Feature Based Object Rendering from Sparse Views

CUI, Chunhui

A Thesis Submitted in Partial Fulfillment

of the Requirements for the Degree of

Doctor of Philosophy

in

Electronic Engineering

The Chinese University of Hong Kong

November 2010

ProQuest LLC.
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106 - 1346

I would like to dedicate this thesis to my loving parents

Acknowledgements

This thesis would not have been possible without the support of many people. It is now my great pleasure to take this opportunity to thank them all.

First and foremost, I would like to express my gratitude towards my supervisor, **Prof. King Ngi Ngan** for his supervision, advice, and guidance from every stage of this research. Most of all, he provided me an exciting working environment with enormous freedom to pursue my own interests and develop new ideas. I am also deeply grateful for his steadfast encouragement and generous financial support during these many years.

I would like to thank the other professors in Image and Video Processing (IVP) Lab, **Prof. Wai Kuan Cham**, **Prof. Thierry Blu**, **Prof. Hung Tat Tsui** and **Prof. Xiaogang Wang** for their inspiring suggestions and comments on my research work. And I would also thank our lab technician **Yuk-Chung Wong**, for helping me a lot on camera setup and software maintenance.

I also express my sincere thanks to my colleagues and friends, **Dr. Hongliang Li, Dr. Wenxian Yang, Dr. Xin Jin, Dr. Zhenzhong Chen, Dr. Yu Liu, Dr. Jie Li, Dr. Jie Dong, Dr. Haiyan Shu, Dr. Dongdong Zhang, Dr. Chun Man Mak, Wei Zhang, Deqing Sun, Wanli Ouyang, Renqi Zhang, Qian Zhang, Qiang liu, Songnan Li, Cong Zhao, Lin Ma, Kai Lam Tang** and **Yichen Fan** for their kind support and help on my work and life in CUHK.

Last but not least, my sincere gratitude goes to my parents for their unconditional and never ending love which brings me great confidence and courage to follow myself in my life.

Abstract

The objective of this thesis is to develop a multiview system that can synthesize photorealistic novel views of the scene captured by sparse cameras distributed in a wide area. The system cost is largely reduced due to the small number of required cameras, and the image capture is greatly facilitated because the cameras are allowed to be widely spaced and flexibly placed. The key techniques to achieve this goal are investigated in this thesis.

The first part of this thesis presents a convenient and flexible calibration method to estimate the relative rotation and translation among multiple cameras. A simple planar pattern is used for accurate calibration and is not required to be simultaneously observed by all cameras. Thus the method is especially suitable for widely spaced camera array. In order to fairly evaluate the calibration results for different camera setups, a novel accuracy metric is introduced based on the deflection angles of projection rays, which is insensitive to a number of setup factors.

The second part of this thesis studies the invariant local features that have been successfully applied to wide baseline matching. However, most existing feature detectors cannot represent and match the image structures near surface boundaries because the image neighborhood usually contains background or multiple foreground surfaces. These boundary features, though not many, are useful for describing the object contour. Scale and affine invariant Fan feature is proposed, which is able to represent and match these boundary structures and meanwhile has good invariance to scale, viewpoint and background changes.

In the third part, we present a novel image-based rendering method based on the global propagation of a bag of invariant features. Our method is able to generate visually pleasant free-view navigation of the focused object, from a very small number of images taken from quite different viewpoints. The bag of invariant features, including the Fan feature, brings good robustness to significant scale and viewpoint changes between wide baseline images. The global propagation enables dense feature correspondences even for low textured surfaces. The feature-based method has been successfully extended to render moving object from multiview video sequences.

摘要

本論文的目標是設計一套有效的多視角系統方案，用少量分步在廣域的相機來捕獲感興趣的場景，帶來逼真的多視角視覺體驗。由於只需要很少的相機來捕獲場景，該方案大量地減少了系統開銷；又由於能夠靈活隨意地佈置這些相機，該方案也極大地簡化了圖像捕獲的過程。本論文將深入討論這個多視角系統中用到的核心技術。

本論文的第一部分提出了一個方便靈活的相機校正方法來估算多個相機之間相對位移和旋轉。該方法借用一個平面的網格圖來達到精確校正的目的。因為不要求所有相機同時觀察網格圖，該方法特別適用於分步在廣域的相機陣列。此外，為了能公平地評估不同相機設置的校正結果，本文介紹了一種基於投影射線偏差角的準確度度量方法。由於該度量方法對很多相機的設置因素都不敏感，因而能為不同的相機校正結果提供一個公平的測量。

本論文的第二部分重點研究了局部不變特徵。這種圖像特徵已經被成功地應用於長基線匹配。然而目前大部分特徵探測器都無法表徵和匹配那些靠近物體邊緣的圖像結構。這主要是因為這些結構的圖像鄰域往往都含有背景或是多個前景表面。儘管這些邊界特徵的數量並不可觀，但是對於物體輪廓的表徵卻起著非常重要的作用。為此，本文提出了尺度和仿射不變的扇形特徵來提取和匹配這些邊界的圖像結構。這種新型的扇形特徵對尺度，視角和背景的變化具有很好的不變性。

本論文的第三部分闡述了一種新型的基於不變特徵全局性傳播的圖像渲染方法。該方法僅需要輸入極少的不同視角的圖像，即可實現對聚焦對象的自由視角導航。由於使用了包括扇形特徵在內的不變特徵包，該方法對圖像中顯著的尺度和視角變化也具有很好的魯棒性。而全局性特徵傳播技術即使對於低紋理的物體表面也能夠生成密集的特徵對應。此外，這種基於不變特徵的渲染方法也能夠成功地根據多視角視頻來渲染移動的物體。

Publications

Journal Papers

- **Chunhui Cui** and King Ngi Ngan, "Plane-based external camera calibration with accuracy measured by relative deflection angle," *Signal Processing: Image Communication*, vol. 25, no. 3, pp. 224–234, 2009.
- **Chunhui Cui**, Qian Zhang and King Ngi Ngan, "Multi-view Video Based Object Segmentation – A Tutorial," *ECTI Transactions on Electrical Engineering, Electronics and Communications*, Thailand, vol. 7, no. 2, pp. 90–105, 2009.
- **Chunhui Cui** and King Ngi Ngan, "Scale and Affine Invariant Fan Feature," *IEEE Transaction on Image Processing*, Accept by minor revision.
- **Chunhui Cui** and King Ngi Ngan, "Wide Baseline Object Rendering Based on Global Feature Propagation," under review.

Conference Papers

- **Chunhui Cui** and King Ngi Ngan, "A Novel Geometric Filter for Affine Invariant Features," In *Proceedings of 2010 IEEE International Conference on Image Processing (ICIP2010)*, Hong Kong, September 2010.
- **Chunhui Cui** and King Ngi Ngan, "Automatic Scale Selection for Corners and Junctions," In *Proceedings of 2009 IEEE International Conference on Image Processing (ICIP2009)*, Eygpt, November 2009.

- **Chunhui Cui**, Wenxian Yang and King Ngi Ngan, "External Calibration of Multi-camera System Based on Efficient Pair-wise Estimation," *Advances in Image and Video Technology, Pacific Rim Symposium (PSIVT)*, Chile, December 2007.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation and Objective

Multiview imaging system has attracted considerable attention recently due to its increasingly wide range of applications and the decreasing cost of digital cameras. One of the most important research topics is Virtual View Synthesis (VVS) that brings various interesting entertainment services such as three-dimensional television (3DTV), virtual reality, computer games, art and cinema, etc. VVS aims to synthesize the virtual images that would have been seen from novel viewpoints other than those of the input images. This technique has already been applied to films such as "The Matrix". It allows a director to modify a scene by changing the viewpoint without the trouble and expense of reshooting it. It is also the key component of 3DTV that would allow a spectator to watch a show from any desired viewpoint.

The Image-based or Video-based view synthesis technique, also named Image-based or Video-based Rendering (IBR/VBR), is able to generate novel views purely based on input images or videos, without any prior geometrical model. This technique becomes more and more popular due to its flexibility and generality, and is the focus of this thesis. Based on conventional photos, a description of scene content in terms of geometry can be derived based on matching techniques. From this more abstract scene description, rendering techniques are able to create novel views of the recorded scene. Though IBR has been intensively researched recent years, there are still many new and challenging issues to be addressed. In

particular, most IBR approaches require the imaging system to capture the scene or object using a large number of cameras and usually generate novel views that cannot depart much from the camera array positions. A densely sampled camera array is expensive and requires a lot of manual efforts to set up. Moreover, processing a large number of input images requires large data storage and high memory cost. On the other hand, in the past few years there is an increasing demand for IBR from the images acquired using simple devices such as family photographs. One of the interesting applications would be to produce 3D photos from a small set of pictures easily captured by common consumer cameras.

The goal of this thesis is to develop a multiview rendering system that is built up by a very sparse camera array, yet is able to synthesize photorealistic virtual images observed from novel viewpoints. Since our system requires only a small number of input views, the cost of the equipments can be largely saved and the effort of calibration and synchronization can be significantly reduced, which makes our system much more practical and flexible. However, the wide spacing between cameras (views) also brings many challenges in synthesizing photorealistic views: The pattern for accurate camera calibration cannot be simultaneously and clearly observed by widely separated cameras; The strong photometric and geometric changes will make it much more difficult to establish correspondences between wide baseline images; Smooth and low-texture regions will be more difficult to match as opposed to dense camera array systems where interpolation can already produce sufficiently accurate results. In this thesis, we aim to address all these problems and enable image-based rendering from sparse views.

In the following parts of this chapter, we first present a brief overview of previous research works related to the topics studied in this thesis, including camera calibration in Section 1.2, invariant local feature in Section 1.3 and image-based rendering in Section 1.4. The outline of this thesis is then given in Section 1.5.

1.2 A Quick Glance of Camera Calibration

Most of the methods for estimating the 3D structure of a scene through image analysis require an accurate a priori knowledge of the acquisition system's model.

The parameters of this model can be estimated through a process called camera calibration, which is based on the analysis of image features of one or more views. The parameters of the model basically include the *intrinsic parameters* that characterize the projection of 3D points onto the image plane of camera, such as camera focal length, principal point and lens distortion[1], and the *extrinsic parameters* that denote the coordinate system transformations (translations and rotations) from the 3D world coordinates to the 3D camera coordinates, or between multiple camera coordinates. On the other hand, the targets that originate the image features can be "artificial", e.g., planar patterns that have been intentionally added to the scene, or "natural", e.g., natural object features such as vertices or corners. The use of the targets gives rise to two main approaches to the calibration problem. One that relies on additional images of the artificial pattern with fully or partially known geometry falls into the category of *pattern-based calibration.* The second approach extracts the natural features from the captured images themselves and tries to estimate both the scene geometry and model parameters, which is therefore called *self-calibration.*

1.2.1 Pattern-based Calibration

Pattern-based calibration usually assumes that an object (mostly a pattern) with precisely known geometry is present in the input images, and computes the camera parameters consistent with a set of correspondences between the features defining the pattern and their observed image projections [11]; [140]. It is often used in conjunction with positioning systems such as a robot arm [118] or a turntable [30] that can repeat the same motion with high accuracy, so that object and calibration chart pictures can be taken separately but under the same viewing conditions.

Pattern-based calibration can be used regardless of scene texture and view separation, but it is difficult to design and build accurate calibration patterns clearly visible from all views, especially when the camera views are wide apart. This is particularly true for 3D patterns [140], which are desirable for uniform accuracy over the visible field, but remains a problem even for printed planar

[1]Please refer to Section 2.2

grids [156], where the plates that the paper is laid on may not be quite flat, and for laser printers [7] that are sometimes surprisingly inaccurate. In addition, the robot arms or turntables [30] used in many experimental setups may not be exactly repetitive. In fact, even a camera attached to a sturdy tripod may be affected during experiments by vibrations from the floor or thermal effects. These minor factors may not be negligible for modern high-resolution cameras.

It is necessary to mention the most commonly adopted pattern-based methods, including those of Tsai [140], Heikkila & Silven [54] and Zhang [156]. These methods are all based on the pinhole camera model and include terms for modeling radial distortion.

Tsai's calibration model [140] assumes that some parameters of the camera are provided by the manufacturer, to reduce the initial guess of the estimation. It requires n features points ($n > 8$) per image and solves the calibration problem with a set of n linear equations based on the radial alignment constraint. A second order radial distortion model is used while no decentering distortion terms are considered. The two-step method can cope with either a single image or multiple images of a 3D or planar calibration grid, but grid point coordinates must be known.

The technique developed by Heikkila & Silven [54] first extracts initial estimates of the camera parameters using a closed-form solution and then a non-linear least-squares estimation employing a the Levenberg-Marquardt algorithm is applied to refine the interior orientation and compute the distortion parameters. The model uses two coefficients for both radial and decentering distortion, and the method works with single or multiple images and with 2D or 3D calibration grids.

Zhang's calibration method [156] requires a planar checkerboard grid to be placed at different orientations (more than 2) in front of the camera. The developed algorithm uses the extracted corner points of the checkerboard pattern to compute a projective transformation between the image points of the n different images, up to a scale factor. Afterwards, the camera interior and exterior param-

eters are recovered using a closed-form solution, while the third- and fifth-order radial distortion terms are recovered within a linear least-squares solution. A final nonlinear minimization of the reprojection error, solved using a Levenberg-Marquardt method, refines all the recovered parameters. Zhang's approach is quite similar to that of [138], which requires at least 5 views of a planar scene.

1.2.2 Self-Calibration

Self-calibration estimates both the scene geometry and the camera parameters, which are consistent with a set of correspondences between scene and image features [105]; [52]. In this process, the intrinsic camera parameters are often supposed to be known a priori [99]. A final bundle adjustment stage is then typically used to fine tune the positions of the scene points and the entire set of camera parameters in a single non-linear optimization [139].

In a typical self-calibration system for example [105], natural features may first be found as "interest points" such as blobs and corners in the input images, before a robust matching technique such as RANSAC [37] is used to simultaneously estimate a set of consistent feature correspondences and camera parameters.

Some approaches propose to improve feature correspondences for robust camera calibration [86]. However, reliable automated self-calibration systems are hard to design, and they may fail for scenes composed mostly of objects with weak textures (e.g., human faces). In this case, manual feature selection and pattern-based calibration are the only viable alternatives.

A few researchers have proposed using scene information to refine camera calibration parameters: Lavest et al. [68] propose to compensate for the inaccuracy of a calibration chart by adjusting the 3D position of the markers that make it up, but this requires special markers and software for locating them with sufficient sub-pixel precision. The calibration algorithms proposed in [55] and [150] exploit silhouette information instead. They work for objects without any texture and are effective in wide baseline situations, but are limited to circular camera motions.

Basically speaking, automated self-calibration methods tend to work well for close-by cameras in controlled environments and may be ineffective for poorly

textured scenes and widely separated input images.

1.2.3 Estimation Methods

A more specific classification of camera calibration methods can be made according to the parameter estimation and optimization technique employed.

Linear techniques are quite simple and fast, but generally cannot handle lens distortion and need a control point array of known coordinates. They can include closed-form solutions, but usually simplify the camera model, leading to low accuracy results. The well-known DLT [1], which is essentially equivalent to an Essential matrix approach, exemplifies such a technique.

Non-linear techniques, such as the extended collinearity equation model forms the basis of the self calibrating bundle adjustment. A rigorous and accurate modeling of the camera interior orientation and lens distortion parameters is provided [16] through an iterative least-squares estimation process.

A combination of linear and non-linear techniques where a linear method is employed to recover initial approximations for the parameters, after which the orientation and calibration are iteratively refined [32]; [140]; [148]; [54]; [156]. This two-stage approach has in most respects been superceded for accurate camera calibration by the bundle adjustment formulation above, which is also implicitly a two-stage process.

1.3 Recent Advances on Invariant Local Features

Image representation and understanding is to handle the passage from pixels to semantic content of the image. Global approaches based on color or texture distribution analyze the image as a whole. Unfortunately, they are not robust against occlusions, background clutter and other content changes, which are introduced by arbitrary imaging conditions. An efficient approach which provides a possible

solution to these problems is based on local invariant features. The local features are local image structures formed by pixels of high intensity variation. These local structures convey more information due to signal changes, and hence are more representative for the image.

Local features have been widely used in quite varied applications, such as wide baseline matching for stereo pairs [9]; [87]; [141], model based recognition [36]; [79]; [100]; [111], object retrieval in video [123]; [124], visual data mining [125], building panoramas [17], and object categorization [24]; [29]; [34]. In all these applications, the most crucial requirement for the local features is that they should correspond to the same pre-image for different viewpoints, i.e., their shape is not fixed but automatically adapts, based on the underlying image intensities, so that they are the projection of the same 3D surface patch. Since the local features are extracted independently of viewing conditions, they are named invariant local features.

Basically, a *feature* is represented by a small local neighborhood in the image and is defined by the coordinate of a keypoint, the size and the shape of a local structure. These can be affected by different image transformations which most frequently are rotation, scaling, perspective deformations as well as changes in pixel intensities. A feature detector should first provide the accurate location of keypoints detected in transformed images; otherwise the point neighborhoods do not correspond to each other. The scale, namely the size of a point neighborhood is the second important parameter to estimate. Finally, each structure has a specific shape which can be deformed under arbitrary viewing conditions. The problem is to determine the shape of the point independently of these conditions.

1.3.1 Scale Invariance

The scale of a local structure is related to the resolution at which the structure is represented in an image. The resolution is determined during the acquisition of images by the parameters of the camera or the scanner, and cannot be artificially increased, although it can be decreased by smoothing and sampling. The scale is, in fact, the factor of relative change in the size of a local structure represented in two images with different resolution. Therefore the term scale is always related

to the resolution at which the structure is presented. Given a local structure, there exists a minimal resolution, below which the structure is meaningless and a maximal resolution, which depends mainly on the constraints defining the local character of the structure. The problem is to select the appropriate scale at which the structure is most representative. Without the property of scale invariance the complexity of a matching algorithm is high in the case of significant scale changes.

Gaussian Scale Space

In the discrete domain of digital images the scale parameter is also discretized. Thus, the scale space representation is a set of images represented at different discrete levels of resolution. Koenderink [64] showed that the scale space must satisfy the diffusion equation for which the solution is a convolution with the Gaussian kernel. Furthermore he showed that this kernel is unique for generating a scale space representation. The uniqueness of the Gaussian kernel was concerned with different formulations by Babaud [6], Lindeberg [74] and Florack [40]. These results lead to the conclusion that the convolution with the Gaussian kernel is the best solution to the problem of constructing a multi-scale representation. The bi-dimensional Gaussian function is defined by:

$$G(\sigma) = \frac{1}{2\pi\sigma^2} \exp(-\frac{x^2 + y^2}{2\sigma^2}) \tag{1.1}$$

Different levels of the scale space representation are generally created by convolution with the Gaussian kernel:

$$L(\mathbf{x}; \sigma) = G(\sigma) * I(\mathbf{x}) \tag{1.2}$$

with I the image and $\mathbf{x} = [x, y]^\top$ the point location. The Gaussian kernel is circularly symmetric and parameterized by one scale factor σ. A coarse scale image is obtained by smoothing the fine scale image. This operation is repeated on consecutive coarser levels to obtain the multi-scale representation. The scale factor must be distributed exponentially as $\sigma_n = \sigma_0 s^n$, in order to maintain a uniform change of information between successive levels of resolution.

Normalized Gaussian derivatives

A feature can be extracted at different resolutions by applying an appropriate function at different scales. The detection functions are mostly based on Gaussian scale space derivatives, as the linear derivatives of Gaussians are suitable for modeling the human visual front-end. The aim of the scale space analysis is to explore an image representation on a wide range of scales in order to extract the salient information.

In general, the derivatives at different scales can be computed by smoothing the image with the Gaussian kernel and differentiating the smoothed signal. Another option is to convolve the image with a derivative of a scaled Gaussian kernel. All these methods are equivalent. Given any image function $I(\mathbf{x})$, the mth derivative can be defined by:

$$L_{i_1 \ldots i_m}(\mathbf{x};\sigma) = \frac{\partial^m}{\partial i_1 \ldots \partial i_m} G(\sigma) * I(\mathbf{x}) \tag{1.3}$$

Specifically, for an image, we have $i_1 = x$ and $i_2 = y$.

The amplitude of spatial derivatives, in general, decreases with scale due to the response being smoother on a larger scale. In the case of the structures present at a large range of scales, e.g., a step-edge, we would hope to have the derivative constant over scale. In order to maintain the property of scale invariance the derivative function must be normalized with respect to the scale of derivation. The details of scale invariance properties are described in [75]; [77]. The scale normalized derivative D of order m is defined by

$$D_{i_1 \ldots i_m}(\mathbf{x};\sigma) = \sigma^m L_{i_1 \ldots i_m}(\mathbf{x};\sigma) \tag{1.4}$$

,where σ^m is the necessary normalization factor to achieve scale invariance.

Automatic scale selection

Automatic scale selection and the properties of the selected scales have been extensively studied by Lindeberg [77]. The idea is to select the characteristic scale, for which a given function of the input signal attains an extremum over scales. For a particular descriptor, a scale can be named characteristic if the descriptor computed at this scale conveys more information comparing to those

at other scales. Because of this saliency, the characteristic scale is much more reliable to be repeatedly detected under different viewing conditions.

The scale selection operators for computing the scale function of signal are usually constructed from the combinations of several Gaussian derivatives. Chomat et al. [21] show that the gradient operator is appropriate for selecting the characteristic scale of local features and is robust to noise in the image.

$$\text{Squared Gradient} \quad \sigma^2(L_x^2(\mathbf{x};\sigma) + L_y^2(\mathbf{x};\sigma)) \tag{1.5}$$

The magnitude of the gradient is naturally invariant to rotation and the phase can be used to determine the dominant orientation in the local feature.

The Laplacian of Gaussian (LoG) function is circularly symmetric and has been successfully used by Lindeberg [77] for blob detection and automatic scale selection.

$$\text{Laplacian of Gaussian} \quad \sigma^2|L_{xx}(\mathbf{x};\sigma) + L_{yy}(\mathbf{x};\sigma)| \tag{1.6}$$

The Difference of Gaussian (DoG) operator used by Lowe [80] is an approximation of the Laplacian of Gaussian and allows accelerating the computation of a scale space representation.

$$\text{Difference of Gaussian} \quad |L(\mathbf{x};k\sigma) - L(\mathbf{x};\sigma)| \tag{1.7}$$

A more sophisticated approach is to select the scale for which the trace and the determinant of the Hessian matrix assume a local extremum [76].

$$\max(|\text{trace}(H)|) \quad \text{and} \quad \max(|\det(H)|) \tag{1.8}$$

The Harris corner detector [50] is based on the same idea, but uses the components of the second moment matrix μ instead.

$$\text{Harris Function} \quad \det(\mu(\mathbf{x},\sigma)) - \alpha\,\text{trace}^2(\mu(\mathbf{x},\sigma)) \tag{1.9}$$

Recently, Mikolajczyk and Schmid [92] propose a new feature detector that combines the reliable Harris detector and the Laplacian based scale selection. Their Harris-Laplace detector uses the Harris function (1.9) to localize points in each level of the scale space representation. Next, it selects the points, for which the LoG attains a maximum over scale.

The comparative evaluation on these scale selection operators [91] suggests that the LoG operator[1] achieves the overall best performance in detecting scale invariant features.

1.3.2 Affine Invariance

One of the most frequent image transformations introduced by arbitrary viewing conditions is the perspective transformation. The real images generally represent scenes containing partially smooth objects. The smooth surface can be approximated by a piecewise planar surface. The planar surface undergoes perspective transformation when viewed from different angles. Finally, the perspective transformation can be locally approximated by affine transformation. An affine invariant detector can be seen as a generalization of the scale invariant detector. In the case of affine transformation the scaling can be non-uniform, which is different in each direction.

Tuytelaars and Van Gool [141] proposed two approaches for detecting image features in an affine invariant way. The first one, Edge-based Region (EBR), starts from corners and uses the nearby edges. The first step is the extraction of Harris points, which limits the search regions and reduces the complexity of the method. Two nearby edges, which are required for each point, additionally limit the number of potential features in an image. One point moving along each of the two edges together with the Harris point determines a parallelogram. The points stop at positions where some photometric quantities of the texture covered by the parallelogram go through an extremum. The second method, Intensity-based Region (IBR), is purely intensity-based and starts with extraction of local intensity extremum. Next, they investigate the intensity profiles along rays going out of the local extremum. A marker is placed on each ray in the place, where the intensity profile significantly changes. Finally, an ellipse is fitted to the region determined by the markers.

Lindeberg and Garding [78] developed a method to find blob-like affine features using an iterative scheme in the context of shape from texture. The algorithm explores the properties of the second moment matrix to estimate the

[1] closely followed by its approximation DoG

affine transformation of local patterns. Specifically, they propose to extract the keypoints using the maxima of a scale space representation and to iteratively modify the scale and the shape of the point neighborhood. This approach was later implemented in the domain of matching and recognition by Baumberg [9]. He extracts interest points at several scales using the Harris detector and then adapts the shape of the regions to the local image structure using the iterative procedure proposed by Lindeberg. Mikolajczyk and Schimid [92] further improve the method by iteratively modifying the location, scale and the neighborhood of the keypoint, such that both the keypoint and its neighborhood are extracted in an affine invariant way. The resulting features are called Harris Affine if the Harris detector is employed for keypoint extraction or Hessian Affine if Hessian matrix is used instead.

A Maximally Stable Extremal Region (MSER) [87] is a connected component of an appropriately thresholded image. The word "extremal" refers to the property that all pixels inside the MSER have either higher or lower intensity than all the pixels on its outer boundary. The "maximally stable" in MSER describes the property optimized in the threshold selection process. The extremal regions have a number of desirable properties. Firstly, a monotonic change of image intensities leaves the regions unchanged, since it depends only on the ordering of pixel intensities that is preserved under monotonic transformation. Secondly, continuous geometric transformations preserve topology-pixels from a single connected component. Thus after a geometric change locally approximated by an affine transform, homography or even continuous non-linear warping, the transformed extremal region will still be an extremal region. The covariance matrix analysis [120]; [100] can then be employed to normalize both the original and the transformed extremal regions to uniform shape and size. Thirdly, an extremal region is stable because its support is ensured to be virtually unchanged over a range of thresholds. Finally, since no smoothing is involved, both very fine and very large structure is detected, enabling multiple scale selection.

1.3.3 Local description and Similarity Measure

Once the local features are identified in the image, their properties must be captured by descriptors. There are many possibilities to describe the local image structures. The description is necessary for comparing and finding similar structures. The problem is to compute a complete representation that is simultaneously compact and easy to manipulate. The description should be invariant to possible photometric and geometric image transformations. Given the affine invariant features, we can compensate for the affine geometric deformation and compute an affine invariant descriptor. However, the invariance to rotation and illumination changes must also be handled in the process of description. In addition, a reliable similarity measure is required in every application related to matching. The similarity measures are generally determined by the type of descriptors.

The simplest descriptor is a vector of image pixels. Cross-correlation can then be used to compute a similarity score between two descriptors. However, the high dimensionality of such a description results in a high computational complexity for matching. Yet the region can be subsampled to reduce the dimension.

Distribution-Based Descriptors

These techniques use histograms to represent different characteristics of appearance or shape. A simple descriptor is the distribution of the pixel intensities represented by a histogram. A more expressive representation was introduced by Johnson and Hebert [60], where the proposed spin image is a histogram of the point positions in the neighborhood of a 3D interest point. The spin image was recently adapted to images [70], where the two dimensions of the histogram are quantized pixel locations and the intensity value.

Lowe [80] proposed a scale invariant feature transform (SIFT), which combines a scale invariant feature detector and a descriptor based on the gradient distribution in the detected regions. The SIFT descriptor is represented by a 3D histogram of 4×4 gradient locations and 8 orientations. The contribution to the location and orientation bins is weighted by the gradient magnitude. The quantization of gradient locations and orientations makes the descriptor robust

to small geometric distortions and small errors in the region detection. To obtain illumination invariance, the descriptor is normalized by the square root of the sum of squared components.

Gradient location-orientation histogram (GLOH) [93] is an extension of the SIFT descriptor designed to increase its robustness and distinctiveness. The histogram is computed at a log-polar location grid with three bins in radial direction and 8 in angular direction. The central bin is not divided in angular directions. The gradient orientations are quantized in 16 bins.

Geometric histogram [4] and shape context [10] implement the same idea and are very similar to the SIFT descriptor. Both methods compute a histogram describing the edge distribution in a region. These descriptors were successfully used, for example, for shape recognition of drawings for which edges are reliable features.

The size of the descriptors can be further reduced by Principle Component Analysis (PCA) [61]. For example, PCA-SIFT [63] descriptor is a vector of image gradients in x and y direction computed within the support region. The gradient region is sampled at 39×39 locations, therefore, the vector is of dimension 3042. The dimension is reduced to 36 with PCA.

Basically, the Euclidean distance is used to compare the histogram descriptors, including SIFT, GLOH, PCA-SIFT, shape context, and spin images. The measure is easy to compute and has been widely used in matching [93].

Other Descriptors

A set of image derivatives computed up to a given order approximates a point neighborhood. The properties of local derivatives (local jet) were investigated by Koenderink and van Doorn [65]. Florack et al. [39] derived differential invariants, which combine components of the local jet to obtain rotation invariance. Freeman and Adelson [42] developed steerable filters, which steer derivatives in a particular direction given the components of the local jet. Steering derivatives in the direction of the gradient makes them invariant to rotation. Baumberg [9] and Schaffalitzky and Zisserman [112] proposed using complex filters derived from the family $K(x, y, \theta) = f(x, y)\exp(\imath\theta)$. where θ is the orientation. For the function

$f(x, y)$, Baumberg uses Gaussian derivatives and Schaffalitzky and Zisserman apply a polynomial. These filters differ from the Gaussian derivatives by a linear coordinates change in filter response domain.

Generalized moment invariants have been introduced by Van Gool et al. [143] to describe the multispectral nature of the image data. The moments characterize shape and intensity distribution in a region. They are independent and can be easily computed for any order and degree. However, the moments of high order and degree are sensitive to small geometric and photometric distortions. Computing the invariants reduces the number of dimensions. These descriptors are therefore more suitable for color images where the invariants can be computed for each color channel and between the channels.

The Mahalanobis distance is usually employed to measure the similarity of abovementioned descriptors, including differential invariants, steerable filters, complex filters and moment invariants. In order to compute the Mahalanobis distance, one need to first train the covariance matrix from adequate image samples.

1.3.4 Outlier Filter

The descriptors and the similarity measure are necessary but not sufficient to obtain correct point-to-point correspondences. The correctness of the correspondences has to be verified by an additional algorithm that takes into account a global and local geometric relation between the images.

Various outlier rejection approaches have been proposed by inferring the global spatial transform of local features [28]; [80]. These global filters, however, have two major problems: 1) they cannot deal with non-rigid deformation, and 2) they are sensitive to high number of outliers in the correspondence set. To overcome these problems, the use of semi-local geometry information has been explored in the literature. Schmid and Mohr [115] use a fixed number of local features around a given feature to determine its semi-local structure. A similar method has been proposed in [19], where the typical features are combined with shape context to describe the spatial configuration of neighboring feature points. Semi-local constraints are also used by Tuytelaars and Van Gool [141], where an iterative

method rejects mismatches based on homographies between matches of semi-local features. Carneiro and Jepson [19] also propose an efficient pair-wise grouping method. The pair-wise relation is measured by the consistency of scale, distance and heading between a pair of matches. The three consistency measures are then combined together to present robustness to the scaling, translation and rotation. In this way the method makes full use of the available information provided by typical scale invariant features such as SIFT [80].

There are a few works focusing on designing special geometric filters for the state-of-the-art affine invariant features. Lazebnik et al. [69] propose to measure the geometric consistency of triples of matches. The local affine geometry is estimated only by keypoint locations, but the consistency measure takes into account the shape and size information by examining the variation of major and minor axes. The early contraction [36] measures the pair-wise consistency of affine geometry to identify outliers. However, this filter only applies to intersecting regions, but is suitable for match propagation.

1.4 Overview of Image Based Rendering

Image-based rendering (IBR) refers to techniques and representations that allow 3-D scenes and objects to be visualized in a realistic way without full 3-D model reconstruction. IBR uses images as the primary substrate. The potential for photorealistic visualization has tremendous appeal, and it has been receiving increasing attention over the years. Applications such as video games, virtual travel, and E-commerce all benefit from this technology. Recently proposed methods for IBR are described and discussed in [121]. In this section we give a brief overview on IBR techniques.

1.4.1 Matching

For any IBR system that relys on geometric information, matching is the key technique to infer the 3D structure from 2D images. With cameras calibrated, we can compute 3D points by triangulating the 2D correspondences between images. In terms of image acquisition, matching can be classified as narrow

baseline matching and wide baseline matching. In wide baseline matching, the viewpoints of input images are widely separated. This configuration can benefit the accuracy of triangulation and reduce the system cost, but brings difficulties like significant change of image content and severe occlusion. The most successful technique that enables wide baseline matching is invariant local features. Please refer to Section 1.3 for detailed introduction.

The narrow baseline matching, or dense stereo, has been intensively studied over the years and has achieved its maturity. Basically the input images are densely sampled and hence only small differences exist between neighboring images. Therefore, dense depth map can be estimated to represent the 3D scene instead of a sparse set of correspondences as in wide baseline matching. Dense stereo algorithms consist of three fundamental elements, namely the representation, the objective function, and the optimization technique. The representation refers to how the images are used to decide depth or disparity: independent pixels [158], voxels [109], rectangular local windows [102], lines and contours [19], or segments [159]. The objective function specifies the weighting of the data term relative to the regularization term, and indicates how occlusion is handled. Finally, optimizing the objective function can take various forms, such as winner-take-all, dynamic programming [101], graph cuts [12], and belief propagation [132]. For a comprehensive survey on dense stereo techniques, please refer to [111].

1.4.2 Image Based Rendering

Over the past decade, there emerged a number of techniques to synthesize new images of novel views from images. Figure 1.1 lists some main techniques according to how much geometric information is used: with no geometry, with implicit geometry and with explicit geometry. These techniques are briefly introduced as follows.

Concentric mosaics are a generalization of cylindrical panoramas that allows the viewer to explore a circular region and experience horizontal parallax and lighting effects [53]. In this case, instead of using a single cylindrical image, slit cameras are rotated along planar concentric circles. A series of concentric

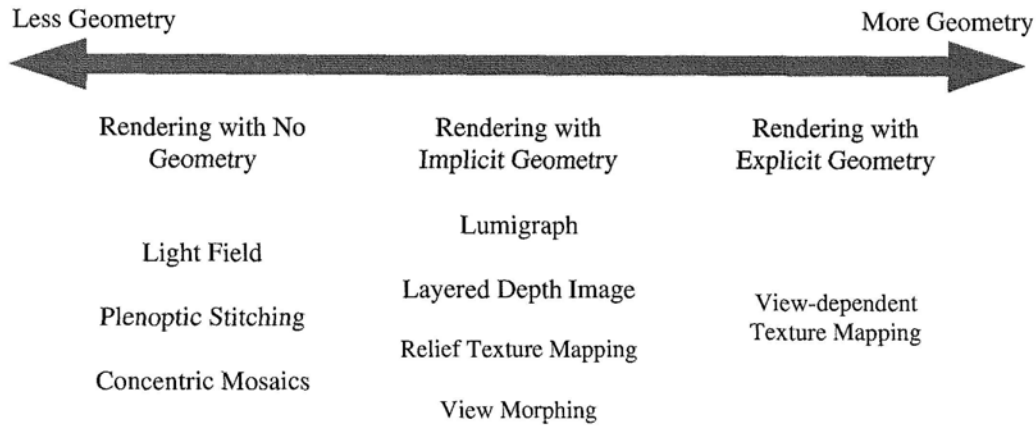


Figure 1.1: Overview of IBR techniques

manifold mosaics [104] are created by composing the slit images acquired by each camera along their circular paths [53]. Thus, a cylindrical panorama is equivalent to a single mosaic for which the axis of rotation passes through the camera's center of projection. In a set of concentric mosaics, all slit images associated to a given column are acquired at the same angle. The use of slit images significantly reduces the amount of data required to approximate the plenoptic function [2]

Plenoptic stitching [3] is a clever technique that gives the viewer the ability to explore (walkthough) unobstructed environments of arbitrary sizes and shapes. In order to provide appropriate sampling for most viewpoints in the environment, an omnidirectional video camera is moved over a grid. Along these paths, the position and orientation of the camera are tracked and stored in synchrony with the corresponding video frames. The intersections among the several paths define image loops. Each loop is segmented as part of a pre-processing step. During a walkthrough, the image loop containing the current viewpoint is used to reconstruct the desired view.

The light field [71] is a function that describes, for any given point, the radiance perceived in a particular direction in free space, which is equivalent to the definition of plenoptic function [2]. Light field and lumigraph [16] rendering

create novel views of scenes or objects by resampling a database of images representing a discrete sample of the plenoptic function. In this representation, a ray is parameterized by its intersections with two parallel planes.

A layered depth image (LDI) [119] is an image with depth that supports multiple samples (color and depth information) per sampling ray. Each element of the image consists of an ordered list of samples. LDIs can be warped in occlusion compatible order. In this case, as a "pixel" is ready for warping, all samples along the corresponding ray are warped. The sample furthest from the novel center of projection (COP) is warped first and the closest is warped last.

Relief texture mapping is an extension to conventional texture mapping that supports the representation of 3D surface detail and view-motion parallax [103]. This effect is achieved by pre-warping the so-called relief textures and mapping the resulting images onto flat polygons. Relief textures, in turn, are parallel projection images with depth. The use of parallel-projection images with depth greatly simplifies the pre-warping, and the rendering of complete 3D objects. The construction process of a relief texture is that the corresponding surface is orthogonally projected onto a reference plane and depth is measured as the per-pixel distance from the plane to the sampled point.

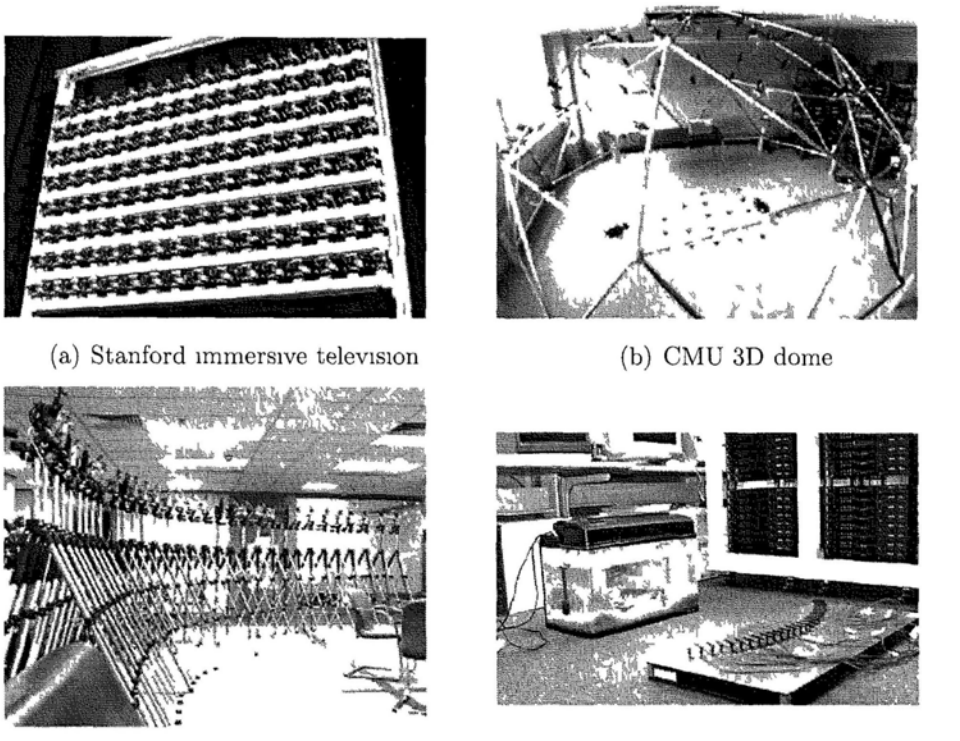
From two input images, view morphing technique [116] reconstructs any viewpoint on the line linking two optical centers of the original cameras. Intermediate views are exactly linear combinations of two views only if the camera motion associated with the intermediate views is perpendicular to the camera viewing direction. If the two input images are not parallel, a pre-warp stage can be employed to rectify two input images so that corresponding scan lines are parallel. Accordingly, a post-warp stage can be used to un-rectify the intermediate images. Scharstein [113] extends this framework to camera motion in a plane. He assumes, however, that the camera parameters are known.

Texture maps are widely used in computer graphics for generating photo-realistic environments. Texture-mapped models can be created using a CAD modeler for a synthetic environment. For real environments, these models can

be generated using a 3D scanner or applying computer vision techniques to captured images. Unfortunately, vision techniques are not robust enough to recover accurate 3D models. In addition, it is difficult to capture visual effects such as highlights, reflections and transparency using a single texture-mapped model. To obtain the visual effect of a reconstructed architectural environment, Debevec et al. [26] used view-dependent texture mapping to render new views, by warping and compositing several input images of an environment. A three-step view-dependent texture mapping method was also proposed later by Debevec et al. [27] to further reduce the computational cost and to have smoother blending. This method employs visibility pre-processing, polygon-view maps, and projective texture mapping.

1.4.3 Rendering from Sparse Views

IBR technique is closely related to the camera layout of a multiview system. A few typical and well-known multiview systems are shown in Figure 1.2. Basically they use the small baseline camera setup, and as a result can capture a large number of reference images to provide adequate overlap between neighboring views for accurate depth estimation. However such a densely sampled camera array not only is expensive but also requires a lot of manual efforts to set up. Besides, rendering from a large number of images also demands extensive data storage and has high memory cost. Furthermore, In the past few years, there is an increasing demand for rendering new scenes from images acquired using simple devices such as family photographs [126]; [112]. The wide baseline configuration [128] is a good solution to overcome the problems of small baseline setup and to meet the demand for the newly emerged applications. Instead of using a densely sampled image sequence, the novel views of the scene is synthesized from a small number of images taken from very different viewpoints, as shown in Figure 1.3. However, the wider spacing between cameras also brings more challenges in producing locally consistent geometries and hence photorealistic views. Especially, the strong photometric and geometric changes between widely separated images make it much more difficult to establish correspondences.



