ABSTRACT

Title of dissertation: APPEARANCE MODELING UNDER GEOMETRIC CONTEXT FOR OBJECT RECOGNITION IN VIDEOS

Jian Li
Doctor of Philosophy, 2006

Dissertation directed by: Professor Rama Chellappa
Department of Electrical and Computer Engineering

Object recognition is a very important high-level task in surveillance applications. This dissertation focuses on building appearance models for object recognition and exploring the relationship between shape and appearance for two key types of objects, human and vehicle. The dissertation proposes a generic framework that models the appearance while incorporating certain geometric prior information, or the so-called *geometric context*. Then under this framework, special methods are developed for recognizing humans and vehicles based on their appearance and shape attributes in surveillance videos.

The first part of the dissertation presents a unified framework based on a general definition of geometric transform (GeT) which is applied to modeling object appearances under geometric context. The GeT models the appearance by applying designed functionals over certain geometric sets. GeT unifies Radon transform, trace transform, image warping etc. Moreover, five novel types of GeTs are introduced and applied to fingerprinting the appearance inside a contour. They include GeT

based on level sets, GeT based on shape matching, GeT based on feature curves, GeT invariant to occlusion, and a multi-resolution GeT (MRGeT) that combines both shape and appearance information.

The second part focuses on how to use the GeT to build appearance models for objects like walking humans, which have articulated motion of body parts. This part also illustrates the application of GeT for object recognition, image segmentation, video retrieval, and image synthesis. The proposed approach produces promising results when applied to automatic body part segmentation and fingerprinting the appearance of a human and body parts despite the presence of non-rigid deformations and articulated motion.

It is very important to understand the 3D structure of vehicles in order to recognize them. To reconstruct the 3D model of a vehicle, the third part presents a factorization method for structure from planar motion (SfPM). Experimental results show that the algorithm is accurate and fairly robust to noise and inaccurate calibration. Differences and the dual relationship between planar motion and planar object are also clarified. Based on our method, a fully automated vehicle reconstruction system has been designed.

APPEARANCE MODELING UNDER GEOMETRIC CONTEXT FOR OBJECT RECOGNITION IN VIDEOS

by

Jian Li

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2006

Advisory Committee:
Professor Rama Chellappa, Chairman/Advisor
Professor Larry Davis
Professor P. S. Krishnaprasad
Professor Min Wu
Dr. Shaohua (Kevin) Zhou

DEDICATION

To my parents.

ACKNOWLEDGMENTS

I would like to express my gratitude to all those who made this thesis possible. First of all, I want to thank my advisor, Professor Rama Chellappa, who has sustained the funding for my research, and has been giving me great guidance both academically and personally. His scholarly and honest attitude will always have a positive influence on my life.

I am also thankful to all my committee members, Professor Larry Davis, Professor P. S. Krishnaprasad, Professor Min Wu, and Dr. Shaohua (Kevin) Zhou. Professor Larry Davis has given me very encouraging comments about my research in the past, which means a lot to me. Professor P. S. Krishnaprasad and Professor Min Wu were in my proposal committee and have given me valuable suggestions. Dr. Shaohua Zhou has given me tremendous help and encouragement in my research, especially when I felt lost.

My days in Center for Automation Research have given very important experiences in my life, whether in terms of the interaction with fellow students, or the enlightening weekly seminars. I really appreciate this opportunity. What I learned in this center will always be a solid part of my background and useful for my future careers.

I would also like to thank all my colleagues, especially Haibin Ling, Jie Shao, Aswin Sankaranarayanan, Yang Ran, and Zhanfeng Yue for helpful discussions. I

really enjoyed my collaborations with them.

I would also thank three of my friends in other departments, who have given me some insightful suggestions from other perspectives. They are Bin Cheng from Department of Math in University of Maryland, Dongdong Wang from Department of Geography in University of Maryland, and Professor Jingyi Yu who is at Department of Computer Science in University of Delaware.

Last but not least, I am grateful for every bit of support and education my parents have given me. It always encourages me to recall my dad's persistent attempt to pass a calculus exam in his thirties while he had never learned high school math. I will be indebted forever to my mom's arduous support of my college education when she was laid off. They are my heros. This thesis is dedicated to them, Jiazhi Li and Jinrong Hu.

Table of Contents

List of Figures

Chapter 1

Introduction

1.1 Overview

In surveillance applications, object recognition is a very important high-level task both in itself and for other tasks including event and activity analysis. For example, if we can identify the object observed in one camera as one of the objects in the database, we can make inference about the activities near the surveillance sites based on the time of the observations. Or if this object is the same one as seen by another camera earlier, we can track this object persistently across camera networks.

In order to identify and track an object in surveillance videos, there are three different cues, three major difficulties, and three key types of objects. First we have a general discussions about these issues and then explain the focus of this thesis along these issues.

1.1.1 Three Types of Cues

There are three different cues for object tracking and recognition in videos. They include appearance, shape, and motion. Depending on the application, we can use different combinations of these cues for identification purposes.

An appearance mainly relates to the intensity pattern of the image of an ob-

ject. This pattern can be purely based on the statistics of intensity values, such as the intensity histogram. It can also be combined with spatial distribution of the patterns, for example textures. Another choice is to extract features as representation of the appearance, which include wavelet features [7, 36] at feature points, and representations of feature lines or curves.

The appearance of the object depends on both the intrinsic properties of the object, such as the albedo and the specular coefficient [2, 62], the illumination condition of the environment, such as lighting and shadows, and the color calibration of the camera.

A 2D shape concerns with the geometry of a 3D object projected onto a 2D image. Ideally, an accurate estimate of the 3D structure of the object can greatly improve the matching between two objects. Some major ways of reconstructing the 3D shape include structure from motion [23, 28], shape from shading [60, 61], shape from contour [25], or simply fitting a 3D deformable model to a 2D image [5]. In some cases where the projection model can be approximated using an affine camera and the object does not have much out-of-image-plane rotation, a 2D shape model may suffice for recognition purposes, for example, as in gait recognition [22, 53]. Even though the task is much simpler than processing 3D shapes, complex 2D deformations or articulated motions may cause difficulty in 2D shape matching [34].

Motion signature is another cue for object identification. It can be inferred by tracking a set of feature points, or from a dense optical flow field [35]. For example, when we classify a blob in videos to be a vehicle or a pedestrian, the periodic nature of the underlying motion can be used [45]. In gait analysis, the dynamics of the

movements is combined with the shape for improved recognition [53].

The three cues are usually coupled with one another. Structures of feature points or feature boundaries from intensity patterns are usually used to describe the shape. Appearance and shape variations over time are used to infer the motion.

In different applications, we will employ different cues. For example, for classification between vehicles and humans, apart from motion signature, we can also use the shape of the object [45]. For identifying a vehicle under arbitrary pose, both appearance and 3D shape information are needed [27, 28].

There are two reasons that appearance is often the most stable and the most used cue. First, as the intensity pattern is our primary observation while shape and motion are usually inferred from it, appearance keeps the most basic form of information and often contains fewer errors from processing. Second, the intensity pattern possesses a rich representation, while shape and motion are often represented as distributions of some simplified primitives such as feature points. On the other hand, the rich representation also causes the curse of dimensionality. So we usually need to use statistical pattern recognition techniques to reduce the dimension and study the structure of the data distribution [11].

Therefore the appearance feature is often used for identification whenever possible or necessary. For example, in the case of persistent tracking, if we assume the appearance of the object does not change over a short period of time, it becomes the most reliable cue, whether in human or vehicle recognition. In contrast, for human identification, gait analysis usually requires side views from the camera [22].

Clearly as the shape and motion of the object change, the appearance will

change as well. Thus the key in matching two appearances is to establish the correspondences based on shape and motion information.

1.1.2 Three Types of Correspondences

In general, correspondence is a broad term. Most of the time it means the spatial mapping between two sets of features from two image planes. The two sets usually come from the same patch on a 3D object, or similar patches on two objects that have similar topology. For example, the point in one image can be matched to another point in another image, which gives the correspondence of points. Similarly, there are feature curve correspondence and region correspondence. More details of spatial correspondences are discussed in section 2.1.2.

Correspondence can also mean the temporal alignment between two sequences. For example, when we have two videos of walking humans, we can temporally align the two sequences according to the pose of the human. Then if we play the two videos after alignment, the two people will walk in a synchronized fashion. In gait analysis, dynamic time warping based on dynamic programming [10] has been used for finding temporal correspondences [22].

Correspondence can also be used to describe the mapping of intensity values of two images. For example, in color calibration between two cameras, the color vector in one image is mapped to another color vector in the second image. The mapping can be affine or a non-linear function. It relates the two intensity values correspondingly. Similarly the illumination changes can be compensated if we find

the intensity correspondences under different lightings.

In this dissertation, we focus on the first type of correspondence. Essentially we are trying to reduce the variations in appearances due to geometric changes caused by shape and motion. The appearance is modeled under certain geometric prior information, or the so-called geometric context, so that the uncertainty due to spatial correspondence can be reduced. We study how to exploit the characteristics of the shape and the motion of the object for appearance modeling under geometric context.

1.1.3 Three Types of Objects

There are three major types of objects of primary importance in surveillance applications. They include human faces, vehicles, and walking humans. Human faces and walking humans are considered as two categories because they have very different properties and are used for human identification in very different scenarios.

For human faces, usually the appearance is represented as a normalized rectangular template. Then statistical learning is applied to the vectorized template to deal with changes in object poses and illuminations [31, 29]. Other model based approaches such as Active Appearance Models (AAM) [9], Elastic Graph Matching (EGM) [26], and complete 3D deformable models of faces [5] are also worthy of consideration.

Because the structure of the human face is relatively stable across different people, the shape formed by the feature points on the face is easier to model. In

Active Shape Model (ASM) [9] and AAM, Gaussian distribution is used to model small deformations of each feature point. But this model does not work for walking humans, where the topology between body parts changes significantly across poses. Moreover, since the shape of the face in 3D can be roughly approximated by the half sphere of an ellipsoid, it does not have an ad-hoc 3D structure like vehicles. These properties make the appearance of a human face very different from those of vehicles and walking humans.

Figure 1.1: *Examples of walking humans and turning vehicles in surveillance videos.*

In Fig. 1.1.3, we illustrate examples of pedestrians and vehicles in surveillance videos. We can observe the changes in the appearances of these two objects due to changes in the underlying geometric structures. But each of them has their own characteristics.

A walking human is a typical example of an object with non-linear deformation and articulated motion of body parts [30]. These deformations cause difficulties when two appearances of pedestrians need to be matched. However when different

people are at the same pose, their body parts topology are roughly the same. This property can be used to build an articulation-invariant appearance model.

Vehicle is a typical example of objects with complex 3D structures in surveillance videos [27, 28]. In order to identify these objects, we have to take the 3D structure into account. Appearance matching can only be done after the 3D pose is estimated, especially when there are significant view changes as the vehicle moves and turns on the ground plane. We develop a method to estimate the 3D structure of the vehicle from its planar motion, which helps to identify the vehicle.

1.2 Focus of this thesis

In this thesis, we focus on some of the key issues in human and vehicle identification. In particular, we build a generic framework that models the appearance of objects under certain geometric context. Then under this framework, an appearance model for a walking human is built and used for human identification. For objects that have complex 3D structures like vehicles, we develop a method to reconstruct the 3D model of an object from its planar motion. A brief introduction to the three parts of this dissertation is as follows.

1.2.1 Geometric Transform

In computer vision literature, traditionally the 2D affine transformation is used to register two rigid rectangular templates before the appearances are matched or statistical learning algorithms are applied. But quite often, appearance modeling

inside arbitrary shapes with non-rigid motion is needed. In section 1.1.1, shape and appearance are discussed as different cues for object recognition. But the shape of the object provide important geometric prior information for matching the appearance. Spatial correspondences can be inferred from the shape and help to align the two appearance patches before matching. This becomes particularly important when we need to fingerprint the appearance of humans with articulated motion or vehicles under arbitrary poses. So in Chapter 2, we propose a generic framework that models the appearance under certain shape context or combines appearance and shape information. It can be used for object recognition under various kinds of deformations.

The unifying framework is based on a general definition of Geometric Transform (GeT) [30]. The GeT incorporates the geometric context by applying designed functionals over certain geometric sets of an image. We show that linear and non-linear image transformations, Radon Transform, and trace Transform are special cases of GeT. We also propose some innovative ways of generating the geometric sets, such as from the contour boundary, or from skeletons of the shape, rather than simply from some feature points as in Active Appearance Model [8]. In the case when we only use sets of straight lines as in Radon Transform, we propose a multi-resolution representation that combines both shape and appearance information.

Chapter 2 is organized as follows. Section 2.1 gives the definition of geometric transform. The section starts with the motivation of such a transform, then formulates it mathematically by changing elements in Radon transform. Then we show that several common methods are special cases of this transform. The sec-

tion ends with a discussion of general principles for designing GeTs for appearance modeling. Section 2.2 focuses on GeTs designed to model the appearance inside arbitrary contours. Two types of GeTs are proposed. The first one uses the level set representation to generate a proper curve set. The second one generates dense correspondences inside the contour through matching the contour boundaries. This GeT is very useful when modeling the appearances of objects with articulated motion of body parts. It is crucial for building the geometric appearance model of humans in Chapter 3. Section 2.3 introduces three additional types of GeTs. Skeleton based GeT uses the correspondence between two skeletons of the shape to generate a dense mapping inside two contours. It is complimentary to the GeT based on shape matching introduced in section 2.2.2 when no canonical poses are available. An alternative functional is proposed in section 2.3.2 which can deal with occlusions that do not change the convex hull of the shape. A multi-resolution GeT is proposed in section 2.3.3 by changing the indicator function into a kernel function.

1.2.2 Articulation Invariant Model for Walking Human

An important type of GeT is the one based on shape matching. It is used as a generic tool for modeling the appearance inside two contours. In Chapter 3 this GeT is used to build a geometric appearance model for walking humans. Two important applications of this method are illustrated in Chapter 3. In section 3.1, we work on pedestrian appearance matching on a still-image-to-still-image setting. Assuming that body parts here have been segmented, we test each type of GeT by designing

them for body part recognition and combined human identification. GeT based on shape matching without using parts information gives superior recognition results compared to rigid template matching with body parts information. In section 3.2, the setting is video-to-video matching, again without prior body parts segmentation. First, this section give a complete illustration of how to use GeT based on shape matching for modeling objects with articulated motion. The training phase learns the shape space of typical poses. Then based on the shape space constructed for each pose, one can smooth the noisy human silhouette that results due to background subtraction errors. Second, this section shows how to use GeT for automatic body part segmentation. Once we have the segmentation, part-based recognition is used for human identification and surveillance video retrieval. This approach helps to solve the hard cases when the pedestrian changes only part of clothes such as putting on a jacket. This section also gives examples of how to use the skeleton based GeT for synthesis of human appearances at arbitrary poses.

1.2.3 Structure from Planar Motion

In fingerprinting vehicles in videos, we often need to extract a 3D model of the vehicle before modeling its appearance across different views. Therefore, in Chapter 4, a factorization approach is proposed for structure from planar motion that can be used to reconstruct the 3D model of a vehicle in surveillance videos [27, 28]. Our method reduces the rank constraints and simplifies the reconstruction into one singular value decomposition (SVD). It can be implemented efficiently without

estimating the fundamental matrices or epipoles [47]. Experimental results show that the algorithm is accurate and fairly robust to noise and inaccurate calibration.

Based on this factorization method, a system that can automatically reconstruct the 3D model of the vehicles is built. In order to construct a complete model when some faces are not observed, a priori information of the object can be used. In this case, an example of using the GeT for registering a 3D vehicle model in a tracking application is illustrated.

Chapter 4 is organized as follows. Section 4.1 gives the derivation of standard planar factorization. It presents a motivation for studying planar motion and a literature review of factorization approach. Then it discusses two key observations for planar motion and provides the mathematical solution based on these observations. The section concludes with a comparison of methods for general motion and quantitative and qualitative evaluations of our methods. Section 4.2 follows the line of work by Irani *et. al* [18], and discusses how to deal with feature points that have directional uncertainty. We approach this problem in two different ways and find a linear approximation of the formed measurement matrix using the observations. The experiment results show improvement over standard planar factorization when feature points have anisotropic observation noise. Section 4.3 clarifies two confusing terms, planar motion and planar object, and show they have a dual relation. The factorization for planar motion proposed in this dissertation has a dual theorem for a planar object. Section 4.4 introduces the automatic 3D vehicle model reconstruction system. The system is very efficient and can be easily implemented in real-time. As an extension, the section discusses how a prior 3D model can be used

for 3D tracking. The model-based approach can be combined with the factorization method for a complete model reconstruction, as discussed in the future work of this chapter.

1.3 Key contributions

Here are some key contributions of this dissertation:

- A unifying definition of GeT provides a generic framework for appearance modeling under geometric context. Several existing methods including the Radon transform, the trace transform, and image warping are special instances of our definition.

- Five novel types of geometric transforms are proposed, including the GeT based on level sets, the GeT based on shape matching, the GeT based on feature curves, the GeT with functionals to deal with occlusion, and the multi-resolution GeT. Each of these transforms has different properties and can be designed to model certain types of contoured appearances.

- Modeling the appearances inside arbitrary contours is studied in depth. Different kinds of shapes and motion models are taken into account, so that the selected transform domain representation has proper alignment for direct matching.

- GeT based on shape matching and GeT based on skeletons provide interfaces between shape matching and appearance modeling inside the shape. While the

representation of the shape usually takes a lower dimensional manifold such as contour boundaries or shape skeletons instead of the entire masked region or volume, the shape matching results generally provide correspondences in a lower dimensional surface, and these correspondences can be used to generate dense correspondences inside the entire region or volume. So essentially, these two GeTs use shape matching to obtain registration between two image planes.

- The geometric appearance model for a human provides a way to model the human appearance with articulated motion of body parts. While the GeT based on shape matching itself may seem to be a simple image warping technique, when used properly, this GeT can be applied to model complex motions. The key steps are: classification of shapes to be different poses, building part structures for shapes at each typical pose, and building correspondences for part structures at different poses. After these steps, the appearances inside any shape that belongs to these poses can be modeled, and each appearance can be transformed from one pose to another. Another advantage is that this model is implicit without feature points, unlike many deformable models.

- The GeT based on shape matching also provides an important cue for segmentation. The central idea is to register a new shape with a canonical shape at the same pose, then based on the shape matching, the segmentation inside the canonical shape can be transformed to obtain the segmentation inside the new shape. This method is illustrated in section 3.2 for successful body part segmentation of pedestrians. Although in some cases the segmentation results

may be sensitive to errors in shape matching, they can provide an important initialization based on which other segmentation methods can be used for refinement. This idea can be easily extended to applications such as medical segmentation. For example, if we can match the boundary of a leg bone with the boundary with a mean shape of the leg bone, then each sub-structure of this leg bone can be inferred from the substructure of the mean shape.

- For vehicles, we propose a simple planar factorization method that is used in an automatic 3D vehicle model reconstruction system. In section 4.1, we explained the importance of planar motion in surveillance applications. This planar factorization significantly simplifies the factorization for general motion. It is a key observation for understanding the structure of vehicles in surveillance videos.

- Planar factorization under uncertainty presents a view on how to deal with anisotropic noise with feature points, when the formed measurement matrix is a non-linear function of the observed feature points coordinates. Two different ways of approaching this problem and a proof for the validity of the approximations are given.

- The relationship between planar motion and planar objects is clarified and a dual relation is discovered between them. The analysis explores the concept of spatial and temporal rigidity, and their consequences on the duality relation between planar motion and a planar object. A dual theorem for planar factorization is also presented.

Chapter 2

Geometric Transform

2.1 Definition of Geometric Transform

Researchers familiar with image processing know about geometric transformation. Usually it is a synonym for image warping which establishes correspondences between two image planes, since the word *transformation* comes from coordinates transformation. In this section, we provide a broader definition of geometric transform, which in general is not necessarily a one-to-one mapping from the image domain to the transform domain. This definition characterizes a more general view of geometric corrections required for appearance models. We show under this definition, how different types of geometric context can be incorporated into the transform, and provide certain deformation invariant appearance representations.

First we illustrate the intuitive motivations behind the transform. Then the definition is formulated by changing the two key elements in Radon transform: *geometric set and geometric functional*. Special cases of this definition are explained, including trace transform and image warping. In the end, the general principles for designing GeTs for appearance modeling are discussed.

2.1.1 Background

The most common way of modeling the appearance of an object is by using templates, in which images of objects are usually cropped out from certain regions before learning algorithms are applied. Usually a 2D affine transformation is used to align the templates and derive pixel-wise correspondences. However, this is ineffective for modeling pose/view variations or large non-rigid motions, which have to be dealt with by the learning algorithm.

Quite often not enough sample images are available for learning these variations. In these cases incorporating prior knowledge such as geometric information can improve the recognition rate as well as tackle the problem of undersampling. In this section, we propose a generic way to incorporate geometric prior knowledge, which is also referred to as *geometric context*, into appearance models. The context can be based on a model, or inferred from the contour, or derived from prior knowledge of the underlying motion. In particular we want to model the appearance inside a contour, such as the appearance of humans with articulated motion and vehicles under different views after background subtraction. In these cases, objects have very large deformations and self-occlusions, rendering rigid transformations such as 2D affine transform insufficient to capture the correspondences. A general definition of GeT is introduced and guidelines are given to use it to transform an image for deformation-invariant modeling.

Many methods of incorporating prior knowledge have been proposed in the literature. The most comprehensive way is to have a full 3D generative model. Then

attempts for recovering the imaging process can be carried out. For example, it has been successfully used in face recognition [5] to deal with pose changes. However this approach is computationally intensive and requires significant prior information. Also, there is no guarantee of convergence to the global optimal solution, thus ineffective for low resolution imagery [62]. A simplified 2D model is found in Active Appearance Model (AAM) [9]. In AAM, the statistical behavior of a set of feature points to be tracked is modeled and used to generate a *normalized* appearance in the mean shape. It requires explicit detection of feature points and can only deal with small deformations that obey Gaussian distributions, so it is ineffective for tasks like modeling the appearance of pedestrians. Elastic Graph Matching (EGM) [26] is also a popular method of extracting the appearance signature with some prior knowledge of the geometric structure. Wavelet filters are often used to extract features at fiducial points that have certain link architectures. Similar to AAM, it needs feature points and an explicit model. In [20, 41], the authors use the trace transform to generate invariant features with respect to a group of affine transformations. Their approach uses the property of the transform to deal with 2D rigid motions as well as small non-linear deformations. It has the advantage of not using explicit models, but does not have the capacity to include complicated prior knowledge.

In this dissertation, we propose a transform based approach. We aim at providing a general framework that models large object deformations. When modeling the apperance inside a contour, we wish to incorporate the implicit knowledge inferred from the contour itself. Thus, the transform is used to represent the visual

pattern with certain invariance before we recognize it.

Inspired by the difficulties in appearance modeling inside a contour and a motivating example of GeT, Radon transform, a unifying definition is given in section 2.1.3. Many existing methods are shown to be special instances of GeT. Finally the guidelines for designing GeT that incorporates prior knowledge are given, along with comparisons with other descriptors such as SIFT.

2.1.2 Inspirations

2.1.2.1 Appearance inside a contour

The idea of *geometric transform* first comes up when the appearance inside a contour is to be modeled. The matching of two appearances can be viewed as a comparison of two 2D functions with compact support regions. For regular image matching, usually the two images are registered through an affine transform before the pixel-wise difference is compared. But for appearance inside a contour, since the support region has an arbitrary shape, it is hard to find a *transform* for direct comparison.

The role of such a *transform* is to align the corresponding parts. The key to accurate matching is to find the correspondences. Once the correspondences inside the two regions are established, comparisons can be made directly in the transform domain. Note that the correspondences do not have to be pixel-wise. They can be curve-to-curve, region-to-region or even volume-to-volume in the case of 3D data or videos. In Fig. 2.1, different types of correspondences are illustrated. Therefore,

correspondences can be viewed as a mapping between two sets. If each set contains more than one point, certain statistics can be computed over that set yielding features such as the mean or the variance, or even the histogram. These ideas are the basis of *geometric set* and *geometric functional* in our unifying definition.

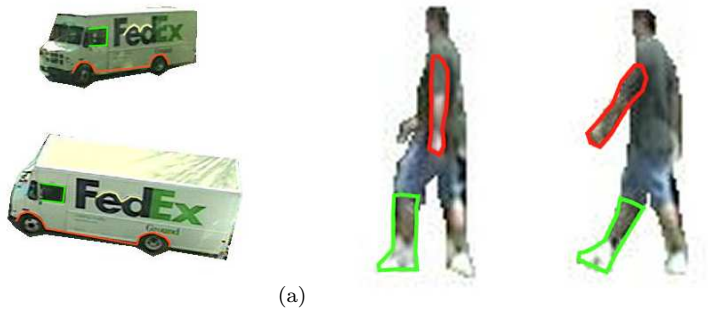


Figure 2.1: *Illustration of different types of correspondences. (a) The correspondence of curves across different views. (b) The correspondence of regions across different poses. Marked contours with the same color show the boundary of two corresponding regions.*

Another important fact is that these correspondences can be based on either explicit or implicit prior knowledge. AAM uses an explicit model for faces [9], where the feature points are tracked and used to generate the correspondences. But to model appearances inside an arbitrary contour, quite often there is no explicit model available. On the other hand, the contour boundary contains very important information about the correspondence. It can be used to infer the correspondence implicitly. Further, such knowledge can be combined with feature points to generate correspondences.

So our goal is to find a transform domain representation, where the support

region of the two appearance patches are properly registered based on explicit or implicit knowledge. The registration is not necessarily pixel-wise. In the transform domain, geometric deformations are taken care of for direct comparisons.

