64.78
135	50.30	51.02	69.15	62.34	61.47		335	30.88	30.80	34.93	43.94	46.12
136	33.94	33.86	50.01	48.85	38.24		336	51.15	51.37	62.54	56.58	63.25
137	90.97	90.22	110.65	96.63	111.6		337	31.75	32.12	50.16	46.38	40.85
138	47.21	47.07	53.1	49.45	56.86		338	22.36	22.40	28.41	34.24	40.84
139	32.82	32.42	43.46	35.68	39.74		339	368.95	547.41	273.31	275.76	278.56
140	79.24	80.44	93.32	86.1	87.39		340	137.26	174.75	149.87	130.78	145.75
141	40.86	40.57	47.82	55.33	59.25		341	106.52	105.64	115.55	125.8	120.35
142	26.88	26.84	38.29	35.25	38.21		342	149.55	148.16	177.36	163.91	164.1
143	22.48	22.28	35.07	34.36	33.9		343	247.12	1275.30	142.92	147.47	152.17
144	15.51	15.61	20.44	27.65	32.5		344	106.36	104.82	121.03	115.8	130.6
145	13.13	13.01	23.23	22.92	26.87		345	118.19	114.65	140.2	126.46	148.83
146	247.94	364.91	152.34	138.83	146.58		346	102.67	116.92	124.32	122.83	129.31
147	1082.00	4112.70	586.3	524.31	538.21		347	73.20	71.55	95.36	83.23	86.86
148	53.95	54.62	71.45	69.3	70.21		348	89.71	89.47	116.82	108.26	113.63
149	146.65	147.38	165.3	167.15	176.33		349	100.31	98.91	117.7	116.33	115.2
150	88.71	88.37	94.12	98.05	104.33		350	52.82	53.51	60.94	72.9	64.87
151	40.48	40.81	48.11	44	57.62		351	99.15	97.46	123.55	116.17	128.83
152	111.03	112.96	134.09	116	140.37		352	59.99	60.22	69.19	80.21	76.13
153	70.75	71.73	83.72	86.88	94.65		353	36.05	35.61	49.67	40.91	51.97
154	36.23	35.77	41.74	43.97	44.52		354	76.15	76.78	101.79	90.65	101.83
155	94.70	92.97	116.05	113.82	108.13		355	46.63	46.05	49.76	58.8	52.7
156	50.17	48.97	60.97	63.48	63.33		356	31.77	31.64	51.03	33.56	50.11
157	35.97	35.62	52.45	46.67	39.47		357	68.08	67.93	71.92	80.98	73.32
158	78.07	77.83	97.99	97.2	97.1		358	96.80	95.31	116.63	104.67	115.43
159	44.82	44.85	61.68	54.42	64.45		359	86.84	86.98	108.68	104.16	107.75
160	29.83	29.61	46.29	39.36	35.98		360	71.38	71.44	77.94	80.09	90
161	55.77	56.06	60.41	66.18	64.83		361	66.34	66.56	80.06	73.07	74.05
162	36.90	36.63	41.52	53.84	47.21		362	98.63	97.19	121.79	119.68	114.28
163	24.07	24.22	40.8	31.39	40.98		363	71.35	73.45	82.95	87.07	87.84
164	861.97	2765.10	181.78	164.88	223.83		364	75.64	74.81	99.58	89.11	95.16
165	1264.50	5293.10	726.74	703.42	740.3		365	76.09	76.13	90.49	80.79	97.9
166	155.47	158.75	181.25	171.73	173.84		366	85.74	83.88	111.42	105.28	110.15
167	145.27	141.29	173.56	171.6	180.86		367	52.71	52.93	74	69.27	74.59
168	44.40	44.60	59.17	63.6	62.67		368	66.44	65.55	72.8	70.3	81.34
169	100.61	100.28	119.37	110.99	123.03		369	65.74	65.13	78.32	72.64	91.46
170	94.60	97.14	119.55	113.99	123.1		370	43.13	43.03	56.47	60.26	47.11
171	38.48	39.24	57.82	50.72	51.51		371	222.86	221.87	250.54	240.43	252.13
172	92.99	90.42	111.42	110.26	121.62		372	56.79	56.52	78.96	59.59	66.62
173	66.36	66.22	83.57	87.27	90.76		373	63.22	62.76	82.08	68.41	78.03
174	36.06	35.94	52.16	46.06	54.67		374	55.14	55.60	71.05	60.58	73.54
175	75.59	74.22	87.09	89.38	88.61		375	56.66	56.45	68.39	75.71	67.23
176	57.81	55.21	86.03	80.65	84.56		376	99.11	104.78	117.38	118.78	126.49
177	31.27	31.05	40.23	37.6	51.3		377	52.11	52.17	71.49	63.83	62.08
178	63.52	63.51	84.87	69.05	80.23		378	63.84	62.58	78.04	76.78	87.08
179	45.40	44.78	56.75	65.13	61.47		379	100.83	100.94	123.43	105.68	111.56
180	27.29	27.21	31.48	34.42	32.89		380	59.02	57.92	64.77	75.49	65.18
181	6684.00	8097.40	5992.02	4705.08	5637.18		381	69.90	69.20	90.58	77.08	93.08
182	1771.50	5466.70	160.86	164.17	193.04		382	83.84	82.03	91.7	94.81	94.27
183	4186.00	3743.10	3508.97	3189.26	3460.83		383	58.44	58.10	78.73	71.32	65.52
184	720.51	3116.10	315.86	312.34	318.56		384	57.98	57.71	82.55	73.37	70.84
185	102.88	101.09	120.72	123.89	120.98		385	36.15	35.81	48.8	44.13	40.91
186	44.20	44.10	53.33	62.11	61.44		386	73.60	70.94	84.52	89.92	91.13
187	104.59	104.82	128.6	122.5	121.72		387	52.67	52.83	60.25	71.34	69.65
188	67.71	66.84	82.56	73.36	85.27		388	62.91	63.29	77.84	76.2	81.5
189	39.77	39.96	49.94	49.38	56.05		389	90.59	89.39	115.31	102.67	115.48
190	93.80	92.42	105.61	115.99	105.36		390	57.30	57.75	64.14	64.32	76.64
191	63.04	62.31	71.21	72.45	77.6		391	51.87	52.42	70.26	62.97	65.71
192	31.77	31.63	40.05	47.56	40.64		392	69.92	69.97	93.69	74.11	90.13
193	73.53	73.82	97.25	79.43	87.04		393	38.77	38.61	47.8	54.16	60.25
194	50.05	49.01	59.48	69.83	66.09		394	55.82	54.60	71.79	67.55	69.51
195	29.23	29.17	36.66	45.71	46.83		395	108.04	105.70	124.83	119.85	122.65
196	69.54	70.17	81.09	80.99	78.11		396	37.42	37.72	45.08	48.09	55.5
197	38.88	38.60	54.14	47.69	54.27		397	62.71	62.91	82.6	66.99	72.67
198	31.46	31.51	35.78	37.64	49.84		398	75.36	72.66	91.54	90.75	84
199	5899.20	5570.60	4975.04	3872.25	4329.69		399	34.17	34.21	41.44	42.23	44.95
200	63.97	70.77	66.18	62.11	70.61		400	7.03	6.61	18.27	16.74	18.94

Appendix D

Trials for Learning Algorithm With and Without Vector Composition for 2D MRI Dataset

D.1 Learning Algorithm With and Without Vector Composition

MSEs of Images Registered Using Learning Algorithm With and Without Vector Composition (experiment described in section 3.5.1)

Trial no.	LK	Degraded LK	Without Vector Composition	With Vector Composition
1	93.55	145.52	159.54	150.34
2	206.17	289.01	252.3	251.77
3	136.65	193.89	174.74	167.36
4	100.84	143.45	158.42	155.73
5	90.89	133.58	138.53	134.3
6	153.87	223.29	210.43	204.95
7	96.30	139.58	159.24	149.81
8	151.23	221.61	220.94	216.76
9	146.99	209.90	193.23	183.4
10	106.67	144.48	118.77	115.76
11	100.85	128.76	116.23	109.22
12	115.93	166.25	190.78	184.12
13	107.85	141.88	115.91	110.52
14	90.83	128.97	135.58	128.6
15	109.11	158.78	110.13	109.46
16	39.66	53.29	73.69	71.91
17	120.54	153.63	172.1	170.82
18	154.17	233.98	192.52	182.53
19	128.31	193.54	163.06	161.35

20	82.14	113.10	83.18	82.85
21	82.14	113.10	138.68	133.07
22	101.25	147.03	104.62	103.89
23	101.60	134.88	105.82	103.13
24	184.06	273.92	233.1	231.2
25	82.14	113.10	82.54	85.85
26	101.25	147.03	162.44	157.83
27	112.29	171.59	182.26	172.44
28	112.78	165.85	163.42	161.86
29	101.75	145.75	110.92	102.36
30	92.25	125.34	149.9	143.45

Appendix E

Learning Algorithm with Vector Composition with 3D Gated CT Dataset

E.1 Using Learning Algorithm with Vector Composition

MSEs of Registered Images Using Learning Algorithm without Vector Composition and with Vector Composition

Trial no.	LK	Degraded LK	Wtihout Vector compo-sition	Wtih Vector compo-sition		Trial no.	LK	Degraded LK	Wtihout Vector compo-sition	Wtih Vector compo-sition
1	51.778	51.87	61.69	58.78		201	401.28	636.92	326.82	325.21
2	32.894	33.055	35.36	33.94		202	79.626	78.25	96.26	92.81
3	42.785	42.226	43.34	40.22		203	42.35	42.151	52.27	49.81
4	88.983	89.909	104.17	102.42		204	127.95	127.24	137.01	133.05
5	40.676	40.997	46.04	44.29		205	67.506	67.466	75.84	75.03
6	55.56	54.098	62.59	62.39		206	42.666	42.137	50.1	46.79
7	85.23	82.255	94.96	94.76		207	87.59	87.066	106.89	104.19
8	82.606	80.914	90.35	89.99		208	53.625	53.492	71.35	70.35
9	42.315	41.814	51.73	49.35		209	32.867	32.446	42.95	41.05
10	60.793	61.131	73.72	72.76		210	73.902	73.608	94.55	92.95
11	75.714	74.699	80.6	77.23		211	53.998	53.951	64.72	62.32
12	89.72	87.704	107.19	103.76		212	35.115	33.792	43.88	40.68
13	39.078	39.437	48.92	45.07		213	56.942	56.87	63.04	62.62
14	51.789	51.867	63.35	61.39		214	41.907	42.677	47.82	44.53
15	93.042	93.628	102.94	102.06		215	24.88	24.863	39.45	36.09
16	102.3	100.55	112.5	111.6		216	513.3	886.18	173.13	171.71
17	41.534	41.884	55.2	53.05		217	857.31	1238.2	309.22	307.5
18	59.787	58.925	66.73	63.68		218	86.586	87.005	97.97	95.68
19	72.637	72.918	90.52	89.13		219	48.975	48.637	63.64	60.84

20	88.52	86.825	110.05	108.21		220	111.92	112.94	124.1	121.13
21	43.214	43.223	55.99	53.43		221	54.803	54.638	59.16	56.13
22	65.407	64.216	80.11	76.44		222	45.177	44.656	57.1	55.54
23	30.689	30.611	48.56	47.91		223	92.457	91.414	103.62	101.9
24	33.598	34.01	46.81	43.95		224	50.168	49.567	51.56	47.73
25	97.503	95.536	111.9	109.59		225	35.089	35.631	48.95	46.66
26	60.106	59.218	77.4	75.67		226	82.421	81.16	98.96	95.56
27	65.916	64.436	68.98	65.44		227	38.401	37.712	43	41.89
28	93.74	89.671	106.72	105.15		228	36.336	36.233	44.12	41.63
29	66.026	65.098	87.97	87.25		229	66.537	65.567	83.79	81.44
30	59.38	57.409	75.99	73.46		230	41.274	41.356	52.41	48.56
31	110.41	108.75	119.73	117.23		231	28.986	29.269	31.61	31.27
32	57.572	57.206	72.05	70.74		232	4144	5467.1	1778.28	1776.28
33	44.969	44.576	54.92	51.71		233	316.89	336.4	293.19	291.1
34	13.769	13.999	20.43	16.43		234	102.93	113.99	90.88	90.52
35	129.92	129.4	141.28	137.36		235	58.416	58.452	68.31	64.69
36	70.236	70.781	83.97	83.46		236	94.168	92.204	104.23	100.69
37	76.858	74.395	81.92	80.99		237	45.113	44.811	57.75	55.99
38	140.83	134.86	160.61	160.52		238	87.759	85.788	88.08	84.95
39	74.72	73.988	86.01	83.58		239	43.145	43.307	44.99	44.4
40	56.159	56.121	74.79	74.35		240	52.81	52.586	71.31	68.83
41	126.67	124.03	145.34	143.71		241	63.57	64.053	80.76	79.72
42	79.035	77.249	95.22	91.68		242	41.186	40.724	43.36	41.58
43	66.167	63.646	83.98	81.79		243	32.92	32.385	44.67	41.29
44	108.06	103.11	119.79	118.31		244	58.508	57.617	76.14	75.36
45	43.948	43.834	58.49	57.66		245	34.004	33.926	38.73	37.51
46	43.699	42.354	63.01	61.25		246	29.628	29.513	45.74	43.81
47	63.858	63.574	76.69	72.87		247	1587.8	3113.9	536.91	535.56
48	30.728	30.643	47.39	46.89		248	187.29	893.97	145.75	142.56
49	23.701	23.341	37.06	35.18		249	454.13	848.79	429.99	426.04
50	133.19	129.16	149.26	145.83		250	94.115	122.74	83.3	82.66
51	72.593	67.252	93.28	93.11		251	70.128	69.641	87.18	86.23
52	149.82	144.35	169.34	166.57		252	109.62	108.05	124.48	121.67
53	81.312	81.256	101.41	97.49		253	77.568	77.41	85.17	83.67
54	59.822	60.174	73.68	72.55		254	48.767	48.009	61.91	58.02
55	72.942	71.873	83.29	82.75		255	86.819	85.458	95.71	91.82
56	71.748	68.099	90.71	87.97		256	50.297	50.034	61.97	59.4
57	128.65	126.12	151.86	148.22		257	36.546	36.09	53.61	50.17
58	44.719	44.693	48.16	45.72		258	52.606	51.802	69.27	67.66
59	54.799	54.679	61.04	57.44		259	37.319	37.166	44.48	41.95
60	134.3	130.77	153.66	152.89		260	30.505	30.052	43.34	39.4
61	46.571	46.901	58.54	55.52		261	48.028	47.069	55.78	53.54
62	39.707	39.956	42.38	40.99		262	29.85	29.746	47.9	44.17
63	70.622	69.01	86.15	84.48		263	36.166	35.275	40.06	37.18
64	37.642	38.209	53.49	52.87		264	765.44	2593.7	202.01	200.07
65	32.269	32.253	34.47	31.19		265	242.5	1016.1	139.64	137.08
66	172.73	170.31	187.44	184.94		266	89.381	97.771	77.07	73.52
67	90.56	90.627	106.93	103.98		267	113.47	111.8	129.3	128.51
68	74.335	76.137	95.53	92.31		268	70.159	70.28	86.18	84.6
69	74.258	72.237	87.48	87.21		269	40.335	39.249	56.03	52.06
70	86.289	86.464	98.41	94.61		270	95.427	95.035	100.11	98.5
71	78.252	76.021	86.68	84.69		271	58.694	58.749	67.94	65.3
72	68.885	69.012	88.31	85.29		272	35.789	35.614	43.32	39.71
73	66.792	67.442	85.92	82.95		273	65.888	65.269	76.42	72.44
74	86.829	88.025	106.42	103.1		274	43.119	42.574	54.83	52.22
75	50.806	51.41	68.71	68.08		275	31.965	31.685	35.9	35.47
76	40.086	39.657	46.35	44.52		276	51.494	50.85	68.22	68.08
77	96.432	94.166	111.8	109.33		277	34.802	34.469	47.01	44.54
78	43.701	43.177	59.16	55.43		278	24.646	24.583	27.83	25.56
79	35.59	35.014	39.29	35.95		279	540.89	428.74	556.36	552.51
80	62.271	62.499	78.68	75.1		280	436.53	1206.8	170.9	167.92
81	26.945	26.708	42.95	40.62		281	472.66	1530.6	356.85	354.2
82	18.308	18.379	22.33	20		282	131.94	251.03	123.15	121.06
83	231.27	279.15	211.87	208.45		283	156.59	156.46	166.61	165.57
84	324.01	407.17	203.59	203.45		284	88.446	86.362	100.47	96.62
85	271.61	318.32	94.45	90.91		285	62.711	62.113	74.94	72.78
86	152.17	162.13	147.49	145.86		286	118.12	117.71	125.4	125.28

