WEAK CALIBRATION AND IMAGE-BASED RENDERING ALGORITHMS

BY

YAKUP GENC

B.S., Technical University of Istanbul, 1991
M.S., University of Illinois at Urbana-Champaign, 1995

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Computer Science
in the Graduate College of the
University of Illinois at Urbana-Champaign, 1999

Urbana, Illinois

WEAK CALIBRATION AND IMAGE-BASED RENDERING ALGORITHMS

Yakup Genc, Ph.D.
Department of Computer Science
University of Illinois at Urbana-Champaign, 1999
Jean Ponce, Advisor

This thesis introduces two novel techniques for the analysis and synthesis of image sequences: a linear algorithm for weak calibration of a stereo rig from point correspondences, and an algorithm for image-based rendering without explicit three-dimensional reconstruction based on point and line correspondences.

By recasting the epipolar constraint in a projective setting with an appropriate basis choice, we first show that Jepson's and Heeger's linear subspace algorithm for infinitesimal motion estimation can be generalized to the finite motion case. This yields a linear method for weak calibration. The algorithm has been implemented and tested on both real and synthetic images, and it is compared to other linear and non-linear approaches to weak calibration.

We then show that the set of all images of a rigid scene taken by a Euclidean camera is a six-dimensional variety, and we introduce a parameterization (called parameterized image variety, or PIV in short) of this variety for weak perspective and paraperspective cameras in terms of the image positions of three reference points. This parameterization can be estimated via linear least-squares and non-linear least-squares with low-degree equations.

We use parameterized image varieties of both point and line features to synthesize new images from a set of pre-recorded pictures without actual three-dimensional reconstruction (image-based rendering) in an integrated framework. The method has been implemented and extensively tested on real data sets.

Finally, we show how to adapt recent advances in statistically-unbiased least-squares methods to our image-based rendering approach. The point-based PIV involves equations with

bilinear or higher-order data dependencies and we show how to efficiently estimate its parameters by adapting Leedan's and Meer's technique for bilinear estimation problems.

ABSTRACT

This thesis introduces two novel techniques for the analysis and synthesis of image sequences: a linear algorithm for weak calibration of a stereo rig from point correspondences, and an algorithm for image-based rendering without explicit three-dimensional reconstruction based on point and line correspondences.

By recasting the epipolar constraint in a projective setting with an appropriate basis choice, we first show that Jepson's and Heeger's linear subspace algorithm for infinitesimal motion estimation can be generalized to the finite motion case. This yields a linear method for weak calibration. The algorithm has been implemented and tested on both real and synthetic images, and it is compared to other linear and non-linear approaches to weak calibration.

We then show that the set of all images of a rigid scene taken by a Euclidean camera is a six-dimensional variety, and we introduce a parameterization (called parameterized image variety, or PIV in short) of this variety for weak perspective and paraperspective cameras in terms of the image positions of three reference points. This parameterization can be estimated via linear least-squares and non-linear least-squares with low-degree equations.

We use parameterized image varieties of both point and line features to synthesize new images from a set of pre-recorded pictures without actual three-dimensional reconstruction (image-based rendering) in an integrated framework. The method has been implemented and extensively tested on real data sets.

Finally, we show how to adapt recent advances in statistically-unbiased least-squares methods to our image-based rendering approach. The point-based PIV involves equations with bilinear or higher-order data dependencies and we show how to efficiently estimate its parameters by adapting Leedan's and Meer's technique for bilinear estimation problems.

Decidated to my parents, Elmas and Osman Genc

ACKNOWLEDGMENTS

I would like to thank my advisor, Professor Jean Ponce, for his guidance and support throughout my graduate studies at the University of Illinois. It has been a pleasure to work with him. I also would like to thank to the other members of my thesis committee, Professors Seth Hutchinson, David Kriegman and Dan Roth for their input and time.

Thanks also extend to Professor Peter Meer for his collaboration and contributions, and to my colleagues in the Computer Vision and Robotics research group for their helpful discussions, including (but not limited to) Kevin Knickels, Peter Leven, Sung-il Pae and Attawith Sudsang.

Finally, I wish to thank my wife, Emine Genc, for her endless patience and support, and my daughter, Hilal, and my son, Ibrahim Said, for the joy they bring to my life.

This research was partially supported by a fellowship from the Turkish Ministry of Education, by the National Science Foundation under grants IRI-9224815 and IRI-9634312 and by the National Aeronautics and Space Administration under grant NAG 1-613.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Figure **Page**

CHAPTER 1

INTRODUCTION

The human visual system processes a continuous stream of images to deduce information about the world. Computer vision tries to imitate these capabilities by processing sequences of images obtained by a digital camera.

In this context, two images of a rigid scene are related by the *epipolar geometry.* The estimation of the epipolar geometry from point correspondences is called *weak calibration.* The synthesis of new views of a scene from a set of pre-recorded pictures is called *image-based rendering.* In this thesis, we investigate these two problems and develop algorithms to solve them.

It has been shown by Jepson and Heeger in [43, 50] that motion estimation for calibrated cameras undergoing infinitesimal displacements can be reduced to a linear problem. A primary objective of this thesis was to answer the following question: is it possible to generalize Jepson's and Heeger's method to the finite motion case? As we will see in Chapter 3, the answer to this question is yes when the motion analysis problem is considered in a projective setting. In this case, motion estimation reduces to the estimation of the epipolar geometry, a process known as weak calibration.

More precisely, two uncalibrated perspective images of a rigid object are related by the epipolar geometry (or equivalently by the *fundamental matrix* which is a 3×3 singular matrix) [21]. When the cameras are calibrated, i.e., the intrinsic parameters of the cameras are known, this relation is captured by the *essential matrix* [63]. The image of a scene point in one image lies on the corresponding epipolar line. The epipolar lines in one image go through the epipole, i.e., the intersection of the image plane with the baseline connecting the centers of the two camera. The epipolar line associated with a scene point is the intersection of the image plane and the

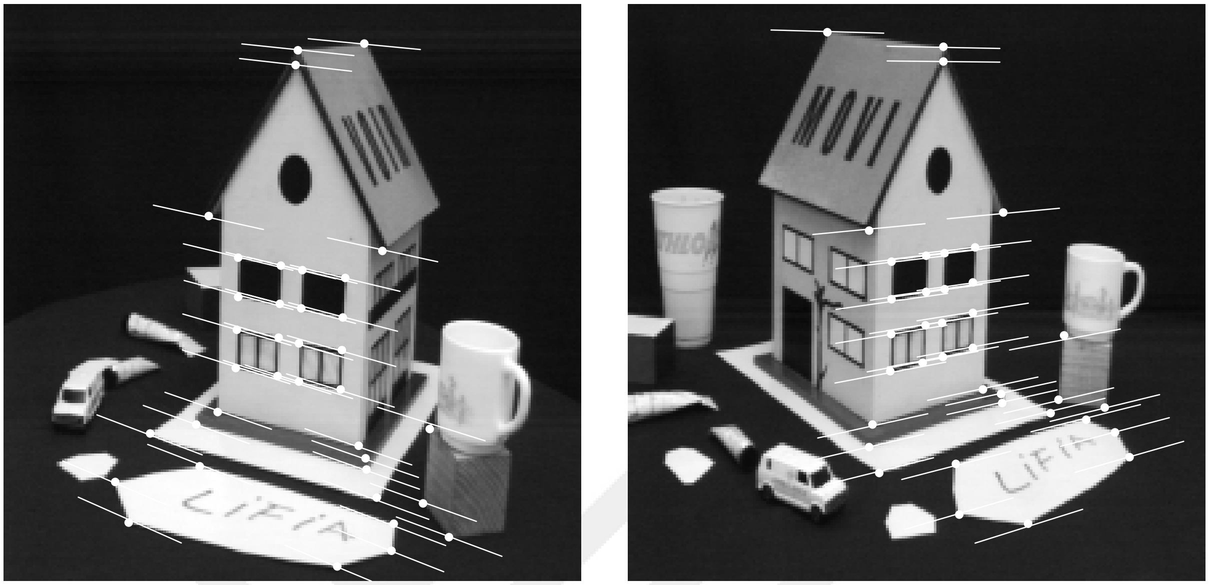


Figure 1.1 Recovered epipolar lines for a pair of images.

plane spanned by the point and the centers of two cameras. See Figure 1.1 for an example of recovered epipolar lines for a pair of images using the algorithm introduced in Chapter 3.

Developing accurate and reliable methods for weak calibration is very important in practice because of the central role of epipolar geometry in stereo and motion algorithms: in particular, Faugeras [19] and Hartley, Gupta and Chang [39] have shown that it is possible to reconstruct the three-dimensional projective structure of a scene observed by a weakly calibrated stereo rig without any knowledge of the intrinsic parameters of the cameras (see [55, 75, 99, 88] for related work). As demonstrated by Xu *et al.* [117] [80] and Deriche *et al.* [16] [120], estimating the epipolar geometry from initial point correspondences is also the key to establishing further correspondences in image matching. Luong and Faugeras [68] discuss many other applications, including the construction of stereo projective invariants [31, 34, 35, 99], the synthesis of new images from old ones [57, 95], convex hull computation [91], image rectification [38, 94], motion segmentation [80, 110], and self calibration [21, 36].

We propose in Chapter 3 a linear algorithm for weak calibration that generalizes Jepson's and Heeger's method to the finite motion case. This algorithm has been implemented and comparisons with other techniques are presented.

Second, we consider the problem of image-based rendering from point and line correspondences. This class of techniques operates solely on a fixed set of pre-recorded pictures of a rigid scene to synthesize new views of the scene without actually constructing a three-dimensional model, offering an alternative to the classical model-based rendering methods.

In recent years, there has been a proliferation of research on image-based rendering in both computer vision [57, 95, 56, 3] and computer graphics [9, 30, 61]. The computer graphics techniques are based on the idea that the set of all visual rays (light field) is four-dimensional, and can thus be characterized from a two-dimensional sample of images of a rigid scene. In contrast, the computer vision techniques only use a discrete set of views among which point correspondences have been established by feature tracking or conventional stereo matching.

Unfortunately, previous image-based rendering techniques based on ideas from affine or projective geometry generate pictures that may be affinely or projectively deformed. Another primary objective of this thesis was to determine whether it was possible to construct correct images without estimating the three-dimensional structure of the scene or the motion of the camera. We have indeed developed such an algorithm presented in Chapter 5 that takes into account Euclidean constraints directly.

A desirable property that an image-based rendering system should have is that even a restricted range of motion in the input should be sufficient to render correct images for far apart views. For example, the image-based rendering system developed in this thesis can generate a view from the top of a building even though only small set of frontal images of the building are available (see Figure 1.2 for an example).

1.1 Thesis Organization

In Chapter 2, the main elements of epipolar geometry are summarized and various approaches to weak calibration are reviewed. This chapter also clarifies the relationship between Jepson's and Heeger's linear subspace approach [50, 43, 51] to infinitesimal motion analysis and the Longuet-Higgins characterization of epipolar geometry.

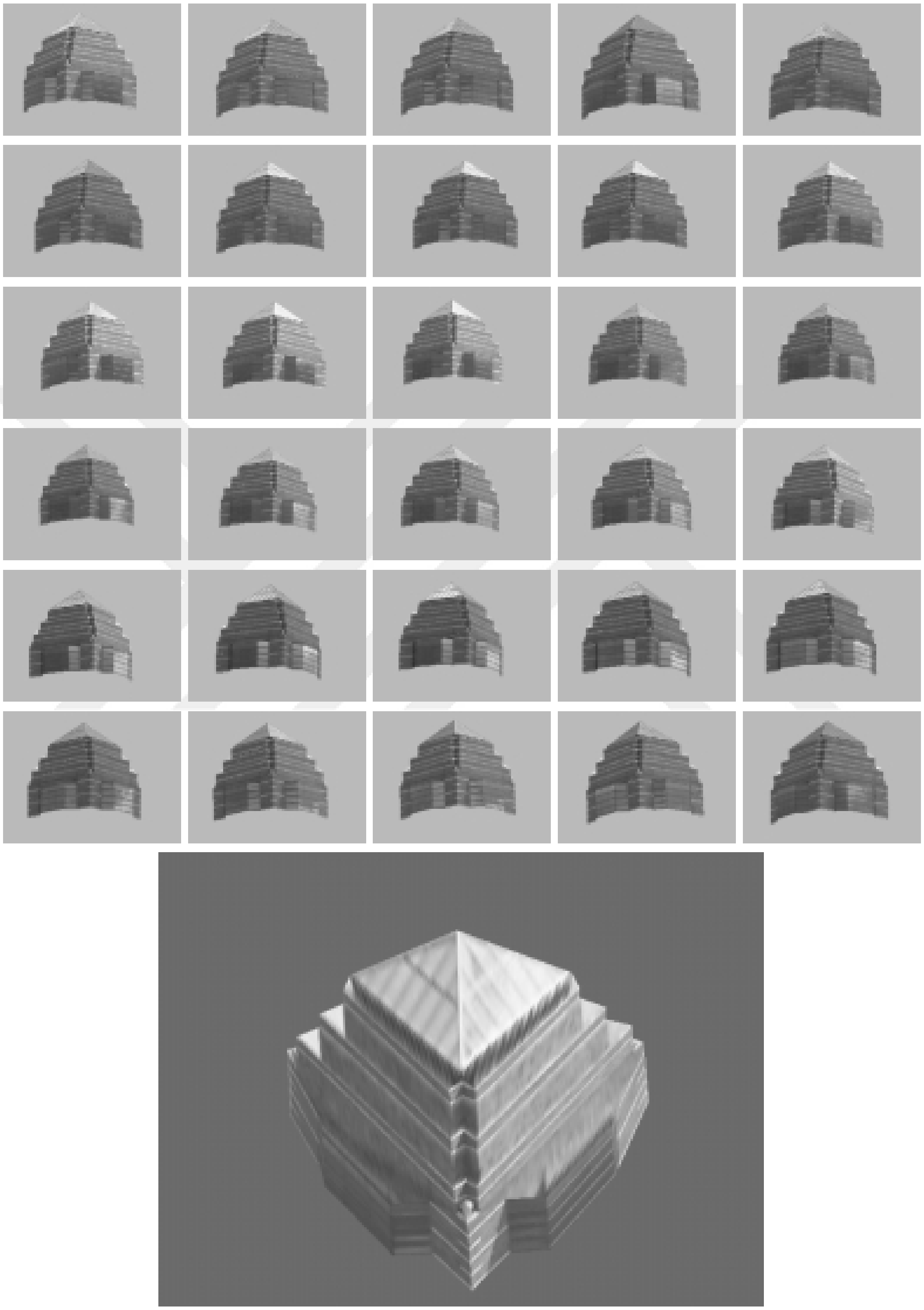


Figure 1.2 Example image synthesis: 30 images (top 6 rows) are used to obtain the synthesized image for a novel view (larger image).

In Chapter 3, we introduce a linear algorithm for estimating the epipolar geometry. We first show that Jepson's and Heeger's linear subspace technique for infinitesimal motion estimation can be generalized to the finite motion case by choosing an appropriate basis for projective space. This yields a linear method for weak calibration. The proposed algorithm has been implemented and tested on both real and synthetic images, and it is compared to other linear and non-linear approaches to weak calibration.

In Chapter 4, we review various image-based rendering techniques. In Chapter 5, after discussing different projection models and their properties, we introduce a new approach to image-based rendering taking into account the Euclidean constraints associated with real cameras. We assume that a set of point correspondences are given throughout the sequence. The method is based on a new parameterization of the set of all weak perspective or paraperspective images of a rigid scene. The method has been implemented and tested on real data. We also present an analysis of degenerate camera and scene configurations for which our algorithm may fail. We then generalize the point-based approach to handle line correspondences. Note that using lines has the advantage that they can be located very accurately in images. In image-based rendering, the effective use of line correspondences requires the prediction of the end points of the line segments. We propose a method for solving this problem and integrating point and line features to render new images of the scene in a common framework.

In Chapter 6, we investigate the use of statistically-unbiased least-squares methods in image-based rendering applications. In particular, the estimation of the parameters of the PIV involves solving a set of equations with bilinear or higher-order data dependencies. The approach proposed by Leedan and Meer [59, 60] for bilinear data dependency problems is adapted to the estimation of the PIV parameters for the weak perspective and paraperspective cases.

Finally, we conclude in Chapter 7 by summarizing the results of this thesis and giving some future research directions.

Part of the material in this thesis has been published in a number of papers: The weak calibration method in Chapter 3 was first presented in [86] and [87]. The point-based image-based rendering algorithm in Chapter 5 was presented in [26]. Finally, the bilinear estimation algorithm of Chapter 6 was presented in [27].

CHAPTER 2

WEAK CALIBRATION

The geometric information contained in two images taken by uncalibrated perspective cameras is completely captured by the *epipolar geometry* of the two images, i.e., by the knowledge of the *epipoles* and of the homography, called *epipolar transformation*, relating the pencils of *epipolar lines* in the two images, or equivalently, by the knowledge of the *fundamental matrix* [21, 39, 33, 64, 68]. This chapter reviews current approaches to the problem of estimating the epipolar geometry from point correspondences, a process known as *weak calibration*.

Developing accurate and reliable methods for weak calibration is very important in practice because of the central role of epipolar geometry in stereo and motion algorithms: in particular, Faugeras [19] and Hartley, Gupta and Chang [39] have shown that it is possible to reconstruct the three-dimensional projective structure of a scene observed by a weakly calibrated stereo rig without any knowledge of the intrinsic parameters of the cameras (see [55, 75, 99, 88] for related work). As demonstrated by Xu *et al.* [117] [80] and Deriche *et al.* [16] [120], estimating the epipolar geometry from initial point correspondences is also the key to establishing further correspondences in image matching. Luong and Faugeras [68] discuss many other applications, including the construction of stereo projective invariants [31, 34, 35, 99], the synthesis of new images from old ones [57, 95], convex hull computation [91], image rectification [38, 94], motion segmentation [80, 110], and self calibration [21, 36].

We will assume throughout the presentation that a static scene is observed by a mobile perspective camera. In the infinitesimal case we assume that the motion field is known. In the finite-motion case we suppose that discrete point correspondences used as input to the weak calibration process have been correctly established by a separate matching process. Let us just mention that there is a vast literature on image matching, including various methods for detecting feature points [76, 18, 32, 81, 92, 15], hypothesizing initial correspondences (e.g., correlation [28, 79, 24], feature tracking [75, 108, 118] and structural matching [4, 98, 45, 115]), establishing new correspondences once an initial estimate of the epipolar geometry has been obtained [117, 80], and finding and rejecting false matches using techniques from robust statistics [83, 97, 110, 109]. Two recent papers by Deriche *et al.* describe an integrated approach to image matching that combines several of these ideas and achieves impressive results [16, 120].

The rest of this chapter is organized as follows. Section 2.1 summarizes the main elements of epipolar geometry. Section 2.2 presents previous approaches to weak calibration, including the non-linear distance minimization technique of Luong *et al.* [65, 68], Hartley's normalized eight-point algorithm [37], and the virtual parallax approach of Boufama and Mohr [6]. Section 2.3 clarifies the relationship between Jepson's and Heeger's linear subspace approach to infinitesimal motion analysis [50, 43, 51] and the Longuet-Higgins characterization of epipolar geometry [63].

2.1 Epipolar Geometry

Consider two perspective cameras with optical centers $\boldsymbol{C}$ and $\boldsymbol{C}'$ and image planes Π and Π' (Figure Figure 2.1). If $\boldsymbol{P}$ is a point observed by the cameras and $\boldsymbol{p}, \boldsymbol{p}'$ denote its two images, then the point $\boldsymbol{p}$ (resp. $\boldsymbol{p}'$) lies on the line where the plane formed by $\boldsymbol{C}$, $\boldsymbol{C}'$ and $\boldsymbol{P}$ intersects the image plane Π (resp. Π'). This line goes through the point $\boldsymbol{e}$ (resp. $\boldsymbol{e}'$) where the baseline joining the optical centers $\boldsymbol{C}$ and $\boldsymbol{C}'$ intersects the plane Π (resp. Π'). The points $\boldsymbol{e}$ and $\boldsymbol{e}'$ are called the *epipoles* associated with the two cameras, and the line passing through $\boldsymbol{e}'$ and defined by the intersection of the plane formed by the points $\boldsymbol{C}$, $\boldsymbol{C}'$, and $\boldsymbol{p}$ is called the *epipolar line* associated with the point $\boldsymbol{p}$.

This yields the well-known epipolar constraint: "if $\boldsymbol{p}$ and $\boldsymbol{p}'$ are images of the same point, then $\boldsymbol{p}'$ lies on the epipolar line associated with $\boldsymbol{p}$", which is of course of fundamental importance in stereo vision.

Longuet-Higgins [63] was the first person to give a simple algebraic characterization of the epipolar constraint, namely

$$\boldsymbol{p}^{'T}\mathcal{E}\boldsymbol{p} = 0, \tag{2.1}$$

where $\mathcal{E} = [\boldsymbol{t}_\times]\mathcal{R}$, and the point $\boldsymbol{p}$ (resp. $\boldsymbol{p}'$) has been identified with the vector of its homogeneous image coordinates $(u, v, 1)^T$ (resp. $(u', v', 1)^T$). Here $[\boldsymbol{a}_\times]$ denotes the skew-symmetric

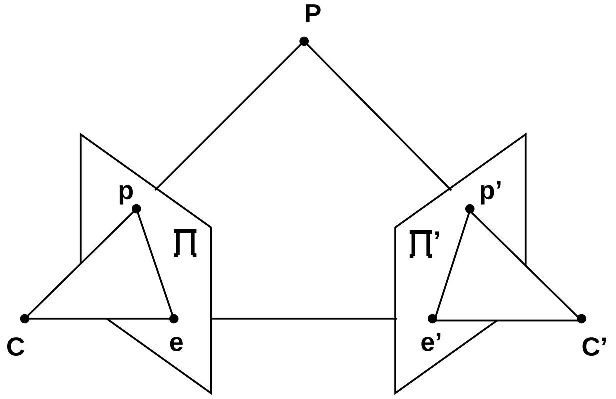


Figure 2.1 Epipolar geometry.

matrix such that $[\boldsymbol{a}_\times]\boldsymbol{x} = \boldsymbol{a} \times \boldsymbol{x}$ is the cross-product of the vectors $\boldsymbol{a}$ and $\boldsymbol{x}$. The matrix $\mathcal{E}$ is called the *essential matrix* representing the motion from the first to the second camera defined by the rotation $\mathcal{R}$ and the translation $\boldsymbol{t}$. As shown by Huang and Faugeras, essential matrices are characterized by the fact that they are singular with two equal non-zero singular values [47].

The Longuet-Higgins relation holds for *calibrated* cameras, i.e., cameras whose internal parameters are known so that image positions can be expressed in *normalized* coordinates. As shown by Faugeras, Luong and Maybank [21, 64, 68], it is still possible to write the epipolar constraint for *uncalibrated* cameras as

$$\boldsymbol{p}^{'T}\mathcal{F}\boldsymbol{p} = 0, \tag{2.2}$$

where the matrix $\mathcal{F}$, called the *fundamental matrix*, is not, in general, an essential matrix. It has rank two, and the eigenvector of $\mathcal{F}$ (resp. $\mathcal{F}^T$) corresponding to its zero eigenvalue is the epipole $\boldsymbol{e}$ (resp. $\boldsymbol{e}'$). Note that $\mathcal{F}\boldsymbol{p}$ (resp. $\mathcal{F}^T\boldsymbol{p}'$) represents the epipolar line corresponding to the point $\boldsymbol{p}$ (resp. $\boldsymbol{p}'$) in the second (resp. first) image.

Alternatively, the epipolar geometry is also determined by the position of the epipoles $\boldsymbol{e}$ and $\boldsymbol{e}'$ and by the *epipolar transformation*: as shown in [21, 65, 68], the correspondence between epipolar lines is captured by a homography, i.e., if the epipolar line l with slope τ in the first image matches the epipolar line l' with slope τ' in the second image, then

$$\tau \rightarrow \tau' = \frac{a\tau + b}{c\tau + d} \quad \text{with} \quad \tau = \frac{v - \beta}{u - \alpha} \quad \text{and} \quad \tau' = \frac{v' - \beta'}{u' - \alpha'}, \tag{2.3}$$

where a, b, c, d are the coefficients of the homography, $\boldsymbol{p} = (u, v)^T$ and $\boldsymbol{p}' = (u', v')^T$ are corresponding points, and $\boldsymbol{e} = (\alpha, \beta)^T$ and $\boldsymbol{e}' = (\alpha', \beta')^T$ are the epipoles. This homography is the epipolar transformation, and as shown in [21, 65, 68], the coefficients of the fundamental matrix can be computed from the positions of the epipoles $\boldsymbol{e}$ and $\boldsymbol{e}'$ and a, b, c, d, and vice versa. In particular,

$$\mathcal{F} = \begin{pmatrix} b & a & -a\beta - b\alpha \\ -d & -c & c\beta + d\alpha \\ d\beta' - b\alpha' & c\beta' - a\alpha' & -c\beta\beta' - d\beta'\alpha + a\beta\alpha' + b\alpha\alpha' \end{pmatrix}. \tag{2.4}$$

2.2 Current Approaches to Weak Calibration

As mentioned by Faugeras [20], the problem of estimating the epipoles and the epipolar transformations compatible with seven point correspondences was first posed by Chasles [8] and solved by Hesse [44] (see the article by Sturm [103] for an analysis of Hesse's method and the paper by Faugeras and Maybank [72] for a modern account). Hesse's approach is mostly of theoretical interest since it only exploits the minimum number of point correspondences necessary to estimate the epipolar geometry and is thus unable to deal with noise. More robust approaches to weak calibration from a large number of point correspondences have been proposed recently in the computer vision community: Luong *et al.* [65, 68] have proposed various linear and non-linear least-squares methods for estimating the fundamental matrix, which captures the epipolar geometry in algebraic form. In particular, they have shown that, although Longuet-Higgins' eight-point algorithm [63] generalizes to the uncalibrated case and can be used to estimate the fundamental matrix via linear least squares, its performance is in general poor in the presence of noise. This has prompted Luong *et al.* to propose an iterative non-linear algorithm that minimizes the distance between image points and the corresponding epipolar lines. The reliability and accuracy of this technique have been established through extensive experimentation in [65, 68].