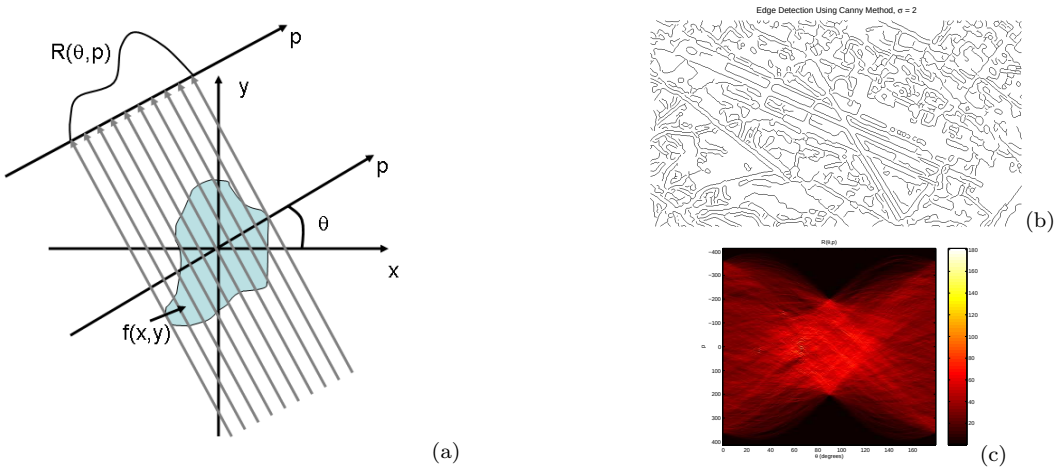
2.1.2.2 Radon Transform

To build a transform that solves the problem discussed in the last section, we first briefly review a special instance of GeT: the Radon transform, which has all the key features of our framework. In 2D, the Radon transform (RT) [19, 21] is defined as

$$\boldsymbol{R}(\theta, p) = \int\int I(x, y)\delta(x\cos\theta + y\sin\theta - p)dxdy. \tag{2.1}$$

RT applies integral operations to image $I(x, y)$ along a set of lines as illustrated in Fig. 2.2. It can also be viewed as a line projection to obtain the directional histogram.

RT has been extensively studied in computer tomography (CT) [21]. In CT, the focus is on image reconstruction from the transform domain. Given enough resolution in θ and p, the image can be fully reconstructed using filtered back-projection according to the Fourier slice theorem or using algebraic reconstruction techniques. In computer vision, the basic use of RT has been for line detection, which is also referred to as the Hough transform [13]. Usually an edge detector is applied to an image, then lines are detected through locating peaks in the Radon transform of the edge map. In Fig. 2.2, an edge map and its Radon transform are illustrated. By changing the geometric sets into arbitrary shapes such as circles,



(a) (b) (c)

Figure 2.2: *Illustration of Radon transform and its use for line detection. (a) Radon transform is essentially a line projection. (b) An edge map. (c) The Radon transform of the edge map in (b). Line detection is through locating the peaks in the transform domain and finding the lines corresponding to these peaks.*

rectangles or even non-parametric shapes, other shapes can be detected in a similar fashion. Many nice properties of RT such as the Fourier slice theorem are very useful to GeT as well.

RT carries two important elements: the geometric sets of straight lines, and the functionals defined over these sets, which are integrals. Through arbitrary choices of these two elements, we provide a general definition of geometric transform.

2.1.3 A unifying definition

Definition: Given any set $S \subset R^p$ and any function defined over it $f : S \mapsto R^q$, a *geometric functional* G_S is a functional that takes as input the f value over the set S, i.e., $G_S : f \to R^r$. We call S a *geometric set*. If we have a collection of sets $\{S(\omega)\}$ parameterized or indexed by ω, where each $S(\omega) \subset R^p$ is a geometric

set, the *geometric transform* of the function $f : R^p \mapsto R^q$ is the mapping of $\{S(\omega)\}$ to R^r by applying $G_{S(\omega)}(.)$ to f, i.e.,

$$\boldsymbol{R}(\omega) = \boldsymbol{R}(S(\omega)) = G_{S(\omega)}(f). \tag{2.2}$$

Remarks:

- Function $f : R^p \mapsto R^q$: Usually it corresponds to the intensity function of an image. For example, a 2D color image has $p = 2$ and $q = 3$. In most cases considered in this dissertation, f is chosen as the image intensity defined in a compact region $\Omega \subset R^2$, which is usually inside a contour. We call f the *appearance* inside Ω, sometimes denoted as A. However, the domain of interest in (2.2) is not limited to an image plane that lies in R^2. It can be generalized to $x - y - t$ plane in the spatial-temporal domain or $x - y - n$ domain where n refers to the index of the camera when we have multiple cameras.

- Geometric set $S(\omega)$: ω can be viewed as the transform domain coordinates. The mapping from the transform domain to the original image domain is through S. In other words, $S : \Pi \mapsto \Theta$ if $\Pi = \{permissible\ \omega\}$ and $\Theta = \{all\ subsets\ of R^p\}$. Using different choices of mappings S, we can obtain different types of GeTs. For example, if $S(\omega)$ only contains a point, it is called the GeT based on a point set.

- Geometric functional $G_{S(\omega)}(.)$: If we denote the function space as $\Im = \{f | f : R^p \mapsto R^q\}$, then the resulting functional is $G : \Im \mapsto R^r$. The only difference of geometric functional from a regular functional is that it is also a function

of the selected geometric set $S(\omega)$. For example, if the functional calculates the mean of a function, then the geometric functional gives the mean function value over the set $S(\omega)$.

- Dimension of the transform r: The dimension of the transform domain depends on the dimension of ω. But the dimension of the transformed output is r, which depends on the transform value. In this dissertation, mostly $r = 1$, meaning each set $S(\omega)$ is associated with a scalar that depends on the functional $G(.)$ operating on a function f over set $S(\omega)$. Ideally, the scalar gives the *signature* of the function f.

- The transform $\boldsymbol{R}$: Because $\boldsymbol{R}$ depends on both $\{S(\omega)\}$ and f, this definition is two-fold. If $\{S(\omega)\}$ is fixed, $\boldsymbol{R}$ is the GeT of f. If we fix f, the transform can be regarded as either the mapping $\boldsymbol{R} : \{S(\omega)\} \rightarrow R^r$ or the mapping $\boldsymbol{R} : \omega \rightarrow R^r$ according to the function f and functional $G_{S(\omega)}(.)$.

The key to using GeT as a knowledge-based representation is to embed the geometric context in the selection of sets $\{S(\omega)\}$. Intuitively, as discussed in section 2.1.2, a geometric set reflects the correspondences, and the functional $G(.)$ extracts the feature vector by obtaining the desired statistics over the set $S(\omega)$. But in practice, the choices of these two elements may have other interpretations. Details of how to select the set and functional are provided in sections 2.2, 2.3.1, and 2.3.2.

2.1.4 Special instances

It is interesting to show that many existing transforms and methods are special cases of the general definition of GeT in (2.2).

2.1.4.1 Radon transform

As mentioned above, RT is a special GeT. In n-dimensional RT [12, 19, 21], the collection of sets $S(\omega)$ are hyperplanes parameterized by $\mathbf{n}$ and p, such that $S(\omega) = \{x \in R^n | \mathbf{x}^T\mathbf{n} - p = 0\}$. So $\omega = \{\mathbf{n}, p\}$. The functional G is an integral operating on the set S.

$$\boldsymbol{R}(\mathbf{n}, p) = G_{S(\mathbf{n},p)}(f) = \int_S f(\mathbf{x})d\mathbf{x} = \int_\Omega f(\mathbf{x})\delta(\mathbf{x}^T\mathbf{n} - p)d\mathbf{x} \tag{2.3}$$

2.1.4.2 Trace transform

In 2D, by changing the functionals defined on the line set, RT is generalized to the trace transform. The trace transform has been successfully used for object recognition in [20, 41].

We give some examples of functionals in trace transform. In trace transform, the geometric set remains as straight lines. Denote points $\mathbf{x} = (x, y)$ in the line set $S(\mathbf{s}, p)$ as

$$x = ps_1 + ts_2, y = ps_2 - ts_1, -\infty < t < +\infty,$$

where $\mathbf{s}$ is the line normal, then function $f(\mathbf{x}) = f(t)$ for $\mathbf{x} \in S$. The geometric functional G can be defined as

$$G_S(f) = (\int |f(t)|^q dt)^r,$$

$$G_S(f) = (\int tf(t)dt)/(\int f(t)dt),$$

and so on. Different choices of functionals G give different statistics. In [41], the authors focus on designing the combinations of functionals so that the extracted features vary or remain invariant under a group of affine transforms, which is very useful for recognition of appearances inside a contour. However because they limit the geometric sets to be straight lines, their methods lack the ability to capture object appearances with large non-rigid motions.

2.1.4.3 Image Warping

Consider the case when point sets are selected as geometric sets, i.e., $S(\omega) = \{\mathbf{x}\}$. If the functional is an identity mapping, i.e., $G_{S(\omega)}(f) = f(\mathbf{x})$, then the definition in (2.2) generalizes the traditional definition of geometric transformation of images [57], which includes affine transformation, perspective transformation etc. Usually transformation refers to the type of the transform that only changes coordinates and there is a one-to-one mapping between the original domain and the transform domain. In this case, ω can simply be the new coordinate in the transform domain, say, $\omega = (\tilde{x}, \tilde{y})$, and the transformation of coordinate system is implemented in the mapping $S(\omega) = S((\tilde{x}, \tilde{y})) = \{(x, y)\}$. For example, if

$$\begin{bmatrix} x \\ \\ y \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} \\ \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} \tilde{x} \\ \\ \tilde{y} \end{bmatrix} + \begin{bmatrix} t_x \\ \\ t_y \end{bmatrix}, \tag{2.4}$$

then the GeT becomes a 2D affine transformation. Similarly the GeT becomes a perspective transformation with the help of depth information.

Here we further extend the ways of generating point sets so that more complicated image warping can also be included in GeT. Consider the feature point based warping used in AAM [9], a set of feature points are registered, i.e., in terms of the geometric set, $S(\tilde{\mathbf{x}}_p^c) = \{\mathbf{x}_p^c\}$ for known $\{\mathbf{x}_p^c\}_{p=1}^P$ and $\{\tilde{\mathbf{x}}_p^c\}_{p=1}^P$. An equivalent GeT to image warping is

$$S(\tilde{\mathbf{x}}) = \{\sum_{i=1}^{P} h_i(\tilde{\mathbf{x}})\mathbf{x}_i^c\}, \tag{2.5}$$

and the function $h_i(.)$ is an interpolator that satisfies $h_i(\tilde{\mathbf{x}}_j^c) = 1$ for $i = j$, $h_i(\tilde{\mathbf{x}}_j^c) = 0$ for $i \neq j$. In [9], it is shown that by properly choosing h_i, the corresponding warping can be reduced to piecewise affine or thin plate splines. Moreover, from the view of generating point sets for GeT, we can go beyond feature point based methods and use feature curves to find the dense correspondences. This is illustrated in section 2.3.1.

2.1.5 Designing GeT for appearance modeling

The special cases discussed above have been proven very useful in appearance modeling. Here we further exploit the key elements in GeT and develop more complicated transforms for different purposes. As seen in section 2.1.4, the geometric set is a crucial aspect of generalization from the RT in (2.3), which can be written as

$$\boldsymbol{R}(S(\omega)) = \int f(\mathbf{x})\chi_{S(\omega)}(\mathbf{x})d\mathbf{x}, \tag{2.6}$$

where the integral takes place in an arbitrary set $S(\omega)$ and χ is the indicator function whether $\mathbf{x}$ belongs to $S(\omega)$. Equation (2.6) is similar to the Hough transform applied

to detecting different shapes, but here the integrand is the intensity function so our model can extract appearance signatures. Proper selection of the set can help to find a representation with certain invariance. Ideally, the selected set incorporates meaningful prior knowledge or corresponds to regions of homogeneous distribution. Mostly $S(\omega)$ in (2.2) reflects correspondences inferred from prior knowledge. But for $S(\omega)$ in some variants of RT, the use of line sets is to obtain features with certain properties.

For modeling appearance inside two contours, our focus is on finding a transform domain representation that is invariant to relative motion between the two contours. Several typical kinds of motions studied in this thesis and the preferred GeTs are as follow.

- *2D rigid motion:* Use designed functionals in the trace transform as discussed in [41] or use the property of RT with respect to an affine transform for other line set based GeT.

- *3D rigid motion:* For example, vehicles. Preferably a 3D model can be used to generate the set as in the example in section 4.4.1. Then the color of each face can be fitted with a distribution.

- *Bending:* Such as human arms in Figure 2.4. GeT based on level set can be used as described in 2.2.1.

- *Local deformation:* For small local deformations, a multi-resolution GeT discussed in section 2.3.3 can be applied. For larger ones, we can apply a GeT

based on shape matching discussed in section 2.2.2 or AAM [9] with a set of feature points to generate the point sets as discussed in section 2.1.4.3.

- *Articulated motion of parts:* Objects with articulated motion (walking human) can be segmented into parts using a GeT based on shape matching as discussed in sections 2.2.2 and 3.2. The geometric set can be generated from the segmented parts or from a skeleton model in section 2.3.1.

- *Occlusion:* When a section of the contour is occluded, but its convex hull does not change much, an occlusion invariant GeT can be used with line sets and selected functionals as shown in section 2.3.2.

In summary, the set generation can be from the contour itself, using feature curves, and from an explicit model as discussed in sections 2.2, 2.3.1 and 4.4.1 respectively. Other possible ways of finding the geometric set can be through an analysis of appearances such as color-based segmentation, from dynamic relations across time such as using a motion model, or from multi-view relations when we have multiple cameras. They are beyond the scope of this dissertation.

In Fig. 2.3, we illustrate the relations of all the GeTs discussed in this dissertation. Three different elements in the original RT are changed: the geometric functional, the indicator function, and the geometric set. By changing the indicator function from a Dirac delta function to a Gaussian kernel, a multi-resolution GeT is introduced in section 2.3.3. According to the size of the geometric set, we have point set and curve set based GeTs. According to different applications of the transform, we have contour-driven GeT and traditional image warping. Detailed discussions

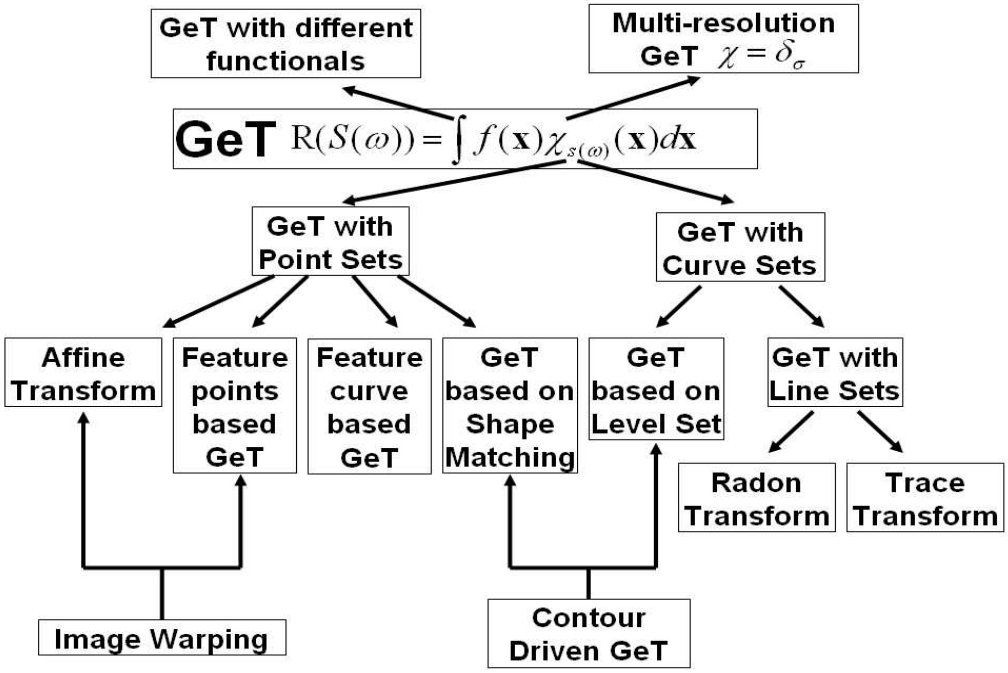


Figure 2.3: *The hierarchical graph of GeT families illustrates the relations between different types of GeTs.*

on how to design each part of the framework are provided later.

It is interesting to compare GeT with other local descriptor based appearance modeling such as SIFT [37][38]. There are three major differences: 1) Both methods can be used for deformation-invariant appearance models. GeT is generally a top-down approach designed for a global representation that incorporates prior knowledge and is invariant to certain deformations and articulations, while SIFT is generally used as a bottom-up approach that matches some local feature points before comparing the two images. 2) In the following sections, our method is shown to be specially suitable for modeling the appearance inside a contour, since the contour itself can provide useful knowledge for establishing correspondences. 3) As a general framework, GeT does not require feature points. But the matched feature points using SIFT can be incorporated in the GeT framework.

2.2 Contour-driven GeT

As mentioned in section 2.1.2, the idea of GeT first comes up when we try to model the appearance inside arbitrary contours. Since the relative motion between two shapes can be arbitrary deformation or even contains the articulated motion of parts, existing methods such as affine transformation or AAM cannot be directly applied. Essentially the spatial correspondence between the regions inside two arbitrary shapes needs to be implied. In this section, we propose two types of GeTs that make such inference based on the contour itself. In these methods, neither explicit models or feature points are required. Both curve sets and point sets can be generated from the contour.

The first one uses the level set representation to generate a proper curve set. The second one generates dense correspondences inside the contour through matching the contour boundaries, which is very useful when modeling the appearances of objects with articulated motion of body parts. The application is illustrated extensively in Chapter 3 for building the geometric appearance model of humans.

2.2.1 GeT based on level set

The *Level set* is an implicit way of representing the contour. Usually $\{\mathbf{x}|\phi(\mathbf{x}) > 0\}$ corresponds to the region inside the contour, $\{\mathbf{x}|\phi(\mathbf{x}) < 0\}$ is the outside region, and $\{\mathbf{x}|\phi(\mathbf{x}) = 0\}$ is the contour boundary. Though for a given contour, such a $\phi(\mathbf{x})$ is not unique, mostly $\phi(\mathbf{x})$ is selected as the signed distance transform, where the contour boundary is fixed and $\phi(\mathbf{x})$ is the solution of the Eikonal equation [40]:

$\|\nabla\phi\|^2 = 1$.Another choice of $\phi(\mathbf{x})$ is from the Poisson equation with the same boundary condition [14]: $\triangle\phi = \phi_{xx} + \phi_{yy} = -1$, which gives a smoother solution since the equation is of second order.

If the Eikonal equation is used, each level set corresponds to the point with equal distance to the contour boundary. The geometric set can be generated from these level sets. Then the GeT becomes

$$\boldsymbol{R}(c) = \int f(\mathbf{x})\delta(\phi(\mathbf{x}) - c)d\mathbf{x}. \tag{2.7}$$

For $c > 0$, the integral is over the level set inside the contour. Because when the contour translates or rotates, the relative position of each level set does not change, so the transform in (2.7) is translation and rotation invariant, and it can be easily made scale invariant by changing c. In addition, it is not sensitive to the bending of the contour as shown in Fig. 2.4.

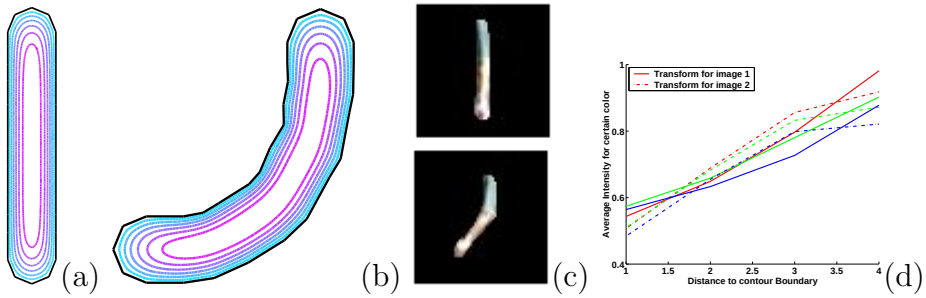


Figure 2.4: *Illustration of the contour-driven GeT applied to bending. Images (a) and (b): using the level set as the geometric set makes it insensitive to bending. Images (c) and (d): two arm images, and the average intensities of the r, g, and b color components along each level for the two arm images.*

We show that this selection of set is particularly useful for modeling the appear-

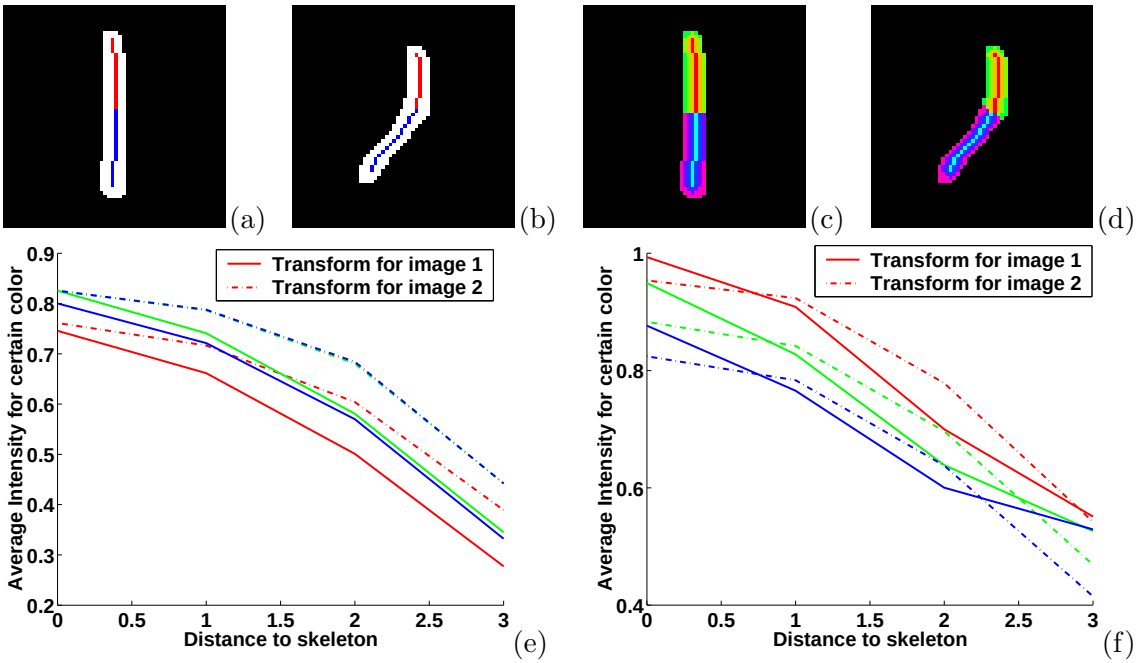


Figure 2.5: *Illustration of modified GeT based on level set designed for human arms. (a)(b) The skeleton of the contour is obtained by thinning the contour, and the skeleton is divided into upper and low parts. (c)(d) The geometric sets consist of points with equal distance to the upper or lower skeletons. Each color represents a geometric set. (e) The GeT finds the average intensity of the two arm images in Fig. 2.4 over the geometric sets for the upper part of the arm. (f) GeT for the lower part of the arm.*

ance of a single component contour with bending and small distortion, for example when it is applied to modeling human arms in section 3.1. In Fig. 2.4, we plot the curves of $\boldsymbol{R}(c)$ displaying the average intensity of arm images along different levels c. We observe that these curves are clustered together. This indicates that our transform is somewhat insensitive to bending. The essence of this GeT is the rough correspondences of these curves generated from the level set.

In practice, since the contour boundary is noisy and it may change the topology of the level set, so a modified version can be used instead. First, the contour region is thinned to a skeleton as in Figs. 2.5 (a) and (b). Then the level set curve is generated from the distance transform with respect to the skeleton. The modified level set still keeps the topology shown in Fig. 2.4, but each set contains points with equal distance to the skeleton instead of the contour boundary. These sets are marked with different colors in Figs. 2.5 (c) and (d).

The above method is good for the contour with a single skeleton. For improved modeling of the appearance of human arms, these sets are further divided into upper and lower parts according to whether the point is closer to the upper or the lower skeleton. The functional calculates the average intensity over these sets. In Figs. 2.5 (e) and (f), we see that the resulting transforms of the two arm images are very close to each other. This modified transform is used for bending invariant matching of human arms in section 3.1.

2.2.2 GeT based on shape matching

Point sets can be generated by matching two contours [54]. In this case, the transform domain coordinate is denoted as $\tilde{\mathbf{x}} \equiv \omega$ and the mapping $S(\tilde{\mathbf{x}}) = \{\mathbf{x}\}$ reflects the dense point-to-point correspondences inferred from shape matching. Here we focus on contour based shape matching [32], among which a descriptor called shape context raises the benchmark in [3][34] and finds correspondences without feature points. We incorporate this idea into a GeT and use it to model the appearance of pedestrians, where the objects have articulated motion of parts and self occlusions.

A GeT based on shape matching is defined as follows: suppose we have two 2D regions Γ_0, Γ_1 bounded by two contours C_0 and C_1 respectively. Denote the intensity function in region Γ_0 as A_0. Let the two contours be represented by the sampled points on them, i.e., $C_0 = \{\mathbf{x}_i^c | i = 1, ..., N_0\}$, $C_1 = \{\tilde{\mathbf{x}}_i^c | i = 1, ..., N_1\}$. Then by applying any shape matching method to the two sets of points, one-to-one mapping of their subsets are found as $\tilde{\mathbf{x}}_i^m \leftrightarrow \mathbf{x}_i^m$, for $i = 1, ..., M$ and $M \leq \min(N_0, N_1)$. Design geometric sets for the interpolated dense correspondences as

$$S(\tilde{\mathbf{x}}) = \{\sum_{i=1}^{M} h_i(\tilde{\mathbf{x}}) \mathbf{x}_i^m\}, \tag{2.8}$$

for $\mathbf{x} \in \Gamma_0$, $\tilde{\mathbf{x}} \in \Gamma_1$ and $h_i(.)$ satisfies $h_i(\tilde{\mathbf{x}}_j^m) = 1$ for $i = j$, $h_i(\tilde{\mathbf{x}}_j^m) = 0$ for $i \neq j$ as shown in section 2.1.4.3. *Identity mapping* is used as functionals over these sets. Then the corresponding GeT, denoted as $\boldsymbol{R}_{\Gamma_0\Gamma_1}$, is the GeT of the function A_0 based on shape matching between Γ_0 and Γ_1. $\boldsymbol{R}_{\Gamma_0\Gamma_1}$ transforms the appearance A_0 inside the contour C_0 to the appearance A_1 inside the contour C_1. Although here we still

use point based image warping as in (2.5), $\mathbf{x}_i^m$'s are not necessarily points at places with distinctive features such as corners or points with large curvatures.

