ABSTRACT

Title of dissertation: Measuring Deformations and Illumination Changes in Images with Applications to Face Recognition

Anne Jorstad, Doctor of Philosophy, 2012

Dissertation directed by: Professor David Jacobs
Department of Computer Science and
University of Maryland Institute for
Advanced Computer Studies

This thesis explores object deformation and lighting change in images, proposing methods that account for both variabilities within a single framework. We construct a deformation- and lighting-insensitive metric that assigns a cost to a pair of images based on their similarity. The primary applications discussed will be in the domain of face recognition, because faces provide a good and important example of highly structured yet deformable objects with readily available datasets. However, our methods can be applied to any domain with deformations and lighting change. In order to model variations in expression, establishing point correspondences between faces is essential, and a primary goal of this thesis is to determine dense correspondences between pairs of face images, assigning a cost to each point pairing based on a novel image metric.

We show that an image manifold can be defined to model deformations and illumination changes. Images are considered as points on a high-dimensional man-

ifold given local structure by our new metric, where costs are based on changes in shape and intensity. Curves on this manifold describe transformations such as deformations and lighting changes to connect nearby images, or larger identity changes connecting images far apart. This allows deformations to be introduced gradually over the course of several images, where correspondences are well-defined between every pair of adjacent images along a path. The similarity between two images on the manifold can be defined as the length of the geodesic that connects them.

The new local metric is validated in an optical flow-like framework where it is used to determine a dense correspondence vector field between pairs of images. We then demonstrate how to find geodesics between pairs of images on a Riemannian image manifold. The new lighting-insensitive metric is described in the wavelet domain where it is able to handle moderate amounts of deformation, and allows us to derive an algorithm where the analytic geodesics between images can be computed extremely efficiently. To handle larger deformations in addition to changes in illumination, we consider an algorithmic framework where deformations are modeled with diffeomorphisms. We present preliminary implementations of the diffeomorphic framework, and suggest how this work can be extended for further applications.

Measuring Deformations and Illumination Changes in Images with Applications to Face Recognition

by

Anne Jorstad

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2012

Advisory Committee:
Professor David Jacobs, Chair/Advisor
Professor Philippe Burlina
Professor Doron Levy
Professor Dianne O'Leary
Professor Min Wu, Dean's Representative

Acknowledgments

First and foremost I would like to thank my advisor, Professor David Jacobs, for his guidance, direct advice, positive and constructive feedback, and unassuming brilliance. My successes past and future are due to his encouragement and mentorship.

I am grateful to the members of my thesis committee, Professors Philippe Burlina, Doron Levy, Dianne O'Leary, and Min Wu, for their helpful and insightful feedback, both on my thesis in its final stages, and throughout the years along the way.

My mentors during the seven summers I spent as an industrial research intern during my undergraduate and graduate years provided me with the motivation to get me to and through graduate school. I would like to express my sincerest gratitude to Dr. Thomas Grandine and Dr. Jan Vandenbrande of The Boeing Company, and Dr. Philippe Burlina and Dr. Daniel DeMenthon of the Johns Hopkins University Applied Physics Laboratory for encouraging me with their interesting problems and rational outlooks on the world. You were the role models that inspired me to keep working. And I would like to thank Dr. Dan'l Pierce, formerly of The Boeing Company, for reaching out and handing an entering college freshman math major his business card at a luncheon, thereby connecting me to the world of industrial research and development that I hope to contribute to throughout my career.

I would very much like to thank Professor Alain Trouvé of the Ecole Normale Supérieure de Cachan for inviting me to study with him for six months, resulting

not only in publications but in opening my mind to new ways of thinking. (And for providing me with the opportunity to live in Paris!)

Much credit is also due to many teachers I had along the way, perhaps most importantly my elementary school teachers Lorena Huber, Steve Anderson, Marlene Erickson, Sharon Held, and Shelly Perkins. I would also like to mention my math team coaches Judy Theil, Patricia Leffler, and Ilyse Wagner, my ballet teachers Kathy Milligan and Jennifer Carroll, and all the many music instructors I have had over the years especially my oboe teacher Glen Danielson.

To all my friends from Seattle (the ballet girls and AMOK), Ithaca (the Clarinets, and the whole Big Red Marching Band), Madison (the mathletes and Choi Tae Kwon Do), College Park (the Graduate Student Government, everybody who came to the Computer Vision Student Seminars, and my math friends who had lives outside of math), Paris (the thésards at CMLA), and throughout the world, you have kept me sane and interested in all aspects of life, thank you. To my boyfriend Dave Karpuk, your support has been invaluable. To my academic siblings, Sameer, Carlos, Daozheng, João, Arijit, Abhishek, and Angjoo, you are the siblings I never had, please stay family!

And to my parents, whose guidance and support for 29 years will ground me for the next 70. Thank you for everything.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

This thesis explores object deformation and lighting change in images. We want to be able to meaningfully compare two images of the same object, specifically when the object has deformed and/or the illumination of the scene has changed. We aim to develop measures of similarity between two images where images of the same object are assigned a low matching energy cost, while images of different objects have a higher cost. The overriding goal of all work in this thesis is to compute image comparison costs that can then be used for recognition purposes. The primary application discussed here will be in the domain of face recognition, because faces provide a good and important example of highly structured yet deformable objects with readily available datasets that include large changes in expression and lighting. However, our methods are not designed specifically for faces, and can be applied to any domain with deformations and lighting change.

Our primary motivation is to show that an image manifold can be explicitly defined to elegantly model deformations and illumination changes in images. $M \times N$ images are considered as points on a high-dimensional manifold of images, and we present a metric that gives local structure to the manifold, where costs are based on changes in shape and intensity. Curves on this manifold describe transformations such as deformations and lighting changes to connect nearby images, or larger

identity changes connecting images far apart. The similarity between two images on the manifold can be defined as the length of the geodesic, or shortest path, that connects them.

The primary contributions of this thesis are as follows. We present a novel lighting-insensitive metric based on the effect of lighting in 3D scenes, where the metric is a function of image gradients and the differences of image gradients, inspired by the known result that image gradients across object boundaries are insensitive to variations in lighting. We show that this local metric is meaningful by applying it to compute dense correspondence vector fields between two images in an optical flow-like setting, where the goal is to generate meaningful image matching costs in the presence of deformation and lighting changes rather than perfect object tracking. A new framework for optimizing flow fields is implemented, making use of the Sobolev gradient and a global kernel, leading to increased stability against deformation.

We then demonstrate how to find geodesics between pairs of images on a Riemannian image manifold using our new metric. Instead of calculating a single correspondence vector field between two images, the metric is integrated along the geodesic, which is discretized into a sequence of images varying along the path connecting the two given images on the manifold. The benefit of using geodesics is that a face can deform slowly through several steps, making the algorithm robust to large expression change and lighting variations. The lighting-insensitive metric is converted into the wavelet domain, where it is able to handle moderate amounts of deformation, comparable in size to the support of the wavelet basis functions. We show that in this formulation the geodesics through each wavelet basis location

are independent, and can be calculated analytically so that no optimization scheme is required and there is no risk of converging to local minima. This allows for an extremely fast image comparison computation.

To handle larger deformations in addition to large changes in illumination, we consider an algorithmic framework where a deformation can be described as a diffeomorphism (a smooth invertible function between differentiable manifolds), and the geodesic flow through diffeomorphisms is sought. To calculate the discretized geodesic path, a computational optimization scheme based on the gradient descent method is required. We present preliminary implementations of the diffeomorphic framework, calculating diffeomorphisms for complete face databases, and suggest how this work can be extended for further useful applications.

1.1 Outline of Proposed Methods

We start in Chapter 2 by exploring some of the most straightforward methods for obtaining correspondences between images, to see where they break down. We first consider an image morphing algorithm used in computer graphics applications presented by Beier and Neely [8]. This method is based on matching corresponding line segments in images, and is seen to be not very robust in regions of the image not explicitly matched. We then look at optical flow, specifically the robust optical flow algorithm of Black and Anandan [13]. In the optical flow framework, each point in a first image is matched to some point in a second image, resulting in more generally meaningful set of correspondences than the first simple morphing

algorithm. However, the correspondence field is penalized by a cost function that usually requires too much smoothness to be able to match all individually deformed image patches, or is not smooth enough to be meaningful. This chapter is not meant to be a comprehensive study of all possible known methods that handle deformation, but simply serves to motivate the later chapters where new research is presented.

In order to model variations in expression, establishing point correspondences between faces is essential. Our methods determine dense correspondences between pairs of images, assigning a cost to each point pairing based on a novel image metric. The research presented in this thesis was conceived with the idea that the mathematical structures of geodesic paths and diffeomorphisms on image manifolds should be powerful tools for handling deformations in images. If we consider an image to be a point on a high dimensional image manifold, then following a path away from that image through the manifold is like watching a sequence of images that get progressively more different from the original. This allows deformations to be introduced gradually over the course of several images, where correspondences are well-defined between every pair of adjacent images on the path. A manifold is a generalized often high-dimensional surface, diffeomorphisms are smooth, bijective mappings between images represented as points on a manifold, and geodesics are the locally shortest paths between two points on a manifold; these ideas will be expanded later, or see [28, 93]. As we aim to measure image similarity, we use a Riemannian manifold, in which a local metric gives structure to the manifold by penalizing certain types of image change, in the same way a hill requires more work from a walker in some directions than others. In order for the manifold structure to

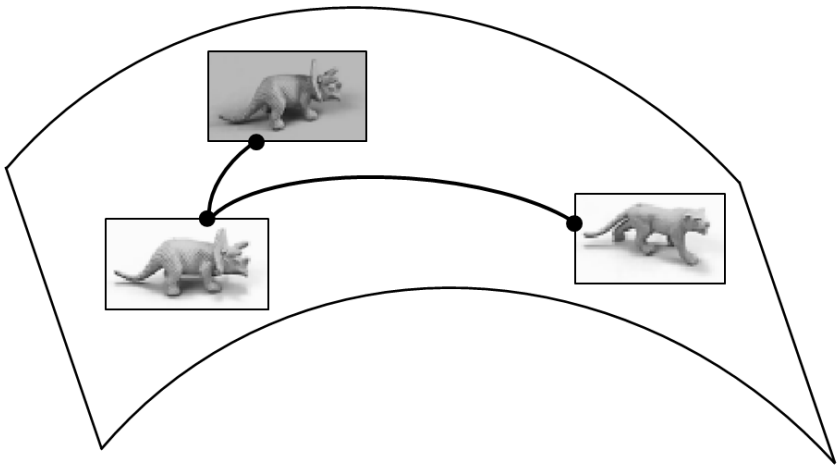


Figure 1.1: A high-dimensional manifold, where each point on the manifold is an $M \times N$-dimensional image, and geodesics connecting more similar images are shorter (images from [50]).

be useful, images of the same object should be close together on the manifold, while images of very different objects should be far apart; see Figure 1.1. The length of the geodesic connecting two images can therefore be used as a measure of image similarity.

In Chapter 3 we present a new metric for measuring image patch similarity in the presence of illumination change. It is well known that using image gradients instead of intensities directly is less sensitive to changes in lighting, for example from [35, 55]. The metric we present has similar properties to the gradient direction, but assigns a higher cost to changes when the image gradient is small than to changes when the image gradient is large, by scaling the gradient of the image change by the norm of the image gradient. We discuss why this can be useful, and in later chapters this metric is used in a geodesic framework. The direction of the gradient

in an image patch can be compared to the direction of the gradient in a different patch, and the resulting difference measure is the angle difference. Using our new metric where the relative magnitudes of the gradient affects the cost to match them, a meaningful path can be traced from the first gradient to the second on a manifold that provides more information than the simple angle difference.

The new local metric is used in an optical flow-like framework in Chapter 3, where every pixel in the first image is matched to some pixel in the second image, resulting in a correspondence vector field. This vector field can be thought of as defining a small movement from one image to the next along a path on a manifold, allowing both shape and intensity to be modified locally. The cost of a given correspondence field is defined by the new illumination metric plus a regularization term, and this cost is minimized using an optimization scheme. The first image can then be warped along this vector field to result in an image that is in correspondence with the second image using only pixel values from the first image; see Figure 1.2. This study verifies that the local metric we present is meaningful.

Chapter 3 also presents a regularization term that was chosen to result in an efficient Sobolev gradient [66]. In order to calculate the optimal correspondence vector field, a minimization scheme must be implemented, and the smooth properties of the Sobolev gradient allow a gradient descent-based scheme to progress further before breaking out at a local minimum. This algorithm computes the x- and y-components of the illumination and regularization costs at each pixel for each image pair, and all this data is fed into a simple Naïve Bayes classification machine learning routine that learns to discriminate between same-person and different-person image

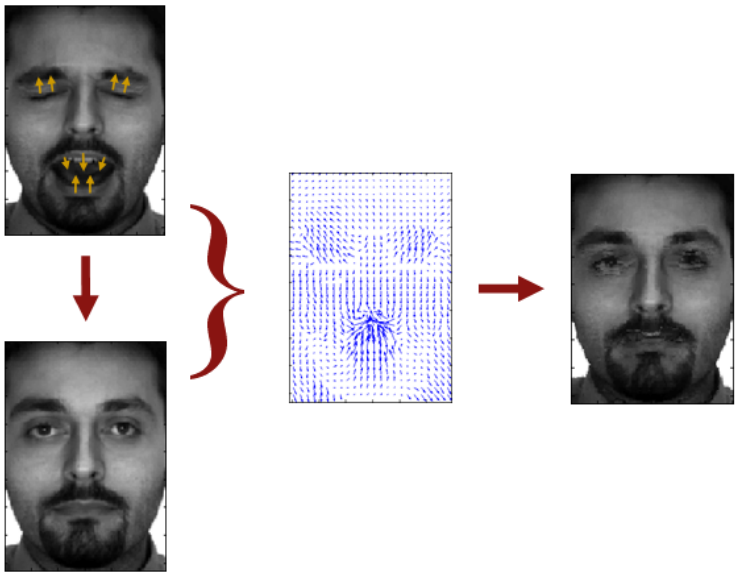


Figure 1.2: Correspondence vectors are found from the top left image to the bottom left image, then the pixels from the top image are warped along these vectors to be put in correspondence with the bottom image.

pairs. Identification results are presented on the AR Face Database [61], where the identity of an unknown image is declared to be that of the known image that results in the lowest image matching cost.

The new lighting-insensitive metric is used in a geodesic framework in Chapter 4. In this work only lighting change is considered (no deformations), and the metric is re-expressed in the wavelet domain where we show that the minimizing geodesic path between any pair of corresponding image gradients can be calculated analytically. In order to compute the matching cost of two images in the presence of lighting changes, each point location is compared separately in wavelet space, where a geodesic path is constructed between the values at that location in each image. The lengths of these curves are summed for the overall image matching cost. A

lookup table is pre-calculated, containing the discretized matching costs for every pair of input values, so that at runtime the computations required to compare two images involve simply projecting each image to its wavelet coefficients and referencing a lookup table, making this an extremely fast algorithm in practice. Because wavelets are fundamentally a multi-scaled representation of an image, the coarsest scales are insensitive to small image changes, so comparing wavelet coefficients provides some insensitivity to moderate deformations. Strong results are seen in the face recognition task where both lighting and expressions are varied. The speed of this algorithm allows many images to be compared very fast.

A method to explicitly handle image deformations using a diffeomorphic framework is described in Chapter 5. This chapter presents an initial exploration of several extensions of the previous work using diffeomorphisms. A sparse set of facial feature points is found automatically on each face image using published methods. Geodesics are then calculated between these known corresponding points, and the geodesics between the rest of the image points are calculated based on spline interpolation, following the work of [20, 83]. The spline is chosen to have certain desirable properties, relating to an appropriate cost function to be minimized and appropriate geometry for the image domain. The result is a geodesic path through diffeomorphisms between two images, along which a first image deforms into correspondence with a second. Directly, the minimization procedure required to go from initial input paths connecting the corresponding points to the true geodesic path through diffeomorphisms is very expensive, but any set of paths defines a smooth, invertible diffeomorphism which is in itself a desirable relation between two images.

The lighting-insensitive metric discussed earlier can be added to this framework, to calculate diffeomorphisms and geodesics that follow the geometric image deformations and handle intensity changes. An image can then be morphed along the diffeomorphic path connecting it to another image, and the resulting images can be meaningfully compared. Preliminary results are again presented on the face recognition task.

Since the entire diffeomorphic path between two images is known, all the intermediate images along this path can be generated, like a movie of one face image deforming into another. If two faces of an individual showing different expressions are provided, then all the images connecting those expressions can be generated, and a new image can be compared to all the intermediate images. When a new image is closer in expression to one of the intermediate images than to either of the known images, it is seen to match more closely to the generated images. This is significant because it is very common in face recognition tasks for an image to be declared more similar to an image of a different person showing a similar expression than to an image of the same person with a different expression. We can now overcome this problem if we are provided with images of every individual only at the expression extremes. Preliminary results are provided on the Cohn-Kanade AU-Coded Facial Expression Database [56]. We foresee many future applications of the methods presented here.

1.2 Related Work

Traditional Face Recognition:

Recognizing faces in the presence of expression variation has been studied for many years. Expression variations are often simply ignored, and faces are compared as if there were no deformations, accepting that the pixels in regions that have been deformed will not match well to any image, so only the parts of the face that have not deformed will provide meaningful comparisons. This is the case with Principal Component Analysis (PCA, also known as eigenfaces) [82], which finds the best low-dimensional linear linear subspace that captures the most important variations in a dataset, and then the coefficients projecting an image into this subset can be meaningfully compared. This is also true for Linear Discriminant Analysis (LDA) [9] which, instead of finding the best subspace representation, finds the best classification, and in the expression case results in learning which parts of the image have high intraclass variability and discounting.

Handling expression variation explicitly requires the knowledge of how individual points from two faces correspond. The Active Appearance Models of Cootes et al. [24] separated shape information from texture information by identifying corresponding feature points in each image (hand-selecting 68 feature points on each face), warping feature points to their average locations while interpolating all other points, and mapping the texture values respectively to achieve "shape-free patches" that could then be compared directly. Identifying this many points by hand is unreasonable for large datasets, so several works deal with uniform grids on face images.

For example Dynamic Link Matching by Lades et al. [49] fit a uniform grids over face images and allowed each node to distort locally, interpolating the distortions for all other points, and representing each grid point by its Gabor wavelet transform, which was met with limited success. Another general method for handling deformations is the Pictorial Structures of Felzenszwalb and Huttenlocher from [31], where cost functions for deformations specific to faces were learned that depend on the local image similarity and the amount of deformation required to arrive at this similarity.

The methods studied in this thesis are all model-based and require no training or learning stages (with one proof-of-concept exception), and so we do not focus on those face recognition algorithms that do involve training as they form a distinct body of work. However, we note that the current state-of-the-art algorithms for face recognition rely heavily on learning methods. We argue that combining robust models, such as those presented in this thesis, with successful learning algorithms will result in advances in the state-of-the-art face recognition techniques. Significant learning-based face recognition methods include the 3D morphable model work of Blanz and Vetter [14], which learned 3D models of faces from textured 3D scans of heads, then modeled new 2D face images fitting parameters for 3D shape and texture, producing impressive results but requiring heavy computation. In [89], Wright et al. projected an unknown image onto the space of known images and enforced sparseness of coefficients, relying on the fact that the most compact representation of a face is likely to be from faces of the same class. Currently, the most robust results on the most general face recognition datasets are obtained by meth-

ods that consider many different representations of faces together, including local and global descriptors. These methods compare faces by comparing the relative responses across all descriptors. For example in [48], Kumar et al. compared many different face patches to known reference patches, resulting in similarity histograms that can then be compared for identity verification. In [92], Yin et al. also divided faces into patches, extracting descriptors such as SIFT and LBP to be compared to reference patches, and then pose change was modeled by using a corresponding patch for each reference patch in a gallery pose for image comparison.

Illumination Insensitivity:

While it has been shown that there can be no truly illumination-invariant image measure in the general case [22], significant work has been done to develop models which are insensitive to illumination change. The insensitivity of the gradient to lighting change has been shown numerous times such as in Lowe's SIFT descriptors [55], where normalized gradients are used as features so that the descriptors are invariant to affine changes in illumination. The self-quotient image model scales image intensity values by smoothed versions of the local intensity at every location, thereby removing the effects of shading to normalize an image, and this idea was successfully applied to lighting-variant face recognition by Wang et al. in [86]. The self-quotient image was combined with a total variation model for better edge preservation by Chen et al. in [23]. Georghiades et al. in [33] presented the illumination cone model, where it was observed that the set of all images of a single object in a fixed pose but varying illumination form a convex cone in the space of images, and this idea was used to reconstruct the shape and albedo of faces from training images

of faces taken under different lighting conditions. Multi-scaled wavelets have been used in the past to efficiently represent lighting variations in works including the lighting model of [67]. Triggs [78] selected image keypoints that were robust against scale, orientation, and illumination change by maximizing an eigenvalue-based local stability criterion that compensates for linear illumination changes.

It was shown by Basri and Jacobs in [5] that all effects of Lambertian lighting on a 3D object can be modeled very accurately in nine dimensions using as basis functions the first nine spherical harmonics. This idea was successfully applied to face recognition by Zhang and Samaras in [95]. Gopalan and Jacobs [35] compare several simple lighting-insensitive representations for illumination-insensitive face recognition, and the gradient direction was found to generally be the most robust of these methods (self-quotient, correlation filters, eigenphases, whitening). A lighting-insensitive wavelet-based face recognition algorithm is presented by Zhang et al. in [96], where using the relation $\log(I) = \log(R) + \log(L)$ between intensity (I), reflectance (R), and illuminance (L), the reflectance term is reduced using thresholding in a multi-scale wavelet domain. In [76], Tan and Triggs apply robust preprocessing and a ternary extension of the Local Binary Pattern (LBP) texture descriptor to lighting-insensitive face recognition.

Wavelets for Deformation Insensitivity:

Wavelets have been used to obtain insensitivity to group actions in the work of Bruna and Mallat [19]. Rubner's Earth Mover's Distance (EMD) [71] is a measure that can handle certain types of deformation, and an efficient approximation of this method was presented by Shirdhonkar and Jacobs in [73] that performs its

calculations in the wavelet domain. EMD has previously been shown to handle moderate deformations in [36], which performs a fast contour matching algorithm to judge similarities between sets of local shape descriptors.

Optical Flow:

Illumination changes and deformations have been studied together in a variety of works. Many attempts to solve this problem have used optical flow to find correspondences between scene points as they deform and in the presence of illumination change; optical flow will be defined explicitly in Section 2.4. The traditional optical flow methods of Lucas and Kanade [3] and the regularized version by Horn and Schunck [41] were updated to include a robust error function to allow multiple motions to be modeled in a single image sequence by Black and Anandan [13]. Negahdaripour [64] relaxed the standard optical flow brightness constancy assumption to allow intensity to change according to multiplication by a scalar and addition by a constant. Kim et al. incorporated this approach into a robust optical flow framework in [46]. Relaxing the brightness constancy constraint to incorporate time-dependent physical causes of lighting change, such as changes in the surface orientation with respect to the direction of light sources, was presented by Haussecker and Fleet in [40]. A gradient constancy constraint for robustness to illumination change was integrated into the coarse-to-fine algorithm of Brox et al. in [17]. Papenberg et al. add higher order derivative constancy terms including a Laplacian constancy term and a Hessian constancy term to the the gradient constancy term in [69]. A structure-texture decomposition was proposed in [87] that treats an image as a composition of geometric structure and fine-scaled texture details, and uses total

variation minimization to minimize illumination artifacts such as shadows. Zimmer et al. handle violations of the brightness constancy assumption in [98] by considering a complementarity between the data and smoothness terms, incorporating photometric invariant channels along with gradient constancy. Brox and Malik use a variational model with rich local descriptors in [18] to accurately handle large displacements better than previous coarse-to-fine methods. SIFT Flow was used in [54] to determine correspondences based on pixelwise SIFT features, and is able to find correspondences in and align images of different but related scenes, unlike traditional optical flow which requires the scenes being matched to be very similar. Face recognition is performed by aligning images to a query using SIFT flow. Glocker et al. use Markov Random Fields in [34] to dynamically solve for an optical flow estimation as a discrete multi-labeling problem with the goal of effectively handling image morphing.

Although optical flow was initially developed for the rigid object motion tracking problem, it has been successfully applied in face recognition. For example in the work of Beymer and Poggio [12], the flow was calculated between a face and a small variation in pose of that same face. The flow between a new face and the original face was calculated to find correspondences, and then the flow field from the original face was applied to a new face to generate a new pose of the new face, which could then be used for comparison. Martinez [60] used the length of the flow vectors to weight the importance of each pixel before performing image differencing on expression variant image pairs. A robust optical flow method was developed by Hseih et al. [42] based on 15 key points that are manually selected on each face to

help drive the flow calculation. In Ma et al. [57], the optical flow of Brox is used to compute flow between gradients, successfully capturing facial deformations and generating synthetic expressions on faces.

Other Methods for Modeling Deformation and Illumination Variation Together:

Other strategies for processing illumination variation and deformations together involve template matching with affine transformations in a Lucas-Kanade-type algorithm. Examples of this include the work by Hager and Belhumeur [37] which handled illumination and small changes in pose together, treating occlusions as statistical outliers, and the work by Tzimiropoulos et al. [84] which iteratively maximized image correlation based on gradients that capture the orientation of image structures rather than pixel intensities.

Solving the expression and lighting problems together in faces has been attempted in several recent works. Zhao and Gao [97] used only pixels from an edge map to determine the best point pair correspondences between images based on location and Gabor jet information. Xie and Lam [90] also found correspondences between edge pixels, developing a cost function based on Euclidean distances, Gabor maps and gradient directions at each pixel. In a separate work [91] Xie and Lam modeled a face as a grid of tiles each of which was allowed to translate, rotate and vary intensity linearly to match a second image. Song et al. [74] combined binary edge features with gray scale information using mutual information. In [43], James presented a method in which a simple local descriptor is calculated at each pixel, descriptors at the same coordinates in two images are compared, and the

number of sufficiently similar descriptor pairs based on a threshold were tallied, resulting in a surprisingly robust cost function. Thesholding was also seen to produce strong results in the work of Gass et al. in [32], which computes smooth warps between expression-variant face images using local features, and handles occlusions by thresholding local distances.