Recently, Hartley [37] has shown that the poor characteristics of the eight-point method can be traced to the fact that the corresponding matrices are ill-conditioned, so that adding a simple preprocessing step (translating the data so it is centered at the origin, then scaling it so the average distance to the origin is $\sqrt{2}$ pixel) improves the algorithm's performance to a level comparable to the one achieved through non-linear minimization.

Following an original idea by Couapel [12], Boufama and Mohr [6] have proposed a different linear algorithm, based on the notion of virtual parallax, to compute one of the epipoles and the homography between a plane and the images (see the papers by Demey, Zisserman and Beardsley [14] and Luong and Faugeras [66, 68] for related approaches assuming that groups of coplanar points have been identified). They have shown that this is sufficient to completely determine the epipolar geometry and have also reported good results.

The two linear approaches mentioned so far use a redundant set of parameters: the eight-point algorithm does not take into account the rank-two property of fundamental matrices, thus it estimates eight parameters instead of the seven coefficients that define the epipolar geometry, while the virtual parallax algorithm linearizes the estimation process by introducing another extraneous variable. In contrast, in Chapter 3 we propose an approach which estimates the epipolar geometry through singular value decomposition [116], yet only involves the variables of interest, namely the position of the epipoles and the epipolar transformation. The core of this approach is the computation of the epipoles, which is based on the choice of an appropriate basis for projective space and on the construction of two sets of linear constraints corresponding to certain combinations of the original epipolar constraints. This construction has been inspired by Jepson's and Heeger's linear subspace method for infinitesimal motion estimation [50, 43, 51], and it is related to the linearized weak calibration method of Lawn and Cipolla [58].

So far, we have assumed that point correspondences are available. There are two types of possible errors (outliers) in these matches, i.e., noise in feature location and false matches. The above methods and [59] can handle the noise in the feature location to some degree. However, when there are false matches, other robust methods are needed such as [109].

We will propose in Chapter 3 a new method for weak calibration and compare it to other approaches [65, 68, 37, 6]. We briefly describe these here for completeness (see [119] for a review of other techniques that are not covered here).

2.2.1 Linear Methods

Note that the epipolar constraint (2.2) is a linear equation in the nine coefficients of the fundamental matrix $\mathcal{F}$. Since (2.2) is homogeneous in the coefficients of $\mathcal{F}$, it follows that observing eight point correspondences is in principle sufficient to estimate the fundamental matrix. When $n \geq 8$ correspondences are available, $\mathcal{F}$ can be estimated using linear least squares by minimizing

$$\sum_{i=1}^{n} (\boldsymbol{p}_i'^T \mathcal{F} \boldsymbol{p}_i)^2 \tag{2.5}$$

with respect to the coefficients of $\mathcal{F}$ under the constraint that the vector formed by these coefficients has unit norm.

This method generalizes Longuet-Higgins' eight-point algorithm [63] to the uncalibrated case but ignores the rank-two property of fundamental matrices. To enforce this constraint, Luong *et al.* [65, 68] have proposed to use the matrix $\mathcal{F}$ output by the eight-point algorithm as the basis for a two-step estimation process: first, use linear least squares to compute the position of the epipoles $\boldsymbol{e}$ and $\boldsymbol{e}'$ that minimize $|\mathcal{F}\boldsymbol{e}|^2$ and $|\mathcal{F}^T\boldsymbol{e}'|^2$; second, substitute the coordinates of these points in (2.3): this yields a linear parameterization of the fundamental matrix by the coefficients of the epipolar transformation, which can now be estimated by minimizing (2.5) via linear least squares.

Recently, Hartley [37] has proposed a normalized eight-point algorithm and has also reported excellent results. His approach is based on the observation that the poor performance of the plain eight-point method is due, for the most part, to poor numerical conditioning. Thus Hartley has proposed to translate and scale the data so it is centered at the origin and the average distance to the origin is $\sqrt{2}$ pixel. This dramatically improves the conditioning of the linear least-squares estimation process. Accordingly, his method is divided into four steps: first, transform the image coordinates using appropriate translation and scaling operators $\mathcal{T} : \boldsymbol{p}_i \to \hat{\boldsymbol{p}}_i$ and $\mathcal{T}' : \boldsymbol{p}_i' \to \hat{\boldsymbol{p}}_i'$. Second, use linear least squares to compute the matrix $\hat{\mathcal{F}}$ minimizing

$$\sum_{i=1}^{n} (\hat{\boldsymbol{p}}_i'^T \hat{\mathcal{F}} \hat{\boldsymbol{p}}_i)^2.$$

Third, enforce the rank-two constraint; this can be done using the two-step method of Luong *et al.* described earlier, but Hartley uses instead a technique, suggested by Tsai and Huang [111] in the calibrated case, which constructs the singular value decomposition [116]

$$\hat{\mathcal{F}} = \mathcal{U}\mathcal{S}\mathcal{V}^T$$

of $\hat{\mathcal{F}}$. Here, $\mathcal{S} = \mathrm{diag}(r, s, t)$ is a diagonal 3×3 matrix with entries $r \geq s \geq t$, and $\mathcal{U}, \mathcal{V}$ are orthogonal 3×3 matrices. The rank-two matrix $\hat{\mathcal{F}}'$ minimizing the Frobenius norm of $\hat{\mathcal{F}} - \hat{\mathcal{F}}'$ is simply $\hat{\mathcal{F}}' = \mathcal{U}\mathrm{diag}(r, s, 0)\mathcal{V}^T$ [111]. Fourth, set $\mathcal{F} = \mathcal{T}'^T \hat{\mathcal{F}}' \mathcal{T}$. This is the final estimate of the fundamental matrix.

The recent linear algorithm for weak calibration due to Boufama and Mohr [6] and based on an original idea of Couapel [12] is too involved to be described in detail here, so we will just give a rough sketch of their approach and refer the reader to [6] for further details: the main idea is that a set of coplanar points and their images define two homographies between the plane they lie in and the two image planes, and thus a third homography between the image planes themselves; this homography and the knowledge of one of the epipoles provide a new parameterization of the epipolar geometry. (It should be noted that a similar idea was proposed earlier by Luong and Faugeras [66, 68], who have given linear and non-linear methods for weak calibration based on the estimation of the homographies associated with several groups of four or more coplanar points; these groups can be found, when they exist, by using generate-and-test methods such as the ones described in [22, 31, 102, 91].)

In general, four coplanar scene points are not guaranteed to exist, and the *virtual parallax* associated with one of these points is the difference between the position of its actual image projection and the position it would assume if it belonged to the plane formed by the other three points. Boufama and Mohr have shown that the virtual parallax associated with a set of image correspondences can be used (in conjunction with an appropriate choice for projective bases of the two image planes) to estimate the homography and one epipole, thus the whole epipolar geometry. This estimation process can be linearized through an appropriate change of variables, which includes the introduction of an extraneous parameter. Once an estimate of the epipolar geometry has been obtained via linear least squares, it can be refined if necessary by going back to the original non-linear equations and using non-linear least squares. As shown in [6], both the linear and non-linear variants of this method also give very good results and automatically ensure that the rank-two constraint is satisfied. As noted in [119], another reason for the good performance is due to the change of projective coordinates that results in data normalization similar to normalized eight-point algorithm.

2.2.2 Nonlinear Methods

An alternative to the eight-point algorithm is to minimize the distance between the image points and the corresponding epipolar lines, i.e.,

$$\sum_{i=1}^{n}[\mathrm{d}^2(\boldsymbol{p}_i', \mathcal{F}\boldsymbol{p}_i) + \mathrm{d}^2(\boldsymbol{p}_i, \mathcal{F}^T\boldsymbol{p}_i')],$$

which is a non-linear problem, regardless of the parameterization chosen for the fundamental matrix. The minimization can be initialized using the result of the eight-point algorithm; this is the method proposed by Luong *et al.* [65, 68]. In the experiments described in [65, 68], both the parameterization defined by (2.3) and a simple rank-two, seven-parameter parameterization proposed by Robert *et al.* [91] are used with similar results, far superior to those obtained using the eight-point method.

2.3 The Infinitesimal Case: Jepson's and Heeger's Approach

So far we have considered finite displacements between two uncalibrated cameras. We now turn to the case of *infinitesimal* displacements between *calibrated* cameras since this is the setting for Jepson's and Heeger's linear subspace method [50, 51], which is the inspiration for our approach to weak calibration.

Suppose that a moving perspective camera with unit focal length observes some scene. Let $\boldsymbol{p} = (u, v, 1)^T$ denote the position of an image point, and $\dot{\boldsymbol{p}} = (\dot{u}, \dot{v}, 0)^T$ denote the corresponding motion field; it is well known (see [46] for example) that if the instantaneous motion of the camera is described in a coordinate system attached to the scene by the translational velocity $\boldsymbol{t}$ and the rotational velocity $\boldsymbol{\omega}$, then

$$\begin{pmatrix} \dot{u} \\ \dot{v} \end{pmatrix} = \begin{pmatrix} -1 & 0 & u \\ 0 & -1 & v \end{pmatrix} (\frac{1}{z}\boldsymbol{t} + \boldsymbol{\omega} \times \boldsymbol{p}).$$

Eliminating z among the two scalar components of this vector equation yields, after some simple algebraic manipulation:

$$\boldsymbol{p}^T([\boldsymbol{t}_\times][\boldsymbol{\omega}_\times])\boldsymbol{p} + (\boldsymbol{p} \times \dot{\boldsymbol{p}}) \cdot \boldsymbol{t} = 0. \tag{2.6}$$

Equation (2.6) is simply the instantaneous form of the Longuet-Higgins relation (2.1) which captures the epipolar geometry in the discrete, calibrated case. (Viéville and Faugeras [113] have recently derived an equation similar to (2.6) which characterizes the motion field of a camera with varying intrinsic parameters.)

Note that in the case of pure translation we have $\boldsymbol{\omega} = 0$, thus $(\boldsymbol{p} \times \dot{\boldsymbol{p}}) \cdot \boldsymbol{t} = 0$, which yields the well known result that the motion field points toward the epipole (or *focus of expansion*) in the case of pure translational motion.

Let us now turn to the relationship between the infinitesimal Longuet-Higgins relation (2.6) and Jepson's and Heeger's linear subspace method [50, 51]. We can rewrite (2.6) as

$$(\boldsymbol{p} \times \dot{\boldsymbol{p}}) \cdot \boldsymbol{t} = \boldsymbol{t}^T([\boldsymbol{p}_\times]^2)\boldsymbol{\omega}. \tag{2.7}$$

Let us now suppose that we observe the motion field $\dot{\boldsymbol{p}}_i$ at n image points $\boldsymbol{p}_i$ $(i = 1, .., n)$. We define a vector of n coefficients $\boldsymbol{\xi} = (\xi_1, .., \xi_n)^T$ and the vector $\boldsymbol{\tau}(\boldsymbol{\xi}) = \sum_{i=1}^n \xi_i \boldsymbol{p}_i \times \dot{\boldsymbol{p}}_i$. It follows from (2.7) that

$$\boldsymbol{\tau}(\boldsymbol{\xi}) \cdot \boldsymbol{t} = \boldsymbol{t}^T(\sum_{i=1}^n \xi_i [\boldsymbol{p}_{i\times}]^2)\boldsymbol{\omega}.$$

Hence, for any value of $\boldsymbol{\xi}$ such that $\sum_{i=1}^n \xi_i [\boldsymbol{p}_{i\times}]^2 = 0$, we have $\boldsymbol{\tau}(\boldsymbol{\xi}) \cdot \boldsymbol{t} = 0$, which is a linear constraint on $\boldsymbol{t}$. Because the matrix $[\boldsymbol{p}_{i\times}]^2$ has the form

$$[\boldsymbol{p}_{i\times}]^2 = \begin{pmatrix} -1 - v_i^2 & u_i v_i & u_i \\ u_i v_i & -1 - u_i^2 & v_i \\ u_i & v_i & -u_i^2 - v_i^2 \end{pmatrix},$$

the corresponding condition on the coefficients ξ_i is that

$$\sum_{i=1}^n \xi_i (1, u_i, v_i, u_i v_i, u_i^2, v_i^2) = 0.$$

This system of 6 equations in n unknowns admits an $(n-6)$-dimensional space of solutions, and constructing a basis for this subspace yields $n - 6$ linear constraints on $\boldsymbol{t}$, which can then be found via linear least squares. This is Jepson's and Heeger's linear method for estimating the translational component of motion, and we see that it can be derived from the infinitesimal epipolar constraint.

CHAPTER 3

A LINEAR SUBSPACE ALGORITHM

Jepson's and Heeger's method does not generalize in a straightforward way to the discrete motion case, partly because finite rotations introduce additional non-linear constraints. Lawn and Cipolla [58] have proposed a linearized subspace method using a first-order approximation of the rigid body transformation associated with small image regions to derive linear constraints on the translation direction and the position of the epipoles.

In contrast, the linear subspace method proposed in this chapter reformulates the motion equations in a projective coordinate system, where the camera orientation and more generally the location of the image plane are irrelevant, and the motion is completely determined by the displacement of the optical center (see [107] for a related approach in the two-dimensional, calibrated case). Computing this displacement allows us to weakly calibrate our two cameras, i.e., to find the two epipoles and compute the epipolar transformation.

The rest of this chapter is organized as follows: Section 3.1 describes the new algorithm to weak calibration in detail. Then, Section 3.2 discusses the implementation details and experiments along with comparisons to the method discussed in the previous chapter.

3.1 A New Approach to Weak Calibration

We assume in the rest of this section that the reader is familiar with elementary notions of analytical projective geometry [106] such as projective bases, projective coordinates and cross-ratios. A short introduction to these concepts is given in Appendix A for completeness.

3.1.1 A Projective Version of the Epipolar Constraint

Let us choose four scene points –say $\boldsymbol{A}_1$, $\boldsymbol{A}_2$, $\boldsymbol{A}_3$ and $\boldsymbol{A}_4$– and the optical center $\boldsymbol{C}$ of the first camera as a projective basis. We assign to these five points the homogeneous projective coordinates $\boldsymbol{A}_1 = (1,0,0,0)^T$, $\boldsymbol{A}_2 = (0,1,0,0)^T$, $\boldsymbol{C} = (0,0,1,0)^T$, $\boldsymbol{A}_3 = (0,0,0,1)^T$ and $\boldsymbol{A}_4 = (1,1,1,1)^T$. Following Eq. (A.1) in Appendix A, the projective coordinates of a point $\boldsymbol{P}$ in this basis are $(k_1, k_2, k_3, 1)^T$, where

$$\begin{cases} k_1 = [\boldsymbol{A}_2\boldsymbol{C}\boldsymbol{A}_1, \boldsymbol{A}_2\boldsymbol{C}\boldsymbol{A}_3, \boldsymbol{A}_2\boldsymbol{C}\boldsymbol{A}_4, \boldsymbol{A}_2\boldsymbol{C}\boldsymbol{P}], \\ k_2 = [\boldsymbol{C}\boldsymbol{A}_1\boldsymbol{A}_2, \boldsymbol{C}\boldsymbol{A}_1\boldsymbol{A}_3, \boldsymbol{C}\boldsymbol{A}_1\boldsymbol{A}_4, \boldsymbol{C}\boldsymbol{A}_1\boldsymbol{P}], \\ k_3 = [\boldsymbol{A}_1\boldsymbol{A}_2\boldsymbol{C}, \boldsymbol{A}_1\boldsymbol{A}_2\boldsymbol{A}_3, \boldsymbol{A}_1\boldsymbol{A}_2\boldsymbol{A}_4, \boldsymbol{A}_1\boldsymbol{A}_2\boldsymbol{P}], \end{cases}$$

and $\boldsymbol{LMN}$ denotes the plane spanned by the three points $\boldsymbol{L}$, $\boldsymbol{M}$ and $\boldsymbol{N}$, while $[\Pi_1, \Pi_2, \Pi_3, \Pi_4]$ denotes the cross-ratio of the pencil of planes Π_1, Π_2, Π_3, Π_4.

The two cross-ratios k_1 and k_2 are easily computed from image measurements by intersecting the corresponding pencils of planes with the first image plane: if $\boldsymbol{a}_i$ denotes the image of $\boldsymbol{A}_i$ for $i = 1,..,4$, and $\boldsymbol{p}$ denotes the image of $\boldsymbol{P}$, we have

$$\begin{cases} k_1 = [\boldsymbol{a}_2\boldsymbol{a}_1, \boldsymbol{a}_2\boldsymbol{a}_3, \boldsymbol{a}_2\boldsymbol{a}_4, \boldsymbol{a}_2\boldsymbol{p}], \\ k_2 = [\boldsymbol{a}_1\boldsymbol{a}_2, \boldsymbol{a}_1\boldsymbol{a}_3, \boldsymbol{a}_1\boldsymbol{a}_4, \boldsymbol{a}_1\boldsymbol{p}], \end{cases}$$

where $\boldsymbol{mn}$ denotes the line joining the points $\boldsymbol{m}$ and $\boldsymbol{n}$, and $[l_1, l_2, l_3, l_4]$ denotes the cross-ratio of the pencil of lines l_1, l_2, l_3, l_4 (Figure 3.1). The coordinates k_1, k_2 are simply the non-homogeneous plane projective coordinates of the point $\boldsymbol{p}$ in the basis formed by $\boldsymbol{a}_1$, $\boldsymbol{a}_2$, $\boldsymbol{a}_3$, $\boldsymbol{a}_4$, where the basis points have homogeneous projective coordinates $\boldsymbol{a}_1 = (1,0,0)^T$, $\boldsymbol{a}_2 = (0,1,0)^T$, $\boldsymbol{a}_3 = (0,0,1)^T$ and $\boldsymbol{a}_4 = (1,1,1)^T$.

To emphasize the fact that k_1 and k_2 can be directly measured in the image, while k_3 cannot, let us rename these coordinates as $u = k_1$, $v = k_2$, and $z = k_3$. It should be clear that, from now on, the coordinates that we will use for both scene points and their image projections are projective coordinates. We will only switch back to the usual affine image coordinates in Section 3.1.5. Of course, the affine coordinates of the image points can be computed from the projective coordinates through a projective coordinate transformation. The corresponding 3×3 matrix can be computed by writing the change of coordinates for the four points $\boldsymbol{a}_i$, $i = 1,..,4$.

Let $\boldsymbol{C}'$ denote the optical center of the second camera, with homogeneous coordinates $(\alpha, \beta, \gamma, 1)$ in the basis $(\boldsymbol{A}_1, \boldsymbol{A}_2, \boldsymbol{C}, \boldsymbol{A}_3, \boldsymbol{A}_4)$. Let $(u', v', z', 1)$ denote the coordinates of the

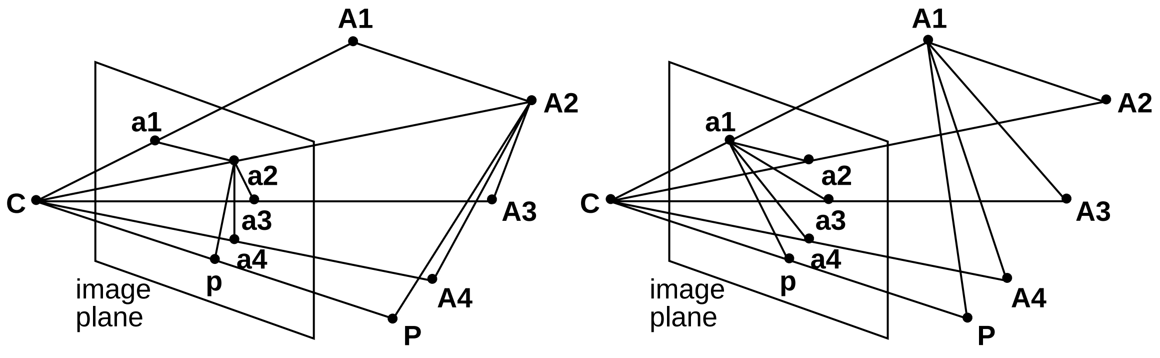


Figure 3.1 The pencils of planes and lines used to compute the $u = k_1$ (left) and $v = k_2$ (right) projective coordinates.

point $\boldsymbol{P}$ in the basis $(\boldsymbol{A}_1, \boldsymbol{A}_2, \boldsymbol{C}', \boldsymbol{A}_3, \boldsymbol{A}_4)$. We compute the coefficients of the projective transformation mapping $(u', v', z', 1)^T$ into $\rho(u, v, z, 1)^T$ by writing the change of coordinates for the five points $\boldsymbol{A}_i$ $(i = 1, .., 4)$ and $\boldsymbol{C}'$, and obtain

$$\begin{cases} \rho u = (\gamma - \alpha) u' + \alpha z', \\ \rho v = (\gamma - \beta) v' + \beta z', \\ \rho z = \gamma z', \\ \rho = (\gamma - 1) + z'. \end{cases}$$

In turn, eliminating ρ and z' between the first, second, and fourth of these equations yields, after some simple algebraic manipulation

$$\boldsymbol{p}^{'T} \mathcal{F} \boldsymbol{p} = 0 \quad \text{where} \quad \mathcal{F} = \begin{pmatrix} \gamma - \alpha & 0 & 0 \\ 0 & \gamma - \beta & 0 \\ 0 & 0 & \gamma - 1 \end{pmatrix} [\boldsymbol{e}_\times], \tag{3.1}$$

$\boldsymbol{e} = (\alpha, \beta, 1)^T$ is the image position of the first epipole, and, as before, $\boldsymbol{p} = (u, v, 1)^T$ and $\boldsymbol{p'} = (u', v', 1)^T$. Equation (3.1) is just another incarnation of the epipolar constraint, $\mathcal{F}$ being the fundamental matrix.

It should be noted that Carlsson [7] has used the double algebra to derive an alternative construction of the fundamental matrix in a projective setting. The end result is of course the same with an appropriate choice of projective basis.

3.1.2 Setting up the Linear Constraints

Reversing the order of the terms, the epipolar constraint can be rewritten as

$$\boldsymbol{e}^T \mathcal{Q} \begin{pmatrix} \gamma - \alpha \\ \gamma - \beta \\ \gamma - 1 \end{pmatrix} = 0 \quad \text{where} \quad \mathcal{Q} = [\boldsymbol{p}_\times] \begin{pmatrix} u' & 0 & 0 \\ 0 & v' & 0 \\ 0 & 0 & 1 \end{pmatrix}. \tag{3.2}$$

Let us now define the vectors $\boldsymbol{f} = (\alpha, \beta, \alpha\beta)^T$ and $\boldsymbol{q} = (v - vu', uv' - u, u' - v')^T$. After some more simple algebraic manipulation, we can rewrite (3.2) as

$$\gamma(\boldsymbol{p} \times \boldsymbol{p}') \cdot \boldsymbol{e} = \boldsymbol{q} \cdot \boldsymbol{f}.$$

Suppose that beside the four reference correspondences, we observe n others, namely $\boldsymbol{p}_i \leftrightarrow \boldsymbol{p}'_i$, with $i = 1, .., n$. Given a vector of n coefficients $\boldsymbol{\xi} = (\xi_1, .., \xi_n)^T$, let us define the vectors $\boldsymbol{\tau}(\boldsymbol{\xi}) = \sum_{i=1}^n \xi_i(\boldsymbol{p}_i \times \boldsymbol{p}'_i)$ and $\boldsymbol{\chi}(\boldsymbol{\xi}) = \sum_{i=1}^n \xi_i \boldsymbol{q}_i$. It follows that

$$\gamma \boldsymbol{\tau}(\boldsymbol{\xi}) \cdot \boldsymbol{e} = \boldsymbol{\chi}(\boldsymbol{\xi}) \cdot \boldsymbol{f}. \tag{3.3}$$

In particular, for any value $\boldsymbol{\xi_\tau}$ such that $\boldsymbol{\tau}(\boldsymbol{\xi_\tau}) = 0$, (3.3) provides a linear constraint on $\boldsymbol{f}$, while for any value $\boldsymbol{\xi_\chi}$ such that $\boldsymbol{\chi}(\boldsymbol{\xi_\chi}) = 0$, the same equation provides a linear constraint on $\boldsymbol{e}$. Since the vectors $\boldsymbol{\xi_\tau}$ and $\boldsymbol{\xi_\chi}$ are each defined by three equations in n unknowns, they form $(n-3)$-dimensional subspaces of $\mathbb{R}^n$.

Because of the particular structure of our problem, both the linear constraints on $\boldsymbol{e}$ and $\boldsymbol{f}$ will prove useful for computing $\boldsymbol{e}$. In particular, it is easy to show (see Appendix B) that the vectors $\boldsymbol{\tau}(\boldsymbol{\xi_\chi})$ and $\boldsymbol{\chi}(\boldsymbol{\xi_\tau})$ are both orthogonal to the vector $(1, 1, 1)^T$. Since the vectors $\boldsymbol{\tau}(\boldsymbol{\xi_\chi})$ are also orthogonal to $\boldsymbol{e}$, they only span a one-dimensional line. Likewise, the vectors $\boldsymbol{\chi}(\boldsymbol{\xi_\tau})$ span a line orthogonal to both $\boldsymbol{f}$ and $(1, 1, 1)^T$.

As shown below, representative vectors $\boldsymbol{\tau}_0 = (a_\tau, b_\tau, c_\tau)^T$ and $\boldsymbol{\chi}_0 = (a_\chi, b_\chi, c_\chi)^T$ for the lines spanned respectively by the vectors $\boldsymbol{\tau}(\boldsymbol{\xi_\chi})$ and $\boldsymbol{\chi}(\boldsymbol{\xi_\tau})$ can be identified through singular value decomposition.

