Deformable Surface Recovery and Its Applications

ZHU, Jianke

A Thesis Submitted in Partial Fulfilment
of the Requirements for the Degree of
Doctor of Philosophy
in
Computer Science and Engineering

December 2008

Abstract of thesis entitled:
Deformable Surface Recovery and Its Applications
Submitted by ZHU, Jianke
for the degree of Doctor of Philosophy
at The Chinese University of Hong Kong in December 2008

Recovering deformable surfaces is an interesting and beneficial research problem for computer vision and image analysis. An effective deformable surface recovery technique can be applied in a variety of applications for surface reconstruction, digital entertainment, medical imaging and Augmented Reality. While considerable research efforts have been devoted to deformable surface modeling and fitting, there are only few schemes available to tackle the deformable surface recovery problem efficiently. This thesis proposes a set of methods to effectively solve the 2D nonrigid shape recovery and 3D deformable surface tracking based on a robust progressive optimization scheme. The presented techniques are also applied to a variety of real-world applications.

To tackle the 2D nonrigid shape recovery problem, this thesis first presents a novel progressive finite Newton optimization scheme, which is based on the local feature correspondences. The key of this approach is to formulate the nonrigid shape recovery as an unconstrained quadratic optimization problem which has a closed-form solution for a given set of observations.

For the appearance-based method, a deformable Lucas-Kanade algorithm is proposed which triangulates the template image into small patches and constrains the deformation through the second order derivatives of the mesh vertices. It is formulated into a sparse regularized least squares problem which is able to reduce the computational cost and the memory requirement. The inverse compositional algorithm is applied to efficiently solve the optimization problem. Furthermore, we present a fusion approach to take advantage of both the appearance information and the local features.

As for the 3D deformable surface recovery, the key challenge arises from the difficulty in estimating a large number of 3D shape parameters from noisy observations. In this thesis, 3D deformable surface tracking is formulated into an unconstrained quadratic problem that can be solved very efficiently by resolving a set of sparse linear equations. Furthermore, the robust progressive finite Newton method developed for nonrigid surface detection is employed to handle the large outliers.

Without resorting to an explicit deformable mesh model, the nonrigid surface detection can be treated as a generic regression problem. A novel velocity coherence constraint is imposed on the deformable shape model to regularize the ill-posed optimization problem. To handle the large outliers, a progressive optimization scheme is employed.

In addition to the methodologies studied and evaluated in computer vision, this thesis also investigates the nonrigid surface recovery in some real-world multimedia applications, such as Near-duplicate image retrieval and detection. In contrast to conventional approaches, the presented technique can recover an explicit mapping between two near-duplicate images with a few deformation parameters and find out the correct correspondences from noisy data effectively. To make the proposed technique applicable to large-scale applications, an effective multilevel ranking scheme is presented that filters out the irrelevant results in a coarse-to-fine manner. To overcome the extremely small training size challenge, a semi-supervised learning method is employed to improve the performance using unlabeled data. Extensive evaluations show that the presented method is clearly effective than conventional approaches.

可變形曲面恢復及應用

論文摘要

在計算機視覺和圖像分析領域，恢復可變形曲面是非常有意義同有價值的研究課題。變形曲面恢復技術有著廣泛的應用，如非剛性表面重建、數字娛樂、醫學影像和擴充現實。儘管過去已經有大量關於建立和擬合可變形曲面模型的研究，目前仍然缺少比較有效而且快速的方法來解決這個問題。本論文提出了一系列方法來解決二維非剛性形狀恢復和三維可變形曲面重構，而且這些方法皆基於魯棒的漸進優化方法。同時這項技術被應用在包含多媒體檢索在內的許多現實問題中。

本論文首先提出了新穎的漸進式有限牛頓優化方法來解決二維形狀恢復問題。這個方法的關鍵是我們把二維形狀恢復問題闡明成無約束的二次優化問題，而且在給定一組觀測值的前提下，可以得到閉合解。

其次，本論文提出了基於外觀的可變形 Lucas-Kanade 算法。該方法先將模板圖像劃分成三角形小塊，然後用劃分三角網格的坐標的二階導數來約束變形模型的變形幅度。而且相關優化問題被闡明成了稀疏正則最小平方問題，從而同時減少了計算量和內存需求量。這個優化問題則可以用逆組成算法來高效地解決。另外，我們也提出了融合算法用來同時利用外觀和局部特徵信息。

對於三維曲面恢復問題，最大的困難來自于從有噪聲的觀測中估計出大量的三維形狀參數。本論文將三維曲面重構問題闡述成無約束的二次優化問題，並通過解一系列稀疏綫形方程組來得到最終優化解。進一步而言，用於二維形狀恢復的漸進式有限牛頓優化方法可以用來處理觀測結果中的大量異常值。

此外，本論文還將非常性形狀檢測闡明成一般性回歸問題，從而不需要訴諸于顯性的可變形網格模型。爲了約束形狀變形和解決不適定問題，我們在這個一般性回歸模型基礎上外加了新穎的速度一致約束。

除了以上在計算機視覺領域的方法論研究和評估，本論文也探討非剛性曲面恢復在一些實際多媒體領域的應用，比如近似副本圖像的檢索和檢測。跟傳統方法不同，本論文提出的方法不但可以通過大量變形參數恢復出兩個近似副本圖像之間的顯性映射關係，而且可以非常有效地從有噪聲的大量觀測中找出正確的點對點對應關係。為了使提出的方法應用到大規模數據集上，我們提出了有效的多層排序方案使用由粗到精的方式來過濾非相關中間結果。爲了克服小樣本訓練難題，本論文提出了半監督學習方法，可以利用無標簽樣本的信息來提升檢索性能。

Acknowledgement

I would like to thank my supervisor, Prof. Michael R. Lyu, for his patient guidance, encouragement and advice he has provided throughout my time as his student. He taught me a lot on research, presentation and English writing. This thesis would not have been possible without his guidance and enthusiasm.

I also appreciate all the support and help from Prof. Thomas S. Huang at UIUC. He kindly invited me to visit his IFP group. I made lots of friends there, and learnt a lot form them. Prof. Steven C.H. Hoi at NTU has helped me so much on research. We have collaborated several great work. Many thanks to Prof. Shuicheng Yan at NUS, we have discussed a lot on various topics, and worked together during my UIUC trip. I also want to thank my other collaborators, Mr. Zenglin Xu at CUHK, and Prof. Rong Jin at MSU.

I would also like to thank Prof. Leo Jia and Prof. K. H. Wong for their suggestions on this work. My special thanks to Prof. Long Quan who kindly served as the external committee for this thesis, and provided some insightful comments.

I must show my gratitude to my colleagues in our group including Prof. Irwin King, Dr. Xia Cai, Dr. Haixuan Yang, Dr. Kaizhu Huang, Mr. Hao Ma, Mr. Hongbo Deng and many others, who gave me encouragement and kind help.

I also wish to express my gratitude to my parents and family, who always supported me during this period.

This work is dedicated to my parents.

Contents

Abstract **i**

Acknowledgement **v**

1 Introduction **1**

1.1 Deformable Surface Recovery 1

1.1.1 2D nonrigid shape recovery 2

1.1.2 3D deformable surface recovery 3

1.2 Motivation 4

1.3 Challenges 5

1.4 Main contributions of the thesis 7

1.5 Thesis outline 10

2 Background Study **13**

2.1 Overview 13

2.2 Physics-based Model 15

2.3 Interpolation 16

2.4 Data Embedding 19

2.4.1 Principal Component Analysis 19

2.4.2 Active Appearance Model 20

2.4.3 3D Morphable Model 23

2.4.4 Factorization Method 25

2.5 Convex optimization 26

2.5.1 Linear Program 26

2.5.2 Quadratic Program 27

2.5.3 Cone Program 27
2.5.4 Second Order Cone Program 28

3 Progressive Finite Newton Optimization 29
3.1 Motivation 29
3.2 Methodology and Overview 32
3.3 Nonrigid Surface Detection 34
3.3.1 2D Nonrigid Surface Model 35
3.3.2 Nonrigid Surface Recovery 36
3.3.3 Finite Newton Formulation 37
3.3.4 Progressive Finite Newton Optimization . 40
3.4 Experimental Results 43
3.4.1 Experimental Setup 43
3.4.2 Computational Efficiency 45
3.4.3 Performance of Nonrigid Surface Recovery 47
3.4.4 Augmented Reality 47
3.4.5 Medical Image 49
3.5 Discussions 49
3.6 Summary 53

4 Fusing Features and Appearance 54
4.1 Motivation and Methodology 54
4.2 Fusing Features and Appearance 57
4.2.1 Overview 57
4.2.2 Mesh Model 58
4.2.3 Fusing Features and Appearance Approach 59
4.2.4 Deformable Lucas-Kanade Algorithm . . . 61
4.2.5 Fusion Approach Optimization 66
4.3 Experimental Results 67
4.3.1 Experimental Setup 68
4.3.2 Evaluation on the Deformable Lucas-Kanade Algorithm 68
4.3.3 Fusion Approach 72

4.4 Discussions 74
4.4.1 Deformable Lucas-Kanade algorithm . . . 74
4.4.2 Fusion Approach 75
4.5 Summary 76

5 3D Deformable Surface Tracking 78
5.1 Motivation 78
5.2 Fast 3D Deformable Surface Tracking 82
5.2.1 Problem Definition 82
5.2.2 Convex Optimization Formulations 84
5.2.3 Unconstrained Quadratic Optimization . . 87
5.3 Experimental Results 89
5.3.1 Experimental Setup 89
5.3.2 Synthetic Data Comparison 91
5.3.3 Performance on Real Data 94
5.4 Summary 98

6 Velocity Coherence Regression 99
6.1 Motivation and Methodology 99
6.2 Velocity Coherence Regression 103
6.2.1 Theoretical Framework 104
6.2.2 Velocity Coherence Regularization 105
6.2.3 Link to Gaussian Process Regression . . . 107
6.3 Nonrigid Surface Detection 107
6.3.1 Robust Velocity Coherence Regression . . 108
6.3.2 Optimization 109
6.4 Experimental Results 110
6.4.1 Experimental Setup 111
6.4.2 Nonrigid Surface Detection 111
6.4.3 Medical Image 116
6.5 Discussion 116
6.6 Summary 118

7 Near-duplicate Keyframe Retrieval 119

7.1 Motivation . . . 119
7.2 Related Work . . . 123
7.3 Nonrigid Image Matching . . . 125
7.3.1 Nonrigid Image Matching . . . 125
7.3.2 Case Studies: Detecting Various NDKs . . 126
7.4 Multi-Level NDK Retrieval . . . 128
7.4.1 Framework Overview . . . 128
7.4.2 Formulation as a Machine Learning Task . 128
7.4.3 Semi-supervised Support Vector Machine . 130
7.5 Experiments . . . 133
7.5.1 Experimental Testbeds and Setup . . . 133
7.5.2 Feature Extraction . . . 135
7.5.3 Experiment I: Ranking on Global Features 137
7.5.4 Experiment II: Re-ranking with NIM on Local Features . . . 140
7.5.5 Evaluation of Computational Cost . . . 145
7.6 Summary . . . 146

8 Conclusion and Future Work 148

8.1 Conclusion . . . 148
8.2 Future Work . . . 150

A Optimization With Lighting 153

B Solution of Velocity Coherence Regression 155

C List of Publications 157

Bibliography 161

List of Figures

1.1 Example of the large bending deformations. . . . 5
1.2 An example of face images with different illuminations. 6
1.3 Perspective distortion and occlusions. 7

2.1 Tracking faces using proposed method in the video sequences. 24
2.2 An example of 3D face fitting using 3D mophable model. 25

3.1 Detecting nonrigid surfaces in real-time video. . . 31
3.2 The robust estimator. 37
3.3 Progressive Newton approach to nonrigid surface detection. 41
3.4 Probability of success of the robust estimator function. 44
3.5 A Starbucks pad is used as the deformable object for nonrigid surface detection. 45
3.6 The first row shows the initialization results using the modified RANSAC method, and the second row shows the results. 46
3.7 Re-texturing of a shirt print. 48
3.8 Re-texturing a picture on a piece of paper. 50
3.9 Medical image registration. 51
3.10 Medical image registration with missing data. . . 52
3.11 Results with large errors and some failure cases. . 52

4.1 Recovering nonrigid shapes. 57
4.2 Mesh model. 58
4.3 Overview of the 2D shape recovery algorithm. . . 60
4.4 Deformable Lucas-Kanade Algorithm. 62
4.5 The Jacobian $\frac{\partial W}{\partial x_i}$. 63
4.6 The gradient $\nabla T \frac{\partial W}{\partial \mathbf{s}}$. 65
4.7 An example of the deformable Lucas-Kanade fitting to a single image. 69
4.8 The Root Mean Square Error. 70
4.9 The Root Mean Square Error (RMSE) with given regularization parameter. 71
4.10 RMSE comparison of different methods. 71
4.11 Comparison of the deformable Lucas-Kanade method and Fusion approach. 73
4.12 Comparison of DLK method and Fusion approach. 74
4.13 Pattern occluded by a piece of paper. 75
4.14 Recovering the cover of a magazine in a real-time video. 76
4.15 Diminishing a picture on a piece of paper. 77

5.1 Recovering highly deformable surfaces from video sequences. 80
5.2 Synthetic meshes with 96 vertices for evaluation. . 90
5.3 The performance comparison of the QO, QP and SOCP methods. 92
5.4 Comparison of the performance of the QO and SOCP on the synthetic data. 93
5.5 Tracking the deformable surface with sever bending. 95
5.6 Tracking the deformable surface with waveform deformation. 95
5.7 Tracking the deformable surface with two sharp folds in it. 96
5.8 Recovering the deformation of a bag. 97

5.9 Recovering the deformation of a piece of cloth. . 97

6.1 Detecting nonrigid surfaces in real-time video. . . 103
6.2 Point set registration with large outliers. 112
6.3 Point set registration with pose variation. 112
6.4 Model image and test image used in the numerical comparison. 113
6.5 Nonrigid surface detection on a piece of paper. . . 114
6.6 Re-texturing a picture on a piece of paper. 115
6.7 Applying the proposed method to medical image registration. 117

7.1 Some near-duplicate keyframes examples selected from TRECVID2003 video corpus. 121
7.2 Examples of the detection results on various near-duplicate keyframe cases. 127
7.3 A multi-level ranking framework. 129
7.4 Semi-Supervised SVM Re-ranking Algorithm . . . 132
7.5 Cumulative accuracy of similarity measure and features using Nearest Neighbor Ranking. 138
7.6 Comparison of the proposed semi-supervised ranking method using S^{3}VM algorithm with other appearance based methods. 139
7.7 Cumulative accuracy of NDK retrieval using NIM method on the TRECVID2003 dataset. 140
7.8 Comparison of cumulative accuracy of NDK retrieval results on the TRECVID2003 dataset. . . 142
7.9 Comparison of cumulative accuracy of NDK retrieval results on the TRECVID2004 dataset. . . 143
7.10 Failure examples in showing when the presented method may fail. 145
7.11 Computational efficiency and retrieval performance on the TRECVID2003 dataset. 147

List of Tables

3.1 Computational time of proposed method at each step. 47

4.1 Computational time of the deformable Lucas-Kanade algorithm on different 2D mesh models. 70

5.1 Average computational time per frame (seconds) . 94

6.1 The accuracy on synthesized dataset with different standard deviations *std*. The mean square error (MSE) per vertex is adopted as the metric. . 113

7.1 Comparison of overall time cost of 300 queries on the TRECVID2003 dataset. 146

7.2 Comparison of overall time cost of 1,953 queries on the TRECVID2004 dataset. 146

Chapter 1

Introduction

Deformable surface recovery is essentially a computer vision task with a variety of applications in image alignment [116], surface reconstruction, medial imaging, object detection [76], object recognition [20], augmented reality [123], human computer interaction and digital entertainment. In this chapter, we describe the main problem, the motivation of our research, and the challenges of the research topic. The key contributions of this thesis are also described.

1.1 Deformable Surface Recovery

Deformable surface recovery from images is always an interesting research problem in computer vision and image analysis. There are various kinds of input data sources in the deformable surface recovery problem, such as single still image, multiple uncalibrated image, monocular video, stereo video and photometric stereo. Among these data acquisition methods, photometric stereo [7] and structure light method [111] require either the strict illumination configuration or the active input light, which limit their scope of application. Without making the strict assumption on the working environment, in this thesis, we focus on the problem related to recovering the deformable surface from single image and monocular video, which has attracted increasing attention in recent years. As for the stereo-based methods [16], they are able to take advantage of the effective deformable surface recovery algorithm developed for the monocular video and still images.

To make it clear, the deformable surface recovery problem is studied in 2D and 3D separately. 2D nonrigid shape recovery only locates the landmark points, while 3D deformable surface recovery deals with the more challenging problem of estimating the 3D model from the 2D observations.

1.1.1 2D nonrigid shape recovery

The main goal of 2D nonrigid shape recovery is to extract the deformable shape's structure from a given input image. In general, most of the current nonrigid shape recovery methods can be divided into two categories.

The first group is dependent on the extracted salient features in the image. In this group, there are two different methods to estimate the nonrigid mapping function from the local feature information. One is to take advantage of local feature matching in order to find correspondences [123], and employ a smoothing method to eliminate the false matches. The other directly matches two salient point sets by treating nonrigid shape recovery as a general graph matching problem [20, 26].

The second group is based on the appearance information, which tries to minimize the residual image between the synthesized template image and the input image [23].

Unlike nonrigid shape recovery, **nonrigid surface detection** [76] does not require any initialization or a prior pose information, which provides a fully-automatic solution. Note that only the local feature correspondence-based methods are employed in the nonrigid surface detection task, while the global appearance-based approaches [124] usually require good initialization to avoid stucking at the local minima.

1.1.2 3D deformable surface recovery

3D deformable models are extensively studied in computer graphics and computer vision for modelling, visualization, simulation and animation. Based on these models, 3D deformable surface recovery aims to reconstruct the visible surfaces from the laser-scanner data, stereo disparity map, image and video sequence. Various approaches have been proposed in the literature.

First of all, 3D surface can be directly extracted from the depth map data obtained by a laser-scanner, which usually contains less outliers than many other methods. To build the triangulated mesh surface, either the geometry-based triangulation method [43] or the implicit surface model fitting [18, 117] can be adopted in practice.

Without a dedicated device, the deformable surface can be directly recovered from the images, which may contain a large number of outliers. To fit the noisy disparity data obtained from image stereo, a prior generic mesh model is employed to constrain the surface deformation, which leads to a regularized least square optimization problem [45, 47].

Structure from motion method shows the promising results on recovering the 3D deformable shape from image sequences [19], which directly factorizes the tracked 2D point locations. Moreover, the shape deformation is represented as a linear combination of the shape basis.

In terms of the online application, **3D deformable surface tracking** takes advantage of the temporal information from consecutive frames in a video clip, and imposes the temporal and spatial consistence constraints to regularize the surface deformation.

1.2 Motivation

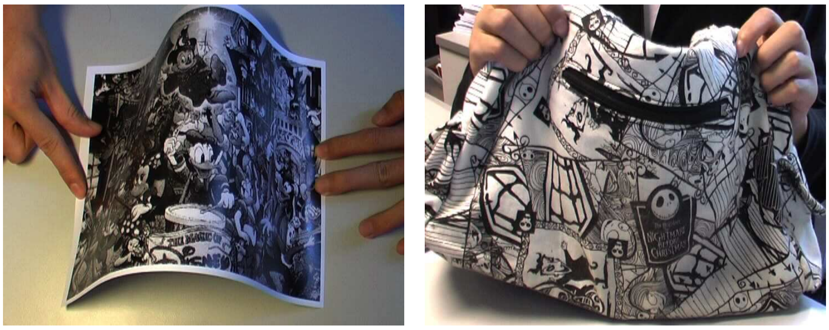
Given an annotated image example or a 3D textured mesh model, it is interesting to find out some automatic methods to spatially align the pre-defined model to the input image containing the object. This process can be employed to locate the object from the still image and extract the detailed structure information at the same time. Moreover, the object classification performance can be remarkably boosted by properly aligning the testing images to a predefined template. In addition, the motion of the deformable surface can be captured by tracking through a video sequence, and the results can be further used in computer animation. Therefore, the main objective of this thesis is how to effectively recover the deformable surface from images. However, deformable surface recovery is known to be very challenging due to involving the highly ill-posed optimization problem.

To tackle this critical problem, various deformable models are introduced to make the problem tractable by constraining the searching space of the deformation parameters. Generally speaking, there are two steps to build a deformation model. The first step is to find out an effective way to represent the deformable surface, which could be either an explicit triangulated mesh model or a set of points. In the second step, the regularization is imposed on the deformable surface to model and constrain the various kinds of deformation. Note that the regularization method is vital to deformable surface recovery which usually contains noisy observations and associates with the ill-posed optimization.

In this thesis, we try to explore different deformation models and propose novel deformable surface recovery methods for the real-world applications. We start from a local feature-based method that facilitates an automatic solution to detect the non-rigid surface. And then, the appearance is taken into account to exploit more information. For the even more challenging 3D deformable surface tracking, the temporal motion models over video sequences are employed to handle the ambiguity issue in scene depth.

1.3 Challenges

As a computer vision task, there are lots of challenges. In the following, we discuss the major aspects which have to be taken into account to develop a robust deformable surface recovery system.



(a) Paper

(b) Bag

Figure 1.1: Example of the large bending deformations. (a) A piece of paper is severely bended. (b) A bag is bended.

(1) **Noisy data.**
As only local feature descriptors are compared, the incorrect matches cannot be avoided in the feature-based method. On the other hand, the deformable surface is usually highly dynamic and represented by a large number of deformation parameters. Thus, it is difficult to directly apply the robust techniques widely used in statistics to remove the spurious matches. Furthermore, deformable surface recovery requires a sufficient number of correct correspondences in order to obtain high registration accuracy.

(2) **Deformations.**
Deformation usually occurs in the real-world applications. Fig. 1.1 shows a piece of paper and a bag under severe bending. Although a surge of research efforts have been devoted to the feature descriptors and matching, there is still a lack of an effective descriptor or matching scheme to handle the general deformations.

(3) **Ambiguity.**
Due to using the 2D observations only, solving the depth ambiguity issue is a challenging problem in the 3D monocular deformable surface tracking.

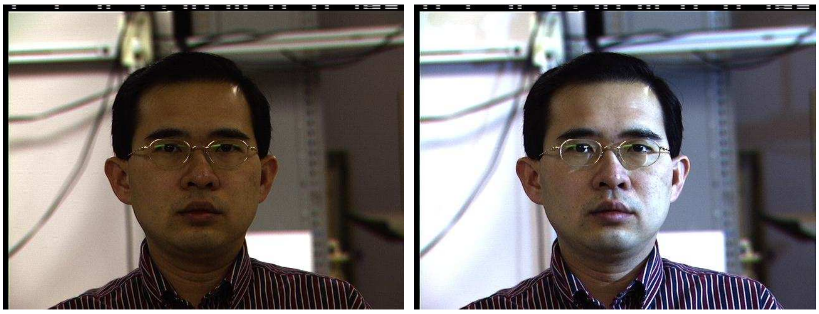
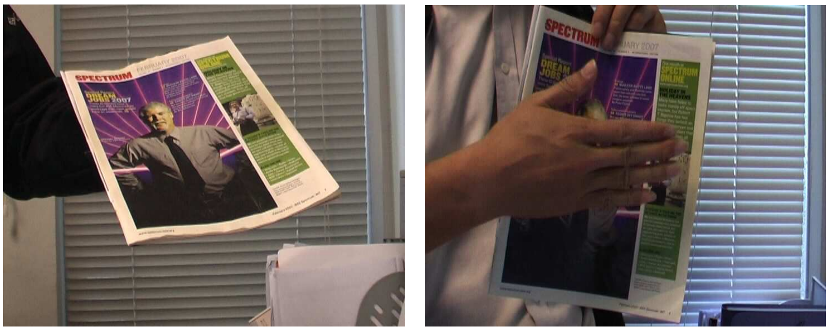


Figure 1.2: An example of face images with different illuminations.

(4) **Illumination changes.**
Illumination change in images is an important issue to be taken into account. Fig. 1.2 shows sample face images with different illuminations from the CMU PIE dataset [85]. In order to develop a robust system, we also consider that it should be capable of recovering the deformable surface under different illumination conditions.

(5) **Perspective distortions.**
Perspective distortions are mainly introduced by the camera lens, which needs to be properly handled in practice. Fig. 1.3(a) shows an example of perspective distortions.



(a) Perspective distortion

(b) Occlusion

Figure 1.3: (a) Magazine cover with perspective distortions. (b) Magazine cover is occluded by hand.

(6) **Occlusions.**
Partial occlusion is always an issue to be concerned in object detection and tracking. Fig. 1.3(b) shows that a magazine cover is partially occluded by hand. Also, the nonconvex surface may encounter the self-occlusion issue which requires to be properly handled.

1.4 Main contributions of the thesis

This thesis intends to develop the techniques that can effectively build the deformation models and efficiently recover the deformable surface. To this end, we have investigated both 2D nonrigid shape recovery and 3D deformable surface tracking problems respectively. A progressive finite Newton optimization scheme is proposed to attack the nonrigid surface detection problem. By taking advantage of the appearance information, a deformable Lucas-Kanade algorithm is presented for image alignment with large deformations. Furthermore, 3D deformable surface tracking is formulated into an unconstrained quadratic optimization problem which is reduced to a set of sparse linear equations. The main contributions of this thesis can be further summarized as follows:

(1) **Nonrigid surface detection.**

There are two different methods for nonrigid surface detection that will be presented in this thesis:

First, a novel progressive finite Newton optimization scheme for the nonrigid surface detection problem is proposed, which is reduced to only solving a set of linear equations. The key of this method is to formulate the nonrigid surface detection as an unconstrained quadratic optimization problem that has a closed-form solution for a given set of observations. Moreover, a progressive active-set selection scheme is employed, which takes advantage of the rank information of the detected correspondences.

In contrast to the first method, the second approach to nonrigid surface detection is formulated as a generic regression problem which does not require an explicit deformable mesh model. In addition, the proposed velocity coherence regression is equivalent to a special case of Gaussian Progress regression. Furthermore, the velocity coherence constraints are employed as the regularization term in this method.

(2) **Image alignment.**

The conventional Lucas-Kanade algorithm for image alignment only estimates either the affine transformation or the homography between the template image and the target image, which usually does not consider the local deformations. To deal with the image alignment with large deformations, we propose a novel deformable Lucas-Kanade algorithm which triangulates the template image into small patches and constrains the deformation through the second order derivatives of the mesh vertices. The presented deformable Lucas-Kanade algorithm is further formulated into a sparse regularized least squares problem, which is able to reduce the computational cost and the memory requirement. To solve the optimization problem in the deformable Lucas-Kanade fitting, an efficient inverse compositional algorithm is employed.

(3) **2D shape recovery.**
A fusion approach is presented to tackle the nonrigid shape recovery problem, which is able to take advantage of both the appearance information and the local feature correspondences. Moreover, the inverse compositional algorithm is also employed to deal with the associated optimization problem.

(4) **3D deformable surface tracking.**
Referring to the recent Second Order Cone Programming (SOCP) method [80], we first reformulate the 3D deformable surface tracking into an unconstrained quadratic optimization problem. Then, a closed-form solution for this problem is proposed, which is reduced to only solving a set of sparse linear equations. Based on this new framework, the progressive finite Newton optimization scheme is adopted to handle large noisy observation.

(5) **Near-duplicate keyframe retrieval and detection.**
We apply the proposed nonrigid surface detection method to retrieving the near-duplicate keyframes from real-world video corpora, which is an important problem in the multimedia domain. In contrast to the conventional methods, the proposed method takes consideration of the spatial coherence between two near-duplicate images, which is able to handle the local deformations. Moreover, this technique can recover an explicit mapping between two near-duplicate images with a few deformation parameters and find out the correct correspondences from noisy data effectively. To make this technique applicable to large-scale applications, we suggest an effective multi-level ranking

scheme that filters out the irrelevant results in a coarse-to-fine manner. In order to overcome the extremely small training size challenge in the near-duplicate keyframe retrieval, a semi-supervised learning method is employed to improve the performance by taking advantage of unlabeled data.

1.5 Thesis outline

This thesis reviews the main methodology in deformable surface recovery, and proposes some approaches to tackle this challenging research topic. To solve the associated optimization problem, an effective coarse-to-fine scheme has been presented. In addition to solving the conventional image registration and object tracking problem in computer vision, this thesis also employs the proposed technique to tackle the problems rising in multimedia domain, such as near-duplicate image retrieval and detection. The rest of this thesis is organized as follows:

- **Chapter 2.**
  This chapter reviews some background knowledge of the recent work on deformable surface modelling and recovery. Moreover, the main methodology and problems will be described.

- **Chapter 3.**
  In this chapter, we first introduce the feature-based approach to nonrigid surface detection, which offers a promising automatic solution. A novel progressive finite Newton approach is proposed to attack the associated optimization problem. Also, the 2D Finite Element Model is employed to regularize the surface deformation. This method will be evaluated on various applications, such as real-time re-

texturing the nonrigid surface and medical image registration.

- **Chapter 4.**
  This chapter presents a fusion approach to recover 2D nonrigid shape, which takes advantage of both the appearance information and the local features. Moreover, a deformable Lucas-Kanade algorithm is proposed for image alignment with large deformations. Extensive evaluations on these two methods will be illustrated.

- **Chapter 5.**
  Deformable surface recovery in 3D environment is more challenging than its 2D counterpart. In this chapter, we propose an effective closed-form solution for the 3D deformable surface tracking problem. A novel unconstrained quadratic optimization formulation is presented, which is reduced to a set of sparse linear equations. Moreover, the progressive finite Newton scheme described in Chapter 3 is employed to gradually reject the outlier matches. Extensive evaluations on both synthetic and real-world data will be discussed.

- **Chapter 6.**
  Instead of using the explicit triangulated mesh model in the previous part of this thesis, this chapter presents a robust velocity coherence regression method, in which the nonrigid surface detection is formulated as a generic regression problem. The velocity coherence constraint is imposed to regularize the surface deformation. Evaluations on this method will be presented.

- **Chapter 7.**
  In this chapter, we apply the proposed 2D nonrigid surface detection method to tackle the near-duplicated keyframe

retrieval problem, which catches more and more attention in multimedia community. To accelerate the method for large-scale data, a Multi-Level Ranking framework is presented, which takes advantage of the semi-supervised ranking techniques. Extensive evaluations on keyframes from TRECVID 2003 and TRECVID 2004 video corpora will be studied.

- **Chapter 8.**
  Finally, this chapter summarizes the whole thesis and addresses some directions to be explored in future work.

□ **End of chapter.**

Chapter 2

Background Study

During the past several decades, extensive research efforts in the computer vision community have focused on the problem on deformable object modeling and tracking [76, 116, 117, 122, 123]. There is a rich basis for deformable surface recovery. In this chapter, we first take a brief overview of the deformable surface modeling and recovery methods, and then review them in detail in the subsequent sections.

2.1 Overview

The interests in deformable surface recovery are very closely related to problems such as motion capture [99], simulation, image registration [6], feature matching [69] and object recognition [20]. Deformable models offer an attractive approach to tackling such kind of problems. This is because these models are able to represent the complex shapes and broad shape variability of anatomical structures. Also, deformable models overcome many of the limitations of conventional low-level image processing techniques, which provide compact and analytical representations of object shape and incorporates anatomic knowledge. Generally speaking, deformable surface models stem from the fusion of geometry, physics, statistical machine learning, and optimization theory. Geometry is the main tool to represent the object surface. Moreover, physics law imposes constraints on how the shape may vary over space and time. Furthermore, the statistical machine learning methods show excellent performance on modelling the surface deformations directly from examples and constraining the searching space of the deformation parameters. Finally, optimization approximation theory provides a mathematical foundation to estimate the deformation parameters from the noisy observations. The continuous development and refinement of these models should remain an important area of research into the future.

Prior models are essential to making the ill-posed optimization in deformable surface recovery trackable. Numerous deformable models have been proposed in the literature, which can be roughly divided into several categories. First of all, since surface deformation is essentially a natural phenomena, it is intrinsically studied from the geometry aspect. Thus, lots of physics-based models have been presented. The finite element method [91] is the most representative physical model used in deformable surface recovery. Second, the underlying problem for deformable surface recovery is to find out an optimal mapping function that fits to the input data. This can be tackled by the general data interpolation technique [98], which wins success in the point set matching problem. Third, the deformations are able to be directly modeled from examples by taking advantage of the advance in data embedding. Moreover, the statistical regularization theory provides a convenient tool to constrain the search space of the deformation parameters. Finally, the factorization method [19] makes use of both temporal motion information and linear subspace assumption, and simultaneously recovers the object motion and nonrigid shape from the video sequence. These models will be introduced in the following sections.

2.2 Physics-based Model

Physics-based model has been extensively investigated in the past two decades. In general, the physical interpretation views deformable models as elastic bodies which respond naturally to applied forces and constraints. Typically, deformation energy functions defined in terms of the geometric degrees of freedom are associated with the deformable model. The energy grows monotonically as the model deforms away from a specified natural shape, and often includes the terms that constrain the smoothness of the model. Taking a physics-based view of classical optimization, external energy functions are usually defined in terms of the data to be fitted, which give rise to external forces to deform the model.

From the above studies, Kass et. al [51] introduced Active Contour Models (or 'Snakes') which are energy minimizing curves. In the original formulation, the energy has an internal term which aims to impose smoothness on the curve, and an external term which encourages movement toward image features. They are often used to approximate the locations and shapes of object boundaries in images based on the reasonable assumption that boundaries are piecewise continuous or smooth. Later, this method was extended to 3D surface modelling by deformable superquadrics [91] and elastically deformable ballon model [22, 66]. However, since no deformable model other than the second order smoothness term is imposed, they are not optimal for the objects which have a known shape. This problem can be alleviated by incorporating the data-driven model that will be introduced in Section 2.4.

The finite element method is one of the most representative physics-based model, which comes from the mechanical engineering field and provides an analytic surface representation. In [45], the triangulation facets are treated as C^0 finite elements, which approximates the sum of square of the derivatives of displacements across the surface. This leads to an effective regularization term to prevent deformations at neighboring vertices of the mesh facets from being too different. Furthermore, the mesh model with hexagonally connected vertices has been successfully used in 3D reconstruction [32] and real-time nonrigid surface detection [76, 123]. This model mainly imposes the penalization of the squared second-order derivative of the mesh vertex coordinates.

Besides the explicit mesh representation, Free Form Deformation enables the researchers to parameterize an arbitrary mesh that may be irregular or with a large number of vertices, in terms of a relatively small number of control points and therefore parameters. By taking advantage of the Free Form Deformation representation, the implicit surfaces method [46, 47] shows promising results on fitting 3D deformable surface from the quite noisy image stereo data.

In addition to the above physics-based models, physical constraint plays a very important role in regularizing the deformable surface. For example, the temporal motion information is commonly used in nonrigid shape recovery from video sequences [19]. Moreover, Salzmann [82] synthesizes the paper-like deformable surface examples by strictly enforcing the degree of freedom for each mesh vertex.

2.3 Interpolation

Generally speaking, deformable surface recovery can be formulated as a problem of correspondence: finding an optimal mapping between one set of points and another set of points. Such mapping can be found by the interpolation method [98] which is highly effective to search for the optimized nonlinear mapping function between the source and the target. Also, such nonlinear mapping function shown to be very promising in representing various image distortions induced by different kinds of deformations. Therefore, lots of nonrigid shape recovery approaches are developed on top of the data interpolation method, such as Radial Basis Functions [6] and Thin-Plate Spline [20], which are originally used in regression analysis.

Thin-Plate Spline is the most popular mapping function used in point set matching problem. Formally, Thin-Plate Spline is defined by the centers and coefficients as follows:

$$f(\mathbf{u}) = \sum_{i=1}^{n} \begin{bmatrix} w_i^x \\ w_i^y \end{bmatrix} \phi(|\mathbf{u} - \mathbf{u}_i|) + A\mathbf{u} + \mathbf{t} \tag{2.1}$$

where $\phi(r) = r^2 \log r$ and $\mathbf{u} = [\ x \quad y\ ]^\top$. A and t are affine transformation parameters. The matrix $W \in R^{n\times 2}$ is defined as below:

$$W = \begin{bmatrix} w_1^x & w_2^x & \dots & w_n^x \\ w_1^y & w_2^y & \dots & w_n^y \end{bmatrix}^\top,$$

which specifies the nonlinear mapping with n centers.

Thin-Plate Spline is mainly penalized by bending energy E_b, which is defined as the integral of the squares of the second derivatives [89]:

$$E_b = \int\int (f_{xx}^2 + 2f_{xy}^2 + f_{yy}^2) dxdy$$

Usually, the point set matching problem minimizes the sum of residual errors and penalty term. To deal with the associated optimization problem and the noisy observations, various approaches have been proposed in the literature. Several representative methods are described in the following part of this section.

Chui et al. [20] present a coarse-to-fine approach to jointly determine the correspondences and nonrigid transformation between two point sets through deterministic annealing and softassign. Moreover, shape-context feature is used to find the local point correspondences. Additionally, an iterative algorithm is employed to estimate both the correspondences and the nonlinear mapping function.

In the most recent studies, the probabilistic approach for the nonrigid point set matching is attracting increasing research interests [49, 67, 71]. The point set matching is interpreted as a mixture density estimation problem [38], where one point set represents the centers of Gaussian mixture models and the other represents sample data. This problem is usually solved by the Expectation Maximization (EM) algorithm. Another idea is to model each of the two point sets by a kernel density function and then measure the similarity. In [96], Tsin and Kanade proposed a kernel correlation based approach to register the nonrigid point set, which minimizes the L_2 norm between the distributions. Later, Jian and Vemuri [49] extended this approach via representing the density by Gaussian mixture models. Furthermore, Myroneko et al. [71] presented a coherent point drift method for nonrigid point set registration, which does not make an explicit assumption of the transformation model.

The point set matching problem can also be simply viewed as a graph-matching problem [26], in which graph nodes represent feature points extracted from either an input image or a model image and graph edges represent relationships between feature points. The problem of graph matching is to find a mapping between the two node sets that preserves as much as possible the relationships between nodes. Because of its combinatorial nature, graph matching is either solved exactly in a very restricted setting or approximately.

In contrast to the point set matching method, Bartoli and Zisserman [6] present an intensity-based scheme, which directly minimizes the registration residual errors. In this scheme, both control point positions and transformation coefficients are optimized simultaneously.

2.4 Data Embedding

As a data-driven method, data embedding is intensively studied in the statistical machine learning community, which is widely used in image processing, computer vision and computer graphics. There are three important approaches in data embedding: Principal Component Analysis [33], ISOMAP [90] and Locally Linear Embedding [79]. In this thesis, we only focus our attention on the conventional Principal Component Analysis-based methods.

Benefiting from the excellent reconstruction and projection characteristics, Principal Component Analysis is employed to build deformable model directly from the examples [23, 82, 13]. Among the various data-driven approaches, Active Appearance Model and 3D Morphable Model win great success in recent years, which take into account of both the structure and appearance information. In this section, we first introduce Principal Component Analysis. Then, two representative approaches, Active Appearance Model and 3D morphable model, are presented respectively. Finally, the factorization method is introduced.

2.4.1 Principal Component Analysis

Principal Component Analysis is commonly used in signal processing, statistics, machine learning and image analysis. The fundamental idea in Principal Component Analysis is to find the components so that they explain the maximum amount of variance possible by some linear transformed components. Therefore, Principal Component Analysis is mainly used to remove the redundancy from the data. The representation given by

Principal Component Analysis is an optimal linear dimension reduction technique in the mean-square sense, and noises may be reduced.

Given a set of data $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_l\}$ taking values in an n dimensional feature space, the Principal Component Analysis of a random vector $\mathbf{x}$ factorizes its covariance matrix by eigen-decomposition. An important property of Principal Component Analysis is its optimal signal reconstruction in the sense of minimum mean-square-error when only a subset of principal components is used to represent the original signal. Therefore, the data $\mathbf{x} \in R^n$ can be projected on the low dimension space:

$$\mathbf{b} = P^\top(\mathbf{x} - \bar{\mathbf{x}}) \tag{2.2}$$

where $P \in R^{n \times m}$ is the projection matrix that is made of the eigenvectors corresponding to the m largest eigenvalues. Moreover, the low dimensional vector $\mathbf{b} \in R^m$ captures the most expressive features of the original data.

2.4.2 Active Appearance Model

Active Appearance Model [1, 23, 24] has been proven to be a very successful method in fitting statistical models of appearance onto new images. Active Appearance Model is taking the analysis-through-synthesis approach to the extreme, which has been successfully applied in numerous different applications [1, 23]. It establishes a compact parameterizations of object variability, as learnt from a training set by estimating a set of latent variables. The modeled object properties are usually shape and pixel intensities. If only considering the shape variations, this degenerated model is named as Active Shape Model [25] which manipulates a shape model to describe the location of structures in a target image. In contrast to Active Shape Model, Active Appearance Model is able to synthesize novel photo-realistic images. Therefore, we will only introduce Active Appearance Model in detail in the following part of this section.