87	83.999	85.209	92.53	92.38		287	69.258	70.173	76.32	73.53
88	112.62	113.49	124.34	121.36		288	36.277	36.605	55.1	53.02
89	73.381	72.51	78.05	77.43		289	81.855	81.288	99.24	99
90	139.06	136.15	148.57	147.99		290	55.696	55.443	70.08	66.52
91	72.555	72.83	90.92	88.5		291	33.334	34.016	50.4	49.08
92	54.58	54.216	69.3	68.28		292	69.541	67.283	78.05	77.13
93	118.8	119.92	134.74	133.44		293	42.574	42.303	51.39	50.93
94	63.37	64.494	80.35	78.74		294	23.073	23.201	34.44	33.2
95	49.323	49.124	55	53.37		295	35.99	35.358	56.3	55.39
96	105.68	104.35	124.94	123.4		296	18.718	18.799	30.14	27.53
97	41.799	42.092	55.49	53.05		297	15.99	16.169	23.13	22.87
98	36.517	35.98	45.02	44.35		298	178.87	178.73	192.53	191.43
99	41.183	40.373	50.33	49.58		299	175.88	178.61	194.65	193.52
100	31.331	31.726	41.09	40.71		300	210.43	210.09	226.65	223.13
101	27.543	27.38	42.91	41.62		301	376.91	410.02	293.25	291.47
102	7.7697	7.8756	21.42	18.34		302	373.82	1336.3	127.64	124.62
103	267.45	267.65	278.91	277.97		303	853.49	1778.8	246.89	244.48
104	115.52	114.68	120.26	117.3		304	6978.7	12552	6194.36	6191.23
105	840.16	1450.6	642.48	639.71		305	115.1	115.47	123.55	123.09
106	778.31	1518.4	450.8	447.5		306	116.17	115.85	125.1	121.19
107	547.22	735.77	266.33	263.02		307	102.02	100.11	122.08	118.69
108	631.96	1570.3	98.43	97.26		308	60.696	59.251	69.16	68.96
109	71.9	76.031	89.45	88.21		309	91.67	90.591	115.3	113.44
110	153.31	148.83	167.85	165.76		310	66.039	64.844	80.19	78.89
111	80.265	78.035	96.86	95.56		311	39.961	39.553	48.89	46.37
112	53.209	54.216	72.78	69.45		312	95.788	94.322	106.53	105.61
113	150.5	150.41	164.8	161.56		313	60.657	60.892	66.68	64.36
114	60.227	60.638	73.52	71.29		314	31.674	31.59	45.04	42.63
115	37.332	36.893	52.2	51.15		315	69.983	69.013	77.92	75.52
116	112.28	111.7	133.13	130.41		316	47.417	47.673	60.25	58.46
117	47.519	47.42	50.27	49.34		317	21.646	21.304	33.17	33.03
118	35.273	35.024	43.42	41.59		318	1.5052	1.6892	14.4	12.34
119	102.65	102.59	108.68	107.14		319	118.41	119.74	137.75	136.12
120	49.88	49.701	66.83	64.68		320	258.16	221.37	232.35	231.92
121	31.794	32.025	40.93	36.96		321	317.89	350.14	302.68	300.84
122	62.285	61.95	71.87	68.85		322	550.76	2080.5	112.22	110.42
123	41.928	41.776	48.98	45.06		323	1078.9	2094.2	379.38	377.18
124	25.117	25.062	36.37	35.43		324	4707.4	5967.2	1968.44	1965.22
125	185.2	186.99	195.7	193.59		325	100.01	118.53	85.8	83
126	373.72	300.85	239.79	239.58		326	123.56	122.08	135.38	131.89
127	136.21	170	134.92	131.89		327	82.887	83.666	91.11	90.9
128	401.8	812.35	338.99	336.58		328	95.767	95.633	109.23	108.35
129	196.91	236.89	175.22	171.79		329	69.635	68.79	76.72	74.88
130	473.21	1453	183.11	179.16		330	98.198	96.683	110.37	106.54
131	251.65	600.16	225.91	222.19		331	75.786	76.198	92.85	89.69
132	66.674	65.541	84.63	82.99		332	37.596	37.92	48.05	46.24
133	69.493	69.532	76.08	76.08		333	63.359	63.708	83.21	81.88
134	40.284	40.224	44.37	42.21		334	47.177	48.436	52.91	52.67
135	50.302	51.023	62.34	61.51		335	30.88	30.8	43.94	40.98
136	33.938	33.864	48.85	47.97		336	51.149	51.369	56.58	54.55
137	90.971	90.216	96.63	95.33		337	31.752	32.117	46.38	45.58
138	47.205	47.065	49.45	49.07		338	22.356	22.396	34.24	32.53
139	32.822	32.42	35.68	32.69		339	368.95	547.41	275.76	275.09
140	79.242	80.443	86.1	83.11		340	137.26	174.75	130.78	127.77
141	40.862	40.566	55.33	53.16		341	106.52	105.64	125.8	124.33
142	26.879	26.836	35.25	33.9		342	149.55	148.16	163.91	160.14
143	22.476	22.278	34.36	31.03		343	247.12	1275.3	147.47	147.4
144	15.505	15.607	27.65	25.44		344	106.36	104.82	115.8	112.48
145	13.13	13.009	22.92	19.09		345	118.19	114.65	126.46	123.95
146	247.94	364.91	138.83	135.26		346	102.67	116.92	122.83	120.68
147	1082	4112.7	524.31	522.88		347	73.201	71.552	83.23	80.63
148	53.95	54.624	69.3	67.11		348	89.705	89.469	108.26	105.35
149	146.65	147.38	167.15	165.76		349	100.31	98.914	116.33	115.95
150	88.707	88.374	98.05	95.56		350	52.815	53.506	72.9	69.39
151	40.48	40.809	44	40.81		351	99.145	97.459	116.17	116.11
152	111.03	112.96	116	113.02		352	59.987	60.224	80.21	79.03
153	70.748	71.729	86.88	86.38		353	36.045	35.614	40.91	40.19

154	36.229	35.773	43.97	40.68		354	76.149	76.784	90.65	86.94
155	94.701	92.974	113.82	113.72		355	46.631	46.051	58.8	58.53
156	50.168	48.967	63.48	61.82		356	31.772	31.636	33.56	31.24
157	35.974	35.617	46.67	43.74		357	68.078	67.932	80.98	78.43
158	78.067	77.834	97.2	94.07		358	96.801	95.306	104.67	102.06
159	44.822	44.851	54.42	52.95		359	86.839	86.978	104.16	100.7
160	29.833	29.609	39.36	36.38		360	71.382	71.439	80.09	79.87
161	55.772	56.056	66.18	62.61		361	66.34	66.557	73.07	69.8
162	36.899	36.633	53.84	52.87		362	98.628	97.186	119.68	117.56
163	24.07	24.22	31.39	30.87		363	71.346	73.449	87.07	84.29
164	861.97	2765.1	164.88	163.98		364	75.637	74.806	89.11	88.26
165	1264.5	5293.1	703.42	702.02		365	76.089	76.127	80.79	78.62
166	155.47	158.75	171.73	170.58		366	85.739	83.883	105.28	102.47
167	145.27	141.29	171.6	167.89		367	52.708	52.926	69.27	65.44
168	44.396	44.604	63.6	63.39		368	66.438	65.549	70.3	68.52
169	100.61	100.28	110.99	108.62		369	65.741	65.134	72.64	72.3
170	94.604	97.139	113.99	113.34		370	43.127	43.03	60.26	60.03
171	38.479	39.235	50.72	47.37		371	222.86	221.87	240.43	237.91
172	92.985	90.422	110.26	109.59		372	56.79	56.517	59.59	56.41
173	66.358	66.216	87.27	85.26		373	63.215	62.756	68.41	65.65
174	36.061	35.943	46.06	42.06		374	55.138	55.599	60.58	59.2
175	75.589	74.223	89.38	87.96		375	56.664	56.454	75.71	71.92
176	57.812	55.205	80.65	80.46		376	99.107	104.78	118.78	116.7
177	31.274	31.053	37.6	36.75		377	52.106	52.166	63.83	60.01
178	63.516	63.512	69.05	67.46		378	63.838	62.579	76.78	76.49
179	45.402	44.779	65.13	63.8		379	100.83	100.94	105.68	104.85
180	27.294	27.208	34.42	33.5		380	59.018	57.924	75.49	72.39
181	6684	8097.4	4705.08	4701.34		381	69.895	69.203	77.08	73.42
182	1771.5	5466.7	164.17	161.44		382	83.843	82.03	94.81	91.68
183	4186	3743.1	3189.26	3185.41		383	58.438	58.1	71.32	70.14
184	720.51	3116.1	312.34	310.59		384	57.98	57.713	73.37	72.76
185	102.88	101.09	123.89	120.13		385	36.147	35.806	44.13	40.74
186	44.197	44.099	62.11	62.09		386	73.603	70.936	89.92	86.78
187	104.59	104.82	122.5	120.06		387	52.672	52.833	71.34	70.26
188	67.709	66.839	73.36	70.16		388	62.906	63.291	76.2	75.29
189	39.774	39.957	49.38	48.45		389	90.588	89.391	102.67	101.39
190	93.8	92.422	115.99	112.26		390	57.303	57.746	64.32	61
191	63.035	62.31	72.45	69.4		391	51.87	52.418	62.97	59.68
192	31.77	31.627	47.56	44.25		392	69.922	69.97	74.11	71.83
193	73.527	73.815	79.43	77.14		393	38.767	38.608	54.16	51.87
194	50.052	49.013	69.83	66.66		394	55.824	54.598	67.55	66.41
195	29.233	29.168	45.71	44.39		395	108.04	105.7	119.85	117.05
196	69.537	70.167	80.99	80.1		396	37.423	37.72	48.09	44.9
197	38.88	38.597	47.69	46.44		397	62.714	62.907	66.99	65.22
198	31.46	31.514	37.64	35.3		398	75.359	72.657	90.75	88.97
199	5899.2	5570.6	3872.25	3868.93		399	34.171	34.21	42.23	40.37
200	63.967	70.766	62.11	60.95		400	7.0318	6.6065	16.74	15.62

Appendix F

Trials of Different Data Embedding Methods for 2D MRI Dataset

F.1 Different Data Embedding Methods for 2D MRI Dataset

MSEs of Images Registered with Learning Algorithm with Different Data Embedding Methods(experiment described in section 4.3.1)

Trial no.	LK	Degraded LK	PCA	LLE	Laplacian Eigenmaps	LTSA
1	93.55	145.52	135.92	233.06	120.00	152.16
2	206.17	289.01	266.20	441.06	250.77	253.56
3	136.65	193.89	170.00	313.32	176.19	158.13
4	100.84	143.45	148.63	239.50	128.04	117.93
5	90.89	133.58	120.46	249.05	87.90	121.03
6	153.87	223.29	352.36	454.85	177.78	348.31
7	96.3	139.58	150.16	269.81	121.62	155.12
8	151.23	221.61	546.81	404.32	168.67	592.35
9	146.99	209.9	187.07	288.82	156.03	166.26
10	106.67	144.48	142.61	292.85	152.55	141.87
11	100.85	128.76	132.04	192.43	121.63	96.00
12	115.93	166.25	181.41	274.42	168.71	124.71
13	107.85	141.88	115.61	240.05	122.94	108.11
14	90.83	128.97	119.02	245.95	140.08	141.86
15	109.11	158.78	188.44	262.68	119.67	219.48
16	39.66	53.29	227.45	104.99	88.55	175.46
17	120.54	153.63	165.44	260.72	166.82	151.00
18	154.17	233.98	185.02	394.44	172.60	162.02
19	128.31	193.54	168.74	316.20	135.98	126.93
20	82.14	113.1	112.43	188.51	125.51	136.76
21	82.14	113.1	126.53	171.70	137.65	118.77
22	101.25	147.03	139.36	253.99	120.69	166.86
23	101.6	134.88	143.81	207.09	140.01	142.28
24	184.06	273.92	224.18	405.75	215.10	230.98

25	82.14	113.1	374.79	218.70	124.36	361.32
26	101.25	147.03	105.75	234.36	139.04	152.68
27	112.29	171.59	223.54	446.97	122.79	286.87
28	112.78	165.85	109.20	270.82	153.73	106.41
29	101.75	145.75	103.46	233.07	157.59	158.74
30	92.25	125.34	128.63	236.77	124.67	146.19

Appendix G

Trials of Data Embedding Methods for 3D Gated CT Dataset

G.1 Different Data Embedding Methods

MSEs of Registered Images with Learning Algorithm Using Different Data Embedding Methods

Trial no.	LK	Degraded LK	PCA	LLE	Laplacian Eigen-maps	LTSA	Trial no.	LK	Degraded LK	PCA	LLE	Laplacian Eigen-maps	LTSA
1	51.78	51.87	58.71	86.55	57.94	65.86	201	401.28	636.92	324.15	473.56	321.01	360.25
2	32.89	33.06	34.98	51.97	37.04	45.37	202	79.63	78.25	81.57	121.94	87.10	98.03
3	42.79	42.23	44.02	65.53	43.85	49.83	203	42.35	42.15	49.85	69.68	47.20	54.43
4	88.98	89.91	100.27	147.12	102.02	116.30	204	127.95	127.24	130.27	197.47	131.59	149.43
5	40.68	41.00	46.82	66.72	46.03	51.33	205	67.51	67.47	69.85	102.68	72.05	85.11
6	55.56	54.10	59.86	86.65	62.66	68.21	206	42.67	42.14	51.89	70.16	50.40	52.32
7	85.23	82.26	96.78	136.67	94.68	104.59	207	87.59	87.07	95.63	141.31	93.06	106.08
8	82.61	80.91	95.73	136.56	94.33	101.83	208	53.63	53.49	58.06	83.26	59.51	66.13
9	42.32	41.81	48.16	71.02	51.80	54.56	209	32.87	32.45	38.29	56.30	35.65	40.89
10	60.79	61.13	63.38	99.22	63.61	75.11	210	73.90	73.61	83.70	121.20	84.39	90.81
11	75.71	74.70	82.22	116.52	81.51	91.70	211	54.00	53.95	57.99	89.47	61.95	70.68
12	89.72	87.70	101.03	145.30	100.96	115.53	212	35.12	33.79	42.24	58.51	37.81	43.36
13	39.08	39.44	44.08	66.04	44.88	53.77	213	56.94	56.87	65.15	93.68	62.52	72.55
14	51.79	51.87	60.14	88.06	61.93	69.49	214	41.91	42.68	48.13	69.92	48.96	53.60
15	93.04	93.63	102.11	155.68	101.30	117.36	215	24.88	24.86	29.29	37.66	31.80	31.94
16	102.30	100.55	117.31	165.16	110.09	129.60	216	513.30	886.18	162.24	237.85	164.49	188.20
17	41.53	41.88	48.13	69.12	44.74	55.62	217	857.31	1238.20	304.41	454.25	304.60	346.75
18	59.79	58.93	66.43	93.22	63.23	72.56	218	86.59	87.01	87.64	130.72	88.65	96.20
19	72.64	72.92	83.16	128.18	84.75	96.48	219	48.98	48.64	61.66	88.66	61.40	69.33
20	88.52	86.83	94.72	143.28	97.56	108.44	220	111.92	112.94	117.16	176.43	115.79	135.22
21	43.21	43.22	44.85	71.09	46.54	54.61	221	54.80	54.64	58.86	84.21	58.37	67.53
22	65.41	64.22	75.05	103.90	70.73	85.45	222	45.18	44.66	49.65	75.35	49.30	57.61
23	30.69	30.61	37.49	49.99	32.27	39.18	223	92.46	91.41	102.78	154.75	106.05	115.61
24	33.60	34.01	36.47	52.78	39.34	39.88	224	50.17	49.57	55.07	79.63	56.90	66.43
25	97.50	95.54	108.73	155.79	106.04	122.74	225	35.09	35.63	43.40	58.63	40.76	45.35
26	60.11	59.22	67.07	97.91	64.42	77.18	226	82.42	81.16	96.23	136.60	94.25	105.45
27	65.92	64.44	73.53	107.12	69.68	80.41	227	38.40	37.71	44.41	63.00	41.56	47.41
28	93.74	89.67	109.03	157.01	107.77	119.94	228	36.34	36.23	40.20	62.36	41.06	42.74