Manifolds and Differential Geometry:

Some of the methods presented in this thesis treat images as points on an image manifold, and aim to find geodesic paths between images. These methods do not fall into the category of manifold learning, being instead in the domain of analytical geometry. In manifold learning algorithms, a large number of data points are given, which are assumed to be sampled from an unknown manifold of much lower intrinsic dimension than the dimension of the data points. Computations are performed with the aim of reducing the dimension of the data while preserving the local and sometimes some global structure between points, such as following a sequence of adjacent points to trace an approximate geodesic. Important Manifold Learning algorithms include Isomap [77], Locally Linear Embedding [70], Laplacian Eigenmaps [11], and Hessian Eigenmaps [30]. In our case, we know the image manifold explicitly, as we define the metric that gives structure to it, and we consider only two images at a time. Our work is more related to that of Absil et al. [1], who work with the Grassman Manifold, which is the space of all fixed-dimension linear subspaces of a given Euclidean space. However, the properties of the image manifold we use are defined by the metric we define to give it structure, and we perform computations using the metric and the known images to calculate geodesic distances in the manifold

itself. We adapt the theory presented in several of the works below to construct a manifold using a new local metric, and demonstrate applications of this framework using datasets larger than the examples presented in most of these works.

Computing geodesic paths through diffeomorphisms for image comparison applications has been explored in several works. Beg et al. [7] defined a framework to solve for large deformation diffeomorphisms, using Euler-Lagrange equations to minimize a cost function based on the a norm of a diffeomorphism through time and the difference between the image morphed by the diffeomorphism and the image to which it was being matched. In the work of Ashburner [2], diffeomorphic image registration was computed by finding the best coefficients over a chosen set of spline basis functions, an optimization problem solved using a Levenberg-Marquardt strategy. Both these algorithms were applied to medical imaging datasets including brain imagery. An evaluation of 14 nonlinear deformation algorithms was presented in [47] with applications to brain imaging.

Diffeomorphisms based on a sparse set of point correspondences using the Thin-Plate Splines deformation framework of Bookstein [15] have been defined. Camion and Younes allowed for inexact correspondence matching in [20] and minimized a data consistency term in addition to the norm on the diffeomorphism itself. Twining et al. [83] enforced exact matching so they only minimized the norm on the diffeomorphism to calculate the geodesic through diffeomorphisms. The former of these algorithms was only applied to a small number of displaced grid points, the latter was applied to a handful of simple images.

Trouvé and Younes studied deformations belonging to Lie groups with Lie al-

gebras acting on Riemannian image manifolds in [79]. The algorithm was applied to a handful of face images undergoing moderate pose changes or deformations. Geodesic shooting was used by Miller et al. in [63] to generate complex deformations. Garcin and Younes used a multiscaled wavelet approach in [94] to perform hierarchical energy minimization to arrive at a geodesic between two images that is more likely to be globally optimal. There is also a body of geodesic methods which do not involve diffeomorphisms, such as the work of Wirth et al. [88] which computed geodesic paths between images that were represented using level sets.

The algorithms presented in this thesis involve finding correspondences between points in two images. These corresponding points will then be compared using metrics insensitive to changes in scene illumination. In this work we do not consider cast shadows, such as those caused by the nose onto the cheek when lighting is from one side, and we will not directly consider pose change, although small amounts of pose change can be effectively handled as deformations. A robust general object recognition system should combine the work of this thesis with an algorithm specifically developed to handle changes in pose, for example [21]. Even stronger results will be obtained when machine learning techniques are applied to the data output by the methods presented, as certain types of deformations provide significant information about the object being deformed. For example, a face can naturally deform into a smile, but if the relative location of a cheekbone changes between two images, these images are not likely to be of the same person, and this information can be captured by simple machine learning algorithms. However, this is not the primary objective of this thesis, and here we study general deformations and lighting

changes. We handle expression and lighting variation within a single framework by constructing deformation and lighting insensitive measures that assign a cost to a pair of images based on their similarity.

Chapter 2

Finding Correspondences to Model Deformations

In this chapter we set up the problems we address in this thesis, and present some common methods that might be used as first attempts at solutions. This chapter does not present new research, it is meant to provide motivation for the research in the following chapters. We explore the situation in which two images of an object are provided, but the object has undergone a nonlinear deformation in one of the images. We would like to be able to quantitatively compare these images, to determine if they are in fact of the same object, and further, we would like to be able to recognize and appropriately handle the images even when the lighting in the scene has changed. We will apply the algorithms developed in this thesis to face recognition, as faces are a good and important example of objects that can undergo moderate deformations and experience extreme lighting changes. However, the primary focus of this work is the study of geometric and image properties.

2.1 Point Correspondences for Image Warping

In order to be able to quantitatively compare two images of an object that has deformed, it is essential to determine how the points in the images correspond. Once point correspondences are determined, the points can be compared to see how similar they are, for example in intensity or in image features that capture

relationships between neighboring pixels, such as the direction of the gradients of the intensity. Without knowing how points correspond, we have no way of comparing specific parts of an image.

Specific image feature point computation is a well-studied field, starting generally with the Harris corner detector [38], the Scale-Invariant Feature Points (SIFT) [55], and the Speeded Up Robust Features (SURF) [6]. Feature point detectors have been developed for many applications, and specifically for faces there are algorithms that can reliably find 4 to 9 points on fairly uncontrolled images of faces. For example, the commercial OMRON algorithm [25] detects the nine facial feature points as seen in Figure 2.1.

This small number of points is sufficient to be able to rigidly align face images, which is done by finding the average location of each feature point across a dataset, then using the Least Squares or RANSAC [39] method to determine the affine transformation that most closely aligns the feature points of each individual image to the average position. Affine transforming each image accordingly puts them in a standard frame that is more meaningful for comparison, and our algorithms assume all face images have been pre-aligned. However, such a small number of feature points only provides enough information for a rigid affine transform, and does not provide enough information to stretch and shrink individual parts of an image to effectively warp all points into correspondence with another image. Several recent algorithms including [10, 27, 53] automatically find much larger numbers of points or curves on each face, and while these algorithms are becoming very accurate, there is still some loss of robustness to variations in pose, lighting, expression and occlusion when

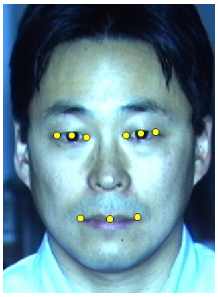
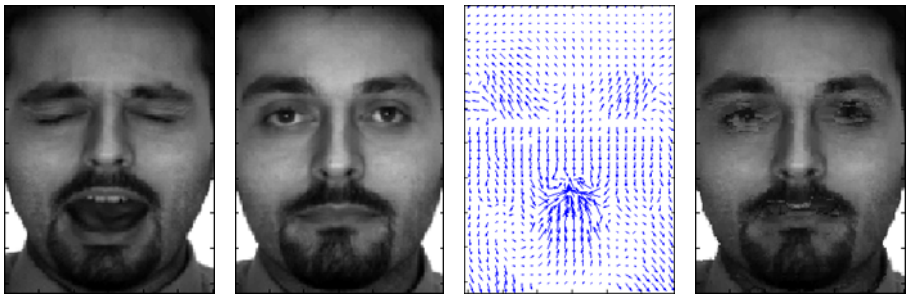


Figure 2.1: Nine face feature points as detected by the Omron algorithm.

searching for so many very specific points.

We aim to deform one image so that it matches a second image as closely as possible, thereby obtaining dense point correspondences between images. The deformation will not be based on directly finding individual point correspondences in the second image for each pixel in the first image, but instead will be based on minimizing a cost function that compares pairings between the two images for all pixels over a single global correspondence field. The deformed image is then compared to the second image, and image pairs with high similarity are assumed likely to be of the same object, person, or scene. As an example, see Figure 2.2.

To test algorithms throughout this thesis, face recognition tasks will be performed on the expression and lighting subset of the AR Face Database [61], a publicly available database created by Aleix Martinez and Robert Benavente in the Computer Vision Center (CVC) at the Universitat Autònoma de Barcelona. This is a widely used dataset and so we will be able to compare to other published methods. This database includes frontal images of individuals on a plain background displaying large variations in expression in addition to variations in lighting, and these



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

Figure 2.2: The pixels from (a) are warped into correspondence with (b) via the correspondence vector field in (c), resulting in the final image (d).

image variations can be compared to the well-lit neutral expression of each person. The different expressions are seen in Figure 2.3(b), and the different lightings are seen in Figure 2.3(c). For our experiments, each non-neutral face is compared to every neutral face in the dataset, and a cost is calculated for each pairing. We will define identity based on nearest neighbor matching, meaning that for a given non-neutral face, if the pairing that returns the lowest matching cost came from the neutral image of the same individual, then we deem the non-neutral image to have been correctly identified. We use the standard crops of the AR Face Database [62], consisting of 50 men and 50 women, and unless otherwise noted we rescale each crop so that it is 83×59 pixels, half the length and width of the original images, as face recognition tasks tend to work the best on images of this scale.

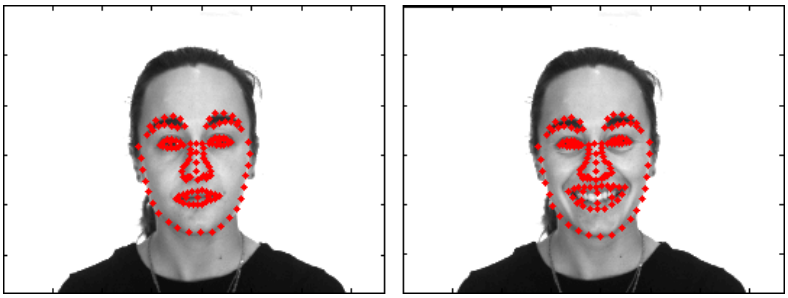
(a) (b) (c)

Figure 2.3: The variations of one person from the standard croppings of the AR Face Database [62]: (a) neutral, (b) expressions, (c) lightings.

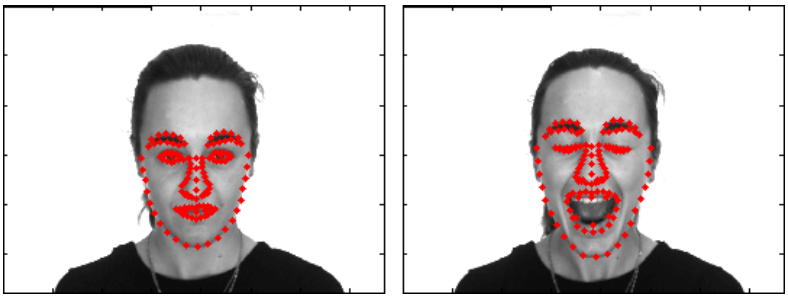
2.2 Current Facial Feature Point Detection

Some of the algorithms we present will require the knowledge of a small set of facial feature points, with feature points more varied than those found in Figure 2.1, and in this section we describe how we obtained appropriate feature points for later use. In [27], 98 facial feature points are manually determined for the expression variant images of the AR Face Database, so we will use these publicly available points. To find the feature points in the lighting variant images, the algorithm of [10] for automatically finding 29 face points was applied to the images, using parameters learned from the datasets as described by the authors. We therefore expect and observe that the facial feature points we use for the lighting variant images are less precisely located than those in the expression variant part, but are for the most part acceptably accurate. The supplied feature points are seen in Figure 2.4.

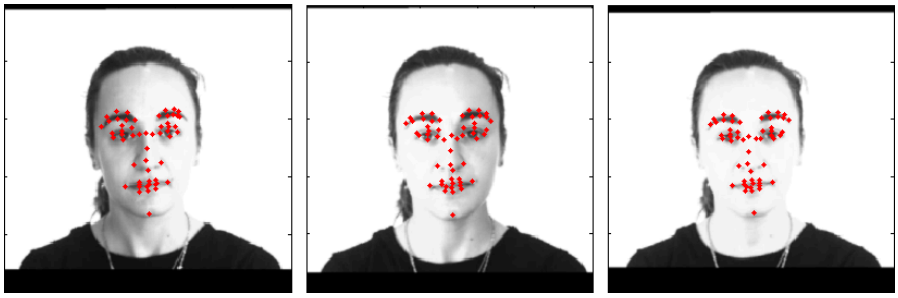
Fourteen semantically meaningful points found by both facial feature point detection algorithms were selected on each face, as shown in Figure 2.5. From these points, the images were aligned using affine transformations so that the locations



(a)



(b)



(c)

Figure 2.4: Feature points found on the AR Face Database. (a)-(b) The feature points found by Ding and Martinez in [27] are used for the expression variation images, (c) the feature points found by Belheumer et al. in [10] are used for the lighting variation images.

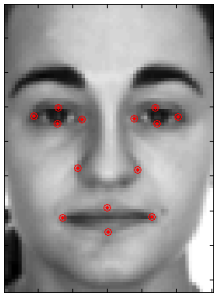


Figure 2.5: The 14 selected feature points used in some of the algorithms in this thesis, on a cropped face.

of the points in each aligned image were as close as possible to the average point locations. Both feature point detection algorithms required the full uncropped face images as input, as they make use of information relating to the shape and position of the head and hair, and so we need to crop the images to just the face regions. To create the standard cropping of the AR Face Database, the individual faces were morphed to a standard position as described in [62]. We use the same subset of individuals, but apply only an affine transformation to align the image, as opposed to any nonlinear image morphing, during preprocessing. The average feature point locations of the standard AR croppings are used for alignment, and the images are cropped to the same regions of the face as the standard croppings as closely as possible given that the amount of preprocessing is different.

2.3 Background: Simple Warping

To demonstrate that finding correspondences between deformed images is a challenging task, we will first look at a straightforward way to attempt to match

images of two objects, using warping methods developed for computer graphics applications to attempt to warp the one image to be in correspondence with the other. The commonly cited algorithm of Beier and Neely [8] is based on corresponding line segments in each image, and warps an image so that its segments are aligned with the location of the corresponding segments in a second image. The appropriate warping for all other image points can then be determined by interpolating the warping of the known lines.

In order to determine corresponding line segments, corresponding points defining the endpoints of the segments must be known, and we will use the facial feature points found from [10] and [27] to obtain these points as described in Section 2.2. From the 14 known feature points, 13 lines were selected for use in the warping algorithm of [8], as shown in Figure 2.6(a).

The warping at each point is defined as a weighted sum of the deformations of every known feature line, where the weightings are determined by the distance to each known line. For an individual pixel, the weight of each line is defined by

$$\text{wgt} = \left(\frac{L^{\,p}}{a + d} \right)^{b}, \tag{2.1}$$

where L is the length of the line, p controls how the length of the line affects the weighting, a large a encourages smoother warpings at the expense of precision, d is the distance from the pixel to the nearest point on the line, and b controls how the distance affects the weighting. For the experiments here, $p = 0$, $a = 1$, and $b = 1$, which disregards the length of the lines, weighs the importance of the lines

linearly with respect to the inverse of their distance away from the point, and only enforces moderate smoothness. The amount a single pixel is required to move can be defined by a deformation vector at each point. The collection of these vectors is a vector field that attempts to match the first image with the second, putting them into correspondence. The results of two image warpings can be seen in Figure 2.6. This method works well for small deformations, but it handles poorly the large changes in facial expression seen here.

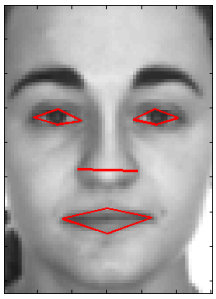
Given two images, using the warping algorithm described above, image I_2 is warped to match image I_1, producing warped image I_2^w. An image similarity measure is used to compare I_2^w to I_1. Image differencing is the simplest image comparison metric, where image intensity differences are calculated at each pixel, and these differences are summed:

$$E_{\text{imdiff}} = \sum_{i,j} \|I_1(i,j) - I_2^w(i,j)\|. \tag{2.2}$$

Both the L_1 and L_2 distances are considered.

It was found that warping from the variant face to the neutral face produced more accurate results when compared to warping the neutral face to the variant face. In the first direction, features that appear in a variant expression, such as an open mouth, can be diminished when warping to neutral, but when starting with the neutral face, there are no pixels corresponding to the inside of the mouth that can be warped to match these pixels in the second image. See Figure 2.7.

Identification results on the expression variation part of the AR Face Database



(a)

(b)

(c)

Figure 2.6: (a) The line segments used for warping. (b)-(c) Two examples using the line-based warping method, warping the image on the left to match the image in the middle, creating the image on the right.

(a)

(b)

Figure 2.7: Faces warped from neutral to variations, opposite the direction from Figure 2.6. Empirically this was found to produce weaker identification results.

Table 2.1: Identification results on the expression variation part of the AR Face Database achieved using the warping algorithm of [8] with image differencing.

	Smile	*Frown*	*Scream*	*Overall*
E_{imdiff} L_1 norm	98.0%	98.0%	83.0%	93.0%
E_{imdiff} L_2 norm	84.0%	95.0%	75.0%	84.7%

achieved using the simple warping algorithm described above are presented in Table 2.1. The results show that simple face warping does aid in expression-insensitive face recognition, but there is still much room for improvement, which we will explore below.

The results obtained in this study are not entirely fair, as the feature point-based implementation could probably be improved with further parameter exploration, and it is possible that the 14 feature points chosen are not the ideal set of face feature points to be used for warping. A better set of points might include a partial outline of the face, to give more meaning to the deformation of the entire face including the hairline. However, we would like to study methods of generating dense correspondences where all the point correspondences are meaningful, and so we elect instead to move on to more meaningful methods, starting with optical flow.

2.4 Dense Point Correspondences from Optical Flow

A collection of algorithms developed for establishing dense point correspondences between images is provided in the optical flow framework [3]. The traditional optical flow algorithm was developed to track points on rigid objects through frames in a video sequence. The amount of movement between images was small, and although sides of an object might rotate out of view over the course of several frames, the rigid object experienced no non-rigid deformations. Since its original inception, optical flow has been extended for many uses, such as tracking non-rigid objects, including expression-variant faces as discussed in the section on related work.

Optical flow determines the displacement of every pixel in an image to the most similar pixel in a second image, returning the displacement vectors as a vector field over the image. The method is completely automatic; the only input required is the two images, which are assumed to be reasonably well aligned; see Figure 2.2. Traditional optical flow is based on the intensity constraint equation, which assumes that corresponding object points in two images will have near equal gray-scale values, so that at point (x, y, t), its intensity $I(x, y, t)$ is assumed to satisfy

$$I(x + \delta x, y + \delta y, t + \delta t) = I(x, y, t). \tag{2.3}$$

A first order Taylor expansion of this equation gives

$$I(x + \delta x, y + \delta y, t + \delta t) = I(x, y, t) + \frac{\partial I}{\partial x}\delta x + \frac{\partial I}{\partial y}\delta y + \frac{\partial I}{\partial t}\delta t, \tag{2.4}$$

and plugging equation (2.3) into equation (2.4) leads to the traditional optical flow

equation:

$$\nabla I \cdot w + I_t = 0, \tag{2.5}$$

where $w = [\delta x \ \delta y]^T = [u \ v]^T$ and $\delta t = 1$. This relation defines an energy to be minimized, which we will refer to as the brightness energy,

$$E_b = \nabla I \cdot w + I_t. \tag{2.6}$$

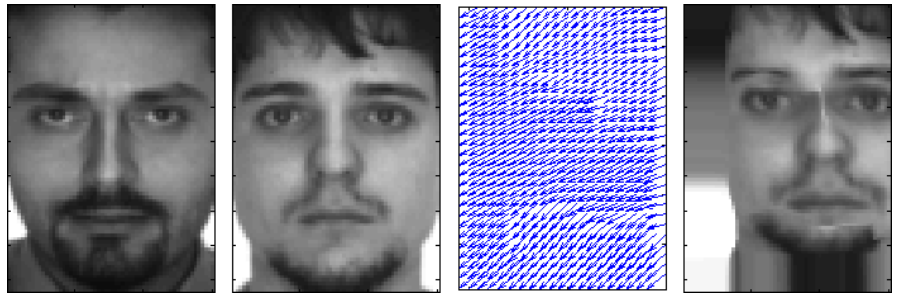
As this equation is under-determined, a second constraint must be added, and it is standard to use this equation to enforce smoothness or regularity by minimizing the gradient, as in [41], referenced as the regularization energy

$$E_r^2 = \left(\frac{\partial u}{\partial x}\right)^2 + \left(\frac{\partial u}{\partial y}\right)^2 + \left(\frac{\partial v}{\partial x}\right)^2 + \left(\frac{\partial v}{\partial y}\right)^2 = |\nabla \delta x|^2 + |\nabla \delta y|^2. \tag{2.7}$$

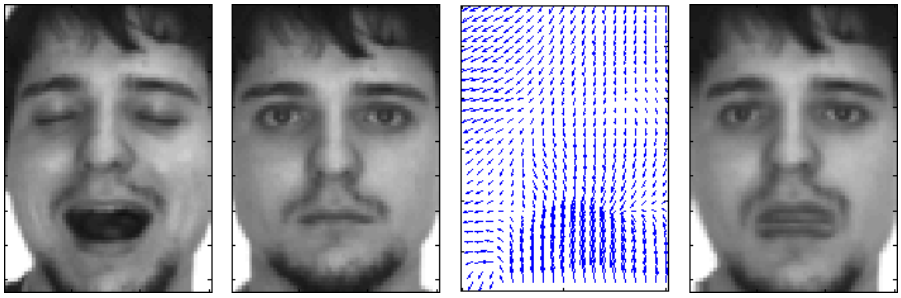
Black and Anandan [13] incorporate a robust error function ρ to limit the effect of outliers, allowing multiple distinct motions to be handled within a single image pair. Instead of solving a least squares fit of all points, the effect of outliers is reduced using an error ρ-function with each energy term, which limits blurring at motion boundaries. The full energy to be minimized is

$$E_{B\&A} = \int_\omega \left(\rho_b(E_b^2) + \lambda \rho_r(E_r^2)\right) dxdy \tag{2.8}$$

for weighting constant λ. This equation is minimized using a coarse-to-fine strategy, solving the problem at a coarse scale, then warping the image to a finer scale, adding



(a)



(b)

Figure 2.8: Two examples using Black and Anandan optical flow to attempt to automatically put two images into correspondence. The correspondence vector field is from the far left image to the next image, and the warped image on the right is the second image warped backwards along this correspondence field to be in correspondence with the first image.

more outliers, and repeating the process. At each level, the solution is obtained numerically using the Successive Over Relaxation (SOR) scheme. This method is seen to handle boundaries much more reliably than the method of [41] alone. The optical flow implementation used here is based largely on the implementation from [75]. The results from applying this robust optical flow algorithm can be seen in Figure 2.8.

Using the Black and Anandan optical flow algorithm, we warp images from the AR Face Database and perform a basic nearest neighbor recognition test using image differencing from equation(2.2) as before. Results are presented in Table 2.2. We see that warping based on dense optical flow out-performs feature point-based warping. This is not surprising, as with optical flow, every pixel is deformed to a meaningful location, whereas using sparse correspondences cannot reliably deform regions of points. We see in the examples above that warping based on sparse points breaks down faster than the robust optical flow, for the algorithm has no way to deal with a mouth being closed, for example, which is equivalent to introducing occlusions. The optical flow algorithm is able to find similar pixels for every point, whereas the point-based warping can do nothing but drag along all the pixels between the very few correspondences that it knows for sure.

Because the direct optical flow results are seen to be strong, we perform further rudimentary tests on this method. This thesis aims to handle both deformations and lighting variations together, and so we test the optical flow method on the lighting variation subset of the AR Face Database. Several optical flow methods have been developed to be insensitive to changes in scene lighting, such as in [46] where lighting change is modeled by multiplication by a scalar and addition by a constant, an idea from [64], and in [17] which incorporates a gradient constancy constraint for robustness against illumination change. We do not explore these methods further as we will take a somewhat different approach to handling lighting change. Here we simply demonstrate that the standard Black and Anandan flow, which does not aim to explicitly handle lighting variation, expectedly performs poorly between images

with different illuminations.

A standard method for comparing images in the presence of lighting change is to compare the direction of the gradients of the images. If $[dx\ dy]^T = \nabla I$ are the elements of the gradient of the intensity values of image I, then the direction of the gradient at every point is

$$\theta = \tan^{-1} \frac{dy}{dx} \mod \pi. \tag{2.9}$$

When there is significant lighting change in a scene, comparing the gradient directions θ is much more meaningful than comparing image intensities directly, because although the gray scale values of the pixels may have changed significantly, the angles of the surfaces in the images and hence the directions of the image gradients (mod π) have remained the same. In order to compare gradient directions at individual points, the difference between their angles is computed, taking the smaller of $\Delta\theta$ and $\pi - \Delta\theta$, so the difference in gradient directions $\|.\|_{\text{GD}}$ is calculated as

$$d_a = \|\theta_1 - \theta_2\| \tag{2.10}$$

$$d_b = \pi - d_a \tag{2.11}$$

$$\|.\|_{\text{GD}} = \min(d_a, d_b). \tag{2.12}$$

We compare the gradient directions of the optical flow output and include these results in Table 2.2, where $E_{\text{GD}} = \sum_{ij} \|\theta_2(i,j) - \theta_1(i,j)\|_{\text{GD}}$. It is seen that using the gradient direction does increase the accuracy of the identification results, but the results are still poor in the lighting case.

Table 2.2: Identification results on the expression and lighting variation subsets of the AR Face Database achieved using Black and Anandan Optical Flow.

Expressions	*Smile*	*Frown*	*Scream*	*Overall*
E_{imdiff} L_1 norm	100%	99%	89%	96%
E_{imdiff} L_2 norm	99%	98%	76%	91%
E_{GD} L_1 norm	100%	100%	91%	97%
E_{GD} L_2 norm	100%	100%	88%	96%
Lightings	*From left*	*From right*	*From both sides*	*Overall*
E_{imdiff} L_1 norm	19%	16%	0%	11.7%
E_{imdiff} L_2 norm	12%	7%	0%	6.3%
E_{GD} L_1 norm	59%	52%	3%	38%
E_{GD} L_2 norm	58%	50%	3%	37%

The results shown in this chapter are not meant to be comprehensive, and these methods could be further tuned to perform better on the tasks presented. Our purpose here was simply to explore some commonly used techniques to observe where they break down. We have seen that having meaningful dense correspondences is essential for handling deformations, but that the most straightforward approaches to finding correspondences do not work sufficiently well. A variety of solutions to these problems have been proposed, some of which will be highlighted in the following chapters. This thesis will focus on exploring and improving methods for finding dense pixel correspondences across deformations, and methods for comparing these corresponding pixels in the presence of lighting variation.

Chapter 3

A Deformation and Lighting Insensitive Metric for Face Recognition Based on Dense Correspondences

The work from this chapter was published in the proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) in June 2011, [44].

