Dense Optical Flow Estimation using Diffusion Distances

Szymon Wartak

MSc

University of York
Computer Science

January 2010

Abstract

Diffusion maps have been shown to model relations between points by considering the overall connectivity of the graph. This report outlines how we can apply the diffusion framework to dense optical flow estimation where diffusion maps are used to embed distributions of local spatial gradients. We review the problem of dense optical flow estimation and several broad types of approaches to computing accurate estimate of the flow. We then review the diffusion framework and its predecessors in the manifold learning literature. Local image features are recorded by diffusion distances calculated from the graph Laplacian whose kernel function depends on inter-pixel intensity differences in a certain neighbourhood. These features are then used in a correlational optical flow estimation algorithm to illustrate the improvement to the dense estimate of optical flow by using a richer description of features as the elementary unit in the estimation.

By considering systems of correlation vectors from image neigbourhoods, we also increase the smoothness of the estimate. The present work compares several smoothing principles, including the vector mean, vector median, marginal median which are based on both the maximum correlation and minimum rank of correlation vectors from the correlation matrix.

A large number of very accurate estimates, spread through the image can be identified based on level of consensus with the estimates from surrounding pixels, which we term as confidence. We use this confidence information as a basis for smoothing the motion estimate by filling regions with poor confidence with estimates from neighboring high confidence regions.

The proposed methodology was applied on two distinct image sequences from the Middlebury data set, as well as a fluid motion data set. Results show the robustness of our method to the different types of input data.

Contents

1 Introduction **1**

1.1 Background 1

1.2 Research Contribution 2

1.3 Applications 3

2 Dense optical flow estimation **4**

2.1 Regularization 5

2.2 Feature based methods 7

2.3 Probabilistic models 10

2.4 Summary 12

3 Manifold Learning **13**

3.1 Kernel eigenmap methods 14

3.2 Other manifold methods 16

3.3 Computational tractability 17

3.4 The Diffusion Framework 18

3.5 Commute Times 20

3.6 Summary 21

4 Optical Flow estimation using Diffusion Distances **22**

4.1 Diffusion kernel 22

4.2 Markov process and diffusion distances 23

4.3 Alternative representations 24

4.4 Constructing the correlation matrix 24

4.5 Estimating optical flow from the correlation matrix 26

4.6 An effective postprocessing method 28

4.7 Summary 29

5 Experimental results **31**
5.1 Diffusion distances as feature descriptors 33
5.2 Estimating optical flow from the correlation matrix 35
5.3 An effective postprocessing method 43
5.4 Object data - Middlebury data set 45
5.5 Fluid data - cloud motion 51
5.6 Comparison to other representations and estimation methods 52
5.7 Summary 56

6 Conclusions **57**
6.1 Summary 57
6.2 Future Work 58

List of Tables

5.1 Angular and flow error comparison 48
5.2 Comparison of representations 54
5.3 Comparison of representations and scale factors 55

List of Figures

4.1 Gaussians for two scale values 23

5.1 Image data and flow field colour coding 32
5.2 Diffusion features representation examples 33
5.3 Representation of moving point in Hamburg taxi sequence . . 34
5.4 Errors for various flow estimation methods 37
5.5 Estimate with classical approach 38
5.6 Estimate with vector sum (7-by-7 39
5.7 Estimate with vector sum (5-by-5) 39
5.8 Estimate with weighted vector sum 40
5.9 Estimate with marginal median 40
5.10 Estimate with vector median 41
5.11 Vector sum aggregation 1 42
5.12 Vector sum aggregation 2 42
5.13 Threshold angular errors 44
5.14 Threshold flow errors 45
5.15 Threshold data proportions 46
5.16 Confidence maps for Dimetrodon and Venus 47
5.17 Angular errors from complete Baker et al (2007) data set . . 48
5.18 Flow errors from complete Baker et al (2007) data set 48
5.19 Dimetrodon postprocessing results 49
5.20 Venus postprocessing results 50
5.21 Flow estimates from fluid motion sequence 52
5.22 Flow estimates from fluid motion sequence 53
5.23 Flow estimates from fluid motion sequence 53
5.24 Comparison to Cuzol et al (2005) 54

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

Szymon Wartak

Acknowledgements

A number of people have contributed to this thesis as well as the underlying research. I want to thank my supervisor, Dr. Adrian Bors for his guidance during my degree and for helping with the writing and corrections to the thesis. I am also grateful to Prof. Edwin Hancock and Prof. Richard Wilson as well as the other students in the Computer Vision and Pattern Recognition group for the helpful conversations and discussions.

I would like to thank Filomena Ottaway and the staff members of the Computer Science department for their support.

As always, I am grateful to my parents, fiancee and brother who continuously offer ideas and inspiration in all aspects of life.

Chapter 1

Introduction

1.1 Background

Modeling features as well as dimensionality reduction for data representation occupies an important place in fields such as information theory and machine learning. Diffusion maps have been developed for representing data using sets of mutual distances with the aim to define their relational structure [1]. The adjacency matrix representing the graph Laplacian is constructed with the entries defined as probabilities of transition between any pair of points. Similar to other data mining and machine learning methods [2, 3, 4, 5], diffusion maps represent a graph-based approach employing data reduction by means of the eigendecomposition of the adjacency matrix. The eigenvalues define the significance of the diffusion directions as indicated by the eigenvectors. Here, diffusion maps are applied to representing local image features with the aim of estimating the optical flow from image sequences.

Many optical flow estimation algorithms use pixel brightness as the fundamental unit on which the algorithms operate. Various regularization based [6, 7] and registration based [8, 9] algorithms match individual or groups of pixels to minimize the brightness discrepancy between the source of motion in the first image, and its destination in the second image.

An alternative approach involves preprocessing the image to extract features, such as edges, which are then used as the fundamental unit of the optical flow estimation algorithm. While the features are well defined, their extraction is prone to error and may be ambiguous from image to image. Zitnick, Jojic, and Kang [10] address the problem of ambiguous segmentations by formulating the problem of optical flow estimation and segmentation as a single generative model with appearance and flow constraints. This produces temporally consistent segmentations that can be matched much more accurately and efficiently. Ren [11] uses a contrast boundary map to calculate the intervening contour [12] that defines the affinity between any pair of points. Optical flow is calculated at corners and edges, where the most robust flow information is usually present in shape motion. Finally, motion elsewhere in the image is spatially integrated from the estimates at edges and corners using the pairwise affinities.

Again, boundary-based optical flow estimation imposes limitations that make those methods unsuitable for such data as the motion of fluids, where hard boundaries do not exist. Corpetti *et. al.* used the vorticity-stream formulation to recover dense motion of water vapours [7]. Similar to other regularization based methods, the authors employed the continuity equation which again uses the pixel brightness as the fundamental unit in optical flow estimation.

1.2 Research Contribution

Chapter 2 reviews a broad range of methods for dense motion estimation and manifold learning algorithms, including the diffusion framework and commute times, which extend from the framework. Preliminary results show that the diffusion framework [1] can be used to extract the key elements of local features. This provides a representation based on diffusion distances that is robust against noise, capable of detecting the direction of features and describes how each point in the image relates to its broader neighbourhood in the image. The improvement in optical flow estimation by preprocessing images to extract dense feature information is demonstrated by applying a correlational optical flow estimation algorithm.

In addition to simple correlational flow estimation, we examine several statistics that aggregate neighbouring information in an attempt to provide a better estimate. Using regional information can in general be a useful way to mitigate the effects of noise, but some statistics will prove better than others depending also on the image data.

We also look at a second way of incorporating regional information. The confidence in the estimates is variable as the noise in an image is not uniform, and certain well defined features will yield higher relative correlations. For example, low textured regions will have similar correlations in a search region compared to textures with greater pixel contrast. Thus, we propose a confidence measure that identifies good estimates, and we use these higher confidence estimates to fill in regions where estimates are less certain.

Several novel results are presented in this thesis. Firstly, we outline how the properties of diffusion maps makes them suitable for representing image features. Secondly, we analyze several methods for incorporating regional information to improve the flow estimate. And thirdly, we introduce a confidence measure that we show identifies estimates with small error rates.

1.3 Applications

Applications of our method extend beyond optical flow estimation. Two potential applications are motion segmentation and video indexing.

In this thesis we mainly examine the applicability of diffusion distances to optical flow estimation. However, commute times, which are related to diffusion distances (see Section 3.5) have been shown to be applicable to scene segmentation by Qiu and Hancock [13]. In their paper Qiu and Hancock show that commute times naturally partition a non-uniformly distributed set of points lying on a circle. In this case, using diffusion distances requires tuning the scale parameter t, while commute times do not require parameter tuning. In addition to these embeddings, we could use similarities between the motion estimates to perform motion segmentation, similar to work by Robles-Kelly et al [14] or more recently, Vidal et al [15]. In combination, spatial and motion based segmentation can provide a good understanding of the objects in a scene.

Video indexing can also be performed using optical flow estimates [16, 17]. It is possible as well to use features described by the diffusion maps from Section 3.1 to find patterns in an image and use this information in conjunction with optical flow information to provide a rich indexing method.

We begin by looking at the literature in optical flow estimate and manifold learning in Sections 2 and 3, respectively. In Chapter 4, we introduce the details of the feature representation, the optical flow estimation methods and postprocessing. Then we present results from these stages of the flow estimation in Chapter 5, and conclude in Chapter 6.

Chapter 2

Dense optical flow estimation

Dense optical flow estimation, due to the sparseness of identifiable features, such as edges and corners, has predominantly relied upon gradient-based methods to compute the dense motion field [6, 7, 18]. Gradient based methods use the brightness constancy assumption in a global energy function optimization. The brightness constancy assumption states that points roughly conserve their intensity, E, over displacements corresponding to the ground truth motion, which is estimated by v:

$$\frac{dE}{dt} = \nabla E \cdot v + \frac{\partial E}{\partial t} \approx 0. \tag{2.1}$$

This one constraint is insufficient due to the aperture problem, which leads to ambiguous optical flow estimates in untextured regions. Because of the lack of meaningful variation in brightness in these regions, estimates from other regions with higher signal-to-noise ratios, such as corners or edges, must be used. Horn and Schunck's [6] algorithm computed the flow in untextured regions through an isotropic regularization term in the global energy function,

$$\sum_{\Omega} \left(\nabla E(a,t) \cdot v(a,t) + \frac{\partial E(a,t)}{\partial t} \right) + \alpha \left(\nabla v(a,t) \right), \tag{2.2}$$

where $v(a,t)$ is the estimate of the velocity at time t at point $a = (x,y)$ in the image place Ω. Recent gradient based algorithms use more complex regularization terms to deal with anisotropic features in the image.

The image registration algorithm by Lucas and Kanade [8] is the second seminal algorithm of optical flow estimation. The authors take a feature based approach as their algorithm matches local windows in a pair of images. It involves finding a value of h for which two functions yield the best match. This is done by finding the minimum of

$$E = \sum \left(F(\mathbf{x} + \mathbf{h}) - G(\mathbf{x}) \right), \tag{2.3}$$

where $\mathbf{x}$ and $\mathbf{h}$ are two dimensional row vector coordinates and offsets, respectively, in images F and G. The method iterates in a Newton-Raphson like fashion until the h that corresponds to the best match is found. The method can be generalized to match scaled, rotated or sheared objects by incorporating the linear transformation A for $G(\mathbf{x}) = F(\mathbf{x}A + \mathbf{h})$, into the iteration.

In this chapter, we review extension of these two seminal optical flow estimation algorithms, as well as combinations of the two and other feature based methods. We also review several stochastic algorithms, which have become more prevalent very recently in the field.

2.1 Regularization

Since the seminal work in dense optical flow estimation by Horn and Schunck [6], many regularization schemes have been suggested. These schemes are largely anisotropic, which reflects the relation of motion between neighbouring points in an image. That is, all neighbouring points do not need to have similar optical flow estimate, which can be seen most clearly at boundaries of moving objects. Tschumperlé and Deriche apply anisotropic diffusion PDEs to smooth the vector field [19]. Their method disassembles the regularization process into the smoothing itself and the underlying geometry that drives the smoothing.

Penalty functions are often applied to each of the constraints in the energy function being optimized in order to yield higher or lower degrees of smoothing of the vector field with respect to the constraints. Horn and Schunck applied a constant function α. Other work has used L^1 [20] and L^2 [7] norms as the penalty functions, the latter of which tends to smooth our natural discontinuities in the motion field.

Recent gradient-based methods incorporate estimates of features such as vorticity [7] or segments [21], into the energy minimization to regularize the motion field. However, because the regularization is one constraint in optimization of the global energy function, gradient based methods are limited by the choice of features, which are often specific to the properties of the images in the sequence. There is also the problem that the features that constrain the energy minimization are computed from (initially) very rough estimates of motion based on differences in brightness that often contain a lot of noise. This means that gradient descent methods, which are often used for the optimization, may end up in a local minimum instead of the global minimum, potentially resulting in a poor estimate of optical flow.

Memin et al [22, 7] have done several studies on optical flow in fluid data. In [7], Corpetti et al. use an improvement to the brightness constancy assumption called the *continuity equation*, and apply a regularization scheme based on the divergence and vorticity of the current estimate of the motion field. The authors use two aspects in the regularization to ensure convergence on a smooth motion field that accurately depicts the divergence and vorticity,

$$\sum_{\Omega} |div\mathbf{x} - \xi|^2 + \lambda f_2(|\nabla\xi|) + \sum_{\Omega} |curl\mathbf{x} - \zeta|^2 + \lambda f_2(|\nabla\zeta|), \tag{2.4}$$

where $\mathbf{d}$ is the displacement and f_2 is the quadratic penalty function. The divergence, *div*, and vorticity, *curl*, are defined as $\nabla\mathbf{x}$ and $\nabla\mathbf{x}^{\perp}$. ξ and ζ are the current estimates of the divergence and vorticity. The quadratic function is suitable in this case as hard boundaries generally do not exist in the meteorological data set used. The first part of each integral encourages a less erratic convergence of the divergence and vorticity, while the second part yields more spatially smooth estimates of each.

The brightness constancy assumption is broken is such cases as the occlusion of an object, and in the motion of fluids, which are compressible. The continuity equation, which is based on the density of a physical quantity is used instead of the brightness constancy assumption. The continuity equation given by,

$$\partial\rho/\partial t + div(\rho v) = 0, \tag{2.5}$$

where ρ is the density and v is the velocity field. By analogy, the image brightness represents the density of the clouds in the satellite images in Figure 5.1. The relation to the brightness constancy equation can be seen in the case of zero divergence in the continuity equation in which case the two are equal.

Cuzol, Hellier and Memin [22] also describe motion field in terms of the divergence and vorticity. Their iterative algorithm, similar to Corpetti et al [7], involves estimating the motion from the continuity equation and estimating the vorticity and divergence at each iteration. However, the optimization problem extracts the divergence and vorticity using the Helmholtz decomposition, and uses a low dimensional representation $f(\gamma, z)$ consisting of the Dirac delta function centred at z and strength γ.

Xu et al. [21] introduce an algorithm that first estimates affine motion of the parts of a segmented image. This sort of optical flow estimation assumes that all objects are rigid, which introduces errors in the estimate as the assumption generally does not hold for real and synthetic image sequences. To resolve this, their algorithm builds a confidence map from the combined motion and colour segmentation that describes the confidence in the initial flow estimate. The confidence map combines a pixel-wise coherence estimate, and a segment-wise confidence estimate that excludes parts of segments that undergo occlusion. The final estimate of the optical flow is computed by optimizing a global energy function that gives higher influence to more confident points.

2.2 Feature based methods

Many dense optical flow estimation algorithms do use the presence of features for the regularization of the motion vector field [7, 19]. Feature based methods differ from these as they employ features as the fundamental unit of the optical flow estimation procedure. Several types of features can be used in various ways to determine the optical flow. Feature based methods can employ tracking the movement of specific features such as edges or corners, matching a block of an image to surrounding areas in the subsequent image to see where it has moved, or performing an eigendecomposition on a tensor based on a three dimensional (2 space and 1 time), gradient based tensor to determine the principal components of the gradient, and hence the direction the 'matter' in the part of the image is moving.

Castelow et al. [23] present a least squares algorithm that estimates the optical flow using Canny edgels and an area around the edgel called a support neighbourhood. The edgel representation from the Canny edge detector provides a rich and accurate description of the edges in an image, including the location, orientation and the strength of the edge defined by the magnitude of the gradient at the point of inflection. Although only optical flow perpendicular to the edge can be obtained, due to the same aperture problem affecting pixel based comparisons, edges are sparser than pixels leading to an improvement in computation time.

Traditionally, the problematic aspect of feature matching methods has been locating the features that are later matched during the motion estimation. One problem is that features may change shape, size, or brightness making selecting the same feature and comparing the same locations of the feature during optical flow estimation difficult. Castelow et al. match edgels from the Canny edge detector to both previous and next frames to improve consistency. Zitnick, Jojic and Kang take a stochastic approach to address this issue. The authors use a generational model with spatial and temporal constraints to produce consistent segmentations from which optical flow is estimated [10]. Points that belong to a segment in one image must belong to the same segment in the next image. Segments are defined by Gaussian distributions representing their colours and coordinates. To make the algorithm more tractable, the optimization is split into two parts based on the colour and motion segmentations. The algorithm iterates between the two, optimizing the similarity of the segments defined by either motion or colour.

Early work in feature based methods largely centred around matching medium-sized windows in one image to the most similar window in a second image. Measures of correlation such as the sum of squared differences are often minimized to obtain the best measure of optical flow [24]. However, feature based methods while simple in principle, have been less popular for use in dense optical flow estimation due to several problems.

