國立交通大學

資訊科學與工程研究所

博士論文

透視不變性於電腦視覺之應用

Using Perspective Invariants in Some Computer Vision Applications

研究生：高肇宏

指導教授：莊仁輝　教授

中華民國九十六年六月

透視不變性於電腦視覺之應用
Using Perspective Invariants in Some Computer Vision Applications

研究生：高肇宏　　Student：Jau-Hong Kao

指導教授：莊仁輝 教授　　Advisor：Jen-Hui Chuang

國立交通大學
資訊工程系
博士論文

A Dissertation
Submitted to Department of Computer Science
College of Electrical Engineering and Computer Science
National Chiao Tung University
in partial Fulfillment of the Requirements
for the Degree of
Doctor of Philosophy
in

Computer Science

June 2007

Hsinchu, Taiwan, Republic of China

中華民國九十六年六月

國　立　交　通　大　學

博碩士論文全文電子檔著作權授權書

（提供授權人裝訂於紙本論文書名頁之次頁用）

本授權書所授權之學位論文，為本人於國立交通大學資訊科學與工程研究所多媒體工程 組，95 學年度第 2 學期取得博士學位之論文。

論文題目：透視不變性於電腦視覺之應用

指導教授：莊仁輝教授

■ 同意

本人茲將本著作，以非專屬、無償授權國立交通大學與台灣聯合大學系統圖書館：基於推動讀者間「資源共享、互惠合作」之理念，與回饋社會與學術研究之目的，國立交通大學及台灣聯合大學系統圖書館得不限地域、時間與次數，以紙本、光碟或數位化等各種方法收錄、重製與利用；於著作權法合理使用範圍內，讀者得進行線上檢索、閱覽、下載或列印。

論文全文上載網路公開之範圍及時間：	
本校及台灣聯合大學系統區域網路	■ 立即公開
校外網際網路	■立即公開

授 權 人：高肇宏

親筆簽名：______________________

中華民國　　　　年　　　　月　　　　日

國立交通大學

博碩士紙本論文著作權授權書

（提供授權人裝訂於全文電子檔授權書之次頁用）

本授權書所授權之學位論文，為本人於國立交通大學資訊科學與工程研究所 多媒體工程 組，95 學年度第 2 學期取得博士學位之論文。

論文題目：透視不變性於電腦視覺之應用
指導教授：莊仁輝教授

■ 同意

本人茲將本著作，以非專屬、無償授權國立交通大學，基於推動讀者間「資源共享、互惠合作」之理念，與回饋社會與學術研究之目的，國立交通大學圖書館得以紙本收錄、重製與利用；於著作權法合理使用範圍內，讀者得進行閱覽或列印。

本論文為本人向經濟部智慧局申請專利(未申請者本條款請不予理會)的附件之一，申請文號為：____________________，請將論文延至____年____月____日再公開。

授 權 人：高肇宏

親筆簽名：______________________

中華民國　　　年　　　月　　　日

國家圖書館博碩士論文電子檔案上網授權書

ID:GT008823804

本授權書所授權之學位論文，為本人於國立交通大學資訊科學與工程研究所 多媒體工程 組，95 學年度第 2 學期取得博士學位之論文。

論文題目：透視不變性於電腦視覺之應用

指導教授：莊仁輝教授

茲同意將授權人擁有著作權之上列論文全文（含摘要），非專屬、無償授權國家圖書館，不限地域、時間與次數，以微縮、光碟或其他各種數位化方式將上列論文重製，並得將數位化之上列論文及論文電子檔以上載網路方式，提供讀者基於個人非營利性質之線上檢索、閱覽、下載或列印。

□ 論文全文上載網路公開之範圍及時間：2007.6.21 公開

※ 讀者基於非營利性質之線上檢索、閱覽、下載或列印上列論文，應依著作權法相關規定辦理。

授權人：高肇宏

親筆簽名：________________

民國　　年　　月　　日

國立交通大學

論文口試委員會審定書

本校　資訊科學與工程研究所　博士班　　高肇宏　　君

所提論文:　透視不變性於電腦視覺之應用

Using Perspective Invariants in Some Computer Vision Applications

合於博士資格水準、業經本委員會評審認可。

口試委員：

____________________　____________________

____________________　____________________

____________________　____________________

____________________　____________________

____________________　____________________

指導教授：

_______________　_______________　_______________

系主任：

中華民國九十六年　月　日

透視不變性於電腦視覺之應用

學生：高肇宏　　　　　　　　　　　　　　　　指導教授：莊仁輝 教授

國立交通大學資訊科學與工程研究所

摘　　要

電腦視覺的主要目標，是發展能以良好效率及精確性完成特定影像分析工作的穩定系統。其中，基於透視投影幾何的方法，常與參考點所提供的已知資訊有所關連。交比不變性在此類投影轉換中，扮演了一個重要的角色。事實上，它是許多辨識和影像重建方法的基礎，同時也是處理電腦視覺問題最重要的技術之一，例如平面特徵的辨認、三度空間的定位，以及在自動導航領域之應用。在本篇論文中，我們應用交比發展了產生線光源影子的方法，以及利用交比進行臉孔辨識的系統。基於此不變性，我們可以避免大量且繁複的有關三維空間資訊的計算，例如攝影機的校正，和物體完整結構的重建，而直接使用交比去求得物體二維或三維結構的相對量測值。我們也發展了一個新的方法，去預測及描述交比計算過程所產生的定位誤差的特性。此方法並不直接進行整個影像空間的誤差計算。由於此方法可提供我們對此類誤差的傳播較具象的概念，因此能幫助我們有效率地選擇參考點。另外，我們也提出了以影像特徵點間的區域相似度和整體的幾何關係為基礎的影像特徵對應計算方法。此演算法可做為影像分析的前處理運算，並可望適用於包括本論文所提出的各種即時電腦視覺應用。

關鍵字：交比，透視投影，不變性，線光源影子，臉孔辨識，定位誤差，特徵點對應

Using Perspective Invariants Cross-Ratio in Computer Vision

Student：Jau-Hong Kao　　　　Advisor：Dr. Jen-Hui Chuang

Department of Computer Science
National Chiao Tung University

ABSTRACT

One of the main purposes of computer vision is to develop a reliable system that can carry out its tasks with satisfactory efficiency and precision in a realistic environment. Approaches based on projective geometry are often associated with reference points given as prior knowledge. As a geometric invariant under projective transformations, cross-ratio is the basis of many recognition and reconstruction algorithms. In fact, cross-ratio-based approaches are important techniques to address various computer vision problems, such as planar feature recognition, 3-D localization and autonomous navigation applications. In this dissertation, applications of cross-ratio in shadow generation and identity verification are investigated. The common idea of these algorithms is to use cross-ratio to determine measurements of object structure without tedious and expensive computation to infer 3-D information, including object modeling and camera calibration. Meanwhile, for error analysis of cross-ratio-based approaches, we derive efficient means to predict and to describe the characteristic of localization error. The approach allows one to select appropriate reference image points by providing corresponding regions of localization error. Finally, an efficient approach for finding correspondences between image features based on local similarity and global constraints is also conducted as an applicable stage of image analysis, which will be suitable for various real-time applications of computer vision, including those developed in this dissertation.

keywords：cross-ratio, perspective projection, invariant, shadow of linear light source, identity verification, localization error, feature point correspondence

Index

List of Figures

1. Introduction

One of the main purposes of computer vision is to develop a reliable system that can carry out its tasks, e.g., reconstruction of scene structures, with satisfactory efficiency and precision in a realistic environment. There are basically two classes of methods to reconstruct 3-D structures from 2-D images. The first class involves strategies relying on camera calibration to establish reconstruction matrices while the second class consists of approaches based on projective geometry associated with reference points given as prior knowledge. An invariant is a property of a geometric configuration that does not change when a transformation is applied to that geometric configuration. As a geometric invariant under projective transformations, cross-ratio is the basis of many recognition and reconstruction algorithms which are based on projective geometry [1][2]. In fact, cross-ratio-based approaches are important techniques to address various computer vision problems, such as planar feature recognition, 3-D localization and autonomous navigation applications.

In general, an immediate advantage to use cross-ratio measurement in computer vision applications is that most calibration works, which aims to obtain 3-D data, are no longer necessary. Since various tasks can thus be processed with simple 2-D computation, such a view-invariant measurement plays a very important role in this dissertation. The proposed shadow generation approach uses cross-ratio to compute 2-D quantities which are traditionally obtained with 3-D scene and projection data. In the proposed face identification approach, it is found that cross-ratio measurements are actually equivalent to relative affine structure described in [3], which is then utilized to derive relative 3-D information of facial features.

On the other hand, as indicated in [4][5], the quality of scene reconstruction and structure inference strongly depends on the quality of the image data. In addition to other possible measurement uncertainties, 2-D coordinates of feature points in an image plane will always have quantization errors due to limited image resolution. Hence, values of projective coordinates, i.e., pairs of cross-ratios with respect to some given reference points, will also be noisy. Since the error in the calculation of cross-ratio will also propagate in the subsequent computations, it must be carefully analyzed and controlled so as to avoid too much negative influence on the final reconstruction results. In this dissertation, we propose an efficient way of analyzing localization error for systems which use cross-ratio for planar localization. We first inspect the linear nature of localization error due to small inaccuracy in cross-ratio measurements. Properties of the localization error due to two dimensional noises of reference image points are then investigated. Based on our computationally efficient error analysis, we derive means to predict and to describe the characteristic of localization error, assisting ones to select reference image points accordingly and to efficiently provide picture of resultant localization error in advance, without generating similar results by using a large amount of synthesized noisy data. In order to efficiently obtain reliable correspondences between image features, we also conduct a method to find point correspondences using local similarity and global constraint. Such an approach can be conducted as a preprocess stage of image analysis. Related works for shadow generation, face identification, error analysis of cross-ratio, and the determination of image feature correspondence will be presented in the following subsections.

1.1 Survey of shadow generation of linear light source

Shadows increase the perception of image realism, and can also enhance users' spatial awareness. Several shading and shadowing algorithms, which are based on 3-D information of the scene, have been proposed for directional lights, point lights, spot lights, and so on [6][7]. There are two general-purpose shadow algorithms for interactive applications. The approaches based on shadow volume are continuous methods working in object space and the ones based on shadow map are discrete methods that sample depth images of the scene [8]. Among them, illumination from linear and area light sources generates penumbras along shadow boundaries which notably enhance the photo-realism of an image. For the illumination due to a linear light source, sampling methods, which represents the light source with a series of point light sources, are often used [9]. This is because a shadowing algorithm for a point light source is simple. However, if the point samples are too sparse, serious aliasing artifacts will occur. On the other hand, for very dense samples, the computation cost will become excessive.

Instead of sampling, another type of shadowing algorithms, e.g., the ones use light clipping process, determines the illumination of a point on the object surface by identifying the portions of each linear light source visible from that point. However, when many complex objects cast shadows onto rugged object surfaces, the cost for the light clipping process is extremely high. To reduce the cost, Bao et al. [7] proposed an extension of the BSP tree-based shadowing algorithm which is originally developed for point light sources [10][11].

In [12], an algorithm which can precisely generate shadows due to a linear light source for complex (curved) objects other than planar polygons is introduced. The algorithm computes the illumination of each point in the penumbra by using an integral function to evaluate the diffuse and specular effects simultaneously. In order to reduce the higher cost due to more complex object shapes, only the objects occupying the subdivisions that intersect the light triangle are considered. In [13], a ray-oriented buffer is proposed to improve the rendering performance by reducing the computation time for the selection of candidate polygons. The authors of [14] proposed a soft shadow algorithm to produce penumbra regions for linear light sources. It is not an exact method and will produce artifacts if the light source is severely undersampled. Other model-based approaches are based on radiosity methods, which need to address the problem of high computation cost [15][16].

On the other hand, geometries between 2-D image features are also used to derive 3-D information in some approaches, such as [17]. In that method, a special setup of the scene and some assumptions speed up the computation of the camera calibration. However, these simplified calibration model are often sensitive to the quality of imaging device and depend on the sophisticate image processing techniques. In addition, there has been some work on generating shadows from image-based scene representations [18].

Previous works such as those in [6]-[19] are generally based on 3-D geometry. Relevant 3-D data required in the shadow computation includes the object model, the location of light sources and the surfaces on which the object shadows are to be cast. In this dissertation, an algorithm which obtains realistic shadowing effects purely using 2-D information is proposed. By using cross-ratios, the system neither utilizes calibrated cameras nor performs 3-D reconstruction of the scene.

1.2 Survey of identity verification using multiple facial images

Machine recognition of faces has been a very active research topic in recent years [20]-[23]. Face recognition technology for still and video images has potentially numerous commercial and law enforcement applications. These applications range from static matching of well-formatted photographs such as passports, credit cards, driver's licenses, and mug shots, to real-time matching of surveillance video images presenting different constraints in terms of various processing requirements. Although humans seem to recognize faces in cluttered scene with relative ease, machine recognition which often spans several disciplines such as image processing, pattern recognition, computer vision, and neural networks is a much more daunting task. In particular, the problem can be formulated as follows: Given still or video images of a scene, identify one or more persons in the scene using a stored database of faces. A complete face recognition system generally includes two main stages. The first stage is the face detection stage that determines the existence of one or more faces in an image. Techniques used in this stage involve segmentation of faces from cluttered scenes and extraction of features from the face region. The challenges are mainly due to the fact that the position, orientation and size of face regions in an arbitrary image are usually unknown [24][25][26]. A survey of face detection techniques can be found in [27]. The second stage is the recognition stage which deals with the identification and matching problems. The goal is to determine the identities of the target faces obtained in the first stage. Considering important works developed so far in the recognition stage in the engineering literature, a brief survey on the face recognition researches in recent years is provided in what follows.

Most of existing face recognition algorithms are 2-D based. In terms of the nature of the facial features utilized, these 2-D algorithms can generally be divided into two major categories: structure-based approaches and statistics-based approaches. The class of structure-based ones uses structural facial features, which are mostly local structures, e.g., the shapes of mouth, nose, and eyes [28]-[31]. In [30], an automated recognition system that uses a top-down control strategy directed by a generic model of expected feature characteristics is developed. They proposed an elastic graph matching model which extracts the feature vectors from image lattices based on a set of 2-D Gabor filters. The main advantage of a structure-based face recognition method is the low sensibility to irrelevant data, e.g., moving hair or background, since it only handles data of interest instead of using all image data indiscriminately. The main disadvantage of such approaches is the high complexity in feature extraction.

The statistics-based approaches basically use the whole 2-D image as facial features [20] [32][33][34]. In this category of approaches, the principal component analysis (PCA) exhibits particular importance [35]. The principal components, e.g., Eigenface [36][37], of training face images are calculated and then used as a set of orthonormal basis. The complete space can be represented effectively by a significant small subset of these orthonormal facial images and the dimension of the feature space of facial images is thus reduced. Moreover, theoretical neuroscience has contributed to account for the view-invariance perception, which is also the underlying idea of our work for identify verification, of universals such as the explicit perception of featural parts and wholes in visual scenes. A survey of recent developments in theoretical neuroscience for machine vision can be found in [38]. These unsupervised learning methods are used to make predictive perceptual models of the spatial and temporal statistical structure in natural visual scenes. In particular, given the spatial-temporal continuity of the statistics of sensory input, invariant object recognition might be implemented using a learning rule that uses a trace of previous neural activity capturing the same object under different transforms in the short time scale. By first relating a modified Hebbian rule to error correction rules and exploring a number of error correction rules that can be applied to invariant pattern recognition, learning rules related to temporal difference learning are developed in [39]. The analysis of temporal difference learning provides a theoretical framework for better understanding the operation and convergence properties of rules useful for learning invariant representations. In contrast to structure-based approaches, statistics-based ones are more straightforward and simple. However, it happens that important local features are used with small factor of importance. As for theoretical neuroscience, it is not yet obvious whether the full power of learning rules is expressed in the brain, and the practical applications in face recognition are needed for the understanding of the performance. The work in [39] provides suggestions about how they might be implemented. Although the above 2-D based face recognition approaches produce satisfactory results under normal conditions, their performance can deteriorate quickly by varying lighting condition or large change of the viewing geometry.

As the face recognition technology is an essential tool for law enforcement agencies' efforts to combat crime, fake or duplicated facial images which can easily cheat the 2-D based facial recognition systems raise problems of interest [21]. To avoid such problems, a few 3-D model-based face recognition are proposed wherein 3-D feature points are reconstructed which provide important information for facial recognition. In [40] a method based on Karhonen-Loeve expansion is developed to reconstruct 3-D face features. The method is claimed to be independent on lighting conditions. In [41], the reconstruction of face surface is made rotation-invariant. A similar approach based on a depth map obtained from stereo images to perform face segmentation and recognition can be found in [42]. In [43], a model-matching approach is provided to reduce the computational cost of 3D-based facial recognition algorithms.

In this dissertation, we propose a novel approach to identify a person with facial images using 3-D information of facial feature points. Three reference points are first extracted to construct a reference plane in every image. By calculating a view-invariant relative depth with respect to the obtained reference plane for each relevant feature point, an efficient face recognition algorithm is developed using relative affine structure introduced in [3], which are found in our work to be equivalent to some cross-ratio measurement.

1.3 Survey of error analysis of cross-ratio-based planar localization

Recently, more and more computer vision researchers are paying attention to error analysis so as to fulfill various accuracy requirements arising from different applications. One of the main purposes of computer vision is to develop a reliable system that can carry out its tasks, e.g., reconstruction of scene structures, with satisfactory efficiency and precision in a realistic environment. There are basically two classes of methods to reconstruct 3-D structures from 2-D images. The first class involves strategies relying on camera calibration to establish reconstruction matrices while the second class consists of approaches based on projective geometry associated with reference points given as prior knowledge.

As a geometric invariant under projective transformations, cross-ratio is the basis of many recognition and reconstruction algorithms which are based on projective geometry [1][2]. For example, cross-ratios calculated from vertices of polygons are used in [44]-[48] to recognize planar features in a 3-D environment. In addition to recognition, given prior knowledge about a scene, object structure can also be reconstructed using cross-ratio. For example, an approach that transforms relative affine structure defined in [3] into equivalent cross-ratio measurement is used to determine relative 3-D face structure from facial images in an identity recognition system [49]. Such a projective invariant can also be utilized to match trajectories across video streams and applied to image retrieval problems [50][51]. For autonomous navigation of vehicles, cross-ratio is often used to identify artificial landmarks or beacons placed in the environment [52]-[56].

As indicated in [4][5], the quality of scene reconstruction and structure inference strongly depends on the quality of the image data. In addition to other possible measurement uncertainties, 2-D coordinates of feature points in an image plane will always have quantization errors due to limited image resolution. Hence, values of projective coordinates, i.e., pairs of cross-ratios with respect to some given reference points, will also be noisy. Since the error in the calculation of cross-ratio will also propagate in the subsequent computations, it must be carefully analyzed and controlled so as to avoid too much negative influence in the final reconstruction results. Some studies of cross-ratio are proposed to assess its use in invariant-based recognition systems [4][5][57][58][59]. These studies mainly focus on robust estimations of the cross-ratio regarding to the uncertainty in measurement, or concerning with the analysis of error propagation, so as to provide relevant information of quality estimation in different steps of a vision system. However, such sensitivity analysis only considers the computation of cross-ratio itself instead of the final localization or reconstruction results or autonomous navigation applications.

In this dissertation, we propose an efficient way of analyzing localization error for systems which use cross-ratio for planar localization. Based on such a computationally efficient error analysis, one may obtain the picture of resultant regions of localization error in advance, instead of generating similar results by using a large amount of synthesized noisy data for a particular cross-ratio configuration, and select proper reference image points accordingly.

1.4 Survey of finding point correspondence using local similarity and global constraint

Extraction and correspondence establishment for image features have been intense areas of research in computer vision for decades. Feature extraction analyzes images and obtains meaningful image features, e.g., corners, which abstract the scene structure to reduce the amount of data for further computation. Establishing correspondences of image features, which is also a fundamental topic in vision, forms the basis for stereo depth computation as well as most optical flow algorithms. Given two images of the same scene, a pixel in one image corresponds to another pixel in the other image if both pixels are projections along lines of sight of the same physical element in space. Regarding the state of the art in related researches, it seems that no general solution to the correspondence problem exists, due to ambiguous matches. Some common causes include occlusion, lack of texture, and photometric distortions, etc.

A number of constraints for feature extraction (e.g., corner points should be well localized as stated in [60]), correspondence determination (e.g., via epipolar geometry [61]), and some other assumptions (e.g., image brightness constancy and surface smoothness), are exploited to make the problem tractable. Hager et al. refer to constraints on a small region surrounding the pixel of interest as *local* constraints, and constraints on scan-lines or on the entire image as *global* constraints [62]. They outline the principle methods for exploiting both local and global constraints. In general, local methods can be very efficient. However, they are sensitive to locally ambiguous regions in an image such as uniform textures and occlusion regions. On the other hand, global methods are usually more robust by providing additional support for regions difficult to match locally. But the computation is generally more expensive. Recently, SIFT is proposed and used to describe and match digital image content between views [63]. However, while the purpose is to compute features invariant to transformations, the SIFT description is typically too expensive especially when the transformation among images is not significant, e.g., between two consecutive frames of a video sequence.

In this dissertation we propose a novel approach to feature extraction and correspondence establishment for images of indoor scenes. Since image transforms include mainly pan and tilt in common reconstruction scenarios, we assume the scaling and roll are insignificant.