3.1 Introduction

Face recognition is a challenging problem, complicated by variations in pose, expression, lighting, and the passage of time. Significant work has been done to solve each of these problems separately. We consider the problems of lighting and expression variation together, proposing a method that accounts for both variabilities within a single model. We construct a deformation and lighting insensitive metric that assigns a cost to a pair of images based on their similarity. In order to model variations in expression, establishing point correspondences between faces is essential. Our method determines a dense correspondence flow field between pairs of faces, assigning a cost to each pixel pairing based on a novel image metric.

There are two main contributions in this chapter: 1) we present a new lighting-insensitive metric based on the effect of lighting in 3D scenes, and 2) we present a new framework for optimizing flow fields making use of the Sobolev gradient and

a global kernel, leading to increased stability against deformation. The algorithm presented here is able to find reliable correspondences between images that are taken under very different conditions, and the cost function based on these correspondences results in very good recognition accuracy across classes of structured images with variations in deformation and lighting.

Our new deformation and lighting insensitive metric is a function of image gradients and the difference of image gradients, inspired by the known result that image gradients are insensitive to variations in lighting. To find the best pixel correspondences between image pairs, we minimize the sum of the proposed photometric matching costs at each pixel, added to a regularization term that enforces smoothness across adjacent pixel correspondences using a global kernel. Our optimization scheme minimizes over the correspondence flow field making use of a Sobolev gradient, which is smoother and results in superior rates of convergence. The optimization returns correspondence costs for each image pair, which can be compared to make decisions on identity. Based on the photometric and regularization costs calculated at each pixel, we learn a Naïve Bayes Maximum Likelihood model of how same-person and different-person image pairs typically correspond, and we apply this knowledge to improve our results. Experiments are presented on the AR Face Database, and our method is seen to be competitive with the current state-of-the-art.

The standard method for finding dense correspondences is to determine the optical flow between images. Methods of optical flow have traditionally been developed to measure rigid object motion between images in a video sequence. We

emphasize that while we construct a method that involves determining a flow field between pairs of images, our goal is to compute a distance between image pairs, and we are not proposing a new method for solving problems in the general optical flow framework. We will sometimes accept incorrect pixel correspondences if this allows the overall image matching cost to be meaningful.

We review the use of optical flow for face recognition in Section 3.2, and present our new metric in Section 3.3. Our optimization scheme is described in Section 3.4, a probabilistic model is introduced in Section 3.5 to improve our results, and experiments are presented in Section 3.6.

3.2 Optical Flow for Face Recognition

Optical flow determines the displacement of every pixel in an image to the most similar pixel in a second image, returning a vector field over the image. It was discussed in detail in Section 2.4, and many applications of optical flow to face recognition were described in Section 1.2. However, there are limits to using traditional optical flow. The flow between faces is highly nonrigid, often with very large object deformations, and does not involve any intermediate frames between two images separated in time. For example see the expression extremes when comparing Figure 2.3(a) with the third image in Figure 2.3(b), or the lighting variations between Figs. 2.3(a) and 2.3(c). The challenge of this flow problem is demonstrated using the robust Black and Anandan flow [13], and similar results were observed when using the long range Brox flow [17], which also incorporates a gradient constancy constraint

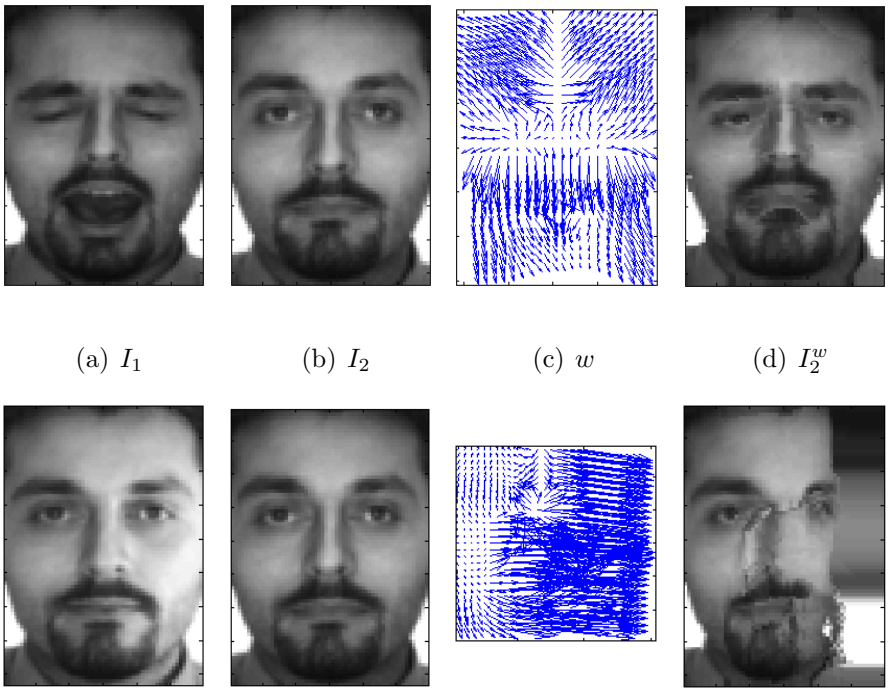
for illumination change robustness. To inspect pixel correspondences, pixels from one image can be traced along the flow and pasted into their corresponding positions to create a warped image. When the weight on the regularization term in (2.8) is very small, it is possible to achieve artificially good-looking results with the Black and Anandan flow, such as in Figure 3.1(d) generated for $\lambda = 10^{-5}$. Pixels from the tongue in I_1 are matched to lip, skin and beard pixels in I_2, creating false correspondences and a very nonsmooth flow. If the regularization weight is turned up then the resulting flow is almost zero everywhere, and no deformations are captured. If lighting changes are introduced, the method completely breaks down; see Figure 3.1(h). We want to construct a new metric that can handle large deformations and is insensitive to lighting changes, to be able to find more accurate costs based on dense correspondences between images.

3.3 A Deformation and Lighting Insensitive Metric

We present a new deformation and lighting insensitive metric, which we will then use in an optical flow-like framework.

3.3.1 The New Metric

Traditional optical flow relies on the intensity constraint equation (2.3) to find correspondences between images. Instead of enforcing consistent intensity, we would like to construct a metric where intensities that change as a result of a lighting change in the scene can still be matched. If $w(\vec{x})$ is the flow from image $I_1(\vec{x})$ to



(a) I_1 (b) I_2 (c) w (d) I_2^w

(e) I_1 (f) I_2 (g) w (h) I_2^w

Figure 3.1: Poor results are achieved when the Black and Anandan flow w is calculated from I_1 to I_2, then the pixels from I_2 are warped backwards along w to generate image I_2^w which corresponds to I_1. The flow here is calculated with a very small regularization weighting. (a)-(d) Change in expression. (e)-(h) Change in lighting.

image $I_2(\vec{x})$, where $\vec{x}_{ij} = (i, j)$ is the pixel in the $(i, j)^{th}$ position, then $I_2(\vec{x})$ can be warped backwards along this flow to match $I_1(\vec{x})$ by defining

$$I_2^w(\vec{x}) = I_2\left(\vec{x} + w(\vec{x})\right). \tag{3.1}$$

Any image warped backwards via w will be denoted with a superscript w. Traditional template matching attempts to minimize the warped image difference

$$E_b^{L^2}(w) = \frac{1}{2}\sum_{i,j} \|I_2^w - I_1\|_{L^2}^2. \tag{3.2}$$

The usual Euclidean metric gives structure to the image manifold in a local neighborhood of I. Letting δI denote an infinitesimal image variation, this infinitesimal metric is $\|\delta I\|_{L^2}$. In the discrete case we take

$$\delta I = I_2^w - I_1, \tag{3.3}$$

so $\|\delta I\|_{L^2}$ is just (3.2). Our new metric instead defines a Riemannian structure on images using the new infinitesimal metric

$$\|\delta I\|_I^2 = \frac{1}{2}\int \frac{\|\nabla \delta I\|^2(x,y)}{\|\nabla I\|^2(x,y) + \epsilon^2} dxdy, \tag{3.4}$$

where ϵ is a small positive constant of the order of the image noise. As a simple approximation of the geodesic distance, we then take our new photometric energy term to be

$$E_b(w) = \frac{1}{2}\sum_{i,j} \frac{\|\nabla(I_2^w - I_1)\|^2}{\|\nabla I_1\|^2 + \epsilon^2}, \tag{3.5}$$

where for the moment the norms and gradients are all taken to follow their standard Euclidean definitions in L^2.

The idea that lighting change on a surface can be represented as multiplication by a scalar and addition by a constant [64] is integrated into the robust optical flow calculation in [46] to develop a lighting-insensitive optical flow algorithm. Our metric goes further, and is designed to be insensitive to intensity changes caused by the

effects of lighting variation in 3D scenes. We normalize by the gradient of the image because a high image gradient often signals a rapid change in scene properties, such as a change in albedo or a point with high curvature. At these locations, a change in lighting conditions can have a significant effect on the image gradient. For example, a brighter light can scale the image gradient. Changing the location of a light can magnify or weaken the gradient at the edge of a polyhedron, as the two sides forming the edge are exposed differently to the light. Therefore, at locations with large image gradients, a significant change in the gradient is often due to lighting effects. At the same time, regions with small image gradients often signal scene regions with uniform albedo and surface normals. For Lambertian objects with uniform albedo and surface normals, variations in lighting cannot induce large gradients. Therefore, while it is not impossible for a lighting change to turn a small gradient into a large one, it is less likely, and so is more heavily penalized by our metric.

The derivation of our new metric removes the restriction that movement between images be less than one pixel, a limitation [4] that comes from applying first order finite differencing to a first order Taylor Expansion (2.4). Many long-range optical flow methods have been developed to get around this restriction, often using hierarchical coarse-to-fine strategies [17]. Our method is able to capture larger movements by optimizing over a dual space related through a global kernel, see Section 3.3.3, and the new method is seen to handle typical face deformations better than traditional optical flow.

In addition to minimizing E_b, a metric based on similarities between the gradients of the intensities, we also want to take into account the total deformation

required to arrive at this similarity, so we include a regularization term E_r that depends on the smoothness of the flow w. Traditional optical flow minimizes the sum of the L_2-norm squared gradients of the flow (2.7). Instead, we introduce a more general Sobolev-type quadratic cost penalizing irregular w,

$$E_r(w) = \frac{1}{2}\langle K^{-1}w,\ w\rangle_{_G}, \tag{3.6}$$

where K is a symmetric positive definite matrix as will be discussed below, and the definition of the G-inner product is given in (3.8).

Equations (3.5) and (3.6) are combined into the proposed Deformation and Lighting Insensitive (DLI) energy function:

$$E_{\text{DLI}}(w) = (1-\lambda)E_b(w) + \lambda E_r(w). \tag{3.7}$$

In our experiments, we take the weighting constant $\lambda = .01$.

3.3.2 The Sobolev Gradient

Since E_b in (3.5) involves derivatives, the usual Euclidean gradient $\nabla E_{\text{DLI}}(w)$ will not be smooth enough to be used in an efficient gradient descent method. Instead we use a Sobolev gradient $\nabla_{_K} E_{\text{DLI}}(w)$, which is smoother and results in superior rates of convergence [66], so the optimization scheme gets caught in fewer local minima, and our algorithm is able to arrive efficiently at more accurate solutions. We first define a general inner product

$$\langle u, v\rangle_{_G} = \sum_{i=1}^{M}\sum_{j=1}^{N}\langle u_{ij}, v_{ij}\rangle_{_{\mathbb{R}^2}}. \tag{3.8}$$

where $G := \mathbb{R}^{M\times N\times 2}$, the dimension of the flow w. Then taking the Sobolev inner product

$$\langle u,\ v\rangle_{K} = \langle K^{-1}u,\ v\rangle_{G} \tag{3.9}$$

used in the regularization term (3.6), the relation between the regular gradient and the Sobolev gradient is given by

$$\nabla_{K} f = K\nabla f, \tag{3.10}$$

where K is a smoothing operator regularizing the Euclidean gradient. To derive (3.10), it is sufficient to consider the variation δf of any smooth function f and follow the framework of differential forms. The definition of the gradient of a function f for any inner product defined by some K is the unique vector written $\nabla_{K} f$ satisfying the following equality for any vector w:

$$\delta f = \langle \nabla_{K} f,\ \delta w\rangle_{K}. \tag{3.11}$$

This can be connected back to the traditional definition of the gradient by observing that for a function f that depends on an N-dimensional vector $\vec{v}$,

$$df(\vec{v}) = \sum_{i=1}^{N}\left(\frac{\partial f(\vec{v})}{\partial v_i}\cdot \delta v_i\right) = \langle \nabla_{\mathbb{R}^N} f(\vec{v}),\ \delta\vec{v}\rangle_{\mathbb{R}^N}. \tag{3.12}$$

From this,

$$\delta f = \langle \nabla f(w),\ \delta w\rangle_{G} \tag{3.13}$$

$$= \langle \nabla_{K} f(w),\ \delta w\rangle_{K} \tag{3.14}$$

$$= \langle K^{-1}\nabla_{K} f(w),\ \delta w\rangle_{G}, \tag{3.15}$$

and equating the first terms of the $\langle ., . \rangle_G$ expressions, we get

$$\nabla f(w) = K^{-1} \nabla_K f(w), \tag{3.16}$$

which is equivalent to (3.10). Since $\nabla E_r(w) = K^{-1} w$ directly from (3.6), we get that

$$\nabla_K E_r(w) = w, \tag{3.17}$$

where K^{-1} no longer appears, and only K is needed for the computation of $\nabla_K E_b = K \nabla E_b$. Here w can be considered as an element of a Reproducing Kernel Hilbert Space (RKHS).

We choose K to be the matrix form of a 2D convolution with a symmetric positive definite kernel k,

$$Ku \equiv k * u, \tag{3.18}$$

where we abuse notation slightly to consider u as an $MN \times 1$ column vector on the left and as an $M \times N$ image on the right. Here k is an $M \times N$ kernel, and K is the $MN \times MN$ matrix representation of this kernel. Multiplying K by the vector representation of u, (3.18) holds for corresponding elements. With this choice of K and periodic boundary conditions, any matrix-vector product involving K can be computed very efficiently with the Fast Fourier Transform (FFT). We therefore accept periodic boundary conditions, as will be discussed further at the end of Section 3.4.2.

3.3.3 Choice of Kernel

The convolution kernel k associated with the matrix K used in (3.6) must be positive definite in order to define an inner product. We select a Gaussian-like kernel for its smoothing properties. The most obvious choice of such a kernel is defined for all (x, y) as

$$k(x, y) = \exp\left(\frac{-1}{s^2}\left(x^2 + y^2\right)\right). \tag{3.19}$$

We will use derivatives of this kernel to define the derivative filters discussed in Section (3.4.2). The scale parameter used is $s = 0.0075p$ where p is the perimeter of the image, this value having been empirically determined to be robust.

When defining (3.6) we instead use a Cauchy kernel which was observed to provide better results experimentally,

$$k(x, y) = \frac{1}{1 + \frac{1}{s^2}\left(x^2 + y^2\right)}, \tag{3.20}$$

where the scale parameter $s = \frac{1}{32}p$.

A second kernel is defined for each s with $s_2 = \frac{s}{4}$, and the final kernel is the weighted average of these two kernels ($\frac{1}{4}$ the kernel with smaller scale, $\frac{3}{4}$ the larger). All parameters and kernel choices were tuned on simple synthetic datasets consisting of grey polygons on a white background, to be as general as possible. At the start of the iterations, the kernel of larger scale dominates, aligning large regions in the image. As the iterations progress, smaller features become more significant and the effect of the smaller kernel predominates.

The kernel has the same dimensions as the image. Convolving with such a global kernel allows our algorithm to capture large-scale image deformations, includ-

ing long-range translations and large rescalings, that other flow algorithms require multiscale methods to achieve.

3.4 The Optimization Scheme

The optimization is performed using a modified gradient descent algorithm. To find a point where the energy function $E(w)$ is minimized, we start with $w = 0$, and at every iteration calculate $\nabla_K E$, then update w using a standard gradient descent update

$$w_{n+1} = w_n - \Delta t \cdot \nabla_K E(w_n). \tag{3.21}$$

In fact, the actual implementation uses a dual variable α_n such that $w_n = K\alpha_n$ initialized at $\alpha_0 = 0$. Using the fact that $\nabla_K E = K\nabla E$, the update becomes

$$w_n = K\alpha_n \tag{3.22}$$

$$\alpha_{n+1} = \alpha_n - \Delta t \cdot \nabla E(w_n), \tag{3.23}$$

which involves only the usual Euclidean gradient. The step size Δt is initially defined to be 0.01. If an iteration results in a cost smaller than the previous cost, we accept the new α_{n+1} and update $\Delta t = 1.1 \cdot \Delta t$. If an iteration results in a larger cost, then the iteration was not successful, and we update $\Delta t = \frac{1}{2} \cdot \Delta t$ and try again. For the next calculation, we use the α_{n+1} which had resulted in too high a cost, as it was found that this helps move away from local minima as in a rudimentary deterministic annealing algorithm, but no α_{n+1} is accepted as a solution if the cost it produces is not smaller than that at the previous accepted step.

The optimization scheme is terminated when either the gradient at the current α is within a small threshold of zero, or when the size of Δt has been decreased to within a small threshold of zero and no nearby α in the direction of the negative gradient has resulted in a smaller overall cost. Like all implementations of the Gradient Descent algorithm, our algorithm will usually stop at a local minimum, but it was observed that optimizing over α using Sobolev gradients allows the optimization scheme to proceed much further before terminating.

3.4.1 The Gradient of the DLI Metric

In order to use a gradient descent method, we must calculate the gradient of the DLI energy function (3.7),

$$\nabla E_{\text{DLI}}(w) = (1-\lambda)\nabla E_b(w) + \lambda \nabla E_r(w). \tag{3.24}$$

Since $\nabla E_r(w) = K^{-1}w = \alpha$ we get

$$\nabla E_r(w) = \alpha, \tag{3.25}$$

and all that remains is to solve for $\nabla E_b(w)$.

3.4.2 The Gradient of the Photometric Norm

For any given definition of the photometric norm E_b, the regular Euclidean gradient can be calculated directly through applications of the chain rule and finite differencing. However, since this cost involves the computation of derivatives of warped images, we will consider a slightly more general situation using low-pass filtered directional derivatives.

Before describing this more general framework, we consider the simple example of the template matching definition of E_b defined in (3.2). For this, the gradient would be calculated as

$$\nabla E_b(w) = (I_2^w - I_1)(\nabla I_2)^w, \tag{3.26}$$

with the warped image gradient term $(\nabla I_2)^w$ resulting from an application of the chain rule. Using the more complex metric for E_b from (3.5), the gradient could be derived similarly.

Instead, to increase robustness, we make use of more general gradient-like filters with larger regions of support than those used by traditional finite difference methods. Instead of calculating a true gradient ∇I we will instead calculate HI for $H = [H_x \; H_y]^T$, where H_x and H_y represent convolutions with more general x- and y-directional derivative filters h_x and h_y of the low-pass kernel k from (3.19).

We introduce a diagonal weighting matrix C on $\mathbb{R}^{MN\times MN}$ with dimensions as in (3.18) to serve as the denominator, with diagonal coefficient

$$C_{ij,ij} = \left(|(H_x I_1)_{ij}|^2 + |(H_y I_1)_{ij}|^2 + \epsilon^2\right)^{-1}. \tag{3.27}$$

The metric (3.5) can now be expressed as

$$E_b(w) = \frac{1}{2}\langle CH_x(I_2^w - I_1),\; H_x(I_2^w - I_1)\rangle_{\mathbb{R}^{M\times N}} \tag{3.28}$$

$$+ \frac{1}{2}\langle CH_y(I_2^w - I_1),\; H_y(I_2^w - I_1)\rangle_{\mathbb{R}^{M\times N}} \tag{3.29}$$

$$= \frac{1}{2}\langle \Delta_C(I_2^w - I_1),\; (I_2^w - I_1)\rangle_{\mathbb{R}^{M\times N}} \tag{3.30}$$

where $\Delta_C = H_x^T C H_x + H_y^T C H_y$ is a discrete Laplacian operator combining the directional derivatives and the weighting factors. Note that the multiplication by C

has a linear cost with respect to the number of pixels, and the multiplication by H_x^T (respectively H_y^T) is a convolution with the adjoint filter of h_x (respectively h_y).

To calculate the gradient of E_b we will make use of the symmetry of the matrix Δ_C to get

$$\frac{\partial E_b}{\partial w_{ij}}(w) = [\Delta_C(I_2^w - I_1)]_{ij} \nabla I_2(\vec{x}_{ij} + w_{ij}) \tag{3.31}$$

or equivalently

$$\nabla E_b(w) = [\Delta_C(I_2^w - I_1)](\nabla I_2)^w. \tag{3.32}$$

To perform the computations efficiently, FFTs are used to compute the convolutions. This means that we accept periodic boundary conditions, despite not having periodic images. In order to avoid driving the optimization by pixels near the boundaries, which are the least important points for our purposes, we multiply the cost function by a weighting function that diminishes the weights of the pixels closest to each boundary smoothly down to zero, thereby approximating periodic boundary conditions. This is implemented by premultiplying (3.27) with this weighting at each point.

3.4.3 The Algorithm

The optimal pixel correspondences between images I_1 and I_2 are determined by the flow w from I_1 to I_2 that minimizes the cost $E_{\text{DLI}}(w)$ from (3.7). The optimization algorithm is summarized in Algorithm 1.

The optimization takes approximately 1 second to converge for a pair of images of dimension 83×59, running Matlab on a 3.16 GHz processor.

Algorithm 1 Find Optimal Correspondences

Input images I_1 and I_2, initialize $\alpha_0 = 0$

repeat

$w_n = k * \alpha_n$

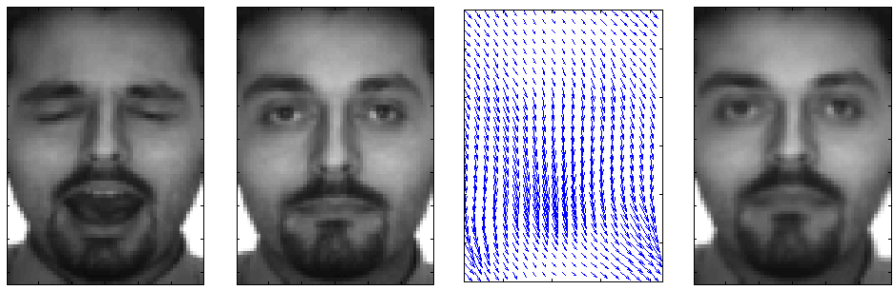
Calculate $\nabla E(w_n)$ from (3.24) using (3.25) and (3.32)

$\alpha_{n+1} = \alpha_n - \Delta t \cdot \nabla E(w_n)$, update Δt

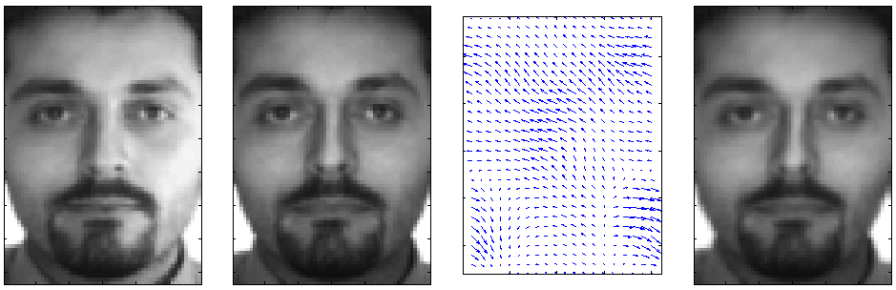
until $\|\alpha_{n+1} - \alpha_n\| <$ threshold

return final matching cost from (3.7)

Inspecting representative image pairs reveals that our algorithm is robust to changes in expression and lighting. In Figure 3.2, the flow w is calculated from I_1 to I_2, then the pixels from I_2 are warped backwards along w to generate I_2^w which corresponds to I_1. We see that in 3.2(d), the top lip and nose from I_2 has been matched very accurately to the location of the top lip in I_1, and the top of the face has been deformed slightly to align with I_1. Below the top lip, the regularization became more important than pixel intensity matching so the rest of the mouth remained smooth, rather than having the discontinuous flow that would be required to match both closed sets of lips in I_2 to the open lips in I_1. We note that generating flows and warped images is not the goal of our algorithm. We are searching for distance values between image pairs, and we accept some imperfect correspondences when this preserves smoothness. It will be seen in Section 3.6 that the smooth correspondences we achieve from our calculations are sufficient to serve as the basis for an accurate identification algorithm. In Figure 3.2(h), the algorithm has accurately detected that although there has been a change in lighting in the



(a) I_1 (b) I_2 (c) w (d) I_2^w



(e) I_1 (f) I_2 (g) w (h) I_2^w

Figure 3.2: Results from our proposed flow calculation. (a)-(d) The algorithm is robust to large deformations, where the top lip has been correctly matched between images while keeping the overall flow smooth. (e)-(h) The algorithm correctly identifies that in spite of significant change in lighting there has been no deformation, and the flow is small.

scene, there is no deformation of the face, and the calculated flow is small, mostly accounting for imperfect alignment between images.

3.5 Learning Typical Correspondence Patterns

Because all images are known to be of faces, typical correspondences between faces can be learned via Naïve Bayes classification to improve the recognition re-

sults. Based on the cost values obtained from the DLI metric, we learn a Gaussian model at each pixel between faces of the same person across variations in expression and lighting, and we learn a separate model for correspondences between faces of different people, also allowing for variations in expression and lighting. The found correspondence costs between an unknown probe face and a known gallery face can then be compared to each model.