29	66.03	65.10	71.63	111.31	76.97	87.01	229	66.54	65.57	75.20	110.47	75.80	87.77
30	59.38	57.41	66.66	99.93	68.83	76.12	230	41.27	41.36	48.15	69.93	49.25	52.47
31	110.41	108.75	118.71	176.14	114.92	134.47	231	28.99	29.27	35.50	47.51	34.17	35.52
32	57.57	57.21	64.92	95.05	67.26	74.33	232	4144.00	5467.10	1728.82	2571.84	1781.20	1930.25
33	44.97	44.58	48.57	72.57	53.72	55.14	233	316.89	336.40	296.67	440.47	293.94	342.35
34	13.77	14.00	18.73	24.96	14.52	22.17	234	102.93	113.99	87.46	127.23	85.72	95.18
35	129.92	129.40	137.77	206.67	139.31	163.72	235	58.42	58.45	61.05	90.58	65.38	71.62
36	70.24	70.78	82.30	121.45	83.18	89.97	236	94.17	92.20	103.47	156.28	103.02	121.64
37	76.86	74.40	86.98	122.28	84.26	99.12	237	45.11	44.81	52.46	74.71	46.65	59.79
38	140.83	134.86	157.54	230.69	155.84	181.81	238	87.76	85.79	89.52	132.99	90.84	104.77
39	74.72	73.99	85.36	125.05	87.00	92.00	239	43.15	43.31	43.45	65.56	49.52	54.77
40	56.16	56.12	60.40	90.50	64.96	70.61	240	52.81	52.59	59.18	84.97	56.95	68.27
41	126.67	124.03	141.06	208.95	141.26	160.54	241	63.57	64.05	66.43	98.10	68.16	77.34
42	79.04	77.25	84.04	124.53	84.85	96.09	242	41.19	40.72	45.88	69.13	45.35	54.11
43	66.17	63.65	78.94	118.07	83.45	93.16	243	32.92	32.39	42.17	57.87	36.46	48.30
44	108.06	103.11	116.56	172.64	115.41	133.88	244	58.51	57.62	68.12	94.68	64.97	74.41
45	43.95	43.83	46.73	67.90	45.60	54.79	245	34.00	33.93	39.36	57.39	40.39	42.52
46	43.70	42.35	47.48	73.13	48.44	56.59	246	29.63	29.51	34.86	54.07	33.37	40.79
47	63.86	63.57	72.31	105.36	69.86	79.67	247	1587.80	3113.90	604.21	837.74	523.40	686.08
48	30.73	30.64	35.52	49.04	35.57	42.22	248	187.29	893.97	136.52	201.40	136.79	155.04
49	23.70	23.34	32.27	44.86	24.97	36.13	249	454.13	848.79	438.41	648.96	434.58	497.44
50	133.19	129.16	143.05	212.02	140.09	164.90	250	94.12	122.74	82.02	118.29	79.15	92.61
51	72.59	67.25	83.65	116.27	77.98	93.97	251	70.13	69.64	76.37	114.57	76.43	85.37
52	149.82	144.35	162.94	235.03	161.95	180.52	252	109.62	108.05	115.12	169.55	113.00	131.42
53	81.31	81.26	87.78	133.61	90.78	99.51	253	77.57	77.41	82.07	122.66	83.76	96.15
54	59.82	60.17	67.09	93.40	67.66	75.76	254	48.77	48.01	53.00	79.62	53.91	62.63
55	72.94	71.87	82.27	122.94	83.63	91.74	255	86.82	85.46	97.97	137.41	91.80	108.23
56	71.75	68.10	90.46	124.57	89.60	95.63	256	50.30	50.03	56.23	83.63	55.10	60.44
57	128.65	126.12	141.98	212.21	143.27	160.37	257	36.55	36.09	39.55	55.35	41.93	43.97
58	44.72	44.69	51.14	73.97	48.96	55.53	258	52.61	51.80	60.79	84.11	61.28	65.95
59	54.80	54.68	59.81	91.59	65.09	70.47	259	37.32	37.17	41.31	59.78	46.55	49.92
60	134.30	130.77	151.62	223.66	149.35	167.36	260	30.51	30.05	37.57	53.99	38.62	41.49
61	46.57	46.90	50.46	81.09	49.83	62.88	261	48.03	47.07	55.75	80.10	55.96	64.19
62	39.71	39.96	42.76	63.33	43.55	50.92	262	29.85	29.75	37.28	47.11	37.18	42.12
63	70.62	69.01	75.33	115.87	81.23	91.17	263	36.17	35.28	41.55	60.90	44.24	45.02
64	37.64	38.21	44.73	64.16	44.21	47.14	264	765.44	2593.70	201.75	296.46	201.63	233.57
65	32.27	32.25	36.80	55.63	32.63	43.38	265	242.50	1016.10	132.41	190.12	128.78	148.44
66	172.73	170.31	185.69	265.86	184.48	204.81	266	89.38	97.77	78.79	119.69	76.52	92.85
67	90.56	90.63	97.72	136.96	107.79	110.58	267	113.47	111.80	126.25	182.81	124.85	141.88
68	74.34	76.14	83.58	123.94	80.50	97.37	268	70.16	70.28	77.87	114.27	78.62	84.85
69	74.26	72.24	77.81	116.53	77.58	91.69	269	40.34	39.25	50.63	67.64	45.39	50.54
70	86.29	86.46	100.67	141.87	95.94	112.88	270	95.43	95.04	102.59	147.78	103.58	114.44
71	78.25	76.02	89.25	129.81	85.15	100.13	271	58.69	58.75	65.00	95.09	61.00	74.44
72	68.89	69.01	83.03	124.51	82.43	97.08	272	35.79	35.61	41.73	62.84	40.01	43.48
73	66.79	67.44	74.03	114.72	74.78	83.34	273	65.89	65.27	74.93	105.97	70.17	80.34
74	86.83	88.03	94.24	137.41	94.86	105.97	274	43.12	42.57	51.24	67.66	45.91	52.31
75	50.81	51.41	58.26	85.70	60.75	69.13	275	31.97	31.69	34.54	52.19	37.84	39.31
76	40.09	39.66	44.88	69.86	45.83	50.88	276	51.49	50.85	54.42	79.97	55.04	60.65
77	96.43	94.17	103.47	159.99	104.53	118.55	277	34.80	34.47	38.21	55.74	42.84	46.38
78	43.70	43.18	45.61	71.66	44.63	55.34	278	24.65	24.58	30.68	38.08	28.55	29.22
79	35.59	35.01	39.62	62.59	38.76	45.16	279	540.89	428.74	552.52	824.70	551.91	631.22
80	62.27	62.50	67.62	104.59	65.79	81.57	280	436.53	1206.80	163.28	238.16	155.15	179.22
81	26.95	26.71	32.14	42.27	27.85	37.90	281	472.66	1530.60	351.94	517.98	347.78	399.78
82	18.31	18.38	24.78	33.70	23.55	29.84	282	131.94	251.03	112.87	170.96	115.53	130.73
83	231.27	279.15	214.96	312.01	210.61	241.09	283	156.59	156.46	164.37	241.83	164.33	182.60
84	324.01	407.17	202.89	298.37	199.81	229.78	284	88.45	86.36	103.62	148.25	100.62	116.15
85	271.61	318.32	88.62	131.35	83.11	96.67	285	62.71	62.11	71.91	100.06	71.41	79.75
86	152.17	162.12	148.15	212.03	141.25	165.62	286	118.12	117.71	125.87	184.25	127.49	146.39
87	84.00	85.21	91.58	132.18	90.07	102.68	287	69.26	70.17	75.23	106.81	71.96	80.23
88	112.62	113.49	120.57	180.88	118.03	136.08	288	36.28	36.61	40.58	62.07	40.97	49.70
89	73.38	72.51	84.02	116.25	76.65	90.61	289	81.86	81.29	88.37	130.95	89.73	97.58
90	139.06	136.15	148.45	217.93	150.07	171.48	290	55.70	55.44	59.91	87.15	56.99	64.58
91	72.56	72.83	81.98	121.37	77.27	88.99	291	33.33	34.02	36.22	56.90	37.39	40.63
92	54.58	54.22	61.07	93.73	65.23	69.16	292	69.54	67.28	79.71	115.77	77.11	88.91
93	118.80	119.92	132.81	199.67	134.90	155.14	293	42.57	42.30	46.89	69.78	47.73	50.78
94	63.37	64.49	68.42	103.27	71.58	78.84	294	23.07	23.20	25.38	35.65	24.51	29.69
95	49.32	49.12	56.26	87.91	59.35	63.21	295	35.99	35.36	45.50	68.67	46.33	49.24

96	105.68	104.35	122.46	177.40	122.81	137.56	296	18.72	18.80	19.18	31.97	24.33	24.37
97	41.80	42.09	49.78	66.70	47.64	53.64	297	15.99	16.17	18.78	25.88	20.24	21.78
98	36.52	35.98	42.42	63.39	44.23	49.09	298	178.87	178.73	184.85	272.71	181.72	211.16
99	41.18	40.37	47.64	70.26	48.33	51.95	299	175.88	178.61	184.93	271.43	181.25	212.52
100	31.33	31.73	36.15	53.56	34.38	40.71	300	210.43	210.09	222.26	328.39	220.82	248.59
101	27.54	27.38	33.08	43.92	31.43	36.95	301	376.91	410.02	297.04	445.49	291.98	342.43
102	7.77	7.88	10.77	12.63	14.30	11.03	302	373.82	1336.30	113.81	171.01	112.39	133.64
103	267.45	267.65	288.15	417.77	278.11	323.46	303	853.49	1778.80	254.65	372.01	246.89	288.54
104	115.52	114.68	123.08	175.89	119.02	135.24	304	6978.70	12552.00	6253.75	9282.74	6188.55	7115.64
105	840.16	1450.60	647.43	958.95	639.06	737.10	305	115.10	115.47	126.61	182.90	126.24	139.86
106	778.31	1518.40	404.88	601.58	442.43	462.59	306	116.17	115.85	124.71	183.23	124.63	140.58
107	547.22	735.77	267.36	394.46	263.60	304.44	307	102.02	100.11	114.44	168.83	111.75	129.65
108	631.96	1570.30	95.98	145.79	101.23	110.20	308	60.70	59.25	71.62	98.98	65.35	75.50
109	71.90	76.03	84.45	122.13	83.39	96.81	309	91.67	90.59	102.95	150.22	102.46	115.54
110	153.31	148.83	165.01	243.49	165.58	188.77	310	66.04	64.84	69.65	104.82	72.57	82.01
111	80.27	78.04	87.41	129.41	84.86	98.82	311	39.96	39.55	45.45	69.73	47.96	54.78
112	53.21	54.22	59.87	89.01	60.71	66.14	312	95.79	94.32	108.93	159.84	104.32	123.75
113	150.50	150.41	156.93	232.42	152.15	178.40	313	60.66	60.89	64.42	98.56	65.87	72.59
114	60.23	60.64	68.78	93.26	65.91	71.91	314	31.67	31.59	34.57	54.81	37.73	41.91
115	37.33	36.89	39.23	64.06	46.31	51.69	315	69.98	69.01	84.77	122.28	81.24	94.67
116	112.28	111.70	119.60	183.98	121.57	141.83	316	47.42	47.67	54.85	73.45	52.35	59.16
117	47.52	47.42	54.88	79.14	52.52	60.40	317	21.65	21.30	24.48	38.39	24.95	27.06
118	35.27	35.02	41.07	58.48	43.27	48.07	318	1.51	1.69	8.72	5.93	4.59	3.65
119	102.65	102.59	111.82	161.96	109.26	130.03	319	118.41	119.74	122.45	181.40	126.70	144.81
120	49.88	49.70	59.14	81.41	58.92	61.21	320	258.16	221.37	230.28	344.04	228.22	259.25
121	31.79	32.03	37.36	50.38	32.33	42.02	321	317.89	350.14	302.97	450.24	301.50	345.35
122	62.29	61.95	71.44	99.06	69.23	79.85	322	550.76	2080.50	101.08	151.42	99.87	116.03
123	41.93	41.78	46.81	73.75	49.88	56.90	323	1078.90	2094.20	387.18	574.28	370.44	446.64
124	25.12	25.06	28.22	45.13	31.24	32.75	324	4707.40	5967.20	2252.14	3341.29	1964.57	2600.58
125	185.20	186.99	195.61	293.13	194.13	222.49	325	100.01	118.53	89.64	131.29	88.81	97.92
126	373.72	300.85	245.49	360.31	240.87	275.64	326	123.56	122.08	133.16	189.43	129.23	148.09
127	136.21	170.00	137.11	200.50	137.69	153.61	327	82.89	83.67	90.34	135.77	90.71	101.10
128	401.80	812.35	348.39	513.30	344.22	385.85	328	95.77	95.63	104.14	153.29	103.53	122.90
129	196.91	236.89	173.95	258.73	173.03	196.49	329	69.64	68.79	73.32	112.07	71.15	82.63
130	473.21	1453.00	183.99	276.91	187.64	209.84	330	98.20	96.68	108.43	163.56	107.90	125.78
131	251.65	600.16	227.59	327.15	219.07	256.60	331	75.79	76.20	81.87	122.83	87.28	98.22
132	66.67	65.54	71.64	103.83	72.93	82.26	332	37.60	37.92	39.80	62.00	44.75	51.05
133	69.49	69.53	74.12	108.41	70.84	82.93	333	63.36	63.71	73.15	106.90	74.87	83.15
134	40.28	40.22	49.47	67.77	46.15	54.69	334	47.18	48.44	51.54	80.38	54.79	61.93
135	50.30	51.02	57.58	79.12	57.49	67.06	335	30.88	30.80	38.77	51.87	37.71	43.69
136	33.94	33.86	36.36	54.85	35.00	42.75	336	51.15	51.37	62.39	92.14	61.59	73.28
137	90.97	90.22	94.27	140.22	96.14	110.48	337	31.75	32.12	40.78	53.85	38.15	40.70
138	47.21	47.07	49.59	73.89	51.53	57.13	338	22.36	22.40	30.64	41.55	29.55	30.40
139	32.82	32.42	39.71	50.84	39.55	43.67	339	368.95	547.41	266.22	390.90	260.46	304.08
140	79.24	80.44	87.04	133.15	86.95	98.68	340	137.26	174.75	130.53	199.43	131.99	149.60
141	40.86	40.57	44.76	64.29	43.17	50.41	341	106.52	105.64	116.55	163.91	113.80	131.01
142	26.88	26.84	34.57	47.11	29.03	37.82	342	149.55	148.16	160.75	234.73	157.05	181.88
143	22.48	22.28	26.45	40.95	23.58	28.89	343	247.12	1275.30	144.55	210.50	137.28	161.41
144	15.51	15.61	18.16	24.14	22.41	19.55	344	106.36	104.82	114.36	167.99	117.25	128.25
145	13.13	13.01	13.94	22.46	13.32	15.20	345	118.19	114.65	130.89	192.24	130.22	145.42
146	247.94	364.91	149.47	193.47	128.94	154.10	346	102.67	116.92	110.12	165.65	109.41	126.61
147	1082.00	4112.70	516.08	768.72	509.48	591.09	347	73.20	71.55	87.96	120.71	81.81	99.16
148	53.95	54.62	58.73	86.24	59.55	72.92	348	89.71	89.47	107.36	157.19	106.89	121.34
149	146.65	147.38	157.92	228.27	152.35	181.09	349	100.31	98.91	106.96	160.51	108.42	122.00
150	88.71	88.37	91.75	139.39	94.60	106.90	350	52.82	53.51	61.34	88.27	56.47	69.51
151	40.48	40.81	44.36	69.03	45.22	54.13	351	99.15	97.46	110.64	162.18	110.26	128.49
152	111.03	112.96	118.64	173.91	116.56	133.69	352	59.99	60.22	64.45	98.95	64.37	73.14
153	70.75	71.73	81.29	116.29	76.10	91.93	353	36.05	35.61	38.50	60.23	42.65	48.95
154	36.23	35.77	41.40	58.35	39.37	50.74	354	76.15	76.78	85.81	130.73	90.39	101.47
155	94.70	92.97	109.59	153.92	104.99	120.07	355	46.63	46.05	48.88	70.95	46.99	56.35
156	50.17	48.97	59.26	78.52	52.44	60.11	356	31.77	31.64	37.53	49.34	37.01	41.76
157	35.97	35.62	39.58	60.17	41.86	48.61	357	68.08	67.93	74.12	106.72	73.02	80.32
158	78.07	77.83	89.99	128.41	88.39	97.68	358	96.80	95.31	101.82	150.52	99.30	118.33
159	44.82	44.85	50.46	75.34	48.00	57.45	359	86.84	86.98	94.74	142.48	96.27	105.93
160	29.83	29.61	37.32	48.55	36.27	38.67	360	71.38	71.44	78.18	110.42	75.70	84.99
161	55.77	56.06	62.26	92.77	63.60	71.18	361	66.34	66.56	74.78	104.06	67.84	83.05
162	36.90	36.63	43.16	61.91	39.77	45.10	362	98.63	97.19	108.01	159.22	103.57	121.38