1.5 Organization of the dissertation

In this dissertation, various applications of cross-ratio are investigated. The common idea of these algorithms is to use cross-ratio to determine measurements of object structure without tedious and expensive computation to infer 3-D information, including object modeling and camera calibration. Meanwhile, for error analysis of cross-ratio, we derive efficient means to predict and to describe the characteristic of localization error. The approach allows one to select reference image points efficiently and to obtain picture of resultant localization error in advance, instead of generating similar results by using a large amount of synthesized noisy data. Finally, the approach for correspondence establishment can be considered as a pre-process for proposed cross-ratio-based applications when the scaling and roll are insignificant. The organization of this dissertation is shown in Figure 1.1.

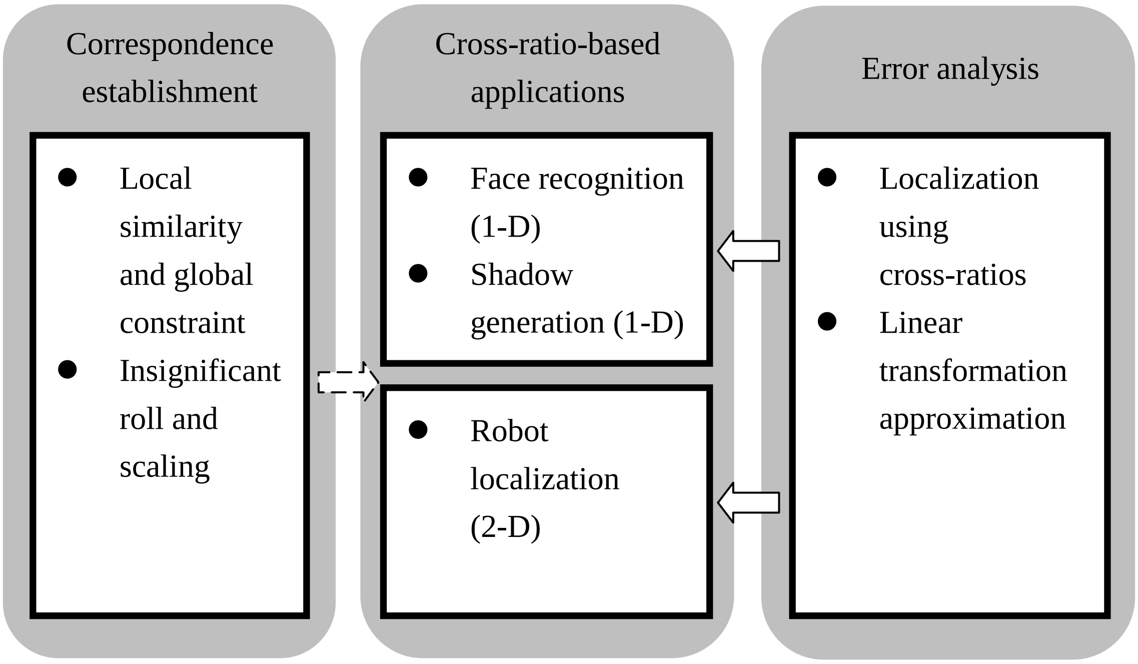

Correspondence establishment
Local similarity and global constraint
Insignificant roll and scaling
Cross-ratio-based applications
Face recognition (1-D)
Shadow generation (1-D)
Robot localization (2-D)
Error analysis
Localization using cross-ratios
Linear transformation approximation

Figure 1.1 Organization of the dissertation.

The remainder of this dissertation is organized as follows. In Chapter 2, an efficient approach to generating object shadows on a base plane due to a linear light source in 3-D space is proposed. With the use of cross-ratio, the computation is performed purely in 2-D image space, thus it needs neither calibrated cameras nor 3-D reconstruction of the scene. Chapter 3 describes an identity verification system based on cross-ratio measurement computed from multiple facial images. The measurement interprets 3-D structure of a face as a set of relative quantities. Using the property of invariant, the system is able to perform identity verification without camera calibration. In addition, iterative training is not required which leads to the issue of convergence in the neural network based face recognition approaches. In Chapter 4, an efficient way of analyzing localization error for systems using cross-ratio for planar localization is proposed. Starting from the 1*st*-order approximation of one dimensional error functions, we eventually investigate conditions for the existence of a nominal boundary of an ellipse due to some circularly distributed 2-D errors. In Chapter 5, we develop an algorithm to find correspondences between images. Under the assumption of insignificant scaling and roll, the approach uses local similarity and global constraint to match image feature points. The system is efficient and suitable to real-time applications. Finally, Chapter 6 gives conclusions of this dissertation.

2. Shadow Generation of Linear Light Source Using Stereo Images without 3D Reconstruction

2.1 Overview

Shadows contribute to the perception of image realism, and can also enhance users' spatial awareness. Several shading and shadowing algorithms based on 3-D information of the scene have been proposed [6]-[19]. However, they are generally based on 3-D geometry. In this chapter, an algorithm which obtains realistic shadowing effects purely using 2-D information is proposed. The system neither utilizes calibrated cameras nor performs any 3-D reconstruction of the scene. Similar to the sampling methods, we use a set of point samples to approximate a linear light source. In the next two sections, we briefly review the definition of the projective invariant cross-ratio and its use in planar point localization, respectively, which plays a fundamental role in algorithms described in following chapters. In Sections 2.4, the approach developed in [64] to generating shadow due to a point light source is briefly reviewed. The approach is then extended to the generation of shadows due to a linear light source in Section 2.5. Experimental results and a brief summary are given in Sections 2.6 and 2.7, respectively.

2.2 Definition of cross-ratio

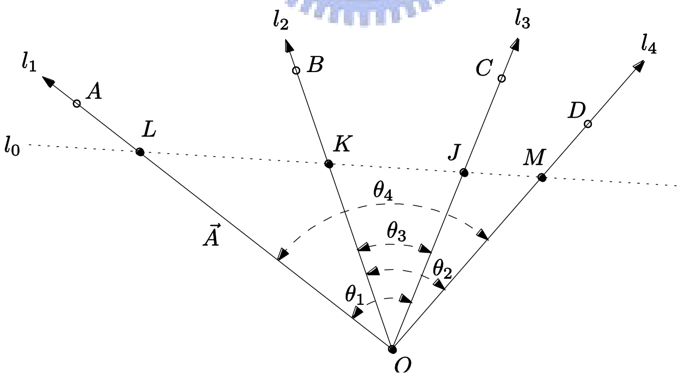


Figure 2.1 Cross-ratio of five coplanar points.

Before the description of a typical framework for 3-D reconstruction of a scene point from four reference points, all in a 2-D space, using the projective invariant cross-ratio, we first review some mathematics involved in the computation. By denoting image points with lowercase letters and scene points with uppercase letters, let *O*, *A*, *B*, *C* and *D* be five coplanar points in a general configuration (with no three of them being collinear), as shown in Figure 2.1. One form of cross-ratio is given as[1]

$$[A,B,C,D]_O \triangleq CR_O \triangleq \frac{\sin\theta_1 \sin\theta_2}{\sin\theta_3 \sin\theta_4} = \frac{\left|\vec{A}\times\vec{C}\right|\left|\vec{B}\times\vec{D}\right|}{\left|\vec{B}\times\vec{C}\right|\left|\vec{A}\times\vec{D}\right|} \tag{1}$$

where $\vec{A}=(A_x, A_y)$ stands for the directed vector $\overrightarrow{OA}$, and so on. Alternately, we can rewrite (1) as

$$[A,B,C,D]_O = \frac{\begin{vmatrix} A_x & C_x \\ A_y & C_y \end{vmatrix}\begin{vmatrix} B_x & D_x \\ B_y & D_y \end{vmatrix}}{\begin{vmatrix} B_x & C_x \\ B_y & C_y \end{vmatrix}\begin{vmatrix} A_x & D_x \\ A_y & D_y \end{vmatrix}} \triangleq \frac{Q_1}{Q_2} \tag{2}$$

where A_x, ..., D_x denote the x-components of vectors $\vec{A}, ..., \vec{D}$, respectively, and Q_1 and Q_2 are the denominator and numerator of CR_O, respectively. With (2), a cross-ratio can be obtained without actually computing the angles between the pencils $\overrightarrow{OA}, \overrightarrow{OB}, \overrightarrow{OC}, \overrightarrow{OD}$.

2.3 Use of cross-ratio in Planar Point Localization

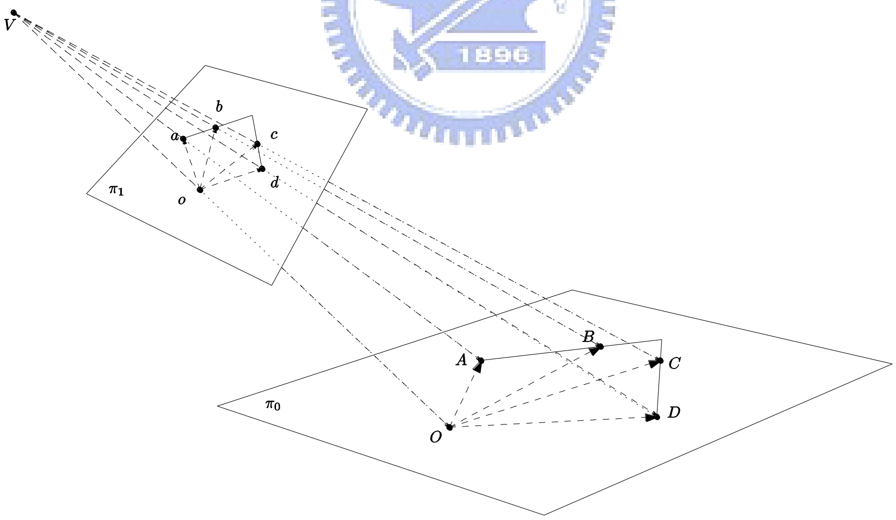


Figure 2.2 A perspective projection of some feature points.

Figure 2.2 illustrates a typical geometry of perspective projection. Five scene points O, A, B, C, D located on a 3-D plane π_0, with no three of them being collinear, are projected on image plane

[1] Note that a total of 24 different cross-ratios k_i, $1 \le i \le 24$, can be defined for a scene point and Eq. (1) corresponds to k_1 defined in [57].

π_1 as *o*, *a*, *b*, *c* and *d*, respectively[2]. The invariant property of cross-ratio guarantees that if the five feature points can be identified in the image plane accurately, the cross-ratio $[a,b,c,d]_o$ obtained by (2) will be identical to the cross-ratio $[A,B,C,D]_O$.

An immediate application of projective invariant is to determine ray directions. For example, if the origin and three of the rest four points are known in Figure 2.1, the vector passing through the fourth point from the origin can be determined easily from *o*, *a*, *b*, *c* and *d* if $[A,B,C,D]_O$ is given. For example, let $D=(X, Y)$ and $\overrightarrow{OD}$ is to be determined. From (1), we have

$$CR_O = \frac{\begin{vmatrix} a_x & c_x \\ a_y & c_y \end{vmatrix}\begin{vmatrix} b_x & X-O_x \\ b_y & Y-O_y \end{vmatrix}}{\begin{vmatrix} b_x & c_x \\ b_y & c_y \end{vmatrix}\begin{vmatrix} a_x & X-O_x \\ a_y & Y-O_y \end{vmatrix}} \triangleq \frac{Q_1}{Q_2},$$

which can be rewritten as

$$\begin{aligned}&(Q_1K_{BC}A_y - Q_2K_{AC}B_y)X + (Q_2K_{AC}B_x - Q_1K_{BC}A_x)Y \\ &= Q_2K_{AC}\begin{vmatrix} B_x & O_x \\ B_y & O_y \end{vmatrix} - Q_1K_{BC}\begin{vmatrix} A_x & O_x \\ A_y & O_y \end{vmatrix}\end{aligned} \tag{3}$$

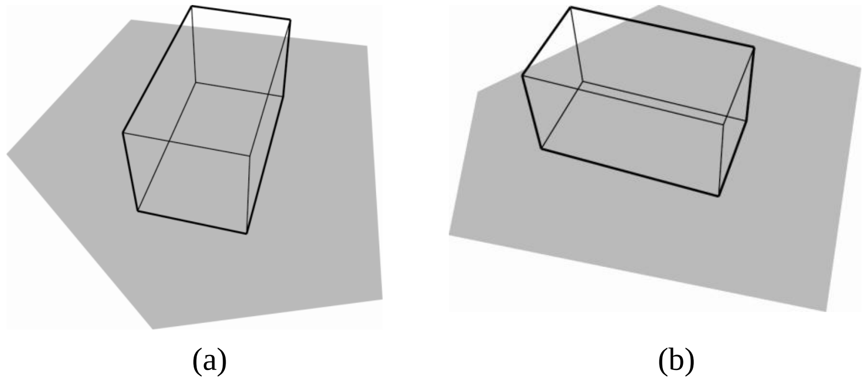
where $K_{AC} = \begin{vmatrix} A_x & C_x \\ A_y & C_y \end{vmatrix}$ and $K_{BC} = \begin{vmatrix} B_x & C_x \\ B_y & C_y \end{vmatrix}$. This is in fact the line equation of l_4 ($\overleftrightarrow{OD}$).

Furthermore, if *O*, *A*, *B*, *C* are known, so as $[A, B, C, D]_O$ and $[B, C, D, O]_A$, we can obtain point *D* by intersecting $\overleftrightarrow{OD}$ and $\overleftrightarrow{AD}$. Accordingly, a localization system can be developed for a mobile robot based on the view-invariant cross-ratio, assuming perfect image acquisition and feature extraction, as described in Chapter 4.

2.4 Shadows of Point Light Sources

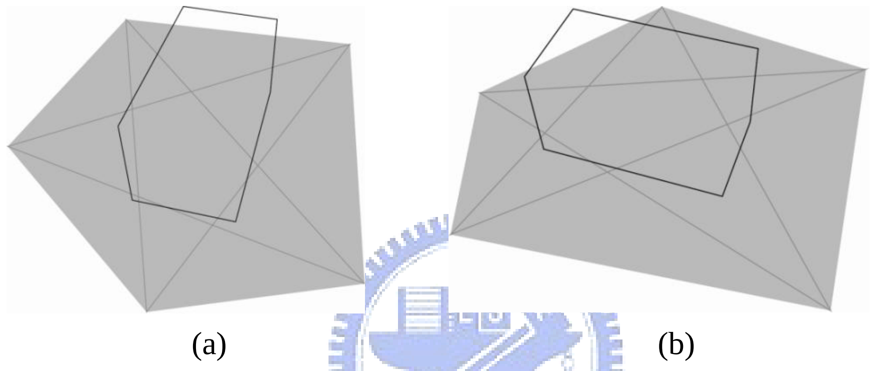
In this section, we briefly review the approach developed in [64] for shadow generation of due to a point light source using 2D image data. The 2D data are extracted, possibly interactively, from pictures taken by some unknown cameras. Given two images *I* and *II*, the approach assumes that the object region, and the base plane where the object's shadow is to be cast upon, can be identified in both *I* and *II*. Moreover, at least five reference points on the base plane can be identified and any three of them must not be collinear in the images. The 3D information of the base plane, e.g., the position in the scene, is not required in the process.

[2] In a robot navigation environment, π_0 can be the ground plane and the five points can be landmarks or beacons placed in the environment, or the robot itself.



(a) (b)

Figure 2.3 (a) A wireframe image (I) rendered at viewpoint V_1. (b) A wireframe image (II) rendered at viewpoint V_2.



(a) (b)

Figure 2.4 (a) The shadow region of the base plane due to a point light source located at V_1. (b) The shadow region of the base plane due to a point light source located at V_2.

Figure 2.3(a) and Figure 2.3(b) show images *I* and *II* obtained from viewpoints V_1 and V_2, respectively, where an object is put on a base pentagon (whose vertices will be treated as reference points). The border segments shown by bold lines in Figure 2.3(a) (Figure 2.3(b)) is the shadow area on the base plane due to a point light source located at V_1 (V_2). In general, if some reference points in image *I* can be located correctly in image *II*, such as intersections of extended lines of the border segments of the shadow region and the diagonals of the pentagon, the shadow region in image *II* can be found, as shown in Figure 2.4(a) and Figure 2.4(b).

Figure 2.5(a) illustrates an object and its shadow thus obtained. A similar shadow generation result is shown in Figure 2.5(b). The detail of the process which accurately locates the intersections and the shadow regions using view-invariant quantities, i.e., cross-ratios, is provided as follows.

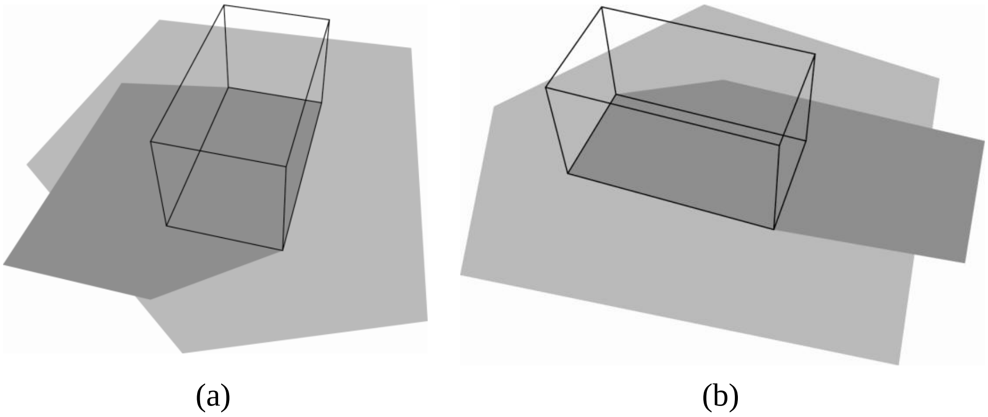


Figure 2.5 (a) An object and its shadow due to a point light source located at V_2. (b) An object and its shadow due to a point light source located at V_1.

Consider a sequence of four points A, B, C, and D located along a straight line in the 3D space. The cross-ratio, in another form, is defined as

$$[A,B,C,D] \triangleq \frac{AC \cdot BD}{BC \cdot AD}. \tag{4}$$

It has been proved in [19] that any linear transformation in homogeneous coordinates, i.e., perspective projection, linear scaling, skewing, etc., preserves the above cross-ratio value. For example, if the image of the four points a, b, c and d can be located accurately along the corresponding image of the line, then we have

$$[A,B,C,D] = [a,b,c,d]. \tag{5}$$

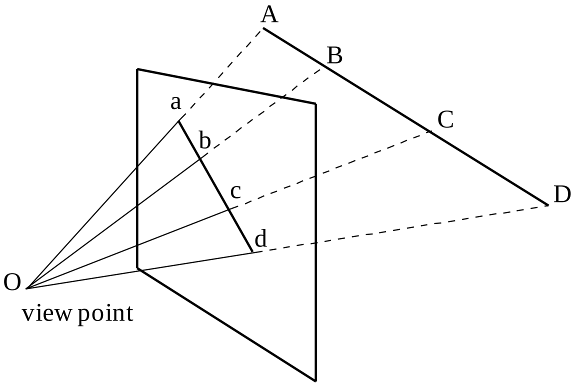


Figure 2.6 A perspective view of four points lying along a straight line.

A straightforward application of the view invariant cross-ratio, is the determination of the location of a point along a line segment in an image. Given the cross-ratio of four points on a line segment, if three of them can be identified accurately in an image, the location of the fourth in the image can be determined easily by solving a linear equation of a single variable. Consider the line segment and its image shown in Figure 2.6. Assume that

$$\frac{AC \cdot BD}{BC \cdot AD} = \frac{ac \cdot bd}{bc \cdot ad} = R\,.$$

If points $a = (a_x, a_y)$, $b = (b_x, b_y)$ and $d = (d_x, d_y)$ can be identified in the image, then we have

$$\frac{(a_x - c_x) \cdot (b_x - d_x)}{(b_x - c_x) \cdot (a_x - d_x)} = \frac{(a_y - c_y) \cdot (b_y - d_y)}{(b_y - c_y) \cdot (a_y - d_y)} = R\,.$$

Hence the location of the fourth point $c = (c_x, c_y)$ can be determined by solving

$$c_x = \frac{Rb_x(a_x - d_x) - a_x(b_x - d_x)}{R(a_x - d_x) - b_x + d_x}$$

and

$$c_y = \frac{Rb_y(a_y - d_y) - a_y(b_y - d_y)}{R(a_y - d_y) - b_y + d_y}\,.$$

For a line segment almost parallel to x-axis (or y-axis) of the image plane, only one of the two equations, which will not result in a near zero denominator, should be used to ensure numerical stability (see [57]). It is also worth noting that if points B and C are too close to each other, the calculation of the cross-ratio in equation (4) may become unstable.

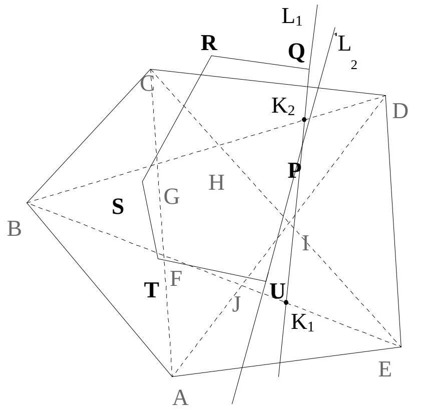


Figure 2.7 The object shadow generated for a point light source located at V_1.

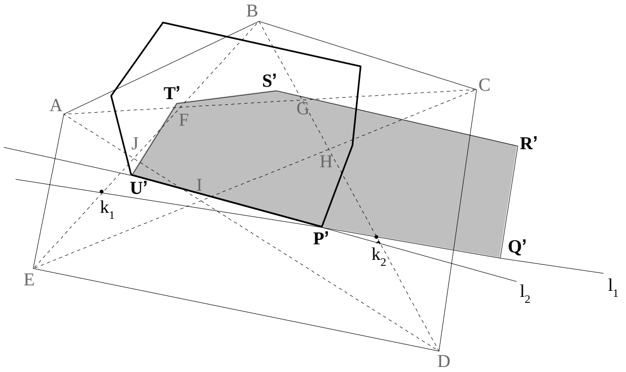


Figure 2.8 A perspective view of four points lying along a straight line.