3.1.3 Solving the Linear Constraints

In this section we give a linear method for computing the vectors $\boldsymbol{\chi}_0$ and $\boldsymbol{\tau}_0$. Computing either of these vectors is an instance of the following problem: we are given a set of vectors $\boldsymbol{x}_i$

and $\boldsymbol{y}_i$ $(i = 1,..,n)$ in $\mathbb{R}^p$, having the property that the set of linear combinations $\sum_{i=1}^n \xi_i \boldsymbol{y}_i$ such that $\sum_{i=1}^n \xi_i \boldsymbol{x}_i = 0$ is d-dimensional. We want to construct this d-dimensional set. This is easily done using singular value decomposition as follows.

We suppose $n > p > d$ (this is true in our application, where n is the number of data points, $p = 3$, and $d = 1$). Construct the $p \times n$ matrices

$$\mathcal{X} = [\boldsymbol{x}_1, \dots, \boldsymbol{x}_n] \quad \text{and} \quad \mathcal{Y} = [\boldsymbol{y}_1, \dots, \boldsymbol{y}_n].$$

The set of vectors $\boldsymbol{\xi}$ such that $\sum_{i=1}^n \xi_i \boldsymbol{x}_i = 0$ is simply the null space of $\mathcal{X}$. Writing the singular value decomposition of $\mathcal{X}$ yields:

$$\mathcal{X} = \mathcal{U}\mathcal{S}\mathcal{V}^T,$$

where $\mathcal{U}$ is an orthogonal $p \times p$ matrix, $\mathcal{S}$ is a $p \times n$ matrix, and $\mathcal{V}$ is an orthogonal $n \times n$ matrix. In the generic case, $\mathcal{X}$ has (maximal) rank p, and a basis for its null space is formed by the rightmost $n - p$ columns of $\mathcal{V}$.

Let $\mathcal{V}_0$ denote the $n \times (n - p)$ matrix formed by these columns, the set of vectors $\sum_{i=1}^n \xi_i \boldsymbol{y}_i$ such that $\sum_{i=1}^n \xi_i \boldsymbol{x}_i = 0$ is thus the subspace spanned by the columns of the matrix $\mathcal{Y}\mathcal{V}_0$.

According to our assumptions, this $p \times (n - p)$ matrix has rank $d < p$. To find the range of $\mathcal{Y}\mathcal{V}_0$, we construct its singular value decomposition

$$\mathcal{Y}\mathcal{V}_0 = \mathcal{U}_1\mathcal{S}_1\mathcal{V}_1^T,$$

and the range is spanned by the leftmost d columns of $\mathcal{U}_1$.

The rank-d hypothesis can be verified by comparing the first d singular values in $\mathcal{S}_1$ to the remaining ones.

3.1.4 Estimating the Position of the Epipoles

Once the vectors $\boldsymbol{\tau}_0$ and $\boldsymbol{\chi}_0$ have been estimated, we can compute the position of the epipoles as follows: since both $\boldsymbol{\tau}_0$ and $\boldsymbol{\chi}_0$ are orthogonal to $(1, 1, 1)^T$, we have $c_\tau = -a_\tau - b_\tau$ and $c_\chi = -a_\chi - b_\chi$, which allows us to rewrite (3.3) as

$$\begin{pmatrix} a_\tau & b_\tau \\ b_\chi\beta & a_\chi\alpha \end{pmatrix} \begin{pmatrix} 1 - \alpha \\ 1 - \beta \end{pmatrix} = 0 \tag{3.4}$$

A necessary condition for this system of two homogeneous linear equations in $1-\alpha$ and $1-\beta$ to admit a non-trivial solution is that its determinant $a_\tau a_\chi \alpha - b_\tau b_\chi \beta$ is zero. Solving for β as a function of α and substituting into the first equation of (3.4) finally yields

$$\alpha = \frac{b_\chi}{a_\tau} \frac{a_\tau + b_\tau}{a_\chi + b_\chi}, \quad \beta = \frac{a_\chi}{b_\tau} \frac{a_\tau + b_\tau}{a_\chi + b_\chi}. \tag{3.5}$$

Substituting these values back into the second equation of (3.4) shows that they are indeed solutions of our problem.

3.1.5 Estimating the Epipolar Transformation

Once α and β have been computed, the affine coordinates of the epipoles are easily computed from the image positions of the points $\boldsymbol{a}_i$ $(i = 1,..,4)$. Accordingly, let us change notation so that u, v (resp. u', v') now denote the affine coordinates of a point in the first (resp. second) image, and α, β and α', β' denote the affine coordinates of the two epipoles. We can rewrite the epipolar transformation (2.3) as a linear constraint on a, b, c, d:

$$((v-\beta)(u'-\alpha'), (u-\alpha)(u'-\alpha'), -(v-\beta)(v'-\beta'), -(u-\alpha)(v'-\beta')) \begin{pmatrix} a \\ b \\ c \\ d \end{pmatrix} = 0, \tag{3.6}$$

and it follows that, given n correspondences, the coefficients a, b, c, d can be estimated (up to an irrelevant scale factor) using linear least squares. Thus the whole epipolar geometry can be recovered through linear least squares and singular value decomposition.

3.1.6 Summary of the Algorithm

Table 3.1 shows the various steps of the proposed approach to weak calibration. It should be noted that, unlike some recent non-linear approaches to the same problem (e.g., [68]), our method does not minimize a physically-significant error such as the distance between image points and the corresponding epipolar lines. On the other hand, it does not require an initial guess. As will be shown in Section 3.2.2, both approaches can be combined by using a non-linear distance-minimization algorithms to “polish” the results obtained by our method, an approach reminiscent of numerical root finding, where a few Newton-Raphson iterations are used to fine-tune the results obtained by reliable methods with slower convergence rate.

1. Pick four points $\boldsymbol{a}_1$, $\boldsymbol{a}_2$, $\boldsymbol{a}_3$ and $\boldsymbol{a}_4$ in the left image and use these points and their matches in the right image as projective image bases.

2. Transform the image points from the Euclidean to the projective frame.

3. Estimate the position of the epipoles as follows:

   (a) Use the SVD-based method of Section 3.1.3 and Eq. (3.3) to compute the vector $\boldsymbol{\tau}_0$.
   (b) Use the SVD-based method of Section 3.1.3 and Eq. (3.3) to compute the vector $\boldsymbol{\xi}_0$.
   (c) Use Eq. (3.5) to compute the projective coordinates of the epipoles.
   (d) Convert the projective coordinates of the epipoles back to Euclidean space.

4. Estimate the remaining parameters of the fundamental matrix using Eq. (3.6) and linear least-squares.

Table 3.1 The linear subspace algorithm.

It should also be noted that centering and scaling the data points as proposed by Hartley [37] will not change the outcome of our algorithm since it uses as input the projective coordinates of the image points in the basis formed by the points $\boldsymbol{a}_i$ $(i = 1, .., 4)$: these coordinates are of course independent of any scaling or centering transformation. In addition, we are not aware of scaling and/or conditioning methods for improving the stability of singular value decomposition: as noted in [11], it is possible to scale the rows and columns of a matrix to improve the reliability of the estimation of its rank. But this will *not* improve the reliability of the estimation of its range and null space, which are exactly the quantities we are trying to estimate.

3.2 Empirical Evaluation of the Proposed Approach

We have implemented the linear subspace approach proposed in Section 3.1 using the LINPACK public-domain package for numerical linear algebra [17]. This section discusses the results of our experiments with real and synthetic data.

3.2.1 Experiments Using Real Data

The real data used in our experiments were kindly provided by Boubakeur Boufama; they consist of point matches between pairs of images of a house model (data sets H(a) and H(b), see Figure 3.2), a bridge (data sets B(a), B(b) and B(c), see Figure 3.3), and a tribunal (data sets

T(a) and T(b), see Figure 3.4). The matches were found automatically by using correlation to track curvature extrema in image sequences [75], and their number varies from 25 to 55 depending on the data set. Figures 3.2 to 3.4 show the images, the data points (small white discs) and the epipolar lines (short white line segments) estimated by our program.

Quantitatively, we have used in all our tests the distance between the data points and the recovered epipolar lines, averaged over the two images and all the data points, as an error measure. The output of the program depends of course on the point matches $\boldsymbol{a}_i \leftrightarrow \boldsymbol{a}'_i$ $(i = 1, .., 4)$ chosen as projective bases for the two images. The results shown in Figures 3.2 to 3.4 correspond to "good" basis choices, obtained for each data set by trying 30 random bases, then selecting the one minimizing our error measure.

It would of course be interesting to develop a better method for selecting an appropriate basis. At the very least, it is possible to ensure that the four points selected are not coplanar, using methods such as the ones described in [22, 31, 102, 91].

To gain a better understanding of the effect of basis choice on the performance of our approach, we have also run our weak calibration program with 500 randomly chosen bases for each data set, and recorded the results in histograms of the average distance between the data points and the estimated epipolar lines. Figure 3.5 shows the outcome of this experiment: qualitatively, the histograms are roughly unimodal, with the largest peak always corresponding to an average error smaller than 1.5 pixel. Quantitative results are shown in the accompanying table. They indicate that a small number of random choices should typically be sufficient to obtain good results: for every data set, at least 10% of the choices yield an error of 1 pixel or less. This means that one out of 30 random choices will achieve this level of performance with a probability of $1 - 0.9^{30} \approx 0.95$, assuming of course that our 500 samples provide a good estimate of the actual probability distribution.

3.2.2 Comparison with Other Methods

We have conducted a number of comparisons between the linear subspace approach and other methods for weak calibration: our LINPACK implementation of Hartley's normalized eight-point algorithm [37], Boufama's and Mohr's implementation of the virtual parallax method [6], Quan's implementation of the non-linear distance minimization technique of Luong *et al.* [65, 68], which uses the output of the plain eight-point method to initialize the minimization,

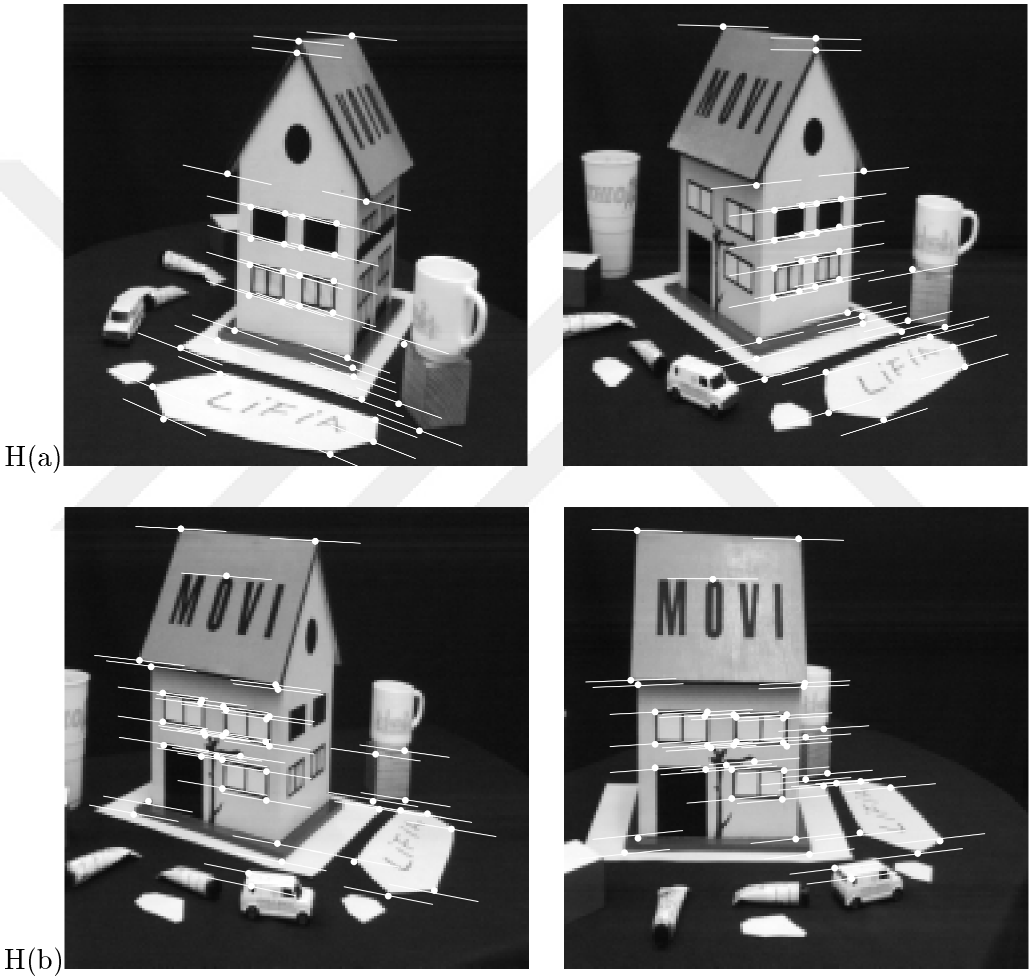


Figure 3.2 Recovered epipolar lines for the house data sets.

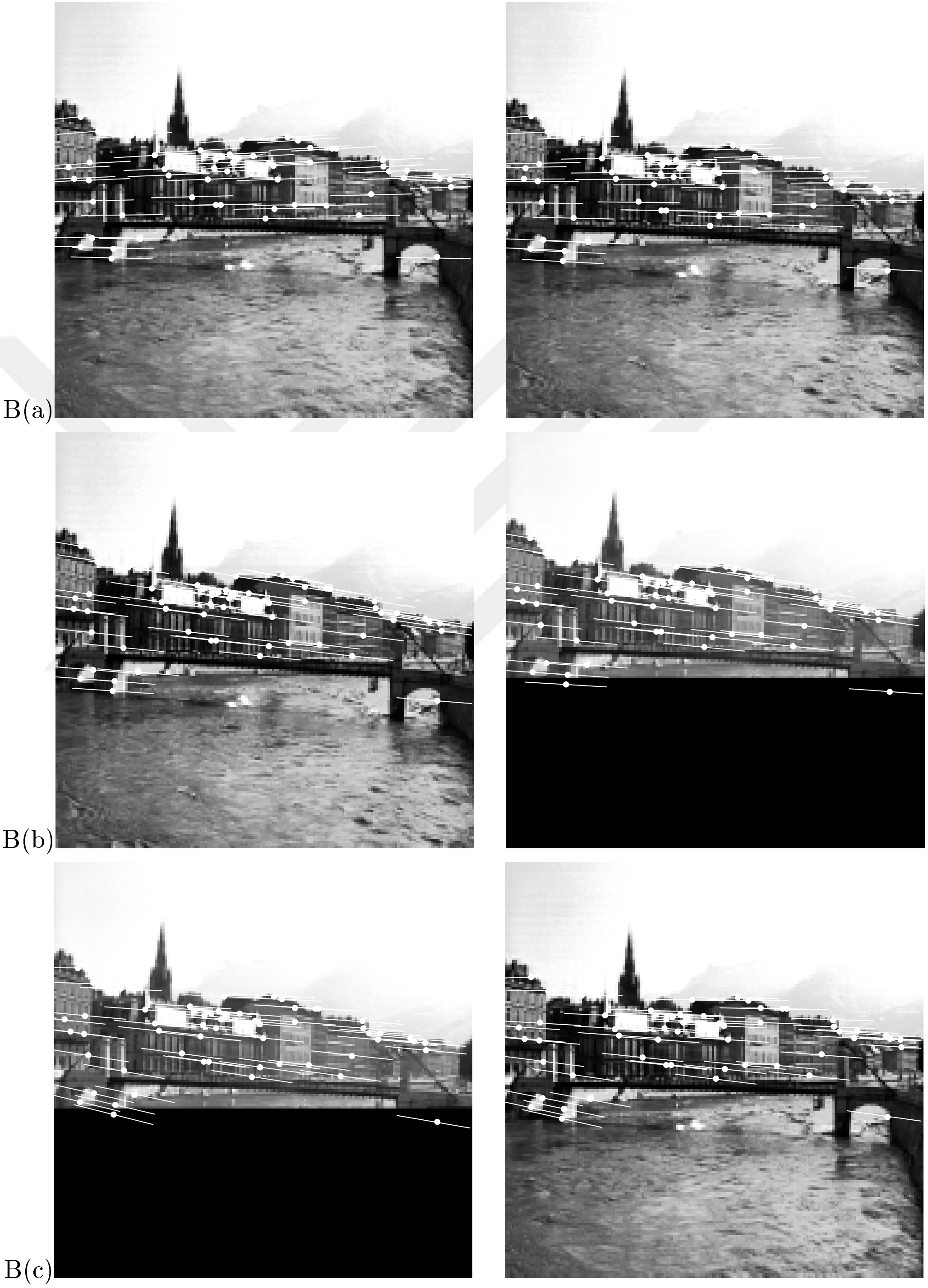


Figure 3.3 Recovered epipolar lines for the bridge data sets.

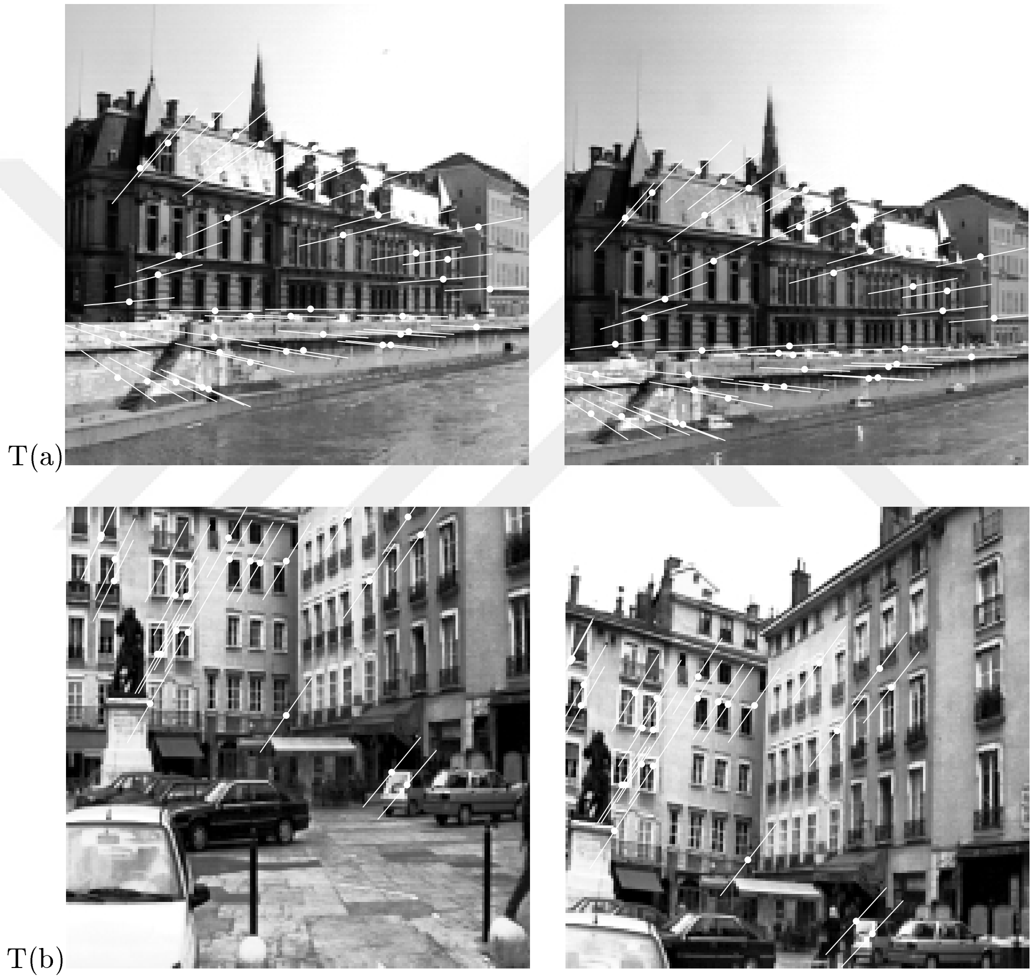


Figure 3.4 Recovered epipolar lines for the tribunal data sets.

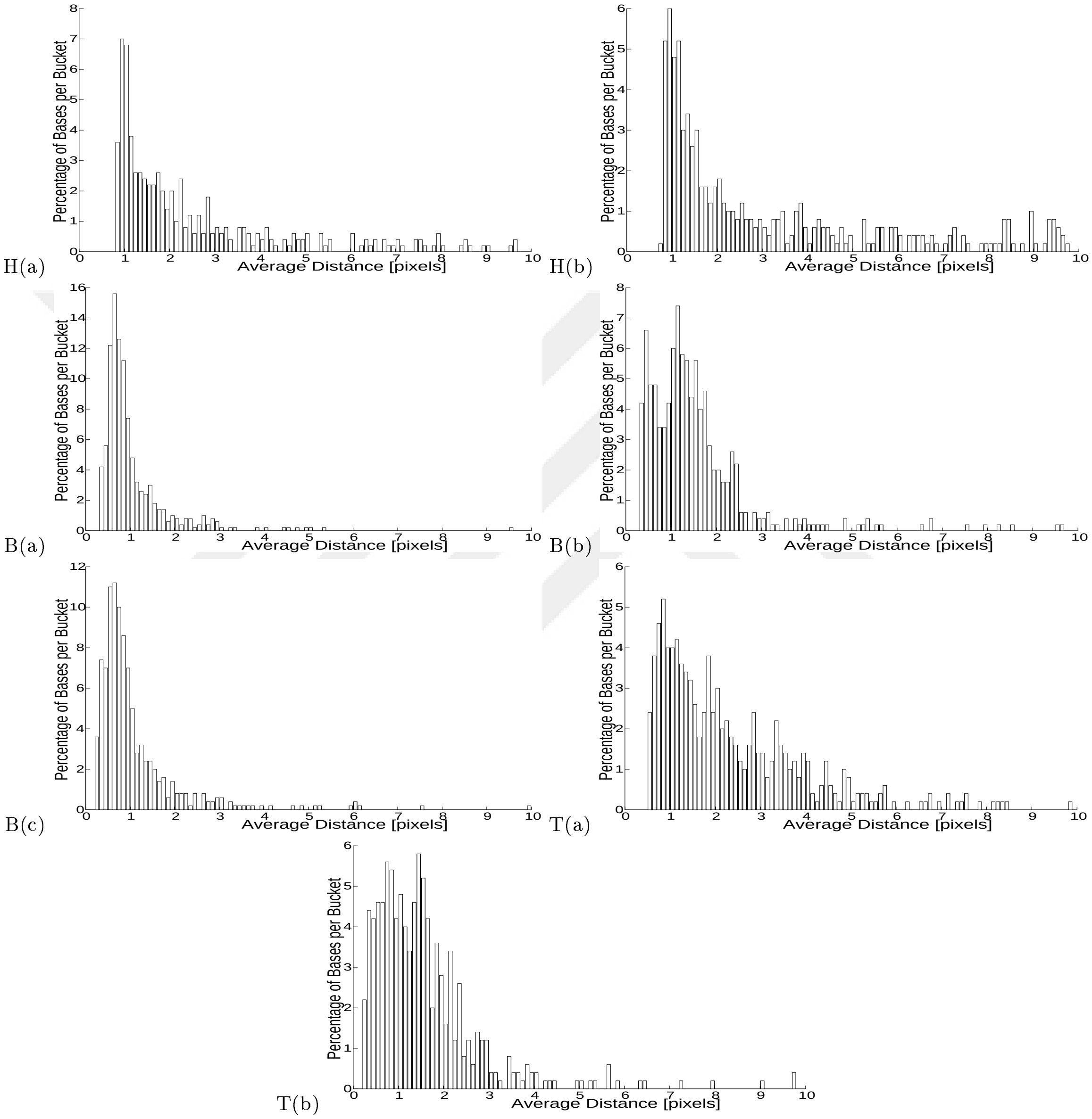


Figure 3.5 Effect of basis choice on the performance of the algorithm. The bucket size is 0.1 pixel, and we only show the part of the histograms corresponding to errors between 0 and 10 pixels.

and our own implementation of the distance minimization algorithm, which relies instead on the output of the linear subspace approach as initial guess. The latter program uses the Levenberg-Marquardt routine from the MINPACK public-domain library [77] to perform the distance minimization.

Before presenting these comparisons, let us say a few words about how we have chosen to illustrate the results of the various algorithms. As explained in Section 2.2, Hartley's normalized eight-point method is divided into four steps: translating and scaling of the data, estimating the matrix $\hat{\mathcal{F}}$ through linear least squares, replacing it by the rank-two matrix $\hat{\mathcal{F}}'$ using singular value decomposition, and finally computing $\mathcal{F}$. We have implemented this technique as well as a variant using the two-step method of Luong *et al.* [65, 68] mentioned earlier, which bypasses the computation of $\hat{\mathcal{F}}'$ and instead enforces the rank-two constraint by computing the epipoles from $\mathcal{F}$ then estimating the epipolar transformation. It is easy to show that the epipoles computed by this method are exactly the same as the epipoles computed using Hartley's original algorithm, but that the epipolar transformations are different. As shown in Figure 3.6(a), this variant has given better results than the original method in our experiments. This has also proven to be the case for the virtual parallax algorithm (Figure 3.6(b)): even though $\mathcal{F}$ is estimated directly, smaller errors are obtained when the epipolar transformation is estimated a posteriori from the epipoles' positions. Thus, for both algorithms, the error plots presented in this section have been constructed by (1) computing the epipoles, either directly or from the fundamental matrix, (2) estimating the epipolar transformation, and (3) using it to construct the epipolar lines. We have also tested both the linear and non-linear versions of the virtual parallax method of Boufama and Mohr, using their own implementation. As expected, the non-linear variant, which uses the output from the linear one as initial guess, has yielded better results in all of our experiments (Figure 3.6(c)). Thus we have used the non-linear version of their approach in all our comparisons. It should also be noted that the virtual parallax method, like ours, relies on the choice of a particular projective basis. In Boufama's and Mohr's current implementation, this choice is fixed, which will bias any comparison in favor of our approach since we always try 30 different bases and pick the best one. In addition, excluding four correspondences (the basis points) from the weak calibration process may also in certain cases bias the comparison in favor of our method since the four points may be very noisy. However, it is unlikely that

N8P	Normalized eight-point algorithm
VP	Virtual parallax algorithm
LS	Linear subspace algorithm
D8P	Distance minimization algorithm starting from plain eight-point guess
DLS	Distance minimization algorithm starting from linear subspace guess

Table 3.2 The abbreviations for the weak calibration algorithms tested in the experiments.

our method would give good results in this case since the whole calibration process depends on these points.