After the correspondences between images have been calculated, at each pixel we have a photometric cost in the x- and y-directions, and a regularization cost in the x- and y-directions (recall that the gradient and the flow w both have x- and y-components),

$$\begin{aligned} E_b(w) &= \frac{1}{2}\sum_{i,j}\frac{\|\nabla(I_2^w - I_1)\|^2_{\mathbb{R}^2}}{\|\nabla I_1\|^2_{\mathbb{R}^2} + \epsilon^2} \\ &= \frac{1}{2}\sum_{i,j}\left(E^x_{b_{ij}}\right)^2 + \left(E^y_{b_{ij}}\right)^2 \end{aligned} \tag{3.33}$$

$$\begin{aligned} E_r(w) &= \frac{1}{2}\langle K^{-1}w,\ w\rangle_G = \frac{1}{2}(K^{-1}w)^T w = \frac{1}{2}\left(K^{-\frac{1}{2}}w\right)^2 \\ &= \frac{1}{2}\sum_{i,j}\left(E^x_{r_{ij}}\right)^2 + \left(E^y_{r_{ij}}\right)^2 \end{aligned} \tag{3.34}$$

The cost vector for an image pair correspondence at each pixel (i,j) is $\vec{E_{ij}} = [E^x_{b_{ij}}\ E^y_{b_{ij}}\ E^x_{r_{ij}}\ E^y_{r_{ij}}]$, and the total cost (3.7) at each pixel can be rewritten as

$$E_{\text{DLI}}(w)_{ij} = (E^x_{b_{ij}})^2 + (E^y_{b_{ij}})^2 + (E^x_{r_{ij}})^2 + (E^y_{r_{ij}})^2. \tag{3.35}$$

We can use Maximum Likelihood estimation to learn the typical Gaussian distribution for the flow costs between same person image pairs at each pixel. Given

training data of many same person image pairs, we calculate the optimal pixel correspondences between each pair using Algorithm 1. For each pixel, a Gaussian is fit through the 4D cost vectors found for that location. The probability that two new images both come from the same person can then be calculated at each pixel.

Assuming pixel independence, we multiply the probabilities over all pixels in an image for the final probability value. We compute the probability P_{same} that two images are from the same person, and probability P_{diff} that two images are not from the same person, repeating the above process using training data from different person image pairs. The ratio $P_{\text{same}}/P_{\text{diff}}$ is used as the final similarity metric between pairs of face images, as this is a more discriminatory metric than P_{same} alone. In practice we calculate the log likelihood ratio. For a new image pair I_1 and I_2, a new set of cost values $\vec{E}^{new}$ is calculated where each pixel location is as in (3.35). The final similarity value for this image pairing is then

$$S(I_1, I_2) = \frac{P_{\text{same}}(\vec{E}^{new}(w))}{P_{\text{diff}}(\vec{E}^{new}(w))}. \tag{3.36}$$

We write this similarity function in terms of the image pair, while in the original DLI energy function (3.7) the cost was written in terms of the flow between the two images.

3.6 Experiments

Experiments are performed on the subset of the AR Face Database [62] dealing with expression and lighting; see Figure 2.3. There are seven images of each individual: a neutral face, three variations in expression (smile, frown, scream), and

three variations in lighting (from the left, from the right, from both sides). The standard 100 person aligned and cropped faces are used, consisting of 50 males and 50 females, several of whom are wearing glasses or have facial hair. We resize each image to be 83×59 pixels, as images of this size return the most accurate results with our algorithm. Similarly resized images have been used successfully in many other algorithms [43, 90, 91]. Our algorithm is fully automatic, so no other input is required.

The neutral faces of all individual are taken to be the gallery, and the other six images of each person are compared to each gallery image separately. We found that warping the neutral images to the non-neutral images is more stable, and so the gallery images take the place of I_2 in our algorithm, and the neutral faces are warped backwards along the calculated flows to generate the I_2^w. Nearest neighbor matching is applied, so that the neutral image that results in the lowest correspondence cost for an unknown non-neutral image defines the identity of the unknown image.

Results are presented from the direct output of the optimization scheme minimizing (3.7) in the first row of Table 3.1. To use the probabilistic model from Section 3.5 to maximize (3.36), half the dataset is used as training data, where the same number of different person image pairs are used as available same person image pairs ($6 \times 50 = 300$), with different person image pairs chosen randomly, given that each type of variation is equally represented. The other half of the data is used for testing. The dataset is divided in half randomly five times, and the average accuracy of the five trials is presented in the second row of Table 3.1. The same testing galleries are used for both the direct and learned methods. The results of

Cost Function	*Expression*	*Lighting*	*Overall*
Direct	82.0%	96.0%	89.0%
After Learning	89.6%	98.9%	94.3%
Smile gallery Direct	77.7%	84.8%	81.3%
Smile gallery After Learning	86.8%	91.2%	89.0%
Borders removed Direct	82.0%	96.0%	89.0%
Borders removed After Learning	85.1%	96.4%	90.7%

Table 3.1: Identification Accuracy found when directly minimizing equation (3.7), and after applying the probabilistic model from equation (3.36). Rows 1-2: for a gallery of neutral faces. Row 3-4: for a gallery of smile faces. Row 5-6: when 10% of the border pixels have been removed from each edge for a gallery of neutral faces.

our algorithm are broken down for each expression and lighting variation in Table 3.2. The lowest observed accuracy is on the challenging "scream" case, where our results are 30% higher than recently reported results [74, 97].

We note that after minimizing the cost function without applying the probabilistic model, recognition accuracy across expression decreases as the image size decreases, while accuracy across lighting increases as the image size decreases; see Table 3.3. This effect is observed less strongly after probabilistic learning has been applied; see Table 3.4. We choose to perform all further tests on images of dimension 83×59 pixels.

Variation	*Accuracy*	*Variation*	*Accuracy*
Smile	97.6%	Left light	98.8%
Frown	91.6%	Right light	99.6%
Scream	79.6%	Both lights	98.4%

Table 3.2: Identification Accuracy broken down by variation for a gallery of neutral faces.

Image Size	*Expression*	*Lighting*	*Overall*
165×119	89.3%	91.3%	90.3%
117×85	85.5%	94.0%	89.7%
83×59	82.0%	96.0%	89.0%
59×43	80.0%	97.3%	88.7%

Table 3.3: Identification Accuracy found when directly minimizing equation (3.7) for a gallery of neutral faces.

To test the gains in robustness coming from our new lighting-insensitive photometric energy norm (3.5), we ran our optimization scheme replacing E_b in (3.7) with the L^2 warped image difference metric from (3.2). Results are presented in the first row of Table 3.5. It is seen that this direct image differencing breaks down when lighting variation is considered, and the new metric presented in this chapter is more accurate in all cases.

To test that our algorithm is robust when both lighting and expression are

Image Size	*Expression*	*Lighting*	*Overall*
165×119	89.3%	92.7%	91.0%
117×85	90.3%	97.6%	93.9%
83×59	89.6%	98.9%	94.3%
59×43	80.0%	97.3%	88.7%

Table 3.4: Identification Accuracy found after applying the probabilistic model from equation (3.36) for a gallery of neutral faces.

varied at once, we use the smile faces as our gallery, and repeat the above experiment, so that all the lighting variation images are being warped from a neutral face with harsh lighting to a smiling face with ambient lighting. See Table 3.1.

The recognition accuracy of many algorithms is directly related to the alignment of the outline of the head and neck. To test that we are capturing true face information and not simply capturing the head and neck outlines, we remove 10% of the pixels on each edge of the image after the flow has been calculated, and determine the matching cost only from the remaining pixels; see Figure 3.3. From Table 3.1 we see that very little accuracy is lost. As a comparison, we consider the simple Gradient Direction method, which has been found to be one of the most robust methods against changes in lighting [35]. This method determines the direction of the image gradient at each pixel, and measures the distance between images as the sum of the angles between their gradient directions at each pixel coordinate. The Gradient Direction accuracy decreases by 7% in this case.

Method	*Expression*	*Lighting*	*Overall*
Proposed Framework with image differencing	84.0%	8.7%	46.3%
Significant Jet Point [97]	80.8%	91.7%	86.3%
Binary Edge Feature and MI [74]	78.5%	97.0%	87.8%
Gradient Direction [35]	86.0%	96.0%	91.0%
Elastic Shape-Texture Matching [90]	98.3%*	97.2%	97.8%*
Elastic Local Reconstruction [91]	99.2%*	98.6%	98.9%*
Proposed Method	**89.6%**	**98.9%**	**94.3%**
Pixel Level Decisions [43]	99.0%	97.0%	98.0%

Table 3.5: Comparison with other methods that address both lighting and expression variation on the AR Face Database using a gallery of neutral expression and lighting. *The challenging "scream" case is not included in these expression tests, so these results are not directly comparable.

When compared to other methods in the literature, the method proposed here is found to be very competitive; see Table 3.5. The AR Face Database is a tightly controlled and therefore relatively simple dataset. With a robust error function incorporated into our algorithm to limit the effect of outliers, we expect that our algorithm will be able to handle much less controlled datasets. Unlike other algorithms [43], our method does not rely heavily on input image alignment, as we

Figure 3.3: 10% of the border pixels have been removed from each edge to test that the cost function is capturing face information and not just head alignment.

calculate dense correspondences based on global considerations. We foresee many ways to extend the unified framework presented in this paper to incorporate more robustness, to be able to handle greater variations that cause other algorithms to fail. Nothing in our algorithm is specific to faces, the method can be applied to any class of images with deformations and lighting variation that exhibit a standard structure.

3.7 Conclusion

Finding reliable image metrics is a fundamental problem in Computer Vision. We have presented an algorithm to perform recognition tasks in the presence of deformation and lighting variations in well-structured images. Our primary contributions are the introduction of a metric that handles lighting variation in a new way, and a method to optimize over this metric. The new lighting-insensitive metric is based on the effect of lighting in 3D scenes. The optimization scheme makes use of smooth Sobolev gradients to efficiently optimize over a flow field that determines dense correspondences between potentially deformed images taken under very dif-

ferent conditions. The mathematics inspiring this work is rigorously motivated. We have validated the efficacy of our metric and optimization scheme by applying them to the problem of expression and lighting variant face recognition. Typical correspondence cost patterns from our metric were learned between face image pairs and a Naïve Bayes classifier was applied to improve recognition accuracy. Our very general algorithm is seen to be competitive with the current state-of-the-art on the AR Face Database, and it lays the groundwork for many possible extensions to handle significantly more challenging datasets.

3.8 Acknowledgments

We would like to thank D. Sun and S. Roth for making their implementation of the Black and Anandan Optical Flow [13] available. We would also like to thank A. James for making the Pixel Level Decisions [43] code available.

Chapter 4

A Fast Illumination and Deformation Insensitive Image Comparison Algorithm Using Wavelet-Based Geodesics

The work from this chapter will be published in the proceedings of the European Conference on Computer Vision (ECCV) in October 2012, [45].

4.1 Introduction

In this chapter we present a fast image comparison algorithm for handling variations in illumination and moderate amounts of deformation using an efficient geodesic framework. As the geodesic is the shortest path between two images on a manifold, it is a natural choice to use the length of the geodesic to determine the image similarity. Distances on the manifold are defined by a metric that is insensitive to changes in scene lighting. This metric is described in the wavelet domain where it is able to handle moderate amounts of deformation, and allows us to derive an algorithm where the complete analytic cost calculation requires only $O(n)$ table lookups, for n the number of pixels in one image, less than 3ms per image comparison. We demonstrate the similarity between our method and the illumination insensitivity achieved by the Gradient Direction. Strong results are presented on the AR Face Database.

Considering images as points on a high dimensional image manifold, defining a

metric to give local structure to the manifold allows paths to be calculated between images along the manifold; see Fig. 1.1. Computer Vision literature frequently uses geodesics in a Manifold Learning framework, where many given images are assumed to lie on a manifold and paths are defined by edges through sets of known images. In this work, we are given only two images, and we consider the geometry of the manifold, as induced by the chosen metric, to calculate the length of the path between them. It is natural to use the length of the geodesic, or locally shortest path, to define the similarity between two images, and geodesics provide significant information about the ways in which images differ. Points along a geodesic curve are images that have morphed part way from the first image to the second, and changes such as lighting and deformations can be introduced gradually through time. Being able to construct and manipulate geodesics has many applications, including accurate image interpolation [80], the ability to extract nonlinear statistics from a set of images on a manifold [81], and image registration [7]. In this work we aim to measure geodesic lengths on an image manifold, and provide a framework that can then be extended for further applications.

Due to their high dimensionality, calculating geodesic distances can be a very expensive task directly, but we show that by working in the wavelet domain with a well-chosen metric, we can solve this problem very efficiently. To define an appropriate manifold of images, we will use a metric that is insensitive to changes in lighting and moderate amounts of deformation. The metric depends on image gradients, as gradients are less sensitive to changes in lighting than are direct pixel intensities. We will achieve results similar to those from the illumination-insensitive Gradient

Direction, but here we also have a meaningful geodesic in addition to a simple difference value. We will show that our lighting cost is insensitive to moderate amounts of deformation when accumulated over several scales.

The primary contributions of this chapter are threefold: 1) a method using geodesics to calculate an illumination-insensitive image comparison cost similar to the Gradient Direction, but useful for applications where manipulating geodesics is required; 2) the insight that local dependencies can be removed by using an appropriate wavelet domain to express an image matching cost function based on gradient terms, allowing the cost computation to be separated into independent problems at every point location in wavelet space; and 3) a very fast calculation of this image comparison cost.

4.2 Geodesics for Object Identification

Identifying objects in pairs of images is made challenging by changes in pose, lighting, deformations, and occlusions. If these changes could be introduced gradually over the course of several images, they would be much easier to handle. If we consider the manifold of images of a single class of object, where every point on the manifold is some instance of that object, then paths through the manifold connecting two images would consist of a continuum of images morphing from the first image to the second, like a video playing over time. The similarity of two instances of an object could then be defined by the length of the geodesic connecting them on the manifold, where shorter paths imply more similar objects; see Fig. 1.1.

Being able to compute and manipulate geodesics has many benefits. We present a framework for calculating the geodesic distance on a specific image manifold that we define, and our method is easily integrable to a wide variety of applications.

Given a manifold of $M \times N$-dimensional images, we define a metric on this manifold so that it has a quantifiable structure, making it a Riemannian manifold [29]. The metric defines how costly it is to take an infinitesimal step in any given direction from any given point, and can be thought of as an $M \times N$-dimensional topographical map, where walking up a hill in one direction costs more than walking downwards in a different direction. On the Euclidean plane, the metric is $d(p_1, p_2) = \|p_2 - p_1\|_2$, but a metric can be defined in many ways as long as it is locally linear and a true metric: that it is always positive except at $p_1 = p_2$ where it is zero, that it is symmetric, and that it satisfies the triangle inequality. The metric chosen to define the manifold can be constructed to heavily penalize certain types of image variations, while allowing other variations to have low costs. For example, we would like an image metric that allows scene lighting to change at little cost, while object instance changes should come with a very high cost.

The length L of a path $I(t)$ from $t = 0$ to $t = 1$ on a manifold is defined, for any given metric $\|\cdot\|$, to be

$$L(I(0), I(1)) = \int_0^1 \left\| \frac{dI}{dt} \right\| dt, \tag{4.1}$$

which a reader might be more familiar with in 2D where $x(t)$ and $y(t)$ are the x-

and y-components of $I(t)$:

$$L(I(0), I(1)) = \int_0^1 \left| \frac{dI}{dt} \right| dt \tag{4.2}$$

$$= \lim_{\Delta \to 0^+} \int_0^{1-\Delta} \left| \frac{I(t+\Delta) - I(t)}{\Delta} \right| dt \tag{4.3}$$

$$= \lim_{\Delta \to 0^+} \int_0^{1-\Delta} \sqrt{\left(\frac{x(t+\Delta) - x(t)}{\Delta} \right)^2 + \left(\frac{y(t+\Delta) - y(t)}{\Delta} \right)^2} dt \tag{4.4}$$

$$= \int_0^1 \sqrt{\left(\frac{dx}{dt} \right)^2 + \left(\frac{dy}{dt} \right)^2} dt. \tag{4.5}$$

In order to calculate the geodesic path connecting $I(0)$ and $I(1)$, we must find the minimum cost path $I(t)$ along the manifold. This becomes an optimization problem, where we want to solve $I_{\text{geod}}(t) = \arg\min_{I(t)} \; L(I(0), I(1))$. Geometrically, a geodesic is a curve whose tangent vectors $\frac{dI}{dt}$ have constant length [29]. It can be shown that the length of the geodesic $I_{\text{geod}}(t)$ is also equal to

$$L_{\text{geod}}(I(0), I(1)) = \min_{I(t)} \sqrt{2E(I(t))}, \tag{4.6}$$

a function of the energy E of the curve [93], where energy is defined as

$$E(I(t)) = \frac{1}{2} \int_0^1 \left\| \frac{dI}{dt} \right\|^2 dt, \tag{4.7}$$

which is familiar from classical mechanics where kinetic energy is $\frac{1}{2}mv^2$. The Cauchy-Schwartz inequality says that $2E \geq L^2$, and for any path at constant speed we have $2E = L^2$. The relation (4.6) can be understood intuitively because the tangent vectors all have constant length c, and so if $\int_0^1 \|c\| dt$ is minimal, then $\frac{1}{2} \int_0^1 \|c\|^2 dt$ must also be minimal, as squaring is a monotonic function. Therefore,

$$I_{\text{geod}}(t) = \arg\min_{I(t)} \int_0^1 \left\| \frac{dI}{dt} \right\| dt = \arg\min_{I(t)} \frac{1}{2} \int_0^1 \left\| \frac{dI}{dt} \right\|^2 dt. \tag{4.8}$$

We will choose an appropriate energy function and use the relation from (4.6) to help us calculate geodesic distances on the image manifold.

The metric defining the manifold on which the geodesics live can be adjusted for various applications, making this an elegant framework to handle an often messy problem, allowing images to update gradually and continuously through time. In the next sections we will discuss the metric and optimization schemes chosen to efficiently solve this problem.

4.3 A Lighting-Insensitive Metric

The pixel-based metric proposed in Chapter 3 was designed to be insensitive to changes in scene illumination, which we combined with a regularization term to handle deformations in an Optical Flow-like framework, calling the combined method the Deformation and Lighting Insensitive (DLI) metric. The lighting-insensitive (LI) term relating two images I_1 and I_2 was presented as

$$E_{\text{LI}}(I_1, I_2) = \frac{1}{2} \sum_{x,y} \frac{\|\nabla \delta I(x,y)\|^2}{\|\nabla I(x,y)\|^2 + \epsilon^2}, \tag{4.9}$$

where $\nabla \delta I$ and ∇I are defined in terms of I_1 and a second image $\hat{I}_2$ that is I_2 warped to match I_1 as closely as possible under certain constraints, so $\delta I = \hat{I}_2 - I_1$ and $\nabla I = \nabla I_1$. The small constant ϵ is of the order of the noise in the image, and ensures that the denominator is never zero.

Using image gradients instead of intensities directly is known to be less sensitive to changes in lighting, for example from [55]. The Gradient Direction is a cost function commonly used when insensitivity to illumination change is desired.

The direction of the image gradient $\theta = \tan^{-1}\left(\frac{I_y}{I_x}\right)$ is calculated at each pixel, then used in a sum-of-squared-differences or L_1-norm image comparison, defining a cost between a given pair of images. This measure is invariant to adding a constant value to the image, or multiplying the image by a scalar, desirable properties for being insensitive to changes in scene illumination. However, it can be argued that a small change in illumination should be penalized less harshly than a large change in illumination.

The metric E_{LI} has similar properties to the Gradient Direction, but is able to respond to different gradient relations appropriately, scaling the gradient of the image change δI by the norm of the image gradient. Changing from a small to a medium gradient norm will be penalized more severely than changing from a medium to a large gradient norm. Comparing two smooth image regions should have a low cost, while comparing a smooth region to a jagged region should have a high cost. The image gradient is small at pixels that correspond to smooth regions of an object, and although a change in scene lighting will result in different pixel intensities, the relative intensities of the pixels will remain similar, and the gradient will remain small, so both the numerator and the denominator will be small, resulting in a low matching cost, and the desired property holds. In an image region where there is a geometric boundary, such as at the edge of a building, a change in scene lighting could affect the distinct surfaces in very different ways, but as the gradient is likely to be large across this boundary, matching a larger $\nabla\delta I$ is permissible at a lower cost as it will be weighted by the image gradient in the denominator. In an image

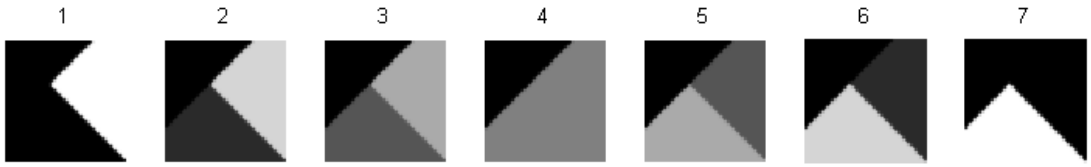
region where there is an albedo change but little geometric change, for example a colored stripe on a white wall, the gradient across this boundary may be large, but as the scene lighting changes, $\nabla \delta I$ will scale with ∇I, so as long as the pixels being compared correspond to the same points in the scene, the matching cost will remain low.

To understand the difference in behavior between the Gradient Direction and our new cost $E_{\text{LI mfld}}$, we provide a simple toy example Fig. 4.1, which could represent a series of images captured as a lighting source moves from one side of a building to another across a corner. Costs are calculated from the leftmost image in Fig. 4.1(a) to all images in the sequence, and these costs are presented in Fig. 4.1(b). As the change in intensity gets larger, the cost of $E_{\text{LI mfld}}$ steadily increases, and when the order of the intensity magnitudes reverses (from image 3 to image 5), this causes a jump in the costs. With Gradient Direction (mod π), the cases where two image regions have the same intensity (images 4 and 7) cause the comparison cost value to blow up, while otherwise the direction of the gradients and hence the costs are not discriminative.

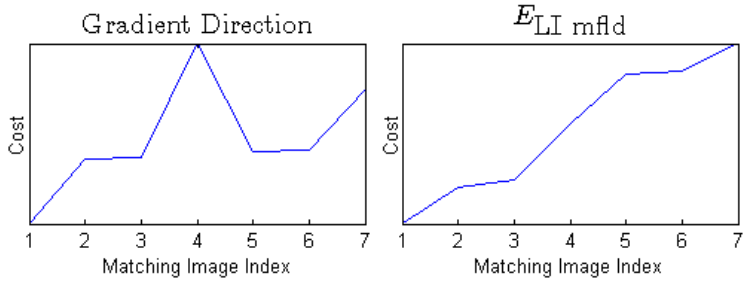
We will use this metric to define a manifold that is insensitive to changes in illumination. Along any curve $I(t)$ (a continuum of images) on the manifold, for small step $\delta t > 0$, $\delta I(t) = I(t + \delta t) - I(t)$. The relation between two images from (4.9) defines a Riemannian structure on images using the infinitesimal norm

$$\|\delta I\|^2_{\text{LI}} = \frac{1}{2} \sum_{x,y} \frac{\|\nabla \delta I(x,y)\|^2}{\|\nabla I(x,y)\|^2 + \epsilon^2}. \tag{4.10}$$

Using this term in the energy function from (4.7), the energy of a curve $I(t)$ on this



(a) Image sequence.



(b) Image matching costs.

Figure 4.1: (a) Image sequence, where each image is compared to image 1, the leftmost image. (b) Gradient Direction and $E_{\text{LI mfld}}$ costs for each image pair in the image sequence.

manifold is

$$E_{\text{LI mfld}}(I(t)) = \lim_{\delta t \to 0^+} \frac{1}{2} \int_0^{1-\delta t} \frac{\|\delta I\|_{\text{LI}}^2}{(\delta t)^2} dt. \tag{4.11}$$

We search for geodesics on this manifold in order to determine the distance $L_{\text{geod}}(I(0), I(1))$ between any given pair of input images $I(0)$ and $I(1)$. To calculate the geodesic from (4.8) we must therefore solve

$$I_{\text{geod}}(t) = \arg\min_{I(t)} \lim_{\delta t \to 0^+} \frac{1}{2} \int_0^{1-\delta t} \sum_{x,y} \frac{\|\nabla \delta I(x,y,t)\|^2}{\|\nabla I(x,y,t)\|^2 + \epsilon^2} \frac{1}{(\delta t)^2} dt \tag{4.12}$$

for fixed $I(0)$ and $I(1)$.

4.3.1 Behavior of the Metric

In this section we will discuss the behavior of the geodesics defined by this lighting-insensitive metric at a single point location (x, y). When the image gradient is near zero, the metric is dominated by the $\frac{1}{\epsilon^2}$ term, and the cost scales nearly linearly with the change in the gradient.

In regions where the image gradients are large, the behavior is more exponential. This can be seen analytically without loss of generality if we consider the case where the gradient is zero in the y dimension in both images, so that there is no change in gradient direction and $\nabla I = I_x$. For clarity let $\epsilon^2 = 0$, and take $I' = \lim_{\delta t \to 0^+} \frac{\delta I}{\delta t}$. This reduces (4.12) to

$$\arg\min_{I(t)} \frac{1}{2} \int_0^1 \left(\frac{I'_x}{I_x} \right)^2 dt, \tag{4.13}$$

which can be solved using the Euler-Lagrange equation [26], a technique that converts a functional to be minimized into a differential equation describing the minimizing function. Specifically, given a functional J of the form

$$J(f) = \min_{f(t)} \int_0^1 F(t, f(t), f'(t)) dt, \tag{4.14}$$

the function $f(t)$ that minimizes $J(f)$ is described by the equation $\frac{\partial F}{\partial f} - \frac{d}{dt}\frac{\partial F}{\partial f'} = 0$. Applying the Euler-Lagrange equation to (4.13), the resulting differential equation

is simplified to

$$\frac{-2(I'_x)^2}{I_x^3} - \frac{2I''_x I_x^2 - 4I_x(I'_x)^2}{I_x^4} = 0 \tag{4.15}$$

$$\frac{2}{I_x^3}\left(-I''_x I_x + (I'_x)^2\right) = 0 \tag{4.16}$$

$$\implies \quad (I'_x)^2 - I_x I''_x = 0. \tag{4.17}$$

It can be shown that $I_x(t) = ce^{rt}$ satisfies this equation for $c, r \in \mathbb{R}$, and any set of boundary conditions $I(0)$ and $I(1)$ will determine the specific values of these variables. We therefore see that when the value of ϵ is small with respect to the magnitudes of $\nabla\delta I$ and ∇I, the gradient of I behaves like an exponential, meaning that I changes exponentially with time. So the cost function we seek to minimize is near linear when the image gradients are near zero, and near exponential when the image gradients are larger, which penalizes scene lighting variation as desired.