Building Active Appearance Model

The main idea in modeling the shape variations is to perform Principal Component Analysis on a set of aligned shapes. Then, a new shape instance $\mathbf{s} \in R^{2n}$ with n landmark points can be synthesized by:

$$\mathbf{s} = \bar{\mathbf{s}} + \mathbf{P_s b_s} \tag{2.3}$$

where $\bar{\mathbf{s}}$ represents the mean shape, and projection matrix $\mathbf{P_s} \in R^{2n \times m}$ contains the m eigenvectors corresponding to the largest eigenvalues. $\mathbf{b_s}$ is a vector formed by a set of deformation parameters, which controls the shape variations. Thus, we can manipulate the shape model by changing $\mathbf{b_s}$. Given a shape instance $\mathbf{s}$, $\mathbf{b_s}$ can be estimated as below:

$$\mathbf{b_s} = \mathbf{P_s}^\top (\mathbf{s} - \bar{\mathbf{s}}) \tag{2.4}$$

Similarly, the appearance model is built by applying Principal Component Analysis on the pixel intensities. To align the training samples, all the images are mapped into the reference frame by piecewise-affine mapping or Thin-Plate Spline in Eqn. 2.1.

Fitting Algorithm

The objective of Active Appearance Model fitting is to directly minimize the residual error between the sampled input image and the synthesized model instance. Generally speaking, there are three different approaches to fitting Active Appearance Model onto a still image.

Multi-linear regression for Active Appearance Model fitting [23, 88] is a statistical approach, which mainly learns the relation between the displacements of model parameters and residual images. This method requires a separate learning step in order to build the Gram matrix from a large collection of the training samples. Under this framework, canonical component analysis [29] can also be employed to estimate the update of the shape and appearance parameters.

Lucas-Kanade framework provides a general solution for an image alignment problem. As the objective of Active Appearance Model fitting is exactly same as the goal of Lucas-Kanade algorithm, it can be treated as an image alignment problem. In [65], Baker and Matthew thoroughly investigate Active Appearance Model fitting under the Lucas-Kanade framework, and propose a very efficient inverse composition algorithm. Based on Baker and Matthews' work, several more specific problems are studied, such as occlusion [36] and mutiview fitting [56]. Similar to the original Lucas-Kanade algorithm, these methods employ the steepest descent optimization scheme. Among the algorithms described in [65], the inverse composition algorithm is the most efficient one, and can reduce the computational cost by pre-computing the Hessian matrix. Comparing to multilinear regression, Lucas-Kanade algorithm-based approaches do not require a training phase, and can take advantage of the advance in steepest descent optimization, such as line search and Levenberg-Marquardt method [15].

Discriminative methods impose Active Appearance Model fitting problem as a classification task. In [27], Constrained Local Model directly searches the surrounding patches to locate the facial landmark points. Moreover, a discriminative appearance model [61] utilizes the weak-classifier to extract the shape from a still image. Similar to Active Shape Model, the discriminative method is optimized for the local feature locality accuracy rather than the reconstruction error.

As many other appearance-based image alignment methods, Active Appearance Model fitting tends to stuck at local minima, and requires good initialization. A remedy is to employ either the multiple-resolution fitting [23] or a feature-based initialization method [124].

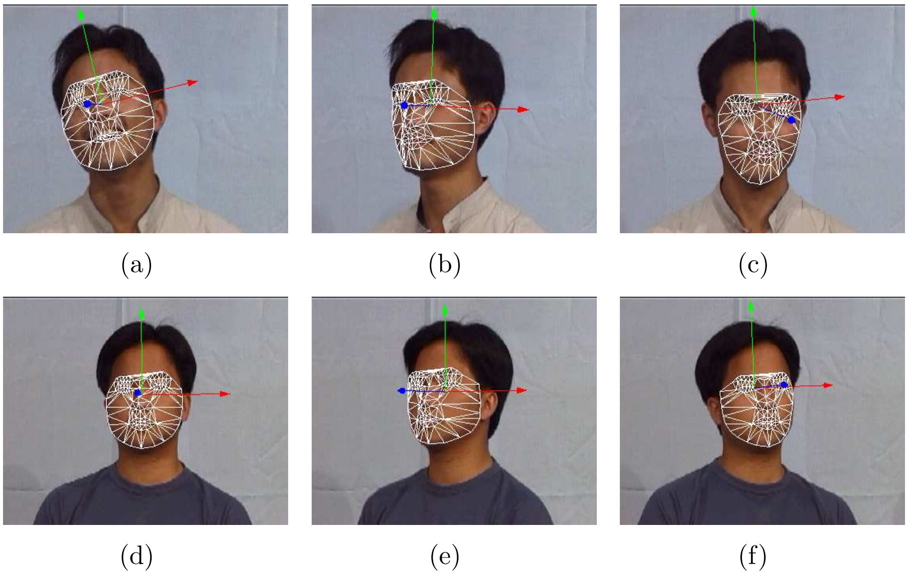
Application

Numerous interesting applications are developed on top of the Active Appearance Model fitting, such as face alignment [23], head tracking [1, 122], facial expression analysis, gaze estimation [105] and Augmented Reality [116]. Fig. 2.1 demonstrates an application based on the Active Appearance Model tracking. we roughly describe the main methodology in the following. In this example, a two-stage scheme is employed for online non-rigid shape recovery toward Augmented Reality applications. First, 3D shape models are built from Active Appearance Model tracking results offline, which does not involve processing of the 3D scan data. Based on the computed 3D shape models, an efficient online algorithm is employed to estimate both 3D pose and non-rigid shape parameters via local bundle adjustment for building up point correspondences. This approach, without manual intervention, can recover the 3D non-rigid shape effectively from either real-time video sequences or a single image. The estimated 3D pose parameters can be used for Augmented Reality registrations, as illustrated in Fig. 2.1.

2.4.3 3D Morphable Model

3D morphable model is first presented by V. Blanz and T. Vetter [13] to synthesize and analyze the facial images, which has been successfully used in a large number of applications, such as face recognition [14, 110], facial expression analysis, face reconstruction and exchanging faces in still images [12].

Similar to Active Appearance Model, 3D morphable model employs Principal Component Analysis to learn a sophisticated statistical model and regularize the deformation parameters. In-



(a) (b) (c)

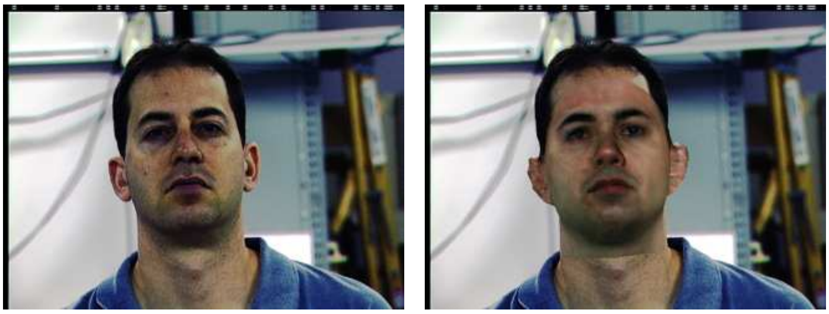
(d) (e) (f)

Figure 2.1: Tracking faces using proposed method in the augmented video sequences, the axis in the displayed frames indicates the current 3D pose of tracked subject.

stead of a sparse 2D mesh model used in Active Appearance Model, 3D morphable model adopts a dense 3D mesh model in order to synthesize the photo-realistic facial image.

By taking advantage of its 3D representation and the rendering techniques in computer graphics, 3D morphable model enjoys several merits. First of all, it offers a promising approach to tackling the pose variations problem in face recognition and head tracking. Second, the lighting problem can be properly handled through introducing some illumination models, such as Phong illumination model [13, 14] and spherical harmonic model [8, 110]. When the light source position is available, cast shadows and specular reflections can be correctly modeled. Finally, this method is able to render photo-realistic image with fewer artifacts.

As another appearance-based fitting method, this technique requires a good initialization for pose and illumination parameters in order to avoid stucking at local minima during the op-



(a) Input image (b) Fitting result

Figure 2.2: An example of 3D face fitting using 3D mophable model. (a) Original input image. (b) The estimated 3D face is overlaid on the input image.

timization process. Moreover, the regularization is essential to tackling the overfitting issue, which is properly handled by Probabilistic Principle Component Analysis. Fig. 2.2 demonstrates a 3D face fitting example, which employs a gradient-based optimization method [115] to estimate the 3D pose, illumination and model parameters.

2.4.4 Factorization Method

Usually, the factorization method first tracks the feature points across the whole video sequence, and then recover both the motion parameters and the nonrigid shape simultaneously.

Bregler et al. [19] proposed a solution for recovering 3D nonrigid shape models from image sequences. Their technique is based on a non-rigid model, in which the 3D shape in each frame is a linear combination of a set of basis shapes. By analyzing the low rank of the image measurements, they proposed a factorization-based method that enforces the orthonormality constraints on camera rotations for reconstructing the non-rigid shape and motion. Later, Torresani et al. [93] extended the method in [19] to initialize the optimization process in the factorization method.

Xiao et al. [105] presented a non-rigid structure-from-motion algorithm that is able to convert an Active Appearance Model into a 3D face model. They described how a non-rigid structure-from-motion algorithm is able to be employed to compute the corresponding 3D shape models from an Active Appearance Model. Their method does not require 3D range data in [14] and also shows fast fitting speeds. They then show how the 3D modes could be used to constrain the Active Appearance Model so that it not only can generate model instances, but also can be generated with the 3D modes. In [116], a similar technique is employed to recover the 3D shape basis for an online tracking application.

Note that these methods [19, 93] have been successfully used in offline non-rigid shape recovery from image sequences through performing factorization analysis on the 2D tracked points. However, it is difficult to directly extend the above methods to an online tracking application.

2.5 Convex optimization

In most recent work, 3D deformable surface recovery is formulated as the convex optimization problems [80, 121] without resorting to a deformation model. These convex problems can be optimally solved very efficiently. The remaining part of this section reviews several major convex optimization problems associated with this thesis. More details on convex optimization theory can be found in [15].

2.5.1 Linear Program

Definition 1. Linear Program (LP): *A convex optimization problem is called a linear program when the objective and constraint functions are all affine. The LP is generally expressed as* *follows:*

$$\begin{aligned} \min_{\mathbf{x}} \quad & \mathbf{c}^\top \mathbf{x} + d \\ s.\ t. \quad & G\mathbf{x} \preceq \mathbf{h}, \\ & A\mathbf{x} = \mathbf{b}, \end{aligned} \tag{2.5}$$

where $\mathbf{x} \in R^n$ *is the optimization variable,* $G \in R^{m\times n}$ *and* $A \in R^{p\times n}$.

2.5.2 Quadratic Program

Definition 2. Quadratic Program (QP): *A convex optimization problem is called a quadratic program if the objective function is convex and quadratic, and constraint functions are all affine. The QP has the following form:*

$$\begin{aligned} \min_{\mathbf{x}} \quad & \frac{1}{2}\mathbf{x}^\top P\mathbf{x} + \mathbf{c}^\top \mathbf{x} + d \\ s.\ t. \quad & G\mathbf{x} \preceq \mathbf{h}, \\ & A\mathbf{x} = \mathbf{b}, \end{aligned} \tag{2.6}$$

where $P \in S^n_+$, $G \in R^{m\times n}$ *and* $A \in R^{p\times n}$.

Obviously, linear program can be viewed as a special case of quadratic program when matrix $P = 0$.

The problem of minimizing the convex quadratic function

$$\|A\mathbf{x} - \mathbf{b}\|_2^2 = \mathbf{x}^\top A^\top A\mathbf{x} - 2\mathbf{b}^\top \mathbf{x} + \mathbf{b}^\top \mathbf{b}$$

is an unconstrained QP. It arises in many fields and has many names, e.g., regression analysis or least-squares approximation. This problem is simple enough to have the well known analytical solution $\mathbf{x} = A^\dagger \mathbf{b}$, where $A^\dagger$ is the pseudo-inverse of A.

2.5.3 Cone Program

In addition to the standard forms of the convex optimization problem, the generalized inequality constraints are another representation. One of the most representative case is the Cone Program (CP), which is defined as below:

Definition 3. Cone Program (CP): *A convex optimization problem with generalized inequalities is called a Cone Program if the objective is linear, and the inequality constraint functions are all affine. The CP has the following general form:*

$$\begin{aligned} \min_{\mathbf{x}} \quad & \mathbf{c}^\top \mathbf{x} + d \\ s.\ t. \quad & F\mathbf{x} + \mathbf{g} \preceq_K \mathbf{0}, \\ & A\mathbf{x} = \mathbf{b}, \end{aligned} \tag{2.7}$$

where $K \subseteq R^k$ is a proper cone.

2.5.4 Second Order Cone Program

Definition 4. Second Order Cone Program (SOCP): *A second order cone program is closely related to quadratic program, which has the following general form:*

$$\begin{aligned} \min_{\mathbf{x}} \quad & f^\top \mathbf{x} \\ s.\ t. \quad & \|A_i\mathbf{x} + \mathbf{b}_i\|_2 \leq \mathbf{c}_i^\top \mathbf{x} + d_i, \quad i = 1, \ldots, m \\ & F\mathbf{x} = \mathbf{g}, \end{aligned} \tag{2.8}$$

where $A \in R^{n_i \times n}$ and $F \in R^{p \times n}$.

A constraint of the form:

$$\|A\mathbf{x} + \mathbf{b}\|_2 \leq \mathbf{c}^\top \mathbf{x} + d$$

is named as a second-order cone constraint, since it is the same as requiring the affine function $(A\mathbf{x} + \mathbf{b}, \mathbf{c}^\top \mathbf{x} + \mathbf{d})$ to lie in the second-order cone in R^{k+1}.

When $A_i = 0$, $i = 1, \ldots, m$, the SOCP is equivalent to a general LP. Similarly, if $\mathbf{c}_i = 0$, $i = 1, \ldots, m$, then SOCP reduces to a quadratic constrained quadratic program (QCQP).

□ **End of chapter.**

Chapter 3

Progressive Finite Newton Optimization

In this chapter, we will introduce the feature-based nonrigid shape recovery technique. A novel progressive finite Newton optimization scheme is proposed for the nonrigid surface detection problem, which is reduced to only solving a set of linear equations. Extensive experiments have been conducted for performance evaluation on various environments, whose promising results show that the proposed algorithm is more efficient and effective than the existing iterative methods.

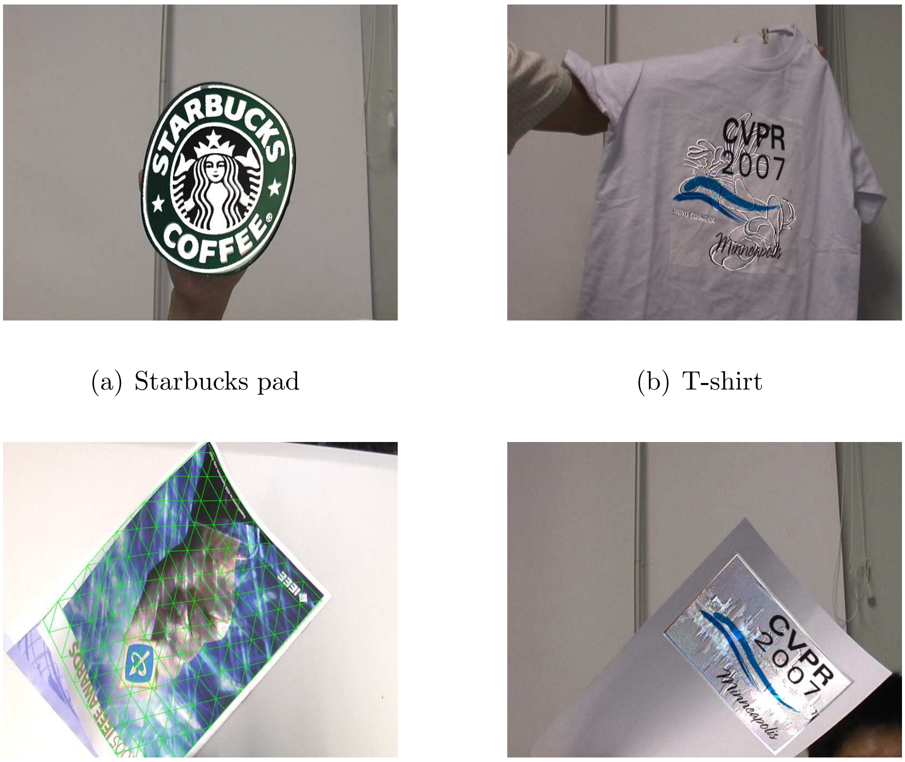
3.1 Motivation

The detection and tracking of the nonrigid objects in images and videos is an interesting and beneficial research issue for computer vision and image analysis [6, 75, 95]. The goal of nonrigid surface detection is to extract the deformable shape's structure from an input image. The difference between nonrigid surface recovery and detection is that the latter does not require any initialization or a priori pose information. An effective nonrigid surface detection technique can be applied in a variety of applications for digital entertainment, medical imaging [6] and Augmented Reality, such as the re-texturing of images and videos [100, 101].

Nonrigid surface detection can usually be treated as the problem of recovering the explicit surface with a few deformation parameters and finding out the correct correspondences from noisy data simultaneously. Many applications have been investigated for deformable object tracking and registration, such as face tracking and modelling [14, 23, 28, 105, 116], and also more generic and more deformable objects [6]. The major problem of these methods is that they tend to be computationally expensive and mainly aim at object recognition and image segmentation tasks rather than the nonrigid surface recovery. However, a real-time and automated solution [75] has recently been proposed, which takes advantage of an iterative robust optimization scheme.

Unlike the rigid object pose estimation, it is difficult to directly employ a robust estimator, such as RANSAC [31] or Hough transform [39], to remove the spurious matches for nonrigid surface detection. Because the nonrigid surface is usually highly dynamic and represented by many deformation parameters, the problem is far more complex than the rigid object detection. Moreover, it requires a sufficient number of correct correspondences in order to obtain high registration accuracy. An alternative strategy is to iteratively solve for both the correspondence and the transformation [10, 75]. However, these methods are either sensitive to initial conditions and parameter choices, or involve too many iterations and a complex optimization procedure. Consequently, they are neither efficient nor effective for real-time applications.

In this chapter, we propose a novel progressive finite Newton optimization scheme for nonrigid surface detection, which has the advantage of solving only a fixed number of linear equations. Moreover, a progressive sample scheme far more efficient than RANSAC is proposed to initialize the optimization process. The previous method [75] is generally accepted as the most ef-



(a) Starbucks pad

(b) T-shirt

(c) Cover

(d) Paper

Figure 3.1: Detecting nonrigid surfaces in real-time video (a-d). (a) The contour is overlaid on the Starbucks pad. (b) T-shirt with shadow. (c) The cover of a magazine. (d) A piece of paper with specular reflection.

fective state-of-the-art method in solving this kind of problem. It employs an implicit iterative scheme for the first order partial differential equation; however, this requires a large number of iterations to solve the problem and remove the outliers simultaneously. We tackle this critical problem from two angles. First, the nonrigid surface detection is formulated as an unconstrained quadratic optimization problem, which inherits a closed-form solution for a given set of observations. Thus, it can be efficiently solved through LU factorization. Then, a progressive sample scheme [21] is employed to initialize the optimization scheme, which can decrease the number of trials significantly. Therefore, the proposed approach requires much fewer iterations than the semi-implicit iterative optimization scheme [76], and it is very efficient for real-time nonrigid surface recovery tasks. To evaluate the performance of the proposed algorithm, extensive experiments have been conducted on such diverse objects as a Starbucks pad, a T-shirt, and the cover of a magazine, as shown in Fig. 3.1.

3.2 Methodology and Overview

Although nonrigid surface detection in general is not new to researchers in the computer vision domain, only a few approaches are automatic and can achieve real-time results. Some appearance-based approaches directly minimize the residual image between the input image and the synthesized model image [23]. Moreover, optical flow information [6, 28] can be incorporated into the optimization scheme to obtain better results. However, the major limitation of these methods is that they tend to become stuck at a local minimum and hence require good initialization. In addition, it is usually difficult to handle the partial occlusion for an appearance-based method. Well-designed markers widely used in motion capture are also applied to recover the structure of a nonrigid surface, such as cloth and paper [100, 101]. As these methods rely on the physical markers, they require the placing of pre-defined patterns on the target surface. Nevertheless, they are capable of high accuracy. On the other hand, feature-based methods [10, 75] try to find out the transformation from the correspondences built by feature matching methods. Thus, these methods can benefit from the recent advances in the feature detection and matching. In [75, 76], J. Pilet et al. proposed an iterative approach to attack the fast nonrigid surface recovery problem. Physical constraints based on the Finite Element Model [95] are employed for regularization. A semi-implicit iterative scheme is proposed to solve the optimization problem.

Recently, several sophisticated feature descriptors [62, 69] have been proposed to handle the wide-baseline matching problem, including images with large deformation [60]. In addition, machine learning methods, such as random classification trees [58], are also employed to find the point correspondences. These methods can take advantage of shifting part of the computational load from the matching phase to the training phase.

It is more complex to handle a large amount of deformation parameters for detecting the nonrigid surface rather than only a few pose parameters used in the rigid object detection. Therefore, there are several challenges when applying conventional robust estimators, such as RANSAC and M-estimator, for the nonrigid surface detection task. One is the lack of a concise function which can estimate the deformed mesh from the correspondences directly; instead, one may need to use a large number of free variables, which can lead to a high computational cost for each prediction step. Obviously, the semi-implicit iterative approach [75] is not efficient enough to deal with this problem. Another challenge is that the RANSAC-based approach requires a large number of trials. This makes the problem even more complex. Moreover, to the best of our knowledge, there is still a lack of criteria for selecting the number of samples for each trial in nonrigid surface detection. In rigid object pose estimation, the sample number is usually set according to the number of free parameters. However, the number of deformation parameters for a nonrigid surface may be larger than the total number of observations. This initialization problem is tackled through a modified RANSAC method. The key is to draw from progressively larger sets of top-ranked correspondences [21], rather than to treat all correspondences as equal and draw random samples uniformly from the full set in RANSAC. Thus, the progressive sample scheme affords large computational savings, and the conventional robust estimator can be engaged for initializing the nonrigid surface detection.

In contrast to the previous work, the proposed approach is based on a progressive finite Newton scheme, in which the optimization problem can be solved very efficiently by a factorization method. In addition to offering computationally highly competitive performance, the proposed modified RANSAC initialization method can further reduce the number of Newton optimization steps.

The rest of this chapter is organized as follows. In Section 3.3, we present the proposed progressive finite Newton solution. Section 3.3.1 describes the nonrigid surface model and mapping function for a feature matching-based method. Section 3.3.2 presents the object function which minimizes the correspondence errors and surface energy. A robust estimator is introduced to deal with the large outliers. In Section 3.3.3, the nonrigid surface detection is formulated as an unconstrained quadratic optimization problem, which is efficiently solved using the factorization method. Section 3.3.4 presents the progressive finite Newton optimization scheme to remove the spurious correspondences, and the progressive sampling method to initialize the optimization. Section 3.4 provides the details of the experimental implementation and describes the experimental results. Limitations are discussed in Section 3.5. Section 3.6 summarizes this chapter.

3.3 Nonrigid Surface Detection

In this section, we describe the present progressive finite Newton optimization scheme for detecting and recovering the nonrigid surface. For tackling the challenges, a mapping function is used to associate the feature correspondences with a mesh model.

Therefore, the nonrigid surface detection turns out to be a problem which minimizes the correspondence error and the surface energy. Moreover, the nonrigid surface detection is formulated into an unconstrained quadratic optimization problem. A progressive scheme is proposed to deal with outliers and find out as many correct correspondences as possible. Finally, a modified RANSAC scheme is introduced to select the initial active set for the optimization scheme.

3.3.1 2D Nonrigid Surface Model

The nonrigid surface is usually explicitly represented by triangulated meshes. As shown in Fig. 3.1(c), a triangulated 2D mesh with N hexagonally connected vertices is employed, which are formed into a shape vector $\mathbf{s}$ as below:

$$\begin{aligned}\mathbf{s} &= \begin{bmatrix} \mathbf{s_x} & \mathbf{s_y} \end{bmatrix}^\top \\ &= \begin{bmatrix} x_1 & x_2 & \dots & x_N & y_1 & y_2 & \dots & y_N \end{bmatrix}^\top\end{aligned}$$

where $\mathbf{x}$ and $\mathbf{y}$ are the vectors of the coordinates of mesh vertices. Assuming that a point $\mathbf{m}$ lies in a triangle whose three vertices' coordinates are (x_i, y_i),(x_j, y_j) and (x_k, y_k) respectively, and $\{i, j, k\} \in [1, N]$ is the index of each vertex. The piecewise affine transformation is used to map the image points inside the corresponding triangle into the vertices in the mesh. Thus, the mapping function $W(\mathbf{m}, \mathbf{s})$ is defined as below:

$$W(\mathbf{m}, \mathbf{s}) = \begin{bmatrix} x_i & x_j & x_k \\ y_i & y_j & y_k \end{bmatrix} \begin{bmatrix} \xi_1 & \xi_2 & \xi_3 \end{bmatrix}^\top \tag{3.1}$$

where (ξ_1, ξ_2, ξ_3) are the barycentric coordinates for the point $\mathbf{m}$.

3.3.2 Nonrigid Surface Recovery

In general, the nonrigid surface detection problem approximates a 2D mesh with $2N$ free variables, which is usually ill-posed. One effective way to attack this problem is to introduce regularization, which preserves the regularity of a deformable surface. The following object function is widely used in deformable surface fitting [47, 51, 75, 76] for energy minimization:

$$E(\mathbf{s}) = E_c(\mathbf{s}) + \lambda_r E_r(\mathbf{s}) \tag{3.2}$$

where $E_c(\mathbf{s})$ is the sum of the weighted square error residuals for the matched points. Also, $E_r(\mathbf{s})$ is the regularization term that represents the surface deformation energy, and λ_r is a regularization coefficient.

A set of correspondences M between the model and the input image can be built through a point matching algorithm. Therefore, a pair of matched points is represented in the form of $\mathbf{m} = \{\mathbf{m}_0, \mathbf{m}_1\} \in M$, where $\mathbf{m}_0$ is defined as the 2D coordinates of a feature point in the training image and $\mathbf{m}_1$ is the coordinates of its match in the input image. Then, the correspondence error term $E_c(\mathbf{s})$ is formulated as below:

$$E_c(\mathbf{s}) = \sum_{\mathbf{m} \in M} \omega_{\mathbf{m}} \mathcal{V}(\delta, \sigma) \tag{3.3}$$

where $\mathcal{V}(\delta, \sigma)$ is a robust estimator, and $\omega_{\mathbf{m}} \in [0, 1]$ is a weight linked with each correspondence.

The regularization term E_r in Eqn. 3.2, also known as internal force in Snakes [51], is composed of the sum of the squared second-order derivatives of the mesh vertex coordinates.

As the mesh is regular, $E_r(\mathbf{s})$ can be formulated through a finite difference:

$$E_r = \mathbf{s}^\top \mathcal{K} \mathbf{s} \tag{3.4}$$

where matrix $\mathcal{K} \in R^{2N \times 2N}$ is defined as below:

$$\mathcal{K} = \begin{bmatrix} K & 0 \\ 0 & K \end{bmatrix}$$

Note that K is a sparse and banded matrix which is determined by the structure of the explicit mesh model [32].

3.3.3 Finite Newton Formulation

In this thesis, we employ a robust estimator $\mathcal{V}(\delta, \sigma)$ with compact support size σ. Moreover, δ is the residual error, which is defined as follows:

$$\delta = \mathbf{m}_1 - W_{\mathbf{s}}(\mathbf{m}_0) \tag{3.5}$$

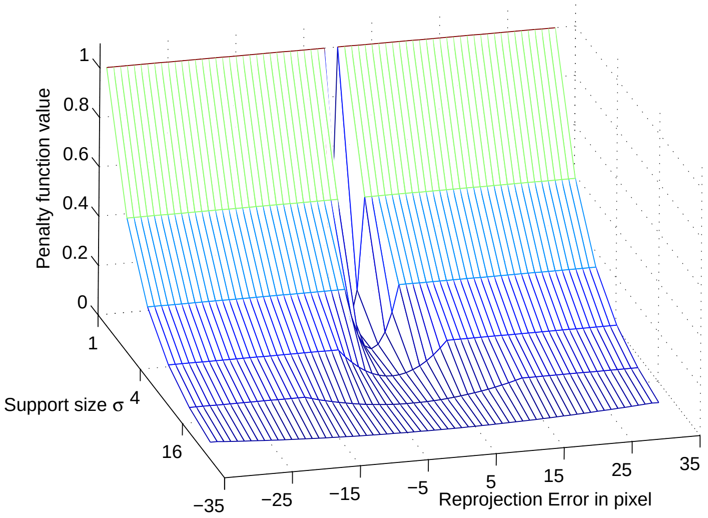


Figure 3.2: The robust estimator that assesses a fixed penalty to residuals larger than a threshold σ.

The robust estimator function $\mathcal{V}(\delta, \sigma)$ that assesses a fixed penalty for residuals larger than a threshold σ is employed in the present work; this approach is relatively insensitive to outliers [15]:

$$\mathcal{V}(\delta, \sigma) = \begin{cases} \frac{\|\delta\|}{\sigma^n}, & M_1 = \{\mathbf{m} | \quad \|\delta\| \leq \sigma^2\} \\ \sigma^{2-n}, & M_2 = \overline{M_1} \end{cases} \tag{3.6}$$

where the set M_1 contains the inlier matches, and M_2 is the set of the outliers. In addition, the order n determines the scale of the residual. As shown in Fig. 3.2, the most correspondences are included when the support σ is large. As σ decreases, the robust estimator becomes narrower and more selective.

Since the robust estimator function is not convex, the associated penalty function approximation problem becomes a hard combinational optimization problem. This problem can be tackled under the finite Newton optimization framework. An augmented vector $\mathbf{t} \in R^N$ containing the barycentric coordinates is defined as below:

$$\mathbf{t}_i = \xi_1 \quad \mathbf{t}_j = \xi_2 \quad \mathbf{t}_k = \xi_3$$

while the remaining elements in the vector $\mathbf{t}$ are all set to zero. Therefore, the residuals for the inlier correspondences can be rewritten as follows:

$$\begin{aligned} \|\boldsymbol{\delta}\| &= (u - \mathbf{t}^\top \mathbf{x})^2 + (v - \mathbf{t}^\top \mathbf{y})^2 \\ &= u^2 + v^2 - 2(u\mathbf{t}^\top \mathbf{x} + v\mathbf{t}^\top \mathbf{y}) + \mathbf{x}^\top \mathbf{t}\mathbf{t}^\top \mathbf{x} + \mathbf{y}^\top \mathbf{t}\mathbf{t}^\top \mathbf{y} \end{aligned}$$

where (u, v) are the coordinates of $\mathbf{m}_1$. Therefore, the error term in Eqn. 3.3 turns out to be

$$E_c = \sum_{\mathbf{m} \in M_1} \frac{\omega_{\mathbf{m}}}{\sigma^n} \left(u^2 + v^2 - 2 \begin{bmatrix} u\mathbf{t} \\ v\mathbf{t} \end{bmatrix}^\top \mathbf{s} + \right.$$
$$\left. \mathbf{s}^\top \begin{bmatrix} \mathbf{t}\mathbf{t}^\top & 0 \\ 0 & \mathbf{t}\mathbf{t}^\top \end{bmatrix} \mathbf{s} \right) + q\sigma^{2-n}$$

where q is the number of outliers.

Let $\mathbf{b} \in R^{2N}$ be defined as below:

$$\mathbf{b} = \begin{bmatrix} \mathbf{b}_x \\ \mathbf{b}_y \end{bmatrix} = \sum_{\mathbf{m} \in M_1} \frac{\omega_{\mathbf{m}}}{\sigma^n} \begin{bmatrix} u\mathbf{t} \\ v\mathbf{t} \end{bmatrix} \tag{3.7}$$

and a matrix $A \in R^{N \times N}$ is equal to

$$A = \sum_{\mathbf{m} \in M_1} \frac{\omega_{\mathbf{m}}}{\sigma^n} \mathbf{t}\mathbf{t}^\top \tag{3.8}$$

Thus, the energy function in Eqn. 3.3 is formulated into an unconstrained quadratic optimization problem, which can be solved by the modified finite Newton method [55, 64].

$$\begin{aligned} E &= \mathbf{s}^\top \begin{bmatrix} \lambda_r K + A & 0 \\ 0 & \lambda_r K + A \end{bmatrix} \mathbf{s} - 2\mathbf{b}^\top \mathbf{s} \\ &+ \sum_{\mathbf{m} \in M_1} \frac{\omega_{\mathbf{m}}}{\sigma^n} (u^2 + v^2) + q\sigma^{2-n} \end{aligned}$$

The finite gradient of the energy function E with respect to $\mathbf{s}$ can be derived as below:

$$\nabla = 2\left(\begin{bmatrix} \lambda_r K + A & 0 \\ 0 & \lambda_r K + A \end{bmatrix} \mathbf{s} - \begin{bmatrix} \mathbf{b}_x \\ \mathbf{b}_y \end{bmatrix} \right) \tag{3.9}$$

and the Hessian [15] can also be computed by

$$H = 2 \begin{bmatrix} \lambda_r K + A & 0 \\ 0 & \lambda_r K + A \end{bmatrix} \tag{3.10}$$

Thus the gradient can be rewritten as below:

$$\nabla = H\mathbf{s} - 2\mathbf{b}$$

Each Newton step will perform the following operation:

$$\mathbf{s} \leftarrow \mathbf{s} - \gamma H^{-1} \nabla \tag{3.11}$$

where γ is the step size. Substituting Eqn. 3.9 into Eqn. 3.11, the equation can be rewritten as follows:

$$\mathbf{s} \leftarrow (1-\gamma)\mathbf{s} + 2\gamma H^{-1}\mathbf{b}$$

Thus, the update equation can be computed by

$$\mathbf{x} \leftarrow (1-\gamma)\mathbf{x} + \gamma[\lambda_r K + A]^{-1}\mathbf{b}_x$$

$$\mathbf{y} \leftarrow (1-\gamma)\mathbf{y} + \gamma[\lambda_r K + A]^{-1}\mathbf{b}_y$$

In the experiment, γ is simply set to one, and no convergence problem occurs in the experiments. Therefore, the new update equation can be derived as below:

$$H\mathbf{s} = \mathbf{b} \tag{3.12}$$

Substituting Eqn 3.10 into Eqn 3.12, the update of the state vector $\mathbf{s}$ can be computed by the following linear equation:

$$\begin{bmatrix} \lambda_r K + A & 0 \\ 0 & \lambda_r K + A \end{bmatrix} \mathbf{s} = \begin{bmatrix} \mathbf{b}_x \\ \mathbf{b}_y \end{bmatrix}$$

Since K is regular, the problem can be further simplified into two linear equations which can be efficiently solved via LU decomposition:

$$\mathbf{s}_\mathbf{x} = (\lambda_r K + A)^{-1}\mathbf{b}_x \tag{3.13}$$

$$\mathbf{s}_\mathbf{y} = (\lambda_r K + A)^{-1}\mathbf{b}_y \tag{3.14}$$

The overall complexity is thus the complexity of one Newton step. Note that the complexity of one step for the proposed method is the same as [48].

3.3.4 Progressive Finite Newton Optimization

Generally speaking, the incorrect matches cannot be avoided in the first stage of the matching process where only local image

```
Algorithm Progressive Finite Newton Approach To Nonrigid Surface Detection
Input
    • Parameters: $\nu$, $\lambda_r$, $\sigma_0$
    • Template image
Pre-compute
1: Build mesh model $\mathbf{s}_0$ for the template image
2: Compute $K$ and $(\xi_1, \xi_2, \xi_3)$ for each keypoint $\mathbf{m}_0$
Nonrigid Surface Detection:
For a given input image
    Obtain $M$ by feature matching
    Select active set by modified RANSAC
    While $\sigma > 2$
        Compute $A$ and $\mathbf{b}$
        Solve linear system: Eqn. 3.13 and Eqn. 3.14
        Calculate residual error $\delta$ and inlier set $M_1$
        $\sigma = \nu \cdot \sigma$
Output
    • mesh vertices $\mathbf{s}$ and total number of inlier matches
End
```

Figure 3.3: Progressive Newton approach to nonrigid surface detection.

descriptors are compared. A coarse-to-fine scheme is introduced to deal with those outliers. The support σ of robust estimator $\mathcal{V}(\delta, \sigma)$ is progressively decayed at a constant rate α. Since the derivatives of $\mathcal{V}(\delta, \sigma)$ are inversely proportional to the support σ, the regularization coefficient λ_r is kept constant during the optimization. For each value of σ, the object function E is minimized through the finite Newton step and the result is employed as the initial state for the next minimization. The minimization of E is directly solved through Eqn. 3.13 and Eqn. 3.14 for a given initial state, and one step is enough to achieve convergence. The optimization procedure stops when σ reaches a value close to the expected precision, which is usually one or two pixels. The algorithm reports a successful detection when the number of inlier matches is above a given threshold. Thus, the whole optimization problem can be solved within a fixed number of steps. This is in contrast to the semi-implicit optimization scheme [76], which involves a few iterations for each σ, and at least 40 iterations in total to ensure the convergence.

In order to select most of the correspondences into the initial active set and avoid getting stuck at local minima, the initial value of σ is usually set to a sufficiently large value. However, this requires a fixed initial state. The method is dependent on the object position, and needs a few iterations to compensate for the errors generated by the pose variations. In the present work, this problem is solved through a modified RANSAC approach. Taking advantage of the concise finite Newton formulation and closed-form solution, the explicit mesh can be directly estimated from a given set of correspondences. Moreover, samples are progressively drawn from larger sets of top-ranked correspondences, which decreases the number of trials significantly. In the experiments, the sampling process stopped within 5 trials. In the worst case, such as when an object does not appear in the scene, it still converges towards RANSAC. Therefore, the output of the proposed progressive sample can be employed as the initial state for the finite Newton optimization. Since the result of progressive sample estimation is quite close to the solution, σ is relatively small. Thus, the proposed progressive scheme requires fewer stages, and is somewhat invariant to the initial position.

From above all, the whole algorithm can be summarized into Fig. 3.3.

3.4 Experimental Results

In this section, we discuss the details of the experimental implementation and report the results of performance evaluation on nonrigid surface detection. It can be concluded that the proposed approach is very efficient for real-time tracking, and can be easily employed for Augmented Reality applications. In addition, same convincing results are obtained for medical image registration, even with missing data.

3.4.1 Experimental Setup

In order to register the mesh model conveniently, a model image is acquired when the nonrigid surface contains no deformation. In order to facilitate real-time Augmented Reality applications, a random-trees based method [58] is used to build the correspondences between the model image and the input image.

Since the number of free variables for nonrigid surface recovery is usually quite large (even up to one thousand), the sample size of each RANSAC iteration becomes a tricky issue. We compare the performance with different sample sizes. In the experiments, the support σ is empirically set to 30, and λ_r is set to a large value to ensure the regularity of the nonrigid surface. Interestingly, the best sample size is found to be three. This is because the nonrigid surface degenerates into a rigid one, and only three points are necessary to determine the position of a rigid surface. Moreover, when the sample size increases, the probability of selecting the inlier data is decreased. Thus, three is the best choice for the sample size.

In the finite Newton optimization, the weighting scheme is beneficial for a single step. However, it changes the scale of the error term in the object function, and so the regularization coefficient λ_r is no longer kept constant during the optimization. In the experiments, all weight coefficients ω are set to one.

A set of synthetic data is used to select the parameters, and the reference mesh is manually registered. The performance is evaluated by the percentage of mesh vertices within 2 pixels of those in the reference mesh. The best regularization coefficient is found to be around 3×10^{-4} by grid searching. Similarly, the initial support σ_0 is set to 80, and decay rate α is 0.5. Fig. 3.4 plots the success probability with different orders n of the robust estimator function. Based on these results, n is set to 4.