163	24.07	24.22	30.29	41.51	28.37	32.33	363	71.35	73.45	78.90	114.25	80.45	90.70
164	861.97	2765.10	167.28	247.77	166.02	191.83	364	75.64	74.81	89.46	128.99	87.07	100.60
165	1264.50	5293.10	702.68	1046.69	701.72	805.45	365	76.09	76.13	82.60	120.00	84.59	91.94
166	155.47	158.75	161.08	238.33	165.96	187.55	366	85.74	83.88	96.05	140.47	97.16	105.89
167	145.27	141.29	162.03	241.93	164.21	187.01	367	52.71	52.93	65.11	91.77	64.22	71.90
168	44.40	44.60	52.23	77.38	51.50	55.23	368	66.44	65.55	73.04	104.68	69.29	82.34
169	100.61	100.28	112.45	162.21	108.60	128.29	369	65.74	65.13	75.66	111.75	73.43	85.32
170	94.60	97.14	105.43	157.17	101.92	120.78	370	43.13	43.03	48.63	70.36	45.32	50.71
171	38.48	39.24	42.98	66.65	44.85	49.06	371	222.86	221.87	238.84	350.64	235.35	267.11
172	92.99	90.42	105.97	160.64	105.71	125.73	372	56.79	56.52	61.17	88.67	65.38	69.32
173	66.36	66.22	83.63	118.87	80.48	95.12	373	63.22	62.76	68.18	101.84	69.45	83.26
174	36.06	35.94	43.87	62.36	41.48	44.36	374	55.14	55.60	60.66	89.64	62.38	68.86
175	75.59	74.22	85.92	121.99	79.57	94.99	375	56.66	56.45	67.20	96.89	60.70	72.29
176	57.81	55.21	70.07	102.19	72.76	84.24	376	99.11	104.78	116.29	171.12	111.95	132.12
177	31.27	31.05	35.68	50.59	32.86	41.20	377	52.11	52.17	58.05	83.48	60.14	68.38
178	63.52	63.51	73.65	106.36	67.77	82.56	378	63.84	62.58	71.68	103.94	69.73	83.56
179	45.40	44.78	53.27	82.12	53.10	64.55	379	100.83	100.94	109.85	158.29	110.34	121.31
180	27.29	27.21	34.79	42.24	28.34	37.88	380	59.02	57.92	65.11	96.88	62.60	70.49
181	6684.00	8097.40	4768.60	7172.54	4698.20	5445.02	381	69.90	69.20	76.93	114.79	76.87	87.40
182	1771.50	5466.70	163.82	242.40	167.05	186.46	382	83.84	82.03	91.95	138.44	90.61	103.95
183	4186.00	3743.10	3529.57	5640.01	3190.17	4358.78	383	58.44	58.10	64.41	96.38	66.36	72.73
184	720.51	3116.10	302.09	450.11	298.13	343.41	384	57.98	57.71	69.33	102.85	68.78	79.55
185	102.88	101.09	113.64	168.99	112.00	125.13	385	36.15	35.81	40.68	56.56	44.47	47.46
186	44.20	44.10	47.27	69.18	48.02	55.87	386	73.60	70.94	83.08	122.20	80.80	92.91
187	104.59	104.82	109.67	168.29	112.91	128.54	387	52.67	52.83	58.63	83.52	58.03	70.40
188	67.71	66.84	74.39	109.26	75.58	90.13	388	62.91	63.29	72.86	103.71	66.45	77.53
189	39.77	39.96	43.32	66.12	46.78	51.38	389	90.59	89.39	99.37	149.89	104.61	115.92
190	93.80	92.42	105.64	148.12	105.41	116.16	390	57.30	57.75	66.58	91.20	63.86	70.94
191	63.04	62.31	71.24	100.31	70.42	76.68	391	51.87	52.42	59.07	83.61	56.09	67.09
192	31.77	31.63	34.26	52.76	35.13	39.17	392	69.92	69.97	79.81	117.55	78.67	90.60
193	73.53	73.82	83.92	119.69	80.27	94.05	393	38.77	38.61	48.87	66.57	42.81	50.47
194	50.05	49.01	59.24	81.02	59.25	61.75	394	55.82	54.60	62.64	88.67	59.48	66.60
195	29.23	29.17	33.40	50.45	31.49	39.68	395	108.04	105.70	123.83	176.75	121.66	137.15
196	69.54	70.17	72.83	107.12	72.32	87.63	396	37.42	37.72	41.18	60.08	45.10	47.31
197	38.88	38.60	45.85	63.37	44.00	51.47	397	62.71	62.91	72.68	100.08	67.85	76.93
198	31.46	31.51	35.57	48.14	38.24	38.88	398	75.36	72.66	84.80	127.74	79.83	98.29
199	5899.20	5570.60	4008.69	6067.61	3862.82	4655.96	399	34.17	34.21	41.06	57.98	37.75	46.68
200	63.97	70.77	68.79	96.13	63.39	74.33	400	7.03	6.61	7.65	16.62	8.52	9.81

Appendix H

Trials of Learning set of Representative Patches Only for 2D MRI Dataset

H.1 A Learning set of Representative Patches Only for 2D MRI Dataset

MSEs of Images Registered with Learning Algorithm With A Learning set of Representative Patches Only (experiment described in section 4.3.3.1)

Trial no.	LK	Degraded LK	With *representative patches only*
1	93.55	145.52	352.16
2	206.17	289.01	289.38
3	136.65	193.89	200.15
4	100.84	143.45	306.99
5	90.89	133.58	136.03
6	153.87	223.29	292.45
7	96.30	139.58	269.19
8	151.23	221.61	298.98
9	146.99	209.90	264.06
10	106.67	144.48	333.94
11	100.85	128.76	812.42
12	115.93	166.25	127.93
13	107.85	141.88	259.62
14	90.83	128.97	282.36
15	109.11	158.78	156.17
16	39.66	53.29	162.12
17	120.54	153.63	120.9
18	154.17	233.98	168.94
19	128.31	193.54	243.95

20	82.14	113.10	205.03
21	82.14	113.10	144.26
22	101.25	147.03	211.02
23	101.60	134.88	147.32
24	184.06	273.92	150.33
25	82.14	113.10	287.79
26	101.25	147.03	140.6
27	112.29	171.59	148.32
28	112.78	165.85	430.25
29	101.75	145.75	781.12
30	92.25	125.34	189.68

Appendix I

Learning Algorithm with *Representative Patches Only* for 3D Gated CT Dataset

I.1 Comparison of Learning Algorithm with Learning Set of Represenative Patches (RP) Only

MSEs of Registered Images with Learning Algorithm Using Learning Set of Representative Patches Only

Trial #	LK	Degraded LK	RP	Trial #	LK	Degraded LK	RP	Trial #	LK	Degraded LK	RP	Trial #	LK	Degraded LK	RP
1	51.78	51.87	134.99	101	27.54	27.38	144.59	201	401.28	636.92	751.88	300	210.43	210.09	680.32
2	32.89	33.06	112.08	102	7.77	7.88	56.29	202	79.63	78.25	212.09	301	376.91	410.02	273.53
3	42.79	42.23	151.03	103	267.45	267.65	618.60	203	42.35	42.15	142.35	302	373.82	1336.3	622.09
4	88.98	89.91	253.27	104	115.52	114.68	263.32	204	127.95	127.24	330.17	303	853.49	1778.8	13275.08
5	40.68	41	131.34	105	840.16	1450.6	1441.62	205	67.51	67.47	208.19	304	6978.7	12552	338.69
6	55.56	54.1	162.59	106	778.31	1518.4	829.64	206	42.67	42.14	184.25	305	115.1	115.47	281.22
7	85.23	82.26	278.89	107	547.22	735.77	599.63	207	87.59	87.07	196.95	306	116.17	115.85	306.04
8	82.61	80.91	253.29	108	631.96	1570.3	263.29	208	53.63	53.49	122.43	307	102.02	100.11	209.14
9	42.32	41.81	105.71	109	71.9	76.03	201.64	209	32.87	32.45	113.78	308	60.7	59.25	252.95
10	60.79	61.13	179.03	110	153.31	148.83	429.33	210	73.9	73.61	191.54	309	91.67	90.59	206.01
11	75.71	74.7	183.16	111	80.27	78.04	270.60	211	54	53.95	178.17	310	66.04	64.84	158.22
12	89.72	87.7	208.45	112	53.21	54.22	157.30	212	35.12	33.79	152.00	311	39.96	39.55	251.24
13	39.08	39.44	101.07	113	150.5	150.41	386.81	213	56.94	56.87	194.37	312	95.79	94.32	151.28
14	51.79	51.87	174.64	114	60.23	60.64	166.78	214	41.91	42.68	122.33	313	60.66	60.89	141.78
15	93.04	93.63	305.13	115	37.33	36.89	141.83	215	24.88	24.86	74.36	314	31.67	31.59	241.40
16	102.3	100.55	299.81	116	112.28	111.7	311.42	216	513.3	886.18	408.63	315	69.98	69.01	161.67
17	41.53	41.88	106.69	117	47.52	47.42	142.98	217	857.31	1238.2	698.80	316	47.42	47.67	77.05
18	59.79	58.93	136.93	118	35.27	35.02	109.71	218	86.59	87.01	261.51	317	21.65	21.3	27.07
19	72.64	72.92	272.62	119	102.65	102.59	262.82	219	48.98	48.64	132.94	318	1.51	1.69	261.47
20	88.52	86.83	225.00	120	49.88	49.7	127.44	220	111.92	112.94	320.03	319	118.41	119.74	569.89
21	43.21	43.22	152.33	121	31.79	32.03	117.96	221	54.8	54.64	156.72	320	258.16	221.37	736.88
22	65.41	64.22	230.29	122	62.29	61.95	153.96	222	45.18	44.66	104.34	321	317.89	350.14	247.73
23	30.69	30.61	124.59	123	41.93	41.78	172.37	223	92.46	91.41	258.31	322	550.76	2080.5	837.17
24	33.6	34.01	169.73	124	25.12	25.06	83.83	224	50.17	49.57	137.25	323	1078.9	2094.2	4043.14
25	97.5	95.54	261.03	125	185.2	186.99	459.02	225	35.09	35.63	152.58	324	4707.4	5967.2	276.79
26	60.11	59.22	178.04	126	373.72	300.85	559.67	226	82.42	81.16	247.76	325	100.01	118.53	277.14
27	65.92	64.44	200.45	127	136.21	170	368.81	227	38.4	37.71	94.32	326	123.56	122.08	277.54
28	93.74	89.67	235.68	128	401.8	812.35	798.87	228	36.34	36.23	83.99	327	82.89	83.67	226.58
29	66.03	65.1	209.96	129	196.91	236.89	371.31	229	66.54	65.57	174.63	328	95.77	95.63	180.85