One problem is that windowing assumes rigid motion for the points in the window. This assumption rarely holds, even for non-deformable object, because movement in the third dimension will usually cause some linear transformation of the object in the two dimensional image. The assumption is especially unsuitable for data where the image does not contain rigid objects, as is the case with fluids. Another problem is that the window may contain points from more than one object. In this case, it is unclear whether the optical flow estimate is accurate for either object, or neither.

These problems can be somewhat alleviated using overlapping correlational windows. Szeliski and Coughlan [25] propose an alternative that does not require overlapping windows - a process that essentially involves computing the optical flow many times at the same point. Their method uses 2-D spline basis functions in the motion field representation. Five basis vectors, weighted, are able to represent most of the variance in the flow. The flow itself is estimated using the Levenberg-Marquardt algorithm with the splines and the motion vectors at their centres being optimized.

Because the correlational approach tends to contain many local minima, multi-scale (or hierarchical) techniques can be used to find a solution that is closer to the global maximum. In this class of techniques, the correlational approach is applied first on a lower resolution version of the original image. The optical flow estimates from the lower resolution are used as priors in progressively higher resolutions, in a coarse-to-fine methodology. Multi-scale techniques also add to the smoothness of the final estimate.

Robust statistics have been proposed by Black and Anandan [18] to resolve the problem of outliers that especially affects methods such as Lucas and Kanade [8] that attempt to provide one estimate from a window of data that may contain several different motions. Optical flow estimation algorithms that apply least squares estimation are particularly affected by outliers, which are known to affect least square disproportionately. The authors consider three broad classes of algorithms that use regression, correlation and regularization. They show that simply applying robust ρ-functions to the terms of the optimization preserves the estimation properties of the terms, while making them more robust to outliers and substructures that deviate from the consensus estimate.

Two similar methods by [26] and [27] determine the presence of piecewise affine motion of the image from the eigenvalues of a three-dimensional structure tensor, with 2 spatial dimensions and a third time dimension. The image sequence $I(\mathbf{x})$ is defined in a volumetric fashion, with the third dimension being time. The eigenvalues of the structure tensor

$$A_p = \langle (\nabla I_p)(\nabla I_p)^T \rangle, \tag{2.6}$$

can be used to determine the type of motion in a local window, I_p, by solving the ordinary eigenvalue problem, $A_p\vec{v} = \lambda\vec{v}$. Depending on whether

0, 1, 2 or 3 eigenvalues are greater than zero, there is either no motion, ambiguous motion because of the aperture problem, motion at a corner or edge, or brightness variation in all directions meaning no coherent noise, respectively.

Many modern algorithms have combined elements from local registration and global optimization involving regularization. One algorithm by Bruhn, Weickert and Schnorr [28] directly combines the original algorithms postulated by [6] and [8]. The authors note that the error term of Horn and Schunck's energy function is based on the gradient from just the central pixel, while Lucas and Kanade use a neighbourhood of pixels. Thus fusing the two methods can be accomplished by replacing the point based gradient with an area based definition of the gradient,

$$E(w) = \sum_{\Omega} \left(w^T J_\rho(\nabla_3 f) w + \alpha |\nabla w|^2 \right) dxdy, \tag{2.7}$$

where w is the current estimate of the optical flow field, ρ is the standard deviation of the Gaussian J_ρ that defines the local gradient window, f is the image data, and α is a parameter weighting the influence of the regularization term. The combined local-global method yields a dense motion field, while being less sensitive to noise because of the larger window for the error term.

2.3 Probabilistic models

Other than Zitnick et al.'s [10] algorithm for optimizing the joint probability distribution for colour and motion segmentation, several other stochastic methods have been applied to optical flow estimation.

Markov random fields (MRFs) have been applied to several very recent optical flow estimation algorithms. Lempitsky, Roth and Rother proposed one such MRF based algorithm that incorporated a non-convex energy function [29]. The energy function is non-convex firstly because the data error term (from the brightness consistency assumption) does not employ a linearized constraint, and secondly because the penalty function is based on robust statistics which are also non-convex. The benefit of using a non-convex energy function over a convex one can be seen at areas of discontinuity in the motion field. Convex functions tend to smooth out real discontinuities, while non-convex ones do not.

Glocker et al. [30] employed MRFs in the context of multi-labeling optimization, where the labels correspond to a quantized version of the displacement space. The optimization involves assigning labels to the nodes of the MRF based on an energy function,

$$E_{MRF}(l) = \sum_{p \in G} V_p(l_p) + \sum_{p \in G} \sum_{q \in N(p)} V_{pq}(l_p, l_q) \tag{2.8}$$

where the unary potentials V_p represent the data and pairwise potentials V_{pq} represent the smoothness term. G is a set of control points uniformly distributed over the image domain, and $N(p)$ are the neighbouring nodes at control point, p. The optimization of the MRF is performed using the Fast-PD method [31], which gives an approximately optimal solution.

A stochastic algorithm designed for the purposes of segmenting video by Chan and Vasconcelos [32], is also suitable for tracking larger objects. An image sequence can be represented using an observed variable y_t and a hidden state variable x_t defined by

$$x_{t+1} = Ax_t + v_t \tag{2.9}$$

$$y_t = Cx_t + w_t, \tag{2.10}$$

where x_t encodes the evolution of the video through a sequence of states, while y_t encodes the appearance, and v_t and w_t are their respective normally-distributed noise processes. Thus, differences in A, which encodes the state transition information, when combined with C, which decodes the observation information from the hidden state variable, contain the motion information. In the implementation, the initial state $p(x_t)$, the state transition $p(x_{t+1}|x_t)$, and the observation information $p(y_t|x_t)$ are all modeled as multivariate Gaussian distributions. The EM algorithm is used to learn the parameters of the joint distribution over a time period $1, \ldots, \tau$,

$$p(x_1^\tau, y_1^\tau) = p(x_1) \prod_{t=2}^{\tau} p(x_t|x_{t-1}) \prod_{t=1}^{\tau} p(y_t|x_t). \tag{2.11}$$

2.4 Summary

We have reviewed various algorithms for optical flow estimation. Regularization, or energy minimization based methods are a large class of algorithms developed on the basis of these algorithms, and we have looked at several modern algorithms that use the regularization approach. These are feature based methods, which use spatial information to improve the flow estimate, and probabilistic models that use methods from statistics to obtain an estimate.

Most of these optical flow estimation algorithms perform quite well on the Middlebury data set discussed in Section 5.4. The earliest methods by Horn and Schunck [6] and Lucas and Kanade [8] are simpler, faster, though have higher error rates. However, some recent methods that use regularization, such as the structure- and motion-adaptive method by Wedel et al [20], have similar or better runtimes and are capable of being adapted for real-time flow estimation.

As we discuss in Section 5.4, the Middlebury data set does include various types of features, high and low textured areas, but it is uncertain how well the methods tested on the Middlebury set will perform on fluid data. For example, Wedel et al's algorithm [20] is designed to deal well with motion field discontinuities, with other approaches also being designed to deal well at boundaries [11, 21]. On the other hand, method such as Cuzol et al's [22] are designed specifically to regularize divergence and vorticity patterns. We test our method on various types of data to show its robustness.

In the next chapter, we review manifold learning algorithms that are the basis of our scheme to represent image features for improved optical flow estimation.

Chapter 3

Manifold Learning

Many areas of artificial intelligence and data mining involve processing large, high dimensional data sets where finding meaningful relations between sets of data points is complicated by problems related to the curse of dimensionality and their data distribution. For example, 64×64 images of faces photographed with varying horizontal and vertical displacements of the camera contain over 4,000 dimensions, though the intrinsic number of dimensions in the data set will just be the two describing the viewing angle.

Classical dimensionality reduction methods such as PCA can extract prevalent features from a set of images. However, the precision of these methods is not adequate for dense optical flow estimation, which requires an accurate depiction of individual features that can be matched between frames to provide an estimate of the optical flow. The kernel eigenmap algorithms [1, 13, 33, 2, 3, 4] reviewed in this chapter all perform eigendecompositions based on the local structure of the neighbourhoods, which is preserved throughout the steps of the algorithm. By rooting the feature extraction process in the local structure, it allows us to use the kernel eigenmap methods for dense optical flow estimation, which requires local features to be intact for an accurate estimate to be possible.

In this Chapter, we introduce the various manifold learning methods that have been researched during the past 10 years. After introducing some general concepts in Section 3.1 and outlining the major methods, we examine computational limitations of these methods in Section 3.3. We then focus on the manifold learning algorithms used in our feature representation scheme. This scheme involves applying the diffusion framework in Section 3.4 and commute times in Section 3.5 on the image data. We discuss the details of the application in Chapter 4.

3.1 Kernel eigenmap methods

A recently developed and powerful set of methods for manifold learning are based on using eigenvectors to define the dimensions in a low dimensional manifold on which the data lie. These methods include two fundamental steps with several variations and extensions that we outline in this chapter. The fundamental steps are

- constructing a graph that connects local neighbourhoods of points using a kernel function, and
- solving the generalized eigenvector problem for the graph.

The result is an embedding of the original data from the high dimensional space, in a low dimensional manifold for which the basis vectors are the k largest eigenvectors. The first in this class of algorithms were published in 2000 in two separate papers by Tenenbaum et al [33] and the second by Roweis and Saul [2].

Tenenbaum et al introduced a geometric framework for nonlinear dimensionality reduction with their isometric feature mapping algorithm, ISOMAP [33]. Previous nonlinear methods were much less efficient computationally and were not always guaranteed to converge. Their method, as with other eigenmap methods, guarantees global optimality of the embedding as the eigenvectors explaining the most variance can be easily selected (by taking those eigenvectors with the highest corresponding eigenvalues) for the low dimensional embedding.

There are 3 steps to Tenenbaum et al's algorithm. The first step is to define the relations between a point and its neighbours. These neighbours can be chosen using a $k - nearest - neighbours$ approach or by selecting all points within some fixed distance, ϵ from the point. The latter method involves selecting a value of ϵ that results in connected graph. The second step of the algorithm is estimating geodesic distances $d_M(i, j)$ between all pairs of points on the manifold M by computing the shortest path in the graph. Finally, the graph G is embedded in a d-dimensional Euclidean space Y by minimizing the cost function

$$E = \|\tau(D_G) - \tau(D_Y)\|_{L^2} \tag{3.1}$$

where D_Y are the Euclidean distances and $\|.\|_{L^2}$ is the L^2 norm. The minimization is achieved by taking the first d eigenvectors of matrix $\tau(D^G)$.

The locally linear embedding (LLE) algorithm by Roweis and Saul [2] also describes the global scope of the relationships between points in a data set, though the eigenvectors are calculated from a matrix describing only local relations between points. Specifically, points are represented as linear combinations of their $k-nearest-neighbours$, where k is a free parameter. The weights in the linear combination are selected so as to minimize the error in reconstructing the original point. As with other manifold learning algorithms in this class, the final step is to embed the points in a lower-dimensional space based on the first d eigenvectors.

Belkin and Niyogi [3] further developed manifold learning theory by showing how Gaussian-like weights between neighbours (Gaussian kernel) in the graph relate to the heat equation, and by relating LLE to Laplacian eigenmaps. The latter method embed the high dimensional data using eigenvectors from the generalized eigenvector problem

$$L\mathbf{f} = \lambda D\mathbf{f}, \tag{3.2}$$

where D is the diagonal matrix, $D_{ii} = \sum_j W_{ji}$ and $L = D - W$ is the Laplacian matrix. Belkin and Niyogi show that LLE can be interpreted as finding the eigenvectors of the iterated Laplacian, L^2, which coincide with the eigenvectors of L.

In Hessian eigenmaps, the Laplacian matrix is replaced by the Hessian matrix in the generalized eigenvector problem. The method involves a n eigendecompositions to calculate the tangent coordinates at the neighbourhoods of each point in the manifold. These tangent coordinates are required to compute an approximation of the Hessian at a point in the high dimensional manifold. Earlier eigenmap methods required that data on the high dimensional manifold is globally isometric to a convex subset of a low dimensional Euclidean space. Donoho and Grimes showed that the assumption of convexity can be removed yielding unwarped embeddings for non-convex data sets [4].

The most recent manifold learning algorithm, diffusion maps [1], is the one used in the present work. Coifman and Lafon show that the aforementioned kernel eigenmap methods are a special case of a general class of algorithms based on diffusion processes. The construction of the kernel eigenmaps based on diffusion follows similar steps to the other eigenmap algorithms. Points in the high dimensional space are related to their neighbours using a Gaussian-like kernel, and the eigenvectors of the graph are used to map the points from the original high dimensional space to the low dimensional manifold. The authors apply a Markov process on the adjacency matrix of the graph by taking higher powers prior to the eigendecomposition, which acts to define structures at different scales. Furthermore, they define the notion of diffusion distances, which describe the distance between a pair of points by considering their relation to all other points in the graph. Diffusion distances and maps are discussed in more detail in Section 3.4 with their application to feature representation described in Chapter 4.

3.2 Other manifold methods

Prior to work by Tenenbaum et al and Roweis and Saul, nonlinear manifold learning algorithms based on MDS and PCA existed in two broad classes. Local linear techniques [34, 35] were not designed to represent global structure, thus being unable to represent the data in a single coordinate system and allow for the determination of the true underlying dimensionality of the data. Nonlinear methods [36, 37] based on greedy optimization did attempt to capture the global structure but were non-polynomial and did not guarantee global optimality.

Locally smooth manifold learning can be used to produce smooth manifolds [5]. Kernel eigenmap methods assume local linearity, from which local distances used for manifold learning are calculated. However, the assumption of linearity may not hold due to a high curvature of the manifold or noise data, making kernel eigenmaps potentially sensitive to the curse of dimensionality. The easiest solution is to consider more data points to learn the manifold, though this is not always possible and may require exponentially more data points. Bengio and Monperrus [5] propose a method for estimating the tangent plane around a point, x, using information from the whole data set. The authors show that by using the whole data set, regular types of structures can be learned that may be used in various parts of the manifold for improve tangent plane estimation. The tangent planes are estimated by minimizing the relative projection error of the nearest neighbours of x onto the tangent plane,

$$\min_{F,\{w_{tj}\}} \sum_t \sum_{j\in N(x_t)} \frac{\|F'(x_t)w_{tj} - (x_t - x_j)\|^2}{\|x_t - x_j\|^2}, \tag{3.3}$$

where x_t is the central point in the tangent plane and x_j are the neighbours. The minimization is done for all points x_t in the data set in order to learn the optimal set of basis vectors, $F'(x)$, for the set of tangent planes. Note that $F'(x)w$ is the projection of a local neighbour onto the tangent plane. Tangent spaces were also used in Donoho and Grimes's Hessian eigenmaps algorithm for the estimation of the Hessian at each point in the high dimensional manifold [4].

Langs and Paragios [38] applied diffusion maps to a segmentation task. The authors model shape variation by aligning the landmarks of the shape, adjusting for rotation, translation and scale with the remaining variation modeled by a multivariate Gaussian. Diffusion distances are then used to capture coherence around a set of landmarks and a clustering method is then applied to perform the segmentation. The algorithm is suitable for landmark tracking assuming that the landmarks are predefined.

Lefevre and Baillet [39] introduced the notion of optical flow estimation on 2-Riemannian manifolds. The authors show that the brightness constancy assumption and regularization methods like the one used by Horn and Schunck can also be applied on data lying on a manifold. The manifold itself is approximated by a tessellation, consisting of n nodes and t triangles, representing the manifold.

3.3 Computational tractability

Issues of computational tractability generally do not arise for classical dimensionality reduction problems where the size of the data set is on the order of hundreds. Tenenbaum et al. [33] used data sets of faces with different orientation and lighting, and the number 2 with different handwritten styling. Both these data sets contained about 100 points, meaning that the adjacency matrix in the high-dimensional space is of size 100×100. Belkin and Niyogi's swiss roll data set contained 2000 points [3]. Sets of this size start to warrant more efficient algorithms than the simple $O(N^2)$ pairwise calculation for determining the neighbours.

The data sets that we consider are often many times bigger. An image of size 640 × 480 yields an adjacency matrix of size 307200 × 307200. The first step of the algorithm that involves determining the neighbours is trivial and involves simply locating the surrounding pixels, which are already organized sequentially in a 2-D image raster, rather than having to construct a graph from unordered high dimensional data. Furthermore, calculating geodesic distances between all pairs of points, as is the case for the ISOMAP algorithm [33], is not necessary in our algorithm. We are applying the diffusion framework to find a description of broader features from only a local set of diffusion distances. Because of the ability of the diffusion distances to describe the broader extent of the feature, we need only to calculate a limited number of local diffusion distances to get a good description of the directions the feature extends from a given point. However, as with related manifold learning methods, the diffusion framework requires computing the eigendecomposition of the adjacency matrix in the high dimensional space. For the case of the 640 × 480 image mentioned above, this requires eigendecomposition for very high dimensional sparse matrices using methods such Arnoldi and Lanczos iteration for Hermitian and non-Hermitian matrices, respectively [40, 41].