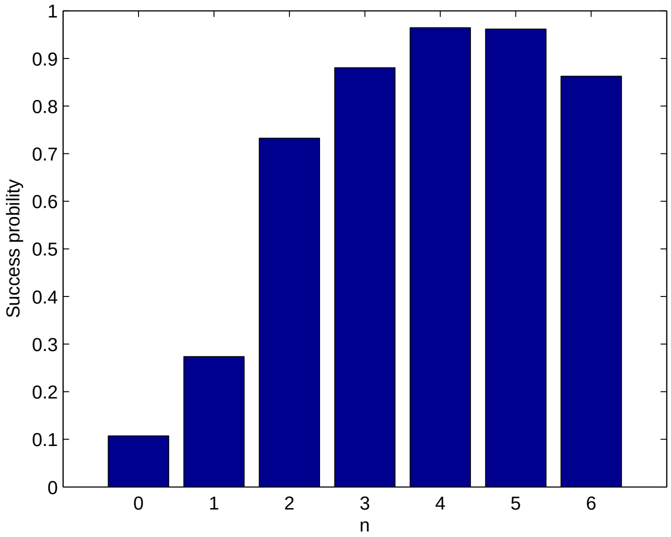
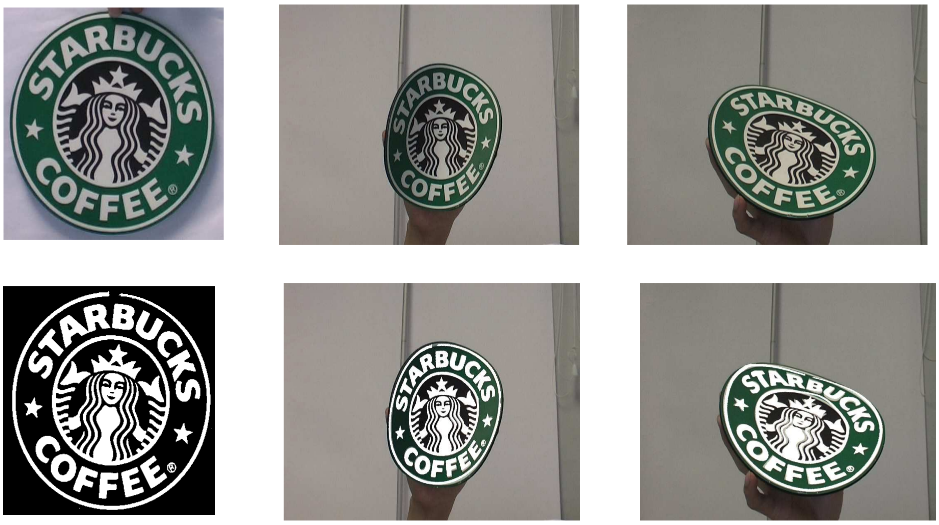


Figure 3.4: Probability of success with different order n of the robust estimator function.

All the experiments reported in this chapter are carried out on a Pentium-4 3.0GHz PC with 1GB RAM, and a DV camera is engaged to capture videos. We also implement a semi-implicit iterative method [76], which is regarded as the state-of-the-art approach.

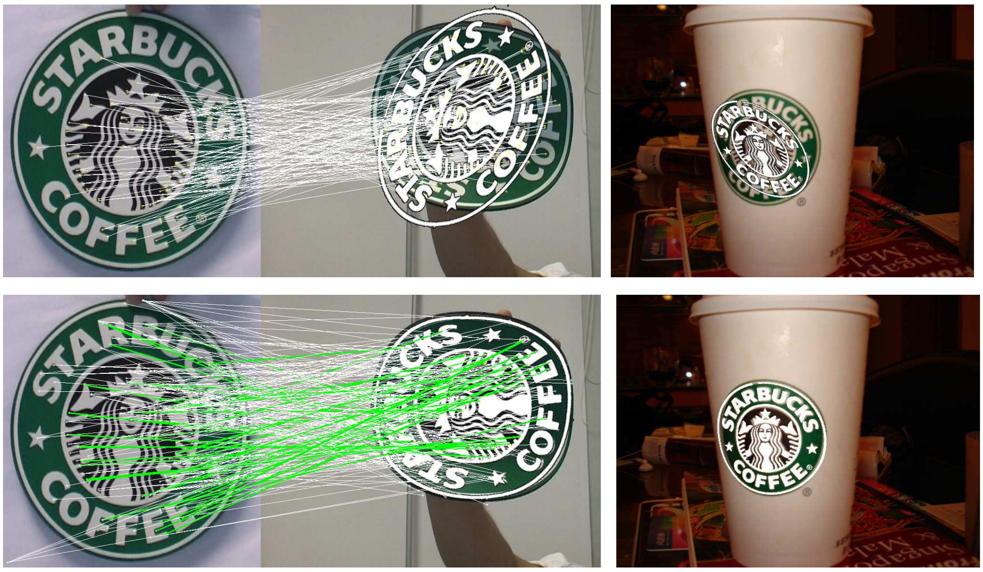


(a) Model image (b) Result (c) Result

Figure 3.5: A Starbucks pad is used as the deformable object. The model image is shown in (a) the contour of the model image is extracted using a simple gradient and filling operator, which is overlaid on the input image. (b) and (c) show the results. The model contains 120 vertices, and the whole process, including image capturing and rendering, runs around 18 frames per second.

3.4.2 Computational Efficiency

The complexity of the proposed method is mainly dominated by the order of Eqn. 3.13 and Eqn. 3.14, which is equal to the number of vertices N in the mesh model. Another important factor is the number of inlier matches, which affects the sparseness of matrix A. This usually differs from one frame to another. For the Starbucks pad with 120 vertices, as shown in Fig. 3.5, the proposed method runs at 18 frames per second on real-time video with the size of 720×576. Fig. 3.6 illustrates the initialization results by the modified RANSAC. We can observe that the proposed method is effective to reject the large outlier matches. Although there are large number of the incorrect matches using the fast random-tree based point matching algorithm [58], which is still far more effective than the conventional normalized cross



(a) Result

(b) Plastic cup

Figure 3.6: The first row shows the initialization results using the modified RANSAC method, and the second row shows the results.

correlation method. As depicted in Table 3.1, the proposed optimization scheme requires around 8 iterations and only takes half of the time of the feature matching algorithm, which is the bottleneck of the whole system. Our implementation [1] of semi-implicit iterative approach [76] needs around 40 iterations to reach the convergence, and runs about 9 frames per second. The improvement is more significant for high resolution mesh. Thus, the proposed method requires far less iterations, and is efficient for real-time applications. We also conduct the experiments without using the modified RANSAC initialization, and start the optimization scheme from a sufficiently large support $\sigma = 1000$. This requires 11 iterations, and the fitting accuracy is worse than the proposed method. In addition, the modified RANSAC initialization can also be used for a semi-implicit method, in which case the number of iterations is reduced to

[1] We use the same parameters setting as [76]. The convergence condition is set to 0.9995, with at most 5 iterations for each support value σ.

around 25.

Table 3.1: Computational time of proposed method at each step.

Total	Match	Optimization	Iteration	Other
57ms	27ms	14ms	$\sim$ 1.9ms	16ms

3.4.3 Performance of Nonrigid Surface Recovery

A Starbucks pad is employed as the deformable object. As shown in Fig. 3.5, the proposed method is robust to large deformations and perspective distortion. In practice, the whole process runs at around 18 frames per second. Fig. 3.8 describes the result of detecting a piece of paper, where similar performance is achieved. As another feature-based method, the performance of the proposed method is closely related to the texture of objects. Better results can be obtained for objects with more texture, because it is easy to find more correct correspondences than with those lacking texture. This problem can be dealt with via incorporating global appearance into the optimization scheme, which will be presented in the next chapter.

3.4.4 Augmented Reality

Once the nonrigid surface is recovered, an immediate application is to re-texture an image. In order to obtain realistic results, the texture should be correctly relighted. As suggested in [76], a re-textured input image is generated by directly multiplying a blank shaded image, which is the quotient of the input image and the warped reference image. The reference image is acquired when the nonrigid surface is lighted uniformly. Moreover, the quotient image is normalized through multiplying the intensity of white color in the reference image. This relighting procedure is easily done by the GPU and requires only a short OpenGL

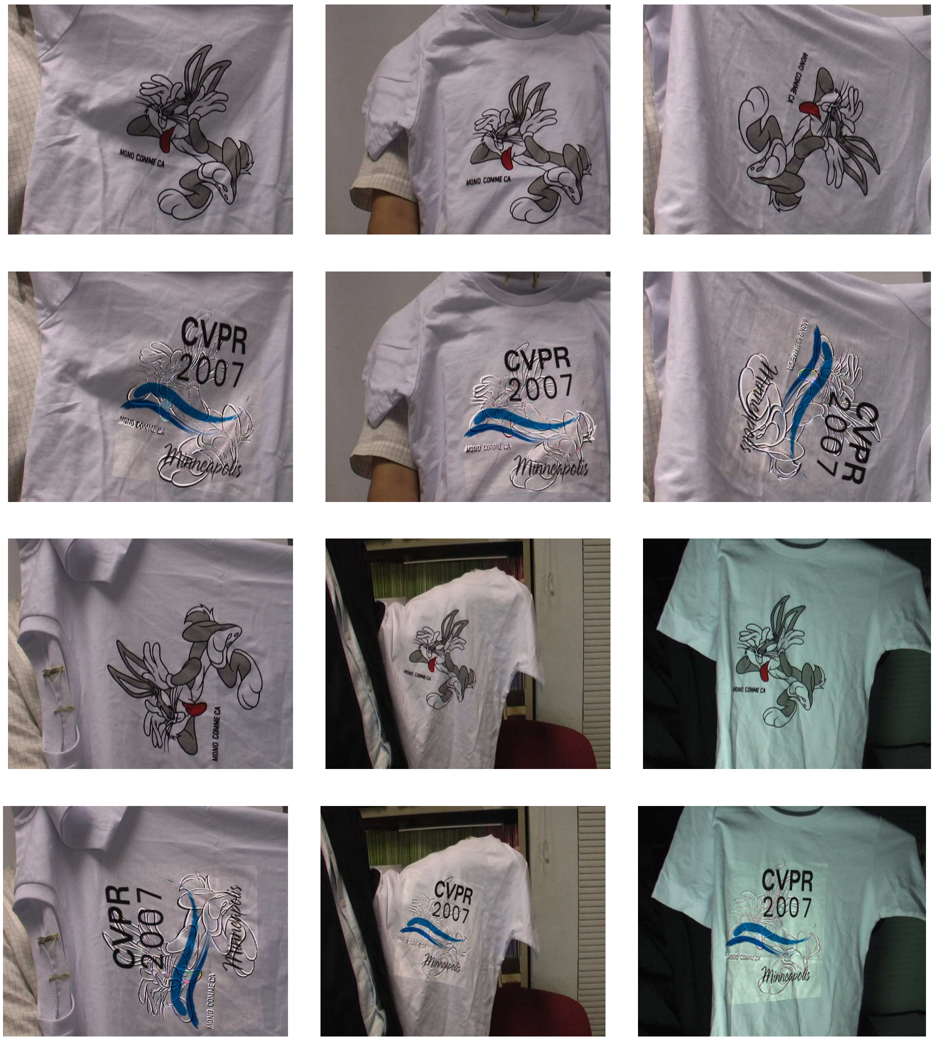


Figure 3.7: Re-texturing of a shirt print. The first row and third row show the 720 × 576 images captured by a DV camera. The second row and forth row show the results of replacing the bunny with the CVPR logo.

shading language program; and the whole process runs at about 17 frames per second. Fig. 3.7 shows the results of re-texturing a T-shirt with a Lambertian surface. It is difficult to estimate a blank shaded image due to dividing near zero intensity values and the use of an uncontrolled optical sensor. However, the visual effect is that the bunny in the input video is re-textured by the CVPR logo. For a specular surface, Fig. 3.8 describes the results on a piece of paper with a saturated region. In addition, the right two columns of Fig. 3.8 show the results in a cluttered environment.

3.4.5 Medical Image

The proposed approach is also evaluated for medical image registration. A pair of sagittal images [74] with size of 256×256 from two different patients are used in the experiments. The source and target images differ in both geometry and intensity. The results are plotted in Fig. 3.9; it can be seen that the source image is successfully registered. In comparison with the locally affine but globally smooth method [74], which takes about 4 minutes, the proposed method only needs 0.2 seconds. Moreover, the sparse correspondences based method can naturally handle the missing data and partial occlusion problem. As shown in Fig. 3.10, the source images with a region removed, the nonrigid shape can still be recovered.

3.5 Discussions

A novel scheme has been proposed for non-rigid surface detection by progressive finite Newton optimization. In comparison with semi-implicit optimization methods [76], the proposed method has several advantages. First, the presented method needs not solve the optimization iteratively for every σ, be-

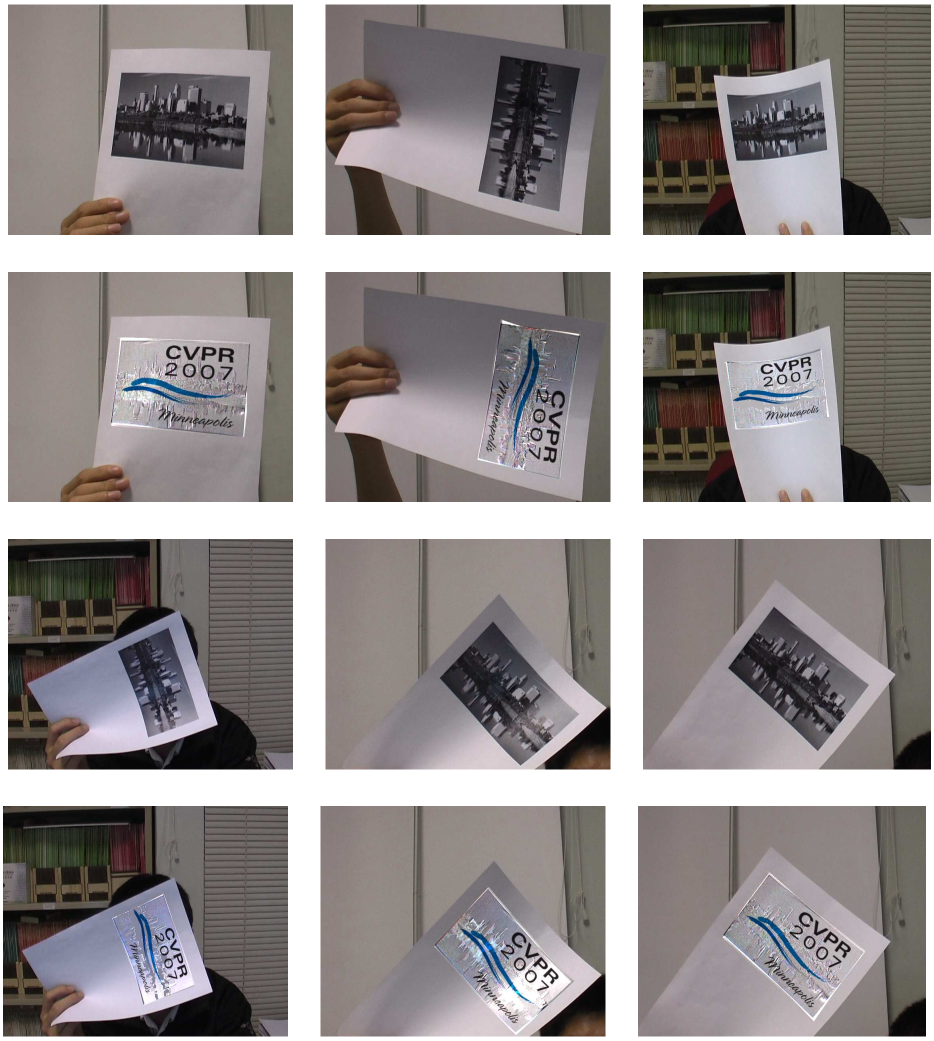
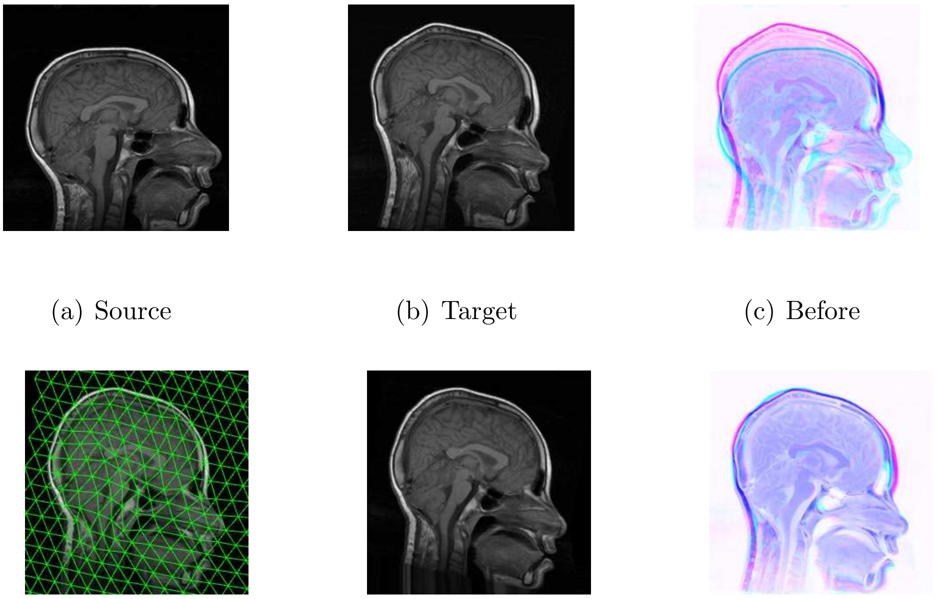


Figure 3.8: Re-texturing a picture on a piece of paper. The first row is the 720 × 576 images captured by a DV camera. The second row is the results of replacing the picture with the CVPR logo.



(a) Source

(b) Target

(c) Before

(d) Registered

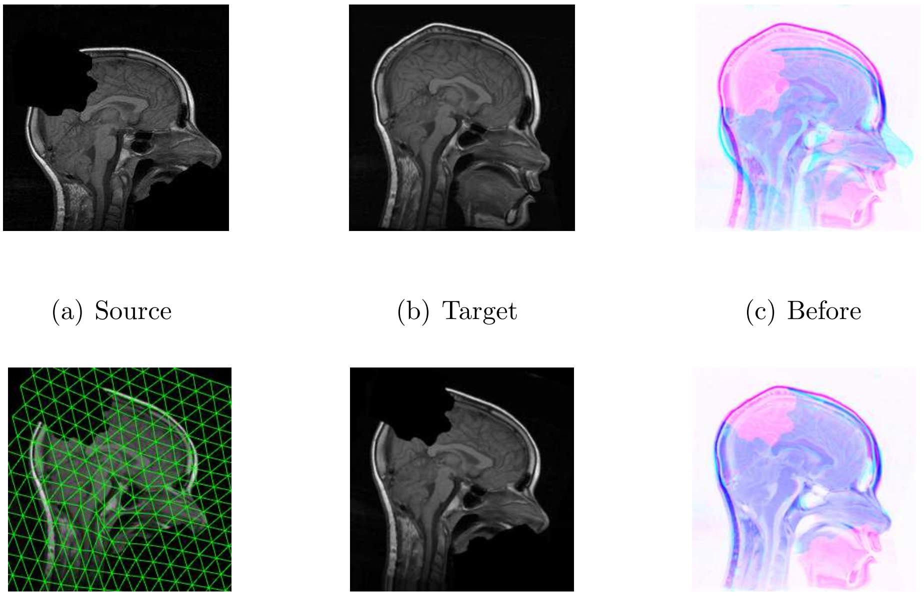
(e) Registered

(f) After

Figure 3.9: Applying the proposed method to medical image. A pair of sagittal images from two different patients is shown. (a,b,e) are the source, target and registered source respectively. (d) is the registered source with mesh model. (c) and (f) are the overlaid images before and after registration.

cause it can be solved in one step directly. Second, the iterative method starts from a sufficiently large support value in order to estimate the location and pose of an object, which leads to a large number of iterations. Thus, the proposed method is far more efficient than the semi-implicit method. Additionally, it is easy to implement the proposed approach, which only involves solving the sparse linear equation, and does not require tuning the viscosity parameters and a sophisticated Levenberg-Marquardt optimization algorithm.

Although promising experimental results have validated the efficiency of the methodology, some limitations should be addressed. First of all, some jitter and errors may occur due to the point matching algorithm or the lack of texture information,



(a) Source (b) Target (c) Before

(d) Registered (e) Registered (f) After

Figure 3.10: Applying the proposed method to medical image registration for the synthetic example with missing data.

as shown in Fig. 3.11. Second, the proposed method may fail in the case of severe folding and self-occlusion. Two failure examples are depicted in Fig. 3.11. Also, the presented method is mainly focused on single deformable surface detection, whereas it is also interesting to study the multiple surfaces case.

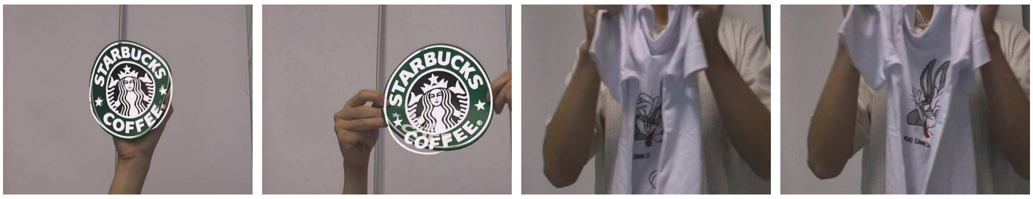


Figure 3.11: Results with large errors and some failure cases.

3.6 Summary

This chapter presents a novel progressive scheme to solve the non-rigid surface detection problem. In contrast to the previous approaches involving iterative and explicit minimization, a progressive finite Newton algorithm is proposed, which directly solves the unconstrained quadratic optimization problem by an efficient factorization method. Moreover, the modified RANSAC scheme takes advantage of the concise formulation and progressive sampling of the top-ranked correspondences, and can handle high-dimensional spaces with noisy data.

Extensive experimental evaluations have been conducted on diverse objects with different materials. The proposed method is very fast and robust, and can handle large deformations and illumination changes. It has been tested in several applications, such as real-time Augmented Reality and medical image registration. The promising experimental results show that the algorithm is more efficient than previous methods.

□ **End of chapter.**

Chapter 4

Fusing Features and Appearance

In this chapter, we present a fusion approach to tackle the nonrigid shape recovery problem, which takes advantage of both the appearance information and the local features. This method can greatly reduce the jittering problem in the previous chapter. Moreover, a deformable Lucas-Kanade algorithm is proposed, which triangulates the template image into small patches and constrains the deformation through the second order derivatives of the mesh vertices. It is further formulated into a sparse regularized least squares problem which is able to reduce the computational cost and the memory requirement. The inverse compositional algorithm is applied to efficiently solve the optimization problem.

4.1 Motivation and Methodology

Image alignment or registration has been an important research topic in computer vision for the past few decades [2], finding a variety of applications in object tracking [70, 76], facial image analysis [23, 65, 105], medical imaging and digital entertainment.

Most of the current nonrigid shape recovery methods can be divided into two categories. The first is dependent on local feature correspondences [75, 82]. The second is based on the appearance, which directly minimizes the residual image between the synthesized template image and the input image [23, 83, 116].

As for the feature-based image alignment methods, we have already throughly reviewed them in Chapter 3. The major limitation of these methods is that they are dependent on the geometric locations of a set of carefully selected salient features, which do not always cover the whole image. Therefore, it is difficult to guarantee the registration accuracy in the regions lacking texture. In addition, it is still hard to handle the feature matching problem involving with large deformations and severe perspective distortions [60].

On the other hand, the appearance-based approach can exploit more of the texture information, and therefore achieves better registration accuracy. In fact, a large number of the appearance-based methods [23, 28, 65, 83] can be viewed as extensions of the original Lucas-Kanade algorithm [2, 3] which has been one of the most widely used techniques in computer vision. These approaches directly minimize the residual image between the input image and the synthesized model image [23, 65].

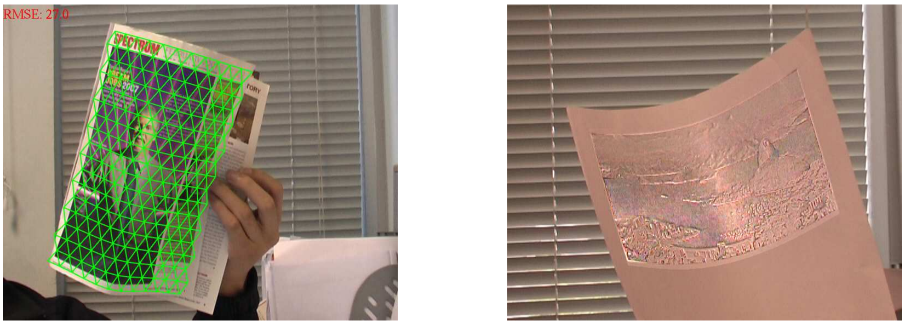
An inverse compositional method [2] has recently been proposed to efficiently solve the optimization problem in the conventional Lucas-Kanade algorithm, reducing the computational cost by pre-computing the Hessian matrix. The original Lucas-Kanade algorithm [4, 11, 63] for image alignment usually estimates either the affine transformation or the homography between the template image and the input image. In order to handle the image alignment problem involved in the deformations, such as facial feature movements, the Lucas-Kanade algorithm has been extended to incorporate linear shape and appearance variations [65]. This extension has been referred to Active Appearance Model. In [34], a feature-driven method is described to make use of the compositional algorithms for the parametric warps. In addition, optical flow information [6, 28] can be incorporated into the optimization scheme to obtain better results. The major limitation of these methods is that they tend to become stuck at a local minimum and hence require good initialization.

As described in Section 2.4.2, Active Appearance Model [23, 65] requires building the statistical shape and appearance models from manually annotated examples. However, relatively few of the appearance-based methods can handle deformable objects using a single template. One such method is the Active blob [83], which mainly employs the Finite Element Model to build the shape variations. Another, proposed by Gumerov et al. [37], requires that the whole outline can be detected. In a more recent study, the repeating properties of a near regular texture are exploited to track new texture tiles in video frames [59].

As discussed in Chapter 2, the appearance-based method tends to be computationally expensive and requires good initialization to avoid the local optima, and only a few automated solutions have been proposed in the literature. Since both the feature and appearance based methods have limitations, there is a need for an automated method which can make use of both the appearance information and the local features.

In this chapter, we propose a novel automated approach to efficiently handle the image alignment with very large non-affine deformations, as shown in Figure 4.1. The major contribution of this chapter is the proposed deformable Lucas-Kanade algorithm, which triangulates the template image into small patches and preserves the regularity of the mesh through the second order derivatives of the mesh vertices. Moreover, the optimization of the proposed deformable Lucas-Kanade algorithm is formulated into a sparse regularized least squares problem, which is able to reduce the computational cost and the memory requirement. The inverse compositional algorithm [2] is applied to efficiently solve the optimization problem. Furthermore, the optimization for the fusion approach is solved through a modified deformable Lucas-Kanade algorithm.

The rest of this chapter is organized as follows. In Section 4.2, we present the proposed fusing features and appearance approach for nonrigid shape recovery. Section 4.3 provides the details of the experimental implementation and describes the experimental results. We discuss limitations of the approach in Section 4.4. Section 4.5 summarizes this chapter.



(a) Cover

(b) Paper

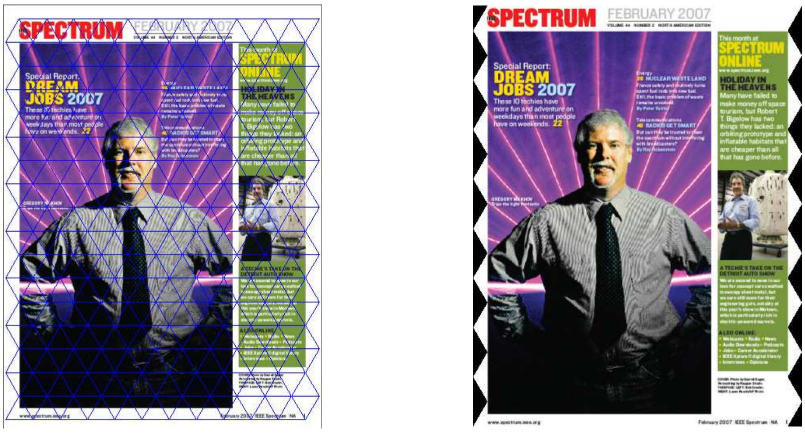
Figure 4.1: Recovering nonrigid shapes. (a) The cover of a magazine. (b) A piece of paper.

4.2 Fusing Features and Appearance

4.2.1 Overview

In this section, we describe the fusion approach to dealing with the nonrigid shape recovery, which takes advantage of both the local features and appearance information. For tackling the challenges, the 2D nonrigid shape model in Chapter 3 is introduced. The proposed algorithm is formulated into an optimization problem which minimizes the correspondence error, the texture difference and the surface energy. The key of the fusion approach is to solve this problem in the following. First, we employ the progressive finite Newton method using the feature correspondences to detect the nonrigid surface. Then, a novel deformable Lucas-Kanade algorithm is proposed to handle the appearance error. Based on these two algorithms, the optimization scheme for the fusion approach is formulated.

4.2.2 Mesh Model



(a) Model mesh $\mathbf{s}_0$

(b) Reference image

Figure 4.2: (a) The mesh model with 216 vertices and 374 triangles. (b) The reference image size of 403 × 516.

As described in Section 3.3 of Chapter 3, the nonrigid shape is explicitly represented by triangulated meshes, which is shown in Fig. 4.2(a). Instead of treating the template image as a whole block, as in [4, 63], this 2D deformable mesh model is employed to triangulate it into small patches, as shown in Fig. 4.2(a). Then, the mesh associated with the model image is defined as the reference mesh $\mathbf{s}_0$. Also, the piecewise affine warp $W(\mathbf{m}, \mathbf{s})$ defined in Eqn. 3.1 is used to map the input image into the reference frame $\mathbf{s}_0$. Fig. 4.2(b) shows an example of the template image in the reference frame.

Based on this triangulated mesh model, we describe in detail the proposed approach to nonrigid shape recovery in the following.

4.2.3 Fusing Features and Appearance Approach

Proposed Algorithm

The aim of the fusion approach is to make use of both the local features and the appearance information.

- **Local feature correspondences.** The correspondence error term $E_c(\mathbf{s})$ is the sum of the weighted square error residuals for the matched points, which is introduced in Section 3.3.

- **Appearance.** In this chapter, we try to handle the appearance error under the Lucas-Kanade framework. The objective of the Lucas-Kanade algorithm is to minimize the sum of the squared errors between the template image T and the input image I warped back onto the coordinate frame of the template. Baker and Matthews [2] have proposed an inverse compositional algorithm which switches the role of the template image T and input image I in the computation of the incremental warp. Using this approach, the computational cost can be reduced by pre-computing the Hessian matrix. Instead of using the affine transformation or homography, as in [4, 63], we directly employ the parameterization of the mesh model vertices $\mathbf{s}$ in this paper. Due to the direct parameterization, $\Delta\mathbf{s}$ is defined as the increments to the mesh vertices. The inverse compositional method is employed to formulate the energy for the appearance E_a. Following the notation in [2, 65], E_a is defined as follows:

$$E_a(\mathbf{s}) = \sum_{\mathbf{x}} [T(W(\mathbf{x}; \Delta\mathbf{s})) - I(W(\mathbf{x}; \mathbf{s}))]^2 \tag{4.1}$$

In general, the nonrigid shape recovery problem approximates a 2D mesh with $2N$ free variables, which is usually ill-posed. One effective way to attack this problem is to introduce regularization, which preserves the regularity of a deformable surface. This leads to the following energy function:

$$E(\mathbf{s}) = E_a(\mathbf{s}) + \alpha E_c(\mathbf{s}) + \lambda_r E_r(\mathbf{s}) \tag{4.2}$$

where α is a weight coefficient, and λ_r is a regularization coefficient. The regularization term $E_r(\mathbf{s})$ is composed of the sum of the squared second-order derivatives of the mesh vertex coordinates, which is defined in Eqn. 3.4.

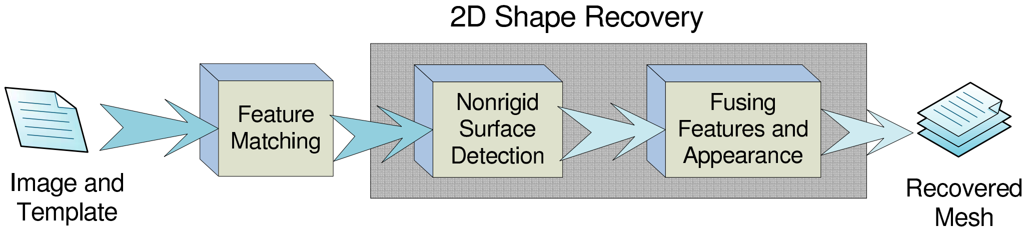


Figure 4.3: Overview of the 2D shape recovery algorithm.

Optimization Framework

To enable an automated solution, we employ the result of minimizing the feature correspondences error to initialize the optimization for the fusion approach. This is because $E_c(\mathbf{s})$ is independent of the image during the optimization, and so it can be computed very efficiently. More specifically, the initial result is obtained by the nonrigid surface detection method, which deals with the following energy minimization problem:

$$E_F(\mathbf{s}) = E_c(\mathbf{s}) + \lambda_r E_r(\mathbf{s}) \tag{4.3}$$

The details of the solution for the above optimization problem have been described in Section 3.3.

In this chapter, the optimization for the fusion approach is based on the Lucas-Kanade framework. To simplify the formulation, we start from only taking consideration of the texture difference $E_a(\mathbf{s})$. Also, the regularization term $E_r(\mathbf{s})$ is introduced to preserve the surface regularity. Thus, the following regularized least squares problem can be obtained:

$$E_A(\mathbf{s}) = E_a(\mathbf{s}) + \lambda_r E_r(\mathbf{s}) \tag{4.4}$$

We name this approach the deformable Lucas-Kanade algorithm, which can be used to solve the optimization for the fusion approach with slight modification.

Therefore, the essence of the fusion approach is to first detect the nonrigid shape using feature correspondences, and then solve the fusion optimization based on the modified deformable Lucas-Kanade algorithm. We will describe it in detail in the following subsections. The overview of the method is shown in Fig. 4.3 where each step is highlighted using a shaded box.

4.2.4 Deformable Lucas-Kanade Algorithm

Taking consideration of appearance information only, we try to solve the optimization problem in Eqn. 4.4 using the inverse compositional method. The deformable Lucas-Kanade algorithm is summarized in Fig. 4.4.

The warp update equation can be defined as follows:

$$W(\mathbf{x}) \leftarrow W(\mathbf{x}) \circ W(\mathbf{x}; \Delta\mathbf{s})^{-1}$$

Performing the first order Taylor expansion on the Eqn. 4.4 gives:

$$\sum_{\mathbf{x}} \left[T(W(\mathbf{x}; \mathbf{s}_0)) + \nabla T \frac{\partial W}{\partial \mathbf{s}} \Delta\mathbf{s} - I(W(\mathbf{x}; \mathbf{s})) \right]^2 + \lambda_r (\mathbf{s} + \Delta\mathbf{s})^\top \mathcal{K} (\mathbf{s} + \Delta\mathbf{s}) \tag{4.5}$$

```
Algorithm Deformable Lucas-Kanade Algorithm
Input
    • Parameters: $\mathbf{s}_0$, $\lambda_r$
    • Template image $T$
Pre-compute
    • 1: Build mesh model $\mathbf{s}_0$ for the template image
    • 2: $K$, $\nabla T \frac{\partial W}{\partial \mathbf{s}}$, and the Hessian matrix $H_2$
Iterate:
    Warp $I$ with $W(\mathbf{x};\mathbf{s})$ to compute $I(W(\mathbf{x};\mathbf{s}))$
    Compute the residual image $I(W(\mathbf{x};\mathbf{s})) - T$
    Compute $\Delta\mathbf{s}$ using Eqn. 4.6
    Update the shape vector $\mathbf{s} \leftarrow \mathbf{s} - \Delta\mathbf{s}$
Until $\|\Delta\mathbf{s}\| < threshold$
Output
    • mesh vertices $\mathbf{s}$
End
```

Figure 4.4: Deformable Lucas-Kanade Algorithm.

where ∇T is the gradient of the template image evaluated at $W(\mathbf{x};\mathbf{s}_0)$, and $\frac{\partial W}{\partial \mathbf{s}}$ is the Jacobian of the warp parameters evaluated at $\mathbf{s}$. Note that $\nabla T \frac{\partial W}{\partial \mathbf{s}}$ is the gradient, and $\mathbf{s}_0$ is the reference mesh shown in Fig. 4.2(a).

Assuming that $W(\mathbf{x};\mathbf{s}_0)$ is the identity warp, the gradient of Eqn. 4.5 with respect to $\Delta\mathbf{s}$ can be derived as below:

$$\sum_{\mathbf{x}} \left[\nabla T \frac{\partial W}{\partial \mathbf{s}}\right]^{\top} \left[T(W(\mathbf{x};\mathbf{s}_0)) + \nabla T \frac{\partial W}{\partial \mathbf{s}} \Delta\mathbf{s} - I(W(\mathbf{x};\mathbf{s}))\right] + \lambda_r \mathcal{K}(\mathbf{s} + \Delta\mathbf{s})$$

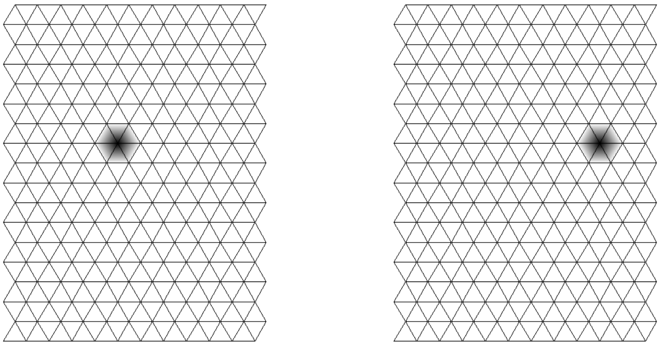
As the above gradient vanishes for optimality, this leads to the following closed-form solution:

$$\Delta \mathbf{s} = H_2^{-1} \sum_{\mathbf{x}} \left[\nabla T \frac{\partial W}{\partial \mathbf{s}} \right]^\top \left[I(W(\mathbf{x};\mathbf{s})) - T(W(\mathbf{x};\mathbf{s}_0)) \right] - \lambda_r H_2^{-1} \mathcal{K} \mathbf{s} \quad (4.6)$$

where H_2 is the $2N \times 2N$ Hessian matrix:

$$H_2 = \sum_{\mathbf{x}} \left[\nabla T \frac{\partial W}{\partial \mathbf{s}} \right]^\top \left[\nabla T \frac{\partial W}{\partial \mathbf{s}} \right] + \lambda_r \mathcal{K} \quad (4.7)$$

Note that the Hessian matrix H_2 is independent of the parameter vector $\mathbf{s}$, and H_2 is kept constant across iterative optimization and can be pre-computed. It is also independent of the gradient matrix $\nabla T \frac{\partial W}{\partial \mathbf{s}}$. Therefore, the warp update of the shape parameters $\Delta \mathbf{s}$ can be computed very efficiently.



(a) $\frac{\partial W}{\partial x_1}$ and $\frac{\partial W}{\partial y_1}$

(b) $\frac{\partial W}{\partial x_2}$ and $\frac{\partial W}{\partial y_2}$

Figure 4.5: The Jacobian $\frac{\partial W}{\partial x_i}$ with respect to the mesh vertices (x_1, y_1) and (x_2, y_2). Only the non-zero part is plotted, and the inverted images are used for better illustration.

Since the coordinates of the mesh vertices $\mathbf{s}$ are directly employed as the warp parameter in the deformable Lucas-Kanade algorithm, the computation of the warp inversion becomes much easier than the linear combination model method in Active Appearance Model [65]. Specifically, the shape vector $\mathbf{s}$ is updated by:

$$\mathbf{s} \leftarrow \mathbf{s} - \Delta\mathbf{s} \tag{4.8}$$

Link to Lucas-Kanade algorithm: The proposed method can be viewed as a natural extension of the Lucas-Kanade algorithm, which is able to handle the deformations rather than the affine transformation. Since the 2D coordinates of the mesh vertices $\mathbf{s}$ is employed as the parameters in deformable Lucas-Kanade algorithm, the degree of freedom is increased. This is useful for handling the image alignment when the deformation is large. Furthermore, the efficient optimization methods for the Lucas-Kanade algorithm [2] can also be applied for the proposed method.

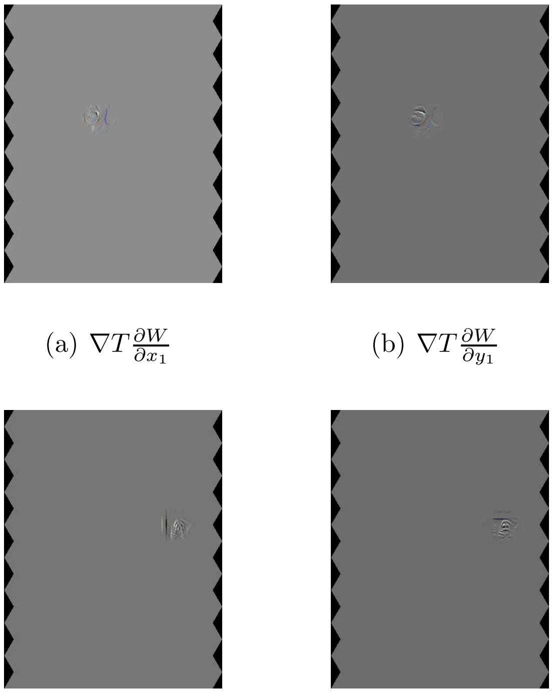
Link to Active Appearance Model [65]: The deformable Lucas-Kanade algorithm can be treated as a kind of Active Appearance Model. It employs a single training example along with certain physical constraints, while Active Appearance Model needs to build both texture and shape models to constrain the searching space.