4.3.2 Disadvantages of Direct Optimization

The most straightforward way to minimize a function is to use a gradient descent scheme, and the function to be minimized in (4.12) does have a well-defined gradient at all points. However, for input images of size $M \times N$, the geodesic path $I(t)$ has dimension $M \times N \times T$, where T is the number of time steps used to discretize the geodesic. The gradient descent scheme easily gets trapped in local minima for such large dimensional problems, no matter what step size update method is used. Further, the cost contribution from each pixel is determined by the distribution of pixel values in a neighborhood around that pixel, as gradients are fundamentally defined as change over a neighborhood, whether they are calculated using a finite

difference filter of small support, or a smoothed gradient filter with broader support. So the definition of the gradient at each point depends on the values of many neighboring points, a cluttered and slow calculation. We avoid these computations by moving the problem into the wavelet domain, where we will show that it can be expressed as $M \times N$ distinct 1D problems that are straightforward to solve.

4.4 Optimization in the Wavelet Domain

We show that moving the norm $E_{\text{LI mfld}}$ into the wavelet domain results in a function that can be minimized over each independent variable separately, thereby vastly simplifying the minimization calculations and resulting in a very fast computation. We will also find that this representation provides insensitivity to moderate amounts of deformation.

4.4.1 Background on Wavelets

For our purposes, wavelets are a set of orthonormal functions that allow local analysis of a function according to scale; for details see [59]. A family of wavelet basis functions is constructed from a single function $\psi(t)$ that is zero everywhere except in a local region of finite support. The family of wavelets $\psi_{s,b}$ takes the original wavelet function ψ, scales it by s and translates it by b according to the relation

$$\psi_{s,b}(t) = \frac{1}{\sqrt{2^s}} \psi \left(\frac{t - 2^s b}{2^s} \right), \tag{4.18}$$

where $s, b \in \mathbb{Z}$. In this work we will be considering the 2D discrete wavelet transform

(DWT), which is defined by the wavelet family $\psi_{s,b}(t)$ and by a scaling function $\phi_{s,b}(t)$ which is a low pass filter that basically downsamples its input by a factor of two. At every scale the wavelet transform has three outputs, defined in the horizontal, vertical and diagonal directions, and a downsampled version of the input that is then processed at the next scale. At one scale the outputs are defined as

$$H(x, y) = \psi(x)\phi(y) \tag{4.19}$$

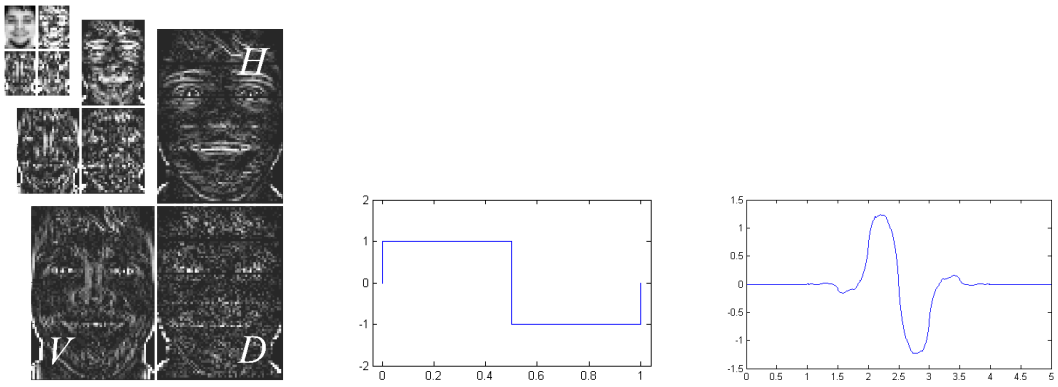
$$V(x, y) = \phi(x)\psi(y) \tag{4.20}$$

$$D(x, y) = \psi(x)\psi(y), \tag{4.21}$$

see Fig. 4.2(a). The horizontal and vertical components are each downsampled using ϕ in one dimension, and transformed by the wavelet ψ in the other dimension. A function I can be rewritten in terms of its orthogonal projection onto wavelet basis functions, where the coefficients of each basis element $\psi_{s,b}$ are defined by $\langle I, \psi_{s,b}\rangle$, and the function as a whole can be expressed as

$$I = \sum_s \sum_b \langle I, \psi_{s,b}\rangle \psi_{s,b}. \tag{4.22}$$

The wavelet transform of an image at the lowest scale ($s = 1$) returns a convolution of the image with the appropriate wavelet functions $\psi_{1,b}(t)$ in each of the three directions, and a convolution with the scaling function which results in an image approximately equivalent to the input image but downsampled by a factor of two in each dimension. For the next scale ($s = 2$), the image that was convolved with the scaling function is then convolved with the high pass directional wavelets in each of the three directions, and again with the scaling function to produce a



(a) (b) (c)

Figure 4.2: (a) 2D Haar wavelet decomposition to three scales, (b) 1D Haar wavelet, (c) 1D biorthogonal spline wavelet.

new approximate image downsampled by another factor of two in each dimension. This process can be repeated until the size of the low pass output is as small as the support of the wavelet function. An example of the output of the DWT is seen in Fig. 4.2(a).

Wavelet functions ψ can be constructed in a wide variety of forms, but for our purposes we consider only functions that have the same general form as a derivative filter. The 1D Haar wavelet at one scale is exactly a simple finite difference filter, see Fig. 4.2(b), and so filtering with a Haar wavelet is equivalent to downsampling by two and filtering with a finite difference filter in each dimension, i.e. extracting the gradient at every other pixel. The critical observation here is that each term of this wavelet transform is independent. Wavelet basis functions can be chosen to be orthogonal, and in this case changing the value of the wavelet coefficient at one location at one scale affects no other coefficients at any scale, for its support of two adjacent points at the next coarsest scale is downsampled by two, so these

points influence no other coefficient. This allows us to define gradients in terms of independent wavelet coefficients. If we filter with a smoother wavelet of a similar gradient-like shape, such as the biorthogonal spline wavelet, see Fig. 4.2(c), this can be considered to be filtering with a smoothed gradient filter, which results in comparable wavelet decompositions but has desirable continuity properties. In this work we will use the family of biorthogonal spline wavelets (with orders nr = 1, nd = 3).

4.4.2 The Lighting Metric in the Wavelet Domain

We rewrite the function $E_{\text{LI mfld}}$ (4.11) in terms of wavelet coefficients. If these coefficients are defined so that $H(m,n)$ is the horizontal component and $V(m,n)$ is the vertical component of a 2D gradient-like wavelet calculated via a discrete wavelet transform, then $H \approx I_x$ and $V \approx I_y$, where each has been downsampled by a factor of two. Using the L^2 norm, E_{LI} can be rewritten approximately as

$$E_{\text{wav}}(I) = \frac{1}{2} \sum_{m,n} \frac{\delta H^2 + \delta V^2}{H^2 + V^2 + \epsilon^2}, \tag{4.23}$$

where H and V depend on point locations (m,n), but we leave this out of the notation for clarity, as (m,n) are fixed inside the sum. The converted cost function does not make use of the diagonal component of the 2D wavelet decomposition, as all terms are expressible using only H and V.

In the wavelet domain, each wavelet basis location is now independent of its neighbors, as the local descriptions of the gradients are handled during the wavelet filtering, a result of the orthogonality of the wavelets as described in the previous

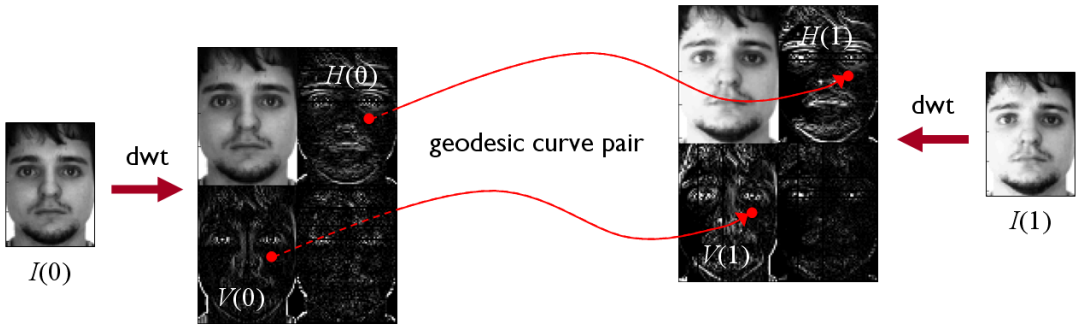


Figure 4.3: Algorithm schematic: The discrete wavelet transform (dwt) is applied to the input images to generate the horizontal and vertical components H and V of the wavelet decomposition at one scale. At each point pair location in $H(0), V(0)$, the geodesic curve is calculated to the corresponding point location in $H(1), V(1)$. These curves are then integrated, and the resulting values from each point pair are summed for the total image matching cost.

section. A primary contribution of this work is the insight that using the wavelet domain to express an image matching cost function based on gradients allows the similarity computation to be separated into independent problems at every point location in wavelet space. We recall that the terms comprising the cost function in the wavelet domain are sampled from the original terms at every other pixel. Again taking $H' = \lim_{\delta t \to 0^+} \frac{\delta H}{\delta t}$ and $V' = \lim_{\delta t \to 0^+} \frac{\delta V}{\delta t}$ so that the $\frac{1}{(\delta t)^2}$ term cancels, the minimization problem (4.12) can be rewritten as

$$I_{\text{geod}}(t) = \frac{1}{2} \sum_{m,n} \underset{H(t),V(t)}{\arg\min} \int_0^1 \frac{H'^2 + V'^2}{H^2 + V^2 + \epsilon^2} dt. \tag{4.24}$$

where H and V are curves through time, and each individual point on the curves is in $\mathbb{R}^{M \times N}$. The $M \times N \times T$ dimensional problem of (4.12) has now been separated into $M \times N$ independent continuous 1D problems to be summed, one for each location

(m, n) in the wavelet domain. The geodesic path at each point location is defined by two 1D curves, $H(t)$ and $V(t)$, which are coupled, meaning that their geodesic paths are co-dependent and are optimized together; see Fig. 4.3. We can calculate the geodesic path for each point location separately, and then the full geodesic path of the image as a whole is simply the combination of all these distinct paths. The starting and ending values $H(0)$, $H(1)$, $V(0)$, $V(1)$ are the coefficients from the wavelet decompositions of the given images $I(0)$ and $I(1)$, and so this reduces to a series of boundary value problems.

The minimization problem in (4.24) is a functional of a form that can be easily converted to a set of differential equations using the Euler-Lagrange equation [26], as described in Sec. 4.3, which can then be solved numerically. We chose to first convert the relation into polar coordinates, as this proved to be more stable to solve numerically. Defining $r = \sqrt{H^2 + V^2}$ and $\theta = \tan^{-1} \frac{V}{H}$, the inner functional to be minimized becomes

$$\arg\min_{r(t),\theta(t)} \int_0^1 \frac{r'^2 + r^2\theta'^2}{r^2 + \epsilon^2} dt. \tag{4.25}$$

Following the vector form of the Euler-Lagrange equation, the differential equations that describe the curves $r(t)$ and $\theta(t)$ that together minimize the term inside the sum for a single point location (m, n) are

$$\begin{aligned} r'' &= r\theta'^2 + (rr'^2 - r^3\theta'^2)(r^2 + \epsilon^2)^{-1}, \\ \theta'' &= 2r^{-1}r'\theta'(r^2(r^2 + \epsilon^2) - 1). \end{aligned} \tag{4.26}$$

This pair of second order equations can be solved as a system of four first order equations using any numerical integration scheme, and we chose to use the

Boundary Value Problem solver from MATLAB. The output is a pair of numerical 1D curves $r(t)$ and $\theta(t)$, starting at $r(0), \theta(0)$ and ending at $r(1), \theta(1)$, that can be converted back to 1D curves $H(t), V(t)$, and that minimizes the cost from (4.24). This process is repeated for each wavelet domain point (m, n) separately. We now have $M \times N$ pairs of geodesic curves. A visual schematic of the algorithm can be seen in Fig. 4.3.

Once all the optimal curves have been found, it remains to integrate along each of them to calculate the cost contribution from each location, and sum these point costs for the overall value of the energy of the image matching. These integrations can be computed numerically, discretizing the curve into T segments and summing the value of the cost function at each of these segments. Once the total energy is calculated, we recall the relation from (4.6) and return the square root of twice the energy value as the true geodesic length.

4.4.3 Limiting Behavior

When ϵ is reduced to 0, equation (4.25) decouples into two separate problems:

$$\arg\min_{r(t)} \int_0^1 \frac{r'^2}{r^2} dt \quad \text{and} \quad \arg\min_{\theta(t)} \int_0^1 \theta'^2 dt. \tag{4.27}$$

These equations are optimized by exponential curves in $r(t)$ and linear curves in $\theta(t)$, and when the boundary values are included, the optimal curves are

$$r(t) = r_0 e^{\ln \frac{r_1}{r_0} t} = r_0 \left(\frac{r_1}{r_0}\right)^t \quad \text{and} \quad \theta(t) = (\theta_1 - \theta_0)t + \theta_0. \tag{4.28}$$

These functions can be integrated analytically, resulting in a total energy of

$$E = \left(\ln \frac{r_1}{r_0}\right)^2 + (\theta_1 - \theta_0)^2 , \tag{4.29}$$

a value determined entirely by the boundary points, invariant to the path connecting them. This is observed to be exactly the cost of the Gradient Direction plus a constant term depending on the ratio of the lengths of the H and V terms in the two images. So we expect the cost reported here to be very similar to the Gradient Direction, but more highly penalizing cases where the difference in gradient norms between the two images is large, while the Gradient Direction is invariant to uniform scalar changes in intensity magnitude. It is reasonable and often desirable to have cases where a uniform intensity change is small be penalized less than cases where the magnitude is large. When the magnitude of r is the same at corresponding pixels in both images, the cost is exactly that of the Gradient Direction. In this limiting case when $\epsilon = 0$ the geodesic path is not meaningful, but for all positive ϵ a geodesic path does exist. When the gradient norms are small, we prefer the linear penalty incurred by the ϵ term, as discussed in Sec. 4.3.1, so that small amounts of noise in smooth regions do not bias the measure.

In practice when ϵ is positive, these properties are consistent, but the geodesic cost is influenced by its entire path on the manifold. The cost to rotate by an angle θ when $r_1 = r_2$ is essentially constant, regardless of the magnitude of r_1. The cost to go from (r_1, θ_1) to (r_2, θ_2) is close to the cost of rotating a constant r by $\theta_2 - \theta_1$ plus the cost of scaling from r_1 to r_2 without any rotation.

4.4.4 Deformation Insensitivity

The algorithm presented above provides a way to compare images that is insensitive to changes in scene lighting conditions. We now claim that this algorithm can also handle moderate amounts of deformation. We first expand our function to include several scales of wavelet coefficients instead of just one. The function to be minimized is now

$$I_{\text{geod}}(t) = \frac{1}{2} \sum_{m,n} \sum_{s} \underset{H(t),V(t)}{\arg\min} \lambda_s \int_0^1 \frac{H'^2 + V'^2}{H^2 + V^2 + \epsilon^2} dt \tag{4.30}$$

where s is the scale of the wavelet, larger scales corresponding to coarser levels of the decomposition. We choose the weighting coefficient on each scale to be $\lambda_s = 2^s$ in order to more highly weight the coarser scales, which we justify from its similarity to the Wavelet Earth Mover's Distance weighting as discussed later in this section. The coarsest images capture global geometric properties while essentially ignoring small image deformations, and in general have coefficients that are smaller in magnitude than those from the smaller scales. This choice of weights was also observed empirically to provide the most accurate results. Using several scales increases the accuracy of our method, as will be seen in the experiments section below. This is expected because we can now consider both global image properties from the coarse scales, and edge details that define specific instances of an object from the finer scales, and both cases are handled appropriately in the presence of lighting change. In our experiments we will use the first three scales of the biorthogonal spline wavelet. The resulting algorithm now involves a separate geodesic curve construction and integration for each scale and location.

We now argue that simply using wavelets adds some insensitivity to deformation. The image pixels within the support of each individual wavelet basis function are handled together during the wavelet transform, so deformations within this region can be modeled together. A similar observation was made previously when the Earth Mover's Distance was explored in the wavelet domain. The Wavelet EMD [73] approximates the Earth Mover's Distance in the wavelet domain, and its cost depends on wavelet coefficients at all scales.

The Earth Mover's Distance (EMD) algorithm [71] provides a way to compare two distributions by measuring the distance and quantity of "mass" that must be moved in order to convert one distribution into the other, where "mass" is thought of as whatever is populating the bins of a histogram. This similarity measure captures certain types of deformation, where no particular geometric structure is preserved or favored, but local changes in mass cost significantly less than global structure modifications. The Wavelet EMD [73] expresses the Earth Mover's Distance in the wavelet domain, converting an algorithm of complexity $O(n^3 \log n)$ into an $O(n)$ algorithm without any significant performance difference, where n is the number of points in an image.

The Wavelet EMD cost depends on wavelet coefficients at all scales. At each individual scale, it limits the distance individual mass units can move to the span of the wavelet at that scale. The weighting on the magnitude of each scales' wavelet coefficients in the distance calculation is $2^{2s} = 4^s$, similar to the weighting we incorporated into our multiscaled cost function (4.30), where our base is 2 instead of 4. When the image gradients are small, our proposed cost function is essentially

linear, as discussed in the end of Sec. 4.3, meaning that it behaves similarly to the Wavelet EMD, and we understand how this new metric is able to handle moderate amounts of deformation, as this is the purpose of the Earth Mover's Distance. When the image gradients are larger, the new metric becomes more exponential, which allows the image comparison to be penalized less heavily when large lighting changes are present.

We have now described a method to compute geodesic paths between images by reformulating the manifold's metric in the wavelet domain. A primary contribution of this work is the insight that using the wavelet domain to express an image matching cost function based on gradients allows the similarity computation to be separated into independent problems at every point location in wavelet space. 1D geodesic paths directed by the cost function can be analytically computed and numerically integrated at each point, and the sum of these point costs can be summed for the overall image matching cost, i.e. the length of the geodesic between the images on the manifold. We will now discuss how these calculations can be further optimized to create a very computationally efficient algorithm.

4.5 The Faster Algorithm

We will now discuss how to optimize our calculations to create a very computationally efficient algorithm. For any given pair of starting input values $H(0), V(0)$ and ending input values $H(1), V(1)$, the geodesic curve connecting them is always the same, so the cost of this input is always the same. This means that the geodesic

curves can be calculated and integrated offline, and at run time the only computation that has to be performed is to look up the value of the integral for the given $(H(0), V(0), H(1), V(1))$. To further reduce the amount of space and time required, at every point we convert the input $(H(0), V(0), H(1), V(1))$ into polar coordinates, $(r_1, r_2, \theta_1, \theta_2)$, and then rotate so that $\theta_2 = 0$, as these rotated values preserve the relation between the points and will result in the same output cost. This allows us to generate a lookup table of integral values that depends on only three values $(r_1, r_2, \Delta\theta)$ instead of four.

We discretize the space of r values into 40 bins of exponentially increasing size in the range $[0, 1.5]$, as this is the range of wavelet coefficient values observed in practice for images with pixel values in $[0, 1]$, with coarser scales generally consisting of smaller values. We used $\epsilon = 0.01$ in our experiments. The space of $\Delta\theta$ values we discretize into 80 bins of uniform size in the range $[0, 2\pi)$. The resulting costs are symmetric about $\Delta\theta = \pi$, so we really only have to store the first half of these values, and the lookup table to be stored is of dimension $40 \times 40 \times 40$. The online calculation at each location (m, n) in wavelet space consists of converting $(H(0), V(0), H(1), V(1))$ into polar coordinates $(r_1, r_2, \Delta\theta)$, looking up the corresponding integral value in the table, and adding this value to the overall cost being calculated.

This calculation is limited principally by the speed at which a given machine can perform a lookup in a $40 \times 40 \times 40$ array, which is in general a very fast operation. The cost of this calculation is on the order of milliseconds, fast enough to use in practice when many image comparisons must be computed very quickly. On a 3.16

GHz machine running MATLAB in serial, this takes on average 1.3×10^{-3} seconds for a pair of images with 5000 pixels each. We emphasize that the lookup table is application-independent; once it has been generated, which takes 1.5 hours, the same table can be used for any pair of images from any domain.

An outline of the algorithm used for our experiments is provided in Algorithm 2.

Algorithm 2 Calculating the wavelet-based image matching cost along lighting-insensitive geodesics.

Input images I_1 and I_2

Compute wavelet transform of each image for scales $s = 1, 2, 3$:

$$H_{1,s}, V_{1,s} \leftarrow I_1, \quad H_{2,s}, V_{2,s} \leftarrow I_2$$

for all scales $s = 1, 2, 3$ **do**

for all points locations (m, n) in wavelet image at scale s **do**

$[r_1, r_2, 0, \Delta\theta] \leftarrow \text{polar}([H_{1,s}(m,n), V_{1,s}(m,n), H_{2,s}(m,n), V_{2,s}(m,n)])$

Calculate length of geodesic minimizing (4.25) from $[r_1, 0]$ to $[r_2, \Delta\theta]$

or

Look up value in bin containing $[r_1, r_2, \Delta\theta]$ from pre-generated table

end for

sum values, weight by λ_s

end for

return final sum is image matching cost

4.6 Experiments

4.6.1 Face Recognition

One class of object that is regularly presented with large amounts of lighting variation and moderate amounts of deformation is the human face. Although nothing in our algorithm is specific to faces, the limited amount of deformation present with expression change, along with potentially high variations due to lighting change, make them a relevant application of our work. We use a common face dataset studied for this problem, the subset of the AR Face Database [62] that contains variation in expression and lighting. We reduce the size of the standard cropped AR images by a factor of two in each dimension, as face recognition algorithms routinely perform the best on images of this scale, and so the images we compare are 83×59 pixels in dimension, and are smoothed slightly before processing. We use a neutral face from each of the 100 people in the dataset as gallery images, and the three variations in expression and the three variations in lighting for each person comprise the test set; see Fig. 2.3. The identity of each test image is determined by the gallery image returning the lowest cost pairing.

The algorithm presented here is a fast method for comparing images in the presence of lighting change and moderate deformations, and so we compare to other lighting and deformation insensitive algorithms that do not require training data. It was shown in [35] that the Gradient Direction method, described around equation (2.9), consistently performs better than the other standard pixel-based lighting-insensitive methods (Self-Quotient, luminance map estimation, Eigenphases, Whiten-

ing), so we compare to Gradient Direction. We also compare to the results of the Deformation and Lighting Insensitive metric (E_{DLI}) from Chapter 3, and we expect our calculations to be much faster. Other works that present a cost function to handle both lighting change and deformations include that of [97], which calculates image point correspondences using edge maps and Gabor jet information, and [74] which uses mutual information to combine binary edge features with grayscale information. We also compare to simple image differencing and to normalized cross-correlation [52], where the template is a full image, as these methods are frequently used to compare images when many comparisons must be completed very fast. As our method is based on an L^2 metric, we use the L^2 norm on each of these measures for valid comparison. Results on the AR Face Database are presented in Table 4.1 for both algorithm speed and accuracy.

We see that our method achieves more accurate results than the Gradient Direction method on the lighting variation images, and significantly more accurate results on the expression variation images, as expected. This confirms the insensitivity of the method to lighting change, with the added benefit that we are able to construct geodesic information which allows for meaningful extensions such as mapping and interpolating large image variations. The accuracy of the method is also above that of the E_{DLI} work where the lighting metric was first presented, which also handled deformations explicitly, and our calculations here are 10^3 times faster than that work, making our method useful in template matching applications where the original method was prohibitively slow.

The previous best results on this dataset, as far as the authors are aware, were

produced by Pixel-Level Decisions in [43], where simple thresholding was applied to pixel differences of a chosen image property. Standard deviation calculated within a window around each pixel was the property that provided the best results. The differences between these standard deviations at every pixel location in each image were computed, and the total number of pixel differences less than a pre-determined threshold were counted for the final similarity value. We present these results here to demonstrate that the surprisingly strong results achieved from this extremely simple algorithm can be applied to other pixel-based methods, and we use a similar thresholding step on our results as well. [43] also suggests compensating for local error by repeating the procedure with the images shifted a few pixels in every direction, but we do not compare these results as they are not relevant to the ideas in this paper. However, this repeated shifting could be applied to improve the results of any of the these methods. As the threshold value for our point costs in wavelet space, we use the cost value that counts the lowest 20% of the point costs across all images, as this was the value used by [43]. The exact threshold value is not sensitive, and we observed that all values thresholding 9% to 47% of the costs resulted in overall accuracies within 1% of each other, and the ideal threshhold on this dataset, if hand-picked, results in an overall accuracy of 98.0%. We see in Table 4.1 that this simple thresholding extension removes 58.6% of the errors in our method.

The proposed algorithm performs well with variations in lighting, and also handles moderate amounts of deformation. Many methods perform very poorly on the scream category of this database, but the multiscaled method presented here achieved 83.0% accuracy in this case, and 93.0% with thresholding, higher than

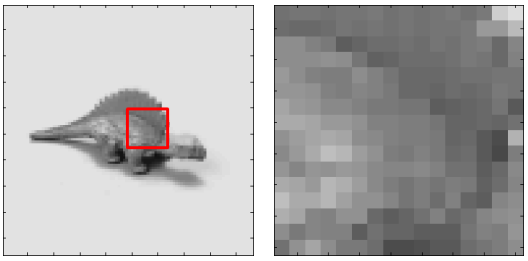
Method	*Time (sec)*	*Expression*	*Lighting*	*Overall*
Image Differencing	3.1×10^{-5}	83.0%	9.0%	46.0%
Normalized Cross-Correlation [52]	7.2×10^{-3}	84.0%	59.3%	71.7%
Significant Jet Point [97]	–	80.8%	91.7%	86.3%
Binary Edge Feature and MI [74]	–	78.5%	97.0%	87.8%
Gradient Direction [35]	3.8×10^{-4}	85.0%	95.3%	90.2%
E_{DLI} (Chapter 3)	1.0×10^{0}	89.6%	98.9%	94.3%
Proposed Method	1.3×10^{-3}	93.7%	96.7%	95.2%
Pixel Level Decisions [43]	5.6×10^{-4}	98.0%	94.0%	96.0%
Proposed Method thresholded	1.3×10^{-3}	97.3%	97.0%	97.2%

Table 4.1: Identification results on the AR Face Database. The *Time* column reports the MATLAB calculation time of a single image pair comparison in seconds, except in two cases where time was not reported and we unable to reproduce the authors' results.

either Gradient Direction (57.0%) or the E_{DLI} metric (79.6%), which was designed to handle deformations as described above.