We now present our comparisons, using the abbreviations in Table 3.2 to designate the various techniques tested in our experiments.

Figure 3.7 compares the five methods using our seven image pairs. It shows that, even though the DLS algorithm always gives the best results, the performances of the five methods are remarkably close over the range of data. In particular, there is no clear winner among the linear algorithms: the only significant difference between them has been measured for the data set T(b), for which the error associated with the VP and LS methods is roughly one half of the error associated with the N8P algorithm. As reported in [6] and confirmed by our experiments, this is also a case where the D8P algorithm fails to recover the correct epipolar geometry, probably because it converges to the wrong local minimum from a bad initial guess found by the plain eight-point algorithm.

It should be noted that we have also used the output of the N8P and VP algorithms to initialize the distance minimization algorithm, with results essentially identical to those obtained using the DLS method. This indicates that the non-linear distance minimization algorithm works very well when provided with adequate initial guesses, and suggests using it to refine the results obtained by any of the other methods, an approach reminiscent of numerical root finding, where a few Newton-Raphson iterations are used to “polish” the results obtained by reliable methods with slower convergence rate.

Figure 3.8 compares the effects of noise on the five methods: we have constructed synthetic data from the image pairs in Figures 3.2 to 3.4 by modifying the image positions of the point correspondences until they perfectly agree with the epipolar geometry shown in the figures. We have then added various amounts of zero-mean Gaussian noise to these data, with standard

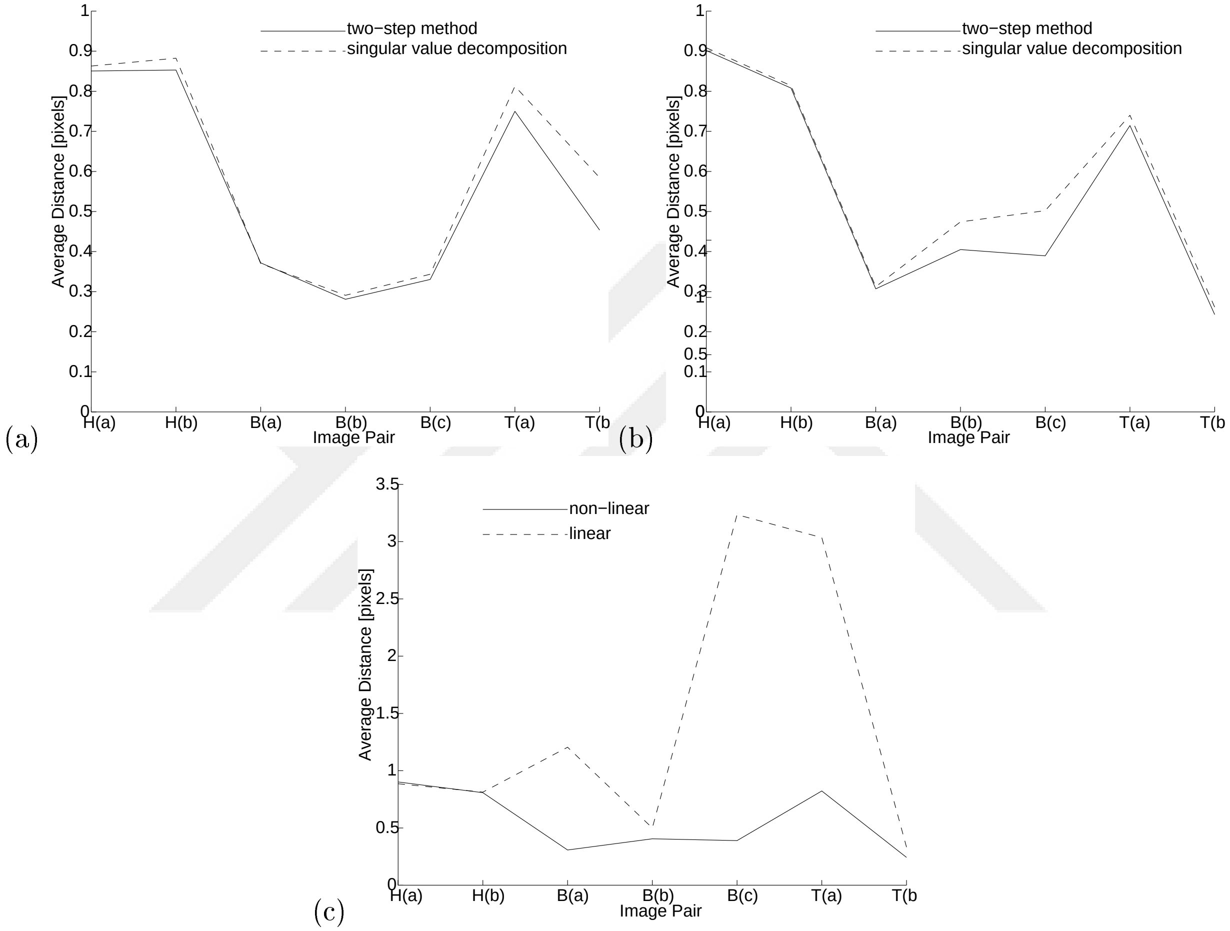


Figure 3.6 Results obtained by running variants of the normalized eight-point and virtual parallax methods on our seven data sets: (a) normalized eight-point algorithm; the dashed curve represents the errors measured when using the singular value decomposition of the fundamental matrix to compute the epipolar lines, while the solid curve represents the errors measured when the epipoles are used as a basis for estimating the epipolar transformation and thus the epipolar lines; (b) similar plot for the non-linear version of the virtual parallax algorithm; (c) linear and non-linear versions of the virtual parallax algorithm.

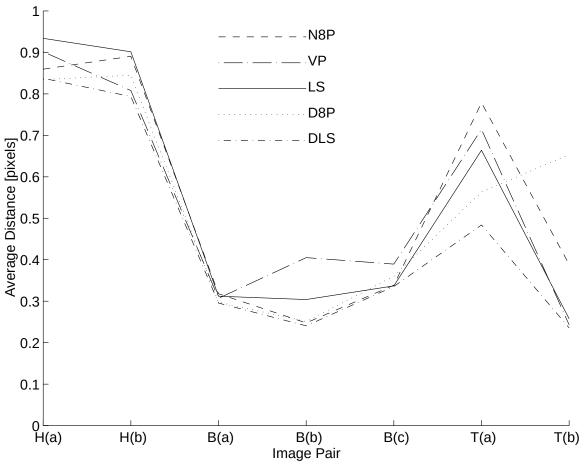


Figure 3.7 A comparison of five weak calibration techniques using real data.

deviation between 0 and 1.5 pixel, and run the five programs. The plots shown in Figure 3.8 have been constructed by averaging the errors measured for 100 runs of each method for each noise level (in the case of the subspace method, each "run" of the algorithm actually corresponds to 30 runs with random basis choices, always picking the best one).

According to these experiments, the N8P, VP and LS methods once again achieve very close performance levels. If we now turn our attention to the non-linear distance minimization algorithms, we observe that the D8P method sometimes yields large errors at high noise levels: in particular, the error measured for data sets B(b) and B(c) is over 3.5 pixels at the 0.5-pixel noise level, and it reaches 17 pixels at the 1.5-pixel noise level (the latter is not shown in Figure 3.8, where all plots are truncated at the 4-pixel error level for readability). The most likely cause for this behavior is again convergence to the wrong local minimum from a bad initial guess found by the plain eight-point algorithm. In contrast, the DLS algorithm yields the best results of all of the implemented methods.

We have conducted a second series of experiments with synthetic data to see how well the epipoles are recovered by the various methods. Following [68], we define the relative error between a ground-truth epipole $\boldsymbol{e}_1 = (\alpha_1, \beta_1)^T$ and its estimate $\boldsymbol{e}_2 = (\alpha_2, \beta_2)^T$ by

$$\frac{1}{2}\min(\frac{|\alpha_1 - \alpha_2|}{\min(|\alpha_1|, |\alpha_2|)}, 1) + \frac{1}{2}\min(\frac{|\beta_1 - \beta_2|}{\min(|\beta_1|, |\beta_2|)}, 1).$$

We then average the relative errors associated with the two epipoles. Figure 3.9 shows the results of our experiments. As before, the plots shown in this figure have been constructed by

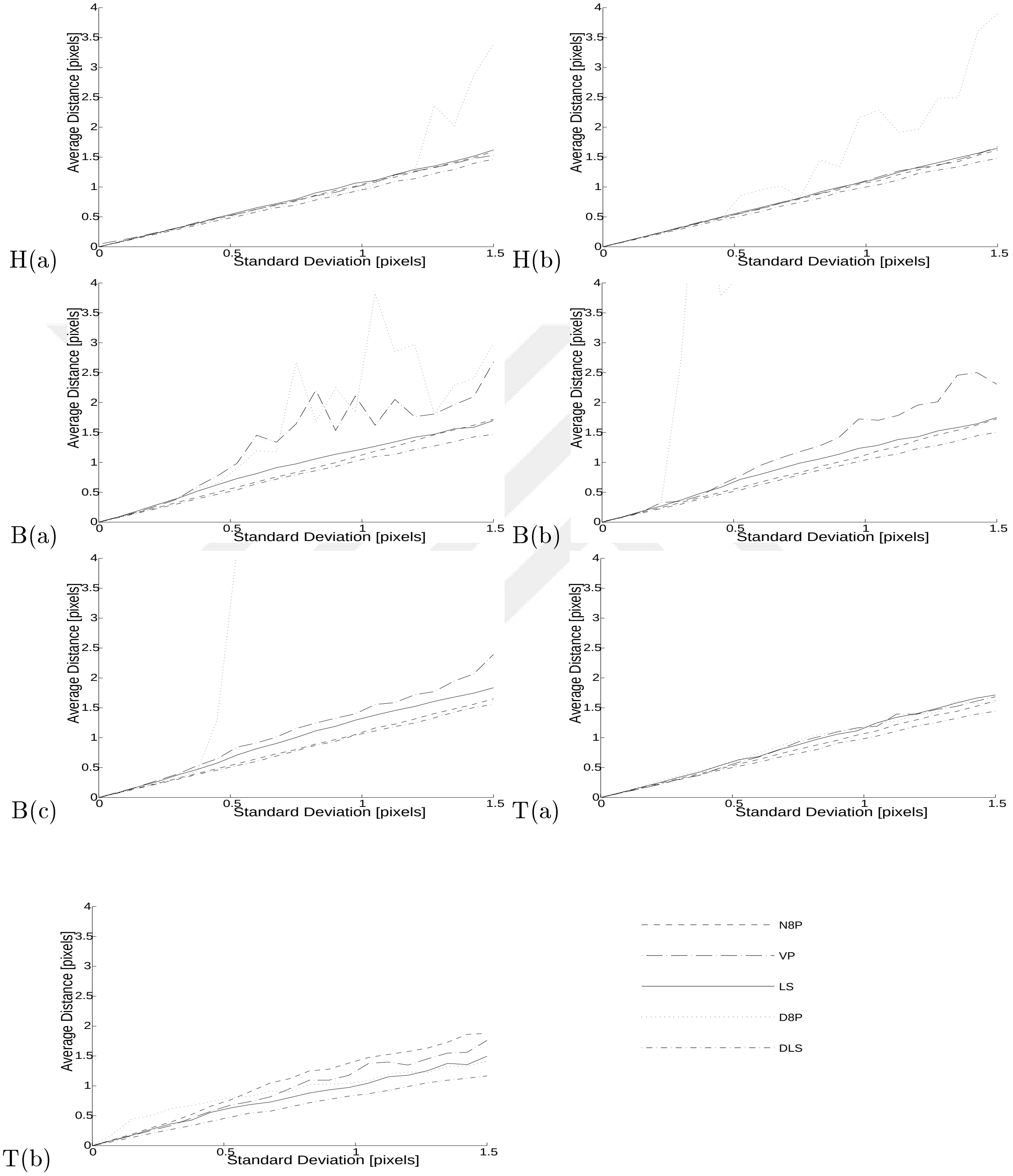


Figure 3.8 Effect of noise on the five methods. Note that all plots are truncated at an error level of 4 pixels.

averaging the errors measured for 100 runs of each method for each noise level, each "run" of the subspace method corresponding to 30 runs with random basis choices, always picking the one that minimizes the average distance to the epipolar lines. These new experiments confirm that the various methods give very close levels of performance, except for the D8P method, which as expected, gives by far the worse results at high noise levels.

3.2.3 Computational Cost

Recovering the complete epipolar geometry using the subspace algorithm requires constructing the singular value decomposition of several matrices. In particular, assuming we have n point correspondences, steps 3(a) and 3(b) have the same complexity. Each step performs the following SVD-based operations: the computation of the null space $\mathcal{V}_0$ of a $3 \times (n-4)$ matrix and the computation of the range $\mathcal{U}_1$ of a $3 \times (n-7)$ matrix. In addition, the computation of the matrix $\mathcal{Y}\mathcal{V}_0$ requires the multiplication of two matrices of sizes $3 \times (n-4)$ and $(n-4) \times (n-7)$. Finally, after computing both the epipoles, step 4 requires the computation of the null space of an $(n-4) \times 4$ matrix. The contribution of the other steps of the algorithm to its overall complexity is negligible.

The normalized eight point algorithm requires the SVD-based computation of the null space of an $n \times 9$ matrix. The cost of the other computations is negligible.

It can be shown that the number of floating point operations required by singular value decomposition for a $p \times q$ matrix where $p \gg q$ is linear in p and quadratic in q [11]. Since the SVD of the transpose of a matrix is the transpose of its SVD, constructing the singular value decomposition of a $q \times p$ matrix where $p \gg q$ is also linear in p and quadratic in q. Thus, the SVD computations involved in the LS method are expected to be cheaper then the corresponding computations in the N8P method. Note however, that as n grows, the computation of the matrix $\mathcal{Y}\mathcal{V}_0$, whose cost is quadratic in n, will eventually dominate the cost of the LS method.

We do not have an estimate of the computational cost of the non-linear algorithms (i.e., the distance minimization algorithm and the non-linear virtual parallax algorithm). The number of steps required to converge to a solution depends both on the input data and the initial guess.

Table 3.3 gives actual running times for all methods, measured using our seven data sets on a SUN SPARCStation 10.

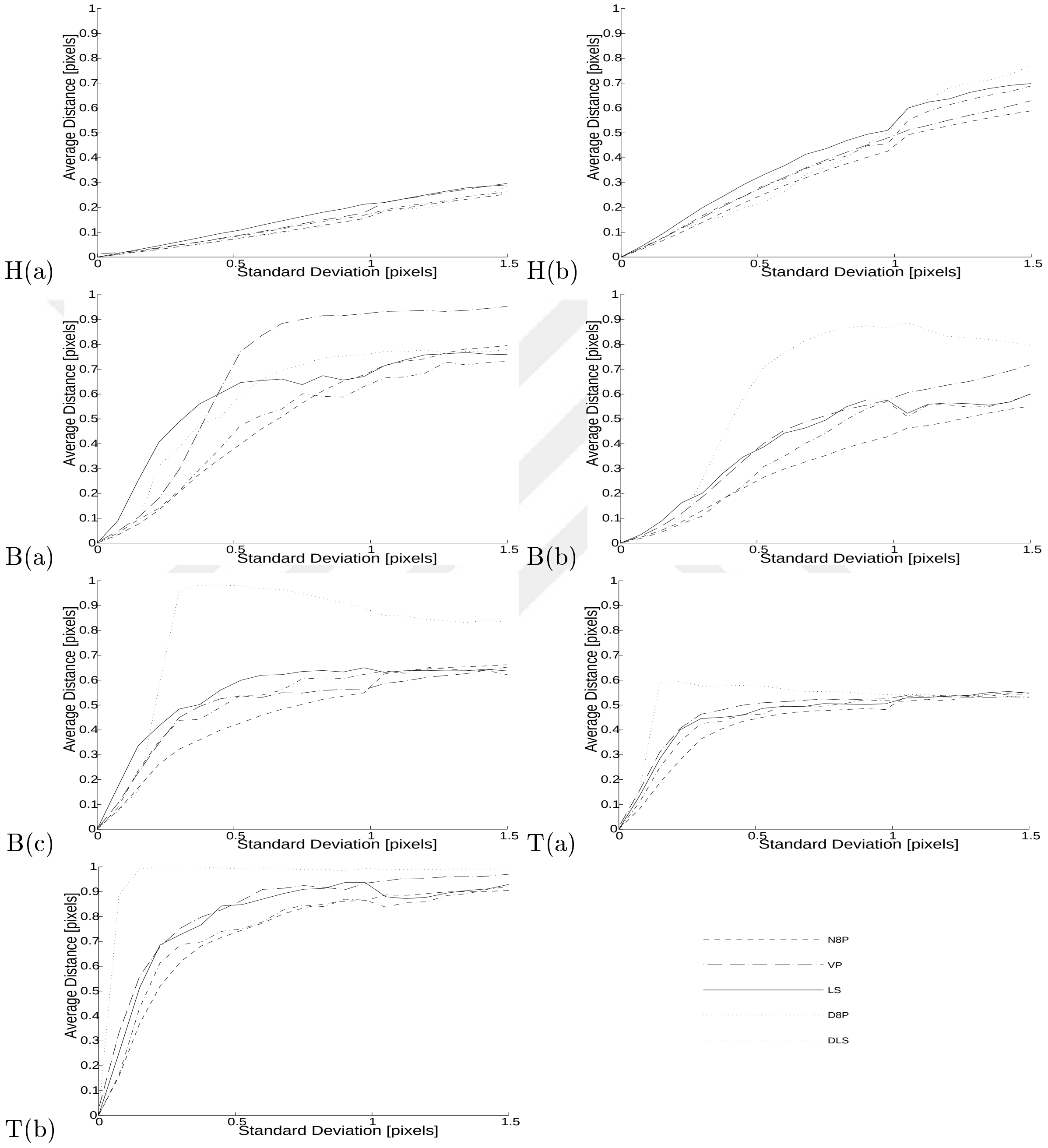


Figure 3.9 Effect of noise on the five methods. Here the relative error in reconstruction of epipoles with respect to ground truth is given. See the text for details.

		Average Running Time of Algorithms [ms]				
Data	n	N8P	VP	LS	30 × LS	DLS
H(a)	37	100	442	22	660	974
H(b)	46	120	471	31	930	2490
B(a)	55	150	502	44	1320	1874
B(b)	55	150	497	44	1320	1896
B(c)	55	150	591	44	1320	3942
T(a)	46	120	483	31	930	4321
T(b)	25	70	454	12	360	786

Table 3.3 Running times of the various methods. We have indicated the running time for both a single run of the LS method and the 30 runs required to select a good basis.

This table confirms the above analysis: a single run of the LS method is cheaper than a run of the N8P method, but 30 runs of the LS method are of course more expensive. The LS/N8P cost ratio increases with n as the quadratic matrix multiplication step becomes more and more costly.

3.3 Discussion

By recasting Jepson's and Heeger's linear subspace method for infinitesimal motion analysis in the setting of finite motion analysis for uncalibrated cameras, we have been able to construct a reliable linear algorithm for weak calibration. The method estimates the right number of parameters and an implicit data normalization is done due to the change of projective coordinates resulting in good performance.

We have compared our algorithm empirically with other approaches to the same problem: the plain and normalized eight-point methods of [37], the virtual parallax algorithm of [6, 12], and the non-linear distance minimization technique of [65, 68]. According to our experiments with both real and synthetic data, there does not appear to be a clear (i.e., statistically significant) winner among the linear methods, which all give very close levels of performance, except for the plain eight-point method, whose performance degrades quickly as noise increases. To refine the results of this comparison, it would be very interesting to also conduct a theoretical analysis of the stability of the linear algorithms, as was done recently by Luong and Faugeras [67, 68] for the non-linear method of Luong *et al.* [65, 68] using both statistical techniques and the notion of *critical surfaces* from photogrammetry (see the books by Maybank [71] and Faugeras [20] for discussions of these surfaces in the computer vision context).

A key part of our approach is the choice of an appropriate basis for projective space, which assumes that four reference points can be measured accurately. An interesting question is whether it is possible to design a different linear subspace algorithm that takes every image measurement equally into account, hopefully averaging out individual errors. Another interesting question is whether the linear subspace approach can be generalized to handle calibrated cameras directly and estimate the full motion in a linear manner.

CHAPTER 4

IMAGE-BASED RENDERING

This part of the thesis is concerned with the problem of rendering new images of a scene from a set of pre-recorded pictures. Before presenting our approach in Chapter 5, we review here current approaches to this problem.

Traditionally, synthesizing a new view of a scene for a virtual camera position has been done by first reconstructing a three-dimensional model from a set of images using computer vision techniques and then rendering the model using computer graphics techniques. As an alternative, recent work in computer graphics [9, 30, 61] and computer vision [57, 95, 56, 3] has demonstrated the possibility of rendering three-dimensional scenes without explicit three-dimensional models (*image-based rendering*). The computer graphics techniques developed by Chen [9], Gortler, Grzeszczuk, Szeliski and Cohen [30], and Levoy and Hanrahan [61] are based on the idea that the set of all visual rays (light field) is four-dimensional, and can thus be characterized from a two-dimensional sample of two-dimensional images of a rigid scene. In contrast, the computer vision techniques proposed by Laveau and Faugeras [57], Seitz and Dyer [95], and Kutulakos and Vallino [56] only use a discrete (and possibly small) set of views among which point correspondences have been established by feature tracking or conventional stereo matching.

One of the earliest attempts to render new images from an existing set of images without three-dimensional reconstruction was the movie-map approach of Lippman [62]. This method was based on an image database containing a vast number of images representing the scene. New views of the scene were rendered by taking the closest image corresponding to the virtual camera position defining the new view. The closest view in the database then morphed to allow small camera motions to get the desired view. With this method, realistic pictures can be synthesized provided that a close enough view is present in the database. Clearly, when a large variation in viewpoint is allowed, one needs to have so many images that the approach becomes unmanageable.

Practical methods for image-based rendering have, however, been developed recently. As shown below, they can be roughly divided into three groups: (1) model-based, (2) transfer-based, and (3) light-field methods. We review these techniques next.

4.1 Model-Based Techniques

Model-based techniques to image-based rendering involve two distinct steps: reconstruction of the three-dimensional model from a set of images using techniques from computer vision, and rendering the model using computer graphics techniques. Besides the model, texture information may also be extracted from the images to be used in the rendering process.

There is a tremendous amount of work in the computer vision literature on model reconstruction from intensity images with different paradigms such as structure from motion (e.g., [55, 108, 19, 105]), shape from stereo (e.g., [70, 79]), shape from shading (e.g., [49, 84]), and other model reconstruction methods (e.g., [13, 104]). The extracted model may be a complete CAD model or just the positions of some geometric features like points and lines.

Once the 3D model representing the scene has been reconstructed, a new view of the scene can be obtained by first setting the virtual camera position, i.e., the position and the orientation of the camera, and then rendering the model using computer graphics techniques. This usually requires additional information such as an illumination model and the surface reflectance properties of the objects in the scene. Using these information, a picture of the scene can be synthesized for example by ray tracing (for more information on ray tracing see [41, Chapter 14]). To achieve more realistic images one can use texture mapping (see [42] for a review) using the texture extracted from the input images. Depending on the details in the model, model-based methods can yield very realistic images.

A complete image-based rendering system that represents the scene using depth maps has been described by Kanade *et al.* in [52, 78]. They have described a visual medium, called *virtualized reality*, where a representation of a real scene is captured as a collection of stereo intensity images along with reconstructed depth maps. A depth map contains the depth of each pixel corresponding to a point in the scene. When a new view is rendered the stereo intensity images provide the texture to be used in texture mapping.

As an example of model reconstruction, we describe below the factorization method of Tomasi and Kanade [108] that will be compared with our approach in later chapters.

4.1.1 Factorization Method of Tomasi and Kanade

An orthographic camera projects a scene P onto its image plane with

$$\begin{cases} u = \boldsymbol{a}^T \boldsymbol{P} + t_x, \\ v = \boldsymbol{b}^T \boldsymbol{P} + t_y, \end{cases} \tag{4.1}$$

where $\boldsymbol{p} = (u, v)^T$ is the image of the point P, the vectors $\boldsymbol{a}^T$ and $\boldsymbol{b}^T$ are the first two rows of the camera rotation matrix, and t_x and t_y are the first two coordinates of the camera translation. Here we are considering a set of scene points P_i (for $i = 1, ..., m$) and their orthographic projections p_i.

Without loss of generality, we can set the origin of the world coordinate frame at the center of mass of the scene, i.e.,

$$\boldsymbol{P}_0 = \frac{1}{m} \sum_{i=1}^{m} \boldsymbol{P}_i. \tag{4.2}$$

The orthographic projection of the point $\boldsymbol{P}_0$ is also the center of mass of the image points, i.e.,

$$\boldsymbol{p}_0 = \frac{1}{m} \sum_{i=1}^{m} \boldsymbol{p}_i. \tag{4.3}$$

When we apply the projection in (4.1) to the point $\boldsymbol{P}_0$, we can solve for the camera translation with

$$\begin{pmatrix} t_x \\ t_y \end{pmatrix} = \frac{1}{m} \sum_{i=1}^{m} \begin{pmatrix} u_i \\ v_i \end{pmatrix}. \tag{4.4}$$

We can now choose the center of image coordinate frame at $\boldsymbol{P}_0$, and rewrite the projection equations in matrix form with (for f orthographic images of m points)

$$\mathcal{W} = \mathcal{M}\mathcal{S}, \tag{4.5}$$

where

$$\mathcal{W} = \begin{pmatrix} u_1^{(1)} & u_2^{(1)} & \ldots & u_m^{(1)} \\ \ldots & \ldots & \ldots & \ldots \\ u_1^{(f)} & u_2^{(f)} & \ldots & u_m^{(f)} \\ v_1^{(1)} & v_2^{(1)} & \ldots & v_m^{(1)} \\ \ldots & \ldots & \ldots & \ldots \\ v_1^{(f)} & v_2^{(f)} & \ldots & v_m^{(f)} \end{pmatrix} \tag{4.6}$$

is the registered measurement matrix,

$$\mathcal{M} = \begin{pmatrix} \boldsymbol{a}^{(1)} \\ \ldots \\ \boldsymbol{a}^{(f)} \\ \boldsymbol{b}^{(1)} \\ \ldots \\ \boldsymbol{b}^{(f)} \end{pmatrix} \tag{4.7}$$

is the motion matrix, and

$$\mathcal{S} = \begin{pmatrix} x_1 & x_2 & \ldots & x_m \\ y_1 & y_2 & \ldots & y_m \\ z_1 & z_2 & \ldots & z_m \end{pmatrix} \tag{4.8}$$

is the shape matrix.