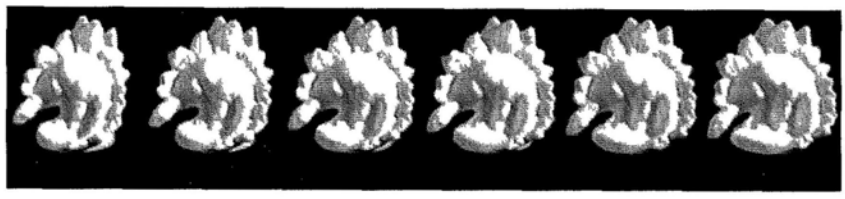
(a) Stanford immersive television (b) CMU 3D dome

(c) MSR-Asia multiview system (d) Nagoya University multiview system

Figure 1 2 Typical multiview system

Recently, local invariant features [95], [93] have been successfully used to address the problem of wide baseline matching For modeling and rendering purpose, however, a sparse set of matches obtained by invariant features is inadequate for producing a satisfactory 3D representation One strategy to generate dense matches between wide baseline images is to introduce the invariance of local features to the general dense matching framework designed for small baseline setup Tola et al [135] proposed to replace the commonly used correlation windows by a fast and robust descriptor, called DAISY They then fed it to a standard graph-cut based depth estimation algorithm Bradley et al [13] proposed a multi-scale window-based stereo algorithm where the horizontal scale of the correlation window is made adaptive to the local surface orientation

Besides, many efforts have been made to address the problem by growing regions or surfaces starting from a small set of extracted features or seed points



(a) Densely sampled multiview images

(b) Sparsely sampled multiview images

Figure 1.3: Multiview images.

[67]; [48]. Strecha et al. [128] develop a dense matching algorithm for multiple wide-baseline images. A sparse set of initial depth estimates is propagated to dense depth map by an inhomogeneous time diffusion process. Lhuillier et al. [72], [73] present a quasi-dense approach to establish surface reconstruction using a greedy match propagation method. Yao et al. [153] further improve this propagation method by introducing the clustering-based photo-consistency and the data-driven depth smoothness. Furukawa et al. [43] propose to represent the scene by a dense set of rectangular patches that cover the surfaces visible in the input images. Their algorithm starts from a sparse set of matched keypoints, and repeatedly expands these to nearby pixel correspondences before using visibility constraint to filter away false matches. The most relevant previous work is [35]; [36], where Ferrari et al. propose to refine the matches of affine invariant features by maximizing the similarity function in the 6D affine space. Later they employ the match refinement to propagate more feature correspondences using the initial matches as the propagation attempts, which has proven to be successful in the application of simultaneous object recognition and segmentation

1.5 Thesis Outline

This thesis aims to develop a multiview rendering system using a very sparse camera array. Accurate multi-camera calibration is an inevitable and important step towards the efficient use of a multiview system. An efficient and flexible multi-camera calibration method is presented in Chapter 2, which is especially suitable for sparsely distributed cameras. Since the input views are widely separated, invariant local features are needed to deal with the severe occlusion and strong photometric and geometric changes between images. In Chapter 3, scale and affine invariant Fan feature is proposed to complement the bag of features for wide baseline matching. The complete feature-based sparse view rendering scheme is discussed in Chapter 4, where the affine consistent outlier filter and the global match propagation technique ensure correct and quasi-dense feature correspondences for sparse view modeling and rendering. In the following, the contents of each chapter are briefly described.

In Chapter 2, we present an efficient external calibration technique for a typical multi-camera system. The technique is very handy in practice using a simple planar pattern. Based on homography, three different pair-wise estimation algorithms, including two-Step, three-Step and non-linear algorithms are proposed to recover the rigid rotation and translation between neighboring cameras. By registering the accurate and reliable partial calibrated structures, the complete calibration of a multi-camera system can be accomplished. The planar pattern is not required to be observed simultaneously by all the cameras, which makes our method more practical and flexible for general calibration purpose. Traditional accuracy measure depends very much on the actual system setup. In order to give fair assessments to the calibration results of different camera systems under different working conditions, a novel accuracy metric is introduced based on the deflection angles of projection rays, which is insensitive to camera focal length, baseline length, scene depth and image resolution. Experimental results using both simulated and real data are presented to verify the validity and performance of the proposed method.

Invariant local features have proven to be very successful in scene representation and image matching for wide baseline camera setting. However, most existing feature detectors assume no surface discontinuity within the keypoints' support regions, and hence have little chance to match the keypoints located on or near the surface boundaries. These keypoints, though not many, are salient and representative. In Chapter 3, we show that they can be successfully matched by using the proposed scale and affine invariant Fan features. Specifically, the image neighborhood of a keypoint is depicted by multiple fan sub-regions, namely Fan features, to provide robustness to surface discontinuity and background change. These Fan features are made scale invariant by using the automatic scale selection method based on the Fan Laplacian of Gaussian (FLOG). Affine invariance is further introduced to the Fan features based on the affine shape diagnosis of the mirror-predicted surface patch. The Fan features are then described by Fan-SIFT, which is an extension of the famous SIFT (Scale Invariant Feature Transform) descriptor. Experimental results of quantitative comparisons show that the proposed Fan feature has good repeatability that is comparable to the state-of-the-art features for general structured scenes. Moreover, by using Fan features we can successfully match image structures near surface discontinuities despite significant scale, viewpoint and background changes. These structures are complementary to those found by the traditional methods, and are especially useful for describing weakly textured scenes, which is demonstrated in our experiments on image matching and later on object rendering in Chapter 4.

In Chapter 4, we present a novel object rendering method based on matching affine invariant features. Our method is able to synthesize photorealistic novel views of still or moving objects from a very small number of images taken from quite different viewpoints, which reduces the system cost and facilitates the capture procedure. Matching the state-of-the-art affine invariant features brings good robustness to significant scale and viewpoint changes between wide baseline images. The resulting correspondences usually have a lot of mismatches. We thus introduce an efficient filter to reject false matches based on the consistency of local affine geometry. Since the initial matches are too sparse to cover the object surface for modeling purpose, we propose to refine and propagate the matches by

optimizing a global function that takes into account both the appearance similarity and the geometric consistency, so that a dense set of correct matches can be produced even for weakly textured surfaces. Finally, a 3D mesh model of the object surface can be constructed, based on which novel views can be synthesized in good quality by a double weighting texturing algorithm. This feature-based rendering scheme can be efficiently extended to render moving objects by a new technique of tracking the affine invariant features across successive frames and accordingly updating the mesh model. Experiments on a few real image and video datasets demonstrate the feasibility of the proposed method.

The thesis is concluded in Chapter 5, where the contributions of this thesis are summarized and the future research directions are discussed.

Chapter 2

Plane-based External Camera Calibration

2.1 Introduction

Camera calibration has always been an essential component of photogrammetric measurement. A vast amount of the vision algorithms in practical use assume pre-calibrated cameras. Traditional photogrammetric calibration [31] employs reference grids and determines the intrinsic matrix $\mathbf{K}$ using images of a known object point array (e.g., a checkerboard pattern). Commonly adopted methods are those of Tsai [140] and Zhang [156]. These methods are all based on the pinhole camera model and include terms for modeling radial distortion. Tsai's calibration model assumes that some parameters of the camera are provided by the manufacturer, to reduce the initial guess of the estimation. It requires n features points ($n > 8$) per image and solves the calibration problem with a set of n linear equations based on the radial alignment constraint. This method can cope with either a single image or multiple images of a 3D or planar calibration grid, but grid point coordinates must be known. Zhang's calibration method requires a planar checkerboard grid to be placed at different orientations (more than 2) in front of the camera. The extracted corner points of the checkerboard pattern are used to compute a projective transformation between the image points of the n different images, up to a scale factor. Afterwards, the camera intrinsic and extrinsic parameters are recovered using a closed-form solution, while the third-

and fifth-order radial distortion terms are recovered within a linear least-squares solution. A final non-linear minimization of the reprojection error, solved using a Levenberg-Marquardt method, refines all the recovered parameters. Zhang's approach is similar to that of Triggs [138], which requires at least 5 views of a planar scene. Recently, the plane-based calibration method has been developed by Malm et al. [83]; [84], where additional constraints are introduced by stereo setup or pure translation scenario. Another category of camera calibration is the multitude of so-called self-calibration algorithms developed during the last decade [89]; [33]; [56]; [81]; [137]; [15]; . These methods do not use a calibration object and only rely on the rigidity of the 3-D scene and on different assumptions on the intrinsic parameters, such as fixed parameters throughout the image sequence. While the self-calibration is very flexible in practical use, it cannot always obtain accurate and reliable results because there are many parameters to estimate while the available constraints are so limited.

Virtual immersive environment usually requires multiple cameras distributed in a wide area to capture scenes of considerable extent. A complete multi-camera calibration is an inevitable and important step towards the efficient use of such systems. For this purpose, many multi-camera calibration methods [131]; [142]; [133] have been developed based on factorization and global constraints. Usually the whole projection matrix P is estimated instead of distinguishing intrinsic and extrinsic parameters. The method proposed in [142] relies on the planar pattern and assumes it to be visible to all cameras. Its applications are limited and unsuitable for wide baseline cases. Other approaches such as [133] and [7] using a laser pointer or virtual calibration object are more flexible, but usually involve elaborate feature detection and tracking, or have some particular requirements on the camera setup (e.g., focusing on the same scene) or the calibration environment (e.g., dark room).

In many practical multi-camera systems designed for real-time 3D video acquisition, the cameras have fixed intrinsic parameters such as the focal length, the principal point and various distortion parameters. By taking into account this fact, it is reasonable and wise to perform the internal camera calibration beforehand for individual camera. Once the multi-camera network has been set up again in a new environment, we only need to estimate the poses of different

cameras, i.e., their relative locations and orientations, and register them together in a world coordinate system. This is generally referred to as external camera calibration. The separation of the internal and external calibration benefits both the accuracy and efficiency of parameter estimation. Specifically, in the presence of noise, there will be less compensating effect between the internal and external parameters, leading to more accurate results of model fitting. Moreover, since fewer parameters are to be estimated, it will in general have lower computational complexity, especially much faster convergence of non-linear optimization. A few works [107]; [20]; [58] have focused their efforts on the external camera calibration, where the intrinsic and distortion parameters are estimated beforehand and regarded as fixed. In [107], Zhang's method [156] is applied to estimate the positions and orientations of the model planes relative to the camera. Using this information, rigid transforms between two cameras are then determined through an arbitrarily chosen plane. Besides, a RANSAC (RANdom SAmple Consensus) [37] procedure is applied to remove possible outliers. A more elaborate approach is presented in [20], where virtual calibration object is used instead of the planar pattern. A structure from motion algorithm is employed to compute the rough pair-wise relationship between cameras. Global registration in a common coordinate system is then performed using a triangulation scheme iteratively. The method proposed in [58] estimates the pair-wise relationship based on the epipolar geometry. Translation and rotation between two cameras are recovered by decomposing the associated essential matrix.

In this chapter, we present an efficient plane-based external calibration method to estimate the relative pose between two neighboring cameras. The technique is simple to use, only requiring the cameras to observe a planar pattern placed at a few different locations and orientations. Generality in the camera position is offered, only reasonable overlap in FOV (field of view) between neighboring cameras is required. Based on homography [52], three different pair-wise estimation algorithms are proposed to recover the rotation and translation between cameras. They are named Two-step, Three-step and Non-linear. The three algorithms impose the orthogonal constraint of rotation in different levels and accordingly achieve different calibration accuracy. The validity of the proposed method is verified through experiments with both simulated and real data. We also show

that accurate pair-wise pose estimation can be reliably employed to register multiple cameras. In addition, a new evaluation metric of calibration accuracy is introduced. It takes into account the impact of both the camera potentials and the stereo setup on the calibration results, specifically including the image resolution, the focal length, the scene depth and the triangulation. Such a metric is based on the deflection angle between the projection rays and called Relative Deflection Angle (RDA). It is convenient to take the RDA measure since it does not require any prior knowledge of the true 3D coordinates of control points. Moreover, compared with traditional metrics, RDA is much less sensitive to the aforementioned factors and as a consequence can provide fairer assessments on the accuracy of calibration results for different cameras and working conditions.

This chapter is organized as follows. Section 2.2 presents the camera model and the basic equations from homography that constitute the foundation of our calibration method. Section 2.3 describes the details of the three pair-wise pose estimation algorithms. Section 2.4 introduces the proposed RDA metric and discusses its properties. Experimental results on synthetic data and real images are presented in Section 2.5 to demonstrate the proposed method. Section 2.6 concludes this chapter.

2.2 Basic Equations

In this section, we briefly introduce the pinhole camera model and the two-view homography that provide the basis for our calibration method.

2.2.1 Pinhole Camera Model

Let $\mathbf{M} = [X, Y, Z]^T$ represent the coordinates of any visible 3D point in the world coordinate system. Its projection onto the image plane is denoted by $\mathbf{m} = [u, v]^T$. The homogeneous coordinates of $\mathbf{M}$ and $\mathbf{m}$ are represented by $\widetilde{\mathbf{M}} = [X, Y, Z, 1]$ and $\widetilde{\mathbf{m}} = [u, v, 1]$. Based on the widely accepted pinhole camera model,the relationship between a 3D point $\mathbf{M}$ and its image projection $\mathbf{m}$ is given

by

$$\widetilde{\mathbf{m}} \cong \mathbf{P}\widetilde{\mathbf{M}} = \mathbf{K}[\mathbf{R}|\mathbf{t}]\widetilde{\mathbf{M}}, \text{ with } \mathbf{K} = \begin{bmatrix} f_u & \gamma & u_0 \\ 0 & f_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.1}$$

where $\cong$ indicates "equal up to scale". $\mathbf{P}$ is called the projection matrix. $\mathbf{R}$ and $\mathbf{t}$ are the rotation and translation that relate the world coordinate system to the camera coordinate system. $\mathbf{K}$ is the camera intrinsic matrix, with (u_0, v_0) the coordinates of the principal point, f_u and f_v the scale factors in the image u and v axes, and γ the parameter describing the skew of the two image axes.

Common cameras usually have visible lens distortion, especially the radial components. Please refer to [148] for a detailed introduction of distortion model.

2.2.2 Two-view Homography Induced by A Plane

Suppose two cameras capture a planar pattern simultaneously as shown in Figure 2.1. Let $\mathbf{m}_1$ and $\mathbf{m}_2$ denote the projections of the same 3D point $\mathbf{M}$ on the plane π onto the camera 1 and camera 2, respectively. Their homogeneous coordinates $\widetilde{\mathbf{m}}_1$ and $\widetilde{\mathbf{m}}_2$ are related by a homography induced by the plane

$$\widetilde{\mathbf{m}}_2 \cong \mathbf{H}\widetilde{\mathbf{m}}_1 \tag{2.2}$$

where $\mathbf{H}$ is the 3×3 homography matrix.

To give an explicit expression of the homography $\mathbf{H}$, let C1 and C2 denote the coordinate systems of camera 1 and camera 2, respectively, and let $\mathbf{K}_1$, $\mathbf{K}_2$ be their intrinsic matrices. Here let $\mathbf{R}$ and $\mathbf{t}$ represent the rotation and translation from C1 to C2 as shown in Figure 2.1. Without loss of generality, we choose C1 as the world coordinate system. Thus the projection matrices for the two cameras are $\mathbf{P}_1 = \mathbf{K}_1[\mathbf{I}|\mathbf{0}]$ and $\mathbf{P}_2 = \mathbf{K}_2[\mathbf{R}|\mathbf{t}]$. The world plane π is defined by

$$\mathbf{n}_\pi^\top \mathbf{M} + d_\pi = 0 \tag{2.3}$$

where $\mathbf{n}_\pi$ is the unit vector in the direction of the plane normal, and d_π is the distance from C1 origin to the plane π.

Since $\mathbf{m}_1$ is the projection of $\mathbf{M}$ onto camera 1, we have $\widetilde{\mathbf{m}}_1 = s\mathbf{P}_1\widetilde{\mathbf{M}} = s\mathbf{K}_1[\mathbf{I}|\mathbf{0}]\widetilde{\mathbf{M}} = s\mathbf{K}_1\mathbf{M}$. The 3D point $\mathbf{M}$ should also satisfy the plane equation

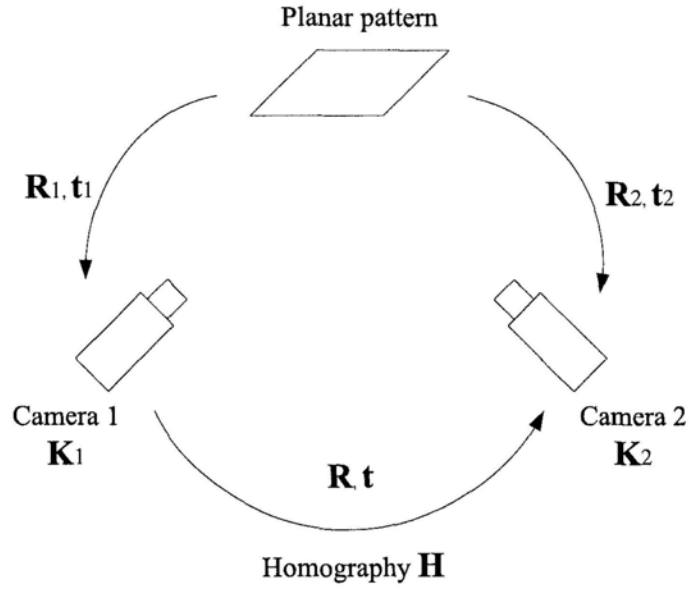


Figure 2.1: Homography between two views.

(2.3). This determines the unknown scale $s = -\mathbf{n}_\pi^\top \mathbf{K}_1^{-1}\widetilde{\mathbf{m}}_1/d_\pi$. Substitude $\mathbf{M} = \mathbf{K}_1^{-1}\widetilde{\mathbf{m}}_1/s$ into $\widetilde{\mathbf{m}}_2 \cong \mathbf{P}_2\widetilde{\mathbf{M}} = \mathbf{K}_2[\mathbf{R}|\mathbf{t}]\widetilde{\mathbf{M}} = \mathbf{K}_2(\mathbf{R}\mathbf{M} + \mathbf{t})$, we can deduce

$$\widetilde{\mathbf{m}}_2 \cong \mathbf{K}_2(\mathbf{R}\mathbf{K}_1^{-1}\widetilde{\mathbf{m}}_1 + s\mathbf{t}) = \mathbf{K}_2(\mathbf{R} - \mathbf{t}\cdot\mathbf{n}_\pi^\top/d_\pi)\mathbf{K}_1^{-1}\widetilde{\mathbf{m}}_1 \tag{2.4}$$

Let $\mathbf{n} = \mathbf{n}_\pi/d_\pi$ denote the plane normal. According to (2.2) and (2.4), we have

$$\lambda\mathbf{H} = \mathbf{K}_2(\mathbf{R} - \mathbf{t}\cdot\mathbf{n}^\top)\mathbf{K}_1^{-1} \tag{2.5}$$

where λ is an unknown arbitrary scalar.

2.3 Pair-wise Pose Estimation

We apply Zhang's method [156] to do the intrinsic calibration for each individual camera. Thus only the external calibration is necessary every time the cameras are moved and refocused to capture new 3D video. Planar pattern such as a checkerboard is widely used in calibration due to its flexibility and convenience. The main drawback of using the planar pattern lies in the fact that it is difficult

to make it visible to all cameras. However, we only use the planar pattern to estimate the relative relationship between two neighboring cameras. It is practical to make the pattern visible to both cameras, because in most multi-view systems two neighboring cameras generally have sufficient common Field Of View (FOV). As for global registration, the transform from one camera to another can be easily computed by chaining those associated neighboring transforms together. It is argued that the chaining procedure is prone to errors. However, accurate and stable pair-wise calibration can benefit the accuracy of transform chaining. This will be demonstrated by the experiments presented in sub-section 2.5.3.

An easy way to do the pair-wise $(\mathbf{R}, \mathbf{t})$ estimation is to utilize the single camera calibration results. Zhang's method [156] can also recover the positions and orientations of the model planes relative to the camera, such as $(\mathbf{R}_1, \mathbf{t}_1)$ and $(\mathbf{R}_2, \mathbf{t}_2)$ in Figure 2.1. Using this information, $(\mathbf{R}, \mathbf{t})$ between two cameras can be determined through an arbitrarily chosen model plane. Ideally, $(\mathbf{R}, \mathbf{t})$ between cameras should be invariant irrespective of the plane through which they are computed. However in the presence of noise, the $(\mathbf{R}, \mathbf{t})$ estimates computed through different planes actually differ from each other. Simply using these estimates may result in very unstable calibration results, since the accuracy of corner detection highly depends on the plane location and orientation. This will be verified later in our simulations.

Based on homography (2.5), we propose a robust pair-wise estimation method to recover the relative relationship between two cameras using multiple stereo images. First, homography $\mathbf{H}$ is estimated by point correspondences, and then follows the calculation of the unknown scale λ and the plane normal $\mathbf{n}$. Finally, $(\mathbf{R}, \mathbf{t})$ between two cameras can be estimated by three different algorithms: Two-step, Three-step and Non-linear method.

2.3.1 Homography Estimation

With sufficient point correspondences, the homography matrix $\mathbf{H}$ can be computed based on (2.2). The algorithm described in [156] is applied to estimate the homography. As shown in Figure 2.1, each image pair, one view from camera 1

and the other from camera 2, leads to a homography $\mathbf{H}$. Suppose there are totally P image pairs and then we can estimate P homographies $\mathbf{H}_i$ $(i = 1, 2, \ldots P)$ induced by different planes.

2.3.2 Calculation of n and λ

The plane normal $\mathbf{n}$ also varies with the moving pattern, thus P different stereo views lead to P different normals $\mathbf{n}_i$ $(i = 1, 2, \ldots P)$. To compute each plane normal with respect to the C1 coordinates system, Zhang's method [156] is first employed to estimate the plane position and orientation $(\mathbf{R}_1, \mathbf{t}_1)$ relative to C1, where the plane π is now assumed on $Z = 0$ of the world coordinate system, as shown in Figure 2.2.

Let us express $\mathbf{R}_1$ by means of its column vectors $\mathbf{R}_1 = [\mathbf{R}_1\, \mathbf{R}_2\, \mathbf{R}_3]$. It can be easily shown that the third column vector $\mathbf{R}_3$ is a unit vector in the direction of Z axis of the world coordinate system, that is to say $\mathbf{R}_3 = \mathbf{n}_\pi$. Also the translation $\mathbf{t}_1$ is the origin of the world coordinate system w.r.t the C1 coordinates system. Since $\mathbf{R}_3$ is orthogonal to the plane π, the distance from C1 origin to the plane can be computed by $d_\pi = |\mathbf{t}_1| \cos\theta_{rt} = |\mathbf{R}_3||\mathbf{t}_1| \cos\theta_{rt} = \mathbf{R}_3^\top \mathbf{t}_1$, where θ_{rt} denotes the angle between $\mathbf{R}_3$ and $\mathbf{t}_1$. Therefore the plane normal $\mathbf{n}$ can be calculated by $\mathbf{R}_3$ and $\mathbf{t}_1$ as

$$\mathbf{n} = \mathbf{n}_\pi / d_\pi = \mathbf{R}_3 / (\mathbf{R}_3^\top \mathbf{t}_1) \tag{2.6}$$

The collineation matrix $(\mathbf{R} - \mathbf{t} \cdot \mathbf{n}^\top)$ in (2.5) has an important property that its median singular value is equal to one [157]. This can be employed to compute the unknown scalar λ. Let us define $\mathbf{G} = \lambda \mathbf{K}_2^{-1} \mathbf{H} \mathbf{K}_1$. From (2.5), we have

$$\mathbf{G} = \lambda \mathbf{K}_2^{-1} \mathbf{H} \mathbf{K}_1 = \mathbf{R} - \mathbf{t} \cdot \mathbf{n}^\top \tag{2.7}$$

Let $(\sigma_1, \sigma_2, \sigma_3)$ denote the singular values of matrix $\mathbf{K}_2^{-1} \mathbf{H} \mathbf{K}_1$ in descending order $(\sigma_1 \geq \sigma_2 \geq \sigma_3 > 0)$. Since the collineation matrix $\mathbf{G} = \lambda \mathbf{K}_2^{-1} \mathbf{H} \mathbf{K}_1$ has median singular value equal to one, we have

$$\lambda \sigma_2 = 1 \tag{2.8}$$

Note that matrix $\mathbf{K}_1$, $\mathbf{K}_2$ and $\mathbf{H}$ are known, so we can compute λ by (2.8) and then recover the matrix $\mathbf{G}$.

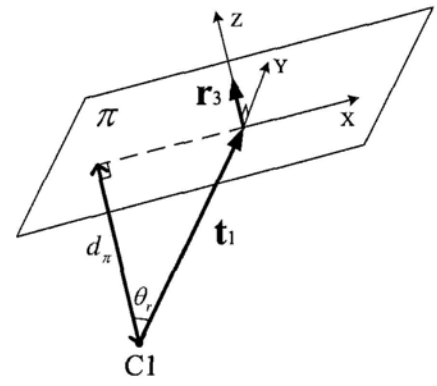


Figure 2.2: Geometry between the model plane and camera center.

2.3.3 (R,t) Estimation

It is straightforward to derive from (2.7) a linear equation (2.9) to solve the rotation $\mathbf{R}$ and translation $\mathbf{t}$. Here vec($\mathbf{X}$) denotes the vectorization of matrix $\mathbf{X}$ formed by stacking the columns of $\mathbf{X}$ into a single column vector, $\mathbf{I}_3$ and $\mathbf{I}_9$ denote 3×3 and 9×9 identity matrices, respectively, and $\otimes$ denotes the kronecker product.

$$\underbrace{\begin{bmatrix} \mathbf{I}_9 & -\mathbf{n}\otimes\mathbf{I}_3 \end{bmatrix}}_{9\times12} \underbrace{\begin{bmatrix} \mathrm{vec}(\mathbf{R}) \\ \mathbf{t} \end{bmatrix}}_{12\times1} = \underbrace{\mathrm{vec}(\mathbf{G})}_{9\times1} \tag{2.9}$$

As each model plane introduces a normal $\mathbf{n}_i$, a homography $\mathbf{H}_i$ and thus a matrix $\mathbf{G}_i$, by stacking P equations we have

$$\underbrace{\begin{bmatrix} \mathbf{I}_9 & -\mathbf{n}_1\otimes\mathbf{I}_3 \\ & \vdots \\ \mathbf{I}_9 & -\mathbf{n}_P\otimes\mathbf{I}_3 \end{bmatrix}}_{9P\times12} \underbrace{\begin{bmatrix} \mathrm{vec}(\mathbf{R}) \\ \mathbf{t} \end{bmatrix}}_{12\times1} = \underbrace{\begin{bmatrix} \mathrm{vec}(\mathbf{G}_1) \\ \vdots \\ \mathrm{vec}(\mathbf{G}_P) \end{bmatrix}}_{9P\times1} \tag{2.10}$$

Because of noise in the data, the computed matrix $\mathbf{R}$ does not in general satisfy the orthogonal property of a rotation matrix. Thus the best orthogonal matrix $\mathbf{R}'$ should be solved to approximate the original $\mathbf{R}$ using a method described in [157]. However, the orthogonal approximation causes a severe problem here. The ($\mathbf{R}$,$\mathbf{t}$) solution of equation (2.10) is best in the least square sense. After orthogonal approximation, the obtained ($\mathbf{R}'$, $\mathbf{t}$) no longer fits this equation well and may

result in erroneous calibration results. Therefore it is necessary to impose the orthogonal constraint $\mathbf{R}^\top\mathbf{R} = \mathbf{I}$ in the $(\mathbf{R},\mathbf{t})$ estimation procedure so that the matrix $\mathbf{R}$ is as close to orthogonal as possible, consequently less deviation will be caused by orthogonal approximation from $\mathbf{R}$ to $\mathbf{R}'$.

2.3.3.1 Two-Step Method with Implicit Orthogonal Constraint

In this section, we first derive an implicit constraint imposed in vector $\mathbf{t}$ based on homography and the orthogonal property of matrix $\mathbf{R}$. Then a Two-step method is proposed to estimate the pair-wise $(\mathbf{R},\mathbf{t})$, where the implicit orthogonal constraint is imposed making the $\mathbf{R}$ solution closer to orthogonal.

Implicit orthogonal constraint

Following (2.7), we have

$$\mathbf{R} = \mathbf{G} + \mathbf{t}\cdot\mathbf{n}^\top = \begin{bmatrix} \mathbf{g}_1^\top + t_1\mathbf{n}^\top \\ \mathbf{g}_2^\top + t_2\mathbf{n}^\top \\ \mathbf{g}_3^\top + t_3\mathbf{n}^\top \end{bmatrix}$$

where $\mathbf{t} = [t_1\ t_2\ t_3]^\top$ and $\mathbf{g}_i^\top$ $(i = 1,2,3)$ are the three row vectors of $\mathbf{G}$. The three row vectors $\mathbf{g}_i^\top + t_i\mathbf{n}^\top$ $(i = 1,2,3)$ form an orthonormal basis of $\mathbb{R}^3$, i.e., we have

$$\begin{cases} (\mathbf{g}_i^\top + t_i\mathbf{n}^\top)(\mathbf{g}_i + t_i\mathbf{n}) = 1 \\ (\mathbf{g}_j^\top + t_j\mathbf{n}^\top)(\mathbf{g}_j + t_j\mathbf{n}) = 1 \\ (\mathbf{g}_i^\top + t_i\mathbf{n}^\top)(\mathbf{g}_j + t_j\mathbf{n}) = 0 \end{cases} \qquad (i,j \in [1,2,3];\ i \neq j)$$

Note that $\mathbf{g}_i^\top\mathbf{n} = \mathbf{n}^\top\mathbf{g}_i$ $(i = 1,2,3)$, we then have

$$\begin{cases} \mathbf{g}_i^\top\mathbf{g}_i + t_i^2\mathbf{n}^\top\mathbf{n} + 2t_i\mathbf{n}^\top\mathbf{g}_i = 1 & \text{(a)} \\ \mathbf{g}_j^\top\mathbf{g}_j + t_j^2\mathbf{n}^\top\mathbf{n} + 2t_j\mathbf{n}^\top\mathbf{g}_j = 1 & \text{(b)} \\ \mathbf{g}_i^\top\mathbf{g}_j + t_it_j\mathbf{n}^\top\mathbf{n} + t_i\mathbf{n}^\top\mathbf{g}_j + t_j\mathbf{n}^\top\mathbf{g}_i = 1 & \text{(c)} \end{cases} \quad (i,j \in [1,2,3];\ i \neq j) \qquad (2.11)$$

(2.11-a)$\times t_j/t_i$+(2.11-b)$\times t_i/t_j$ makes

$$\frac{t_i}{t_j}(\mathbf{g}_j^\top\mathbf{g}_j - 1) + \frac{t_j}{t_i}(\mathbf{g}_i^\top\mathbf{g}_i - 1) + 2(t_it_j\mathbf{n}^\top\mathbf{n} + t_i\mathbf{n}^\top\mathbf{g}_j + t_j\mathbf{n}^\top\mathbf{g}_i) = 0$$

Together with (2.11-c), we can eliminate the terms involving $\mathbf{n}$ in (2.11) and derive the equation (2.12) with a single unknown quantity $k_{ij} = t_i/t_j$. (2.12) is one of the

necessary conditions to guarantee the orthogonality of $\mathbf{R}$. Also note that (2.12) no longer involves the normal $\mathbf{n}$. As a consequence, less noise disturbance will be introduced for estimating the ratio of $\mathbf{t}$ elements.

$$2\mathbf{g}_i^\top \mathbf{g}_j = \frac{t_i}{t_j}(\mathbf{g}_j^\top \mathbf{g}_j - 1) + \frac{t_j}{t_i}(\mathbf{g}_i^\top \mathbf{g}_i - 1) \qquad (i, j \in [1,2,3];\ i \neq j) \tag{2.12}$$

Similarly, if we define $\mathbf{n} = [n_1\ n_2\ n_3]^\top$ and $\mathbf{G} = [\mathbf{g}'_1\ \mathbf{g}'_2\ \mathbf{g}'_3]$ with $\mathbf{g}'_i$ $(i = 1,2,3)$ the three column vectors of $\mathbf{G}$, we will have $\mathbf{R} = [\mathbf{g}'_1 + n_1\mathbf{t} \quad \mathbf{g}'_2 + n_2\mathbf{t} \quad \mathbf{g}'_2 + n_2\mathbf{t}]$ and can derive the equation (2.13), which is another implicit orthogonal constraint on matrix $\mathbf{G}$ and normal $\mathbf{n}$. (2.13) could be employed to examine the input data $\mathbf{G}_i$ and $\mathbf{n}_i$ $(i = 1,2,\ldots,P)$. Those severely violate this constraint should be rejected as outliers, making the estimation more robust to very noisy data.

$$2\mathbf{g}'^\top_i \mathbf{g}_j = \frac{n_i}{n_j}(\mathbf{g}'^\top_j \mathbf{g}'_j - 1) + \frac{n_j}{n_i}(\mathbf{g}'^\top_i \mathbf{g}'_i - 1) \qquad (i, j \in [1,2,3];\ i \neq j) \tag{2.13}$$

Two-step method

The proposed Two-step method is based on the implicit orthogonal constraint derived above. At the first step, we gather P such equations as (2.12) corresponding to $\mathbf{G}_i$ $(i = 1,2,\ldots,P)$ and compose simultaneous quadratic equations. Solving this problem by least square metric, we obtain the uniform ratio of the three elements of $\mathbf{t}$ vector $\mathbf{t}_1 : \mathbf{t}_2 : \mathbf{t}_3 = 1 : k_{21} : k_{31}$. Thus the original 3-DOF (Degree of Freedom) $\mathbf{t}$ vector is reduced to a single scale s as

$$\mathbf{t} = s \begin{bmatrix} 1 \\ k_{21} \\ k_{31} \end{bmatrix} \tag{2.14}$$

Based on (2.14), we then rewrite (2.9) as (2.15). At the second step, we solve the simultaneous linear equations generated by stacking P such equations as (2.15). Once s is estimated, vector $\mathbf{t}$ is readily computed by (2.14).

The Two-step method imposes the implicit orthogonal constraint (2.12) and (2.13) in the estimation explicitly, while keeping the problem linear. ($\mathbf{R}$,$\mathbf{t}$) estimated by this method not only conform to the homography geometry, but also satisfy the orthogonal constraint $\mathbf{R}^\top\mathbf{R} = \mathbf{I}$ much better. Still $\mathbf{R}$ is not perfectly

orthogonal and further orthogonal approximation is necessary. However, less deviation will be induced by the approximation, because $\mathbf{R}$ is much closer to its orthogonal approximation $\mathbf{R}'$.

$$\underbrace{\text{vec}(\mathbf{G})}_{9\times 1} = \text{vec}(\mathbf{R}) - \mathbf{n} \otimes \begin{bmatrix} 1 \\ k_{21} \\ k_{31} \end{bmatrix} s = \underbrace{\begin{bmatrix} \mathbf{I}_9 & -\mathbf{n} \otimes \begin{bmatrix} 1 \\ k_{21} \\ k_{31} \end{bmatrix} \end{bmatrix}}_{9\times 10} \underbrace{\begin{bmatrix} \text{vec}(\mathbf{R}) \\ s \end{bmatrix}}_{10\times 1} \tag{2.15}$$

2.3.3.2 Three-Step Method

By good external calibration, we mean that ($\mathbf{R}$,$\mathbf{t}$) should not only conform to the homography geometry, but also satisfy the orthogonal constraint $\mathbf{R}^\top\mathbf{R} = \mathbf{I}$. Though such a calibration result cannot be obtained completely by a linear method, it is possible to improve the two-step method through several additional linear optimization steps to further impose the orthogonal constraint of rotation.

As we know, in three dimensions a rotation can be defined by a single angle of rotation θ, and the direction of a unit vector $\mathbf{v} = [x, y, z]^\top$, about which to rotate. According to Euler's rotation theorem, the 3×3 rotation matrix $\mathbf{R}$ has one real eigenvalue equal to unity, and the unit vector $\mathbf{v}$ is the corresponding eigenvector, i.e.,

$$\mathbf{R}\mathbf{v} = \mathbf{v} \tag{2.16}$$

It follows from (2.5) and (2.16) that

$$\mathbf{G}\mathbf{v} = \mathbf{v} - \mathbf{t} \cdot \mathbf{n}^\top \cdot \mathbf{v} \tag{2.17}$$

If we know the matrix $\mathbf{G}_i, \mathbf{n}_i$ $(i = 1, 2, \ldots, P)$ and $\mathbf{t}$, vector $\mathbf{v}$ can be estimated by solving the linear equation (2.18), which is the accumulation of P such equations as (2.17).

$$\begin{bmatrix} \mathbf{G}_1 - \mathbf{I} + \mathbf{t} \cdot \mathbf{n}_1^\top \\ \vdots \\ \mathbf{G}_P - \mathbf{I} + \mathbf{t} \cdot \mathbf{n}_P^\top \end{bmatrix} \mathbf{v} = \mathbf{0} \tag{2.18}$$

According to Eckart-Young-Mirsky (EYM) theorem, the solution to (2.18), in matrix form as $\mathbf{B}\mathbf{v} = \mathbf{0}$, is the right singular vector of $\mathbf{B}$ associated with its smallest singular value.

To recover the rotation matrix, we further estimate the parameter θ based on the rotation representation (2.19), where $\hat{\mathbf{v}}$ indicates the 3×3 skew symmetric matrix corresponding to $\mathbf{v}$

$$\mathbf{R} = \mathbf{v}\mathbf{v}^\top + (\mathbf{I} - \mathbf{v}\mathbf{v}^\top)\cos\theta + \hat{\mathbf{v}}\sin\theta \tag{2.19}$$

In order to guarantee a linear optimization, we estimate two parameters $\cos\theta$ and $\sin\theta$ instead of the single θ and the constraint $\cos^2\theta + \sin^2\theta = 1$ is not imposed in the estimation. Experimental results show that the computed $\cos\theta$ and $\sin\theta$ basically satisfy this constraint.

At this step, we may retain the original result of vector $\mathbf{t}$, or we can refine it together with $\cos\theta$ and $\sin\theta$ while still keeping the optimization problem a linear one. In order to achieve robust results, we choose the latter scheme to estimate $\cos\theta$, $\sin\theta$ and vector $\mathbf{t}$ together by linear equation (2.20), which is derived from (2.7) and (2.19)

$$\begin{bmatrix}\mathrm{vec}(\mathbf{I} - \mathbf{v}\mathbf{v}^\top) & \mathrm{vec}(\hat{\mathbf{v}}) & -\mathbf{n} \otimes \mathbf{I}\end{bmatrix} \begin{bmatrix}\cos\theta \\ \sin\theta \\ \mathbf{t}\end{bmatrix} = \mathrm{vec}(\mathbf{G} - \mathbf{v}\mathbf{v}^\top) \tag{2.20}$$

By stacking P such equations, we have

$$\underbrace{\begin{bmatrix}\mathrm{vec}(\mathbf{I} - \mathbf{v}\mathbf{v}^\top) & \mathrm{vec}(\hat{\mathbf{v}}) & -\mathbf{n}_1 \otimes \mathbf{I} \\ \vdots & \vdots & \vdots \\ \mathrm{vec}(\mathbf{I} - \mathbf{v}\mathbf{v}^\top) & \mathrm{vec}(\hat{\mathbf{v}}) & -\mathbf{n}_P \otimes \mathbf{I}\end{bmatrix}}_{9P \times 5} \underbrace{\begin{bmatrix}\cos\theta \\ \sin\theta \\ \mathbf{t}\end{bmatrix}}_{5 \times 1} = \underbrace{\begin{bmatrix}\mathrm{vec}(\mathbf{G}_1 - \mathbf{v}\mathbf{v}^\top) \\ \vdots \\ \mathrm{vec}(\mathbf{G}_P - \mathbf{v}\mathbf{v}^\top)\end{bmatrix}}_{9P \times 1} \tag{2.21}$$

Based on the above description, the Three-step method is outlined as follows:

1. Use Two-step method to compute the initial estimation of $\mathbf{t}$
2. Estimate vector $\mathbf{v}$ based on (2.18) with $\mathbf{t}$ fixed
3. Estimate parameters $\cos\theta$, $\sin\theta$ and refine $\mathbf{t}$ together based on (2.21) with $\mathbf{v}$ fixed. With $\cos\theta$, $\sin\theta$ and $\mathbf{v}$ already computed, the rotation matrix $\mathbf{R}$ can be recovered by (2.19).

2.3.3.3 Non-linear Method

If we expect the estimated matrix $\mathbf{R}$ to be orthogonal without further orthogonal approximation, we should impose the constraint $\mathbf{R}^\top\mathbf{R} = \mathbf{I}$ explicitly in the maximum likelihood estimation and it turns out to be a constrained non-linear minimization problem

$$\min_{\mathbf{R},\mathbf{t}} \sum_i \left\| \mathbf{G}_i - \mathbf{R} + \mathbf{t} \cdot \mathbf{n}_i^\top \right\|_F \quad \text{subject to } \mathbf{R}^\top\mathbf{R} = \mathbf{I} \tag{2.22}$$

, where the optimum is in the sense of the smallest sum of Frobenius norms.

We may use the Lagrange multiplier to solve the constrained problem, but a better choice is to utilize the angle-axis representation of rotation. As mentioned before, a rotation matrix in three dimensions can be represented by a unit vector $\mathbf{v} = [x, y, z]^\top$ and an angle θ as

$$\mathbf{R}(\mathbf{v},\theta) = \begin{bmatrix} \cos\theta + (1-\cos\theta)x^2 & (1-\cos\theta)xy - (\sin\theta)z & (1-\cos\theta)xz - (\sin\theta)y \\ (1-\cos\theta)xy + (\sin\theta)z & \cos\theta + (1-\cos\theta)y^2 & (1-\cos\theta)yz - (\sin\theta)x \\ (1-\cos\theta)xz - (\sin\theta)y & (1-\cos\theta)yz + (\sin\theta)x & \cos\theta + (1-\cos\theta)z^2 \end{bmatrix} \tag{2.23}$$

We can substitute this compact representation to the minimization term of (2.22), and solve the non-linear problem with the Levenberg-Marquardt algorithm as implemented in Minpack [97]. The required initial guess of ($\mathbf{R}$,$\mathbf{t}$) can be obtained by the Two-step or Three-Step method described before. Experimental results show that matrix $\mathbf{R}$ estimated by the non-linear method is already orthogonal. Hence it avoids the problem caused by orthogonal approximation.