30	59.38	57.41	166.91	130	473.21	1453	466.85	230	41.27	41.36	131.92	329	69.64	68.79	237.03
31	110.41	108.75	315.77	131	251.65	600.16	520.59	231	28.99	29.27	83.84	330	98.2	96.68	192.49
32	57.57	57.21	176.82	132	66.67	65.54	163.96	232	4144	5467.1	3710.83	331	75.79	76.2	90.03
33	44.97	44.58	139.14	133	69.49	69.53	241.96	233	316.89	336.4	717.75	332	37.6	37.92	245.42
34	13.77	14	91.12	134	40.28	40.22	133.09	234	102.93	113.99	223.71	333	63.36	63.71	167.39
35	129.92	129.4	350.68	135	50.3	51.02	147.72	235	58.42	58.45	144.77	334	47.18	48.44	103.33
36	70.24	70.78	244.25	136	33.94	33.86	127.42	236	94.17	92.2	314.87	335	30.88	30.8	202.68
37	76.86	74.4	245.64	137	90.97	90.22	290.11	237	45.11	44.81	121.96	336	51.15	51.37	150.17
38	140.83	134.86	416.39	138	47.21	47.07	173.26	238	87.76	85.79	263.82	337	31.75	32.12	84.74
39	74.72	73.99	240.40	139	32.82	32.42	136.91	239	43.15	43.31	194.13	338	22.36	22.4	615.54
40	56.16	56.12	217.68	140	79.24	80.44	269.42	240	52.81	52.59	122.91	339	368.95	547.41	305.03
41	126.67	124.03	364.48	141	40.86	40.57	150.89	241	63.57	64.05	216.19	340	137.26	174.75	266.74
42	79.04	77.25	202.69	142	26.88	26.84	143.63	242	41.19	40.72	128.50	341	106.52	105.64	345.22
43	66.17	63.65	177.30	143	22.48	22.28	131.97	243	32.92	32.39	107.19	342	149.55	148.16	355.35
44	108.06	103.11	250.67	144	15.51	15.61	39.72	244	58.51	57.62	135.90	343	247.12	1275.3	252.93
45	43.95	43.83	174.50	145	13.13	13.01	57.26	245	34	33.93	138.12	344	106.36	104.82	272.83
46	43.7	42.35	161.20	146	247.94	364.91	353.52	246	29.63	29.51	119.88	345	118.19	114.65	322.29
47	63.86	63.57	148.74	147	1082	4112.7	1155.68	247	1587.8	3113.9	1163.76	346	102.67	116.92	232.20
48	30.73	30.64	165.56	148	53.95	54.62	170.85	248	187.29	893.97	330.19	347	73.2	71.55	300.99
49	23.7	23.34	58.13	149	146.65	147.38	421.46	249	454.13	848.79	958.49	348	89.71	89.47	265.83
50	133.19	129.16	324.91	150	88.71	88.37	218.54	250	94.12	122.74	184.54	349	100.31	98.91	143.69
51	72.59	67.25	228.83	151	40.48	40.81	91.24	251	70.13	69.64	198.54	350	52.82	53.51	287.79
52	149.82	144.35	421.50	152	111.03	112.96	247.25	252	109.62	108.05	329.76	351	99.15	97.46	166.49
53	81.31	81.26	278.46	153	70.75	71.73	238.76	253	77.57	77.41	259.55	352	59.99	60.22	146.15
54	59.82	60.17	148.54	154	36.23	35.77	90.92	254	48.77	48.01	142.65	353	36.05	35.61	258.26
55	72.94	71.87	183.95	155	94.7	92.97	232.25	255	86.82	85.46	220.91	354	76.15	76.78	144.31
56	71.75	68.1	264.18	156	50.17	48.97	178.32	256	50.3	50.03	158.40	355	46.63	46.05	80.72
57	128.65	126.12	379.65	157	35.97	35.62	168.95	257	36.55	36.09	145.01	356	31.77	31.64	159.65
58	44.72	44.69	122.97	158	78.07	77.83	215.84	258	52.61	51.8	147.66	357	68.08	67.93	218.35
59	54.8	54.68	198.62	159	44.82	44.85	169.09	259	37.32	37.17	148.73	358	96.8	95.31	289.23
60	134.3	130.77	393.44	160	29.83	29.61	107.18	260	30.51	30.05	142.39	359	86.84	86.98	176.70
61	46.57	46.9	189.05	161	55.77	56.06	183.76	261	48.03	47.07	184.87	360	71.38	71.44	204.99
62	39.71	39.96	179.11	162	36.9	36.63	181.07	262	29.85	29.75	110.75	361	66.34	66.56	306.70
63	70.62	69.01	160.47	163	24.07	24.22	119.54	263	36.17	35.28	172.88	362	98.63	97.19	231.18
64	37.64	38.21	163.22	164	861.97	2765.1	391.74	264	765.44	2593.7	440.65	363	71.35	73.45	258.78
65	32.27	32.25	88.46	165	1264.5	5293.1	1551.05	265	242.5	1016.1	358.15	364	75.64	74.81	222.20
66	172.73	170.31	480.86	166	155.47	158.75	390.29	266	89.38	97.77	249.48	365	76.09	76.13	211.19
67	90.56	90.63	254.83	167	145.27	141.29	350.50	267	113.47	111.8	274.27	366	85.74	83.88	210.29
68	74.34	76.14	180.14	168	44.4	44.6	131.96	268	70.16	70.28	246.74	367	52.71	52.93	175.18
69	74.26	72.24	202.98	169	100.61	100.28	263.24	269	40.34	39.25	149.37	368	66.44	65.55	232.78
70	86.29	86.46	299.51	170	94.6	97.14	219.87	270	95.43	95.04	215.65	369	65.74	65.13	169.03
71	78.25	76.02	191.50	171	38.48	39.24	144.88	271	58.69	58.75	207.20	370	43.13	43.03	580.54
72	68.89	69.01	221.25	172	92.99	90.42	230.84	272	35.79	35.61	117.01	371	222.86	221.87	206.41
73	66.79	67.44	181.00	173	66.36	66.22	215.71	273	65.89	65.27	227.78	372	56.79	56.52	180.53
74	86.83	88.03	236.59	174	36.06	35.94	143.94	274	43.12	42.57	133.03	373	63.22	62.76	223.49
75	50.81	51.41	146.49	175	75.59	74.22	178.49	275	31.97	31.69	136.98	374	55.14	55.6	147.00
76	40.09	39.66	156.97	176	57.81	55.21	199.81	276	51.49	50.85	120.82	375	56.66	56.45	315.58
77	96.43	94.17	229.26	177	31.27	31.05	166.45	277	34.8	34.47	153.98	376	99.11	104.78	176.53
78	43.7	43.18	143.56	178	63.52	63.51	203.09	278	24.65	24.58	68.30	377	52.11	52.17	187.80
79	35.59	35.01	129.68	179	45.4	44.78	182.68	279	540.89	428.74	1241.63	378	63.84	62.58	302.06
80	62.27	62.5	238.50	180	27.29	27.21	116.57	280	436.53	1206.8	422.18	379	100.83	100.94	174.60
81	26.95	26.71	117.19	181	6684	8097.4	10035.42	281	472.66	1530.6	818.92	380	59.02	57.92	217.54
82	18.31	18.38	42.13	182	1771.5	5466.7	387.18	282	131.94	251.03	313.42	381	69.9	69.2	269.81
83	231.27	279.15	540.89	183	4186	3743.1	6684.11	283	156.59	156.46	410.62	382	83.84	82.03	211.88
84	324.01	407.17	507.74	184	720.51	3116.1	679.29	284	88.45	86.36	211.71	383	58.44	58.1	184.24
85	271.61	318.32	209.03	185	102.88	101.09	299.94	285	62.71	62.11	171.43	384	57.98	57.71	126.90
86	152.17	162.12	372.21	186	44.2	44.1	168.09	286	118.12	117.71	328.10	385	36.15	35.81	224.40
87	84	85.21	202.96	187	104.59	104.82	275.68	287	69.26	70.17	158.42	386	73.6	70.94	170.99
88	112.62	113.49	269.57	188	67.71	66.84	195.42	288	36.28	36.61	94.04	387	52.67	52.83	168.63
89	73.38	72.51	185.55	189	39.77	39.96	138.43	289	81.86	81.29	247.96	388	62.91	63.29	246.92
90	139.06	136.15	411.88	190	93.8	92.42	265.98	290	55.7	55.44	218.88	389	90.59	89.39	224.34
91	72.56	72.83	207.27	191	63.04	62.31	214.93	291	33.33	34.02	138.35	390	57.3	57.75	170.49
92	54.58	54.22	133.22	192	31.77	31.63	130.05	292	69.54	67.28	244.26	391	51.87	52.42	232.99
93	118.8	119.92	291.02	193	73.53	73.82	237.24	293	42.57	42.3	138.37	392	69.92	69.97	101.57
94	63.37	64.49	175.63	194	50.05	49.01	119.50	294	23.07	23.2	116.40	393	38.77	38.61	202.23
95	49.32	49.12	212.88	195	29.23	29.17	84.10	295	35.99	35.36	142.29	394	55.82	54.6	316.52
96	105.68	104.35	332.07	196	69.54	70.17	186.98	296	18.72	18.8	43.80	395	108.04	105.7	159.87
97	41.8	42.09	134.53	197	38.88	38.6	143.06	297	15.99	16.17	63.85	396	37.42	37.72	236.33
98	36.52	35.98	136.50	198	31.46	31.51	107.06	298	178.87	178.73	438.79	397	62.71	62.91	247.50
99	41.18	40.37	160.33	199	5899.2	5570.6	8240.99	299	175.88	178.61	449.68	398	75.36	72.66	85.57
100	31.33	31.73	171.40	200	63.97	70.77	208.01	300	210.43	210.09	528.98	399	34.17	34.21	94.64

Appendix J

Trials of Learning Algorithm With and Without Heuristic Weight of Spatial Distance for 2D dMRI Dataset

J.1 Learning Algorithm With and Without Heuristic Weight of Spatial Distance for 2D dMRI Dataset

MSEs of Images Registered using Learning Algorithm With and Without Heuristic Weight of Spatial Distance (experiment described in section 5.2.1)

Trial no.	LK	Degraded LK	Without heuristic weight of spatial distance	With heuristic weight of spatial distance
1	93.55	145.52	120	90.41
2	206.17	289.01	250.77	229.66
3	136.65	193.89	176.19	157.01
4	100.84	143.45	128.04	95.99
5	90.89	133.58	87.9	78.79
6	153.87	223.29	177.78	157.86
7	96.30	139.58	121.62	85.59
8	151.23	221.61	168.67	145.68
9	146.99	209.90	156.03	140.21
10	106.67	144.48	152.55	123.00
11	100.85	128.76	121.63	98.19
12	115.93	166.25	168.71	158.84

13	107.85	141.88	122.94	96.28
14	90.83	128.97	140.08	136.74
15	109.11	158.78	119.67	104.63
16	39.66	53.29	88.55	62.11
17	120.54	153.63	166.82	137.63
18	154.17	233.98	172.6	146.97
19	128.31	193.54	135.98	123.69
20	82.14	113.10	125.51	94.75
21	82.14	113.10	137.65	114.39
22	101.25	147.03	120.69	93.56
23	101.60	134.88	140.01	116.81
24	184.06	273.92	215.1	214.42
25	82.14	113.10	124.36	119.53
26	101.25	147.03	139.04	104.53
27	112.29	171.59	122.79	103.42
28	112.78	165.85	153.73	119.94
29	101.75	145.75	157.59	149.21
30	92.25	125.34	124.67	102.58

Appendix K

Trials of Learning Algorithm With Heuristic Weight of Spatial Distance for 3D Gated CT Dataset

K.1 Comparison of Learning Algorithm with and without Heuristic Weight

MSEs of Registered Images Using Learning Algorithm With and Without Heuristic Weight of Spatial Distance

Trial no.	LK	Degraded LK	Without heuristic weight (spatial)	With heuristic weight (spatial)		Trial no.	LK	Degraded LK	Without heuristic weight (spatial)	With heuristic weight (spatial)
1	51.78	51.87	57.94	57.53		201	401.28	636.92	321.01	318.14
2	32.89	33.06	37.04	36.24		202	79.63	78.25	87.1	84.87
3	42.79	42.23	43.85	43.31		203	42.35	42.15	47.2	45.07
4	88.98	89.91	102.02	100.73		204	127.95	127.24	131.59	128.09
5	40.68	41.00	46.03	42.23		205	67.51	67.47	72.05	70.48
6	55.56	54.10	62.66	60.53		206	42.67	42.14	50.4	48.56
7	85.23	82.26	94.68	93.68		207	87.59	87.07	93.06	92.23
8	82.61	80.91	94.33	92.58		208	53.63	53.49	59.51	56.48
9	42.32	41.81	51.8	49.12		209	32.87	32.45	35.65	33.46
10	60.79	61.13	63.61	61.42		210	73.90	73.61	84.39	82.96
11	75.71	74.70	81.51	79.08		211	54.00	53.95	61.95	59.15
12	89.72	87.70	100.96	97.51		212	35.12	33.79	37.81	37.37
13	39.08	39.44	44.88	43.35		213	56.94	56.87	62.52	62.49
14	51.79	51.87	61.93	58.93		214	41.91	42.68	48.96	46.57
15	93.04	93.63	101.3	100.67		215	24.88	24.86	31.8	29.16
16	102.30	100.55	110.09	109.86		216	513.30	886.18	164.49	162.17

17	41.53	41.88	44.74	43.38		217	857.31	1238.20	304.6	300.96
18	59.79	58.93	63.23	59.96		218	86.59	87.01	88.65	86.1
19	72.64	72.92	84.75	83.24		219	48.98	48.64	61.4	59.3
20	88.52	86.83	97.56	93.67		220	111.92	112.94	115.79	114.75
21	43.21	43.22	46.54	44.12		221	54.80	54.64	58.37	58.17
22	65.41	64.22	70.73	69.38		222	45.18	44.66	49.3	46.37
23	30.69	30.61	32.27	28.56		223	92.46	91.41	106.05	105.39
24	33.60	34.01	39.34	35.74		224	50.17	49.57	56.9	55.78
25	97.50	95.54	106.04	102.64		225	35.09	35.63	40.76	39.73
26	60.11	59.22	64.42	63.39		226	82.42	81.16	94.25	92.06
27	65.92	64.44	69.68	68.54		227	38.40	37.71	41.56	39.39
28	93.74	89.67	107.77	104.65		228	36.34	36.23	41.06	37.91
29	66.03	65.10	76.97	74.16		229	66.54	65.57	75.8	72.32
30	59.38	57.41	68.83	66.86		230	41.27	41.36	49.25	46.1
31	110.41	108.75	114.92	111.05		231	28.99	29.27	34.17	30.3
32	57.57	57.21	67.26	65.36		232	4144.00	5467.10	1781.2	1780.48
33	44.97	44.58	53.72	49.74		233	316.89	336.40	293.94	290.22
34	13.77	14.00	14.52	12.56		234	102.93	113.99	85.72	85.54
35	129.92	129.40	139.31	137.3		235	58.42	58.45	65.38	64.42
36	70.24	70.78	83.18	80.11		236	94.17	92.20	103.02	102.99
37	76.86	74.40	84.26	82.71		237	45.11	44.81	46.65	43.96
38	140.83	134.86	155.84	154.03		238	87.76	85.79	90.84	87.22
39	74.72	73.99	87	86.47		239	43.15	43.31	49.52	47.23
40	56.16	56.12	64.96	61.92		240	52.81	52.59	56.95	56.33
41	126.67	124.03	141.26	139		241	63.57	64.05	68.16	66.15
42	79.04	77.25	84.85	82.26		242	41.19	40.72	45.35	43.08
43	66.17	63.65	83.45	80.26		243	32.92	32.39	36.46	35.7
44	108.06	103.11	115.41	114.53		244	58.51	57.62	64.97	63.68
45	43.95	43.83	45.6	42.17		245	34.00	33.93	40.39	37.52
46	43.70	42.35	48.44	44.82		246	29.63	29.51	33.37	31.16
47	63.86	63.57	69.86	68.69		247	1587.80	3113.90	523.4	522.83
48	30.73	30.64	35.57	32.66		248	187.29	893.97	136.79	135.26
49	23.70	23.34	24.97	23.61		249	454.13	848.79	434.58	432.99
50	133.19	129.16	140.09	139		250	94.12	122.74	79.15	76.84
51	72.59	67.25	77.98	77.3		251	70.13	69.64	76.43	76.35
52	149.82	144.35	161.95	159.3		252	109.62	108.05	113	110.69
53	81.31	81.26	90.78	88.63		253	77.57	77.41	83.76	80.03
54	59.82	60.17	67.66	64.35		254	48.77	48.01	53.91	53.48
55	72.94	71.87	83.63	82.56		255	86.82	85.46	91.8	88.88
56	71.75	68.10	89.6	88.89		256	50.30	50.03	55.1	51.22
57	128.65	126.12	143.27	141.55		257	36.55	36.09	41.93	39.5
58	44.72	44.69	48.96	47.06		258	52.61	51.80	61.28	58.4
59	54.80	54.68	65.09	61.95		259	37.32	37.17	46.55	45.34
60	134.30	130.77	149.35	148.83		260	30.51	30.05	38.62	36.79
61	46.57	46.90	49.83	49.62		261	48.03	47.07	55.96	55.77
62	39.71	39.96	43.55	41.04		262	29.85	29.75	37.18	35.64
63	70.62	69.01	81.23	81.11		263	36.17	35.28	44.24	42.79
64	37.64	38.21	44.21	43.67		264	765.44	2593.70	201.63	200.48
65	32.27	32.25	32.63	29.85		265	242.50	1016.10	128.78	125.52
66	172.73	170.31	184.48	182.42		266	89.38	97.77	76.52	74.72
67	90.56	90.63	107.79	105.62		267	113.47	111.80	124.85	121.63
68	74.34	76.14	80.5	77.26		268	70.16	70.28	78.62	75.46
69	74.26	72.24	77.58	74.4		269	40.34	39.25	45.39	44.26
70	86.29	86.46	95.94	93.93		270	95.43	95.04	103.58	103.31
71	78.25	76.02	85.15	84.04		271	58.69	58.75	61	60.78
72	68.89	69.01	82.43	81.95		272	35.79	35.61	40.01	37.46
73	66.79	67.44	74.78	71.24		273	65.89	65.27	70.17	68.47
74	86.83	88.03	94.86	90.98		274	43.12	42.57	45.91	42.29
75	50.81	51.41	60.75	56.98		275	31.97	31.69	37.84	36.17
76	40.09	39.66	45.83	43.28		276	51.49	50.85	55.04	54.43
77	96.43	94.17	104.53	104.16		277	34.80	34.47	42.84	40.68
78	43.70	43.18	44.63	44.33		278	24.65	24.58	28.55	24.8
79	35.59	35.01	38.76	38.03		279	540.89	428.74	551.91	549.27
80	62.27	62.50	65.79	65.67		280	436.53	1206.80	155.15	153.57
81	26.95	26.71	27.85	24.96		281	472.66	1530.60	347.78	346.74
82	18.31	18.38	23.55	22.97		282	131.94	251.03	115.53	112.13
83	231.27	279.15	210.61	208.07		283	156.59	156.46	164.33	160.55