Computing Gradient $\nabla T\frac{\partial W}{\partial \mathbf{s}}$

Recall that the destination of the pixel $\mathbf{x}$ under the piecewise affine warp $W(\mathbf{x};\mathbf{s})$ depends on the vertices of the mesh $\mathbf{s}$. According to the definition of $W(\mathbf{x};\mathbf{s})$ in Eqn. 3.1, the Jacobian of the warp $W(\mathbf{x};\mathbf{s})$ with respect to the mesh vertices $\mathbf{v}(x_i, y_i)$ can be derived as below:

$$\frac{\partial W}{\partial x_i} = \begin{bmatrix} \xi_1 & 0 \end{bmatrix}^\top \qquad \text{and} \qquad \frac{\partial W}{\partial y_i} = \begin{bmatrix} 0 & \xi_1 \end{bmatrix}^\top$$

It can easily be found that the non-zero parts of $\frac{\partial W}{\partial x_i}$ and $\frac{\partial W}{\partial y_i}$ are equal. As shown in Fig. 4.5, the Jacobians can be illustrated as the images with the same size of reference frame; in fact, each image is the Jacobian with respect to the vertex $\mathbf{v}$. Moreover, it can also be observed that the warp Jacobian is quite sparse,



(a) $\nabla T \frac{\partial W}{\partial x_1}$

(b) $\nabla T \frac{\partial W}{\partial y_1}$

(c) $\nabla T \frac{\partial W}{\partial x_2}$

(d) $\nabla T \frac{\partial W}{\partial y_2}$

Figure 4.6: The gradient $\nabla T \frac{\partial W}{\partial \mathbf{s}}$ with respect to the mesh vertices (x_1, y_1) and (x_2, y_2).

having non-zero values only in the triangles around the vertex $\mathbf{v}$. Next, we compute $\nabla T \frac{\partial W}{\partial \mathbf{s}}$ by multiplying the gradient of the template image with the warp Jacobian matrix, resulting in the images plotted in Figure 4.6.

Remark Since the dimensionality of the texture is usually very high, the gradient $\nabla T \frac{\partial W}{\partial \mathbf{s}}$ becomes quite a large matrix. Fortunately, both the gradient and the Hessian H_2 are the sparse matrices in the proposed deformable Lucas-Kanade algorithm, and this can greatly reduce the computational cost and memory requirement and make the problem tractable. Moreover, this also leads to a sparse regularized least squares problem in Eqn. 4.4.

Lighting

In order to minimize the effect of global lighting variation, we apply a scaling a and an offset o to the template image T. Therefore, the energy function of the proposed deformable Lucas-Kanade algorithm can be rewritten as follows:

$$\sum_{\mathbf{x}}[aT(W(\mathbf{x};\Delta\mathbf{s})) + o \cdot \mathbf{1} - I(W(\mathbf{x};\mathbf{s}))]^2 + \lambda_r \mathbf{s}^\top \mathcal{K}\mathbf{s} \tag{4.9}$$

Similarly, we can employ an extended inverse compositional algorithm [4, 5] to solve this optimization problem. See Appendix A for details.

4.2.5 Fusion Approach Optimization

Based on the deformable Lucas-Kanade algorithm, we describe the optimization scheme for the proposed fusion approach in Section 4.2.3.

Let us define a matrix $B \in R^{2N\times 2N}$, which is equal to

$$B = \frac{1}{\sigma^n}\begin{bmatrix} A & 0 \\ 0 & A \end{bmatrix} \tag{4.10}$$

Therefore, we can rewrite E_c as below:

$$E_c = \mathbf{s}^\top B\mathbf{s} - 2\mathbf{b}^\top\mathbf{s} + q\sigma^{2-n} + \sum_{\mathbf{m}\in M_1} \frac{1}{\sigma^n}(u^2+v^2)$$

Performing the first order Taylor expansion on the energy function Eqn. 4.2 gives:

$$\begin{aligned}&\sum_{\mathbf{x}}\left[T(W(\mathbf{x};\mathbf{s}_0)) + \nabla T\frac{\partial W}{\partial \mathbf{s}}\Delta\mathbf{s} - I(W(\mathbf{x};\mathbf{s}))\right]^2 \\ &+\alpha(\mathbf{s}+\Delta\mathbf{s})^\top B(\mathbf{s}+\Delta\mathbf{s}) - 2\alpha\mathbf{b}^\top(\mathbf{s}+\Delta\mathbf{s}) + q\sigma^{2-n} \\ &\quad + \sum_{\mathbf{m}\in M_1}\frac{1}{\sigma^n}(u^2+v^2) + \lambda_r(\mathbf{s}+\Delta\mathbf{s})^\top\mathcal{K}(\mathbf{s}+\Delta\mathbf{s})\end{aligned} \tag{4.11}$$

The solution to this problem is:

$$\Delta\mathbf{s} = H_3^{-1}\sum_{\mathbf{x}} \left[\nabla T \frac{\partial W}{\partial \mathbf{s}}\right]^{\top} [I(W(\mathbf{x};\mathbf{s})) - T(W(\mathbf{x};\mathbf{s}_0))] \\ -\alpha H_3^{-1}(B\mathbf{s} - \mathbf{b}) - \lambda_r H_3^{-1}\mathcal{K}\mathbf{s} \quad (4.12)$$

where H_3 is the Hessian matrix:

$$H_3 = \sum_{\mathbf{x}} \left[\nabla T \frac{\partial W}{\partial \mathbf{s}}\right]^{\top} \left[\nabla T \frac{\partial W}{\partial \mathbf{s}}\right] + \alpha B + \lambda_r \mathcal{K} \quad (4.13)$$

Again, we can compute the warp update through Eqn. 4.8.

In order to reduce the computational cost, the gradient and part of the Hessian for the deformable Lucas-Kanade algorithm is pre-computed. Since the inlier set is slightly changed in the fusion optimization phase, matrix B can be viewed as a constant. Therefore, the Hessian H_3 is computed once for each input image through Eqn. 4.13. The optimization procedure stops when $\|\Delta\mathbf{s}\|$ is close to the given threshold or the number of iterations exceeds the limit.

To tackle the lighting variations, we only need to make slight modification on the method described in Fig 4.4. Specifically, we add the initialization step, and pre-compute the matrix B and H_3 for each input image. Furthermore, Eqn. 4.12 is employed to compute the update for the shape vector $\mathbf{s}$.

4.3 Experimental Results

In this section, we discuss the details of the experimental implementation and report the results of performance evaluation on nonrigid shape recovery. First, the various evaluations are performed on the the deformable Lucas-Kanade algorithm. Then, the fusion approach is tested.

4.3.1 Experimental Setup

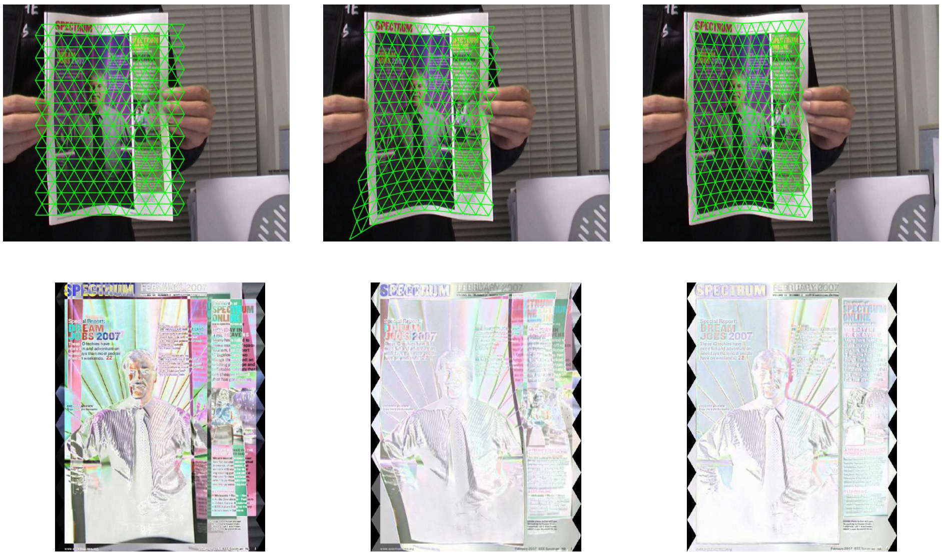
In order to register the mesh model conveniently, a model image is acquired when the nonrigid surface contains no deformation. In order to facilitate real-time augmented reality applications, a random-trees based method [58] is used to build the correspondences between the model image and the input image. The semi-implicit iterative method [76] is implemented as the state-of-the-art approach. All the experiments reported in this paper are carried out on a Pentium-4 3.0GHz PC with 1GB RAM, and a DV camera was engaged to capture videos.

4.3.2 Evaluation on the Deformable Lucas-Kanade Algorithm

Deformable Lucas-Kanade Fitting

The parameters for deformable Lucas-Kanade algorithm are found by grid searching; and the regularization parameter λ is set to 10^5. Moreover, the texture mapping is efficiently performed by OpenGL. Fig. 4.7 shows an example of the proposed deformable Lucas-Kanade algorithm fitting to a single image, which employs the template image and mesh model illustrated in Fig. 4.2. Fig. 4.7(a) displays the initial configuration, Fig. 4.7(b) the result after 30 iterations, and Fig. 4.7(c) the final converged result after 58 iterations.

The conventional Lucas-Kanade algorithm is also evaluated with the inverse compositional method, using the same initial position as the method. However, it fails to converge in this case due to the the large non-affine deformation. Fig. 4.8 plots the root mean square error (RMSE) curve for the proposed method. In the case of the deformable Lucas-Kanade algorithm, the RMSE is relatively large (28.5), which is mainly due to the difference between the optical sensor and the printing device. However, it can be observed that the mesh is accurately reg-



(a) Initialized (b) 30 iterations (c) Converged

Figure 4.7: An example of the deformable Lucas-Kanade fitting to a single image. The first row is the result mesh overlaid on the input image. The second row displays the residual images; the inverted image is used for better illustration.

istered on the input image in Fig. 4.7(c). Since the lighting variations are considered in the proposed method, the RMSE drops rapidly in the first few iterations .

Computational Efficiency

The complexity of the proposed method is mainly dominated by the size of the template image and the number of the vertices N in the mesh model. Another factor is the number of inlier feature matches, which affects the sparseness of matrix B. In the experiments, three models are built to perform the evaluation, as summarized in Table 4.1. The mesh model C_1 is shown in Fig. 4.2. C_2 is obtained by increasing the edge length, giving fewer mesh vertices than C_1. C_3 is built by reducing both the template image and the mesh size to 75% of C_1. We evaluate the computational cost of the proposed method for the

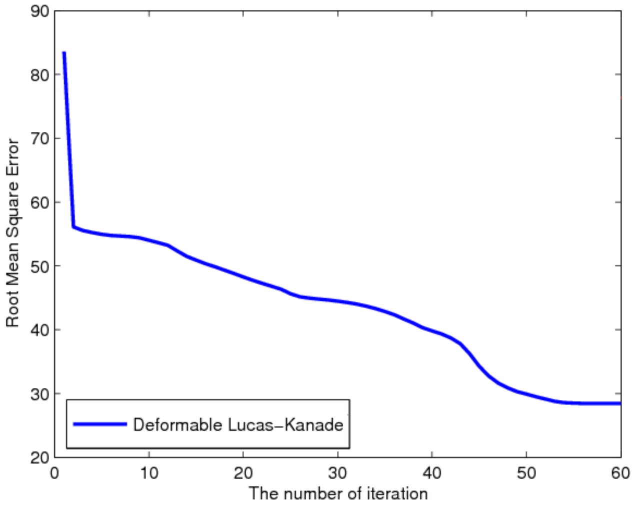
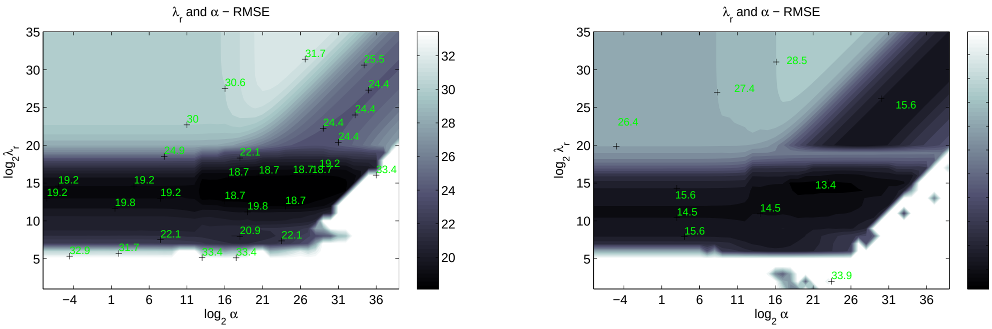


Figure 4.8: The Root Mean Square Error between the template and input images against the number of iterations.

nonrigid surface recovery task on realtime videos with the size of 720 × 576. Table 4.1 summarizes the experimental results on different models. We observe that the dimensionality of the appearance determines the time complexity of the deformable Lucas-Kanade algorithm. Therefore, gray images are easier to track. The number of mesh vertices N has a great influence on the initialization step, but posts a limited impact on the computational time in the optimization.

Table 4.1: Computational time of the deformable Lucas-Kanade algorithm on different 2D mesh models.

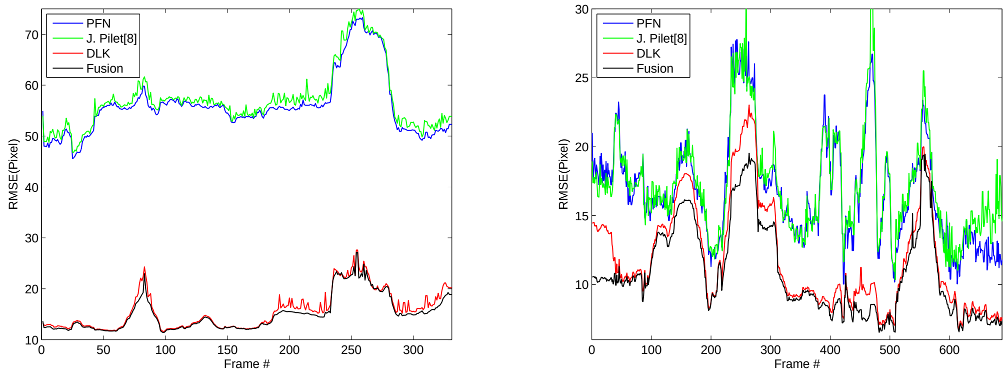
	Vertices	Size	FPS	Initialization	Iteration
C_1 (color)	216	198660	2.4	84ms	$\sim$ 35.7ms
C_1 (gray)	216	198660	4.4	84ms	$\sim$ 16.7ms
C_2 (gray)	96	194670	4.7	68ms	$\sim$ 16.2ms
C_3 (gray)	216	109538	6.1	77ms	$\sim$ 10.2ms



(a) Magazine cover

(b) Paper

Figure 4.9: The Root Mean Square Error (RMSE) with given regularization parameter λ_r and weight coefficient α. Two sets of data are used for evaluation.



(a) Magazine cover video sequence

(b) Paper video sequence

Figure 4.10: The Root Mean Square Error (RMSE) comparison of the progressive finite Newton (PFN) method, the semi-implicit method [75], the deformable Lucas-Kanade (DLK) method and fusion approach on two videos. (a) As the model image and input video are from different stheces, RMSE for the feature-based method is much larger than that for the fusion method. (b) Both the model image and input video are captured by the same device and under similar lighting conditions, so RMSE is relatively low. Sample frames are shown in Fig. 4.11 and Fig. 4.12.

4.3.3 Fusion Approach

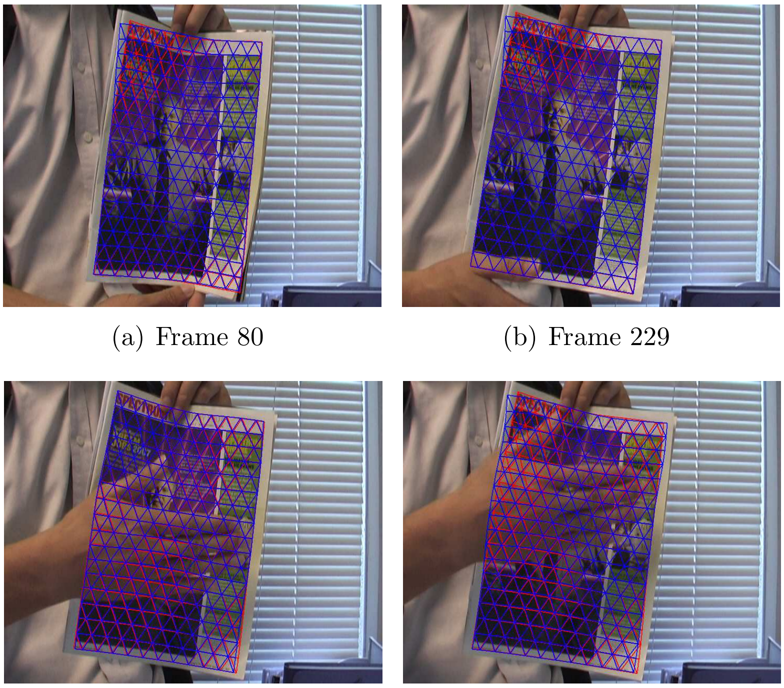
It is shown that the proposed fusion approach is able to be used for nonrigid shape recovery tasks.

Parameter Settings

Two datasets are engaged for searching the parameters. One is the magazine cover as illustrated in Fig. 4.11, and the other is a piece of paper in Fig. 4.12. For each dataset, we select ten testing images containing deformations, and then evaluate the proposed fusion approach using different λ_r and α. In the experiment, RMSE is used as the performance measurement. Also, a condition ($\|\Delta\mathbf{s}\| < 2.0$) is employed as the success criteria, and the failure cases are set to the highest RMSE. Fig. 4.9 plots the mean RMSE of ten tests. It can be observed that there is a broad area with low RMSE for selecting λ_r and α; the lowest RMSE is found in the middle dark region. Therefore, the local features are useful to improve the fitting accuracy, and there is a large range for choosing the weight coefficient α. When α becomes larger, the result is more similar to those from the feature-based method; and there is a constant ratio between λ_r and α. It can also be found that the optimization seldom converges with small λ_r. Furthermore, as shown in the upper part of each figure, large λ_r may lead to over-smoothing. $\lambda_r = 2\times 10^4$ and $\alpha = 10^6$ are used in the following experiments.

Performance Evaluation

Two videos were captured for performance evaluation, which are the magazine cover and a piece of paper. To investigate the occlusion problem, the magazine cover is occluded by hand in some frames. For simplicity, the feature-based method in Section 3.3 is denoted as "PFN". The deformable Lucas-Kanade algorithm in Section 4.2.4 is denoted as "DLK", which is equivalent to the



(a) Frame 80

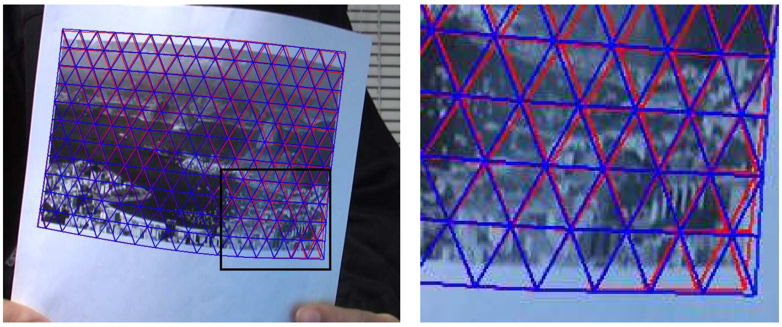
(b) Frame 229

(c) Frame 249

(d) Frame 271

Figure 4.11: Comparison of the deformable Lucas-Kanade (DLK) method (blue) and Fusion approach (red) on the magazine video. The magazine cover is occluded by hand in (c-d).

fusion approach with $\alpha = 0$. The proposed fusion approach is denoted as "Fusion". Fig. 4.10 shows the results of the comparison between two feature-based methods (PFN and J. Pilet [75]) and two appearance-based methods (DLK and Fusion). From the experimental results, it can be first observed that fusion approach consistently obtains the lowest RMSE. Second, it can be found that PFN is slightly better than the own implementation of the J. Pilet [75]. Further, comparing the two appearance-based methods, DLK may suffer drift problem in some frames as shown in Fig. 4.11 and Fig. 4.12. For those frames containing small deformations, the two methods obtain very similar results. Also, the proposed fusion approach and DLK method are able to handle the partial occlusion well without other treatment such as the robust loss functions, which is mainly due to the direct parameterizations and the regularization method. In addition, a piece of paper is used to occlude the patterns on the paper, and the results are shown in Fig. 4.13. Since the inverse compositional optimization starts from a good initialization, the optimization for the fusion approach usually requires around 8 iterations.



(a) Frame 251

(b) Zoomed region

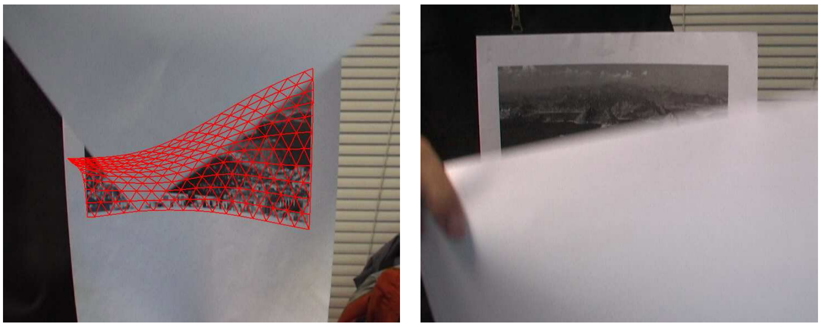
Figure 4.12: Comparison of the deformable Lucas-Kanade (DLK) method (blue) and Fusion approach (red) on the paper video. Results are shown at frame 251.

Fig. 4.14 illustrates the results of recovering the nonrigid shape from a real-time video. It can be observed that the fusion method is robust to large deformations and perspective distortions. In Fig. 4.15, it shows the results of erasing the patterns on the recovered surfaces using the method in [76]. It can be seen that both the shadows and the specular regions are also correctly estimated. In addition, the artifacts in the resulting images are mainly due to an uncontrolled optical sensor.

4.4 Discussions

4.4.1 Deformable Lucas-Kanade algorithm

We discuss several major differences of the proposed deformable Lucas-Kanade algorithm compared with the previous work. In



(a) Detected

(b) Failed

Figure 4.13: Pattern occluded by a piece of paper.

contrast to the conventional Lucas-Kande algorithm for image alignment [2], the proposed deformable Lucas-Kanade algorithm can handle the large deformations rather than the affine transformation or the homography. Different from the Active Appearance Model [23, 65], the proposed approach does not require a set of representative training examples to build the shape and the texture models. Comparing to other deformable template matching methods, such as Active blob [83], the proposed deformable Lucas-Kanada algorithm has several advantages. First, the deformable model is more flexible. Second, the optimization of the proposed approach is an efficient sparse problem, which can reduce the computational cost by pre-computing the gradient and Hessian.

4.4.2 Fusion Approach

In contrast to the feature-based image alignment methods presented in Chapter 3, the fusion approach can deal with large deformations and perspective distortions, in which correct feature correspondences are difficult to obtain. Also, the jitter is greatly reduced in the fusion approach. Furthermore, the proposed fusion method can handle the partial occlusion, which is mainly due to the triangulated mesh model and the regular-

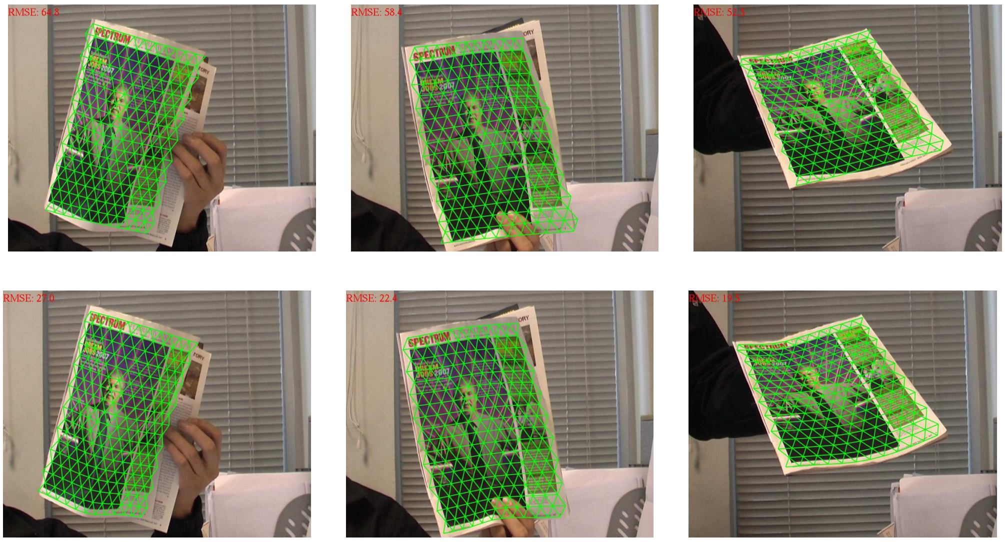


Figure 4.14: Recovering the cover of a magazine in a real-time video with the size of 720 × 576. The first row shows the initialization results using feature correspondences only. The second row shows the results with the fusion approach. Moreover, Root Mean Square Error (RMSE) is shown at the left corner in each image.

ization method. In this thesis, however, we have not discussed the illumination issue for the appearance and fusion approaches, which may be tackled by employing illumination model in the literature, such as Phong model used in [14].

4.5 Summary

This chapter presents a fusion approach to solve the nonrigid shape recovery problem, which takes advantage of both the appearance information and the local features.

In contrast to the conventional Lucas-Kanade algorithm, the proposed approach employs a deformable mesh model and can handle image alignment when the deformation is large. The proposed deformable Lucas-Kanade algorithm is formulated into a sparse regularized least squares problem, which can be efficiently

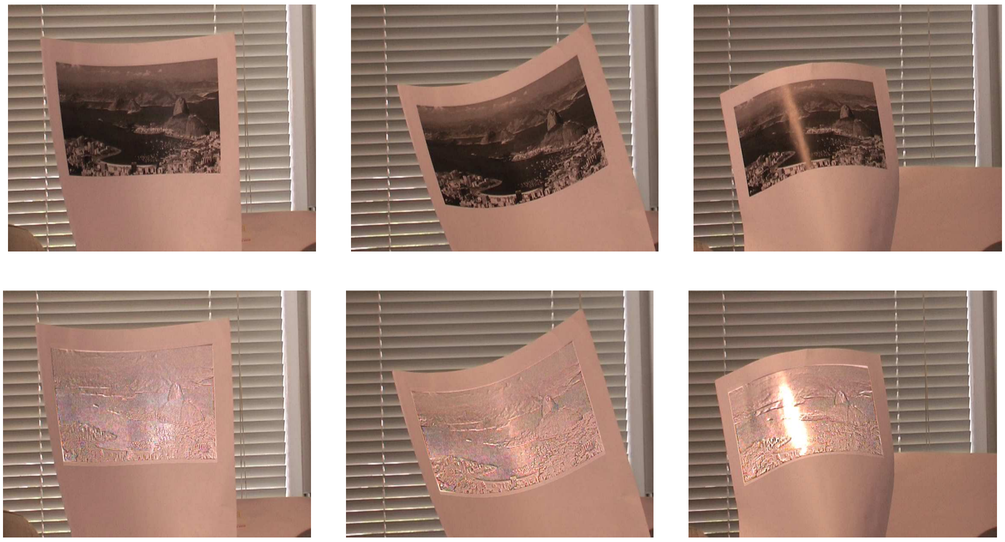


Figure 4.15: Diminishing a picture on a piece of paper. The first row shows the 720 × 576 images captured by a DV camera. The second row shows the results of diminishing the texture on the paper.

solved by the inverse compositional method. Finally, the optimization problem for the fusion approach is tackled under the deformable Lucas-Kanade algorithm framework.

Extensive experiments have been conducted on the proposed deformable Lucas-Kanade algorithm and fusion approach, which evaluate them in image alignment and nonrigid surface recovery tasks. The experimental results demonstrate that these algorithms are promising for image alignment and nonrigid object tracking.

□ **End of chapter.**

Chapter 5

3D Deformable Surface Tracking

Compared with 2D nonrigid shape recovery in the previous two chapters, this chapter presents an effective approach to 3D deformable surface tracking, which is able to estimate the depth information from the 2D observations. In contrast to the recent Second Order Cone Program method [80], we reformulate the problem into an unconstrained quadratic optimization problem. The new formulation can be solved very efficiently by resolving a set of sparse linear equations. Based on the new framework, the progressive finite Newton optimization scheme described in Chapter 3 is employed to handle large outliers. An extensive set of experiments have been conducted to evaluate the performance on both synthetic and real-world testbeds, from which the promising results show that the proposed algorithm not only achieves better tracking accuracy, but also executes significantly faster than the previous solution.

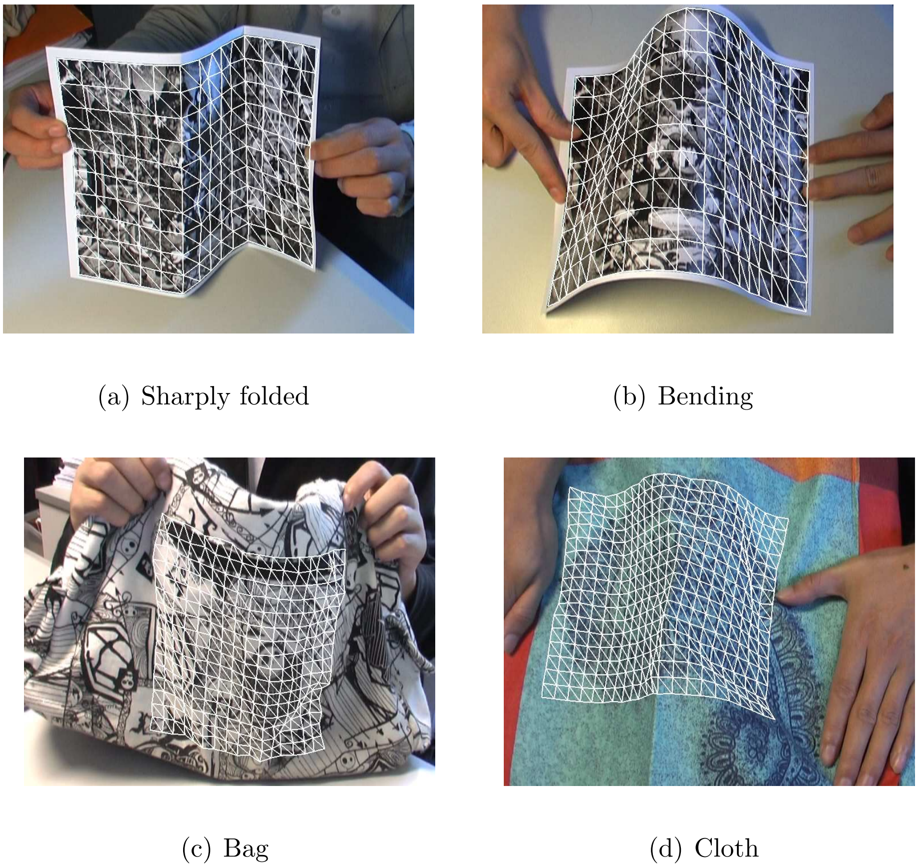
5.1 Motivation

3D deformable surface modeling and tracking has attracted extensive research interest due to its significant role in many computer vision applications [6, 82, 95, 101, 116, 124]. This work is mainly motivated from the SOCP method [80], the convex optimization [50] and quasiconvex optimization [52] to the triangulation problem. Moreover, it is important to note that this work is closely related to previous work on structure from motion [19] as well as nonrigid surface detection and tracking [76, 101, 123, 124].

Factorization methods are widely used in 3D deformable surface recovery. Bregler et al. [19] proposed a solution for recovering 3D nonrigid shapes from video sequences, which factorizes the tracked 2D feature points to build the 3D shape model. In this approach, the 3D shape in each frame is represented by a linear combination of a set of basis shapes. A similar method was applied to the Active Appearance Models fitting results in order to retrieve the 3D facial shapes from video sequences [105]. Based on the factorization method, a weak constraint [81] can be introduced to handle the ambiguities problem by constraining the frame-to-frame depth variations. In addition, machine learning techniques have also been applied to building the linear subspace from either the collected data or the synthetic data. Although some promising results have been achieved in 3D face fitting [14] and deformable surface tracking [82], these methods usually require a large number of training samples to obtain sufficient generalization capability.

As for 2D nonrigid surface detection, we have investigated in the previous two chapters. J. Pilet et al. [76] proposed a real-time algorithm which employs a semi-implicit optimization approach to handle noisy feature correspondences. In contrast, several image registration methods [6, 20] tend to be computationally expensive and are mainly aimed at object recognition.

Since the deformable surface is usually highly dynamic and represented by many deformation parameters, the prior models are often engaged in dealing with the ill-posed optimization problem of deformable surface recovery. As described in Chap-



(a) Sharply folded

(b) Bending

(c) Bag

(d) Cloth

Figure 5.1: Recovering highly deformable surfaces from video sequences (a-d). (a) A piece of paper with well-marked creases. (b) Severely bending. (c) Bag surface. (d) A piece of cloth.

ter 2, a variety of methods have been proposed to create these models, such as the interpolation method [6, 20], the data embedding method [82, 105, 116] and physical models [32, 47, 76]. The major problem of these models is that their smoothness constraints usually limit their capability of accurately recovering sharply folded and creased surfaces.

Instead of using the strong prior models, M. Salzmann et al. recently formulated the problem generally as a Second Order Cone Programming (SOCP) problem without engaging the unwanted smoothness constraints [80]. Although they have demonstrated some promising results on tracking deformable surfaces from 3D to 2D correspondences, their approach is computationally expensive while handling a large number of SOCP constraints for a large set of free variables. In this chapter, we apply the principles they have described, and investigate new techniques to address the shortcomings.

Specifically, we propose a novel unconstrained quadratic optimization formulation for 3D deformable surface tracking, which requires only the solution of a set of sparse linear equations. In the proposed approach, we first show that the SOCP formulation can be viewed as a special case of a general convex optimization feasibility problem. Then, a slack variable is introduced to rewrite the SOCP formulation into a series of Quadratic Programming (QP). Furthermore, the SOCP constraints are converted into a quadratic regularization term, which leads to a novel unconstrained optimization formulation. Finally, we show that the resulting unconstrained optimization problem can be solved efficiently by the robust progressive finite Newton optimization scheme introduced in Chapter 3, which can handle large outliers. Hence, not only is the proposed solution highly efficient, but also it can directly handle noisy data in an effective way. To evaluate the performance of the proposed algorithm, extensive experiments have been conducted on both synthetic and real-world data, as shown in Fig. 5.1.

The rest of this chapter is organized as follows.Section 5.2 presents the proposed 3D deformable surface tracking solution using a novel unconstrained quadratic optimization method. Section 5.3 shows the details of the experimental implementation and evaluates the experimental results. Section 5.4 discusses some limitations and sets out the conclusion.

5.2 Fast 3D Deformable Surface Tracking

In this section, we first formally define the 3D deformable surface tracking problem. Then, an optimization framework is presented for treating the 3D deformable surface tracking problem as a general convex optimization feasibility problem. We then revisit previous SOCP work that can be viewed as a special case of the general convex optimization framework. With a view to improving the efficiency of the optimization, we present techniques to relax the SOCP constraints properly and propose two new optimization formulations. One is a QP formulation and the other is an efficient unconstrained quadratic optimization.

5.2.1 Problem Definition

The 3D deformable surface is explicitly represented by triangulated meshes. As shown in Fig. 5.1, a triangulated 3D mesh with n vertices is employed in this chapter, and the vertices' coordinates are formed into a shape vector $\mathbf{s}$ as below:

$$\mathbf{s} = \left[\begin{array}{ccccccccc} x_1 & \dots & x_n & y_1 & \dots & y_n & z_1 & \dots & z_n \end{array} \right]^\top$$

in which $\mathbf{v}_i = (x_i, y_i, z_i)^\top$ is defined as the coordinates of the i^{th} mesh vertex. The shape vector $\mathbf{s}$ is the variable to be estimated.

Given a set of 3D to 2D correspondences $\mathcal{M}$ between the surface points and the image locations, a pair of matched points is defined as $\mathbf{m} = (\mathbf{m}_S, \mathbf{m}_I) \in \mathcal{M}$, where $\mathbf{m}_S$ is the 3D point on the surface and $\mathbf{m}_I$ is the corresponding 2D location on the input image.

Assuming that the surface point $\mathbf{m}_S$ lies on a facet whose three vertices' coordinates are $\mathbf{v}_i, \mathbf{v}_j$ and $\mathbf{v}_k$ respectively, and $\{i, j, k\} \in [1, n]$ is the index of each vertex. The piecewise affine transformation is used to map the surface points $\mathbf{m}_S$ inside the corresponding triangle into the vertices in the mesh:

$$\mathbf{m}_S = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} x_i & x_j & x_k \\ y_i & y_j & y_k \\ z_i & z_j & z_k \end{bmatrix} \begin{bmatrix} \xi_1 & \xi_2 & \xi_3 \end{bmatrix}^\top$$

where $(\xi_1, \xi_2, \xi_3)^\top$ are the barycentric coordinates for the surface point $\mathbf{m}_S$.

As in [80], the 3×4 camera projection matrix $\mathbf{P}$ is assumed to be known and remains constant. This does not mean that the camera is fixed, since the relative motion with respect to the camera can be recovered during the tracking process. Hence, with the projection matrix $\mathbf{P}$, we can compute $\mathbf{m}_I = [\ u \quad v\ ]^\top$, the 2D projection of the 3D surface point $\mathbf{m}_S$, as follows:

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \frac{P_{1,1}x+P_{1,2}y+P_{1,3}z+P_{1,4}}{P_{3,1}x+P_{3,2}y+P_{3,3}z+P_{3,4}} \\ \frac{P_{2,1}x+P_{2,2}y+P_{2,3}z+P_{2,4}}{P_{3,1}x+P_{3,2}y+P_{3,3}z+P_{3,4}} \end{bmatrix} \tag{5.1}$$

In order to directly represent the projection by the variables $\mathbf{s}$, an augmented vector $\mathbf{a} \in \mathbb{R}^{3n}$ is defined as below:

$$\begin{aligned} \mathbf{a}_i &= \xi_1 P_{1,1} \quad \mathbf{a}_{i+n} = \xi_1 P_{1,2} \quad \mathbf{a}_{i+2n} = \xi_1 P_{1,3} \\ \mathbf{a}_j &= \xi_2 P_{1,1} \quad \mathbf{a}_{j+n} = \xi_2 P_{1,2} \quad \mathbf{a}_{j+2n} = \xi_2 P_{1,3} \\ \mathbf{a}_k &= \xi_3 P_{1,1} \quad \mathbf{a}_{k+n} = \xi_3 P_{1,2} \quad \mathbf{a}_{k+2n} = \xi_3 P_{1,3} \end{aligned}$$

The remaining elements of the vector $\mathbf{a}$ are all set to zero. Similarly, we define other two vectors $\mathbf{b}, \mathbf{c} \in \mathbb{R}^{3n}$ accordingly, and then rewrite Eqn. 5.1 as follows:

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \frac{\mathbf{a}^\top \mathbf{s}+P_{1,4}}{\mathbf{c}^\top \mathbf{s}+P_{3,4}} \\ \frac{\mathbf{b}^\top \mathbf{s}+P_{2,4}}{\mathbf{c}^\top \mathbf{s}+P_{3,4}} \end{bmatrix} \tag{5.2}$$

Definition 5. *The 3D deformable surface tracking problem is to estimate the 3D shape (or mesh)* $\mathbf{s}$ *from a set of 3D to 2D correspondences* $\mathcal{M}$ *in a video sequence, in which the projection formula is based on Eqn. 5.2.*

5.2.2 Convex Optimization Formulations

General Convex Formulation

Since it is impossible to find a perfect projection that can ideally match all the 3D to 2D correspondences in practice, let γ denote the upper bound for the reprojection error of each correspondence pair $\mathbf{m} \in \mathcal{M}$. As a result, for each 2D image observation $\mathbf{m}_I = [\ \hat{u}\ \ \hat{v}\ ]^\top$, the following inequality constraint will be satisfied:

$$\left\| \frac{\mathbf{a}^\top \mathbf{s} + P_{1,4}}{\mathbf{c}^\top \mathbf{s} + P_{3,4}} - \hat{u}, \frac{\mathbf{b}^\top \mathbf{s} + P_{2,4}}{\mathbf{c}^\top \mathbf{s} + P_{3,4}} - \hat{v} \right\|_p \leq \gamma \text{ for } \mathbf{m} \in \mathcal{M}, \qquad (5.3)$$

where $p \geq 1$ is a constant integer and the inequality constraint is known as a p-norm cone constraint [15]. As a result, the 3D deformable surface tracking problem can be formulated as a general convex optimization problem:

$$\begin{array}{ll} \min\limits_{\gamma \geq 0, \mathbf{s}} & \gamma \\ \text{s. t.} & \left\| \frac{\mathbf{a}^\top \mathbf{s} + P_{1,4}}{\mathbf{c}^\top \mathbf{s} + P_{3,4}} - \hat{u}, \frac{\mathbf{b}^\top \mathbf{s} + P_{2,4}}{\mathbf{c}^\top \mathbf{s} + P_{3,4}} - \hat{v} \right\|_p \leq \gamma \\ & \text{for each } \mathbf{m} \in \mathcal{M}. \end{array}$$

In the above optimization, γ is usually set by the bisection algorithm [50, 80]. Hence, the tracking problem can be regarded as a feasibility problem for the above general convex optimization.