2.4 Accuracy Measure Based on Relative Deflection Angle

Measurement based on 3D reconstruction error [140] is widely used to evaluate the calibration accuracy. However, this metric needs the prior knowledge of the true 3D positions of test points which is usually unavailable in common multi-view capturing systems, especially for those composed of off-the-shelf cameras. On the other hand, many factors actually can contribute to a small reconstruction error besides the accurate calibration. For example, accurate detection of the control

points can lead to good measures by this metric, but it can be achieved by: (1) using lens with larger focal length so that the array of pixels focus on a smaller area of the scene; (2) reducing the distance between the scene and the cameras to take a close look; (3) directly increasing the image resolution. Besides, one can always increase the baseline between two cameras to reduce the triangulation uncertainty, which will make the reconstruction more accurate. In conclusion, this measurement depends very much on the actual system setup, such as camera focal length, baseline length, scene depth and image resolution. A good evaluation metric should be insensitive to these factors so as to give fair assessments to the calibration results of different camera systems under different working conditions. In [32], this metric is modified as the depth error from triangulation divided by the actual depth for the purpose of reducing the impact of scene depth. The factor of camera potential is further taken into account in the NSCE (Normalized Stereo Calibration Error) criterion [148] so that the focal length and resolution of the digital camera will not benefit the measurement. While all these metrics require the true point position to compute the reconstruction error, the wide baseline stereo setup can still result in good measures as long as the triangulation is involved. To address these problems, we propose to measure the calibration accuracy based on the relative deflection angle (RDA). It is defined as

$$\text{RDA} = \text{E}[\theta_{err}]/\theta_{sys} \tag{2.24}$$

where θ_{err} denotes the deflection angle caused by calibration inaccuracy and θ_{sys} denotes the deflection angle due to system error. In a stereo setup, one camera is chosen as the reference, e.g., the camera 2 is chosen as the reference camera in Figure 2.3. Both θ_{err} and θ_{sys} are computed for the reference camera.

The error angle θ_{err} is defined as the included angle $\angle\mathbf{QC_2M}$ in Figure 2.3, where C2 represents the pinhole of camera 2, i.e., the origin of the C2 coordinate system; $\mathbf{Q}$ denotes the 3D control point and $\mathbf{M}$ denotes its 2D projection on the image plane of camera 2. By ideal projection, the three points should be collinear. In practice, a deflection angle $\angle\mathbf{QC_2M}$ will exist due to the calibration error. The projection ray $\mathbf{C_2Q}$ is recovered by chaining the transform $(\mathbf{R}_1, \mathbf{t}_1)$ from $\mathbf{Q}$ to $\mathbf{C_1}$ (the origin of C1 coordinate system) and the transform $(\mathbf{R},\mathbf{t})$ from C1 to C2. The two transforms are estimated by the external camera

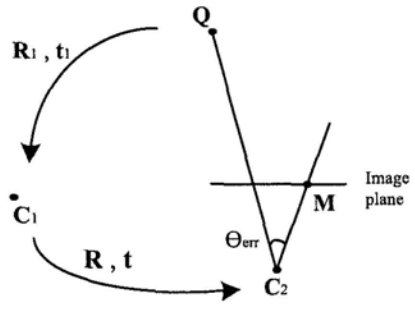


Figure 2.3: Definition of θ_{err}.

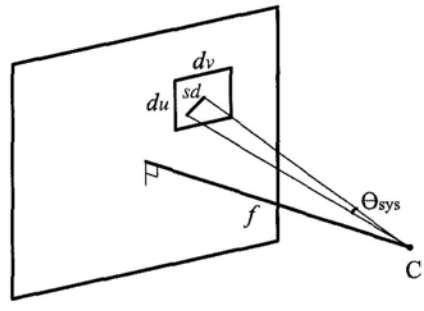


Figure 2.4: Definition of θ_{sys}.

calibration. Before computing the backward projection ray $\mathbf{C_2M}$, un-distortion should first be performed according to the distortion parameters estimated. The position of $\mathbf{M}$ in the image plane is then determined by corner detection, and the 3D coordinate of $\mathbf{M}$ with respect to C2 can be easily figured out as we have the intrinsic parameters. We can see that the computation of $\angle\mathbf{QC_2M}$ involves the intrinsic matrix, the distortion parameters and the relative transform between the two cameras. Therefore the deflection angle θ_{err} integrates the errors from both internal and external calibration. The mean value $\mathrm{E}[\theta_{err}]$ over all the control points can be reasonably utilized to reveal the inaccuracy of stereo camera calibration.

The digitization noise of camera may cause some inherent uncertainty of the projection [148] which is modeled by the system deflection angle θ_{sys} in this thesis. Consider a pixel rectangle of size $d_u \times d_v$ as shown in Figure 2.4, uniform digitization noise in the rectangle has a standard deviation $sd = \sqrt{(d_u^2 + d_v^2)/12}$. This digitization error corresponds to the uncertainty of the projection of a 3D point onto the image plane, which can be measured as the angle of the deflected projection rays, as shown in Figure 2.4. Let f be the camera focal length and assume that the focal length has a much larger scale than the pixel size. The system deflection angle is defined to be

$$\theta_{sys} \approx \tan(\theta_{sys}) \approx \frac{sd}{f} = \frac{\sqrt{(d_u^2 + d_v^2)/12}}{f} = \sqrt{(f_u^{-2} + f_v^{-2})/12} \tag{2.25}$$

where $f_u = f/d_u$ and $f_v = f/d_v$. We can see that θ_{sys} is associated with the image resolution and the camera focal length.

The proposed RDA metric (2.24) is much less sensitive to the digital image resolution and the camera focal length because these two factors have similar effect on both the numerator and denominator. For example, if the cameras with high resolution are used, the digitization uncertainty θ_{sys} and the resulting inherent error of corner detection will be reduced in a similar way. Accurate corner detection means less data noise, thus the calibration error θ_{err} will also decrease accordingly, keeping the RDA measurement unchanged. Thus the RDA metric still provides a fair and meaningful calibration evaluation regardless of the changing camera parameters. Moreover, both the scene depth and the stereo triangulation will not affect the two rays $\mathbf{C_2Q}$ and $\mathbf{C_2M}$, and of course the camera characteristics. As a result, these factors should have little impact on the RDA metric as well. Note that taking a close look does not reduce the uncertainty of corner detection on the image plane measured in Euclidean space, i.e., will not benefit the deflection angle θ_{err}. Above analysis is based on the ideal case that the digitization uncertainty is the only concern for corner detection. However, in practice there are many other factors that affect the corner detection, including the detection algorithm itself. The actual performance of RDA metric will be discussed in sub-section 2.5.2.

2.5 Experimental Results

2.5.1 Simulation on (R,t) Estimation

This simulation is to evaluate the performance of different ($\mathbf{R}$,$\mathbf{t}$) estimation algorithms, especially their sensitivity to data noise. We consider the scenario that two cameras capture a planar pattern with 9×12 50mm square corners. The cameras are assumed to be distortion-free and their intrinsic matrices are specified to be the same with two of our digital cameras which are internally calibrated by Zhang's method. Total 30 different plane poses are used for calibration, where the 30 different translations and rotations from the plane to the first camera are manually specified. Then the second camera is translated and rotated by $\mathbf{t} = [-1010, -110, 160]^\top$ and $\mathbf{R} = [0.9397\ -0.0179\ 0.3416;\ 0\ 0.9986\ 0.0523;\ -0.3420\ -0.0492\ 0.9384]$ with respect to the first camera such that the two cameras are

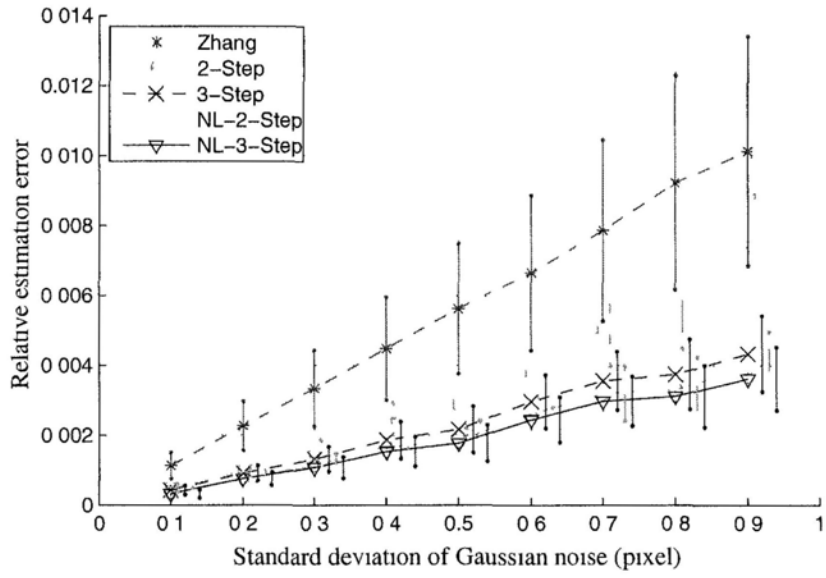


Figure 2.5. Relative estimation error $\|\mathbf{R} - \widetilde{\mathbf{R}}\|/\|\mathbf{R}\|$.

widely separated and can simultaneously capture the plane. The translation $\mathbf{t}$ and rotation $\mathbf{R}$ are referred to as the ground truth for later evaluation. We then add zero mean uncorrelated Gaussian noise to the image projections of the control points (square corners on the plane) from a standard deviation of 0.1 pixels up to 0.9 pixels in step of 0.1. The relative estimation errors $\|\mathbf{t} - \widetilde{\mathbf{t}}\|/\|\mathbf{t}\|$ and $\|\mathbf{R} - \widetilde{\mathbf{R}}\|/\|\mathbf{R}\|$ are used to measure the calibration accuracy, where $(\widetilde{\mathbf{R}}, \widetilde{\mathbf{t}})$ are the estimates and $(\mathbf{R},\mathbf{t})$ are the ground truth. The results of 50 simulations are summarized in Figure 2.5 and Figure 2.6, where the curves show the mean values of errors and the length of vertical line segments indicates the standard deviation of error at each noise level. The vertical line segments are horizontally shifted for clear display. Non-linear methods initialized by two-step and three-step methods are indicated by NL-2-Step and NL-3-Step, respectively.

The results of Zhang's method [156] are also presented for comparison. Note that Zhang's method uses a non-linear minimization to estimate the intrinsic parameters of a camera together with the rotations and translations of different plane poses relative to the camera. Here, the external calibration is to recover the fixed rotation and translation $(\mathbf{R},\mathbf{t})$ between two cameras rather than the

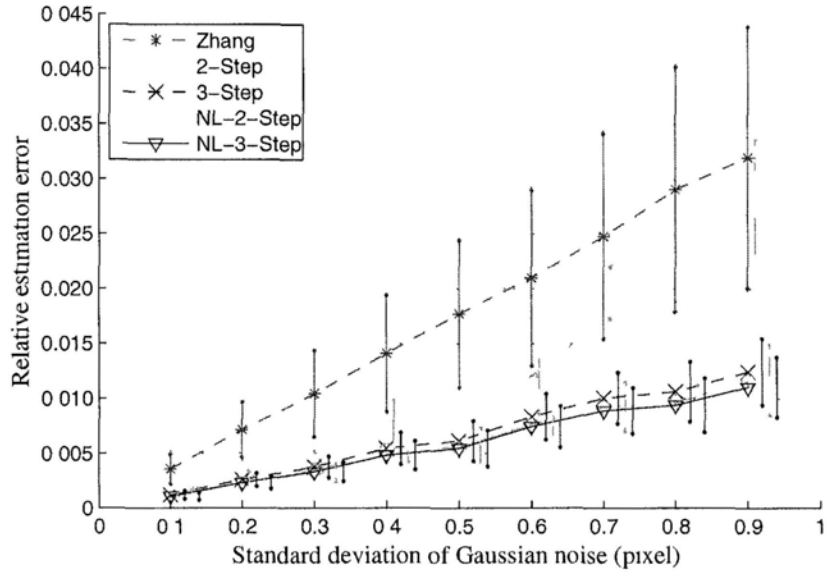


Figure 2.6: Relative estimation error $\|\mathbf{t} - \widetilde{\mathbf{t}}\|/\|\mathbf{t}\|$.

plane poses. To this end, we can compute a $(\mathbf{R},\mathbf{t})$ estimate for each plane pose by simply chaining the rotation and translation of the plane relative to camera 1 and those relative to camera 2. In our simulation, we have tried all 30 different plane poses independently and the results shown in Figure 2.5 and 2.6 present their overall performance. In comparison, the algorithms proposed in this thesis compute a single estimate of $(\mathbf{R},\mathbf{t})$ by efficiently combining all the plane poses.

Figure 2.5 and Figure 2.6 clearly show that our methods, including two-step, three-step and non-linear methods, perform much better than simple chaining of the plane poses obtained by Zhang's method, in terms of both the accuracy and the stability. In addition, the results also demonstrate that combining multiple stereo images in an efficient way does benefit the external camera calibration. By imposing the implicit orthogonal constraint, two-step method can already provide a calibration much better than Zhang's chaining. The performance is further improved by the three-step method and it should be emphasized that such an improvement is significant for highly noisy data. As expected, the non-linear method initialized by three-step achieves the best calibration results, closely followed by the non-linear method initialized by two-step and the three-step method. We can

see that, compared with two-step, the three-step method is a better initialization for non-linear method, and its performance is already very close to that after the non-linear optimization. If computation is a concern, the three-step method will be a nice approximation of the non-linear method since it only involves solving a few linear equations.

2.5.2 Test on RDA Metric

The proposed RDA metric is tested on different camera setups using real data. Two cameras are placed focusing on the checkerboard pattern with 9×12 50mm square corners. In the first experiment, the baseline of the two cameras is kept 1m fixed. We change the distance of the pattern from the cameras as well as the pattern orientations. The distance between the pattern and the baseline midpoint is selected as roughly 2m, 2.5m, 3m and 3.5m. In the second experiment, the camera setup is varied in terms of the baseline length, which is roughly set to 0.6m, 0.8m, 1m and 1.2m. Accordingly, the distance between the pattern and the baseline midpoint is slightly adjusted such that the distances from the pattern to the two cameras are the same and kept unchanged. The pattern is placed in different orientations to obtain multiple stereo images. In the last experiment, we change the camera focal length while keeping the baseline length and the viewing distance fixed as 1m and 2m, respectively. The focal length of the reference camera is adjusted to 50mm, 45mm, 40mm and 35mm. For all three experiments, the non-linear method initialized by three-step is applied to estimate the external parameters.

For comparison, we apply another error metric to measure the calibration accuracy, in addition to RDA. Let $\{\mathbf{x}^1\}$ and $\{\mathbf{x}^2\}$ be the two set of original control points detected in the first and second views, respectively. We then employ the optimal triangulation method [51] to compute two set of image points $\{\hat{\mathbf{x}}^1\}$ and $\{\hat{\mathbf{x}}^2\}$ that are consistent with the estimated fundamental matrix and closest to the original control points. The mean square distance MSD defined in (2.26) will be used as the second error metric to measure the calibration results, where N is

Viewing distance (m)	RDA	Change ratio of RDA	MSD	Change ratio of MSD
2.0	0.9173	-	0.0150	-
2.5	0.9862	0.0751	0.0267	0.7800
3.0	1.1438	0.1598	0.0435	0.6292
3.5	1.4170	0.2389	0.0860	0.9770

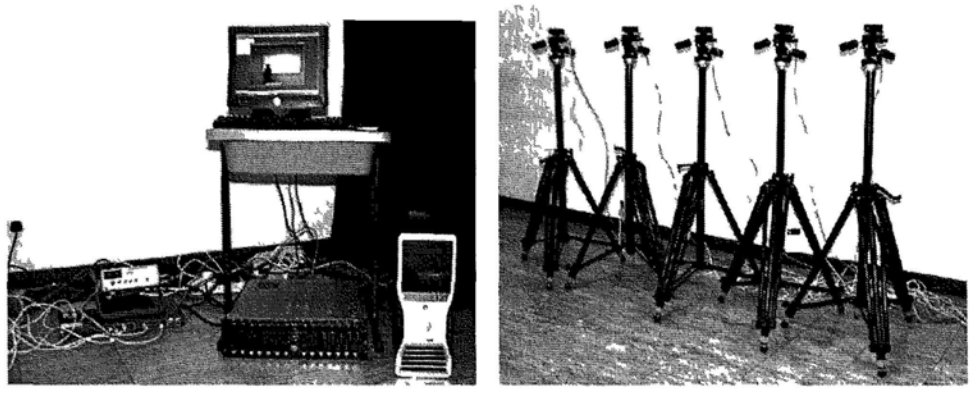
Table 2.1: Accuracy measures for different viewing distances

Baseline length (m)	RDA	Change ratio of RDA	MSD	Change ratio of MSD
1.2	0.9150	-	0.0131	-
1.0	0.9173	0.0025	0.0150	0.1450
0.8	0.9395	0.0242	0.0181	0.2093
0.6	0.9531	0.0145	0.0228	0.2547

Table 2.2: Accuracy measures for different baseline lengths

Focal length (m)	RDA	Change ratio of RDA	MSD	Change ratio of MSD
50	0.9173	-	0.0150	-
45	0.9643	0.0512	0.0194	0.2933
40	1.0332	0.0715	0.0240	0.2371
35	1.0824	0.0476	0.0277	0.1541

Table 2.3: Accuracy measures for different focal lengths



(a) The control units computer, switch, synchronizer and harddisk array

(b) Five Prosilica GC650C cameras

Figure 2.7: Our multiview system.

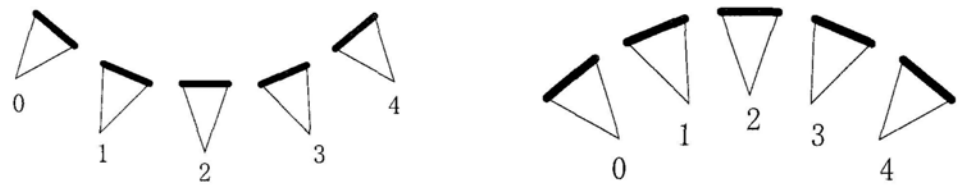
the total number of control points used in calibration.

$$\text{MSD} = \frac{1}{2N}\sum_{k=1}^{N}\left(\left\|\hat{\mathbf{x}}_k^1 - \mathbf{x}_k^1\right\|^2 + \left\|\hat{\mathbf{x}}_k^2 - \mathbf{x}_k^2\right\|^2\right) \tag{2.26}$$

Experimental results on different camera setups are summarized in Table 2.1, Table 2.2 and Table 2.3. To see the change of the RDA measures, we compute the change ratio as $(\text{RDA}_i - \text{RDA}_{i-1})/\text{RDA}_{i-1}$, where RDA_i and RDA_{i-1} indicate two successive RDA measures when the camera setup is changed. Similarly we compute the change ratio of MSD. From Table 2.2 and Table 2.3, we can observe that the RDA metric presents good invariance to the variation of baseline length and focal length, whereas the MSD measures benefit a lot from small digitization uncertainty (large focal length) and accurate triangulation (wide baseline). In case of different viewing distances in Table 2.1, though there is a clear increasing trend of RDA as the viewing distance increases, the RDA measures are still relatively stable compared with the MSD results.

2.5.3 Multi-camera Calibration

In this experiment, the proposed pair-wise $(\mathbf{R},\mathbf{t})$ estimation methods are applied to calibrate a multi-camera system, and the calibration results are then evaluated



(a) (b)

Figure 2.8: Different setups of multi-camera system.

by the proposed RDA metric. For comparison, simple chaining of Zhang's results [156] and Svoboda's multi-camera calibration method [133] are applied as well.

Real images are captured by our multiview system composed by five Prosilica GC650C cameras, as shown in Figure 2.7. The intrinsic parameters for the five cameras are estimated individually beforehand using Zhang's method. Our task now is to recover the external relationship among different cameras. To this end, A checkerboard pattern with 9×12 50mm square corners is used for providing control points. Undistortion is performed on the captured images before the detection of control points. Two different system setups are tested as sketched in Figure 2.8. In the first case 2.8(a), the five cameras are placed focusing on the same scene. Tested calibration methods include three-step method (3-Step), non-linear method initialized by three-step (NL-3-Step), chaining of Zhang's result (Zhang) and Svoboda's multi-camera calibration method (Svoboda). In the second case 2.8(b), the five cameras are placed looking at different scenes, only two adjacent cameras have sufficient common FOV. In this case, Svoboda's method is inapplicable, hence we only compare the results of 3-Step, NL-3-Step and Zhang. In both cases, the cameras are indexed from 0 to 4 sequentially, thus there are totally four neighboring pairs: (0-1), (1-2), (2-3) and (3-4). For each pair, 30 different plane poses are captured for calibration.

As we have neither the ground truth of the external parameters nor those of the control points w.r.t the camera coordinate system, we apply the proposed RDA metric to measure the calibration errors. To investigate the error introduced by the chaining of pair-wise $(\mathbf{R},\mathbf{t})$ estimation, we emphasize the calibration results of camera pairs: (0-2), (0-3) and (0-4). For methods of 3-Step, NL-3-Step and

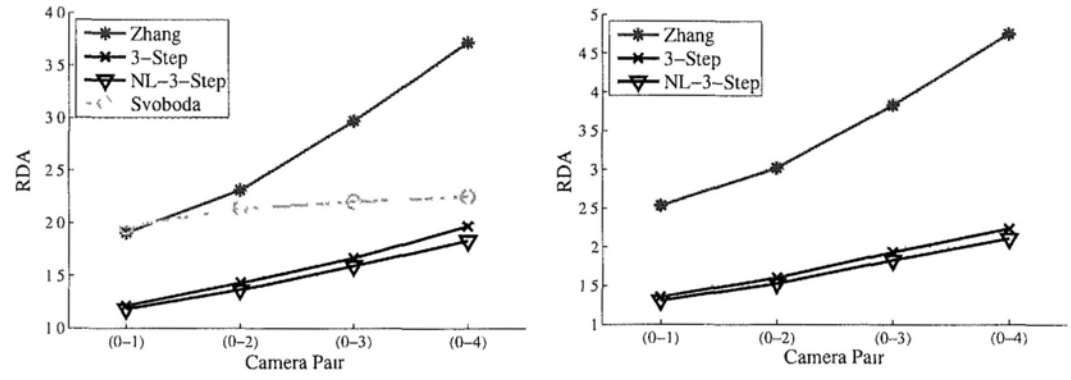


Figure 2.9: RDA results of different multi-camera calibration methods.

Zhang, (**R**,**t**) between (0-2), (0-3) and (0-4) pairs are computed by chaining the (**R**,**t**) results across the neighboring pairs (0-1), (1-2), (2-3) and (3-4).

The results of different calibration methods for camera setup 2.8(a) and 2.8(b) are shown in Figure 2.9. Two important observations should be noted. First, we can see from both Figure 2.9(a) and 2.9(b) that 3-Step and NL-3-Step achieve much better RDA measures than Zhang for (0-2), (0-3) and (0-4) camera pairs, which demonstrates that accurate pair-wise (**R**,**t**) estimation does help suppress the error accumulation of transform chaining. Second, as shown in Figure 2.9(a), when few transform chainings are performed, 3-Step and NL-3-Step methods outperform Svoboda's method in terms of calibration accuracy. However, we can see a trend that the calibration errors of 3-Step and NL-3-Step will exceed that of Svoboda when more chainings are involved. The reason is that Svoboda's method does not have the problem of error accumulation in transform chaining. In addition, when the cameras do not focus on the same scene points, e.g., setup 2.8(b), traditional methods such as Svoboda's are not applicable, while our method still works. This shows the flexibility and universality of pair-wise chaining.

2.6 Conclusion

In this chapter, we present a convenient and efficient method to calibrate a typical multi-camera system. Relative relationship between neighboring cameras is

recovered by the proposed pair-wise pose estimation method. Three different algorithms are proposed based on homography. By imposing the implicit orthogonal constraint, Two-Step method can already obtain accurate calibration results. The performance can be further improved by the Three-step and non-linear methods. As the orthogonal constraint is fully imposed in the non-linear method, it achieves the best calibration results in terms of robustness to data noise, as demonstrated by the simulation results. Our Experiments also show that multiple cameras can be reliably registered by chaining the accurate pair-wise poses estimated by the Three-Step and Non-linear methods. Such a way is more flexible and universal than the traditional multi-camera calibration methods, where all the cameras are required to capture the same control points. In case that there is no ground truth of the coordinates of 3D control points relative to the cameras, which is usually the case for a common multi-view capturing system, the RDA metric proposed in this thesis can be employed to provide a fair and meaningful evaluation of the calibration results for different cameras under different working conditions.

Chapter 3

Scale and Affine Invariant Fan Features

3.1 Introduction

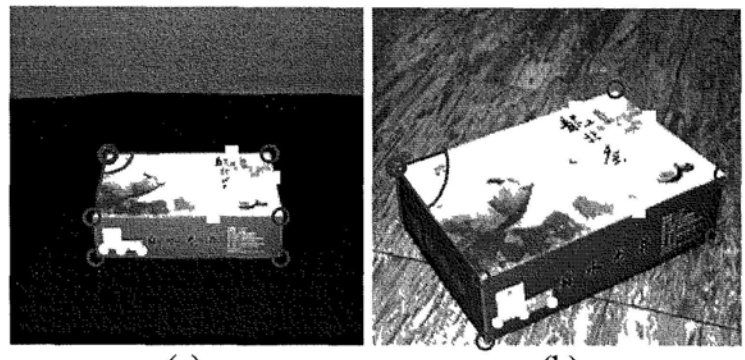
Local image features have proven to be very successful in wide baseline matching and object recognition [155] as well as many other applications. Their robustness to partial visibility allows for successful matching even in severe cluttered scenes. Their good discriminative property provides high confidence in recognition. Basically the feature-based schemes consist of two steps. First, the keypoints and their associated support regions are extracted from the image. Together they are referred to as *features*. Second, the descriptors are composed to summarize the features' appearance such as the shape and texture. For extensive investigation and comparison on feature detectors and descriptors, one can refer to [95]; [93]. The major problem of designing local features is how to obtain the invariance under different viewing conditions.

There is a considerable body of previous research on scale invariant features. In the early eighties, Crowley et al. [22]; [23] proposed to search for local extrema in the 3D scale-space representation. A local 3D extremum, (x, y, σ), in the scale space indicates a local feature with the keypoint located on (x, y) and the region extent (window size) determined by the scale parameter σ. In [77], Lindeberg proposed a systematic methodology for automatic scale selection. The basic idea is to select the characteristic scales, for which a given function attains extrema

over scales. The scale is characteristic in the sense that it responds to some salient signal change in the image and consequently can be repeatedly detected under different viewing conditions. Lindeberg proved that the scale-normalized Gaussian derivatives are good choices to compute the multi-scale function. Specifically, he suggested to use the scale-normalized Laplacian-of-Gaussian (LoG) to detect blob-like features. Later, Lowe [79] proposed the Difference of Gaussian (DoG) as the approximation of scale-normalized LoG to accelerate the computation of scale-space representation. In detailed experimental comparisons, Mikolajczyk [91] found that the scale-normalized LoG produces the most stable features compared to a range of other Gaussian derivative functions, such as squared gradient, Hessian and Harris corner function. Actually, a number of feature detectors [91], [92], [80] have adopted the scale-normalized LoG to select the characteristic scales. Other methods like MSER (Maximally Stable Extreme Regions) [87], EBR (Edge Based Region) and IBR (Intensity Based Region) [111] use different approaches to achieve scale invariance, yet the similar idea is using the salient intensity changes to indicate the characteristic local structures. Kadir et al. [62] proposed a different scale selection method, where local complexity is used instead as a measure of saliency and the salient scale is selected at the entropy extremum of the local descriptors.

To achieve rotation invariance, the common method is to describe the features using some rotationally invariant image measures, such as the generalized moments [96], the local jets [65] and RIFT [70]. In Lowe's SIFT [79], the free rotation is determined by estimating the dominant gradient orientation.

As an important step towards viewpoint invariance, affine invariance is highly desired for local features. Actually affine transformation is sufficient to locally model the image distortion arising from viewpoint changes, provided that (1) small surface patches can be thought of as being comprised of coplanar points; (2) perspective effect can be ignored at a local scale. In the mid nineties, Lindeberg et al. [78] developed a method to detect blob-like affine features in the context of shape from texture. It explores the properties of the second moment matrix and iteratively estimates the affine transformation of local patterns. This shape estimation method was later used for matching and recognition by Baumberg [9]. He used a multi-scale Harris detector to extract the keypoints and then



(a) (b)

Figure 3.1: What kind of extra keypoints can be matched by using Fan features?

employed the iterative procedure proposed by Lindeberg to adapt the shape of the point neighborhood to the local image structure. Mikolajczyk and Schimid [92] went a step further by iteratively modifying the location, scale and the neighborhood of a keypoint, such that both the keypoint and its associated support region are extracted in an affine invariant way. Apart from the second moment matrix, the covariance matrix (or region moments) is also widely used for affine invariant image normalization [120], as is employed by [87]; [100]; [41] to cope with geometric deformation introduced by viewpoint change.

However, the basic assumption for affine invariant features [95] does not hold for the keypoints located on or near the object boundaries. Conventional methods such as SIFT [79], Harris & Hessian Affine [92] and MSER [87] probably fail to match these keypoints because the point neighborhood cannot be modeled by a single planar surface due to depth discontinuity. Figure 3.1 gives an example of these keypoints such as the 3D corners and junctions (red circle), the keypoints along the boundaries (golden square) and the keypoints close to the boundaries (green dots). Note that, for those "green dots", conventional methods may adapt their support regions to small or highly deformed ones that do not cross the surface boundaries. However, small regions are usually not distinctive enough for reliable matching, and the highly deformed regions basically have low repeatability of detection under significant viewpoint changes. Therefore, it is difficult for conventional methods to match these "green dots". In [127], SIFT has been improved by incorporating the object boundary information to guide anisotropic

smoothing. The "green dots" can now be saved because the background clutter can be eliminated providing the accurate object boundaries. However, in practice it is nontrivial to obtain the object boundaries, especially from a single image. This is why in [127] stereo disparity map is employed, which, however, cannot be obtained as the prior information in applications such as wide baseline matching.

The basic idea to address the problem of surface discontinuity is straightforward. That is to divide the keypoint neighborhood into multiple sub-regions, each of which can now be reasonably assumed to represent a planar surface or just background. The sub-regions are described separately and all attached to the keypoint as independent signatures. As long as one of them exists in both views and can be matched successfully, the correspondence of the keypoints can be established accordingly. This is illustrated in Figure 3.1, where the two upper-left box corners in image (a) and (b) can be matched according to the corresponding upper box surfaces bounded by the red lines and the blue arcs. Similar ideas are shared by a few works, including 3D singularity [144] [145], EBR [141], Edge-based feature [94] and Edge descriptor [90]. The major differences lie in (1) how to select the anchor points with high repeatability; (2) how to divide the point neighborhood into sub-regions that represent meaningful surfaces; (3) how to make the sub-regions scale and affine invariant so that they can be consistently extracted in different views.

Regarding (1), 3D singularity [144]; [145] and EBR [141] only aim at the well formed edge corners and junctions like the "red circles" in Figure 3.1, which are salient as argued in [111] [145] but are rare in natural images. On the contrary, edge features [94]; [90] focus on extracting the keypoints along the boundaries such as the "golden squares" in Figure 3.1. Edge-based feature [94] selects edge points as keypoints as long as the LoG filter detects characteristic scales. As a result, the features may be duplicated and not distinctive enough. Edge descriptor [90] chooses anchor points along the edge where the scale envelopes attain their extrema. The extrema are salient and stable under viewpoint changes, but the computation of scale envelope highly relies on the continuity of edges which is difficult to guarantee in different images, and hence may hinder the features' repeatability. In comparison, we propose a unified framework to extract and match both the edge corners and junctions and other salient points along the

edges. The keypoints are selected from the edges that are efficiently and carefully detected to favor accurate surface boundaries. The repeatability of keypoints is guaranteed by a multi-scale selection scheme where the edges are not involved.

Regarding (2) and (3), the method of region extraction in [144]; [145] totally relies on the complete and straight edges that are usually difficult to detect in natural images. The EBR feature [141] is more practically designed by incorporating both edges and textures to detect scale and affine invariant regions. Yet the continuity of edges is still required for feature extraction. As a consequence, the matched EBR features are found to be quite limited. On the other hand, edge features [94]; [90] extract only scale invariant half-regions. In [94], a single LoG scale is selected for both the two half-regions, which is not reasonable because the two regions are supposed to represent different surfaces with independent extents (or one of them is background). The scale selection is improved in [90], where the two sides divided by the edge can have different LoG scales. However, continuous edges are required again to guide the scale selection. In this chapter, we propose to divide the point neighborhood into multiple regular fan sub-regions, namely Fan features, by a method of edge association which does not rely on the continuity and completeness of edges. To achieve scale invariance for each Fan feature, we propose the Fan Laplacian of Gaussian (FLOG) filter to select its characteristic scales. Support for FLOG is given in terms of theoretical investigation and real image experiments. To cope with geometric deformation, affine normalization is further applied to the Fan features, where the affine shape is diagnosed from the mirror-predicted surface patch. This in general gives us a better shape estimation result than the traditional way. Note that both the scale selection and the affine normalization are based on textures, rather than edges.

Finally, the scale and affine invariant Fan features are described by the Fan-SIFT, an extension of the famous SIFT descriptor. Fan grids are carefully designed to replace the square grids used in SIFT. Strong gradients arising from the region boundaries are efficiently suppressed by a boundary mask. In addition, local Gaussian weighting is introduced to each fan grid, to reduce the boundary effect of strong gradients shifting across the neighboring grids.

The remaining parts of this chapter are organized as follows. Section 3.2 describes the FLOG based scale selection method. Section 3.3 presents the method

to extract scale and affine invariant Fan features. Section 3.4 introduces the Fan-SIFT descriptor and Section 3.5 discusses the matching strategy based on Fan features. The experimental results are given in Section 3.6 and Section 3.7 concludes the chapter.

3.2 Automatic Scale Selection By FLOG

In order to achieve scale invariance for fan sub-regions, a novel automatic scale selection method is proposed based on FLOG. In this section, we first give the definition of the FLOG kernel and prove its transformation property under uniform scaling, which is emphasized in [77] as the fundamental requirement on a scale selection mechanism. We then describe the FLOG based scale selection method and demonstrate its feasibility using some simple image patterns.

3.2.1 Scaling Property of FLOG Response

The standard LoG kernel in the polar coordinate system is defined in (3.1), where σ is the standard deviation of Gaussian.

$$LoG(r;\sigma) = \frac{r^2 - 2\sigma^2}{2\pi\sigma^6}\exp(-\frac{r^2}{2\sigma^2}) \tag{3.1}$$

The FLOG kernel can be interpreted as the standard LoG normalized by a factor of σ^2 and bounded within a fan domain $D = \{(r,\theta)|r \geq 0,\ \theta_1 \leq \theta \leq \theta_2\}$. Formally it is defined in (3.2).

$$FLOG_{\theta_1\theta_2}(r,\theta;\sigma) = \begin{cases} \sigma^2 LoG(r;\sigma) & \theta_1 \leq \theta \leq \theta_2 \\ 0 & \text{otherwise} \end{cases} \tag{3.2}$$

Figure 3.2 shows an example of the FLOG kernel. We can see that FLOG preserves the isotropy of LoG within the fan domain. When the fan domain expands to the circle domain, the FLOG kernel becomes exactly the same with the scale-normalized LoG [77]. In this sense, FLOG is an extension of the scale-normalized LoG. Next, we investigate the behavior of the integration of FLOG and input signal under uniform scaling.

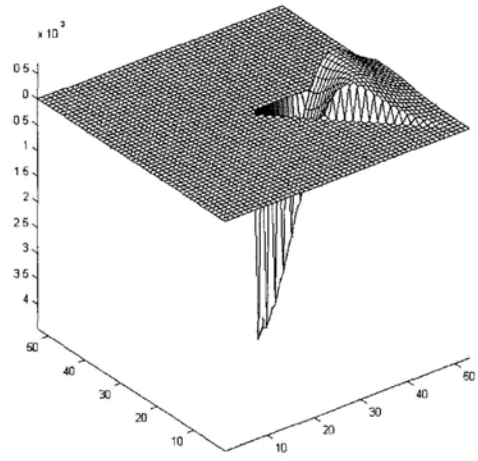


Figure 3.2: FLOG kernel with included angle equal to 45°.

Consider two 2D signals f and f', where f' is obtained by uniformly scaling the spatial variables of f, i.e.,

$$f'(x', y') = f(x, y) \tag{3.3}$$

$$[x'\ y']^\top = s[x\ y]^\top \tag{3.4}$$

Accordingly, in the polar coordinate system, it holds that

$$f'(r', \theta') = f(r, \theta), \quad r' = sr, \quad \theta' = \theta \tag{3.5}$$

Suppose that the scale parameters are transformed by the same factor in the two domains, i.e.,

$$\sigma' = s\sigma \tag{3.6}$$

According to (3.1) and (3.2), we then have

$$FLOG_{\theta_1\theta_2}(r', \theta'; \sigma') = \frac{1}{s^2} FLOG_{\theta_1\theta_2}(r, \theta; \sigma) \tag{3.7}$$

By equations (3.5), (3.6) and (3.7), we can derive that

$$\begin{aligned}
&\iint_{D'} f'(x',y') \cdot FLOG_{\theta_1\theta_2}(x',y';\sigma')\,\mathrm{d}x'\,\mathrm{d}y' \\
=\ &\int_{\theta_1}^{\theta_2}\int_0^{\infty} f'(r',\theta') \cdot FLOG_{\theta_1\theta_2}(r',\theta';\sigma')r'\,\mathrm{d}r'\,\mathrm{d}\theta' \\
=\ &s^2\int_{\theta_1}^{\theta_2}\int_0^{\infty} f'(r',\theta') \cdot FLOG_{\theta_1\theta_2}(r',\theta';\sigma')r\,\mathrm{d}r\,\mathrm{d}\theta \\
=\ &\iint_{D} f(x,y) \cdot FLOG_{\theta_1\theta_2}(x,y;\sigma)\,\mathrm{d}x\,\mathrm{d}y
\end{aligned}$$

,where

$$D = \{(r,\theta)|r \geq 0,\ \theta_1 \leq \theta \leq \theta_2\}\,,\ D' = \{(r',\theta')|r' \geq 0,\ \theta_1 \leq \theta' \leq \theta_2\}$$

It is rewritten as

$$\iint_{D'} f'(x',y') \cdot FLOG_{\theta_1\theta_2}(x',y';\sigma')\,\mathrm{d}x'\,\mathrm{d}y' = \iint_{D} f(x,y) \cdot FLOG_{\theta_1\theta_2}(x,y;\sigma)\,\mathrm{d}x\,\mathrm{d}y \tag{3.8}$$

This means that the integration of FLOG and the input signal, called *FLOG response*, is equal in the two domains, provided that the spatial positions and the scale parameters are related according to (3.4) and (3.6). Let's look at the FLOG response as a function of the scale parameter σ. Based on above derivation, if the image pattern is rescaled by a constant scaling factor s, then the scale at which the FLOG response assumes its extrema will be multiplied by the same factor. Here, to guarantee the scale invariance, σ^2 is introduced to normalize the FLOG kernel, which is consistent with the scale-normalized LoG [77].

3.2.2 FLOG-based Scale Selection

As suggested in (3.8), the FLOG response can commute with the size change. This gives us a solution to detect the characteristic scales for a given sub-region attached to a keypoint. First, according to the fan shape of the sub-region, we choose a FLOG kernel with two appropriate directions θ_1 and θ_2. We then compute the multi-scale FLOG response centered on the keypoint, i.e., the corner of the fan sub-region. Finally, the extrema of FLOG response are detected and

the corresponding scale parameters are chosen as the characteristic scales of the fan sub-region. Ideally, if the image pattern within the sub-region undergoes uniform scaling, the characteristic scales selected by this method before and after the scaling will indicate consistent image contents.

Intuitively, the characteristic scales can be repeatedly detected because they respond to salient signal changes. To see how the extrema of FLOG response capture the salient signal changes, let us consider a simple fan step signal:

$$f_s(r,\theta) = \begin{cases} 1 & 0 \le r \le r_0,\ \theta_1 \le \theta \le \theta_2 \\ 0 & \text{otherwise} \end{cases}$$

The extrema of its FLOG response can be found as follows:

$$\frac{\partial}{\partial\sigma}\int_{\theta_1}^{\theta_2}\int_0^{\infty} f_s(r,\theta)\cdot FLOG_{\theta_1\theta_2}(r,\theta;\sigma)r\,\mathrm{d}r\,\mathrm{d}\theta = 0 \quad \Rightarrow \quad \sigma = r_0/\sqrt{2} \tag{3.9}$$

We can see that the scale parameter σ that makes the FLOG response attain its extremum is related to the distance from the step signal change, i.e., r_0, by a factor of $1/\sqrt{2}$.

In Figure 3.3, the scale selection method is tested using some fan image patterns. For comparison, both the FLOG kernel and the scale-normalized LoG kernel are applied. Suppose that we are only concerned with the extent of the fan patterns. Thus we compute the multi-scale responses using the two kernels centered in the fan corner. The scale parameter is set as $\sigma = 1.2^{k-1}$ $(k = 1, 2, \ldots, 20)$. The scales detected by LoG and FLOG are represented by the red circles and green arcs, with their radius equal to the detected σ. The two directions for FLOG kernel are specified manually, as indicated by the two blue lines.