The factorization method of Tomasi and Kanade [108] is based on the observation that the registered measurement matrix $\mathcal{W}$ is at most rank 3. For noisy registered observation matrix, the rank property is not correct anymore but a possible solution to the shape and rotation matrices can be recovered using the three greatest singular values of the noisy registered measurement matrix, thus, factoring it into (using singular value decomposition)

$$\mathcal{W} = \mathcal{U}\mathcal{S}^{1/2}\mathcal{S}^{1/2}\mathcal{V}^T = \hat{\mathcal{M}}\hat{\mathcal{S}}. \tag{4.9}$$

Note that the matrices $\hat{\mathcal{M}}$ and $\hat{\mathcal{S}}$ are determined up to a linear transformation, i.e.,

$$\hat{\mathcal{M}}\hat{\mathcal{S}} = (\hat{\mathcal{M}}'\mathcal{A}^{-1})(\mathcal{A}\hat{\mathcal{S}}'),$$

and $\hat{\mathcal{M}}'$ and $\hat{\mathcal{S}}'$ are also possible solutions to the motion and shape matrices. A normalization can be done by using the orthographic projection constraints,i.e., the fact that the rotation matrices are orthonormal yielding $3f$ nonlinear constraints on the vectors $\boldsymbol{a}$ and $\boldsymbol{b}$ for each image. The factorization method has been extended to weak perspective and paraperspective projections [85] as well as to line features [90].

4.2 Transfer-Based Methods

The computer vision techniques for image-based rendering without actual three-dimensional reconstruction proposed by Laveau and Faugeras [57], Seitz and Dyer [95], Kutulakos and Vallino [56], and Avidan and Shashua [3] use only a discrete (and possibly small) set of views among which point correspondences have been established by feature tracking or conventional stereo matching. These approaches are related to the classical problem of *transfer* in photogrammetry: given the image positions of a number of tie points in a set of reference images and in a new image, and given the image positions of a ground point in the reference images, predict the position of that point in the new image [5] (Figure 4.1).

In the projective case, Laveau and Faugeras [57] have proposed to first estimate the pairwise epipolar geometry (i.e., weak calibration, see Chapter 2) between the set of reference views, then reproject the scene points into a new image. The new image is specified by the positions of the new optical center in two reference images (i.e., the epipoles) and the position of four reference points in the new image. By definition, the epipolar geometry constrains the possible reprojections of points in the reference images. In the new image, the projection of the scene point is at the intersection of two epipolar lines associated with the point and two reference images (Figure 4.1).

Once the feature points have been reprojected, realistic rendering is achieved using classical computer graphics techniques such as ray tracing and texture mapping ([41, Chapters 13, 14]). Laveau and Faugeras have developed a three-dimensional ray tracing method to handle occlusions without direct geometric reconstruction. For some cases, this method may require more than two images to fully solve visibility problems.

The main contribution of this work is the demonstration that a virtual view can be rendered without any three-dimensional reconstruction. However, since they do not enforce the

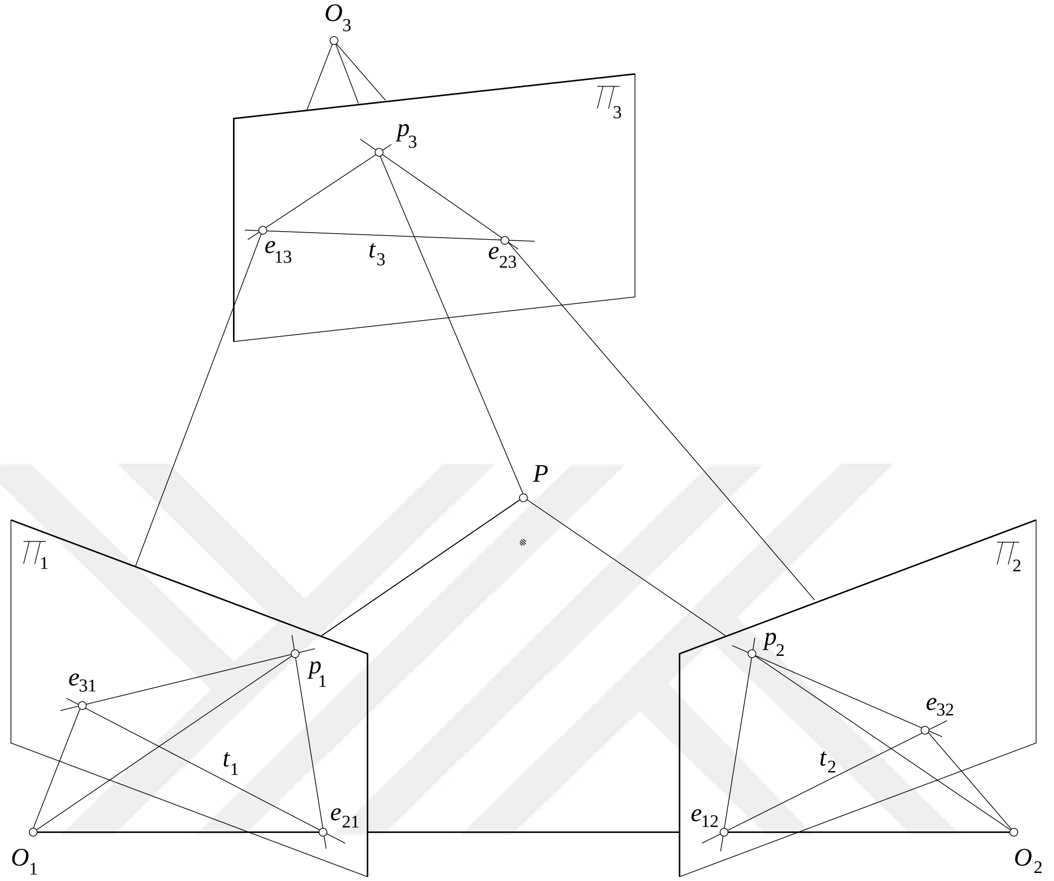


Figure 4.1 Transferring the images of the scene point P in two reference images p_3 into a third one as the intersection of the corresponding epipolar lines.

Euclidean constraints, the rendered images are ambiguous (i.e., they are in general separated from the "correct" pictures by arbitrary planar projective transformations) unless additional scene constraints are taken into account.

Related methods have been proposed by several authors in both the affine and projective cases [3, 56, 95, 73]. For three images taken by a calibrated pinhole camera, Avidan and Shashua [3] have used *trifocal tensors* [100] to constrain the reprojection of the points in these images. In other words, the three images of a scene point obey certain multilinear matching constraints captured by the trifocal tensor that can be constructed from three images in a linear fashion. This tensor can then be used to construct a general warping function from the reference images to the virtual image. This method can be used to render new images either from two or three reference pictures by setting the virtual camera position and the orientation. Even though no three-dimensional reconstruction is performed, the small rotations between the cameras used at modeling time are actually recovered (motion estimation). An advantage of this approach is that using trilinear tensors ensures that there is no degenerate virtual camera configuration.

Kutulakos and Vallino in [56] have proposed another transfer method based on the affine projection model. They have observed that, given two or more images of four points in the scene, the affine coordinates of another point can be extracted in a linear fashion without any three-dimensional information about the camera position or the reference points in the scene. In fact, when the image of the four reference points are known in the image plane, the projection of any other point is given by

$$\begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} u_1 & u_2 & u_3 & u_0 \\ v_1 & v_2 & v_3 & v_0 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix},$$

where $(u_0, v_0)^T$, $(u_1, v_1)^T$, $(u_2, v_2)^T$, $(u_3, v_3)^T$ are the projections of four reference points and $(x, y, z, 1)^T$ is the homogeneous vector of the point's affine coordinates. This equation does not involve any motion parameters but implicitly reconstructs the scene up to an affine transformation. Once again a new view is defined by the image positions of the reference points.

For more restricted virtual camera motions, Seitz and Dyer [96] use a technique called view morphing to render novel images of a rigid scene from two images. Their method assumes that the virtual camera center lies on the baseline (i.e., the line connecting the centers of the two camera associated with the reference images). The method works on point correspondences which are given a priori and involves first rectification of the two reference images so that the image planes are parallel, then linear interpolation of the new image from the rectified reference images, and finally inverse rectification of the morphed image. The new image will be physically valid in the sense that the virtual camera preserves rigid transformations. This method is also essentially based on the transfer idea in that when the reference images are rectified, any virtual image whose center lies on the baseline is a linear combination of two reference images.

4.3 Light-Field Methods

The computer graphics techniques developed by Chen [9], Gortler, Grzeszczuk, Szeliski and Cohen [30], Levoy and Hanrahan [61], and McMillan and Bishop [74] are based on the idea that the set of all visual rays (light field) is four-dimensional, and can thus be characterized from a two-dimensional sample of two-dimensional images of a rigid scene. The light field is a simplified form of the *plenoptic function* [1]. The plenoptic function at a point (x, y, z) determines the radiant energy emitted from the scene along a ray passing through this point. More precisely, the plenoptic function is normally written as

$$p = P(x, y, z, \theta, \phi, \lambda, t)$$

where $(x, y, z)^T$ is the position of the point of interest (for example the position of the optical center of a camera), θ and ϕ are angular coordinates defining the orientation of the ray, λ is the wavelength at which the radiance is measured and t is the time. When a snapshot of the function is considered (which is the case for a static picture), time and wavelength are fixed, then the number of parameters drop to 5. Furthermore, when we assume that the radiance is independent of the position of the point of interest along the corresponding rays (which is the case for a camera taking images of a scene), the parameter space (or the light field) becomes the four-dimensional set of light rays.

The image-based rendering approaches based on the light field approximate the plenoptic function and create a lookup table from a large set of images representing the scene. At synthesis time a virtual camera view is set and the view is interpolated from the lookup table. Essentially, image morphing can be viewed as interpolation of the visual rays (or the light field or the plenoptic function) arriving at a particular pixel. Then, any method based on pixel interpolation must answer two questions: how to establish pixel correspondences between images and how to define a new camera position. The methods described in this section will address these question. Furthermore, the quality of the synthesized images will depend on the reference images used. The closer the virtual view is to the reference images, the better the quality of the synthesized images. Therefore, these methods tend to utilize a large number of images.

As we have already discussed, the light-field methods do not rely on particular image features, and work instead with the image pixels which is one of the main distinctions between the methods discussed in this section and the transfer-based methods of the previous section. The transfer-based methods use a set of features and synthesize new images by first predicting the images of these features and texture mapping the rest of the picture. The transfer-based methods in principle assume that the scene is composed of Lambertian surfaces (texture mapping will not work perfectly for other types of models) whereas the light-field methods do not assume anything about the reflectance properties of the scene.

The first method we will review is the view interpolation method of Chen and Williams [10] which generates intermediate images by morphing images stored at nearby viewpoints. The morphing process requires a pixel correspondence map that is computed using a priori known range data and the camera transformations. When the virtual camera is set, reference images at nearby viewpoints in the image database are retrieved and the new image is morphed from these reference images. Since morphing is a linear interpolation of the visual rays arriving at a particular pixel, this method will generate physically correct images when the virtual image plane is parallel to the baseline connecting the two reference images.

Despite similarities in the way a new view is generated, the view morphing technique of Seitz and Dyer [95] differs from the view interpolation method of Chen and Williams in several ways. First, Seitz and Dyer do not require a known range map. Instead, they use a set of feature correspondences to rectify the reference images so that pixel correspondences can be established. A second difference is in they way the virtual camera position is set by these two methods. The view interpolation method of Chen and Williams assumes that the camera motions are known and the reference images are retrieved from the database accordingly, whereas Seitz and Dyer render new images which are restricted to lie on the baseline of two reference frames. So, the view morphing technique of Seitz and Dyer is closely related to the transfer-based methods.

A similar technique has appeared in Chen's image-based rendering system, called QuickTimeVR [9], where the scene is represented by a set of cylindrical panoramas instead of regular intensity images. A new view is synthesized by warping the cylindrical image whose center of projection is closest to that of the new view. Cylindrical panaromas represents a restricted view of the scene, therefore, the virtual camera position is in principle restricted.

In the methods we have discussed so far, the plenoptic function (or light field) concept have been used implicitly without actually characterizing it from the input images. Instead, these methods keep the images in the database and morph the reference images closest to the desired

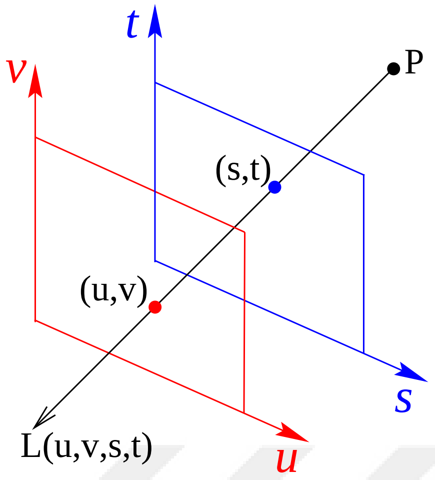


Figure 4.2 Parameterization of the light field.

view. As we have mentioned already, image morphing is the interpolation of the visual rays arriving at a particular pixel. The methods we will review below actually recover the light field explicitly and use it to render new images instead of keeping the images as a representation of the plenoptic function.

The technique developed by Levoy and Hanrahan [61], called light-field rendering, parameterizes the visual rays with four parameters with L(u, v, s, t) (see Figure 4.2). To synthesize a virtual view, they describe an efficient method to compute the ray parameters for the new view. These parameters are used to find the corresponding rays in the lookup table for the new view. The storage requirement for this method can be very large.

A similar representation by Gortler, Grzeszczuk, Szeliski and Cohen [30] is called the Lumigraph (a representation of the four-dimensional light field). It uses the same parameterization for the visual rays (see Figure 4.2). The Lumigraph is a continuous four-dimensional space. A discrete set of input images yields a discrete Lumigraph. So the continuous Lumigraph needs to be reconstructed from the discrete Lumigraph to synthesize intermediate views. Let us first assume that the discrete Lumigraph is placed on grid points with data value $x(i, j, p, q)$. Furthermore, let us associate the basis functions $B_{i,j,p,q}$ with each grid point to reconstruct the continuous Lumigraph as the linear sum (considering that we have M subdivisions in s and t

dimensions and N subdivisions in u and v dimensions)

$$\tilde{L}(s,t,u,v) = \sum_{i=0}^{M}\sum_{j=0}^{M}\sum_{p=0}^{N}\sum_{q=0}^{N} x(i,j,p,q)B_{i,j,p,q}(s,t,u,v),$$

where $\tilde{L}$ is the continuous Lumigraph. In [30], a quadrilinear basis function whose value is 1 at the grid point and 0 at all neighboring grid points.

Even though it can generate very realistic images, like light-field rendering, the Lumigraph requires a large number of images. This is a characteristic of light-field methods. On the other hand, the transfer-based methods require far less input images at the expense of image quality.

CHAPTER 5

PARAMETERIZED IMAGE VARIETIES

The set of all images of a rigid scene observed by a mobile camera with fixed intrinsic parameters forms a six-dimensional variety. This is intuitively obvious since, for a mobile camera with fixed internal parameters, the image of a rigid scene is completely determined by the six parameters defining the position and orientation of the camera.

We propose to construct an explicit representation of V, the *Parameterized Image Variety* (or PIV) from a set of point and line correspondences established across a sequence of weak perspective or paraperspective images. The PIV associated with a rigid scene is parameterized by the position of three image points, and it can be used to synthesize new pictures of this scene from arbitrary viewpoints, with applications in the virtual reality domain.

Like other recent approaches to image synthesis without explicit three-dimensional models [57, 95, 56, 40], our method completely by-passes the estimation of the motion and structure parameters, and works fully in image space. Previous techniques exploit the affine or projective structure of images [55, 19, 39] but ignore the Euclidean constraints associated with real cameras; consequently, as noted in [57], the synthesized pictures may be subjected to affine or projective deformations. Our method takes Euclidean constraints into account explicitly and outputs correct images.

In the remainder of this chapter, we first review the various projection models used in our approach and show how they determine the set of all possible images of a rigid scene. Then a parameterization of the set of all images of a fixed set of points and lines based on images of three points is described. We also show how both the point and line PIVs can be integrated in a general framework for image synthesis without explicit three-dimensional models, and present experiments to validate our approach.

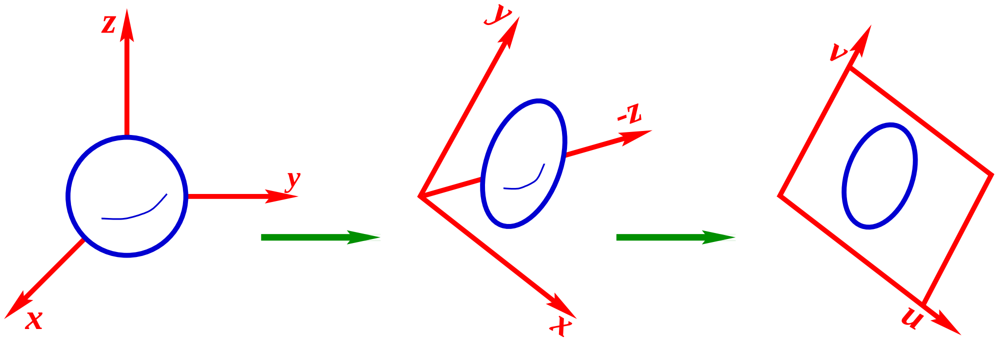


Figure 5.1 Affine camera model.

5.1 The Set of Images of a Rigid Scene

The set of all images of m points and n lines (corresponding to the two coordinates defining each point and line) rigidly attached to each other can be embedded in a $2(m+n)$-dimensional vector space E, but it forms in fact a low-dimensional subspace V of E: as will be shown below, V is a variety of E (i.e., a subspace defined by polynomial equations) of dimension eight for affine cameras, and an eleven-dimensional variety for projective cameras. But V is only a *six-dimensional* variety of E for weak perspective, paraperspective [82, 2], and full perspective cameras.

5.1.1 Affine Cameras

Let us first consider an *affine* camera observing some 3D scene, i.e., let us assume that the scene, represented in some fixed world coordinate system, is first submitted to a 3D affine transformation and then orthographically projected onto the image plane of the camera (see Figure 5.1).

We denote the coordinate vector of a scene point P in the world coordinate system by $\boldsymbol{P} = (x, y, z)^T$. Let $\boldsymbol{p} = (u, v)^T$ denote the coordinate vector of the projection p of P onto the image plane, the affine camera model can be written as

$$\boldsymbol{p} = \mathcal{M}\boldsymbol{P} + \boldsymbol{p}_0 \tag{5.1}$$

with

$$\mathcal{M} = \begin{pmatrix} \boldsymbol{a}^T \\ \boldsymbol{b}^T \end{pmatrix} \quad \text{and} \quad \boldsymbol{p}_0 = \begin{pmatrix} u_0 \\ v_0 \end{pmatrix}.$$

Note that $\boldsymbol{p}_0$ is the image of the origin of the world coordinate system.

Suppose we observe a fixed set of points P_i $(i = 1, .., m)$ with coordinate vectors $\boldsymbol{P}_i$, and let $\boldsymbol{p}_i$ denote the coordinate vectors of the corresponding image points. Writing (5.1) for all the scene points yields

$$\begin{pmatrix} \boldsymbol{u} \\ \boldsymbol{v} \end{pmatrix} = \begin{pmatrix} \boldsymbol{x} & \boldsymbol{y} & \boldsymbol{z} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{1} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{x} & \boldsymbol{y} & \boldsymbol{z} & \boldsymbol{0} & \boldsymbol{1} \end{pmatrix} \begin{pmatrix} \boldsymbol{a} \\ \boldsymbol{b} \\ \boldsymbol{p}_0 \end{pmatrix},$$

where

$$\boldsymbol{u} = \begin{pmatrix} u_1 \\ \dots \\ u_m \end{pmatrix}, \boldsymbol{v} = \begin{pmatrix} v_1 \\ \dots \\ v_m \end{pmatrix}, \boldsymbol{x} = \begin{pmatrix} x_1 \\ \dots \\ x_m \end{pmatrix}, \boldsymbol{y} = \begin{pmatrix} y_1 \\ \dots \\ y_m \end{pmatrix}, \boldsymbol{z} = \begin{pmatrix} z_1 \\ \dots \\ z_m \end{pmatrix}, \boldsymbol{1} = \begin{pmatrix} 1 \\ \dots \\ 1 \end{pmatrix}, \boldsymbol{0} = \begin{pmatrix} 0 \\ \dots \\ 0 \end{pmatrix}.$$

In other words, the set of images of m points is an eight-dimensional vector space V_p embedded in $\mathbb{R}^{2m}$. Given $f \geq 2$ views of the m points, a basis for this vector space can be identified by performing the singular value decomposition of the $2m \times f$ matrix

$$\mathcal{O} = \begin{pmatrix} \boldsymbol{u}^{(1)} & \dots & \boldsymbol{u}^{(f)} \\ \boldsymbol{v}^{(1)} & \dots & \boldsymbol{v}^{(f)} \end{pmatrix},$$

where $\boldsymbol{u}^{(i)}$ and $\boldsymbol{v}^{(i)}$ $(i = 1, .., f)$ denote the vectors $\boldsymbol{u}$ and $\boldsymbol{v}$ associated with frame number i. Once a basis for V has been constructed, new images can be constructed by assigning arbitrary values to $\boldsymbol{a}$, $\boldsymbol{b}$ and $\boldsymbol{p}_0$.

An alternative method is to pick four scene points, say A_0, A_1, A_2 and A_3 as an affine world basis: let α, β and γ denote the affine coordinates of a point P in this basis, so that

$$P = (1 - \alpha - \beta - \gamma)A_0 + \alpha A_1 + \beta A_2 + \gamma A_3.$$

If a_i denotes the image of the point A_i $(i = 0, 1, 2, 3)$ and p denotes the image of the point P, we have

$$p = (1 - \alpha - \beta - \gamma)a_0 + \alpha a_1 + \beta a_2 + \gamma a_3, \tag{5.2}$$

and it follows that α, β and γ can be computed from $f \geq 2$ images through linear least squares. Once these values are known, new images can be generated by specifying the image positions of the points a_i and using (5.2) to compute all the other point positions. This is essentially the method proposed by Kutulakos and Vallino [56], and its roots can be found in the classical paper by Koenderink and Van Doorn on affine structure from motion [55].

Suppose now that we observe a point $\boldsymbol{P}$ with a fixed set of cameras $\mathcal{M}^{(i)}$ $(i = 1, .., f)$, and let $\boldsymbol{p}^{(i)}$ denote the image of $\boldsymbol{P}$ corresponding to the projection matrix $\mathcal{M}^{(i)}$. We have

$$\boldsymbol{t} = \boldsymbol{o} + x\boldsymbol{c}_1 + y\boldsymbol{c}_2 + z\boldsymbol{c}_3,$$

where

$$\boldsymbol{t} = \begin{pmatrix} \boldsymbol{p}^{(1)} \\ \dots \\ \boldsymbol{p}^{(f)} \end{pmatrix}, \boldsymbol{o} = \begin{pmatrix} \boldsymbol{p}_0^{(1)} \\ \dots \\ \boldsymbol{p}_0^{(f)} \end{pmatrix}, \boldsymbol{c}_1 = \begin{pmatrix} a_1^{(1)} \\ b_1^{(1)} \\ \dots \\ a_1^{(f)} \\ b_1^{(f)} \end{pmatrix}, \boldsymbol{c}_2 = \begin{pmatrix} a_2^{(1)} \\ b_2^{(1)} \\ \dots \\ a_2^{(f)} \\ b_2^{(f)} \end{pmatrix}, \boldsymbol{c}_3 = \begin{pmatrix} a_3^{(1)} \\ b_3^{(1)} \\ \dots \\ a_3^{(f)} \\ b_3^{(f)} \end{pmatrix},$$

which shows that the set of images taken by the cameras is the three-dimensional affine subspace of $\mathbb{R}^{2f}$ spanned by the point $\boldsymbol{o}$ and the vectors $\boldsymbol{c}_1$, $\boldsymbol{c}_2$ and $\boldsymbol{c}_3$.

This is of course just a re-statement of several well-known results: in particular, if the center of mass of the observed feature points is chosen as the origin of the image coordinate system, it can be shown that an affine image is the linear combination of three model images [112], that the measurement matrix

$$\mathcal{O}' = \begin{pmatrix} u_1^{(1)} & \dots & u_m^{(1)} \\ \dots & \dots & \dots \\ u_1^{(f)} & \dots & u_m^{(f)} \\ v_1^{(1)} & \dots & v_m^{(1)} \\ \dots & \dots & \dots \\ v_1^{(f)} & \dots & v_m^{(f)} \end{pmatrix}$$

has rank 3 [108, 85] (note that $\mathcal{O} \neq \mathcal{O}'$), and that the image trajectories of a scene point are linear combinations of the trajectories of three reference points [114].