This GeT can be used to obtain a *pose-invariant* representation of a pedestrian's appearance. Suppose two images of pedestrians are to be matched after background subtraction. It is difficult to compare directly because of differences in poses and sizes of the silhouettes. However, if we focus on the side view of the person, a walking human usually has six typical poses as in Fig. 2.6(a). Although each person may have a different shape and walk differently, the topology of body parts remains roughly the same for different people at the same pose. We can use this property to *normalize* appearances at the same pose. By *normalization* we mean warping the appearance to be inside a canonical shape through GeT for pixel-wise comparison. This way, shape variations of different people are handled in a fashion similar to obtaining a *normalized* appearance of a face inside a mean shape in the AAM.

So given only one image of a pedestrian with an arbitrary pose, we can obtain the *normalized* appearance of pedestrians at all other poses as illustrated in Fig. 2.6(d), by using GeT based on shape matching. We assume to have canonical silhouettes at six typical poses $\{\Gamma_i | i = 1, ...6\}$ as shown in Fig. 2.6(a), along with eight-part segmentation $\{\gamma_i^k | i = 1, ...6, k = 1, ..., 8\}$. We first *normalize* the pedestrian's appearance inside the canonical silhouette for the closest pose, before synthesizing the pedestrian's *normalized* appearance at other poses. Denote the appearance inside the pedestrian's silhouette Γ_0 as A_0. Here are the two steps:

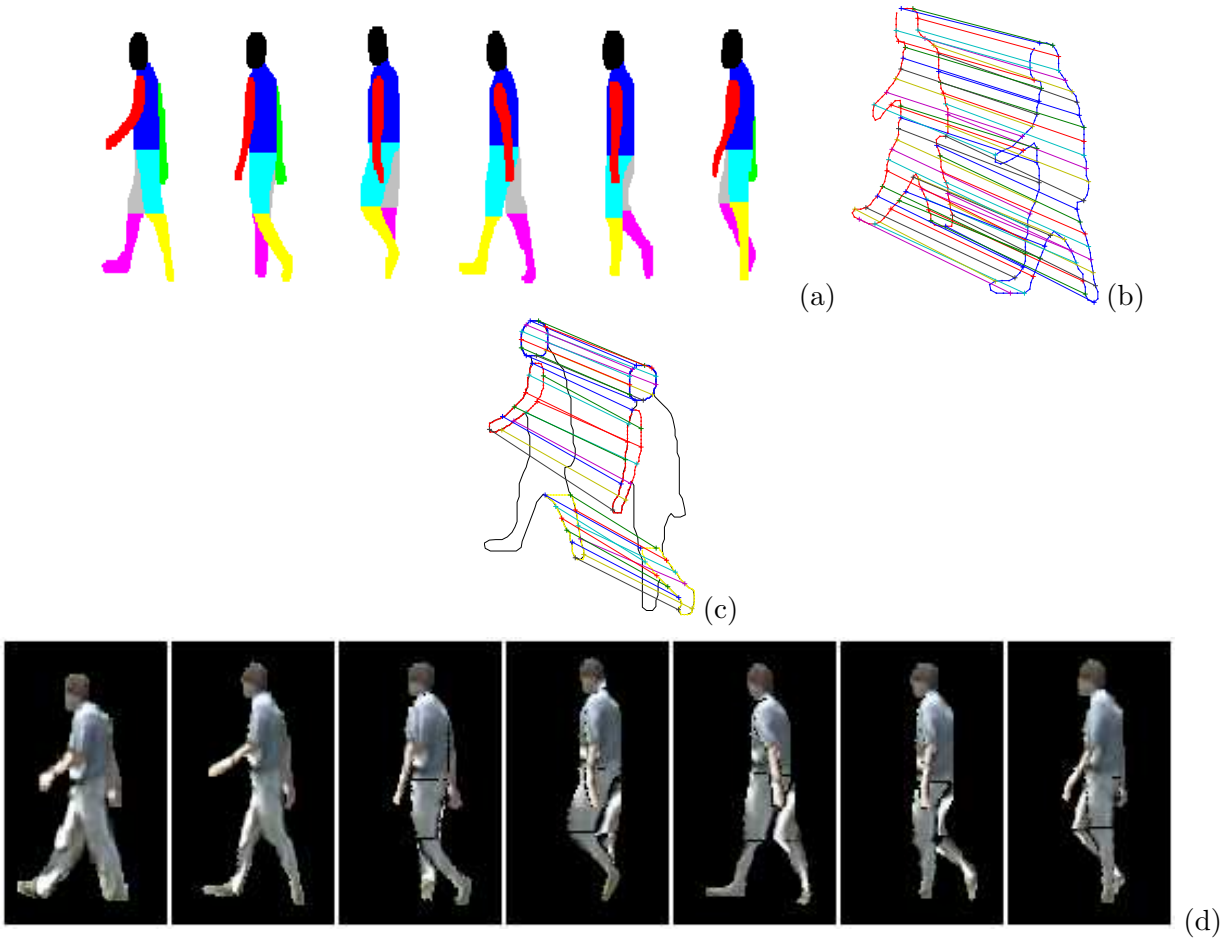


Figure 2.6: *(a) Canonical silhouettes of six typical poses of a walking human along with segmentation of body parts taken from the USF database [42]. (b) Shape matching between a pedestrian's silhouette and the canonical silhouette at a similar pose. The corresponding points are used for the GeT* $\boldsymbol{R}_{\Gamma_0\Gamma_j}$*. (c) Shape matching between parts at different poses used to generate the GeT* $\boldsymbol{R}_{\gamma_j^k\gamma_i^k}$*. Here we show the matching of head, left arm and left lower leg. By applying GeT to each part in a certain order, the appearance can be transformed from one pose to another. (d) The first image is the sample image and the second image is the synthetic image at the closest pose, followed by synthesized normalized appearance at the remaining five typical poses.*

1. Use shape matching to find the most similar pose:

   $j = argmin_{i=1,...6}\ MatchCost(\Gamma_0, \Gamma_i)$. Use GeT $\boldsymbol{R}_{\Gamma_0\Gamma_j}$ to find the normalized appearance A_j inside Γ_j, as illustrated in Fig. 2.6(b).

2. Synthesize from pose j to pose i. For body part k, transfer the appearance a_j^k inside γ_j^k to appearance a_i^k inside γ_i^k by applying $\boldsymbol{R}_{\gamma_j^k\gamma_i^k}$ to a_j^k, as illustrated in Fig. 2.6. Transform each part in the order from the part farthest from the camera to the part closest to the camera, so that self-occlusions are dealt with.

The final representation of the appearance does not depend on the initial pose. Results in Fig. 2.6 show that the designed GeTs capture the structure and deformation of the parts very well. After applying GeT, appearances inside two contours can be compared directly in the transform domain. Here for shape matching, we use the inner distance based shape context method [34], which is insensitive to articulations. $h_i(.)$'s are chosen to generate thin plate spline interpolations.

Fig. 2.6 shows how GeT can deal with large deformations and articulations without feature points, while AAM is known to be ineffective for dealing with large deformations of feature points that do not obey Gaussian distributions [9]. The idea of modeling appearances based on shape matching has been illustrated in [3], but here we formulate using a GeT that can handle articulations and self-occlusions, and apply it to model the appearances of real world objects. This method is applied to appearance based pedestrian recognition in section 3.1.

From the above example, we summarize in Fig. 2.7 the GeT based on shape matching for deformation invariant models that serve as an interface between shape

matching and appearance modeling. The key assumption is: for the contour of an object at a given time, one can always find the contour of any other object at a corresponding time, and the regions inside these two contours have similar topology. These two contours are considered to be of the same pose.

a. *Construct a normalized shape space for each typical pose.* This has three steps. a1: All the training shapes are classified based on their corresponding poses. a2: For each pose, an examplar shape is selected as the reference shape. All the masks of the shapes are normalized using the GeT based on matching between these shapes and the reference shape. a3: The normalized masks are used to construct the shape space, and the mean shape of each pose is found as $\{\Gamma_i | i = 1, ..., K\}$.

b. *Define correspondences between regions bounded by the mean shapes.* These regions are segmented into parts. Then the shape matching between the corresponding parts is used to find dense correspondences.

c. *Given a new appearance A_0 inside a contour Γ_0, classify its pose and normalize the appearance for this pose.* The pose can be found by finding the closest match Γ_j among the mean shapes. The normalized appearance A_j is from $A_j = \boldsymbol{R}_{\Gamma_0\Gamma_j}[A_0]$.

d. *Synthesize the appearance for other poses through the GeT based on the matching of parts, or obtain parts segmentation from the GeT of the segmentation of the mean shapes.*

Figure 2.7: *GeT based on shape matching for deformation invariant models.*

This whole framework is illustrated in section 3.2. In the GeT used in Fig. 2.6, the training phases of steps (a) and (b) are simplified by using canonical contours instead of learning the shape space. The construction of shape space is explained in

section 3.2.1. The segmentation in step (d) is performed as follows: the binary mask of each part is generated from part γ_i of Γ_j. Based on the shape matching between Γ_j and Γ_0, the mask image χ_{γ_i} is transformed using $\boldsymbol{R}_{\Gamma_j\Gamma_0}$, then the transformed mask gives the segmentation of part γ_i of Γ_0. Body parts segmentation is illustrated in section 3.2.2.

A final note is about the interaction between shapes and appearances. Here the focus is to obtain an appearance model independent of the shapes, so pixel-wise comparison can be made. It is good for application such as appearance based recognition of pedestrians at different poses. In some applications, shape information needs to be combined with appearance information for identification purposes.

2.3 Miscelaneous GeTs

We introduce three additional types of GeTs. Instead of using the contour boundary, the feature curve/skeleton based GeT uses the correspondence between two skeletons of the shape to generate dense correspondences inside two contours. It can be viewed as an interface between skeleton-based shape matching and appearance modeling. It is also complimentary to the GeT based on shape matching proposed in section 2.2.2 when no canonical poses are available.

In section 2.3.2, an alternative functional is proposed to deal with occlusions that do not change the convex hull of the shape. This section concludes with a presentation of multi-resolution GeT (MRGeT) which is formed by changing the indicator function into a kernel function.

The application of skeleton based GeT is illustrated in section 3.2.3 for synthesizing the appearance of human with arbitrary poses. The other two GeTs are illustrated in section 3.1 and designed to model the appearance of body parts.

2.3.1 Feature curve/skeleton based GeT

Point sets are used in GeT based on shape matching and AAM. These sets are generated from matched contour points or feature points. But sometimes we only have correspondences of some feature curves or skeletons. Direct interpolation becomes difficult since the points on the curve may be nearly colinear, as illustrated in Fig. 2.8(b). In this section, a feature curve based point set generation is proposed. The feature curves can be curves along the intensity edges of the image as

in Fig. 2.1 (a). Or it can be curves inferred from the shapes such as the skeleton. The GeT based on the latter is used as an interface between the skeleton based shape matching [46] and appearance modeling. The skeleton can be generated from morphological operations, the medial axis space[46], principal curves or some prior models space[50].

For example in Fig. 2.8, in order to deal with bending, we can also use the correspondence between two skeletons of the shapes. The local coordinate system along the skeleton can be specified through differential geometry. For example in Fig. 2.8, at every point on the skeleton, the y-axis in Fig. 2.8 is the tangent vector of the curve, while the x-axis is the normal vector. This coordinate system can be used to generate dense point-to-point correspondences by retaining the local coordinates at each point.

A GeT based on feature curves is defined as follows. Suppose we have two matched curves as in Fig. 2.8, $C_0 = \{(x(s), y(s))|s \in [0,1]\}$, $C_1 = \{(\tilde{x(s)}, \tilde{y(s)})|s \in [0,1]\}$ and for $s \in [0,1]$,

$$S((\tilde{x}(s), \tilde{y}(s))) = \{(x(s), y(s))\}. \tag{2.9}$$

Then one simple way of generating $S((\tilde{x}, \tilde{y}))$ for any $(\tilde{x}, \tilde{y})$ inside the contour is as follows:

1. Define the local coordinate system for every point on C and $\widetilde{C}$ that reflects local correspondences. For example, the tangent and normal vectors of the curve at that point can be chosen as bases. But for end points, joints, or

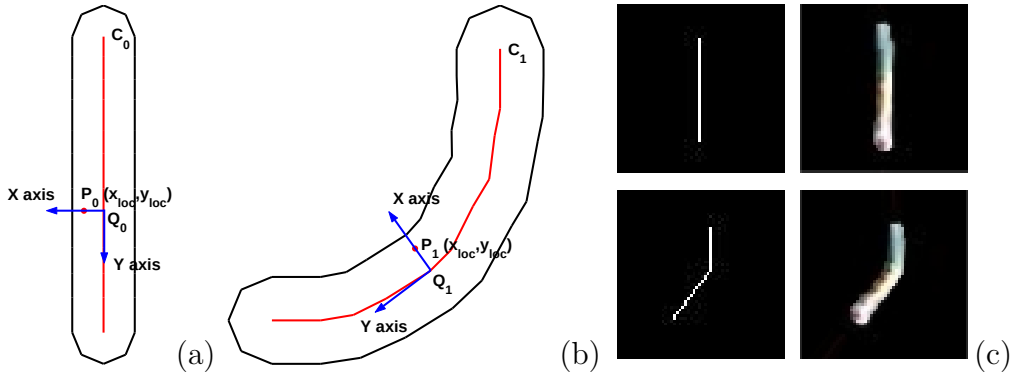


Figure 2.8: *Illustration of mapping through local coordinate systems defined by skeletons. Images (a,b) show how to map P_1 to P_0 according to the local coordinate system at Q_1, which is the closest point to P_1 on curve C_1. C_0 and C_1 are matched curves and Q_0 corresponds to Q_1. Image (c): The skeletons of arm images in Fig. 2.4 and the synthetic appearance generated using the skeleton based GeT from one arm to another are shown.*

discontinuities, the system needs to be chosen carefully.

2. For each $P_1 = (\tilde{x}, \tilde{y})$ inside the contour, find $Q_1 = argmin_{Q \in C_1} |Q - P_1|$. Then find the local coordinate of P_1 at point Q_1, denoted as (x_{loc}, y_{loc}).

3. $S((\tilde{x}, \tilde{y}))$ will be $P_0 = (x, y)$ that has the local coordinate (x_{loc}, y_{loc}) (rescale if necessary) at point Q_0, which is the corresponding point of Q_1 on curve C_0.

The corresponding GeT is denoted as $\boldsymbol{R}_{C_0 C_1}$. In Fig. 2.8, we illustrate synthetic images of human arms from a skeleton based GeT. Here the skeleton is obtained by thinning the shape. The synthetic images are very close to the real ones.

For a more complicated skeleton based GeT that can be used for synthesizing the appearance of a human with arbitrary articulations, the following steps can be

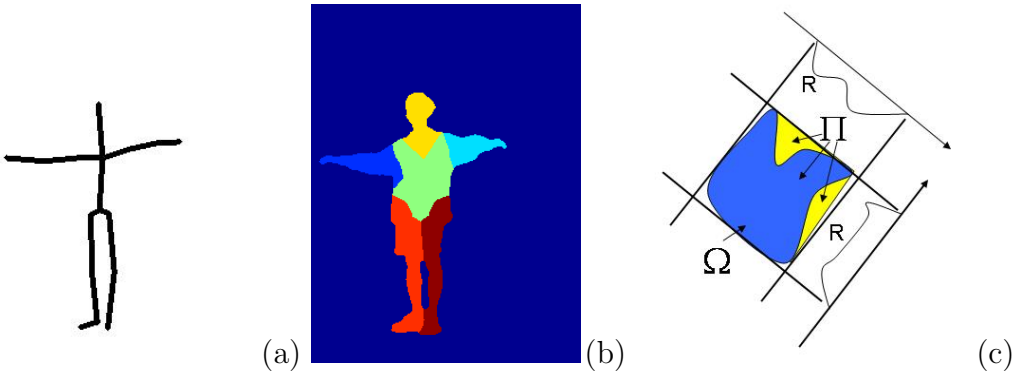
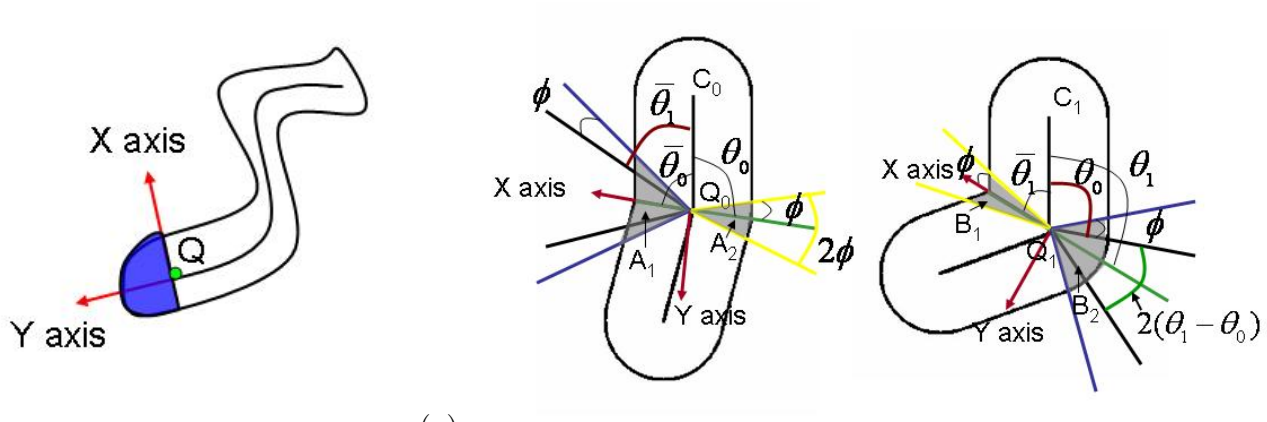


Figure 2.9: *(a,b) Illustration of skeleton based transform for the appearance of human with articulations: (a) The skeleton of a human silhouette. (b) The part segmentation. (c) Illustration of reconstructing the convex hull from the support of the transform.* Ω *is the original contour.* Π *is the convex hull of this contour. From the support of the transform* R *along each direction, the convex hull* Π *can be found.*

followed. First, associate each segment of the skeleton with a body part, then the human silhouette is divided into parts according to which skeleton each point is closest to. This is illustrated in Fig. 2.9. Second, for each body part, the GeT discussed above is used. The transform of each part is applied in the order of from the part farthest from the camera to the closest part, to deal with occlusions. Also note that, in order to have smooth synthesis near the boundary between body parts, a small margin is added to each part boundary. This method is illustrated in the experimental section.

The transform near the end points and joint points has special properties. Retaining the local coordinates near the end point leads to a rigid transformation of the nearby region. For example, in Fig. 2.10(a), at the end point Q, the coordinate system is selected the same way as in Fig. 2.8, then the region marked in blue is



(a) (b)

Figure 2.10: *Illustration of coordinate systems near the end points and joint points. (a) At end points, the blue region has rigid transformations according to the location and normal direction of Q. (b) At joint points, the X axis is selected to be the bisector of the two tangent lines. Expansion and shrinking is through the mappings from A_1 to B_1 and from A_2 to B_2. The angle corresponding to these four regions are: $A_1 : 2(\bar{\theta}_0 - \bar{\theta}_1) + 2\phi$,$A_2 : 2\phi$, $B_1 : 2\phi$,$B_2 : 2(\theta_1 - \theta_0) + 2\phi$. Bending is handled through linearly warping these regions along the angler direction. The rest of the region follows the original scheme.*

transformed rigidly according to the position of Q and the normal of the curve at Q. The blue region contains points whose closest point on the curve is Q.

The first order derivatives at joints may be discontinuous. For example, Fig. 2.10(b) shows the joints Q_0 on C_0 and Q_1 on C_1. In this case, the angle changes due to bending, and some regions near the joint have non-rigid distortion. The goal is to find proper selection of these regions A_1,A_2,B_1 and B_2, as well as the mapping between A and B. A selection scheme is proposed in Fig. 2.10(b). The x-axis is chosen to be the bisector of the angle between two tangent lines at the joint point. The green line marks this bisector. $\theta_1 > \theta_0$ is assumed, then the area is expanded from A_2 to B_2. If the original scheme is used, for C_1 on the right, the region inside the angle θ_0 marked with red color finds its corresponding part as the region inside the angle θ_0 on the left. However, for region inside the angle $2(\theta_1 - \theta_0)$, the appearance information is missing. In order to fill in this gap, the region A_2 to be expanded is selected to be inside the yellow angle 2ϕ, where ϕ is a free parameter called the marginal angle here. It determines the area of A_2. Since points in A_2 do not follow the original scheme, the gap region on the right needs to include the marginal angle. So B_2 is selected as the region inside the angle $2(\theta_1 - \theta_0) + 2\phi$. Points outside these two regions follow the original scheme. The mapping from A_2 to B_2 expands A_2 linearly along the angler direction. The mapping between A_1 and B_1 is similar.

Not all GeTs are invertible. Recall that the basic RT is fully reconstructible with proper selection of the resolution of the projection [19, 21]. However, when the geometric sets are changed into arbitrary ones as in (2.6), the inverse transform

may not be available. Intuitively, we need to have enough number of projections along these sets to make an invertible transform. In this case, it is possible to form algebraic equations of pixel values and apply algebraic reconstruction techniques [21] to solve these equations. For some applications such as image synthesis, the transform needs to be inverted so that the appearance can be reconstructed from the transform. A simple way of building an invertible transform is to select point sets that carry a bijective mapping [1] from (x, y) to $(\tilde{x}, \tilde{y})$ in the transform domain. This is discussed in sections 2.2.2 and 2.3.1. But the inherent problem with imposing point-to-point correspondence is that when the resolution is low, the correspondence is not reliable. In that case using a geometric set that contains more than one point may be more reliable. The multi-resolution representation in section 2.3.3 also helps to solve this problem.

Feature curve based GeT is complementary to GeT based on shape matching when no canonical shapes are available or when feature curves can be more reliably tracked. For example, the appearance of a human with arbitrary articulations cannot be handled by methods in section 2.2.2. But it is possible to use the skeleton based GeT and the result is shown in section 3.2.3. One may argue that the feature curves can be represented by a set of points, then similar interpolation as in Eq. (2.8) can be used. But all the interpolation methods including thin plate spline only work well for region inside or near the convex hull of these points. So when the convex hull of the feature curves does not cover most of the contour region, such as the case in Fig. 2.8 (a), simple interpolations are not effective.

[1] A mapping which is one-to-one and onto and thus invertible.

2.3.2 Selection of geometric functional

In (2.6), the geometric functional can be changed along the geometric set as in trace transform [41] to obtain different statistics. Some examples of the functionals are listed in section 2.1.4. We now show a useful geometric functional that helps to deal with occlusions.

The following geometric functional can be used to find the average intensity over set S.

$$G_s(f(\mathbf{x})) = \frac{\int_s f(\mathbf{x})d\mathbf{x}}{\int_s H(f(\mathbf{x}))d\mathbf{x}}, \tag{2.10}$$

where $H(.)$ is the Heaviside function and we set $H(0) = 0$. Here f corresponds to the intensity in a contour, and $f(x) > 0$ for $x \in \Omega$, and $f(x) = 0$ outside Ω. So $H(f(x, y))$ is equivalent to $\chi(\Omega)$ and gives the mask of the contour region.

Now we show why the functional in (2.10) makes a GeT insensitive to occlusions that do not change the convex hull of the shape. First, the set S is selected to be straight lines, so the GeT is

$$\boldsymbol{R}(\theta, p) = \frac{\int f(x, y)\delta(x\cos\theta + y\sin\theta - p)dxdy}{\int H(f(x, y))\delta(x\cos\theta + y\sin\theta - p)dxdy}. \tag{2.11}$$

In (2.11), if f is constant inside Ω, then the transform will be constant when the line passes through the contour region and zero otherwise as shown in Fig. 2.9(c). Therefore, the shape information is partly lost. From the support of the transform, we can only reconstruct Π, the visual hull of the contour. This reconstruction is well studied in computational geometry. For all the contours that have the same convex hull and the same intensity inside, their GeTs are identical.

So the GeT in (2.11) is unable to differentiate these appearances. However

it becomes useful when the appearance inside the convex hull instead of the exact shape is to be modeled. For example, part of the contour is missing but the visual hull does not change much. Fig. 2.11 illustrates cases like this, when a human walks sideways to the camera and the torso is partly occluded. If the average intensity along each line does not change much because of occlusions, such a GeT can be used for occlusion invariant appearance models.

To illustrate the accuracy of such a representation, we study the reconstructions from two different transforms in Fig. 2.11. The first one is from the original RT of a function $f(\mathbf{x})$ inside Ω as in (2.1). The construction is through filtered back-projection. As observed from the figure, the exact shape and appearance are recovered. But the appearance in the missing part is not inferred. The other one is from the GeT defined in (2.11), which finds the average intensity along each line. From the the support of GeT, the binary mask of convex hull is obtained as $\chi(\Pi)$. Then the RT of $f(\mathbf{x})$ inside the convex hull Π is estimated as

$$\tilde{\boldsymbol{R}}(\theta, p) = \boldsymbol{R}(\theta, p) \int \chi(\Pi)\delta(x\cos\theta + y\sin\theta - p)dxdy, \tag{2.12}$$

which is the product of the average intensity along each direction and the corresponding RT of the binary mask. Finally reconstruction is achieved by applying the filtered back-projection to the estimated RT $\tilde{\boldsymbol{R}}$. This reconstruction gives an estimate of the appearance inside the convex hull. A comparison of this estimate with the true appearance inside the convex hull in Fig. 2.11 shows fairly accurate reconstruction. Thus using such a GeT helps to represent the appearance with partial occlusions. In section 3.1, a method based on this GeT is used for fingerprinting tor-

sos with occlusions. It essentially gives a very good estimate of the average intensity along each line even for regions of non-uniform intensities.