The above cross-ratio properties are used in [64] to derive object shadows for stereo images. For example, the blocked base region with solid border segments shown in Figure 2.4(a) is identified in Figure 2.5(b) as the shadow region. The two figures are redrawn in Figure 2.7 and Figure 2.8, respectively. In Figure 2.7 and Figure 2.8, if the five reference point can be located precisely, one can identify four reference points along each diagonal of the pentagon, e.g., *B*, *F*, *J* and *E* along $\overleftrightarrow{BE}$. In Figure 2.7, in order to specify the extended line of a border segment of the blocked base region, say L_1, at least two of its intersections with the extended lines of the diagonals of the pentagon need to be found. (Here, we choose $\overleftrightarrow{BD}$ and $\overleftrightarrow{BE}$ which intersect with L_1 with angles closest to 90 degrees in Figure 2.7 so that the intersections can be found more accurately.)

Along $\overleftrightarrow{BE}$ ($\overleftrightarrow{BD}$), its intersection with L_1, K_1 (K_2), and three of the four reference points on the diagonal are used to calculate the cross-ratio value which is then used to locate the intersection k_1 (k_2) in Figure 2.8. Thus l_1 can be determined by k_1 and k_2. Note that to improve the numerical stability in the calculation, the selection of the three reference points is based upon the requirement that none of the quantities in the denominator of equation (4) has an extremely small value. Working in exactly the same manner, l_2, as well as other lines containing the border segments can also be determined in Figure 2.8. With these lines, points *P'*, *Q'*, *R'*, *S'*, *T'* and *U'*, which are the vertices of the shadow region shown in Figure 2.8 are computed.

Assume Image *I* is taken from the light source viewpoint and Image *II* is taken from the eye viewpoint, and let L_i, $(1 \le i \le n)$, denote the lines of the border segments of the base plane region blocked by the object in Image *I*. The algorithm of the above shadow generating process for Image *II* is summarized as *Shadow_Point_Source*. Figure 2.5(a) (Figure 2.5(b)) shows the result for a point light source located at the viewpoint from which Figure 2.4 (b) (Figure 2.4(a)) is obtained.

ALGORITHM 1 *Shadow_Point_Source*

```
- For each $L_i$ ($1 \le i \le n$) {
    - Select two of the extended lines of the diagonals of the pentagon $D_{i1}$, $D_{i2}$ which intersect $L_i$ with angles closest to 90°, identify the intersections $K_{i1}$, $K_{i2}$.
    - For each $D_{ij}$ ($j = 1, 2$) {
        - Select one of the two intersections of $D_{ij}$ with two other diagonals, $X_{ij}$, which guarantees that the following cross-ratio calculation will be numerically stable.
        - Calculate the cross-ratio $R$ of the four reference points (two endpoints of the diagonal, $X_{ij}$ , and $K_{ij}$).
      }
    - Identify the positions of $K_{i1}$ and $K_{i2}$ in image $II$, i.e., $k_{i1}$ and $k_{i2}$.
    - Derive the line $l_i$, which is the image of $L_i$, defined by $k_{i1}$ and $k_{i2}$.
  }
- Determine the vertices of the shadow border by calculating the intersections of pairs of $l_i$'s containing two adjacent border segments.
```

2.5 Shadows of Linear Light Sources

A linear light source casts soft shadows which include umbra and penumbra. In this section, a process of generating shadows due to a linear light source based on the algorithm for a point light source is described. The proposed shadow generation approach is very efficient since the computations use only 2D image data.

Given stereo images of an object obtained from two viewpoints, the proposed approach can generate in either image the shadow due to the linear light source connecting the two viewpoints. To simplify the shadow generation process, the umbra and penumbra regions are not explicitly identified by the proposed approach. Instead, the shadow is formed by superposing shadow regions due to a series of point light sources sampled along the linear light source. The first and the last shadow regions of the two ends of the linear light source are generated by the approach presented in Section 2.4. In order to derive the remaining shadow regions, the proposed approach first identifies trajectories of shadows of object features such as vertices and edges in an image as a point light source moves along the line connecting the two viewpoints, as discussed in Section 2.5.1. The shadow algorithm which uses these feature trajectories to derive the object shadow due to a linear light source is provided in Section 2.5.2.

2.5.1 Trajectories of shadows of object features due to point light sources moving along a straight line

In this subsection, the point light source algorithm described in the previous section is extended to locate trajectories of shadows of object features due to point light sources moving along a straight line in an image. Object shadows due to a linear light source can then be obtained from these object features. The object image obtained from viewpoint V_1 which is shown in Figure 2.4(a) is redrawn here in Figure 2.9 to explain this trajectory locating process for object features.

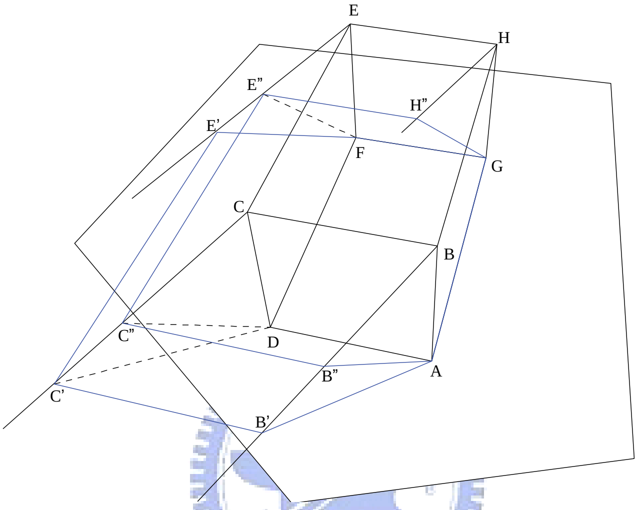


Figure 2.9 Edge and vertex features of an object in image space.

Assume that X, $X \in \{A, B, C, D, E, F, G, H\}$, is the image of an object vertex obtained at viewpoint V_1, and X', $X' \in \{A', B', C', D', E', F', G', H'\}$, is the shadow of X due to a point light source located at viewpoint V_2. The correspondence between Xs and X's can be established by the framework developed in [47] wherein the correspondence between base vertices are determined first. It is easy to see that in image space the shadows of object features due to a series of point light sources lying on the extended line of $\overline{V_1V_2}$, have the following two geometric properties:

(i) For a vertex X of an object, its shadows lie on the straight line $\overleftrightarrow{XX'}$. (For example, E, E' and E'' are collinear as shown in Figure 2.9.)

(ii) The extended lines of the shadows of each object edge are either intersected at the same point or parallel to each other. (Note that the parallel case is only associated with special 3D configurations of an object edge.)

For each X in Figure 2.9, the straight line described in (i) and the intersection point of the extended lines in (ii) can be obtained easily by identifying X' with *Shodow_Point_Source*. Therefore, given a vertex (or/and an edge) of an object shadow due to a point light source located on $\overline{V_1V_2}$, one can derive the remaining vertices and edges of the shadow by intersecting corresponding linear trajectories of object vertices described in (i) and the extended lines of object edges given in (ii). An algorithm to derive shadow due to a linear light source using these features will be presented next.

2.5.2 The Generation of an Object Shadow due to a linear light source

In this subsection, a shadow generation algorithm which uses shadows of object features discussed in the previous subsection to generate object shadows due to a linear light source is given. Consider $B'' \in \{\overline{BB'}\}$ in Figure 2.9 which is the shadow of object vertex *B* due to a point light source lying on $\overleftrightarrow{V_1V_2}$. According to (ii), one can easily find the straight line $\overline{B''C''}$ passing through *B''* and parallel to $\overline{BC}$ and $\overline{B'C'}$. It intersects $\overline{CC'}$ at *C''*, which is the shadow of *C*. Similarly, for each *X*, $X \in \{A, B, C, D, E, F, G, H\}$, one can obtain *X''*, where $X'' \in \{A'', B'', C'', D'', E'', F'', G'', H''\}$. (Note that in this example *A''*, *D''*, *F''* and *G''* coincide with *A*, *D*, *F* and *G*, respectively, since they are actually the base points.) With this shadow boundary identifying process, object shadows due to a series of point light sources located along a straight line can also be generated in an image.

In particular, given two images *A* and *B* obtained from two different viewpoints V_1 and V_2, respectively, if all object vertices can be seen in both images, then the object shadow corresponding to a point light source on $\overleftrightarrow{V_1V_2}$ can be obtained if the shadow of an arbitrary, non-base object vertex is given. Assume X_i, $1 \le i \le n$, are object vertices viewed from V_1, and X_i', $1 \le i \le n$, are shadows of these vertices computed with *Shadow_Point_Source* for a point light source located at V_2. Given the shadow X_s'' of an arbitrary non-base object vertex X_s due to a point light source on $\overleftrightarrow{V_1V_2}$, the process to locate X_i'', $1 \le i \le n$, and thus to identify the shadow region of the object in image *A*, can be formulated as follows.

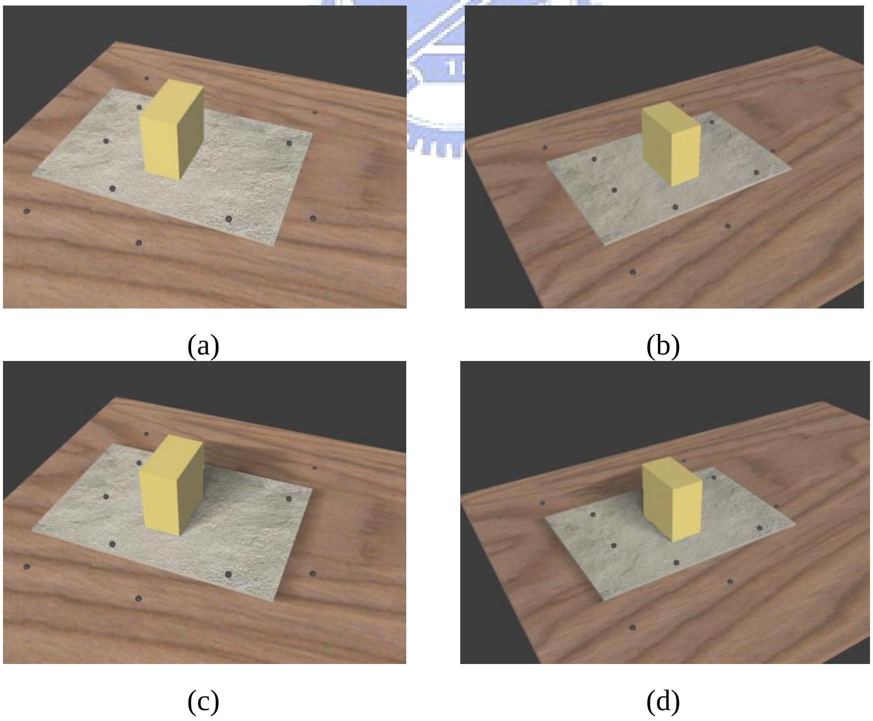
ALGORITHM 2 ***Shadow_Point_Source_ on_ Line***

```
(1) Find all base points which satisfy $X_i = X_i'$.
(2) For each non-base vertex $X_j$ adjacent to $X_s$, determine $\overline{X_S''X_J''}$.
(3) Calculate $X_j''$ which is the intersection of $\overline{X_JX_J'}$ and $\overline{X_S''X_J''}$.
(4) If there are unprocessed non-base vertices $X_i$,
        (i)  $X_s'' \leftarrow X_j''$.
        (ii) Go to Step 2.
(5) Connect the vertices $X_i''$, $1 \le i \le n$, into a shadow region.
```

Two images are sufficient to generate the shadow if all of the projections of the utilized vertices on object boundary can be identified in both of them. In general, it is not always true that all the needed object vertices can be identified in two images. In this case, we will need additional image(s) to ensure that these vertices can be seen from at least two view points. In fact, we only need to consider vertices which will become vertices of the shadow boundary. For instance, *B* in Figure 2.9 needs correspondence and *F* does not need however. On the other hand, if an object vertex has been identified as a base point, possibly interactively, only one image is sufficient for that vertex. The developed approach can then be used to obtain shadows of the object with minor modifications, i.e., by separating vertices into groups that utilize different sets of images and generating all the needed trajectories of object features.

In order to derive soft shadows, an intensity buffer is used to calculate the illuminations of all image points. Depending on the density of the virtual point light sources sampled between the two view points, there is a trade-off between the quality of the synthesized image and the computation time.

2.6 Experimental Results



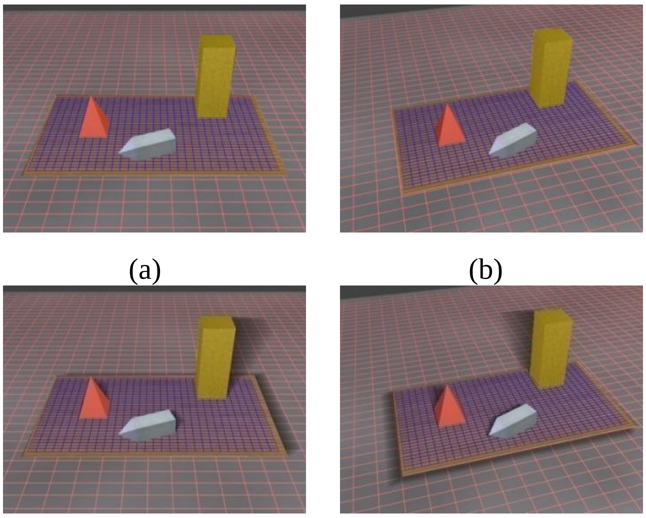
(a) (b)

(c) (d)

Figure 2.10 (a) and (b): Two images of an object placed on a base plane. Reference points on both planes are marked with black color. (c) and (d): Shadow generation results of (a) and (b), respectively.

This section illustrates experimental results for both synthetic and real images. In complicated scenes, shadows of an object are often projected onto multiple planes. The algorithm developed in previous sections can deal with such scenes if at least five reference points are identifiable in two or more images for each projection plane. In general, shadow regions may go beyond the boundary of a projection plane. In that case, the boundary of the projection plane needs to be specified in the image. One may need to clip certain portions of the derived shadow regions which are outside a projection plane with a finite size or blocked by other polygon objects (or projection planes). Although in our implementation, the shapes of projection planes and the geometric relationships among them, e.g., the visibility with respect to different viewpoints, are established manually in advance, the proposed approach can be enhanced by importing a fraction of 3D information or by applying other algorithms such as the BSP tree. Figure 2.10 illustrates shadow generation examples of scenes with multiple projection planes. A box is placed on the base plane hung in the air. Shadows are cast upon the base plane and the ground plane underneath the base plane, with five reference points also shown for both planes. Note that shadows of the base plane are also cast upon the ground plane.

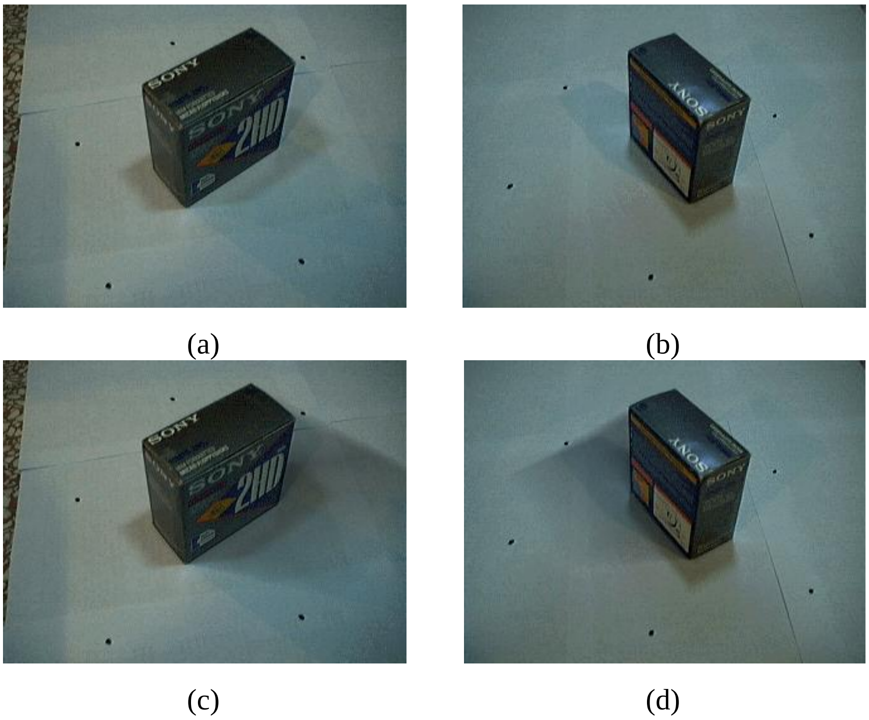
The developed algorithms are implemented with Microsoft Visual Basic 6.0 running on a Pentium II PC. All images are rendered in 32-bit RGBA color mode and the sizes are 800 by 600 pixels. The computation time for shadow region generation is much less than that for updating the intensity buffers. For example, Figure 2.10 and Figure 2.11 take 0.48, and 0.81 seconds for shadow region generation, and 1.07, 2.16 seconds for updating the intensity buffers, respectively. The time spends for updating the intensity buffers is roughly proportional to the amount of accesses of the intensity buffers. Though our implementation does not use special instruction for fast memory access provided by modern CPU, we believe the use of this support of hardware can extremely improve the overall performance of the proposed approach.



(a) (b)

(c) (d)

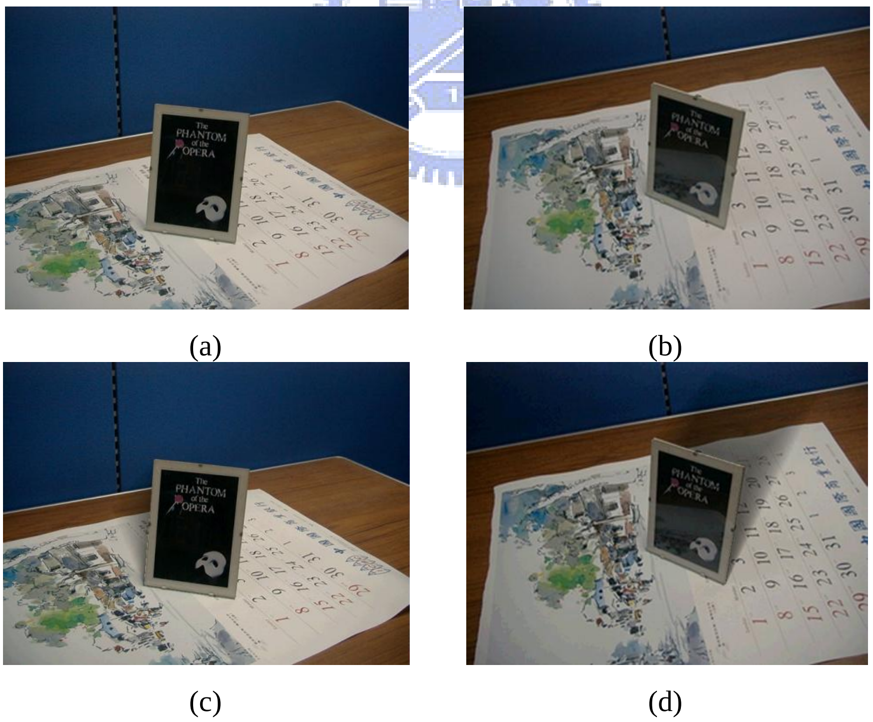
Figure 2.11 Shodows generated for multiple objects using 20 virtual point light sources.



(a) (b)

(c) (d)

Figure 2.12 A real image example. (a) and (b): Source images. (c) and (d): Shadows generated with 30 virtual point light sources are blended.



(a) (b)

(c) (d)

Figure 2.13 Another real image example. (a) and (b): source images. (c) and (d): Shadows generated with 30 virtual point light sources are blended.

Figure 2.12 and Figure 2.13 demonstrate shadow generation examples of real scenes. In real scenes, shadows may already exist in the original images. It can be seen that additional shadows added by the proposed algorithm blend naturally into the images. For the examples considered in this section, the time spent in calculating a shadow due to a point light source for the real scenes is in fact less than that for synthetic ones since the geometry of the latter is more complex.

2.7 Summary

In this chapter, we propose a novel approach to generating object shadows due to a linear light source. The purpose is to obtain realistic shadowing effects with very limited data and simple calculations. The shadows are derived without 3D reconstruction of the scene. A shadow generation algorithm which uses trajectories of shadows of object features on base plane due to a linear light source connecting two viewpoints is developed. Satisfactory shadow generation results are obtained for synthetic as well as real images.

3. Identity Verification by Relative 3-D Structure Using Multiple Facial Images

3.1 Overview

Machine recognition of faces has been a very active research topic in recent years [20]-[23]. As indicated in Section 1.2, most of existing face recognition algorithms are 2-D based. In this chapter, we propose a novel approach to identify a person with facial images. The approach uses a measure of 3-D facial structure without explicit 3-D reconstruction. Three reference points are first extracted to construct a reference plane in every image. By calculating a view-invariant relative depth, i.e., relative affine structure with respect to the obtained reference plane introduced in [3] which is equivalent to cross-ratio measurement, for each relevant feature point, an efficient face recognition algorithm is developed. The approach is a structure-based method which has the advantage of low sensibility to irrelevant data. Compared with other 3-D approaches that require specific structures in Euclidean space [40][41], the proposed method uses only a few facial feature points and requires no camera calibration. In addition, iterative training is not required which leads to the issue of convergence in the neural network approaches. Experimental results show that the developed approach performs satisfactorily with an experimental facial image database.