As we can see in Figure 3.3(a) and 3.3(d), when no clutter exists around the fan corner, both two kernels can correctly reflect the extents of the fan patterns. Specifically, the scales detected by the two kernels are exactly the same, roughly $1/\sqrt{2}$ of the fan radius. However, when other patterns coexist, LoG attempts to find some uniform scales for the whole point neighborhood, leading to undesired or inaccurate scales as shown in Figure 3.3(b) and 3.3(e). In comparison, FLOG only concerns the given sub-region. Signal changes elsewhere will never affect the scale selection for the target sub-region. Therefore identical scales are repeatedly detected despite the nearby clutters, as we compare the Figure 3.3(a) and 3.3(b)

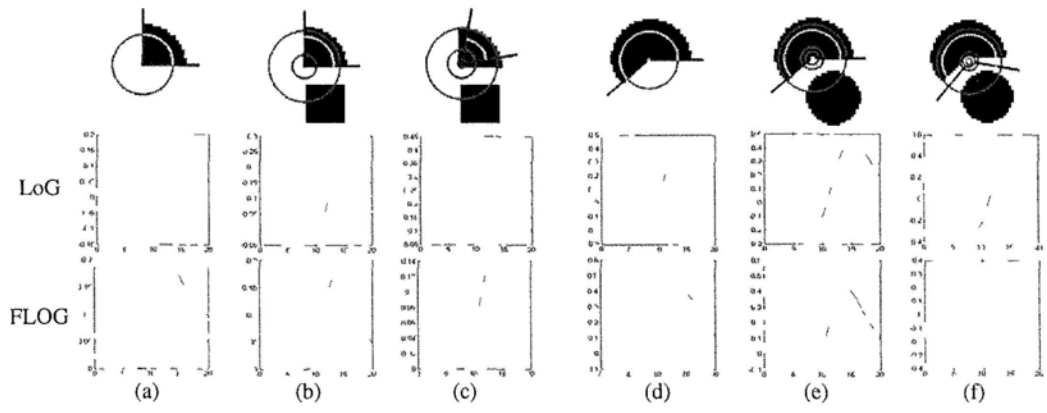
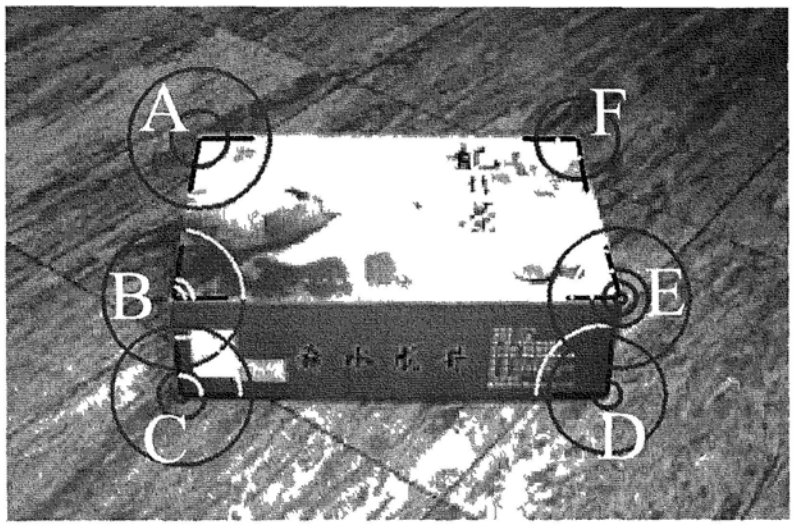


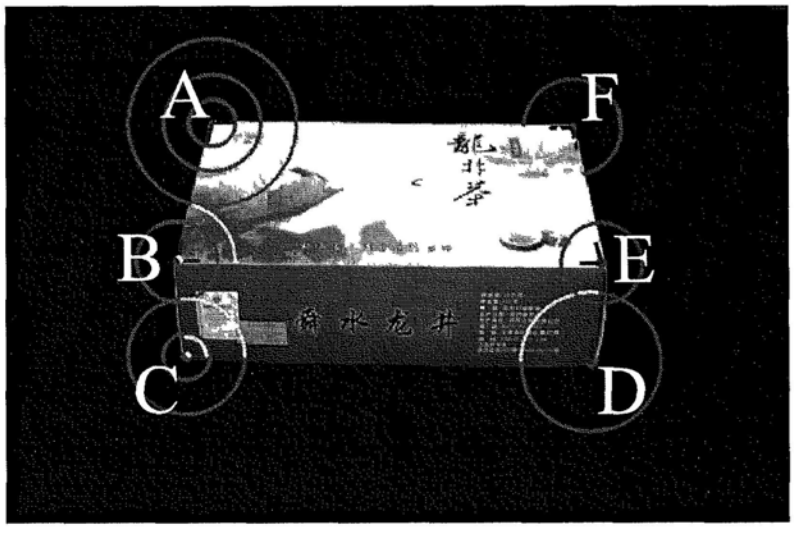
Figure 3.3: Automatic scale selection for fan image patterns. The first row shows the scales detected by the scale-normalized LoG (*red circle*) and the FLOG (*green arc*). The second and the third rows present the corresponding multi-scale responses computed using the scale-normalized LoG kernel and the FLOG kernel, respectively. The *horizontal* axis is the parameter k ($\sigma = 1.2^{k-1}$). The *vertical* axis is the integration response.

and Figure 3.3(d) and 3.3(e). In Figure 3.3(c) and 3.3(f), considerable errors of direction estimation and keypoint localization are introduced. As we can see, these errors have little influence on the extents detected by FLOG, except for a new scale arising in Figure 3.3(f) because an additional signal change is included in the fan sub-region. However, if there is no salient signal change within the region, the FLOG response may not present any extremum and could be more sensitive to errors and noise, which is true for LoG as well.

More experiments are conducted on real images. To test the proposed scale selection method under uniform scaling, we first took a picture of a box with clutter background as shown in Figure 3.4(a), and another three pictures of the same box but with clean background at different scales, i.e., the 2nd, 3rd and 4th pictures with scale times 1.2, 1.7 and 2.5 compared to the 1st picture, respectively. The fourth picture is given in 3.4(b). It is downscaled for the purpose of display. We then apply both LoG and FLOG to the four pictures to detect the scales for the six box corners indicated by A$\sim$F as shown in Figure 3.4. The keypoints and the directions for FLOG are specified manually. To evaluate the performance, we



(a) Scales detected in the 1st picture with clutter background



(b) Scales detected in the 4th picture with clean background (downscaled for display)

Figure 3 4 Detect the scales of corners using FLOG(green) and LoG(red)

True Scale Change Ratio	$\sigma_2/\sigma_1 = 1.2$		$\sigma_3/\sigma_1 = 1.7$		$\sigma_4/\sigma_1 = 2.5$	
Method / Corner	LoG	FLOG	LoG	FLOG	LoG	FLOG
A	1.44	1.2	1.728	1.728	2.986	2.488
B	0.695	1.0	1.0	1.439	1.441	2.073
C	1.2	1.2	1.728	1.728	2.488	2.488
D	1.44	1.2	2.074	1.728	3.583	2.488
E	1.728	1.2	2.488	1.439	3.583	2.488
F	1.44	1.0	2.074	1.439	2.986	2.073
Average Error	0.292	0.067	0.363	0.144	0.705	0.138

Table 3.1: Scale change ratio detected by LoG and FLOG

compute the ratio of the scale detected in the 2nd, 3rd and 4th picture to the scale detected in the 1st picture, i.e., $ratio = \sigma_i/\sigma_1$ $(i = 2, 3, 4)$, where σ_i is the scale detected in the ith picture. The ground truth of the three ratios should be 1.2, 1.7 and 2.5, respectively. In case of multiple scales detected for a corner, we choose the best one to compute the scale change ratio, i.e., choose the scale that leads to the ratio closest to the true ratio.

The results are summarized in Table 3.1. As we can see, FLOG performs much better than LoG in the sense of recovering the true scale changes between the pictures. The detailed results of the scales selected by the two kernels are shown in Figure 3.4. The red circles and green arcs indicate the scales detected by LoG and FLOG, respectively. Basically, the scales detected by FLOG are consistent in the two images, while most of the scales detected by LoG capture different image contents due to background clutter or the presence of multiple surfaces.

In Figure 3.5, we investigate the sensitivity of FLOG to the directions. For each corner in the first picture, three direction combinations are tested: the original combination of accurate direction (θ_1, θ_2) and two combinations of inaccurate directions $(\theta_1 - 10°, \theta_1 + 10°)$ and $(\theta_1 + 10°, \theta_1 - 10°)$. The detected scales corresponding to the three combinations are indicated by green, red and blue arcs in Figure 3.5, respectively. We can see that in most cases the same scales are detected for the corners regardless of the change of directions, except for some small

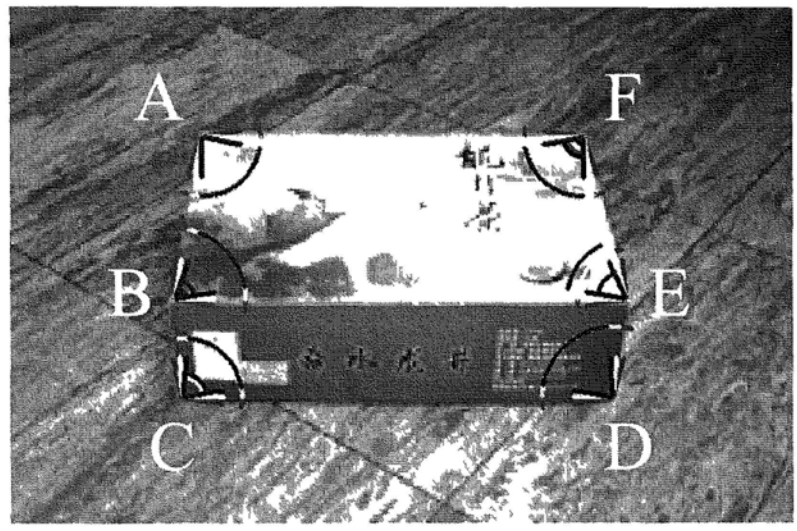


Figure 3.5: Scales of corners detected by FLOG with different combinations of fan directions.

scales arising from weak signal changes, e.g., the corner F. Of course, different scales may be detected when the image content bounded by the two directions varies a lot, e.g., the corner E. Basically the proposed method has good tolerance to the estimation error of directions. In practice, it is applicable to estimate the fan directions for FLOG according to meaningful image edges that represent the object contours.

3.3 Scale and Affine Invariant Fan Feature

In this section, we describe how to extract from images the Fan features that are invariant to scale and affine change. Basically it consists of four steps: (1) keypoint detection; (2) edge association; (3) scale selection; (4) affine normalization. In the following sub-sections, the details of each step are described.

3.3.1 Keypoint Detection

As the Fan features are specially designed for the keypoints located on surface boundaries, a natural choice will be to extract the keypoints from image bound-

aries [85]; [82]. However, for the sake of accurate localization and computational efficiency, we use a canny edge detector [18] with two improvements. First, in order to guarantee the accurate localization of edges and keypoints, the intensity gradients are computed at a fine scale. However, many clutters will arise from detailed textures at the fine scale. Therefore, the texture suppression technique [47] is employed before doing non-maximum suppression. Second, after hysteresis thresholding, there are usually many short and weak edge fragments. An edge cleaning procedure is then introduced to eliminate these fragments, as we believe that strong and long edges are more likely to represent object boundaries. The cleaning algorithm starts at each edge endpoint and tracks the edge fragment until arriving at another endpoint or a junction. The score of the tracked edge fragment is measured as the sum of the gradient magnitudes of all the associated edge points. Fragments with score less than a threshold are eliminated from the edge map. To achieve better results, the edge cleaning procedure is repeated a few times, with the threshold increased a little bit at each time. Typically 2 or 3 iterations are sufficient.

Keypoints should present good repeatability under various imaging conditions. Here, we propose a multi-scale selection scheme to select salient keypoints from the edge points. Let E denotes the set of edge points detected in the image. Let $H(\mathbf{e}, \sigma_k)$ denotes the Harris measure [50] of an edge point $\mathbf{e} \in E$ at the scale $\sigma_k = \sigma_0^{k-1}$ $(k = 1, 2, \ldots, K)$. The spatial neighbor of $\mathbf{e}$ is defined as

$$N(\mathbf{e}) = \{\epsilon \in E, \epsilon \neq \mathbf{e} \mid \|\epsilon - \mathbf{e}\|_2 \leq D_1\} \tag{3.10}$$

At each scale σ_k, we select a subset S_k of the salient edge points as the candidate keypoints by performing the non-maximum suppression as

$$S_k = \{\mathbf{e} \in E \mid H(\mathbf{e}, \sigma_k) \geq H(\epsilon, \sigma_k);\ \forall \epsilon \in N(\mathbf{e})\} \tag{3.11}$$

Note that the keypoints that represent the same local structure but are detected at different scales may shift a little from each other. We believe that the more scales a local structure survives, the more stable it is. We then track each candidate keypoint across scales, trying to find its affinities at different scales and group

Figure 3.6: Results of edge detection and keypoint extraction for two wide baseline images.

them together as a single representative keypoint. Specifically, for a keypoint $\mathbf{p}_k \in S_k$ detected at scale σ_k, its affinity $\mathbf{p}_{k+1}$ at the next scale is defined as

$$\mathbf{p}_{k+1} = \arg\min_{\mathbf{x} \in S_{k+1}} \|\mathbf{x} - \mathbf{p}_k\|_2 \quad \text{subject to } \|\mathbf{x} - \mathbf{p}_k\|_2 \leq D_2 \tag{3.12}$$

If $\mathbf{p}_{k+1}$ exists, it will be removed from the set S_{k+1}, and we then try to find $\mathbf{p}_{k+2}$ for $\mathbf{p}_{k+1}$. Otherwise, the tracking is stopped and we obtain a group of keypoints $\{\mathbf{p}_m, \mathbf{p}_{m+1}, \ldots, \mathbf{p}_{m+n}\}$. This group of keypoints will be combined into a single representative $\mathbf{p}_m$, i.e., the one detected at the finest scale, and its saliency is measured by $n+1$, i.e., the number of consecutive scales it survives. The tracking will be performed for each candidate keypoint $\mathbf{p} \in S_k$ $(k = 1, 2, \ldots, K)$ until all the keypoints have been removed from the candidate sets. Finally, we keep those representatives whose saliency is not smaller than T. In our experiments, the above parameters are empirically set to $\sigma_0 = 1.4, K = 5, T = 3$. The distance measure D_1 and D_2 are efficiently implemented by 7×7 and 3×3 windows, respectively. Figure 3.6 shows an example of the edges and the keypoints detected in two wide baseline images. We can see the high repeatability of the keypoints. Note that a few keypoints may not have associated edges or characteristic scales (see sub-section 3.3.2 and 3.3.3) and hence are removed later.

3.3.2 Edge Association

For each keypoint, nearby edge fragments are associated to guide its neighborhood division. Inspired by [82], we first approximate each edge fragment by one or several straight line segments, as represented by the dotted lines in Figure 3.7. Then the correlation score between the keypoint $\mathbf{p}$ and a line segment l_k within the local window W is calculated by (3.13), where $d(\mathbf{p}, l_k)$ is the Euclidean distance of the keypoint $\mathbf{p}$ from the line segment l_k. The parameter δ is used to control the distance tolerance and is set to a small number such that the score drops fast as the distance increases. A Gaussian weighting $G(\mathbf{x} - \mathbf{p}; \sigma)$ is introduced centered on $\mathbf{p}$ to emphasize the edge points near the keypoint.

$$Score_k = \sum_{\mathbf{x} \in l_k, \mathbf{x} \in W} G(\mathbf{x} - \mathbf{p}; \sigma) \cdot \exp(-d(\mathbf{p}, l_k)^2 / \delta^2) \tag{3.13}$$

Line segments with high scores are chosen to associate with the keypoint. In practice, as the truly related line segments usually have salient scores, a simple thresholding is sufficient. Finally, a line emitting from the keypoint is fitted to each associated line segment, indicating a division direction. Accordingly, multiple sub-regions are constructed around the keypoint. An example of edge association is shown in Figure 3.7. The three green dotted lines are the line segments associated to the yellow keypoint. Accordingly, the keypoint neighborhood is divided into three sub-regions indicated by the red lines.

In our experiments of wide baseline matching, we choose to discard those sub-regions whose included angles are larger than 200°, because most of them capture either the background or multiple physical surfaces. Background sub-regions probably have no correspondences since the content of background could change a lot in wide baseline images. As for the regions comprised of multiple surfaces, we cannot use a single affine transform to model its geometric deformation. By removing these regions, there will be less clutter in the final feature matching.

3.3.3 Scale Selection

The characteristic scales for each sub-region are automatically selected by FLOG as described in Section 3.2. As more than one scale can be detected for a sub-

Figure 3.7: Edge association. The yellow dot represents the keypoint. By edge association, the three line segments (green dotted lines) are associated to the keypoint. The estimated division directions are indicated by the three red solid lines through the yellow dot.

region, multiple scale invariant Fan features with different extents and from different sub-regions can be extracted for a single keypoint. For discrete implementation of the FLOG kernel, we face the problem of finite sampling approximation. In our experiments, the mask size of FLOG is set heuristically to $1 + ceil(3\sigma)$. To restore the zero mean property for the discrete FLOG mask, all the positive coefficients are uniformly scaled such that their sum equals to the absolute sum of all the negative coefficients. Of course, this procedure will slightly distort the mask shape. By experiments we find that it usually leads to more distinctive extrema of FLOG response, but has little influence on the scales where the extrema are detected.

Figure 3.8 gives some examples of the scale invariant Fan features detected in wide baseline image pairs. The scales selected by LoG are also displayed for comparison. We can observe that the Fan features together with their FLOG scales can be detected consistently between widely separated views, whereas the scales selected by LoG are largely affected by nearby clutters. As suggested in (3.9), Figure 3.3 and Figure 3.8, in general the region extent detected by FLOG is shrunk compared to the location of salient signal change. To make the Fan features more distinctive, the detected sub-regions should be further enlarged to include the signal changes. However, large regions may lose the local properties such as the local planarity and the robustness to occlusion. In our experiments, the extents of all the sub-regions are enlarged by three times.

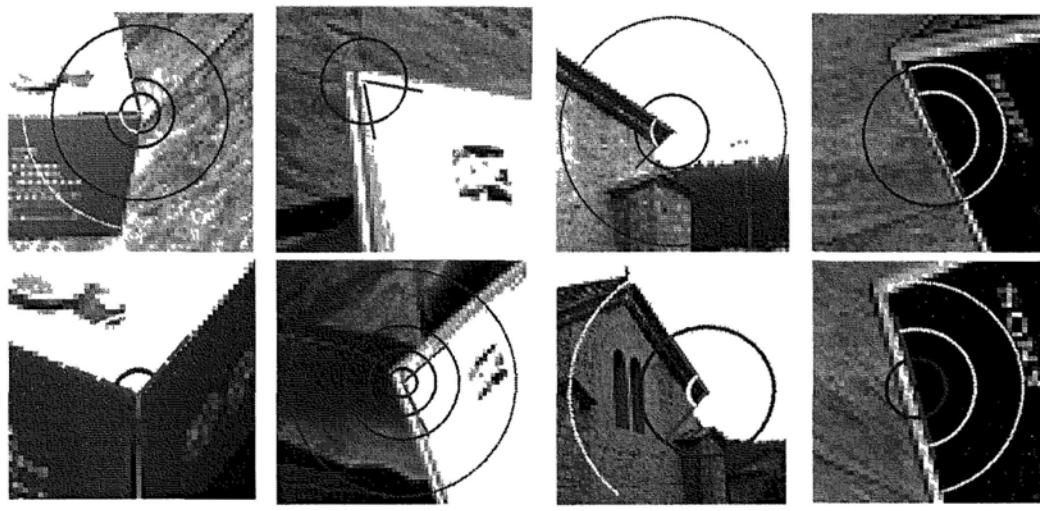


Figure 3.8: Scale invariant Fan features detected in real images taken from quite different viewpoints. The red lines are the division directions estimated by edge association. The scales selected by FLOG and LoG are indicated by the green arcs and the blue circles, respectively.

3.3.4 Affine Normalization

In addition to the scale change, fan sub-regions may suffer geometric deformation when observed from different viewpoints. Under the assumption that each sub-region represents a locally planar surface, such a deformation can be modeled by an affine transform and hence can be addressed by affine normalization. This will make the scale invariant Fan features further possess affine invariance.

The second moment matrix [78]; [9]; [92] can be used to measure the affine deformation of an isotropic structure. This method works well for a circular support region, but is not suitable for a fan sub-region. Indeed, other sub-regions attached to the keypoint should never be involved in estimating the affine shape of the concerned sub-region, because they are supposed to represent different physical surfaces.

On the other hand, the covariance matrix [120]; [100]; [87] has also been successfully employed to diagnose the affine shape. For an image region S with arbitrary shape, the local image moments M_{ij}, the region centroid $\mathbf{x}_c$ and the covariance matrix $\mathbf{C}$ can be computed by (3.14), (3.15) and (3.16), respectively,

where $I(x, y)$ denotes the image intensity at (x, y).

$$M_{ij} = \iint_S x^i y^j I(x, y)\, \mathrm{d}x\, \mathrm{d}y \tag{3.14}$$

$$\mathbf{x}_c = [\overline{x},\ \overline{y}]^\top = [M_{10}/M_{00},\ M_{01}/M_{00}]^\top \tag{3.15}$$

$$\mathbf{C} = \begin{bmatrix} M_{20}/M_{00} - \overline{x}^2 & M_{11}/M_{00} - \overline{x} \cdot \overline{y} \\ M_{11}/M_{00} - \overline{x} \cdot \overline{y} & M_{02}/M_{00} - \overline{y}^2 \end{bmatrix} \tag{3.16}$$

Let λ_a and λ_b be the largest and smallest eigenvalues of the covariance matrix $\mathbf{C}$, respectively. Let $\mathbf{v}_a$ and $\mathbf{v}_b$ be the two corresponding eigenvectors. An important property of $\mathbf{C}$ is that its $\mathbf{v}_a$ and $\mathbf{v}_b$ indicate the semi-major and semi-minor axes of the elliptical (affine) shape of S, and λ_a and λ_b are proportional to their squared lengths. If $\lambda_a \neq \lambda_b$, which is usually the case in practice, we can use the affine transformation given in (3.17) to project the anisotropic image pattern to an isotropic one. Here, $\mathbf{A}$ denotes the affine transform matrix, $\mathbf{x}$ and $\hat{\mathbf{x}}$ are the image coordinates before and after the affine transformation, respectively. s is a scaling factor. In this chapter, it is decided to normalize $s\lambda_a^{-1/2}$ to 1, such that the image pattern is only expanded in the direction of $\mathbf{v}_b$.

$$\hat{\mathbf{x}} = \mathbf{A}(\mathbf{x} - \mathbf{x}_c) = s \begin{bmatrix} \lambda_a^{-1/2} & 0 \\ 0 & \lambda_b^{-1/2} \end{bmatrix} \begin{bmatrix} \mathbf{v}_a^\top \\ \mathbf{v}_b^\top \end{bmatrix} (\mathbf{x} - \mathbf{x}_c) \tag{3.17}$$

Note that the affine normalization is performed centered on the estimated region centroid $\mathbf{x}_c$ that is definitely not the position of the keypoint to which the fan sub-region is attached. As shown in Figure 3.9(a), directly computing the covariance matrix on the fan sub-region will give us a diagnosis of the affine deformation centered on the region centroid, i.e., the red dots in Figure 3.9(a). Affine normalization based on this shape estimation, as indicated by the green ellipses in Figure 3.9(a), cannot accurately compensate the true deformation centered on the yellow colored keypoint. Figure 3.9(b) shows the considerable differences of the normalized sub-regions detected in the two images I_1 and I_2 with significant viewpoint change. Here the fan directions are represented by the red and blue lines, and the region extent determined by the FLOG scale is indicated by the green arcs. Figure 3.9(c) shows the corresponding affine shapes in the original

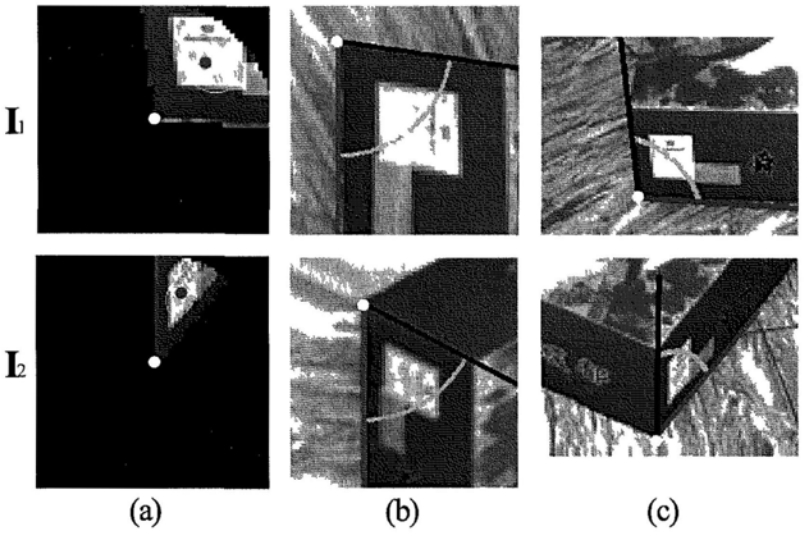


Figure 3.9: Traditional affine normalization applied to the fan sub-regions detected in two images I_1 and I_2. (a) Original image patches for affine shape diagnosis; (b) Normalized image patches; (c) Corresponding affine shapes in the original images

images, where we can also observe that the sub-regions are inconsistently detected in the two images.

Here we introduce a simple and efficient method to address this problem. First recall that the covariance matrix relies on the anisotropy of the region pattern to estimate the affine shape (similar principle for the second moment matrix). Suppose that a sub-region represents an incomplete planar surface attached to a keypoint, in comparison with a circular feature whose support region is a complete surface around the keypoint. To estimate the affine shape of a sub-region attached to a keypoint, we propose to predict the image pattern of the missing surface part by mirroring the known sub-region with the keypoint as the center, as shown in Figure 3.10(a). For a mirror-predicted image patch, its region centroid is guaranteed to locate on the keypoint, and hence the affine shape diagnosed by the covariance matrix, as indicated by the green ellipses in Figure 3.10(a), can give us a better estimation of the local geometric deformation around the keypoint. The improvement in affine normalization can be clearly observed as we compare

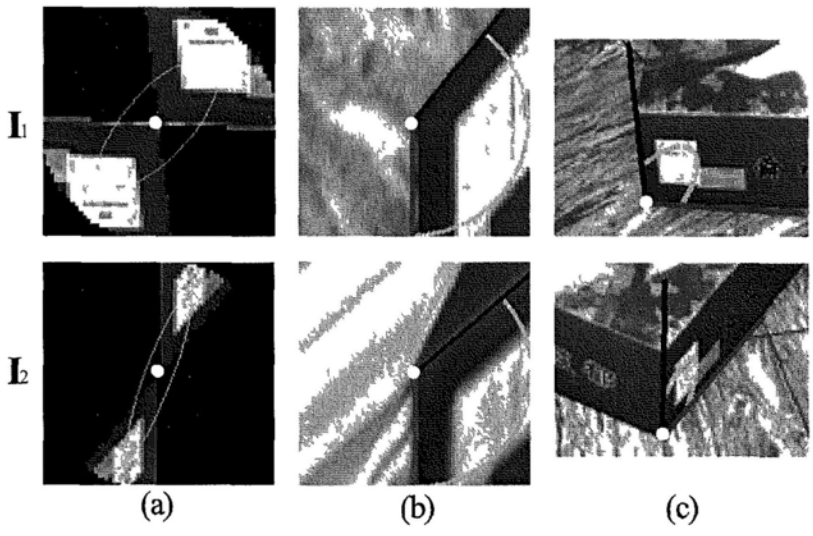


Figure 3 10: Improved affine normalization applied to the fan sub-regions detected in two images I_1 and I_2. (a) Mirror-predicted image patches for affine shape diagnosis, (b) Normalized image patches; (c) Corresponding affine shapes in the original images

Figure 3.9(b) and (c) with Figure 3.10(b) and (c), where the appearances of the two sub-regions normalized by using the mirror prediction are much more similar than using the traditional way.

In addition, there are a few sub-regions whose fan angles are larger than 180°. For these fan regions, we clip them into half circles before performing the mirror-prediction, so as to guarantee the good localization of region centroid. That is to say, the clipped parts will not be used for diagnosing the local affine shape. Recall that the maximal fan angle is restricted to 200°. The regions clipped out are trivial compared to the remaining parts. As a consequence, the clipping will not affect much the affine shape estimation.

An iterative estimation method similar to [92] can be employed to further improve the scale and affine normalization of Fan features. To save computations, however, we adopt a single scale selection plus affine normalization, and we found that it works well in the experiments. To conclude the whole section, the proposed method can efficiently extract consistent sub-regions from two images with

significant viewpoint change, and normalize the regions to present quite similar appearance, which is essential for feature description and matching.

3.4 Fan-SIFT Descriptor

The well known SIFT descriptor [80] is an invariant and stable representation of region appearance by a weighted histogram of gradient locations and orientations. It performs best in the context of matching and recognition [93]. The Fan SIFT descriptor proposed in this section is an extension of the SIFT descriptor for describing the fan sub-regions. The technical details are given below.

First, the intensity gradients are computed in the normalized image patch generated by the method described in Section 3.3, as shown in Figure 3.11. The smoothing scale σ_g for computing the gradients is chosen as $\sigma_g = k\sigma_s\sqrt{\theta/2\pi}$, where σ_s is the FLOG scale, θ is the fan angle of the normalized sub-region, and the control parameter k is set to $1/3$ in our experiments, so as to preserve the fine texture details for high discrimination. Following [80], the gradients are weighted by a global Gaussian function centered on the keypoint to provide the robustness to occlusion to some extent.

As can be observed in Figure 3.11(a), there are always strong gradients around the sub-region boundaries. These gradients actually depict the region shape rather than its inner texture. To suppress these boundary gradients, we introduce a boundary mask defined in (3.18), where $\mathbf{x}$ is the sample position and $dis(\mathbf{x})$ is the minimal distance from $\mathbf{x}$ to the two boundaries. The threshold ϵ_2 is set to $1+3\sigma_g$ by taking into account the diffusion of Gaussian smoothing for computing gradients. The threshold ϵ_1 is simply set to $\epsilon_2/2$. Unlike the one in [127], our suppression is only performed on samples very close to the region boundaries, such that the inner texture can be well preserved for the purpose of discrimination. Figure 3.11(b) shows the relevant gradients after the boundary suppression. Gradients outside the sub-regions are eliminated as well.

$$Mask\mathbf{x} = \begin{cases} 0 & , dis(\mathbf{x}) \leq \epsilon_1 \\ 1 & , dis(\mathbf{x}) \geq \epsilon_2 \\ \frac{\exp(dis(\mathbf{x})-\epsilon_1)-1}{\exp(dis(\epsilon_2-\epsilon_1)-1} & , \text{otherwise} \end{cases} \tag{3.18}$$

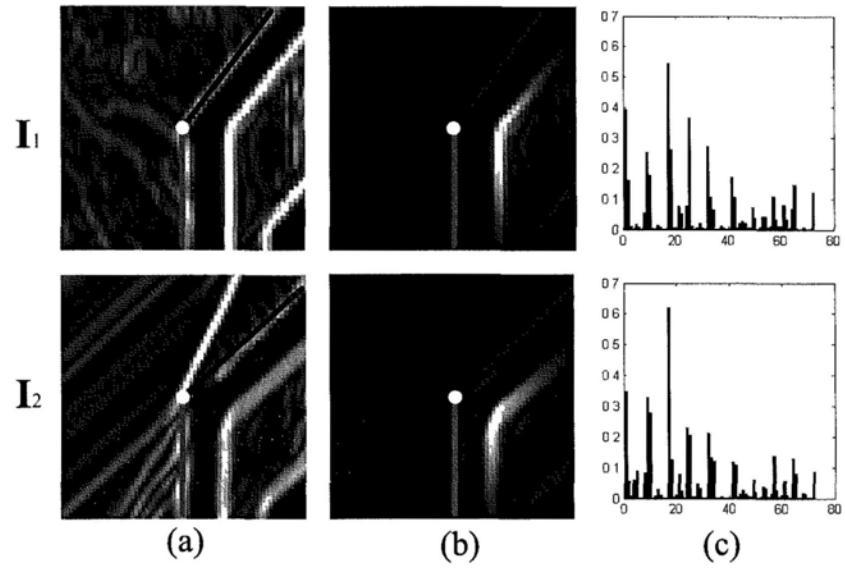


Figure 3.11: Computation of Fan-SIFT descriptor. (a) Gradients computed in the normalized image patch. (b) Relevant gradients after global Gaussian weighting and boundary suppression. Gradients outside the sub-regions are also eliminated. (c) Fan-SIFT descriptors (normalized histogram with 72 bins).

Next, fan grids are introduced to distribute the gradients into 9 discrete locations, as shown in Figure 3.12. The width of the fan region is 3 times of the FLOG scale, i.e., $w = 3\sigma_s$. The radius of the three fan rings are set to $a = w/3$, $b = 2w/3$ and $c = w$, and the fan angles for each ring are equally divided, such that all the fan grids have the same areas[1], i.e., all the discrete locations have the same number of gradient samples. To achieve rotationally invariant description, a coordinate system is aligned to the direction from the fan vertex to the region centroid, which is unique once the keypoint and the fan region are determined. In this way, we avoid estimating the dominant gradient orientation as the SIFT does. Gradient orientations are then computed in this coordinate frame and are quantized into 8 bins.

Matching gradients while allowing for shift is the key idea that leads to the success of the SIFT descriptor [93]. To further improve the tolerance of gradient

[1] $S_f = \theta\sigma_s^2/2$

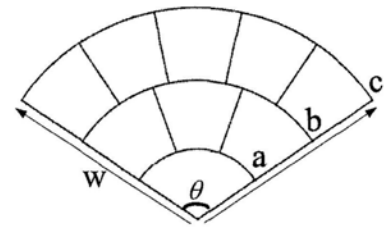


Figure 3.12: Design of fan grids for Fan-SIFT descriptor

shift, a local Gaussian weighting is introduced to each individual fan grid centered on its centroid. The scale of the local Gaussians is determined as $\sigma_l = \frac{1}{2}\sigma_s\sqrt{\theta/2\pi}$, such that the area of the fan grid S_f is equal to the area of a circle with radius of $2\sigma_l$. The local Gaussian weightings can suppress the gradients near the grid boundaries and consequently reduce the influence of the strong gradients shifting across the neighboring grids to the descriptor.

Finally a histogram of gradient locations and orientations is built in a way similar to [80]. Note that for each gradient sample, its contribution to the histogram is weighted by its gradient magnitude, the boundary mask, the global and the local Gaussian weightings. Based on the histogram, a vector with $9 \times 8 = 72$ dimensions is composed as the Fan-SIFT descriptor. The descriptor is further normalized into a unit vector to compensate for affine changes in illumination. Figure 3.11 illustrates the computation of Fan-SIFT descriptors for the two Fan features in different images. As we can see in Figure 3.11(c), the two Fan-SIFT descriptors can be reliably matched.

3.5 Matching based on Fan Features

The similarity of two Fan features is measured by the Euclidean distance between their descriptors. The *nearest neighbor distance ratio* [80]; [93] is employed to match the descriptors. Specifically, two descriptors $\mathbf{v}_A$ and $\mathbf{v}_B$ are matched if $\|\mathbf{v}_A - \mathbf{v}_B\|/\|\mathbf{v}_A - \mathbf{v}_C\| < t$, where the descriptors $\mathbf{v}_B$ and $\mathbf{v}_C$ are the first and second nearest neighbor to $\mathbf{v}_A$. The threshold t is set to 0.8 in our experiments.

To obtain the tentative correspondences of keypoints based on the matching of Fan features, we introduce the following strategy. Two keypoints $\mathbf{p}_1$ and $\mathbf{p}_2$ are

matched as long as one of the Fan features attached to $\mathbf{p}_1$ can be matched to one of those attached to $\mathbf{p}_2$. This is based on our assumption that each fan sub-region represents a local physical surface attached to the keypoint. As a result, any one of them can be used as the signature of the keypoint.

To automatically reject false matches, we apply a global and a semi-local outlier filter. The global one is the epipolar test [52] using RANSAC [37]. Matches that violate the estimated epipolar geometry will be removed. The semi-local filter is based on the consensus of neighboring local affine transforms. Please refer to Section 4.3 for more details. This affine consistency filter can be applied to any affine invariant features such as Affine Hessian & Harris [92], MSER [87], EBR & IBR [141] and the proposed Fan feature.

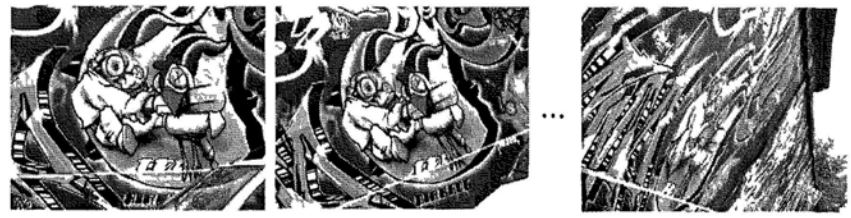
3.6 Experimental Results

To evaluate our method, we compare the proposed Fan feature with Harris Affine, Hessian Affine [92]; [95] and EBR [141], all of which have been efficiently implemented[1] and are publicly available. Different features basically capture different image structures. The Harris and Hessian features detect corner-like and blob-like structures within object surfaces [93]. Both EBR features and Fan features are extracted from edges. EBR arises from well-formed edge junctions, while Fan feature aims at both edge junctions and the salient points along edges. Through the following experiments, we show that not only does the Fan feature possess good invariance property that is comparable to the state-of-the-art features [95], but also it can successfully match image structures near surface discontinuities, and hence contributes to the variety of the bag of features.

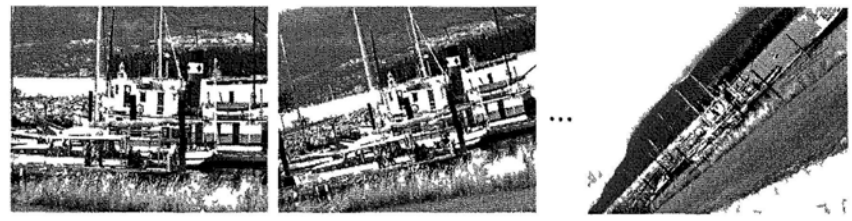
3.6.1 Repeatability under Scale, Viewpoint and Lighting Change

In this sub-section, we follow the standard test [95] to evaluate the repeatability and accuracy of Fan features under scale, viewpoint and lighting changes. The

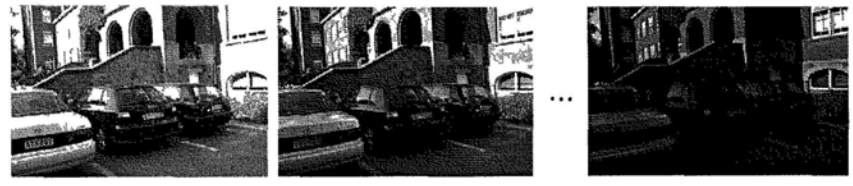
[1] http://www.robots.ox ac uk/~vgg/research/affine/



(a) Graffiti sequence



(b) Boat sequence

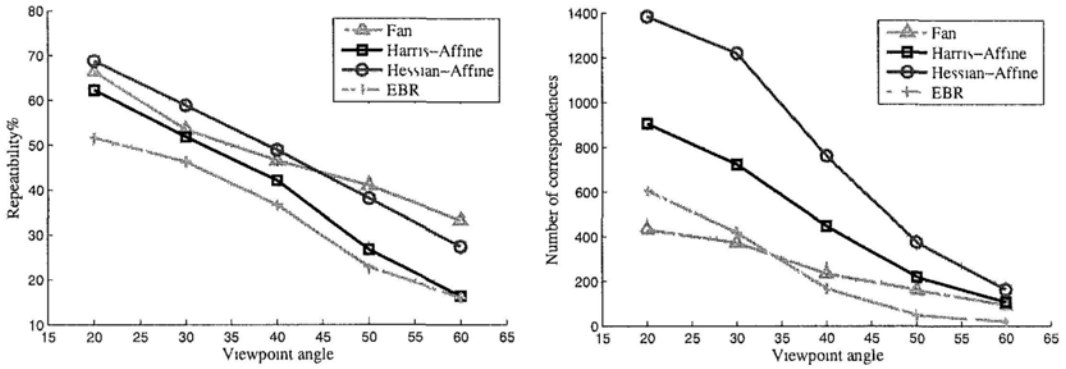


(c) Leuven sequence

Figure 3.13: Standard test images.

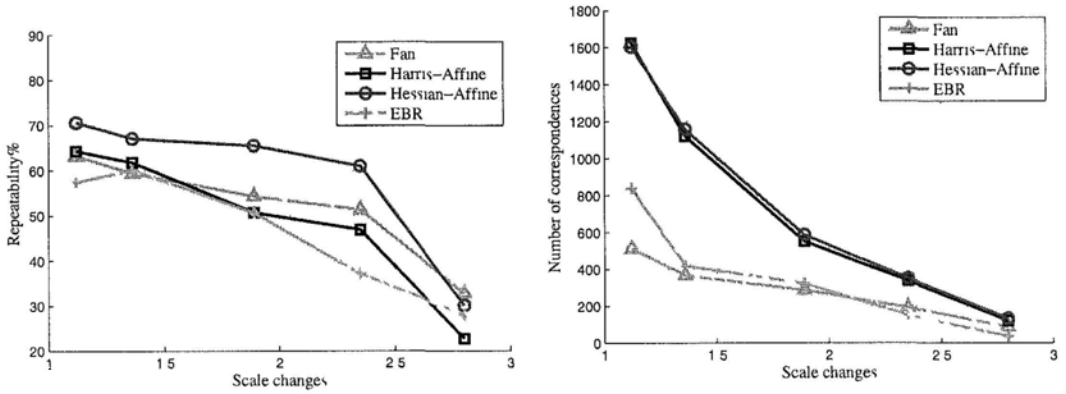
Graffiti, Boat and Leuven sequences are shown in Figure 3.13, and the test results for the three sequences are shown in Figure 3.14, Figure 3.15 and Figure 3.16, respectively. The image pairs in these sequences can be related by a single homography. Thus we can measure the accuracy of two corresponding features by the overlap of their elliptical regions which are mapped onto the same image by the known homography. The support region of a Fan feature is deemed as a complete ellipse in this test. Following [95], the region size is normalized to a radius of 30 pixels prior to computing the overlap measure, and two features are considered as a correspondence if the overlap error is smaller than 40%. The repeatability score is computed as the ratio between the number of correspondences and the smaller of the number of features extracted in a pair of images

Figure 3.14(a) shows that for the Graffiti scene, the Fan feature has better repeatability than Harris Affine and EBR under viewpoint changes. Though



(a) Repeatability score (b) Number of correspondences

Figure 3.14: Test of viewpoint invariance for Graffiti sequence.



(a) Repeatability score (b) Number of correspondences

Figure 3.15: Test of scale (+ rotation) invariance for Boat sequence.