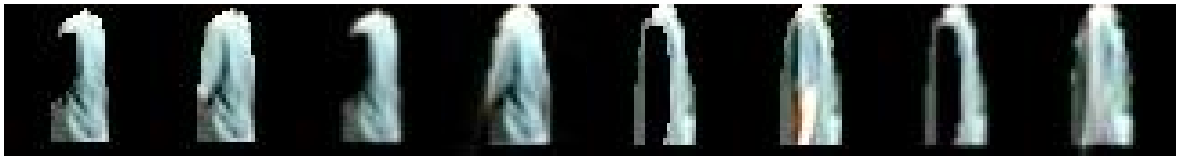


Figure 2.11: *Two illustrations of partially occluded human torsos as examples of when the contour changes but the convex hull remains similar. Images 1 to 4 contain a partially occluded torso, the ground truth appearance inside the convex hull containing the torso, the reconstructed appearance from the RT in image 1, and the reconstructed appearance by using filtered back projection from the average intensity times the RT of the convex hull as in (2.12). Images 5 to 8, show another set of illustration as in 1 to 4. Note in image 6, the ground truth image has outliers because the arm occludes the torso.*

2.3.3 Multiresolution GeT (MRGeT)

The resolution problem becomes a primary concern when using explicit model based methods, because it is not reliable to impose point-to-point correspondences. Here we propose a multiresolution geometric transform (MRGeT) that can deal with noisy observations and inexact contour extraction at the proper scale space, as well as properly combine the appearance and shape information. Specifically, we can change the $\chi(.)$ in (2.6) into the following kernel function:

$$\delta_\epsilon(x) = \frac{1}{\sqrt{2\pi\epsilon^2}} \exp(-\frac{x^2}{2\epsilon^2}),$$

where ϵ determines the resolution of the kernel. Note $\lim_{\epsilon \to 0} \delta_\epsilon(x) = \delta(x)$. If x is the distance of the point from the geometric set S, then by replacing $\chi(.)$ with δ_ϵ in (2.6) will produce a weighted integral in the neighborhood of the set S. Such a kernel is also denoted as $\chi_S^\epsilon(x)$. In the case of basic RT, δ_ϵ corresponds to a line spread function and ϵ determines the width of the spread.

The multiresolution representation can be used to combine the shape and appearance information. Consider introducing a kernel to the functional defined in (2.10). The key is to use different resolution parameters in the numerator and the denominator. For the basic geometric set of straight lines, we have

$$\boldsymbol{R}(\theta, p) = \frac{\int f(x,y)\delta_{\epsilon_1}(x\cos(\theta) + y\sin(\theta) - p)dxdy}{\int H(f(x,y))\delta_{\epsilon_2}(x\cos(\theta) + y\sin(\theta) - p)dxdy}. \tag{2.13}$$

By changing ϵ_1 and ϵ_2, we achieve different representations for various purposes.

- When $\epsilon_1 \to 0$, $\epsilon_2 \to 0$, $\boldsymbol{R}(\theta, p)$ corresponds to the average intensity over a straight line, as discussed in section 2.3.2.

- When $\epsilon_1 \to 0$, $\epsilon_2 \to +\infty$, the denominator will be almost constant. So $\boldsymbol{R}(\theta, p)$ will be a scaled RT of $f(x, y)$. It can be fully reconstructed using filtered backprojection.

- Other combinations of ϵ_1 and ϵ_2 will give combined representations of shape and appearance at different resolutions. ϵ_1 adjusts the resolution of the appearance. ϵ_2 depends on if we need to model the appearance in the actual shape or its convex hull. Using a bigger ϵ_2 allows a more accurate description

of the shape, since $\boldsymbol{R}(\theta, p)$ is closer to a scaled RT. Using a smaller ϵ_2 will make $\boldsymbol{R}(\theta, p)$ closer to the average intensity along the line, thus modeling the appearance inside the convex hull. It will help to handle occlusions that do not change the convex hull of the shape. When the shape is closer to its convex hull, ϵ_2 can be bigger so that the GeT is close to being fully reconstructible.

Another nice property of $\boldsymbol{R}(\theta, p)$ is that it still carries properties of a basic RT with respect to the similarity transform. Suppose $\tilde{f}(x, y) = f(T(x, y))$, where

$$T(x, y) = s \begin{bmatrix} \cos(\alpha) & \sin(\alpha) \\ -\sin(\alpha) & \cos(\alpha) \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix},$$

then the transform of $\tilde{f}$ can be easily shown as

$$\tilde{\boldsymbol{R}}(\theta, p) = \boldsymbol{R}(\theta - \alpha, t_x \cos(\theta - \alpha) + t_y \sin(\theta - \alpha) + sp). \tag{2.14}$$

Following (2.14), the registration with respect to the similarity transform can be easily obtained. If we register two contours by first aligning their centroids and then scale them according to the ratio of area, the only unknown in (2.14) is α, which simply corresponds to a translation in the transform domain. Thus it is very easy to match the appearances inside two contours when they are related by a 2D similarity transform.

Chapter 3

Modeling objects with articulated motion

3.1 Human Identification

Chapter 2 mainly dealt with the theoretical basis of GeT. In Chapter 3, we apply all the proposed GeTs to real world scenarios. We show how GeT can be used as a generic tool for modeling the appearance inside various contours.

In this section, GeTs are designed to incorporate geometric context into appearance modeling for objects with articulated motion, bending, and local deformations. The identity of objects is linked to the representation of appearances in the transform domain. The appearances of human and body parts provide very good examples for our study. We use them to illustrate all the methods introduced in the last chapter. It is useful to model the appearance of a human because sometimes the appearance is more reliable than gait, for example, in the application of persistent tracking.

In this section appearance matching is set in a still-image-to-still-image framework. Segmented body parts are assumed to be given, and we test each type of GeT by designing them for body parts recognition and combined human identification. We show that GeT based on shape matching without parts information gives superior recognition results compared to rigid template matching with body parts segmentation.

Figure 3.1: *Sample of USF database from 3 classes. Walking pedestrians with manually segmented body parts. The first image for each class is in the gallery. The second image is in the probe set.*

We compare three approaches with the same setting using the USF database [42] as in Fig. 3.1, where the body parts have been manually segmented and the size of each image is around 125×72. The first two use the part information while the third one does not. *Approach I*: we design GeT for each body part and study the matching of the exact appearances of parts as well as combined recognition of humans. The task of matching the exact appearances of parts is still not easy because of low resolution, poor quality imagery and errors in segmentation. We apply different transforms for each body part according to their motions and possible occlusions. Because the right arms and right legs are often occluded in the dataset, we did not match these parts for recognition purposes. *Approach II*: as a comparison, we directly match each part using rigid templates, then combine them for human identification. *Approach III*: without using the information of manually segmented parts, we apply GeT based on shape matching to deal with articulation, as discussed in Section 2.2.2.

The experimental setting is as follows. There are 71 classes in this dataset. For each class, we have one image in the gallery and 28 in the probe set that are taken

under different conditions. We classify the probe image according to its distance to the gallery image either in the transform domain as in approach I and III, or in the pixel domain as in approach II. For approach I, each part is represented using the designed GeTs before distances are calculated. The properties of MRGeT in (2.14) are used to align the two parts.

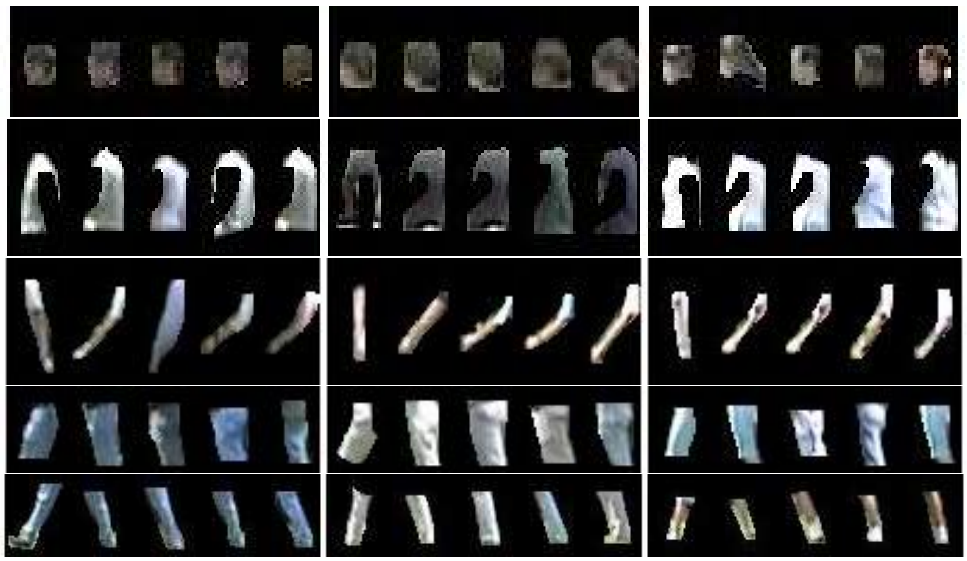


Figure 3.2: *Sample results for matching body parts using GeT. Probe images from three classes are illustrated, corresponding to subjects in Fig. 3.1. Here each class has five images for one part. The first image is the probe image. The second image is the correct match in the gallery using GeT for parts. The next three images show the top 3 matches in the gallery. The ranks of the correct match for each class and each part are: from top to bottom, 2,1,58 for head. 3,1,1 for torso, 18,11,1 for the left arm, 5,16,3 for the left upper leg and 2,5,11 for the left lower leg. The ranks of the correct match of human by combining parts are 1,1,4 for approach I, and 6,13,10 for approach II.*

For approach I, our choice of GeT is as follows.

- *Head:* Use MRGeT in (2.13) with $\epsilon_1 = 2$, $\epsilon_2 = 4$. Choose $\epsilon_2 = 4$ because the actual shape is close to its convex hull.

- *Torso:* Since occlusion needs to be considered while the convex hull of the shape does not change much, we use MRGeT in (2.13) with $\epsilon_1 = 2$, $\epsilon_2 = 2$ so that the transform is close to the average intensity along the line.

- *Left Arm:* We use the modified GeT based on level set as discussed in section 2.2.1 and illustrated in Fig. 2.5.

- *Left Upper Leg:* Mainly 2D rigid motion, choose $\epsilon_1 = 2$ and $\epsilon_2 = 3$ in (2.13). Small ϵ_2 can allow a certain degree of occlusion.

- *Left Lower Leg:* Mainly 2D rigid motion, choose $\epsilon_1 = 2$ and $\epsilon_2 = 4$.

For approach II, each part is matched using the sum of squared distances by only allowing rigid transformations. Note for the torso region, we use the appearance inside the convex hull to reduce the effect of occlusion. For the above two methods, the distances of each part are normalized to a standard log-normal distribution as illustrated in Fig. 3.3(c). For combined recognition, we classify the probe image to the class that has the least weighted distance. The results for matching each body part as well as their heuristically chosen weights are shown in Fig. 3.2 and Table 3.1. The part numbering is in the order shown in Fig. 3.2. Figures 3.3(a)(b)(d) show the CMC curves for matching each body part and the combined recognition results.

For approach III, the image in the gallery set is transformed to the normalized appearance at all six poses using the GeT described in Section 2.2.2. The image in the probe set is transformed to the normalized appearance at its closest pose and the corresponding *mirror* pose. By the *mirror* pose, we mean two similar silhouettes with different topology of parts, such as pose 1 and 4 in Fig. 2.6(a). This helps to obtain more robust matching results. Then we match the transformed image with the normalized gallery image at the same pose and choose the closest match. This way we accomplish human identification without part segmentation.

Part No.	1	2	3	4	5	All
GeT	45.3	65.6	27.0	31.1	26.8	87.9
Templates	36.4	52.4	21.6	20.4	26.2	64.9
GeT(no parts)	-	-	-	-	-	69.0
weights	0.2	0.4	0.13	0.13	0.13	

Table 3.1: *Top One Recognition Rate (%)*

As we can see, approach I outperforms approach II for each body part and both I and III do better than II for combined recognition of humans. Comparing approach I and II for part recognition, we observe that GeT gives 10% higher recognition rates for parts 2 and 4. It is mainly due to the ability of GeT for handling occlusions. Overall, matching the exact appearances of parts is a difficult task, as we see in Fig. 3.2. Part 3 usually contains very few pixels and is very blurred, thus the contour-driven GeT only gets 6% higher rate than rigid templates. For part 5, the GeT method is only slightly higher, because part 5 displays mostly 2D rigid motion

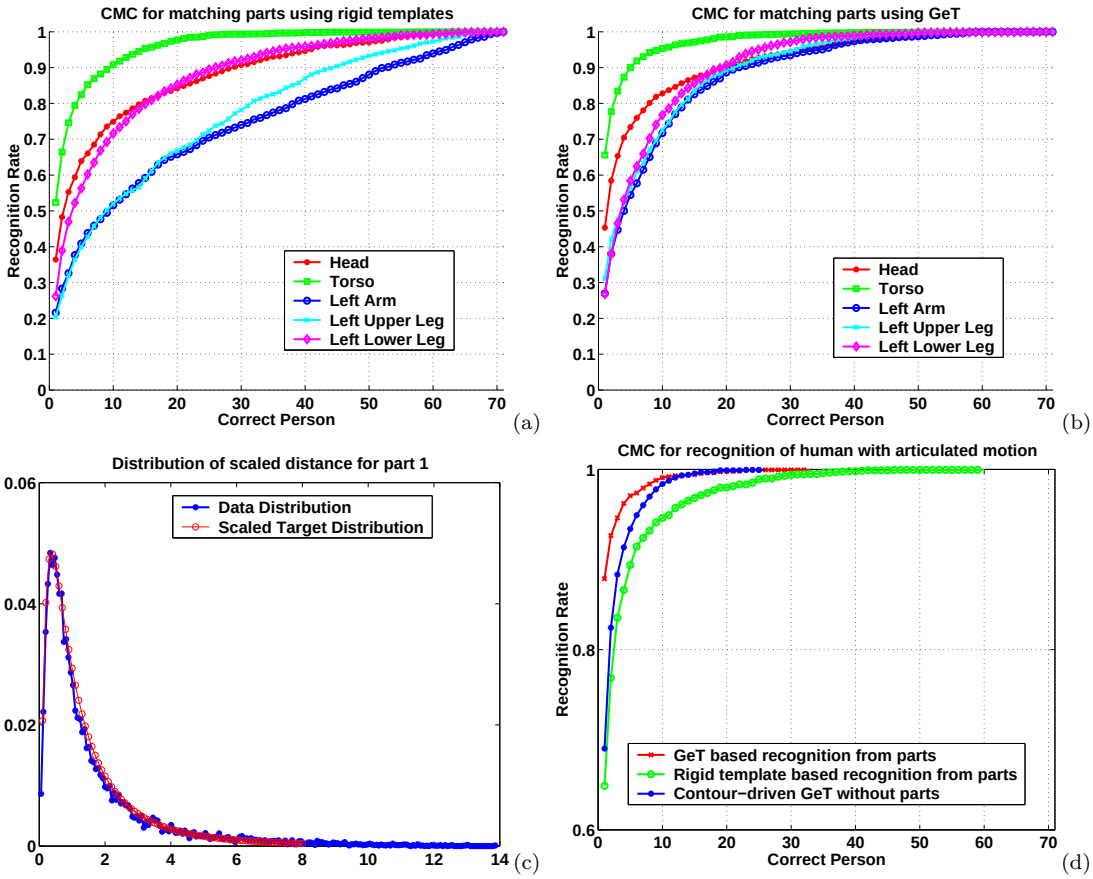


Figure 3.3: *(a)(b) Cumulative matching curves (CMC) of matching parts for two methods. (c) Illustration of normalizing the distance for part 1(head). We observe that the distance has similar distribution as a log-normal distribution with the shape and scale parameters σ,m. The data is normalized to have $m = \sigma = 1$ and shown along with the scaled target distribution. (d) Combined recognition rate of human appearance for all three cases.*

with no occlusion. For overall recognition, despite the non-rigid motion that the probe images have with respect to the gallery images, GeT's top one recognition with part information is as high as 87.9%, while the contour-driven GeT without part segmentation gives 69.0%. The superior performance of approach III over II shows that the designed GeT handles the articulation better even though approach II uses the part information.

3.2 Body Part Segmentation and Video Retrieval

In the last section, we showed that the GeT based on shape matching gives better recognition results even without using any information on body parts. In this section, we apply the GeT to video-to-video setting without prior segmentation of body parts. We will follow the framework proposed in Fig. 2.7 for automatic body parts segmentation, part-based human identification, and surveillance video retrieval using the Honeywell database. First, this section gives a complete illustration of the framework in Fig. 2.7, showing how to use GeT based on shape matching for modeling objects with articulated motion. It starts with a training phase that learns about shape space of typical poses. Then based on the shape space constructed for each pose, the noisy human silhouette due to background subtraction errors can be smoothed. Second, this section shows how to use GeT for automatic body part segmentation. Once we have the segmentation, color features for each body part can be extracted and used for human identification and surveillance video retrieval. That helps to solve difficult cases when the pedestrian changes only part of clothes such as putting on a jacket. Third, as an extensive discussion, this section gives examples of how to use the skeleton based GeT for synthesis of human appearance at arbitrary poses.

In the Honeywell database, there are 30 subjects in one camera where they walk along similar paths as illustrated in Fig. 3.4. People change part of their clothing in different videos, so that there are two different kinds of identity: clothing identity ID1 and person identity ID2. ID1 requires the same person with the same clothing.

ID2 requires the same person but allows different clothing. Each subject has one or two non-overlapping short sequences of about 21 frames each. In total, there are 54 short sequences. Based on ID1, there are 30 classes and based on ID2, there are only 9 classes. Our goal is to match these short sequences. Given one short video, we try to retrieve similar videos from the database according to both ID1 and ID2. A system that automatically segments the body part and extracts the signature of its appearance is used to accomplish this goal.

Figure 3.4: *Honeywell database along with background subtraction results. It contains 54 short sequences. There are 30 classes (each considered as a subject) based on ID1 and 9 classes based on ID2. From left to right, except for subjects one and six, the neighboring four subjects are the same person, i.e., having the same ID2. For example, subjects two to five are the same person with different clothing.*

Compared with the USF database, this set is more difficult because of noisy background subtraction and the absence of good canonical templates of typical poses like the one in Fig. 2.6(a). But the key assumption for the algorithm in Fig. 2.7 still holds: each body part has similar topology for different people with the same pose.

The outline of our method is as follows. Step (a) in Fig. 2.7 is carried out to construct a normalized shape space, followed by step (b) of manual part segmentation of the mean shapes. These steps form the training phases of our algorithm. Following the ideas in steps (c) and (d), when a query sequence comes in, first these frames are temporally aligned with the mean shapes, then the silhouette is normalized and projected onto the shape space to be smoothed. The smooth silhouette is again matched with the mean shapes and GeT based on this matching is used to produce parts segmentation. Finally, since each part has many samples from all the frames, instead of the small region in a single image as in section 3.1, the color features can be more reliably extracted. Video retrieval and human identification are done based on these features. Implementation details of each step in Fig. 2.7 are given below.

3.2.1 Shape Space Construction

Because the silhouettes are very noisy, the left and right boundaries of the mask along each row are used for shape representation, as illustrated in Fig. 3.5. One video sequence is taken from each subject for training purposes. Step (a1) of Fig. 2.7 is implemented through the temporal alignment of these 30 training sequences. The alignment takes two steps. First, the period is estimated through matching the shapes within one sequence, then the length of each sequence is cut to one period. Second, dynamic time warping (DTW) is used to find the best sequence alignment. Both methods have been well studied in the gait recognition literature

[53]. Subject ten is selected as the reference. The alignment results with respect to the reference are shown in Fig. 3.5.

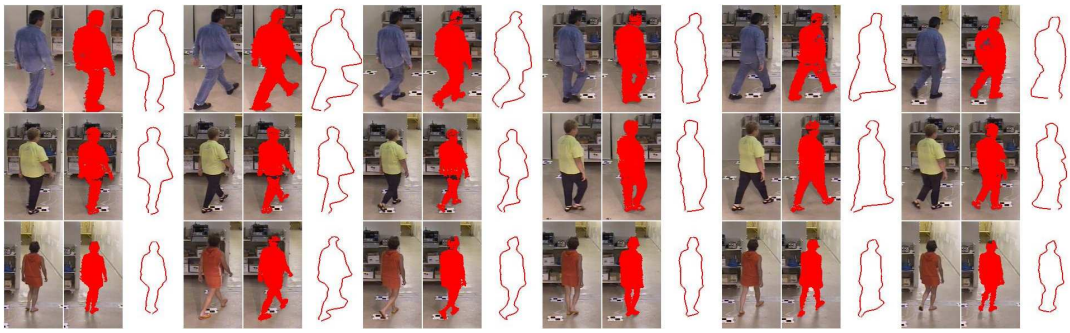


Figure 3.5: *Each sequence is temporally aligned with respect to subject ten using dynamic time warping. Illustrated here are six typical poses along with background subtraction results and contour representations using row boundaries.*

In step (a2), all the binary masks corresponding to the same pose are *normalized* using the GeT based on shape matching between the contours of each subject and subject ten. In other words, the normalization step applies $\tilde{\boldsymbol{R}}_{\Gamma_p^s \Gamma_p^{10}}$ to binary mask $\chi_{\Gamma_p^s}$, where s indexes the subject and p indexes the pose. Note the transform is denoted as $\tilde{\boldsymbol{R}}$ because the matching between Γ_p^s and Γ_p^{10} is based on their row boundaries. This normalization, as illustrated in Fig. 3.6(a), accounts for the variation in sizes, and it essentially reduces the factor of non-linear deformations between the shape of different people so that a linear learning method such as the principal component analysis (PCA) can be applied .

In step (a3), six typical poses are selected and the PCA is used to construct a shape space for each pose in the *normalized* domain. In Fig. 3.6(b), the mean shapes found by thresholding the average masks are illustrated. For each pose,

thirty normalized masks, one from each subject is used to learn the shape space using PCA. Since all these masks are very noisy, we only keep one eigen-vector and the mean shape in each constructed shape space. Though this construction is limited, but with the noisy data available, it is a fair approximation which can generate fairly smooth masks when noisy masks are projected into this space, and is better than using merely mean shapes since it leaves room for the projected mask to have customized shapes.

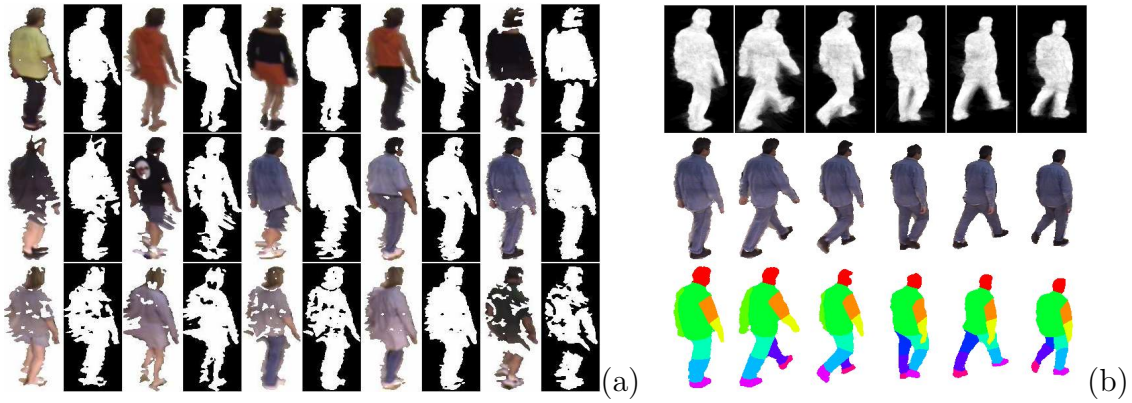


Figure 3.6: *Illustration of how to construct the shape space. (a) For pose one, all the appearances and masks in the training data are 'normalized' using the GeT based on shape matching. These masks are used to construct a shape space. (b) Top row: average masks for each typical poses. Middle row: appearance inside the mean shapes, which is obtained by thresholding the first row. Bottom row: manual part segmentation for the mean shapes. It contains eleven parts as listed in Table 3.2.*

3.2.2 Body Part Segmentation

Once the mean shape of each pose denoted as $\bar{\Gamma}_p$ is found, they can be manually segmented into part γ_p^q as shown in Fig. 3.6(b) and described in step (b) of Fig. 2.7. q is the index for each part. The one-time manual segmentation result is used as the reference for the automatic segmentation below.

Now for the probing phase, given a short walking sequence, the following procedures are carried out. The shape matching algorithm used in this section is based on the standard matching technique using shape context [3].

- Following step (c), align the new sequence with the mean shape through DTW for pose classification. The pose p for each frame f is determined.

- Normalize the noisy masks and project into the shape space of the corresponding pose to obtain a smooth silhouette. The normalization uses $\tilde{\boldsymbol{R}}_{\Gamma_p^f \bar{\Gamma}_p}(.)$. The normalized masks are projected onto the eigen-shapes to obtain the smooth silhouette $\tilde{\Gamma}_p^f$. The normalized appearance $\tilde{A}_p^f$ inside the smooth silhouette $\tilde{\Gamma}_p^f$ is also obtained and illustrated in Fig. 3.7(a).

- Following step (d), based on the shape matching between the mean shape $\bar{\Gamma}$ and the smooth mask $\tilde{\Gamma}$, part segmentation can be obtained. That is, apply $\boldsymbol{R}_{\bar{\Gamma}_p \tilde{\Gamma}_p^f}$ to $\chi_{\gamma_p^q}$ to obtain the part segmentation for $\tilde{\Gamma}_p^f$. Note this transform $\boldsymbol{R}$ is through the matching of exact contours bounding $\bar{\Gamma}_p$ and $\tilde{\Gamma}_p^f$, unlike that of row boundaries in $\tilde{\boldsymbol{R}}$ above.

- Color features of each part are obtained and used as the signature for matching

during video retrieval and human identification tests. All the pixels of each body part across time are counted. In Fig. 3.7(b), the results of part segmentations and two dominant colors for each body part are illustrated. The two dominant colors are the two means of a Gaussian mixture model. In the matching test, the color histogram is used for a more accurate description of color distribution.