In the following sections, we first introduce related projection geometry for one and two cameras. The geometrical relationships between two cameras such as parallax and relative affine structure are discussed in Section 3.3, together with the geometrical meaning of such a structure which is expressed in terms of the invariant under perspective projection, i.e., cross-ratio. Algorithms for face recognition using relative affine structure are presented in Section 3.4. Simulation results for an experimental facial image database are given in Section 3.5. Finally, a brief summary is given in Section 3.6.

3.2 Projective geometry for one and two cameras

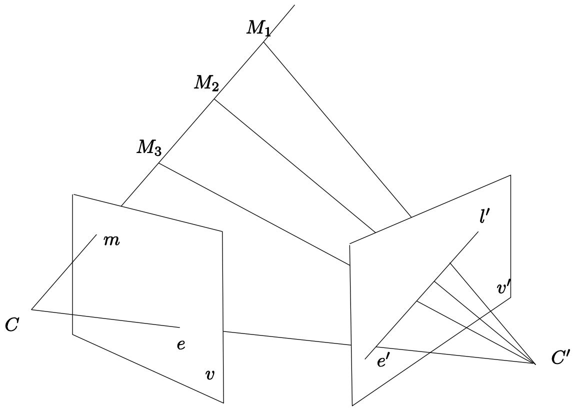


Figure 3.1 A scene with two cameras and three 3D points.

The basic procedure of projecting 3D points onto an image by a perspective camera can be described as

$$m \propto PM\ , \tag{6}$$

where $\propto$ denotes the equality up to a scaling factor, P is the 3×4 projection matrix, $M = \begin{bmatrix} X & Y & Z & 1 \end{bmatrix}^T$ and $m = \begin{bmatrix} x & y & 1 \end{bmatrix}^T$ represent the homogeneous coordinates of a 3D world point and the corresponding image point, respectively. In general, the image coordinate system is defined in terms of image pixels. The general form of the projection matrix can be represented as

$$P_{euc} \propto KP_0T = \begin{bmatrix} f_x & s & p_x \\ 0 & f_y & p_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} I \,|\, 0 \end{bmatrix} \begin{bmatrix} R & t \\ 0_3^T & 1 \end{bmatrix}. \tag{7}$$

In (7), *K* gives the intrinsic parameters of the camera, the imaging system. As for *T*, it describes the location and orientation of the camera with respect to the world coordinate system. It is a 4×4 matrix describing the pose of the camera in terms of a rotation *R* and a translation *t*, which give the extrinsic parameters. For an ideal camera model, both *K* and *T* are identity matrices and (7) becomes

$$m \triangleq P_0M\ . \tag{8}$$

Consider two cameras taking pictures of an object, as illustrated in Figure 3.1, wherein *C* and C_0 are the two optical centers of the two cameras and *v* and v_0 are their associated image planes, respectively. The projection of *C’* on *v*, *e* = *PC’*, observed from *C* and the projection of *C* on *v’* observed from *C'*, *e'* = *P'C*, are defined as the epipoles of the two cameras, respectively. Without loss of generality, we assume that the world coordinate system is aligned with the image coordinate system of camera *C*, thus the projection matrices for *C* and *C'* become

$$P = K_{3\times3}\left[I_{3\times3}\,|\,0\right] = \left[K\,|\,0\right], \tag{9}$$

$$P' = K'_{3\times3}\left[R_{3\times3}\,|\,t_{3\times1}\right] = \left[K'R\,|\,K't\right]. \tag{10}$$

In addition, we have, by definition, $PC = K_{3\times3}\left[I_{3\times3}\,|\,0_{3\times1}\right]C_{4\times1} = 0$ or $C \propto \left[0 \quad 0 \quad 0 \quad 1\right]^T$. Since e_0 is the projection of *C* on v_0,

$$e' = P'C = K't\,. \tag{11}$$

Consider a 3D point *M* whose depth is *z* with respect to the camera coordinate system of camera *C*. Its projection on the image plane *v*, from (8), is equal to

$$m \propto PM = K\widetilde{M}$$

with

$$M = \begin{bmatrix} \widetilde{M} \\ 1 \end{bmatrix} = \begin{bmatrix} zK^{-1}m \\ 1 \end{bmatrix}$$

if *m* is normalized as $(x, y, 1)^T$. The projection on image plane v_0 is then

$$m' \propto P'M \propto K'RK^{-1}m + \frac{1}{z}K't\,. \tag{12}$$

With the above geometrical relationships and coordinate transformations between two cameras, the authors of [3] derived the view invariant relative affine structure. The following section provides a brief review, together with its explicit geometric meaning.

3.3 Relative affine structure and its geometric meaning

In [3], an affine framework for perspective views is proposed which is captured by a simple equation based on an invariant called relative affine structure. It is shown in [3] that the framework unifies projection tasks including Euclidean, projective and affine in a natural and simple way.

While the algebraic form of the relative affine structure is given clearly in [3], as reviewed next, the direct relationship between the relative affine structure and a view-invariant cross-ratio under perspective projection is derived at the end of this section.

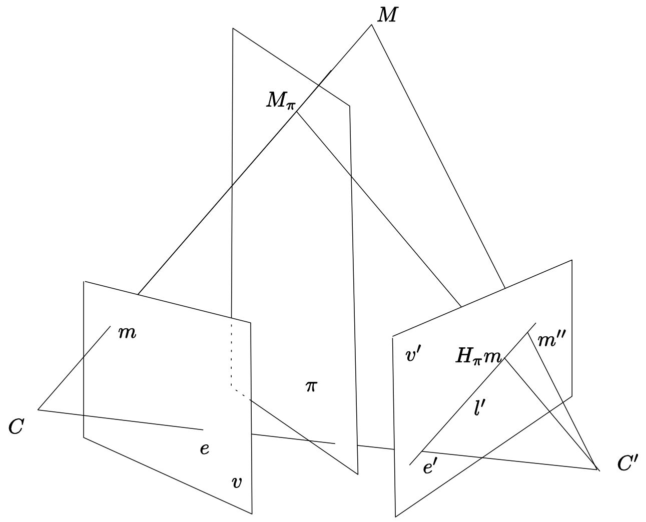


Figure 3.2 An example of parallax. M is a point which is not on the reference plane π.

Given a reference plane π where the image points m and m' are projections of a 3D point $M_\pi \in \pi$ on image planes v and v', respectively. The homography induced by π can be obtained by $M_\pi = H_1 m$ and $M_\pi = H_2 m'$ as follows:

$$m' = H_2^{-1} M_\pi = H_2^{-1} H_1 m = H_\pi m \,. \tag{13}$$

Since H_π has eight entries (nine minus a scale factor), H_π can be determined uniquely by solving a system of linear equations obtained from three point correspondences in general positions on π and the relationship $e' = H_\pi e$. Moreover, once H_π is computed we can use it to determine positions of points on p from a single image.

The homogeneous coordinates of π can be written as

$$\pi = \begin{bmatrix} n_{3\times 1} \\ d_\pi \end{bmatrix}, \tag{14}$$

where n and d_π describe the normal vector and the depth of π, respectively. For the projection m of M_π on the image plane v, we have $m = PM_\pi = [K \,|\, 0] M_\pi$. Since the depth of M_π is unknown, we can assume that

$$M_\pi = \begin{bmatrix} (K^{-1}m)_{3\times 1} \\ \rho \end{bmatrix}. \tag{15}$$

On the other hand, since M_π is on π, we have

$$\rho = \frac{-1}{d_\pi} n^T K^{-1} m. \tag{16}$$

Now, by projecting M_π on v', we have

$$m' = H_\pi m \propto P' M_\pi = K'(R - \frac{tn^T}{d_\pi})K^{-1}m. \tag{17}$$

For more general scenes wherein not all of the 3D points are co-planar, parallax will be produced. For instance, M is a 3D point which is not on the plane π in Figure 3.2. m'' and $H_\pi m$ are projections of M and M_π on v', respectively. From (12), (13), (17) and $e' = K't$, we have

$$m'' \propto K'RK^{-1}m + \frac{1}{z}K't = H_\pi m + (\frac{zn^T K^{-1} m + d_\pi}{d_\pi z})e'. \tag{18}$$

For a point $M = [zK^{-1}m, \quad 1]^T$ which is not on the reference plane π, the distance from M to π is equal to

$$d = \pi^T M = zn^T K^{-1} m + d_\pi. \tag{19}$$

Substituting (19) into (18), we have

$$m'' \propto H_\pi + \left(\frac{d}{d_\pi z}\right) e' = H_\pi m + \beta e'. \tag{20}$$

Since the value of the parallax term β in (17) is normalized, d_π can be dropped out, as stated in [3]. If we let $\beta_0 = 1$ for a reference point M_0 which is not on the reference plane (see Figure 3.3), we are left with

$$d_\pi = \frac{d_0}{z_0}$$

and (20) can be rewritten as

$$m^{''} \cong H_\pi m + \lambda e' \tag{21}$$

with λ being the relative affine structure. In the following paragraph, we will investigate its relationship with cross-ratio.

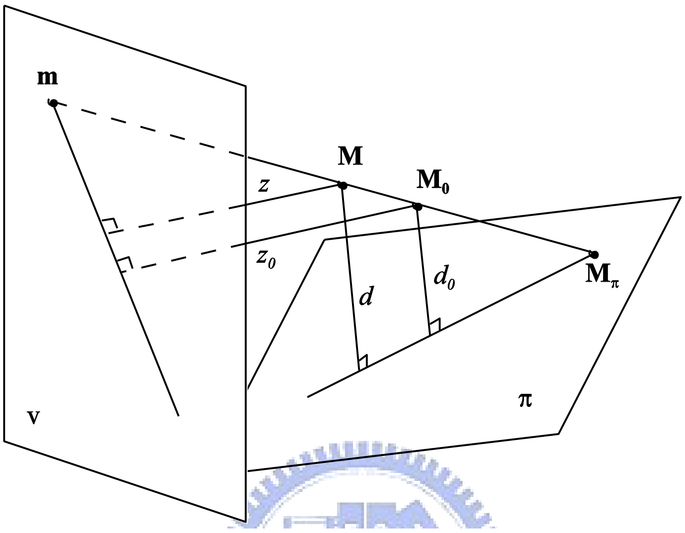


Figure 3.3 The geometry of the relative affine structure. *z* and *z*$_0$ are depths of *M* and *M*$_0$ with respect to *v*, respectively.

In (21), it is not difficult to see that λ is an invariant quantity since the variables z_0, z, d_0, d are governed by camera *C* only. Consider Figure 3.3, by extending $\overline{MM_0}$, we can obtain two intersection points *m* and M_π, which are on *v* and the reference plane π, respectively. By triangular similarity, we have

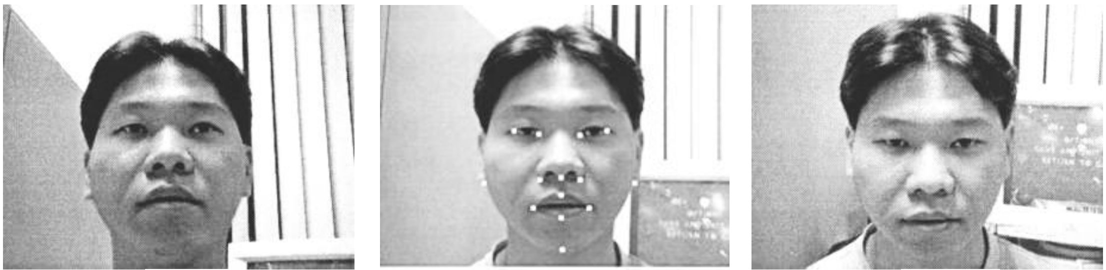
$$\lambda = \frac{z_0}{z}\frac{d}{d_0} = \frac{\overline{mM_0}}{\overline{mM}}\frac{\overline{MM\pi}}{\overline{M_0M_\pi}} = CR(m, M, M_0, M_\pi)\,. \tag{22}$$

This leads to a conclusion that relative affine structure is in fact a measure of cross-ratio. Since it is view-invariant, λ can be used as a useful feature to describe object structure. Algorithm 3 summarizes the process to calculate the relative affine structure for *n* pairs of image points. By calculating relative affine structures of facial features of persons, we have developed an identity verification system based on face recognition using λ, as discussed next.

ALGORITHM 3 ***Computation of relative affine structure for n pairs of image points***

- Calculate the fundamental matrix F with 8 pairs of correspondences.
- Derive the epipoles e and e' using $F^T e^{'} = 0$ and $Fe = 0$.
- Derive the homography H_π of the reference plane with an epipole and 3 pairs of point correspondences.
- Choose a pair of correspondence m_0 and $m_0^{'}$ where m_0 and $m_0^{'}$ are image points on the left image and the right image, respectively.
- Scale H_π such that $m_0^{'} \cong H_\pi m_0 + e^{'}(\lambda_0 = 1)$.
- Obtain λ_i with $m_i^{'} \cong H_\pi m_i + \lambda_i e^{'}, 1 \leq i \leq n-1$.

3.4 Face recognition using relative affine structures

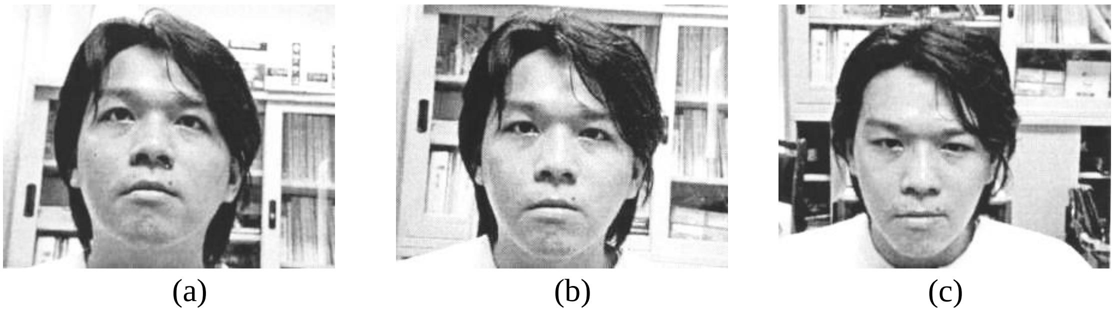


(a) (b) (c)

Figure 3.4 Face images of person *A*. From left to right side, the images are labeled as A_u, A_f, and A_d, respectively.

With the properties of the view-invariant relative affine structure investigated in the previous section, this section presents the proposed approach to face recognition using such invariants. Recall that the relative affine structure of an object point depends only on the configuration of the first camera *C*, the position of the reference plane π and the reference point M_0. So, two facial images are used first to derive the relative affine structure for each feature point. The first image is denoted as the reference image and the extracted facial features are stored together with the obtained relative affine structures. To verify the identity of a new facial image, a new set of relative affine structures are obtained by the reference facial image and the new image. The similarity between the stored relative affine structures and the new set of relative affine structures is evaluated. Finally, the identity is verified by checking whether the similarity is higher than some specified thresholds.

In this chapter, in order to focus on the correctness of the theory, feature points are obtained *manually* from facial images taken from different points of view. On each given face image, fifteen feature points including eye and mouth corners, nose tip, ear lobes, etc. are extracted as shown in Figure 3.4(b). The image of the front view of person *A* is labeled as A_f while the upward and downward looking facial images are labeled as A_u and A_d, respectively. In the same manner, three images are also taken for each person in the database. For example, Figure 3.5 shows the images obtained for person *B*.



(a) (b) (c)

Figure 3.5 Face images of person *B*. From left to right side, the images are labeled as B_u, B_f, and B_d, respectively.

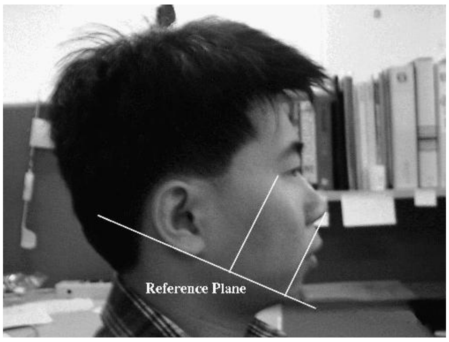


Figure 3.6 Face image of side view of person *F*. The reference plane is defined by the two ear lobes and the chin. The 2D projections on images of these three feature points are used to calculate relative affine structures.

Table 1 shows the relative affine structures obtained for persons *A* and *B* with A_u and B_u being the reference images, respectively. Since the reference plane is defined by right ear lobe (point 14), right ear lobe (point 13) and chin (point 15), as illustrated in Figure 3.6, the relative affine structure values of these three points are all zeros. The value of the relative affine structure of the nose tip (point 12), which is the reference point M_0, is defined as 1 for normalization. Since the depth from the camera to a person is usually several meters, z_0/z in (22) is close to 1. Thus, the values of other relative affine structures given in (22) are close to d/d_0. From Figure 3.6, we can see that the ratio for the eye corner is close to unity, the ratio for the mouth corner is about 0.4, while the ratios for the upper and lower lips are about 0.65 and 0.45, respectively.

In our experiments, we use six groups of facial images for persons *A* through *F* (see Figure 3.7 for facial images C_f through F_f). Each group consists of three images from three different points of view. With a personal computer equipped with a 333 MHz Pentium II processor and memory of 128

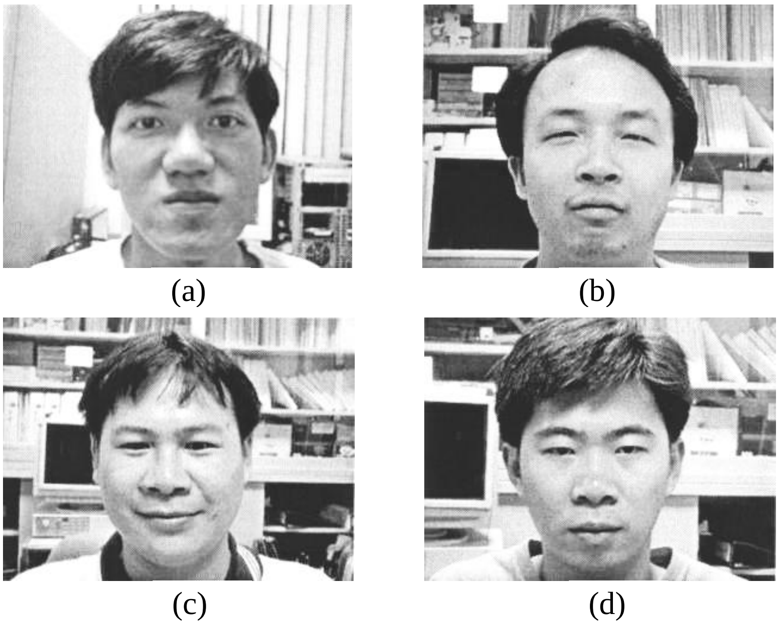
MB, the program is implemented with MATLAB 6.1 under Microsoft Windows 2000 and spends 0.1 seconds to obtain the relative affine structure for each data set, e.g., $A_u\_A_f$ in Table 1 with A_u being the reference image. A database is used to store such information obtained from the facial images. Details of the verification process using this database and possible ways to improve the stability of the verification results are given in the next section.

Table 1. Relative affine structures obtained for persons A (λ_{1i}) and B (λ_{2i}) using $A_u\_A_f$ and $B_u\_B_f$, respectively.

i	Feature point	λ_{1i}	λ_{2i}
1	Right eye corner (outer)	0.9951	1.0111
2	Right eye corner (inner)	0.9050	1.0391
3	Left eye corner (inner)	0.8112	1.0400
4	Left eye corner (outer)	0.7242	1.0590
5	Mouth corner (right)	0.4358	0.4594
6	Mouth corner (left)	0.4228	0.3430
7	Upper lip	0.6663	0.6598
8	Lower lip	0.4748	0.4518
9	Nose (right)	0.7256	0.7436
10	Nose (left)	0.6808	0.7281
11	Nose (center)	0.7734	0.8849
12	Nose (tip)	1.0000	1.0000
13	Ear lobe (right)	0.0000	0.0000
14	Ear lobe (left)	0.0000	0.0000
15	Chin	0.0000	0.0000

3.5 Experimental results

This section gives some experimental results of face recognition. For example, given the relative affine structures previously stored in the database for $X_u\_X_f$ and a facial image of an unknown person Y, we can investigate the identity of Y by evaluating the similarity between the relative affine structures for $X_u\_X_f$ and that for X_u_Y. The result of the comparison is then transformed into a score of matching error. If the score exceeds a threshold, the unknown person Y is not identified as person X.


(a) (b)

(c) (d)

Figure 3.7 Facial images (a) C_f, (b) D_f, (c) E_f, (d) F_f.