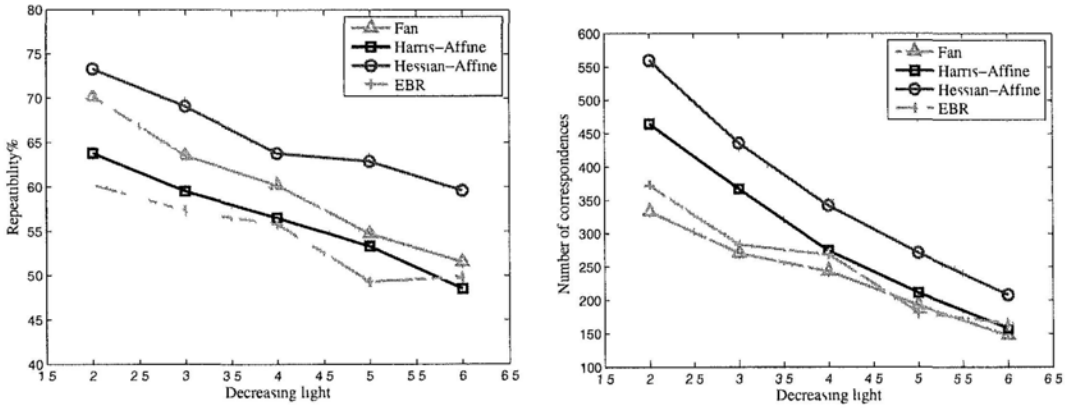
Hessian Affine performs the best for small and median viewpoint angles, Fan feature exceeds it in the case of large viewpoint changes. The test result for scale invariance in Figure 3.15(a) is slightly different. The repeatability of Fan feature falls below Harris Affine for small scale change, yet still better than EBR. And the gap between Hessian Affine and Fan feature becomes larger for small and median scale changes. Similar results can be observed in Figure 3.15(a) for linear lighting changing. We can conclude that the Fan feature has comparable repeatability to the state-of-the-art feature detectors. On the other hand, the invariance of the feature under the studied transformation is reflected in the slope of the curves, i.e., how much does a given curve degrade with increasing transformations. In this sense, the Fan feature has outstanding invariance to scale, viewpoint and lighting changes compared with the other features.

Figure 3.14(b), Figure 3.15(b) and Figure 3.16(b) indicate that the correspondences of Fan features are much fewer than those found by the other features. This is because the Fan features are essentially extracted in a smaller number due to the strict selection through several steps (See Section 3.3). But we note that it still contributes a lot to the match quantity in case of significant image changing. In addition, like all edge-based features, Fan feature performs worse for purely textured scenes such as the Wall and Bark sequences [95]. For this kind of scenes, Fan feature is therefore not recommended.

3.6.2 Scale, Viewpoint and Background Invariance

Images used in the standard test [95] are mostly of planar scenes with no background change or clutter. Experiments presented in this sub-section will further take into account the background variation around the surface discontinuity, in addition to the scale and viewpoint changes. Figure 3.17 shows the test images[1]. The first group (S0∼S4) is used to test the scale invariance. Attempt is made to match S0–S1, S0–S2, S0–S3 and S0–S4. The scale changes of the four image pairs are $1/1.2^k$ ($k = 1, 2, 3, 4$) successively. The second group (V0∼V4) is used to test the viewpoint invariance. We try to match the image pairs of V0–V1, V0–V2, V0–V3 and V0–V4, where the viewpoint changes are approximately 15, 30, 45

[1] All images are with resolution of 300×200.



(a) Repeatability score (b) Number of correspondences

Figure 3.16: Test of lighting invariance for Leuven sequence.

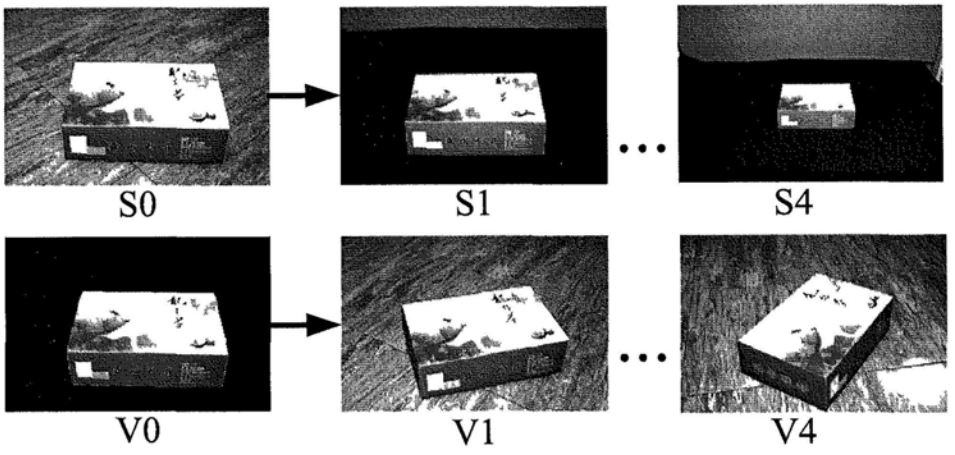
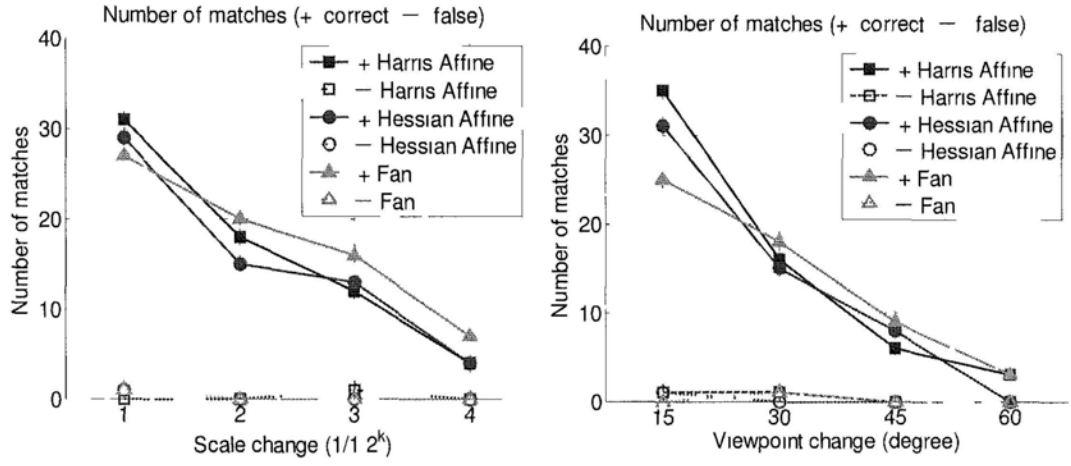


Figure 3.17: Test images with different scales and viewing angles



(a) scale & background invariance (b) viewpoint & background invariance

Figure 3.18: Performance of different features on invariance test.

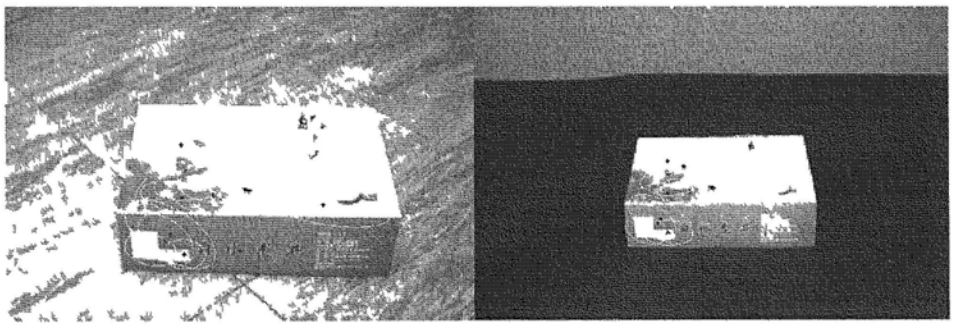
and 60 degree. All the image pairs have quite different backgrounds, so as to test the background invariance in the meantime. Note that the 3D box in the test images provides sufficient surface textures to raise Harris and Hessian features Meanwhile it is well structured for extracting edge-based features like EBR and Fan feature.

As the image pairs can no longer be related by simple homographies, we now focus on the performance of actual feature matching based on the feature descriptors. Through our experiments, the standard SIFT descriptor is used to describe the Harris Affine, Hessian Affine and EBR, while the Fan feature uses the Fan-SIFT descriptor instead The similarity measure based on Euclidean distance and the strategy of nearest neighbor distance ratio (t=0.8) are adopted to initially match these features. We then apply the semi-local filter (Section 4.3) to reject false matches.

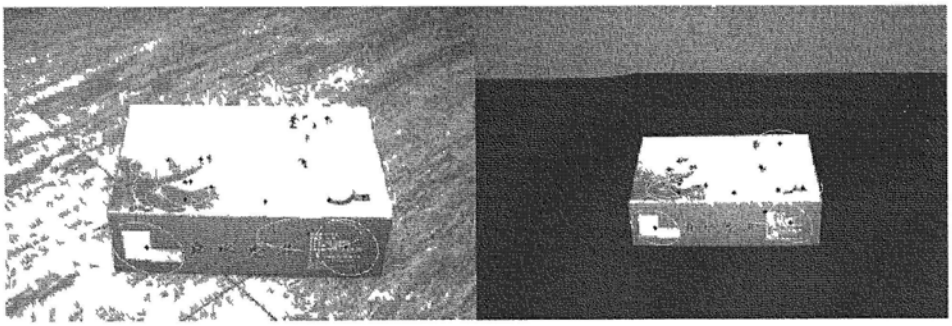
Test results are presented in Figure 3.18, where the solid marks indicate the number of correct matches and the hollow ones indicate the false matches As EBR generates few matches for small scale and viewpoint changes and totally fails in case of large changes, it is not plotted in Figure 3.18. From the results we can see that when there is only slight change in scale or viewpoint, Harris and Hessian Affine produce more correct matches than Fan feature This again indicates that

Fan feature may contribute less to the match quantity in case of small image changes. However, when these changes become more significant, Fan feature tends to preserve more correct matches than Harris and Hessian Affine, which also suggests that the Fan feature has better invariance under scale, viewpoint and background changes.

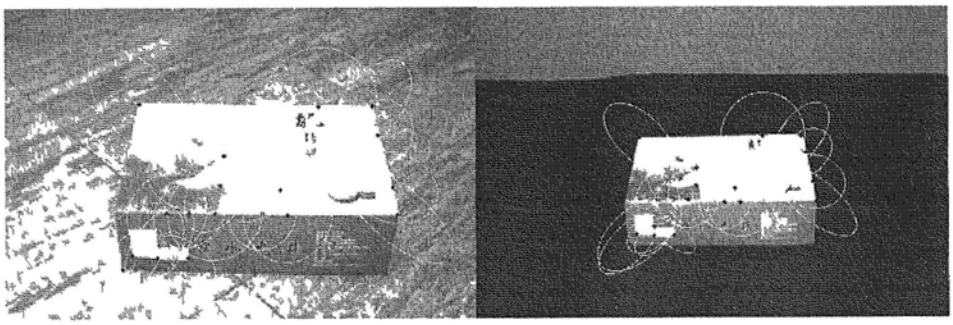
In Figure 3.19, some typical matching results are shown for visual comparison, where the keypoints are represented by the red dots and their associated support regions are represented by the green ellipses. False matches are indicated by red ellipses instead. Note that the support region of Fan feature is only a fan part of the ellipse which can be distinguished by the clear image edges. As we can see in Figure 3.19(a), (b) and (c), the keypoints extracted by Harris and Hessian Affine are quite different from those by Fan feature, while EBR find few matches as shown in Figure 3.19(d). Hessian Affine generally extracts image blobs. Harris Affine detects image corners, but has little chance to match the corners on or near the object boundaries, because their support regions probably contain different backgrounds. Note that for a keypoint near the object boundary, Harris and Hessian Affine may adapt its support region to a small or highly deformed one to avoid crossing the surface boundary. However, such features are usually less distinctive or unstable under scale or viewpoint change. In comparison, Fan features are specially designed to save the keypoints on or near surface boundaries. As shown in Figure 3.19(c), most keypoints matched by Fan fea-tures are located on the box boundaries, including the 3D box corners. Since the keypoints are extracted from edges, some of them may arise from salient surface textures. In this case, even when the keypoints are close to the surface boundaries, they can still be matched by Fan features provided that one of the sub-regions is distinctive enough and does not cross the surface boundary. Similar observations can be found as well in the image matching results presented in the next sub-section. In conclusion, the Fan feature is complementary to the classical circular features such as SIFT [80], Harris Affine and Hessian Affine [92].



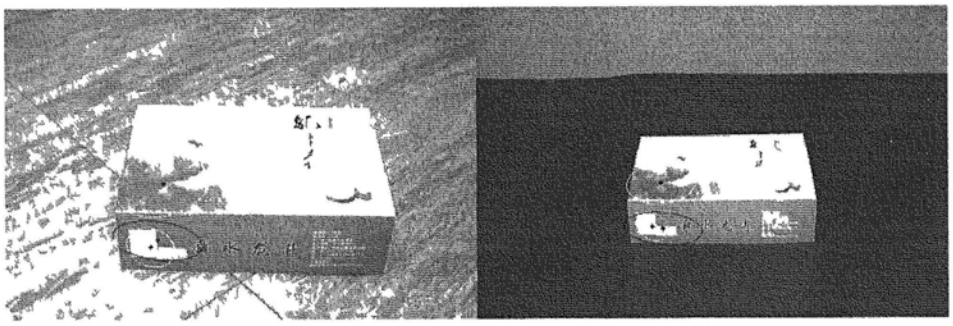
(a) S0-S2 by *Hessian Affine*



(b) S0-S2 by *Harris Affine*



(c) S0-S2 by *Fan feature*



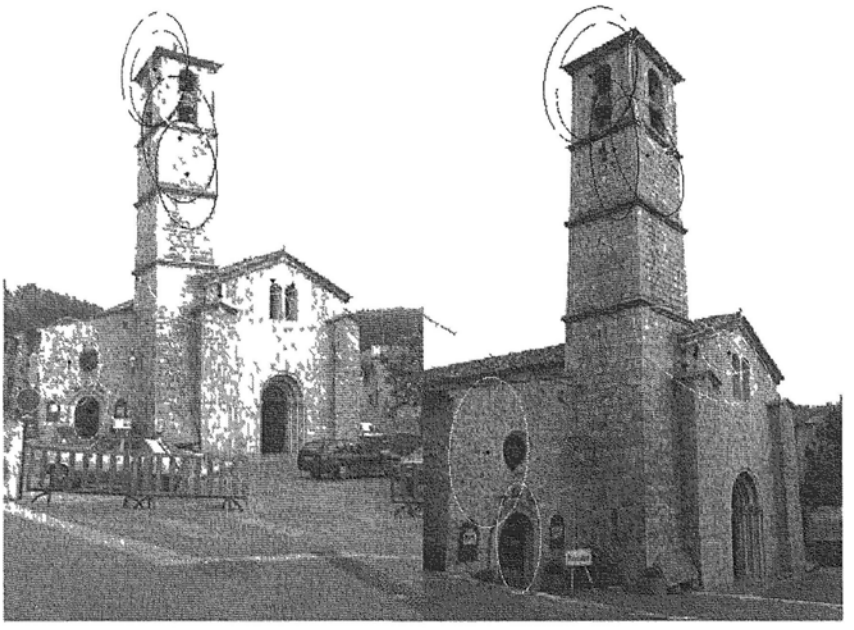
(d) S0-S4 by *Fan feature*

Figure 3.19 Selected matching results from the test on scale & background invariance

(a) Fan feature (correct/false = 31/4)

(b) Harris & Hessian Affine (correct/false = 20/3)

Figure 3.20 Test on *Church* (300 × 450) viewpoint change + scale change + lack of texture



(a) EBR (correct/false = 8/4)

Figure 3.20: Test on *Church* (300 × 450): viewpoint change + scale change + lack of texture (*continue*)

3.6.3 Wide Baseline Image Matching

More matching examples are presented to demonstrate the utility and effectiveness of Fan feature for matching structured scenes with significant scale, viewpoint (pose) and background changes between images. The image contents vary from man-made objects such as buildings and pencil bags, to natural creatures like birds and butterflies, and most of them have annoying background clutters.

The matching results are visually shown in Figures 3.20, 3.21, 3.22 and 3.23, where the image resolution, the number of correct and false matches and the major difficulties in matching the images are annotated as well. The results of Fan feature are compared with EBR and Harris & Hessian Affine which is an efficient combination of both Harris Affine and Hessian Affine[1]. EBR fails for the image pairs in Figure 3.21 and Figure 3.22, and hence is not displayed there. As we can see, the Fan feature consistently outperforms the other features in terms

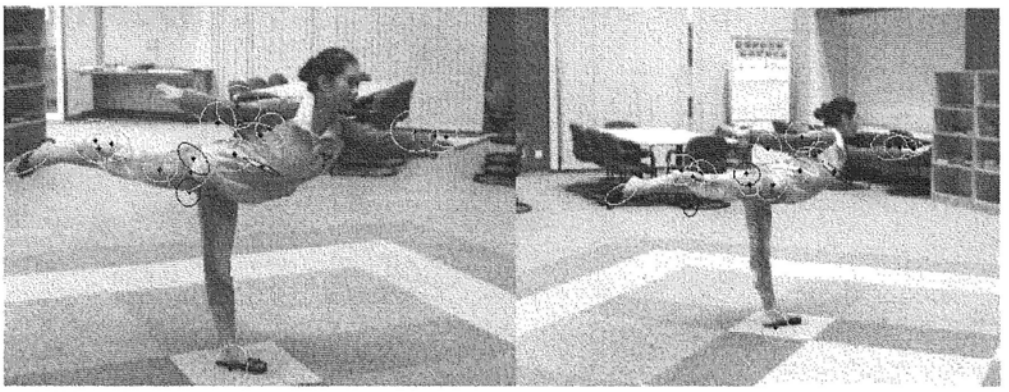
[1]http //www robots ox ac uk/~vgg/research/affine/

(a) Fan feature (correct/false = 13/1)

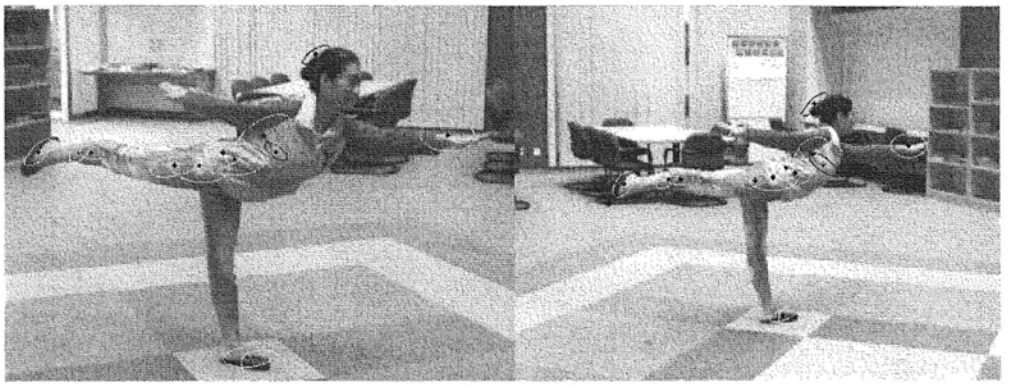
(b) Harris & Hessian Affine (correct/false = 9/0)

Figure 3.21 Test on *Butterfly* (400×300) pose change + background clutter + homogeneous texture

of the number of correct matches. It is necessary to point out that besides the strong image changes, the test objects do not possess many distinctive textures, which result in only a small number of correspondences. Since the textures at small scales are not distinctive enough, the support regions need to be enlarged to increase the discriminating power. Large Harris and Hessian features, however, are very likely to cover surface discontinuities and as a consequence cannot be matched in case of changing viewpoints or backgrounds. In comparison, there is less risk of crossing surface discontinuity by matching large sub-regions. This is also why a few Fan features survive in Figure 3.23(a) and (b), while Harris and Hessian Affine fail to pass the semi-local filter. In this sense, Fan feature is



(a) Fan feature (correct/false = 15/3)

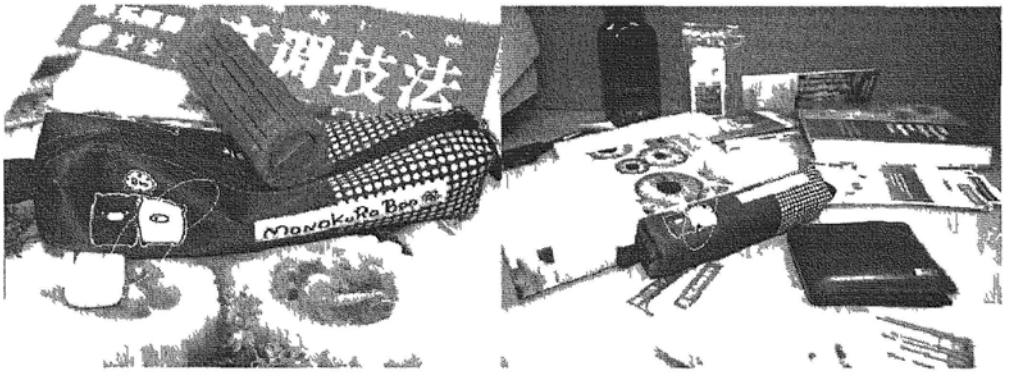


(b) Harris & Hessian Affine (correct/false = 11/2)

Figure 3.22: Test on *Yoga* (600 × 450): significant viewpoint change (about 75°) + scale change + background clutter + lack of texture

superior in matching the weakly textured surfaces under changing viewpoints or backgrounds.

Another important observation is that the matched regions found by Fan feature is complementary to those found by other features, which is true even for EBR in Figure 3.20. Keypoints on surface boundaries are successfully matched by Fan features despite the changing viewpoints and the background clutter. In comparison, most matched Harris and Hessian features are far away from the surface discontinuities. Their support regions are restricted within the object surfaces, unless the covered backgrounds have little change between the images.



(a) *Pencil Bag* (450 × 300) viewpoint change + scale change + severe clutter + non rigid deformation

(b) *Toucan* (300 × 200) pose change + background clutter + lack of texture

Figure 3 23 Correspondences established by Fan features, where Harris & Hessian Affine fails

Feature + Descriptor	Run time (sec)	Number of features
Fan feature + Fan-SIFT	1 95	354
Harris Affine + SIFT	5 48	1374
Hessian Affine + SIFT	3 62	926

Table 3 2 Computation time of feature extraction and description for the leftmost image in Figure 3 20

3.6.4 Computational Complexity

The extraction and description of Fan features involve a number of steps. The edge detection is performed at a single scale, and is slightly slower than the standard canny edge detector due to additional texture suppression and edge cleaning. The Harris measure is computed at 5 scales, but only for the edge points, so is the non-maximum suppression (NMS). Keypoint tracking is only for the edge points that survive NMS. Basically the keypoint selection is very fast and the number of detected keypoints is typically much smaller than Harris and Hessian Affine. For each keypoint, we then perform edge association (Sub-regions larger than 200 degree are discarded), scale selection (12 scales are explored), affine normalization and Fan-SIFT description. There is no iteration of scale and shape adaptation as is used in Harris and Hessian Affine.

Table 3.2 gives the computation time measured on a Core Duo T2400 1.83GHz Windows laptop, for the leftmost (300×450) image shown in Figure 3.20. It also gives the number of features extracted from this image. Though the run time may change depending on the image content, the table gives a reasonable indication of typical time consumption. Note that the Fan feature is implemented without any optimization. In addition, because the Fan features are extracted in smaller quantity and the Fan-SIFT descriptor has lower dimensions, matching Fan features is much faster than matching Harris and Hessian Affine.

3.7 Conclusion

In this chapter, scale and affine invariant Fan features are proposed to match the keypoints located on or near surface boundaries. Multiple Fan features are attached to a single keypoint to provide robustness to image content change around depth discontinuity (including the background change). For each Fan feature, its characteristic scale is selected based on the proposed FLOG kernel. Its affine shape is diagnosed from the mirror-predicted surface patch. In this way, the Fan features can be consistently extracted from two images despite scale change and geometric deformation. Fan-SIFT descriptor is then introduced to depict the feature's texture content. The Fan features are not extracted in a large

quantity because the keypoints are carefully selected to guarantee the saliency and repeatability. Experimental results show that the Fan features have good repeatability for structured scenes and have superior invariance to strong scale, viewpoint and background changes. Moreover, the Fan feature is complementary to traditional circular features, especially for describing the surfaces that are weakly textured or close to the object boundaries. In the next chapter, we will show that the combination of Fan features and Harris & Hessian Affine features provide us a good initial set of correspondences for rendering the object by wide baseline images. Adding Fan features into the bag of features may also benefit other applications like object recognition and image retrieval. However, it is out of the scope of this thesis.

Chapter 4

Feature-based Object Rendering from Sparse Views

4.1 Introduction

Image-based rendering (IBR) is to use a collection of images capturing the same scene or object from different viewing positions for synthesizing the virtual image which would be seen from a new viewpoint. IBR techniques have many potential applications, such as virtual reality [108] and free viewpoint television [134]. In this chapter, we are interested in developing an IBR system that is able to render the focused objects of interest from a very small set of widely separated images

In terms of scene representation, strategies for IBR can be divided into three categories according to how much geometric information is involved. They are rendering with no geometry, rendering with implicit geometry like correspondence, and rendering with explicit geometry. Two of the best-known rendering techniques that require no geometry are light field [71] and lumigraph [46], whose success comes from the usage of a large number of densely sampled images. On the other hand, a natural approach for IBR is to explicitly compute a 3D model of the scene or object. Virtual images can then be synthesized by projecting the reconstructed 3D model onto novel views. Typical examples are texture mapped rendering of stereo reconstructions [114]; [15]; [130]; [13], volumetric techniques [57]; [136]; [147]; [122], space carving [117]; [66]; [88]; [14], and surface evolution [30]; [106]; [44]; [154]. These approaches in general require fewer images yet

higher computational load. Implicit geometry techniques typically use the view dependent geometry to guide the selection of color at each pixel in the novel view. Recent works [59]; [38]; [151]; [159] using Implicit geometry have demonstrated high efficiency for rendering complex scenes. Using view dependent geometry may introduce unpleasant flicking during rendering. Techniques such as the spatial-temporal view interpolation [116] have been proposed to ensure the temporal continuity and produce flicker-free rendering. We refer the reader to [118] for a more thorough discussion of the related techniques.

The camera setup of an IBR system plays a crucial role in determining its applications and designs. Previous works mainly focus on the small baseline setup, where there are a large number of reference images that provide adequate overlap between neighboring views for accurate depth estimation or 3D reconstruction. Such a sequence of images can be acquired either by recording video of the scene or the object [52] which, however, are limited to static ones, or by a densely sampled camera network [119] that is expensive and inconvenient to set up. Besides, rendering with a large number of images also means the requirement of large data storage and high memory cost. In the past few years, there is an increasing demand for rendering new scenes from images acquired using simple devices such as family photographs [126]; [112]. Furthermore there is a wide class of applications where the images are not taken for the purpose of modeling, but reconstruction or rendering is desirable afterwards [126]; [128]; [129]. All these applications introduce the wide baseline configuration, where the novel views are synthesized from a small number of still images taken from very different viewpoints. However, the wider spacing between the cameras brings more challenges in producing locally consistent geometries and hence photorealistic views. This is because the large occlusions become more of an issue and the strong photometric and geometric changes make it much more difficult to establish correspondences between widely separated images.

Recently, local invariant features [95]; [93] have made wide baseline matching possible, and hence the viewpoints can be put further apart. Basically, in wide baseline matching the local invariant features are first extracted independently from two images, and then characterized by invariant descriptors, based on which the correspondences are finally established. Thanks to the intense research works

done these years, the local features have been developed to be invariant not just to translation, but also to rotation [118], scale change [77]; [80] and affine deformation [92]; [87]; [141], which are very common in wide baseline images. Please refer to Section 3.1 for detailed introduction of recently proposed invariant feature detectors and descriptors. Based on the correspondences between two widely separated images, multiview matching algorithms [112] [5] have been proposed to track the correspondences across multiple wide baseline images.

For modeling and rendering purpose, however, a sparse set of matched features obtained by these methods is inadequate for producing a satisfactory 3D representation. Recently, Tola et al. [135] propose to replace the commonly used correlation windows by a fast and robust descriptor, called DAISY. They feed it to a graph-cut based depth estimation algorithm, which is reported to produce dense depth maps with good quality in wide baseline stereo. Besides, a few techniques have been developed to address the problem by growing regions or surfaces starting from a small set of extracted features or seed points [67]; [48]. Strecha et al. [128] develop a dense matching algorithm for multiple wide baseline images. A sparse set of initial depth estimates is propagated to dense depth map by an inhomogeneous time diffusion process. Lhuillier et al. [72]; [73] present a quasi-dense approach to establish surface reconstruction using a greedy match propagation method. Yao et al. [153] further improve this propagation method by introducing the clustering-based photoconsistency and the data-driven depth smoothness.

Furukawa et al. [43] propose to represent the scene by a dense set of rectangular patches that cover the surfaces visible in the input images. Their algorithm starts from a sparse set of matching keypoints, and repeatedly expands these to nearby pixel correspondences before using visibility constraint to filter away false matches. The most relevant previous work is [35]; [36], where Ferrari et al. propose to refine the matches of affine invariant features by maximizing the similarity function in the 6D affine space. Later they employ the match refinement to propagate more feature correspondences using the initial matches as the propagation attempts, which has proven to be successful in the application of simultaneous object recognition and segmentation.

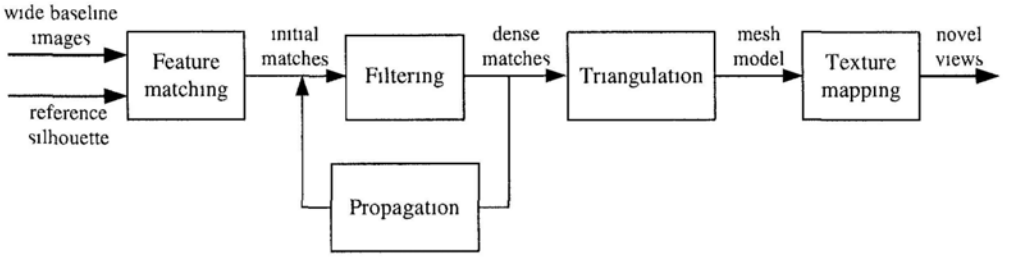


Figure 4 1: The framework of the proposed feature-based image rendering scheme

In this chapter, we present a wide baseline IBR method based on the affine invariant features. Fig. 4.1 shows the framework of the proposed feature-based rendering scheme The inputs include a small set of wide baseline images that capture the object from different viewpoints and a reference silhouette of the object defined in one view. We start from a sparse correspondences of affine invariant features obtained by initial matching, and then produce a quasi-dense set of accurate matches by iteratively filtering outliers and propagating new ones. Finally a triangular mesh model is constructed, based on which view dependent texture mapping is performed to synthesize novel views of the object. This IBR scheme has also been extended to video-based rendering (VBR) for applications like free-view TV. The contributions of this work includes: 1) propose a novel and efficient geometric filter to remove mismatches of affine invariant features based on the pair-wise affine consistency; 2) present a global match refinement and propagation method that takes into account both the appearance similarity and the geometry consistency, and hence can successfully deal with the low-texture regions for which the local method [36] usually fails; 3) introduce a double-weighting texture blending algorithm that is able to provide realistic and smooth free-view navigation. 4) present a video based rendering scheme by tracking the affine invariant features across successive frames. While there are a number of modeling and rendering methods proposed in recent years, our rendering system stands out because of the following desired features: (1): It requires few images (even two) to work and only needs the silhouette from one view to define the object of interest, which largely reduces the user effort; (2) It is robust to the photometric and geometric changes arising from strong wide baseline images; (3)

it is able to deal with low-texture surfaces that are common for general scenes or objects; (4) It can be efficiently extended to render moving object.

The rest of this chapter is organized as follows. Section 4.2 introduces the affine invariant features used in our method and the initial matching step. Section 4.3 proposes the pair-wise affine consistency measure and the outlier filter based on it. Section 4.4 presents a global method to refine and propagate affine invariant features. Section 4.5 discusses the triangulation and the view-dependent rendering algorithm. Section 4.6 extends the feature-based scheme to video-based rendering. Section 4.7 shows the experimental results of the proposed method in rendering still and moving objects, and Section 4.8 concludes the chapter.

4.2 Affine invariant Features and Initial Matching

One of the major difficulties in wide baseline stereo is the significant geometric deformation and illumination change. Local invariant features [95]; [93] have been shown to be a very successful tool to address these problems. The features are extracted locally to handle clutters and partial occlusions, and are carefully normalized to achieve the scale, rotation and affine invariance. These properties make them very suitable for our task. In this chapter, we employ three kinds of invariant features that capture quite different image regions, so as to cover the object surface to the greatest extent for guiding the match propagation later. The Affine Harris features [92] mainly represent the texture corners, the Affine Hessian features [92] tend to depict the image blobs, and the Fan features (please refer to Section 3.3) [25] are used to describe the partial surfaces near the object boundaries. All these features are extracted in a way invariant to rotation, scale change and local affine deformation, and are called affine invariant features in this thesis.

An affine invariant feature has elliptical support region and can be represented by $\mathbf{f} = [\mathbf{x}, a, b, o, \theta, \mathbf{v}]$, where $\mathbf{x}$ is the image coordinates of the feature's keypoint, a and b are the lengths of the semi-major and semi-minor axes of the

feature's elliptical region, o indicates the orientation of the major axis, θ represents the region's dominant orientation which for instance can be estimated by gradient histogram [80], and $\mathbf{v}$ is the feature descriptor that summarizes the region appearance. The Affine Harris and Affine Hessian features are described by the well-known SIFT descriptor [80], which is a stable texture representation by a weighted histogram of gradient locations and orientations. The Fan features are described by the Fan-SIFT descriptor (please refer to Section 3.4), which is an extension of the SIFT descriptor to fan-shaped regions. Both the SIFT and Fan-SIFT descriptors are normalized into a unit vector and only the gradients are taken into account in the description. Therefore, the descriptors are also invariant to linear illumination change.

In the reference image I_R, the affine invariant features are extracted only from the object region defined by the given silhouette. For the other target images I_T[1], the features are detected in the entire image space. Then the initial matching aims to find for each feature $\mathbf{f}_R$ in I_R its correspondence $\mathbf{f}_T$ in I_T. In this stage, the similarity of two features is simply measured by the Euclidean distance of their descriptors:

$$sim(\mathbf{f}_R, \mathbf{f}_T) = \|\mathbf{v}_R - \mathbf{v}_T\| \tag{4.1}$$

And the *nearest neighbor distance ratio* (please refer to Section 3.5) is employed to establish the initial correspondences between the reference and target features.

4.3 Affine Consistency and Outlier Filtering

Though the similarity measure (4.1) based on feature descriptors is widely used as the benchmark of feature matching [93], it only relies on the region appearance that may not be sufficiently discriminative to ensure correct matches, especially for smooth regions with low textures. In this section, we propose an efficient geometric filter to further remove the mismatches from the correspondence set by making use of the remaining information provided by an affine invariant feature, i.e., $(\mathbf{x}, a, b, o, \theta)$. We first introduce the *pair-wise affine consistency* by taking into account both the neighborhood correlation between the features and the

[1] One or two target images in our experiments

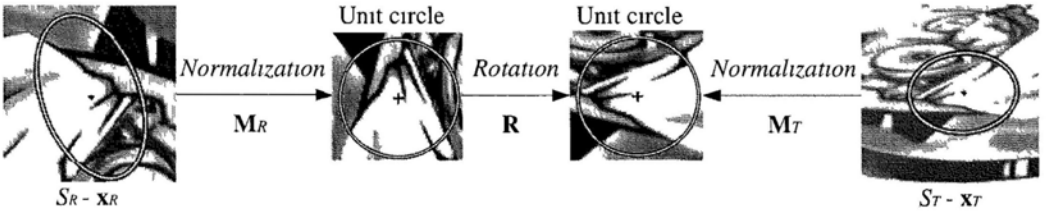


Figure 4.2: Local affine transform estimated from a correspondence of affine invariant features [95]

consistency of local affine geometry between the matches. We then present an iterative algorithm to efficiently filter out the false matches based on the affine consistency measure

4.3.1 Local Affine Transform from a Match of Affine Invariant Features

Let S_R and S_T denote the support regions of two matched features $\mathbf{f}_R$ and $\mathbf{f}_T$, respectively. The two regions can be centered on $(0,0)$ by $S_R - \mathbf{x}_R$ and $S_T - \mathbf{x}_T$, where $\mathbf{x}_R$ and $\mathbf{x}_T$ are the coordinates of the corresponding keypoints. Since (a_R, b_R, o_R) and (a_T, b_T, o_T) are available, we then can normalize their elliptical shapes into unit circles by affine transforms $\mathbf{M}_R$ and $\mathbf{M}_T$, respectively. As shown in Figure 4.2, the two unit circles are now related by a pure rotation $\mathbf{R}$ [95] which can be determined by the dominant orientations of $\mathbf{f}_R$ and $\mathbf{f}_T$, i.e., θ_R and θ_T . Therefore, the two regions can be related by (4.2), which actually is an affine transform that can be represented by (4.3) in homogeneous coordinates.

$$S_T - \mathbf{x}_T = \mathbf{M}_T^{-1}\mathbf{R}\mathbf{M}_R(S_R - \mathbf{x}_R) \tag{4.2}$$

$$\mathbf{A}\mathbf{f} = \begin{bmatrix} \mathbf{A} & \mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix}, \quad \text{where } \mathbf{A} = \mathbf{M}_T^{-1}\mathbf{R}\mathbf{M}_R,\ \mathbf{t} = \mathbf{x}_T - \mathbf{A}\mathbf{x}_R \tag{4.3}$$

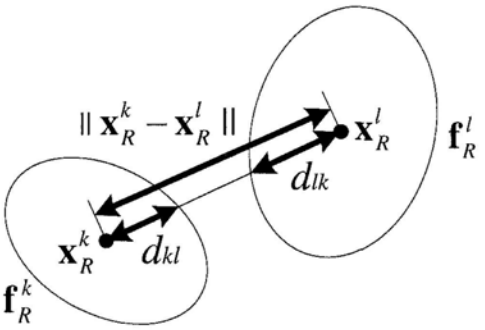


Figure 4.3: Normalized spatial distance between two features

4.3.2 Pair-wise Affine Consistency

An affine transform is sufficient to locally model the image distortion arising from viewpoint changes [95]. Suppose that two neighboring features are located on the same surface which is approximately planar. Their support regions will undergo similar affine transforms when the viewpoint changes. These two features are called *affine consistent*, and they will support each other to survive the proposed geometric filter.

Let $(\mathbf{f}_R^k, \mathbf{f}_T^k) \in \Omega$ $(k = 1, \ldots, N)$ be a target match in the correspondence set Ω. Let $\mathbf{A}\mathbf{f}^k$ denote the affine transform that relates their support regions S_R^k and S_T^k. Let $(\mathbf{f}_R^l, \mathbf{f}_T^l) \in \Omega$ $(l \neq k)$ be a nearby match whose support regions are S_R^l and S_T^l. The pair-wise affine consistency measure $AC(k,l)$ between the match $(\mathbf{f}_R^k, \mathbf{f}_T^k)$ and $(\mathbf{f}_R^l, \mathbf{f}_T^l)$ is defined as

$$AC(k,l) = \exp(-dis(\mathbf{f}_R^k, \mathbf{f}_R^l)^2/\delta) \cdot \frac{S_T^l \cap \mathbf{A}\mathbf{f}^k S_R^l}{S_T^l \cup \mathbf{A}\mathbf{f}^k S_R^l} \tag{4.4}$$

The first term of $AC(k,l)$ measures the neighborhood correlation between the two features $\mathbf{f}_R^k$ and $\mathbf{f}_R^l$ in the reference image. The spatial distance between the two features not only depends on the location of their keypoints, but also is normalized according to the size and shape of their support regions. The normalized spatial distance between the two features $\mathbf{f}_R^k$ and $\mathbf{f}_R^l$ is computed by (4.5).

$$dis(\mathbf{f}_R^k, \mathbf{f}_R^l) = \left\| \mathbf{x}_R^k - \mathbf{x}_R^l \right\| / (d^{kl} + d^{lk}) \tag{4.5}$$

Here d^{kl} denotes the distance from the keypoint $\mathbf{x}_R^k$ to the intersection of the elliptical region and the ray from $\mathbf{x}_R^k$ and through $\mathbf{x}_R^l$, as illustrated in Figure 4.3. It can be easily computed by (4.6)

$$d^{kl} = \sqrt{(a_R^k \cos\phi^{kl})^2 + (b_R^k \sin\phi^{kl})^2} \tag{4.6}$$

, where $\phi^{kl} = ori(\mathbf{x}_R^l - \mathbf{x}_R^k) - o_R^k$ and $ori(\ )$ is the vector orientation. d^{lk} is similarly defined.

If $dis(\mathbf{f}_R^k, \mathbf{f}_R^l) = 1$, the support region of $\mathbf{f}_R^k$ will be tangential to that of $\mathbf{f}_R^l$. In case of $dis(\mathbf{f}_R^k, \mathbf{f}_R^l) < 1$, the two regions will share some overlaps, and when $dis(\mathbf{f}_R^k, \mathbf{f}_R^l) > 1$, they will have no intersection. Simply by thresholding the normalized distance, we can determine the neighboring features in a way adaptive to the features' region size and shape. In general, neighboring features with small $dis(\mathbf{f}_R^k, \mathbf{f}_R^l)$ (e.g., smaller than 1) will have high probability to undergo similar local affine deformations when the viewpoint changes. Thus, it is reasonable to use the first term of $AC(k,l)$ for measuring to what degree the two reference features $\mathbf{f}_R^k$ and $\mathbf{f}_R^l$ are spatially correlated and accordingly how reliable the two matches $(\mathbf{f}_R^k, \mathbf{f}_T^k)$ and $(\mathbf{f}_R^l, \mathbf{f}_T^l)$ can support each other to pass the geometric filter.

The second term of $AC(k,l)$ is to measure the consistency of the local affine geometry estimated from the two matches $(\mathbf{f}_R^k, \mathbf{f}_T^k)$ and $(\mathbf{f}_R^l, \mathbf{f}_T^l)$. Ideally, if the two reference features $\mathbf{f}_R^k$ and $\mathbf{f}_R^l$ undergo the same affine transform, we will have $\mathbf{A}\mathbf{f}^k S_R^l = \mathbf{A}\mathbf{f}^l S_R^l = S_T^l$. In practice, however, the two regions $\mathbf{A}\mathbf{f}^k S_R^l$ and S_T^l will differ from each other, as shown in Figure 4.4. Here the difference of the two regions is used to measure the inconsistency of the affine transforms $\mathbf{A}\mathbf{f}^k$ and $\mathbf{A}\mathbf{f}^l$. Specifically, the difference between $\mathbf{A}\mathbf{f}^k S_R^l$ and S_T^l is quantified by their overlap in image area. In (4.4), $S_T^l \cap \mathbf{A}\mathbf{f}^k S_R^l$ is the intersection of the two regions, which is then normalized by their union $S_T^l \cup \mathbf{A}\mathbf{f}^k S_R^l$. The overall AC value ranges from 0 to 1. A large $AC(k,l)$ value indicates that the two matches $(\mathbf{f}_R^k, \mathbf{f}_T^k)$ and $(\mathbf{f}_R^l, \mathbf{f}_T^l)$ are not only spatially correlated but also affine consistent, and hence are very likely to be a pair of correct matches.