84	324.01	407.17	199.81	196.65		284	88.45	86.36	100.62	99.12
85	271.61	318.32	83.11	80.84		285	62.71	62.11	71.41	71.14
86	152.17	162.12	141.25	139.74		286	118.12	117.71	127.49	126.76
87	84.00	85.21	90.07	86.78		287	69.26	70.17	71.96	69.65
88	112.62	113.49	118.03	116.81		288	36.28	36.61	40.97	40.22
89	73.38	72.51	76.65	75.37		289	81.86	81.29	89.73	88.56
90	139.06	136.15	150.07	146.93		290	55.70	55.44	56.99	55.15
91	72.56	72.83	77.27	75.26		291	33.33	34.02	37.39	36
92	54.58	54.22	65.23	64.19		292	69.54	67.28	77.11	75.84
93	118.80	119.92	134.9	131.97		293	42.57	42.30	47.73	45.89
94	63.37	64.49	71.58	70.93		294	23.07	23.20	24.51	23.57
95	49.32	49.12	59.35	55.66		295	35.99	35.36	46.33	46.22
96	105.68	104.35	122.81	121.92		296	18.72	18.80	24.33	21.7
97	41.80	42.09	47.64	47.31		297	15.99	16.17	20.24	19.6
98	36.52	35.98	44.23	43.94		298	178.87	178.73	181.72	178.51
99	41.18	40.37	48.33	45.25		299	175.88	178.61	181.25	179.61
100	31.33	31.73	34.38	31.11		300	210.43	210.09	220.82	219.51
101	27.54	27.38	31.43	28.47		301	376.91	410.02	291.98	289
102	7.77	7.88	14.3	11.27		302	373.82	1336.30	112.39	109.41
103	267.45	267.65	278.11	274.27		303	853.49	1778.80	246.89	246.19
104	115.52	114.68	119.02	117.16		304	6978.70	12552.00	6188.55	6188.08
105	840.16	1450.60	639.06	635.91		305	115.10	115.47	126.24	125.54
106	778.31	1518.40	442.43	440.74		306	116.17	115.85	124.63	122.12
107	547.22	735.77	263.6	259.83		307	102.02	100.11	111.75	108.38
108	631.96	1570.30	101.23	101.23		308	60.70	59.25	65.35	63.31
109	71.90	76.03	83.39	79.46		309	91.67	90.59	102.46	101.79
110	153.31	148.83	165.58	163.3		310	66.04	64.84	72.57	69.71
111	80.27	78.04	84.86	83.48		311	39.96	39.55	47.96	44.34
112	53.21	54.22	60.71	58.48		312	95.79	94.32	104.32	103.45
113	150.50	150.41	152.15	150.95		313	60.66	60.89	65.87	62.38
114	60.23	60.64	65.91	65.27		314	31.67	31.59	37.73	36.88
115	37.33	36.89	46.31	43.65		315	69.98	69.01	81.24	77.89
116	112.28	111.70	121.57	118.83		316	47.42	47.67	52.35	48.92
117	47.52	47.42	52.52	49.35		317	21.65	21.30	24.95	22.85
118	35.27	35.02	43.27	41.87		318	1.51	1.69	4.59	2.68
119	102.65	102.59	109.26	108.26		319	118.41	119.74	126.7	123.14
120	49.88	49.70	58.92	57.54		320	258.16	221.37	228.22	227.96
121	31.79	32.03	32.33	31.01		321	317.89	350.14	301.5	299.46
122	62.29	61.95	69.23	65.52		322	550.76	2080.50	99.87	97.39
123	41.93	41.78	49.88	46.85		323	1078.90	2094.20	370.44	367.5
124	25.12	25.06	31.24	30.09		324	4707.40	5967.20	1964.57	1963.65
125	185.20	186.99	194.13	191.7		325	100.01	118.53	88.81	88.72
126	373.72	300.85	240.87	237.81		326	123.56	122.08	129.23	128.67
127	136.21	170.00	137.69	134.31		327	82.89	83.67	90.71	87.63
128	401.80	812.35	344.22	340.61		328	95.77	95.63	103.53	99.65
129	196.91	236.89	173.03	170.64		329	69.64	68.79	71.15	69.6
130	473.21	1453.00	187.64	187.36		330	98.20	96.68	107.9	103.92
131	251.65	600.16	219.07	218.2		331	75.79	76.20	87.28	85.98
132	66.67	65.54	72.93	69.45		332	37.60	37.92	44.75	44.2
133	69.49	69.53	70.84	69.18		333	63.36	63.71	74.87	73.33
134	40.28	40.22	46.15	43.5		334	47.18	48.44	54.79	52.54
135	50.30	51.02	57.49	54.36		335	30.88	30.80	37.71	35.17
136	33.94	33.86	35	34.01		336	51.15	51.37	61.59	59.43
137	90.97	90.22	96.14	93.93		337	31.75	32.12	38.15	36.89
138	47.21	47.07	51.53	50.62		338	22.36	22.40	29.55	28.91
139	32.82	32.42	39.55	39.52		339	368.95	547.41	260.46	259.85
140	79.24	80.44	86.95	83.88		340	137.26	174.75	131.99	131.44
141	40.86	40.57	43.17	43.08		341	106.52	105.64	113.8	110.97
142	26.88	26.84	29.03	27.45		342	149.55	148.16	157.05	155.19
143	22.48	22.28	23.58	22.57		343	247.12	1275.30	137.28	136.83
144	15.51	15.61	22.41	21.6		344	106.36	104.82	117.25	114.45
145	13.13	13.01	13.32	10.68		345	118.19	114.65	130.22	129.51
146	247.94	364.91	128.94	125.28		346	102.67	116.92	109.41	106.19
147	1082.00	4112.70	509.48	509.45		347	73.20	71.55	81.81	79.75
148	53.95	54.62	59.55	56.57		348	89.71	89.47	106.89	104.7
149	146.65	147.38	152.35	149.15		349	100.31	98.91	108.42	107.59
150	88.71	88.37	94.6	90.97		350	52.82	53.51	56.47	53.34

151	40.48	40.81	45.22	41.32		351	99.15	97.46	110.26	108.15
152	111.03	112.96	116.56	116.08		352	59.99	60.22	64.37	62.09
153	70.75	71.73	76.1	74.02		353	36.05	35.61	42.65	40.96
154	36.23	35.77	39.37	36.08		354	76.15	76.78	90.39	87.51
155	94.70	92.97	104.99	102.44		355	46.63	46.05	46.99	46.7
156	50.17	48.97	52.44	48.63		356	31.77	31.64	37.01	34.63
157	35.97	35.62	41.86	38.07		357	68.08	67.93	73.02	69.57
158	78.07	77.83	88.39	84.53		358	96.80	95.31	99.3	97.5
159	44.82	44.85	48	47.73		359	86.84	86.98	96.27	93.66
160	29.83	29.61	36.27	34.52		360	71.38	71.44	75.7	74.48
161	55.77	56.06	63.6	62.31		361	66.34	66.56	67.84	65.41
162	36.90	36.63	39.77	39.23		362	98.63	97.19	103.57	102.46
163	24.07	24.22	28.37	27.83		363	71.35	73.45	80.45	77.25
164	861.97	2765.10	166.02	162.79		364	75.64	74.81	87.07	83.89
165	1264.50	5293.10	701.72	699.62		365	76.09	76.13	84.59	80.78
166	155.47	158.75	165.96	162.19		366	85.74	83.88	97.16	95.38
167	145.27	141.29	164.21	160.25		367	52.71	52.93	64.22	62.39
168	44.40	44.60	51.5	49.86		368	66.44	65.55	69.29	66.89
169	100.61	100.28	108.6	107.11		369	65.74	65.13	73.43	70.06
170	94.60	97.14	101.92	101.01		370	43.13	43.03	45.32	45.19
171	38.48	39.24	44.85	43.07		371	222.86	221.87	235.35	234.6
172	92.99	90.42	105.71	104.64		372	56.79	56.52	65.38	61.6
173	66.36	66.22	80.48	78.64		373	63.22	62.76	69.45	65.66
174	36.06	35.94	41.48	39.75		374	55.14	55.60	62.38	60.57
175	75.59	74.22	79.57	78.54		375	56.66	56.45	60.7	57.46
176	57.81	55.21	72.76	72.22		376	99.11	104.78	111.95	108.23
177	31.27	31.05	32.86	31.18		377	52.11	52.17	60.14	57.45
178	63.52	63.51	67.77	65.74		378	63.84	62.58	69.73	68.24
179	45.40	44.78	53.1	51.8		379	100.83	100.94	110.34	108.72
180	27.29	27.21	28.34	25.6		380	59.02	57.92	62.6	60.84
181	6684.00	8097.40	4698.2	4696.43		381	69.90	69.20	76.87	74.15
182	1771.50	5466.70	167.05	165.3		382	83.84	82.03	90.61	88.75
183	4186.00	3743.10	3190.17	3187		383	58.44	58.10	66.36	62.55
184	720.51	3116.10	298.13	294.87		384	57.98	57.71	68.78	67.36
185	102.88	101.09	112	108.99		385	36.15	35.81	44.47	43.12
186	44.20	44.10	48.02	44.87		386	73.60	70.94	80.8	77.21
187	104.59	104.82	112.91	110.91		387	52.67	52.83	58.03	55.84
188	67.71	66.84	75.58	73.35		388	62.91	63.29	66.45	63.45
189	39.77	39.96	46.78	44.26		389	90.59	89.39	104.61	104.11
190	93.80	92.42	105.41	105.02		390	57.30	57.75	63.86	62.04
191	63.04	62.31	70.42	69.44		391	51.87	52.42	56.09	55.79
192	31.77	31.63	35.13	32.67		392	69.92	69.97	78.67	76.01
193	73.53	73.82	80.27	79.05		393	38.77	38.61	42.81	39.99
194	50.05	49.01	59.25	56.18		394	55.82	54.60	59.48	55.8
195	29.23	29.17	31.49	30.42		395	108.04	105.70	121.66	119.02
196	69.54	70.17	72.32	72.16		396	37.42	37.72	45.1	42.34
197	38.88	38.60	44	42.82		397	62.71	62.91	67.85	64.43
198	31.46	31.51	38.24	36.01		398	75.36	72.66	79.83	77.96
199	5899.20	5570.60	3862.82	3858.94		399	34.17	34.21	37.75	35.91
200	63.97	70.77	63.39	60.63		400	7.03	6.61	8.52	5.3

Appendix L

Trials of Learning Algorithm With and Without Heuristic Weight of Intensity for 2D dMRI dataset

L.1 Learning Algorithm With and Without Heuristic Weight of Intensity for 2D dMRI dataset

MSEs of Images Registered using Learning Algorithm With and Without Heuristic Weight of Intensity (experiment described in section 5.3)

Trial no.	LK	Degraded LK	Without heuristic weight of intensity	With heuristic weight of intensity
1	93.55	145.52	90.41	96.30
2	206.17	289.01	229.66	209.79
3	136.65	193.89	157.01	133.95
4	100.84	143.45	95.99	101.52
5	90.89	133.58	78.79	86.70
6	153.87	223.29	157.86	157.92
7	96.30	139.58	85.59	96.18
8	151.23	221.61	145.68	152.47
9	146.99	209.90	140.21	144.21
10	106.67	144.48	123.00	103.23
11	100.85	128.76	98.19	95.83
12	115.93	166.25	158.84	115.43
13	107.85	141.88	96.28	99.48
14	90.83	128.97	136.74	93.94
15	109.11	158.78	104.63	95.85
16	39.66	53.29	62.11	40.01
17	120.54	153.63	137.63	121.52

18	154.17	233.98	146.97	149.98
19	128.31	193.54	123.69	123.02
20	82.14	113.10	94.75	78.11
21	82.14	113.10	114.39	80.71
22	101.25	147.03	93.56	101.01
23	101.60	134.88	116.81	99.75
24	184.06	273.92	214.42	188.76
25	82.14	113.10	119.53	74.32
26	101.25	147.03	104.53	92.90
27	112.29	171.59	103.42	112.77
28	112.78	165.85	119.94	102.01
29	101.75	145.75	149.21	98.19
30	92.25	125.34	102.58	93.83

Appendix M

Trials of Learning Algorithm with One Iteration and Two Iterations for 2D dMRI dataset

M.1 Learning Algorithm with One Iteration and Two Iterations for 2D dMRI dataset

MSEs of Images Registered Using Learning Algorithm with One Iteration and Two Iterations (experiment described in section 5.4.1)

Trial no.	LK	Degraded LK	With one iteration	With two iterations
1	93.55	145.52	96.30	90.017
2	206.17	289.01	209.79	212.05
3	136.65	193.89	133.95	145.2
4	100.84	143.45	101.52	100.62
5	90.89	133.58	86.70	91.87
6	153.87	223.29	157.92	158.22
7	96.30	139.58	96.18	96.39
8	151.23	221.61	152.47	151.65
9	146.99	209.90	144.21	144.68
10	106.67	144.48	103.23	103.67
11	100.85	128.76	95.83	90.54
12	115.93	166.25	115.43	117.14
13	107.85	141.88	99.48	98.42
14	90.83	128.97	93.94	93.59
15	109.11	158.78	95.85	95.91
16	39.66	53.29	40.01	42.86
17	120.54	153.63	121.52	115.84
18	154.17	233.98	149.98	150.13
19	128.31	193.54	123.02	124.23
20	82.14	113.10	78.11	76.835
21	82.14	113.10	80.71	81.412

22	101.25	147.03	101.01	101.39
23	101.60	134.88	99.75	92.77
24	184.06	273.92	188.76	188.47
25	82.14	113.10	74.32	76.26
26	101.25	147.03	92.90	91.34
27	112.29	171.59	112.77	113.23
28	112.78	165.85	102.01	103.30
29	101.75	145.75	98.19	99.19
30	92.25	125.34	93.83	102.32