Table 2. Relative affine structures for $A_{u}\_A_f$ (λ_{1i}) and $A_{u}\_A_d$ (λ_{2i}), and their dissimilarity Ds_i = max($\lambda_{1i}/\lambda_{2i}$, $\lambda_{2i}/\lambda_{1i}$).

i	Feature point	λ_{1i}	λ_{2i}	Ds_i
1	Right eye corner (outer)	0.9951	0.9510	1.0463
2	Right eye corner (inner)	0.9050	0.8961	1.0100
3	Left eye corner (inner)	0.8112	0.8183	1.0087
4	Left eye corner (outer)	0.7242	0.7189	1.0073
5	Mouth corner (right)	0.4358	0.4271	1.0204
6	Mouth corner (left)	0.4228	0.4409	1.0428
7	Upper lip	0.6663	0.6719	1.0084
8	Lower lip	0.4748	0.4871	1.0259
9	Nose (right)	0.7256	0.7243	1.0018
10	Nose (left)	0.6808	0.6853	1.0066
11	Nose (center)	0.7734	0.7593	1.0185
12	Nose (tip)	1.0000	1.0000	1.0000
13	Ear lobe (right)	0.0000	0.0000	1.0000
14	Ear lobe (left)	0.0000	0.0000	1.0000
15	Chin	0.0000	0.0000	1.0000
Overall dissimilarity				1.2141

Table 2 shows the relative affine structure values for the fifteen facial features calculated for $A_{u}\_A_f$ and $A_{u}\_A_d$. Here, the dissimilarity between two corresponding relative affine structures, say λ_{1i} and λ_{2i}, is calculated as Ds_i = max($\lambda_{1i}/\lambda_{2i}$, $\lambda_{2i}/\lambda_{1i}$). For feature points lay on the reference plane, the relative affine structures are 0's by definition and the dissimilarity values are set to 1. Eventually, the overall dissimilarity between these two set of relative affine structures are defined as the product of all Ds_i's. For this example, the person with facial image A_d will be identified as person *A* since the overall dissimilarity, denoted as $DS(A_u\_A_f, A_u\_A_d)$, is very close to 1.

Table 3 gives results similar to that in Table 2 but using facial image B_d of person *B* in place of A_d. It is readily observable that there are major differences between quite a few corresponding relative affine structure pairs. In particular, if $\lambda_{1i} \times \lambda_{2i} < 0$, that means the feature points are not on the same side of the reference plane in the 3D space, the dissimilarity value are set to 2 which leads to a big contribution to the overall dissimilarity. Since the overall dissimilarity of this example exceeds the threshold, person *B* is not identified as person *A*.

Table 3. Relative affine structures for $A_u\_A_f$ (λ_{1i}) and $A_u\_B_d$ (λ_{2i}), and their dissimilarity $Ds_i = \max(\lambda_{1i}/\lambda_{2i},\ \lambda_{2i}/\lambda_{1i})$

i	Feature point	λ_{1i}	λ_{2i}	Ds_i
1	Right eye corner (outer)	0.9951	1.0243	1.0293
2	Right eye corner (inner)	0.9050	2.3284	2.5727
3	Left eye corner (inner)	0.8112	3.5765	4.4088
4	Left eye corner (outer)	0.7242	4.5082	6.2254
5	Mouth corner (right)	0.4358	43.423	99.632
6	Mouth corner (left)	0.4228	-2.701	2.0000
7	Upper lip	0.6663	-3.049	2.0000
8	Lower lip	0.4748	-6.584	2.0000
9	Nose (right)	0.7256	-2.721	2.0000
10	Nose (left)	0.6808	-0.186	2.0000
11	Nose (center)	0.7734	-0.711	2.0000
12	Nose (tip)	1.0000	0.9999	1.0000
13	Ear lobe (right)	0.0000	0.0000	1.0000
14	Ear lobe (left)	0.0000	0.0000	1.0000
15	Chin	0.0000	0.0000	1.0000
Overall dissimilarity (*DS*)				4.63E+05

To further improve the stability of the verification system, every facial image can be used as the reference image and a composite measure of dissimilarity, say the geometric mean of individual results, can be obtained. Table 4 shows the result of the verification of A_f using A_u and A_d while Table 5 shows similar results by using B_f in place of A_f. The composite dissimilarity 1.5821 in Table 4 indicates that the person with facial image A_f can be identified as person *A*. On the other hand, it is obvious that B_f is not a facial image of person *A* since the composite dissimilarity in Table 5 is too high.

By using the composite dissimilarity measure, a more robust identity verification system is developed and more experimental results are obtained. Table 6 shows the composite dissimilarity for the verifications of facial images A_u through F_u using frontal and downward looking facial images of different people. Similarly, Table 7 verifies facial images A_f through F_f and Table 8 verifies facial images A_d through F_d, respectively. It can be seen from these results that the threshold for similarity can be set comfortably at 2.5 for the composite dissimilarity that every person in our database can be correctly verified with the proposed approach.

Table 4. Verification of A_f using A_u and A_d

	Overall dissimilarity
$DS(A_u\_A_f, A_u\_A_d)$	1.2141
$DS(A_f\_A_u, A_f\_A_d)$	1.9270
$DS(A_d\_A_u, A_d\_A_f)$	1.6926
Composite dissimilarity	1.5821

Table 5. Verification of B_f using A_u and A_d

	Overall dissimilarity
$DS(A_u\_B_f, A_u\_A_d)$	499057.33
$DS(B_f\_A_u, B_f\_A_d)$	631.10
$DS(A_d\_A_u, A_d\_B_f)$	1955.54
Composite dissimilarity	8508.22

Table 6. Composite dissimilarities for the verification of facial images A_u through F_u

	A_f, A_d	B_f, B_d	C_f, C_d	D_f, D_d	E_f, E_d	F_f, F_d
A_u	**1.58**	22925.14	1509.41	1.6456.43	110.70	1995.84
B_u	95.19	**1.87**	439.52	1.48E+05	1.50E+06	1.23E+05
C_u	12.17	211.89	**1.94**	1.22E+05	351.99	86389.45
D_u	4.08E+05	122.75	3055.30	**1.61**	50602.31	10.73
E_u	7.85E+06	2861.12	38943.89	62857.37	**1.70**	810.17
F_u	2.89E+05	2348.07	206.42	32115.82	1.29E+07	**1.89**

Table 7. Composite dissimilarities for the verification of facial images A_f through F_f

	A_u, A_d	B_u, B_d	C_u, C_d	D_u, D_d	E_u, E_d	F_u, F_d
A_f	**1.58**	13738.93	9625.68	26941.43	8.06E+06	1.02E+06
B_f	8508.22	**1.87**	372.59	18755.60	1.15E+06	8985.88
C_f	11998.25	14275.11	**1.94**	2791.98	2013.52	5650.15
D_f	3042.51	310.53	51170.61	**1.61**	425738	195.46
E_f	6979.54	1.90E+06	171595	3329.84	**1.70**	105005
F_f	151.60	2.03E+06	1146.79	1262.61	134511	**1.89**

Table 8. Composite dissimilarities for the verification of facial images A_d through F_d

	A_u, A_f	B_u, B_f	C_u, C_f	D_u, D_f	E_u, E_f	F_u, F_f
A_d	**1.58**	149.81	11.38	2654.73	1336.72	162.87
B_d	705.13	**1.87**	422.78	3.18	4536.51	16885.11
C_d	10160.21	24.72	**1.94**	34498.32	6919.54	762.24
D_d	8318.70	1.09E+07	5.61E+05	**1.61**	1.75E+05	20261.21
E_d	14.15	3155.22	24.01	781.89	**1.70**	423.75
F_d	726.86	219.91	8222.20	4.88	5292.08	**1.89**

As for the sensitivity of the proposed algorithm, the relative affine structure is actually cross-ratio in a form which is quite stable numerically. This can be seen from Figure 3.3 that the error of feature detection, in terms of variance of image pixels on the image plane, will results in minor change in the depth of the spatial structure, e.g., z and z_0, associated with a face. From above simulation results, it seems that differences among face structures of different individuals are much more significant than the differences due to the error of feature detection of facial images of the same person, which gives the robustness of the proposed approach.

3.6 Summary

This chapter presents a study on computer vision technique and its application in face recognition to achieve identity verification. The explicit relationship between the relative affine structure and the cross-ratio - an invariant under perspective projection, is addressed. Subsequently, relative affine structures derived from multiple images are used for face recognition. The proposed method neither requires camera calibration nor reconstructs 3D models. Moreover, as long as feature points of facial images are located accurately, the orientation and depth of the face are allowed to very more freely. As shown in our preliminary experiments, the proposed approach does achieve satisfactory results given the feature points of facial images. Slightly large scale of face database can be established for further investigation of the performance.

4. Practical Error Analysis of Cross-ratio-based Planar Localization

4.1 Overview

As mentioned in Section 1.3, cross-ratio plays an important role in many recognition and reconstruction algorithms which are based on projective geometry [1][2]. Thus, error associated with the calculation of cross-ratio must be carefully analyzed and controlled so as to avoid too much negative influence in the final reconstruction results. However, most of studies of sensitivity analysis of cross-ratio only consider the computation of cross-ratio itself instead of the final localization or reconstruction results. In this chapter, we propose an efficient way of analyzing localization error for systems which use cross-ratio for planar localization.

Through the 1*st*-order approximation of the derived one dimensional error function, we first inspect the linear nature of localization error due to small inaccuracy in image data. Similar properties of the localization error due to two dimensional noises are then investigated. In particular, an approximation of a nominal boundary of error ellipse can be determined efficiently for one of image points being affected by radially symmetric errors of a fixed magnitude. Based on such a computationally efficient error analysis, one may obtain the picture of resultant regions of localization error in advance, instead of generating similar results by using a large amount of synthesized noisy data for a particular cross-ratio configuration, and select proper reference points in an image accordingly.

The rest of this chapter is organized as follows. Reconstruction error resulted from cross-ratio computation due to 1-D and 2-D noises in image data is formulated in Section 4.2. We determine how error propagates, including its direction and magnitude range, through a linear approximation of such a cross-ratio-based formulation. In Section 4.3, synthesized noises are added to real data in the experiments for the verification of the theoretical investigations. Finally, a summary is given in Section 4.4.

4.2 Error analysis of cross-ratio-based localization

Referring to Section 2.3 and (3), one can develop a localization system based on the invariant cross-ratio, assuming perfect image acquisition and feature extraction. However, measurement uncertainty and system noise, such as quantization errors of 2-D coordinates of feature points in an image plane due to limited image resolution, usually occur in practice. These uncertainties will propagate through computation process, resulting in erroneous localizations or reconstructions. In this section, we will investigate how the error propagates in the reconstruction process of the above localization system.

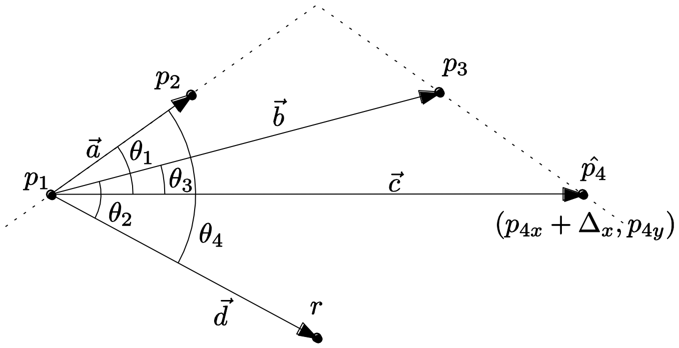


Figure 4.1 A general configuration of coplanar points where p_1 is the origin to compute cross-ratio CR_{p1}.

A general configuration of coplanar points for cross-ratio-based localization is shown in Figure 4.1. Here, we also denote image points with lowercase letters and scene points with uppercase letters (not shown). Assume P_1, P_2, P_3, P_4 are known planar points in 3-D space with P_1 and P_4, as well as p_1 and p_4, being origins of two cross-ratios. The position of a scene point R (or the location of a robot), which corresponds to image point r, can be determined with the procedure described in Section 2.3. For simplicity, let $\overrightarrow{p_1 r} = (d_x, d_y)$ and assume the location of p_4 has noise Δx along x-direction and is extracted as $\hat{p}_4$, we have

$$[p_2, p_3, \hat{p}_4, r]_{p_1} \triangleq CR_{p1} = \frac{\begin{vmatrix} a_x & c_x+\Delta_x \\ a_y & c_y \end{vmatrix}\begin{vmatrix} b_x & d_x \\ b_y & d_y \end{vmatrix}}{\begin{vmatrix} b_x & c_x+\Delta_x \\ b_y & c_y \end{vmatrix}\begin{vmatrix} a_x & d_x \\ a_y & d_y \end{vmatrix}} \triangleq \frac{\hat{q}_1}{\hat{q}_2} = \frac{q_1 - k_{bd} a_y \Delta_x}{q_2 - k_{ad} b_y \Delta_x}, \tag{23}$$

where $k_{bd} = \begin{vmatrix} b_x & d_x \\ b_y & d_y \end{vmatrix}$ and $k_{ad} = \begin{vmatrix} a_x & d_x \\ a_y & d_y \end{vmatrix}$. From (3) and (23) we have

$$\begin{aligned}&(\hat{q}_1 K_{BC} A_y - \hat{q}_2 K_{AC} B_y)X + (\hat{q}_2 K_{AC} B_x - \hat{q}_1 K_{BC} A_x)Y \\ &= \hat{q}_2 K_{AC} \begin{vmatrix} B_x & P_{1x} \\ B_y & P_{1y} \end{vmatrix} - \hat{q}_1 K_{BC} \begin{vmatrix} A_x & P_{1x} \\ A_y & P_{1y} \end{vmatrix}\end{aligned}$$

which yields the line equation of $\overrightarrow{P_1 R}$

$$\begin{aligned}&\left[(q_1 - k_{bd}a_y\Delta_x)K_{BC}A_y - (q_2 - k_{ad}b_y\Delta_x)K_{AC}\,B_y\right]X\\&+\left[(q_2-k_{ad}b_y\Delta_x)K_{AC}B_x - (q_1-k_{bd}a_y\Delta_x)K_{BC}\,A_x\right]Y\\&=(q_2-k_{ad}b_y\Delta_x)K_{AC}\begin{vmatrix}B_x & P_{1x}\\B_y&P_{1y}\end{vmatrix}-(q_1-k_{bd}a_y\Delta_x)K_{BC}\begin{vmatrix}A_x&P_{1x}\\A_y&P_{1y}\end{vmatrix}.\end{aligned}\tag{24}$$

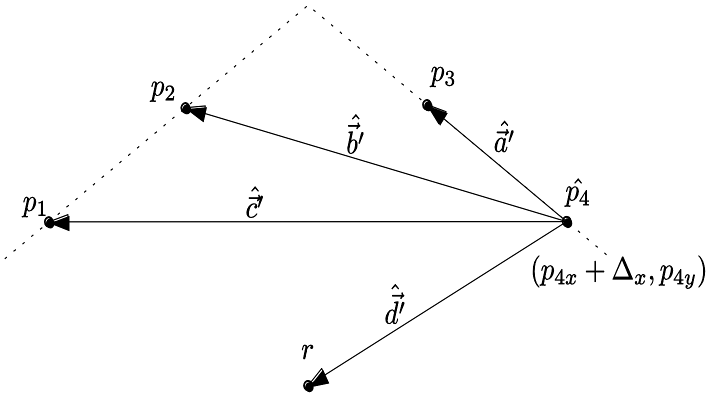


Figure 4.2 $\hat{p}_4$ is used as the origin to compute cross-ratio $CR_{\hat{p}4}$

On the other hand, with $\hat{p}4$ being the origin, as shown in Figure 4.2, $CR_{\hat{p}4}$ can be computed as

$$\begin{aligned}&\frac{\begin{vmatrix}a'_x-\Delta_x & c'_x-\Delta_x\\ a'_y & c'_y\end{vmatrix}\begin{vmatrix}b'_x-\Delta_x & d'_x-\Delta_x\\ b'_y & d'_y\end{vmatrix}}{\begin{vmatrix}b'_x-\Delta_x & c'_x-\Delta_x\\ b'_y & c'_y\end{vmatrix}\begin{vmatrix}a'_x-\Delta_x & d'_x-\Delta_x\\ a'_y & d'_y\end{vmatrix}}\triangleq\frac{\hat{q}'_1}{\hat{q}'_2}\\&=\frac{q'_1+u_1k'_{ac}\Delta_x+u_2k'_{bd}\Delta_x+u_1u_2\Delta_x^2}{q'_2+u_3k'_{bc}\Delta_x+u_4k'_{ad}\Delta_x+u_3u_4\Delta_x^2},\end{aligned}\tag{25}$$

where

$$k'_{ac}=\begin{vmatrix}a^{'}_x & c^{'}_x\\ a^{'}_y & c^{'}_y\end{vmatrix},\ k'_{bd}=\begin{vmatrix}b^{'}_x & d^{'}_x\\ b^{'}_y & d^{'}_y\end{vmatrix},\ k'_{bc}=\begin{vmatrix}b^{'}_x & c^{'}_x\\ b^{'}_y & c^{'}_y\end{vmatrix},\ k'_{ad}=\begin{vmatrix}a^{'}_x & d^{'}_x\\ a^{'}_y & d^{'}_y\end{vmatrix},$$

and $u_1=b'_y-d'_y, u_2=a'_y-c'_y, u_3=a'_y-d'_y, u_4=b'_y-c'_y$. Similarly, from (3) and (25), we can obtain line equation of $\overrightarrow{\hat{P}_4R}$ as

$$\begin{aligned}&(\hat{q}'_1K'_{BC}A'_y-\hat{q}'_2K'_{AC}B'_y)X+(\hat{q}'_2K'_{AC}B'_x-\hat{q}'_1K'_{BC}A'_x)Y\\&=\hat{q}'_2K'_{AC}\begin{vmatrix}B'_x & P_{4x}\\ B_{y'} & P_{4y}\end{vmatrix}-\hat{q}'_1K'_{BC}\begin{vmatrix}A'_x & P_{4x}\\ A'_y & P_{4y}\end{vmatrix}\end{aligned}\tag{26}$$

where (P_{4x}, P_{4y}) is coordinate of image point P_4. It is easy to see that (24) and (26) are of the form

$$\begin{cases} \alpha_1' X + \beta_1' Y = \gamma_1' \\ \alpha_2' X + \beta_2' Y = \gamma_2' \end{cases}$$

Therefore, by solving the above equations, the robot position can be obtained as

$$R \triangleq (R_x, R_y) = \left(\frac{\begin{vmatrix} \gamma_1' & \beta_1' \\ \gamma_2' & \beta_2' \end{vmatrix}}{\begin{vmatrix} \alpha_1' & \beta_1' \\ \alpha_2' & \beta_2' \end{vmatrix}}, \frac{\begin{vmatrix} \alpha_1' & \gamma_1' \\ \alpha_2' & \gamma_2' \end{vmatrix}}{\begin{vmatrix} \alpha_1' & \beta_1' \\ \alpha_2' & \beta_2' \end{vmatrix}} \right) \tag{27}$$

To simplify (27), by skipping high order terms of Δx, we have

$$\begin{aligned}
\begin{vmatrix} \hat{\gamma}_1 & \hat{\beta}_1 \\ \hat{\gamma}_2 & \hat{\beta}_2 \end{vmatrix} \approx \begin{vmatrix} \gamma_1 & \beta_1 \\ \gamma_2 & \beta_2 \end{vmatrix} + \Bigg[& (q_1 u_1 k_1' + q_1 u_2 k_2' - q_1' k_2 a_y) K_2 K_2' \left(\begin{vmatrix} R_x' A_x & A_x' \\ R_x A_y & A_y' \end{vmatrix} + A_x A_x' (P_{1y} - P_{4y}) \right) \\
& - (q_1 u_3 k_3' + q_1 u_4 k_4' - q_2' k_2 a_y) K_2 K_1' \left(\begin{vmatrix} R_x' A_x & B_x' \\ R_x A_y & B_y' \end{vmatrix} + A_x B_x' (P_{1y} - P_{4y}) \right) \\
& + (q_2 u_3 k_3' + q_1 u_4 k_4' - q_2' k_4 b_y) K_1 K_1' \left(\begin{vmatrix} R_x' B_x & A_x' \\ R_x B_y & A_y' \end{vmatrix} + B_x A_x' (P_{1y} - P_{4y}) \right) \\
& + (q_2 u_1 k_1' + q_2 u_2 k_2' - q_1' k_4 b_y) K_1 K_2' \left(\begin{vmatrix} R_x' B_x & A_x' \\ R_x B_y & A_y' \end{vmatrix} + B_x A_x' (P_{1y} - P_{4y}) \right) \Bigg] \Delta_x \\
\triangleq \begin{vmatrix} \gamma_1 & \beta_1 \\ \gamma_2 & \beta_2 \end{vmatrix} + \Bigg[& M_1 \left(\begin{vmatrix} R_x' A_x & A_x' \\ R_x A_y & A_y' \end{vmatrix} + A_x A_x' (P_{1y} - P_{4y}) \right) + M_2 \left(\begin{vmatrix} R_x' A_x & B_x' \\ R_x A_y & B_y' \end{vmatrix} + A_x B_x' (P_{1y} - P_{4y}) \right) \\
& + M_3 \left(\begin{vmatrix} R_x' B_x & A_x' \\ R_x B_y & A_y' \end{vmatrix} + B_x A_x' (P_{1y} - P_{4y}) \right) + M_4 \left(\begin{vmatrix} R_x' B_x & A_x' \\ R_x B_y & A_y' \end{vmatrix} + B_x A_x' (P_{1y} - P_{4y}) \right) \Bigg] \Delta_x,
\end{aligned}$$

$$\begin{vmatrix} \hat{\alpha}_1 & \hat{\beta}_1 \\ \hat{\alpha}_2 & \hat{\beta}_2 \end{vmatrix} \approx \begin{vmatrix} \alpha_1 & \beta_1 \\ \alpha_2 & \beta_2 \end{vmatrix} + \left[M_1 \begin{vmatrix} A_x & A'_x \\ A_y & A'_y \end{vmatrix} + M_2 \begin{vmatrix} A_x & B'_x \\ A_y & B'_y \end{vmatrix} + M_3 \begin{vmatrix} B_x & B'_x \\ B_y & B'_y \end{vmatrix} + M_4 \begin{vmatrix} B_x & A'_x \\ B_y & A'_y \end{vmatrix} \right] \Delta_x,$$

and

$$\begin{vmatrix} \hat{\alpha}_1 & \hat{\gamma}_1 \\ \hat{\alpha}_2 & \hat{\gamma}_2 \end{vmatrix} \approx \begin{vmatrix} \alpha_1 & \gamma_1 \\ \alpha_2 & \gamma_2 \end{vmatrix} + \left[M_1 \left(\begin{vmatrix} R_y A_x & A'_x \\ R'_y A_y & A'_y \end{vmatrix} + A_y A'_y (P_{4x} - P_{1x}) \right) + M_2 \left(\begin{vmatrix} R_y A_x & B'_x \\ R'_y A_y & B'_y \end{vmatrix} + A_y B'_y (P_{4x} - P_{1x}) \right) \right.$$
$$\left. + M_3 \left(\begin{vmatrix} R_y B_x & B'_x \\ R'_y B_y & B'_y \end{vmatrix} + B_y B'_y (P_{4x} - P_{1x}) \right) + M_4 \left(\begin{vmatrix} R_y B_x & A'_x \\ R'_y B_y & A'_y \end{vmatrix} + B_y A'_y (P_{4x} - P_{1x}) \right) \right] \Delta_x$$

where $\begin{vmatrix} \alpha_1 & \beta_1 \\ \alpha_2 & \beta_2 \end{vmatrix}$, $\begin{vmatrix} \gamma_1 & \beta_1 \\ \gamma_2 & \beta_2 \end{vmatrix}$ and $\begin{vmatrix} \alpha_1 & \gamma_1 \\ \alpha_2 & \gamma_2 \end{vmatrix}$ are the corresponding noise-free terms. Thus, the approximate location of robot R becomes

$$\hat{R}_x = \frac{\begin{vmatrix} \hat{\gamma}_1 & \hat{\beta}_1 \\ \hat{\gamma}_2 & \hat{\beta}_2 \end{vmatrix}}{\begin{vmatrix} \hat{\alpha}_1 & \hat{\beta}_1 \\ \hat{\alpha}_2 & \hat{\beta}_2 \end{vmatrix}}, \quad \hat{R}_y = \frac{\begin{vmatrix} \hat{\alpha}_1 & \hat{\gamma}_1 \\ \hat{\alpha}_2 & \hat{\gamma}_2 \end{vmatrix}}{\begin{vmatrix} \hat{\alpha}_1 & \hat{\beta}_1 \\ \hat{\alpha}_2 & \hat{\beta}_2 \end{vmatrix}}. \tag{28}$$

Since (28) has the form of

$$\begin{cases} \hat{R}_x = \dfrac{E\Delta x + F}{I\Delta x + J} & (29) \\ \hat{R}_y = \dfrac{G\Delta x + H}{I\Delta x + J} & (30) \end{cases}$$

for constants E through J, we can obtain the following linear equation by eliminating Δx [3]

$$\hat{R}_y = \frac{IH - GJ}{IF - EJ} \hat{R}_x + \frac{GF - EH}{IF - EJ}. \tag{31}$$

The above equation gives the trajectory of the reconstructed locations of robot R due to relatively small image extraction errors in x-component of p_4 in Figure 4.1 (and Figure 4.2).