4.6.2 Template Matching

As a proof of concept that our method can be used effectively in template-matching scenarios where many image comparisons must be made very fast, we consider the NORB Object Recognition Dataset [50], which consists of images of 50 toys imaged under 6 lighting conditions, 9 elevations, 18 azimuths, and many backgrounds; see Figure 4.4. We take as our template a 16×16 patch from an image of a single toy (the stegosaurus) with a plain background under good lighting conditions with an elevation and azimuth of 0 degrees; see Figure 4.4(a) - (b). We search for the best match of this patch centered at every pixel in the 108×108 pixel images of the same toy appearing in cluttered scenes; see Figure 4.4(c). We use all 6 lighting conditions, and to add some "deformation" we include pose variations with the azimuth at 0, 20 and -20 degrees, keeping the elevation at 0. The dataset contains two distinct images in each of these settings (the dataset also contains stereo images, but we only use one of each stereo pair for our purposes), and so we compare the template patch to $108 \times 108 = 11,664$ positions in each of $6 \times 3 \times 2 = 36$ images. If the center of the best matching location is within 8 pixels of the true location (as defined by hand), we declare it to be correct. As fast techniques that might be used for template matching, we compare our proposed method to Normalized Cross Correlation and the Gradient Direction, and the results are presented in Table 4.2.



(a) (b)

(c)

Figure 4.4: Images from the NORB Dataset. (a) The full image from which the template was cropped. (b) The template used. (c) Some images in which the best match for the template was sought.

We see that in this simple trial, the method presented in this work significantly outperforms the other two techniques. We also tested other patches from other animal toys in the NORB dataset, but found that if the patches were not sufficiently smooth, for example the head of the triceratops, then all three methods performed poorly, identifying the correct location less than 50% of the time.

The proposed algorithm is seen to produce accurate identification results, and the computation time required is extremely small. We emphasize that no training data or learning stage is required for our algorithm.

Method	*Localization Accuracy*
Normalized Cross-Correlation [52]	25.0%
Gradient Direction [35]	61.1%
Proposed Method	80.6%

Table 4.2: Template matching results on a subset of the NORB Dataset. If the center of the region most closely matched to the template is within 8 pixels of the true location, the location was declared to be correct.

4.7 Conclusion

We have presented a fast algorithm for handling illumination changes and moderate deformations applicable to any class of images. Geodesic distances were calculated between pairs of images, as defined on an image manifold given structure by an illumination-insensitive metric that was based on the change in image gradients. The metric was calculated in the wavelet domain, where each point location contributed independently to the overall image comparison cost, allowing geodesic costs to be computed extremely efficiently using a pre-calculated lookup table. Using wavelets at multiple scales allowed for insensitivity to moderate deformations in a manner similar to the Wavelet Earth Mover's Distance. Strong results were presented on the AR Face Database, where our algorithm is seen to be both extremely fast and accurate, and we demonstrated that because this method is so fast, it can also be applied successfully in situations where Normalized Cross-Correlation is of-

ten used, where many image comparisons must be computed in a very short amount of time. Using geodesics to calculate image comparisons instead of simple pixel differences allows our method to be incorporated into a wide array of applications where having information along a morphing path is relevant.

Chapter 5

Diffeomorphisms For General Image Comparison

In order to correctly handle deformations in images, we must explicitly account for displacements between regions due to movement, such as the opening of a mouth. This is in contrast to the algorithm of Chapter 4, where images were assumed to be aligned and we calculated geodesics through the space of illumination changes. We would like to solve the image registration problem, which is to determine a dense correspondence of points between any pair of images, especially across large object deformations. One interpretation of a dense correspondence is a one-to-one and onto mapping, which is called a bijection. Being one-to-one and onto is more restrictive than is sometimes desired to handle real-world scenarios in which occlusions and previously unseen image regions regularly appear, but there are many situations in which these restrictions are preferred, especially in medical imaging. This is a valuable first step towards more general transformations, because it will allow us to use a diffeomorphic framework, resulting in deformations that are smooth and have the very powerful property of invertability.

We will search for diffeomorphisms between images on a Riemannian manifold. In Chapter 4, we considered image manifolds where each point on the manifold was an image. Here we also study manifolds where each point is a diffeomorphism, and the origin is the identity diffeomorphism. A Riemannian manifold is a generalized

differentiable surface where the tangent space at each point on the manifold has an inner product, so that distances and angles have meaning. The distance between two points on the manifold is the length of the (shortest) geodesic connecting them, as defined by a metric. A diffeomorphism is a smooth bijective mapping between points on manifolds (technically between points on any two manifolds, but in our case all manifolds are the same manifold), which should be thought of as a deformation through time; see [28, 93] for a complete description of the mathematics. A good mental image of a diffeomorphism is a 2D uniform mesh being deformed within the plane, where no point can cross over any other point because the transformation is one-to-one and onto, but regions can be stretched and shrunk; for example see Figure 5.1. Working with full diffeomorphisms is more meaningful than simple correspondence vector fields as studied in Chapter 3, because with diffeomorphisms, images can deform gradually through time, and the deformations are guaranteed to be smooth and invertible. This allows large image changes to be dealt with more robustly, and in principle changes like occlusions could be introduced slowly and handled explicitly, although in this thesis we consider only deformations and lighting changes.

In this chapter we present an initial foray into the application of diffeomorphisms to image comparisons. We describe a diffeomorphic framework based on a body of literature on the topic, and define a way to apply this framework to face images. We calculate diffeomorphisms and geodesics through diffeomorphisms between images on a manifold, and compare images deformed by diffeomorphisms. Optimization methods and challenges are discussed. We then show how intermediate images

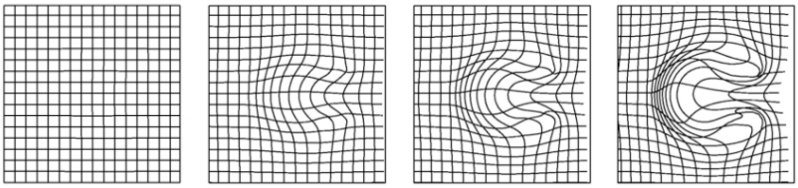


Figure 5.1: An example of a diffeomorphism between two grids, with two intermediate configurations shown (images from [2]).

can be generated along a diffeomorphic path between two known face images of different expressions, and that these generated images are useful for matching to faces of new expressions. One of the goals of this chapter is to present the diffeomorphic framework in a manner accessible to a somewhat broader audience than has previously been a part of the computational diffeomorphism community. The results presented here are rudimentary, and are meant to demonstrate that diffeomorphisms are a useful, manageable, and elegant tool for face recognition with many potential extensions.

5.1 A Diffeomorphic Framework

There is a body of work studying diffeomorphisms between images, often with applications to medical imaging, including [2, 7], and we will set up a diffeomorphic framework in a similar manner here. In order to make the image registration problem well-defined, we need an image-pairing energy cost that is minimized when the images are in correct correspondence. This energy cost function will have a term measuring the amount of deformation required to put the images in correspondence,

and later we will also consider incorporating a term measuring the amount of pixel similarity achieved after the deformation, with a weighting between the two terms.

Consider an image I_0 as a uniform grid of pixels. The image I_0 will be deformed to be put in correspondence with image I_1, and we will think of this deformation as happening through time, so that at time t, $t \in [0,1]$, I_t is an image in a video sequence deforming I_0 to I_1. Let $I : \Omega \to \mathbb{R}$ be an image in the domain $\Omega \subseteq \mathbb{R}^2$, and for a gray scale image we can write that at every location $\vec{x} = [x, y] \in \Omega$, $I(\vec{x}) = c$ for some scalar intensity value c. The pixels of image I are defined on a uniform grid, where pixel i is at location $\vec{x}_i$. To deform the image, the pixels of image I are displaced by a vector field $\vec{v}$, where $\vec{v} : \Omega \to \mathbb{R}^2$, so that point i at location $\vec{x}_i$ is displaced by vector $\vec{v}_i$, and the final location of point i after the deformation is $\vec{x}_i + \vec{v}_i$.

A full diffeomorphism from $t = 0$ to $t = 1$ is defined by a vector field $\vec{v}(t)$ at each time t that deforms the pixels an infinitesimal amount in the given directions at each point. The location and corresponding vector of point i at time t are written respectively as $\vec{x}_i(t)$ and $\vec{v}_i(t)$. A transformation between two images is written as $\phi : \Omega \to \Omega$, and as image I is deformed through time, a path ϕ_t through the space of transformations is traced on the manifold of diffeomorphisms; see Figure 5.2. The final deformation $\phi_1(I_0)$ deforms image I_0 to be in correspondence with I_1, and at any point along the transformation, $\phi_t(I_0) = I_t$ is an image. In other words, $\vec{v}(t)$ defines the direction of the infinitesimal transformation at any given time t, while ϕ_t defines the full transformation from $\vec{x}(0)$, the initial uniform grid, to $\vec{x}(t)$, the locations of the grid points at time t, and depends on each of the $\vec{v}_t$. It is interesting

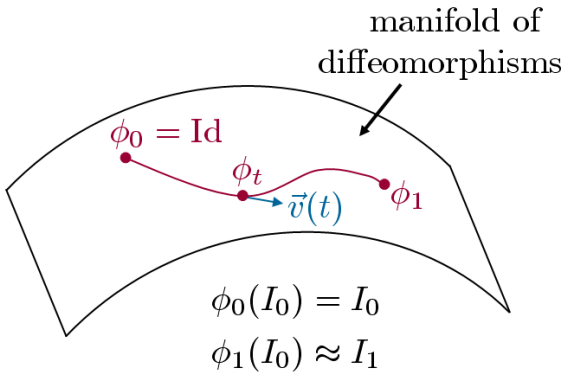


Figure 5.2: Visualization of the manifold of diffeomorphisms. At $t = 0$ the diffeomorphism ϕ_0 is the identity mapping, and at $t = 1$ the diffeomorphism ϕ_1 is the mapping that morphs image I_0 to be in correspondence with image I_1. At time t, the diffeomorphic change is in the direction of $\vec{v}(t)$.

to note that $\phi_1 = \phi_0 + \int_0^1 \vec{v}_t(\phi_t)dt$, and that differentiating with respect to time yields $\frac{d}{dt}\phi_t = \vec{v}_t(\phi_t)$.

In practice, the path ϕ_t must be discretized, and therefore at time $t \in [0, 1]$, the vector field $\vec{v}(t)$ consists of finite non-infinitesimal values. If $\vec{v}$ is not sufficiently smooth and Δt is not sufficiently small, then there could be instances where the paths of points from I_t to $I_{t+\Delta t}$ overlap those of their neighbors, making the mapping from I_t to $I_{t+\Delta t}$ not bijective. However, assuming sufficient smoothness and sufficiently small Δt, this mapping is one-to-one, onto, and differentiable, making the transformation a diffeomorphism.

In order to compute a deformation in this diffeomorphic framework, the vector fields $\vec{v}(t)$ must be known explicitly at every t. The optimal transformation from I_0 to I_1, that warps I_0 into correspondence with I_1, is the one that minimizes a given

energy function over the vector fields at each time.

To enforce smoothness on the vector fields, the cost function used to define the image registration penalizes the norm of the vector field by including a $\|\vec{v}(t)\|$ term to be defined. The energy function may also include a distance function $d(\cdot,\cdot)$ that penalizes discrepancy between $\phi_1(I_0)$ and I_1. So the energy function to be minimized is

$$E(\vec{v}) = \int_0^1 \|\vec{v}(t)\|dt + \lambda d\left(\phi_1(I_0), I_1\right), \tag{5.1}$$

and $\hat{\vec{v}}$, the optimal $\vec{v}$, is calculated as

$$\hat{\vec{v}} = \arg\min_{\vec{v}} E(\vec{v}), \tag{5.2}$$

for some definition of the norms, often the L^2-norm, and some relative weighting λ.

In order to determine the minimum vector fields $\vec{v}$, an optimization scheme must be employed, and it is standard to use a scheme based on Gradient Descent. An initial $\vec{v}(t)^0$ is chosen, and at each iteration is updated via

$$\vec{v}(t)^{k+1} = \vec{v}(t)^k - \epsilon \nabla E(\vec{v}(t)), \tag{5.3}$$

for some stepsize ϵ. This requires the calculation of $\nabla E(\vec{v}(t))$, which can in general be quite complicated, or the use of finite differences [51] or automatic differentiation [65].

We point out that what is really being calculated here is a geodesic flow: the sequence of vector fields that result in the minimum energy path between two given

images.

5.2 Diffeomorphisms Based on Sparse Correspondences

We consider human faces being deformed by various expressions as an application of diffeomorphisms, and we want to define the diffeomorphism problem in an appropriate manner. When a face deforms, typical actions include the opening and closing of the mouth and eyes. These actions introduce new regions or remove existing regions of the face, implying that one-to-many or many-to-one matchings of pixels between images would be technically correct, for example matching all teeth pixels from an open mouth to the boundary between two lips on a closed mouth. However, diffeomorphisms are bijections by definition, and so a mapping that is not one-to-one and onto is not allowed. We accept this property because the resulting maps are smooth and invertible, both very desirable properties, and we will address the potentially disappearing points appropriately.

Diffeomorphisms have been applied to medical imaging in several works, where regions with structures are assumed to deform but to always be present, but less work has been applied to faces. Individual face images have been transformed by diffeomorphisms in studies such as [79, 83], but we are not aware of any cases where diffeomorphisms have been applied to an entire face database with quantitative comparison goals. To properly address all the complexities of face deformation, a robust method to handle regions of the face visible in some images but not others, such as teeth, and occlusions in general, must be incorporated. However, that is not

the goal of this thesis. Here we explore the applications of diffeomorphisms to face recognition, laying a groundwork for further developments.

Rather than try to force points to match regions of the face that might be absent in some images, we elect to use a diffeomorphism algorithm based on a sparse set of face feature points. We use points that appear in all face images (as long as they are sufficiently frontal), and can be automatically detected using published facial feature point detection algorithms. We use the same two fiducial point detection algorithms [10, 27] as described in Chapter 2 Section 2.2 to collect 14 points on each face; see Figure 2.5. We then require the diffeomorphisms we calculate to exactly match these 14 points. The rest of the face image points will be interpolated using an appropriate spline-based interpolation at every time step, and we will then also consider penalizing based on how well the pixels match.

We chose to use the method of bounded diffeomorphisms based closely on the work of Twining et al. [83], which itself is derived from the work of Camion and Younes [20]. Using thin-plate splines penalized by the bending energy to describe a warping between two corresponding sets of points was first presented in the seminal work of Bookstein in [15]. This work is extended to generating diffeomorphic flows in a body of work including [20, 83]. The goal of these works is to construct diffeomorphisms given a sparse set of landmark points and their displacements. The input to the algorithm is the initial and final positions of N points in 2D space. The algorithm determines the appropriate path that each point must travel through time, as penalized by an energy cost function, so that the path of each point is a geodesic. The algorithm also smoothly interpolates the rest of the plane using thin-

plate splines to generate a dense diffeomorphism from the starting to the ending configurations.

The underlying idea behind the diffeomorphism calculations is that the energy cost E_{diffeo} of a diffeomorphism is defined by a differential operator, L, and that the desired configuration defining the diffeomorphism is that of minimum energy. For the moment we will consider only one time step so that time can be removed from the equations for simplicity, but the time variable will be added back later. For point locations $\vec{x}$ and their velocities $\vec{v}$,

$$E_{\text{diffeo}}(\vec{v}(\vec{x})) = \int_{\mathbb{R}^2} \|L\vec{v}(\vec{x})\|^2 d\vec{x}. \tag{5.4}$$

L is often taken to be the Laplacian, $L = \nabla^2 = \sum_i \left(\frac{\partial}{\partial x_i}\right)^2$, but in principle can be any linear operator, and we will require it to be self-adjoint (that is, $\langle Lx, y\rangle = \langle x, Ly\rangle$) so that the energy can be expressed as the following

$$\begin{aligned} E_{\text{diffeo}}(v) &= \int_{\mathbb{R}^2} \|L\vec{v}(\vec{x})\|^2 d\vec{x} = \int_{\mathbb{R}^2} \langle L\vec{v}(\vec{x}), L\vec{v}(\vec{x})\rangle d\vec{x} = \int_{\mathbb{R}^2} \langle \vec{v}(\vec{x}), L^2\vec{v}(\vec{x})\rangle d\vec{x} \\ &= \int_{\mathbb{R}^2} \vec{v}(\vec{x}) \cdot L^2\vec{v}(\vec{x}) dx. \end{aligned} \tag{5.5}$$

It is natural to use the Laplacian operator to define energy because it has a meaningful physical interpretation. A configuration with minimum Laplacian energy corresponds to a situation with minimal second derivatives, and arises for example in physics when heat or a gas has fully diffused and a system is at equilibrium.

In [20], the full energy cost is defined to be the sum of the diffeomorphism energy with a second term, which penalizes the difference between the final position

of the landmark points and their intended corresponding locations. Instead, we use the energy function defined in [83], where exact matching is imposed, meaning that the final position of the points is not a free variable, and this second term is zero, and so E_{diffeo} is the complete cost to be minimized.

The diffeomorphism is defined through a linear combination of basis functions that determine the interpolations between all image points, and these basis functions are chosen to have properties desirable for the given application. The basis functions for the interpolating spline will be the Green's function (described below) of the operator L, and the conditions placed on the spline will therefore determine L, where one of the conditions is that the spline be expressible as a Green's function. Using a Green's function in this way allows calculations to be performed with the diffeomorphism.

Given a linear operator L, the Green's function $G(x, s)$ is defined so that $LG(x, s) = \delta(x - s)$. This function is useful in finding functions $u(x)$ that satisfy the relation $Lu(x) = f(x)$, for a given L and $f(x)$, via the following derivation:

$$\int LG(x,s)f(s)ds = \int \delta(x-s)f(s) \qquad \text{from } LG(x,s) = \delta(x-s)ds \quad (5.6)$$

$$= f(x) \qquad \text{from the definition of the } \delta\text{-function} \quad (5.7)$$

therefore, as $Lu(x) = f(x)$,

$$Lu(x) = \int LG(x,s)f(s)ds = L\int G(x,s)f(s)ds \tag{5.8}$$

$$\implies u(x) = \int G(x,s)f(s)ds, \tag{5.9}$$

and so an expression for the $u(x)$ that satisfies $Lu(x) = f(x)$ can always be found, given the Green's function $G(x,s)$ of L. The Green's function of the Laplacian is $G(x,s) = \frac{1}{|x-s|}$. The Green's function of an operator is not necessarily unique, and can in general be a distribution rather than a proper function; for more details see [26].

In the diffeomorphism setup, the velocity $\vec{v}(\vec{x})$ is expressed as a linear combination of the Green's functions $G(x,s)$ of the operator L^2 calculated at each of the N landmark points, so that

$$\vec{v}(\vec{x}) = \sum_{i=1}^{N} \alpha_i G(\vec{x}, \vec{x}_i). \tag{5.10}$$

The Green's functions are therefore the basis functions defining the diffeomorphism, and so a Green's function with desirable geometric properties is chosen, and the corresponding L^2 is accepted as defining the energy. In order to plug the Green's function into the energy definition of (5.5), we use the definition $L^2G(x,s) = \delta(x-s)$ to obtain an expression for $L^2\vec{v}$,

$$L^2\vec{v}(\vec{x}) = L^2\sum_{i=1}^{N} \alpha_i G(\vec{x}, \vec{x}_i) = \sum_{i=1}^{N} \alpha_i L^2 G(\vec{x}, \vec{x}_i) = \sum_{i=1}^{N} \alpha_i \delta(\vec{x} - \vec{x}_i) \tag{5.11}$$

and plug this into the equation to arrive at a new expression for E_{diffeo} that will be used in calculations:

$$
\begin{aligned}
E_{\text{diffeo}}(\vec{v}) &= \int_{\mathbb{R}^2} \vec{v}(\vec{x}) \cdot L^2 \vec{v}(\vec{x}) d\vec{x} \\
&= \int_{\mathbb{R}^2} \vec{v}(\vec{x}) \cdot \sum_{i=1}^{N} \alpha_i \delta(\vec{x} - \vec{x}_i) d\vec{x} \\
&= \sum_{i=1}^{N} \int_{\mathbb{R}^2} \vec{v}(\vec{x}) \cdot \alpha_i \delta(\vec{x} - \vec{x}_i) d\vec{x} \\
&= \sum_{i=1}^{N} \vec{v}(\vec{x}_i) \cdot \alpha_i \\
&= \sum_{i=1}^{N} \left(\sum_{j=1}^{N} \alpha_j G(\vec{x}_i, \vec{x}_j) \right) \cdot \alpha_i \\
&= \sum_{i=1}^{N} \sum_{j=1}^{N} \langle \alpha_i, \alpha_j \rangle G(\vec{x}_i, \vec{x}_j). \qquad (5.12)
\end{aligned}
$$

This final definition of the diffeomorphic energy is the function we will minimize in order to calculate the desired diffeomorphisms.

In the work of Camion and Younes [20], the Green's function from the original Bookstein thin-plate splines, relating to the bending energy, is used: $G(x, s) = -r^2 \log r^2$, where $r = \sqrt{x^2 + s^2}$. Here we will instead use the clamped-plate spline model presented in [83], because this Green's function is zero and has zero derivative on the unit circle. This is an ideal situation for faces, where we can allow a face within a given circle to deform while keeping the background stationary. We resize each cropped rectangular face image to be unit square, thereby compressing the face regions to be more circular. The Green's function is defined as

$$G(\vec{x},\vec{s}) = |\vec{x}-\vec{s}|^2\left(\frac{1}{2}(A^2-1)-\log A\right), \text{ where} \tag{5.13}$$

$$A(\vec{x},\vec{s}) = \frac{\sqrt{|\vec{x}|^2|\vec{s}|^2 - 2\vec{x}\cdot\vec{s}+1}}{|\vec{x}-\vec{s}|}. \tag{5.14}$$

For any given arrangement of points $\vec{x}_i$, both $\vec{v}_i$ and $G(\vec{x}_i,\vec{x}_j)$ can be directly calculated, and so the α_i from equation (5.10) can be computed by solving a system of linear equations. Bringing back the time variable, we define

$$\vec{v}(\vec{x}_i(t)) = \frac{\vec{x}_i(t+\Delta t)-\vec{x}_i(t)}{\Delta t}, \tag{5.15}$$

and G is an $N\times N$ matrix with entries $G(\vec{x}_j(t),\vec{x}_i(t))$. For any configuration of points $\vec{x}_i(t)$, in order solve for the $\alpha_i(t)$ at each t we can invert the following relation, which in discrete time involves solving a system of linear equations

$$\vec{v}(\vec{x}_j(t)) = \sum_{i=1}^{N}\alpha_i(t)G(\vec{x}_j(t),\vec{x}_i(t)). \tag{5.16}$$

Given the $\vec{v}(\vec{x}_i(t))$ and the $G(\vec{x}_j(t),\vec{x}_i(t))$, the energy of any given configuration can be then calculated from equation (5.12).

In order to find the point configuration resulting in minimum energy, which defines the diffeomorphism, a gradient descent scheme is used as described above. For the starting configuration, we linearly interpolate between the N feature points, so that $\vec{x}_i(t) = \frac{T-t}{T-1}\vec{x}_i(1) + \frac{t-1}{T-1}\vec{x}_i(T)$ for $t = 1,...,T$. Matlab's standard `fminunc()` unconstrained optimization routine is used to automatically estimate the gradient and use the BFGS method to estimate the Hessian; for details see [68].

Finite difference calculations can be quite cumbersome, and in order to have an algorithm that runs efficiently it would be nice to have an analytic expression of the gradient. The energy equation depends on α terms which are solved via a system of linear equations depending on v from equation (5.16). For any given number of landmark points N, it would technically be possible to write down an explicit derivative across this system, but this expression would depend on every term many times over, and the number of arithmetic computations required here is the same order of magnitude as the finite difference itself, while being much more complicated. A way to get around this problem would be to use automatic differentiation [65]. With automatic differentiation, the computer keeps track of which elementary arithmetic operations and functions are executed to calculate a function, and applies the chain rule (potentially thousands of times) to calculate the numeric value of the derivative at any given point. This derivative calculation has the same complexity as the calculation of the original function. However, simple Matlab implementations of automatic differentiation are not able to handle a system of linear equations. The use of a more complete C implementation of automatic differentiation will be considered in the future. For this work we use standard finite differences, accepting the lack of efficiency for our initial studies. With $T = 10$ times steps and $N = 14$ landmark points, using Matlab on a 3.16 GHz machine computing serially, our computations take approximately .8 seconds per iteration, and require roughly 150 iterations per image pair, so to calculate the geodesic diffeomorphism between one pair of images takes approximately 2 minutes. We note that this is independent of image size, and depends only on the number of corresponding points

and time steps.

Figures 5.3 and 5.4 show the output of a diffeomorphism between two face images for $T = 10$ time steps. The paths that each of the N points traverse through time and space are plotted. The points in the first image are on a uniform grid, and this grid is deformed via the clamped-plate spline diffeomorphic interpolation in order to correspond to the points in the second image, resulting in the final deformation presented. We see that while the paths through time of the more horizontal moving features are near linear, the paths of the edges of the eyes and the edges of the mouth have converged to a more rounded trajectory. For very small tolerances and random initial positions, this same minimal configuration is reached, implying that this is likely to be not just a local but the global path of least energy, that is, the optimal geodesic.