Let us now consider a line Δ parameterized by its direction $\boldsymbol{\Omega}$ and the vector $\boldsymbol{D}$ joining the origin of the world coordinate system to its projection onto Δ (see Figure 5.2). We can parameterize the image δ of Δ onto the image plane by the image vector $\boldsymbol{d}$ that joins the origin

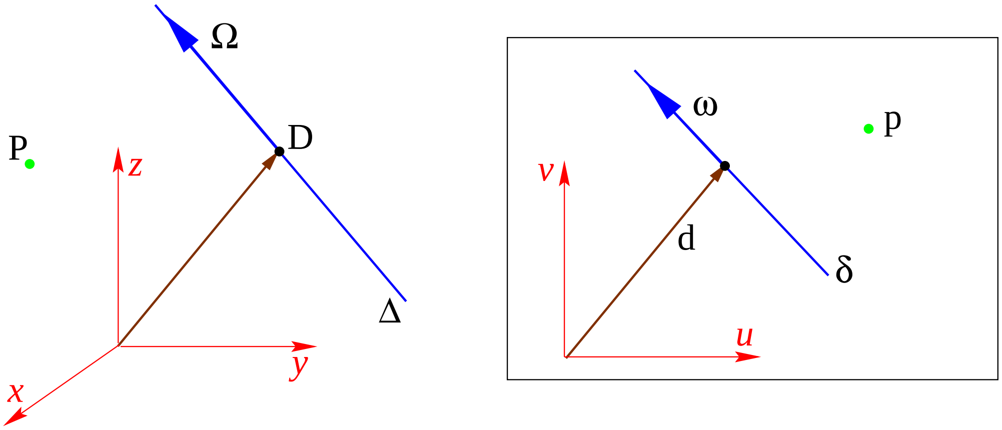


Figure 5.2 Geometric setting for points and lines in 3D.

of the image coordinate system to its orthogonal projection onto δ. This vector is defined by the two constraints

$$\begin{cases} \boldsymbol{d} \cdot (\mathcal{M}\boldsymbol{\Omega}) = 0, \\ \boldsymbol{d} \cdot (\mathcal{M}\boldsymbol{D} + \boldsymbol{p}_0) = |\boldsymbol{d}|^2. \end{cases} \tag{5.3}$$

It follows that the set of all affine images of n lines is an eight-dimensional variety V_l (i.e., a subspace defined by polynomial constraints) embedded in $\mathbb{R}^{2n}$ and defined by the $2n$ equations in $2n+8$ unknowns (namely, the coordinates of the vectors $\boldsymbol{d}_i$ $(i = 1, .., n)$ associated with the n lines and the coordinates of the vectors $\boldsymbol{a}$, $\boldsymbol{b}$ and $\boldsymbol{p}_0$) obtained by writing (5.3) for the n lines.

Combining the above results for points and lines we conclude that the set of all affine images of m points and n lines is an eight-dimensional variety V embedded in $\mathbb{R}^{2(m+n)}$.

5.1.2 Euclidean Constraints

Let us now assume a fixed world coordinate and suppose that the camera observing the scene has been calibrated so that image points are represented by their normalized coordinate vectors. Here we are going to look at three Euclidean versions of affine cameras to approximate perspective projection (see Figure 5.3).

Observing that for an orthographic camera (see Figure 5.3) $\boldsymbol{a}^T$ and $\boldsymbol{b}^T$ are the first two rows of a rotation matrix shows that an orthographic camera is an affine camera with the additional constraints

$$|\boldsymbol{a}|^2 = |\boldsymbol{b}|^2 = 1 \quad \text{and} \quad \boldsymbol{a} \cdot \boldsymbol{b} = 0. \tag{5.4}$$

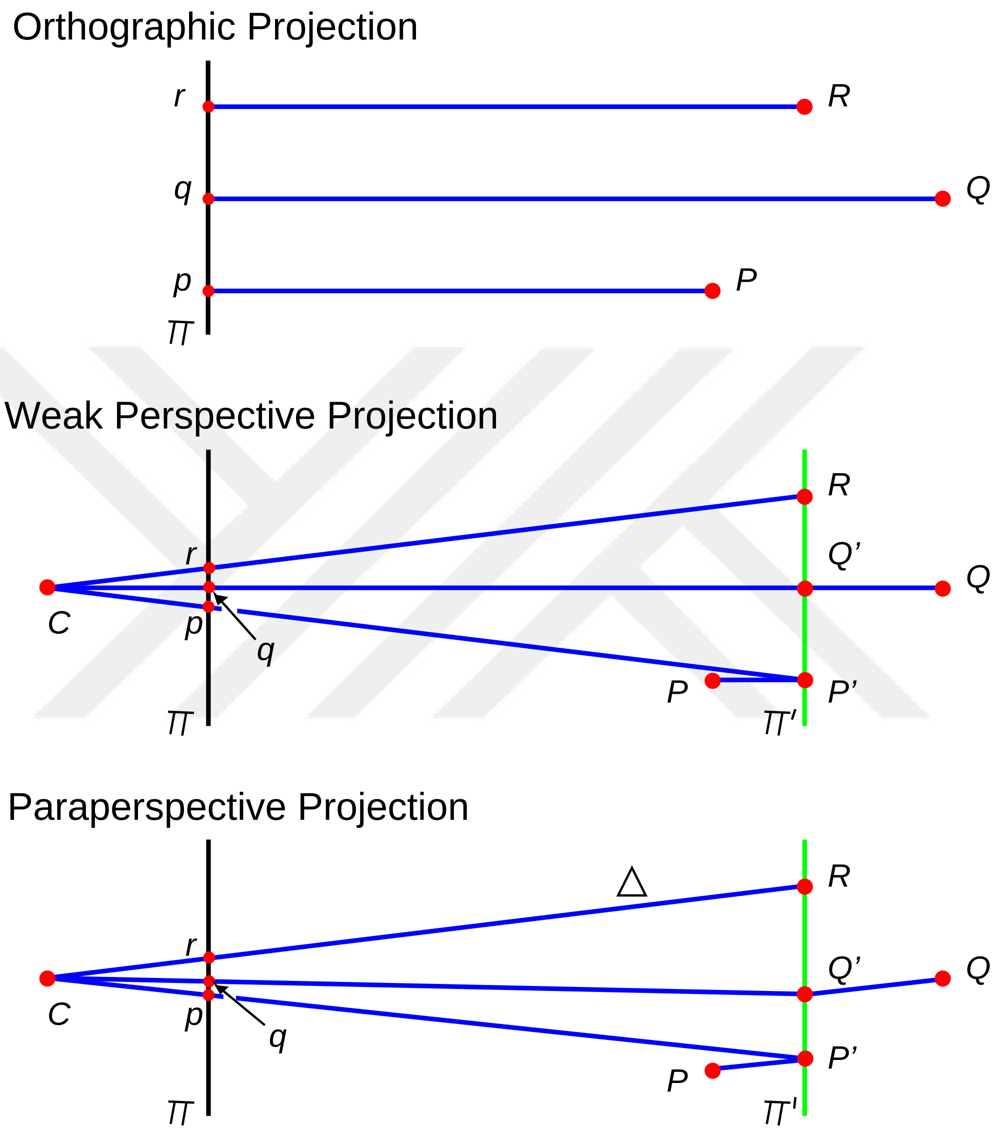


Figure 5.3 Euclidean projection models: In orthographic projection a point is projected onto the image plane orthographically. In weak perspective projection a point is first projected onto a reference plane orthographically and followed by a perspective projection. Finally, a paraperspective projection is obtained by first projecting the scene points onto the reference plane parallel to the line joining a reference point on the reference plane and followed by a perspective projection. Amomg these projection models, paraperspective projection is the best approximation to the perspective projection.

In turn, a weak perspective (or scaled orthographic) camera is an affine camera with the two constraints

$$|\boldsymbol{a}|^2 = |\boldsymbol{b}|^2 \quad \text{and} \quad \boldsymbol{a} \cdot \boldsymbol{b} = 0. \tag{5.5}$$

Finally, as shown in Appendix C, a paraperspective camera is an affine camera with the constraints

$$\boldsymbol{a} \cdot \boldsymbol{b} = \frac{u_r v_r}{2(1+u_r^2)}|\boldsymbol{a}|^2 + \frac{u_r v_r}{2(1+v_r^2)}|\boldsymbol{b}|^2 \quad \text{and} \quad (1+v_r^2)|\boldsymbol{a}|^2 = (1+u_r^2)|\boldsymbol{b}|^2, \tag{5.6}$$

where (u_r, v_r) denote the coordinates of the image of the reference point associated with the scene (see [85] for the use of similar constraints in Euclidean shape and motion recovery). It should be noted that under this projection model, the vectors $\boldsymbol{a}$ and $\boldsymbol{b}$ do not form a basis of the image plane. Instead, they form a basis of the vector plane orthogonal to the line joining the optical center of the camera to the reference point (Appendix C).

As expected, the paraperspective constraints reduce to the weak perspective constraints when $u_r = v_r = 0$, and the weak perspective constraints reduce in turn to the orthographic constraints when $|\boldsymbol{a}^2| = |\boldsymbol{b}^2| = 1$.

As shown earlier, the set of affine images of a fixed scene is an eight-dimensional vector space. If we restrict our attention to orthographic, weak perspective, or paraperspective cameras, the set of images becomes the 5D or 6D variety of this 8D vector space defined respectively by the constraints (5.4), (5.5), or (5.6). It should be noted that the case of perspective and projective cameras is similar: this time however, the projection equations involve a 3 × 4 matrix. They are bilinear in the image coordinates and the rows of this matrix, and define an 11D variety. The Euclidean constraints associated with perspective cameras yield five additional quadratic constraints, which again reduce the set of images of a scene to a 6D variety.

5.2 A New Approach to Image-Based Rendering

We propose a parameterization of the six-dimensional variety formed by the weak perspective images of m points and n lines in terms of the image positions of three points in the scene. This parameterization defines the *Parameterized Image Variety* (or *PIV*) associated with the scene.

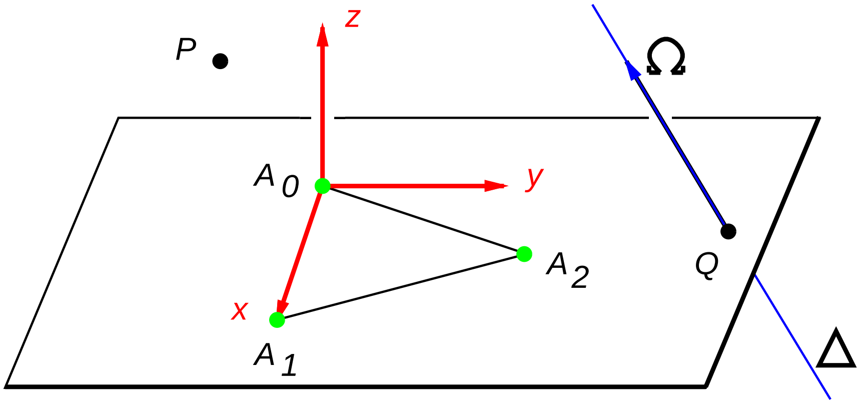


Figure 5.4 Geometric setup for the point and the line PIVs.

Let us suppose that we observe three points A_0, A_1, A_2 whose images are not collinear. We can choose (without loss of generality) a Euclidean coordinate system such that the coordinate vectors of the four points in this system are $\boldsymbol{A}_0 = (0,0,0)^T$, $\boldsymbol{A}_1 = (1,0,0)^T$, $\boldsymbol{A}_2 = (p,q,0)^T$ (Figure 5.4). The values of p and q are nonzero but unknown.

These points will be used to parameterize the PIV in the next two sections. We start by constructing the point PIV, since the equations derived in this case will also be used to construct the line PIV in Section Section 5.2.2. In both cases, the image rendering process is divided into two steps: a training stage, where the parameters of the PIV are estimated from an input image stream, and a synthesis stage, where the parameters of the points and/or lines of interest are estimated from the position of the three reference points in the new image. A potential advantage of lines over points is that they can be located very accurately in edge maps using least squares and they can facilitate the production of a good triangulation.

5.2.1 The Point PIV

Let us consider a point P and its projection p in the image plane, and denote by $\boldsymbol{P} = (x,y,z)^T$ and $\boldsymbol{p} = (u,v)^T$ their coordinate vectors. The values of (x,y,z) are of course unknown. We will also assume that $u_0 = v_0 = 0$ since we can go back to the general case via an image translation. From here on, we focus on the weak perspective projection case. The paraperspective projection case is considered later.

Applying (5.1) to A_1, A_2 and P yields

$$\boldsymbol{u} \overset{\text{def}}{=} \begin{pmatrix} u_1 \\ u_2 \\ u \end{pmatrix} = \mathcal{A}\boldsymbol{a}, \quad \text{and} \quad \boldsymbol{v} \overset{\text{def}}{=} \begin{pmatrix} v_1 \\ v_2 \\ v \end{pmatrix} = \mathcal{A}\boldsymbol{b}, \tag{5.7}$$

where

$$\mathcal{A} \overset{\text{def}}{=} \begin{pmatrix} \boldsymbol{A}_1^T \\ \boldsymbol{A}_2^T \\ \boldsymbol{P}^T \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ p & q & 0 \\ x & y & z \end{pmatrix}.$$

In turn, this implies that

$$\boldsymbol{a} = \mathcal{B}\boldsymbol{u}, \quad \text{and} \quad \boldsymbol{b} = \mathcal{B}\boldsymbol{v}, \tag{5.8}$$

where

$$\mathcal{B} \overset{\text{def}}{=} \mathcal{A}^{-1} = \begin{pmatrix} 1 & 0 & 0 \\ \lambda & \mu & 0 \\ \alpha/z & \beta/z & 1/z \end{pmatrix},$$

and

$$\lambda = -\frac{p}{q}, \ \mu = \frac{1}{q}, \ \alpha = -(x + \lambda y), \ \beta = -\mu y.$$

Using (5.8) and letting $\mathcal{C} \overset{\text{def}}{=} z^2 \mathcal{B}^T \mathcal{B}$, the weak perspective constraints (5.5) can be rewritten as

$$\begin{cases} \boldsymbol{u}^T \mathcal{C} \boldsymbol{u} - \boldsymbol{v}^T \mathcal{C} \boldsymbol{v} = 0, \\ \boldsymbol{u}^T \mathcal{C} \boldsymbol{v} = 0, \end{cases} \tag{5.9}$$

with

$$\mathcal{C} = \begin{pmatrix} \xi_1 & \xi_2 & \alpha \\ \xi_2 & \xi_3 & \beta \\ \alpha & \beta & 1 \end{pmatrix}, \quad \text{and} \quad \begin{cases} \xi_1 = (1 + \lambda^2) z^2 + \alpha^2, \\ \xi_2 = \lambda \mu z^2 + \alpha \beta, \\ \xi_3 = \mu^2 z^2 + \beta^2. \end{cases}$$

Equation (5.9) defines a pair of linear constraints on the coefficients ξ_i $(i = 1, 2, 3)$, α and β; they can be rewritten as

$$\mathcal{N}_{\mathrm{d}} \xi = 0, \tag{5.10}$$

where

$$\mathcal{N}_{\mathrm{d}} \stackrel{\text{def}}{=} \begin{pmatrix} \boldsymbol{d}_1^T - \boldsymbol{d}_2^T \\ \boldsymbol{d}^T \end{pmatrix} \tag{5.11}$$

with

$$\begin{cases} \boldsymbol{d}_1^T \stackrel{\text{def}}{=} (u_1^2, 2u_1u_2, u_2^2, 2u_1u, 2u_2u, u^2), \\ \boldsymbol{d}_2^T \stackrel{\text{def}}{=} (v_1^2, 2v_1v_2, v_2^2, 2v_1v, 2v_2v, v^2), \\ \boldsymbol{d}^T \stackrel{\text{def}}{=} (u_1v_1, u_1v_2 + u_2v_1, u_2v_2, u_1v + uv_1, u_2v + uv_2, uv), \end{cases}$$

and $\boldsymbol{\xi} \stackrel{\text{def}}{=} (\xi_1, \xi_2, \xi_3, \alpha, \beta, 1)^T$.

When the four points A_0, A_1, A_2, and P are rigidly attached to each other, the five structure coefficients ξ_1, ξ_2, ξ_3, α and β are fixed. For a rigid scene formed by m points, choosing three of the points as a reference triangle and writing (5.10) for the remaining ones yields a set of $2m - 6$ quadratic equations in $2m$ unknowns, which do indeed define a parameterization of the set of all weak perspective images of the scenes. This is the PIV.

Note that the weak perspective constraints (5.10) are linear in the five structure coefficients. Thus, given a collection of images and point correspondences, we can compute these coefficients through linear least squares. We describe in Section 5.2.1.1 how to estimate these coefficients using linear least squares and we will discuss alternative estimation techniques in Chapter 6.

Once the vector $\boldsymbol{\xi}$ has been estimated, we can specify arbitrary image positions for our three reference points. Equation (5.10) yields two quadratic constraints on the two unknowns u and v. Although this system should *a priori* admit four solutions, it admits exactly two real solutions: indeed, if we let

$$\boldsymbol{u}_2 \stackrel{\text{def}}{=} \begin{pmatrix} u_1 \\ u_2 \end{pmatrix}, \quad \boldsymbol{v}_2 \stackrel{\text{def}}{=} \begin{pmatrix} v_1 \\ v_2 \end{pmatrix},$$

and

$$\mathcal{E} \stackrel{\text{def}}{=} \begin{pmatrix} \xi_1 - \alpha^2 & \xi_2 - \alpha\beta \\ \xi_2 - \alpha\beta & \xi_3 - \beta^2 \end{pmatrix} = \begin{pmatrix} (1 + \lambda^2)z^2 & \lambda\mu z^2 \\ \lambda\mu z^2 & \mu^2 z^2 \end{pmatrix},$$

then (5.5) can easily be rewritten as

$$\begin{cases} X^2 - Y^2 + e_1 - e_2 = 0, \\ 2XY + e = 0, \end{cases} \tag{5.12}$$

where

$$\begin{cases} e_1 = \boldsymbol{u}_2^T \mathcal{E} \boldsymbol{u}_2, \\ e_2 = \boldsymbol{v}_2^T \mathcal{E} \boldsymbol{v}_2, \\ e = 2\boldsymbol{u}_2^T \mathcal{E} \boldsymbol{v}_2, \end{cases} \quad \text{and} \quad \begin{cases} X = u + \alpha u_1 + \beta u_2, \\ Y = v + \alpha v_1 + \beta v_2. \end{cases}$$

To compute the solutions of (5.12), we introduce the variable Z defined by

$$\begin{cases} X^2 = \dfrac{1}{2}(Z - e_1 + e_2), \\ Y^2 = \dfrac{1}{2}(Z + e_1 - e_2). \end{cases}$$

Squaring the second equation of (5.12) and taking into account the fact that, since $Z = X^2 + Y^2$, its value is positive, yields

$$Z = \sqrt{(e_1 - e_2)^2 + e^2}. \tag{5.13}$$

In turn, this yields two opposite solutions for X and two opposite solutions for Y. Only two of the four possible pairs are solutions of the original equations, and they are determined by the fact that $2XY + e = 0$. Once X and Y have been solved for, the values of u and v are trivially obtained.

5.2.1.1 Estimation of the Structure Parameters

In order to compute the five structure parameters in $\boldsymbol{\xi}$, let us rewrite the weak perspective constraints (5.10) for all available images (i.e., for $i = 1, ..., f$):

$$\begin{cases} \boldsymbol{d}_i^T \boldsymbol{\xi} = 0, \\ \boldsymbol{e}_i^T \boldsymbol{\xi} = 0, \end{cases} \tag{5.14}$$

where

$$\boldsymbol{d}_i \overset{\text{def}}{=} \begin{pmatrix} u_{1i}^2 - v_{1i}^2 \\ 2(u_{1i}u_{2i} - v_{1i}v_{2i}) \\ u_{2i}^2 - v_{2i}^2 \\ 2(u_{1i}u_i - v_{1i}v_i) \\ 2(u_{2i}u_i - v_{2i}v_i) \\ u_i^2 - v_i^2 \end{pmatrix} \quad \text{and} \quad \boldsymbol{e}_i \overset{\text{def}}{=} \begin{pmatrix} u_{1i}v_{1i} \\ u_{1i}v_{2i} + u_{2i}v_{1i} \\ u_{2i}v_{2i} \\ u_{1i}v_i + u_iv_{1i} \\ u_{2i}v_i + u_iv_{2i} \\ u_iv_i \end{pmatrix} \tag{5.15}$$

Note that the weak perspective constraints in (5.14) are linear in the five structure coefficients. In particular, given a collection of f images of four points, we can compute these coefficients through linear least squares: let us define the $6 \times f$ data matrix

$$\mathcal{D} \overset{\text{def}}{=} (\boldsymbol{d}_1, \dots, \boldsymbol{d}_f, \boldsymbol{e}_1, \dots, \boldsymbol{e}_f), \tag{5.16}$$

where $\boldsymbol{d}_i$ and $\boldsymbol{e}_i$ are the values of the data-dependent vectors defined in (5.15) associated with frame number i $(i = 1, \dots, f)$. The linear-least-squares solution of the over-constrained linear system $\mathcal{D}^T \hat{\boldsymbol{\xi}} = 0$ is $\hat{\boldsymbol{\xi}} = \frac{1}{\chi_6}\boldsymbol{\chi}$, where $\boldsymbol{\chi} = (\chi_1, \chi_2, \chi_3, \chi_4, \chi_5, \chi_6)^T$ is the eigenvector of the 6×6 matrix $\mathcal{D}\mathcal{D}^T$ corresponding to its smallest eigenvalue. Alternatively, $\boldsymbol{\chi}$ can be computed as the sixth column of the 6×6 matrix $\mathcal{V}$ in the *singular value decomposition* (or *SVD*) $\mathcal{U}\mathcal{W}\mathcal{V}^T$ of the $f \times 6$ matrix $\mathcal{D}^T$.

5.2.2 The Line PIV

We establish the PIV associated with lines by considering the line position and the line direction separately. Line positions are parameterized using affine notions only.

5.2.2.1 Line Position

Let us now consider a line Δ and assume that its intersection with the reference plane spanned by the points A_0, A_1 and A_2 is transversal (Figure 5.4). Without loss of generality, we can parameterize this line by the affine coordinates (χ_1, χ_2) of the point Q where the line intersects the reference plane in the basis (A_0, A_1, A_2), i.e.,

$$Q = A_0 + \chi_1 A_1 + \chi_2 A_2,$$

and by the coordinate vector $\boldsymbol{\Omega} = (x, y, 1)^T$ of its direction in the Euclidean world coordinate system.

Let δ be the image of Δ, we can parameterize this line by the position of the image q of the point Q and the *unit* coordinate vector $\boldsymbol{\omega} = (\cos\theta, \sin\theta)^T$ of its direction. If we take as before a_0 as the origin of the image plane, and denote by d the distance between a_0 and δ, the equation of δ is

$$-u \sin\theta + v \cos\theta - d = 0, \tag{5.17}$$

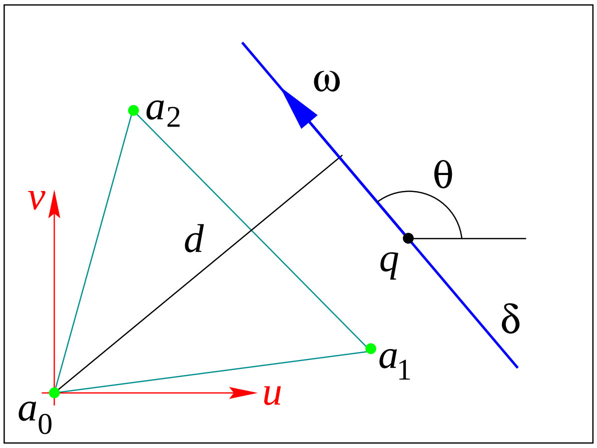


Figure 5.5 Parameterization of the line δ.

where (u, v) denote image coordinates. Since the point Q lies in the reference plane, the affine coordinates of q in the coordinate system a_0, a_1, a_2 are also χ_1 and χ_2, i.e.,

$$q = a_0 + \chi_1 a_1 + \chi_2 a_2, \tag{5.18}$$

and substituting in (5.17) yields

$$(u_1 \sin\theta - v_1 \cos\theta)\chi_1 + (u_2 \sin\theta - v_2 \cos\theta)\chi_2 + d = 0, \tag{5.19}$$

which is a linear equation in the affine coordinates χ_1 and χ_2. Given several images of the line Δ, we can thus use (5.19) to estimate χ_1 and χ_2 via linear least squares. These affine coordinates can then be used to predict the position of q in any new image once a_0, a_1 and a_2 have been specified in (5.18).

5.2.2.2 Line Orientation

Let us now turn to the prediction of θ. Note that the equations derived in Section 5.2.1 still apply when we take $\boldsymbol{P} = \boldsymbol{\Omega}$ and $\boldsymbol{p} \overset{\text{def}}{=} (u, v)^T = \rho\boldsymbol{\omega}$, where ρ is an image-dependent scale factor. Note that since the overall value of ρ is irrelevant, we can take $z = 1$ with the assumption that the line is not parallel to the plane spanned by the three reference points.

There are two differences with the point case: (a) in the line case, the five parameters ξ_i $(1 = 1, 2, 3)$, α and β are not independent since there is no z parameter to take into account, and (b) the two equations in (5.9) contain terms in ρ, that depend on the image considered. We can remove this dependency by eliminating ρ between the two equations (of course, this will introduce nonlinearities). We will deal with the redundancy of the structure parameters later in this section.