Appendix N

Trials of Learning Algorithm with Iteration for 3D Gated CT Dataset

N.1 Comparison of Learning Algorithm With and Without Iteration

MSEs of Registered Images Using Learning Algorithm With and Without Iteration

Trial no.	LK	Degraded LK	Without iteration	With iteration		Trial no.	LK	Degraded LK	Without iteration	With iteration
1	51.778	51.87	54.526	53.317		201	401.28	636.92	320.46	318.71
2	32.894	33.055	33.792	33.808		202	79.626	78.25	80.57	80.497
3	42.785	42.226	42.933	43.072		203	42.35	42.151	42.983	42.983
4	88.983	89.909	95.819	96.8		204	127.95	127.24	128.95	128.97
5	40.676	40.997	41.312	41.283		205	67.506	67.466	67.399	68.024
6	55.56	54.098	56.808	56.736		206	42.666	42.137	44.845	44.845
7	85.23	82.255	90.076	89.854		207	87.59	87.066	90.728	90.728
8	82.606	80.914	87.436	85.503		208	53.625	53.492	54.657	54.87
9	42.315	41.814	44.939	44.948		209	32.867	32.446	34.22	34.22
10	60.793	61.131	61.859	61.859		210	73.902	73.608	77.683	77.683
11	75.714	74.699	77.144	77.668		211	53.998	53.951	56.974	57.028
12	89.72	87.704	95.863	95.68		212	35.115	33.792	36.644	36.644
13	39.078	39.437	41.385	41.385		213	56.942	56.87	59.418	59.418
14	51.789	51.867	55.98	56.07		214	41.907	42.677	44.533	44.528
15	93.042	93.628	99.962	100.06		215	24.88	24.863	25.287	25.287
16	102.3	100.55	109.22	109.37		216	513.3	886.18	160.78	160.71
17	41.534	41.884	44.716	44.728		217	857.31	1238.2	300.21	300.21
18	59.787	58.925	62.159	62.299		218	86.586	87.005	83.879	83.879
19	72.637	72.918	81.005	81.024		219	48.975	48.637	54.937	54.937
20	88.52	86.825	93.986	93.61		220	111.92	112.94	114.72	114.72
21	43.214	43.223	43.844	43.934		221	54.803	54.638	55.529	55.529
22	65.407	64.216	68.56	68.685		222	45.177	44.656	47.107	47.091
23	30.689	30.611	32.254	31.679		223	92.457	91.414	101.19	101.19
24	33.598	34.01	33.628	34.819		224	50.168	49.567	50.671	51.556
25	97.503	95.536	101.57	101.63		225	35.089	35.631	37.332	37.332
26	60.106	59.218	61.279	61.5		226	82.421	81.16	88.609	88.609
27	65.916	64.436	67.97	67.97		227	38.401	37.712	39.274	39.121

28	93.74	89.671	102.15	101.92		228	36.336	36.233	37.205	37.146
29	66.026	65.098	71.196	70.907		229	66.537	65.567	73.083	73.409
30	59.38	57.409	62.861	63.318		230	41.274	41.356	42.965	42.965
31	110.41	108.75	111.65	113.38		231	28.986	29.269	29.439	29.439
32	57.572	57.206	60.47	60.47		232	4144	5467.1	1775.4	1710
33	44.969	44.576	47.829	47.803		233	316.89	336.4	293.17	293.17
34	13.769	13.999	13.966	13.966		234	102.93	113.99	83.773	83.664
35	129.92	129.4	137.65	136.06		235	58.416	58.452	60.006	60.263
36	70.236	70.781	77.462	77.446		236	94.168	92.204	101.51	101.51
37	76.858	74.395	81.42	80.888		237	45.113	44.811	46.411	46.411
38	140.83	134.86	152.58	153.87		238	87.759	85.788	87.779	87.779
39	74.72	73.988	80.343	80.313		239	43.145	43.307	42.957	42.978
40	56.159	56.121	58.201	58.352		240	52.81	52.586	55.115	55.115
41	126.67	124.03	135.9	136.37		241	63.57	64.053	64.174	64.066
42	79.035	77.249	80.825	80.825		242	41.186	40.724	42.825	42.825
43	66.167	63.646	77.04	77.569		243	32.92	32.385	36.27	36.27
44	108.06	103.11	111.94	112.9		244	58.508	57.617	61.468	61.468
45	43.948	43.834	44.438	44.372		245	34.004	33.926	34.599	34.528
46	43.699	42.354	46.161	45.749		246	29.628	29.513	31.561	31.561
47	63.858	63.574	67.78	67.881		247	1587.8	3113.9	523.08	596.01
48	30.728	30.643	31.659	31.774		248	187.29	893.97	135.06	139.01
49	23.701	23.341	24.495	24.488		249	454.13	848.79	429.95	429.95
50	133.19	129.16	139.61	140.79		250	94.115	122.74	76.836	76.836
51	72.593	67.252	76.813	76.813		251	70.128	69.641	71.807	71.995
52	149.82	144.35	155.32	155.72		252	109.62	108.05	112.91	112.91
53	81.312	81.256	85.1	85.119		253	77.568	77.41	78.727	78.727
54	59.822	60.174	62.688	62.688		254	48.767	48.009	51.17	51.17
55	72.942	71.873	76.838	77.234		255	86.819	85.458	91.567	91.567
56	71.748	68.099	82.613	82.561		256	50.297	50.034	52.255	52.318
57	128.65	126.12	136.36	136.37		257	36.546	36.09	36.922	36.892
58	44.719	44.693	47.914	47.918		258	52.606	51.802	54.832	54.963
59	54.799	54.679	58.381	58.498		259	37.319	37.166	39.659	39.659
60	134.3	130.77	145.64	145.45		260	30.505	30.052	31.736	31.736
61	46.571	46.901	49.312	49.312		261	48.028	47.069	49.694	49.617
62	39.707	39.956	41.374	41.374		262	29.85	29.746	31.117	31.117
63	70.622	69.01	74.957	72.712		263	36.166	35.275	38.628	38.628
64	37.642	38.209	38.372	38.366		264	765.44	2593.7	197.75	196.66
65	32.269	32.253	32.609	32.6		265	242.5	1016.1	125.85	125.82
66	172.73	170.31	180	177.23		266	89.381	97.771	75.626	75.626
67	90.56	90.627	102.17	90.393		267	113.47	111.8	120.27	120.27
68	74.335	76.137	78.777	79.256		268	70.159	70.28	72.569	72.569
69	74.258	72.237	77.134	77.134		269	40.335	39.249	43.468	43.468
70	86.289	86.464	94.101	94.101		270	95.427	95.035	97.704	97.704
71	78.252	76.021	84.425	84.402		271	58.694	58.749	60.503	60.468
72	68.885	69.012	79.037	79.037		272	35.789	35.614	37.356	37.356
73	66.792	67.442	71.853	71.716		273	65.888	65.269	68.285	68.285
74	86.829	88.025	92.193	91.856		274	43.119	42.574	44.843	44.762
75	50.806	51.41	54.539	54.649		275	31.965	31.685	33.422	33.422
76	40.086	39.657	42.891	42.891		276	51.494	50.85	52.83	52.374
77	96.432	94.166	102.54	102.66		277	34.802	34.469	36.132	36.128
78	43.701	43.177	44.287	44.287		278	24.646	24.583	25.059	25.059
79	35.59	35.014	37.231	37.348		279	540.89	428.74	546.74	543.7
80	62.271	62.499	64.12	63.885		280	436.53	1206.8	155.06	155.06
81	26.945	26.708	27.652	27.597		281	472.66	1530.6	343.54	343.44
82	18.308	18.379	18.629	18.564		282	131.94	251.03	111.49	111.17
83	231.27	279.15	210.56	208.25		283	156.59	156.46	158.68	158.68
84	324.01	407.17	195.53	195.21		284	88.446	86.362	96.044	96.044
85	271.61	318.32	80.249	80.298		285	62.711	62.113	65.259	65.259
86	152.17	162.12	139.51	140.76		286	118.12	117.71	121.17	121.17
87	83.999	85.209	85.499	85.587		287	69.258	70.173	70.891	70.891
88	112.62	113.49	115.79	115.79		288	36.277	36.605	39.615	39.708
89	73.381	72.51	75.922	76.066		289	81.855	81.288	84.193	84.282
90	139.06	136.15	146.32	146.32		290	55.696	55.443	56.569	56.569
91	72.555	72.83	76.116	76.116		291	33.334	34.016	34.655	34.705
92	54.58	54.216	59.046	58.826		292	69.541	67.283	75.007	75.007
93	118.8	119.92	130.23	130.2		293	42.574	42.303	42.593	42.764
94	63.37	64.494	65.652	65.147		294	23.073	23.201	23.775	23.761

95	49.323	49.124	54.01	54.01		295	35.99	35.358	40.784	40.791
96	105.68	104.35	117.16	117.76		296	18.718	18.799	18.848	18.834
97	41.799	42.092	43.214	43.214		297	15.99	16.169	16.513	16.464
98	36.517	35.98	39.262	39.262		298	178.87	178.73	179.95	180.06
99	41.183	40.373	43.506	43.506		299	175.88	178.61	180.75	180.71
100	31.331	31.726	32.126	32.117		300	210.43	210.09	216.44	215.92
101	27.543	27.38	27.709	28.001		301	376.91	410.02	291.81	291.85
102	7.7697	7.8756	8.1933	7.8808		302	373.82	1336.3	111.96	111.75
103	267.45	267.65	277.73	277.73		303	853.49	1778.8	245.98	245.98
104	115.52	114.68	115.52	115.49		304	6978.7	12552	6185.4	6185.1
105	840.16	1450.6	636.03	636.91		305	115.1	115.47	121.53	121.52
106	778.31	1518.4	436.1	399.8		306	116.17	115.85	118.64	118.64
107	547.22	735.77	259.19	259.19		307	102.02	100.11	108.26	108.33
108	631.96	1570.3	94.351	94.036		308	60.696	59.251	65.012	65.012
109	71.9	76.031	79.286	82.302		309	91.67	90.591	100.26	99.781
110	153.31	148.83	159.7	160.05		310	66.039	64.844	68.076	68.076
111	80.265	78.035	81.58	81.58		311	39.961	39.553	42.462	42.462
112	53.209	54.216	56.89	56.897		312	95.788	94.322	102.3	102.29
113	150.5	150.41	150.9	150.67		313	60.657	60.892	62.378	62.58
114	60.227	60.638	61.468	61.452		314	31.674	31.59	32.001	32.001
115	37.332	36.893	39.572	39.572		315	69.983	69.013	77.071	77.071
116	112.28	111.7	117.83	117.87		316	47.417	47.673	48.597	48.716
117	47.519	47.42	49.161	48.39		317	21.646	21.304	22.633	22.633
118	35.273	35.024	37.71	37.71		318	1.5052	1.6892	1.7054	1.7054
119	102.65	102.59	108.61	108.31		319	118.41	119.74	121.14	120.94
120	49.88	49.701	52.749	52.749		320	258.16	221.37	225.82	225.39
121	31.794	32.025	32.3	32.285		321	317.89	350.14	298.26	298.56
122	62.285	61.95	65.653	65.442		322	550.76	2080.5	97.297	97.404
123	41.928	41.776	45.127	45.152		323	1078.9	2094.2	365.68	382.57
124	25.117	25.062	27.277	27.277		324	4707.4	5967.2	1960.6	1885.4
125	185.2	186.99	190.78	191		325	100.01	118.53	84.251	84.42
126	373.72	300.85	238.63	238.67		326	123.56	122.08	125.79	125.79
127	136.21	170	133.48	132.7		327	82.887	83.666	87.918	87.918
128	401.8	812.35	337.83	337.55		328	95.767	95.633	100.19	100.3
129	196.91	236.89	168.25	168.25		329	69.635	68.79	70.68	70.999
130	473.21	1453	181.01	180.43		330	98.198	96.683	105.02	106.21
131	251.65	600.16	218.38	218.38		331	75.786	76.198	80.501	80.501
132	66.674	65.541	69.347	69.347		332	37.596	37.92	39.277	39.277
133	69.493	69.532	70.07	70.065		333	63.359	63.708	69.768	70.086
134	40.284	40.224	42.334	42.298		334	47.177	48.436	49.429	49.429
135	50.302	51.023	52.674	52.683		335	30.88	30.8	32.411	32.411
136	33.938	33.864	33.969	33.969		336	51.149	51.369	55.69	56.44
137	90.971	90.216	90.697	90.697		337	31.752	32.117	32.762	32.829
138	47.205	47.065	48.741	48.598		338	22.356	22.396	22.696	22.723
139	32.822	32.42	33.259	33.212		339	368.95	547.41	259.68	259.48
140	79.242	80.443	84.804	84.804		340	137.26	174.75	129.22	128.42
141	40.862	40.566	42.736	42.736		341	106.52	105.64	110.36	109.79
142	26.879	26.836	27.491	27.499		342	149.55	148.16	155.24	155.24
143	22.476	22.278	22.996	22.996		343	247.12	1275.3	137.02	137.02
144	15.505	15.607	15.763	16.014		344	106.36	104.82	110.43	110.43
145	13.13	13.009	13.205	13.189		345	118.19	114.65	125.14	126.22
146	247.94	364.91	128.14	127.31		346	102.67	116.92	108.37	108.37
147	1082	4112.7	509.39	509.91		347	73.201	71.552	80.77	80.766
148	53.95	54.624	58.039	58.144		348	89.705	89.469	101.96	102.06
149	146.65	147.38	152.27	152.4		349	100.31	98.914	105.75	105.75
150	88.707	88.374	90.095	90.095		350	52.815	53.506	55.942	55.942
151	40.48	40.809	41.598	41.598		351	99.145	97.459	107.38	107.38
152	111.03	112.96	114.84	114.84		352	59.987	60.224	63.367	63.377
153	70.748	71.729	74.741	74.741		353	36.045	35.614	37.057	37.057
154	36.229	35.773	38.731	38.764		354	76.149	76.784	83.879	83.922
155	94.701	92.974	102.41	102.36		355	46.631	46.051	46.955	46.955
156	50.168	48.967	52.385	52.257		356	31.772	31.636	32.458	32.458
157	35.974	35.617	37.642	37.575		357	68.078	67.932	68.274	68.253
158	78.067	77.834	85.044	85.044		358	96.801	95.306	97.524	97.141
159	44.822	44.851	45.841	45.841		359	86.839	86.978	90.372	90.372
160	29.833	29.609	31.057	31.057		360	71.382	71.439	73.635	73.606
161	55.772	56.056	57.715	57.856		361	66.34	66.557	67.652	67.652