Because a diffeomorphism is one-to-one and onto, features that do not exist in one image cannot appear in a diffeomorphism of that image; its existing features can only be stretched and shrunk. Therefore when matching a face with a closed mouth to a face with an open mouth, the lip region will stretch, but nothing which was inside of the mouth can appear. Also, because diffeomorphisms are a mathematical tool and do not know anything about faces, using only a single point on the bottom of the lips to represent the opening of a mouth is not enough information to force the diffeomorphic path to open as a human mouth opens (convexly), and instead it pulls the lip pixels down concavely like a weight on a string, as in Figure 5.4(d). This could be fixed by using an outline of points along all structures that potentially open and close, but this loses the strength of the algorithm based on an extremely sparse

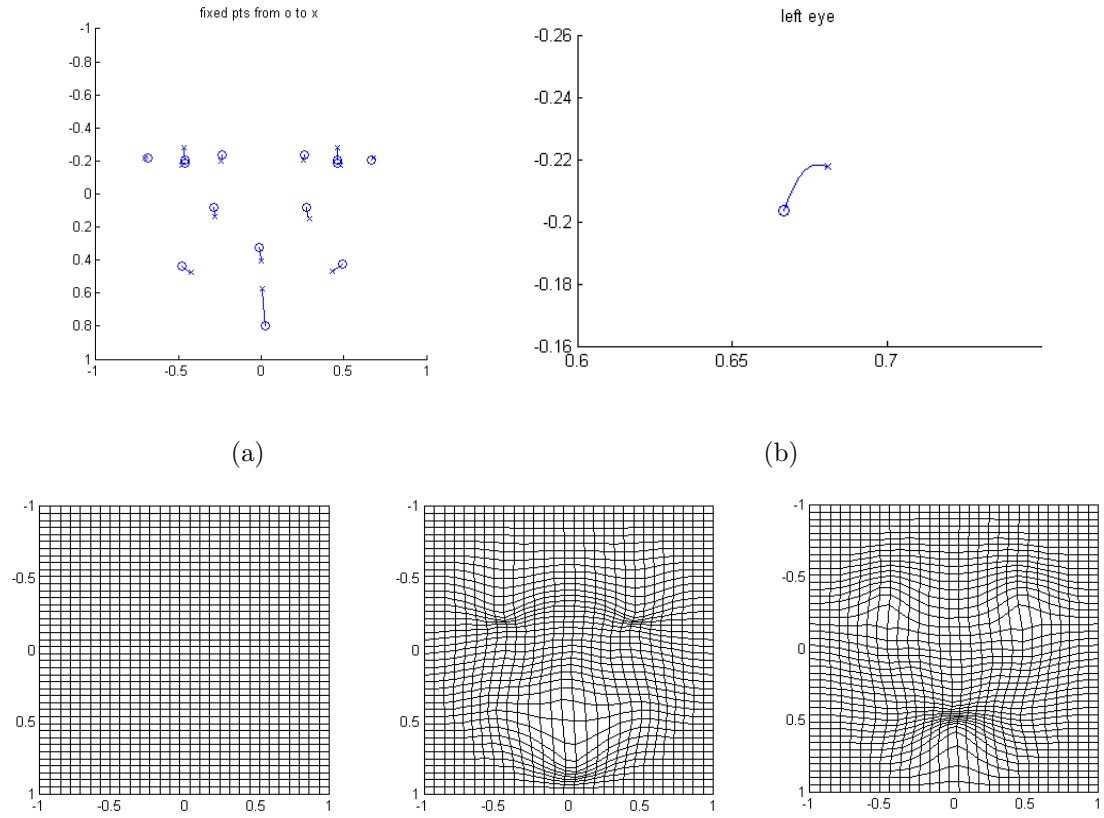
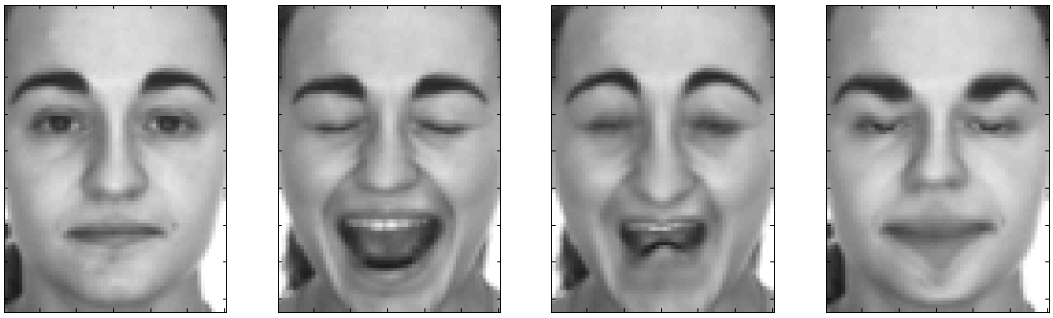


Figure 5.3: The minimal energy diffeomorphism is using 10 time steps. (a) The geodesic path for each of the 14 input fiducial face points. (b) Close-up on the geodesic of the furthest left point on the left eye. (c) The original mesh of points in the first images, (d) the final positions of the points in the diffeomorphism from neutral to scream, (e) the final positions of the points in the diffeomorphism from scream to neutral (note only points within the circle are allowed to move).



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

Figure 5.4: (a) Neutral face, (b) scream face, (c) scream face warped backwards along the diffeomorphism from 5.3(d), (d) neutral face warped backwards along the diffeomorphism from 5.3(e).

set of points. For the studies performed here, we use this unnatural deformation as it occurs mathematically.

Our objective is not to generate realistic images, but is instead to show that the robust mathematical structures of manifolds and diffeomorphisms can be applied in the domain of real images, producing meaningful ways of measuring deformations and image similarity.

We note that the calculated geodesics are symmetric, so the optimal path from image 1 to image 2 is the same optimal path as from image 2 to image 1. Numerically, the gradient descent scheme can reach slightly different values, but we observe that in practice as long as a sufficient number of iterations are computed these numerical errors are negligible.

5.3 Incorporating the Intensity Cost into the Diffeomorphisms

A robust algorithm would calculate diffeomorphisms whose optimal paths are based both on geometric deformations and on relative image intensity values. To the energy function (5.12) we will consider adding a second term as in equation (5.1), the wavelet-based lighting-insensitive intensity cost described in Chapter 4, equation (4.30). This energy cost is already based on geodesic paths, and so for a given pair of intensity values, the intensity values of the entire geodesic path can be found. We now wish to minimize

$$E_{\text{total}}(v) = (1 - \lambda) * E_{\text{diffeo}}(v) + \lambda * E_{\text{Wlgt}}(I(v)), \tag{5.17}$$

where λ is a weighting constant, and $I(v)$ is the set of images generated for a given path v.

The lighting cost function $E_{\text{Wlgt}}(I(v))$ is calculated in wavelet space, and so in order for the calculations to be more efficient, the entire diffeomorphism calculation can be performed in wavelet space. Wavelets are local basis functions, and so the locations of the feature points on a face are still meaningful in wavelet space (resized to match the appropriate wavelet scale). The new term in the cost function will calculate the cost between each successive pair of images along the diffeomorphism through time. The individual geodesic paths of the coupled H and V components are generated from the initial and final intensity values, and so at time t the values can be extracted from these curves. The cost comparing times t and $t + 1$ is the integral along the coupled curves over this time step, however it is faster to use the

pre-generated lookup table to determine these incremental costs.

The problem with using a lookup table is that if we want to use finite differences to calculate the gradient for a gradient descent scheme, values from a lookup table equate to step functions and are not smooth. If a very small step size is used to calculate a finite difference, the same bin in the table will be accessed for both points and the effective gradient will appear to be zero. If a much larger step size is used, the results are unpredictable and not meaningful. Therefore, the lookup table cannot be used for gradient calculations. The original energy function 4.30 could be minimized, but this would require solving a boundary value problem for each pixel pair at each step. Alternately, we can make use of the limiting behavior of the function, as discussed in Section 4.4.3. We know that the photometric energy function behaves similarly to the simple sum $E = \left(\ln \frac{r_1}{r_0}\right)^2 + (\theta_1 - \theta_0)^2$, and as this function is much more efficient to calculate, it can be used in calculations as a good approximation.

One possible way to efficiently optimize (5.17) might be to consider an alternating gradient descent optimization scheme, where first steps are taken in the direction of $-\nabla E_{\text{diffeo}}(v_1)$, then steps are taken in the direction of $-\nabla E_{\text{Wlgt}}(I(v_2))$, then steps are taken towards $v_1 = v_2$, as described in [16]. However, these methods work best when the two separate functions being optimized are truly decoupled, but here the independent variable of E_{Wlgt} is $I(v)$. The full image path I depends on the point geodesics $\vec{x}_i(t)$, which is calculated for all points in space using the relation from equation (5.16). Therefore, the calculations required to compute E_{Wlgt} involve the calculations from the computation of $E_{\text{diffeo}}(v)$, and the two terms cannot be

treated as truly distinct.

In absolute terms, the size of the $E_{\text{Wlgt}}(I(v))$ values are much smaller than those of $E_{\text{diffeo}}(v)$, and so the value of λ should be small. However, if λ is taken so that the two energies are of roughly the same magnitude, then the minimizing path differs very little from the path used when only $E_{\text{diffeo}}(v)$ is minimized. This is not true for larger λ, but for much larger λ the computation starts to lose meaning. With this in mind, we decide that it is reasonable to calculate the optimal diffeomorphic path based entirely on $E_{\text{diffeo}}(v)$, and consider the geodesics from $E_{\text{Wlgt}}(I(v))$ on the output.

5.4 Diffeomorphism Experiments

The speed at which we are currently able to perform gradient calculations in Matlab is prohibitively slow for running our algorithm on large datasets that require tens of thousands of image comparisons. However, if we reduce the number of time steps computed from 10 to 6 and limit the number of gradient descent steps allowed per image pair to 50, we were able to compute the approximate geodesics between the challenging scream case and all neutral images of the AR Face Database. From the optimal paths $x_i(t)$, the full path of each image point can be reconstructed from the neutral to the scream image and from the scream to the neutral image. From the point paths from neutral to scream, the pixels from the scream image can be warped backwards to be put in correspondence with the neutral image, and this new image is compared with the given neutral image. Similarly the pixels from the neutral image

Accuracy from each warping direction	*scream to neutral*	*neutral to scream*
image differencing	76%	78%
E_{Wlgt}	81%	82%

Table 5.1: Identification results on the challenging scream case of the AR Face Database, calculating the geodesic diffeomorphism and warping one image along this path to be put in correspondence with the other for image comparison.

can be warped backwards along the point paths from the scream to neutral; see Figure 5.4. We compare the resulting image pairs using simple image differencing, and using our multi-scaled method from Chapter 4, and the identification results are presented in Table 5.1.

We observe that, not surprisingly, the unnatural deformation of the mathematical diffeomorphisms are not sufficient for handling extreme human expression variation, and these recognition results are comparable to those presented in Chapter 4 for the scream case. We note that as seen previously in this thesis, warping a neutral face to a variant face provides better results than warping the variant face to match the neutral face.

Although we cannot currently calculate the geodesic diffeomorphisms between all image pairs in the AR Face Database (600 variations $\times$ 100 neutral faces $=$ $60,000$ image comparisons), we note that the optimal full image diffeomorphism is never very far from that generated by the input paths that linearly interpolates between then input points through time. These input diffeomorphisms are not the

desired geodesics, but they are still diffeomorphisms with all the desirable properties of diffeomorphisms, namely smoothness and invertibility. We perform the identification task on the full dataset using the paths from our input diffeomorphisms as we used the geodesic diffeomorphisms the experiment comparing the scream and neutral faces above, but with $T = 10$. The results are presented in Table 5.2.

We see that although meaningful identification information is being captured and the results are good, the identification accuracy achieved here based on the diffeomorphisms (but no geodesics) is not stronger than previous methods. However, we point out that the average recognition rate across all expression variations is higher than the method from Chapter 3 before the learning stage was applied, and higher than the method from Chapter 4 before the thresholding was applied. This implies that if further simple data manipulation techniques were applied to this data, that very strong expression-insensitive recognition rates could be achieved.

5.5 Generating Intermediate Images Along Diffeomorphisms

We would like to generate the intermediate images along the geodesic as one face morphs to another, and be able to perform computations with these images. Many algorithms exist from computer graphics to interpolate realistic-looking video sequences between given images, such as the work of Shechtman et al. [72] which uses small pieces of the input images to generate smoothly morphing intermediate images, and that of Mahajan et al. [58] which generates plausible image interpolations by copying pixel gradients along interpolated paths between images, in a framework

Accuracy from each warping direction	*variation to neutral*	*neutral to variation*
image differencing smile	99.0%	98.0%
image differencing frown	99.0%	98.0%
image differencing scream	75.0%	78.0%
image differencing ave. expressions	**91.0%**	**91.3%**
image differencing left lighting	22.0%	22.0%
image differencing right lighting	19.0%	22.0%
image differencing both lightings	2.0%	2.0%
image differencing ave. lightings	**14.3%**	**15.3%**
E_{Wlgt} smile	99.0%	99.0%
E_{Wlgt} frown	99.0%	99.0%
E_{Wlgt} scream	79.0%	82.0%
$\boldsymbol{E_{\mathbf{Wlgt}}}$ **ave. expressions**	**92.3%**	**93.3%**
E_{Wlgt} left lighting	92.0%	92.0%
E_{Wlgt} right lighting	93.0%	94.0%
E_{Wlgt} both lightings	84.0%	81.0%
$\boldsymbol{E_{\mathbf{Wlgt}}}$ **ave. lightings**	**89.7%**	**89.0%**

Table 5.2: Identification results on the full AR Face Database using the input diffeomorphisms, warping one image along this path to be put in correspondence with the other for image comparison.

related to optical flow, and handle occlusions using transition points. Again, our goal is not to create the most visually pleasing images, but instead to create images whose similarity can be meaningfully measured using diffeomorphisms.

Face recognition tasks are regularly performed by comparing an unknown image to every element of a set of known images, and using nearest neighbor matching so the identity of the most similar face is taken to define the identity of the unknown face. Often, multiple images of each person are known in advance, but each image is treated separately. Being able to generate the images along a geodesic path between known images of a person would allow for the comparison of an unknown image to all images in the space between the known images. For example, if we are given the image of a person in both neutral and screaming poses, along the path between them might be an image more similar to that person's smile than either of the given images. Being able to generate intermediate images will make it possible to compare images to the entire convex hull of a local set of known images on the face manifold; see Figure 5.5. Computing convex hulls on arbitrary manifolds is an open problem that we do not claim to be solving, as there are many situations where the region that the convex hull contains is not clear or is the entire surface (for example three points on a sphere that are not contained in a single hemisphere). However, if face images of an individual are assumed to be sufficiently close together on a Riemannian image manifold which is sufficiently smooth, then the geodesics connecting the images are likely to bound a meaningful closed convex region on the manifold.

The diffeomorphism from image 1 to image 2 is defined by the locations of the N feature points at each of T time steps. With this information, it is possible to

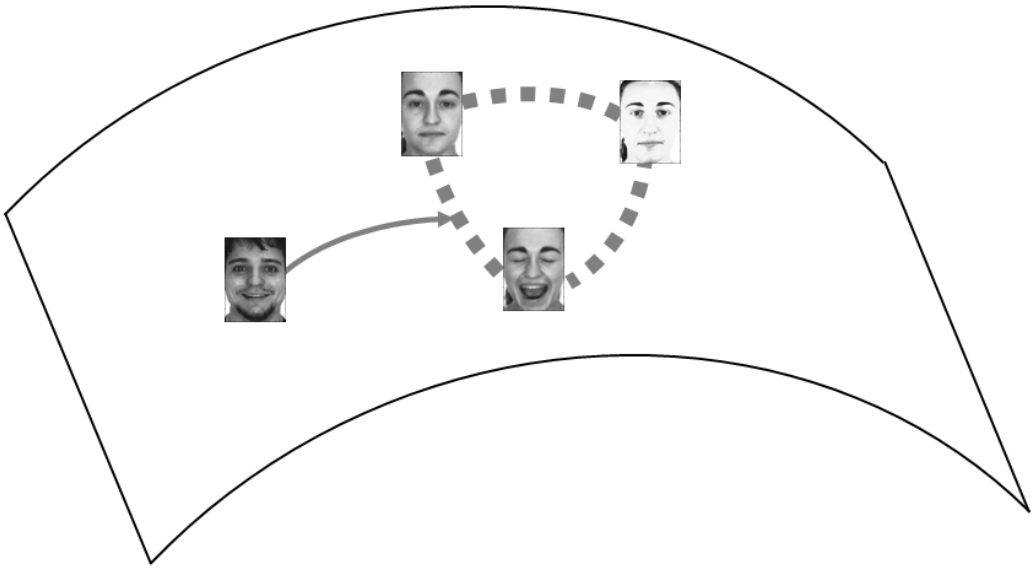


Figure 5.5: The convex hull of a set of known images of an individual, and the geodesic from an unknown face to the known convex set.

generate the diffeomorphic path for each point in an image, as we know each $\vec{x}_i(t)$ and $\alpha_i(t)$, so we can solve for each $\vec{v}(\vec{x}(t))$ via equation (5.16), where the $\vec{x}_j(t)$ are replaced with the general $\vec{x}(t)$. From the $\vec{v}(\vec{x}(t))$ and the initial conditions $\vec{x}(0)$ and $\vec{x}(1)$, we can generate the positions of each point at each time step, $\vec{x}(t)$, via

$$\vec{x}(t) = \vec{x}(t-1) + \vec{v}(t-1), \quad t = 2, ..., T-1. \tag{5.18}$$

The boundary condition $\vec{x}(T) = \vec{x}(T-1) + \vec{v}(T-1)$ is enforced because the variable $\vec{v}$ calculated from equation (5.15) is used in equation (5.16).

Knowing the full path of each point in image 1 tells us the corresponding points in image 2. At each time step, the pixels from image 1 can be positioned at their locations in space at that time, generating new images. However, when regions

in image 2 do not appear in image 1, those regions in the intermediate images have no corresponding pixels. Extrapolation from the known pixels is possible, but it is better instead to use the pixels from image 2 and warp backwards along the flow as done previously in this thesis, for example in 5.4(c) and (d), so that each pixel location is assigned an intensity value. If pixel (i, j) in image 1 corresponds to pixel $(i + v_x, j + v_y)$ in image 2, where (v_x, v_y) are generally not integers, then the corresponding intensity value is determined via bilinear interpolation from the four nearest pixels to the corresponding point location in image 2.

In order to determine the intensity values at each point along the geodesic path between $I_1(i, j)$ and $I_2(i + v_x, j + v_y)$, we use the intensity values along the geodesics generated by the lighting-insensitive metric from Chapter 4. The calculated geodesics are defined by the horizontal and vertical components of the 2D wavelet transform, and so the method from Chapter 4 provides the H and V values at any point along the geodesic. The diagonal components and the approximate image were not used in the calculation, so we must determine their intermediate values another way. The most obvious way to do this is to simply linearly interpolate the known values at each end. Linearly interpolating the approximate image somewhat defeats the purpose of having an algorithm that can explicitly handle deformations, but since we are using the first three scales of wavelets, the final approximate image is 6×4 or 11×8 for input images of size 42×30 and 83×60 respectively. These images are so small that most details are lost, and so using linear interpolations of these images to generate the intermediate images is not unreasonable. In order to generate the intermediate images, we generate a new lookup table that contains the

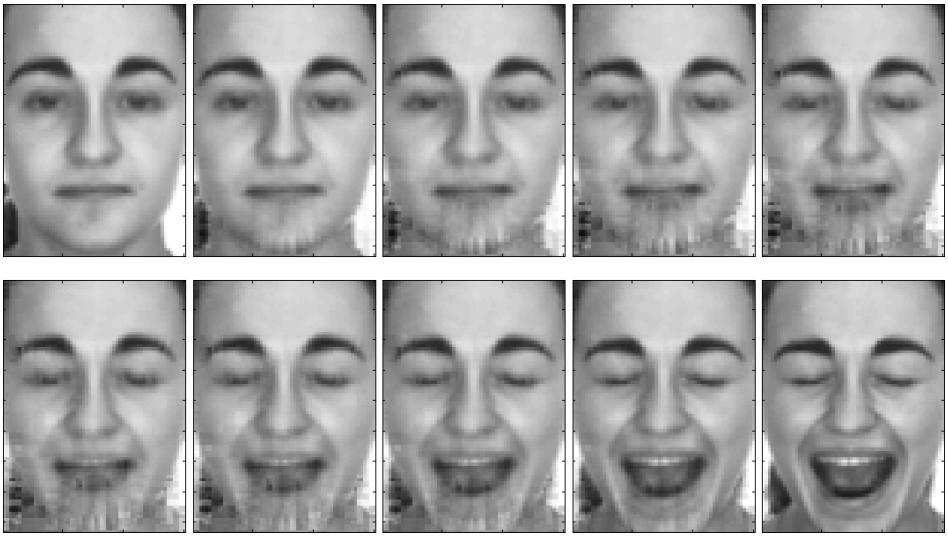


Figure 5.6: Intermediate images for 10 time steps calculated using the horizontal and vertical wavelet coefficients from Chapter 4, with the diagonal and approximate coefficients interpolated linearly from the input first and last images.

values of the horizontal and vertical components at each of $T = 10$ time steps along each geodesic. Intermediate images along the geodesic path are generated in this manner, reconstructing the images from the wavelet coefficients at the first three scales, and the resulting images are show in Figure 5.6.

We see that in the regions of the images that undergo significant deformation, namely around the mouth, these linearly interpolated approximate and diagonal coefficients are not sufficient to recreate smooth images. The artifacts seen are not a property of the geodesic values being taken from a lookup table, as the same images were observed when the true geodesic paths are calculated at each point, a significantly slower calculation. The wavelet horizontal and vertical geodesic paths are quite nonlinear when the starting and ending values are very different, most

Figure 5.7: Intermediate images for 10 time steps calculated using the horizontal and vertical wavelet coefficients from Chapter 4, with the diagonal and approximate coefficients interpolated linearly from the input first and last images when the corresponding points have $\Delta\theta < \frac{\pi}{2}$, and where the intensity values are linearly interpolated for the points with $\Delta\theta > \frac{\pi}{2}$.

significantly when the θ values in equation (4.25) are very different. These are the correct meaningful geodesic paths that measure the shortest distance between the two input images, but unfortunately as they do not depend on the diagonal and approximate coefficients, they do not provide the value of these coefficients. As a proof of concept, if the corresponding points with $\Delta\theta$ larger than $\frac{\pi}{2}$ are linearly interpolated, while the true intensity geodesics are used for the other point pairs, the image sequence in Figure 5.7 is achieved, showing that it is these large $\Delta\theta$ cases that are causing the lack of smoothness between the coefficients in the reconstructed image.

As using the wavelet-based intensity geodesics does not create visually pleasing intermediate images, we will instead interpolate the intensity values linearly along the image path from image 2 to image 1. In order to generate intermediate images that use pixel values from both images, we calculate both the diffeomorphism from image 1 to image 2 and its intermediate images $I_2(t)$ based on the pixels from image 2, and also the diffeomorphism from image 2 to image 1 and its intermediate images $I_1(t)$ based on the pixels from image 1. We note that because the diffeomorphism based on the sparse landmark points is invertible, the diffeomorphic path does not need to be recalculated in each direction, only the locations of the discrete pixels from each image after T steps need to be solved separately for each of the two images. The final intermediate images are determined by a weighted average of these two sets of images, where the weights are determined linearly according to t:

$$I(t) = \frac{T-t}{T-1} I_1(t) + \frac{t-1}{T-1} I_2(t), \quad t = 1, ..., T. \tag{5.19}$$

The output of the linearly interpolated intermediate images is seen in Figure 5.8. The diffeomorphism is calculated only within the highlighted ellipse, as described above, so points outside this ellipse are purely linearly interpolated between the two images and are meaningless for this study. These intermediate images are seen to be reasonably realistic, with the eyes closing and the mouth opening through the sequence of images.

We can now use these intermediate images in the recognition task. Using the expression subset of the AR Face Database [61] as before, we take the extreme

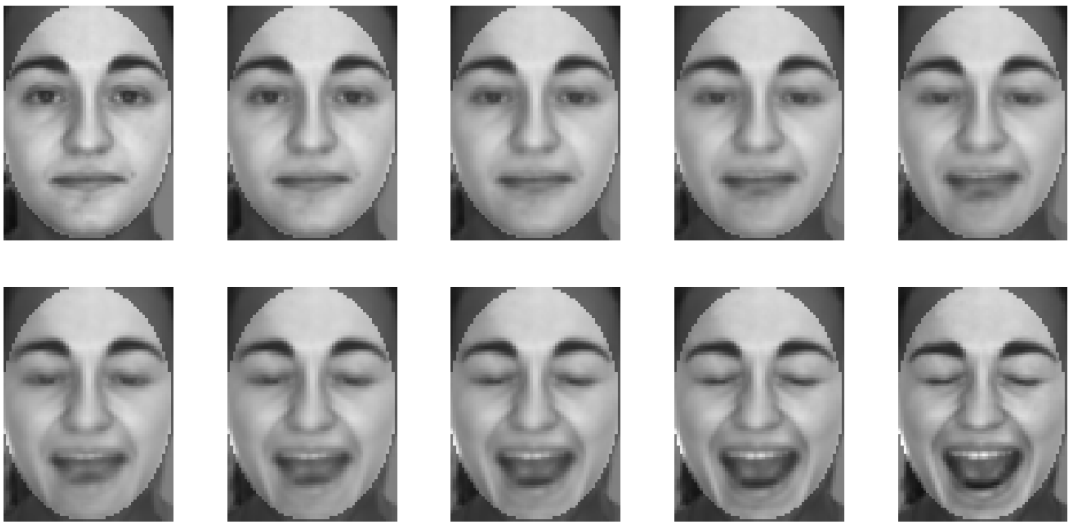


Figure 5.8: Intermediate images for 10 time steps. The first and last images are input, the rest are generated by the proposed algorithm based on linear interpolation. The diffeomorphism is calculated only within the highlighted ellipse.

images of scream and neutral for each individual and generate the intermediate images along the geodesic path connecting them, so that we now have T images of each individual. We then use all the generated images as the gallery images, and compare the smile and frown probes to the entire gallery set, assigning identity from nearest neighbor matching. We compare images using our wavelet-based lighting-insensitive metric from Chapter 4. As a baseline we compare to the case when the only images in the gallery are the original neutral and scream examples. The results of this experiment are presented in Table 5.3.

We see that the identification accuracy for the smile case has improved as compared to the case where only the neutral and scream images are known, but for the frown case has remained the same. However, the results on this subset of the

Accuracy for each probe image	*Smile*	*Frown*
E_{Wlgt} with gallery of neutral and scream	99%	98%
E_{Wlgt} with gallery of intermediate images	100%	98%

Table 5.3: Identification accuracy using nearest neighbor matching on the expression variation subset of the AR Face Database, where the neutral and scream faces are known and the gallery of each person consists of all 10 intermediate images generated by the proposed algorithm.

AR Face Database were already essentially saturated, and so it is not possible to say how much our method actually added to the recognition accuracy. Regardless, it is reasonable to interpret the results by considering that the movement of the face from neutral to scream raises the outer lips, passing through a position more similar to smile than either the neutral or the scream cases. However, in a frown the outer lips move downward, in the opposite direction of the scream. The fact that the results improved for the smile case, when relevant data is being generated, but not for the frown, implies that the observed improvement is a result not of simply having more data, but of having meaningful data, as desired. Unfortunately the expression subset of the AR Face Database is not a good dataset to use for this experiment, as the only challenging case is the scream image, and as data from this image must be used to generate the intermediate images it cannot be used as a probe image in the experiments.