When $p = 2$, the p-norm cone constraint above reduces to the well-known SOCP constraint. In the following discussion, we will show that a recently proposed SOCP formulation can be viewed as a special case of the above general convex optimization feasibility problem.

SOCP Formulation

The recent work in [80] formulates the 3D deformable surface tracking problem as an SOCP feasibility problem, which can be viewed as a special case of the above general convex optimization with $p = 2$:

$$\begin{aligned} \min_{\gamma \geq 0, \mathbf{s}} \quad & \gamma \\ \text{s. t.} \quad & \left\| \frac{\mathbf{a}^\top \mathbf{s} + P_{1,4}}{\mathbf{c}^\top \mathbf{s} + P_{3,4}} - \hat{u}, \frac{\mathbf{b}^\top \mathbf{s} + P_{2,4}}{\mathbf{c}^\top \mathbf{s} + P_{3,4}} - \hat{v} \right\| \leq \gamma \\ & \text{for each } \mathbf{m} \in \mathcal{M}. \end{aligned} \tag{5.4}$$

where the 2-norm notation $\| \cdot \|_2$ is by default written as $\| \cdot \|$ without ambiguity. To handle the outliers, the method described in [84] is employed to remove the set of matches whose reprojection errors equal the minimal γ.

In practice, to regularize the deformable surface, an additional constraint is introduced to prevent irrational changes of the edge orientations between two consecutive frames [80]. Assuming that the shape $\mathbf{s}^t$ at time t is known, and that the orientation of the edge linking the vertices $\mathbf{v}_i^t$ and $\mathbf{v}_j^t$ will be similar at time $t + 1$. For each edge in the triangulated mesh, the corresponding constraint can be formulated as below:

$$\left\| \mathbf{v}_i^{t+1} - \mathbf{v}_j^{t+1} - \boldsymbol{\theta}_{ij}^t \right\| \leq \lambda L_{i,j} \tag{5.5}$$

where $L_{i,j}$ is the original length of the edge. $\boldsymbol{\theta}_{ij}^t$ is the difference of the two vertices $\mathbf{v}_i^t$ and $\mathbf{v}_j^t$ at time t normalized by the original edge length, namely

$$\boldsymbol{\theta}_{ij}^t = L_{i,j} \frac{\mathbf{v}_i^t - \mathbf{v}_j^t}{\|\mathbf{v}_i^t - \mathbf{v}_j^t\|}$$

Also, λ is a coefficient to control the regularity of the deformable surface. Again, the above inequality constraint is also an SOCP constraint. As a result, the tracking problem is formulated as an SOCP feasibility problem [1] with a number of SOCP constraints, which can be solved by some bisection algorithm [50, 80].

[1] The SOCP optimization problem is solved by Sedumi: http://sedumi.mcmaster.ca.

A major problem of the above formulation is that the number of correspondences $|\mathcal{M}|$ is often much larger than the number of variables for ensuring sufficient correct matches, and thus the SOCP formulation has to engage a large number of SOCP constraints. Specifically, if n_e denotes the number of edges in the mesh model, the above SOCP formulation should have $(|\mathcal{M}| + n_e)$ SOCP constraints in total. Solving the above SOCP optimization directly leads to very high computational cost in practice.

QP Formulation

The drawback of the SOCP formulation lies in the large number of SOCP constraints. In this part, we present a QP formulation by removing the SOCP constraints. Specifically, for each of the SOCP constraints in Eqn. 5.4, we can rewrite it equivalently as

$$[(\mathbf{a} - \hat{u}\mathbf{c})^\top \mathbf{s} + d_u]^2 + [(\mathbf{b} - \hat{v}\mathbf{c})^\top \mathbf{s} + d_v]^2 \leq \gamma(\mathbf{c}^\top \mathbf{s} + d_w)^2$$

where $d_w = P_{3,4}$, $d_u = P_{1,4} - \hat{u}d_w$, and $d_v = P_{2,4} - \hat{v}d_w$. Further, a slack variable $\epsilon(\mathbf{m})$ can be introduced for each $\mathbf{m} \in \mathcal{M}$ and the inequality constraint can be rewritten as the following equality:

$$[(\mathbf{a} - \hat{u}\mathbf{c})^\top \mathbf{s} + d_u]^2 + [(\mathbf{b} - \hat{v}\mathbf{c})^\top \mathbf{s} + d_v]^2 + \epsilon(\mathbf{m})^2 = \gamma(\mathbf{c}^\top \mathbf{s} + d_w)^2$$

In addition, the SOCP constraints can be replaced in Eqn. 5.5 with 1-norm cone constraints. As a result, the original formulation can be rewritten by a min-max optimization formulation:

$$\begin{aligned} \min_{\gamma \geq 0} \ \max_{\mathbf{s}} \ & \sum_{\mathbf{m} \in \mathcal{M}} \epsilon(\mathbf{m})^2 \\ \text{s. t.} \ & \left\| \mathbf{v}_i^{t+1} - \mathbf{v}_j^{t+1} - \boldsymbol{\theta}_{ij}^t \right\|_1 \leq \lambda L_{i,j} \\ & \text{for each edge } (\mathbf{v}_i, \mathbf{v}_j) \text{ in the mesh.} \end{aligned}$$

in which the objective function can be expressed as:

$$\sum_{\mathbf{m} \in \mathcal{M}} \epsilon(\mathbf{m})^2 = -(\mathbf{s}^\top \mathbf{H}\mathbf{s} + 2\mathbf{g}^\top \mathbf{s} + d) \tag{5.6}$$

where $\mathbf{H} \in \mathbb{R}^{3n\times 3n}$, $\mathbf{g} \in \mathbb{R}^{3n\times 1}$ and $d \in \mathbb{R}$ are defined as:

$$\begin{aligned}
\mathbf{H} &= \sum_{\mathbf{m}\in\mathcal{M}} (\mathbf{a}-\hat{u}\mathbf{c})(\mathbf{a}-\hat{u}\mathbf{c})^\top + (\mathbf{b}-\hat{v}\mathbf{c})(\mathbf{b}-\hat{v}\mathbf{c})^\top - \gamma\mathbf{c}\mathbf{c}^\top \\
\mathbf{g} &= \sum_{\mathbf{m}\in\mathcal{M}} d_u(\mathbf{a}-\hat{u}\mathbf{c}) + d_v(\mathbf{b}-\hat{v}\mathbf{c}) - \gamma\mathbf{c} \\
d &= \sum_{\mathbf{m}\in\mathcal{M}} d_u^2 + d_v^2 - \gamma d_w^2
\end{aligned}$$

It is clear that the above objective function is quadratic. For the tracking task to be an optimization feasibility problem, γ is assumed to be known. Hence, the min-max optimization becomes a standard QP problem. To solve it, we also employ the bisection algorithm and engage an interior-point optimizer [2].

5.2.3 Unconstrained Quadratic Optimization

The QP formulation still has to include a number of 1-norm cone constraints. To address it, we present an unconstrained quadratic optimization formulation that completely relaxes all constraints. Specifically, instead of engaging the SOCP constraints in Eqn. 5.5, we integrate such constraints into the objective function by treating it as a weighted penalty function, which converts the complex SOCP constraints into a simple quadratic term. This leads to the following unconstrained minimization formulation:

$$\min_{\gamma,\mathbf{s}} \; -\sum_{\mathbf{m}\in\mathcal{M}} \epsilon^2 + \mu \sum_{k=1}^{n_e} \eta_k^2 \qquad (5.7)$$

where μ is a regularization coefficient, and η_k is a variable to constrain the regularity of the k^{th} edge:

$$\eta_k = \left\| \mathbf{v}_i^{t+1} - \mathbf{v}_j^{t+1} - \boldsymbol{\theta}_{ij}^t \right\|$$

[2] http://www.mosek.com/

Moreover, the edge regularization term can be expressed as:

$$\sum_{k=1}^{n_e} \eta_k^2 = \mathbf{s}^\top \mathbf{Q}\mathbf{s} - 2\mathbf{f}^\top \mathbf{s} + \varphi \tag{5.8}$$

where $\mathbf{Q} \in \mathbb{R}^{3n\times 3n}$, $\mathbf{f} \in \mathbb{R}^{3n\times 1}$ and $t \in \mathbb{R}$ are defined as:

$$\begin{aligned} Q &= \sum_{k=1}^{n_e} \tilde{\mathbf{a}}\tilde{\mathbf{a}}^\top + \tilde{\mathbf{b}}\tilde{\mathbf{b}}^\top + \tilde{\mathbf{c}}\tilde{\mathbf{c}}^\top \\ \mathbf{f} &= \sum_{k=1}^{n_e} \theta_x \tilde{\mathbf{a}} + \theta_y \tilde{\mathbf{b}} + \theta_z \tilde{\mathbf{c}}, \\ \varphi &= \sum_{k=1}^{n_e} \|\boldsymbol{\theta}_k\| \end{aligned}$$

where $\boldsymbol{\theta}_k = (\theta_{kx}, \theta_{ky}, \theta_{kz})^\top$ is used to denote $\boldsymbol{\theta}_{ij}^t$. For the k^{th} edge with vertices $\mathbf{v}_i$ and $\mathbf{v}_j$, three augmented vectors $\tilde{\mathbf{a}}, \tilde{\mathbf{b}}$ and $\tilde{\mathbf{c}} \in \mathbb{R}^{3n}$ are defined as follows:

$$\begin{aligned} \tilde{\mathbf{a}}_i &= 1 \qquad \tilde{\mathbf{b}}_{i+n} = 1 \qquad \tilde{\mathbf{c}}_{i+2n} = 1 \\ \tilde{\mathbf{a}}_j &= -1 \qquad \tilde{\mathbf{b}}_{j+n} = -1 \qquad \tilde{\mathbf{c}}_{j+2n} = -1 \end{aligned}$$

and the remaining elements in $\tilde{\mathbf{a}}, \tilde{\mathbf{b}}$ and $\tilde{\mathbf{c}}$ are all set to zero. By substituting Eqn. 5.6 and Eqn. 5.8 into Eqn. 5.7, thus, this leads to the following unconstrained minimization formulation:

$$\min_{\gamma\geq 0, \mathbf{s}} \; \mathbf{s}^\top (\mathbf{H} + \mu\mathbf{Q})\mathbf{s} + 2(\mathbf{g} - \mu\mathbf{f})^\top \mathbf{s} + d + \varphi \tag{5.9}$$

Remark In the above formulation, $\mathbf{H}$, $\mathbf{g}$ and d are all related to the upper bound variable γ, which seems like a complicated optimization problem. Fortunately, we find that the upper bound γ plays the same role as the support of the robust estimator in [76, 123], which is able to handle large outliers. Therefore, the above problem can be perfectly solved by the progressive finite Newton method as proposed in Chapter 3, which makes the proposed method capable of handling large outliers. Specifically, the upper bound γ starts at a large value, and then is progressively decreased at a constant rate. For each value of the upper bound γ, we can simply solve the following linear equation:

$$(\mathbf{H} + \mu\mathbf{Q})\mathbf{s} = -\mathbf{g} + \mu\mathbf{f} \tag{5.10}$$

where $\mathbf{H}$ and $\mathbf{g}$ are computed with the inlier matches only. We employ the results from the previous step to compute the inlier set. Obviously, the square matrix $\mathbf{Q}$ is kept constant for the given triangulated mesh, and $\mathbf{f}$ only needs to be computed once for each frame. Since both $\mathbf{H}$ and $\mathbf{Q}$ are sparse matrices, the above linear system can be solved very efficiently by a sparse linear solver. Owing to its high efficiency, the proposed solution enables us to handle very large scale 3D deformable surface tracking problems with high resolution meshes.

5.3 Experimental Results

In this section, we present the details of the experimental implementation and report the empirical results on 3D deformable surface tracking. First, an evaluation is performed on synthetic data for comparison with the convex optimization method. Then, results of the proposed approach are demonstrated in various environments, which indicate that the presented method is both efficient and effective for 3D deformable surface tracking.

5.3.1 Experimental Setup

All the experiments reported in this chapter are carried out on an Intel Core2 Duo 2.0GHz Notebook Computer with 2GB RAM, and a DV camera was engaged to capture the videos. For simplicity, the QP formulation is denoted as “QP”, and the proposed unconstrained quadratic optimization method is denoted as “QO”. All the methods are implemented in Matlab, in which some routines were written in C code. Instead of relying on the 2D tracking results as in [80], we directly employ the SIFT method [62] to build the 3D to 2D correspondences by matching the model image and the input image. The planar surface with a template image is used due to its simplicity. Moreover, the non-planar surface can be employed by embedding the texture into 2D space.

For the SOCP method, the similar parameters settings are used as given in [80]. Specifically, in the experiments, λ is set to 0.1, and the bisection algorithm stops when the maximal reprojection error is below one pixel. For the proposed QO method, the regularization parameter μ is found by grid searching, which is set to 5×10^4 for all experiments. The decay rate for the upper bound γ is set to 0.5.

To initialize the 3D tracking, the first frame is registered by the 2D nonrigid surface detection method described in Chapter 3, and then estimate the camera projection matrix $\mathbf{P}$ from 3D to 2D correspondences. In fact, the tracking usually starts from a surface that is slightly deformed. This method works well in practice, and it can automatically fit to the correct positions even when the initialization is not very accurate.

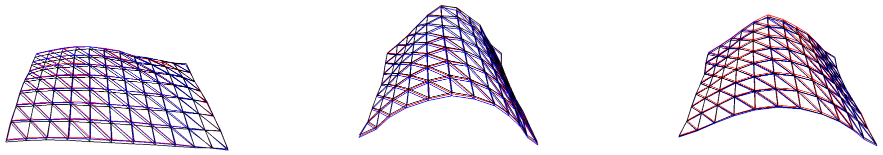


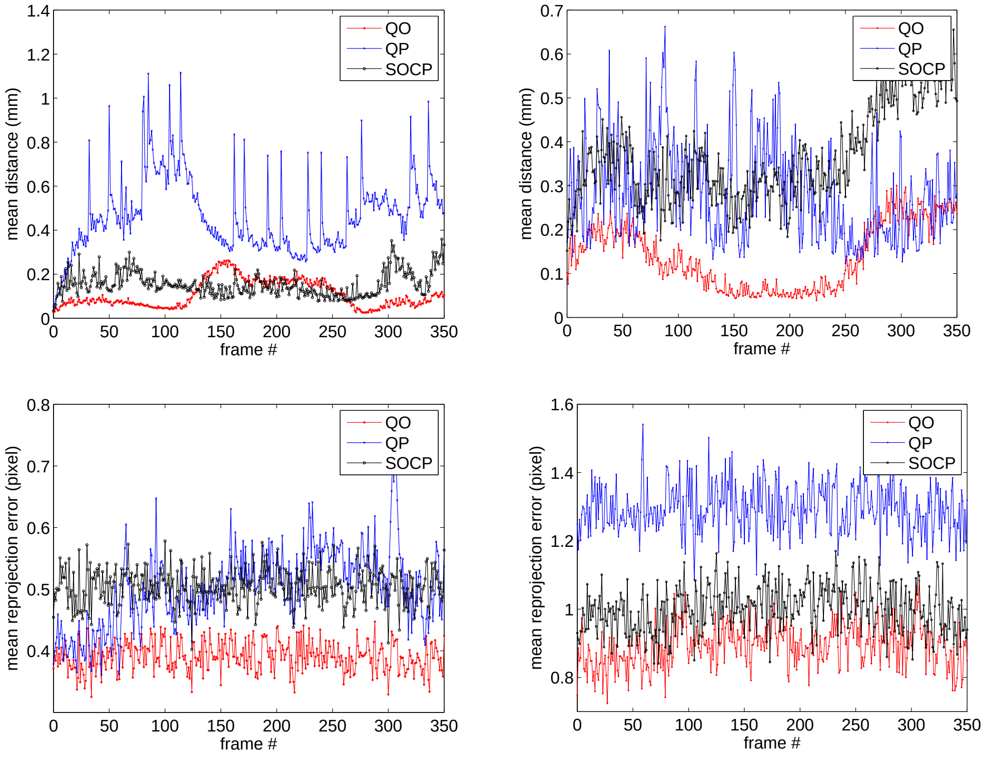
Figure 5.2: Synthetic meshes with 96 vertices for evaluation. The 2D observations corrupted by noise having a normal distribution with $\sigma = 2$. Results for SOCP (black) and QO (blue) are shown with ground truth (red), at frame 94, 170 and 220.

5.3.2 Synthetic Data Comparison

A sequence of 350 synthetic meshes is generated by simulating a surface bending process as shown in Fig. 5.2. The total size of the mesh is $280mm \times 200mm$. Given a perspective projection matrix $\mathbf{P}$, the 2D correspondences are obtained by projecting the 3D points defined by piecewise affine mapping, where the barycentric coordinates are randomly selected. Two sets of experiments are conducted on the synthetic data. Firstly, we conduct the experiments on 2D observations with a small amount of added noises. Secondly, the performance of SOCP and QO methods are evaluated on data with large outliers. The number of correspondences in each facet is set to 5 for the first experiment, and 10 for the second one.

Experiment I

In the first experiment, two cases of noisy data are evaluated, for which the noise is added to all the 2D observations based on a normal distribution with different standard deviations $\sigma = 1, 2$. Fig. 5.3 shows the results of the comparison between the QO, QP and SOCP methods. It can be observed that the proposed QO method achieves the lowest re-projection errors for both cases. When $\sigma = 1$, both QO and SOCP are more effective than the QP formulation in 3D reconstruction performance. Indeed, there is some large jittering for the QP method in 3D reconstruction. This may be due to the L_1 norm relaxation of the constraints that may cause ambiguities in depth. Also, the SOCP method slightly outperforms the QO method when the surface is highly deformed, as observed around frame 170 in Fig. 5.2. When the standard deviation of the noise increases, we found that the proposed QO method achieves better and more steady results than the other two methods. This shows that the QO method is more resilient to noises.



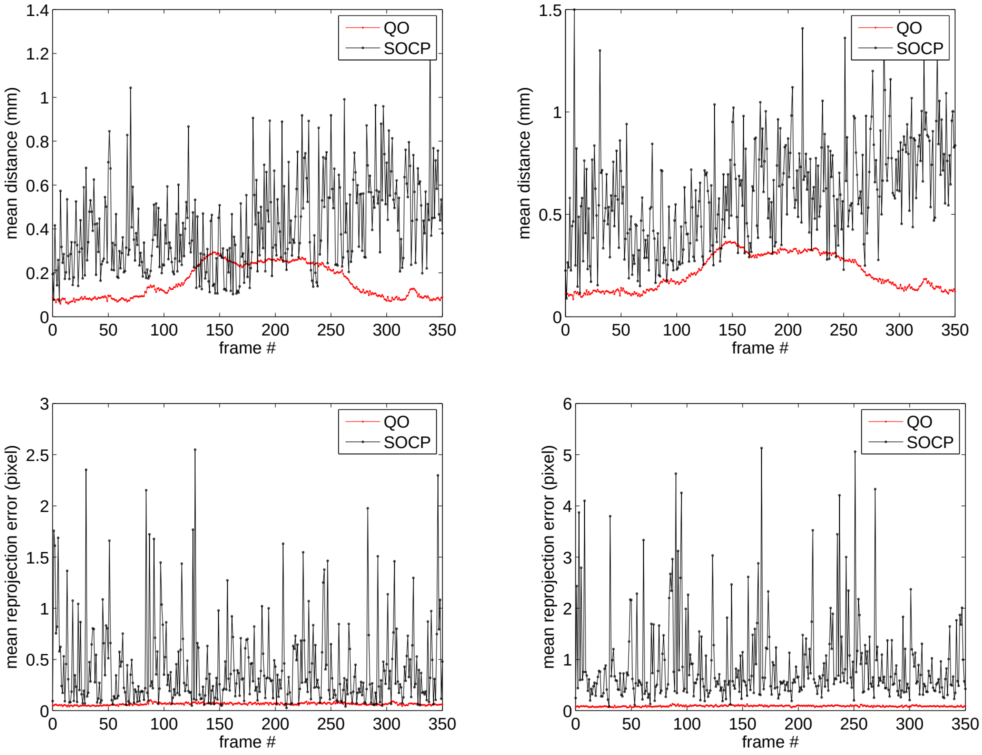
(a) $\sigma = 1$

(b) $\sigma = 2$

Figure 5.3: The performance comparison of the QO, QP and SOCP methods on the 350 synthetic meshes with little added noise. The first row shows the average distance between ground truth and recovery results. The second row is the mean reprojection errors.

Experiment II

In the second experiment, the experiments are conducted on the synthetic data partially corrupted by noises (40% and 60% respectively) with standard deviation $\sigma = 10$. The experimental results shown in Fig. 5.4 demonstrate that the proposed QO approach is very robust, and more effective than the SOCP method in dealing with large outliers. Furthermore, we observe that the results achieved by the QO approach are rather smooth. In contrast, large jittering is observed in the results from the SOCP method. In the experiments, the number of inliers for the QO



(a) 40%

(b) 60%

Figure 5.4: Comparison of the performance of the QO and SOCP on the synthetic data with large outliers. The first row shows the average distance between ground truth and recovery results. The second row is the mean reprojection errors.

method is larger than that for the SOCP method. Specifically, when the percentage of outliers is 60%, the average inlier rate is around 39% for QO, and below 30% for the SOCP method.

Computational Efficiency

The complexity of the proposed QO method is mainly dominated by the order of Eqn. 5.10, which is equal to $3n$. Another important factor is the number of inlier matches, which affects the sparseness of the system matrix. This number usually differs from one frame to another. For the synthetic data with

96 vertices, as shown in Fig. 5.2, the proposed method runs at about 29 frames per second on the synthetic data. As shown in Table 5.1, the proposed QO method takes 0.034 seconds per frame. On the other hand, the QP and SOCP methods require 10 seconds and 5 seconds per frame respectively. On average, the proposed QO method is over 140 times faster than the SOCP method.

Table 5.1: Average computational time per frame (seconds)

Quadratic Optimization (QO)	Quadratic Programming (QP)	SOCP [80]
0.034	10.0	5.0

5.3.3 Performance on Real Data

Next, we investigate the 3D deformable surface tracking performance on some real deformable surfaces based on a piece of paper, a bag and a piece of cloth. Since only the QO method is efficient enough in practice, we evaluate only the QO method on the real data. To ensure that a sufficient number of correct correspondences are found, all the objects are well-textured.

Paper

As shown in Fig. 5.5, the proposed method is robust in handling large bending deformations. In practice, the whole process runs at around one frame per second on the DV size video sequence with a 187-vertex mesh model. The SIFT feature extraction and matching takes most of the time, whereas the optimization procedure only requires 0.1 seconds for each frame. Fig. 5.7 shows that a sharply folded surface is retrieved, and the well-marked creases can be accurately recovered.

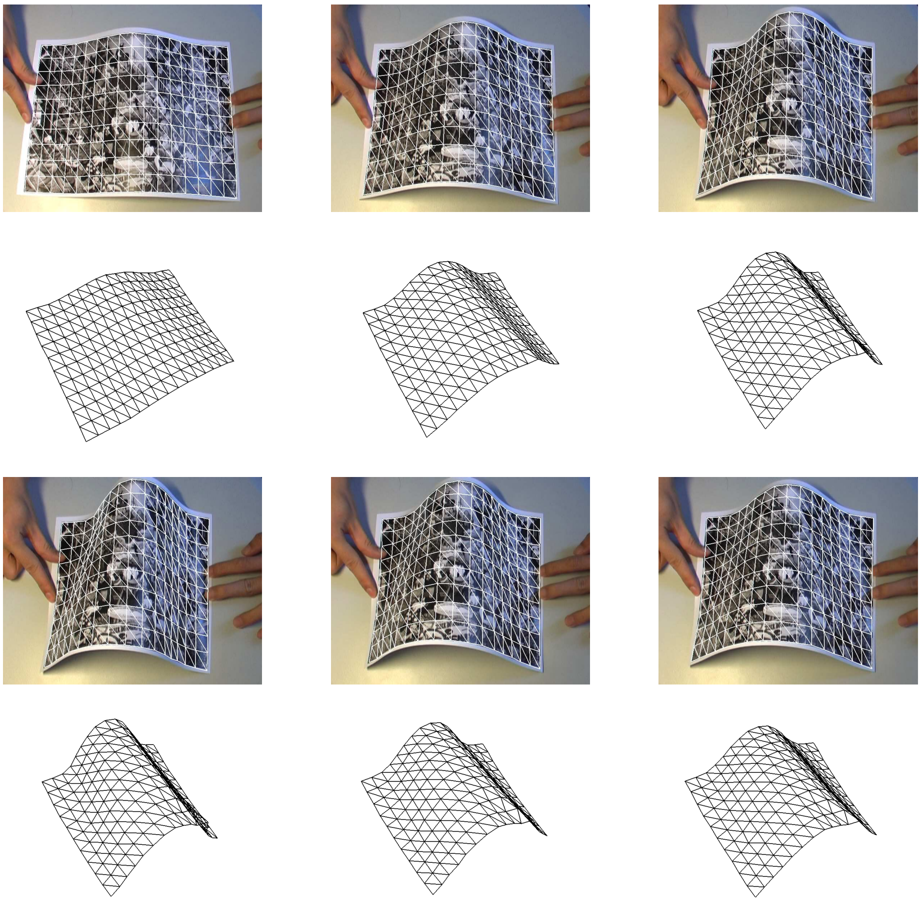


Figure 5.5: We use a piece of paper as the deformable object. The deformable surface is recovered from a 300 frame video. The first row shows the images captured by a DV camera size of 720 × 576 overlaid by the reprojection of the recovered mesh. The second row is a different projective view of the recovered 3D deformable surface.

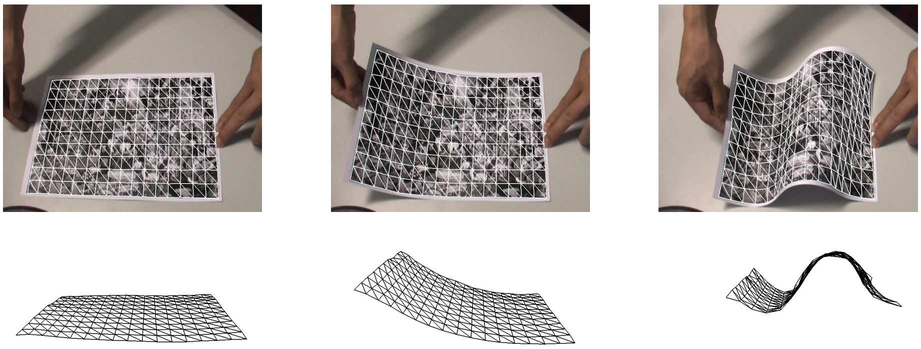


Figure 5.6: Tracking the deformable surface with waveform deformation.

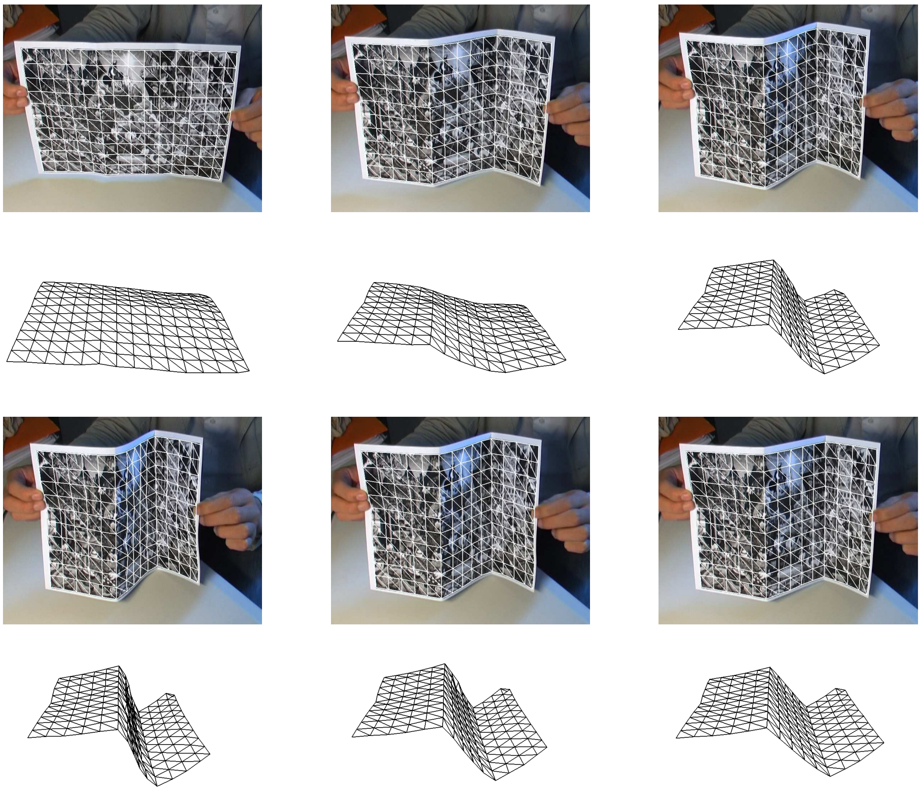


Figure 5.7: Tracking the deformable surface with two sharp folds in it. The creases are correctly recovered.

Bag and Cloth

To evaluate the performance on materials less rigid than a piece of paper, we reconstruct the surfaces of a bag and a piece of cloth with the proposed method. For the high efficiency of the proposed solution, we can handle real-world objects with high resolution mesh very fast. Fig. 5.8 shows the tracking results of the bag surface. The optimization procedure only takes about 0.2 seconds to process a mesh with 289 vertices. Similarly, Fig. 5.9 shows the tracking results of a piece of cloth. From these results, it can be concluded that the proposed method is able to recover the deformable surfaces accurately with the high resolution mesh.

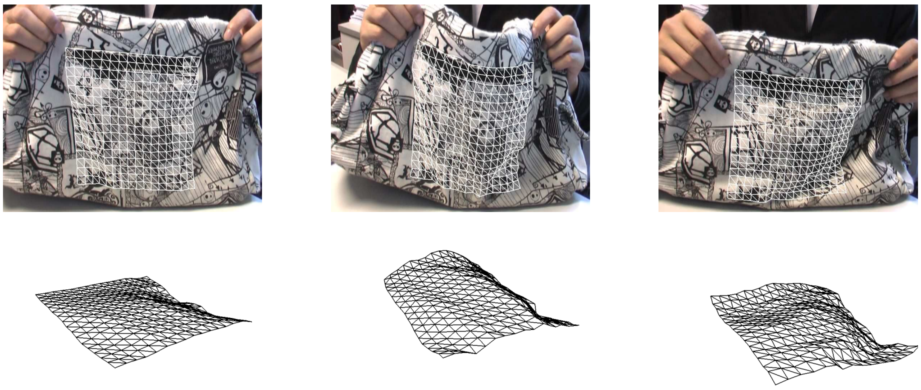


Figure 5.8: Recovering the deformation of a bag.

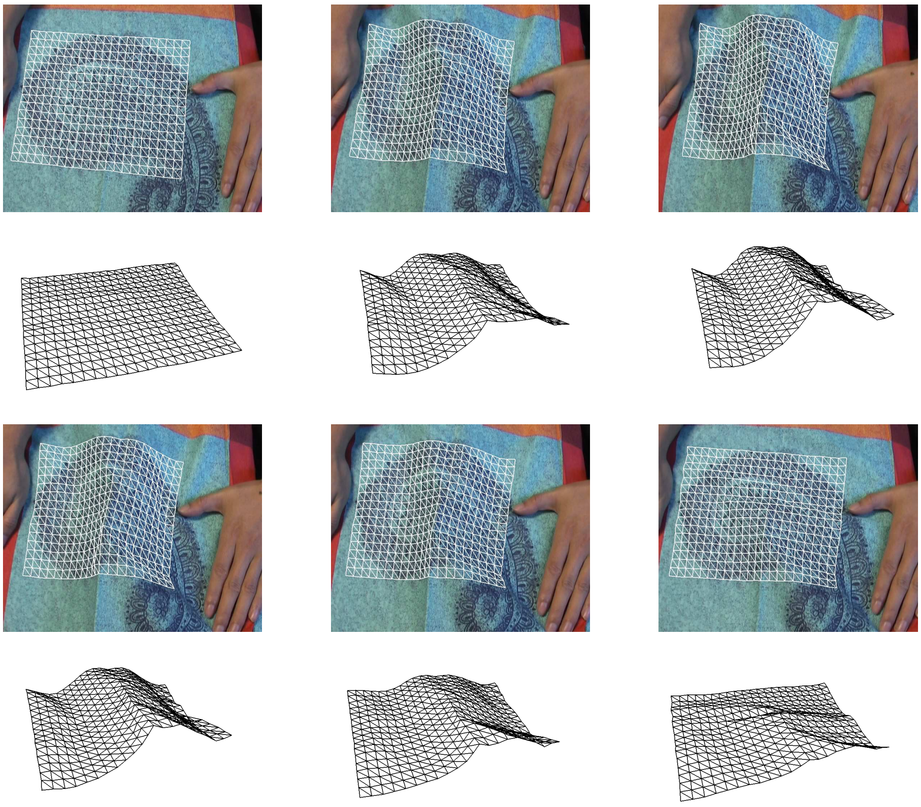


Figure 5.9: Recovering the deformation of a piece of cloth.

5.4 Summary

We have proposed a novel solution for the 3D deformable surface tracking by formulating the problem into an unconstrained quadratic optimization. Compared with previous convex optimization approaches, the proposed method enjoys several major advantages. Firstly, the presented method is very efficient without involving complicated SOCP constraints. Secondly, the proposed approach can handle large outliers and is more resilient to noises. Compared with the previous SOCP method, we have improved both the efficiency and robustness performance significantly. Furthermore, different from the previous SOCP approach that usually requires a sophisticated SOCP solver, the proposed method can be implemented easily in practice, requiring the solution of only a set of linear equations. Also, the optimization method used in this chapter might be applicable to other similar problems solved by SOCP. Extensive experimental evaluations are performed on objects made of different materials. The experimental results demonstrate that the proposed method is significantly more efficient than the previous approach, and is also rather robust to noises. Promising tracking results show that the proposed solution is able to handle large deformations that often occur in real-world applications.

Although promising experimental results have validated the efficiency and effectiveness of the methodology, some limitations should be addressed. First of all, self-occlusion problem has not yet been studied. Also, in some situations some jitter may occur due to a lack of texture information. Finally, this method requires the temporal information, which may limit its scope of the application.

□ **End of chapter.**

Chapter 6

Velocity Coherence Regression

In this chapter, we present a velocity coherence regression approach to nonrigid surface detection. Comparing to the methods introduced in the previous three chapters, the proposed approach does not require an explicit deformable mesh model. To handle the large numbers of outliers, an incremental outlier rejection scheme is presented. Several experiments have been conducted for performance evaluation, and encouraging experimental results demonstrate that the presented method is both effective and robust.

6.1 Motivation and Methodology

As mentioned in Chapter 3, nonrigid surface detection [76] can typically be treated as a problem of fitting a mapping function and rejecting outliers matching without homologies. The difference between nonrigid surface recovery and detection is that the latter does not require any initialization or a priori pose information. If the point correspondences contain no outliers, finding the nonrigid mapping function only requires the solution of a simple linear equation [123]. However, this ideal case seldom happens in any computer vision problem, and the outliers could comprise up to 90% of the points for a typical point matching problem [76]. Therefore, it is difficult to directly apply the robust techniques widely used in statistics. This is because a typical statistical estimator requires that the inliers must be the absolute majority of the data in order to obtain a reasonable solution [68, 119]. Moreover, some registration methods [6, 20] tend to be computationally expensive and mainly aim at object recognition rather than nonrigid surface recovery.

Chapter 3 has already thoroughly reviewed various approaches to nonrigid shape recovery, which employs either the feature-correspondences or appearance information. Alternatively, the method presented in this chapter is closely related the nonrigid point set matching methods, which are investigated in the following.

Considerable research efforts have been expended on the nonrigid shape matching from point sets in the image analysis and computer vision community [49, 67, 71]. Extensive studies can be found in the literature [10, 71]. Rangaranjan et al. [20] intend to establish a consistent correspondence between two point sets and recover the mapping function with the best alignment. They present a coarse-to-fine approach to jointly determine the correspondences and nonrigid transformation through deterministic annealing and soft-assign. The major problem of this approach is that the stability of the registration result is not guaranteed in the case of data with large outliers, and hence a good stopping criterion is required. In the most recent studies, the probabilistic approach for the nonrigid point set matching is attracting increasing research interests [49, 67, 71]. The point set matching is interpreted as a mixture density estimation problem [38], where one point set represents the centers of Gaussian mixture models and the other represents sample data. This problem is usually solved by the Expectation Maximization (EM) algorithm. Another idea is to model each of the two point sets by a kernel density function and then measure the similarity.

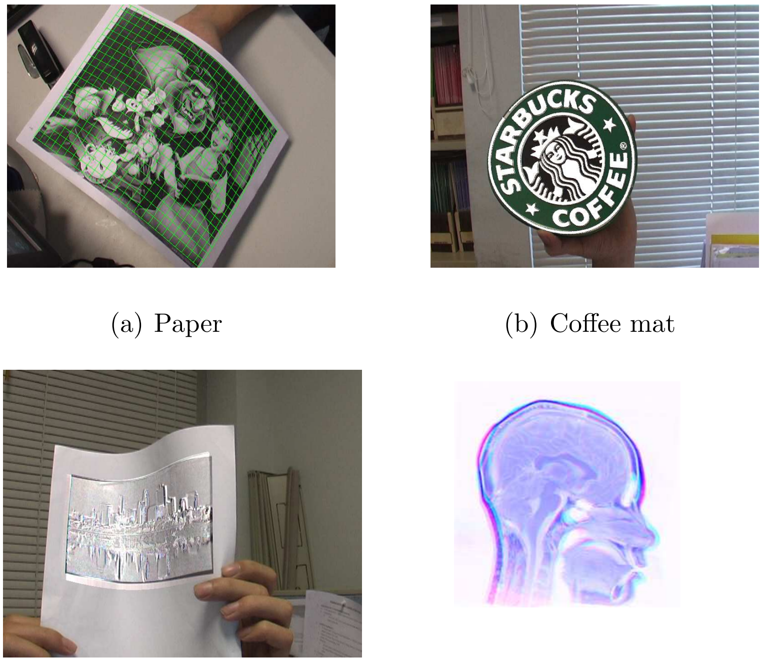
In [96], Tsin and Kanade proposed a kernel correlation based approach to register the nonrigid point set, which minimizes the L_2 norm between the distributions. Later, Jian and Vemuri [49] extended this approach via representing the density by Gaussian mixture models. All these methods employed the Thin-Plate Spline to obtain the smooth nonlinear transformation. Recently, Myroneko et al. [71] presented a coherent point drift method for nonrigid point set registration, which does not make any explicit assumption of the transformation model.

In contrast to the nonrigid surface detection problem, the methods for the point set matching [10, 71] tend to be computationally expensive, and few of them can be applied to point sets extracted from real images; an exception is the most recent part-based approach [67]. On the other hand, the nonrigid surface detection approach has already been applied to track nonrigid objects in real-time video [76, 123].

In nonrigid surface detection, the regularized deformable models are vitally important for dealing with the problem of the many outliers and ill-posed optimization, and thus are able to make the problem tractable by constraining the search space. We have introduced several regularization methods in Chapter 2, such as the Finite Element Model, the Thin-Plate Spline and the data embedding method. The Finite Element Model based regularization approach has been extensively studied in [32, 47, 76], which reported promising results in fitting noisy image data and handling deformable 3D objects. Using the Finite Element Model, however, the nonrigid surface should be explicitly represented by a triangulated mesh. Moreover, the triangulated Finite Element Model heavily relies on the quadratic regularization term, and does not always accurately represent large deformations [82]. Alternatively, the Thin-Plate Spline is well-known interpolation method widely used in point set registration, and mainly penalizes the second order derivatives [89]. Finally, the data embedding techniques, such as Principal Component Analysis [82, 105], can also be engaged as the regularization technique, although PCA requires a large number of training samples to obtain sufficient generalization capability.