In general, as will be demonstrated with simulation results in the next section, if the 2-D image error of a feature point is within a reasonably small range, it can also be transformed approximately linearly into a planar region in the 3-D space of the reconstructed scene[4]. In particular, such a linear transformation of coordinate system will transform a circular region of image error into an elliptic

[3] One can show that by applying Taylor series expansion to (28), which gives a linear relationship between 2-D image extraction error and 3-D localization error, a linear equation identical to (31) can also be obtained.

[4] A formal derivation of such a property is omitted for brevity.

one in the above planar region. Therefore, with only transformations of the image error in two linearly independent directions, an approximate ellipse of reconstruction error can be obtained. Such error ellipses will be useful indicators for one to choose among point features in an image to establish the probabilistically most accurate planar location system using cross-ratios.

4.3 Simulation results

We conduct a series of simulations for the error analysis of cross-ratio-based planar localization for a real robot with synthesized noises added to some reference points in an image. The real scene used in our simulations is set up as shown in Figure 4.3. In these simulations, we consider the situation when extraction noises only affect a single image point. First, we investigate the characteristics of the localization error assuming 1-D noise along *x*-direction, as discussed in the previous section, as well as along other directions. Subsequently, nominal boundary of an error ellipse due to two dimensional noises is computed to approximate the real error region resulted from circularly distributed image inaccuracy. Finally, we give a cross-ratio-based localization scheme which adopts the proposed error analysis method to assist the selection of reference image points to optimize the reconstruction results.

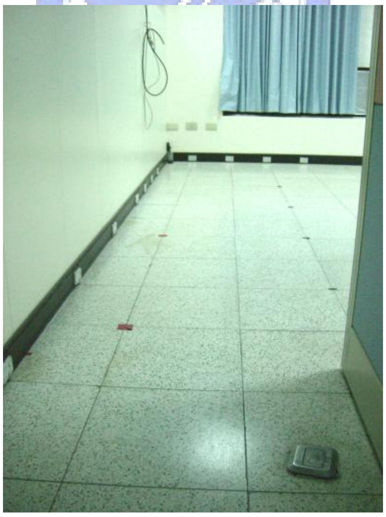


Figure 4.3 The scene which provides real data that used in our experiments. The reference points attached on the wall are co-planar. The size of each tile is $40\text{cm} \times 40\text{cm}$.

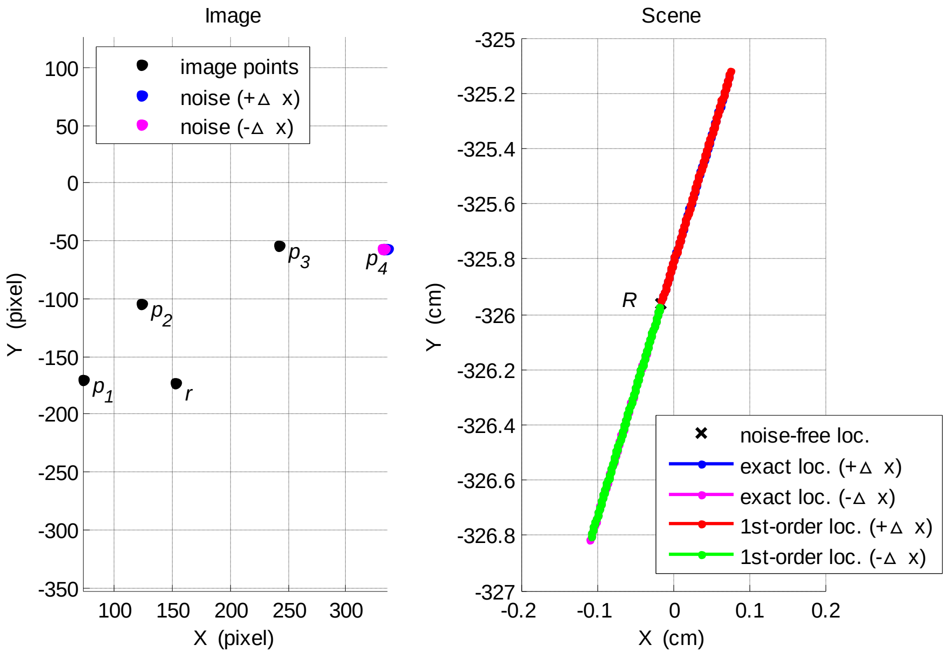


Figure 4.4 (Left) Extracted feature points in an input image. p_1 … p_4 are identified as images of reference points, and r is identified as the robot. Image extraction noises within a range of ± 2 pixels along x-direction are added to p_4. (Right) Trajectory of reconstructed robot locations: blue and magenta points are obtained by (27) while red and green lines are obtained by (31). The former are hardly visible since they are almost entirely covered by the latter. R is the robot location in 3-D space resulted from the noise-free extraction of image points.

Figure 4.4 illustrates the trajectory of the reconstructed robot locations due to noises within the range of ± 2 pixels along x-direction being added to p_4[5]. The locations obtained from linear equation (31) are represented in red and green color, corresponding to deviations of p_4 into $+x$ and $-x$ directions, respectively. Points in blue and magenta colors represent similar results but computed with original rational equation (27). One can see that the latter, which are drawn first, are hardly visible since (31) gives a nearly perfect approximation of the former.

Figure 4.5 illustrates results similar to Figure 4.4 but due to noise added to y-coordinate of p_4. In fact, similar results (which are omitted for brevity) can be obtained for 1-D noises in arbitrary directions. In general, if the 2-D image errors are within a reasonable small range, the errors can also be transformed approximately linearly into the 3-D space of the reconstructed scene. Figure 4.6 illustrates the trajectory of the reconstructed robot locations due to circularly distributed image extraction noises of 2 pixels added to p_4 as well as the error ellipse obtained from the linear

[5] It is assumed in the rest of the chapter that two cross-ratios involved in the computation use p_1 and p_4 as origins, respectively.

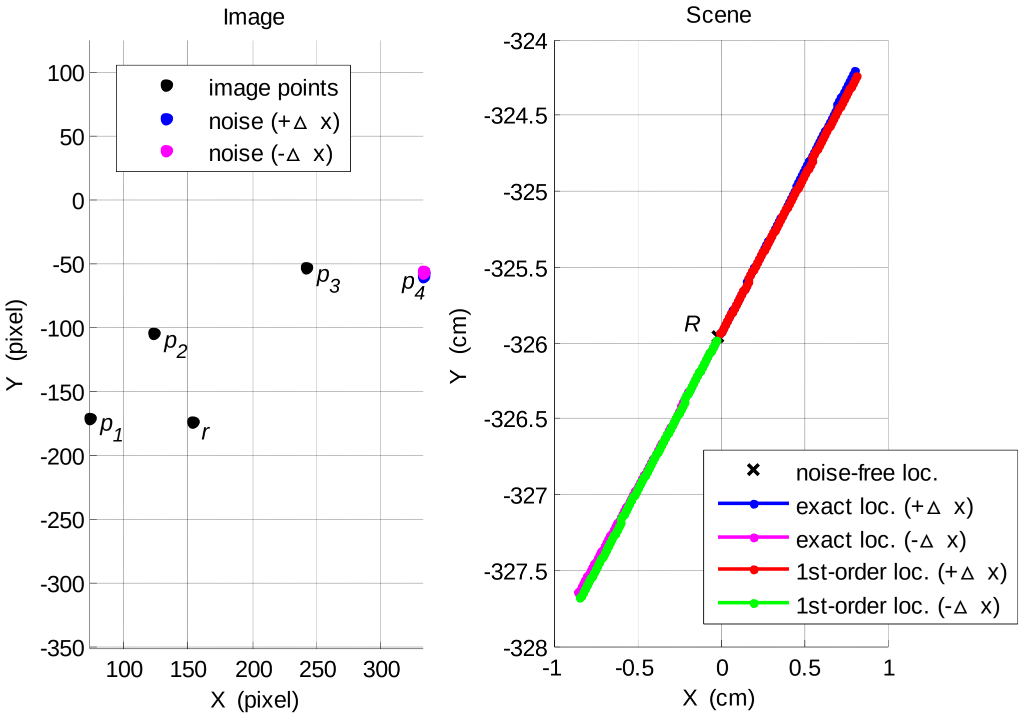


Figure 4.5 (Left) Extracted feature points in the input image. $p_1 \dots p_4$ are identified as images of reference points, and r is identified as the robot. Image extraction noises within a range of ± 2 pixels along y-direction are added to p_4. (Right) Trajectory of reconstructed robot locations: blue and magenta points are obtained by (27) while red and green lines are obtained by (31). R is the robot location in 3-D space resulted from noise-free extraction of image points.

transformation

$$\begin{bmatrix}\Delta\widehat{R}_x\\ \Delta\widehat{R}_y\end{bmatrix}=\begin{bmatrix}a & b\\ c & d\end{bmatrix}\begin{bmatrix}\Delta x\\ \Delta y\end{bmatrix}, \tag{32}$$

which is derived from $(\Delta x, \Delta y) = (2,0)$ and $(\Delta x, \Delta y) = (0,-2)$ (corresponding to reconstructed robot locations R_1 and R_2, respectively)[6]. It can be shown that the orientation of such an ellipse is

$$\theta_{ellipse} = \frac{1}{2}\arctan(\frac{-2(ac+bd)}{c^2+d^2-a^2-b^2}) \tag{33}$$

while its semimajor axis and semiminor axis are

$$a_{ellipse} = \sqrt{\frac{2(ad-bc)^2\sin(2\theta_{ellipse})}{(a^2+b^2+c^2+d^2)\sin(2\theta_{ellipse})-2(abcd)}} \tag{34}$$

[6] Eq. (32) can be used to derive the approximate error ellipse only if there is a linear relationship between image and reconstruction errors Various ways of inspecting such a relationship exist, but is not discussed here for brevity.

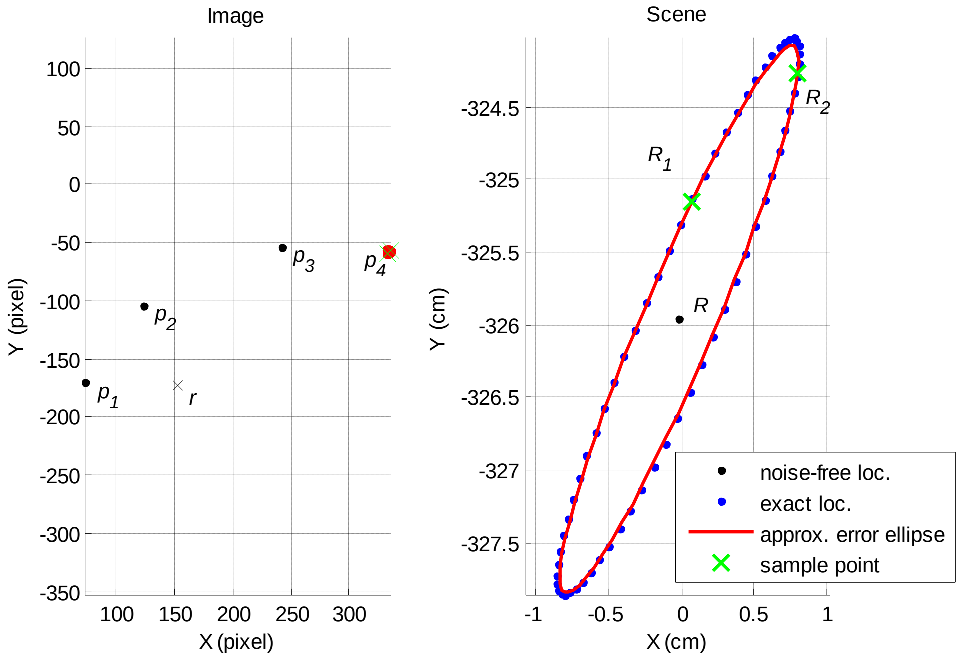


Figure 4.6 (Left) Extracted feature points in the input image. $p_1 \ldots p_4$ are identified as images of reference points, and r is identified as robot image. Circularly distributed image extraction noises of 2 pixels are added to p_4. (Right) Trajectory of reconstructed robot locations: blue points are obtained by (8), while the approximate error ellipse is obtained using (13). The reconstructed locations of robot due to image errors $(\Delta x, \Delta y) = (2, 0)$ and $(\Delta x, \Delta y) = (0, -2)$ on p_4 are at R_1 and R_2, respectively.

and

$$b_{ellipse} = \sqrt{\frac{2(ad-bc)^2 \sin(2\theta_{ellipse})}{(a^2+b^2+c^2+d^2)\sin(2\theta_{ellipse})+2(abcd)}}, \tag{35}$$

respectively.

One can easily see that the above elliptical trajectory can be used to appropriately express the spatial characteristics of the localization error without the computation of a lot of reconstructed robot locations using the expensive high-order equation (27). For the application of a general cross-ratio-based localization algorithm to a scene where many image features (points) can be extracted, multiple choices of reference points, as well as the origins for the computation of cross-ratios, are possible. Figure 4.7 shows simulation results similar to that given in Figure 4.6 but using reference point p_5 in place of p_2. According to the results obtained with either (27) or (32), localization results in Figure 4.7 give similar worst-case error (4.2cm), but with approximately twice the ellipse area, compared to that in Figure 4.6.

The above results suggest that when there are multiple choices of reference points or cross-ratio origins, one can perform the proposed analysis to predict possible localization errors for each choice and select the *optimal* one accordingly. For each choice, one needs to ensure first that the noises are restricted to a reasonable range that (32) obtained using two noisy samples of the reference point of interest can appropriately describe the localization error. Subsequently, an optimal choice can be determined by using (33), (34) and (35) to compare the direction of error, worst-case error, average error magnitude, or other metrics suggested by specific applications.

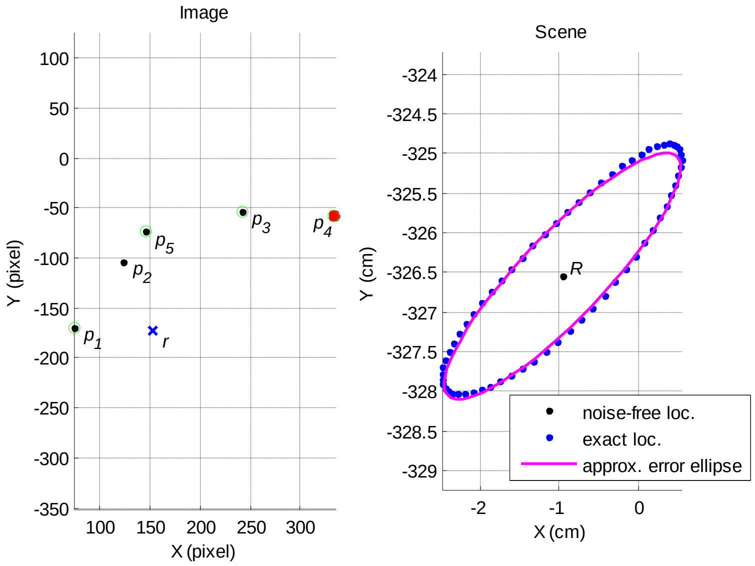


Figure 4.7 (Left) Extracted feature points in the input image. p_1 … p_5 are identified as images of reference points, and r is identified as robot image. Circularly distributed image extraction noises of 2 pixels are added to p_4. p_1, p_5, p_3, p_4 are selected to compute cross-ratios to locate the robot R in the scene. (Right) Trajectory of reconstructed robot locations. Blue points are obtained by (27), while the approximate error ellipse is obtained using (32).

4.4 Summary

As a geometric invariant under projective transformations, cross-ratio is the basis of many recognition and reconstruction algorithms which are based on projective geometry. We propose an efficient way to approximately analyze localization error for systems, which use cross-ratio for planar localization, by establishing a linear relationship between the error and small inaccuracy in measurements of image features due to 1-D and 2-D noises in the image space. Such an analysis will be useful for one to choose among point features, as well as cross-ratio origins, in an image to establish the probabilistically most accurate planar location system. The proposed approach is applicable whenever multiple choices of image features are available, which happens frequently in various computer vision applications, e.g., in robot navigation systems.

5. Finding Point Correspondence Using Local Similarity and Global Constraint Under Insignificant Scaling and Roll

5.1 Overview

Regarding the state of the art in research works related to extraction and correspondence establishment for image features, as partly discussed in Section 1.4, it seems that no general solution to the correspondence problem exists, due to ambiguous matches. In this chapter we propose a novel approach to feature extraction and correspondence establishment for images of indoor scenes. We assume the scaling and roll are insignificant (less than ±5° of roll angle) since image transforms include mainly pan and tilt in common reconstruction scenarios, especially those associated with video data. The proposed algorithm first extracts corner points from the images as feature points by the Harris corner detector. The image gradients obtained as by-products from the corner detector are then classified into nine groups (represented by nine colors) according to their directions and magnitudes. Thus, local gradient directions in the vicinity of each feature point form a color code. As a non-parametric local transform, the color code summarizes local structure of image features. Finally, the point correspondences are obtained by comparing the color codes, and by utilizing some global relationships among feature points. The developed method not only achieves satisfactory efficiency but also resolves the ambiguous problems effectively.

The reminder of this chapter is structured as follows: in Section 5.2, related works are briefly reviewed, as an extension of Section 1.4. The extraction of feature points based on Harris corner detector is discussed in Section 5.3. The color code transform and its properties are also described. The correspondence determination algorithm using the color code is presented Section 5.4. Finally, experiments are presented in Section 5.5 and a summary is given in Section 5.6.

5.2 Related works

In general, feature extraction methods can be categorized into two classes: surrounding pixel-based and edge structure-based methods. The surrounding pixel-based approaches extract feature points by analyzing local gradients or curvatures with the colors and grey levels surrounding the pixel of interest [65][66][67][68]. The edge structure-based approaches obtain feature points using geometrical features such as edges, followed by finding specific geometric structures like maximum curvatures and/or intersections of edges [69][70][71]. The edge structure-based approaches generally have the disadvantages of high computational cost. In the rest of this section, discussions are mainly given for surrounding pixel-based approaches which motivates the corner detector-based algorithm presented in the next section.

Many works have been carried out on corner detection. As indicated in [60], corner detection should satisfy a number of important criteria, such as all the true corners should be detected and no false corners should be detected, etc. Brady et al. [67] proposed an approach known as SUSAN which uses a circular mask scanning the entire image. The basis for SUSAN is the concept that each image point is associated with it a local area of similar brightness, which contains much information about the geometric structure among surrounding image pixels. For each point in the image, a circular region centered at that point is considered. The number of pixels within the circular region having similar brightness to the center point then provides specific information to produce an edge strength image. Finally, moment calculations are applied to find edge directions followed by corner feature derivation.