Figure 3.7: *Normalized appearance of pedestrians along with parts segmentation and appearance signature extraction. (a) The original image, background subtraction, and 'normalized' appearance* $\tilde{A}_p^f$ *in the smooth silhouette. (b) Each column of the original image is followed by two columns showing the part segmentation for the smooth silhouette marked with the two dominant colors for each body part estimated through a Gaussian mixture model.*

As seen in Fig. 3.7(b), the segmentation and the color signature are pretty accurate. The video retrieval tests based on the color histogram of each body parts are shown in Fig. 3.8. In the tests, the query video is matched with the remaining 53 sequences and the top three matches based on the color of each body part are illustrated. The Bhattacharayya distance [4] is used as the measure between his-

tograms. The part-based retrieval gives very interesting results. For example, the leg based retrieval finds other subject wearing jeans of similar colors. The changes in clothing for the lower body do not affect the retrieval results based on the torso or the arm.

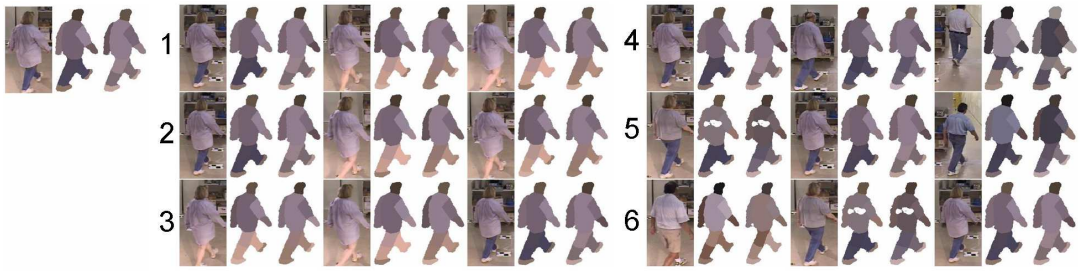


Figure 3.8: *Illustration of retrieval results based on the color of each body part. The triplet of images in the first column includes a sample image from the query video along with the image showing two dominant colors of each body part. The other two columns show the top three retrieval results based on each body part in the following order: 1. head, 2. right lower arm, 3. torso, 4. right upper leg, 5. right lower leg, and 6. right shoe.*

To give some quantitative analysis of our method, two experiments are carried out. Experiment I uses one sequence from each subject as the gallery. Note that these thirty sequences are selected from the same walking videos used for shape space construction, but each sequence is only partially overlapping with the selected sequences for shape learning. The remaining twenty-four sequences make up the probe sets. The correct match is according to ID1, so there are thirty classes in total. Experiment II matches each sequence with other fifty-three sequences. The correct match is decided using ID2, so there are nine classes and fifty-four test

data. The matching results for each body part and the combined recognition rates are shown in Table 3.2. The weights for the combined recognition are heuristically chosen, based roughly on the area of each body part.

Part No.	1	2	3	4	5	6	7	8	9	10	11	All
A	75.0	70.8	79.2	62.5	87.5	66.7	50.0	62.5	58.3	45.8	62.5	95.8
B	87.5	100.0	87.5	83.3	100.0	75.0	62.5	79.2	79.2	70.8	75.0	100.0
C	87.5	100.0	91.7	83.3	100.0	83.3	79.2	83.3	79.2	75.0	79.2	100.0
D	100.0	96.3	85.2	87.0	98.1	81.5	79.6	75.9	85.2	83.3	77.8	96.3
E	100.0	96.3	87.0	94.4	100.0	88.9	90.7	88.9	90.7	87.0	88.9	98.1
F	100.0	98.1	90.7	94.4	100.0	94.4	94.4	92.6	92.6	88.9	92.6	100.0
G	85.2	72.2	51.9	61.1	66.7	59.3	48.1	55.6	55.6	55.6	44.4	66.7
H	79.6	38.9	18.5	31.5	42.6	31.5	22.2	33.3	27.8	31.5	25.9	42.6
I	0.161	0.081	0.081	0.048	0.161	0.081	0.161	0.048	0.048	0.065	0.065	

Table 3.2: *Recognition rates for the Honeywell database. The order of the column is: 1. head, 2. right upper arm, 3. right lower arm, 4. left arm, 5. torso, 6. upper right leg, 7. lower right leg, 8. upper left leg, 9. lower left leg, 10. right foot, 11. left foot, and 12. all parts combined. Each row represents the following. A-C: top one, two, three recognition rates for Exp I. D-F: top one, two, three recognition rates for Exp II. The above rates show the percentage of query videos whose correct match is in the top n matches. G-H show the percentage that all top two or top three matches are correct ones in Exp II. I gives the heuristically chosen weight of each part for combined recognition.*

For experiment I, we observe that torso, the right lower arm, head and the right upper arm give the best recognition results, meaning they are reliable cues to identify the same person with the same dress. For experiment II, head shows the

highest rate, because subjects with the same ID2 have the same hair color, making it the most reliable cue to determine ID2. Torso and the right upper arm regions also give very high rates. Since there may be more than one sample belonging to the same class in the gallery during queries, the rates that all top two or top three matches belong to the correct class are also listed. The combined top one recognition rates are 95.8% and 96.3% respectively.

The role of GeT in the algorithm is very important. The first transform $\tilde{\boldsymbol{R}}$ reduces the effect of different sizes, and makes sure that the appearance of a pedestrian can be compared at the same scale. The normalization also helps to build a linear shape space. The second transform $\boldsymbol{R}$ is used to infer body parts segmentation. This idea can be used broadly in other segmentation methods.

3.2.3 Synthesis

In section 2.3.1, we introduce a way of generating point sets from the skeletons. Here we apply the corresponding GeT to image synthesis.

Figure 3.9: *Images 1 and 2: The original image and its skeleton. Images 3,4 and 5: Synthesis results, the ground truth and the skeleton. Next 3 images: Another set of synthetic imagery.*

For this experiment, we assume that the corresponding skeletons across frames are already found. Then we use the skeleton based GeT proposed in section 2.3.1 for image synthesis. The appearance of a human with articulated motion in subsequent frames can be generated from the GeT of the appearance in the first frame. The results are shown in Fig. 3.9 together with the ground truth, following the methods illustrated in Fig. 2.9.

We observe that the curve based GeT can handle articulated motion with large nonlinear deformation and occlusions. Also in many cases feature curves are easier to track or can be generated from the contour boundary. Then a curve based GeT is more desirable for image synthesis. Although synthetic images have artifacts due to out-of-plane rotation etc., our method uses a very simple model and can be further developed for image based rendering.

3.2.4 Conclusion

In summary, a general definition of geometric transform is given to unify Radon transform, trace transform and image warping. We show how to design each element of GeT, particularly the geometric set and functional, to incorporate geometric context into appearance modeling. GeT is shown to be useful in a broad range of applications. Future work includes further exploration of contour driven GeT and designing GeT for multi-view sequences.

Chapter 4

Structure from Planar Motion

4.1 Planar Factorization

In Chapter 4, we study the characteristics of vehicles in surveillance videos. It is crucial to study the 3D geometry of planar motion in perspective cameras to understand the correspondence problem for vehicles. A very efficient planar factorization method is proposed and implemented in a complete automatic vehicle reconstruction system.

This section gives the derivation of standard planar factorization. We start with the importance of studying planar motion and present a literature review of factorization approach. Then we present two key observations for planar motion which enable a simple mathematical formulation. The formed measurement matrix has lower rank than general motion, and it is a non-linear function of the observed feature point coordinates. We obtain the 3D structure of the vehicle through applying SVD over this measurement matrix. Our method is compared with methods for general motion and quantitative and qualitative evaluations of our methods are provided.

4.1.1 Background

Structure from planar motion is motivated by applications such as parking lot surveillance and traffic scene analysis. Planar motion is ubiquitous in surveillance videos, simply because most objects can be assumed to move on the typically planar ground. However, an approach that fully exploits the constraints of motion on the ground plane has not been reported. Planar motion is also quite often confused with the motion of a planar object. Throughout this thesis, *planar motion* means that the motion of a 3D object is constrained to lie on a plane. In a later section, we clarify the differences and elaborate on a dual relationship between planar motion and planar object. In this dissertation, we focus on a monocular sequence captured by a stationary perspective camera, in which a rigid object moves on the ground plane.

Among all structure from motion (SfM) methods, the factorization approach [51] has been very popular because as a batch processing method, the reconstruction can be easily carried out through singular value decomposition (SVD). Generally, the observation matrix is factorized as a bilinear product of motion and shape matrices. SVD is then used to find the factorized components, at the same time denoising the data. Our goal is to find a factorization method specialized for planar motion.

The factorization approach for SfM has been studied extensively in the last decade [23]. The essence of factorization approach lies in finding the right rank constraint, which corresponds to the lowest rank among all the factor matrices. The main goal is to find a lower rank condition by exploiting the property of certain

camera model, shape or motion [1]. Rank constraints have been found for different camera models, or different types of objects such as planar objects, rigid, and non-rigid objects. Another branch of study has focused on the rank constraints for certain kinds of motion, such as linear and planar motions.

The study of rank constraints for certain types of motion can be found in [15] [39] [44] [55] etc. In [15], Han et al. study the case of linear motion with constant speed for multiple objects, and develop a method of scene reconstruction through factorization. In [44], Quan et al. proposed a method for decomposing a 2D image into two 1D images, so that the structure can be reconstructed from images captured under constrained planar motion, but they restrict the image plane to be perpendicular to the motion plane. In [39] and [55], matrices formed by displacements of feature points are shown to have a lower rank condition under constrained motion. Then iterative estimation of motion and depths is carried out. Although these methods can be used to recover the motion plane, they do not support a direct factorization. We propose a generic factorization method for structure from planar motion under perspective projection. A measurement matrix specialized for planar motion is formed in order to lower the rank condition. We exploit the constraint of planar motion to find a simple scaling method instead of using the fundamental matrix and epipoles as in [47], where the authors propose a factorization approach for a perspective camera.

Other methods for structure from planar motion (SfPM) such as [39][49][55] have been proposed. Many of them need non-linear optimizations or iterative schemes and sometimes cannot guarantee a global optimal solution. Our method

only requires SVD and linear operations with no iterations. Our formulation is similar to [49], where the authors consider the reconstruction of perspective depths from correspondences across only two frames.

In our method, the camera does not have to be calibrated. For an uncalibrated sequence, our method requires the estimation of the focal length and ground plane constraints (GPC), so that the motion plane is known. We use vanishing points and lines [16][33][56] to find the GPC.

In this dissertation, *planar factorization* refers to the factorization method for structure from planar motion. The rest of Chapter 4 is organized as follows. Section 4.1.2 gives the derivation of the standard planar factorization, followed by a comparison with factorization for general motion and both quantitative and qualitative analysis of the algorithm. Since the measurement matrix is formed differently, the algorithm has different properties. The detailed theoretical analysis is given in section 4.2 along with a method for handling directional uncertainties in observations. Section 4.3 clarifies the difference between planar object and planar motion and explains their dual relationship. Section 4.4 shows how to use our method to build a fully automated vehicle reconstruction system. Section 4.4.2 concludes Chapter 4.

4.1.2 Standard planar factorization

In this section, we derive the factorization method for structure from planar motion and present the detailed algorithm. Then we compare our method with the factorization method for general motion under a perspective projection.

We do not assume that the camera is calibrated. The calibration and ground plane constraints can be estimated through a number of ways. In the next section, we show a simple semi-automatic calibration scheme using vanishing points and lines. Other automated calibration methods can also be applied here. Once the calibration and GPC is estimated, a matrix is constructed from the observations, which after properly scaling has a rank of at most 3.

4.1.2.1 Derivation of our method

Consider the selection of the camera coordinate system (CCS) and the world coordinate system (WCS) shown in Figure 4.1, where the x-y plane in WCS lies on the ground plane. We focus on the typical case that the image plane is not parallel to the ground plane.

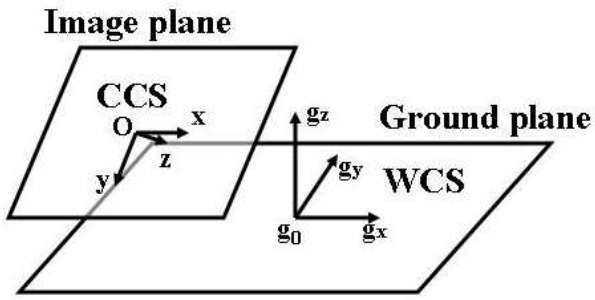


Figure 4.1: *Selection of world coordinate and camera coordinate systems. O is the camera center.*

In CCS, denote the direction of x- y- and z-axis of WCS as $\mathbf{g}_x$, $\mathbf{g}_y$ and $\mathbf{g}_z$ respectively, and the origin of WCS as $\mathbf{g}_0$. f is the focal length. As mentioned above, all of these vectors can be estimated.

Suppose a point p is the image of a point P in the 3D space. Its representation is $p_c = (x, y, f)^T$ in the CCS. Then $P_c = \lambda(x, y, f)^T$, where λ is the perspective depth. Its representation in WCS is,

$$P_w = \begin{pmatrix} \mathbf{g}_x^T \\ \mathbf{g}_y^T \\ \mathbf{g}_z^T \end{pmatrix} (\lambda \begin{pmatrix} x \\ y \\ f \end{pmatrix} - g_0), \tag{4.1}$$

and we denote $\begin{bmatrix} \mathbf{g}_x & \mathbf{g}_y & \mathbf{g}_z \end{bmatrix} = \begin{bmatrix} \rho_1^T \\ \rho_2^T \\ \rho_3^T \end{bmatrix}$ for convenience.

According to our assumption, the only unknown in the above equation is λ. Here are some key observations about the factorization method. If the feature points are from a rigid object moving on the ground plane, then each of them will move on a plane parallel to the ground plane, and hence their z-coordinates in WCS will remain constant across all frames. Because of rigid motion, its x- and y-coordinates in WCS will take a 2D Euclidean Transform.

Note in [49], similar constraints are used for only two frames. In [49], quadratic equations of perspective depths are formed using two facts: the displacement vectors between corresponding 3D feature points at two time instants are parallel to the ground plane, and the 3D distance between any pair of feature points at each time instant does not change over time because of rigidity. But since they only consider a pair of frames and there is no denoising process, the coefficients of the equations are very sensitive to noise, yielding solutions that are often unstable and sometimes

even imaginary. As shown below, our method makes use of multiple frames and the factorization approach, and therefore, is more robust to noise and incorrect calibration.

Suppose we have an image sequence from a stationary perspective camera, in which a rigid object moves on the ground. Assume that N visible image points lie on the object, such as a vehicle being tracked over M frames. We will use factorization to estimate the structure of the vehicle.

Let λ_{ti} be the perspective depth of point i in frame t. Then from (4.1), all points should satisfy

$$P_w^{ti} = \begin{bmatrix} \mathbf{g}_x^T \\ \mathbf{g}_y^T \\ \mathbf{g}_z^T \end{bmatrix} (\lambda_{ti} \begin{bmatrix} x_{ti} \\ y_{ti} \\ f \end{bmatrix} - g_0) \equiv \lambda_{ti} \begin{bmatrix} u_{ti} \\ v_{ti} \\ w_{ti} \end{bmatrix} - \mathbf{s} \tag{4.2}$$

$\mathbf{s}$ should be the same for all points across all frames. Then if we form a matrix from the x and y components of P_w^{ti}, and using the fact that they take a 2D Euclidean transform, we will have

$$\boldsymbol{W} \text{ as } \begin{bmatrix} \lambda_{11}u_{11} & \lambda_{12}u_{12} & ... & \lambda_{1N}u_{1N} \\ \lambda_{11}v_{11} & \lambda_{12}v_{12} & ... & \lambda_{1N}v_{1N} \\ ... & ... & & ... \\ ... & ... & & ... \\ \lambda_{M1}u_{M1} & \lambda_{M2}u_{M2} & ... & \lambda_{MN}u_{MN} \\ \lambda_{M1}v_{M1} & \lambda_{M2}v_{M2} & ... & \lambda_{MN}v_{MN} \end{bmatrix}$$

$$= \begin{bmatrix} r_{11}^1 & r_{12}^1 & t_x^1 \\ r_{21}^1 & r_{22}^1 & t_y^1 \\ ... & & \\ r_{11}^M & r_{12}^M & t_x^M \\ r_{21}^M & r_{22}^M & t_y^M \end{bmatrix} \begin{bmatrix} x_1 & x_2 & ... & x_N \\ y_1 & y_2 & ... & y_N \\ 1 & 1 & ... & 1 \end{bmatrix} \tag{4.3}$$

where $\boldsymbol{W}$ is the rescaled observation matrix. $\boldsymbol{R}_t = \begin{bmatrix} r_{11}^t & r_{12}^t \\ r_{21}^t & r_{22}^t \end{bmatrix}$ is an orthogonal matrix and corresponds to the rotation matrix for frame t. $\begin{bmatrix} t_x^k & t_y^k \end{bmatrix}^T$ corresponds to the translation vector for frame k. $\begin{bmatrix} x_i & y_i \end{bmatrix}^T$ are the x-y coordinates of point i, but because of the ambiguity in selecting the reference coordinates for motion and structure, they are not necessarily defined as WCS. Also note that the contribution from **s** is absorbed into $\begin{bmatrix} t_x^k & t_y^k \end{bmatrix}^T$.

Therefore the rescaled matrix $\boldsymbol{W}$ will have a rank of at most 3. To find the scale, we use the fact that the z-component of each feature point in WCS remains constant across all frames, i.e. $z(P_w^{ti}) = z(P_w^{si})$ for $s = 1, ..., M$. Then we can obtain $\lambda_{ti} w_{ti} = \lambda_{si} w_{si}$. Thus the ratio of λs can be recovered along each column in $\boldsymbol{W}$.

Based on this equality, we can set $\lambda_{ti} w_{ti} = c_i$, for $t = 1, ..., M$, and $i = 1, ..., N$. Typically $w_{ti} \neq 0$ and $c_i \neq 0$ unless the feature point lies on the vanishing line of the ground plane. So we substitute λ_{ti} and move all the unknowns to the right hand side, which results in,

$$\tilde{\boldsymbol{W}} \text{ as } \begin{bmatrix} u_{11}/w_{11} & u_{12}/w_{12} & ... & u_{1N}/w_{1N} \\ v_{11}/w_{11} & v_{12}/w_{12} & ... & v_{1N}/w_{1N} \\ ... & ... & & ... \\ u_{M1}/w_{M1} & u_{M2}/w_{M2} & ... & u_{MN}/w_{MN} \\ v_{M1}/w_{M1} & v_{M2}/w_{M2} & ... & v_{MN}/w_{MN} \end{bmatrix} \tag{4.4}$$

$$= \begin{bmatrix} \alpha_{11} & \alpha_{12} & ... & \alpha_{1N} \\ \beta_{11} & \beta_{12} & ... & \beta_{1N} \\ ... & ... & & ... \\ \alpha_{M1} & \alpha_{M2} & ... & \alpha_{MN} \\ \beta_{M1} & \beta_{M2} & ... & \beta_{MN} \end{bmatrix}$$

$$= \begin{bmatrix} r_{11}^1 & r_{12}^1 & t_x^1 \\ r_{21}^1 & r_{22}^1 & t_y^1 \\ ... & & \\ r_{11}^M & r_{12}^M & t_x^M \\ r_{21}^M & r_{22}^M & t_y^M \end{bmatrix} \begin{bmatrix} x_1/c_1 & x_2/c_2 & ... & x_N/c_N \\ y_1/c_1 & y_2/c_2 & ... & y_N/c_N \\ 1/c_1 & 1/c_2 & ... & 1/c_N \end{bmatrix}$$

$$= \begin{bmatrix} \mathbf{m}_1^T \\ \mathbf{n}_1^T \\ ... \\ \mathbf{m}_M^T \\ \mathbf{n}_M^T \end{bmatrix} \begin{bmatrix} \mathbf{s}_1 & \mathbf{s}_2 & ... & \mathbf{s}_N \end{bmatrix} \tag{4.5}$$

$$\text{as } \tilde{\boldsymbol{M}}\tilde{\boldsymbol{S}} \tag{4.6}$$

where

$$\mathbf{m}_t = \begin{bmatrix} r_{11}^t & r_{12}^t & t_x^t \end{bmatrix}^T, \mathbf{n}_t = \begin{bmatrix} r_{21}^t & r_{22}^t & t_y^t \end{bmatrix}^T,$$

$$\mathbf{s}_i = \begin{bmatrix} x_i/c_i & y_i/c_i & 1/c_i \end{bmatrix}^T, \text{and } \begin{bmatrix} \alpha_{ti} \\ \beta_{ti} \end{bmatrix} = \begin{bmatrix} u_{ti}/w_{ti} \\ v_{ti}/w_{ti} \end{bmatrix}. \tag{4.7}$$

The symbols in (4.7) are not used in this section. But they facilitate discussions in section 4.2.

The matrix $\tilde{\boldsymbol{W}}$ can be directly calculated from observations, calibration parameters, and GPC. It is a bilinear product of motion and shape matrices on the x-y plane in WCS. Its rank is at most 3.

Once we have the rank constraints of an observation matrix, factorization methods [51] can be applied as follows. Because of noise, the rank of $\tilde{\boldsymbol{W}}$ is generally higher. So the estimation of the true observation matrix is through the following optimization

$$argmin_{\hat{\boldsymbol{W}}} || \tilde{\boldsymbol{W}} - \hat{\boldsymbol{W}} ||_F. \tag{4.8}$$

subject to the constraint that the rank of $\hat{\boldsymbol{W}}$ is 3. In (4.8), $||.||_F$ denotes the Frobenius norm, which gives the sum of squared differences between the elements of two matrices. Suppose the SVD of $\tilde{\boldsymbol{W}}$ is $\tilde{\boldsymbol{W}} = \boldsymbol{U} diag(\sigma_1, \sigma_2, ..., \sigma_s) \boldsymbol{V}^T$ and $\sigma_1 \geq \sigma_2 \geq ... \geq \sigma_s$, then the solution to (4.8) is $\hat{\Phi} = \boldsymbol{U}_{2M\times 3} diag(\sigma_1, \sigma_2, \sigma_3) \boldsymbol{V}_{N\times 3}^T$, where only the three largest singular values and singular vectors are retained. So the SVD is applied to find the rank three matrix $\hat{\boldsymbol{W}}$ which is closest to a noisy $\tilde{\boldsymbol{W}}$ in terms of the Frobenius norm. Set $\hat{\boldsymbol{M}} = \boldsymbol{U}_{2M\times 3} diag(\sigma_1^{0.5}, \sigma_3^{0.5}, \sigma_3^{0.5})$ and $\hat{\boldsymbol{S}} = diag(\sigma_1^{0.5}, \sigma_3^{0.5}, \sigma_3^{0.5}) \boldsymbol{V}_{N\times 3}^T$.

Because $\hat{\boldsymbol{W}} = \hat{\boldsymbol{M}}\hat{\boldsymbol{S}} = (\hat{\boldsymbol{M}}\boldsymbol{T})(\boldsymbol{T}^{-1}\hat{\boldsymbol{S}})$ for any 3x3 non-singular matrix $\boldsymbol{T}$, we need additional constraints to eliminate the ambiguity. Rewrite $\boldsymbol{T} = [\boldsymbol{T}_1\, \boldsymbol{T}_2]$, where $\boldsymbol{T}_1$ and $\boldsymbol{T}_2$ are 3x2 and 3x1 matrices respectively. Suppose the correct motion matrix $\boldsymbol{M} = \hat{\boldsymbol{M}}\boldsymbol{T}$, then

$$\hat{\boldsymbol{M}}\boldsymbol{T}_1 = \begin{bmatrix} \boldsymbol{R}_1 \\ \boldsymbol{R}_2 \\ \cdots \\ \boldsymbol{R}_M \end{bmatrix} \tag{4.9}$$

Using the orthogonal property of $\boldsymbol{R}_t$, certain elements along the tri-diagonals in $(\hat{\boldsymbol{M}}\boldsymbol{T}_1)(\hat{\boldsymbol{M}}\boldsymbol{T}_1)^T = \hat{\boldsymbol{M}}(\boldsymbol{T}_1\boldsymbol{T}_1^T)\hat{\boldsymbol{M}}^T$ are known. So the linear least square estimate (LLSE) can be used to recover $\boldsymbol{Q} \equiv \boldsymbol{T}_1\boldsymbol{T}_1^T$. Then $\boldsymbol{T}_1$ can be estimated through SVD of $\boldsymbol{Q}$ and it is unique up to a 2x2 rotational matrix. The ambiguity comes from the selection of the reference coordinate system. If we select the first frame as the reference frame, namely $\boldsymbol{R}_1 = \boldsymbol{I}_{2\times 2}$ and $t_x^1 = t_y^1 = 0$, then $\boldsymbol{T}_1$ can be uniquely recovered and $\boldsymbol{T}_2$ can be recovered up to a scale factor. $\boldsymbol{T}_2$ can be solved using the known elements in $\hat{\boldsymbol{M}}\boldsymbol{T}_2$. The scale factor can be arbitrarily chosen and it is an inherent ambiguity when a monocular sequence is used.

After $\boldsymbol{T}$ is found, the desired shape matrix is $\boldsymbol{S} = \boldsymbol{T}^{-1}\hat{\boldsymbol{S}}$. Using the the last row of $\boldsymbol{S}$, we can easily find the perspective depth of each feature point, thus recovering the 3D structure. It is only recovered up to a scale factor because of the ambiguity in $\boldsymbol{T}_2$.