In this chapter, a novel robust velocity coherence regression approach is proposed for the nonrigid surface detection, which has the advantage of employing the velocity coherence constraints [108] to regularize the deformation of the nonrigid surface. In contrast to the method presented in Chapter 3, the proposed approach does not require an explicit triangulated mesh representation for the nonrigid surface. Moreover, a coarse-to-fine scheme is proposed to handle large numbers of outliers. The key of the proposed robust velocity coherence regression method is to take advantage of imposing the velocity coherence smoothness on the underlying mapping. According to [108], the velocity coherence constraint penalizes the derivatives of all orders of the underlying velocity field, while both the Thin-Plate Spline and the Finite Element Model regularization methods can only penalize the second order derivative. To the best of our knowledge, few studies have been done to formulate the nonrigid surface detection as a generic regression problem. To evaluate the performance of the proposed algorithm, extensive experiments have been conducted on such diverse objects as a piece of paper, a coffee mat, T-shirt and a medical image, as shown in Fig. 6.1.

The rest of this chapter is organized as follows. In Section 6.2, the nonrigid surface detection problem is first formulated as a generic regression problem, and then a velocity coherence regression approach is proposed. Section 6.3 presents a robust velocity coherence regression method along with an incremental outlier threshold scheme to handle large numbers of outliers in the nonrigid surface detection. Section 6.4 provides the details of the experimental implementation and describes the experimental re-



(a) Paper

(b) Coffee mat

(c) A piece of paper

(d) Medical image

Figure 6.1: Detecting nonrigid surfaces in real-time video (a-d). (a) Paper with the registered mesh. (b) The contour is overlaid on the coffee mat. (c) A piece of paper. (d), Medical image registration.

sults. We discuss limitations in Section 6.5. Section 6.6 sets out the conclusion.

6.2 Velocity Coherence Regression

In this section, we describe the proposed velocity coherence regression approach for nonrigid surface detection. To tackle this challenge, the nonrigid surface detection is formulated as a regression problem under the regularization networks framework. Then, the velocity coherence regression is proposed to address this problem. Finally, we discuss the connections of the method with the Gaussian process regression and the Gaussian mixtures models.

6.2.1 Theoretical Framework

Consider two images I_m and I_t, which are the model image and the target image respectively. Let $\mathbf{x}_i$ be defined as the 2D coordinates of a feature point in the model image I_m, and $\mathbf{y}_i$ is the coordinates of its match in the target image I_t. A set of correspondences $M = \{(\mathbf{x}_i, \mathbf{y}_i) \in R^d\}_{i=1}^N$ between the model and target images are obtained through a point matching algorithm, where $d = 2$ and N is the total number of matched pairs.

The goal of the nonrigid surface detection is to find a function f to map the points in the model surface into the target image I_t. It can be viewed as a regression problem that has the input data of the correspondence pairs $(\mathbf{x}, \mathbf{y})$. Let vector $\mathbf{y}$ denote the measurements target of the input vector $\mathbf{x}$. We denote $\boldsymbol{\epsilon}$ as the noise variable, and then the function f can be written as $f(\mathbf{x}) = \mathbf{y} + \boldsymbol{\epsilon}$.

Given the set of correspondences M, the posterior probability of the mapping function f can be derived as below:

$$p(f|M) \propto p(M|f)p(f)$$

where $p(M|f)$ is the conditional probability of M given f, and $p(f)$ is the a priori probability of the random field f.

Assuming the noise variables $\boldsymbol{\epsilon}_i$ are normally distributed with variance σ. Thus, we can write $p(M|f)$ as:

$$p(M|f) \propto \exp\left(-\frac{1}{2\sigma^2}\mathrm{tr}\left[(Y - f(X))^\top (Y - f(X))\right]\right)$$

where $X \in R^{N\times d}$ and $Y \in R^{N\times d}$ are the matrices of the model and target points. In addition, the a priori probability $p(f)$ embodies a priori knowledge of the function f, and can be used to constrain the nonrigid surface model. As suggested in [30], $p(f)$ can be written as follows:

$$p(f) \propto \exp(-\frac{\lambda}{2}\phi(f))$$

where λ is a regularization coefficient, and $\phi(f)$ is a smoothness function.

Finally, finding the maximum a posteriori estimate of function f is equivalent to minimizing the following log-posterior energy function:

$$\begin{aligned} E_1 &= -\ln p(M|f) - \ln p(f) \\ &= \frac{1}{2\sigma^2}\mathrm{tr}\left[(Y - f(X))^\top (Y - f(X))\right] + \frac{\lambda}{2}\phi(f) \quad (6.1) \end{aligned}$$

which is the same problem as for the Tikhonov Regularization [30, 92].

6.2.2 Velocity Coherence Regularization

In this chapter, the velocity coherence constraint is employed to impose smoothness on the underlying mapping function. A continuous velocity function v is defined as follows:

$$v(\mathbf{x}) = f(\mathbf{x}) - \mathbf{x}$$

Instead of applying the regularization to the mapping function f, Andriy et al. [71] suggest constraining the velocity function v. This is similar to the technique used in optical flow estimation [44]. Therefore, according to the regularization networks [30], the smoothness function $\phi(v)$ is defined as:

$$\phi(v) = \frac{\lambda}{2}\int_{R^d} \frac{|\tilde{v}(\mathbf{s})|^2}{\tilde{G}(\mathbf{s})} d\mathbf{s}$$

where $\tilde{v}$ and $\tilde{G}$ are the Fourier transform of the velocity function v and a real symmetric function G respectively. $\tilde{G}$ is a positive function that tends to zeros as $\|\mathbf{s}\| \to \infty$; thus $1/\tilde{G}$ becomes a high-pass filter. Replacing the regularization term in the regularization networks (Eqn. 6.1) with $\phi(v)$, the energy function can be rewritten as below:

$$E_2 = \frac{1}{2\sigma^2}\mathrm{tr}\left[(Y - f(X))^\top (Y - f(X))\right] + \frac{\lambda}{2}\int_{R^d} \frac{|\tilde{v}(\mathbf{s})|^2}{\tilde{G}(\mathbf{s})} d\mathbf{s} \quad (6.2)$$

It is named as the velocity coherence regression in this thesis. It is shown (see the appendix B for a sketch of the proof) that the solution of the velocity coherence regression has the following form:

$$v(\mathbf{x}) = \sum \boldsymbol{\alpha}_i G(\mathbf{x} - \mathbf{x}_i)$$

where $\boldsymbol{\alpha}_i$ is a d-dimensional coefficient vector. Therefore, the mapping function f can be rewritten as below:

$$f(\mathbf{x}) = \mathbf{x} + \sum \boldsymbol{\alpha}_i G(\mathbf{x} - \mathbf{x}_i) \tag{6.3}$$

As discussed in [30], there are various choices for selecting the kernel form for G, such as Gaussian, multivariate splines and multiquadric, amongst others. In this chapter, we choose the Radial Basis Function (RBF) kernel function k, since it not only fulfills the requirements for the positive function G but also leads to a velocity coherence regularization [71]. Moreover, the regularization term $\phi(v)$ with a Gaussian kernel form is equivalent to the one used in [108], which can be derived as the sum of weighted squares of all order derivatives of the velocity field:

$$\int_{R^d} \frac{|\tilde{v}(\mathbf{s})|^2}{\tilde{G}(\mathbf{s})} d\mathbf{s} = \int_{R^d} \sum_{m=1}^{\infty} \frac{\rho^{2m}}{m!2^m} (D^m v) d\mathbf{s}$$

where D is a derivative operator such that $D^{2m}v = \nabla^{2m}v$. Let $K \in R^{m \times m}$ denote a kernel matrix with elements:

$$K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\frac{1}{2\rho^2}\|\mathbf{x}_i - \mathbf{x}_j\|^2)$$

where ρ is the width of the RBF kernel. In nonrigid surface detection, ρ is related to physical characteristics such as elasticity. For example, the nonrigid object becomes more rigid when ρ increases. Substituting Eqn. 6.3 into the energy function Eqn. 6.2, we can derive the following problem:

$$E_2 = \frac{1}{2\sigma^2}\mathrm{tr}[(Y - X - K\boldsymbol{\alpha})^\top (Y - X - K\boldsymbol{\alpha})] \\ + \frac{\lambda}{2}\mathrm{tr}(\boldsymbol{\alpha}^\top K \boldsymbol{\alpha}) \tag{6.4}$$

where matrix $\boldsymbol{\alpha} \in R^{m \times d}$ is a variable. Therefore, the derivatives of the energy function $E_2(\boldsymbol{\alpha})$ with respect to the variable $\boldsymbol{\alpha}$ vanish for optimality:

$$\frac{\partial E_2}{\partial \boldsymbol{\alpha}} = -\frac{1}{\sigma^2} K(Y - X - K\boldsymbol{\alpha}) + \lambda K \boldsymbol{\alpha}$$

which leads to the following linear equation:

$$\left(K + \sigma^2 \lambda I\right) \boldsymbol{\alpha} = Y - X \tag{6.5}$$

6.2.3 Link to Gaussian Process Regression

Assume the velocity function $v(\mathbf{x})$ is a zero mean Gaussian Process with squared exponential covariance function [78]:

$$v(\mathbf{x}) \sim \mathcal{GP}(0, k(\mathbf{x}_i, \mathbf{x}_j))$$

Then, the mapping function $f(\mathbf{x})$ can be written as a Gaussian Process with the deterministic mean function $\mathbf{x}$:

$$f(\mathbf{x}) \sim \mathcal{GP}(\mathbf{x}, k(\mathbf{x}_i, \mathbf{x}_j))$$

The prediction can be computed by:

$$f^*(X) = X + K(K + \sigma^2 I)^{-1}(Y - X) \tag{6.6}$$

6.3 Nonrigid Surface Detection

Generally speaking, the great challenge for nonrigid surface detection is to deal with the large numbers of outliers which are mainly introduced by the local feature matching. In the rest of this section, the proposed robust velocity coherence regression method is presented to attack this critical problem.

6.3.1 Robust Velocity Coherence Regression

Since the outliers are overemphasized using the L_2 norm in the velocity coherence regression, we employ a robust estimator $\mathcal{V}(\delta, \sigma)$ which assesses a fixed penalty for the residual $\delta = \mathbf{y} - f(\mathbf{x})$, which is larger than the variance σ. Moreover, this approach is relatively insensitive to the outliers [15]:

$$\mathcal{V}(\delta, \sigma) = \begin{cases} \|\delta\|, & M_1 = \{(\mathbf{x}, \mathbf{y}) | \quad \|\delta\| \leq \sigma^2\} \\ \sigma^2, & M_2 = \overline{M_1} \end{cases} \tag{6.7}$$

where the set M_1 contains the inlier matches, and M_2 is the set of the outliers. Note that this robust estimator is the same as the one used in Chapter 3 except that the order is fixed to two in the velocity coherence regression. As in EM-ICP [35], we introduce a weight ω_i associated with each correspondence, and then reformulate the energy function as follows:

$$E_3 = \frac{1}{2\sigma^2} \sum_{i=1}^{N} \omega_i \mathcal{V}(\mathbf{y}_i - f(\mathbf{x}_i), \sigma) + \frac{\lambda}{2} \mathrm{tr}(\boldsymbol{\alpha}^\top K \boldsymbol{\alpha}) \tag{6.8}$$

Note that a feature point in the model image I_m may be matched with multiple points in the target image I_t. Simply summing them together is equivalent to matching them with the center of these points in I_t, which may not be effective and efficient. In this case, we only retain the correspondences with the highest match score. Moreover, ω_i is the posterior probability, which decays exponentially as a function of distance, so that the large numbers of outliers have little influence on the minimization:

$$\omega_i = \frac{e^{-\frac{1}{2\sigma^2}\|\mathbf{y}_i - f(\mathbf{x}_i)\|^2}}{\sum_{j=1}^{N} e^{-\frac{1}{2\sigma^2}\|\mathbf{y}_i - f(\mathbf{x}_i)\|^2}}$$

The modified finite Newton method [54] can be employed to solve the unconstrained optimization problem in Eqn. 6.8, and the derivatives of E_3 can be derived as below:

$$\frac{\partial E_3}{\partial \boldsymbol{\alpha}} = -\frac{1}{\sigma^2}(Y - X - K\boldsymbol{\alpha})^\top W I^0 K + \lambda K \boldsymbol{\alpha}$$

where W is a diagonal weight matrix with $W_{ii} = \omega_i$ and I^0 is an $N \times N$ matrix with inlier entries being one and others zero. Therefore, we can obtain the following solution:

$$\left(W I_0 K + \sigma^2 \lambda I\right) \boldsymbol{\alpha} = W I^0 (Y - X) \tag{6.9}$$

Let $K' \in R^{l \times l}$ denote the matrix part of the inlier block in the original kernel matrix K, and l is the number of inlier matches. Since l is always less than N, the above linear system can be reduced to a smaller problem:

$$\left(W' K' + \sigma^2 \lambda I\right) \boldsymbol{\alpha}' = W'(Y' - X') \tag{6.10}$$

where W', Y' and X' refer to the inlier block of the corresponding matrices. The above linear system can be efficiently solved by LU decomposition. Moreover, it can be observed that the outliers are not involved into the computation. In addition, both the precision and computational cost of the method are dependent only on the number of inlier points.

After computing the mapping function f, the model image I_m can be warped to the target image I_t by predicting the velocity fields using Eqn. 6.3.

6.3.2 Optimization

In order to facilitate the velocity coherence regression, both the model and target point sets are normalized with zero mean and unit variance, which is equivalent to translating and scaling the point sets. To handle the large numbers of outliers, we introduce an incremental outlier threshold scheme. The variance σ that is also the support of the robust estimator $\mathcal{V}(\delta, \sigma)$ is progressively decayed at a constant rate γ. Since the derivatives of $\mathcal{V}(\delta, \sigma)$ are inversely proportional to the support σ, the regularization coefficient λ is kept constant during the optimization. For each value of σ, the object function E_3 is minimized until the inlier set no longer changes, and then the result is employed as the initial state for the next minimization. The minimization of E_3 is solved through Eqn. 6.10 with a given initial state, after which the inlier set is updated. To deal with the transformation, we re-normalize the input point set with respect to the inliers.

In order to select most of the correspondences into the initial active set, and to avoid getting stuck at local minima, the initial value of σ is usually set to a sufficiently large value. The optimization procedure stops when σ reaches a value close to the expected precision, and then the algorithm reports a successful detection when the number of inlier matches is above a given threshold. Ultimately, the proposed optimization scheme involves two or three iterations for each σ, and around twenty iterations in total to ensure the convergence.

Fast Computation: Since the optimization procedure only involves the part of the Kernel matrix K that is constant in the whole process, it can be pre-computed in order to save computational cost. Similarly, the projection matrix can also be pre-computed, which maps the mesh from the model to the target.

6.4 Experimental Results

In this section, we present the details of the experimental implementation and report the results of performance evaluation on nonrigid surface detection. It is shown that the proposed approach is both effective and efficient for real-time tracking, and can be easily employed for augmented reality applications. In addition, similarly convincing results are obtained for medical image registration.

6.4.1 Experimental Setup

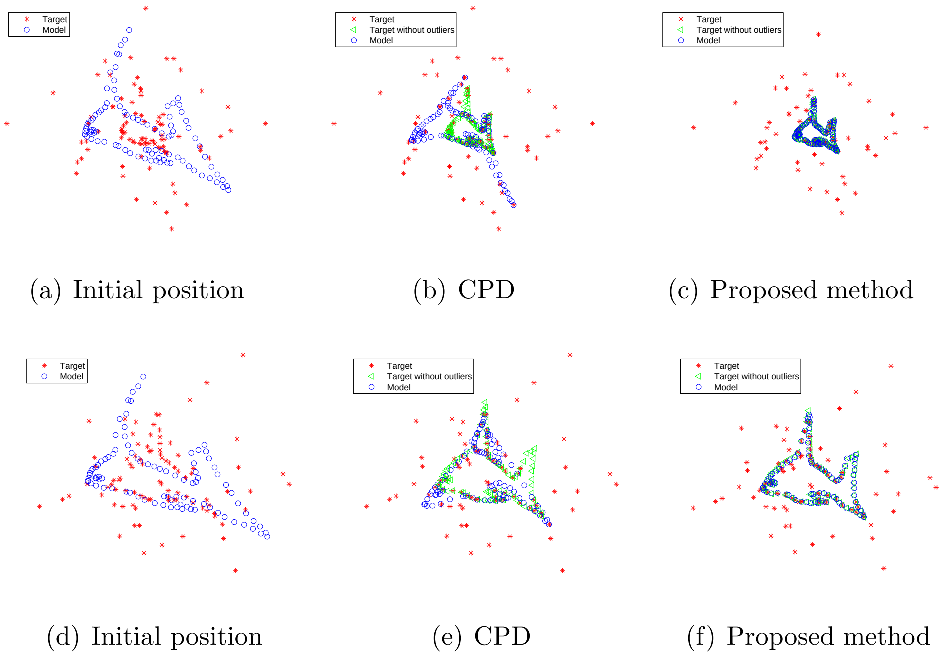
All the experiments reported in this chapter were carried out on a Pentium-4 3.0GHz PC with 1GB RAM, and a DV camera was employed to capture videos. As in Chapter 3, a random-trees based method [58] is used to build the correspondences between the model image and the target image, and a model image is acquired when the nonrigid surface contains no deformation.

In the experiments, a set of synthetic data is used to select the parameters, and the reference mesh is manually registered. The performance is evaluated by measuring the percentage of mesh vertices within two pixels of those in the reference mesh. The RBF kernel width ρ is set to 2.0, and the best regularization coefficient is found to be around 0.1 by grid searching. Similarly, the initial support is fixed to 1.0, and the decay rate is 0.7.

6.4.2 Nonrigid Surface Detection

The proposed robust velocity coherence regression is first evaluated on a point set registration problem. The targets are corrupted by the noise (55%) having a normal distribution with a standard deviation of two. Fig. 6.2 plots the registration results, which indicates that the proposed method can handle the larger outliers. The proposed method requires 11 iterations to obtain the convergence. However, CPD algorithm [71] needs above 100 iterations, and fails to achieve the convergence. Furthermore, CPD algorithm cannot handle the large pose variation, as shown in Fig. 6.3.

Then, the proposed approach is compared with with the state-of-the-art methods, such as the semi-implicit optimization [76] and the progressive finite Newton method presented in Chapter 3. A set of synthesized correspondences is generated



(a) Initial position

(b) CPD

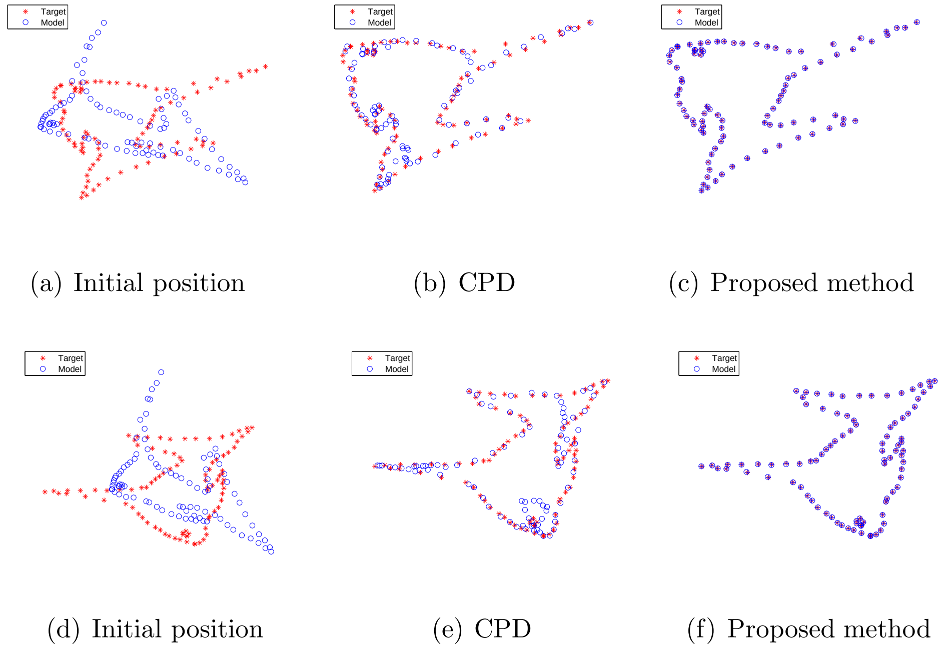
(c) Proposed method

(d) Initial position

(e) CPD

(f) Proposed method

Figure 6.2: Point set registration with large outliers.(a) initial position of model and target points; (b) CPD algorithm [71]; (c) proposed method.



(a) Initial position

(b) CPD

(c) Proposed method

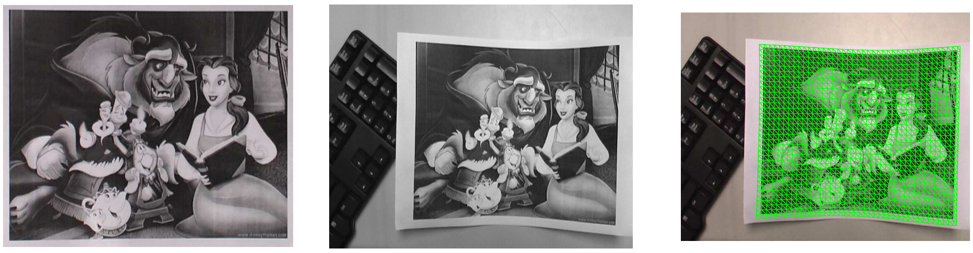
(d) Initial position

(e) CPD

(f) Proposed method

Figure 6.3: Point set registration with pose variation.(a) initial position of model and target points; (b) CPD algorithm [71]; (c) proposed method.

by a given transformation between the model image and the test image, as shown in Fig. 6.4. The experiments are conducted on the observations with added Gaussian noise. Each experiment is repeated with 50 runs. Table 6.1 shows the experimental results with different standard deviations for the added noise, which indicates that the proposed method performs very similarly to the gradient method. Moreover, the performance of nonrigid surface detection algorithm is mainly determined by the accuracy of the feature matching methods. However, it can be also observed that the implementation of semi-implicit method does not perform well. This is mainly due to lack of a Levenberg-marquardt scheme to properly tune the parameters.



(a) Model image (b) Test image (c) Test image with overlaid mesh

Figure 6.4: Model image and test image used in the numerical comparison. The overlaid mesh is employed as the ground truth mapping.

Table 6.1: The accuracy on synthesized dataset with different standard deviations *std*. The mean square error (MSE) per vertex is adopted as the metric.

Method	$std = 1$	$std = 2$	$std = 5$	$std = 8$	$std = 10$
[76]	0.88 ± 0.09	1.72 ± 0.10	4.01 ± 0.33	6.26 ± 0.68	7.45 ± 0.71
[123]	0.72 ± 0.02	1.43 ± 0.05	3.59 ± 0.14	5.69 ± 0.29	7.08 ± 0.38
VCR	0.74 ± 0.02	1.45 ± 0.04	3.86 ± 0.09	5.68 ± 0.17	7.13 ± 0.37

Fig. 6.5 illustrates the results on a piece of paper. To eval-

uate the effectiveness of the proposed method, the grid mesh is mapped from the model image to the input image. It can be observed that the velocity coherence regularization technique is robust to the large deformations. The regularization networks with the robust estimator is also evaluated in this case, which fails to converge for all the cases.

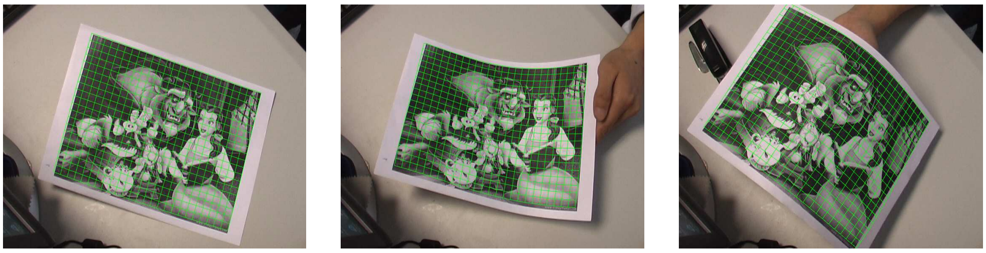


Figure 6.5: Nonrigid surface detection on a piece of paper, the detected mesh model is overlaid on the input images size of 720 × 576.

Complexity: The complexity of the proposed method is determined by the order of Eqn. 6.10, which is equal to the number of inlier matches. On the other hand, the complexity of the Finite Element Model-based method [76, 123] is dominated by the number of vertices in the mesh model. For the paper video, as shown in Fig. 6.6, the proposed method runs around 15 frames per second on real-time video with size of 720 × 576, which requires about 20 iterations to achieve the convergence. As described in Chapter 3, our own implementation of the semi-implicit iterative approach [76] needs around 40 iterations to reach the convergence, and runs about 9 frames per second with a mesh of 120 vertices. Thus, the proposed method is more efficient than the semi-implicit iterative approach. As for the progressive finite Newton approach, it runs around 18 frames per second.

Augmented Reality: The proposed method is also applied to re-texturing an image. To obtain realistic results, the texture should be correctly relighted. As suggested in [76], a re-textured

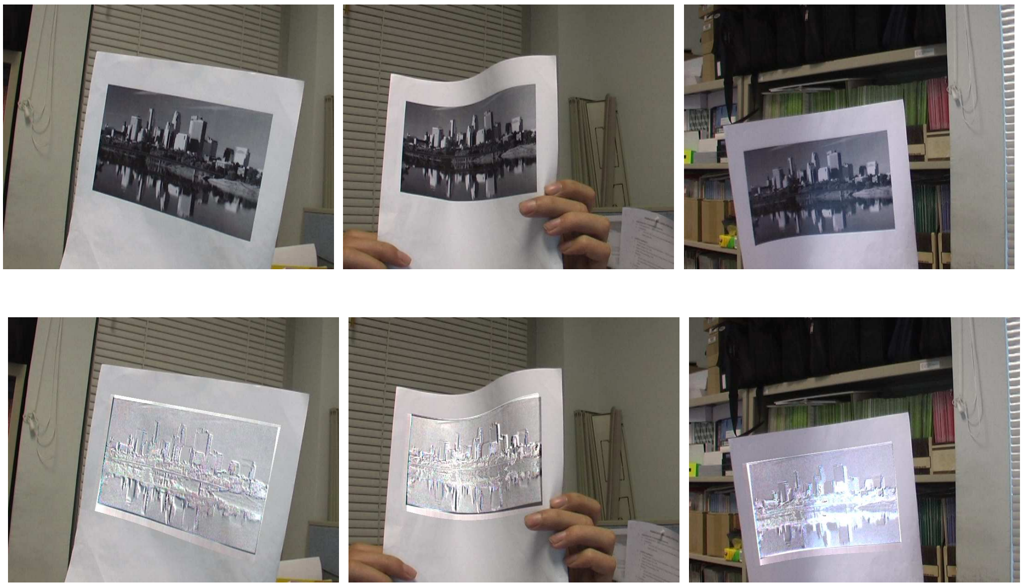


Figure 6.6: Re-texturing a picture on a piece of paper. The first row is the 720 × 576 images captured by a DV camera. The second row is the results of replacing the pure white pattern.

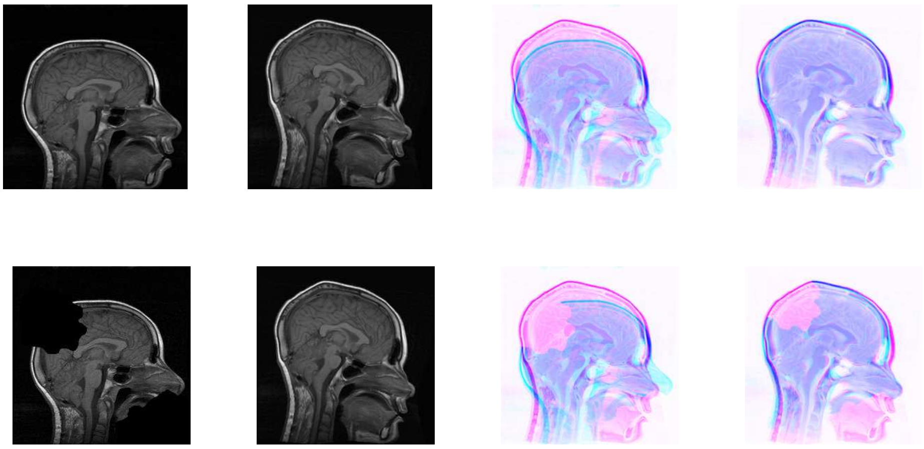
input image is generated by multiplying a blank shaded image, which is the quotient of the input image and the warped reference image. This relighting procedure is easily done by the GPU and requires only a short OpenGL shading language program; and the whole process runs at around 15 frames per second. Fig. 6.6 describes the results on a piece of paper with a saturated region. In addition, the right two columns of Fig. 6.6 show the results in a cluttered environment, and the last one shows the result with partial occlusion. As it is another feature-based method, the performance of the proposed method is closely related to the texture of objects. Specifically, better results can be obtained for objects with more texture, because it is easy to find more correct correspondences than with those lacking texture.

6.4.3 Medical Image

The proposed approach is also evaluated for medical image registration. A pair of sagittal images [74] with the size of 256×256 from two different patients are used in the experiments. The source and target images differ in both geometry and intensity. The results are plotted in Fig. 6.7; it can be seen that the source image is successfully registered. In comparison with the locally affine but globally smooth method [74], which takes about four minutes, the proposed method can solve the problem within half a second. Moreover, the sparse correspondences-based method can naturally handle the missing data and the partial occlusion problem. As shown in Fig. 6.7, even with the source images in a region removed, the nonrigid shape can still be recovered. Since it is a fully automated approach, we can employ the fitting result to initialize other local methods [74] in order to further improve the registration accuracy.

6.5 Discussion

A robust velocity regression method with an incremental outlier threshold scheme has been proposed. Note that the proposed methodology could be applied to solving other flow related estimation problems. In this thesis, however, we restrict its application to nonrigid surface detection. Comparing to the semi-implicit iterative method [76] and the progressive finite Newton approach in Chapter 3, the proposed method makes no assumption about the model except for the velocity coherence, which is independent of an explicit mesh model. Moreover, the velocity coherence regularization can be infinite order, while the Finite Element Model-based methods only consider the second order regularization. Similar to the progressive finite Newton method, it is easy to implement the proposed approach, which



(a) Source (b) Target (c) Before (d) After

Figure 6.7: Applying the proposed method to medical image registration. A pair of sagittal images from two different patients is shown. (a,b,d) are the source, target and registered source respectively. (c) and (e) are the overlaid images before and after registration. The second row displays the synthetic example with missing data.

only involves solving a linear equation, and does not require tuning of the viscosity parameters or a sophisticated Levenberg-Marquardt optimization algorithm. Although the linear system of the present method is not a sparse one as in [76], experimental results indicate that the proposed approach is more efficient than the semi-implicit iterative method. This is mainly because the present optimization scheme requires fewer iterations and the problem size is greatly reduced. While it is difficult to establish ground truth in non-rigid registration, we just show the qualitative comparisons empirically. In the experiments, the accuracy largely depends on the feature matching algorithm, and all methods perform similarly.

Although promising experimental results have validated both the effectiveness and efficiency of the proposed approach, some limitations still exist. First of all, as this is another feature correspondence-based method, some jitter may occur due to the point matching algorithm or the lack of texture information, just like the method presented in Chapter 3. Second, the robust estimator used in the method cannot handle multi-structured data, while detection of multiple nonrigid surfaces is still an open issue.

6.6 Summary

It is clear that the proposed novel approach to nonrigid surface detection is powerful and effective. It offers several distinct advantages over the semi-implicit method. Firstly, this method makes no assumption about the model except for the velocity coherence. Moreover, the robust velocity coherence regression takes advantage of the robust estimator and progressive optimization scheme, and can handle the data with large numbers of outliers. In addition, in contrast to the previous approaches involving iterative and explicit minimization, the proposed optimization scheme requires fewer iterations. Finally, the proposed method is both robust and efficient, and can handle large deformations and illumination changes.

The proposed approach has been tested in several applications, such as real-time Augmented Reality and medical image registration. Encouraging experimental results show that the proposed approach is both effective and promising.

□ **End of chapter.**

Chapter 7

Near-duplicate Keyframe Retrieval

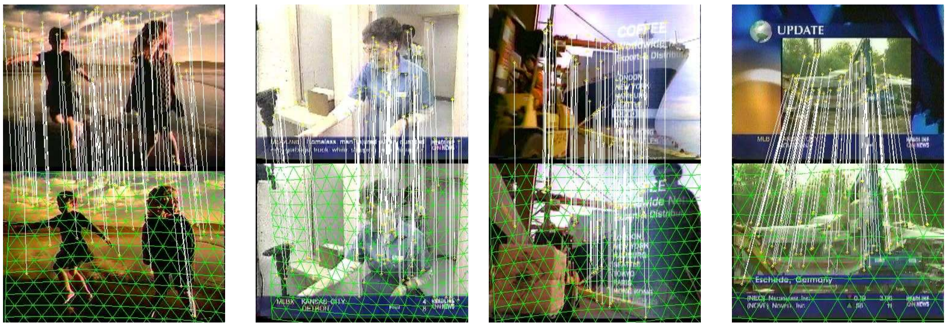
In this chapter, we apply the technique developed in Chapter 3 to tackle the task in multimedia domain: near-duplicate keyframe retrieval from real-world video corpora. In contrast to previous approaches, the presented technique can recover an explicit mapping between two near-duplicate images with a few deformation parameters and find out the correct correspondences from noisy data effectively. To make the presented technique applicable to large-scale applications, we suggest an effective multilevel ranking scheme that filters out the irrelevant results in a coarse-to-fine manner. In the proposed scheme, to overcome the extremely small training size challenge, a semi-supervised learning method is employed for improving the performance using unlabeled data.

7.1 Motivation

Near-Duplicate Keyframes (NDK) refer to the pairs of keyframes in a video corpus, for which the two keyframes of a pair are closely similar to each other apart from minor differences due to the variations of capturing conditions, rendering conditions, or editing operations [109, 113, 120]. NDK detection and retrieval techniques are beneficial for many real applications, such as news video search [94] and copyright infringement detection [53, 77]. NDK retrieval is a challenging research problem due to some well-known factors. One is that videos from different sources may be captured by devices with different hardwares under a variety of illumination conditions. Moreover, video editing often produces extra photometric and geometric transformations and occludes the original video by adding captions. Figure 7.1 shows some examples of pairs of duplicate keyframes extracted from the TRECVID2003 video corpus.

In the past years, there has been a surge of research attention on this topic in the multimedia community [53, 77, 103, 104, 109, 113, 120]. Some conventional methods extend content-based image retrieval (CBIR) techniques for the NDK detection and retrieval task; these often employ global features extracted from the whole image, such as color moment and color histogram [77, 109]. Although these methods are usually very efficient in finding identical copies, they may not be very accurate for real NDKs as they often fail to address the variations of lighting changes, viewpoint changes, and occlusions.

Alternatively, some recent approaches using local feature point correspondences can deal with the illumination variations and geometric transformations by exploring the recent advances in local feature descriptors [69]. These approaches often incur heavy computational cost in feature matching. Nevertheless, some efficient solutions have been proposed. For example, Ke et al. [53] proposed an efficient method using PCA-SIFT and locality-sensitive hashing indexing. However, their method often makes a *rigid* projective geometry assumption, which may suffer from some outlier matches due to lens changes and small object movements. Zhang and Chang [109] presented a stochastic Attributed Relational Graph (ARG) matching framework, which involves a computationally intensive process of stochastic belief



(a) $19/\frac{52}{0.73}/92$ (b) $26/\frac{58}{0.76}\ /107$ (c) $15/\frac{75}{0.72}/104$ (d) $56/\frac{76}{0.68}/148$

Figure 7.1: Some near-duplicate keyframes examples selected from TRECVID2003 video corpus. The caption of each subfigure shows the total number of inlier matches with each of the three methods: projective geometry, OOS-SIFT method (PE is below the number of inliers), and the presented NIM method. Since $PE > 0.5$, OOS-SIFT method failed in (a-d).

propagation. Zhao et al. [113] proposed a one-to-one symmetric (OOS) matching method, which applies a local smoothing constraint to remove the outlier matches. In [72], Pattern Entropy (PE) is employed as similarity measure for OOS method. Similar to other *bipartite graph matching* methods, the OOS method considers only pairwise matches and fails to explore the *spatial coherence* between the two sets of interest points in two NDKs. As shown in Figure 7.1, illumination variations, occlusions and zooming lead to large PE, in which $PE \leq 0.5$ is considered as NDK pair [72].

In contrast to previous approaches employing either rigid projective models or bipartite graph matching, in this chapter, we apply the nonrigid surface detection method developed in Chapter 3 to retrieving near-duplicate keyframes. Instead of detecting a patch of deformable surface from video, this method is employ to matching two images. Therefore, we rename the technique presented in 3 as **Nonrigid Image Matching** (NIM) which will be used in the following part of this chapter. Unlike the previous approaches, the proposed NIM method assumes that there may exist nonrigid transformations between the two NDKs. The key to solving the NIM problem is based on the methodology presented in Chapter 3, which takes advantage of a closed-form solution for a given set of correspondences. Since the presented method takes consideration of local deformations, it often obtains more inlier matches than regular rigid projective models and the OOS graph matching method. This characteristic plays a critical role in duplicate similarity matching. Figure 7.1 shows some examples along with the total numbers of inlier matches found by three different methods on the same set of extracted SIFT features [62].

Compared to the previous approaches, the proposed NIM method not only delivers better retrieval performance, but also enjoys some other salient merits. For example, the method is able to find the exact matching region between two NDKs, which is usually not obtained by conventional methods. This attractive feature is important for part-based or sub-image detection and retrieval. In addition, the presented method is rather efficient, processing about ten pairs of keyframes per second on a regular PC with moderate configuration. To further accelerate the presented technique for large-scale applications, I suggest a Multi-Level Ranking (MLR) framework for efficient NDK retrieval, which integrates three different ranking components in a unified solution: nearest neighbor ranking, semi-supervised ranking, and NIM-based ranking.

In summary, this chapter includes three main contributions. First of all, the **Nonrigid Image Matching** technique is applied to retrieving and detecting NDK, which is significantly different from the conventional approaches. The presented technique overcomes some limitations with the existing approaches and hence offers better performance for solving the NDK detection and retrieval tasks. Secondly, to enable the proposed technique applicable to large-scale applications, we suggest a

Multi-Level Ranking framework that can effectively filter out irrelevant results so as to significantly reduce the sample size for the NIM comparisons. Although this is not the first use of the MLR approach by multimedia researchers [40, 41], our contribution is to validate its effectiveness by improving the NIM scheme in the NDK retrieval tasks. The third major contribution is to employ a **Semi-Supervised Ranking** (SSR) method with a *Semi-Supervised Support Vector Machine* (S^3VM) for improving the NDK learning task, which often has extremely few labeled data. The SSR method effectively improves the filtering performance of traditional supervised learning approaches by taking advantage of unlabeled data information.

The rest of this chapter is organized as follows. Section 7.2 reviews some existing approaches for NDK detection and retrieval. Section 7.3 proposes the nonrigid image matching method for detecting NDK with local feature correspondences. Section 7.4 presents a multi-level ranking scheme together with a semi-supervised SVM method for NDK retrieval. Section 7.5 provides experimental results and details of the experimental implementation. Section 7.6 sets out the conclusions.

7.2 Related Work

There are numerous research efforts devoted to the near-duplicate image/keyframe detection and retrieval in the multimedia community [53, 77, 102, 104, 109, 120]. In general, most of the existing approaches can be roughly divided into two categories: *appearance-based methods* and *local feature-based methods.*

The appearance-based methods often measure the similarity between two keyframes based on the extracted global visual features, such as color histogram [109] and color moments [112]. These methods are advantageous for their high efficiency since keyframes are often compactly represented in the vector space and thus can be solved efficiently by adapting conventional CBIR methods and mature data indexing techniques [77]. But they are often not very robust to illumination changes, partial occlusions, and geometric transformations.

On the other hand, the local feature-based methods detect local keypoints in two keyframes and measure their similarity by counting the number of correct correspondences between two keypoint sets. Keypoints are the salient regions detected over image scales and their descriptors are often invariant to certain transformations and variations. They overcome the limitations of the global appearance-based methods, and thus often achieve better performance [53, 113]. But they may incur a heavy computational cost for the matching of two keypoint sets, which may contain more than one thousand keypoints.