This time we use (5.13) instead of (5.10) to parameterize the PIV since it yields simpler equations. First note that

$$\begin{cases} X = \rho\cos\theta + \alpha u_1 + \beta u_2, \\ Y = \rho\sin\theta + \alpha v_1 + \beta v_2, \end{cases}$$

and therefore

$$\sin\theta X - \cos\theta Y = \zeta,$$

where

$$\zeta \overset{\text{def}}{=} (u_1\sin\theta - v_1\cos\theta)\alpha + (u_2\sin\theta - v_2\cos\theta)\beta$$

is independent of ρ. Squaring this equation and using the variable Z defined in the previous section yields

$$2\zeta^2 = Z + (e_1 - e_2)\cos 2\theta + e\sin 2\theta.$$

Evaluating ζ^2 and substituting the value of Z in (5.13) yields

$$f_1 + f_2 - (g_1 - g_2)\cos 2\theta - g\sin 2\theta = \sqrt{(e_1 - e_2)^2 + e^2}, \tag{5.20}$$

where

$$\begin{cases} f_1 \overset{\text{def}}{=} \boldsymbol{u}_2^T \mathcal{F} \boldsymbol{u}_2, \\ f_2 \overset{\text{def}}{=} \boldsymbol{v}_2^T \mathcal{F} \boldsymbol{v}_2, \end{cases} \quad \mathcal{F} \overset{\text{def}}{=} \begin{pmatrix} \alpha^2 & \alpha\beta \\ \alpha\beta & \beta^2 \end{pmatrix}, \quad \begin{cases} g \overset{\text{def}}{=} 2\boldsymbol{u}_2 \mathcal{G} \boldsymbol{v}_2, \\ g_1 \overset{\text{def}}{=} \boldsymbol{u}_2 \mathcal{G} \boldsymbol{u}_2, \\ g_2 \overset{\text{def}}{=} \boldsymbol{v}_2 \mathcal{G} \boldsymbol{v}_2, \end{cases} \quad \text{and} \quad \mathcal{G} \overset{\text{def}}{=} \begin{pmatrix} \xi_1 & \xi_2 \\ \xi_2 & \xi_3 \end{pmatrix}.$$

At this point, we still have a redundant set of five structure parameters. Let $\gamma \overset{\text{def}}{=} \sqrt{\alpha^2 + \beta^2}$, we can now define a minimal set of four structure parameters ε_1, ε_2, ε_3 and Θ by the equations

$$\begin{cases} \varepsilon_1 = (1 + \lambda^2)/\gamma^2, \\ \varepsilon_2 = \lambda\mu/\gamma^2, \\ \varepsilon_3 = \mu^2/\gamma^2, \end{cases} \quad \text{and} \quad \Theta \overset{\text{def}}{=} \text{Arg}(\alpha, \beta).$$

With this notation, (5.20) becomes

$$i_1 + i_2 - (j_1 - j_2)\cos 2\theta - j\sin 2\theta = \sqrt{(h_1 - h_2)^2 + h^2}, \tag{5.21}$$

where

$$\begin{cases} h_1 \stackrel{\text{def}}{=} \boldsymbol{u}_2^T \mathcal{H} \boldsymbol{u}_2, \\ h_2 \stackrel{\text{def}}{=} \boldsymbol{v}_2^T \mathcal{H} \boldsymbol{v}_2, \\ h \stackrel{\text{def}}{=} 2\boldsymbol{u}_2 \mathcal{H} \boldsymbol{v}_2, \end{cases} \qquad \mathcal{H} \stackrel{\text{def}}{=} \begin{pmatrix} \varepsilon_1 & \varepsilon_2 \\ \varepsilon_2 & \varepsilon_3 \end{pmatrix},$$

and

$$\begin{cases} i_1 \stackrel{\text{def}}{=} \boldsymbol{u}_2^T \mathcal{I} \boldsymbol{u}_2, \\ i_2 \stackrel{\text{def}}{=} \boldsymbol{v}_2^T \mathcal{I} \boldsymbol{v}_2, \\ i \stackrel{\text{def}}{=} 2\boldsymbol{u}_2 \mathcal{I} \boldsymbol{v}_2, \end{cases} \qquad \mathcal{I} \stackrel{\text{def}}{=} \begin{pmatrix} \frac{1}{2}(1+\cos 2\Theta) & \sin 2\Theta \\ \sin 2\Theta & \frac{1}{2}(1-\cos 2\Theta) \end{pmatrix},$$

and $j \stackrel{\text{def}}{=} h + i$, $j_1 \stackrel{\text{def}}{=} h_1 + i_1$, $j_2 \stackrel{\text{def}}{=} h_2 + i_2$.

Given a set of line correspondences, the four structure parameters ε_1, ε_2, ε_3 and Θ can estimated via non-linear least squares. At synthesis time, (5.21) becomes a trigonometric equation in 2θ, with two solutions that are easily computed in closed form. Each of this solution only determines θ up to a π ambiguity, which is immaterial in our case.

Note that directly minimizing the error corresponding to (5.21) is a biased process. A better method is to minimize

$$\sum_{i=1}^{p} (\theta_i - \hat{\theta}_i)^2,$$

where $\hat{\theta}_i$ is the line orientation empirically measured in image number i, and θ_i is the orientation predicted from (5.21). This is a constrained minimization problem, and the derivatives of θ_i with respect to the structure parameters are easily computed from the partial derivatives of (5.21) with respect to θ and these parameters.

5.2.3 From Lines to Line Segments

Infinite lines are inappropriate for direct rendering. Thus we must associate with each line a line segment. On the other hand, while lines can be localized very accurately in the input images, the position of their endpoints cannot in general be estimated reliably since most edge finders behave poorly near edge junctions. Additional line breaks can also be introduced by the program that segments edges into straight lines. Here we present a method for computing a rough estimate of the endpoints of a line from its PIV.

Let R denote one of the endpoints of the segment associated with the line Δ, and let r denote its image. We have $\boldsymbol{r} - \boldsymbol{q} = l\boldsymbol{\omega}$, and $\boldsymbol{R} - \boldsymbol{Q} = L\boldsymbol{\Omega}$, and we can once again write $\rho\boldsymbol{\omega} = \mathcal{M}\boldsymbol{\Omega}$, where this time $\rho = l/L$. The (signed) distance l is known at training time and unknown at synthesis time, while the (signed) distance L is unknown at training time and known at synthesis time. We have:

$$X^2 + Y^2 = Z = \gamma^2 T,$$

where

$$T \overset{\text{def}}{=} \frac{1}{\gamma^2}\sqrt{(e_1 - e_2)^2 + e^2} = \sqrt{(h_1 - h_2)^2 + h^2},$$

and expanding this equation yields a quadratic equation in ρ:

$$\rho^2 - 2\gamma\rho[(u_1 \cos\Theta + u_2 \sin\Theta)\cos\theta + (v_1 \cos\Theta + v_2 \sin\Theta)\sin\theta] + \gamma^2(\sigma - T) = 0. \qquad (5.22)$$

Note that we can compute γ from the vector $(\varepsilon_1, \varepsilon_2, \varepsilon_3)^T$ as

$$\gamma^2 = \frac{\varepsilon_3}{\varepsilon_1\varepsilon_3 - \varepsilon_2^2}.$$

This equation (5.22) has the form

$$\rho^2 - 2c\rho + d = 0.$$

To avoid sign problems due to the fact that Θ is only known up to a π ambiguity and that the sign of l and L cannot be determined, we square this equation and introduce the new variables $l' \overset{\text{def}}{=} l^2$ and $L' \overset{\text{def}}{=} l'^2$. We obtain

$$d^2 L'^2 + 2l'(d - 2c^2)L' + l'^2 = 0.$$

Its solutions are of course the same as before, but this equation can now be used to estimate L^2 via non-linear least squares since its coefficients only depend on 2θ, 2Θ and l^2. During image generation we estimate $|l|$ as $|\rho L|$, and give a sign to l using tracking from a real image. The value of l used at that time does not have to be a precise estimate of the actual line extent as long as it remains consistent over the synthetic frames.

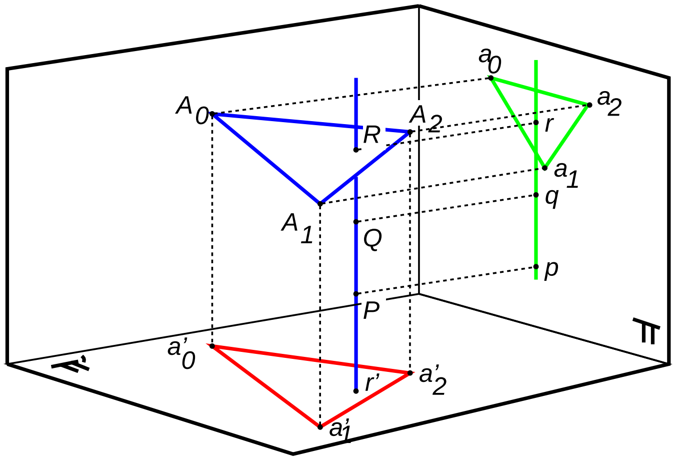


Figure 5.6 Z-buffering.

5.3 Image Synthesis

Once the PIV parameters have been estimated, the scene can be rendered from a new viewpoint by specifying interactively the image positions of the three reference points, and computing the corresponding image positions of all other points and lines. To create a shaded picture, we can construct a constrained Delaunay triangulation of these line segments and points (using, for example Shewuck's Triangle public-domain software [101]) whose vertices and edges form a superset of the input points and line segments. Texture mapping is then easily achieved by using the same triangulation in one of the input images. This section details the main stages of the rendering process.

5.3.1 Hidden-Surface Removal

Here we show how traditional z-buffer techniques can be used to perform hidden-surface elimination even though no explicit three-dimensional reconstruction is performed. Our aim here is to assign relative depth values to the vertices of the traingles obtained from points and line segments as described above.

Let Π denote the image plane of one of our input images, and Π' the image plane of our synthetic image. To render correctly two points P and Q that project onto the same point r' in the synthetic image, we must compare their depths.

Let R denote the intersection of the viewing ray joining P to Q with the plane spanned by the reference points A_0, A_1 and A_2, and let p, q, r denote the projections of P, Q and R into the reference image. Suppose for the time being that P and Q are two of the points tracked in the input image; it follows that the positions of p and q are known. The position of r is easily computed by remarking that its coordinates in the affine basis of Π formed by the projections a_0, a_1, a_2 of the reference points are the same as the coordinates of R in the affine basis formed by the points A_0, A_1, A_2 in their own plane, and thus are also the same as the coordinates of r' in the affine basis of Π' formed by the projections a'_0, a'_1, a'_2 of the reference points.

The ratio of the depths of P and Q relative to the plane Π is simply the ratio $\overline{pr}/\overline{qr}$. Not that deciding which point is actually visible requires orienting the line supporting the points p, q, r, which is simply the epipolar line associated with the point r'. A coherent orientation should be chosen for all epipolar lines (this is easy since they are all parallel to each other). Note that this does not require explicitly computing the epipolar geometry: given a first point p', one can orient the line pr, then use the same orientation for all other point correspondences. The orientations chosen should also be consistent over successive frames, but this is not a problem since the direction of the epipolar lines changes slowly from one frame to the next, and one can simply choose the new orientation so that it makes an acute angle with the previous one.

5.3.2 Rendering

Given an input triangulation, the entire scene can now be rendered using z-buffering as follows:

1. pick the correct orientation for the epipolar lines (using one of the point correspondences and the previous orientation);

2. compute, for each of the data points P, the position of r in the reference image and the "depth" $\overline{pr}$ and store it as its "z" coordinate;

3. render the triangles forming the scenes using a z-buffer algorithm with orthographic projection along the z-axis.

Texture mapping is easily incorporated in the process.

It should be noted that this process can generate two families of images corresponding to the initial choice of epipolar line orientation. The choice can be made by the user during interactive image synthesis.

5.4 The Paraperspective Case

In this section, we extend the weak perspective case to the paraperspective case. We will choose $\boldsymbol{a}_0$ as the reference point for the paraperspective projection in the rest of this discussion, i.e., $u_r = u_0$ and $v_r = v_0$ in (5.6).

5.4.1 The Paraperspective Point PIV

The construction of the paraperspective point PIV follows the same steps as in the weak perspective case: using (5.6) yields

$$\begin{cases} \psi_1 \boldsymbol{u}^T \mathcal{C} \boldsymbol{u} - \psi_2 \boldsymbol{v}^T \mathcal{C} \boldsymbol{v} = 0, \\ 2\boldsymbol{u}^T \mathcal{C} \boldsymbol{v} - \phi_1 \boldsymbol{u}^T \mathcal{C} \boldsymbol{u} - \phi_2 \boldsymbol{v}^T \mathcal{C} \boldsymbol{v} = 0, \end{cases} \tag{5.23}$$

where

$$\psi_1 = 1 + v_0^2, \quad \psi_2 = 1 + u_0^2, \quad \phi_1 = \frac{1}{\psi_2} u_0 v_0, \quad \text{and} \quad \phi_2 = \frac{1}{\psi_1} u_0 v_0. \tag{5.24}$$

This can as before be rewritten as a linear system in $\boldsymbol{\xi}$:

$$\mathcal{N}_{\mathrm{d}} \boldsymbol{\xi} = 0, \tag{5.25}$$

where this time

$$\mathcal{N}_{\mathrm{d}} \stackrel{\text{def}}{=} \begin{pmatrix} \psi_1 \boldsymbol{d}_1^T - \psi_2 \boldsymbol{d}_2^T \\ 2\boldsymbol{d}^T - \phi_1 \boldsymbol{d}_1^T - \phi_2 \boldsymbol{d}_2^T \end{pmatrix}. \tag{5.26}$$

Once the structure parameters have been estimated via linear least squares, we can predict new image coordinates by rewriting (5.23) as

$$\begin{cases} \psi_1 (X^2 + e_1) - \psi_2 (Y^2 + e_2) = 0, \\ 2XY + e - \phi_1 (X^2 + e_1) - \phi_2 (Y^2 + e_2) = 0. \end{cases} \tag{5.27}$$

To improve the symmetry of our problem, let us introduce as before the variable Z defined by

$$\begin{cases} X^2 = \dfrac{\psi_2}{2}(Z - \vartheta) \\ \\ Y^2 = \dfrac{\psi_1}{2}(Z + \vartheta) \end{cases} \quad \text{where} \quad \vartheta \stackrel{\text{def}}{=} \frac{1}{\psi_1 \psi_2}(\psi_1 e_1 - \psi_2 e_2),$$

Substituting in the second equation of (5.27) yields

$$2XY - u_0 v_0 Z + \chi = 0, \tag{5.28}$$

where

$$\chi = e - \phi_1 e_1 - \phi_2 e_2.$$

In turn, squaring (5.28) yields a quadratic equation in Z only:

$$(1 + u_0^2 + v_0^2)Z^2 + 2u_0 v_0 \chi Z - (\psi_1 \psi_2 \vartheta^2 + \chi^2) = 0 \tag{5.29}$$

whose roots have opposite signs since the degree of the equation is 2 and constant coefficients have opposite signs.

The positive solution is the correct one since $\psi_1 \psi_2 Z = \psi_1 X^2 + \psi_2 Y^2$. This yields two opposite values for X and two opposite values for Y. Two of the four possible pairs are solutions of the original equations: they are determined by the fact that $2XY = u_0 v_0 Z - \chi$.

5.4.2 The Paraperspective Line PIV

Finding the intersection of the line with the reference plane, i.e., the image of line position, proceeds as before in Section 5.2.2 since this point is defined using purely affine notions.

To parameterize the line orientation for the paraperspective images, we can introduce as before the quantity

$$\zeta = \sin\theta X - \cos\theta Y.$$

Squaring this parameter and introducing the variable Z yields

$$2\zeta^2 = (\frac{\psi_1 + \psi_2}{2} + \frac{\psi_1 - \psi_2}{2}\cos 2\theta - u_0 v_0 \sin 2\theta)Z + (\frac{\psi_1 - \psi_2}{2} + \frac{\psi_1 + \psi_2}{2}\cos 2\theta)\vartheta + \chi \sin 2\theta.$$

Since we also have

$$2\zeta^2 \quad = \quad f_1 + f_2 - (f_1 - f_2)\cos 2\theta - f \sin 2\theta,$$

we can solve for Z and replace it in (5.29) to obtain the desired parameterization of the line PIV for paraperspective images. The parameters are $\varepsilon_1, \varepsilon_2, \varepsilon_3$ and Θ as described before in Section 5.2.2.

At synthesis time, once again, we have a trigonometric equation in θ with two solutions for which the correct solution can be chosen using tracking.

5.5 Adding a Second Pass to the Algorithm

In this section we show how to compute the common parameters $1+\lambda^2$, $\lambda\mu$ and μ^2 from both line and point PIVs computed in Section 5.2 and to refine the rest of the parameters for point and line PIVs.

5.5.1 Integrating Point and Line PIVs to Estimate Structure Parameters

Once the structure parameters associated with all the points and lines have been computed, these can be used to construct a refined estimate of the parameters $1+\lambda^2$, $\lambda\mu$ and μ^2 that are common across all features. Indeed, the vectors $(\xi_1-\alpha^2, \xi_2-\alpha\beta, \xi_3-\beta^2)^T$ and $(\varepsilon_1, \varepsilon_2, \varepsilon_3)^T$ associated with the various lines and points all belong to the one-dimensional vector space spanned by $(1+\lambda^2, \lambda\mu, \mu^2)^T$. Note that

$$\begin{pmatrix} \xi_1-\alpha^2 \\ \xi_2-\alpha\beta \\ \xi_3-\beta^2 \end{pmatrix} = z^2 \begin{pmatrix} 1+\lambda^2 \\ \lambda\mu \\ \mu^2 \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_3 \end{pmatrix} = \frac{1}{\gamma^2} \begin{pmatrix} 1+\lambda^2 \\ \lambda\mu \\ \mu^2 \end{pmatrix}.$$

A representative unit vector $(\eta_1, \eta_2, \eta_3)^T$ can be found via singular value decomposition, and we have

$$\begin{cases} \psi\eta_1 = 1+\lambda^2, \\ \psi\eta_2 = \lambda\mu, \\ \psi\eta_3 = \mu^2, \end{cases}$$

for some ψ. Solving this system yields:

$$\begin{cases} 1+\lambda^2 = \eta_1\eta_3/(\eta_1\eta_3-\eta_2^2), \\ \lambda\mu = \eta_2\eta_3/(\eta_1\eta_3-\eta_2^2), \\ \mu^2 = \eta_3\eta_3/(\eta_1\eta_3-\eta_2^2). \end{cases}$$

5.5.2 The Refined Point PIV

Once the common structure parameters $1+\lambda^2$, $\lambda\mu$ and μ^2 have been estimated, a better estimate of the point PIV can be constructed via linear least squares. Here we consider the more general case of paraperspective images and use the derivations from Section 5.4.1.

In particular, we can use these parameters to compute $\mathcal{E}' \overset{\text{def}}{=} \mathcal{E}/z^2$ directly. Equation (5.27) can now be rewritten as

$$\begin{cases} \psi_1(e_1' + X'^2) - \psi_2(e_2' + Y'^2) = 0, \\ e' + 2X'Y' - \phi_1(e_1' + X'^2) - \phi_2(e_2' + Y'^2) = 0, \end{cases} \tag{5.30}$$

where, this time,

$$\begin{cases} e_1' = \boldsymbol{u}_2^T \mathcal{E}' \boldsymbol{u}_2, \\ e_2' = \boldsymbol{v}_2^T \mathcal{E}' \boldsymbol{v}_2, \\ e' = 2\boldsymbol{u}_2^T \mathcal{E}' \boldsymbol{v}_2, \end{cases}$$

and

$$\begin{cases} X' = X/z = \boldsymbol{c} \cdot \boldsymbol{u} = \boldsymbol{c}^T \mathcal{A} \boldsymbol{a} = \boldsymbol{n} \cdot \boldsymbol{a}, \\ Y' = Y/z = \boldsymbol{c} \cdot \boldsymbol{v} = \boldsymbol{c}^T \mathcal{A} \boldsymbol{b} = \boldsymbol{n} \cdot \boldsymbol{b}, \end{cases}$$

where $\boldsymbol{n} = (0,0,1)^T$ is the normal to the reference triangle. Since the vectors $\boldsymbol{a}$ and $\boldsymbol{b}$ are orthogonal to the line joining the optical center of the camera to the reference point, it follows that X', Y', and the affine flow are all zero when this line is parallel to $\boldsymbol{n}$.

We can use (5.30) to solve for X' and Y' as before. Once these values have been computed for each frame, we can estimate the value of α, β, and z for each point using linear least squares and the relation

$$\begin{pmatrix} u_1^{(1)} & u_2^{(1)} & X'^{(1)} & -u^{(1)} \\ v_1^{(1)} & v_2^{(1)} & Y'^{(1)} & -v^{(1)} \\ \dots & \dots & \dots & \dots \\ u_1^{(f)} & u_2^{(f)} & X'^{(f)} & -u^{(f)} \\ v_1^{(f)} & v_2^{(f)} & Y'^{(f)} & -v^{(f)} \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \\ z \\ 1 \end{pmatrix} = 0.$$

This assumes of course that the signs chosen for the values of X' and Y' for all frames are consistent. To achieve this, for f images of m points, we construct during the *first pass* of the algorithm the $m \times 2f$ matrix

$$\begin{pmatrix} X_1^{(1)} & \dots & X_1^{(f)} & Y_1^{(1)} & \dots & Y_1^{(f)} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ X_m^{(1)} & \dots & X_m^{(f)} & Y_m^{(1)} & \dots & Y_m^{(f)} \end{pmatrix} = \begin{pmatrix} z_1 \\ \dots \\ z_m \end{pmatrix} (X'^{(1)}, \dots, X'^{(f)}, Y'^{(1)}, \dots, Y'^{(f)}).$$

This matrix has by definition rank 1, and it follows that an estimate of the vector $(X'^{(1)},\dots,X'^{(f)},Y'^{(1)},\dots,Y'^{(f)})$ can be computed (up to an unknown scale factor) through singular value decomposition. This estimate can be used to assign consistent signs to the coefficients $X'^{(i)}$ and $Y'^{(i)}$ recomputed during the second pass of the algorithm.

At rendering time, we solve (5.30) for X' and Y' for each new frame, then compute the values of u and v for each point by using the relationship

$$u = zX' + \alpha u_1 + \beta u_2, \quad v = zY' + \alpha v_1 + \beta v_2.$$

5.5.3 The Refined Line PIV

Once again, when the common parameters $1+\lambda^2$, $\lambda\mu$ and μ^2 are given the two parameters α and β can be obtained via linear least squares as follows.

As we have done for the point case, we can compute $\mathcal{E}$ directly. When we take $\boldsymbol{p} = \rho\boldsymbol{\omega}$, (5.27) can be rewritten as

$$\begin{cases} \psi_1(e_1 + X^2) - \psi_2(e_2 + Y^2) = 0, \\ e + 2XY - \phi_1(e_1 + X^2) - \phi_2(e_2 + Y^2) = 0, \end{cases} \tag{5.31}$$

where

$$\begin{cases} e_1 = \boldsymbol{u}_2^T \mathcal{E} \boldsymbol{u}_2, \\ e_2 = \boldsymbol{v}_2^T \mathcal{E} \boldsymbol{v}_2, \\ e = 2\boldsymbol{u}_2^T \mathcal{E} \boldsymbol{v}_2, \end{cases}$$

and

$$\begin{cases} X = \rho\cos\theta + \alpha u_1 + \beta u_2 = \boldsymbol{c}\cdot\boldsymbol{u} = \boldsymbol{c}^T\mathcal{A}\boldsymbol{a} = \boldsymbol{n}\cdot\boldsymbol{a}, \\ Y = \rho\sin\theta + \alpha v_1 + \beta v_2 = \boldsymbol{c}\cdot\boldsymbol{v} = \boldsymbol{c}^T\mathcal{A}\boldsymbol{b} = \boldsymbol{n}\cdot\boldsymbol{b}, \end{cases}$$

where $\boldsymbol{n} = (0,0,1)^T$ is the normal to the reference triangle.

We can use (5.31) to solve for X and Y as before. Once these values have been computed for each frame, we can estimate the value of α and β for each line using linear least squares and the relation

$$\begin{pmatrix} v_1^{(1)}\cos\theta^{(1)} - u_1^{(1)}\sin\theta^{(1)} & v_2^{(1)}\cos\theta^{(1)} - u_2^{(1)}\sin\theta^{(1)} & X^{(1)}\sin\theta^{(1)} - Y^{(1)}\cos\theta^{(1)} \\ \dots & \dots & \dots \\ v_1^{(f)}\cos\theta^{(f)} - u_1^{(f)}\sin\theta^{(f)} & v_2^{(f)}\cos\theta^{(f)} - u_2^{(f)}\sin\theta^{(f)} & X^{(f)}\sin\theta^{(f)} - Y^{(f)}\cos\theta^{(f)} \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \\ 1 \end{pmatrix} = 0.$$

1. Training:
   - (a) Pick three points $\boldsymbol{a}_0$, $\boldsymbol{a}_1$ and $\boldsymbol{a}_2$ as the reference points and center all points at $\boldsymbol{a}_0$.
   - (b) Compute the point PIVs using the method of Section 5.2.1 or Section 5.4.1.
   - (c) Compute the line PIVs using the method of Section 5.2.2 or Section 5.4.2.
   - (d) Compute refined point and line PIVs as described in Section 5.5.
2. Rendering:
   - (a) Define a new view by setting image coordinates of $\boldsymbol{a}_0$, $\boldsymbol{a}_1$ and $\boldsymbol{a}_2$ at the desired view.
   - (b) Predict the image of points and lines using methods of Section 5.2.1 (or Section 5.4.1) and Section 5.2.2 (or Section 5.4.2).
   - (c) Resolve ambiguities using tracking from a previous image.
   - (d) Render the images using the method described in Section 5.3.2.

Figure 5.7 The image-based rendering algorithm.

As before, this assumes that the signs chosen for the values of X and Y for all frames are consistent. Once we have the signs of X' and Y' for each frame and since the signs of X (resp. Y) and X' (resp. Y') are the same, we can use the results from the second pass of point features and choose the signs for X and Y accordingly.

At rendering time, we solve (5.31) for X and Y for each new frame, then compute the value of θ for each point by using the relationship

$$(X - \alpha u_1 - \beta u_2)\cos\theta - (Y - \alpha v_1 - \beta v_2)\sin\theta = 0.$$

5.6 Summary of the Algorithm

Figure 5.7 shows the various steps of the proposed approach to image-based rendering using parameterized image varieties for point and line correspondences. The approach has two distinct steps: (1) computing the point and line PIVs (training), and (2) predicting images using the computed PIVs (rendering).

It should be noted that three reference points must be selected among the input point correspondences. We can always pick the center of the mass for each image (using the points that are visible in all images) as the first reference point.