162	36.899	36.633	37.925	37.948		362	98.628	97.186	102.92	102.77
163	24.07	24.22	24.773	24.785		363	71.346	73.449	74.856	74.856
164	861.97	2765.1	162.89	162.89		364	75.637	74.806	82.085	82.085
165	1264.5	5293.1	699.33	699.37		365	76.089	76.127	79.105	78.918
166	155.47	158.75	160.09	160.09		366	85.739	83.883	92.789	92.789
167	145.27	141.29	157.33	157.26		367	52.708	52.926	58.437	58.437
168	44.396	44.604	47.107	47.107		368	66.438	65.549	69.043	69.043
169	100.61	100.28	107.33	107.33		369	65.741	65.134	70.593	70.573
170	94.604	97.139	101.06	101.06		370	43.127	43.03	43.565	43.565
171	38.479	39.235	40.793	40.793		371	222.86	221.87	231.98	231.98
172	92.985	90.422	103.41	103.41		372	56.79	56.517	59.207	59.024
173	66.358	66.216	78.6	78.814		373	63.215	62.756	67.486	67.039
174	36.061	35.943	37.626	37.682		374	55.138	55.599	58.188	58.179
175	75.589	74.223	78.311	78.375		375	56.664	56.454	60.523	60.523
176	57.812	55.205	68.013	68.021		376	99.107	104.78	110.86	110.85
177	31.274	31.053	32.471	32.471		377	52.106	52.166	54.302	54.455
178	63.516	63.512	67.527	67.465		378	63.838	62.579	68.373	68.083
179	45.402	44.779	51.092	51.092		379	100.83	100.94	104.53	104.39
180	27.294	27.208	27.804	27.804		380	59.018	57.924	60.23	60.23
181	6684	8097.4	4691.9	4691.9		381	69.895	69.203	72.167	72.042
182	1771.5	5466.7	161.12	161.03		382	83.843	82.03	90.243	90.243
183	4186	3743.1	3187.4	3225.1		383	58.438	58.1	61.217	61.217
184	720.51	3116.1	296.95	297.25		384	57.98	57.713	65.283	65.283
185	102.88	101.09	108.99	109.01		385	36.147	35.806	37.873	37.873
186	44.197	44.099	45.106	44.967		386	73.603	70.936	78.77	78.858
187	104.59	104.82	107.81	108.11		387	52.672	52.833	55.387	55.387
188	67.709	66.839	72.729	72.658		388	62.906	63.291	65.65	66.699
189	39.774	39.957	40.117	40.117		389	90.588	89.391	97.861	97.861
190	93.8	92.422	99.032	99.315		390	57.303	57.746	60.831	60.828
191	63.035	62.31	63.76	63.753		391	51.87	52.418	55.501	55.769
192	31.77	31.627	32.714	32.714		392	69.922	69.97	73.645	73.588
193	73.527	73.815	78.236	78.236		393	38.767	38.608	39.255	43.153
194	50.052	49.013	53.044	53.044		394	55.824	54.598	57.183	57.443
195	29.233	29.168	30.02	30.02		395	108.04	105.7	116.39	115.8
196	69.537	70.167	71.399	71.399		396	37.423	37.72	39.251	39.263
197	38.88	38.597	40.361	40.407		397	62.714	62.907	66.066	65.987
198	31.46	31.514	31.898	31.924		398	75.359	72.657	76.09	80.256
199	5899.2	5570.6	3860	3859.1		399	34.171	34.21	34.696	36.083
200	63.967	70.766	61.875	61.852		400	7.0318	6.6065	7.4213	7.4213

Appendix O

Learning Algorithm With MPI-Sintel Dataset

O.1 Comparing Learning Algorithm with Different Optical Flow Methods

Average Angular Errors and Average Endpoint Erros of Learning Algorithm and Other Optical Methods with MPI-Sintel Dataset; the learning set for each pair was consisted of immediately preceeding three frames of the source frame.

Trial#	Sequence	Source	Target	Average Angular Error (AAE)					Average Endpoint Error (AEPE)				
				HS	BA	LDOF	DF	LA	HS	BA	LDOF	DF	LA
1	alley_1	28	30	2.6752	2.6752	2.2580	1.9797	2.1551	0.3138	0.3138	0.2829	0.2403	0.2420
2		29	31	2.6967	2.6967	2.1157	1.9553	2.1157	0.2522	0.2522	0.2180	0.1926	0.1968
3		30	32	3.0423	3.0423	2.4501	2.3313	2.5545	0.2229	0.2229	0.1870	0.1710	0.1817
4		31	33	3.2033	3.2033	3.0219	2.8566	2.9540	0.1860	0.1860	0.1674	0.1495	0.1604
5		32	34	2.9226	2.9226	2.2239	1.9637	2.3967	0.1595	0.1595	0.1312	0.1123	0.1316
6		33	35	2.3563	2.3563	1.6822	1.4903	1.9035	0.1525	0.1525	0.1230	0.1067	0.1251
7		34	36	1.5247	1.5247	1.0191	0.8864	1.1295	0.1288	0.1288	0.1035	0.0903	0.1026
8		35	37	1.3135	1.3135	0.8622	0.7489	0.9945	0.1196	0.1196	0.0980	0.0850	0.1008
9		36	38	1.2493	1.2493	0.8515	0.7104	0.9264	0.1212	0.1212	0.1019	0.0869	0.0992
10	ambush_7	15	17	17.8010	8.1898	13.1910	6.3565	8.8675	1.0952	0.4340	0.4606	0.4137	0.4058
11		16	18	15.8500	7.8548	6.6756	4.8916	7.2717	1.0262	0.3187	0.2477	0.3215	0.2950
12		17	19	16.5070	8.2334	7.0972	5.3116	7.5000	1.0526	0.3307	0.2509	0.3426	0.3215
13		18	20	16.4460	8.2076	7.5605	5.3001	7.6188	1.0233	0.3128	0.2551	0.2745	0.3302
14		19	21	17.2860	7.0973	6.7353	3.9372	6.4951	1.0034	0.2687	0.2302	0.2187	0.2862
15		20	22	18.3830	6.6956	6.4405	4.3420	5.6891	0.9998	0.2538	0.2389	0.1840	0.2358
16		21	23	17.6850	7.0990	7.6519	4.9099	6.0014	0.8784	0.3033	0.2597	0.2421	0.2284
17		22	24	16.1080	7.3990	6.3084	4.5098	6.8539	0.7893	0.3152	0.2868	0.2696	0.3470
18		23	25	18.4790	7.9696	10.2990	5.2364	7.5990	0.6331	0.2502	0.3038	0.1856	0.2651
19		26	28	29.1760	10.1800	14.9190	8.6563	11.0080	1.6739	0.5239	0.6286	0.4181	0.4771
20		27	29	29.3060	9.6347	20.3780	9.3846	10.8110	1.6797	0.4938	0.7798	0.4182	0.4397
21		28	30	31.8950	10.8500	30.1010	9.7019	12.5500	2.1071	0.6151	1.4242	0.5223	0.5921
22		34	36	31.2580	11.9040	43.0980	9.3270	11.3320	2.3457	1.0699	5.3070	0.5661	0.5368
23		35	37	29.6470	11.0740	43.2010	8.5383	9.7787	2.0817	0.9781	5.2006	0.5293	0.4832

24		36	38	37.4730	27.5020	46.9570	13.3570	14.8270	4.8441	5.4609	7.3664	1.5868	1.2334
25	bamboo_1	28	30	7.8433	4.1872	4.1808	4.2473	4.2195	1.6543	0.8295	0.8668	0.8478	0.8848
26		29	31	6.2366	4.2737	3.5729	4.1994	4.0295	1.7181	0.8393	0.8141	0.8401	0.8268
27		30	32	6.3258	3.9054	3.2694	3.8130	3.6109	1.1716	0.7949	0.7282	0.7778	0.7805
28		31	33	7.3774	4.8140	3.5273	4.2727	4.5941	1.3841	0.9902	0.8386	0.9307	0.9731
29		32	34	7.0744	5.3139	4.2774	4.5239	4.6683	1.4102	1.0087	0.9057	0.9218	0.9564
30		33	35	7.2357	5.0702	3.8350	4.5168	4.7110	1.4709	0.9442	0.8253	0.8986	0.9250
31		34	36	7.9318	4.8660	4.4591	4.7085	4.9161	1.3899	0.8671	0.8363	0.8648	0.8848
32		35	37	7.2389	4.4036	3.9589	4.2420	4.2096	1.3782	0.7934	0.7426	0.7923	0.8030
33		36	38	7.0075	4.3896	3.6807	4.1169	4.1713	1.3959	0.7940	0.7337	0.7681	0.7588
34		37	39	7.9344	5.1683	3.9999	4.4287	4.7247	1.6771	0.9125	0.8621	0.8370	0.8932
35	bamboo_2	3	5	10.8430	6.5172	6.2946	7.3416	5.8664	1.2506	0.7819	0.8008	0.8760	0.7629
36		4	6	9.5300	7.1071	6.6859	6.9904	6.7264	0.9579	0.7162	0.6734	0.7356	0.7085
37		5	7	8.3547	5.2857	4.9035	4.9991	4.7846	0.5171	0.2939	0.2795	0.2799	0.2801
38		6	8	8.7102	5.0999	4.5279	4.8583	4.5194	0.5488	0.2860	0.2559	0.2653	0.2737
39	market_2	28	30	14.2190	9.3888	8.2933	7.9794	7.5666	1.8444	1.6468	1.5630	1.5206	1.1659
40		29	31	11.9100	7.7938	7.0036	6.6312	6.9180	1.3002	0.9500	0.9902	0.7728	0.8306
41		30	32	11.5030	7.2886	6.5801	6.1274	6.7413	1.1134	0.8757	0.8363	0.7245	0.7567
42		36	38	9.8715	6.1863	5.2970	5.5750	5.6363	0.6697	0.6017	0.5881	0.6024	0.5066
43		37	39	8.5430	5.4319	4.6965	4.7717	5.1845	0.3275	0.2096	0.1841	0.1852	0.1985
44		38	40	8.0559	5.2292	4.3968	4.4877	4.6894	0.2889	0.1891	0.1633	0.1683	0.1737
45		39	41	8.0153	5.0060	4.3103	4.3715	4.5289	0.2924	0.1872	0.1635	0.1678	0.1736
46		40	42	8.3753	5.1725	4.3928	4.2750	4.5511	0.3228	0.1971	0.1677	0.1658	0.1750
47		41	43	8.3358	5.2234	4.3082	4.3701	4.7134	0.3409	0.2109	0.1795	0.1830	0.1926
48		42	44	8.3632	5.6311	4.6347	4.6604	5.0775	0.3944	0.2479	0.2222	0.2133	0.2216
49		43	45	8.8124	5.5541	4.6574	4.7823	5.0473	0.4454	0.2570	0.2290	0.2295	0.2311
50		44	46	9.0352	5.7680	4.7425	4.8479	5.1216	0.4882	0.2887	0.2400	0.2418	0.2470
51		45	47	8.7450	5.6535	4.6116	4.6810	5.0043	0.4230	0.2600	0.2124	0.2168	0.2301
52		46	48	9.4195	5.7510	4.9463	5.0537	5.3232	0.4756	0.2515	0.2326	0.2284	0.2369
53		47	49	9.5033	6.0625	5.4353	5.5531	5.7413	0.4781	0.2727	0.2530	0.2582	0.2673
54	market_6	10	12	23.8820	16.5870	8.6951	17.6920	9.3539	29.1520	26.4410	20.9150	20.9150	17.8530
55		11	13	24.7320	10.7990	10.1160	13.4720	11.2890	27.9890	21.0980	22.2990	22.2990	21.2110
56		12	14	27.3700	11.2150	13.4240	9.1220	12.5390	30.4530	20.2730	23.0670	23.0670	20.3490
57		13	15	16.3550	12.7760	14.1710	9.4032	8.1656	19.6990	17.8500	20.3880	20.3880	12.4460
58		14	16	11.6270	11.0610	10.8530	11.1990	7.0946	14.0640	14.0680	15.6610	15.6610	10.3560
59		15	17	17.5760	9.3579	7.6542	8.0173	7.0487	15.1570	9.0872	9.5359	9.5359	7.2250
60		16	18	15.7920	6.6652	6.1895	5.3838	4.8939	12.0320	7.1127	7.2171	7.2171	5.0453
61		17	19	16.9510	6.0423	6.7421	4.9935	4.8455	13.4620	6.3187	7.3686	7.3686	4.4070
62		18	20	17.8960	6.1430	5.9627	8.3143	4.7285	14.2680	6.3002	6.2863	6.2863	4.5158
63		19	21	10.8170	6.3045	5.0586	4.4805	4.4007	9.1580	5.6627	4.8660	4.8660	4.2750
64		20	22	14.4310	5.9356	4.7298	4.9381	4.3999	10.1660	5.2550	4.2027	4.2027	4.1513
65		21	23	7.8090	6.5890	4.8954	4.8082	4.3362	6.2336	5.2899	3.6274	3.6274	3.2796
66		22	24	8.0475	6.6058	4.8571	4.7549	4.1452	5.5721	4.7069	3.3537	3.3537	3.0976
67		23	25	11.9130	8.1465	5.9118	6.8977	5.2055	6.7209	5.5283	4.0074	4.0074	3.5354
68		24	26	10.5120	9.4274	6.1424	6.2904	6.6398	6.3551	5.5391	3.3698	3.3698	3.5581
69		25	27	22.6660	14.9790	8.4009	14.2370	11.2570	15.4320	10.4520	6.4898	6.4898	7.9621
70	shaman_2	3	5	3.7132	2.5297	2.3782	2.4205	2.3663	0.3029	0.2367	0.2260	0.2261	0.2135
71		4	6	3.7371	2.4645	2.4500	2.4156	2.3867	0.2489	0.1853	0.1967	0.1887	0.1806
72		5	7	3.6413	2.5798	2.4513	2.4835	2.4657	0.2202	0.1716	0.1673	0.1621	0.1607
73		6	8	4.0927	3.1359	2.8979	2.8751	2.9860	0.2613	0.2137	0.2029	0.1883	0.1944
74		7	9	4.9502	4.0923	3.9288	3.8127	4.1502	0.3826	0.3400	0.3281	0.3108	0.3253
75		38	40	8.3602	5.4474	5.9433	6.7021	6.0057	0.4544	0.3249	0.3246	0.3294	0.3198
76		39	41	8.7507	5.6011	5.9636	6.6802	6.1899	0.4555	0.3102	0.3053	0.3244	0.3181
77		40	42	9.6258	5.9717	6.3237	7.1598	6.8400	0.4429	0.2946	0.2870	0.3038	0.3078
78		43	45	9.1029	6.5714	6.1061	6.0824	6.1948	0.3019	0.2249	0.1967	0.1972	0.2021
79		44	46	8.1659	5.9874	5.4676	5.4519	5.7113	0.2470	0.1882	0.1629	0.1629	0.1700
80		45	47	8.0631	5.9388	5.3977	5.3783	5.7516	0.2486	0.1853	0.1681	0.1650	0.1721
81		46	48	7.5197	5.4968	5.5251	5.0491	5.3649	0.2942	0.2314	0.2233	0.2079	0.2128
82		47	49	7.7900	5.3964	5.8603	5.5146	5.5205	0.4046	0.3173	0.3129	0.3018	0.2898
83	shaman_3	37	39	10.0170	6.1719	9.0703	5.2795	5.4207	1.9320	0.9698	1.3507	0.8574	0.8427

84		38	40	12.2230	5.8646	7.6231	5.4704	5.4806	2.8896	0.9953	1.2626	0.9416	0.9415
85		39	41	17.3800	6.6017	7.5747	6.0066	5.8246	3.5967	1.0920	1.2998	1.0811	0.9946
86		40	42	19.0790	6.8284	6.9759	6.0970	6.6365	3.5102	1.0921	1.3009	1.2150	1.0611
87		41	43	15.4420	6.4149	7.1215	5.9063	6.9024	2.6228	1.0189	1.3154	1.1499	1.1484
88		42	44	10.6250	6.2158	7.2978	5.7222	6.6844	1.7789	0.9550	1.1717	1.0144	1.0690
89		43	45	8.9583	5.9630	7.3726	5.9386	6.4491	1.3891	0.8960	1.0876	0.8890	0.9298
90		44	46	8.6681	5.8491	7.5398	5.6422	6.0870	1.2426	0.8694	1.0473	0.8100	0.8514
91		45	47	8.9358	6.2083	7.1821	6.1171	6.0959	1.2184	0.8980	0.9973	0.8544	0.8431
92		46	48	8.8933	6.3397	7.3568	5.8972	5.9121	1.2434	0.9808	0.9989	0.8751	0.8775
93		47	49	9.4228	6.6954	7.6191	6.6318	6.6131	1.4312	1.1240	1.0243	1.0000	0.9967