Recently, local feature-based methods have been actively studied. Sivic et al. [87] employed the local keypoints approach for object matching and retrieval in movies. Ke et al. [53] employed the compact PCA-SIFT feature and speeded up the search of nearest keypoints with the locality sensitive hashing technique for duplicate image detection and retrieval. Zhao et al. [113] proposed an OOS matching approach to NDK detection and reported state-of-the-art performance. The key of the OOS method is to eliminate noisy outliers during the one-to-one bipartite graph matching process. Most of these methods fall in the same category of point-to-point bipartite graph matching.

The NIM technique proposed in this chapter goes beyond conventional point-to-point bipartite graph matching methods. In contrast to existing techniques, the presented method is able to recover the explicit nonrigid mapping between two near-duplicate keyframes with nonrigid transformation models and can effectively find the correct correspondences from noisy data. Though similar techniques are actively being studied for tracking in computer vision and graphics [123, 124], to the best of our knowledge, we are the first to study it comprehensively for NDK retrieval tasks.

7.3 Nonrigid Image Matching

In this section, we present the nonrigid image matching approach to near-duplicate keyframe detection. We first give the formulation of the nonrigid image matching problem, and then solve it by a coarse-to-fine optimization technique.

7.3.1 Nonrigid Image Matching

Instead of assuming an affine transformation or projective geometry as in the conventional methods, we employ the nonrigid mapping relation between the NDKs. Therefore, the proposed method can tackle not only geometric transformations and viewpoint changes, but also small object movements. *Nonrigid Image Matching* refers to the problem of recovering the explicit mapping between the two images with a few deformation parameters and finding out the correct correspondences from noisy data simultaneously. It has been successfully applied to real-time nonrigid surface tracking in computer vision [76, 123, 124]. Unlike the nonrigid image registration, the NIM method is fully automatic and does not require manual initialization.

As mentioned in Section 7.1, the underlying technique in NIM is the same as the nonrigid surface detection. Therefore, the technical detail for NIM is omitted in this chapter; please refer to Chapter 3 for the detailed description. Since the definitions of task for nonrigid surface detection and NIM are quite different, it is necessary to make some clarifications. Comparing to the nonrigid surface detection task in Chapter 3, NIM does not make restriction on the acquisition of the template image, in which the images being compared is directly employed in the matching process. Moreover, any image in the comparison pair can be selected as the template image, and the other one is viewed as the input image. In NDK retrieval, the query image is usually served as the template image. To reduce the computational cost, the stiffness matrix and barycentric coordinates for the query image are pre-computed in each query. Furthermore, NIM is optimized for finding as many as possible correct inlier matches rather than precisely registering the model mesh onto the input image in the nonrigid surface tracking task. To this end, NIM adopts a relatively small regularization coefficient in order to allow the large deformations.

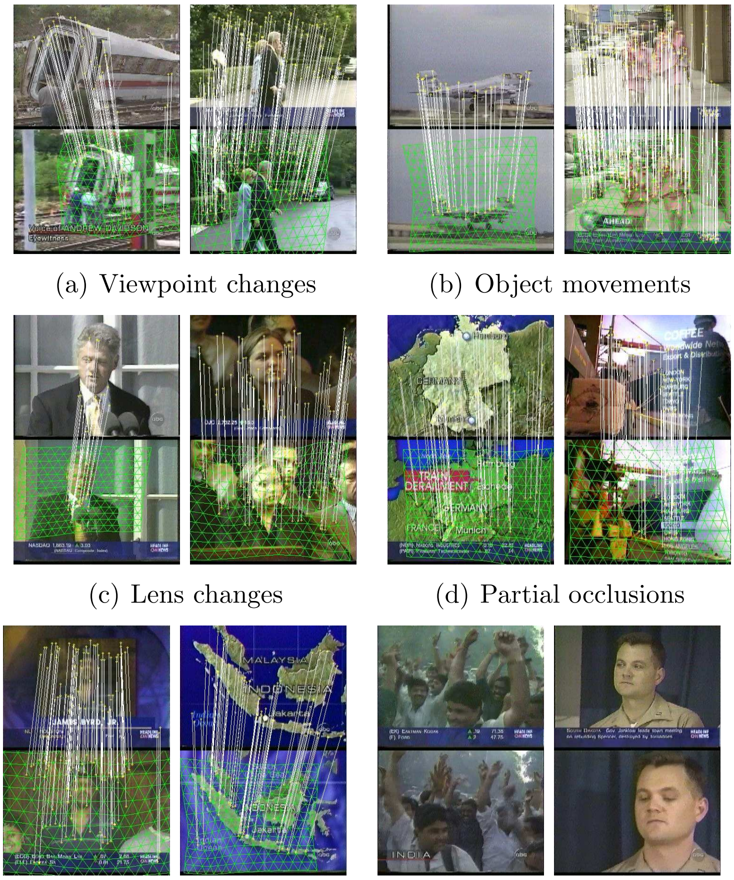
7.3.2 Case Studies: Detecting Various NDKs

To illustrate how the proposed NIM technique can effectively detect various NDKs appearing in news video domains, we show part of the detection results to demonstrate the advantages of the presented technique.

Figure 7.2 shows some examples of the successful detection results for various NDKs. All results on the duplicate pairs from Columbia's TRECVID2003 dataset are available [1]. In particular, the proposed NIM technique can effectively detect a variety of NDKs including, but not limited to, the following cases:

- **Viewpoint change.** This is very common for the shots extracted from news video sequences.
- **Object movement.** This is due to the relative movements caused by the camera or some objects.
- **Lens change.** This case is caused by the changes of camera lens, such as zooming in or zooming out.
- **Partial occlusion.** This case arises from the added captions or text descriptions in the videos.

[1] http://www.cse.cuhk.edu.hk/~jkzhu/dup_detect.html



(a) Viewpoint changes (b) Object movements

(c) Lens changes (d) Partial occlusions

(e) Subimage duplicates (f) Failure cases

Figure 7.2: Examples of the detection results on various near-duplicate keyframe cases.

- **Subimage duplicate.** Such duplicates could be caused either by lens changes or some editing effects.

We also investigate the failure cases, which are shown in Fig. 7.2(f). This is mainly due to either the difficulty in finding the nonrigid mapping or too few feature points in the keyframes.

7.4 Multi-Level NDK Retrieval

7.4.1 Framework Overview

Although the proposed NIM is efficient for matching two images in comparison with conventional local feature matching techniques [104, 113], directly applying NIM to large-scale applications could still be computationally intensive. To improve the efficiency and scalability of the proposed solution, we employ a Multi-Level Ranking (MLR) framework for efficiently tackling the NDK retrieval task. This strategy has been widely used, which is also shown to be successful in multimedia retrieval [40, 41]. In particular, the multi-level ranking scheme integrates three different ranking components:

- **Nearest Neighbor Ranking (NNR).** This is to rank the keyframes with simple nearest neighbor search.
- **Semi-Supervised Ranking (SSR).** This is to rank the keyframes with a semi-supervised ranking method.
- **Nonrigid Image Matching (NIM).** This is to rank the keyframes by applying the proposed NIM method.

The first two ranking components are based on global features for efficiently filtering out the irrelevant results, and the last component provides a fine re-ranking based on the local features. Figure 7.3 shows the proposed MLR framework, which attacks the NDK retrieval task in a coarse-to-fine ranking manner. This makes the proposed NIM solution applicable to large-scale real-world applications.

7.4.2 Formulation as a Machine Learning Task

The NDK retrieval problem can be formulated as a machine learning task with a query set of labeled image examples $\mathcal{Q} =$

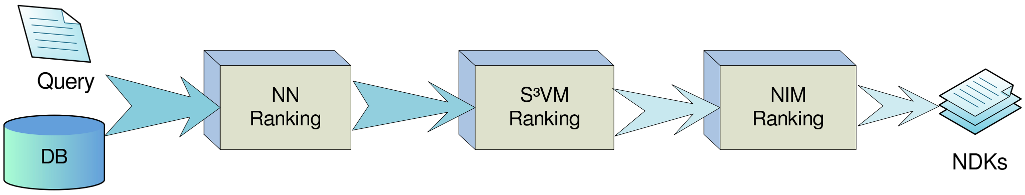


Figure 7.3: A multi-level ranking framework.

$\{(\mathbf{x}_1, +1), \ldots, (\mathbf{x}_l, +1)\}$ and a gallery set of unlabeled image examples $\mathcal{G} = \{\mathbf{x}_{l+1}, \ldots, \mathbf{x}_{l+u}\}$, where each image example $\mathbf{x}_i \in R^d$ is represented in a d-dimensional feature space. The goal of the learning task is to find the relevant near-duplicate examples from $\mathcal{G}$ that are closest to being exact duplicates of examples in $\mathcal{Q}$.

The learning task is tough on account of two difficulties. One is that there is no negative examples available, as only a query set $\mathcal{Q}$ will be provided in the retrieval task. The other is the small sample learning issue: Very few labeled examples will be provided in the retrieval task. To overcome the first difficulty, we adopt the idea of pseudo-negative examples used in previous multimedia retrieval approaches [107]. Specifically, we can conduct a query-by-example retrieval for ranking the unlabeled data in $\mathcal{G}$ based on their distances from the examples in the query set. Then we select a short list of most dissimilar examples as the negative examples based on the Nearest Neighbor ranking results.

To this end, with both positive and negative examples, we can formulate the learning task as a general binary classification task, which can then be solved by existing classification techniques. In the proposed approach, we apply Support Vector Machines (SVM) for the learning task. SVM is a well-known and state-of-the-art learning technique [97], which we briefly review here. SVM is used for learning an optimal hyperplane with maximal margin, and can learn nonlinear decision boundaries by exploiting powerful kernel tricks. SVM can be generally formulated in a regularization framework:

$$\min_{f \in \mathcal{H}_K} \frac{1}{l} \sum_{i=1}^{l} \max(0, 1 - y_i f(\mathbf{x}_i)) + \lambda \|f\|_{\mathcal{H}_K}^2 \tag{7.1}$$

where f is the hyperplane function $f(\mathbf{x}) = \sum_{i=1}^{l} \alpha_i k(\mathbf{x}, \mathbf{x}_i)$, k is some kernel function, and $\mathcal{H}_K$ is the associated reproducing kernel Hilbert space.

While SVM can be applied for solving the learning task, its performance may be poor when there are very limited number of labeled examples. This is a critical issue of an NDK retrieval since only extremely few positive examples will be provided. To overcome the second difficulty, we next introduce a semi-supervised learning technique for exploring both labeled and unlabeled data for the retrieval tasks.

7.4.3 Semi-supervised Support Vector Machine

To overcome the challenge of small sample learning, we suggest a semi-supervised retrieval (SSR) approach to attack the learning task via a semi-supervised SVM technique. Semi-supervised learning has been extensively studied in recent years, and numerous approaches have been proposed to exploit it [106, 114, 118]. In this chapter, we employ a unified kernel learning approach for semi-supervised SVM. The key idea is to first learn a data-dependent kernel from the unlabeled data, and then apply the learned kernel to train a supervised SVM based on the regularization learning framework. In the presented approach, we adopt the kernel deformation principle for learning a data-dependent kernel from unlabeled data [86].

The main idea of kernel deformation is to first estimate the geometry of the underlying marginal distribution from both labeled and unlabeled data, and then derive a data-dependent kernel by incorporating estimated geometry [86]. Let $\mathcal{H}$ denote the original Hilbert space reproduced by kernel function $k(\cdot,\cdot)$, and $\widetilde{\mathcal{H}}$ denote the deformed Hilbert space. In [86], the authors assume the following relationship between the two Hilbert spaces:

$$< f, g >_{\tilde{\mathcal{H}}} = < f, g >_{\mathcal{H}} + \mathbf{f}^{\top} M \mathbf{g}$$

where $f(\cdot)$ and $g(\cdot)$ are two functions, $\mathbf{f} = (f(\mathbf{x}_1), \ldots, f(\mathbf{x}_1))$ evaluates the function $f(\cdot)$ for both labeled and unlabeled data, and M is the distance metric that captures the geometric relationship among all the data points. The deformation term $\mathbf{f}^{\top} M \mathbf{g}$ is introduced to assess the relationship between the functions $f(\cdot)$ and $g(\cdot)$ based on the observed data. Given an input kernel k, the explicit form of the new kernel function $\tilde{k}$ can be derived as below:

$$\tilde{k}(\mathbf{x}, \mathbf{y}) = k(\mathbf{x}, \mathbf{y}) + \kappa_{\mathbf{y}}^{\top} \mathbf{d}(\mathbf{x})$$

where $\kappa_{\mathbf{y}} = (k(\mathbf{x}_1, y), \ldots, k(\mathbf{x}_n, y))^{\top}$. The coefficient vector $\mathbf{d}(\mathbf{x})$ can be computed by: $\mathbf{d}(\mathbf{x}) = -(I + MK)^{-1} M \kappa_{\mathbf{x}}$, where $K = [k(\mathbf{x}_i, \mathbf{x}_j)]_{n \times n}$ is the original kernel matrix for all the data, and $\kappa_{\mathbf{x}} = (k(\mathbf{x}_1, z), \ldots, k(\mathbf{x}_n, z))^{\top}$. To capture the underlying geometry of the data, a common approach is to define M as a function of graph Laplacian L, for example, $M = L^p$ where p is an integer. A graph Laplacian is defined as $L = \mathrm{diag}(S\mathbf{1}) - S$, where $\mathbf{1}$ denotes a vector with all one elements. Moreover, $S \in R^{n \times n}$ is a similarity matrix and each element $S_{i,j}$ is calculated by:

$$S_{ij} = S_{ji} = \begin{cases} e^{-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|_2^2}{2\varsigma^2}}, & \mathbf{x}_i \text{ and } \mathbf{x}_j \text{ are adjacent,} \\ 0, & \text{otherwise,} \end{cases}$$

where ς denotes the kernel width for a graph Laplacian. Various similarity measures can be used to build the adjacent matrix, such as L_1 norm, L_2 norm, and cosine similarity.

```
Algorithm Semi-Supervised SVM Re-ranking
Input
    • $X$: extracted the features for all images in the dataset
    • $k$: input kernel function
    • Regularization parameter $\gamma_A$ and $\gamma_I$, the graph Laplacian parameters
Procedure
1: Perform nearest neighbor ranking for each query image, and record the most dissimilar samples.
2: Calculate initial kernel matrix $K$: $K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$
3: Compute graph Laplacian $L$ and semi-definite positive matrix $M$
4: Calculate each element of semi-supervised kernel matrix:
        $\tilde{k}(\mathbf{x}, \mathbf{y}) = k(\mathbf{x}, \mathbf{y}) - \kappa_{\mathbf{y}}^{\top}(I + MK)^{-1} M \kappa_{\mathbf{x}}$
5: For each query image:
    • Train SVM with semi-supervised kernel $\tilde{K}$.
    • Rank the samples in the gallery set by the trained SVM.
Output
    • Rank list for each query image
End
```

Figure 7.4: Semi-Supervised SVM Re-ranking Algorithm

Consequently, the new kernel k can be formulated as follows:

$$\tilde{k}(\mathbf{x}, \mathbf{y}) = k(\mathbf{x}, \mathbf{y}) - \kappa_{\mathbf{y}}^{\top}(I + MK)^{-1} M \kappa_{\mathbf{x}} \tag{7.2}$$

Hence, replacing the kernel k in Eqn. 7.1 by the kernel $\tilde{k}$ in Eqn. 7.2, we can train the semi-supervised SVM classifier. Note that Eqn. 7.2 can also be used to compute the kernel for transductive learning, and the new deformed kernel matrix $\tilde{K} \in R^{n \times n}$ can be derived as below:

$$\tilde{K} = K - \mathcal{K}(I + MK)^{-1} MK \tag{7.3}$$

It can be simplified through the Kailath Variant:

$$\tilde{K} = (I + KM)^{-1}K$$

Moreover, the above equation is equal to

$$\tilde{K} = K(I + MK)^{-1} \tag{7.4}$$

From above all, we summarize the complete S^{3}VM re-ranking algorithm into Fig. 7.4.

7.5 Experiments

In this section, the empirical study of the proposed techniques for NDK retrieval is reported. Two key techniques will be evaluated comprehensively in the experiments. The first experiment is to examine the effectiveness of the *Multi-Level Ranking* scheme for filtering out the irrelevant results. In particular, we would like to examine whether the semi-supervised ranking method using S^3VM is more effective than the conventional ranking approaches. The second and more important experiment is to evaluate the performance of the proposed NIM technique for NDK retrieval in comparison with some state-of-the-art approaches. In the following experiments, quantitative evaluations are mainly reported.

7.5.1 Experimental Testbeds and Setup

To conduct comprehensive evaluations, we employ two benchmark datasets for NDK retrieval as the experimental testbeds. One is the widely used Columbia's TRECVID2003 dataset [109], which consists of 600 keyframes with 150 near duplicate image pairs and 300 non-duplicate images extracted from the TRECVID2003 corpus [109]. All the keyframes are with the same size, 352 ×

264. The other is CityU's TRECVID2004 dataset[2] recently collected by Ngo et al. [72]. It contains 7,006 keyframes with 3,388 near-duplicate image pairs, which are selected from the TRECVID2004 video corpus. In the TRECVID2004 dataset, the near-duplicate image pairs involve a total of 1,953 keyframes, representing about 28% of the whole collection. Note that one keyframe may be associated with several near-duplicate pairs.

To make a fair comparison with the state-of-the-art approaches, we adopt the evaluation protocol used in [113]. Specifically, all NDK pairs are adopted as queries for performance evaluation. Each query set $\mathcal{Q}$ contains a single keyframe image; other remaining keyframes are regarded as the gallery set $\mathcal{G}$. For the retrieval task, each algorithm produces a list of relevant results by ranking the keyframes in the gallery set. To evaluate the retrieval performance, the average *cumulative accuracy* metric is adopted as a performance metric [113], in which the accuracy is measured by judging whether the retrieved keyframe is one of the corresponding pairwise duplicates in the ground truth query set. As a yardstick for assessing the performance, we compare the proposed method with the recently proposed OOS matching algorithm [113], one state-of-the-art method for NDK detection and retrieval.

For the experimental setups, the kernel function used in both SVM and S^{3}VM is an RBF kernel with fixed width. Regarding the parameter settings, the penalty parameter C of SVMs is set to 10 (or $\gamma_A = 10^{-1}$) and the graph regularization parameter of S^{3}VM is set to $\gamma_I = 10^{-1}$.

All the experiments in this chapter were carried out on a notebook computer with Intel Core-2 Duo 2.0GHz processor and 2GB RAM. All the proposed methods are implemented in Matlab, for which some routines are written in C code. The code

[2] `http://vireo.cs.cityu.edu.hk/research/NDK/ndk.html`

can be downloaded for verification purpose[3].

7.5.2 Feature Extraction

Feature extraction is a key step for NDK retrieval. In the experiments, both global and local features are considered. The two types of features have their advantages and disadvantages. We believe an appropriate fusion of them will compensate their shortcomings, and therefore improve the overall effectiveness and efficiency.

Global Feature Extraction

The global feature representation techniques have been extensively studied in image processing and CBIR community. A wide variety of global feature extraction techniques were proposed in the past decade. In this chapter, we extract four kinds of effective global features:

- **Grid Color Moment.** We adopt the grid color moment to extract color features from keyframes. Specifically, an image is partitioned into 3×3 grids. For each grid, we extract three kinds of color moments: color mean, color variance and color skewness in each color channel (R, G, and B), respectively. Thus, an 81-dimensional grid color moment vector is adopted for color features.

- **Local Binary Pattern (LBP).** The local binary pattern [73] is defined as a gray-scale invariant texture measure, derived from a general definition of texture in a local neighborhood. In the experiment, a 59-dimensional LBP histogram vector is adopted.

- **Gabor Wavelets Texture.** Gabor wavelets is an effective feature image representation method widely used in [42,

[3] http://www.cse.cuhk.edu.hk/~jkzhu/dup_detect.html

125]. To extract Gabor texture features, each image is first scaled to 64×64 pixels. The Gabor wavelet transform [57] is then applied on the scaled image with 5 levels and 8 orientations, which results in 40 subimages. For each subimage, 3 moments are calculated: mean, variance and skewness. Thus, a 120-dimensional vector is used for Gabor texture features.

- **Edge.** An edge orientation histogram is extracted for each image. We first convert an image into a gray image, and then employ a Canny edge detector [17] to obtain the edge map for computing the edge orientation histogram. The edge orientation histogram is quantized into 36 bins of 10 degrees each. An additional bin is used to count the number of pixels without edge information. Hence, a 37-dimensional vector is used for shape features.

In total, a 297-dimensional vector is used to represent all the global features for each keyframe in the datasets.

Local Feature Extraction

Interest point detection and matching is a fundamental research problem in computer vision. Many effective approaches have been proposed in the literature. One of the most widely used methods is the SIFT [62], which computes a histogram of local oriented gradients around the interest point and stores the bins in a 128-dimensional vector. To improve the SIFT, Ke et al. [53] proposed an extended method by applying Principle Component Analysis [33] on the gradient image, which then yields a 36-dimensional descriptor that is more compact and faster for matching. However, the PCA-SIFT has been empirically shown to be less distinctive than the original SIFT in a comparative study [69], and is also slower than the original SIFT in the feature computation. Instead of using SIFT or PCA-SIFT, we adopt SURF [9], another emerging local feature descriptor to detect and extract local features, which takes advantage of fast feature extraction using integral images for image convolutions. Specifically, a 64-dimensional feature vector is used for representing each keypoint with SURF. Compared to the SIFT, it is more compact and hence reduces the computational cost for keypoint matching.

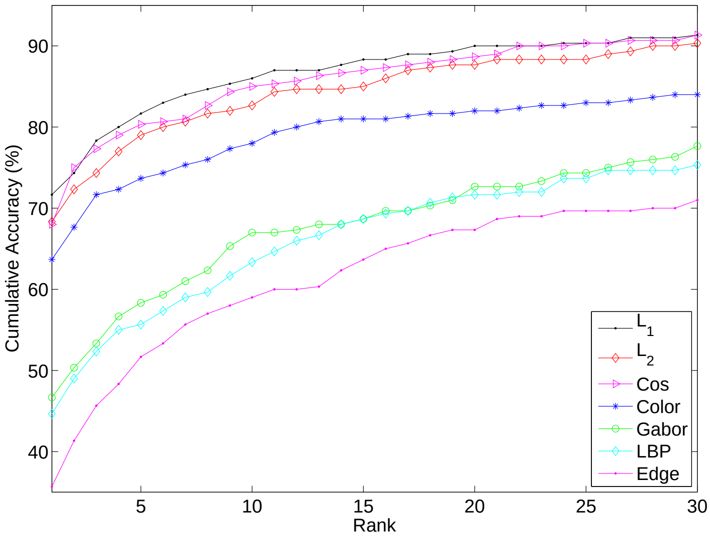
7.5.3 Experiment I: Ranking on Global Features

In this part, the effectiveness of the proposed multi-level ranking scheme is evaluated for filtering out the irrelevant keyframes by ranking on global features. We will first evaluate the retrieval performance of the global features with nearest neighbor ranking, and then evaluate the semi-supervised ranking approach based on S^{3}VM.

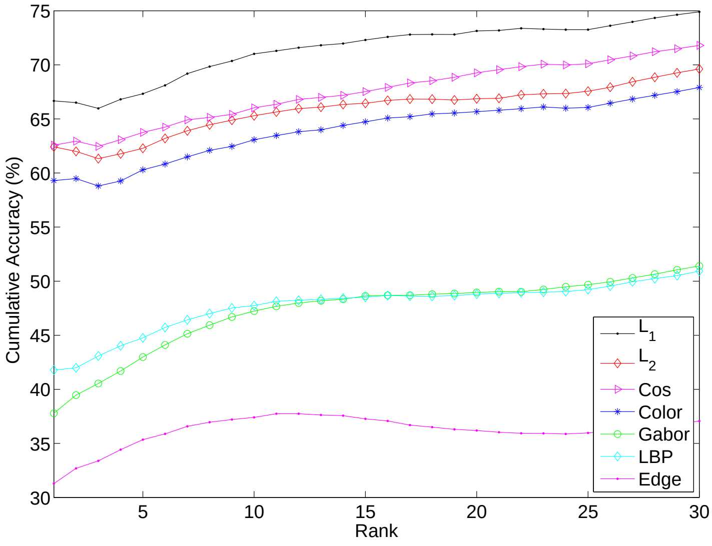
Effectiveness of Global Features

To examine how effective the global features are, we measure the retrieval performance of different distance measures with the global features on both datasets, as shown in Figure 7.5. From the results, we first observe that different distance metrics have different impacts on the retrieval results with the same global features. In particular, the L_1 norm outperforms both the L_2 norm and the cosine metric on both datasets, and the cosine similarity is slightly better than the L_2 norm. As a result, we employ the L_1 norm as the distance measure in all of the remaining experiments.

In addition, we also assess the performance of each component of the global features as well as the combined features. From the results shown in Figure 7.5, we can see that the approaches with the combined features clearly outperform the approaches with individual features. For the individual features,



(a) TRECVID2003 Dataset



(b) TRECVID2004 dataset

Figure 7.5: Cumulative accuracy of similarity measure and features using Nearest Neighbor Ranking on the TRECVID2003 dataset (600 keyframes) and the TRECVID2004 dataset (7006 keyframes).

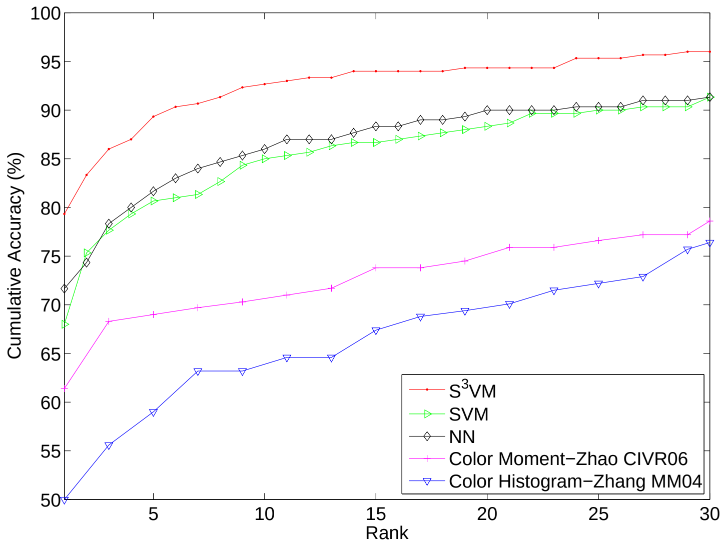
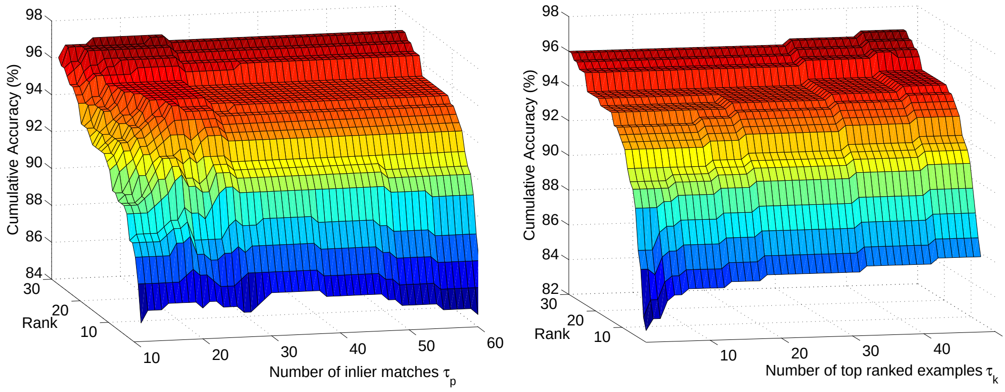


Figure 7.6: Comparison of the proposed semi-supervised ranking method using S^{3}VM algorithm with other appearance based methods on the TRECVID2003 dataset.

we found that the grid color moments method outperforms the other three methods.

Performance of the S^{3}VM Method

Finally, we compare the proposed semi-supervised ranking approach using the S^{3}VM method with other conventional appearance-based methods on global features, such as the approaches with color histogram [109] and color moments [112]. Note that we employ the Nearest Neighbor ranking results to select the most dissimilar examples as the negative samples for training S^{3}VM. Figure 7.6 shows the experimental results on the two datasets. Obviously, S^{3}VM significantly outperforms the color moment and color histogram methods. Specifically, S^{3}VM obtains about 33% improvement over the color moment method on the TRECVID2003 dataset. Compared with the supervised ranking methods including Nearest Neighbor ranking and SVM ranking, S^{3}VM achieves



(a) different τ_p values

(b) different τ_k values

Figure 7.7: Cumulative accuracy of NDK retrieval using NIM method on the TRECVID2003 dataset. (a). There is a wide range available from which to select the threshold value. The image pairs with below 30 inlier matches are viewed as non-duplicate in the experiments. (b) The overall accuracy grows with the number of top-K returns. We choose 50 as a trade-off between the accuracy and computational time.

significantly better results, with around 10% improvement over the two conventional ranking methods.

7.5.4 Experiment II: Re-ranking with NIM on Local Features

Parameter Settings

The last key ranking stage for the MLR scheme is the NIM ranking using the proposed NDK matching technique. To deploy the NIM technique for the NDK retrieval task, we need to determine some parameter settings. In general, the total number of mesh vertices determines the computational complexity and the deformation accuracy of the NIM method. Empirically, we adopt a 14 × 16 mesh for all of the experiments. The regularization coefficient λ_r is set to 5×10^{-5} to allow large deformations. The order ν of the robust estimator is set to 4. The initial support is 100 and the decay rate is 0.5. We find the optimization of each

NIM task requires around 9 iterations to achieve convergence.

Evaluation on the Choices of Two Thresholds

For the proposed NIM approach, there are two threshold parameters that can affect the resulting accuracy and efficiency performance. These are: (1) the minimal number of inlier matches for reporting positive NDKs, denoted by τ_p, and (2) the number of top ranked examples to be matched by NIM, denoted by τ_k.

The first threshold parameter τ_p determines the threshold for predicting positive results. Normally, the smaller the value of τ_p, the higher the recall (the hit rate). At the same time, the precision is likely to drop with decreasing τ_p. Hence, it is important to determine an optimal threshold parameter. Although we do not have a theoretical approach to this, choosing a good τ_p value empirically seems not too difficult. To justify this, we evaluate the performance by varying the τ_p values. Figure 7.7(a) shows the surface of cumulative accuracies with the top 30 returned results on the TRECVID2003 dataset when τ_p varies from 10 to 50 (where τ_k is fixed to 50). From the results, we can see that good results can be obtained when setting the threshold τ_p between 15 and 30.

The second threshold parameter τ_k determines how many examples returned by the S^{3}VM ranking will be engaged for the NIM matching. Hence, it affects both the accuracy and efficiency performance. In general, the larger the value of τ_k is, the more computational cost is incurred. However, τ_k value that is too small is likely to degrade the retrieval performance. Hence, choosing a proper τ_k value is important to balance the tradeoff between accuracy and efficiency performance. To see how τ_k affects the performance, Figure 7.7(b) shows the surface of cumulative accuracies with the top 30 returned results obtained by varying τ_k from 1 to 50 (with τ_p fixed to 30). From the results, we can see that the cumulative accuracy increases when τ_k in-

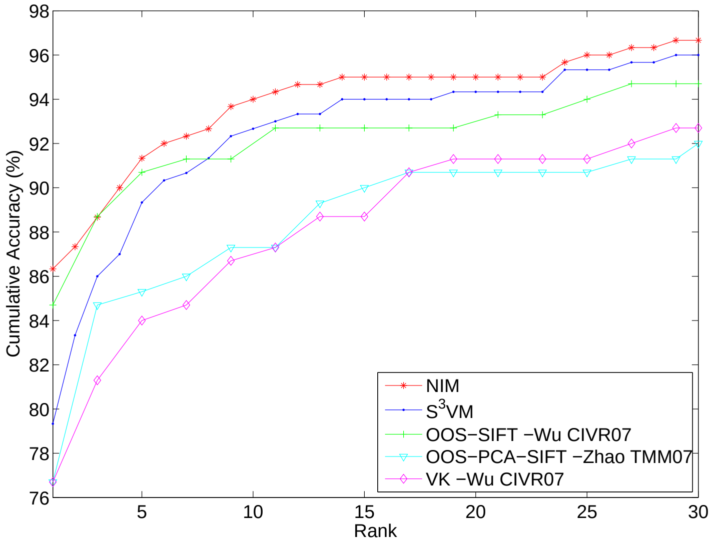


Figure 7.8: Comparison of cumulative accuracy of NDK retrieval results on the TRECVID2003 dataset (600 images).

creases and tends to converge when τ_k approaches 50. Therefore, in the rest of the experiments, we simply fix τ_k to 50 to achieve good efficiency. We will evaluate the efficiency performance in a subsequent part of this chapter.

Comparisons of NDK Retrieval Performance

To examine the performance of the proposed NIM technique for retrieving NDKs, we compare the presented method with several state-of-the-art methods, including the OOS-SIFT method [104], the OOS-PCA-SIFT method [113], and the Visual Keywords (VK) methods [113]. Figure 7.8 and Figure 7.9 show the experimental results of the cumulative accuracy of the top 30 returned keyframes on the two datasets respectively.

For the TRECVID2003 dataset, it is relatively small and widely used as a benchmark testbed for NDK retrieval in literature. From the experimental results, we can draw several observations. First of all, the proposed S^{3}VM method with global

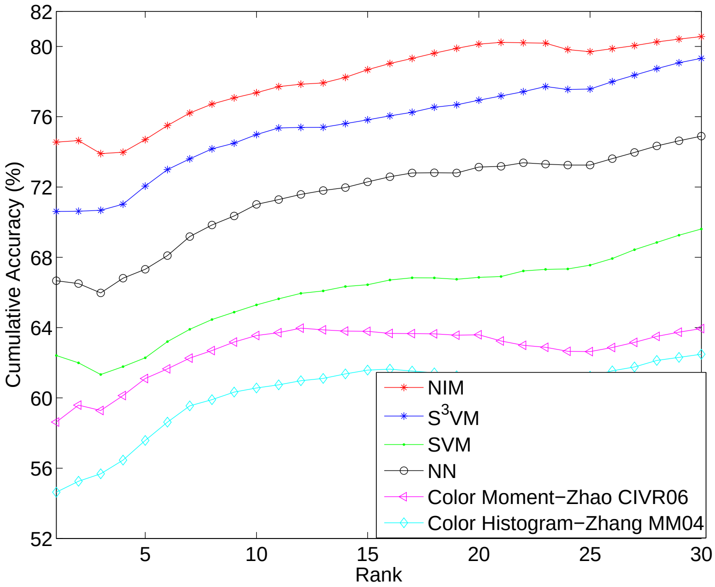
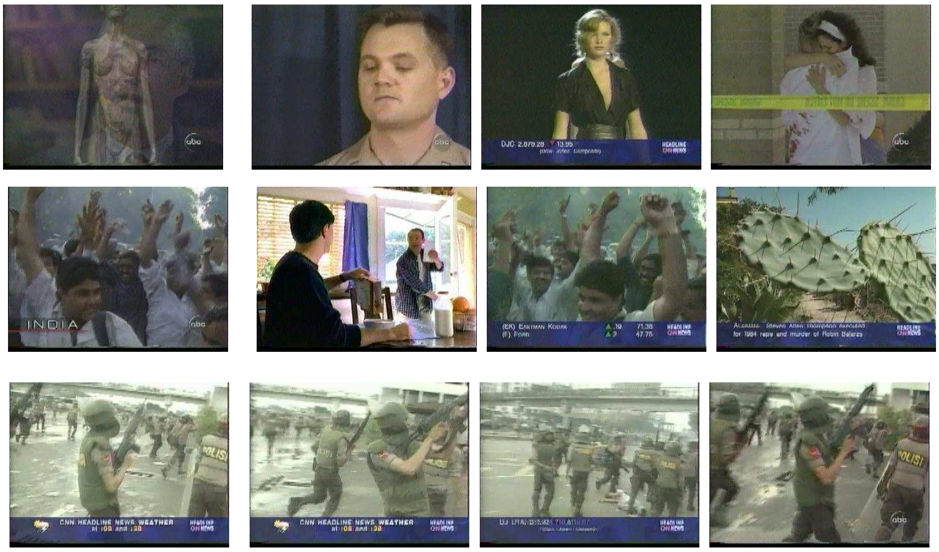


Figure 7.9: Comparison of cumulative accuracy of NDK retrieval results on the TRECVID2004 dataset (7006 images).

features outperforms the OOS-PCA-SIFT method [113] and the VK method [104], which use local features. This again validates the effectiveness of the proposed semi-supervised ranking technique with S^{3}VM. Second, the proposed NIM algorithm with local features is significantly better than the S^{3}VM method. In particular, NIM achieves more than 8% improvement on the rank-one accuracy over S^{3}VM. Finally, among all compared methods, the proposed NIM method achieves the best performance, outperforming the state-of-the-art OOS-SIFT method [104].

Turning next to the TRECVID2004 dataset, due to its large size, we have a difficulty in comparing the presented method with other existing methods, such as the OOS-SIFT and OOS-PCA-SIFT methods, which are computationally very intensive. Therefore, we only compare the proposed method with some conventional approaches. Figure 7.9 shows the experimental results on the TRECVID2004 dataset. Similar to the previous dataset, NIM achieves the best performance among all the compared methods on this dataset. For other compared methods, S^{3}VM performs significantly better than both supervised SVM and NN methods.

Finally, to give more insights of the proposed technique, we are interested in checking when our method may fail. To this purpose, we show some failure cases in Fig. 7.10, where all of the top one retrieved examples by the proposed method are not the true duplicates. We here briefly analyze these cases and attempt to find some possible reasons. For the first case as shown in the first row, all of the top 3 retrieved examples are not the true duplicates. The main reason is because the query image is too blur and too smooth to extract the discriminative feature points by the local feature descriptor. In fact, this is a common challenge faced by most of existing keypoints based methods. For the second case, the first and the third examples are not the true duplicates, while the true duplicate is ranked at the second position instead of the first position. This difficulty in finding a strong explicit mapping between the duplicate pair is possibly due to poor image quality as well as too large changes in lighting and camera capture conditions. For the last case, among the top 3 retrieved examples, only the last example is the true duplicate. In fact, the first two examples are visually very similar to the query image, but they are not labeled as the true duplicates according to the ground truth. The result shows that when there are too many similar images but essentially not real duplicates, the performance of the proposed method may be degraded due to the engagement of the filtering stage with the global features.



(a) Query Image (b) Top three retrieval results

Figure 7.10: Failure examples in showing when the presented method may fail.

7.5.5 Evaluation of Computational Cost

Finally, we empirically examine the efficiency performance of the proposed NIM and S^{3}VM methods. Both the global appearance features and local features are extracted offline. Table 7.1 and Table 7.2 summarize the overall computational time for comparing all pairs of keyframes on both datasets. From the results, we can see that NIM is more efficient than the OOS-SIFT method [104] and less efficient than the VK method which simply computes the similarity of visual words. Note that VK method often requires much preprocessing time cost for extracting the visual keywords offline. In addition, we clearly see that the methods using global features are significantly more efficient than the ones using local feature matching. This again validates the importance and effectiveness of the proposed multi-level ranking scheme for improving the efficiency. Finally, we also plot the computational cost and retrieval accuracy with respect to the number of top ranked examples (τ_k) to be compared by NIM in Figure 7.11. The results show that the larger the value of τ_k, the higher the computational cost and the better the matching accuracy. In particular, we found that the cumulative accuracy tends to converge to the best result when τ_k approaches to 50. In real-world applications, one can choose an appropriate τ_k to balance the tradeoff between accuracy and efficiency. For example, when τ_k equals to 10, each query for NIM takes about 1 second and achieves rather high cumulative accuracy, about 93%.

Table 7.1: Comparison of overall time cost of 300 queries on the TRECVID2003 dataset.

NIM	S^{3}VM	NN	OOS [104]	VK [104]
15.8min	3sec	1sec	6.5hour	1.5min

Table 7.2: Comparison of overall time cost of 1,953 queries on the TRECVID2004 dataset.