4.3.3 Outlier Filter based on Affine Consistency

Given the correspondence set Ω with N candidate matches, we first build the $N \times N$ **AC** matrix whose entries are $AC(k,l)$ $(k,l \in [1, \quad , N])$, where $AC(k,l) =$

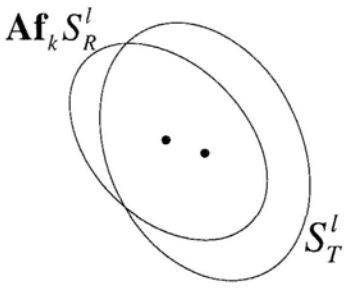


Figure 4.4: Inconsistency of local affine geometry

0 if $k = l$. The AC score for a match indexed by k is calculated by

$$\mathrm{AC}_k = \sum_l AC(k, l) \tag{4.7}$$

In general, $AC(k, l) \neq AC(l, k)$ due to the asymmetry of the second term in (4.4). However, in practice the two values are usually very close, thus we can reasonably assume symmetric **AC** matrix to reduce half of the computations. To further reduce the computations, we only evaluate the AC measures for neighboring matches with $dis(\mathbf{f}_R^k, \mathbf{f}_R^l) < th_1$[1]. The AC measures of far away match pairs have very small values and contribute little to the AC scores anyway.

Next, we iteratively remove the inconsistent matches from Ω and meanwhile update the **AC** matrix. The algorithm is outlined as follows.

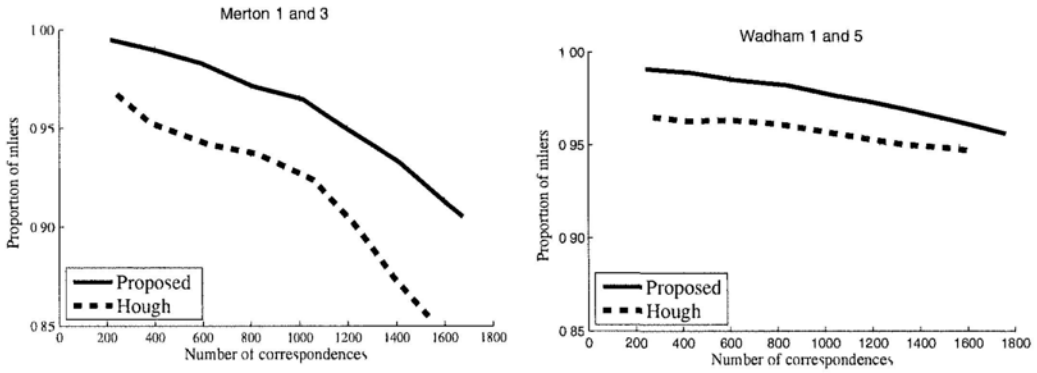
1. Compute the AC scores for all matches in the current set Ω by (4.7) based on the current **AC** matrix;
2. Remove from Ω all the matches $(\mathbf{f}_R^k, \mathbf{f}_T^k)$ whose $\mathrm{AC}_k = 0$, and update the **AC** matrix by deleting the corresponding rows and columns;
3. For the remaining matches in Ω, find the one with the smallest AC score, i.e., $\mathrm{AC}_{\min}$.
4. If $\mathrm{AC}_{\min} > th_2$[2], stop. Otherwise, remove the match and update the **AC** matrix accordingly, then go to step 1.

[1] th_1 is set to 2 in our experiments.
[2] th_2 is set to 0.1 in our experiments.

Figure 4.5 Wide baseline Images. Top row shows the frames 1 and 3 of the Merton sequence Bottom row shows the frames 1 and 5 from the Wadham set

Matches with no support are directly rejected as mismatches because isolated correct matches are very rare in practice. Furthermore, the worst match in terms of AC score is the most probable mismatch in the remaining correspondence set. Removing this match from the **AC** matrix will largely reduce the AC scores of other nearby mismatches with similar wrong affine transforms, but has little influence on the AC scores of nearby correct matches. As a result, the nearby mismatches are more likely to be filtered out in following iterations

We evaluate the proposed geometric filter in two applications· wide baseline matching and non-rigid motion. We use Affine Harris and Affine Hessian [92] to extract the affine invariant features. The initial matches are generated based on the SIFT descriptors. We then apply the proposed geometric filter to reject the outliers, resulting in the final correspondence set For comparison, we also present the results of Hough clustering using the same configuration in [80], which is an efficient and effective mismatch rejection step based on the global spatial configuration



(a) Results for Merton

(b) Results for Wadham

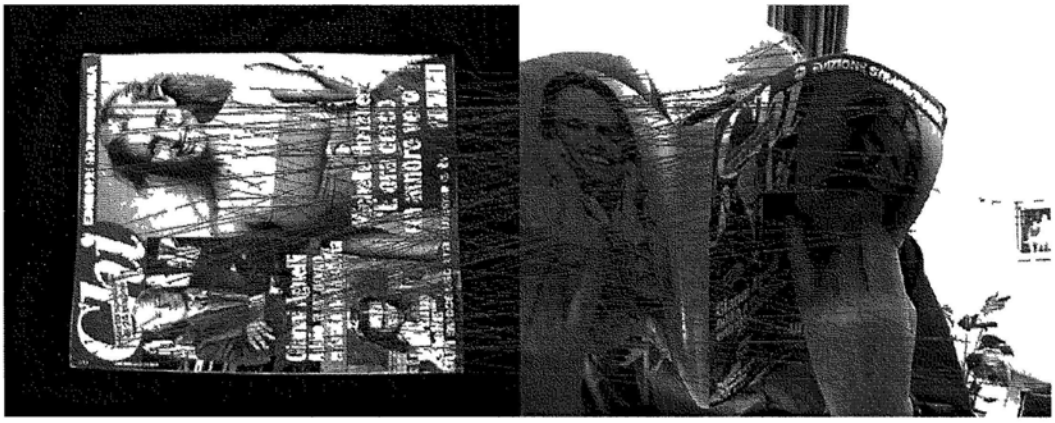
Figure 4.6: Inlier ratio of the final correspondence sets obtained by Hough clustering and the proposed geometric filter.

Wide Baseline Matching

Images used in this experiment are displayed in Figure 4.5. They are available at the Oxford's Visual Geometry Group's webpage. The performances of different filters are assessed in terms of the proportion of inliers within the final correspondence set. An inlier is considered to be a match that conforms to the ground-truth epipolar geometry (the deviation of the feature keypoint from the epipolar line is within 4 pixels). Figure 4.6 presents the proportion of inliers in terms of the set size for different filters. The variation of the set size is obtained by changing the nearest neighbor distance ratio in the initial matching. As we can see from Figure 4.6, the proposed filter achieves higher inlier ratio than the Hough clustering throughout the varying set size.

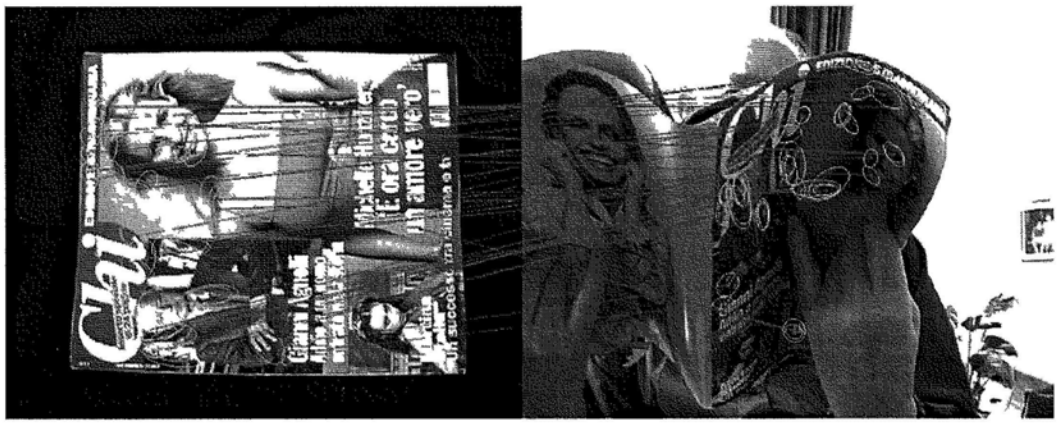
Non-rigid deformation

We now consider more difficult cases, where the model and test images present significant non-rigid deformation and severe clutters. The matching results are visually shown in Figure 4.7 and Figure 4.8, where (a) is the initial matching result, (b) and (c) shows the final correspondences obtained by the Hough clustering and by our geometric filter, respectively. The green ellipses plotted in (b) and (c) indicates the features' support regions. As we can see, the proposed fil-



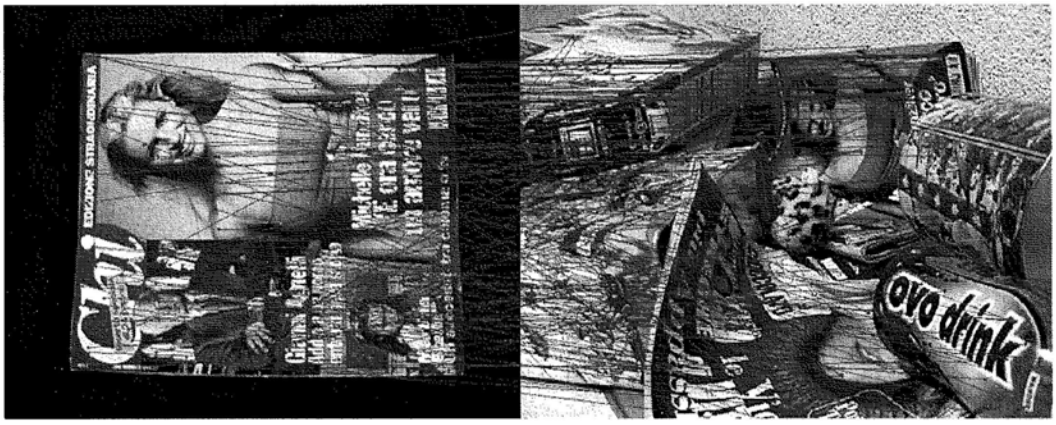
(a) Initial matches (219)

(b) By Hough clustering (14)

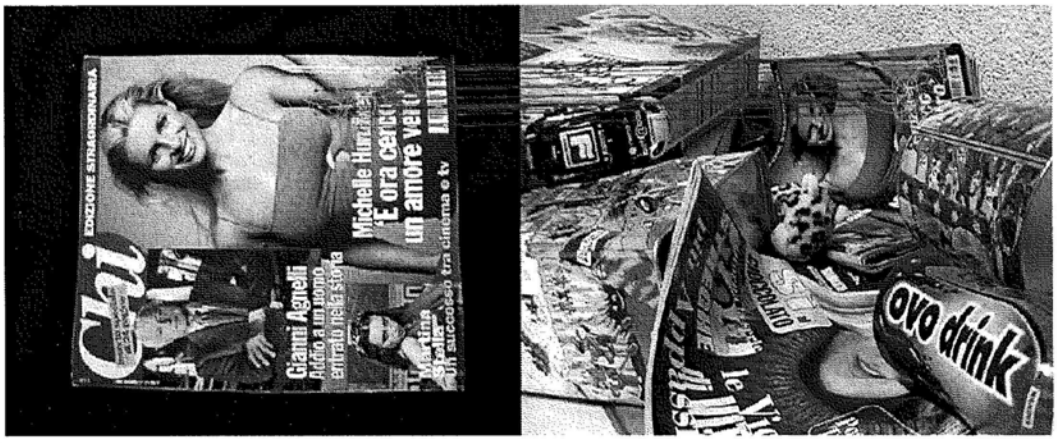


(c) By the proposed filter (48)

Figure 4.7 Matching for the Michelle model [36] Case 1



(a) initial matches (286)



(b) By Hough clustering (29)

(c) By the proposed filter (55)

Figure 4.8: Matching for the Michelle model [36]: Case 2 .

ter can preserve much more correct matches than the Hough clustering, despite the considerable non-rigid motion from the model image to the test image. Also note that there exist a large number of mismatches in the initial correspondence set, which demonstrates the strong power of the proposed filter in dealing with extensive clutters. One can also refer to [19] for their results on the same image set. The Affine Harris and Hessian features combined with our filter can produce more correct matches distributed in wider regions of the object compared with their methods. In addition, the runtime of the proposed filter (implemented in non-optimized Matlab code) measured on a Core Duo T2400 1.83GHz laptop is around 1.02s and 1.68s for case 1 in Figure 4.7 and case 2 in Figure 4.8, respectively, which is very efficient.

4.4 Global Refinement & Propagation for Affine Invariant Features

The success of the match propagation [36] encourages us to apply this method to sparse view rendering, where the initial matches obtained are still quite limited and sparse even when combining the three types of features. In practice, however, we found that this method works well for the textured regions, but usually fails to generate accurate correspondences for smooth regions without any salient textures. Some examples are given in Figure 4.9, where the red dots in (a), (b) and (c) show the keypoints detected in the reference image, their initial correspondences found in the target image and the refined results by Ferrari's method [35], respectively. The green ellipses show the support regions of the features that fail to be accurately matched by their refinement method, and the yellow lines indicate the regions' dominant gradient orientations. As we can see, most of these regions are not highly textured. As a consequence, the local appearance alone is not sufficiently powerful to guide the refinement. In this section, we propose a global method to refine and propagate the affine invariant features by incorporating the local appearance with the pair-wise affine consistency. The improved method can now successfully handle the smooth regions that occur frequently in object rendering.

Figure 4.9: Some matching examples where the global refinement method outperforms the local method. (a): features in the reference image; (b), (c), (d): corresponding features in the target image; (b): initial correspondences; (c): results by local refinement [35]; (d): results by global refinement.

4.4.1 Global Function for Match Refinement

Our global refinement method is based on the observation that the surface orientations change smoothly except for the case of depth discontinuities. This inspires us to impose some smooth constraint on the local affine transforms of nearby features. The original method of match refinement [35] tries to find for each match of features the best affine transform individually. We now attempt to simultaneously find the optimal set of affine transforms for all the matches by maximizing the global function defined in (4.8), where $(\mathbf{f}_R^k, \mathbf{f}_T^k) \in \Omega$ $(k = 1, \ldots, N)$. The three terms in (4.8) are described below.

$$\begin{aligned} F(\{\mathbf{A}\mathbf{f}^k\}) = & \sum_{k=1}^{N} \frac{1 + \mathrm{NCC}(S_R^k, \mathbf{A}\mathbf{f}^k S_R^k)}{2} \\ & + \lambda_1 \sum_{k=1}^{N} \exp(-\frac{\overline{\mathrm{DL}}(S_R^k, \mathbf{A}\mathbf{f}^k S_R^k)}{\gamma}) \\ & + \lambda_2 \sum_{k=1}^{N} \sum_{l=1}^{M^k} w^l AC(k,l) \end{aligned} \tag{4.8}$$

The first and the second terms are defined similarly as in [35]. They are used to measure the appearance similarity of the two matching regions S_R^k in I_R and $S_T^k = \mathbf{Af}^k S_R^k$ in I_T that are related by the affine transform $\mathbf{Af}^k$. The first term measures the intensity similarity and the second term measures the color similarity. Together they are called the data term. NCC is the widely used normalized cross-correlation between the regions' intensity patterns, and is normalized to [0,1]. $\overline{\mathrm{DL}}$ is the average pixel-wise Euclidean distance in the CIE-L*a*b* color space [152], which is better for measuring the color similarity than the RGB space from the perceptual standpoint. The three color bands are also normalized independently to achieve the illumination invariance to some extent. The equivalence between Euclidean and perceptual distances holds for small distances only, while the larger distance only indicates that the colors are perceptually different. By taking into account this fact, we choose the exponential measure ranging from 0 to 1 for the color term, where the control parameter γ is set to 50 through our experiments.

The major improvement of the global method lies in the introduction of the third term, i.e., the smoothness term, in the global function (4.8). In case of a smooth region (or a very small region) for which the local appearance is not discriminative enough to guide the correct match refinement, regularization can be achieved by further maximizing the affine consistency of the nearby matches. To this end, the predefined pair-wise affine consistency measure $AC(k,l)$ is conveniently employed to regularize the set of affine transforms $\{\mathbf{Af}^k\}$ in the global function. Besides, it is reasonable to use the normalized spatial distance $dis(\mathbf{f}_R^k, \mathbf{f}_R^l)$ for defining the neighborhood of an affine invariant feature $\mathbf{f}_R^k$ in the reference image, so that both the keypoint distance and the shape and size of the support regions are taken into account.

However, choosing all the nearby features for regularization may result in some over-smooth problem just as in the dense stereo. Features that are close in the image domain may belong to quite different physical surfaces (e.g., those around the surface discontinuities). Thus the corresponding affine transforms between two views can be completely different and uncorrelated. As smoothing across the depth discontinuity is highly undesired, we need to carefully select the associated

features for regularization. For the purpose of selective regularization, we define for each match $(\mathbf{f}_R^k, \mathbf{f}_T^k) \in \Omega$ its affinity set Ψ^k as

$$\Psi^k = \left\{ (\mathbf{f}_R^l, \mathbf{f}_T^l) \in \Omega \ (l \neq k) \mid dis(\mathbf{f}_R^k, \mathbf{f}_R^l) < th_1, \ AC(k,l) > th_3 \right\} \tag{4.9}$$

Basically, Ψ^k is a sub-set of the nearby matches of $(\mathbf{f}_R^k, \mathbf{f}_T^k)$ with $dis(\mathbf{f}_R^k, \mathbf{f}_R^l) < th_1$ in the reference image. By requiring $AC(k,l) > th_3$, the matches in the sub-set Ψ^k are associated with the affine transforms that are very similar to that of the match $(\mathbf{f}_R^k, \mathbf{f}_T^k)$, and hence they are called the affinities of $(\mathbf{f}_R^k, \mathbf{f}_T^k)$. In a sense, only the nearby matches with similar affine transforms are used for regularization. Thus two matches from the two surfaces with quite different orientations are naturally excluded from each other's affinity set. Note that if the initial matches are not totally erroneous, but a little inaccurate, the AC measure can be confidently used for indicating the initial affinity relationship.

In order to determine the affinity set, we use a simple thresholding on AC measures instead of cluster algorithms to speed up the process. Actually, we further limit the size of Ψ^k to save computations, since a small number of reliable affinities are sufficient to regularize the transforms. Thus the smoothness term in the global function (4.8) is composed of the weighted sum of the AC measures between a match $(\mathbf{f}_R^k, \mathbf{f}_T^k)$ and its M^k closest affinities. The strategy of choosing the weights w^l will be discussed in the next sub-section. Here note that the weights are normalized such that $\sum_{l=1}^{M^k} w^l = 1$. Thus the smoothness term also ranges in [0,1].

Finally, the two Lagrange parameters λ_1 and λ_2 in (4.8) are experimentally set to 2 and 1, respectively.

4.4.2 Implementation of the Global Optimization

The match refinement is now an expensive global optimization problem over a large set of affine transforms $\{\mathbf{Af}^k\}$ $(k = 1, \ldots, N)$. To make this problem tractable, we decompose the global optimization into the iterations of sequential maximization problems, each of which can be formulated as the maximization of the function $f(\mathbf{Af}^k)$ defined in (4.10) over the 6D space of a single affine transform $\mathbf{Af}^k$, with the associated affinity transforms $\{\mathbf{Af}^l \mid (\mathbf{f}_R^l, \mathbf{f}_T^l) \in \Psi^k\}$ fixed. The

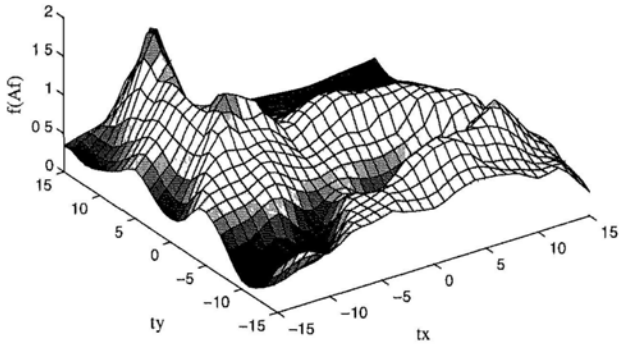


Figure 4.10: A typical case of $f(\mathbf{Af}^k)$ over the space of (t_x, t_y) with fixed (s_x, s_y, θ, h), where the gradient descent algorithm starting from $(0, 0)$ fail to reach the global maxima at $(-8, 12)$.

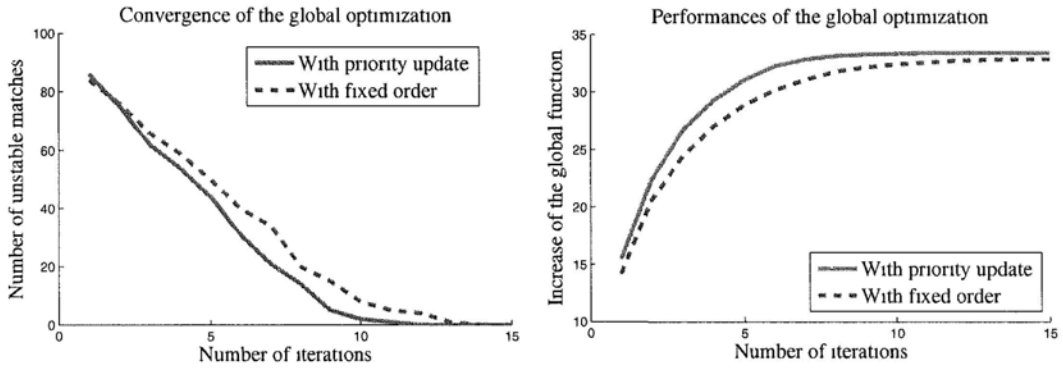
value of $f(\mathbf{Af}^k)$ (ranging from 0 to 4) provides a combined evaluation of a match $(\mathbf{f}_R^k, \mathbf{f}_T^k)$ in terms of both the appearance and the local geometry. Apparently, the choice of the weight w^l in (4.10) should reflect the confidence of using the affinity $(\mathbf{f}_R^l, \mathbf{f}_T^l)$ for regularization. Thus, a natural choice of w^l in the current iteration is the f value of $(\mathbf{f}_R^l, \mathbf{f}_T^l)$ in the last iteration. As the smoothness term is not available before the first iteration, the initial weights are set to the data term only. Then the weights are updated according to the f values after every iteration.

$$f(\mathbf{Af}^k) = \frac{1 + \mathrm{NCC}(S_R^k, \mathbf{Af}^k S_R^k)}{2} + \lambda_1 \exp(-\frac{\overline{\mathrm{DL}}(S_R^k, \mathbf{Af}^k S_R^k)}{\gamma}) + \lambda_2 \sum_{l \in \Psi^k} w^l AC(k, l) \tag{4.10}$$

Now the sub-problem is to maximize $f(\mathbf{Af}^k)$ over the 6D affine space $(t_x,t_y,s_x,s_y,\theta,h)$, where (t_x, t_y) is the 2D translation, (s_x, s_y) are the scales in x and y directions, and θ and h are the rotation and shear, respectively, which is similar to the individual local refinement [35] except for the introduction of the smoothness term. The original similarity function [35] generates highly non-convex spaces with frequent and diverse foldings. That is the reason why they propose a step-wise searching algorithm instead of using the gradient descent. Though an additional smoothness

term is introduced to the proposed $f(\mathbf{Af}^k)$ function, its behavior over the affine space is similar to the similarity function [35]. Figure 4.10 shows a typical case of $f(\mathbf{Af}^k)$ over the space of (t_x, t_y) with (s_x, s_y, θ, h) fixed, which presents a highly non-convex appearance. Actually, we have implemented an inverse compositional algorithm [8] for the maximization problem, but unfortunately the algorithm frequently gets stuck in undesired local maxima. On the other hand, full search of the affine space is too expensive to evaluate. With these considerations, we adopt the step-wise searching algorithm [35] to do the job. Note that in one iteration, only one step is made in the 6D space of $\mathbf{Af}^k$ to approach the maxima of $f(\mathbf{Af}^k)$, i.e., only one of the 6 parameters, which brings the largest increase to $f(\mathbf{Af}^k)$, is updated in an iteration. It is unnecessary to fully maximize the $f(\mathbf{Af}^k)$ in one iteration because the affinities of $\mathbf{Af}^k$ may also change through iterations and they will affect the regularization of $\mathbf{Af}^k$, i.e., affect the smoothness term of $f(\mathbf{Af}^k)$. We find that this modified step-wise algorithm performs much better than the gradient descent and generally produces satisfactory results in our experiments.

The order of the sequential maximization may take a crucial role in the behavior of the optimization because of the smoothness term imposed. Therefore, we need to carefully set the priority of the individual refinements. On one hand, matches with smaller f values should have higher priority, because they are more likely to be inaccurate matches that badly need to be regularized by others nearby. On the other hand, matches with high f values are probably correct and accurate, and as a result can be reliably used for regularizing other nearby features, which means that matches with reliable (high f value) affinities should be given preference in the individual refinements. Based on these considerations, the priority of a match or equivalently its associated affine transform is quantified by the mean f value of its affinities over the f value itself, as in (4.11). After every iteration, the priorities of the matches are updated according to their current f values, and a new order is determined for the sequential maximizations in the next iteration. Besides, in order to guarantee and accelerate the convergence, we stop examining a match $(\mathbf{f}_R^k, \mathbf{f}_T^k)$ in following iterations if the associated transform $\mathbf{Af}^k$ makes no change in one maximization attempt. Such a match is called a stable match. Finally, we only keep the stable matches that are good enough, specifically those



(a) (b)

Figure 4.11: Performances of the global optimizations using different order strategies.

with $f(\mathbf{Af}^k) > th_3$[1].

$$\text{priority}(\mathbf{Af}^k) = \frac{\sum_{l=1}^{M^k} f(\mathbf{Af}^l)}{M^k f(\mathbf{Af}^k)} \tag{4.11}$$

Figure 4.11 presents a performance comparison of the global optimizations with and without the priority update. The results are obtained by applying the global refinement to the 87 initial matches shown in Figure 4.9(a) and (b), and they are typical for other test images. From Figure 4.11(a) and (b), we can see that a reasonable order of sequential maximizations can enhance the convergence of the global optimization in terms of the speed and the achieved convergence value. One can also note that the differences are not that salient, meaning that the algorithm is not very sensitive to the order. The convergence of the whole algorithm typically takes 10 to 15 iterations though the number of initial matches may vary a lot. Also note from Figure 4.11(a) that the number of unstable matches drops quickly, which means that the computations keep decreasing proportionally through iterations. It is reported [35] that the local method typically takes 3 to 10 iterations for each match. Let N be the number of matches to be refined and let us approximate the curve in Figure 4.11(a) by a straight line. Thus the global method takes $5N$ to $7.5N$ evaluations of f[2] over the bounded

[1] th_3 is set to 2.1 in our experiments

[2] A combined measure of similarity and affine consistency

6D space, while the local method takes $3N$ to $10N$ evaluations of the similarity over the same space. Since evaluating the similarity is much more expensive than evaluating the affine consistency, the global and local methods will have similar computational cost.

In Figure 4.9(c) and (d), we select some features for a visual comparison of the matching results refined by the local and global methods, where we can clearly see the improvements brought by the global method in terms of the accuracy of keypoint location and region shape.

4.4.3 Global Match Propagation of Inner & Boundary Features

The match propagation aims to generate more feature correspondences from the initial seed matches. The basic idea is that if two adjacent features $\mathbf{f}_R^k$ and $\mathbf{f}_R^l$ in the reference image I_R are located on the same physical surface, they will be mapped onto $\mathbf{f}_T^k$ and $\mathbf{f}_T^l$ in the target image I_T by similar affine transforms. Let Θ be the set of seed matches and Γ_R be a set of newly added features in the reference image I_R. Recall that each match $(\mathbf{f}_R^k, \mathbf{f}_T^k) \in \Theta$ is associated with an affine transform $\mathbf{A}\mathbf{f}^k$. Thus, Ferrari et al. [36] proposed to choose for each new feature $\mathbf{f}_R^n \in \Gamma_R$ the best affine transform $\mathbf{A}\mathbf{f}^k$ from the seed matches Θ in terms of the similarity function, which is called the best propagation attempt and used to generate the initial match of $\mathbf{f}_R^n$, i.e., $\mathbf{f}_T^n \in \Gamma_T$ in the target image I_T with $S_T^n = \mathbf{A}\mathbf{f}^k S_R^n$. Then this initial match $(\mathbf{f}_R^n, \mathbf{f}_T^n)$ is further refined to generate more accurate correspondence. For more details, please refer to the early and major expansion in [36].

The target application of this work is object rendering rather than recognition or segmentation for [36]. There are several additional issues worth noting. First, since we try to construct a mesh model, which is coarse but sufficient to render the whole object defined by the given silhouette, we need to produce not only the matches of features within the object but also the matches along the object contour. These two kinds of features are named the inner and boundary features. As shown in Figure 4.12(a), the inner features are uniformly sampled within the object silhouette. They have circular support regions of radius R and are spaced

by R to densely cover the object. The odd and even rows offset one other by R as well. The boundary features are extracted in a way to capture the shape of the object silhouette, as shown in Figure 4.13. Specifically, the object contour is first approximated by a few straight line segments, and then a boundary keypoint is either selected as the intersection of two adjacent segments or uniformly sampled along a segment. The support region of a boundary feature is also bounded by a circle of radius R, but it covers only the region within the object silhouette. Actually some inner features cross the object silhouette as well, thus their support regions should also exclude the parts outside the silhouette. Here, the choice of the parameters, i.e., the region size and the sample rate, trades the mesh quality for computational cost. They could be adaptively selected according to the complexity of the target object or simply determined based on the users' desires.

Second, as mentioned before, smooth regions are ubiquitous for general objects, for which the local refinement usually fails to yield accurate matches due to the poor discriminative power of local appearance. To address the problem, we employ the combined f measure (4.10) to select the best propagation attempt instead of using the purely appearance-based similarity measure [35]. To refine the initial matches, we then apply the proposed global refinement method by taking into account both the appearance likelihood and geometry consistency. The detailed algorithm of global propagation is given as follows:

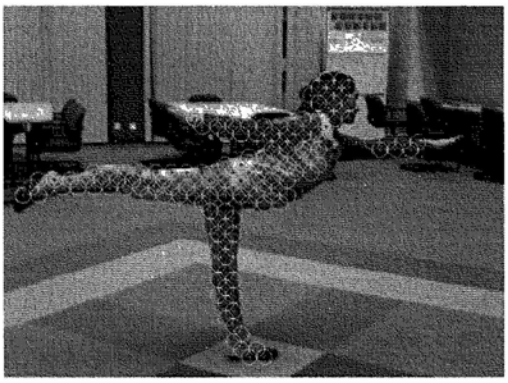
Initialization

For each feature $\mathbf{f}_R^n \in \Gamma_R$, initialize its match $\mathbf{f}_T^n \in \Gamma_T$ by following three steps:

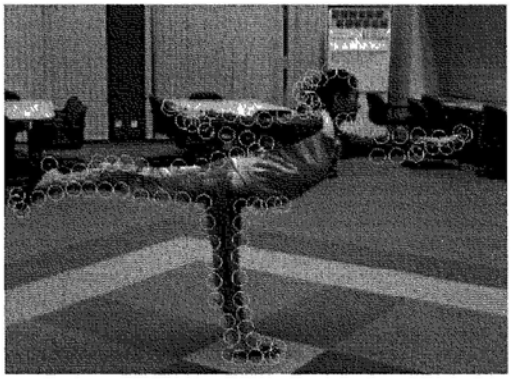
1. Find the nearby seed matches as the candidates. Specifically the candidate set is defined as

$$\Phi^n = \left\{ (\mathbf{f}_R^k, \mathbf{f}_T^k) \in \Theta \mid dis(\mathbf{f}_R^k, \mathbf{f}_R^n) < th_1 \right\}$$

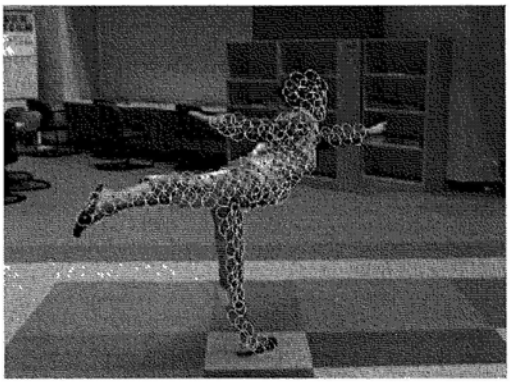
2. For each candidate match $(\mathbf{f}_R^k, \mathbf{f}_T^k) \in \Phi^n$, define an propagation attempt



(a) Inner features in I_R

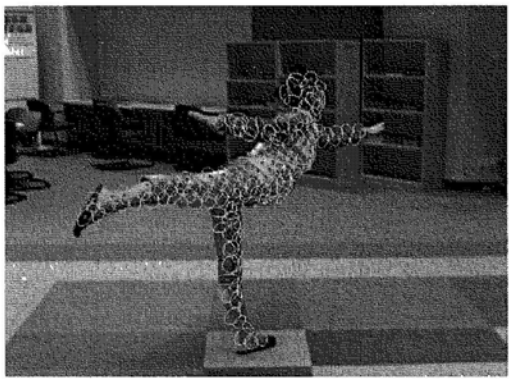


(b) Boundary features in I_R

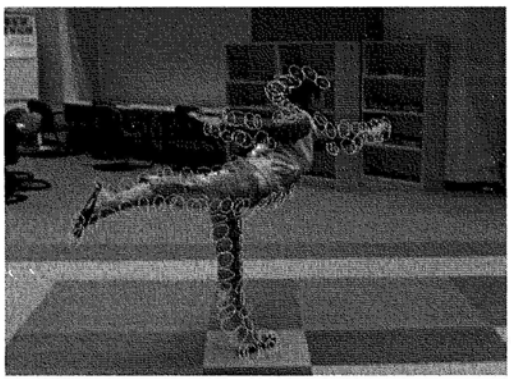


(c) Inner features in I_T (Global)

(d) Boundary features in I_T (Global)



(e) Inner features in I_T (Local)



(f) Boundary features in I_T (Local)

Figure 4.12: A comparison of the global and local match propagation.

Figure 4.13: Boundary features sampled along the object contour.

$(\mathbf{f}_R^n, \mathbf{f}_T^{nk})$ with $S_T^{nk} = \mathbf{A}\mathbf{f}^k S_R^n$. Then evaluate the quality of $(\mathbf{f}_R^n, \mathbf{f}_T^{nk})$ by

$$f(\mathbf{f}_R^n, \mathbf{f}_T^{nk}) = \frac{1 + \mathrm{NCC}(S_R^n, \mathbf{A}\mathbf{f}^k S_R^n)}{2} + \lambda_1 \exp(-\frac{\overline{\mathrm{DL}}(S_R^n, \mathbf{A}\mathbf{f}^k S_R^n)}{\gamma}) + \lambda_2 \sum_{l \in \Psi^k} w^l AC(k,l)$$

, where the affinity set Ψ^k (4.9) is selected from the seed matches Θ only;

3. Find the best propagation attempt as the initialization of match $(\mathbf{f}_R^n, \mathbf{f}_T^n)$.

$$(\mathbf{f}_R^n, \mathbf{f}_T^n) = \arg\max_k f(\mathbf{f}_R^n, \mathbf{f}_T^{nk})$$

Propagation

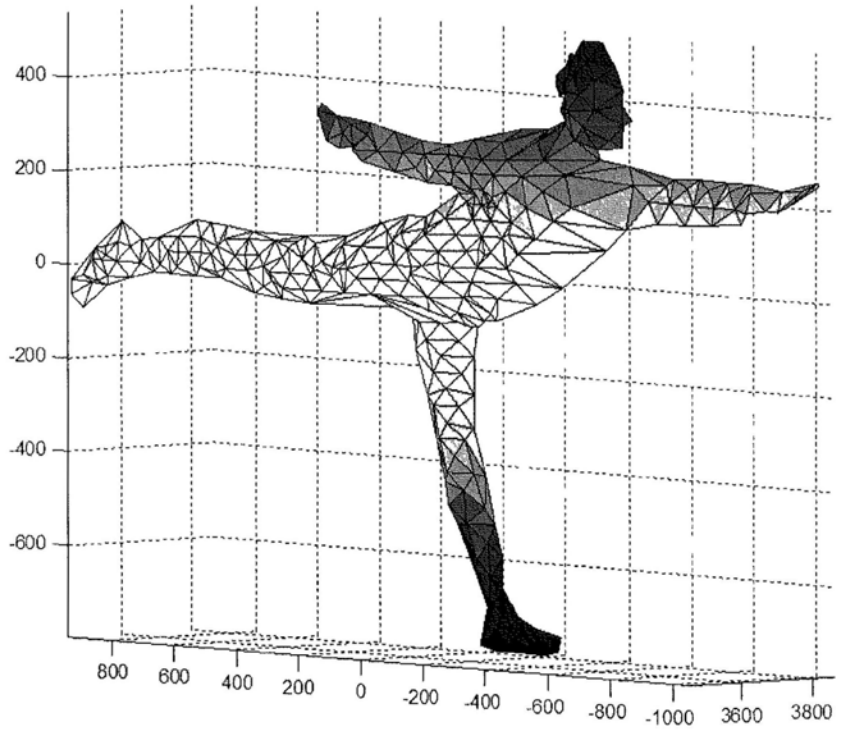
1. Initialize the matches of inner features Ω_I with the support of the initial matches Ω_O, i.e., $\Theta_I = \Omega_O$. Refine Ω_I by maximizing the global function $F(\{\mathbf{A}\mathbf{f}^k\})$ (4.8) over the space $(\mathbf{f}_R^k, \mathbf{f}_T^k) \in \Omega_I$;

2. Initialize the matches of boundary features Ω_B with the support of both Ω_O and Ω_I, i.e., $\Theta_B = \Omega_O \cup \Omega_I$. Refine Ω_B by maximizing the global function over the space $(\mathbf{f}_R^k, \mathbf{f}_T^k) \in \Omega_B$. Since the boundary features are sparsely distributed without sufficient overlap, the matches in Θ_B are also used to regularize Ω_B, i.e., the affinity set Ψ^k for a boundary feature is selected from $\Omega_B \cup \Omega_O \cup \Omega_I$ instead of Ω_B.

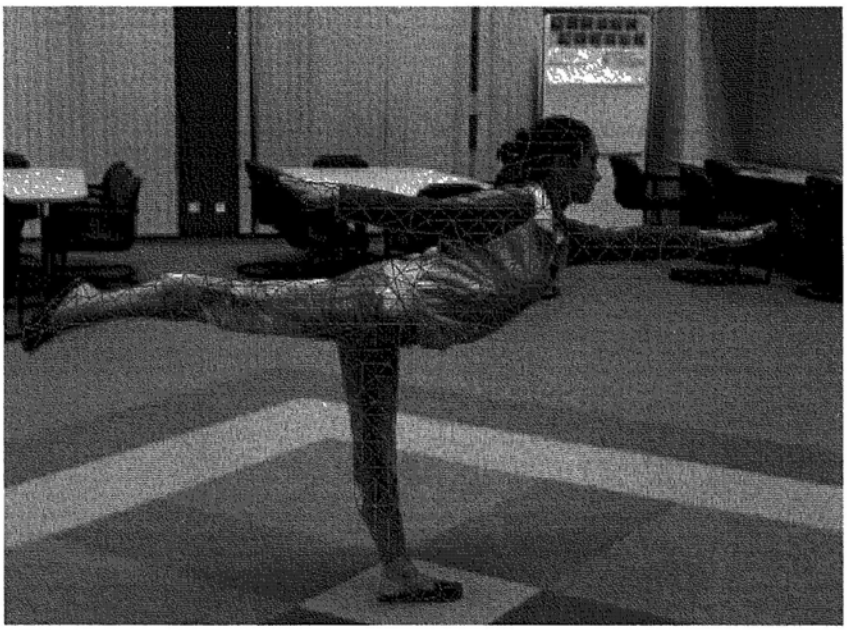
The performances of the global match propagation are visually shown in Figure 4.12. To demonstrate the improvements, we also present the results of the local propagation [36] for comparison. Figures 4.12(a) and (b) show the inner and boundary features sampled in the reference image I_R. Figures 4.12(c) and (d) show the corresponding features in the target image I_T obtained by the global propagation, and Figures 4.12(e) and (f) present the propagation results by the local method. Note that only good matches are kept and shown in Figure 4.12. We can observe that the features in I_T propagated by the global method present more smoothly changed facets thanks to the smoothness term imposed. The regularization also improves the accuracy of the keypoint location and the region shape, especially for the features in the smooth regions, such as the girl's arms, head and left leg. Another observation is that the benefits of regularization to the boundary features are more salient than to the inner features. This is because the local appearance of boundary features is generally less discriminative than the inner features, as the regions outside the silhouette are excluded from the features' support regions.

4.5 Triangulation and Texture Mapping

In the target application, the input images are captured from widely separated views. Thus it is impractical to require that all the features are consistently detected in all the input images. A match in two images is sufficient to compute a 3D keypoint with the cameras calibrated. Matches across more views in general produce more accurate 3D keypoints. Given a set of 3D points that lie on the object surface, one can define an exponential number of surface triangulations that fit the data. By taking into account the sparsity of features in wide baseline stereo and the possible non-smoothness of the object surface, we employ the image consistent surface triangulation [98] to find a particular mesh surface that is closest to the true object surface in the sense that the appearances of the meshes are most consistent across different views. Specifically, this method compares the re-projected images of the predicted surface with the actual images and uses this measure to select between competing triangulations and so overcome the surface ambiguity problem. The space of triangulation is fully searched by using edge



(a)



(b)

Figure 4.14: Triangulation result. (a): the 3D mesh model; (b): the triangular meshes projected onto the reference image.