In [66], Harris et al. extended Moravec's approach [65] and proposed a combined corner and edge detector based on a local auto-correlation function. The approach defines a measure of corner and edge quality or response, which is then used to select isolated corner pixels and to thin the edge pixels. The feature extraction algorithm of the proposed approach is based on Harris corner detector because of its efficiency and the consistency with the criteria stated in [60].

5.3 Extraction of corner features and color codes

We start this section by a review of Harris corner detector. Then, we continue with the development of the feature extraction algorithm based on the gradients computed by such a detector. As an improvement of Moravec's corner detector, Harris corner detector functions by considering a local window in the image, and determines the average changes of local image intensity due to small shifts of the window in various directions. Denoting the image intensities as *I*, the change *E* produced by a shift (*x*,*y*) is given by

$$E_{x,y} = \sum_{u,v} w_{u,v} \left| I_{x+u,y+v} - I_{u,v} \right|^2 \tag{36}$$

where *w* specifies the weighted image window. To reduce the sensitivity of image noise, Harris and Stephens define *w* as a Gaussian function, i.e., $w_{u,v} = e^{-(u^2+v^2)/2\sigma^2}$. According to some mathematical manipulations given in [65], the response in (36) can be approximately expressed in the following matrix form

$$E(x, y) = (x, y)M(x, y)^T \tag{37}$$

where *M* is a symmetric matrix

$$M = \begin{bmatrix} A & C \\ C & B \end{bmatrix}$$

with

$$A = \left(\frac{\partial I}{\partial x}\right)^2 w_{u,v}, \quad B = \left(\frac{\partial I}{\partial y}\right)^2 w_{u,v}, \quad C = \left(\frac{\partial I}{\partial x}\right)\left(\frac{\partial I}{\partial y}\right) w_{u,v}.$$

The gradients are computed using a convolution window. Figure 5.1 shows an image of indoor scene *I*, and gradient images $\partial I / \partial x$ and $\partial I / \partial y$, respectively.

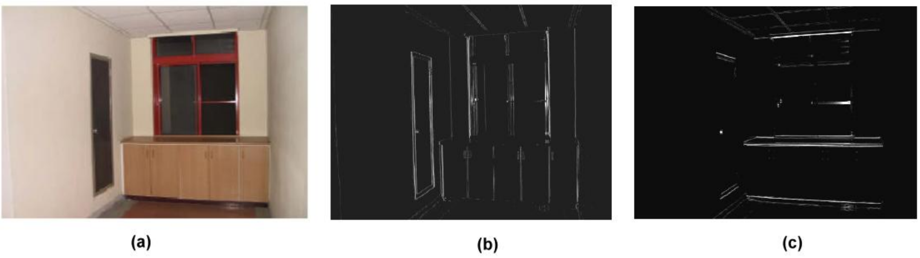


Figure 5.1 (a) An image of indoor scene *I*. (b) The image of $\partial I / \partial x$, and (c) the image of $\partial I / \partial y$.

With some analysis for the two eigenvalues, α and β, of *M*, Harris derived the following formulation for the corner response

$$R = Det(M) - k[Tr(M)]^2 \tag{38}$$

with

$$\begin{gathered} Det(M) = AB - C^2 = \alpha\beta \\ Tr(M) = A + B = \alpha + \beta \end{gathered} .$$

In particular, when *k*=0.04, (38) can be rewritten as

$$R = 0.92AB - C^2 - 0.04(A^2 + B^2) \tag{39}$$

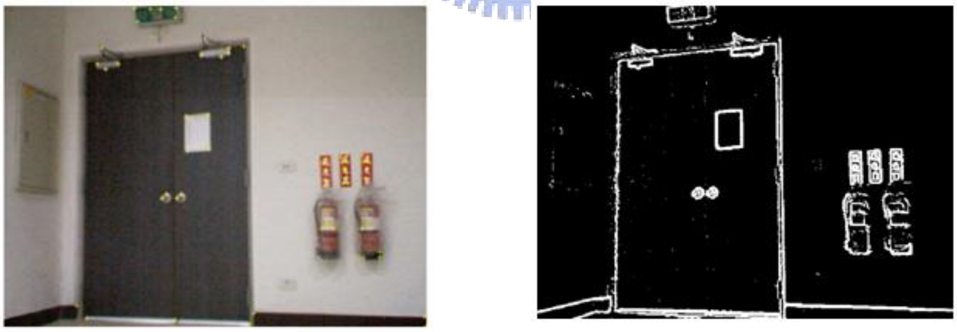
Often, it is very likely that corresponding pixels in two images taken from two closely located viewpoints will have very similar local gradients. The calculations of the image gradients in (37) motivate the idea in this chapter of using them as additional support in the correspondence determination of corner points. Note that no extra effort is needed in obtaining the gradients, which often requires the most time-consuming process. For each pixel of the image, we can form its gradient vector directly from (37) as

$$\nabla I = (X, Y) = (\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y}),$$

and obtain its cotangent magnitude $\cot\theta = \nabla I_x / \nabla I_y$. To obtain a reliable subset of the image pixels whose gradient is obvious and less noisy, the image pixels can be thresholded by

$$|\nabla I| > T \tag{40}$$

where T is a predefined threshold. In our experiments, T is defined as 10. Figure 5.2(a) shows an image of an indoor scene overlaid with the detected corners. The thresholded gradient image is shown in Figure 5.2(b).



(a) (b)

Figure 5.2 (a) Stereo images of an indoor scene overlaid with the detected corners, and (b) the thresholded gradient image of the image.

In this chapter, we make use of non-parametric local transforms as the addition support for finding correspondence of image features. Instead of exploiting local intensity values as [72], we use gradient directions in the transforms, which relay on the quantization of local gradient directions, rather than on the gradients themselves. Specifically, Figure 5.3 shows a quantization of the cotangent magnitudes of the gradient direction into eight sections, each covers 45°. For instances, the *white* cluster corresponds to $\nabla I_x > 10$ and $\left|\frac{\nabla I_x}{\nabla I_y}\right| > 3.2$, while the *cyan* cluster corresponds to $\nabla I_x + \nabla I_y > 14$ and $0.31 < \frac{\nabla I_x}{\nabla I_y} < \frac{1}{0.31}$ with $\nabla I_x > 0$.

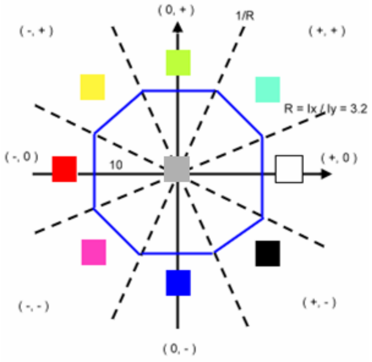


Figure 5.3 The color code representing eight clusters of directions.

With the quantized gradient directions, e.g., the representation using different colors, we now develop a way to determine the correspondence of feature points. The idea is to establish a word of color code by identifying local gradient directions for pixels surrounding each feature point obtained from Harris corner detector. As illustrated in Figure 5.4, a window containing nine 3×3 cells is defined with the feature point located at the center of the central cell. The representative color of each 3×3 cell is defined as the majority of the associated 3×3 colors of gradient directions. For the example shown in Figure 5.4, the resultant 8-color codeword for this particular feature (corner) point consists of blue, blue, grey, red, red, red, magenta and blue, clockwise starting from the upper right cell.

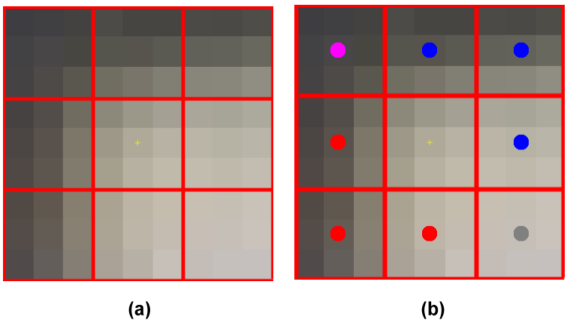


Figure 5.4 The color code transform of a detected corner point. (a) The nine 3×3 windows defined for the corner point, and (b) the colors which form a color code of the corner point.

An indoor scene, as shown in Figure 5.5, is used as an example for the above transform. First, corner points of the scene are obtained by Harris corner detector. The intensity gradients produced by the corner detector are then used to calculate the color code for each corner point. For the upper right corner of the door window in the left image, the associated color code is exactly the same as that in the right image, as shown in the two close-up 9×9 windows in Figure 5.5(a). For the lower right corner of the door window, on the other hand, there is a difference between a pair of the corresponding colors as shown in Figure 5.5(b). Under stable lighting conditions, the probability of having such differences is relatively low. However, the differences may still occur due to quantization noises associated with image acquisition, etc. Fortunately, the corner point shown in Figure 5.5(b) is likely to be matched correctly since there are only few unmatched colors. To improve the robustness of correspondence determination, consistency check and additional global constraints are also considered in the proposed approach, as discussed in the next section.

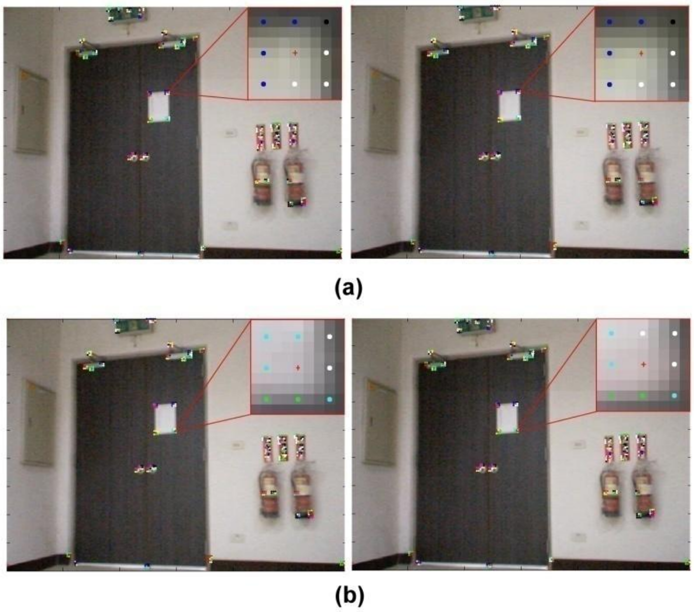


Figure 5.5 The color codes of corner points. (a) The two color codes are the same. (b) The two color codes are not the same.

5.4 Correspondence establishment using color code

In the paradigm of establishing correspondence of feature points in different images, several local methods as well as global methods are proposed, as summarized in [62]. Local methods are efficient but sensitive to locally ambiguous regions while global methods are less sensitive but computationally expensive. Therefore, it is plausible to exploit local and global constraints simultaneously when computing the correspondence for feature points of images. Figure 5.6 shows the block-diagram of the proposed correspondence determination approach which combines local matching, a consistency check, and iterative global matching. The basic idea of the approach to local matching is similar to that of the rank transform and the census transform proposed in [72]. The proposed color code transform is gradient-based, which captures the spatial structure of the region surrounding a corner point without additional cost. Moreover, a consistency check mechanism is adopted to remove incorrect local matching results, which are mostly introduced at object edges due to occlusions. The global matching uses constraints of distances and relative angular positions among feature points to establish additional correspondences. With the already matched feature point pairs utilized as geometrical references during each iteration, the correspondences of remaining feature points are established incrementally.

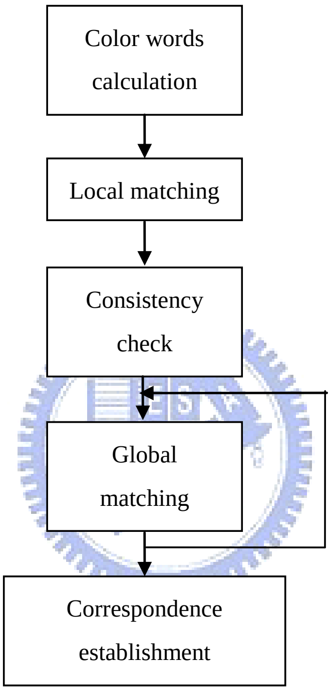


Figure 5.6 The proposed approach to correspondence establishment of corner points using color code.

Similar to the rank and census transforms, color code transform does not allow pixels from a small faction to contribute in a manner proportional to their intensity. However, the approach differs from rank and census transforms in that it does not rely heavily upon the intensity of the center pixel. Besides, it does not perform autocorrelation thus can achieve higher efficiency. To develop the approach to local matching using color codes, we first define two quantitative measurements. Let C_{diff} be the number of different color pairs between two color codes associated with two corner points, each in one of the two images under investigation, and let C_{dist} be the relative distance between a pair of colors. Define k as the quantized angular distance between the two non-gray colors.

For the *i*-th pair of colors, $1 \le i \le 8$, we have[7]

$$C_{dist}^{i} = \begin{cases} k, \text{ if } k \ge 2. \\ 0, \quad \text{if } k < 2 \text{ , or if at least one of the colors is gray.} \end{cases} \qquad (41)$$

For example, if the *j*-th pair of colors are black and green, respectively, then $C_{dist}^{j} = 3$. (Note that $C_{dist} \le 4$ is always true.) Thus, we obtain C_{diff} between two color codes as

$$C_{diff} = \sum_{i=1}^{8} C_{diff}^{i}$$

where

$$C_{diff}^{i} = 1, \text{ if } C_{dist}^{i} \ge 2 .$$

With the two quantitative measures defined above, we now describe the local matching algorithm to determine the correspondence of two corner points. Given two color codes associated with two corner points, each of them consists of 8 colors, if the two words are exactly the same (C_{diff} = 0), the two corner points are matched perfectly. Otherwise, the words are further investigated by thresholding C_{diff} to see whether the corner points match to each other. Figure 5.7 shows an example where one corner point is zoomed-in in two images to illustrate the two somewhat different color codes. One can see that there are only two pairs of different colors between the two color codes, i.e., cyan-white ($C_{dist}^{3} = 0$) and gray-cyan ($C_{dist}^{5} = 0$). Hence C_{diff} = 0 and the correspondence between the two corner points can be established.

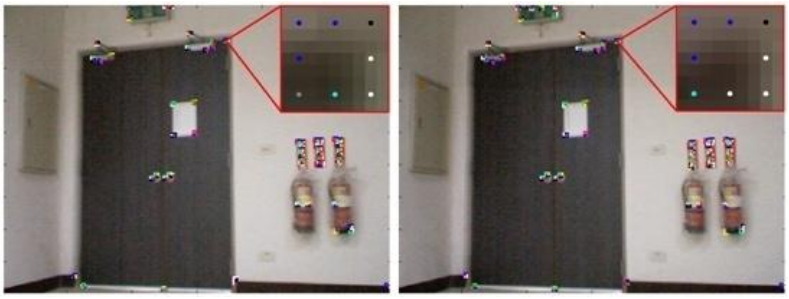


Figure 5.7 A pair of stereo images of an indoor scene overlaid with color codes of corner feature points.

[7] To improve the robustness of the computation a threshold of unity is chosen to suppress the noise in the calculation of gradient direction.

To further enhance the robustness of feature point correspondence, the proposed approach also addresses the problem of ambiguity due to occlusion and noise through a consistency check after the local matching of color codes. In Figure 5.8, the corner point *A* in the left image is matched with the corner point *A'* in the right image, and vise versa. This pair of feature points thus passes the check and satisfies the principle of symmetric correspondence [73]. On the other hand, corner point *B* in the left image matches to the corner point *C'* in the right image while *C'* matches to *C* in the left image. Hence, the correspondence between *B* and *C'* cannot be established since the symmetric correspondence property is violated. Nonetheless, *C* and *C'* do match to each other, so the correspondence is established. More results of performance improvement of feature point matching with the consistency check will be shown in the next section.

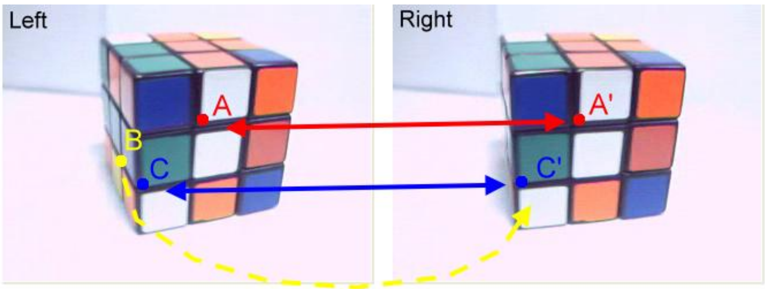


Figure 5.8 An example of consistency check.

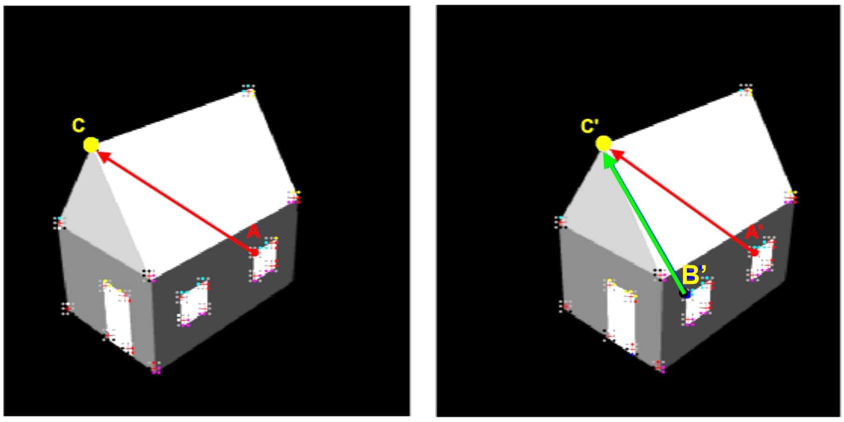


Figure 5.9 Determining the correspondence of point *A* using global constraints.

After performing the consistency check, some global constraints are used to further resolve the problem of ambiguity in the matching of feature points, e.g., due to similar local intensity gradients. The global constraints are based on geometrical relationships among corner points, which are assumed to change little between two images. The correspondences based on global constraints are established in an order such that correspondences with less ambiguity are determined first. The already established correspondences are thus used as geometrical references in subsequent processes. The global constraints utilized in the proposed approach include distances and relative angular positions among feature points. Figure 5.9 shows an example of stereo images where the two windows of a house have very similar local intensity gradients. In particular, it is easy to see that point *A* can be matched to both *A'* and *B'* in the right image if only local matching and the consistency check are performed. However, if *C* and *C'* are corner points whose correspondences have been established already, they can be used as the geometrical references to resolve the ambiguity. Consider vector $\overrightarrow{AC}$ and its length $\left|\overline{AC}\right|$ in the left image, and the same for vectors $\overrightarrow{A'C'}$, $\overrightarrow{B'C'}$ in the right image. It can be seen that $\left\|\overline{A'C'}-\overline{AC}\right\|$ is much less than $\left\|\overline{B'C'}-\overline{AC}\right\|$ and the angle between vectors $\overrightarrow{AC}$ and $\overrightarrow{A'C'}$ is much less than that between $\overrightarrow{AC}$ and $\overrightarrow{B'C'}$. These relative geometric relationships suggest the following formulation of global constraints for the proposed approach.

Assume that we want to determine the correspondence for a feature point *A*. Given a set of *I* already matched feature points, $\{(P_1, P_1'), \ldots, (P_I, P_I')\}$, and a set of *J* candidate points, $\{A_1', A_2', \ldots, A_J'\}$, for *A* to match. Let

$$D_j = \sum_{i=1}^{I} \left(\left|\overline{P_i A}\right| - \left|\overline{P'_i A'_j}\right| \right) \tag{42}$$

be the total length difference between $\overline{P_i A}$ and $\overline{P_i' A_j'}$, $1 \le i \le I$. In addition, we compute the sum of error in the cosine of relative angular positions,

$$C_j = \sum_{i=1}^{I} (1 - \cos\theta_i) = \sum_{i=1}^{I} \left[1 - \left(\frac{\overrightarrow{P_i A} \cdot \overrightarrow{P'_i A'_j}}{\left|\overline{P_i A}\right| \left|\overline{P'_i A'_j}\right|} \right) \right]. \tag{43}$$

Besides such global geometrical information, we also incorporate the differences in color codes between *A* and *A*$_j$,

$$CG_j = C_{diff}^{j} + C_{dist}^{j}. \tag{44}$$

Finally, the overall score of matching can be calculated as

$$G_j = \frac{D_j - D_{\min}}{D_{\max} + D_{\min}} + \frac{C_j - C_{\min}}{C_{\max} + C_{\min}} + \frac{CG_j - CG_{\min}}{CG_{\max} + CG_{\min}}. \tag{45}$$

for each A'_j, where D_{max}, D_{min}, C_{max}, C_{min}, CG_{max} and CG_{min} are maximum and minimum values of D_j, C_j, CG_j, respectively. It is easy to see that if a candidate for correspondence is similar to A in their color code patterns and also in their locations (and angular positions) relative to the earlier matched feature points, (45) will have a small value.

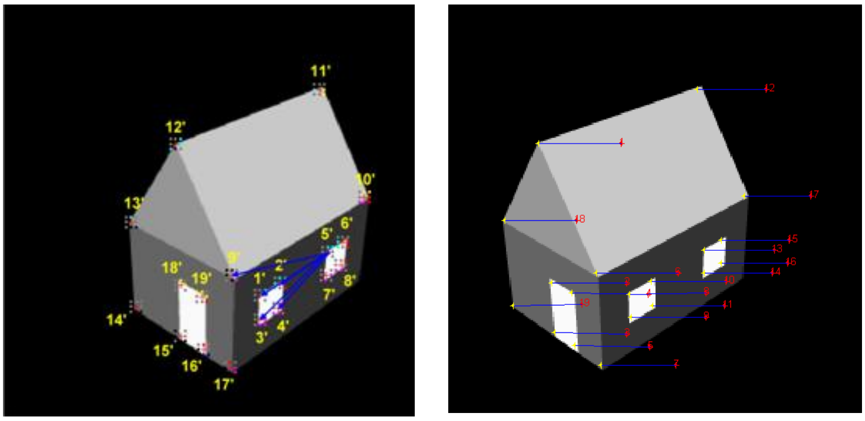
Based on the approach described above, comprehensive experiments for correspondence establishment can be carried out, as presented in the next section.