The algorithm can be summarized as follows,

Standard Planar Factorization:

a. Calculate the matrix $\tilde{\boldsymbol{W}}$ from (4.2) and (4.4), using the observations, estimated calibration, and GPC.

b. Use SVD to get $\tilde{\boldsymbol{W}} = \boldsymbol{U} diag(\sigma_1, \sigma_2, ..., \sigma_s)\, \boldsymbol{V}^T$, where $s = min(2M, N)$. Use the 3 largest singular values and the corresponding singular vectors to get an initial estimate of motion and shape matrices, i.e., set $\hat{\boldsymbol{M}} = \boldsymbol{U}_{2M\times 3} diag(\sigma_1^{0.5}, \sigma_3^{0.5}, \sigma_3^{0.5})$ and $\hat{\boldsymbol{S}} = diag(\sigma_1^{0.5}, \sigma_3^{0.5}, \sigma_3^{0.5})\, \boldsymbol{V}_{N\times 3}^T$.

c. Find $\boldsymbol{T} = [\boldsymbol{T}_1\, \boldsymbol{T}_2]$ and eliminate the ambiguity. First find LLSE of $\boldsymbol{Q} \equiv \boldsymbol{T}_1\, \boldsymbol{T}_1^T$ from known elements in $\hat{\boldsymbol{M}}(\boldsymbol{T}_1\, \boldsymbol{T}_1^T)\hat{\boldsymbol{M}}^T$ according to Eq. (4.9). Set the first frame as the reference frame, then use $\boldsymbol{R}_1 = \boldsymbol{I}_{2\times 2}$ and $t_x^1 = t_y^1 = 0$ to further remove the ambiguity in $\boldsymbol{T}_1$ and find $\boldsymbol{T}_2$ up to a scale. Set an arbitrary scale for $\boldsymbol{T}_2$.

d. Reconstruct the shape and motion matrices as $\boldsymbol{S} = \boldsymbol{T}^{-1}\hat{\boldsymbol{S}}$ and $\boldsymbol{M} = \hat{\boldsymbol{M}}\boldsymbol{T}$ respectively. Then $1/s_{3i} = \lambda_{ti} w_{ti}$ are used to find λ_{ti}, the perspective depth, and hence the structure is reconstructed.

4.1.2.2 Comparisons with the factorization method for general motion

The factorization method was originally developed for orthographic projection [51] and then extended to paraperspective projection [43]. It was generalized for perspective projection in [47], in which the measurement matrix of rank at most 4 was formed. The key factor in factorization under perspective projection lies in

the scaling of the measurement matrix according to perspective depths. In [47], the authors use epipolar constraints to recover the ratio of scalings. However, the estimation of the fundamental matrix and epipoles is not an easy task.

The general method can be used to handle the special case of planar motion. However, we find a simpler and more efficient formulation which is tailored to deal with the constrained case. In summary, our method is different from the general method in the following ways,

1. We form the measurement matrix using only the x-y coordinates in WCS instead of CCS. The measurement matrix is shown to come from points taking a 2D Euclidean transform, which has a rank of at most 3. In [47], the homogeneous coordinates of projections scaled by the depths are gathered together to form a matrix of rank at most 4. So the rank condition is reduced for planar motion.

2. We use the property that the z-component for each feature point is constant in WCS to find the right scaling factor, which is a lot easier than estimating the fundamental matrix and epipoles. However, we do need one-time calibration and estimation of the motion plane, which can be done fairly easily using vanishing points and lines.

3. We have an Euclidean reconstruction instead of a projective reconstruction, because more ambiguities are removed in our formulation. The structure can be recovered up to a scale, while in [47], it is recovered up to a 4x4 non-singular matrix.

4.1.2.3 Experiments

We first explain the simple calibration method we use in our experiments. Then we do quantitative and qualitative analysis of our method by applying it to some real and synthetic sequences.

4.1.2.4 Calibration through vanishing points and lines

Many methods are available for automatic or semi-automatic calibration and recovery of the ground plane constraints. We use vanishing points and lines [16][33][56][58]. We make use of parallel and perpendicular lines, which are very often present in man-made environments.

In a perspective camera model, the images of parallel lines typically will intersect at the vanishing point corresponding to these lines. Geometrically, it is the intersection of the image plane with a ray passing through the camera center and parallel to those lines. Algebraically, the direction of the ray $\mathbf{d}$ and the vanishing point $\mathbf{v}$ is related as $\mathbf{v} = \boldsymbol{K}\mathbf{d}$ [16], where $\boldsymbol{K}$ is the calibration matrix.

The angle between two such rays is,

$$\cos\theta = \frac{\mathbf{v}_1^T(\boldsymbol{K}^{-T}\boldsymbol{K}^{-1})\mathbf{v}_2}{\sqrt{\mathbf{v}_1^T(\boldsymbol{K}^{-T}\boldsymbol{K}^{-1})\mathbf{v}_1}\sqrt{\mathbf{v}_2^T(\boldsymbol{K}^{-T}\boldsymbol{K}^{-1})\mathbf{v}_2}} \tag{4.10}$$

The vanishing line for the ground plane corresponds to the intersection of the image plane with the plane passing through the camera center and parallel to the ground plane. Once calibration is done and the vanishing line is known, the ground plane normal can be calculated.

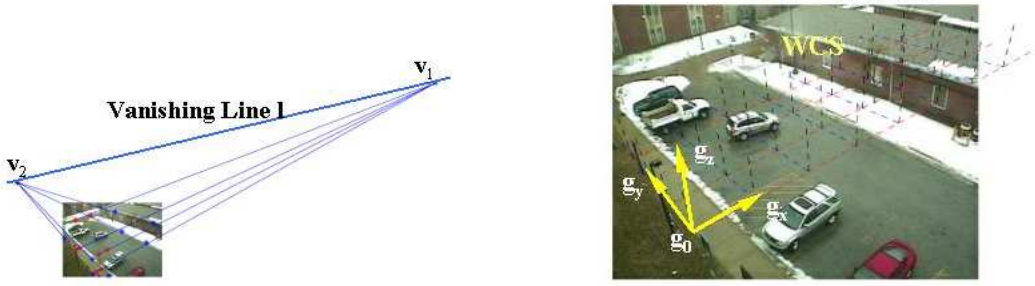


Figure 4.2: *Illustration of the calibration results. Left: Locating vanishing points by forming parallel lines. Right: The WCS used in our work.*

We assume a pinhole camera model, and the image center to be the central projection point. Then the only unknown in $\boldsymbol{K}$ is the focus f.

First, to find f, two vanishing points $\mathbf{v}_1$ and $\mathbf{v}_2$ for two perpendicular lines on the ground plane are located semi-automatically. Using Eq. (4.10), in which $\cos\theta = 0$, f can be found by solving a quadratic equation. Second, the line passing through the two vanishing points is the vanishing line of the ground plane. Hence the plane normal is known. We can set the WCS as $\mathbf{g}_x = \boldsymbol{K}^{-1}\mathbf{v}_1/\|\boldsymbol{K}^{-1}\mathbf{v}_1\|$, $\mathbf{g}_y = \boldsymbol{K}^{-1}\mathbf{v}_2/\|\boldsymbol{K}^{-1}\mathbf{v}_2\|$, and $\mathbf{g}_z = \mathbf{g}_x \otimes \mathbf{g}_y$. $\mathbf{g}_0$ can be selected by arbitrarily setting the perspective depth of the image of a point that lies on the ground plane. Results are shown in Figure 4.2.

4.1.2.5 Quantitative Analysis on Synthetic Data

Using the calibration data, we synthesize very realistic tracking results. We do some quantitative analysis on these sequences by adding noise to the tracked feature points and changing the calibration.

Figure 4.3 shows a typical example of the synthetic data. Twenty-six points

are tracked over forty frames. Note that the feature points on the vehicle are chosen to be points that can be tracked on a real vehicle, shown in Figure 4.6.

Figure 4.3: *A typical synthetic sequence. Top: The projected model. The car makes a* 90^0 *turn. Bottom: Synthetic tracking results of feature points with complete data.*

The noise we add to each feature point is drawn from an i.i.d. isotropic Gaussian distribution. We vary the standard deviation σ to test how robust the method is to noise. Two kinds of calibration error are studied here: error from the focal length f, and from the angle between $\mathbf{g}_x$ and the ground plane, denoted as ϕ. Here we assume that $\mathbf{g}_y$ can be reliably estimated, i.e., $\mathbf{g}_y$ lies on the ground plane. That requires very accurate estimation of one vanishing point, which comes from only one set of parallel lines and is very plausible in man-made environments.

First, we consider the case that all the feature points are tracked over all the frames. The reconstruction results under two conditions are shown in Figure

4.5. Here we consider the relative shape reconstruction error instead of the error in the estimation of perspective depth, because in most applications we care more about the error in the reconstructed 3D vehicle model . Besides, The error between the estimated shape matrix $\boldsymbol{S}$ and the ground truth gives a more comprehensive evaluation of the results than using perspective depths. The relative errors in shape and motions are calculated through comparing $\boldsymbol{S}$ and $\boldsymbol{M}$ respectively with ground truth matrices after scale normalization. For correct calibration $f = 690$, $\phi = 0^o$, and for wrong calibration $f = 690$, $\phi = -4.6^o$. The same condition applies to Figures 4.4 (a) and (b). As can be seen in Figure 4.5, the reconstruction does not change much visually when noise and incorrect calibration are considered.

In Figure 4.4 (a), the relative construction error is plotted as a function of σ for both correct and wrong calibrations. For each σ, the experiments are repeated 40 times and the average error is plotted. For correct calibration, the structure can be reconstructed perfectly with small σs. When the noise level σ increases to about 7 pixels, performances under correct and incorrect calibrations become almost the same. (c) and (d) show how the reconstruction error changes as the calibration error varies. In both cases, noise with $\sigma = 2$ is added to all the feature points. The reconstruction error is less than 2% when $\delta f = \pm 50$ or $\phi = \pm 5^o$. (b) shows the rank condition by using the ratio of the 3rd to the 4th largest singular value of the scaled measurement matrix. Ideally, the 4th singular value should be zero and the ratio approaches infinity, which is true when σ is close to zero with correct calibration. When σ is large or the calibration is incorrect, the ratio decreases. As can be seen from these figures, the algorithm degrades gracefully with increasing noise level and

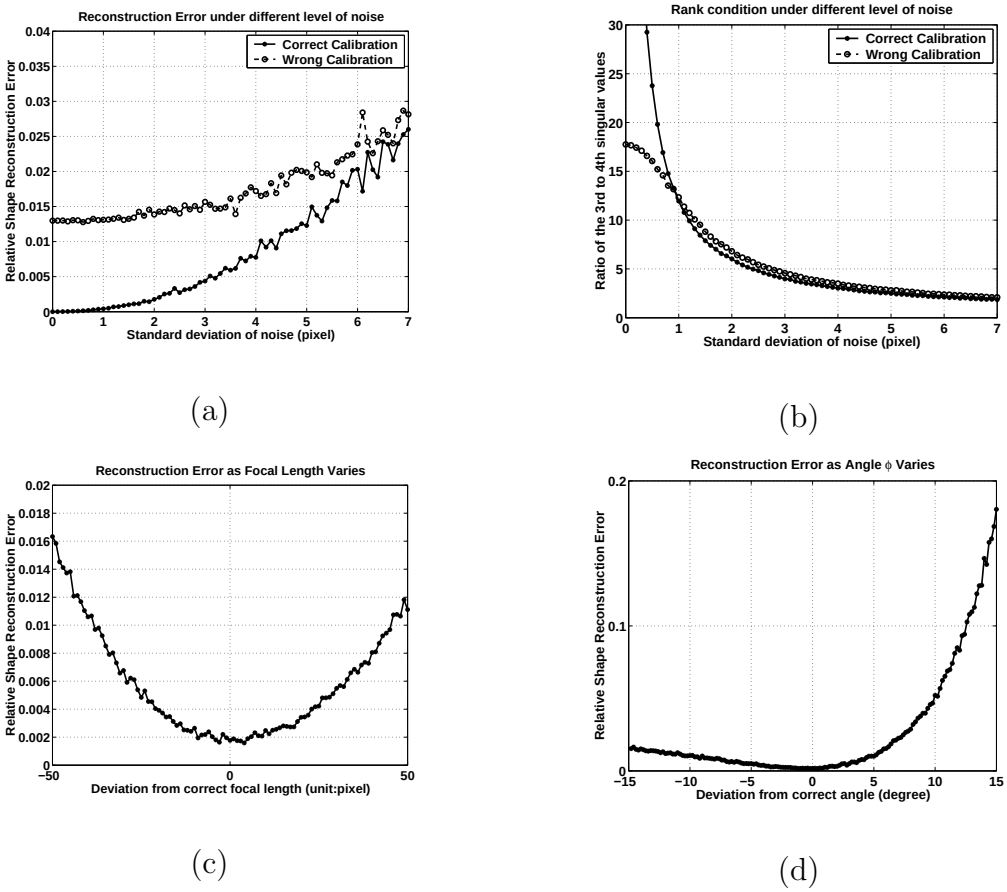


Figure 4.4: *Quantitative analysis of reconstruction error and rank condition. Suppose during estimation of GPC,* $\mathbf{g}_y$ *lies on the ground plane, and the angle between* $\mathbf{g}_x$ *and the ground plane is* ϕ*. For correct calibration* $\phi = 0$*. (a) plots the reconstruction error in the case of correct and wrong calibrations. Here for wrong calibration,* $\phi = -4.6^o$ *and the focal length* f *is correct. (b) The ratio of the 3rd to 4th largest singular values is shown to analyze the rank condition of scaled measurement matrix. The calibration condition is the same as in (a). (c) Reconstruction error as the estimated focal length* f *deviates and* ϕ *remains zero. The true focal length is* $f = 690$*. (d) Reconstruction error as estimation of* $\mathbf{g}_x$ *changes and* $\mathbf{g}_y$ *is assumed to be on the ground plane.* f *is set as* 690*. Noise of* $\sigma = 2$ *is added in (c) and (d).*

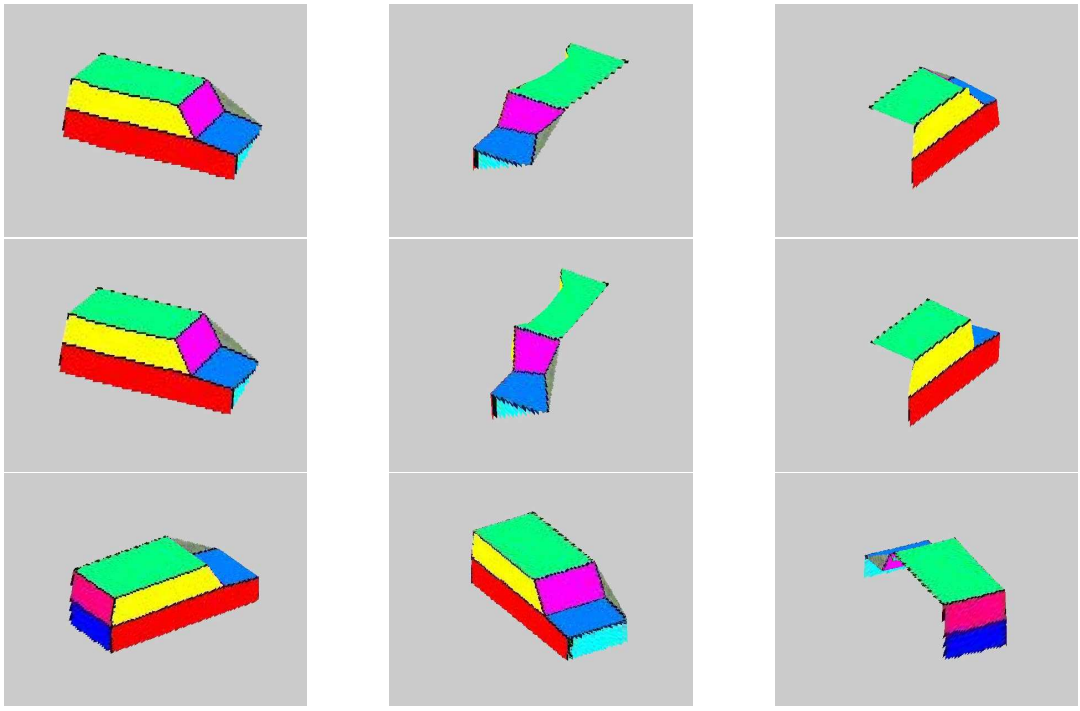


Figure 4.5: *3D reconstruction results for synthetic data. The first two rows are results from observations with no missing data. First row: correct calibration with no noise. Second row: wrong calibration with additive noise of* $\sigma = 5$. *The third row shows the reconstruction results from observations with missing data. So some feature points are not tracked over all the frames. Texture mappings from three frames with different views are fused together to generate a more complete 3D model.*

incorrect calibration.

Second, we also consider the case when some points are occluded in some of the frames. Thirty-four points are tracked in sections of the forty frames. We apply SVD with missing data reported in [6] [63] so that we can have a more complete reconstruction. The reconstruction results for the case with missing data are shown in Figure 4.5. [1]

[1]The video of the results can be found at the author's website *www.cfar.umd.edu/∼lij*.

The 3D models shown in Fig. 4.5 and the rest of the thesis are generated from the depths of the feature points as follows. The factorization method is applied to obtain the perspective depth of each feature point. To map the texture image onto the 3D model, first the dense depth map can be interpolated from the depths of the feature points, so given one frame, each pixel inside the convex hull of the tracked feature points can be associated with a depth value and color. Then using (4.2), the corresponding 3D point of each pixel in WCS can be found. The neighboring four pixels can form a quadrilateral mesh which makes up a 3D model with colors. This 3D model is rotated for qualitative analysis of reconstruction results. For generating reconstruction results with missing data, texture mappings from several different views have to be fused together to construct a complete 3D model. This requires generating the 3D model for two or three selected frames, followed by a transformation of these 3D models to the same reference coordinate system for display. The example with missing data corresponds to noise free and correct calibration. Fig. 4.5 shows that the reconstructed model correctly captures the structure of the vehicle, such as parallel and perpendicular faces.

4.1.2.6 Qualitative Analysis on Real Data

The reconstruction for a real sequence is shown in Figure 4.6. Note that the image of the object is of low resolution and it is taken on a rainy day, making the image more blurry. Thirty points are tracked over forty frames on the vehicle with missing data. Because of the low resolution and blurry effects, many of them cannot

be reliably tracked even manually. However, our method still correctly captures the structures of most faces. The reconstruction is visually realistic. In Fig. 4.7, we show more reconstruction results with manually marked feature points.

Figure 4.6: *Reconstruction results for a real sequence. Top row: The real sequence taken on a rainy day and the tracked feature points. Bottom row: Reconstruction with texture mapping.*

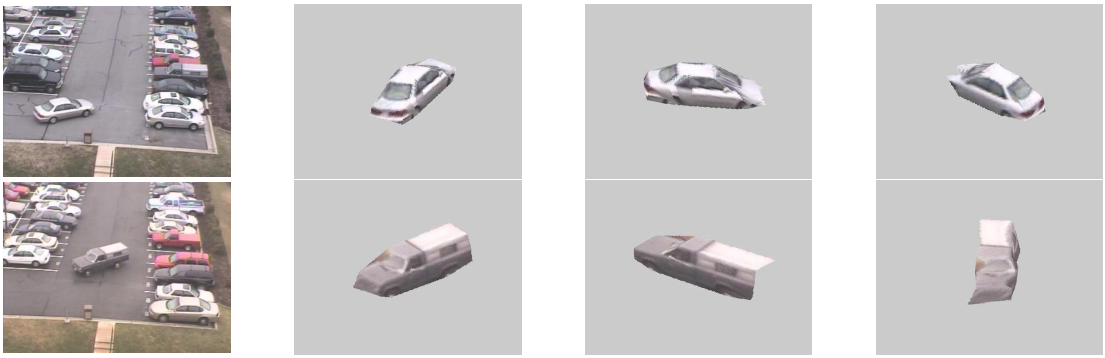


Figure 4.7: *Illustration of reconstruction results. The leftmost column are sample input images, followed by examples of reconstruction results.*

4.2 Planar factorization with uncertainty

In [18], the authors study how to deal with anisotropic noise associated with observations in the context of factorization algorithms. In this section, we also develop an algorithm to deal with noisy observations, which is closely related to the error analysis of the factorization approach discussed in [48].

In [18], each element of the matrix to be factorized is a linear function of the observations. So if we assume that the observations are corrupted by Gaussian noise, these elements take Gaussian distributions. Then instead of minimizing the *Frobenius norm*, one can minimize the *Mahalanobis distance*. But in planar factorization, the elements are not linear function of observations, which makes it difficult to directly use the *Mahalanobis distance*. Approximations are made in order to recast the planar factorization problem into the form in [18].

In order to make things clear, two approaches are illustrated in the following subsection. In the first approach we consider the distribution of elements of $\tilde{\boldsymbol{W}}$. In the second approach the factorization problem is viewed as a parameter estimation problem and the solution is obtained using maximum likelihood estimation (MLE).

4.2.1 Approach I

If $\begin{bmatrix} \alpha_{ti} & \beta_{ti} \end{bmatrix}^T$ takes the Gaussian distribution $N(\begin{bmatrix} \mathbf{m}_t^T\mathbf{s}_i \\ \mathbf{n}_t^T\mathbf{s}_i \end{bmatrix}, \boldsymbol{Q}_{ti})$, then as shown in [18], the following Mahalanobis distance can be minimized to give the MLE of $\mathbf{m}$,$\mathbf{n}$ and $\mathbf{s}$.

$$D(\boldsymbol{M},\boldsymbol{S})=\sum_{t,i}[\begin{bmatrix} \alpha_{ti}-\mathbf{m}_t^T\mathbf{s}_i & \beta_{ti}-\mathbf{n}_t^T\mathbf{s}_i \end{bmatrix} \boldsymbol{Q}_{ti} \begin{bmatrix} \alpha_{ti}-\mathbf{m}_t^T\mathbf{s}_i \\ \beta_{ti}-\mathbf{n}_t^T\mathbf{s}_i \end{bmatrix}] \tag{4.11}$$

However, since $\begin{bmatrix} \alpha_{ti} & \beta_{ti} \end{bmatrix}^T$ is not a linear transformation of $\begin{bmatrix} x_{ti} & y_{ti} \end{bmatrix}^T$, and if we assume that $\begin{bmatrix} x_{ti} & y_{ti} \end{bmatrix}^T$ is corrupted with Gaussian noise, the distribution of $\begin{bmatrix} \alpha_{ti} & \beta_{ti} \end{bmatrix}^T$ is no longer Gaussian.

Our assumption is: Each feature point is corrupted with mutually independent anisotropic Gaussian noise such that

$$\begin{bmatrix} x_{ti} \\ y_{ti} \end{bmatrix} = \begin{bmatrix} \tilde{x}_{ti} \\ \tilde{y}_{ti} \end{bmatrix} + \begin{bmatrix} n_{ti}^x \\ n_{ti}^y \end{bmatrix}, \tag{4.12}$$

where $\begin{bmatrix} \tilde{x}_{ti} \\ \tilde{y}_{ti} \end{bmatrix}$ is the mean and also the true location of each feature point while $\begin{bmatrix} n_{ti}^x \\ n_{ti}^y \end{bmatrix}$ is distributed as $N(0,\Lambda_{ti})$.

Then the probability distribution function of $\begin{bmatrix} \alpha_{ti} & \beta_{ti} \end{bmatrix}^T$ can be found as shown in Appendix I,

$$f_{\alpha_{ti},\beta_{ti}}(\xi_1,\xi_2) = (\frac{f}{\rho_3^T\xi})^2 \frac{1}{2\pi|\Lambda_{ti}|^{\frac{1}{2}}} \cdot \tag{4.13}$$

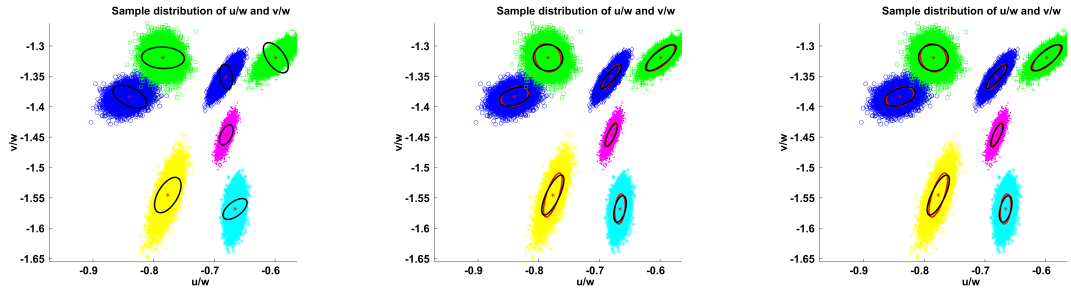
$$\exp[-\frac{1}{2}(\frac{f}{\rho_3^T\xi}\begin{bmatrix} \rho_1^T \\ \rho_2^T \end{bmatrix}\xi - \begin{bmatrix} \tilde{x}_{ti} \\ \tilde{y}_{ti} \end{bmatrix})^T\Lambda_{ti}^{-1}(\frac{f}{\rho_3^T\xi}\begin{bmatrix} \rho_1^T \\ \rho_2^T \end{bmatrix}\xi - \begin{bmatrix} \tilde{x}_{ti} \\ \tilde{y}_{ti} \end{bmatrix})]$$

where $\xi = \begin{bmatrix} \xi_1 & \xi_2 & 1 \end{bmatrix}^T$. Equation (4.13) helps us to understand the statistical behavior of elements in matrix $\tilde{\boldsymbol{W}}$. But there are two major difficulties in using (4.13) in the factorization approach. First, $\tilde{x}_{ti}$ and $\tilde{y}_{ti}$ cannot be directly cast in the form of $\mathbf{m}_t^T\mathbf{s}_i$ and $\mathbf{n}_t^T\mathbf{s}_i$, so the MLE based on (4.13) cannot be changed into the form of (4.11). Second, there is no analytical form for the covariance matrices of this distribution. In the second approach, the MLE is transformed to the form in (4.11), and essentially an estimate of the covariance matrix is given to consider the directional uncertainties.

In Fig. 4.8, we illustrate the distribution of α and β under our assumptions. These samples are generated from the same synthetic data in section 4.1.2.5. The distributions of seven points at same time instants are shown. If the distribution of Gaussian noise is $N(0, \Lambda)$, and suppose the eigenvalues of Λ are λ_1, λ_2 and $\lambda_1 \leq \lambda_2$. Its ellipticity is $e = \sqrt{\lambda_2/\lambda_1}$. Here Gaussian noise with elipticity 2 and $\sqrt{\lambda_1}$ selected randomly between 1 to 3 is added to the observation, then (4.2) and (4.7) are used to generate samples of α and β.