3.4 The Diffusion Framework

Here we review the diffusion framework, which we use to extract feature information from the image [1, 42]. For a set of points $I = \{x_1, \ldots, x_n\}$, a random walk is constructed by considering the probabilities of moving from x_i to its neighbours, $\{x_{j_1}, \ldots, x_{j_k}\}$. Neighbouring points can be selected by various criteria discussed in Section 4.1. The choice to consider only the relations between neighbours stems from work on manifold learning where it is believed that relevant relational information can only be determined at a local level [3, 2]. Probabilities arise from a kernel function, $k(x, y)$ that defines the similarity between two points, x and y. In the present application to feature detection, we use the kernel function

$$k(x, y) = e^{-(x-y)^2/a}, \tag{3.4}$$

for two points x and y, and a scale factor, a. The kernel function guarantees the symmetry of the adjacency matrix and yields non-negative probabilities relating the points.

In order to construct a normalized graph Laplacian using the kernel function, $k(x, y)$, we can normalize the kernel by the local measure of degree in the graph $d(x) = \sum_{z \in I} k(x, z)$, and define the similarity of the pairs of points as a probability:

$$p(x, y) = \frac{k(x, y)}{d(x)}. \tag{3.5}$$

This probability of transition from x to y (note: k is symmetric but p is not) can be thought of as occurring in one time step. If we define an adjacency matrix, P, using these probabilities, we can consider probabilities of transition, $p_t(x, y)$ for more than one time step by taking higher powers of P forming Markov chains. The result from the Markov chain contains feature information about the data set I, while higher values of t increase the propagation of this information to the broader neighbourhood around the point of origin, x.

The scale factor, a, from the kernel function is the other factor determining the extent of the Markov process from a starting point with lower values of a inhibiting the propagation of information across noisier features. Though we do not explore the effects of this parameter in this paper, suffice to say that higher values of a may fail to record meaningful variations in intensities resulting in poorer boundary marking.

Coifman and Lafon introduced the *diffusion distance* metric between two points given by

$$D_t(x, y) = \left(\sum_{z \in I} \frac{(p_t(x, z) - p_t(y, z))^2}{\phi(z)} \right)^{\frac{1}{2}}, \tag{3.6}$$

where $\phi(z)$ is the stationary distribution defined by

$$\phi(z) = d(z) / \sum_{u \in I} d(u). \tag{3.7}$$

The metric defines the distance between two points by considering the probabilistic relations between those points and all other points in the graph. In particular, it integrates all the paths of length t starting at the points x or z as calculated in the Markov process [43].

Diffusion distances are calculated with the formula derived in [1]:

$$D_t(x,y) = \left(\sum_{l \geq 1} \lambda_l^{2t} \left(\psi_l(x) - \psi_l(y) \right)^2 \right)^{\frac{1}{2}}, \tag{3.8}$$

using the eigenvalues and eigenvectors of P from $P\psi_l = \lambda_l \psi_l$. ψ_0 is constant and the corresponding $l = 0$ is omitted.

3.5 Commute Times

Qiu and Hancock [13] showed that diffusion distances can be replaced by commute times, for a more robust estimate of the distance between two points. Because diffusion distances use a fixed value of t, the path length, they can be influenced by the specific choice of t that may not reflect the overall connectivity of the graph very well. Commute times, however, are a sum of all path lengths between two points, in both directions (i.e., going from x to y and from y to x).

This can be shown analytically [13] by summing diffusion distances $D_t(x,y)$ between a given pair of points, (x,y), over all possible path lengths,

$$\sum_{t=0}^{\infty} D_t^2(x,y) = \sum_{t=0}^{\infty} \sum_{i=1}^{m} (\lambda_P)_i^{2t} (\psi_i(x) - \psi_i(y))^2 \tag{3.9}$$

and using the properties of power series where $\sum_{t=0}^{\infty} (\lambda_P)_i^{2t} = \frac{1}{1-(\lambda_P)_i}$ to get,

$$\sum_{t=0}^{\infty} D_t^2(x,y) = \sum_{i=1}^{m} \frac{1}{1-(\lambda_P)_i} (\psi_i(x) - \psi_i(y))^2 = \sum_{i=1}^{m} \frac{1}{\lambda_i^{'}} (\psi_i(x) - \psi_i(y))^2, \tag{3.10}$$

which is identical to the commute time measure up to a constant [13].

3.6 Summary

We have reviewed the major manifold learning algorithms in this chapter. This class of algorithms finds the underlying relations between data points by establishing correspondences between points and their nearest neighbours, and using these correspondences to embed the data in a low dimensional manifold. In addition to being able to extract meaningful information and embed the data, some of the latest algorithms [1, 13] also define distances between points in the low dimensional manifolds. The latter of these is particularly important in the present work as it allows for a fine-grained description of the features that are subsequently used in the optical flow estimation. In the next chapter, we detail the diffusion distance based representation and discuss optical flow estimation methods that use these diffusion distances. We also introduce a confidence measure that can be used to select accurate flow estimates.

Chapter 4

Optical Flow estimation using Diffusion Distances

In this chapter, we detail how the theoretical framework of diffusion maps can be applied to dense optical flow estimation. There are three stages to the estimation, which are the feature extraction and representation with diffusion distances, estimation of the optical flow from the correlation matrix, and propagating high confidence estimates in the postprocessing step.

The first stage involves representing local image features by diffusion feature representations (DFRs). DFRs comprise each point's diffusion distances to surrounding points. Correlations between local DFRs are stored in a correlation matrix used in estimating the optical flow in the second stage.

The estimation of the flow can be done using any of a number of methods that we introduce in Section 4.5. These methods aggregate correlations from regions of the image using one of several statistics: vector mean, weighted vector mean, vector median and marginal median. We also look at how these aggregations compare with using just the estimate at each point.

In the final stage, we take the flow field and apply an iterative postprocessing step based on a confidence measure. This step propagates estimates at points with a high confidence isotropically.

4.1 Diffusion kernel

Let us consider a block of pixels $B = \{I_x | x = 1, \ldots, d\}$. We evaluate the anisotropic diffusion kernel for each of the blocks of pixels as $k(I_x, I_y) =$

$e^{-(I_x - I_y)^2/a}$ where I_x and I_y are the normalized image greyscale values in the range [0,1) located at x and y, respectively, and a represents the scale of the diffusion kernel. The negative exponential function and parameters used yield high degrees of similarity for pixels of equal or close to equal brightness, while yielding low similarity scores for pixels further away on the grey scale, as is depicted in Figure 4.1. This allows for the detection of near equiluminescent features, while maintaining good sensitivity to boundaries in the subsequent diffusion distance metric.

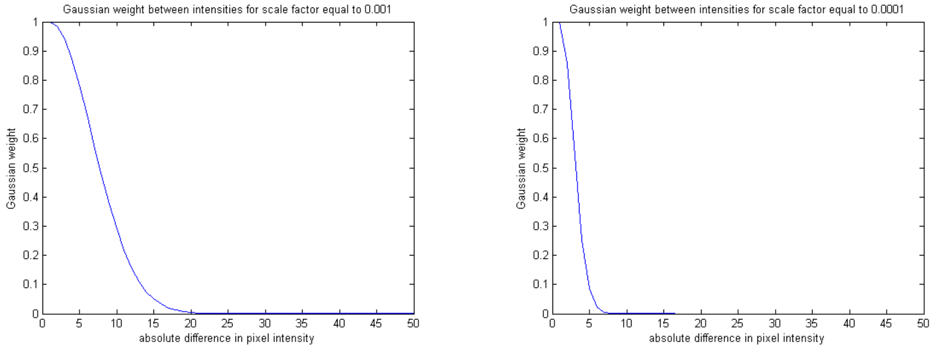


Figure 4.1: Gaussian functions of the normalized differences in pixel intensity for two *scale* values that were used in our experiments $a = 0.001$ and $a = 0.0001$.

Belkin and Niyogi [3] suggested *minimum distance neighbourhoods* and *n nearest neighbours* to select the number of neighbouring points in the adjacency matrix, P. The raster images used in the present application allows for a fixed window around the central pixel scheme to be used. We call this the *kernel window.*

4.2 Markov process and diffusion distances

Having computed the diffusion kernel, we proceed to run the Markov process and compute the diffusion distances from its eigendecomposition as described in the review of the diffusion framework.

The degrees of similarity defined in P are between the neighbouring pixels in the image. The result of the Markov process, P^t, is a metric of similarity after t time steps. And finally, the diffusion distances provide a metric of similarity by integrating all the paths in the image graph of length

t between two points. Each of these stages provide an increasingly broad description of the feature - the Markov process by propagating information from immediate neighbours into the broader neighbourhood, and diffusion distances by integrating path information from the whole image. Our results confirm that the broader descriptions translate into more accurate estimates of optical flow.

The DFRs from the final stage provide an efficient representation of feature orientation around a given pixel as it relates pairs of points based on all paths of length t that connect the points. This also makes the representation robust in the presence of noise as the distances are less affected by changes to individual pixels. We give several examples of features represented by diffusion distances in Figure 5.2.

4.3 Alternative representations

In addition to the representation based on diffusion distances, we use two other representations of the image data and compare the effectiveness of each in optical flow estimation.

The first representation is commute times which we have introduced in Section 3.5 as a potential improvement on the representation based on diffusion distances. For the optical flow estimation problem, we can simply replace the diffusion features based on diffusion distances with ones based on commute times. Much of the calculations remain the same, except we are now using a representation that is based on paths of all lengths, which may improve the feature representation and thus the optical flow estimate. This has been discussed in detail in the aforementioned section.

The second alternative representative is our baseline, or the raw image data. Similar to the commute times, we can replace the diffusion distance based representation with the image data simply by taking overlapping regions. This method is faster than the other two as it does not involve the numerous eigendecompositions required for the eigenmap calculations.

4.4 Constructing the correlation matrix

In this approach we propose to use the diffusion framework in order to model the local image features and to estimate the optical flow by matching the

DFRs corresponding to two blocks of pixels from two different frames. The match with the highest Pearson correlation coefficient provides the estimate of the optical flow at the central pixel in the first image. This commonly used coefficient is defined as,

$$r = \frac{1}{n-1} \sum_{i=1}^{n} \left(\frac{X_i - \bar{X}}{s_X} \right) \left(\frac{Y_i - \bar{Y}}{s_Y} \right), \tag{4.1}$$

where X and Y are the two feature vectors and s_X and s_Y are their respective standard deviations. The complex similarity measures, presented in this paper, that serve as a representation of the features can also be adapted for use in variational and other methods for estimating optical flow.

Two parameters that are analyzed in this study are the *diffusion window* and the *search window*. The *diffusion window* defines the size of the DFR blocks being correlated, while the *search window* defines the search area for matching DFR blocks in constructing the correlation matrix. We define the sizes of the two windows in terms of their respective radii, r, where $size = 2 \times r + 1$, which we use in equation 4.2.

Thus, the final optical flow estimation involves finding the point in the second image where the surrounding diffusion distances correlate maximally with the diffusion distances from the source point in the first image:

$$\begin{aligned} \bar{v} = \arg\max_{k,l} \frac{1}{4r_d^2 + 1} \sum_{i=-r_d}^{r_d} \sum_{j=-r_d}^{r_d} & \left(\frac{D_{t,q}(x+i, y+j) - \bar{D}}{s_D} \right) \\ & \left(\frac{D_{t,q+1}(x+i+k, y+j+l) - \bar{D}}{s_D} \right), \end{aligned} \tag{4.2}$$

where p is the image in the sequence, r_d is the radius of the *diffusion window*, k and l are the offsets in the *search window*, with $k, l \leq r_s$, where r_s is the radius of the search window. t is the number of time steps for the Markov chain, as defined earlier, while $\bar{D}$ and s_D are the sample mean and standard deviation of the diffusion distances from the given block B.

Correlation has been used here to relate the DFRs as it is invariant to changes in location and scale which is appropriate for use with the unnormalized DFRs. However, a drawback to using correlations is that the number of correlation operations required to construct the correlation matrix results in a long computation time for each pair of images.

4.5 Estimating optical flow from the correlation matrix

After calculating the correlations of features based on diffusion distances or commute times, or correlations of neighbourhoods of pixels from the raw image data, a correlation matrix is obtained. The values in this matrix are the correlations of the features or pixels centered at each point in the image, to features or pixels centered at neighbouring points in a 7-by-7 search region around the central pixel. [1]

Here, we compare several methods for filtering the correlation matrix to obtain a single optical flow estimate. The methods include,

- **classical approach** maximum correlation from the central point, given in equation 4.2
- **sum** of the local correlations from a 7-by-7 neighbourhood around the central point,
$$\bar{v} = \arg\min_i \sum_{n \in N(z)} R_n(i) \tag{4.3}$$
- **diffusion-weighted** (diffusion distance-weighted) sum of the local correlations from a 7-by-7 neighbourhood around the central point,
$$\bar{v} = \arg\min_i \sum_{n \in N(z)} D(n,i)\dot{R}_n(i) \tag{4.4}$$
- **marginal median** marginal median of the local correlations from a 7-by-7 neighbourhood around the central point,
$$\bar{v} = \arg\min_i median_{n \in N(z)} R_n(i) \tag{4.5}$$
- **vector median** of the local correlations from a 7-by-7 neighbourhood around the central point,
$$\bar{v} = \min\left(\arg\min_{n \in N(z)} \sum_{m \in N(z)} \sqrt{\sum_i \left(R_n(i) - R_m(i)\right)^2}\right) \tag{4.6}$$

[1] there is a distinction between the 7-by-7 search region which is the vector of correlations and the 7-by-7 neighbourhood of points at which the vectors are centred

where $N(z)$ are the neighbouring pixels around z, R_n and R_m are the local correlation vectors centred at n and m, respectively, i are the indices in the correlation vector, and $\bar{v}$ is the estimate of motion. Note that $D(n, i)$ is the diffusion distance between the image points n and i, and the indices in the correlation vector correspond to two-dimensional coordinates from which the offset or flow can be easily calculated.

With the **classical** method, the estimate consists of the maximum correlation from the central pixel's correlation vector. This method is among the least accurate in terms of error rates as it incorporates the least amount of information into the optical flow estimate. The remaining methods incorporate information from a 7-by-7 neighbourhood around the central point.

The **sum** method, which takes the sum of the local correlations the neighbourhood around the central point, and selects the direction estimate corresponding to the minimum of the local sum. The **sum** method is likely to produce low consensus at edges or other areas where the optical flow field is discontinuous. However, because the 7-by-7 region used is not particularly large, the proportion of the points where the neighbourhood contains a discontinuity in the flow field is small.

Nevertheless, we can exclude parts of the neighbourhood that correspond to differing motions from the central point by using the diffusion distances that we have already computed. In many cases, different objects moving in different directions will also have different intensities. This difference in intensities will yield larger diffusion distances, which are used in the **diffusion-weighted** method to calculate instead a weighted sum of correlation vectors. The smaller diffusion distances between similar points weights the estimate accordingly for the flow estimation.

The other two estimation methods investigated are based on the median statistic, namely the **marginal median** and the **vector median**. Mean-based statistics are biased by outliers and can thus yield poor results at motion boundaries. There may be a significant number of outliers to deal with in this case as the effects of noise have grown through the feature representation and correlation stages. Thus we look at the whether median-based statistics can provide an overall improvement by improving estimates in such areas.

The first is the **marginal median**, which calculates the median from the neighbourhood of vectors at each point in the search region. This statistic extends the **sum** statistic by treating each point in the search region independently, but instead takes the median at that point instead of the sum.

The **vector median** method also uses the median statistic, except points in the search region are not treated independently. Instead, the Euclidean distance between each pair of vectors in the neighbourhood is calculated and the one with the smallest sum of total distances is the vector median [44]. From the median vector, the maximum correlation in its search region is used as the estimate.

Finally, we incorporate a multiresolution approach for the detection of large displacements. Our method calculates and stores the correlations between the features in the search region, and this requires significant time and memory. Thus, repeating the correlation step at multiple resolutions is significantly more efficient than correlating features in a larger search space.

Furthermore, our postprocessing method, described in the next section, allows for some tolerance in finding correct estimates. If a correct estimate is not found at all pixels in a region, this may not be a large concern as values from neighbouring pixels will be used. We now look at the postprocessing method.

4.6 An effective postprocessing method

In an evaluation of optical flow estimation algorithms in 1994, Barron et al. [45] noted that there was a lack of confidence measure to highlighting regions contributing to a good overall estimate of optical flow. Bruhn et al. [28] suggested a confidence measure for variational methods based on the contribution of the data term at various points to the overall energy. Areas with low energy have small deviations from model assumptions, which are smooth gradients and brightness consistency.

The flow estimation method in the present work is based on correlating regions of the image. There is an implicit smoothness in the estimate which can be extracted from the initial neighbourhood estimate. Because the flow estimate in a given region of an image can be assumed to be smooth (an assumption often made by modern algorithms), we can assume that the best flow estimate will be the same in a given region. Furthermore, there may be one estimate in the region that consistently produces high correlations, and this is very likely to be the true estimate of the flow.

The measure of confidence that we use is based on a function of the aggregate correlational vector from one of the methods described in the previous section. That is,

$$C(R) = max\left(\frac{\left(\frac{min(R)}{r}\right)^4}{\sum_{r\in R}\left(\frac{min(R)}{r}\right)^4}\right) \tag{4.7}$$

where $r \in R$ and R is a vector of ranks of correlations between source and destination diffusion features of a size 7×7. This measure of confidence is normalized to 1 with the denominator and the exponent has been determined empirically. The measure takes the best correlational match and checks how the estimate compares with other estimates in the vector. High values (close to 1) imply that neighbouring vectors rank a particular estimate consistently very high. We show in Section 5.3 that confident values are also more likely to be accurate estimates.