5.5 Experimental results

This section gives experimental results of the proposed approach. The algorithms are implemented with Matlab 6.5 running on a Pentium III 800Mhz machine under Microsoft Windows XP. The correspondence establishment results are presented first for some test images with readily observable ambiguities, and then for stereo images of real scenes. Some of the images used in the experiments are obtained from the CMU VASC image database [74].

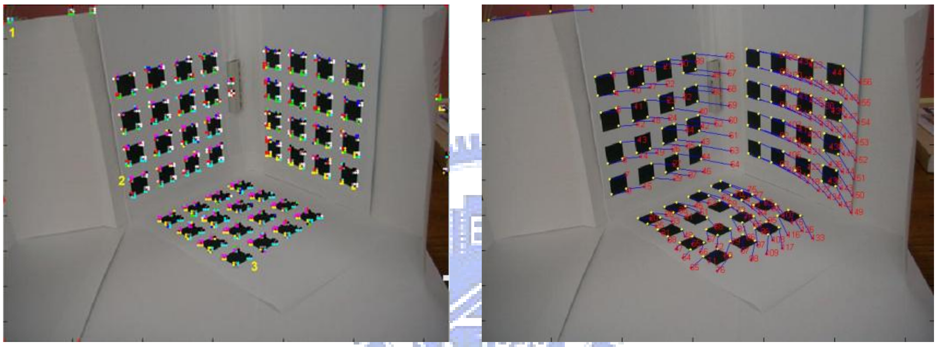
5.5.1 Correspondences for images with readily observable ambiguities

We first performed the experiments for stereo images with readily observable ambiguities. The synthesized images used in the first experiment, as shown in Figure 5.10, are the ‘house1’ images from the CMU VASC image database. The images are of size 250×250 and there is only horizontal translation of the camera. Figure 5.10(a) illustrates the global constraints used to determine the correspondence for point #5 whose vicinity has almost identical local intensity variation as point #1. Figure 5.10(b) presents the final result of the determined correspondences with that ambiguity resolved. In this experiment, all of the 19 correspondences are determined correctly. Figure 5.11 presents another experiment. There are many ambiguities since, as illustrated in Figure 5.11(a), many color codes are very similar. With the proposed approach, we obtain the final results presented in Figure 5.11(b) wherein only 2 of the 156 correspondences are determined incorrectly.



(a) (b)

Figure 5.10 Two synthesized images used in an experiment. (a) The global constraints used to assist the determination of the correspondence of a feature point. (b) The final result of the established correspondences.



(a) (b)

Figure 5.11 Two test images used in another experiment. (a) The color codes of the corner points. (b) The final result of the established correspondences.

5.5.2 Correspondences for stereo images of real scenes

Figure 5.12 shows a sample image obtained from a sequence of images of an indoor scene. The images was taken with Canon PowerShot G1 and the size was 640×450. A total of 36 correspondences are determined among 59 corner features in the left image and 53 corner features in the right image. Among the obtained results, only one correspondence is incorrect and the execution time is 3.23 seconds.

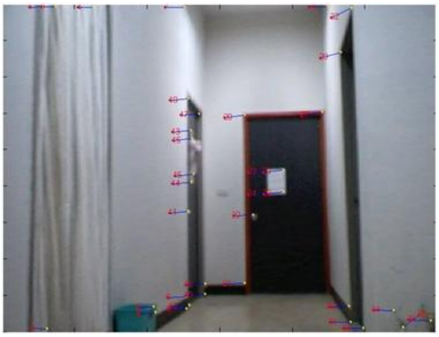
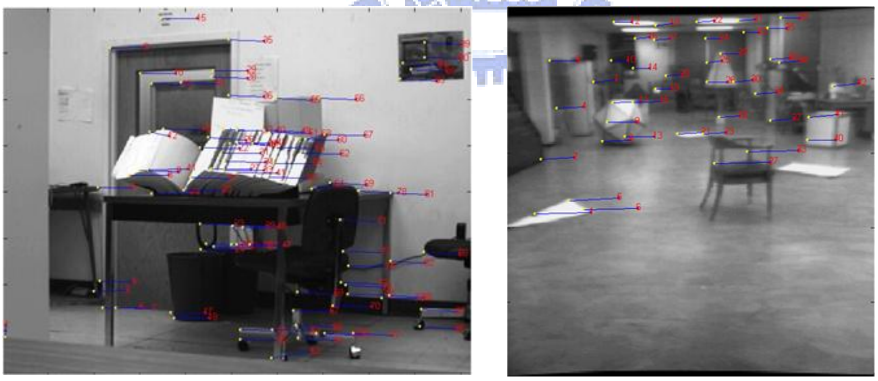


Figure 5.12 A sample image obtained from a sequence of images of an indoor scene. A total of 36 correspondences are determined, with only one of them being incorrect, with the proposed approach.

The next two experimental results are performed for images obtained from the VASC image database. Figure 5.13(a) shows a 512×400 image of a laboratory. A total of 90 correspondences are determined among 142 corner features in the left image and 156 corner features in the right image. Among the obtained correspondence, only four of them are incorrect. The execution time is 12.33 seconds. Figure 5.13(b) uses the ‘cart-alt’ images from the same image database where far-off objects as well as near-by objects are present. A total of 42 correspondences are determined among 83 corner features in the left image and 42 corner features in the right image. The obtained correspondences are all correct and the execution time is 12.5 seconds.



(a) (b)

Figure 5.13 Two samples of Image sequences used in two experiments both overlaid with the established point correspondences.

5.6 Summary

In this chapter, we propose an algorithm using local similarity and global constraint to obtain point correspondence. Image gradients obtained as by-product from corner detector are classified into nine groups, represented by nine distinct colors, according to their directions and magnitudes.

Local gradients in the vicinity of each feature point thus form a color code. The proposed approach obtains point correspondences by comparing these color codes followed by consistency check, and iteratively global matching using some spatial relationships among feature points. Experiments show that the proposed algorithm is not only efficient but also very robust for finding point correspondence among multiple images.

6. Conclusions

As a geometric invariant under projective transformations, cross-ratio is the basis of many recognition and reconstruction algorithms. In fact, cross-ratio-based approaches have major contributions to many important techniques which address various computer vision issues. In this dissertation, some applications of view-invariant cross-ratio are investigated. The common idea of these algorithms is to use cross-ratio to determine object structure without tedious and expensive computation to infer 3-D information, including object model and camera parameters. Meanwhile, for error analysis of cross-ratio, we derive efficient means to predict and to describe the characteristic of localization error. The approach enables one to select appropriate reference image points by efficiently providing approximate regions of localization error in advance, with no need to generate similar results for each configuration of reference points by using a large amount of data with synthetic noise.

In this dissertation, an efficient approach for finding correspondences between image features based on local similarity and global constraints is conducted as an applicable stage of image analysis, which is suitable for real-time applications. Several computer vision applications based on 1-D projective coordinates (cross-ratios) are proposed, which include (i) shadow generation of linear light source, and (ii) identity verification using facial images. According to simulation results, satisfactory results can be obtained for both applications. As for robot localization using 2-D projective coordinates, if multiple choices of reference points exist, the proposed error analysis technique can readily be applied to improve the localization accuracy.

References

[1] K. Kanatani, “Computational cross-ratio for computer vision,” *Computer Vision, Graphics, and Image Processing*, vol. 60, no. 3, pp. 371-381, 1994.

[2] J. L. Mundy and A. Zisserman, *Geometric invariance in computer vision*, MIT Press, Cambridge, MA, 1992.

[3] A. Shashua and N. Navab, ”Relative affine structure: theory and application to 3D reconstruction from perspective views,” *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 18, no. 9, pp. 873-883, 1996.

[4] N. Georgis, M. Petrou and J. Kittler, “Error guided design of a 3D vision system,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 4, pp. 366-379, 1998.

[5] D. Q. Huynh, “The cross-ratio: A revisit to its probability density function,” *British Machine Vision Conference,* vol. 1, pp. 262-271, Bristol, United Kingdom, 2000.

[6] A. Woo, P. Poulin, and A. Fournie, “A survey of shadow algorithms,” *IEEE Computer Graphics and Applications*, vol.10, no.6, pp. 13-32, 1990.

[7] H. Bao and Q. Peng, “Shadow models for linear and area light sources,” *Computers and Graphics*, vol.17, no.2, pp. 137-145, 1993.

[8] S. Gibson, J. Cook, T. Howard, and R. Hubbold, “Rapid shadow generation in real-world lighting environments,” *Proceesings of the 14th Eurographics workshop on Rendering*, pp. 219-229, Leuven, Belgium, 2003.

[9] A. Watt, *3D Computer* Graphics, Addison Wesley, 1993.

[10] N. Chin and S. Feiner, “Near real-time shadow generation using BSP trees,” *Computer Graphics*, vol.23, no.3, pp. 99-106, 1989.

[11] W. C. Thibault and B. F. Naylor, “Set operations on polyhedra using binary space partitioning trees,” *Computer Graphics* (*SIGGRAPH '87 Proceedings*), vol.21, no.4, pp. 153-162, 1987.

[12] P. Poulin, and J. Amanatides, “Shading and shadowing with linear light sources,” *Computers and Graphics*, vol.15, no.2, pp. 259-265, 1991.

[13] T. Tanaka and T. Takahashi, “Fast shadowing algorithm for linear light source,” *Proceedings of Eurographics '95*, vol.14, no.3, pp. 205-216, Dublin, Ireland, 1995.

[14] W. Heidrich, S. Brabec, and H.P. Seidel, “Soft shadow maps for linear lights,” *Proceedings of the 11th Eurographics Workshop on Rendering*, pp. 269-280, Brno, Czech Republic, 2000.

[15] M. F. Cohen and J. R. Wallace, *Radiosity and Realistic Image Synthesis*, Acacemic Press Professional, Boston, MA, 1993.

[16] F. Sillion and C. Puech, *Radiosity and Global Illumination*, Morgan Kaufmann, San Francisco, CA, 1994.

[17] L.-L. Wang and W.-H. Tsai, “Camera calibration by vanishing lines for 3-D computer vision,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol.13, no.4, pp. 370-376, 1991.

[18] B. Keating and N. Max, “Shadow penumbras for complex objects by depth-dependent filtering of multi-layer depth images,” *Proceedings of the 10th Eurographics Workshop on Rendering*, pp. 197-212, Granada, Spain, 1999.

[19] H. S. M. Coxeter, *Projective Geometry*, University of Toronto Press, 1974.

[20] P. Belhhumeur, J. Hespanha and D. Kriegman, “Eigenfaces vs. fisherfaces: Recognition using class specific linear projection,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 711-720, 1997.

[21] R. Chellappa, C. Wilson and S. Sirohey, “Human and machine recognition of faces: A survey,” *Proceedings of IEEE*, vol. 83, no. 5, pp. 705-741, 1995.

[22] A. Samal and P. Iyengar, “Automatic recognition and analysis of human faces and facial expressions: A Survey,” *Pattern Recognition*, vol. 25, pp. 65-77, 1992.

[23] J. Zhang, Y. Yan and M. Lades, “Face recognition: eigenface, elastic matching and neural nets,” *Proceedings of IEEE*, vol. 85, no. 9, pp. 1422-1435, 1997.

[24] H.A. Rowley, S. Baluja and T. Kanade, “Neural network-based face detection,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 1, pp. 23-38, 1998.

[25] G. Yang and T.-S. Huang, “Human face detection in a complex background,” *Pattern Recognition*, vol. 27, no. 1, pp. 53-63, 1994.

[26] S.-H. Jeng, H. Y. Mark Liao, C.-C. Han, M.-Y. Chern and Y.-T. Liu, “Facial feature detection using geometrical face model: an efficient approach,” *Pattern Recognition*, vol. 31, no. 3, pp. 273-282, 1998.

[27] D.J. Kriegman, M.H. Yang and N. Ahuja, ”Detecting faces in images: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 1, pp. 34-58, 2002.

[28] A.R. Mirhosseini and H. Yan, “Human face image recognition: An evidence aggregation approach,” *Computer Vision and Image Understanding*, vol. 71, no. 2, pp. 213-230, 1998.

[29] M. Lades, J.C. Vorbruggen, J. Buhmann, J. Lage, C. von der Malsburg, R.P. Wurtz and W. Konen, “Distortion invariant object recognition in the dynamic link architecture,” *IEEE Transactions on Computers*, vol. 42, no. 3, pp. 300-311, 1993.

[30] T. Kanade, “Picture processing system by computer complex and recognition of human faces,” *doctoral dissertation, Kyoto University*, 1973.

[31] P.J. Phillips, “Matching pursuit filters applied to face recognition,” *IEEE Transactions on Image Processing*, vol. 7, no. 8, pp. 1150-1164, 1998.

[32] M. Bichsel and A.P. Pentland, “Human Face recognition and the face image set's topology,” *Computer Vision, Graphics, and Image Processing: Image Understanding*, vol. 59, no. 2, pp. 254-261, 1994.

[33] S.-H. Lin, S.-Y. Kung and L.-J. Lin, “Face recognition/detection by probabilistic decision-basedneural network,” *IEEE Transactions on Neural Networks*, vol. 8, no. 1, pp. 114-132, 1997.

[34] H. Y. Mark Liao, C.-C. Han, G.-J. Yu, H.-R. Tyan, M.-C. Chen and L.-H. Chen, “Face recognition using a face-only database: A new approach,” *Proceedings of 3rd Asian Conference on Computer Vision*, LNCS, vol. 1352, pp. 742-749, Hong Kong, 1998.

[35] K. Hotta, “View-invariant face detection method based on local PCA cells,” *Proceedings of the 12th International Conference on Image Analysis and Processing*, pp. 57-62, Los Alamitos, CA, 2003.

[36] M. Turk and A. Pentland, “Eigenfaces for recognition,” *Journal of Cognitive Neuroscience*, vol. 3, no. 1, pp. 71-86, 1991.

[37] A. Pentland and M. Turk, “Face recognition using eigenfaces,” *Proceedings of CVPR*, pp. 586-591, Hawaii, 1991.

[38] J.B. Colombe, "A survey of recent developments in theoretical neuroscience and machine vision," *Proceedings of the 32nd Applied Imagery Pattern Recognition Workshop*, pp. 205-216, 2003.

[39] E.T. Rolls and S.M. Stringer, "Invariant object recognition in the visual system with error correction and temporal difference learning," *Network: Computation in Neural Systems*, vol. 12, no. 2, pp. 111-129, 2001.

[40] J.J. Atick, P.A. Griffin and N.A. Redlich, "Face recognition from live video," *Advance Imaging*, vol. 10, no. 5, pp. 58-62, 1995.

[41] Y. Yan and J. Zhang, "Rotation-invariant 3D reconstruction for face recognition," *Proceedings of IEEE Conference on Image Processing*, vol. 1, pp. 156-160, Chicago, IL, 1998.

[42] R. Lengagne, J.P. Tarel and O. Monga, "From 2D images to 3D face geometry," *Proceedings of IEEE Conference on Automatic Face and Gesture Recognition*, pp. 301-306, Vermont, USA, 1996.

[43] A. Eriksson and D. Weber, "Towards 3-dimensional face recognition," *Proceedings of IEEE Africon Conference*, vol. 1, pp. 401-406, Cape Town, South Africa, 1999.

[44] M. I. A. Lourakis, S. T. Halkidis and S. C. Orphanoudakis, "Matching disparate views of planar surfaces using projective invariants," *Image and Vision Computing*, vol. 18, pp. 673-683, 2000.

[45] S. Carlsson, "Projectively invariant decomposition and recognition of planar shapes," *International Journal of Computer Vision*, vol. 17, no. 2, pp. 193-209, 1996.

[46] T. Suk and J. Flusser, "Point-based projective invariants," *Pattern Recognition*, vol. 33, no. 2, pp. 251-261, 2000.

[47] J.-H. Chuang, J.-M. Chiu and Z. Chen, "Obtaining base edge correspondence in stereo images via quantitative measures along C-diagonals," *Pattern Recognition Letters*, vol. 18, no. 1, pp. 87-95, 1997.

[48] J.-M. Chiu, Z. Chen, J.-H. Chuang, and T.-L. Chia, "Determination of feature correspondence in stereo images using a calibration polygon," *Pattern Recognition*, vol. 30, no. 9, pp. 1387-1400, 1997.

[49] J.-H. Chuang, J.-H. Kao and Y.-H. Chen, "Identity verification by relative 3-D structure using multiple facial images," *Pattern Recognition Letters*, vol. 26, no. 9, pp. 1292-1303, 2005.

[50] W. Nunziati, S. Sclaroff and A.D. Bimbo, "An invariant representation for matching trajectories across uncalibrated video streams," *Proceedings of the 4th International Conference on Image and Video Retrieval*, LNCS 3568, pp. 318-327, Singapore, 2005.

[51] S. C. Rajashekahar and V. P. Namboodiri, "Image retrieval based on projective invariance," *Proceedings of International Conference on Image Processing*, pp. 405-408, Singapore, 2004.

[52] V. S. Tsonis, K. V. Chandrinos and P. E. Trahanias, "Landmark-based navigation using projective invariants," *Proceedings of 1998 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 342-347, Victoria, BC, Canada, 1998.

[53] K. Åström, "A correspondence problem in laser guided navigation," O. Eriksson and E. Bengtsson (Eds.), *Proceedings of Symposium on Image Analysis*, pp. 141-144, Uppsala, Sweden, 1992.

[54] K. Åström, "Automatic mapmaking," D. Charnley (Ed.), *Selected Papers from the 1st IFAC International Workshop on Intelligent Autonomous Vehicles*, pp. 181-186, Southampton, UK, 1993.

[55] R. Basri, E. Rivlin and I. Shimshoni, "Image-based robot navigation under the perspective model," *Proceedings of 1999 International Conference on Robotics and Automation*, pp. 2578-2583, Detroit, Michigan, 1999.

[56] J. J. Guerrero and C. Sagüés, "Uncalibrated vision based on lines for robot navigation," *Mechatronics*, vol. 11, no. 6, pp. 759-777, 2001.

[57] J.-S. Liu, J.-H. Chuang, "A geometry-based error estimation of cross-ratios," *Pattern Recognition*, vol.35, no.12, pp. 155-167, 2002.

[58] K. Åström and L. Morin, "Random cross-ratios," *Proceedings of the 9th Scandinavian Conference on Image Analysis*, vol. 2, pp. 1053-1060, Uppsala, Sweden, 1995.

[59] S. J. Maybank, "Stochastic properties of the cross-ratio," *Pattern Recognition Letters*, vol. 17, no. 3, pp. 211-217, 1996.

[60] F. Mokhtarian and R. Suomela, "Robust image corner detection through curvature scale space," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 12, pp. 1376–1381, 1998.

[61] Z. Zhang, R. Deriche, O. Faugeras, and O.T. Luong, "A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry," *Research Report No. 2273*, INRIA Sophia-Antipolis, 1994.

[62] M.Z. Brown, D. Burschka, and G.D. Hager, "Advances in computational stereo," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 8, pp. 993-1008, 2003.

[63] D. Lowe, "Distinctive Image Features From Scale-Invariant Keypoints," *Internal Journal of Computer Vision*, vol. 60, no. 2, pp. 91-110, 2004.

[64] J.-H. Chuang, L.-W. Kuo, H.-J. Kuo, and J.-S. Liu, "Shadow Generation from Stereo Images," *Electronics Letters*, vol.36, no.8, pp. 720-722, 2000.

[65] H.P. Moravec, "Toward automatic visual obstacle avoidance," *Proceedings of the 5th International Joint Conference on Artificial Intelligence*, pp. 584, 1977.

[66] C. Harris and M. Stephens, "A combined corner and edge detector," *Proceedings of the 4th Alvery Vision Conference*, pp. 147-151, Manchester, 1988.

[67] S.M. Smith and J.M. Brady, "SUSAN: A new approach to low level image processing," *International Journal of Computer Vision*, vol. 23, no. 1, pp. 45-78, 1997.

[68] J. Weng, N. Ahuja and T.S. Huang, "Matching two perspective views," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 8, pp. 806-825, 1992.

[69] H. Asada and M. Brady, "The curvature primal sketch," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, pp. 2-14, 1986.

[70] R. Deriche and G. Giraudon, "2-D curve matching using high curvature points : Application to stereo vision," *Pattern Recognition*, pp. 240-242, 1990.

[71] R. Horaud, F. Veillon, and T. Skordas, "Finding geometric and relational structures in an image," *Proceedings of the First European Conference on Computer Vision*, pp. 374-384, Antibes, France, 1990

[72] R. Zabih and J. Woodfill, "Non-parametric local transforms for computing visual correspondence," *Proceedings of the Third European Conference on Computer Vision*, pp. 150-158, Stockholm, Sweden, 1994.

[73] O. Faugeras, B. Hotz, H. Mathieu, T. Vieville, Z. Zhang,P. Eua, E. Theron, L. Moll, G. Berry, J. Vuillemin, P. Bertin, and C. Proy, “Real-time correlation-based stereo: algorithm, implementation and applications,” *Technical Report 2013*, INRIA Sophia-Antipolis, 1993.

[74] CMU VASC Image Database, http://vasc.ri.cmu.edu/idb/.