4.2.2 Approach II

In this approach, instead of finding the distribution of (α, β) from the distribution of (x, y), (α, β) is taken as unknown parameters to be estimated. This way we can find an expression similar to (4.11). A similar approach is used in [18]. If we denote $\xi_{ti} = [\alpha_{ti}, \beta_{ti}, 1]^T$, then



(a) (b) (c)

Figure 4.8: *Illustration of distribution of α and β. The samples are generated from the same synthetic data as in section 4.1.2.5. Gray ellipses in each figure illustrate the estimated covariance matrix using different methods. The eigen-values and eigen-vectors of the estimated covariance matrix determine the size and the orientation of each ellipse. Once all covariance matrices are estimated, Eq. (4.26) is used to decompose the ensemble covariance matrix, so that planar factorization with uncertainty can be used. In this step, the covariance matrices are approximated again. Black ellipses correspond to the final approximation. (a) Simply use the covariance matrix of observation noise Λ_{ti} as the estimation. Black ellipses overlap with gray ellipses in this case because for each feature point, the covariance matrix of noise is assumed to be constant over time. (b) With $\tilde{\boldsymbol{Q}}_{ti}$ in Eqs. (4.22) and (4.25), but in Eq. (4.22), (x_{ti}, y_{ti}) is replaced with $(\tilde{x}_{ti}, \tilde{y}_{ti})$, i.e., it is replaced with observation with no noise. (c) With original $\tilde{\boldsymbol{Q}}_{ti}$ in Eqs. (4.25).*

$$\xi_{ti} = \begin{bmatrix} \alpha_{ti} \\ \beta_{ti} \\ 1 \end{bmatrix} = \begin{bmatrix} u_{ti}/w_{ti} \\ v_{ti}/w_{ti} \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{m}_t^T \mathbf{s}_i \\ \mathbf{n}_t^T \mathbf{s}_i \\ 1 \end{bmatrix}, \tag{4.14}$$

where $\mathbf{m}_t$, $\mathbf{n}_t$ and $\mathbf{s}_i$ can be regarded as the unknown parameters we wish to estimate. If we still make the same assumption as in the last section, the distribution of a feature point can be written as

$$f_{\mathbf{x}_{ti}}(x_{ti}, y_{ti}) = \frac{1}{2\pi|\Lambda_{ti}|^{0.5}}$$
$$\exp(-\frac{1}{2}\begin{bmatrix} x_{ti}-\tilde{x}_{ti} & y_{ti}-\tilde{y}_{ti} \end{bmatrix}\Lambda_{ti}^{-1}\begin{bmatrix} x_{ti}-\tilde{x}_{ti} \\ y_{ti}-\tilde{y}_{ti} \end{bmatrix}) \tag{4.15}$$

where $\begin{bmatrix} \tilde{x}_{ti} \\ \tilde{y}_{ti} \end{bmatrix}$ is the mean and also the true locations of feature points which depends on parameters $\mathbf{m}_t$, $\mathbf{n}_t$ and $\mathbf{s}_i$. Since

$$\begin{bmatrix} u_{ti} \\ v_{ti} \\ w_{ti} \end{bmatrix} = \begin{bmatrix} \mathbf{g}_x^T \\ \mathbf{g}_y^T \\ \mathbf{g}_z^T \end{bmatrix}\begin{bmatrix} \tilde{x}_{ti} \\ \tilde{y}_{ti} \\ f \end{bmatrix}, \tag{4.16}$$

$$\begin{bmatrix} \tilde{x}_{ti} \\ \tilde{y}_{ti} \end{bmatrix} = f\begin{bmatrix} \rho_1^T\xi_{ti}/\rho_3^T\xi_{ti} \\ \rho_2^T\xi_{ti}/\rho_3^T\xi_{ti} \end{bmatrix}. \tag{4.17}$$

After some algebraic manipulations, the log-likelihood of a tracked feature point can be derived as

$$\ln(f_{\mathbf{x}_{ti}}(x_{ti}, y_{ti}|\mathbf{m}_t, \mathbf{n}_t, \mathbf{s}_i)) = -\ln(2\pi|\Lambda_{ti}|^{1/2}) - \frac{1}{2(\rho_3^T\xi_{ti})^2}\cdot$$
$$\xi_{ti}^T\begin{bmatrix} x_{ti}\rho_3^T - \mathrm{f}\rho_1^T \\ y_{ti}\rho_3^T - \mathrm{f}\rho_2^T \end{bmatrix}^T\Lambda_{ti}^{-1}\begin{bmatrix} x_{ti}\rho_3^T - \mathrm{f}\rho_1^T \\ y_{ti}\rho_3^T - \mathrm{f}\rho_2^T \end{bmatrix}\xi_{ti}. \tag{4.18}$$

The MLE of the parameters are

$$\underset{\mathbf{m}_t,\mathbf{n}_t,\mathbf{s}_i}{argmin} \sum_{t,i} \frac{\xi_{ti}^T \begin{bmatrix} x_{ti}\rho_3^T - f\rho_1^T \\ y_{ti}\rho_3^T - f\rho_2^T \end{bmatrix}^T \Lambda_{ti}^{-1} \begin{bmatrix} x_{ti}\rho_3^T - f\rho_1^T \\ y_{ti}\rho_3^T - f\rho_2^T \end{bmatrix} \xi_{ti}}{(\rho_3^T \xi_{ti})^2} \tag{4.19}$$

where $t = 1, ...M$, $i = 1, ..., N$ and ξ_{ti} is related to the parameters through (4.14).

If we look at each element in the summation in (4.19), the minimization problem is similar to a generalized eigen-decomposition problem, after it is rewritten as $\frac{\xi^T \boldsymbol{A} \xi}{\xi^T \boldsymbol{B} \xi}$ with $\boldsymbol{A} = \begin{bmatrix} x_{ti}\rho_3^T - f\rho_1^T \\ y_{ti}\rho_3^T - f\rho_2^T \end{bmatrix}^T \Lambda_{ti}^{-1} \begin{bmatrix} x_{ti}\rho_3^T - f\rho_1^T \\ y_{ti}\rho_3^T - f\rho_2^T \end{bmatrix}$ and $\boldsymbol{B} = \rho_3\rho_3^T$. Recall that in general such a minimization will have a solution as shown in [11] using linear discriminant analysis: $\min_\xi \frac{\xi^T \boldsymbol{A} \xi}{\xi^T \boldsymbol{B} \xi} = \lambda$ where λ is the smallest generalized eigenvalue and the minimizer is the corresponding eigen-vector such that $\boldsymbol{A}\tilde{\xi} = \lambda \boldsymbol{B}\tilde{\xi}$, where $\boldsymbol{B}$ is non-singular. Then the term in (4.19) can be approximated by $\sum_{f,p} ||\xi_{ti} - \tilde{\xi}_{ti}||$. However, in this case, $rank(\boldsymbol{B}) = 1$, so a different approximation is developed.

Essentially an approximation in the form of (4.11) is needed. Note that,

$$\begin{bmatrix} x_{ti}\rho_3^T - f\rho_1^T \\ y_{ti}\rho_3^T - f\rho_2^T \end{bmatrix} \xi_{ti} = \begin{bmatrix} -f & 0 & x_{ti} \\ 0 & -f & y_{ti} \end{bmatrix} \begin{bmatrix} \rho_1^T \\ \rho_2^T \\ \rho_3^T \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \\ 1 \end{bmatrix}, \tag{4.20}$$

so that

$$\begin{bmatrix} \alpha_0 \\ \beta_0 \end{bmatrix} = \begin{bmatrix} \mathbf{g}_x^T\gamma / \mathbf{g}_z^T\gamma \\ \mathbf{g}_y^T\gamma / \mathbf{g}_z^T\gamma \end{bmatrix} \tag{4.21}$$

corresponds to the zero point of each element in the summation, where $\gamma = [x_{ti}, y_{ti}, f]^T$. Factorize $\Lambda_{ti}^{-1} = \boldsymbol{Q}^T \boldsymbol{Q}$ and denote

$$\boldsymbol{Q} \begin{bmatrix} x_{ti}\rho_3^T - f\rho_1^T \\ y_{ti}\rho_3^T - f\rho_2^T \end{bmatrix} = \boldsymbol{Q} \begin{bmatrix} \boldsymbol{P}_1 & \mathbf{p}_2 \end{bmatrix} = \begin{bmatrix} \boldsymbol{A} & \mathbf{a}_2 \end{bmatrix}, \tag{4.22}$$

where $\boldsymbol{A}$ is a 2x2 matrix. Then from (4.21), each element can be rewritten as

$$L = \frac{(\begin{bmatrix} \alpha \\ \beta \end{bmatrix} - \begin{bmatrix} \alpha_0 \\ \beta_0 \end{bmatrix})^T \boldsymbol{A}^T \boldsymbol{A} (\begin{bmatrix} \alpha \\ \beta \end{bmatrix} - \begin{bmatrix} \alpha_0 \\ \beta_0 \end{bmatrix})}{([q_1 \ q_2] \begin{bmatrix} \alpha \\ \beta \end{bmatrix} + q_3)^2} \tag{4.23}$$

We propose using

$$\tilde{L} = \frac{(\begin{bmatrix} \alpha \\ \beta \end{bmatrix} - \begin{bmatrix} \alpha_0 \\ \beta_0 \end{bmatrix})^T \boldsymbol{A}^T \boldsymbol{A} (\begin{bmatrix} \alpha \\ \beta \end{bmatrix} - \begin{bmatrix} \alpha_0 \\ \beta_0 \end{bmatrix})}{([q_1 \ q_2] \begin{bmatrix} \alpha_0 \\ \beta_0 \end{bmatrix} + q_3)^2} \tag{4.24}$$

as the approximation. In Appendix II, we show that $L - \tilde{L} = o[(\alpha - \alpha_0)^2] + o[(\beta - \beta_0)^2]$. So $\tilde{L}$ is a fairly robust approximation to L when (α, β) is close to (α_0, β_0). Denote $\tilde{\boldsymbol{Q}}_{ti} = \tilde{\boldsymbol{A}}_{ti}^T \tilde{\boldsymbol{A}}_{ti}$ where $\tilde{\boldsymbol{A}}_{ti} = \boldsymbol{A}_{ti} / ([q_1 \ q_2] \begin{bmatrix} \alpha_0^{ti} \\ \beta_0^{ti} \end{bmatrix} + q_3)$. Our algorithm attempts to minimize

$$D(\boldsymbol{M}, \boldsymbol{S}) = \sum_{t,i} (\begin{bmatrix} \mathbf{m}_t^T \mathbf{s}_i - \alpha_0^{ti} & \mathbf{n}_t^T \mathbf{s}_i - \beta_0^{ti} \end{bmatrix} \tilde{\boldsymbol{Q}}_{ti} \begin{bmatrix} \mathbf{m}_t^T \mathbf{s}_i - \alpha_0^{ti} \\ \mathbf{n}_t^T \mathbf{s}_i - \beta_0^{ti} \end{bmatrix}) \tag{4.25}$$

Compared to (4.11), this minimization replaces the covariance matrix of $(\alpha_{ti}, \beta_{ti})$ with $\tilde{\boldsymbol{Q}}_{ti}$. So by minimizing (4.25), we essentially approximate the distribution of

(α, β) with a Gaussian distribution of covariance matrix $\tilde{\boldsymbol{Q}}_{ti}$. In Fig. 4.8, we illustrate the estimated covariance matrix along with samples of $(\alpha_{ti}, \beta_{ti})$. Three estimation methods are compared. As we can see, Λ_{ti} gives the wrong estimate in (a). $\tilde{\boldsymbol{Q}}_{ti}$ and its modified version both give estimates very close to the true covariance matrices. After further approximations in step (a) of our algorithm summarized below, the estimates are still good.

Given the suboptimal solution in (4.25), the algorithm can be summarized as follows,

Planar Factorization with Uncertainty:

a. Form and decompose the ensemble covariance matrix $\boldsymbol{E}$.

$$\boldsymbol{E} = \begin{bmatrix} \tilde{\boldsymbol{Q}}_{11} & . & \tilde{\boldsymbol{Q}}_{1N} \\ . & . & . \\ \tilde{\boldsymbol{Q}}_{M1} & . & \tilde{\boldsymbol{Q}}_{MN} \end{bmatrix} \approx \begin{bmatrix} \boldsymbol{A}_1 \\ . \\ \boldsymbol{A}_M \end{bmatrix}_{2M\times 2} \begin{bmatrix} \boldsymbol{C}_1 & . & \boldsymbol{C}_N \end{bmatrix}_{2\times 2N} \quad (4.26)$$

The approximation is arrived at by keeping only the two largest singular values and setting the rest to zero in SVD of $\boldsymbol{E}$.

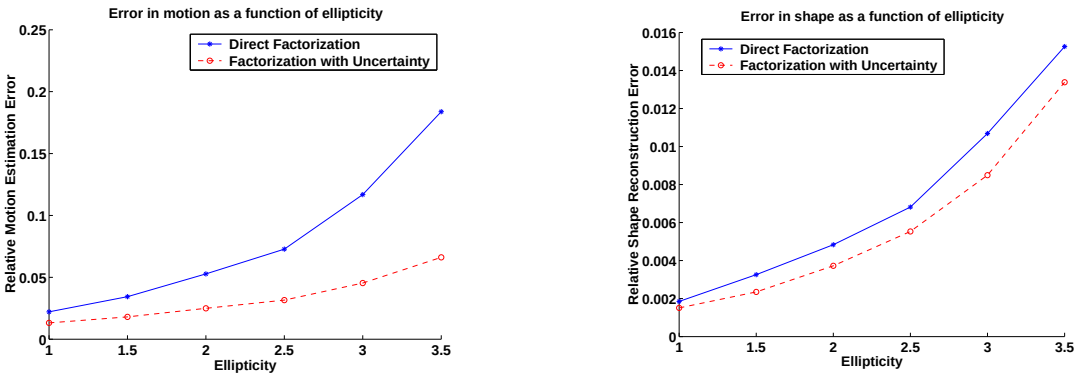
b. For each t and i, compute $\begin{bmatrix} \tilde{\alpha}_0^{ti} \\ \tilde{\beta}_0^{ti} \end{bmatrix} = \boldsymbol{A}_t^T \begin{bmatrix} \alpha_0^{ti} \\ \beta_0^{ti} \end{bmatrix}$. Then follow procedures in [18] to form matrices $\boldsymbol{C}$ and $\begin{bmatrix} \tilde{\alpha}_0 & \tilde{\beta}_0 \end{bmatrix}$. Impose rank 6 constraints on the covariance weighted data, $\begin{bmatrix} \mu & \nu \end{bmatrix} = \begin{bmatrix} \tilde{\alpha}_0 & \tilde{\beta}_0 \end{bmatrix} \boldsymbol{C}$, and obtain $\begin{bmatrix} \hat{\mu} & \hat{\nu} \end{bmatrix}$. If $\boldsymbol{C}$ is well conditioned, find $\begin{bmatrix} \hat{\alpha} & \hat{\beta} \end{bmatrix} = \begin{bmatrix} \hat{\mu} & \hat{\nu} \end{bmatrix} \boldsymbol{C}^{-1}$, then impose rank 3 constraints on $\begin{bmatrix} \hat{\alpha} \\ \hat{\beta} \end{bmatrix} = \tilde{\boldsymbol{M}}_{2M\times 3} \tilde{\boldsymbol{S}}_{3\times N}$, otherwise impose the same constraints with the least square minimization as shown in [18].

c. Recover the motion matrix, $\begin{bmatrix} \hat{\mathbf{m}'}_t^T \\ \hat{\mathbf{n}'}_t^T \end{bmatrix} = (\boldsymbol{A}_t^T)^{-1} \begin{bmatrix} \tilde{\mathbf{m}}_t^T \\ \tilde{\mathbf{n}}_t^T \end{bmatrix}$. Follow step (c) in section 4.1.2 to eliminate the ambiguity and obtain the final reconstructions $\hat{\boldsymbol{M}}$ and $\hat{\boldsymbol{S}}$.

4.2.3 Experimental analysis

In Fig. 4.9, we compare the results of planar factorization with and without uncertainty. This quantitative analysis uses the same synthetic data as in section 4.1.2.5. There is no error in calibration. Anisotropic Gaussian noise $N(0, \Lambda)$ is added to the observation data x_{ti} and y_{ti}. We set $\sqrt{\lambda_1} = 3$ and change the ellipticity $e = \sqrt{\lambda_2/\lambda_1}$, while allowing random orientation of the covariance ellipse but keeping Λ constant for each point over time. For each ellipticity, the experiments are run for 200 times and the average error is shown in Fig. 4.9. Note that factorization with uncertainty does better than direct factorization, especially in terms of motion estimation errors. The motion error varies slowly with the ellipticity for factorization with uncertainty.

Note that in our approach, as $\tilde{Q}_{ti}$ is not directly related to Λ_{ti}, the uncertainty may be directional even if $\lambda_1 = \lambda_2$ for Λ_{ti}. That is, $\tilde{Q}_{ti}$ may still have unequal eigenvalues.



(a) (b)

Figure 4.9: *Comparison of direct factorization and factorization with uncertainty. Ellipticity corresponds to the ratio of eigen-values of the noise covariance matrix* Λ.

4.3 Duality between Planar Motion and Planar Objects

Now we would like to discuss the differences between our work [27] and those in [17, 59] and derive a dual relationship between these approaches. In Appendix A2 of [17], the rank constraints of a planar scene under multiple views are studied. It may appear that our method is a direct derivation from these rank constraints. This is because of confusion between the 3D motion of a planar object (PO) and planar motion (PM) of a 3D object. In [59], the authors use *planar motion* to refer to the motion of a planar object, while throughout this thesis, *planar motion* means that the motion of a 3D object is constrained on a plane. The definitions and detailed comparisons are given later. The term *Planar scene* can also be confusing, but mostly it refers to a *planar object*.

For a planar object, all the feature points lie on one plane, but there is relative motion between the camera and that plane. While for planar motion, there are a set of parallel planes, and each feature point is moving on its own plane, and in our setting there is no relative motion between each plane and the camera. In Appendix A2 of [17], only planar objects are discussed and the conclusion is not applicable to planar motion. However, there is some duality between PO and PM as discussed below.

Another key difference between [17] and our work is the use of affine camera vs. perspective camera. Because for PM discussed in this thesis, there's no relative motion between each plane and the camera, this leads to a simplified approach for perspective cameras. Appendix A2 in [17] discusses affine cameras, and for

perspective cameras, such as in Appendix A3 and A4, they use approximations to obtain the rank constraints. As seen in the literature, most perspective factorizations require either estimating the fundamental matrix, or some approximations. But for PM estimation using a perspective camera, our algorithm is exact.

4.3.1 Formulating Duality

To clearly understand the relation between PO and PM, consider the general setting in structure from motion: a set of feature points $p = (x_{ti}, y_{ti})$ belonging to one object is tracked over time, where $t = 1, ..., M$ is the time index, and $i = 1, ..., N$ is the spatial index for each point. Then our goal is to recover their corresponding 3D points denoted as $P_{ti} = (u_{ti}, v_{ti}, w_{ti})$ in a stationary world coordinate system.

The tracked object is a *planar object* if and only if for any fixed t, points in $\Theta_t \equiv \{P_{ti} | i = 1, ..., N\}$ lie on the same 3D plane Γ_t. The tracked object is undergoing *planar motion* if and only if for any fixed i, points in $\Omega_i \equiv \{P_{ti} | t = 1, ..., M\}$ lie on the same 3D plane denoted as Γ_i and all Γ_is have the same normal for $i = 1, ..., M$. Note that the second definition requires the WCS to be stationary.

The definitions given above are general, but we focus on rigid objects and assume that the camera is stationary. Rigidity leads to important restrictions on the duality between PO and PM. The key to finding the duality is to understand the time and space indices. For PO, the points are spatially distributed on a plane at each time instant, while for PM, each point in space is distributed on a plane across time. It looks as if by swapping the temporal and spatial indices, or transposing an

observation matrix properly, we can interchange the conclusions for PO and PM. However, this proposal is not correct because of two restrictions. First, Γ_ts have to be parallel, so that PO and PM can share the same set of planes. Second, spatial rigidity in both PO and PM prohibits the swapping between time and space. Spatial rigidity means that for any t_1 and t_2, there exists a rigid transformation between corresponding points in Θ_{t_1} and Θ_{t_2}. Literally it means that the points distributed in space have a rigid relationship across time. This kind of rigidity is what we see in the real world. Its dual part can be called temporal rigidity, which means that for any i_1 and i_2, there exists a rigid transformation between corresponding points in Ω_{i_1} and Ω_{i_2}. It suggests that the distribution of each point over time have a rigid relationship across the spatial index. Such rigidity does not belong to any meaningful object in the real world. Since both PO and PM have spatial rigidity, the matrix for decomposition in (4.6) has to be formed the same way and cannot be transposed. The only interchangeable component in PO and PM is the plane related index. In PO, the plane is indexed *temporally* while in PM it is indexed *spatially*. Thus we have the following property.

Duality property between PO and PM: Any theorem for a planar object with parallel Γ_ts has a dual theorem for planar motion through the following changes: keeping the spatial and temporal index of each point but changing the plane related index from t to i. And vice versa by changing the plane related index from i to t.

4.3.2 Dual theorem for planar factorization

The property in the last section is used to find the dual theorem of our planar factorization. The rank constraints for PM proposed in this thesis can be applied to PO under dual condition.

Dual theorem: A planar object moves in the scene but does not change its plane normal $\mathbf{g}_z$ (in CCS) over time with respect to a perspective camera. Suppose $\mathbf{g}_z$ and the camera calibration are found. Then the following matrix $\tilde{\boldsymbol{W}}$ has a rank of at most 3.

$$\tilde{\boldsymbol{W}} \equiv \begin{bmatrix} u_{11}/w_{11} & u_{12}/w_{12} & ... & u_{1N}/w_{1N} \\ v_{11}/w_{11} & v_{12}/w_{12} & ... & v_{1N}/w_{1N} \\ ... & ... & & ... \\ u_{M1}/w_{M1} & u_{M2}/w_{M2} & ... & u_{MN}/w_{MN} \\ v_{M1}/w_{M1} & v_{M2}/w_{M2} & ... & v_{MN}/w_{MN} \end{bmatrix} \tag{4.27}$$

$$= \begin{bmatrix} r^1_{11}/c_1 & r^1_{12}/c_1 & t^1_x/c_1 \\ r^1_{21}/c_1 & r^1_{22}/c_1 & t^1_y/c_1 \\ ... & & \\ r^M_{11}/c_M & r^M_{12}/c_M & t^M_x/c_M \\ r^M_{21}/c_M & r^M_{22}/c_M & t^M_y/c_M \end{bmatrix} \begin{bmatrix} x_1 & x_2 & ... & x_N \\ y_1 & y_2 & ... & y_N \\ 1 & 1 & ... & 1 \end{bmatrix} \tag{4.28}$$

where

$$\begin{bmatrix} u_{ti} \\ v_{ti} \\ w_{ti} \end{bmatrix} = \begin{bmatrix} \mathbf{g}_x^T \\ \mathbf{g}_y^T \\ \mathbf{g}_z^T \end{bmatrix} \begin{bmatrix} x_{ti} \\ y_{ti} \\ f \end{bmatrix}. \tag{4.29}$$

and $\mathbf{g}_x, \mathbf{g}_y$ and $\mathbf{g}_z$ forms an orthonormal basis in R^3.

In this case, c_t has the plane related index and the constraint becomes $\lambda_{ti} w_{ti} = c_t$. Substituting it back in (4.3), we see that these coefficients become part of the motion matrix instead of being part of the shape matrix as in (4.6).

This theorem simplifies the perspective reconstruction of a planar object that does not change its normal over time and is potentially useful.

4.4 A system for automated vehicle model reconstruction

Planar motion is interesting because it is pervasive in surveillance videos. So one important application of our methods is a completely automated vehicle reconstruction system from surveillance videos. In this section, we describe such a system, which is useful for many applications such as object modeling and vehicle identification. In our system, a camera is used to monitor the parking lot; then the system automatically reconstructs the 3D vehicle models from the video input. The system is very efficient and can be easily implemented in real-time. As an extension, we discuss how a prior 3D model can be used for 3D tracking. The model-based approach can be combined with the factorization method for a complete model reconstruction.

In Fig. 4.10, the flowchart of the automatic system is given. It contains four parts:

1. Background subtraction: Here the background image is estimated using the median of each pixel over 40 frames. Then regions with significant variations from the background image are grouped together as possible foreground blobs. In order to deal with illumination changes, in some part of the image, the variations in gradient directions, instead of pixel intensity values, are considered. This is a prototypical block, and other background subtraction algorithms can be applied. The subtraction results are shown in Fig. 4.10 (b).

2. Vehicle Detection: Here we simply consider the size of the blob to remove undesirable blobs such as pedestrians. Again other detection techniques can

be applied here.

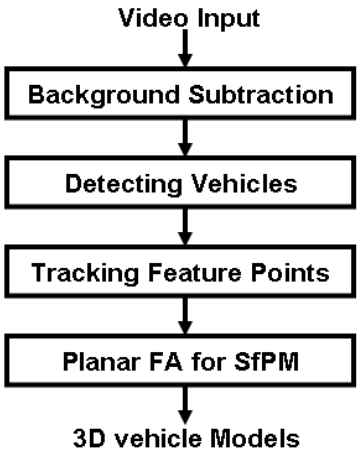
3. Feature Points Tracking: Within the region of detected vehicle, the KLT tracker [52] is used to detect and track feature points over time. To improve the reliability of these tracked feature points, irregular trajectories are removed. Sample tracking results are shown in Fig. 4.10 (c) and (d).

4. Factorization: Finally our factorization method is applied to reconstruct the vehicle model from the tracked feature points. In Fig. 4.11, we illustrate the reconstructed model. Note the calibration step is done beforehand, and the calibration does not change as long as the camera is stationary. Here we apply direct factorization and the directional uncertainty is not considered.