We therefore consider the Cohn-Kanade AU-Coded Facial Expression Database

Figure 5.9: One image sequence from the Cohn-Kanade AU-Coded Facial Expression Database.

[56], which is a dataset very commonly used for testing expression recognition algorithms. The dataset consists of video sequences of people demonstrating extreme facial expressions, moving from neutral to the peak of the expression in an average of 28 frames; for example see Figure 5.9. Facial feature points are also known. This dataset was generated to be used for recognizing expressions, not identities, and as such it is too easy for the identification task. Using simple image differencing with the neutral and extreme images of each expression sequence in the entire database (123 people, 323 sequences) as the gallery, and the intermediate images as the probes, the identification accuracy is already 99.9%. This dataset will be used instead to compare our generated images with the true intermediate images provided, to demonstrate that the intermediate images we generate are meaningful, and useful for image comparison.

We take the neutral and extreme frame of a sequence, and generate images between them, then compare all the true intermediate images to the generated images; see Figure 5.10. There is no expectation that the images at time t in the

Figure 5.10: Generated intermediate images of one sequence from the Cohn-Kanade AU-Coded Facial Expression Database, where only the first and last image of the sequence are provided.

sequence correspond, as humans do not move from one expression to another linearly through time, but our generated images are meaningful if a true intermediate image corresponds more closely to one of the generated intermediate images than it does to the provided extremes. If this is true, then we are able to overcome the often observed challenge that occurs when an automatic face recognition system declares that a face of one expression is more similar to a different face of that same expression than to a face of the same person showing a different expression.

Intermediate images were generated for all 323 expression sequences of the Cohn-Kanade AU-Coded Facial Expression Database, with $T = 10$ images in each sequence, and results are presented in Table 5.4. We see that in the vast majority of cases, the true intermediate images did match more closely to our generated images than they did to the known neutral and extreme expressions.

We have shown that given a sparse set of images, we are able to generate a large number of meaningful intermediate images to make a dataset robust to

	Percentage matching our generated images
Mean	77.5%
Median	80.0%

Table 5.4: The percentage of each expression image sequence from the Cohn-Kanade AU-Coded Facial Expression Database that matched more closely to our generated intermediate images than they did to the true extreme images of their respective sequences.

common errors. In other words, we are able to generate images along the boundary of a convex hull of known images, as long as they are sufficiently close together on the manifold that this is meaningful. This allows an unknown image to be compared not only to known images, but to the set of convex combinations of known images. This is a very powerful idea and we foresee many potentially useful domain-specific extensions of these methods.

5.6 Conclusion

We have shown how smooth, invertible deformations can be applied to face images by using the framework of diffeomorphisms. The diffeomorphic path between two images was defined by a sparse set of corresponding feature points on each face, and geodesics between these feature points were calculated by minimizing an appropriate cost function. These paths were interpolated so that the path through time

of every point in each image was defined, resulting in a full image diffeomorphism between the two images. Neutral faces were deformed based on the diffeomorphic paths, to be put in correspondence with expression and lighting variant faces, and face recognition tasks were performed by comparing these images, with promising results. Intermediate images along the diffeomorphic paths were generated by interpolating pixel values along the paths. Known intermediate images from facial expression video sequences were compared to the generated intermediate images, and the true intermediate images were seen to match more closely to the generated intermediate images than to the extremes of the video sequences, which were the only input into the diffeomorphism algorithm. Being able to generate intermediate images between all known images of an individual provides a way for an unknown image to be compared to the full convex hull of known images of an individual on a manifold.

The result shown here are preliminary, demonstrating the potential utility of some of the possible paths that become available when full diffeomorphisms are generated between pairs of images.

Chapter 6

Conclusion and Future Work

We have constructed a deformation- and lighting-insensitive metric that assigns a cost to a pair of images based on their similarity. The metric is based on the effect of lighting in 3D scenes, comparing image gradients in a new way. In order to explicitly model image deformations, establishing point correspondences between images is essential, and this thesis presented several algorithms for determining dense point correspondences between pairs of images across changes in shape and illumination, assigning a cost to each of these pairings. The methods are inspired by the idea that geodesics and diffeomorphisms on Riemannian image manifolds can provide a robust and elegant way to model changes in shape and lighting. The methods of this thesis were applied to face recognition, but nothing about our work is specific to this domain, and the methods can be applied in any situation where an object with some amount of structure has been deformed.

We proposed a method for finding correspondences between images based on our new metric, using smooth Sobolev gradients to efficiently optimize over a correspondence vector field that determined dense correspondences between potentially deformed images taken under very different conditions. Typical correspondence cost patterns from our metric were learned between face image pairs, and a Naïve Bayes classifier was applied to improve recognition accuracy.

The new local metric was extended in a fast algorithm for calculating geodesic distances between pairs of images on an image manifold with significant illumination variation. The metric was calculated in the wavelet domain, where each point location contributed independently to the overall image comparison cost, allowing geodesic costs to be computed extremely efficiently by referencing a pre-calculated lookup table. Using wavelets at multiple scales allowed for insensitivity to moderate deformations. The speed of this algorithm allowed it to be useful in many real-world scenarios.

We then showed how smooth, invertible deformations can be modeled using the framework of diffeomorphisms. The full diffeomorphic path between two images was constructed from the paths between a sparse set of corresponding feature points in each image, applying spline interpolation with an appropriate set of basis functions to define the paths for all other image points. Faces were deformed based on these diffeomorphic paths to be put in correspondence with other expression-variant faces. Intermediate images along the diffeomorphic path were generated by interpolating pixel values, producing images similar to real intermediate face images when they are known.

Strong results were presented on the expression and lighting variant subset of the AR Face Database for all algorithms presented in this thesis. Instead of simply comparing pixels in two images, using geodesics and diffeomorphisms to calculate image similarities can be incorporated into a wide array of useful applications where having information along a morphing path between two images is relevant. This framework allows large image changes to be introduced gradually and handled ex-

plicitly in a well-defined fashion, and can be applied to calculate image similarities across large datasets. We discuss some of these extensions below.

6.1 Future Directions

The work presented in this thesis can lead to many further studies. The most important next step towards making image diffeomorphisms useful for practical applications is to explicitly and robustly handle occlusions. There are several ways that this could be attempted, including the addition of robust statistical tools such as M-estimators, and redefining the diffeomorphisms so that one-to-many and many-to-one matches are allowed with a certain penalty. Combining the resulting method with an algorithm that explicitly handles changes in pose, such as [21], should then provide a very robust general face recognition system. Domains other than face recognition should be explored with these methods, such as medical imaging and fine-grained visual categorization including animal and plant sub-species identification. The diffeomorphic framework explored here provides an elegant way to handle any type of image variation by allowing the change to be introduced gradually in a well-defined manner.

Optimization schemes can always be improved. For example in Chapter 3, the optimization iterations move towards the optimal solution, but even using Sobolev gradient they terminate at a local minimum long before the true optimal correspondences are reached. Interestingly, this provides enough information for the machine learning algorithm to successfully discriminate which locally optimal vector fields

correspond to same-person image pairs vs different person image pairs. However, a hierarchical method would likely help the iterations progress significantly further before settling on a minima. A stronger optimization method would also benefit the diffeomorphism calculations from Chapter 5, where making use of a full C language implementation of automatic differentiation could help calculate numerically correct diffeomorphism gradients in the same order of time as the function calculation, allowing true geodesic diffeomorphisms to be calculated efficiently.

The results of all the methods presented in this thesis could be improved by applying machine learning methods to the data that they produce. One application was seen in Chapter 3 when Naive Bayes was applied via simple Gaussians fit through the cost data at each pixel, which removed 48% of the errors in the resulting recognition rates. Support Vector Machines [85] with an appropriate kernel, or other statistical regression analysis techniques, should be able to help effectively separate same-person image pair data from different-person image pair data. Machine learning methods can learn the ways in which faces deform naturally, thereby recognizing when a deformation is likely to be between two images of the same person, as compared to an unnatural deformation which would be assumed to come from two different people.

The diffeomorphisms from Chapter 5 could be made more realistic to faces if they are not based on such sparse point correspondences. Perhaps it would help to use basis functions that were based not on points but on curves, so that a convex curve outlining the mouth could be defined. A diffeomorphism scheme not based on sparse points would also help solve this problem, such as methods similar to [2, 7].

It would also be very interesting to study not only interpolation between images, but also extrapolation. Given a set of sparse facial feature locations, a diffeomorphism can be found deforming an entire image to those locations, even if there is no image to be matched with at that location. Images can also be deformed further in the direction of the final known diffeomorphism. There are many potential applications of these ideas to face recognition, and to other domains where explicit deformation and illumination change modeling is required.

The strong results achieved by simple thresholding, as seen in Chapter 4, should be further studied, to determine where the Local Binary Decisions method breaks down, and to easily incorporate an unbiased thresholding step to all methods as appropriate. The Gradient Direction method could probably be made stronger by using a wider gradient filter and by making it multi-scaled. This would probably result in a very strong comparison metric that is robust to moderate amounts of deformation.

A non-isotropic version of the lighting-insensitive metric from equation (3.5) can be defined and explored, where there is a lower cost for intensity changes in the direction of the gradient.

There are also several implementation decisions that should also be further explored, including parameter selections such as the amount of smoothing applied to each image before being processed, the size of the kernels, the discretization used in the lookup table, and the size of the image crops being compared. In order to say that the wavelet version of the cost function presented in Chapter 4 in equation (4.23) is equivalent to the original function (4.10), the original function

could be presented as the sum of every other pixel. The first scale of the Haar wavelet basis functions have a support of width two, and so this would make the wavelet version exact when Haar wavelets are used and only one scale is considered. As the preliminary study presented in Section 4.6.2 implies, the extremely fast and accurate method of Chapter 4 should be extended for applications where fast template matching schemes such as Normalized Cross-Correlation are often used, where many image comparisons must be computed in a very short amount of time.

Bibliography

[1] P.-A. Absil, R. Mahony, and R. Sepulchre. Riemannian Geometry of Grassmann Manifolds with a View on Algorithmic Computation. *Acta Applicandae Mathematicae*, 80:199–220, 2004.

[2] J. Ashburner. A Fast Diffeomorphic Image Registration Algorithm. *NeuroImage*, 38:95–113, 2007.

[3] S. Baker and I. Matthews. Lucas-Kanade 20 Years On: A Unifying Framework. *IJCV*, 56:221–255, 2004.

[4] J. Barron, D. Fleet, and S. Beauchemin. Performance of Optical Flow Techniques. *IJCV*, 12:43–77, 1994.

[5] R. Basri and D. Jacobs. Lambertian Reflectance and Linear Subspaces. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 25:218 – 233, 2003.

[6] H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool. SURF: Speeded Up Robust Features. *Computer Vision and Image Understanding (CVIU)*, 110:346–359, 2008.

[7] M. Beg, M. Miller, A. Trouvé, and L. Younes. Computing Large Deformation Metric Mappings via Geodesic Flows of Diffeomorphisms. *IJCV*, 61:139–157, 2005.

[8] T. Beier and S. Neely. Feature-Based Image Metamorphosis. *ACM SIGGRAPH Computer Graphics*, 26:35 – 42, 1992.

[9] P. Belhumeur, J. Hespanha, and D. Kriegman. Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19:711–720, 1997.

[10] P. N. Belhumeur, D. W. Jacobs, D. J. Kriegman, and N. Kumar. Localizing Parts of Faces Using a Consensus of Exemplars. *CVPR*, pages 545–552, 2011.

[11] M. Belkin and P. Niyogi. Laplacian Eigenmaps and Spectral Techniques for Embedding and Clustering. *Advances in Neural Information Processing Systems*, 14:586691, 2001.

[12] D. Beymer and T. Poggio. Face Recognition From One Example View. *IEEE Conf. on Computer Vision*, page 500, 1995.

[13] M. J. Black and P. Anandan. The Robust Estimation of Multiple Motions: Parametric and Piecewise-Smooth Flow Fields. *Computer Vision and Image Understanding*, 63:75–104, 1996.

[14] V. Blanz and T. Vetter. Face Recognition Based on Fitting a 3D Morphable Model. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25:1063–1074, 2003.

[15] F. Bookstein. Principal Warps: Thin-Plate Splines and the Decomposition of Deformations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11:567585, 1989.

[16] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed Optimization and Statistical Learning Via the Alternating Direction Method of Multipliers. *Foundations and Trends in Machine Learning*, 3:1122, 2011.

[17] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert. High Accuracy Optical Flow Estimation Based on a Theory for Time Warping. *European Conference on Computer Vision*, 4:25–36, 2004.

[18] T. Brox and J. Malik. Large Displacement Optical Flow: Descriptor Matching in Variational Motion Estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 3:500–513, 2011.

[19] J. Bruna and S. Mallat. Classification with Scattering Operators. *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2011.

[20] V. Camion and L. Younes. Geodesic Interpolating Splines. *Energy Minimization Methods for Computer Vision and Pattern Recognition*, pages 513–527, 2001.

[21] C. D. Castillo and D. W. Jacobs. Using Stereo Matching with General Epipolar Geometry for 2-D Face Recognition Across Pose. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31:2298 – 2304, 2009.

[22] H. Chen, P. Belhumeur, and D. Jacobs. In Search of Illumination Invariants. *Computer Vision and Pattern Recognition*, pages 254–261, 2000.

[23] T. Chen, W. Yin, X. S. Zhou, D. Comaniciu, and T. Huang. Total Variation Models for Variable Lighting Face Recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 28:1519 –1524, 2006.

[24] T. Cootes, G. Edwards, and C. Taylor. Active Appearance Models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23:681 – 685, 2001.

[25] O. Corporation. `http://www.omron.com/r_d/coretech/vision/`.

[26] R. Courant and D. Hilbert. *Methods of Mathematical Physics, volume I*, chapter IV. Interscience Publishers, Inc., 1955.

[27] L. Ding and A. Martinez. Features Versus Context: An Approach for Precise and Detailed Detection and Delineation of Faces and Facial Features. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32:2022–2038, 2010.

[28] M. do Carmo. *Differential Geometry of Curves and Surfaces*. Prentice-Hall, Inc., 1976.

[29] M. do Carmo. *Riemannian Geometry*. Birkhuser, 1992.

[30] D. Donoho and C. Grimes. Hessian Eigenmaps: Locally Linear Embedding Techniques for High-Dimensional Data. *Proceedings of the National Academy of Sciences of the United States of America*, 100:55915596, 2003.

[31] P. Felzenszwalb and D. Huttenlocher. Pictorial Structures for Object Recognition. *International Journal of Computer Vision*, 61:55 – 79, 2005.

[32] T. Gass, L. Pishchulin, P. Dreuw, and H. Ney. Warp that Smile on your Face: Optimal and Smooth Deformations for Face Recognition. *IEEE International Conference on Automatic Face & Gesture Recognition and Workshops*, pages 456 – 463, 2011.

[33] A. Georghiades, P. Belhumeur, and D. Kriegman. From Few to Many: Illumination Cone Models for Face Recognition Under Variable Lighting and Pose. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 23:643 –660, 2001.

[34] B. Glocker, N. Paragios, N. Komodakis, G. Tziritas, and N. Navab. Optical Flow Estimation With Uncertainties Through Dynamic MRFs. *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8, 2008.

[35] R. Gopalan and D. Jacobs. Comparing and Combining Lighting Insensitive Approaches for Face Recognition. *CVIU*, 114:135–145, 2010.

[36] K. Grauman and T. Darrell. Fast Contour Matching Using Approximate Earth Movers Distance. *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2004.

[37] G. D. Hager and P. N. Belhumeur. Efficient Region Tracking With Parametric Models of Geometry and Illumination. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20:1025–1039, 1998.

[38] C. Harris and M. Stephens. A Combined Corner and Edge Detector. *Proceedings of the 4th Alvey Vision Conference*, page 147151, 1988.

[39] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision (2nd ed.)*. Cambridge University Press, 2003.

[40] H. Haussecker and D. Fleet. Computing Optical Flow With Physical Models of Brightness Variation. *Pattern Analysis and Machine Intelligence*, pages 661 – 673, 2011.

[41] B. Horn and B. Schunck. Determining Optical Flow. *Artificial Intelligence*, 17:185–203, 1981.

[42] C. Hseih, S. Lai, and Y. Chen. Expression-Insensitive Face Recognition with Accurate Optical Flow. *Proc. of the multimedia 8th Pacific Rim conf. on Advances in multimedia information processing*, pages 78–87, 2007.

[43] A. P. James. Pixel-Level Decisions Based Robust Face Image Recognition. In M. Oravec, editor, *Face Recognition*, chapter 5, pages 65–86. INTECH, 2010.

[44] A. Jorstad, D. Jacobs, and A. Trouvé. A Deformation and Lighting Insensitive Metric for Face Recognition Based on Dense Correspondences. *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2011.

[45] A. Jorstad, D. Jacobs, and A. Trouvé. A Fast Illumination and Deformation Insensitive Image Comparison Algorithm Using Wavelet-Based Geodesics. *Proceedings of the European Conference on Computer Vision*, 2012.

[46] Y.-H. Kim, A. M. Martinez, and A. C. Kak. Robust Motion Estimation Under Varying Illumination. *Image and Vision Computing*, 23, 2005.

[47] A. Klein, J. Andersson, B. Ardekani, J. Ashburner, B. Avants, M. Chiang, G. Christensen, D. Collins, J. Gee, P. Hellier, J. Song, M. Jenkinson, C. Lepage, D. Rueckert, P. Thompson, T. Vercauteren, R. Woods, J. Mann, and R. Parsey. Evaluation of 14 Nonlinear Deformation Algorithms Applied to Human Brain MRI Registration. *NeuroImage*, 46, 2009.

[48] N. Kumar, A. Berg, P. Belhumeur, and S. Nayar. Attribute and Simile Classifiers for Face Verification. *Computer Vision, 2009 IEEE 12th International Conference on*, pages 365–372, 2009.

[49] M. Lades, J. C. Vorbrggen, J. Buhmann, J. Lange, C. V. D. Malsburg, R. P. Wrtz, and W. Konen. Distortion Invariant Object Recognition in the Dynamic Link Architecture. *IEEE Trans. Computers*, 42:300–311, 1993.

[50] Y. LeCun, F. Huang, and L. Bottou. Learning Methods for Generic Object Recognition with Invariance to Pose and Lighting. *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2004.

[51] R. J. LeVeque. *Finite Difference Methods for Ordinary and Partial Differential Equations, Steady State and Time Dependent Problems.* SIAM, 2007.

[52] J. Lewis. Fast Normalized Cross-Correlation. *Vision Interface*, 1995.

[53] L. Liang, R. Xiao, F. Wen, and J. Sun. Face Alignment Via Component-Based Discriminative Search. *European Conference on Computer Vision (ECCV)*, 2008.

[54] C. Liu, J. Yuen, and A. Torralba. SIFT Flow: Dense Correspondence across Scenes and Its Applications. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33:978 –994, 2011.

[55] D. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, 60, 2004.

[56] P. Lucey, J. F. Cohn, T. Kanade, J. Saragih, Z. Ambadar, and I. Matthews. The Extended Cohn-Kande Dataset (CK+): A Complete Facial Expression Dataset for Action Unit and Emotion-Specified Expression. *Third IEEE Workshop on CVPR for Human Communicative Behavior Analysis*, 2010.

[57] W.-C. Ma, A. Jones, J.-Y. Chiang, T. Hawkins, S. Frederiksen, P. Peers, M. Vukovic, M. Ouhyoung, and P. Debevec. Facial Performance Synthesis Using Deformation-Driven Polynomial Displacement Maps. *ACM Trans. Graph.*, 27:121:1–121:10, 2008.

[58] D. Mahajan, F.-C. Huang, W. Matusik, R. Ramamoorthi, and P. Belhumeur. Moving Gradients: A Path-Based Method for Plausible Image Interpolation. *ACM Trans. Graph.*, 28:42:1–42:11, 2009.

[59] S. Mallat. *A Wavelet Tour of Signal Processing*. Elsevier, 2009.

[60] A. Martinez. Recognizing Expression Variant Faces From a Single Sample Image per Class. *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 1:353–358, 2003.

[61] A. Martinez and R. Benavente. The AR Face Database. *CVC Technical Report #24*, 1998.

[62] A. Martinez and A. C. Kak. PCA versus LDA. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23:228 – 233, 2001.

[63] M. I. Miller, A. Trouv, and L. Younes. Geodesic Shooting for Computational Anatomy. *Journal of Mathematical Imaging and Vision*, 24:209–228, 2006.

[64] S. Negahdaripour. Revised Definition of Optical Flow: Integration of Radiometric and Geometric Cues for Dynamic Scene Analysis. *PAMI*, 20:961–979, 1998.

[65] R. D. Neidinger. Introduction to Automatic Differentiation and MATLAB Object-Oriented Programming. *SIAM Review*, 52:545563, 2010.

[66] J. W. Neuberger. *Sobolev Gradients and Differential Equations, 2nd Edition*. Springer, 2010.

[67] R. Ng, R. Ramamoorthi, and P. Hanrahan. All-Frequency Shadows Using Nonlinear Wavelet Lighting Approximation. *SIGGRAPH*, 22:376 – 381, 2003.

[68] J. Nocedal and S. Wright. *Numerical Optimization*. Springer, 1999.

[69] N. Papenberg, A. Bruhn, T. Brox, S. Didas, , and J. Weickert. Highly Accurate Optic Flow Computation with Theoretically Justified Warping. *International Journal of Computer Vision*, 67:141158, 2006.

[70] S. Roweis and L. Saul. Nonlinear Dimensionality Reduction by Locally Linear Embedding. *Science*, 290:pp.2323–2326, 2000.

[71] Y. Rubner, C. Tomasi, and L. Guibas. The Earth Mover's Distance as a Metric for Image Retrieval. *International Journal on Computer Vision*, 40:99 – 121, 2000.

[72] E. Shechtman, A. Rav-Acha, M. Irani, and S. Seitz. Regenerative Morphing. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, San-Francisco, CA, June 2010.

[73] S. Shirdhonkar and D. Jacobs. Approximate Earth Movers Distance in Linear Time. *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

[74] J. Song, B. Chen, W. Wang, and X. Ren. Face Recognition by Fusing Binary Edge Feature and Second-Order Mutual Information. In *IEEE Conf. on Cybernetics and Intelligent Systems*, pages 1046–1050, 2008.

[75] D. Sun and S. Roth. `www.cs.brown.edu/~dqsun/code/ba.zip`, 2008. Implementation.

[76] X. Tan and B. Triggs. Enhanced Local Texture Feature Sets for Face Recognition Under Difficult Lighting Conditions. *IEEE Transactions on Image Processing*, pages 1635–1650, 2010.

[77] J. B. Tenenbaum, V. de Silva, and J. C. Langford. A Global Geometric Framework for Nonlinear Dimensionality Reduction. *Science*, page 23192323, 2000.

[78] B. Triggs. Detecting Keypoints with Stable Position, Orientation, and Scale under Illumination Changes. *European Conference on Computer Vision*, 2004.

[79] A. Trouvé and L. Younes. Metamorphoses Through Lie Group Action. *Foundations of Computational Mathematics*, pages 173–198, 2005.

[80] T. Tung and T. Matsuyama. Dynamic Surface Matching by Geodesic Mapping for 3D Animation Transfer. *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2010.

[81] P. Turaga, A. Veeraraghavan, and R. Chellappa. Statistical Analysis on Stiefel and Grassmann Manifolds with Applications in Computer Vision. *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

[82] M. Turk and A. Pentland. Face Recognition using Eigenfaces. *Proc. IEEE Converence on Computer Vision and Pattern Recognition*, 1991.

[83] C. Twining, S. Marsland, and C. Taylor. Measuring Geodesic Distances on the Space of Bounded Diffeomorphisms. In *In BMVC*, pages 847–856. BMVA Press, 2002.

[84] G. Tzimiropoulos, S. Zafeiriou, and M. Pantic. Robust and Efficient Parametric Face Alignment. *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2011.

[85] V. N. Vapnik. *The Nature of Statistical Learning Theory*. Springer-Verlag New York, 1995.

[86] H. Wang, S. Li, and Y. Wang. Face Recognition Under Varying Lighting Conditions Using Self Quotient Image. *Automatic Face and Gesture Recognition, 2004. Proceedings. Sixth IEEE International Conference on*, pages 819–824, 2004.

[87] A. Wedel, T. Pock, C. Zach, H. Bischof, and D. Cremers. An Improved Algorithm for TV-L1 Optical Flow. In *Statistical and Geometrical Approaches to Visual Motion Analysis*, pages 23–45. Lecture Notes in Computer Science, 2009.

[88] B. Wirth, L. Bar, M. Rumpf, and G. Sapiro. Geodesics in Shape Space via Variational Time Discretization. *Energy Minimization Methods in Computer Vision and Pattern Recognition*, 5681:288–302, 2009.

[89] J. Wright, A. Yang, A. Ganesh, S. Sastry, and Y. Ma. Robust Face Recognition via Sparse Representation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 31:210–227, 2009.

[90] X. Xie and K.-M. Lam. Elastic Shape-Texture Matching for Human Face Recognition. *Pattern Recognition*, 41:396–405, 2008.

[91] X. Xie and K.-M. Lam. Face Recognition Using Elastic Local Reconstruction Based on a Single Face Image. *Pattern Recognition*, 41:406–417, 2008.

[92] Q. Yin, X. Tang, and J. Sun. An Associate-Predict Model for Face Recognition. *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 497–504, 2011.

[93] L. Younes. *Shapes and Diffeomorphisms*. Springer, 2010.

[94] L. G. L. Younes. Geodesic Image Matching: A Wavelet Based Energy Minimization Scheme. *Energy Minimization Methods in Computer Vision and Pattern Recognition*, 2005.

[95] L. Zhang and D. Samaras. Face Recognition From a Single Training Image Under Arbitrary Unknown Lighting Using Spherical Harmonics. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 28:351–363, 2006.

[96] T. Zhang, B. Fang, Y. Yuan, Y. Y. Tang, Z. Shang, D. Li, and F. Lang. Multiscale Facial Structure Representation for Face Recognition Under Varying Illumination. *Pattern Recognition*, 42:251258, 2009.

[97] S. Zhao and Y. Gao. Significant Jet Point For Facial Image Representation and Recognition. *International Conference on Image Processing*, pages 1664–1667, 2008.

[98] H. Zimmer, A. Bruhn, and J. Weickert. Optic Flow in Harmony. *International Journal of Computer Vision*, pages 368–388, 2011.