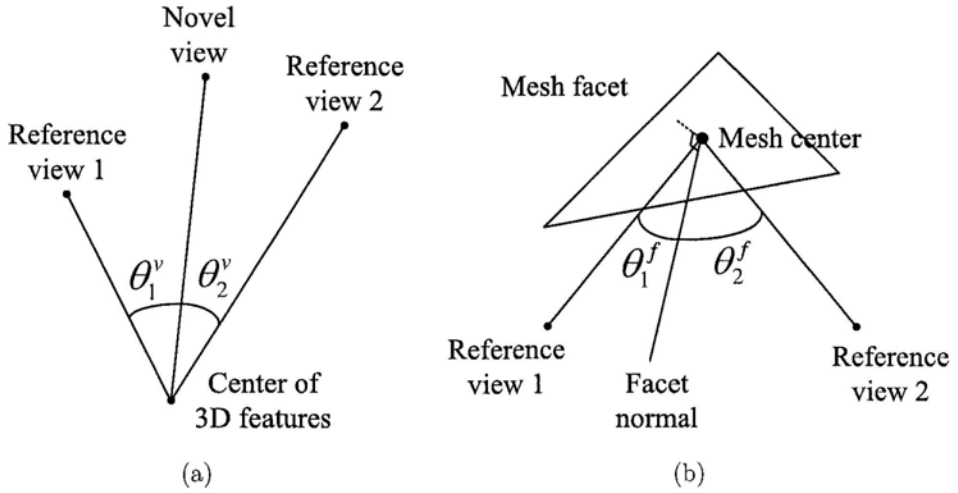


Figure 4.15: Double weighting. (a): view-based weighting; (b): facet-based weighting.

swaps. An example of the triangulation result is given in Figure 4.14, where (a) presents the 3D mesh model and (b) shows the triangular meshes projected onto the reference image.

To synthesize a novel view, two input views that are closest to it are selected as the reference views. If more input views are involved in rendering, the novel view may present undesired blur, because our mesh model is not as detailed as the one reconstructed by dense depth map in narrow baseline case. Recall that a 3D plane defines a homography between two views [52]. For each triangular mesh, i.e., a 3D facet, we map the textures from the two reference views onto the novel view using the respective local homographies, and then blend the two mapped textures together by a double weighting scheme.

The first weighting depends on the geometric relationship between the novel view and the reference views, and is the same for all the meshes. The weight of one reference view is set inversely proportional to the included angle between the novel viewpoint and the reference viewpoint, e.g., the angle θ_1^v for the reference view 1, shown in Figure 4.15 (a). It is thus called view-based weighting and is crucial to the smooth transition of successive viewpoints[1].

[1]Otherwise, there will be annoying flicker when experiencing the novel view navigation.

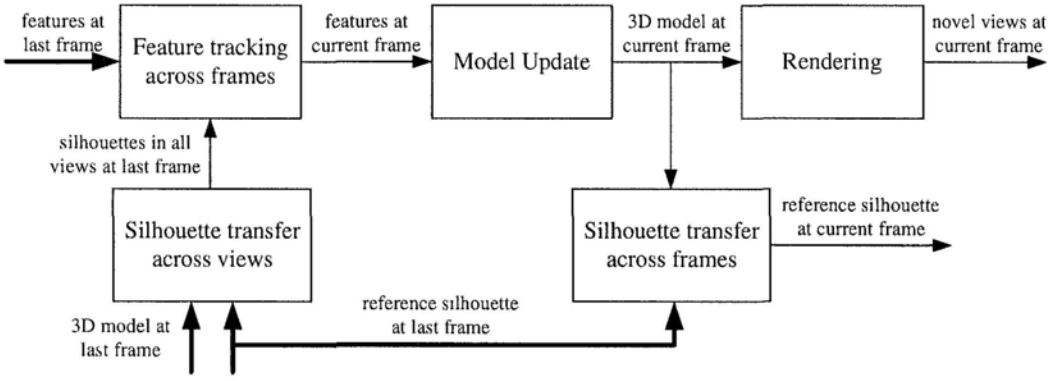


Figure 4.16: The framework of the proposed feature-based video rendering scheme

The second weighting is adapted to different mesh facets, namely facet-based weighting. Given a facet, the weight of one reference view is set inversely proportional to the angle between the facet normal and the line connecting the facet center and the camera center, for instance, the angle θ_1^f for the reference view 1, shown in Figure 4.15 (b). By employing the facet-based weighting, we can preserve for each individual facet more texture details from the reference view that gives a frontal observation of it.

4.6 Extension to Video Rendering

The proposed feature-based image rendering scheme can be efficiently extended to generate free-view video from two or more synchronized sequences captured from quite different viewpoints. The user's additional effort is still to provide the object silhouette in the reference view only at the first frame. Figure 4.16 shows the framework of the proposed feature-based video rendering scheme. In the following sub-sections, we will discuss the related key techniques, including feature tracking, silhouette transfer and model update.

4.6.1 Feature Tracking across Frames

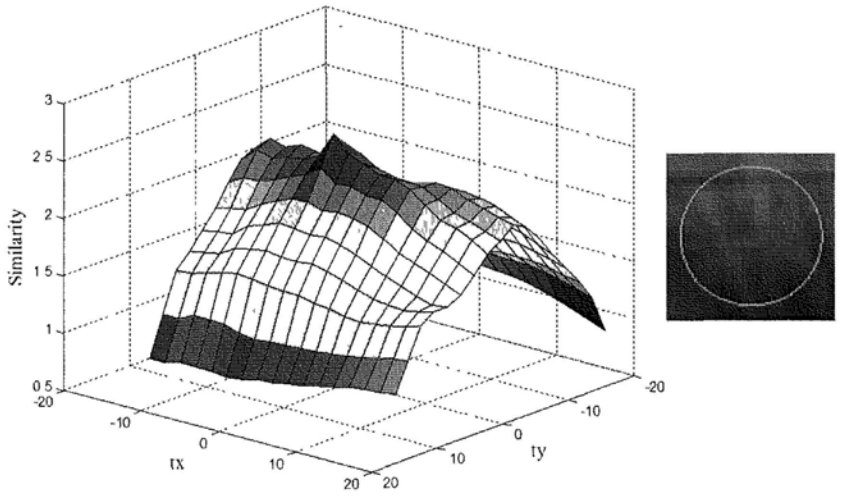
Feature tracking is based on the consistency across time and is performed in a way similar to the match refinement (Section 4.4.1). For each view, given a feature $\mathbf{f}_t^k$ at frame t, attempt is made to find an affine transform $\mathbf{A}\mathbf{f}_t^k$ that maps $\mathbf{f}_t^k$ to its correspondence at the next frame $t+1$, i.e., $\mathbf{f}_{t+1}^k = \mathbf{A}\mathbf{f}_t^k \mathbf{f}_t^k$. Since it is reasonable to assume that the images change little between successive frames, we start from the identity transform and search for the best transform in the 6D affine space to maximize the similarity function, similarly defined as the data term of (4.10).

$$sim(\mathbf{A}\mathbf{f}_t^k) = \frac{1 + \mathrm{NCC}(S_t^k, \mathbf{A}\mathbf{f}_t^k S_t^k)}{2} + \lambda_1 \exp(-\frac{\overline{\mathrm{DL}}(S_t^k, \mathbf{A}\mathbf{f}_t^k S_t^k)}{\gamma}) \qquad (4.12)$$

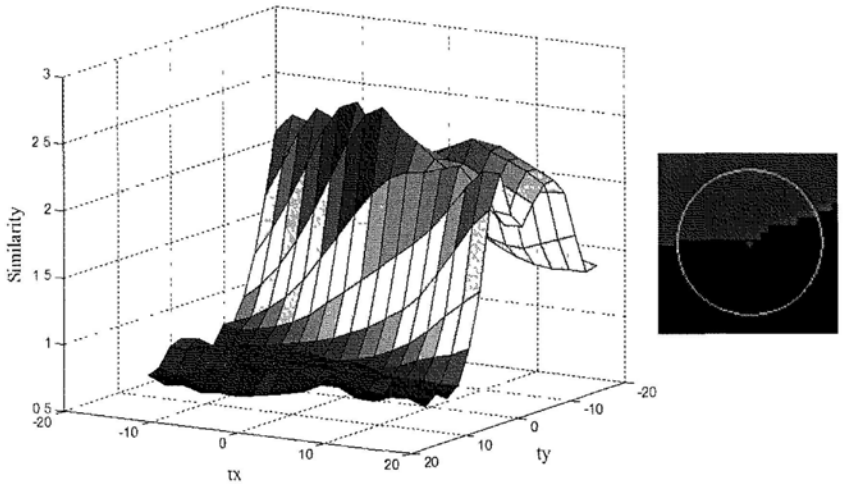
Compared to the step-wise search algorithm in match refinement, here we first perform a full search to find the best translation (t_x, t_y) in a relatively large search range with constant $(s_x, s_y, \theta, h) = (1, 1, 0, 0)$. We then keep the translation fixed and adjust the remaining four parameters in a very small range. Such a modification is based on the observation that in general case translation is the dominant local transform across frames, especially in case of fast movement. Thus we search for the translation much more carefully and at the same time allow for a small amount adjustment of scaling, shear and rotation for general purpose, even non-rigid deformation. Also note that actually the translation estimation is fast despite the large search range because no interpolation is required. This modification to the search algorithm thus saves a lot of computations for feature tracking and significantly accelerates the object rendering in successive frames.

Unfortunately, the local appearance again may be indiscriminative for weakly textured regions, and consequently the maximum of the similarity function in the 2D space of (t_x, t_y) may fail to indicate the best translation. Figure 4.17(b) shows a typical case, where the region of the boundary features has few textures, and the corresponding similarity function presents multiple local peaks with very close values. In comparison, for a region with sufficient textures shown in Figure 4.17(a), the similarity function is well behaved, i.e., the best translation is clearly suggested by a discriminative peak.

To address the tracking problem for the poorly textured features, we propose to predict their translations based on the well textured affinities, and then



(a) For a region with sufficient texture



(b) For a region with insufficient texture

Figure 4.17: Similarity function over the translation space (t_x, t_y) for different regions.

suppress the similarity values of the translations far away from the prediction so that a salient peak can stand out discriminatively just as in Figure 4.17(a). Specifically, the translation estimation becomes a two-pass process. In the first pass, we only determine the translations for the well textured features, for which there is no other peaks within 95% of the highest peak in the similarity func-

tion. In the second pass, we find for each remaining feature (poorly textured) the well textured affinities for the purpose of translation prediction. Since we have already known the 3D positions of the features' keypoints, we can conveniently use the 3D distance between the keypoints to measure the affinity relationship of the features, which is also a clever and efficient way to take into account the possible depth discontinuity. Here the affinities of a poorly textured feature are chosen as the three well textured features that are closest to it in terms of the 3D keypoint distance. Then the prediction is computed as the weighted sum of the translations of the affinities. The weights are set inversely proportional to the 3D keypoint distances and their sum is normalized to one.

4.6.2 Silhouette Transfer across Views and Frames

In order to continuously render the object, we need to track the features across frames in all the input views. Recall that we need the object silhouette to define meaningful support regions for the features close to the boundaries, and we only know the silhouette in the reference view at the first frame. Therefore, we have to transfer the object silhouette to other target views and across successive frames. To compose the silhouette in a target view, we map all the triangular meshes of the reference silhouette image onto this view by local homography of the mesh facet. Similarly, once the features have been successfully tracked across frames, we can generate the reference silhouette at the next frame by mapping the triangular meshes using local affine transform estimated from the matches of the three mesh vertices.

4.6.3 Model Update

As the object moves over time, some of its surfaces will be occluded and some will appear from occlusion, which means that the object model has to be updated across frames. Some features and their correspondences tend to have less similar appearances because the corresponding physical surfaces are gradually occluded due to the object movement. These features should be removed from the correspondence set by checking the consistency of appearance across views. To capture the newly appearing object surfaces, we apply [110] to refine the boundaries for

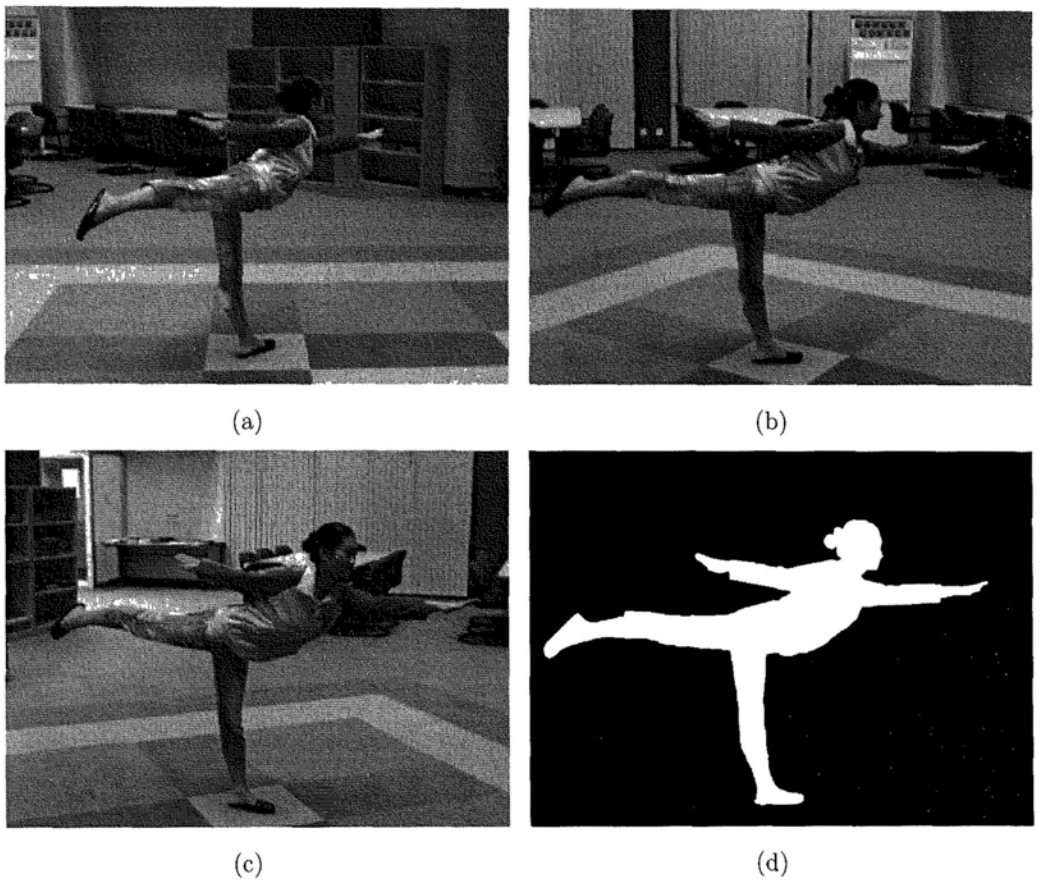
the silhouette image transferred from the last frame based on the image clue, colors. We then add more boundary features to depict the new object contour, and apply the propagation method to find their matches in other views. Finally, as the correspondence set changes, the triangulation should be updated as well to generate the new object model. Note that successive frames change little in practice. We can update the model every few frames to save a lot of computations, while having little influence on the rendering quality.

4.7 Experimental Results

The proposed method is tested on a few real image datasets that are fully calibrated. In practice, one can always make a tradeoff between the modeling accuracy and the manual effort by using precise pattern-based calibration or convenient self-calibration.

Yoga sequence (659×493): Figures 4.18(a), (b) and (c) show the three widely separated input images in this dataset. The angular spacing between two adjacent views, (a)-(b) and (b)-(c) is more than 50 degree. One can observe the distinct backgrounds in the input images, which inherently have few correspondences across views and consequently have little chance of being accurately reconstructed. Our effort is thus only to reconstruct and render the focused foreground, i.e., the girl doing yoga here. Even for the foreground, we can only partially reconstruct the object surface due to the scarcity of the input views. To this end, we choose the middle image Figure 4.18(b) as the reference image with the reference silhouette provided in Figure 4.18(d). The intermediate results of this experiment, including match propagation and triangulation, can be found in Figure 4.12 and Figure 4.14, respectively. Figures 4.19(a) and (b) show two typical virtual views of the foreground generated by the proposed method. The complete rendering results of a look-around sequence can be found in the supplementary material[1]. Despite the difficulties of widely separated views, scarce input images and weakly textured surfaces, the synthesized images are of high fidelity and the look-around sequence presents natural and smooth visual transition across views,

[1] http://www.ee cuhk.edu.hk/~chcui/



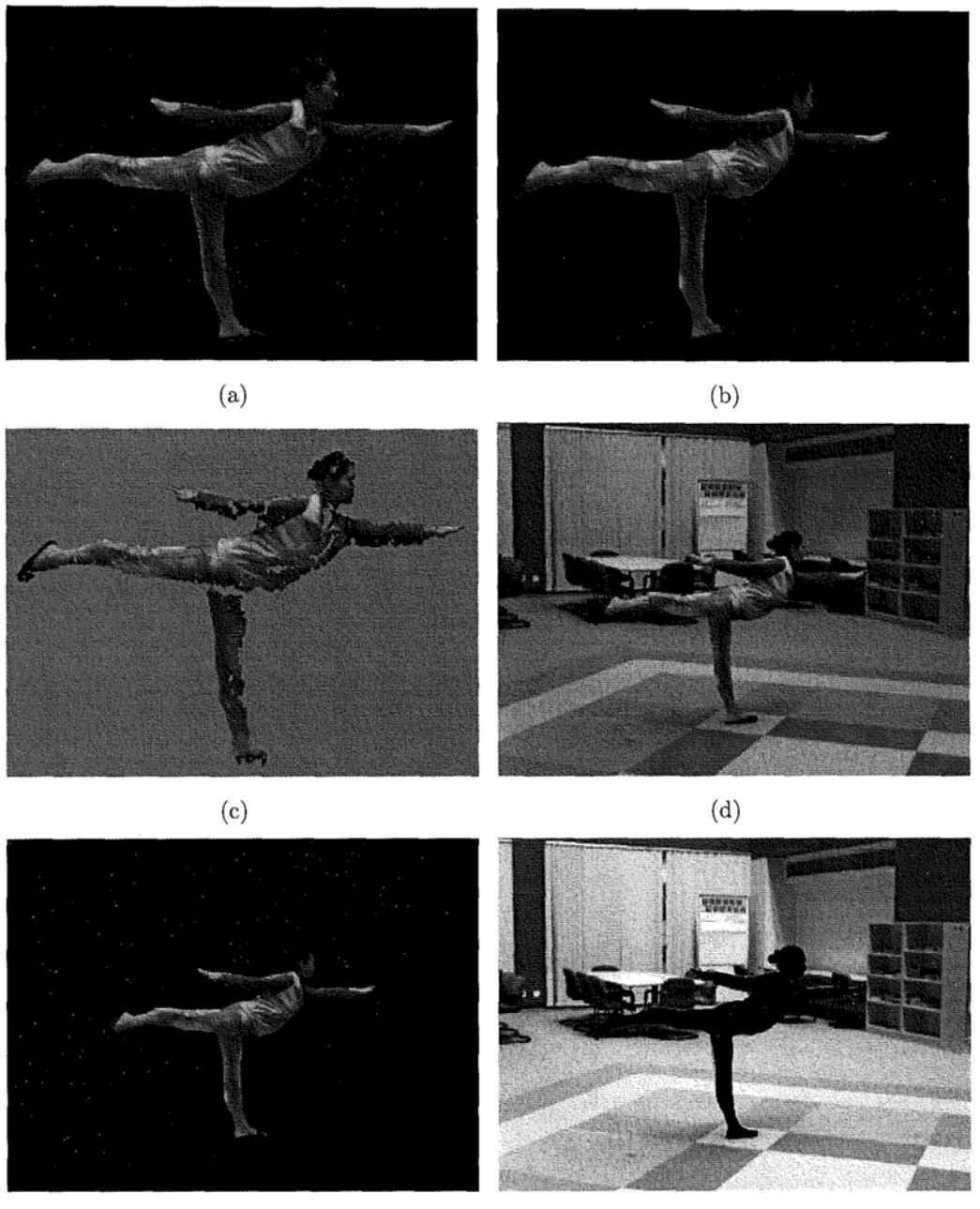
(a) (b)

(c) (d)

Figure 4.18: Inputs of *Yoga sequence*. (a), (b), (c): the three input images; (d): the silhouette of the reference image (b).

including the lighting. One may note from the look-around sequence that the virtual images observed from the leftmost and rightmost viewpoints appear a little weird. This is because of the incompleteness of the object model. As we can see from Figure 4.18, the leftmost and rightmost parts of the object, e.g., the girl's back in 4.18(a) and her left face in 4.18(c), inherently have no correspondences between images.

For comparison, we also present the rendering result of *Yoga* produced by PMVS [43] in Figure 4.19(c) (a full sequence of their results can be found in the supplementary material at the website). PMVS works with at least three input images. In this experiment all three silhouette images are provided for PMVS



(a) (b)

(c) (d)

(e) (f)

Figure 4.19: Rendering results of *Yoga sequence*. (a), (b): two virtual views synthesized by the proposed method; (c): results by PMVS [43] with three silhouette images; (d): image of ground truth. (e): the virtual image of (d) generated by the proposed method; (f): the difference image between (d) and (e).

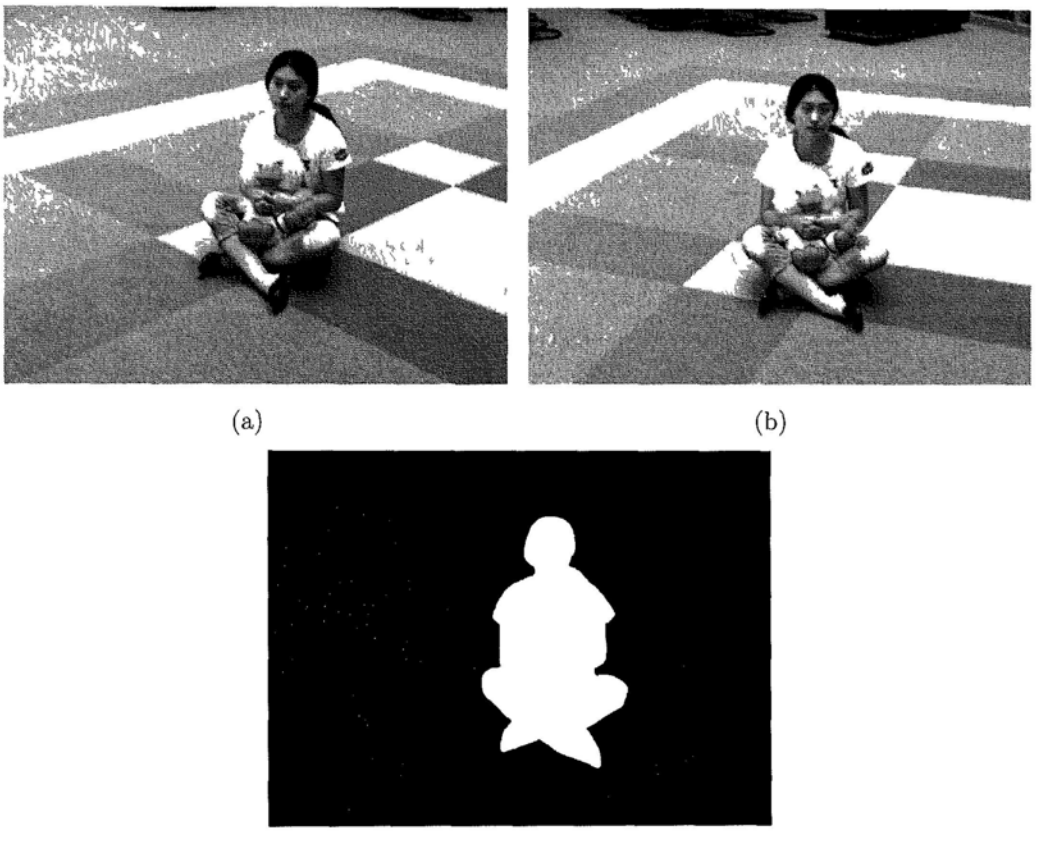
to achieve its best quality. From Figure 4.19(c), we can observe a few patches missing or inaccurately located and we believe that the main reason is the three abovementioned difficulties in this dataset. In this sense, the rendering results of our method are visually much more pleasant.

In Figure 4.19(d), an additional view is captured by our camera and used as the ground truth. Figure 4.19(e) shows the virtual image of the object synthesized by our method, which is observed from the same viewpoint as in Figure 4.19(d). The difference between the virtual image and the ground truth image is shown in Figure 4.19(f), where the darker the image the smaller the length of the color difference vector[1]. As we can see, besides the differences arising from some unmatched object boundaries, there are only a few very detailed differences that can hardly be observed. The main cause of these errors is that the features are sparsely sampled and hence the mesh model is not sufficient fine. Densely sampling more features in match propagation can surely improve the rendering quality of the details.

Girl sequence (659×493): In this experiment, only two wide baseline images are provided as inputs, as shown in Figures 4.20(a) and (b). The three difficulties mentioned above in Yoga sequence also exist here. Moreover, the object surface of Girl is more complex and there are a number of self-occlusions, e.g., the girl's right arm is occluded by the duck toy in Figure 4.20(a). Figure 4.20(c) shows the reference silhouette that defines the object of interest. Figure 4.21(a) shows the triangulation result projected onto the image in Figure 4.20(b). Figures 4.21(b) and (c) are two novel views of the object synthesized by our method. For a complete look-around rendering sequence, please refer to the relevant video in the supplementary material[2]. One may note from Figure 4.21(a) that the girl's right arm is modeled by an inaccurate triangular facet due to the occlusion in Figure 4.20(a). However, thanks to the facet-based weighting in our rendering scheme, the textures from Figure 4.20(a) contribute little to the texture blending of the arm. Therefore the virtual images of the arm are free of color contaminations as shown in Figures 4.21(b) and (c). The comparison to the ground truth is

[1]This measure is more sensitive than intensity difference.

[2]http://www.ee.cuhk.edu.hk/~chcui/



(a) (b)

(c)

Figure 4.20: Inputs of *Girl sequence*. (a), (b): the two input images; (c): the silhouette of the reference image (b).

presented in Figures 4.21(d), (e) and (f). The noticeable differences are either some object boundaries that are not included in the reference silhouette, e.g., the edges along the girl's legs and arms, or some small self-occlusions that cannot be simply modeled by the large-scale triangular facets.

Cityhall sequence (768×512): The original Cityhall sequence [128] consists of seven images of size 1536×1024. In this experiment, we choose only two of them as the input images shown in Figures 4.22(a) and (b), and downsize the images to one fourth of the original size. Different from Yoga and Girl, the Cityhall scene is highly textured and the two input images present strong scale change

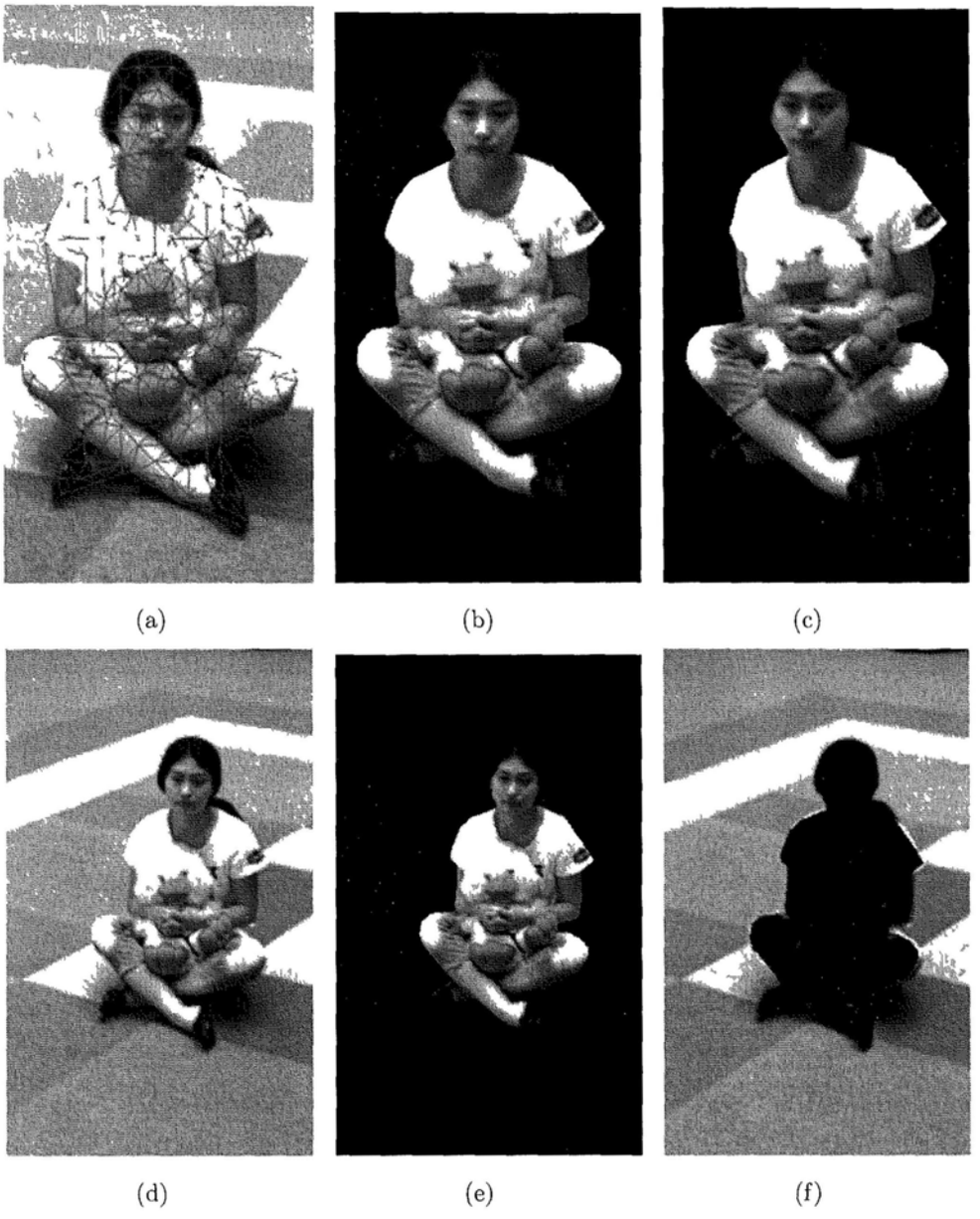
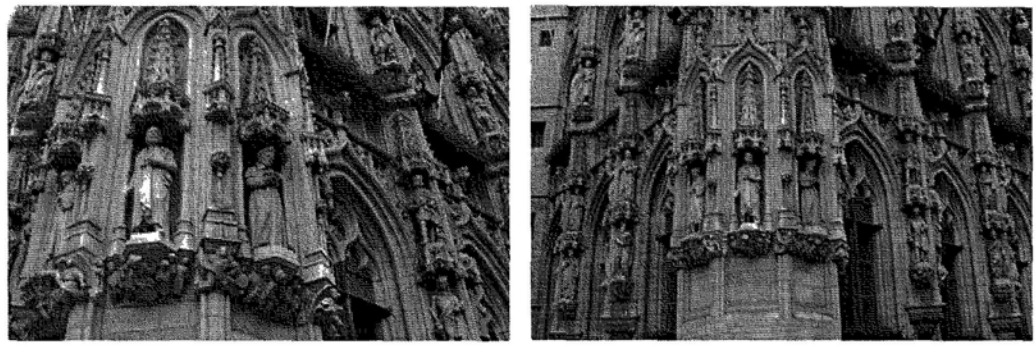
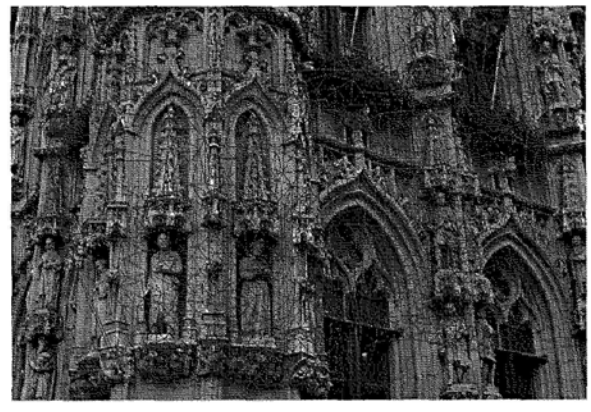


Figure 4.21: Rendering results of *Girl sequence*. (a) the triangulation result projected onto the reference image; (b), (c): two virtual views synthesized by the proposed method; (d) image of ground truth. (e): the virtual image of (d) generated by the proposed method; (f): the difference image between (d) and (e).

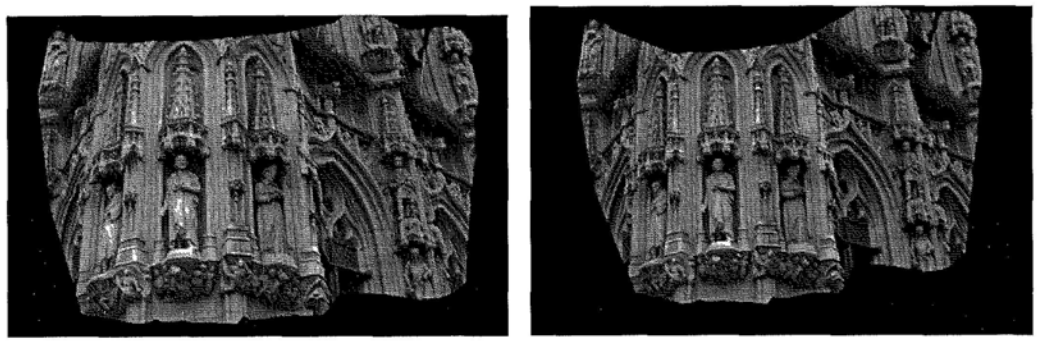


(a) (b)

Figure 4.22: Inputs of *Cityhall sequence*. (a), (b): the two input images.



(a)



(b) (c)

Figure 4.23: Rendering results of *Cityhall sequence*. (a): the triangulation result; (b), (c): two virtual views synthesized by the proposed method.

and perspective transformation. This time the whole image of Figure 4.22(a) is defined as the object of interest. Our method is then used to generate a virtual sequence that shows the visual transition from the viewpoint of Figure 4.22(a) to that of Figure 4.22(b) (please refer to the relevant video in the supplementary material). Figure 4.23(a) shows the triangulation result projected onto the image in 4.22(b). Figures 4.23(b) and (c) show two typical novel views synthesized by our method. It can be observed that most of the annoying distortions come from the occluded parts which cannot be matched and hence are simply modeled by a few coarse and inaccurate triangular facets. The visual quality of the remaining parts is quite satisfactory. Most of them are free of unpleasant artifacts even when the object of interest contains a lot of detailed textures and we have only two input images with significant viewpoint change.

To conclude, though our method may fail to capture very accurate details due to the coarse mesh model used[1], it is capable of generating realistic synthetic views from a very small set of strong wide baseline images. The sparsity of features, however, can simplify the 3D representation and reduce the data storage and memory cost.

Yoga video sequence (659×493): Another advantage of the proposed feature-based scheme is its easy and efficient extension to video rendering. We test the proposed feature-based video rendering scheme on the Yoga multi-view video sequences. The girl doing yoga was photographed from four different viewpoints simultaneously. The obtained four video sequences are synchronized. Three of them are used as the input video sequences and the left one is used as the ground truth for evaluation. The viewpoints of the three input sequences are shown in Figures 4.18(a), (b) and (c). The viewpoint of the ground truth is shown in Figure 4.19(d). Figure 4.24 present the video rendering results, where the color images are the synthesized images of successive frames generated by our method, and the grey images show the color differences between the virtual frames and those of ground truth. As can be observed in Figure 4.24, the quality of video rendering is close to the image rendering results in Figure 4.19. In Figure 4.25, the rendering

[1]Of course finer mesh model can be constructed by propagating more features with the price of more computations.

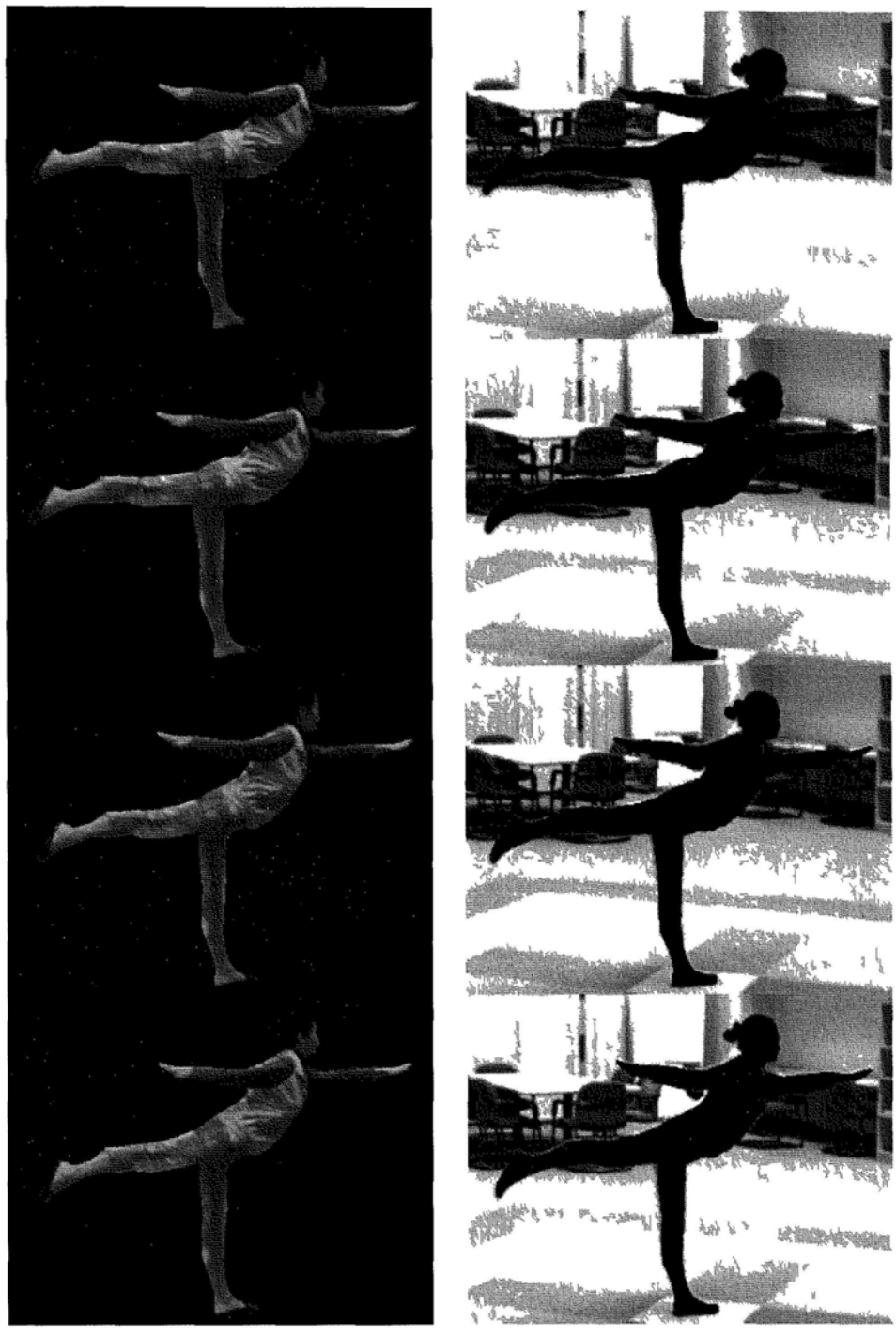


Figure 4.24 Rendering results of *Yoga Video Sequence*. Color images: the virtual images of successive frames synthesized by the proposed method. Grey images: the difference between the virtual images and the ground truth.

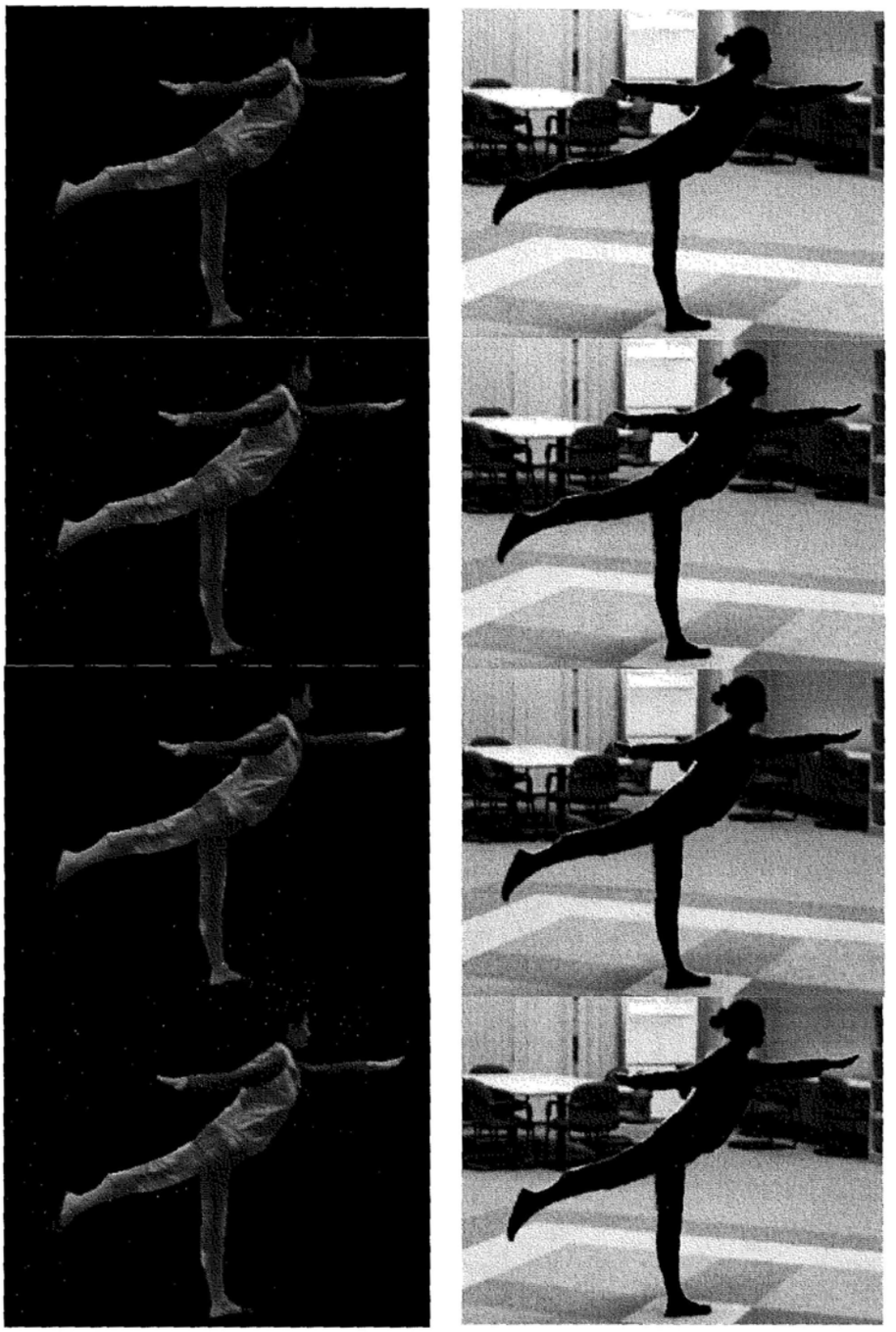


Figure 4.24: Rendering results of *Yoga Video Sequence* (*continue*). Color images: the virtual images of successive frames synthesized by the proposed method. Grey images: the difference between the virtual images and the ground truth.