Given high confidence estimates, we can employ various algorithms to propagate estimates from high confidence areas to areas of low confidence, similar to how energy minimization algorithms regularize the optical flow field. We show in our experimental results (Section 5.3) how a trivial method that propagates estimates at pixels with confidence above a certain threshold produces good results.

4.7 Summary

In this chapter, we have detailed our algorithm for optical flow estimation which comprises three stages. The first involves extracting image features by computing an adjacency matrix based on a function of differences in pixel intensities, and using the adjacency matrix to embed the image in a low dimensional manifold. The aim is to have a representation that will highlight the direction of a feature from a certain reference point, and to be able to do so at every point in the image, allowing for a dense estimate of optical flow. The representation that we use is based on diffusion distances and it satisfies these criteria.

The second stage of the estimation involves aggregating information from the correlation matrix. The correlation matrix has been calculated by correlating the representations based on diffusions distances. We propose several statistics that address the issue of outliers, such as the vector median or a weighted vector sum that uses diffusion distances to select for points that are more related based on feature information. We also use the marginal median and sum statistics to treat flow directions independently when aggregating.

The final stage involves using the confidence measure introduced in Section 4.6 to improve estimates in areas where aggregation has not produced a good flow estimate due to a larger variability in the flow estimates in that region.

The advantage of our algorithm is that it is sequential and can be analyzed at each stage. However, these types of algorithms generally perform poorer than ones that integrate the constraints of each step into one optimization problem, as errors are not allowed to accumulate at each step. We discuss in Section 6.2 how the diffusion distance based representation might be integrated into other optical flow estimation methods that use the one step optimization approach.

We now proceed to test our algorithm empirically in the next section.

Chapter 5

Experimental results

In the following experiments we consider two different data sets: one is the Middlebury data set from which we use the Dimetrodon and Venus image sequences, while the other is an atmospheric image sequence showing cloud movement. The images from the Dimetrodon and Venus sequences as well as the cloud image are shown in Figure 5.1 (a), (b) and (d), respectively.

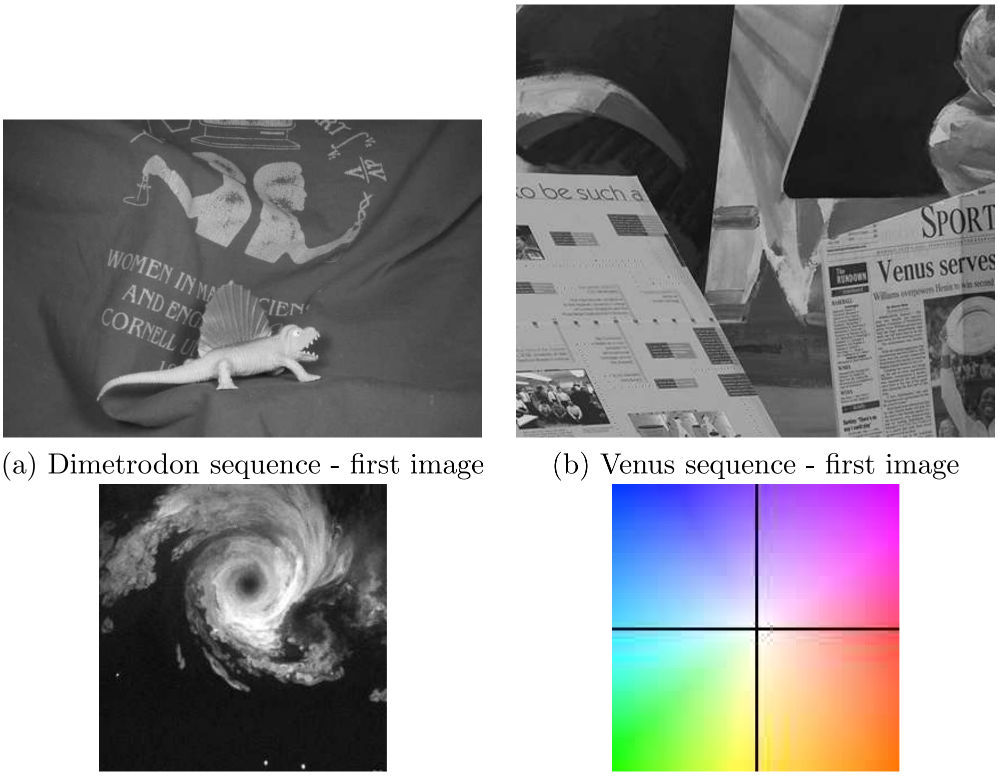
We show results at each of the three stages of the estimation algorithm. The first stage is the computation of the diffusion distances for the diffusion feature representations. In Section 5.1 we briefly discuss how diffusion distances will represent certain types of features, and display several example representations.

The second stage is the estimation of the motion from the correlation matrix. In Section 5.2, we show results from estimations using the five methods described in Section 4.5 and discuss how these results relate to other algorithms.

The third stage is the postprocessing on the flow estimate using the confidence map. In Section 5.3 we motivate the use of the confidence measure defined in equation 4.7 by looking at error rates at various intervals of the confidence measure.

Finally, we look at several parameters that define diffusion distances in Section 5.6 in order to ascertain their effect on error rates from the optical flow estimation using the diffusion distance representation. We also include qualitative and quantitative results from the Middlebury data set in Section 5.4 and the fluid motion data in Section 5.5.

Quantitative results are defined by two error measures that are com-



(a) Dimetrodon sequence - first image

(b) Venus sequence - first image

(c) Cloud sequence - first image

(d) Colour Map

Figure 5.1: Images from the Dimetrodon (a) and Venus (b) image sequences as well as the fluid motion sequence (c). Optical flow estimates for the Middlebury data are coded using the colour map in (d).

monly used in evaluating optical flow estimation algorithms [45, 46]. These measures are *average angular error* defined by

$$AE = arccos\left(\frac{u_{gt}u_e + v_{gt}v_e + 1}{\sqrt{(u_{gt}^2 + v_{gt}^2 + 1)(u_e^2 + v_e^2 + 1)}}\right), \tag{5.1}$$

and originally introduced in [45] and *average flow error* defined by

$$FE = sqrt[(u_{gt} - u_e)^2 + (v_{gt} - v_e)^2], \tag{5.2}$$

where (u_e, v_e) is the estimated flow and (u_{gt}, v_{gt}) is the ground truth flow. Errors for large flows are smaller using the AE measure, while the FE measure provides a less biased measure, especially for zero-flow areas.

We begin by looking at the feature vectors calculated from diffusion distances in the next section.

5.1 Diffusion distances as feature descriptors

We present several examples of features from the object and fluid data sets. Diffusion distances within a certain feature, or in the direction the feature extends, are small as compared to distances to points outside the feature. In all the experiments we have chosen $a = 0.001$ from equation (3.4). The *kernel window*, or feature size, was constant at 5×5, and the number of time steps was $t = 3$. These values provided a good balance between the accuracy of the feature description and computation time. We compare the results from these values of *kernel window* and the scale factor a in Section 5.6. Note that the *kernel windows* in Figure 5.2 are larger than the 5×5 used in the correlation to show how the features extend in the broader neighbourhood of pixels.

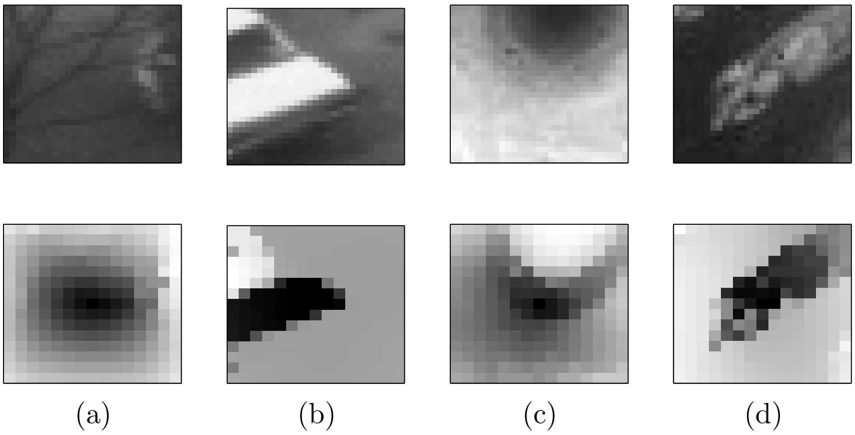


Figure 5.2: Images and their respective diffusion feature representations (DFRs). Note that the *kernel windows* in Figure 5.2 are larger than the 5×5 used in the correlation to show how the features extend in the broader neighbourhood of pixels. Darker areas represent smaller diffusion distances from the central source pixel. Note that large *diffusion windows* are used to illustrate the nature of the diffusion distance metric, whereas for increased efficiency, smaller *diffusion windows* are used in the optical flow estimation. a) and b) are from the object data set, while c) and d) are from the fluid data set

The first pair of images in Figure 5.2 (a), illustrate the shortest diffusion distances (dark patch horizontally stretched) highlighting the gradient of the shading in the pavement. The trees in the foreground do not affect the description of this dominant feature, or the approaching vans on the right side of the image (white patch).

Next, in Figure 5.2 (b) the trunk of the taxi is the main feature. Because of the next near-equal intensity, the diffusion distances are very small for the entire trunk, relative to the surrounding areas. As described earlier, the optical flow estimation correlates DFRs from a point in the first image, to neighbouring points in the second image. The optical flow estimation correlates DFRs between consecutive frames. We demonstrate in Figure 5.3 the close resemblance between DFRs in consecutive frames of the taxi image sequence.

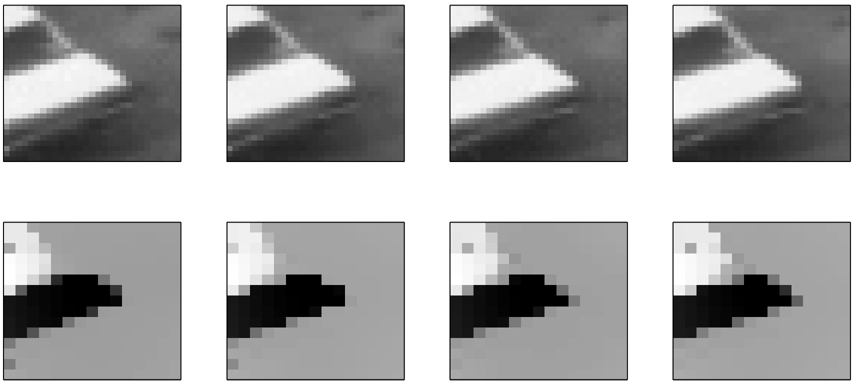


Figure 5.3: A sequence of 4 consecutive frames from the Hamburg taxi sequence. The central pixel has been adjusted to remain on the corner of the trunk of the taxi to demonstrate the close resemblance of the DFR from frame to frame.

In Figure 5.2 (c), the cloud patch of varying intensity maps to a fairly smooth feature in the DFR. The side that extends toward the terminal of the cloud is brighter (larger diffusion distances) than the side that sees a longer extension of the feature. Diffusion distances incorporate all paths between two points, and the direction in which the feature extends will include more paths between the two points yielding a shorter distances. By detecting how far a feature extends from the source point in this way, smaller *diffusion windows* can be used to detect features that extend well beyond the window proper.

Finally, the near equi-luminescent concentric band around the centre of the storm is the main feature in Figure 5.2 (d). Although no clear boundary exists, diffusion distances are calculated from a kernel that yields greater similarity hence shorter diffusion distances for pixels of similar intensity in the band.

Thus, we see from these examples how the diffusion feature representations can highlight some important aspect of the relation between two points in an image, such as the direction the equiluminant pixels extend, while retaining the details required for dense optical flow estimation with the pairwise relations. In Section 5.6 we quantify the benefits in using DFRs by comparing error rates from the Middlebury data set. Next, we evaluate several optical flow estimation methods based on the matrix of correlations of DFRs.

5.2 Estimating optical flow from the correlation matrix

As was described in Section 4.5, after calculating the correlations of features based on diffusion distances or commute times, or correlations of neighbourhoods of pixels from the raw image data, a correlation matrix is obtained. The values in this matrix are the correlations of the features or pixels centered at each point in the image, to features or pixels centered at neighbouring points in a 7-by-7 search region around the central pixel [1].

Results from following methods, described is Section 4.5,

- **classical approach** maximum correlation from the central point,
- **sum** of the local correlations from a 7-by-7 neighbourhood around the central point,
- **diffusion-weighted** (diffusion distance-weighted) sum of the local correlations from a 7-by-7 neighbourhood around the central point,
- **marginal median** marginal median of the local correlations from a 7-by-7 neighbourhood around the central point,

[1] There is a difference between the 7-by-7 search region which is the vector of correlations and the 5-by-5 neighbourhood of points at which the vectors are centred

- **vector median** of the local correlations from a 7-by-7 neighbourhood around the central point,

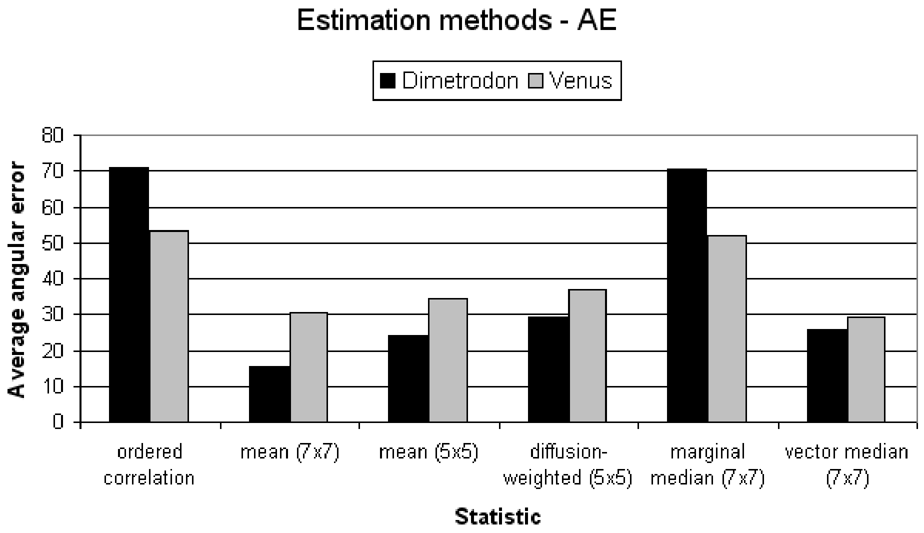
are compared. Correlations are ordered from highest (1) to lowest (49) for the methods that use statistics on the ordered correlations rather than the correlations themselves. The results are from the initial estimation using the noted methods, prior to any postprocessing.

Figure 5.4 shows the results from using various methods of estimation. Using information from the neighbourhood surrounding the central pixel does improve the error rates, except in the case of the marginal median. The improvement is not unusual, as the information can be noisy at several points in a regions, while the estimate in general will point in the direction of the actual motion.

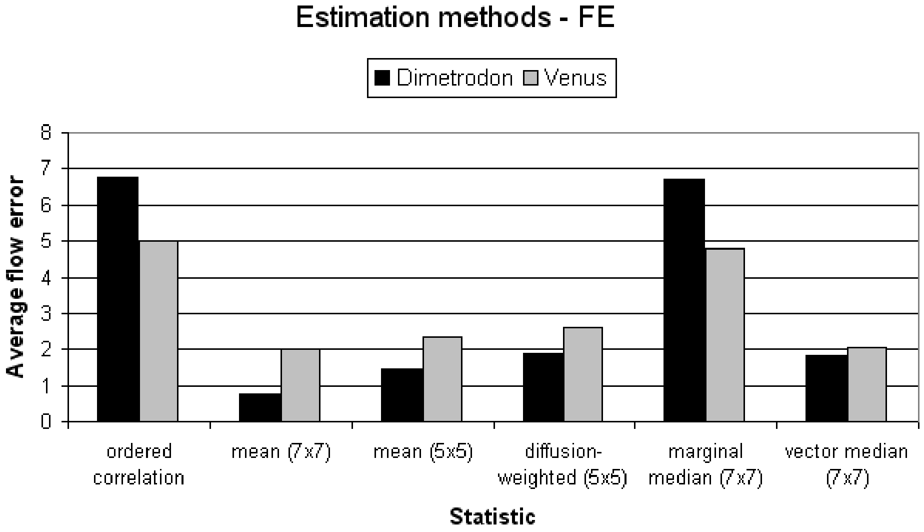
A noteworthy result from Figure 5.4 is also the superiority of the vector median over the marginal median. By treating the individual correlations in their vectors independently, some important information is being lost. This is unusual given that the estimate from the vector sum is quite good, and could mean that the outliers being biased against in the marginal median are contributing important information to the statistic. The vector median does not necessarily bias against these outliers as they are tied in with the complete set of correlations in the vector.

Figures 5.5-5.10 show the optical flow fields for the four of the estimation methods. We see in Figure 5.5 that large areas of correct estimates in the Dimetrodon sequence are very few, while in the Venus sequence there are several such regions. The presence of well defined features in the Venus images allows for accurate estimates at individual points. In the Dimetrodon images, due to the low degree of texture, correlations between various points in the *search window* will be quite similar. This, along with noise in the image data and effects of outliers produces much poorer estimates when no information from neighbouring estimates is used.