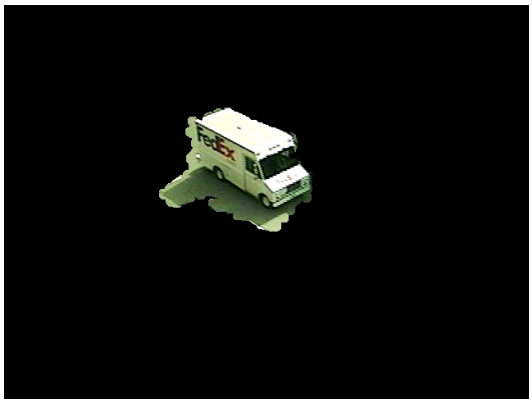
Although the results are still not as good as the manually reconstructed results, it is very promising and other dense depth estimation approaches can be integrated in this framework.

4.4.1 Registration and Tracking

As can be seen from Fig. 4.11, the reconstructed vehicle is only partial because some faces of the vehicle are not observed. In this case, prior 3D models can potentially be used and combined with the reconstructed model to give a complete reconstruction, using properties such as symmetry. But since each type of vehicle has different structures, we can change the geometric parameters and obtain a prior model for each type of vehicles as proposed in [24]. In Fig. 4.12, we illustrate



(a)



(b)

(c)

(d)

Figure 4.10: *(a) Flow chart of the system. (b) Result of background subtraction. (c) and (d) KLT tracking results.*

six types of vehicles, including a sedan, truck, hatchback, wagon, mini-van, and an SUV.

In this section, we discuss how to use a 3D model for 3D tracking of vehicles moving on the ground plane. Our approach is another illustration of the GeT cast in a form that relates to model registration. This GeT resembles the Hough transform for shape matching, but here it is viewed from a different perspective and the notion

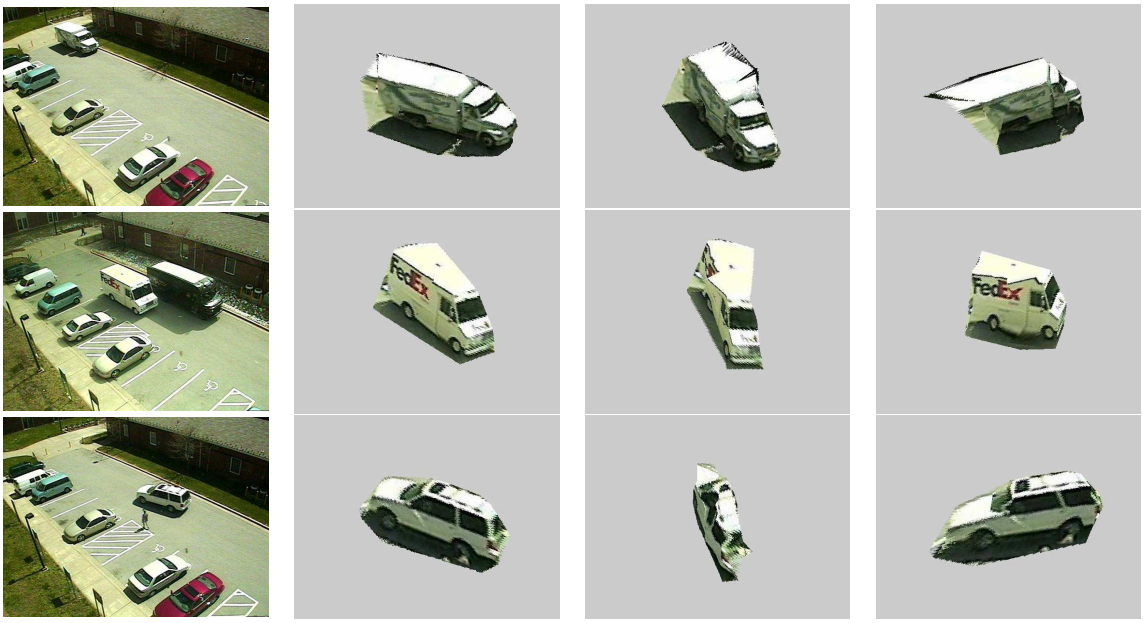


Figure 4.11: *The leftmost column are sample input images, followed by examples of reconstruction results.*

of multi-resolution is included in the matching process.

Specifically, we show that the cost function used for model registration is from MRGeT. For example, if we define the geometric set as the image of a 3D wire-frame model after hidden line removal, and apply the multi-resolution transform over the image of gradient magnitude, then the registration parameters can be found by locating the peak in the transform domain. Mathematically, the transform is defined as,

$$\boldsymbol{R}(S_p) = \int |\nabla I(\mathbf{x})| \frac{1}{\sqrt{2\pi\epsilon^2}} \exp(-\frac{d(\mathbf{x}, S_p)^2}{2\epsilon^2}) d\mathbf{x} = \int |\nabla I(\mathbf{x})| \chi^{\epsilon}_{S_p}(\mathbf{x}) d\mathbf{x}, \qquad (4.30)$$

where S_p corresponds to the projection of the model with its 3D location parameter as p. The value $\arg\max_p \boldsymbol{R}(S_p)$ is the desired registration parameter. The introduction of ϵ allows inexact matching of the model, which has similar advantages as in the case of Hausdorff distance applied to shape detection. The registration result

Figure 4.12: *Examples of 3D vehicle models for sedan, truck, hatchback, wagon, mini-van, and SUV respectively. These models are projected onto a real image taken from a surveillance site.*

is illustrated in Fig. 4.13. If we register the model over time, this method can be used for 3D tracking. Here the type of the vehicle is assumed to be a SUV.

4.4.2 Conclusion and future work

A factorization method for structure from planar motion is proposed and illustrated with experiments. The method fully exploits the constraints, and uses SVD to batch process the data to obtain the shape and motion matrices. It greatly simplifies the factorization approach for general motions under the perspective camera by reducing the rank constraints, as well as avoiding the estimation of the fundamental matrix or using iterative or approximate techniques. Because the planar motion studied in this thesis is very common in surveillance videos and this method is very efficient, it has good potentials in surveillance applications, especially for systems

Figure 4.13: *Left: Image of gradient magnitudes. Right: Registration results. In this example, the ground plane constraint is estimated, so the parameters for set* S_p *as in (4.30) are* (t_x,t_y,ϕ)*, corresponding to translation and rotation motions on the ground plane. The maximum is found by using a deterministic pattern search method.*

related to vehicles, such as vehicle identification or activity analysis. A fully automated vehicle reconstruction system based on this method has been presented. It contains several modular blocks that can be replaced with other background subtraction and tracking algorithms.

Based on experiments done using the vehicle reconstruction system, we feel that feature point tracking is very crucial to the final results. Though factorization with uncertainty has been proposed here to deal with uncertainty in tracking feature points, in real applications, the uncertainty may not be estimated accurately. An alternative direction for future work is to combine the rank constraints with feature point tracking, which can impose some global prior constraints on the feature points and potentially yield improved results.

As seen from section 4.4.1, prior 3D models can be registered with 2D images and used for 3D tracking of the vehicle. Alternatively, the 3D model can potentially be registered with the reconstructed 3D model, to make a complete reconstruction, as well as to determine the type of the vehicle. This approach can be further extended to establish the identities of vehicles across multiple cameras so that the vehicle can be persistently tracked in camera networks. The GeT used in section 4.4.1 can potentially be used to transform a vehicle image into images at different views, based on the geometric models, which makes 2D matching of the appearances possible.

Viewed from a different perspective, the planar factorization can also be used to reconstruct a 3D stationary scene when the camera is moving on a planar platform. For example, if a camera is mounted on a vehicle, and the vehicle is moving on a roughly planar ground, and the scene in the camera is stationary like buildings and road signs *etc.*, then we can track some feature points on these stationary objects, and use the results of this section for 3D reconstruction of these objects. That can provide an interesting way for urban metrology.

Other future work includes possible generalization of our approach to non-rigid motion on the ground plane, SfPM for video captured on moving platforms, and study of motion constrained on non-planar surfaces.

Chapter 5

Summary and future research directions

This dissertation addresses a fundamental problem in appearance modeling: how can we incorporate the geometric prior information based on the shape and motion of the object? The framework we propose essentially transforms the appearance representation geometrically so that in the transform domain, certain correspondences are taken into account for properly aligning the representation. Therefore, in the transform domain, the effects of pose and view variations are reduced and direct matching of appearances is feasible, especially when the appearance inside an arbitrary contour is to be modeled.

Two major types of objects are studied, human and vehicle. Human is an example of an object with articulated motion and vehicle is an example of an object with a complex 3D structure. A geometric appearance model is built for the human based on the geometric transform, and planar factorization is proposed to reconstruct the 3D structure of a vehicle.

This dissertation can be summarized as follows.

- A generic framework for appearance modeling is proposed using a unifying definition of GeT which can incorporate different geometric context.
- Five new types of GeTs are proposed and applied to fingerprinting the appearance inside an arbitrary contour.

- GeT based on shape matching provides a tool for modeling the appearances of humans with articulated motion of body parts.
- GeT based on shape matching also provides a method for segmentation.
- For vehicles, a simple planar factorization method is proposed and used in an automatic 3D vehicle model reconstruction system.
- Planar factorization under uncertainty deals with anisotropic noise with feature points.
- The difference and duality between planar motion and planar objects are clarified.

Our work can be extended in a number of ways, as already discussed in the conclusion section of each part. Here we briefly summarize them:

- In-depth study of all the proposed GeTs will lead to a better understanding of their performance and limitations. It can also possibly lead to new GeTs. For example, the effect of shape matching errors on GeT based on shape matching can help us to understand what kind of shape matching is more desirable in this GeT. Another example is to find ways to combine the explicit model with shape matching for an improved appearance models.
- GeT based on other cues such as multi-view geometry and temporal correspondence can be explored.

- The geometric appearance model in section 3.1 can be potentially extended to multi-view cameras. Then our methods can be used for persistent tracking of humans using camera networks.

- The part segmentation for human in section 3.2 can potentially be used to segment a carried object or to segment body parts when the pedestrian is partially occluded. The geometric appearance model is potentially useful for human identification in crowded environments.

- The reconstructed 3D model of vehicles can also be used for vehicle identification and persistent tracking of the vehicle across cameras. The prior 3D vehicle model used in section 4.4.1 can be combined with the reconstructed model for a complete 3D model or classification of the vehicle type.

- Also videos captured by a camera moving on a plane can be used to reconstruct 3D stationary scenes for urban metrology.

- The rank constraints for planar motion can be used to improve tracking results and thus contribute to improved reconstruction.

- Finally, similar rank constraints can be derived for non-rigid objects moving on a ground plane, or rigid objects moving on a non-planar but known surface.

In all, appearance modeling for human and vehicle can be further studied using the framework and methods proposed in this dissertation.

Appendix A

Distribution of matrix elements

Following the assumption in 4.2.1, and since

$$\begin{bmatrix} x \\ y \\ f \end{bmatrix} = \begin{bmatrix} \rho_1^T \\ \rho_2^T \\ \rho_3^T \end{bmatrix} \begin{bmatrix} u \\ v \\ w \end{bmatrix}, \tag{A.1}$$

so by change of variables from (x, y, f) to (u, v, w), the joint distribution of u, v, w is derived as

$$f_{uvw}(\zeta_1, \zeta_2, \zeta_3) = \frac{\delta(\rho_3^T \zeta - f)}{2\pi|\Lambda|^{0.5}} \cdot$$
$$\exp[-\frac{1}{2}(\begin{bmatrix} \rho_1^T \\ \rho_2^T \end{bmatrix} \zeta - \begin{bmatrix} \tilde{x} \\ \tilde{y} \end{bmatrix})^T \Lambda^{-1} (\begin{bmatrix} \rho_1^T \\ \rho_2^T \end{bmatrix} \zeta - \begin{bmatrix} \tilde{x} \\ \tilde{y} \end{bmatrix})], \tag{A.2}$$

where $\zeta = [\zeta_1, \zeta_2, \zeta_3]^T$ and $\delta(.)$ is the Dirac delta function. The subscripts t, i are omitted for simplicity. Again using a change of variables from (u, v, w) to $(\alpha, \beta, w) = (u/w, v/w, w)$, and integrating out w we get the marginal distribution of (α, β) as

$$f_{\frac{u}{w}, \frac{v}{w}}(\xi_1, \xi_2)$$
$$= \int_{-\infty}^{+\infty} f_{u/w, v/w, w}(\xi_1, \xi_2, \xi_3) d\xi_3$$

$$= \int_{-\infty}^{+\infty} \frac{\xi_3^2 \delta(\xi_3 \rho_3^T \xi - f)}{2\pi |\Lambda|^{0.5}} \exp[-0.5(\xi_3 \begin{bmatrix} \rho_1^T \\ \\ \rho_2^T \end{bmatrix} \xi - \begin{bmatrix} \tilde{x} \\ \\ \tilde{y} \end{bmatrix})^T \cdot$$

$$\Lambda^{-1}(\xi_3 \begin{bmatrix} \rho_1^T \\ \\ \rho_2^T \end{bmatrix} \xi - \begin{bmatrix} \tilde{x} \\ \\ \tilde{y} \end{bmatrix})]$$

$$= \frac{f^2}{(\rho_3^T \xi)^2 2\pi |\Lambda|^{0.5}} \exp[-0.5(\frac{f}{\rho_3^T \xi} \begin{bmatrix} \rho_1^T \\ \\ \rho_2^T \end{bmatrix} \xi - \begin{bmatrix} \tilde{x} \\ \\ \tilde{y} \end{bmatrix})^T \cdot$$

$$\Lambda^{-1}(\frac{f}{\rho_3^T \xi} \begin{bmatrix} \rho_1^T \\ \\ \rho_2^T \end{bmatrix} \xi - \begin{bmatrix} \tilde{x} \\ \\ \tilde{y} \end{bmatrix})],$$

where $\xi = [\xi_1, \xi_2, 1]^T$. This is the distribution shown in Eq. (4.13).

Appendix B

Approximation to MLE

In order to prove that $L - \tilde{L} = o[(\alpha - \alpha_0)^2] + o[(\beta - \beta_0)^2]$ in Eqs. (4.23) and (4.24), it is equivalent to show L and $\tilde{L}$ have the same Taylor expansion up to second order at (α_0, β_0). Obviously,

$$L|_{\mathbf{v}=\mathbf{v}_0} = \tilde{L}|_{\mathbf{v}=\mathbf{v}_0}, \tag{B.1}$$

where $\mathbf{v} = (\alpha, \beta)$ and $\mathbf{v}_0 = (\alpha_0, \beta_0)$. For the first order,

$$\frac{\partial L}{\partial \mathbf{v}} = \frac{2\boldsymbol{A}^T\boldsymbol{A}(\mathbf{v} - \mathbf{v}_0)}{(q_1\alpha + q_2\beta + q_3)^2} - 2\frac{(\mathbf{v} - \mathbf{v}_0)^T\boldsymbol{A}^T\boldsymbol{A}(\mathbf{v} - \mathbf{v}_0)\begin{bmatrix} q_1 \\ q_2 \end{bmatrix}}{(q_1\alpha + q_2\beta + q_3)^3}, \tag{B.2}$$

$$\frac{\partial \tilde{L}}{\partial \mathbf{v}} = \frac{2\boldsymbol{A}^T\boldsymbol{A}(\mathbf{v} - \mathbf{v}_0)}{(q_1\alpha_0 + q_2\beta_0 + q_3)^2}, \tag{B.3}$$

so $\frac{\partial L}{\partial \mathbf{v}}|_{\mathbf{v}=\mathbf{v}_0} = \frac{\partial \tilde{L}}{\partial \mathbf{v}}|_{\mathbf{v}=\mathbf{v}_0} = \mathbf{0}$. For the second order, it is easy to show that

$$\frac{\partial^2 L}{\partial \mathbf{v}^T \partial \mathbf{v}}\ |_{\mathbf{v}=\mathbf{v}_0} = \frac{\partial \tilde{L}}{\partial \mathbf{v}^T \partial \mathbf{v}}\ |_{\mathbf{v}=\mathbf{v}_0} = \frac{2\boldsymbol{A}^T\boldsymbol{A}}{(q_1\alpha_0 + q_2\beta_0 + q_3)^2}. \tag{B.4}$$

Thus, $L - \tilde{L} = o[(\alpha - \alpha_0)^2] + o[(\beta - \beta_0)^2]$.

Bibliography

[1] P. Aguiar and J. Moura, "Rank 1 weighted factorization for 3d structure recovery: Algorithms and performance analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 9, pp. 1134–1149, September 2003.

[2] R. Basri and D. Jacobs, "Lambertian reflectance and linear subspaces," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 2, pp. 218–233, February 2003.

[3] S. Belongie, J. Malik, and J. Puzicha, "Shape matching and object recognition using shape contexts," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 4, pp. 509–522, 2002.

[4] A. Bhattacharayya, "On a measure of divergence between two statistical populations defined by their probability distributions," *Bulletin of the Calcutta Mathematical Society*, vol. 35, pp. 99–110, 1943.

[5] V. Blanz and T. Vetter, "Face recognition based on fitting a 3d morphable model," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 9, 2003.

[6] M. Brand, "Incremental singular value decomposition of uncertain data with missing values," in *Proc. Eur. Conf. Computer Vision*, May 2002, pp. 707–720.

[7] C. Chui, *Wavelets-A Tutorial in Theory and Applications.* Academic Press, 1992.

[8] T. F. Cootes, G. J. Edwards, and C. J. Taylor, "Active appearance models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 6, pp. 681–685, 2001.

[9] T. Cootes and C. Taylor, "Statistical models of appearance for medical image analysis and computer vision," in *Proc. SPIE Medical Imaging*, 2001.

[10] T. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms (Second Edition).* MIT Press and McGraw-Hill, 2001.

[11] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification.* Wiley-Interscience, 2001.

[12] L. Ehrenpreis, *The Universality of the Radon Transform.* Clarendon Press, Oxford, 2003.

[13] D. A. Forsyth and J. Ponce, *Computer Vision, A Modern Approach.* Prentice Hall, 2003.

[14] L. Gorelick, M. Galun, E. Sharon, R. Basri, and A. Brandt, "Shape representation and classification using the poisson equation," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, June 2004, pp. 61–67.

[15] M. Han and T. Kanade, "Reconstruction of a scene with multiple linearly moving objects," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, June 2000.

[16] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge University Press, 2004.

[17] M. Irani, "Multi-frame correspondence estimation using subspace constraints," *Int. J. Comput. Vis.*, vol. 48, no. 3, pp. 173–194, 2002.

[18] M. Irani and P. Anandan, "Factorization with uncertainty," in *Proc. Eur. Conf. Computer Vision*, June 2000.

[19] A. Jain, *Fundamentals of digital image processing.* Prentice-Hall, Inc., 1989.

[20] A. Kadyrov and M. Petrou, "The trace transform and its applications," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 8, pp. 811–828, 2001.

[21] A. C. Kak and M. Slaney, *Principles of Computerized Tomographic Imaging.* Soc. of Industrial and Appl. Math., 2001.

[22] A. Kale, A. Sundaresan, A. Rajagopalan, N. Cuntoor, A. Roy Chowdhury, V. Kruger, and R. Chellappa, "Identification of humans using gait," *IEEE Trans. Image Processing*, vol. 13, no. 9, pp. 1163–1173, September 2004.

[23] T. Kanade and D. D. Morris, "Factorization methods for structure from motion," *Philosophical Trans. of the Royal Society of London, Series A*, vol. 356, no. 1740, pp. 1153–1173, 1998.

[24] D. Koller, K. Daniilidis, and H. Nagel, "Model-based object tracking in monocular image sequences of road traffic scenes," *Int. J. Comput. Vis.*, vol. 10, no. 3, pp. 257–281, June 1993.

[25] K. Kutulakos and S. Seitz, "A theory of shape by space carving," *Int. J. Comput. Vis.*, vol. 38, no. 3, pp. 199–218, July 2000.

[26] M. Lades, C. Vorbrüggen, J. Buhmann, J. Lange, C. von der Malsburg, R. Würtz, and W. Konen, "Distortion invariant object recognition in the dynamic link architecture," *IEEE Trans. on Computers*, vol. 42, pp. 300–311, 1993.

[27] J. Li and R. Chellappa, "A factorization approach for structure from planar motion," in *IEEE Workshop on Motion and Video Computing*, January 2005.

[28] ——, "Structure from planar motion," *to appear on IEEE Trans. Image Processing*, May 2006.

[29] J. Li and S. Zhou, "Probabilistic face recognition with compressed imagery," in *Proc. of IEEE Conf. on Acoustics, Speech, and Signal Processing*, 2004.

[30] J. Li, S. Zhou, and R. Chellappa, "Appearance modeling under geometric context," in *Proc. IEEE Conf. on Computer Vision*, 2005, pp. II: 1252–1259.

[31] J. Li, S. Zhou, and C. Shekhar, "A comparison of subspace analysis for face recognition," in *Proc. of IEEE Conf. on Acoustics, Speech, and Signal Processing*, 2003.

[32] M. Li and C. Kambhamettu, "Nonrigid point correspondence recovery for planar curves using fourier decomposition," in *Proc. of Asian Conf. on Computer Vision*, Jan 2004.

[33] S. Li, S. Tsuji, and M. Imai, "Determining camera rotation from vanishing points of lines on horizontal planes," in *Proc. Int. Conf. Computer Vision*, 1990, pp. 499–502.

[34] H. Ling and D. Jacobs, "Using the inner-distance for classification of articulated shapes," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, June 2005.

[35] H. Liu, R. Chellappa, and A. Rosenfeld, "Accurate dense optical flow estimation using adaptive structure tensors and a parametric model," *IEEE Trans. Image Processing*, vol. 12, no. 10, pp. 1170–1180, October 2003.

[36] ——, "Fast two-frame multiscale dense optical flow estimation using discrete wavelet filters," *Journal of Optical Society America, A*, vol. 20, no. 8, pp. 1505–1515, August 2003.

[37] D. Lowe, "Distinctive image features from scale invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, 2004.

[38] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2003.

[39] J. Oliensis, "Structure from linear or planar motions," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 1996, pp. 335–342.

[40] S. Osher and N. Paragios, *Geometric Level Set Methods*. Springer-Verlag, 2003.

[41] M. Petrou and A. Kadyrov, "Affine invariant features from the trace transform," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 1, pp. 30–44, 2004.

[42] P. Phillips, S. Sarkar, I. Robledo, P. Grother, and K. Bowyer, "The gait identification challenge problem: data sets and baseline algorithm," in *Proc. IEEE Conf. on Pattern Recognition*, 2002, pp. I: 385–388.

[43] C. J. Poelman and T. Kanade, "A paraperspective factorization method for shape and motion recovery," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 3, pp. 206–218, March 1997.

[44] L. Quan, Y. Wei, L. Lu, and H. Shum, "Constrained planar motion analysis by decomposition," *Image and Vision Computing*, vol. 22, no. 5, pp. 379–389, May 2004.

[45] Y. Ran, Q. Zheng, I. Weiss, L. Davis, W. Abd-Almageed, and L. Zhao, "Pedestrian classification from moving platforms using cyclic motion pattern," in *Proc. of IEEE Conf. Image Processing*, 2005, pp. II: 854–857.

[46] T. Sebastian, P. Klein, and B. Kimia, "Recognition of shapes by editing their shock graphs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 5, pp. 550–571, May 2004.

[47] P. Sturm and B. Triggs, "A factorization based algorithm for multi-image projective structure and motion," in *Proc. Eur. Conf. Computer Vision*, 1996, pp. II:709–720.

[48] Z. Sun, A. M. Tekalp, and V. Ramesh, "Error characterization of the factorization method," *Computer Vision and Image Understanding*, vol. 82, no. 2, pp. 110–137, 2001.

[49] T. Tan, K. Baker, and G. Sullivan, "3d structure and motion estimation from 2d image sequences," *Image and Vision Computing*, vol. 11, no. 4, pp. 203–210, May 1993.

[50] T. Tian, R. Li, and S. Sclaroff, "Articulated pose estimation in a learned smooth space of feasible solutions," in *Proc. of IEEE Workshop on Learning in Computer Vision and Pattern Recognition*, 2005.

[51] C. Tomasi and T. Kanade, "Shape and motion from image streams under orthography: A factorization method," *Int. J. Comput. Vis.*, vol. 9, no. 2, pp. 137–154, November 1992.

[52] ——, "Detection and tracking of point features," Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, Tech. Rep. CMU-CS-91-132, April 1991.

[53] A. Veeraraghavan, A. Roy-Chowdhury, and R. Chellappa, "Matching shape sequences in video with applications in human movement analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 12, pp. 1896–1909, December 2005.

[54] R. Veltkamp and M. Hagedoorn, "State-of-the-art in shape matching," Utrecht University, Tech. Rep. UU-CS-1999-27, 1999.

[55] R. Vidal and J. Oliensis, "Structure from planar motions with small baselines," in *Proc. Eur. Conf. Computer Vision*, 2002, pp. 383–398.

[56] L. Wang and W. Tsai, "Camera calibration by vanishing lines for 3-d computer vision," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 13, no. 4, pp. 370–376, April 1991.

[57] G. Wolberg, *Digital Image Warping*. IEEE Computer Society Press, 1994.

[58] A. D. Worrall, G. D. Sullivan, and K. D. Baker, "A simple, intuitive camera calibration tool for natural images," in *Proc. of the 5th British Machine Vision Conference*, September 1994, pp. 781–790.

[59] L. Zelnik-Manor and M. Irani, "Multi-frame estimation of planar motion," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 10, pp. 1105–1116, October 2000.

[60] R. Zhang, P. Tsai, J. Cryer, and M. Shah, "Shape from shading: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 21, no. 8, pp. 690–706, August 1999.

[61] Q. Zheng and R. Chellappa, "Estimation of illuminant direction, albedo, and shape from shading," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 13, no. 7, pp. 680–702, July 1991.

[62] S. Zhou and R. Chellappa, "Image-based face recognition under illumination and pose variations," *Journal of the Optical Society of America, A*, vol. 22, pp. 217–229, 2005.

[63] S. Zhou, R. Chellappa, and D. Jacobs, "Characterization of human faces under illumination variations using rank, integrability, and symmetry constraints," in *Proc. Eur. Conf. Computer Vision*, 2004.