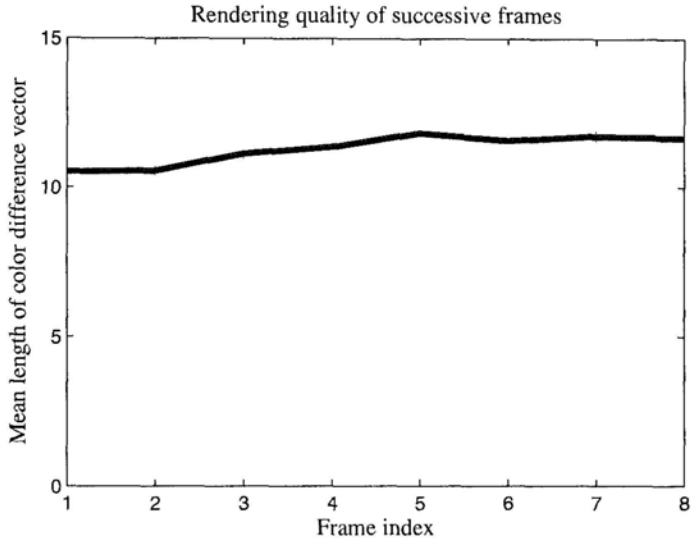


Figure 4.25: Rendering quality of successive frames

quality of successive frames is further quantified by the mean length of the color difference vectors over the whole object region. We can see a slight rising trend of the difference in successive frames mainly due to the accumulated errors of feature tracking. However, the deterioration of rendering quality is relatively slow and thus acceptable, which demonstrates the ability of our method in dealing with large motions and small non-rigid deformation for video-based rendering.

4.8 Conclusion

In this chapter, we have proposed a feature-based rendering scheme that can synthesize high quality novel views of the focused object from strong wide baseline images. Few images are required by the proposed method, so that the effort of capture can be largely reduced for users. We have shown that even two or three images are sufficient to generate a sequence of intermediate virtual views despite the difficulties of significant photometric and geometric changes between the widely separated images. Specifically, first, a bag of affine invariant features is employed to establish the initial sparse correspondences. An efficient geometric filter is then proposed to remove those erroneous matches based on the local

affine consistency. We then propose the global match refinement and propagation to refine the initial matches and propagate more correspondences of inner and boundary features that densely cover the surface of the object. By imposing the affine consistency of affinities in the global function, our method can successfully deal with the features in weakly textured regions, where the local appearance is not sufficiently discriminative to guide correct matches. Finally, a double weighting rendering scheme is introduced to synthesize the novel views based on the 3D mesh model constructed from the matched features. We have also extended the proposed feature-based image rendering scheme to free-view video synthesis based on feature tracking and model update. The efficient 3D representation by sparse features can surely help reduce the burden of data storage and memory cost for video applications.

Chapter 5

Conclusions

Multiview image-based rendering techniques have been intensely studied over the past few years. A number of high-quality algorithms have been developed to synthesize more realistic novel views with fewer cameras required. The work presented in this thesis focus on the development of a multiview system that is able to generate photorealistic novel views based on a very small set of widely separated images. Such a system has a number of desirable features, including low cost in system setup, flexibility and convenience in image capture, but also has a number of technical difficulties arising from the wide baseline setting, which have been discussed in this thesis. The contributions of the research work are summarized below and several directions for the future work are presented at the end of the chapter.

5.1 Contributions

5.1.1 Plane-Based Multi-Camera Calibration

In Chapter 2, we present an efficient plane-based multi-camera calibration method. Based on homography, we propose to estimate the pair-wise pose between neighboring cameras by using multiple images of different plane poses. Three different algorithms are introduced, including Two-Step, Three-Step and the non-linear methods with the orthogonal constraint imposed in different levels. Experimental results show that the combination of multiple images does enhance the accuracy

and stability of parameter estimation, especially for noisy data. The non-linear method initialized by the Three-Step results achieves the best performance, and the Three-Step method has a close performance with much lower computational complexity. The complete calibration of multiple cameras is achieved by registering all the pair-wise calibrated structures. Since only two neighboring cameras are required to simultaneously observe the pattern, our method is more practical and flexible for general calibration purpose, especially suitable for calibrating the widely spaced multi-camera system.

5.1.2 Accuracy Measure by Relative Deflection Angle

In Section 2.4 we present a novel accuracy measure, namely Relative Deflection Angle (RDA), to fairly evaluate and compare the accuracy of calibration results for different camera setups. The new metric is based on the deflection angles of projection rays, which take into account both the calibration inaccuracy and the inherent system errors. Compared with the mean-squared-distance measure, the RDA metric is much less sensitive to image resolution, camera focal length, baseline length and scene depth, which is validated by the experiments on different camera setups using real data.

5.1.3 Automatic Scale Selection for Corners & Junctions

In Section 3.2, we try to address the problem of automatic scale selection for image corners and junctions. Image neighborhood of these features usually contain the background or multiple foreground surfaces, thus cannot be correctly described by a single scale. Fan Laplacian-of-Gaussian (FLOG) kernel is proposed, which is capable of selecting the appropriate scales for independent image partitions that can represent meaningful physical surfaces attached to the corner or junction. Support for the proposed method is given in terms of theoretical investigation and experiments on real images.

5.1.4 Scale & Affine Invariant Fan Feature

In Chapter 3, we propose the scale and affine invariant Fan feature to represent and match the image structures near surface discontinuities, for which most existing feature detectors probably fail because they assume no surface discontinuity within the features' support regions. Our method is to divide the support region into multiple regular fan sub-regions, namely Fan features, each of which is supposed to represent a local smooth surface or background and is used as a distinct signature for matching purpose. In particular, we first propose a multi-scale detector to locate the salient keypoints on image edges. The image neighborhood of a keypoint is then divided into multiple Fan features based on edge-association, to provide robustness to surface discontinuity and background change. The Fan features are made scale invariant by using the automatic scale selection method based on FLOG (Section 3.2), and further made affine invariant based on the shape diagnosis of the mirror-predicted surface patch. Finally, we extend the SIFT descriptor to describe the image content of a fan-shaped region, which is called Fan-SIFT. Experiments of quantitative evaluation show that the Fan features have good repeatability under significant scale, viewpoint and background changes for general structured scenes and especially the low textured scenes. Moreover, our experiments on image matching and object rendering (Chapter 4) have demonstrated that the Fan features can contribute to the variety of the bag of features, especially because it can successfully detect and match those salient image structures near surface boundaries between wide baseline images.

5.1.5 Geometric Filter for Affine Invariant Features

Even though the invariant features can be extracted with high repeatability despite the strong viewpoint change between wide baseline images, their local appearance alone usually does not bring enough discriminative power to support a reliable matching, resulting in a relatively high number of outliers in the correspondence set. In Section 4.3, we present a novel and efficient geometric filter for general scale and affine invariant features. The proposed method detects the mismatches by examining the consistency of local affine geometry between

neighboring matches of affine invariant features. In particular, a pair-wise affine consistency measure is introduced by taking into account the consensus of both the keypoint location and the region size and shape. Experimental results show that the proposed geometric filter not only achieves a higher inlier ratio than the standard Hough clustering, but also presents superior robustness to severe clutters, significant viewpoint changes and non-rigid deformation.

5.1.6 Image-Based Rendering from Sparse Views

In Chapter 4, we present a novel image-based rendering method based on affine invariant features, which is capable of synthesizing photorealistic novel views from a very small number of wide baseline images, and hence can reduce the system cost and facilitate the image capture. In wide baseline setting, correspondences established by the invariant features are in general too sparse to cover the object surface for modeling purpose. In Section 4.4 we propose to refine and propagate the initial matches by optimizing a global function that takes into account both the appearance similarity and the geometric consistency, so that a quasi-dense set of correct matches can be produced even for weakly textured surfaces. Finally, a 3D mesh model with moderate degree of details can be constructed from the quasi-dense set of correspondences, based on which novel views can by synthesized in good quality by a double weighting texturing algorithm descried in Section 4.5. Experiments on difficult image datasets show that the proposed rendering method can generate visually pleasant free-view navigation using only two or three widely separated images as inputs, which outperforms the state-of-the-art object modeling and rendering method.

5.1.7 Video-Based Rendering from Sparse Views

In Section 4.6, we extend the feature-based image-based rendering scheme to render moving objects from wide baseline video sequences. To take advantage of the temporal coherence, an efficient two-pass tracking algorithm is proposed to match both the keypoints and their support regions across successive frames. The two-pass design enables successful tracking of low-textured surfaces and also accounts

for the non-rigid deformation. To minimize the additional user effort, we introduce the silhouette transfer to automatically identify the object boundary across different views and successive frames. To deal with the surface variation arising from object movement, the mesh model is adaptively updated by removing the inconsistent correspondences that cover the occlusions and adding new boundary features to depict the newly appeared surfaces. The feasibility of the video based rendering method is demonstrated by our experiments on real video sequences.

5.2 Future Work

This section concludes with a brief discussion of the future work.

- Improvement to Fan features: Though the Fan features possess good invariance to scale and viewpoint changes, the features are not extracted in a fully scale and viewpoint invariant manner. Actually we have chosen to make a few approximations and assumptions to largely reduce the computations of feature extraction. The method can surely be improved by further taking into account the scale and affine invariance through the whole pipeline. Firstly, the edges are now detected in a single fine scale to preserve details and ensure accuracy. Multi-scale edge detection may benefit some large scale edges and produce more valuable features, but will definitely will increase the computations. Secondly, the FLOG-based scale selection assumes known fan boundaries which are determined by edge association. However, the edge itself is not scale invariant. This requires a more sophisticated method to detect scale invariant fan regions, e.g., to simultaneously estimate the scale and fan directions by examining the FLOG response in a 3D space of scale and two fan directions, but with the cost of much more computations. Finally, to fully achieve scale and affine invariance for fan regions, we could iteratively perform scale selection and affine normalization like the Harris Affine and Hessian Affine features. The question is how to make a good tradeoff between the invariance and the complexity.

- Improvement to feature-based rendering: Firstly, we have not explicitly addressed the problem of self-occlusion in wide baseline stereo. Occluded regions have no chance to be matched and hence their 3D information is not available. In order to render a complete object, we now simply model them by connected triangular facets which of course are not accurate and probably will result in unpleasant artifacts in the virtual images. To address the problem, one way would be to infer the 3D information of occlusions by combining both the image clues and the estimated geometry. Another method would be purely image-based. We could rely on texture synthesis using a large set of training data to render the occluded regions. Secondly, the computational expense of the global refinement and propagation is similar to the local method. The processing time depends on the number and size of the features. In video applications, refinement and propagation are performed only for the first frame. The processing of remaining frames can be accelerated by using feature tracking instead. We intend to develop some fast algorithms for refinement, propagation and tracking to speed up the whole rendering system.

References

[1] Abdel-Aziz, Y. & Karara, H. (1971). Direct linear transformation from comparator coordinates into object space coordinates in close-range photogrammetry. In *Proceedings of the Symposium on Close-Range photogrammetry*, 1–18. 6

[2] Adelson, E. & Bergen, J. (1991). The plenoptic function and the elements of early vision. *Computational models of visual processing*, **1**. 18

[3] Aliaga, D. & Carlbom, I. (2001). Plenoptic stitching: A scalable method for reconstructing 3D interactive walkthroughs. In *ACM SIGGRAPH*, 443–450. 18

[4] Ashbrook, A., Thacker, N., Rockett, P. & Brown, C. (1995). Robust recognition of scaled shapes using pairwise geometric histograms. In *British Machine Vision Conference*, 503–512. 14

[5] Avidan, S., Moses, Y. & Moses, Y. (2004). Probabilistic multi-view correspondence in a distributed setting with no central server. *Europe Conf. Comp. Vision (ECCV)*, 428–441. 92

[6] Babaud, J., Witkin, A., Baudin, M. & Duda, R. (1986). Uniqueness of the Gaussian kernel for scale-space filtering. *IEEE Trans. Pattern Anal. Machine Intell.*, **8**, 26–33. 8

[7] Baker, P. & Aloimonos, Y. (2003). Calibration of a multicamera network. In *IEEE Conf. Comp. Vision & Pattern Recognition Workshop (CVPRW)*, vol. 7. 4, 27

[8] BAKER, S. & MATTHEWS, I. (2004). Lucas-kanade 20 years on: A unifying framework. *Int. Journal of Comp. Vision*, **56**, 221–255. 109

[9] BAUMBERG, A. (2000). Reliable feature matching across widely separated views. In *IEEE Conf. Comp. Vision & Pattern Recognition (CVPR)*, 1774. 7, 12, 14, 52, 68

[10] BELONGIE, S., MALIK, J. & PUZICHA, J. (2002). Shape matching and object recognition using shape contexts. *IEEE Trans. Pattern Anal. Machine Intell.*, **2**, 509–522. 14

[11] BOUGUET, J. (2008). Camera calibration toolbox for matlab http.//www.vision.caltech edu/bouguetj/calib_doc/. 3

[12] BOYKOV, Y., VEKSLER, O. & ZABIH, R. (2001). Fast approximate energy minimization via graph cuts. *IEEE Trans. Pattern Anal. Machine Intell.*, **23**, 1222–1239. 17

[13] BRADLEY, D., BOUBEKEUR, T. & HEIDRICH, W. (2008). Accurate multi-view reconstruction using robust binocular stereo and surface meshing. In *IEEE Conf. Comp. Vision & Pattern Recognition (CVPR)*, 1–8. 21, 90

[14] BROADHURST, A. & CIPOLLA, R. (2000). A statistical consistency check for the space carving algorithm. In *In Proc. ICCV*. 90

[15] BRODSKÝ, T. & FERMÜLLER, C. (2002). Self-calibration from image derivatives. *Int. Journal of Comp. Vision*, **48**, 91–114. 27

[16] BROWN, D. (1971). Close-range camera calibration. *Photogrammetric engineering*, **37**, 855–866. 6

[17] BROWN, M. & LOWE, D. (2003). Recognizing Panoramas. In *Int. Conf. Comp. Vision (ICCV)*, 1218–1225. 7

[18] CANNY, J. (1987). A computational approach to edge detection. *Readings in computer vision: issues, problems, principles, and paradigms*, 184. 64

[19] CARNEIRO, G. & JEPSON, A. (2007). Flexible spatial configuration of local image features. *IEEE Trans. Pattern Anal. Machine Intell.*, **29**, 2089–2104. 15, 16, 104

[20] CHEN, X., DAVIS, J. & SLUSALLEK, P. (2000). Wide area camera calibration using virtual calibration objects. In *IEEE Conf. Comp. Vision & Pattern Recognition (CVPR)*, vol. 2. 28

[21] CHOMAT, O., DE VERDIERE, V., HALL, D. & CROWLEY, J. (2000) Local scale selection for Gaussian based description techniques. In *Europe Conf. Comp. Vision (ECCV)*, 117–134. 10

[22] CROWLEY, J. (1981). A representation for visual information *PhD thesis, Carnegie Mellon University, USA*. 51

[23] CROWLEY, J. & PARKER, A. (1984). A representation for shape based on peaks and ridges in the difference of low-pass transform. *IEEE TRANS. PATTERN ANAL. MACH. INTELLIG.*, **6**, 156–170 51

[24] CSURKA, G., DANCE, C., FAN, L., WILLAMOWSKI, J. & BRAY, C. (2004). Visual categorization with bags of keypoints. In *Workshop on Statistical Learning in Computer Vision, ECCV*, vol. 1, 22. 7

[25] CUI, C. & NGAN, K. (2009). Automatic scale selection for corners and junctions. In *ICIP* 94

[26] DEBEVEC, P., TAYLOR, C. & MALIK, J. (1996). Modeling and rendering architecture from photographs: A hybrid geometry-and image-based approach. In *ACM SIGGRAPH*, 11–20. 20

[27] DEBEVEC, P., YU, Y. & BORSHUKOV, G. (1998). Efficient view-dependent image-based rendering with projective texture-mapping. In *Eurographics Workshop on Rendering*, 105–116. 20

[28] DICKINSON, S., PENTLAND, A. & ROSENFELD, A. (1992) From volumes to views: An approach to 3-D object recognition. *Image Understanding*, **55**, 130–154. 15

[29] DORKÓ, G. & SCHMID, C. (2003). Selection of scale-invariant parts for object class recognition. In *Int. Conf. Comp. Vision (ICCV)*. 7

[30] ESTEBAN, H. *et al.* (2004). Silhouette and stereo fusion for 3D object modeling. *Computer Vision and Image Understanding*, **96**, 367–392. 3, 4, 90

[31] FAUGERAS, O. (1993). *Three-dimensional computer vision: a geometric viewpoint*. the MIT Press. 26

[32] FAUGERAS, O. & TOSCANI, G. (1986). The calibration problem for stereo In *Proceedings*, 15 6, 40

[33] FAUGERAS, O., LUONG, Q. & MAYBANK, S. (1992). Camera self-calibration: Theory and experiments. In *Europe Conf. Comp. Vision (ECCV)*, 321–334. 27

[34] FERGUS, R., PERONA, P. & ZISSERMAN, A. (2003). Object class recognition by unsupervised scale-invariant learning. In *IEEE Conf. Comp. Vision & Pattern Recognition (CVPR)*, vol. 2. 7

[35] FERRARI, V., TUYTELAARS, T. & VAN GOOL, L. (2003). Wide-baseline multiple-view correspondences. In *IEEE Conf. Comp. Vision & Pattern Recognition (CVPR)*, vol. 1. 22, 92, 104, 105, 106, 108, 109, 110, 112

[36] FERRARI, V., TUYTELAARS, T. & VAN GOOL, L. (2004). Simultaneous object recognition and segmentation by image exploration. In *Europe Conf. Comp. Vision (ECCV)*. xii, 7, 16, 22, 92, 93, 102, 103, 104, 111, 115

[37] FISCHLER, M. & BOLLES, R. (1981). Random sample consensus. A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, **24**, 381–395. 5, 28, 75

[38] FITZGIBBON, A., WEXLER, Y. & ZISSERMAN, A. (2005). Image-based rendering using image-based priors. *Int. Journal of Comp. Vision*, **63**, 141–151. 91

[39] FLORACK, L., TER HAAR ROMENY, B., KOENDERINK, J. & VIERGEVER, M. (1991). General intensity transformations and second order invariants In *Seventh Scandinavian Conf. Image Analysis*, 338–345. 14

[40] FLORACK, L., TER HAAR ROMENY, B., KOENDERINK, J. & VIERGEVER, M. (1992). Scale and the differential structure of images. *Image and Vision Computing*, **10**, 376–388. 8

[41] FORSSEN, P. & LOWE, D. (2007). Shape descriptors for maximally stable extremal regions. In *Int. Conf. Comp. Vision (ICCV)*, 59–73. 53

[42] FREEMAN, W. & ADELSON, E. (1991). The design and use of steerable filters. *IEEE Trans. Pattern Anal. Machine Intell.*, **13**, 891–906. 14

[43] FURUKAWA, Y. & PONCE, J. (2007). Accurate, dense, and robust multiview stereopsis. In *IEEE Conf. Comp. Vision & Pattern Recognition (CVPR)*, 1–8. 22, 92, 123, 124

[44] GARGALLO, P., PRADOS, E. & STURM, P. (2007). Minimizing the reprojection error in surface reconstruction from images. In *Int. Conf. Comp. Vision (ICCV)*. 90

[45] GOESELE, M., CURLESS, B. & SEITZ, S. (2006). Multi-view stereo revisited. In *IEEE Conf. Comp. Vision & Pattern Recognition (CVPR)*, vol. 2. 90

[46] GORTLER, S., GRZESZCZUK, R., SZELISKI, R. & COHEN, M. (1996). The lumigraph. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, 43–54. 18, 90

[47] GRIGORESCU, C., PETKOV, N. & WESTENBERG, M. (2004). Contour and boundary detection improved by surround suppression of texture edges. *Image and Vision Computing*, **22**, 609–622. 64

[48] HABBECKE, M. & KOBBELT, L. (2007). A surface-growing approach to multi-view stereo reconstruction. In *IEEE Conf. Comp. Vision & Pattern Recognition (CVPR)*, 1–8. 22, 92

[49] HAN, J. & PARK, J. (2000). Contour matching using epipolar geometry. *IEEE Trans. Pattern Anal. Machine Intell.*, **22**, 358–370. 17

[50] HARRIS, C. & STEPHENS, M. (1988). A combined corner and edge detector. In *Alvey vision conference*, vol. 15, 50. 10, 64

[51] HARTLEY, R. & STURM, P. (1994). Triangulation. In *ARPA Image Understanding Workshop*, 957–966. 45

[52] HARTLEY, R. & ZISSERMAN, A. (2003). *Multiple view geometry in computer vision*. Cambridge Univ Pr. 5, 28, 75, 91, 117

[53] HE, L. & SHUM, H. (1999). Rendering with concentric mosaics. In *ACM SIGGRAPH*, 299–306. 17, 18

[54] HEIKKILA, J. & SILVEN, O. (1997). A four-step camera calibration procedure with implicit imagecorrection. In *IEEE Conf. Comp. Vision & Pattern Recognition (CVPR)*, 1106–1112. 4, 6

[55] HERNANDEZ, C., SCHMITT, F. & CIPOLLA, R. (2007). Silhouette coherence for camera calibration under circular motion. *IEEE Trans. Pattern Anal. Machine Intell.*, **29**, 343–349. 5

[56] HEYDEN, A. & ASTROM, K. (1996). Euclidean reconstruction from constant intrinsic parameters. In *IEEE Int. Conf. on Pattern Recognition (ICPR)*, vol. 13, 339–343. 27

[57] HORNUNG, A. & KOBBELT, L. (2006). Hierarchical volumetric multi-view stereo reconstruction of manifold surfaces based on dual graph embedding. In *IEEE Conf. Comp. Vision & Pattern Recognition (CVPR)*, vol. 1. 90

[58] IHRKE, I., AHRENBERG, L. & MAGNOR, M. (2004). External camera calibration for synchronized multi-video systems. *Journal of WSCG*, **12**, 537–544. 28

[59] IRANI, M., HASSNER, T. & ANANDAN, P. (2002) What does the scene look like from a scene point? *Europe Conf. Comp. Vision (ECCV)*, 600–602. 91

[60] JOHNSON, A. & HEBERT, M. (1997). Object recognition by matching oriented points. In *IEEE Conf. Comp. Vision & Pattern Recognition (CVPR)*, 684–689. 13

[61] JOLLIFFE, I. (2002). *Principal component analysis*. Springer verlag. 14

[62] KADIR, T. & BRADY, M. (2001). Scale, saliency and image description. *Int. Journal of Comp. Vision*, **45**, 83–105. 52

[63] KE, Y. & SUKTHANKAR, R. (2004). PCA-SIFT: A more distinctive representation for local image descriptors. In *IEEE Conf. Comp. Vision & Pattern Recognition (CVPR)*, 511–517. 14

[64] KOENDERINK, J. (1984). The structure of images. *Biological cybernetics*, **50**, 363–370. 8

[65] KOENDERINK, J. & VAN DOORN, A. (1987). Representation of local geometry in the visual system. *Biological cybernetics*, **55**, 367–375. 14, 52

[66] KUTULAKOS, K. & SEITZ, S. (2000) A theory of shape by space carving. *Int. Journal of Comp. Vision*, **38**, 199–218. 90

[67] LABATUT, P., PONS, J. & KERIVEN, R. (2007). Efficient multi-view reconstruction of large-scale scenes using interest points, delaunay triangulation and graph cuts. In *Int. Conf. Comp. Vision (ICCV)*, 1–8. 22, 92

[68] LAVEST, J., VIALA, M. & DHOME, M. (1998). Do we really need an accurate calibration pattern to achieve a reliable camera calibration? *Europe Conf. Comp. Vision (ECCV)*. 5

[69] LAZEBNIK, S., SCHMID, C. & PONCE, J. (2004). Semi-local affine parts for object recognition. In *British machine vision conference*, vol. 2, 959–968. 16

[70] LAZEBNIK, S., SCHMID, C. & PONCE, J. (2005). A sparse texture representation using local affine regions. *IEEE Trans. Pattern Anal. Machine Intell.*, **27**, 1265–1278. 13, 52

[71] LEVOY, M. & HANRAHAN, P. (1996). Light field rendering. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, 42. 18, 90

[72] LHUILLIER, M. & QUAN, L. (2002). Match propagation for image-based modeling and rendering. *IEEE Trans. Pattern Anal. Machine Intell*, **24**, 1140–1146. 22, 92

[73] LHUILLIER, M. & QUAN, L. (2005). A quasi-dense approach to surface reconstruction from uncalibrated images. *IEEE Trans. Pattern Anal. Machine Intell*, 418–433. 22, 92

[74] LINDEBERG, T. (1990). Scale-space for discrete signals. *IEEE Trans. Pattern Anal. Machine Intell.*, **12**, 234–254. 8

[75] LINDEBERG, T. (1993). *Scale space theory in computer vision*. Kluwer Academic Publishers. 9

[76] LINDEBERG, T. (1998). Edge detection and ridge detection with automatic scale selection. *Int. Journal of Comp. Vision*, **30**, 79–116. 10

[77] LINDEBERG, T. (1998). Feature detection with automatic scale selection. *Int. Journal of Comp. Vision*, **30**, 79–116. 9, 10, 51, 56, 58, 92

[78] LINDEBERG, T. & GARDING, J. (1997). Shape-adapted smoothing in estimation of 3-D shape cues from affine deformations of local 2-D brightness structure* 1. *Image and Vision Computing*, **15**, 415–434. 11, 52, 68

[79] LOWE, D. (1999). Object recognition from local scale-invariant features. In *Int. Conf. Comp. Vision (ICCV)*, 1150. 7, 52, 53

[80] LOWE, D. (2004). Distinctive image features from scale-invariant keypoints. *Int. Journal of Comp. Vision*, **60**, 91–110. 10, 13, 15, 16, 52, 72, 74, 81, 92, 95, 100

[81] LUONG, Q. & FAUGERAS, O. (1997). Self-calibration of a moving camera from point correspondences and fundamental matrices. *Int. Journal of Comp. Vision*, **22**, 261–289. 27

[82] MAIRE, M., ARBELÁEZ, P., FOWLKES, C. & MALIK, J. (2008). Using contours to detect and localize junctions in natural images In *IEEE Conf. Comp. Vision & Pattern Recognition (CVPR)*, 1–8. 64, 66

[83] MALM, H. & HEYDEN, A. (2001). Stereo Head Calibration from a planar Object. In *IEEE Conf. Comp. Vision & Pattern Recognition (CVPR)*, vol. 2, 657–662. 27

[84] MALM, H. & HEYDEN, A. (2006). Extensions of plane-based calibration to the case of translational motion in a robot vision setting. *IEEE Transactions on Robotics*, **22**, 322–333. 27

[85] MARTIN, D., FOWLKES, C. & MALIK, J. (2004). Learning to detect natural image boundaries using local brightness, color, and texture cues. *IEEE Trans. Pattern Anal. Machine Intell.*, **26**, 530–549. 64

[86] MARTINEC, D. & PAJDLA, T. (2007). Robust rotation and translation estimation in multiview reconstruction. In *IEEE Conf. Comp. Vision & Pattern Recognition (CVPR)*, 1–8. 5

[87] MATAS, J., CHUM, O., URBAN, M. & PAJDLA, T. (2004). Robust wide-baseline stereo from maximally stable extremal regions. *Image and Vision Computing*, **22**, 761–767. 7, 12, 52, 53, 68, 75, 92

[88] MATUSIK, W., BUEHLER, C., RASKAR, R., GORTLER, S. & MCMILLAN, L. (2000). Image-based visual hulls. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, 369–374. 90

[89] MAYBANK, S. & FAUGERAS, O. (1992). A theory of self-calibration of a moving camera. *Int. Journal of Comp. Vision*, **8**, 123–151. 27

[90] MELTZER, J. & SOATTO, S. (2008). Edge descriptors for robust wide-baseline correspondence. In *IEEE Conf. Comp. Vision & Pattern Recognition (CVPR)*, 1–8. 54, 55

[91] MIKOLAJCZYK, K. (2002). Detection of local features invariant to affine transformations. *PhD thesis, Institut National Polytechnique de Grenoble, France*. 11, 52

[92] MIKOLAJCZYK, K. & SCHMID, C. (2004). Scale & affine invariant interest point detectors. *Int. Journal of Comp. Vision*, **60**, 63–86. 10, 12, 52, 53, 68, 71, 75, 81, 92, 94, 100

[93] MIKOLAJCZYK, K. & SCHMID, C. (2005). A performance evaluation of local descriptors. *IEEE Trans. Pattern Anal. Machine Intell.*, **27**, 1615–1630. 14, 21, 51, 72, 73, 74, 75, 91, 94, 95

[94] MIKOLAJCZYK, K., ZISSERMAN, A. & SCHMID, C. (2003). Shape recognition with edge-based features. In *British Machine Vision Conference*. 52, 54, 55

[95] MIKOLAJCZYK, K., TUYTELAARS, T., SCHMID, C., ZISSERMAN, A., MATAS, J., SCHAFFALITZKY, F., KADIR, T. & GOOL, L. (2005) A comparison of affine region detectors. *Int. Journal of Comp. Vision*, **65**, 43–72. xii, 21, 51, 53, 75, 76, 78, 91, 94, 96, 97

[96] MINDRU, F., MOONS, T. & VAN GOOL, L. (1999). Recognizing color patterns irrespective of viewpoint and illumination. In *IEEE Conf. Comp. Vision & Pattern Recognition (CVPR)*, vol. 1, 368–373. 52

[97] MORE, J. (1977). The Levenberg-Marquardt algorithm: implementation and theory. *Numerical analysis*, 105–116. 39

[98] MORRIS, D. & KANADE, T. (2000). Image-consistent surface triangulation. In *cvpr*, 1332 115

[99] NISTER, D. (2004). An efficient solution to the five-point relative pose problem. *IEEE Trans. Pattern Anal. Machine Intell.*, **26**, 756–770. 5

[100] OBDRZALEK, S. & MATAS, J. (2002). Object recognition using local affine frames on distinguished regions. In *British Machine Vision Conference*, 113–122. 7, 12, 53, 68

[101] OHTA, Y. & KANADE, T. (1985). Stereo by intra-and inter-scanline search using dynamic programming. *IEEE Trans. Pattern Anal. Machine Intell.*, **7**, 139–154. 17

[102] OKUTOMI, M., KATAYAMA, Y. & OKA, S. (2002). A simple stereo algorithm to recover precise object boundaries and smooth surfaces. *Int. Journal of Comp. Vision*, **47**, 261–273. 17

[103] OLIVEIRA, M., BISHOP, G. & MCALLISTER, D. (2000). Relief texture mapping. In *ACM SIGGRAPH*, 359–368. 19

[104] PELEG, S. & HERMAN, J. (1997). Panoramic mosaics by manifold projection. In *IEEE Conf. Comp. Vision & Pattern Recognition (CVPR)*, 338–343. 18

[105] POLLEFEYS, M., VAN GOOL, L., VERGAUWEN, M., VERBIEST, F., CORNELIS, K., TOPS, J. & KOCH, R. (2004). Visual modeling with a hand-held camera. *Int. Journal of Comp. Vision*, **59**, 207–232. 5

[106] PONS, J., KERIVEN, R. & FAUGERAS, O. (2005). Modelling dynamic scenes by registering multi-view image sequences. In *IEEE Conf. Comp. Vision & Pattern Recognition (CVPR)*. 90

[107] PRINCE, S., CHEOK, A., FARBIZ, F., WILLIAMSON, T., JOHNSON, N., BILLINGHURST, M. & KATO, H. (2002). 3d live: Real time captured content for mixed reality. In *Proceedings of the 1st International Symposium on Mixed and Augmented Reality*, 7. 28

[108] RADKE, R., RAMADGE, P., KULKARNI, S. & ECHIGO, T. (2003). Efficiently synthesizing virtual video. *IEEE Transactions on Circuits and Systems for Video Technology*, **13**, 325–337. 90

[109] RECHE-MARTINEZ, A., MARTIN, I. & DRETTAKIS, G. (2004). Volumetric reconstruction and interactive rendering of trees from photographs. In *ACM SIGGRAPH*, 727. 17

[110] ROTHER, C., KOLMOGOROV, V. & BLAKE, A. (2004). Grabcut: Interactive foreground extraction using iterated graph cuts. In *ACM SIGGRAPH*, 314. 121

[111] ROTHGANGER, F., LAZEBNIK, S., SCHMID, C. & PONCE, J. (2003). 3d object modeling and recognition using affine-invariant patches and multi-view spatial constraints. In *IEEE Conf. Comp. Vision & Pattern Recognition (CVPR)*, vol. 2. 7

[112] SCHAFFALITZKY, F. & ZISSERMAN, A. (2002). Multi-view matching for unordered image sets, or How do I organize my holiday snaps?. *Europe Conf. Comp. Vision (ECCV)*, 414–431. 14, 20, 91, 92

[113] SCHARSTEIN, D. (1996). Stereo vision for view synthesis. In *IEEE Conf. Comp. Vision & Pattern Recognition (CVPR)*, 852–857. 19

[114] SCHARSTEIN, D. & SZELISKI, R. (2002). A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *Int. Journal of Comp. Vision*, **47**, 7–42. 17, 90

[115] SCHMID, C. & MOHR, R. (1997). Local grayvalue invariants for image retrieval. *IEEE Trans. Pattern Anal. Machine Intell.*, **19**, 530–535. 15

[116] SEITZ, S. & DYER, C. (1996). View morphing. In *ACM SIGGRAPH*, 21–30. 19

[117] SEITZ, S. & DYER, C. (1997). Photorealistic scene reconstruction by voxel coloring. In *IEEE Conf. Comp. Vision & Pattern Recognition (CVPR)*, 1067–1073. 90

[118] SEITZ, S., CURLESS, B., DIEBEL, J., SCHARSTEIN, D. & SZELISKI, R. (2006). A comparison and evaluation of multi-view stereo reconstruction algorithms. In *IEEE Conf. Comp. Vision & Pattern Recognition (CVPR)*, vol. 1. 3, 91, 92

[119] SHADE, J., GORTLER, S., HE, L. & SZELISKI, R. (1998). Layered depth images. In *ACM SIGGRAPH*, 231–242. 19

[120] SHEN, D. & IP, H. (1997). Generalized affine invariant image normalization. *IEEE Trans. Pattern Anal. Machine Intell.*, **19**, 431–440. 12, 53, 68

[121] SHUM, H., CHAN, S. & KANG, S. (2007). *Image-based rendering*. Springer-Verlag New York Inc. 16

[122] SINHA, S., MORDOHAI, P. & POLLEFEYS, M. (2007). Multi-view stereo via graph cuts on the dual of an adaptive tetrahedral mesh. In *Int. Conf. Comp. Vision (ICCV)*, vol. 1. 90

[123] SIVIC, J. & ZISSERMAN, A. (2003). Video Google: A text retrieval approach to object matching in videos. In *Int. Conf. Comp. Vision (ICCV)*, 1470–1477. 7

[124] SIVIC, J. & ZISSERMAN, A. (2004). Video data mining using configurations of viewpoint invariant regions. In *IEEE Conf. Comp. Vision & Pattern Recognition (CVPR)*, vol. 1. 7

[125] SIVIC, J., SCHAFFALITZKY, F. & ZISSERMAN, A. (2006). Object level grouping for video shots. *Int. Journal of Comp. Vision*, **67**, 189–210. 7

[126] SNAVELY, N., SEITZ, S. & SZELISKI, R. (2008). Modeling the world from internet photo collections. *Int. Journal of Comp. Vision*, **80**, 189–210. 20, 91

[127] STEIN, A. & HEBERT, M. (2005). Incorporating background invariance into feature-based object recognition. In *Seventh Workshop on Applications of Computer Vision*. 53, 54, 72

[128] STRECHA, C., TUYTELAARS, T. & VAN GOOL, L. (2003). Dense matching of multiple wide-baseline views. In *Int. Conf. Comp. Vision (ICCV)*, vol. 2, 1194–1201. 20, 22, 91, 92, 126

[129] STRECHA, C., FRANSENS, R. & VAN GOOL, L. (2004). Wide-baseline stereo from multiple views: a probabilistic account. In *IEEE Conf. Comp. Vision & Pattern Recognition (CVPR)*, vol. 1. 91

[130] STRECHA, C., FRANSENS, R. & VAN GOOL, L. (2006). Combined depth and outlier estimation in multi-view stereo. In *IEEE Conf. Comp. Vision & Pattern Recognition (CVPR)*, vol. 2. 90

[131] STURM, P. & TRIGGS, B. (1996). A factorization based algorithm for multi-image projective structure and motion. In *Europe Conf. Comp. Vision (ECCV)*, 709–720. 27

[132] SUN, J., SHUM, H. & ZHENG, N. (2003). Stereo matching using belief propagation. *IEEE Trans. Pattern Anal. Machine Intell.*, **25**, 787–800. 17

[133] SVOBODA, T., MARTINEC, D. & PAJDLA, T. (2005). A convenient multicamera self-calibration for virtual environments. *Presence: Teleoperators & Virtual Environments*, **14**, 407–422. 27, 48

[134] TANIMOTO, M. (2006). Overview of free viewpoint television. *Signal Processing: Image Communication*, **21**, 454–461. 90

[135] TOLA, E., LEPETIT, V. & FUA, P. (2008). A fast local descriptor for dense matching. In *Proc. CVPR*. 21, 92

[136] TRAN, S. & DAVIS, L. (2006). 3d surface reconstruction using graph cuts with surface constraints. *Europe Conf. Comp. Vision (ECCV)*, 219–231. 90

[137] TRIGGS, B. (1997). Autocalibration and the absolute quadric. In *IEEE Conf. Comp. Vision & Pattern Recognition (CVPR)*, 609–614. 27

[138] TRIGGS, B. (1998). Autocalibration from planar scenes. In *Europe Conf. Comp. Vision (ECCV)*, 89–105. 5, 27

[139] TRIGGS, B., MCLAUCHLAN, P., HARTLEY, R. & FITZGIBBON, A. (1883). Bundle adjustmenta modern synthesis. *Vision Algorithms· Theory and Practice*, 153–177. 5

[140] TSAI, R. (1987). A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses. *IEEE Journal of robotics and Automation*, **3**, 323–344. 3, 4, 6, 26, 39

[141] TUYTELAARS, T. & VAN GOOL, L. (2004). Matching widely separated views based on affine invariant regions. *Int. Journal of Comp. Vision*, **59**, 61–85. 7, 11, 15, 52, 54, 55, 75, 92

[142] UESHIBA, T. & TOMITA, F. (2003). Plane-based calibration algorithm for multi-camera systems via factorization of homography matrices. In *Int. Conf. Comp. Vision (ICCV)*, vol. 2, 966–973. 27

[143] VAN GOOL, L., MOONS, T. & UNGUREANU, D. (1996). Affine/photometric invariants for planar intensity patterns. In *Europe Conf. Comp. Vision (ECCV)*, 642–651. 15

[144] VEDALDI, A. & SOATTO, S. (2005). Features for recognition: Viewpoint invariance for non-planar scenes. In *Int. Conf. Comp. Vision (ICCV)*, vol. 2. 54, 55

[145] VEDALDI, A. & SOATTO, S. (2006). Viewpoint induced deformation statistics and the design of viewpoint invariant features: Singularities and occlusions. In *Europe Conf. Comp. Vision (ECCV)*, 360–373. 54, 55

[146] VEDULA, S., BAKER, S., RANDER, P., COLLINS, R. & KANADE, T. (2005). Three-dimensional scene flow. *IEEE Trans. Pattern Anal. Machine Intell*, 475–480. 91

[147] VOGIATZIS, G., HERNÁNDEZ, C., TORR, P. & CIPOLLA, R. (2007). Multiview stereo via volumetric graph-cuts and occlusion robust photo-consistency. *IEEE Trans. Pattern Anal. Machine Intell*, **29**, 2241–2246. 90

[148] WENG, J., COHEN, P. & HERNIOU, M. (1992). Camera calibration with distortion models and accuracy evaluation. *IEEE Trans. Pattern Anal. Machine Intell.*, **14**, 965–980. 6, 30, 40, 41

[149] WILBURN, B., JOSHI, N., VAISH, V., TALVALA, E., ANTUNEZ, E., BARTH, A., ADAMS, A., HOROWITZ, M. & LEVOY, M. (2005). High performance imaging using large camera arrays. *ACM Transactions on Graphics (TOG)*, **24**, 776. 91

[150] WONG, K. & CIPOLLA, R. (2004). Reconstruction of sculpture from its profiles with unknown camera positions. *IEEE Transactions on Image Processing*, **13**, 381–389. 5

[151] WOODFORD, O. & FITZGIBBON, A. (2005). Fast image-based rendering using hierarchical image-based priors. In *Proc. BMVC*, vol. 1, 260–269. 91

[152] WYSZECKI, G. & STILES, W. (1982). Color Science: Concepts and Methods, Quantitative Data and Formula. *New York: John Wiley and Sons*. 106

[153] YAO, J. & CHAM, W. (2006). 3D modeling and rendering from multiple wide-baseline images by match propagation. *Signal Processing: Image Communication*, **21**, 506–518. 22, 92

[154] ZAHARESCU, A., BOYER, E. & HORAUD, R. (2007). Transformesh: a topology-adaptive mesh-based approach to surface evolution. *ACCV*, 166–175. 90

[155] ZHANG, J., MARSZA, M., LAZEBNIK, S. & SCHMID, C. (2007). Local features and kernels for classification of texture and object categories: A comprehensive study. *Int. Journal of Comp. Vision*, **73**, 213–238. 51

[156] ZHANG, Z. (2000). A flexible new technique for camera calibration. *IEEE Trans. Pattern Anal. Machine Intell.*, **22**, 1330–1334. 4, 6, 26, 28, 31, 32, 33, 43, 48

[157] ZHANG, Z. & HANSON, A. (1995). Scaled Euclidean 3D reconstruction based on externally uncalibrated cameras. In *iscv*, 37. 33, 34

[158] ZITNICK, C. & KANADE, T. (2000). A cooperative algorithm for stereo matching and occlusion detection. *IEEE Trans. Pattern Anal. Machine Intell.*, **22**, 675–684. 17

[159] ZITNICK, C. & KANG, S. (2007). Stereo for image-based rendering using image over-segmentation. *Int. Journal of Comp. Vision*, **75**, 49–65. 17, 91