Figure 5.6 shows the best estimates of the optical flow from the methods described in this section. Both the Dimetrodon and Venus estimates contain large areas of smooth estimates that are close to the ground truth. However, there is a bigger improvement in the Dimetrodon sequence estimate than the Venus sequence estimate when compared to the **classical approach**. When comparing larger regions of smooth flow estimate in the Venus sequence between Figures 5.5 and 5.6, we notice an overlap, with the smooth regions



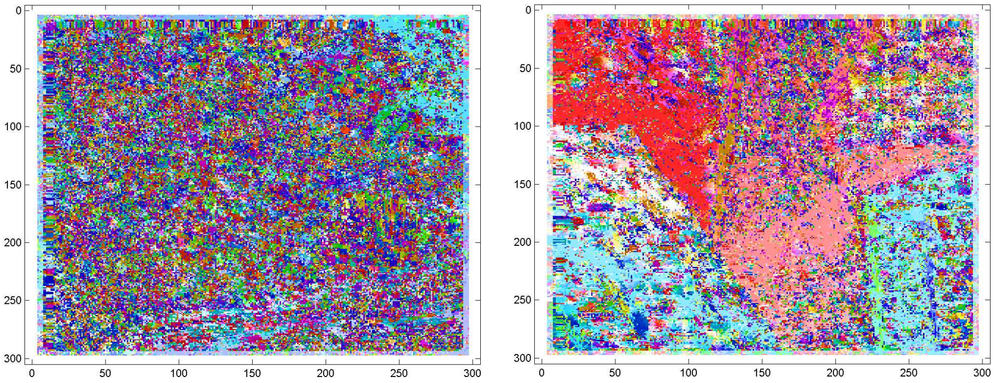
(a) Angular Error



(b) Flow Error

Figure 5.4: Average angular and flow errors for the different optical flow estimation methods outlined in Section 5.2.

from the **classical approach** estimates being extended to surrounding regions with the vector **sum** estimation method. This is not surprising given that the vector **sum** method is using the estimates from the same smooth



Dimetrodon - Classical Approach

Venus - Classical Approach

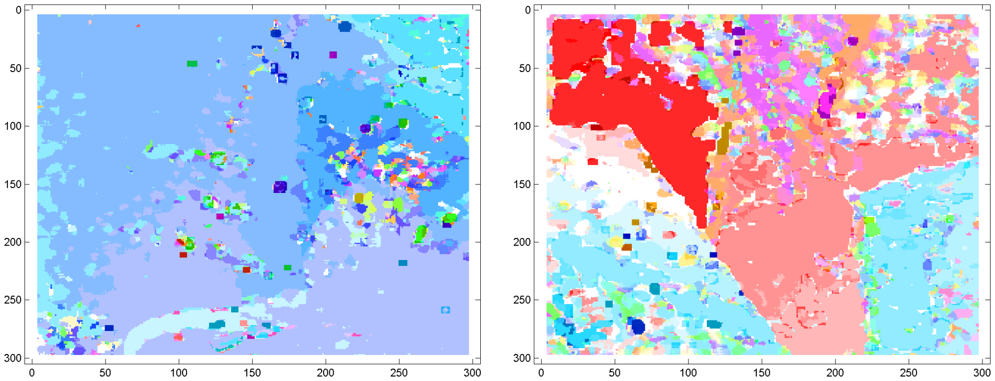
Figure 5.5: Optical flow fields for the Dimetrodon and Venus data sets for the **classical approach** using the diffusion distance based representation.

regions in the **classical approach**, while yielding improvements in other regions from the aggregation of correlational information.

In the case when less information is aggregated in Figure 5.7, there is a slightly decrease in the smoothness of the flow field. In Figure 5.8 where diffusion distances weight the correlation vectors proportional to their (diffusion) distance from the central point, there is a further decrease in smoothness. The diffusion distance weighting may have been expected to improve the flow estimate at the boundaries between objects, though this does not seem to have happened. In fact, the decreased weighting seems to have led to an overall decrease in information proportional to the central point in the neighbourhoods, leading to the slight increase in errors (Figure 5.4).

The flow estimates based on the **marginal median** can be seen in Figure 5.9. Using the median statistic aids in dealing with outliers, which is seen here through the decrease in speckles among smooth regions. However, the marginal median treats individual correlations within the correlation vector independently when computing the statistic for a neighbourhood of vectors which has instead led to larger speckles of poor estimates within regions of smooth estimates in the Venus sequence. This effect can somewhat be seen in the Dimetrodon flow estimate, although this estimate contains much higher error rates and fewer smooth estimate regions.

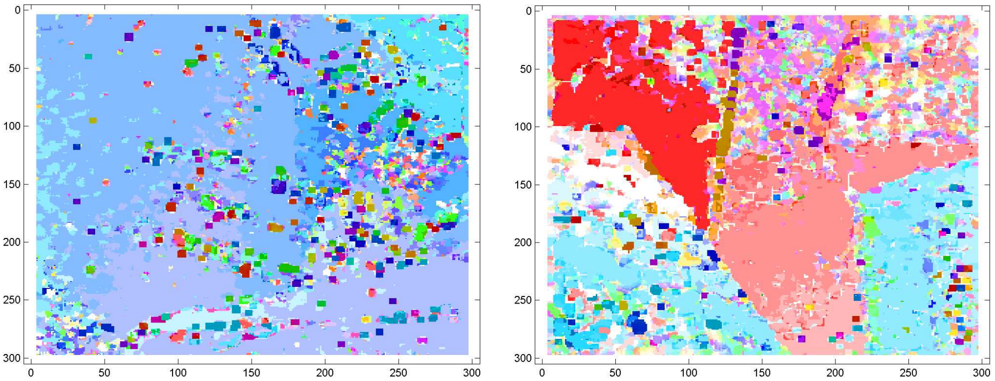
The vector median is quite a different statistic than the vector sum but



Dimetrodon - Vector Sum

Venus - Vector Sum

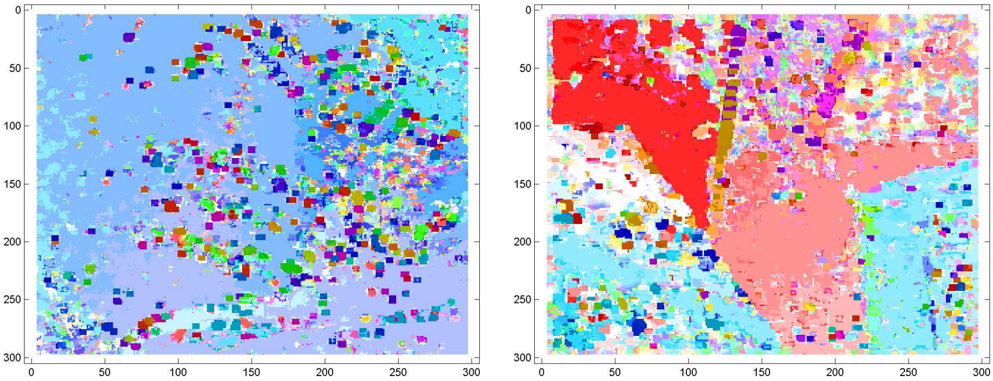
Figure 5.6: Optical flow fields for the Dimetrodon and Venus data sets for the vector **sum** using the diffusion distance based representation and a 7-by-7 neighbourhood of correlation vector to compute the sum.



Dimetrodon - Vector Sum

Venus - Vector Sum

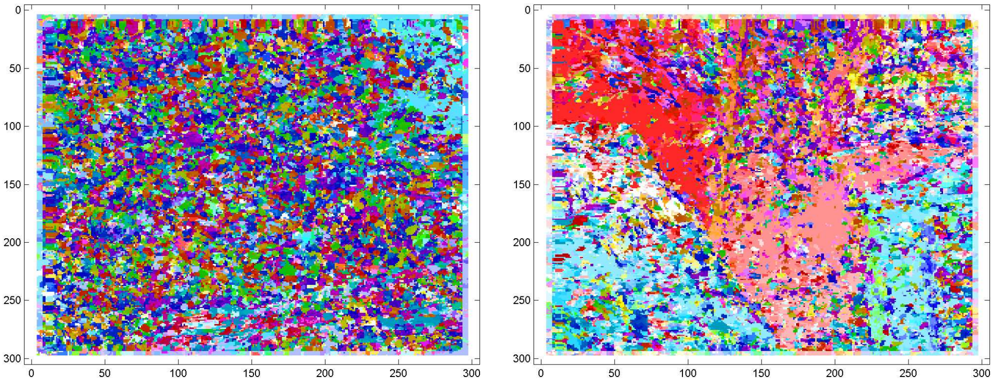
Figure 5.7: Optical flow fields for the Dimetrodon and Venus data sets for the vector **sum** using the diffusion distance based representation and a 5-by-5 neighbourhood of correlation vector to compute the mean.



Dimetrodon - Weighted Vector Sum

Venus - Weighted Vector Sum

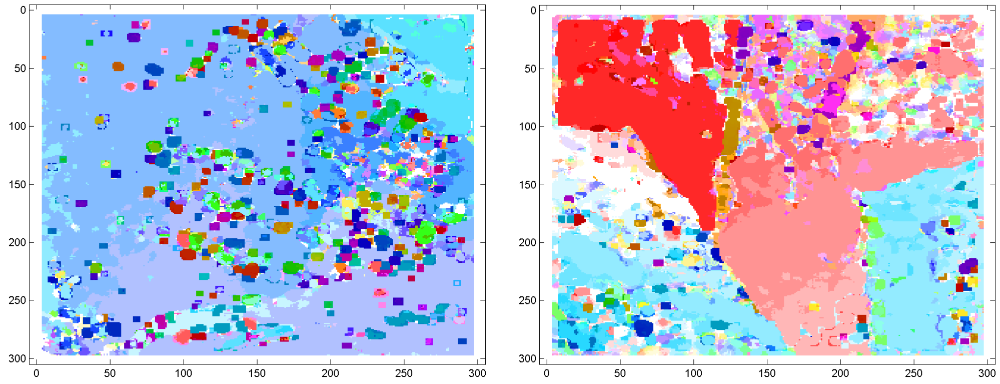
Figure 5.8: Optical flow fields for the Dimetrodon and Venus data sets for the **diffusion-weighted** sum using the diffusion distance based representation and a 5-by-5 neighbourhood of correlation vector to compute the weighted sum.



Dimetrodon - Marginal Median

Venus - Marginal Median

Figure 5.9: Optical flow fields for the Dimetrodon and Venus data sets for the **marginal median** using the diffusion distance based representation.



Dimetrodon - Vector Median

Venus - Vector Median

Figure 5.10: Optical flow fields for the Dimetrodon and Venus data sets for the **vector median** using the diffusion distance based representation.

in Figure 5.10 we see an estimate similar to Figure 5.6. Several smooth regions are larger in the Venus estimate, while regions of poorer estimates have appeared for both sequence for the vector median when compared with the vector sum.

Figures 5.11 and 5.12 highlight the benefit of using an aggregate of information rather than a single data point at two different points in the Dimetrodon data set. The left pair of images in the figures is the vector of the central pixel, while the right pair is the mean of the vectors in a 7-by-7 neighbourhood. In one case (Figure 5.11), the correct estimate will be made, although the peak is not particularly clear. However, in another case (Figure 5.12), there are incorrect estimates, not similar to the correct one, that will be selected.

Finally, we note that the estimates for the Venus sequence are worse than for the Dimetrodon sequence, although this is not necessarily because the data set is inherently difficult. We discuss potential reasons for this when we evaluate the different representations of the data in Section 5.6, where this effect is highlighted by comparing estimates using several different representations of the data.

Overall, we have shown there is a clear benefit to using larger neighbourhoods to aggregate correlational vector data. It is likely that when these neighbourhoods become too big, they begin to significantly overlap objects that may be moving in different directions, causing even poorer estimates

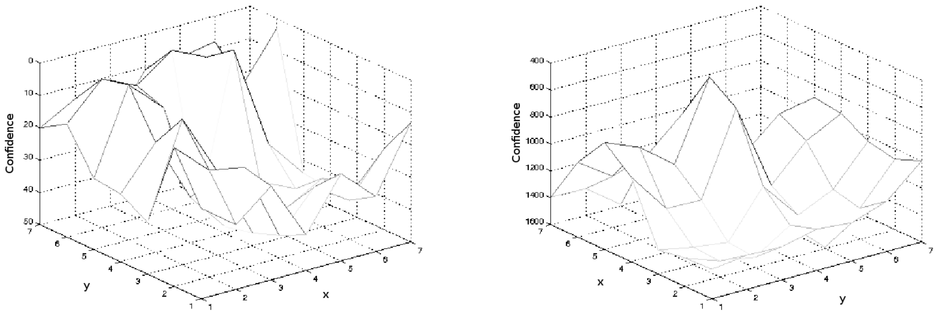


Figure 5.11: Meshes of 7-by-7 neighbourhoods of ordered correlations. The minimum rank, which is chosen as the estimate, is much clearer when taking the sum of the neighbourhood (right) rather than just using data from the cenrtal pixel (left).

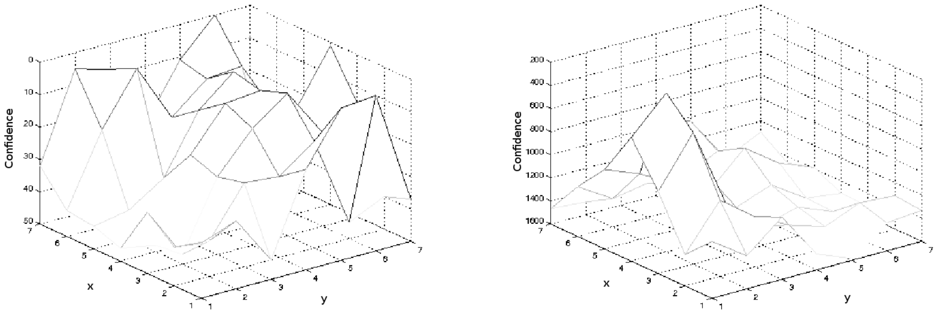


Figure 5.12: Same 7-by-7 meshes as above, but here using the vector from just the central pixel will result in a poor estimate due to noise (left), while the sum of the neighbourhood of vectors produces a clear peak.

at motion boundaries.

Next, we examine a postprocessing method that enhances the optical flow estimates in areas where the estimates are more ambiguous or less confident.

5.3 An effective postprocessing method

As was introduced in Section 4.6, we are using a postprocessing step to take advantage of the method described in that section for localizing high confidence (see equation 4.7) and accurate optical flow estimates. Here, we quantitatively show two effects of this type of estimate confidence. Firstly, we show that when the "true" estimate confidence ranks very high in terms of correlation in a given region, these areas yield low flow error rates. Secondly, we show that with decreasing confidence on the maximum correlation, there is an increase in the error rate.

Figures 5.13 and 5.14 shows these two effects. There is a fairly consistent decrease in both average errors (AE and FE) in both data sequences from the Middlebury data set. Furthermore, this decrease holds irrespective of representation of the data being correlated, even though the representations may be quite different (as is the case of the raw data). The minimum error is for the highest confidence bracket, which also hold the largest proportion of the data (Figure 5.15).

Figure 5.16 (a) depicts the confidence measure described in Figures 5.13-5.14. Lighter regions indicate higher degrees of confidence. Also shown is the motion estimate from the diffusion distance feature representation using the vector **sum** estimation method (b) and the ground truth flow (c). The colour coding for the flows is shown in Figure 5.1 (c).

These results motivate the usage of our postprocessing method. We have shown that the confidence measure corresponds to areas with small error rates and that a fairly sizeable proportion of the two images contain high confidence estimates. In order to produce a vector field with accurate flow estimates throughout, the regions of high confidence should be fairly evenly spread throughout the image. We see in Figure 5.16 that this is not really the case, especially in the Venus confidence map. There are large areas at the top-centre portion of the image where there are few high confidence pixels. Accordingly, the flow field is also much less smooth here and comparing with the ground truth in Figure 5.20 (b) by inspection, there are higher error rates.

This may be problematic for other images that have small regions moving independently as some may prove to have low confidence estimates that may in fact be accurate. The postprocessing will then fill in these regions

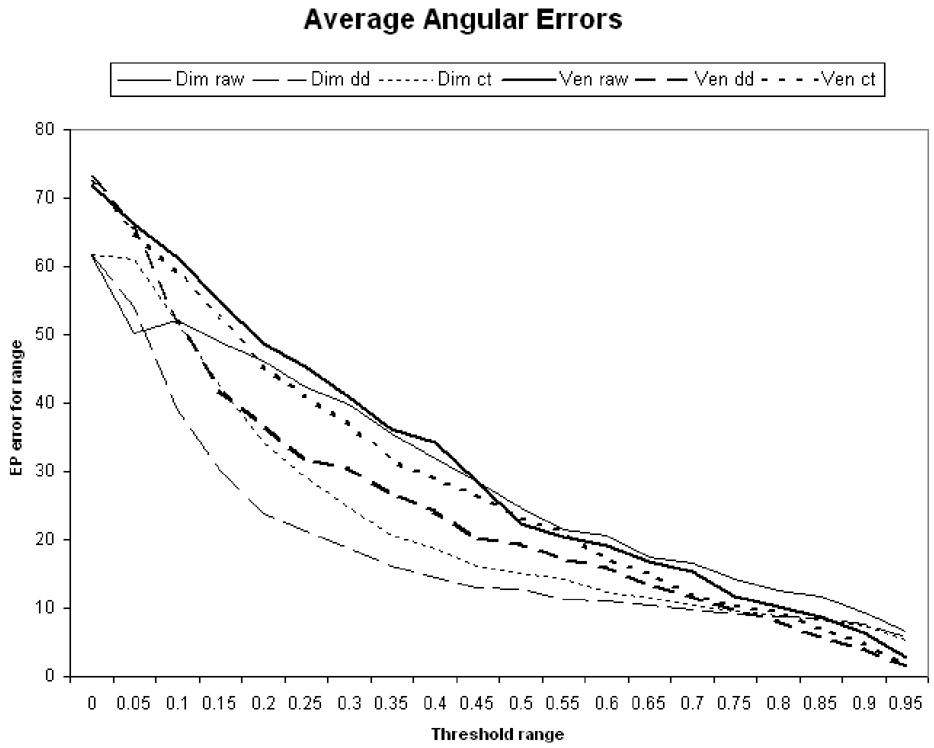


Figure 5.13: Angular error at different confidence thresholds: Higher confidence ranges, in general yield lower errors. They reflect a consensus in 7-by-7 sized neighbourhoods of the optical flow estimate.

with poor estimates. For this reason, methods based on energy minimization that compromise data correspondence with smoothness constraints may provide a mechanism to ensure good estimates with lower confidence are not unnecessarily overwritten.