NIM	S^{3}VM	NN	OOS [104]	VK [104]
103.5min	8.1min	30sec	N/A	N/A

7.6 Summary

This chapter presented a novel nonrigid image matching method for Near-Duplicate Keyframe (NDK) retrieval. In contrast to traditional approaches with either projective geometry or bipartite graph matching, the proposed nonrigid image matching (NIM) algorithm recovers the explicit nonrigid mapping between two NDKs and effectively finds out the correct correspondences by a robust coarse-to-fine optimization scheme. Moreover, the presented method not only can detect the NDK pairs accurately, but also can recover the local deformations between them simultaneously. To further reduce the overall computational cost,

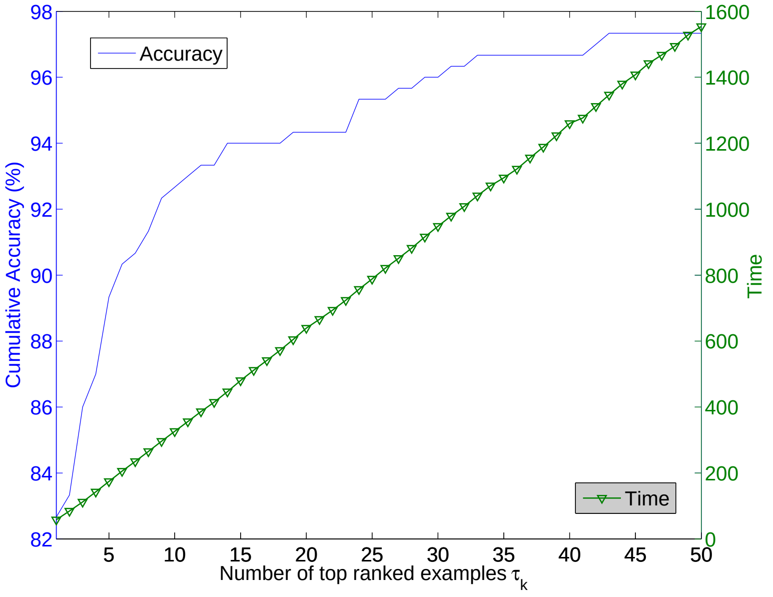


Figure 7.11: Computational efficiency and retrieval performance on the TRECVID2003 dataset. The *left* vertical axis shows mean *cumulative accuracy* of the top 30 returned results, and the *right* vertical axis represents the overall time cost for all 300 queries.

we proposed an effective multi-level ranking scheme together with a semi-supervised ranking technique using semi-supervised SVM (S^{3}VM) to improve the ranking performance with the unlabeled data. Extensive evaluations have been conducted on two testbeds extracted from the TRECVID corpora. The promising experimental results showed that the method is clearly more effective than conventional approaches, especially for dealing with cases involving viewpoint changes and local deformations, which are very common in practice.

□ **End of chapter.**

Chapter 8

Conclusion and Future Work

In this chapter, we briefly summarize this thesis research and discuss some further work.

8.1 Conclusion

In this thesis, we have proposed a few deformation models and deformable surface recovery approaches, and applied them to effectively solve a variety of tasks in the real-world applications. Specifically, we proposed three different approaches to 2D nonrigid shape recovery, one method for 3D deformable surface tracking. These methods all belong to the passive method dealt with single still image or monocular video. The major contributions are concluded in the following.

First of all, a novel progressive finite Newton optimization scheme is proposed to solve the nonrigid surface detection problem, which is formulated as a closed-form solution for a given set of local feature-correspondences. Moreover, a modified RANSAC scheme is employed to select the initial active set. It takes advantage of the concise formulation and top-ranked correspondences, and can handle high-dimensional variable spaces with noisy observations. Furthermore, the presented method is very fast and robust, and provides a fully-automatic solution for real-time nonrigid object detection, Augmented Reality and medical image registration.

The feature-based methods for nonrigid shape recovery may suffer from some jittering issue due to the lack of reliable matches. To tackle this problem, a fusion approach is proposed, which takes advantage of both the appearance information and the local feature correspondences. To allow the large surface deformation, a deformable mesh model is introduced into the conventional Lucas-Kanade framework. This leads to the presented deformable Lucas-Kanade algorithm that can be efficiently optimized by the inverse compositional method. Comparing to the feature-based method, the experimental evaluation demonstrates that the jitter is greatly reduced in the fusion approach. Additionally, the partial occlusion problem is properly handled through starting from a good initialization and imposing the deformable model.

In contrast to the 2D nonrigid shape recovery, 3D deformable surface recovery can estimate the depth information and reconstruct the object surface. In this thesis, we proposed an effective solution for 3D deformable surface tracking, which formulates the problem into an unconstrained quadratic optimization problem with a closed-form solution. Also, the robust progressive finite Newton optimization scheme is applied to handle the noisy observations. Promising 3D deformable surface tracking results show that the proposed solution is robust to noises and large deformations.

Without resorting to an explicit deformable mesh model, we formulated the nonrigid surface detection as a generic regression problem. A novel velocity coherence constraint is imposed to regularize the surface deformation, which leads to the proposed velocity coherence regression approach. Similarly, a progressive optimization scheme is employed to reject the outliers. This approach has been tested in several applications, such as real-time nonrigid surface tracking and medical image registration.

In addition to the methodology studies and evaluations in computer vision, we also investigated the nonrigid shape recovery techniques in some real-world applications in multimedia information retrieval domain. By taking advantage of a very efficient and automatic solution, we applied the nonrigid surface detection method to retrieving the near-duplicate keyframes from the real-world video corpora. In contrast to conventional approaches with either projective geometry or bipartite graph matching, the proposed method recovers the the local deformations between two near-duplicate keyframes and effectively finds out the correct local feature correspondences from the noisy observations. To make it applicable to large scale data, an effective multi-level ranking scheme together with a semi-supervised ranking technique is presented to improve the ranking performance with the unlabeled data. Extensive evaluations on two testbeds extracted from TRECVID video corpora demonstrated that the method is importantly more effective than the conventional approaches, especially in the case of local deformations and viewpoint variations.

8.2 Future Work

Although a substantial number of promising achievements on deformable surface recovery and its applications have been presented in this thesis, there are still numerous open issues that require to be continuously explored in future studies. We briefly describe them in the following.

First, the computational efficiency is always an important issue in a computer vision application, especially for the real-time applications. Although we have developed the efficient algorithms with closed-form solutions, there are still some steps which can be further accelerated. For example, we can consider an efficient GPU-based feature matching algorithm. For large-scale applications, it will be very expensive to directly solve the linear system. By tacking advantage of the spatial information, a multi-scale algorithm can be used to improve the performance. More specifically, an efficient octree structure can be employed to build a simplified multi-resolution mesh model for the mesh model-based methods. Furthermore, a resolution-aware approach can be considered to adaptively select the mesh model.

The second problem is related to finding the reliable correspondences under the scale changes, sever deformations and illumination variations. With the rapid progress in the object recognition, these difficulties can be tackled through introducing some effective feature descriptors. Also, the feature extraction algorithm can be further accelerated by the GPU power. For the 3D deformable surface recovery, we will consider incorporating the appearance information into the energy function in order to exploit more information, which can be built on top of the deformable Lucas-Kanade algorithm described in Chapter 4.

Third, for the feature-based methods, in some situations some jitter may occur due to a lack of texture information. To deal with this problem, the global bundle-adjustment used in 3D reconstruction can be fitted into our proposed gradient-based optimization framework.

Forth, self-occlusion problem has not yet been studied in this thesis. To address the problem in 3D environment, we may consider employing the visible surface detection algorithm. Moreover, small errors did occur in the boundary region. We will consider some silhouette-based methods and incorporate the contour information to solve the ambiguity issue.

There are also a variety of obvious extensions to the existing work we have illustrated in this thesis. First, we only employ the first order approximation method to compute the warp update in the deformable Lucas-Kanade algorithm. Therefore, a second order method can be considered as an immediate extension. Second, we may explore the new deformable model and regularization method via extending the techniques developed in this thesis. Finally, we can apply the presented algorithms to new applications. For example, the 3D deformable surface recovery results can be employed in motion capture and computer animation. Also, the nonrigid surface detection can be applied to many other multimedia applications like video content analysis rather than the near-duplicate image retrieval.

□ **End of chapter.**

Appendix A

Optimization With Lighting

Performing the first order Taylor expansion on Eqn. 4.9 gives:

$$\sum_{\mathbf{x}} \left[(a+\Delta a) \left(T(W(\mathbf{x};\mathbf{s}_0)) + \nabla T \frac{\partial W}{\partial \mathbf{s}} \Delta \mathbf{s} \right) + (o+\Delta o)\cdot \mathbf{1} \right.$$
$$\left. - I(W(\mathbf{x};\mathbf{s})) \right]^2 + \lambda_r (\mathbf{s}+\Delta\mathbf{s})^\top \mathcal{K} (\mathbf{s}+\Delta\mathbf{s})$$

Let D denote $\nabla T \frac{\partial W}{\partial \mathbf{s}}$, the gradient of the above equation can be derived as below:

$$\begin{aligned}
\frac{\partial E_A}{\partial \Delta \mathbf{s}} &= aD^\top \left[(a+\Delta a)(T + D\Delta\mathbf{s}) + (o+\Delta o)\cdot\mathbf{1} - I \right] \\
&\quad + \lambda_r \mathcal{K}(\mathbf{s}+\Delta\mathbf{s}) \\
\frac{\partial E_A}{\partial \Delta a} &= T^\top \left[(a+\Delta a)(T + D\Delta\mathbf{s}) + (o+\Delta o)\cdot\mathbf{1} - I \right] \\
\frac{\partial E_A}{\partial \Delta o} &= \mathbf{1}^\top \left[(a+\Delta a)(T + D\Delta\mathbf{s}) + (o+\Delta o)\cdot\mathbf{1} - I \right]
\end{aligned}$$

The texture difference ΔI is compute by: $\Delta I = I - aT - o$. Also, we define $\mathcal{T} = \begin{bmatrix} T & \mathbf{1} \end{bmatrix}$, $\Delta\mathbf{g} = \begin{bmatrix} \Delta a & \Delta o \end{bmatrix}$ and $H_4 = D^\top D + \frac{\lambda_r}{a^2}\mathcal{K}$. Thus, we can obtain the following equation by neglecting second-order terms:

$$\begin{bmatrix} a^2 H_4 & aD^\top \mathcal{T} \\ a\mathcal{T}^\top D & \mathcal{T}^\top \mathcal{T} \end{bmatrix} \begin{bmatrix} \Delta\mathbf{s} \\ \Delta\mathbf{g} \end{bmatrix} = \begin{bmatrix} aD^\top \Delta I - \lambda_r \mathcal{K}\mathbf{s} \\ \mathcal{T}^\top \Delta I \end{bmatrix}$$

As in [39, 4], we multiply a full-rank matrix $L \in R^{4N \times 4N}$ to the left side of the above equation:

$$L \begin{bmatrix} a^2 H_4 & a D^\top \mathcal{T} \\ a \mathcal{T}^\top D & \mathcal{T}^\top \mathcal{T} \end{bmatrix} \begin{bmatrix} \Delta \mathbf{s} \\ \Delta \mathbf{g} \end{bmatrix} = L \begin{bmatrix} a D^\top \Delta I - \lambda_r \mathcal{K} \mathbf{s} \\ \mathcal{T}^\top \Delta I \end{bmatrix} \quad \text{(A.1)}$$

where L is defined as below:

$$L = \begin{bmatrix} \text{diag}(\mathbf{1}_{2N}) & 0 \\ -\frac{1}{a} \mathcal{T}^\top D H_4^{-1} & \text{diag}(\mathbf{1}_{2N}) \end{bmatrix}$$

Simplifying Eqn. A.1, we can obtain $\Delta \mathbf{g}$ by solving the following equation:

$$\left(Q - G^\top H_4^{-1} G\right) \Delta \mathbf{g} = \mathcal{T}^\top \Delta I - G^\top H_4^{-1} \left(D^\top \Delta I - \frac{\lambda_r}{a} \mathcal{K} \mathbf{s} \right)$$

where $G = D^\top \mathcal{T}$ and $Q = \mathcal{T}^\top \mathcal{T}$. Also, $\Delta \mathbf{s}$ can be computed by:

$$\Delta \mathbf{s} = \frac{1}{a} \left[H_4^{-1} \left(D^\top \Delta I - \frac{\lambda_r}{a} \mathcal{K} \mathbf{s} \right) - H_4^{-1} G \Delta \mathbf{g} \right]$$

Similarly, we compute the warp update through Eqn. 4.8. Note that we can pre-compute G and Q in order to reduce the computational cost. As the regularization coefficient λ_r can be chosen in a very wide range without significantly affecting the results [75, 76], we treat the H_4 as constant (set $a = 1$) and ignore the changes of a during the optimization.

□ **End of chapter.**

Appendix B

Solution of Velocity Coherence Regression

For velocity coherence regression,

$$E = \frac{1}{2\sigma^2}\sum_{i=1}^{n}\|\mathbf{y}_i - f(\mathbf{x}_i)\|^2 + \frac{\lambda}{2}\int_{R^d}\frac{|\tilde{v}(\mathbf{s})|^2}{\tilde{G}(\mathbf{s})}d\mathbf{s}$$

As in [30], the Fourier transform of the continuous function is:

$$v(\mathbf{x}_i) = \int_{R^d}\tilde{v}(\mathbf{s})e^{2\pi j<\mathbf{x}_i,\mathbf{s}>}d\mathbf{s}$$

$$E(\tilde{v}) = \frac{1}{2\sigma^2}\sum_{i=1}^{n}\|\mathbf{y}_i - \mathbf{x}_i - v(\mathbf{x}_i)\|^2 + \frac{\lambda}{2}\int_{R^d}\frac{|\tilde{v}(\mathbf{s})|^2}{\tilde{G}(\mathbf{s})}d\mathbf{s}$$

Moreover, the derivatives of the energy function $E(\tilde{v})$ with respect to $\tilde{v}$ can be derived as below:

$$\frac{\delta E(\tilde{v})}{\delta\tilde{v}} = \frac{1}{\sigma^2}\sum_{i=1}^{n}(\mathbf{y}_i - f(\mathbf{x}_i))e^{2\pi j<\mathbf{x}_i,\mathbf{s}>} + \lambda\frac{\tilde{v}(-\mathbf{t})}{\tilde{G}(\mathbf{t})}$$

Since the derivatives of $E(\tilde{v})$ with respect to $\tilde{v}$ vanish for optimality, and f is real, changing $\mathbf{t}$ in $-\mathbf{t}$:

$$\frac{1}{\sigma^2}\sum_{i=1}^{n}(\mathbf{y}_i - f(\mathbf{x}_i))e^{2\pi j<\mathbf{x}_i,\mathbf{s}>} + \lambda\frac{\tilde{v}(\mathbf{t})}{\tilde{G}(-\mathbf{t})} = 0$$

Therefore,

$$\tilde{v}(\mathbf{t}) = \tilde{G}(-\mathbf{t}) \sum_{i=1}^{n} \frac{(\mathbf{y}_i - f(\mathbf{x}_i))}{\lambda\sigma^2} e^{2\pi j < \mathbf{x}_i, \mathbf{s} >}$$

We define the coefficients $\boldsymbol{\alpha}_i = \frac{(\mathbf{y}_i - f(\mathbf{x}_i))}{\lambda\sigma^2}$, and assume that $\tilde{G}$ is symmetric. Take the Fourier transform of $\tilde{v}$:

$$v(\mathbf{x}) = G(\mathbf{x}) * \sum_{i=1}^{n} \boldsymbol{\alpha}_i \delta(\mathbf{x} - \mathbf{x}_i) = \sum_{i=1}^{n} \boldsymbol{\alpha}_i G(\mathbf{x} - \mathbf{x}_i)$$

□ **End of chapter.**

Appendix C

List of Publications

1. **Jianke Zhu**, Steven C.H. Hoi, Michael R. Lyu and Shuicheng Yan, "Near-Duplicate Keyframe Retrieval by Nonrigid Image Matching," To appear in *ACM Multimedia (MM'2008)*, oral presentation.

2. **Jianke Zhu**, Steven C.H. Hoi, Zenglin Xu and Michael R. Lyu, "An Effective Approach to 3D Deformable Surface Tracking," In *The 10th European Conference on Computer Vision (ECCV'2008).*

3. **Jianke Zhu**, Michael R. Lyu and Thomas S. Huang, "A Fast 2D Shape Recovery Approach by Fusing Features and Appearance," To appear in *IEEE Trans. on Pattern Analysis and Machine Intelligences*, 2008.

4. **Jianke Zhu**, Steven C.H. Hoi and Michael R. Lyu, "Robust Regularized Kernel Regression," To appear in *IEEE Trans. on Systems, Man, and Cybernetics - Part B: Cybernetics*, 2008.

5. **Jianke Zhu**, Steven C.H. Hoi and Michael R. Lyu, "Face Annotation by Transductive Kernel Fisher Discriminant," *IEEE Trans. on Multimedia*, vol. 10, Jan. 2008, pp. 86-96.

6. Steven C.H. Hoi, Rong Jin, **Jianke Zhu** and Michael R.

Lyu, "Semi-Supervised SVM Batch Mode Active Learning for Image Retrieval," In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'2008).*

7. **Jianke Zhu** and Michael R. Lyu, "Progressive Finite Newton Approach To Real-time Nonrigid Surface Detection," In *Proc. Conf. on Computer Vision and Pattern Recognition (CVPR'2007)*, June 19-21, 2007. (Oral presentation, acceptance rate: 5%)

8. **Jianke Zhu**, Steven C.H. Hoi and Michael R. Lyu, "A Multi-Scale Tikhonov Regularization Scheme for Implicit Surface Modelling," In *Proc. Conf. on Computer Vision and Pattern Recognition (CVPR'2007)*, June 19-21, 2007. (Acceptance rate: 28%)

9. Zenglin Xu, Rong Jin, **Jianke Zhu**, Irwin King and Michael R. Lyu, "Efficient Convex Relaxation for Transductive Support Vector Machine," In *Advances in Neural Information Processing Systems (NIPS'2007).* (Acceptance rate: 22%)

10. Zenglin Xu, Kaizhu Huang, **Jianke Zhu**, Irwin King and Michael R. Lyu, "Kernel Maximum a Posteriori Classification with Error Bound Analysis," In *International Conference on Neural Information Processing (ICONIP'2007)*, LNCS 4984, 2008, pp.841-850.

11. Zenglin Xu, **Jianke Zhu**, Michael R. Lyu and Irwin King, "Semi-supervised Spectral Kernel Learning," In *International Joint Conference on Neural Networks (IJCNN'07).*

12. Hongbo Deng, **Jianke Zhu**, Michael R. Lyu, Irwin King, "Two-Stage Multi-Class AdaBoost for Facial Expression Recognition," In *International Joint Conference on Neural Networks (IJCNN'07).*

13. **Jianke Zhu**, Steven C. Hoi, and Michael R. Lyu. "Real-time non-rigid shape recovery via active appearance models for augmented reality." In *Proc. European Conf. Computer Vision*, pages 186–197, 2006. (Acceptance rate: 21%)

14. Steven C.H. Hoi, Rong Jin, **Jianke Zhu** and Michael R. Lyu, "Batch Mode Active Learning and Its Application to Medical Image Classification," In *The 23th International Conference on Machine Learning (ICML'2006)*, Pittsburgh, June 25-29, 2006. (Acceptance rate: 20%)

15. Chon Fong Wong, **Jianke Zhu**, Mang I Vai and Peng Un Mak, "Face Image Retrieval with Relevance Feedback Using Lifting Wavelets Features," In *Applied and Numerical Harmonic Analysis Series*, Tao Qian (ed.), Springer, 2006. .

16. **Jianke Zhu**, Steven C.H. Hoi, Edward Yau and Michael R. Lyu, "Automatic 3D Face Modeling Using 2D Active Appearance Models," In *Proceedings of the 13th Pacific Conference on Computer Graphics and Applications (PG'2005)*, Macau, China, October 12-14 2005.

17. Steven C.H. Hoi, **Jianke Zhu** and Michael R. Lyu, "CUHK at ImageCLEF 2005: Cross-Language and Cross-Media Image Retrieval," In *Proceedings of Cross Language Evaluation Forum (CLEF'2005)*, LNCS 4022, Vienna, Austria, 2006.

18. Chon Fong Wong, **Jianke Zhu**, Mang I Vai and Peng Un Mak, "Face Image Retrieval in Video Sequences Using Lifting Wavelets Transform Feature Extraction," In *Proceedings International Symposium on Computer Electronics*, Macau, 2005.

19. **Jianke Zhu**, Mang I Vai and Peng Un Mak, “Gabor Wavelets Transform and Extended Nearest Feature Space Classifier for Face Recognition,” In *Proceedings Third International Conference on Image and Graphics (ICIG’2004)*, Hong Kong, Dec. 28-30, 2004, pp.372-379.

20. **Jianke Zhu**, Mang I Vai and Peng Un Mak, “A New Enhanced Nearest Feature Space (ENFS) Classifier for Gabor Wavelets Features Based Face Recognition,” In *Proceedings First International Conference on Biometrics Authentication (ICBA’2004)*, LNCS 3072, Hong Kong, July 15-17, 2004, pp.124-131.

□ **End of chapter.**

Bibliography

[1] J. Ahlberg. Using the active appearance algorithm for face and facial feature tracking. In *Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems, 2001. Proceedings. IEEE ICCV Workshop on*, pages 68–72, 2001.

[2] S. Baker and I. Matthews. Lucas-kanade 20 years on: A unifying framework. *Int'l J. Computer Vision*, 56(3):221–255, March 2004.

[3] S. Baker, R. Patil, K. M. Cheung, and I. Matthews. Lucas-kanade 20 years on: Part 5. Technical Report CMU-RI-TR-04-64, RI, CMU, November 2004.

[4] A. Bartoli. Groupwise geometric and photometric direct image registration. In *Proc. British Machine Vision Conference*, Edinburgh, Sep. 2006.

[5] A. Bartoli. Groupwise geometric and photometric direct image registration. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 2008.

[6] A. Bartoli and A. Zisserman. Direct estimation of non-rigid registration. In *Proc. British Machine Vision Conference*, Kingston, Sep. 2004.

[7] R. Basri, D. Jacobs, and I. Kemelmacher. Photometric stereo with general, unknown lighting. *Int. J. Comput. Vision*, 72(3):239–257, 2007.

[8] R. Basri and D. W. Jacobs. Lambertian reflectance and linear subspaces. *IEEE Trans. Pattern Anal. Mach. Intell.*, 25(2):218–233, 2003.

[9] H. Bay, T. Tuytelaars, and L. J. V. Gool. Surf: Speeded up robust features. In *Proc. European Conf. Computer Vision*, pages 404–417, 2006.

[10] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 24(4):509–522, 2002.

[11] M. Berger and G. Danuser. Deformable multi template matching with application to portal images. In *Proc. Conf. Computer Vision and Pattern Recognition*, page 374, 1997.

[12] V. Blanz, K. Scherbaum, T. Vetter, and H.-P. Seidel. Exchanging faces in images. *Comput. Graph. Forum*, 23(3):669–676, 2004.

[13] V. Blanz and T. Vetter. A morphable model for the synthesis of 3d faces. In *SIGGRAPH '99: Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 187–194, 1999.

[14] V. Blanz and T. Vetter. Face recognition based on fitting a 3d morphable model. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 25(9), 2003.

[15] S. Boyd and L. Vandenberghe. *Convex Optimization.* Cambridge University Press, 2004.

[16] M. Z. Brown, D. Burschka, and G. D. Hager. Advances in computational stereo. *IEEE Trans. Pattern Anal. Mach. Intell.*, 25(8):993–1008, 2003.

[17] J. Canny. A computational approach to edge detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 8(6):679–698, 1986.

[18] J. C. Carr, R. K. Beatson, J. B. Cherrie, T. J. Mitchell, W. R. Fright, B. C. McCallum, and T. R. Evans. Reconstruction and representation of 3d objects with radial basis functions. In *SIGGRAPH '01*, pages 67–76, 2001.

[19] C.Bregler, A.Hertzmann, and H.Biermann. Recovering non-rigid 3d shape from image streams. In *IEEE Proc. Conf. Computer Vision and Pattern Recognition*, volume 2, pages 690–696, 2000.

[20] H. Chui and A. Rangarajan. A new point matching algorithm for non-rigid registration. *Comput. Vis. and Image Underst.*, 89(2-3):114–141, 2003.

[21] O. Chum and J. Matas. Matching with prosac- progressive sample consensus. In *Proc. Conf. Computer Vision and Pattern Recognition*, volume 1, pages 220–226, 2005.

[22] L. Cohen and I. Cohen. Deformable models for 3d medical images using finite elements and balloons. In *Proc. Conf. Computer Vision and Pattern Recognition*, pages 592–598, 1992.

[23] T. Cootes, G. Edwards, and C. Taylo. Active appearance models. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 23(6), June 2001.

[24] T. Cootes and P. Kittipanya-ngam. Comparing variations on the active appearance model algorithm. In *Proc. British Machine Vision Conference*, volume 2, pages 837–846, 2002.

[25] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham. Active shape models—their training and application. *Comput. Vis. Image Underst.*, 61(1):38–59, 1995.

[26] T. Cour, P. Srinivasan, and J. Shi. Balanced graph matching. In *Advances in Neural Information Processing Systems 19*. MIT Press, Cambridge, MA, 2007.

[27] D. Cristinacce and T. Cootes. Feature detection and tracking with constrained local models. In 17^{th} *British Machine Vision Conference, Edinburgh, UK*, pages 929–938, 2006.

[28] D. DeCarlo and D. N. Metaxas. Optical flow constraints on deformable models with applications to face tracking. *Int'l J. Computer Vision*, 38(2):99–127, 2000.

[29] R. Donner, M. Reiter, G. Langs, P. Peloschek, and H. Bischof. Fast active appearance model search using canonical correlation analysis. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28(10):1690–1694, 2006.

[30] T. Evgeniou, M. Pontil, and T. Poggio. Regularization networks and support vector machines. *Advances in Computational Mathematics*, 13:1–50, 2000.

[31] M. A. Fischler and R. C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *CACM*, 24(6):381–395, 1981.

[32] P. Fua and Y. Leclerc. Object-centered surface reconstruction: Combining multi-image stereo and shading. *Int'l J. Computer Vision*, 16(1):35–56, Sep. 1995.

[33] K. Fukunaga. *Introduction to statistical pattern recognition.* Academic Press Professional, Inc., 1990.

[34] V. Gay-Bellile, A. Bartoli, and P. Sayd. Feature-driven direct non-rigid image registration. In *Proc. British Machine Vision Conference*, 2007.

[35] S. Granger and X. Pennec. Multi-scale em-icp: A fast and robust approach for surface registration. In *Proc. European Conf. Computer Vision*, pages 418–432, 2002.

[36] R. Gross, I. Matthews, and S. Baker. Active appearance models with occlusion. *Image and Vision Computing*, 24(6):593–604, 2006.

[37] N. A. Gumerov, A. Zandifar, R. Duraiswami, and L. S. Davis. Structure of applicable surfaces from single views. In *Proc. European Conf. Computer Vision*, pages 482–496, 2004.

[38] C. Haili and R. Anand. A feature registration framework using mixture models. *MMBIA*, pages 190–197, 2000.

[39] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000.

[40] C.-H. Hoi, W. Wang, and M. R. Lyu. A novel scheme for video similarity detection. In *CIVR*, pages 373–382, 2003.

[41] S. C. Hoi and M. R. Lyu. A multi-modal and multi-level ranking framework for content-based video retrieval. *To appear in IEEE Transactions on Multimedia*, 2008.

[42] S. C. Hoi, J. Zhu, and M. R. Lyu. Cuhk at imageclef 2005: Cross-language and cross-media image retrieval. In *Proc. Cross Language Evaluation Forum campaign, LNCS 4022*, pages 602–611, 2006.

[43] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle. Surface reconstruction from unorganized

points. In *SIGGRAPH '92: Proceedings of the 19th annual conference on Computer graphics and interactive techniques*, pages 71–78, 1992.

[44] B. Horn and B. Schunck. Determining optical flow. *Artificial Intelligence*, 17(1-3):185 – 203, 1981.

[45] S. Ilic and P. Fua. Using dirichlet free form deformation to fit deformable models to noisy 3-d data. In *Proc. European Conf. Computer Vision*, May 2002.

[46] S. Ilic and P. Fua. Implicit meshes for modeling and reconstruction. In *Proc. Conf. Computer Vision and Pattern Recognition*, June 2003.

[47] S. Ilic and P. Fua. Implicit meshes for surface reconstruction. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 28(2):328–333, February 2006.

[48] S. Ilic, M. Salzmann, and P. Fua. Implicit meshes for effective silhouette handling. *Int'l J. Computer Vision*, 72(2):159–178, 2007.

[49] B. Jian and B. C. Vemuri. A robust algorithm for point set registration using mixture of gaussians. In *Proc. Int'l Conf. Computer Vision*, pages 1246–1251, 2005.

[50] F. Kahl. Multiple view geometry and the l_∞-norm. In *Proc. Int'l Conf. Computer Vision*, pages 1002–1009, 2005.

[51] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *Int'l J. Computer Vision*, 1(4):321–331, Jan. 1988.

[52] Q. Ke and T. Kanade. Quasiconvex optimization for robust geometric reconstruction. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29(10):1834–1847, 2007.

[53] Y. Ke, R. Sukthankar, and L. Huston. Efficient near-duplicate detction and sub-image retrieval system. In *ACM MULTIMEDIA'04*, pages 869–876. ACM, 2004.

[54] O. C. Keerthi, S. and D. Decoste. Building support vector machines with reduced classifier complexity. *Journal of Machine Learning Research*, 8:1–22, August 2006.

[55] S. S. Keerthi and D. DeCoste. A modified finite newton method for fast solution of large scale linear svms. *Journal of Machine Learning Research*, 6:341–361, 2005.

[56] S. C. Koterba, S. Baker, I. Matthews, C. Hu, J. Xiao, J. Cohn, and T. Kanade. Multi-view aam fitting and camera calibration. In *Proc. International Conference on Computer Vision*, volume 1, pages 511 – 518, October 2005.

[57] M. Lades, J. C. Vorbruggen, J. Buhmann, J. Lange, C. von der Malsburg, R. P. Wurtz, and W. Konen. Distortion invariant object recognition in the dynamic link architecture. *IEEE Trans. Computers*, 42(5):300–311, 1993.

[58] V. Lepetit and P. Fua. Keypoint recognition using randomized trees. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 28(9):1465–1479, 2006.

[59] W.-C. Lin and Y. Liu. Tracking dynamic near-regular texture under occlusion and rapid movements. In *Proc. European Conf. Computer Vision*, pages 44–55, 2006.

[60] H. Ling and D. W. Jacobs. Deformation invariant image matching. In *Proc. Int'l Conf. Computer Vision*, pages 1466–1473, 2005.

[61] X. Liu. Generic face alignment using boosted appearance model. In *Proc. Conf. Computer Vision and Pattern Recognition*, 2007.

[62] D. G. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *Int'l J. Computer Vision*, 60(2):91–110, 2004.

[63] B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *IJCAI'81*, pages 674–679, 1981.

[64] O. L. Mangasarian. A finite newton method for classification. *Optimization Methods and Software*, 17(5):913–929, 2002.

[65] I. Matthews and S. Baker. Active appearance models revisited. *Int'l J. Computer Vision*, 60(2):135–164, 2004.

[66] T. McInerney and D. Terzopoulos. A finite element model for 3d shape reconstruction and nonrigid motion tracking. In *Proc. Int'l Conf. Computer Vision*, pages 518–523, 1993.

[67] G. McNeill and S. Vijayakumar. Part-based probabilistic point matching using equivalence constraints. In *Advances in Neural Information Processing Systems 19*, pages 969–976. MIT Press, 2007.

[68] P. Meer. Robust techniques for computer vision. In M. Gerard and K. S. B., editors, *Emerging Topics in Computer Vision*. Prentice Hall, July 2004.

[69] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 27(10):1615–1630, 2005.

[70] E. Munoz, J. M. Buenaposada, and L. Baumela. Efficient model-based 3d tracking of deformable objects. In *Proc. Int'l Conf. Computer Vision*, pages 877–882, 2005.

[71] A. Myronenko, X. Song, and M. Carreira-Perpinan. Non-rigid point set registration: Coherent point drift. In *Advances in Neural Information Processing Systems 19*, pages 1009–1016. MIT Press, 2007.

[72] C.-W. Ngo, W.-L. Zhao, and Y.-G. Jiang. Fast tracking of near-duplicate keyframes in broadcast domain with transitivity propagation. In *ACM MULTIMEDIA'06*, pages 845–854. ACM, 2006.

[73] T. Ojala, M. Pietikainen, and D. Harwood. A comparative study of texture measures with classification based on feature distributions. 29(1):51–59, January 1996.

[74] S. Periaswamy and H. Farid. Medical image registration with partial data. *Medical Image Analysis*, (10):452–464, 2006.

[75] J. Pilet, V. Lepetit, and P. Fua. Real-time non-rigid surface detection. In *Proc. Conf. Computer Vision and Pattern Recognition*, pages 822–828, 2005.

[76] J. Pilet, V. Lepetit, and P. Fua. Fast non-rigid surface detection, registration and realistic augmentation. *Int'l J. Computer Vision*, 76(2):109–122, 2008.

[77] A. Qamra, Y. Meng, and E. Y. Chang. Enhanced perceptual distance functions and indexing for image replica recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 27(3):379–391, 2005.

[78] C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning.* The MIT Press, 2006.

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

[80] M. Salzmann, R. Hartley, and P. Fua. Convex optimization for deformable surface 3-d tracking. In *Proc. Int'l Conf. Computer Vision*, October 2007.

[81] M. Salzmann, V. Lepetit, and P. Fua. Deformable surface tracking ambiguities. In *Proc. Conf. Computer Vision and Pattern Recognition*, 2007.

[82] M. Salzmann, J. Pilet, S. Ilic, and P. Fua. Surface deformation models for non-rigid 3-d shape recovery. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(8):1481–1487, 2007.

[83] S. Sclaroff and J. Isidoro. Active blobs: region-based, deformable appearance models. *Comput. Vis. Image Underst.*, 89(2-3):197–225, 2003.

[84] K. Sim and R. Hartley. Removing outliers using the $L\infty$ norm. In *Proc. Conf. Computer Vision and Pattern Recognition*, pages 485–494, 2006.

[85] T. Sim, S. Baker, and M. Bsat. The cmu pose, illumination, and expression database. *IEEE Trans. Pattern Anal. Mach. Intell.*, 25(12):1615–1618, 2003.

[86] V. Sindhwani, P. Niyogi, and M. Belkin. Beyond the point cloud: from transductive to semi-supervised learning. In *ICML'05*, pages 824–831. ACM Press, 2005.

[87] J. Sivic and A. Zisserman. Video google: A text retrieval approach to object matching in videos. In *Proc. Int'l Conf. Computer Vision*, pages 1470–1477, 2003.

[88] M. Stegmann, B. Ersboll, and R. Larsen. Fame-a flexible appearance modeling environment. *IEEE Trans. on Medical Imaging*, 22(9), 2003.

[89] R. Szeliski and J. Coughlan. Spline-based image registration. *Int. J. Comput. Vision*, 22(3):199–218, 1997.

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

[91] D. Terzopoulos and D. Metaxas. Dynamic 3d models with local and global deformations: Deformable superquadrics. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 13(7):703–714, 1991.

[92] A. Tikhonov and V. Arsenin. *Solutions of Ill-Posed Problems.* Wiley, New York, 1977.

[93] L. Torresani, D. Yang, E. Alexander, and C. Bregler. Tracking and modeling non-rigid objects with rank constraints. In *IEEE Proc. Conf. Computer Vision and Pattern Recognition*, volume 1, pages 493–500, 2001.

[94] TRECVID. TREC video retrieval evaluation. In *http://www-nlpir.nist.gov/projects/trecvid/.*

[95] L. V. Tsap, D. B. Goldgof, and S. Sarkar. Nonrigid motion analysis based on dynamic refinement of finite element models. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 22(5):526–543, 2000.

[96] Y. Tsin and T. Kanade. A correlation-based approach to robust point set registration. In *Proc. European Conf. Computer Vision*, pages 558–569, 2004.

[97] V. N. Vapnik. *Statistical Learning Theory.* John Wiley & Sons, 1998.

[98] G. Wahba. *Spline Models for Observational Data.* CBMS-NSF Regional Conference Series in Applied Mathemtics. Society for Industrial and Applied Mathematics, Philadelphia, 1990.

[99] R. White, K. Crane, and D. A. Forsyth. Capturing and animating occluded cloth. In *ACM SIGGRAPH '07*, page 34, 2007.

[100] R. White and D. Forsyth. Retexturing single views using texture and shading. In *Proc. European Conf. Computer Vision*, pages 70–81, 2006.

[101] R. White and D. A. Forsyth. Combining cues: Shape from shading and texture. In *Proc. Conf. Computer Vision and Pattern Recognition*, pages 1809–1816, 2006.

[102] X. Wu, A. G. Hauptmann, and C.-W. Ngo. Novelty detection for cross-lingual news stories with visual duplicates and speech transcripts. In *ACM MULTIMEDIA'07*, pages 168–177. ACM, 2007.

[103] X. Wu, A. G. Hauptmann, and C.-W. Ngo. Practical elimination of near-duplicates from web video search. In *ACM MULTIMEDIA'07*, pages 218–227. ACM, 2007.

[104] X. Wu, W.-L. Zhao, and C.-W. Ngo. Near-duplicate keyframe retrieval with visual keywords and semantic context. In *ACM CIVR'07*, pages 162–169. ACM, 2007.

[105] J. Xiao, S. Baker, I. Matthews, and T. Kanade. Real-time combined 2d+3d active appearance models. In *Proc. Conf. Computer Vision and Pattern Recognition*, volume 2, pages 535–542, 2004.

[106] Z. Xu, R. Jin, J. Zhu, I. King, and M. R. Lyu. Efficient convex relaxation for transductive support vector machine. In

Advances in Neural Information Processing Systems 2007, NIPS 21, 2007.

[107] R. Yan, A. G. Hauptmann, and R. Jin. Negative pseudo-relevance feedback in content-based video retrieval. In *ACM MULTIMEDIA'03*, pages 343–346, 2003.

[108] A. Yuille and N. Grzywacz. A mathematical analysis of the motion coherence theory. *Int'l J. Computer Vision*, 3(2):155–175, June 1989.

[109] D.-Q. Zhang and S.-F. Chang. Detecting image near-duplicate by stochastic attributed relational graph matching with learning. In *ACM MULTIMEDIA'04*, pages 877–884. ACM, 2004.

[110] L. Zhang and D. Samaras. Face recognition from a single training image under arbitrary unknown lighting using spherical harmonics. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28(3):351–363, 2006.

[111] S. Zhang and S.-T. Yau. High-speed three-dimensional shape measurement system using a modified two-plus-one phase-shifting algorithm. *Optical Engineering*, 46(11), 2007.

[112] W. Zhao, Y. Jiang, and C. Ngo. Keyframe retrieval by keypoints: Can point-to-point matching help? In *CIVR06*, pages 72–81, 2006.

[113] W.-L. Zhao, C.-W. Ngo, H. K. Tan, and X. Wu. Near-duplicate keyframe identification with interest point matching and pattern learning. *IEEE Trans. on Multimedia*, 9(5):1037–1048, 2007.

[114] J. Zhu. Semi-supervised learning literature survey. Technical report, Carnegie Mellon University, 2005.

[115] J. Zhu. A study on 3d face from 2d image, 2007.

[116] J. Zhu, S. C. Hoi, and M. R. Lyu. Real-time non-rigid shape recovery via active appearance models for augmented reality. In *Proc. European Conf. Computer Vision*, pages 186–197, 2006.

[117] J. Zhu, S. C. Hoi, and M. R. Lyu. A multi-scale tikhonov regularization scheme for implicit surface modelling. In *Proc. Conf. Computer Vision and Pattern Recognition*, pages 1–7, 2007.

[118] J. Zhu, S. C. Hoi, and M. R. Lyu. Face annotation by transductive kernel fisher discriminant. *IEEE Trans. on Multimeida*, 10:86–96, 2008.

[119] J. Zhu, S. C. Hoi, and M. R. Lyu. Robust regularized kernel regression. *To appear in IEEE Trans. on Systems, Man, and Cybernetics - Part B: Cybernetics*, 2008.

[120] J. Zhu, S. C. Hoi, M. R. Lyu, and S. Yan. Near-duplicate keyframe retrieval by nonrigid image matching. In *ACM MULTIMEDIA'08*, 2008.

[121] J. Zhu, S. C. Hoi, Z. Xu, and M. R. Lyu. An effective approach to 3d deformable surface tracking. In *Proc. European Conf. Computer Vision*, 2008.

[122] J. Zhu, S. C. Hoi, E. Yau, and M. R. Lyu. Automatic 3d face modeling using 2d active appearance models. In *Proc. 13th Pacific Conf. Computer Graphics and Applications*, pages 133–135, 2005.

[123] J. Zhu and M. R. Lyu. Progressive finite newton approach to real-time nonrigid surface detection. In *Proc. Conf. Computer Vision and Pattern Recognition*, pages 1–8, 2007.

[124] J. Zhu, M. R. Lyu, and T. S. Huang. A fast 2d shape recovery approach by fusing features and appearance. *To appear in IEEE Trans. Pattern Anal. Mach. Intell.*, 2008.

[125] J. Zhu, M. I. Vai, and P. U. Mak. A new enhanced nearest feature space (enfs) classifier for gabor wavelets features based face recognition. In *Proc. Int'l Conf. Biometrics Authentication*, pages 123–131, 2004.