It is also be possible to adapt the method in its present structure to produce a similar effect. We have chosen to use a strict threshold to select regions from which estimates spread, but a more flexible approach might yield more accurate estimates. By considering smaller confidence measurements and weighting them proportionately, non-integer estimates would arise, and good estimates with lower confidence could be taken into account. Such a method could be combined with other weighting factors including diffusion distances or the flow gradient, that would influence the rate of spreading in a particular direction. Such weighting factors are analogous to the constraints

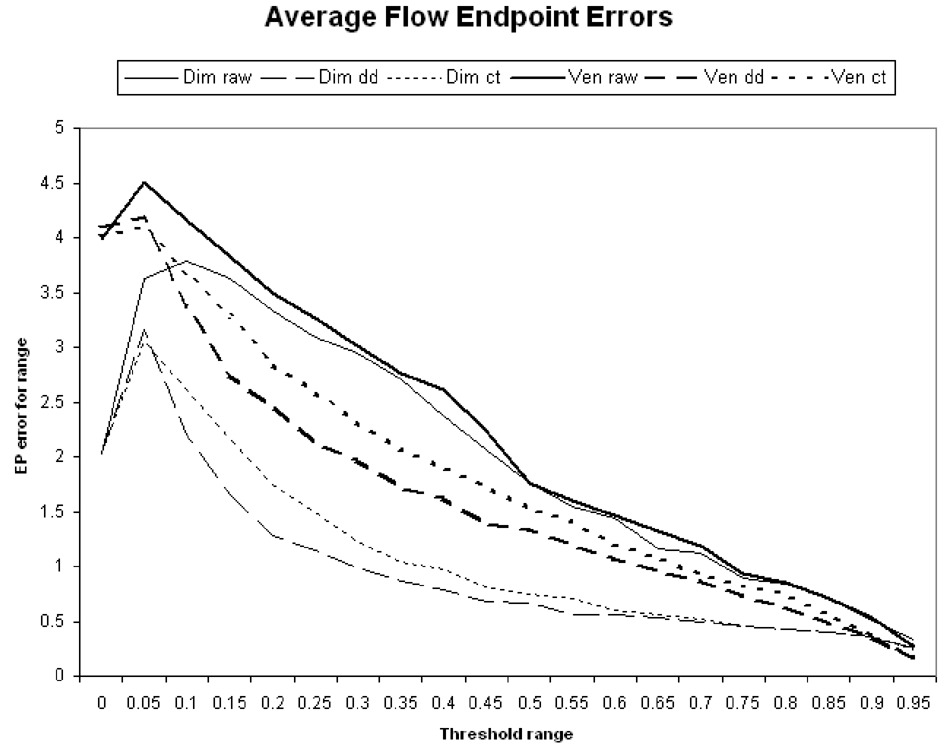


Figure 5.14: Endpoint error at different confidence thresholds: Higher confidence ranges, in general yield lower errors. They reflect a consensus in 7-by-7 sized neighbourhoods of the optical flow estimate.

in an energy minimization approach.

Next, we look in more detail at the results from the Dimetrodon sequence, we look at the Venus sequence from the Middlebury data set, and finally see how well the method performs on fluid motion data.

5.4 Object data - Middlebury data set

The standard data set for comparing optical flow estimation algorithms has been the Middlebury data set [46]. Since the original publication, a large variety of algorithms have been developed and tested against this set, some of which have been introduced in chapter 2. The data set includes ground truth data for which several types of error measures are calculated. The original image sequences have since been updated with newer image sequences.

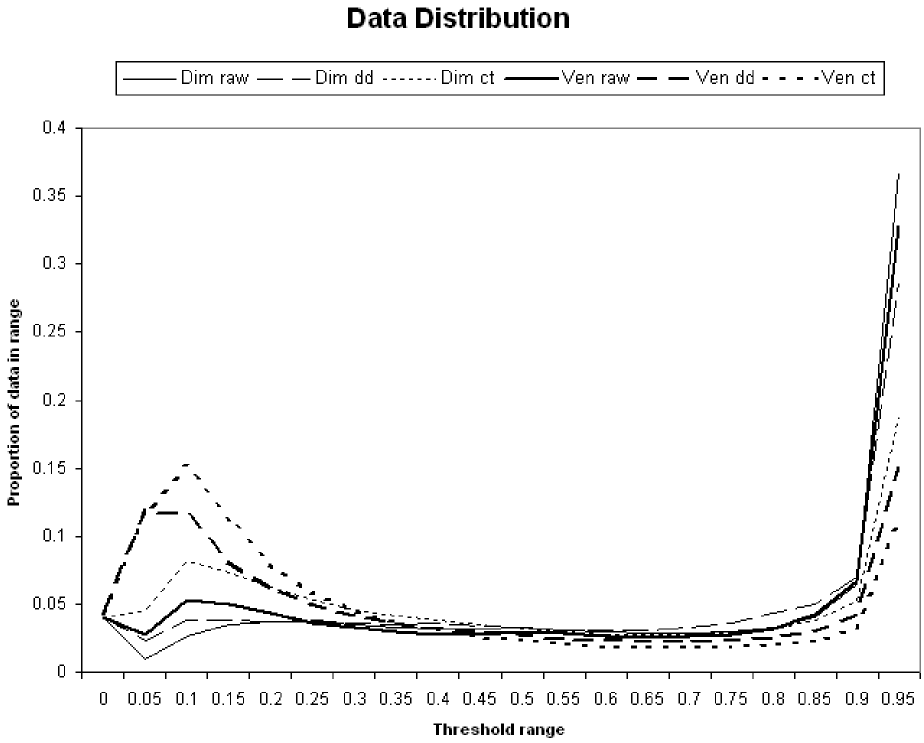
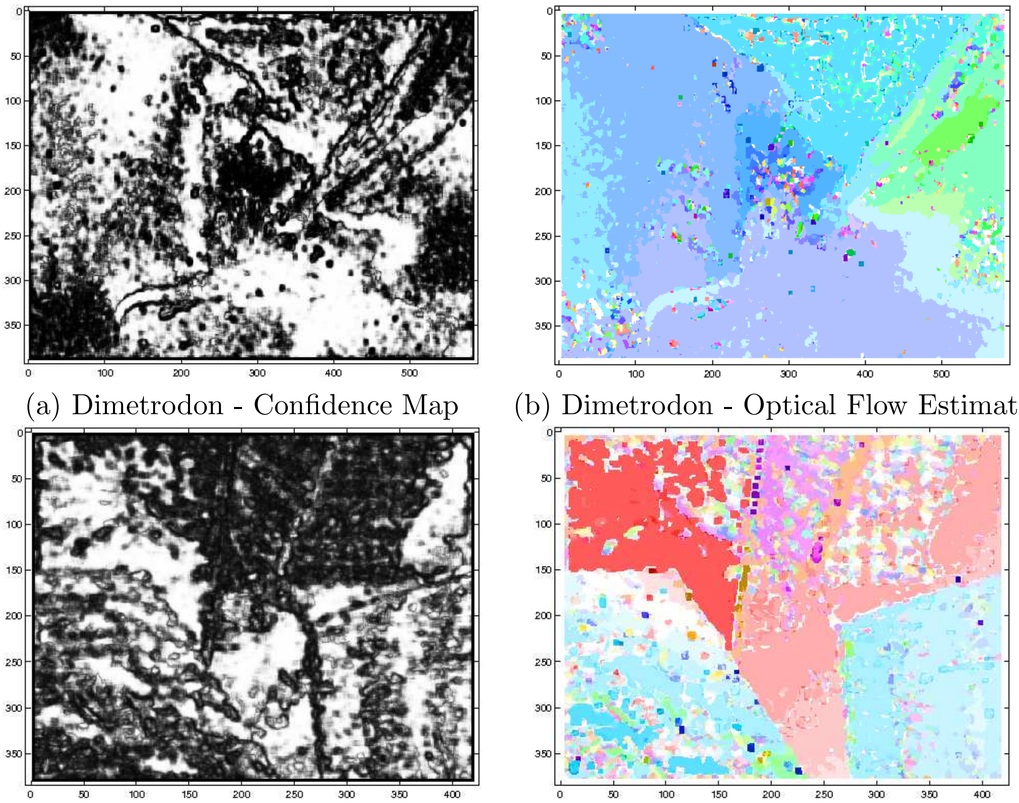


Figure 5.15: The proportion of data falling into each threshold range from Figures 5.13 and 5.14.

The newer set of image sequences use hidden texture and synthetic images that correspond to the Dimetrodon and Venus data sets respectively. A new class of images in a *high-speed camera* category include small regions in the image moving at high speeds, while much of the remainder of the scene remains stationary. This contrasts with the other images in the sequence that usually contain motion of small magnitude throughout the entire image. This type of data poses additional challenges for the flow estimation algorithms due to increased flow difference across object boundaries. The sequences can be found here [47].

In the present research, we used two sequences of images from the original comparison in [46], namely the Dimetrodon and Venus image sequences. The Dimetrodon sequence is a hidden fluorescent texture sequence, which is a real scene that has been spattered with fluorescent paint and photographed. Ground truth motion is computed by tracking the fluorescent paint which



(a) Dimetrodon - Confidence Map (b) Dimetrodon - Optical Flow Estimate

(c) Venus - Confidence Map (d) Venus - Optical Flow Estimate

Figure 5.16: **Dimetrodon:** (a) Map of confidence measurements from the definition in 4.7 (b) Initial flow estimate using sum of neighbourhood vectors **Venus:** (c) Map of confidence measurements from the definition in 4.7 (d) Initial flow estimate using sum of neighbourhood vectors

is used as a marker. This approach allows for the computation of ground truths from low texture data.

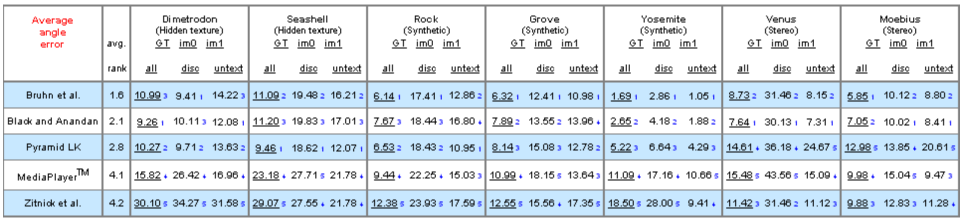
The Venus sequence is a synthetic scene generated using computer graphics. This method of generating images yields highly accurate ground truths and allows for the investigation of the accuracy of optical flow estimation when different types of noise, such as motion blur, are incorporated into the generated images.

We use the average angular (equation 5.1) and flow (equation 5.2) errors. Results for our method and those tested in the original evaluation paper by Baker et al [46] are shown in Table 5.1 for the Dimetrodon and Venus image sequences. We have also included error rates from the original set of

Table 5.1: Angular and flow errors from the Dimetrodon and Venus sequences for the method proposed in this thesis and methods from Baker et al [46].

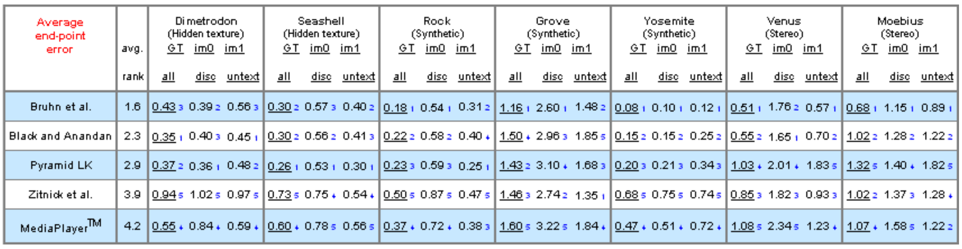
		Angular Error		Flow Error	
	avg rank	Dimetrodon	Venus	Dimetrodon	Venus
Black and Anandan	1.3	9.26_1	7.64_1	0.35_1	0.55_2
Bruhn et al	2.3	10.99_3	8.73_2	0.43_3	0.51_1
Pyramid LK	3.5	10.27_2	14.61_5	0.37_2	1.03_5
Proposed method	3.8	11.45_4	10.40_3	0.50_4	0.87_4
MediaPlayer^{TM}	5.0	15.82_5	15.48_6	0.94_6	0.85_3
Zitnick et al	5.3	30.10_6	11.42_4	0.55_5	1.08_6

sequences in Figure 5.18.



Average angle error	avg.	Dimetrodon (Hidden texture) GT im0 im1			Seashell (Hidden texture) GT im0 im1			Rock (Synthetic) GT im0 im1			Grove (Synthetic) GT im0 im1			Yosemite (Synthetic) GT im0 im1			Venus (Stereo) GT im0 im1			Moebius (Stereo) GT im0 im1		
	rank	all	disc	untext	all	disc	untext	all	disc	untext	all	disc	untext	all	disc	untext	all	disc	untext	all	disc	untext
Bruhn et al.	1.6	10.99_3	9.41_1	14.22_3	11.09_2	19.48_2	16.21_2	6.14_1	17.41_1	12.86_2	6.32_1	12.41_1	10.98_1	1.69_1	2.86_1	1.05_1	8.73_2	31.46_2	8.15_2	5.85_1	10.12_2	8.80_2
Black and Anandan	2.1	9.26_1	10.11_3	12.08_1	11.20_3	19.83_3	17.01_3	7.67_3	18.44_3	16.80_4	7.89_2	13.55_2	13.96_4	2.65_2	4.18_2	1.88_2	7.64_1	30.13_1	7.31_1	7.05_2	10.02_1	8.41_1
Pyramid LK	2.8	10.27_2	9.71_2	13.63_2	9.46_1	18.62_1	12.07_1	6.53_2	18.43_2	10.95_1	8.14_3	15.08_3	12.78_2	5.22_3	6.64_3	4.29_3	14.61_4	36.18_4	24.67_5	12.98_5	13.85_4	20.61_5
MediaPlayer^{TM}	4.1	15.82_4	26.42_4	16.96_4	23.18_4	27.71_5	21.78_4	9.44_4	22.25_4	15.03_3	10.99_4	18.15_5	13.64_3	11.09_4	17.16_4	10.66_5	15.48_5	43.56_5	15.09_4	9.98_4	15.04_5	9.47_3
Zitnick et al.	4.2	30.10_5	34.27_5	31.58_5	29.07_5	27.55_4	21.78_4	12.38_5	23.93_5	17.59_5	12.55_5	15.56_4	17.35_5	18.50_5	28.00_5	9.41_4	11.42_3	31.46_2	11.12_3	9.88_3	12.83_3	11.28_4

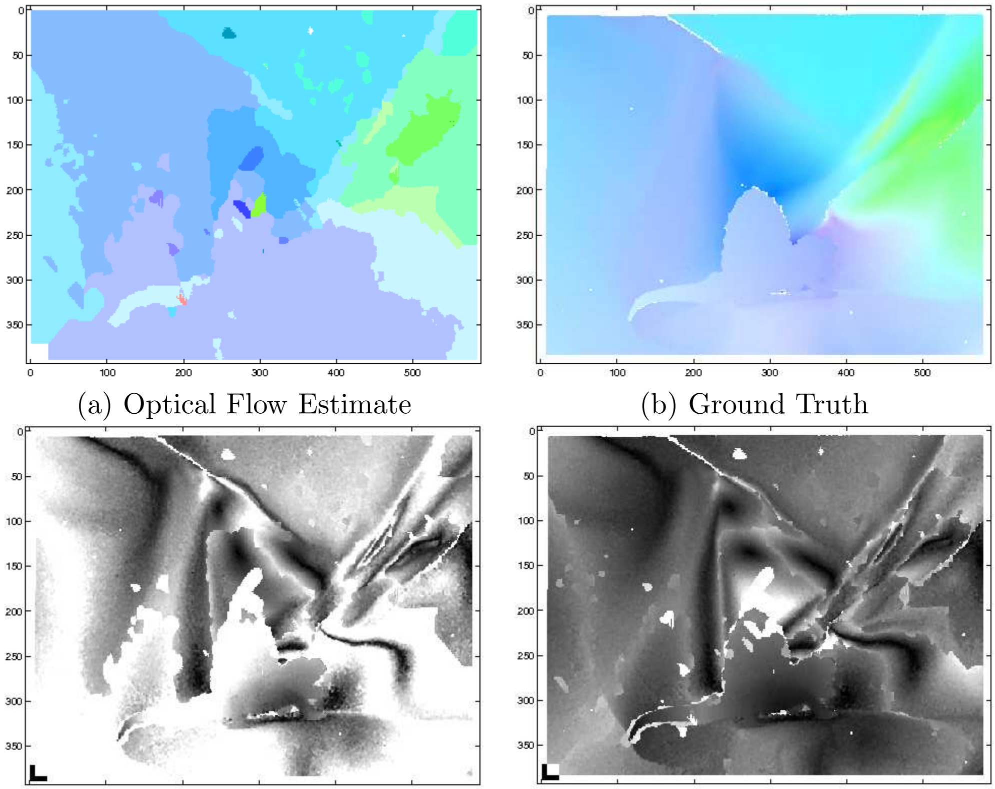
Figure 5.17: The average angular error (AE) measure for the Middlebury data set along with the scores for the top algorithms for each measure.



Average end-point error	avg.	Dimetrodon (Hidden texture) GT im0 im1			Seashell (Hidden texture) GT im0 im1			Rock (Synthetic) GT im0 im1			Grove (Synthetic) GT im0 im1			Yosemite (Synthetic) GT im0 im1			Venus (Stereo) GT im0 im1			Moebius (Stereo) GT im0 im1		
	rank	all	disc	untext	all	disc	untext	all	disc	untext	all	disc	untext	all	disc	untext	all	disc	untext	all	disc	untext
Bruhn et al.	1.6	0.43_3	0.39_2	0.56_3	0.30_2	0.57_3	0.40_2	0.18_1	0.54_1	0.31_2	1.16_1	2.60_1	1.48_2	0.08_1	0.10_1	0.12_1	0.51_1	1.76_2	0.57_1	0.68_1	1.15_1	0.89_1
Black and Anandan	2.3	0.35_1	0.40_3	0.45_1	0.30_2	0.56_2	0.41_3	0.22_2	0.58_2	0.40_4	1.50_4	2.96_3	1.85_5	0.15_2	0.15_2	0.25_2	0.55_2	1.65_1	0.70_2	1.02_2	1.28_2	1.22_2
Pyramid LK	2.9	0.37_2	0.36_1	0.48_2	0.26_1	0.53_1	0.30_1	0.23_3	0.59_3	0.25_1	1.43_2	3.10_4	1.68_3	0.20_3	0.21_3	0.34_3	1.03_4	2.01_4	1.83_5	1.32_5	1.40_4	1.82_5
Zitnick et al.	3.9	0.94_5	1.02_5	0.97_5	0.73_5	0.75_4	0.54_4	0.50_5	0.87_5	0.47_5	1.46_3	2.74_2	1.35_1	0.68_5	0.75_5	0.74_5	0.85_3	1.82_3	0.93_3	1.02_2	1.37_3	1.28_4
MediaPlayer^{TM}	4.2	0.55_4	0.84_4	0.59_4	0.60_4	0.78_5	0.56_5	0.37_4	0.72_4	0.38_3	1.60_5	3.22_5	1.84_4	0.47_4	0.51_4	0.72_4	1.08_5	2.34_5	1.23_4	1.07_4	1.58_5	1.22_2

Figure 5.18: The flow error (FE) measure for the original Middlebury data set along with the scores for the top algorithms for each measure as reported in [46].

The results from the present research are similar to the results from the best algorithms as published in that paper. Specifically, average angular and flow errors following postprocessing based on estimate confidence in Table 5.1, reveal an average AE values of 11.45 and 9.26, and average FE values of 0.50 and 0.35, respectively, for our method and the best reported value in [46] which is from Black and Anandan's algorithm [18] in the Dimetrodon image sequence. We have shown the estimates from the Dimetrodon image sequence in Figure 5.19. The estimate and ground truth are coded using the colour map shown in Figure 5.1. The angular and flow errors are coded such that white regions represent 10 degrees and 1 unit (Euclidean) error.



(a) Optical Flow Estimate

(b) Ground Truth

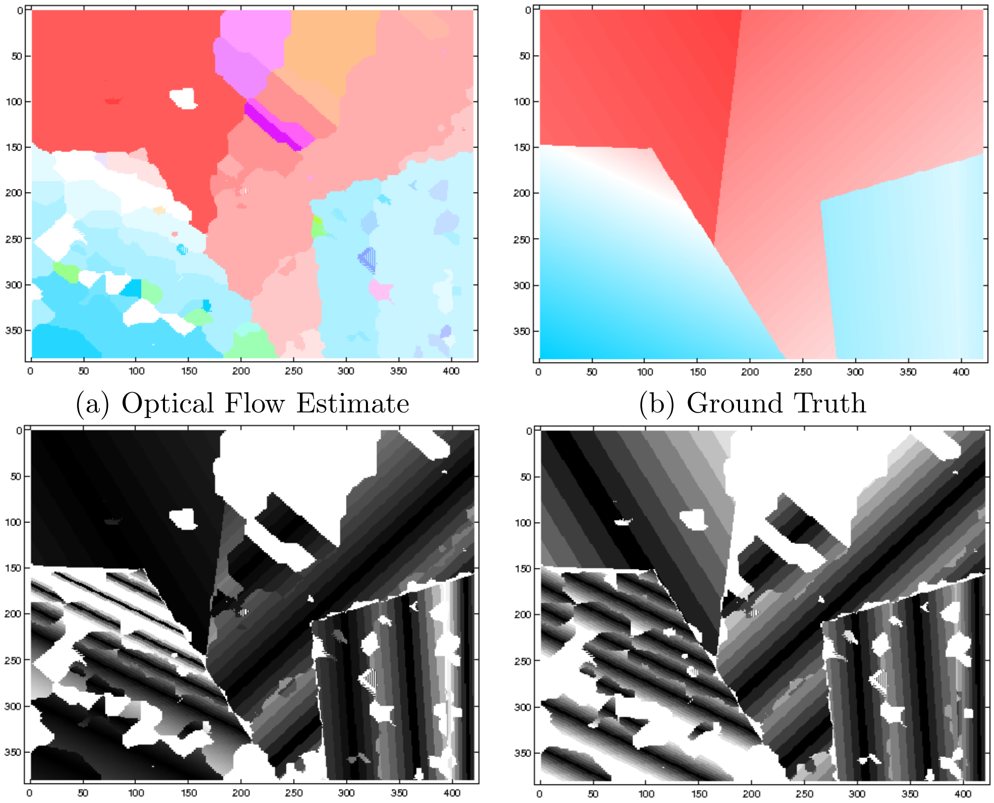
(c) Angular Error

(d) Flow Error

Figure 5.19: Dimetrodon: Estimate following postprocessing (a) along with the ground truth flow (b) with the colour map in Figure 5.1. We have also drawn the angular error (c) with white regions marking 10 degrees or greater error and darker regions with progressively less error. Similarly, the flow error has been drawn in (d) with white regions marking errors greater than 1.

In the Venus image sequence, average AE values of 10.40 and average

FE values of 0.87 for our method also compare favourably with results from algorithms tested in [46]. As with the Dimetrodon sequence, we have shown the estimates from the Venus image sequence in Figure 5.20. The estimate and ground truth are coded using the colour map shown in Figure 5.1. The angular and flow errors are coded such that white regions represent 10 degrees and 1 unit (Euclidean) error.



(a) Optical Flow Estimate

(b) Ground Truth

(c) Angular Error

(d) Flow Error

Figure 5.20: Venus: Estimate following postprocessing (a) along with the ground truth flow (b) with the colour map in Figure 5.1. We have also drawn the angular error (c) with white regions marking 10 degrees or greater error and darker regions with progressively less error. Similarly, the flow error has been drawn in (d) with white regions marking errors greater than 1.

An important aspect to note about the postprocessed results is that the propagated estimates do not extend far, if at all, past the object boundaries. In the Dimetrodon sequence, this can be seen at the dinosaur's tail in Figure 5.19. Looking at the confidence map in Figure 5.16, we see that high confidence estimates just inside and just outside the boundaries of the tail have resulted in a balance that prevented undue propagation.

Similarly, for the Venus sequence, we observe in Figure 5.20 that at the boundaries of the newspapers on the bottom left and bottom right, the boundaries are conserved in the optical flow field. Again, this ties back to the confidence map having high confidence estimates on both sides of the boundaries as seen in Figure 5.16.

Thus we have comparable results to existing algorithms and we proceed to examine the applicability of the method to a different class of data - fluid motion.

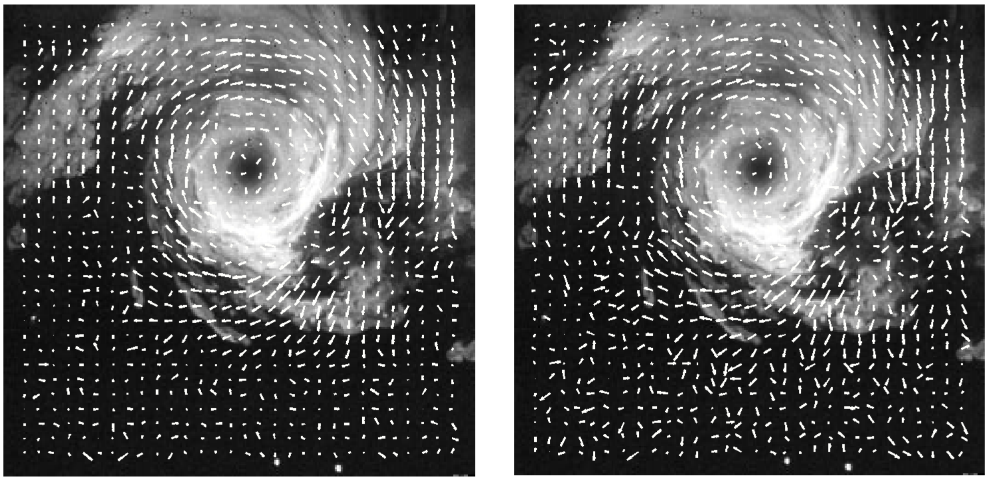
5.5 Fluid data - cloud motion

The fluid flow data set from [22] contains a storm cloud rotating clockwise around its eye. Unlike the object data in the previous section, the boundaries in this image sequence are much less clear as thin and thick layers of clouds occupy nearly the entire image. The complexity of this data sequence lies in the changing and continuous nature of the features. This type of sequence is particularly suitable for the present method, which we have already shown to be robust to noisy data and well suited for describing continuous features based on intensity gradients.

In Figures 5.21, 5.22 and 5.23 we show the results of applying our method to the fluid motion data. In Figure 5.21 are the estimates using the vector sum estimation method. In Figure 5.22 are the same data, but have been smoothed by convolution with a 20-by-20 uniform filter, while in Figure 5.23 are the results after postprocessing. The smoothed and postprocessed results show a good resemblance to the clockwise rotation of the storm clouds (see animation: [48]). Furthermore, following postprocessing, Figure 5.21 shows little or no motion in the bottom third of the image where there are no clouds and hence there should be no motion noted there. Results in Cuzol et al [49] do not motion in areas where there are no clouds due to the nature of their algorithm.

Cuzol et al's [22] optical flow estimation involves an non-linear optimization of parameters that describe the motion in terms of vorticity and divergence, that is essentially a regularization of the optical flow field with respect to these two constraints. The proposed algorithm is a variation on the brightness consistency assumption, with the DFRs used for the motion estimation instead of pixel intensity information. After postprocessing and scale factor equal to 0.0001 (other parameters are also the same as for the Middlebury data set), we see in Figure 5.24 (a) that the estimates of optical flow largely resemble those of the algorithm by Cuzol et al. (Figure 5.24 (b)).

Thus, we have shown that our method works for a variety of data, including low and high-textured data as well as fluid motion data.



(a) Scale factor $a = 0.001$

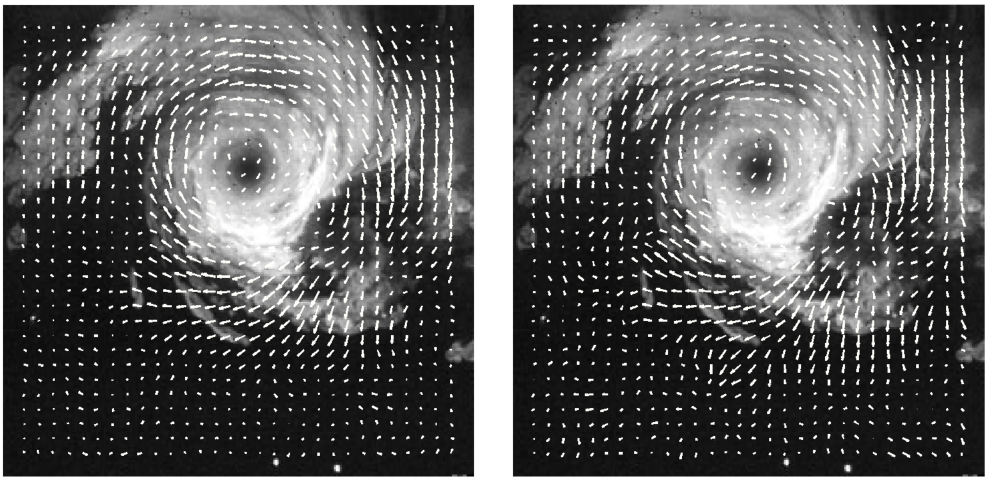
(b) Scale factor $a = 0.0001$

Figure 5.21: Flow estimates for scale factor values $a = 0.001$ and $a = 0.0001$ using the vector sum.

5.6 Comparison to other representations and estimation methods

The results in this chapter have thus far described only one representation of the image data combined with an estimation method. However, several different representations with different parameters were investigated in conducting this research (see Chapter 4). In this section we compare representations based on diffusion distances and commute times, as well as simply using the raw image data in the calculations for the correlation matrix.

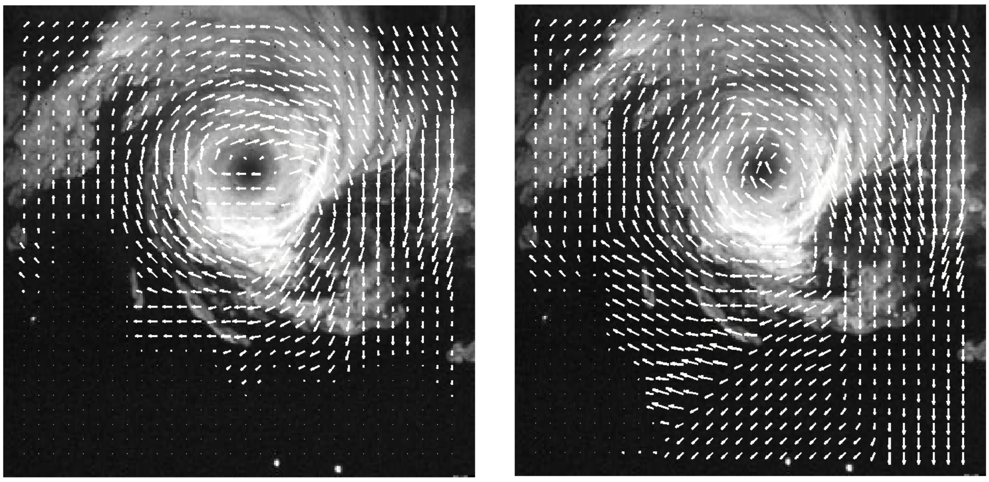
First we show results from several parameter manipulations. The parameters we looked at in more detail are the size of the diffusion feature



(a) Scale factor $a = 0.001$

(b) Scale factor $a = 0.0001$

Figure 5.22: Flow estimates for scale factor values $a = 0.001$ and $a = 0.0001$ using the vector sum and have been filtered with a 20-by-20 uniform filter.



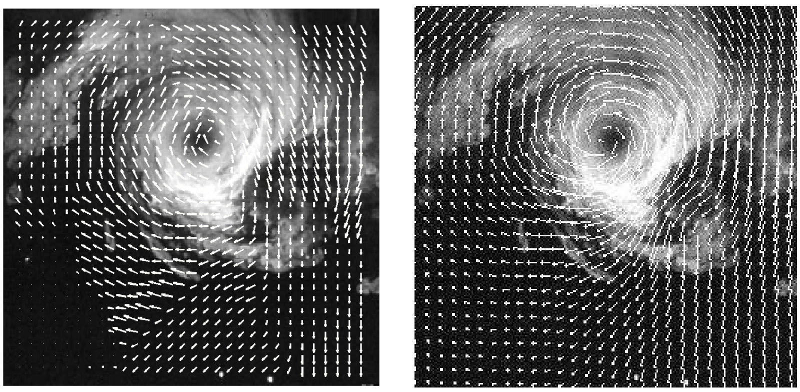
(a) Scale factor $a = 0.001$

(b) Scale factor $a = 0.0001$

Figure 5.23: Flow estimates for scale factor values $a = 0.001$ and $a = 0.0001$ after our postprocessing method has been applied.

window and the scale factor, both of which affect the diffusion feature.

In general, there is an optimal medium for selecting the size of the comparison window used in a correlation or energy minimization as in Bruhn et al [28]. Bruhn and colleagues used an isotropic Gaussian function to select



(a) Our algorithm (b) Cuzol et al (2005)

Figure 5.24: (a) The diffusion feature representation - DFR - estimate of motion with postprocessing. (b) Optical flow estimates from Cuzol et al's work [49].

Table 5.2: Angular and flow errors with variable feature sizes.

		Dimetrodon		Venus	
Representation	Feature Size	AE	FE	AE	FE
Raw Image Data	5x5	1.36	21.07	1.69	23.52
	9x9	3.11	39.41	1.52	18.23
Diffusion Distances	5x5	0.76	15.32	2.00	30.70
	9x9	2.06	29.22	1.95	26.38
Commute Times	5x5	1.14	23.63	2.43	38.54
	9x9	4.03	48.36	3.05	36.93

the feature size, which was bigger than the original single pixel data term used by Horn and Schunck [6]. We used two feature, or *diffusion window* (see Section 4.4) sizes, that are then correlated. These are of size 5-by-5 and 9-by-9.

Our results show (in Table 5.2) that in fact using the smaller feature size produces no advantages in terms of error rates rather than the larger one. In fact, for the Dimetrodon data, using a larger feature size results in a large increase in error rates. This is less true for the Venus data set as the errors are about the same or slightly smaller for the larger feature size.

A second parameter that we investigated deeper is the scale factor from the initial adjacency matrix relating the points in the image based on their

Table 5.3: Angular and flow errors for the three representations and scale factor values a=0.001 and a=0.0001.

		Dimetrodon		Venus	
Representation	scale factor	AE	FE	AE	FE
Raw Image Data	n/a	21.0652	1.3595	23.5198	1.6883
Diffusion Distances	0.0001	15.3172	0.7552	30.6981	1.9998
	0.001	23.787	1.0757	35.5188	2.242
Commute Times	0.0001	23.629	1.1412	38.5388	2.4302
	0.001	23.1382	1.1543	34.041	2.1889

brightness (see equation 3.4). A larger scale factor results in a higher measure of similarity for points with bigger brightness differences. This is because the adjacency matrix is normalized to 1. We have plotted the functions relating points by their intensity in Figure 4.1.

Because the higher scale parameter discriminates less poorly between brightness intensities, the representation of the Dimetrodon images, which is largely composed of low texture data, is poorer. This leads to a bigger deterioration in the error rates than for the Venus sequence, which has more clear textures and higher contrasts between regions in an image. Table 5.3 shows results from two scale parameter values, 0.001 and 0.0001, using the vector sum estimation method described in Section 4.5.

Table 5.3 also shows results prior to postprocessing for the three representations based on feature vectors of the raw image data, diffusion distances and commute times. We note especially the improvement in estimate when using the diffusion distance feature vectors compared with the raw image data in the Dimetrodon sequence. As previously mentioned, this sequence contains mainly low textured data which, when noisy, can result in a noisy correlation matrix. Using diffusion distances is an attempt to mitigate this effect by incorporating a broader stretch of the features around a given point, which may overcome the noise in some cases.

In the Venus image sequence, there is a much better defined texture and the improvement from using diffusion distances is not apparent in this case. In fact, using diffusion distances yields a poorer result in terms of error rates. This is because the transformation of the data is lossy and some details that would have improved the correlations between image regions are lost.

Thus, we see there are some limitations to using the diffusion distance feature vectors over the raw image data.

5.7 Summary

In this chapter, we have verified that the three steps in our optical flow estimation algorithm can produce good flow estimates for different types of data.

In Section 5.1, we showed how the representation based on diffusion distances represents various types of features, including well defined features such as a car boot, and low texture features such as a road with a vertical intensity gradient, or a segment of a cloud.

Next, we evaluated several statistical estimation methods in Section 5.2. For each method, we compared the average angular and average flow errors using the Dimetrodon and Venus image sequences. This allowed us to compare these methods for two data sets with different degrees of texture.

Finally, in Section 5.3 we showed that our confidence measure can identify pixels with low error rates. We use an isotropic propagation method to fill in areas with lower confidence values to obtain a more accurate optical flow field.

It is worth noting the robustness of our postprocessing step. The simple method proposed has been used in very different types of data, including object data from the Middlebury data set as well as fluid motion data used by Cuzol et al [49].

In the next chapter, we summarize these results in more detail and discuss several extensions and improvements to the estimation algorithm presented in this thesis.

Chapter 6

Conclusions

6.1 Summary

In this thesis, we have reviewed the general approaches to optical flow estimation algorithms, as well as manifold learning algorithms in order to examine the possibility of applying the latter to optical flow estimation. We reviewed flow algorithms that use regularization, feature based algorithms as well as probabilistic models for optical flow estimation. Manifold learning was introduced, along with two distance metrics defined in the low dimensional manifold, namely diffusion distances and commute times.

Our present results have shown how the diffusion framework can be used for dense optical flow estimation. An anisotropic kernel is used to define the similarity between pairs of pixels in a neighbourhood. A Markov chain is used in order to model the diffusion process beyond the given neighbourhood. Finally, diffusion distances are computed to calculate the relation between pairs of pixels by considering the overall connectivity of the graph.

Diffusion distances are used to model local image features, and hence can serve as a less noisy fundamental unit in optical flow estimation than raw pixel intensities. This increased robustness can be used to improve optical flow estimates of existing regularization schemes. Correlations of locally defined diffusion distances are used for dense optical flow estimation. The proposed methodology is applied on various image sequences including complex atmospheric movement.

The resulting vector field are used as initialization for a smoothing algorithm based on a confidence measure that we have also proposed here. This iterative algorithm uses the confidence defined at points in the image and propagates estimates at points with high confidence isotropically. This is repeated until all points have estimates that correspond to high confidence measures associated with the initial estimate or one that was propagated to that point.

The algorithm we propose in this thesis computes measures such as diffusion distances in Equation 3.6, our confidence measure in Equation 4.7, and the correlation matrix, that could be used in other applications. Commute times, which are related to diffusion distances (see Section 3.5) have been shown to be applicable to scene segmentation by Qiu and Hancock [13]. Similarly, the correlation matrix can be used for segmentation based on the similarities between the motion estimates as demonstrated by Robles-Kelly et al.[14].

We tested our algorithm on two sequences from the Middlebury data set [46] and a fluid motion image sequence. When comparing the diffusion distance based representation of the image to using the raw image data, we observed a bigger improvement in the Dimetrodon sequence than the Venus sequence, when using the diffusion distance based representation. We attribute this effect to the nature of the data. Namely, that the Dimetrodon sequence comprises low textured image, while the Venus sequence image have clearer textures.

We discuss in more detail in the next section how these measures can be extended, and how various aspects of our algorithm can be extended or adapted to fit into existing algorithms for optical flow estimation.

6.2 Future Work

There are still many aspects of the diffusion distance and commute time based feature representations and their ability to represent noisy features that need to be assessed. The optical flow estimation methods where these representations could be applied have also not been investigated. And finally, while we have assessed several estimation methods using the correlation matrix, the postprocessing method used in the present algorithm uses the confidence measures in a simple way that could in future research be replaced by a more refined method, potentially yielding better results.

In the results from work to date, we see that diffusion distances are well suited for representing features around a certain point in an image. This result is, however, qualitative at present, but can be tested quantitatively. One way of doing so is to take the central point in a circle or square, a observe the change in local diffusion distances as the point is shifted closer to the edges of the shape. The sensitivity of the diffusion distances will depend on the scale-related parameters t from the Markov process.

We use the correlation between DFRs to construct the matrix from which optical flow estimates are inferred. However, the correlation approach to optical flow estimation is not a particularly fast or sophisticated one and could be replaced by other methods for the computation of the correlation matrix or its analogue. One such way would be to extend the correlational approach by including steps to exclude outliers that will skew correlations, or to replace correlation altogether with a different relation metric such as least squares.

A second alternative to using correlation is to incorporate the DFR representations into an existing optical flow estimation algorithm. Various past algorithms use constraint based energy minimization that contain a term based on the image data. We have shown in this thesis improvements in the flow estimate using the correlation based approach, where the image data has been replace by the DFR. In a similar way, the data term in the energy minimization approaches could be replaced by the diffusion distance based representation.

Such a replacement extends beyond the correlation step into the aggregation and estimate spreading analogues of our method, as the energy minimization is an iterative approach that not only matches the data terms, but also regularizes the flow field. Even though energy minimization algorithms are not structured sequentially like the present algorithm is, aspects such as estimate aggregation and use of confidence measures can be incorporated into these methods. For example, we have mentioned a confidence measure in Section 4.6 noted by Bruhn and colleagues [28]. However, those authors have only mentioned the measure to track the confidence of estimates at any point in the image, without necessarily using the confidence. In the present work, we have applied the confidence measure to affect the spreading of estimates throughout the flow field. The diffusion distance can also be used as the regularization term in an energy equation similar to the anisotropic vector gradient norm used by [19]. Finally, other methods, including probabilistic methods remain to be explored as well, in terms of how the diffusion distance representation could be incorporated into the flow estimation.

With all these replacements of the correlation based method, it is important to have a better understanding of how diffusion features describing similar objects relate to each other. Correlation has proven successful, but the reasons for this success will remain unclear without a more detailed investigation of the structure and statistical properties of the diffusion features.

The present method does not incorporate any smoothness measure than might eliminate problematic small areas that retain an incorrect estimate. The method also takes only integer estimates of motion due to maximum statistic from the processed vector and propagates this integer estimate. A better method would take weighted sums based on confidence, and perhaps include a term related to the gradient to resolve the smoothness issue.

In the present work, a two dimensional kernel was used to compute the diffusion distances. This kernel is well suited to describe the features of a particular image, though optical flow occurs in three dimensions because of the extra time dimension. Thus, a natural extension of the work is to use a three dimensional kernel to estimate how the intensity of the image (i.e., the mass of the objects in the image) diffuses in time. Such an approach would likely include several frames of the image sequence in the diffusion maps.

We can also see similarities of this approach to eigenvalue methods by [26] and [27]. These eigenvalue methods only determine the main direction of the feature in a small (though adaptively sized) window, but diffusion distances are defined between the central point and all other points in the window, resulting in significantly richer feature information. The feature information from the eigendecomposition of the adjacency matrix defined in our method may or may not benefit from further feature detection were diffusion distances to be used in the place of image gradient information in an eigenvalue method.

Overall, the fact that we have proposed a new way of representing the data that can improve the optical flow estimate of a previously used algorithm, there are many possibilities that remain to be explored to see how the new representation might be used in other algorithms.

Bibliography

[1] R.R. Coifman and S. Lafon. Diffusion maps. *Applied and Computational Harmonic Analysis*, 21(1):5–30, 2006.

[2] S.T. Roweis and L.K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, 2000.

[3] M. Belkin and P. Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 15(6):1373–1396, 2003.

[4] D.L. Donoho and C. Grimes. Hessian eigenmaps: Locally linear embedding techniques for high-dimensional data. *Proceedings of the National Academy of Sciences*, 100(10):5591–5596, 2003.

[5] Y. Bengio and M. Monperrus. Non-local manifold tangent learning. In *Proceedings of Advances in Neural Information Processing Systems*, pages 129–136. MIT Press, 2005.

[6] B.K.P. Horn and B.G. Schunck. Determining optical flow. *Computer Vision*, 17:185–203, 1981.

[7] T. Corpetti, E. Memin, and P. Perez. Dense estimation of fluid flows. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(3):365–380, 2002.

[8] B.D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *In Proceedings of the International Joint Conference on Artificial Intelligence*, volume 81, pages 674–679, 1981.

[9] G. Le Besnerais and F. Champagnat. Dense optical flow by iterative local window registration. In *Proceedings of the IEEE Computer Society Conference on Image Processing*, volume 1, page 493, 2005.

[10] CW Zitnick, N. Jojic, and S.B. Kang. Consistent segmentation for optical flow estimation. In *Proceedings of the IEEE International Conference on Computer Vision*, volume 2, pages 1308–1315, 2005.

[11] X. Ren. Local grouping for optical flow. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2008.

[12] J. Malik, S. Belongie, T. Leung, and J. Shi. Contour and texture analysis for image segmentation. *International Journal of Computer Vision*, 43(1):7–27, 2001.

[13] H. Qiu and E.R. Hancock. Clustering and embedding using commute times. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(11):1873, 2007.

[14] A. Robles-Kelly, A. Bors, and E. Hancock. Hierarchical iterative eigendecomposition for motion segmentation. In *IEEE ICIP*, volume 2, pages 363–366. Citeseer, 2001.

[15] R. Vidal, R. Tron, and R. Hartley. Multiframe motion segmentation with missing data using PowerFactorization and GPCA. *International Journal of Computer Vision*, 79(1):85–105, 2008.

[16] J.D. Courtney. Automatic video indexing via object motion analysis. *Pattern Recognition*, 30(4):607–625, 1997.

[17] S. Dagtas, W. Al-Khatib, A. Ghafoor, RL Kashyap, P. Res, and B. Manor. Models for motion-based video indexing and retrieval. *IEEE Transactions on Image Processing*, 9(1):88–101, 2000.

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

[19] D. Tschumperlé and R. Deriche. Orthonormal vector sets regularization with PDE's and applications. *International Journal on Computer Vision*, 50(3):237–252, 2002.

[20] A. Wedel, D. Cremers, T. Pock, and H. Bischof. Structure- and motion-adaptive regularization for high accuracy optic flow. In *IEEE International Conference on Computer Vision (ICCV)*, Kyoto, Japan, 2009.

[21] L. Xu, J. Chen, and J. Jia. A Segmentation Based Variational Model for Accurate Optical Flow Estimation. In *Proceedings of the 10th European Conference on Computer Vision*, pages 671–684. Springer-Verlag Berlin, Heidelberg, 2008.

[22] A. Cuzol, P. Hellier, and E. Mémin. A low dimensional fluid motion estimator. *International Journal of Computer Vision*, 75(3):329–349, 2007.

[23] D. Castelow, DW Murray, GL Scott, and BF Buxton. Matching Canny edgels to compute the principal components of optic flow. *Image and Vision Computing*, 6(2):129–135, 1988.

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

[25] R. Szeliski and J. Coughlan. Spline-Based Image Registration. *International Journal of Computer Vision*, 22(3):199–218, 1997.

[26] H. Spies and H. Scharr. Accurate optical flow in noisy image sequences. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 587–592, 2001.

[27] H. Liu, R. Chellappa, and A. Rosenfeld. Accurate dense optical flow estimation using adaptive structure tensors and a parametric model. *IEEE Transactions on Image Processing*, 12(10):1170–1180, 2003.

[28] A. Bruhn, J. Weickert, and C. Schnörr. Lucas/Kanade meets Horn/Schunck: Combining local and global optic flow methods. *International Journal of Computer Vision*, 61(3):211–231, 2005.

[29] V. Lempitsky, S. Roth, and C. Rother. FusionFlow: Discrete-continuous optimization for optical flow estimation. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2008.

[30] B. Glocker, N. Paragios, N. Komodakis, G. Tziritas, and N. Navab. Optical flow estimation with uncertainties through dynamic MRFs. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2008.

[31] N. Komodakis, G. Tziritas, and N. Paragios. Fast, approximately optimal solutions for single and dynamic MRFs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2007.

[32] A.B. Chan and N. Vasconcelos. Modeling, Clustering, and Segmenting Video with Mixtures of Dynamic Textures. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 909–926, 2008.

[33] J.B. Tenenbaum, V. Silva, and J.C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000.

[34] R. Basri, D. Roth, and D. Jacobs. Clustering appearances of 3D objects. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 414–420, 1998.

[35] G.E. Hinton, M. Revow, and P. Dayan. Recognizing handwritten digits using mixtures of linear models. *Advances in Neural Information Processing Systems*, pages 1015–1022, 1995.

[36] R. Hecht-Nielsen. Replicator neural networks for universal optimal source coding. *Science*, 269(5232):1860–1863, 1995.

[37] C.M. Bishop, M. Svensén, and C.K.I. Williams. GTM: The generative topographic mapping. *Neural Computation*, 10(1):215–234, 1998.

[38] G. Langs and N. Paragios. Modeling the structure of multivariate manifolds: Shape maps. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2008.

[39] J. Lefevre and S. Baillet. Optical Flow and Advection on 2-Riemannian Manifolds: A Common Framework. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(6):1081–1092, 2008.

[40] Jane K. Cullum and Ralph A. Willoughby. *Lanczos Algorithms for Large Symmetric Eigenvalue Computations: Vol. I: Theory.* Society for Industrial and Applied Mathematics, 2002.

[41] W. E. Arnoldi. The principle of minimized iterations in the solution of the matrix eigenvalue problem. *Quarterly of Applied Mathematics*, 9:17–29, 1951.

[42] S. Lafon, Y. Keller, and R.R. Coifman. Data fusion and multicue data matching by diffusion maps. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(11):1784–1797, 2006.

[43] S. Lafon and A.B. Lee. Diffusion maps and coarse-graining: A unified framework for dimensionality reduction, graph partitioning, and data set parameterization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(9):1393–1403, 2006.

[44] J. Astola, P. Haavisto, and Y. Neuvo. Vector median filters. *Proceedings of the IEEE*, 78(4):678–689, 1990.

[45] JL Barron, DJ Fleet, and SS Beauchemin. Performance of optical flow techniques. *International journal of computer vision*, 12(1):43–77, 1994.

[46] S. Baker, D. Scharstein, JP Lewis, S. Roth, M.J. Black, and R. Szeliski. A database and evaluation methodology for optical flow. In *Proc. ICCV*, volume 5. Citeseer, 2007.

[47] Flow accuracy and interpolation evaluation, http://vision.middlebury.edu/flow/eval/, December 2009.

[48] A. Cuzol and E. Mémin. A low dimensional motion estimator for fluid flows, http://www.fluid.irisa.fr/demo/annecuzol/estimation.html, 2006.

[49] A. Cuzol and E. Memin. Vortex and source particles for fluid motion estimation. In *Proceedings of the 5th Int. Conf. on Scale-Space and PDE methods in Computer Vision*, pages 254–266. Springer, 2005.