5.7 Degenerate Scene and Camera Configurations

It is of course important to understand what degenerate configurations of scene points and lines and camera positions make the determination of the point line PIVs ill-conditioned. Here we spell out some simple cases in the paraperspective case.

It is easy to show using (5.7) that the matrix $\mathcal{N}_{\mathrm{d}}$ in (5.26) can be rewritten as

$$\mathcal{N}_{\mathrm{d}} = \mathcal{N}_{\mathrm{p}} \mathcal{N}_{\mathrm{str}}, \tag{5.32}$$

where the projection matrix is

$$\mathcal{N}_{\mathrm{p}} \overset{\mathrm{def}}{=} \begin{pmatrix} \boldsymbol{e}^T - \phi_1 \boldsymbol{e}_1^T - \phi_2 \boldsymbol{e}_2^T \\ \psi_1 \boldsymbol{e}_1^T - \psi_2 \boldsymbol{e}_2^T \end{pmatrix}, \tag{5.33}$$

and the structure matrix is

$$\mathcal{N}_{\mathrm{str}} \overset{\mathrm{def}}{=} \begin{pmatrix} 1 & 2p & p^2 & 2x & 2px & x^2 \\ 0 & q & pq & y & py + qx & xy \\ 0 & 0 & q^2 & 0 & 2qy & y^2 \\ 0 & 0 & 0 & z & pz & xz \\ 0 & 0 & 0 & 0 & qz & yz \\ 0 & 0 & 0 & 0 & 0 & z^2 \end{pmatrix}, \tag{5.34}$$

with

$$\begin{cases} \boldsymbol{e}_1^T \overset{\mathrm{def}}{=} (a_1^2, 2a_1a_2, a_2^2, 2a_1a_3, 2a_2a_3, a_3^2), \\ \boldsymbol{e}_2^T \overset{\mathrm{def}}{=} (b_1^2, 2b_1b_2, b_2^2, 2b_1b_3, 2b_2b_3, b_3^2), \\ \boldsymbol{e}^T \overset{\mathrm{def}}{=} (a_1b_1, a_1b_2 + a_2b_1, a_2b_2, a_1b_3 + a_3b_1, a_2b_3 + a_3b_2, a_3b_3). \end{cases}$$

Let us define $\boldsymbol{\eta} \overset{\mathrm{def}}{=} (1, 0, 1, 0, 0, 1)^T$. Using (5.6), (5.7) and (5.8) shows that

$$\mathcal{N}_{\mathrm{p}} \boldsymbol{\eta} = \begin{pmatrix} \boldsymbol{a} \cdot \boldsymbol{b} - \phi_1 |\boldsymbol{a}|^2 - \phi_2 |\boldsymbol{b}|^2 \\ \psi_1 |\boldsymbol{a}|^2 - \psi_2 |\boldsymbol{b}|^2 \end{pmatrix} = 0 \quad \text{and} \quad z^2 \boldsymbol{\eta} = \mathcal{N}_{\mathrm{str}} \boldsymbol{\xi}. \tag{5.35}$$

Equations (5.32) and (5.35) confirm of course that the rank of $\mathcal{N}_{\mathrm{p}}$ is at most 5. If we now consider f images and denote by $\mathcal{N}_{\mathrm{d}}^{(i)}$ (resp. $\mathcal{N}_{\mathrm{p}}^{(i)}$) the value of the matrix $\mathcal{N}_{\mathrm{d}}$ (resp. $\mathcal{N}_{\mathrm{p}}$) for image number i ($i = 1, .., f$), and define the data and projection matrices by

$$\mathcal{N}_{\mathrm{dat}} \overset{\mathrm{def}}{=} \begin{pmatrix} \mathcal{N}_{\mathrm{d}}^{(1)} \\ \dots \\ \mathcal{N}_{\mathrm{d}}^{(f)} \end{pmatrix} \quad \text{and} \quad \mathcal{N}_{\mathrm{pro}} \overset{\mathrm{def}}{=} \begin{pmatrix} \mathcal{N}_{\mathrm{p}}^{(1)} \\ \dots \\ \mathcal{N}_{\mathrm{p}}^{(f)} \end{pmatrix},$$

we obtain

$$\mathcal{N}_{\text{dat}} = \mathcal{N}_{\text{pro}}\mathcal{N}_{\text{str}}.$$

Like $\mathcal{N}_{\text{p}}$, $\mathcal{N}_{\text{dat}}$ has at most rank 5. Degenerate configurations correspond to either $\mathcal{N}_{\text{pro}}$ or $\mathcal{N}_{\text{str}}$ having rank smaller than 5.

5.7.1 Degenerate Point Configurations

First note that the determinant of the matrix $\mathcal{N}_{\text{str}}$ is equal to q^4z^4. Hence this matrix has rank 6 unless $z = 0$. In this case, its last three rows are zero, and it follows that the matrix $\mathcal{N}_{\text{dat}}$ has at most rank 3 for points in the plane of the reference triangle. An explicit parameterization of the null space of $\mathcal{N}_{\text{dat}}$ is given below, and it is used to show that the rank is indeed 3, and that the image synthesis method described in Section 5.2.1 will give the correct results even in that case.

The fact that $\boldsymbol{\xi}$ belongs to the null space of the matrix

$$\begin{pmatrix} 1 & 2p & p^2 & 2x & 2px & x^2 \\ 0 & q & pq & y & py + qx & xy \\ 0 & 0 & q^2 & 0 & 2qy & y^2 \end{pmatrix}$$

can be rewritten as

$$\mathcal{P}\begin{pmatrix} \xi_1 \\ \xi_2 \\ \xi_3 \end{pmatrix} = -\mathcal{Q}\begin{pmatrix} \xi_4 \\ \xi_5 \\ 1 \end{pmatrix},$$

where

$$\mathcal{P} \overset{\text{def}}{=} \begin{pmatrix} 1 & 2p & p^2 \\ 0 & q & pq \\ 0 & 0 & q^2 \end{pmatrix} \quad \text{and} \quad \mathcal{Q} \overset{\text{def}}{=} \begin{pmatrix} 2x & 2px & x^2 \\ y & py + qx & xy \\ 0 & 2qy & y^2 \end{pmatrix}.$$

The inverse of $\mathcal{P}$ is easily shown to be

$$\mathcal{P}^{-1} = \begin{pmatrix} 1 & -2p/q & p^2/q^2 \\ 0 & 1/q & -p/q^2 \\ 0 & 0 & 1/q^2 \end{pmatrix},$$

and it follows that

$$\begin{pmatrix}\xi_1\\ \xi_2\\ \xi_3\end{pmatrix} = -\mathcal{P}^{-1}\mathcal{Q}\begin{pmatrix}\xi_4\\ \xi_5\\ 1\end{pmatrix} = -\begin{pmatrix}2\alpha & 0 & \alpha^2\\ \beta & \alpha & \alpha\beta\\ 0 & 2\beta & \beta^2\end{pmatrix}\begin{pmatrix}\xi_4\\ \xi_5\\ 1\end{pmatrix}.$$

We finally obtain

$$\begin{cases} \xi_1 - \xi_4^2 = -(\xi_4+\alpha)^2, \\ \xi_2 - \xi_4\xi_5 = -(\xi_4+\alpha)(\xi_5+\beta), \\ \xi_3 - \xi_5^2 = -(\xi_5+\beta)^2. \end{cases}$$

In particular, if we define $\nu_1 = \xi_4+\alpha$, $\nu_2 = \xi_5+\beta$, and $\boldsymbol{\nu} = (\nu_1, \nu_2)^T$, we have $e_1 = -(\boldsymbol{\nu}\cdot\boldsymbol{u}_2)^2$, $e_2 = -(\boldsymbol{\nu}\cdot\boldsymbol{v}_2)^2$, and $e = -(\boldsymbol{\nu}\cdot\boldsymbol{u}_2)(\boldsymbol{\nu}\cdot\boldsymbol{v}_2)$, and it follows immediately that the two solutions of (5.27) are $(X^*, Y^*) = (\boldsymbol{\nu}\cdot\boldsymbol{u}_2, \boldsymbol{\nu}\cdot\boldsymbol{v}_2)$ and $(-X^*, -Y^*)$.

We can now determine the solution (u^*, v^*) corresponding to (X^*, Y^*):

$$\begin{cases} u^* = X^* - \xi_4 u_1 - \xi_5 u_2 = \boldsymbol{\nu}\cdot\boldsymbol{u}_2 - (\nu_1-\alpha)u_1 - (\nu_2-\beta)u_2 = \alpha u_1 + \beta u_2, \\ v^* = Y^* - \xi_4 v_1 - \xi_5 v_2 = \boldsymbol{\nu}\cdot\boldsymbol{v}_2 - (\nu_1-\alpha)v_1 - (\nu_2-\beta)v_2 = \alpha v_1 + \beta v_2. \end{cases}$$

Those are of course the correct image coordinates for a point in the reference plane, corresponding to a zero affine flow.

5.7.2 Degenerate Line Configurations

In our analysis of the line PIV we have assumed that the line Δ in the scene intersects the refence plane transversally so we can take $\Omega = (x, y, 1)^T$. Let us now consider the general case where the line direction is given by the unit vector $\Omega = (x, y, z)^T$.

Without going into the details of the derivation, we can analyze the degenerate line configuration as follows: the discussion in the previous section holds with $\boldsymbol{P} = (x, y, z)^T$ and $u = \rho\cos\theta$ and $v = \rho\sin\theta$. Then, by noting that the determinant of the matrix $\mathcal{N}_{\text{str}}$ is equal to q^4z^4, we see that this matrix has rank 6 unless $z = 0$. In this case, the last three rows of $\mathcal{N}_{\text{str}}$ are zero, and it follows that the matrix $\mathcal{N}_{\text{dat}}$ has at most rank 3 for lines parallel to the plane spanned by the three reference points. This is a degenerate case for line features.

5.7.3 Degenerate Camera Configurations

While it is in general difficult to give necessary and sufficient conditions for the rank of $\mathcal{N}_{\text{pro}}$ to be less than 5, it is shown in below that the matrix has at most rank 4 when the motion of the camera is such that the direction of the line joining the optical center of the camera to the scene reference point remains constant.

Let us define the vectors $\boldsymbol{q} = (q_1, q_2, q_3)^T$ and $\boldsymbol{\tau} = (q_1^2, q_1q_2, q_2^2, q_1q_3, q_2q_3, q_3^2)^T$. We have

$$\mathcal{N}_{\text{p}}\boldsymbol{\tau} = \begin{pmatrix} (\boldsymbol{a}\cdot\boldsymbol{q})(\boldsymbol{b}\cdot\boldsymbol{q}) - \phi_1(\boldsymbol{a}\cdot\boldsymbol{q})^2 - \phi_2(\boldsymbol{b}\cdot\boldsymbol{q})^2 \\ \psi_1(\boldsymbol{a}\cdot\boldsymbol{q})^2 - \psi_2(\boldsymbol{b}\cdot\boldsymbol{q})^2 \end{pmatrix}$$

It follows that a necessary and sufficient condition for $\boldsymbol{\tau}$ to be in the null space of $\mathcal{N}_{\text{p}}$ is that

$$\begin{cases} XY - \phi_1 X^2 - \phi_2 Y^2 = 0, \\ \psi_1 X^2 - \psi_2 Y^2 = 0, \end{cases}$$

where $X = \boldsymbol{a}\cdot\boldsymbol{q}$ and $Y = \boldsymbol{b}\cdot\boldsymbol{q}$. This homogeneous system of equations obviously admits $(0, 0)$ as a solution. To see that this is in fact the only solution, we can easily eliminate X and Y between the two equations. This yields the resultant $\psi_1\psi_2 - (\phi_1\psi_2 + \phi_2\psi_1) = 1 + u_0^2 + v_0^2$, which is positive.

Recalling the definition of u_0, v_0, $\boldsymbol{a}$ and $\boldsymbol{b}$, we see that the solution $X = Y = 0$ corresponds to

$$\begin{cases} t_z\boldsymbol{i}\cdot\boldsymbol{q} - t_x\boldsymbol{k}\cdot\boldsymbol{q} = 0 \\ t_z\boldsymbol{j}\cdot\boldsymbol{q} - t_y\boldsymbol{k}\cdot\boldsymbol{q} = 0 \end{cases} \iff (\boldsymbol{R} - \boldsymbol{C}) \times \boldsymbol{q} = 0.$$

In other words, a necessary and sufficient condition for the vector $\boldsymbol{\tau}$ to be in the null space of $\mathcal{N}_{\text{p}}$ is that $\boldsymbol{q}$ be parallel to the line joining the optical center of the camera to the reference point (see Figure 5.8).

5.8 Implementation and Results

In this section, we present the results of the experiments conducted to validate our approach to image-based rendering using point and line features described in the previous sections.

The point-based PIV algorithm is implemented in C. The line-based algorithm is implemented in MATLAB. Interactive image synthesis is programmed in C++ using OpenGL [93] and the OpenGL Utility Toolkit (GLUT) [54] on both UNIX and Windows NT platforms.

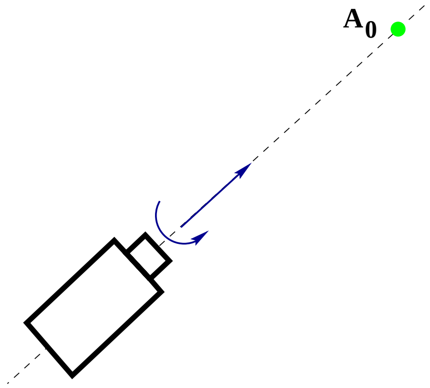


Figure 5.8 A degenerate camera configuration where the parameterization may fail: the camera moves along the line joining the center of projection and the first reference point.

5.8.1 The Data Sets

We have tested our algorithm on seven data sets: the HOUSE data set, kindly provided by Dr. Carlo Tomasi, the KITCHEN data set, kindly provided by the Modeling by Videotaping Research Group at the Department of Computer Science of Carnegie Mellon University, the LQBOX data set kindly provided by Dr. Long Quan and the data sets TOWER, XL1BOX, FLOWER and FACE which are acquired by the author in the Computer Vision and Robotics Laboratory at the Beckman Institute, using a Canon XL1 Digital Camcorder kindly provided by Dr. David Kriegman.

Figures 5.9 and 5.10 show sample pictures from the data sets along with the point and line features. Table 5.1 shows the properties of these data sets. Each data set contains point features tracked in a set of images. The point correspondences for the KITCHEN and HOUSE data sets were provided by their authors. For our own data sets, we have implemented a version of the Harris corner detector [32] and matched the points using a simple RANSAC (random sample consensus) type algorithm [23]. We then eliminated visible bad matches by hand. The line segments were extracted in images using the publicly available software environment VISTA [89] and the line correspondences are established by hand.

We assume that the images in these data sets are obtained using a paraperspective camera and we pick the center of mass of the observed points as our first reference point A_0. The other two reference points are chosen by the user.

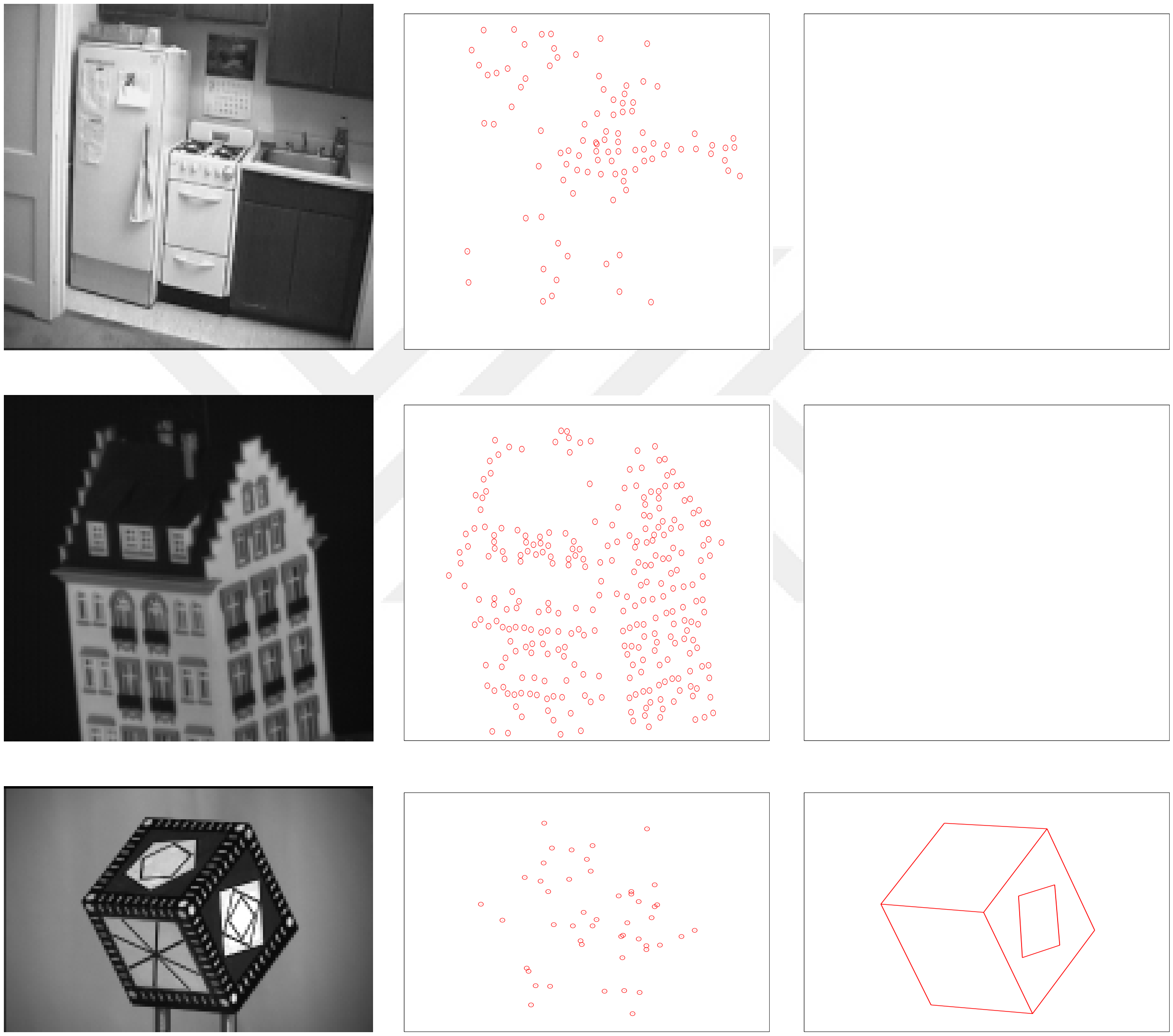


Figure 5.9 First three data sets used in the experiments: The first column shows a sample image. The second column shows the point features and the third column shows the line features.

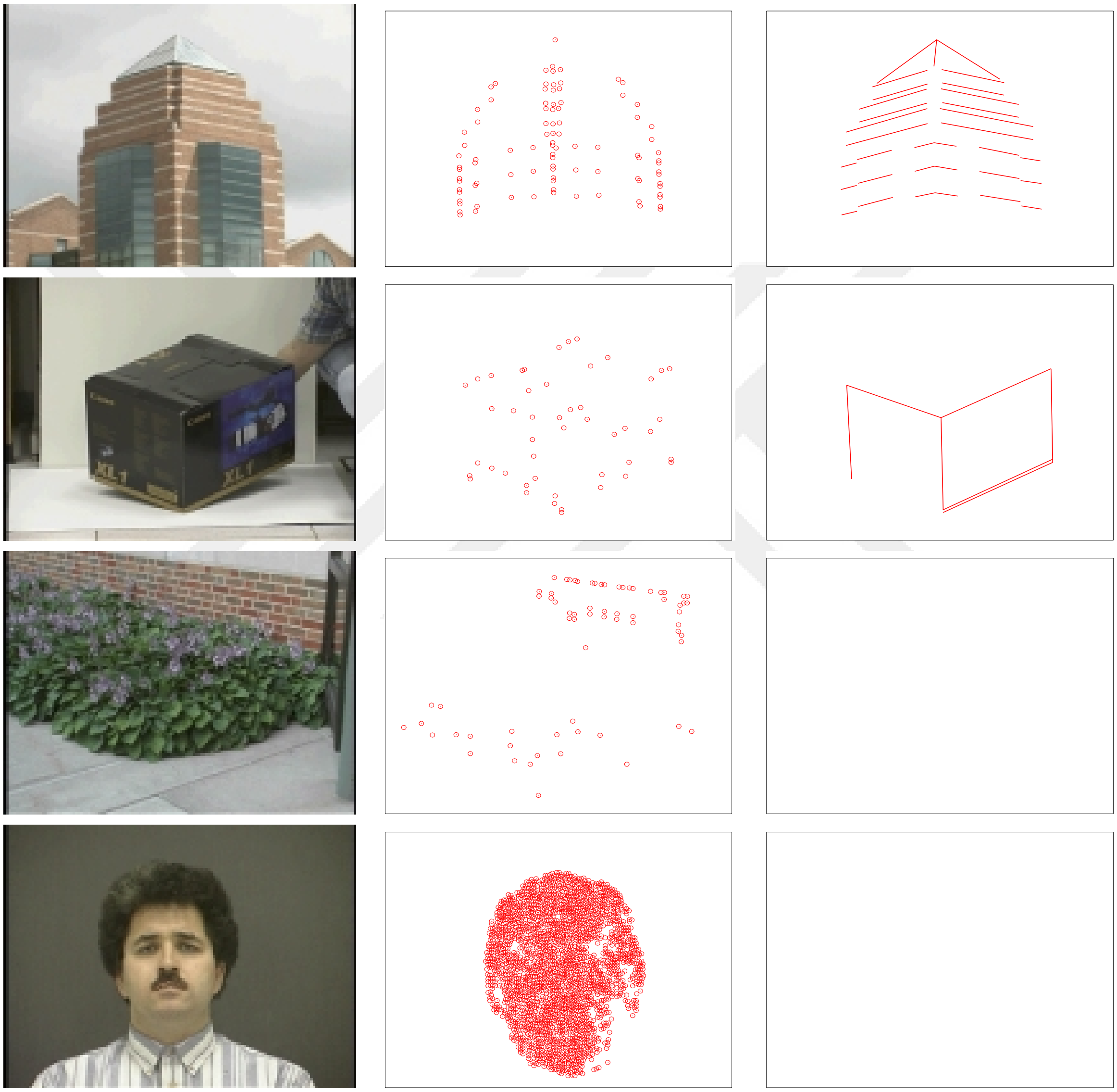


Figure 5.10 Last four data sets used in the experiments: The first column shows a sample image. The second column shows the point features and the third column shows the line features.

Table 5.1 Properties of the data sets used in the experiments.

	# of Images	# of Points	# Lines
KITCHEN	40	97	-
HOUSE	175	275	-
LQBOX	14	48	13
TOWER	30	94	33
XL1BOX	37	47	7
FLOWER	42	67	-
FACE	51	2305	-

5.8.2 The Point PIV Experiments

As we have described in the previous sections, we have four variants of the image-based rendering algorithm for point features: (1) the first pass of the weak perspective PIV (or SW1 in short), (2) the first pass of the paraperspective PIV (SP1), (3) the second pass of the weak perspective PIV (SW2), and (4) the second pass of the paraperspective PIV (SP2). These methods use linear least squares to estimate the parameters.

We first present some quantitative results: we have estimated the point PIVs for each data set using the first 25%, 50% and 75% of the images in each data set and reconstructed the image points for the rest of the images using the computed parameters. We have also used all of the images in the training and reconstructed the image points for all of the images. For each case, we have recorded errors as the difference between the reconstructed and the original image point locations. Figure 5.11 shows the results of this experiment. For each data set the four bars from left to right represents the SW1, SP1, SW2 and SP2 methods. The first three plots show the reconstruction power of our approach. As expected, the larger the number of images used in training, the better the reconstruction for the new views. Even though the second pass does not always improve the results quantitatively, it improves the quality of the synthesized images as we will see below.

Figures 5.12-5.17 shows the synthesized views for the above experiments corresponding to the last frame in each data set and using only the first half of the images in training. In each case, we show, from left to right, the images synthesized by SW1, SP1, SW2 and SP2 algorithms. As it can be seen, the second pass improves results qualitatively.

Figure 5.18 shows completely new images generated for each data set. These pictures are created by changing the position of the three reference points. Note that we present more results for point PIVs in the next chapter where we evaluate the new bilinear estimation algorithm.

Finally, we have created movies from synthesized pictures using our approach. These movies can be found at "http://www-cvr.ai.uiuc.edu/˜ygenc/thesis/index.html".

5.8.3 The Line PIV Experiments

We have carried out experiments on the LQBOX, XL1BOX and TOWER data sets for line-based parameterized image varieties. Figure 5.20 shows the mean errors for the reconstruction of images of line features for different variants of the algorithm, namely, the first passes of the weak perspective and paraperspective algorithms and the second passes (or W1, P1, W2 and P2 in short). Note that the line position is computed using affine notions only so it does not have any variants. We have computed the parameters associated with the line positions and line directions by using a different number of images during training and reconstructing the images of the lines for the rest of the images or for all of the images for the last case. We have recorded the error in reconstruction of the line position in pixels and the line direction in degrees.

Figures 5.21-5.23 show the line features with their extents reconstructed for the last frame in each data set. We have used the first half of the images in each data set in training. The figures also show the original lines as the dashed lines whereas the reconstructed lines are drawn as solid lines. Once again the second pass improves the results considerably.

5.8.4 Combined Results for Line and Point PIVs

We present in Figure 5.24 some new image synthesis results where we used the line and point PIVs together for the LQBOX, TOWER and XL1BOX data sets. An advantage of lines over points is that they can be located very accurately in edge maps using least squares and they can facilitate the production of a good triangulation.

5.9 Conclusions

We have presented a parameterization of the set of all images of a fixed set of points and lines based on images of three reference points. The parameterization (Parameterized Image Variety or PIV) takes into account the Euclidean constraints and produces correct images in the context of image-based rendering. We have integrated the point-based and line-based methods in a common framework for image-based rendering. The method correctly handles occlusions and enables the use of z-buffer and texture-mapping techniques for rendering images without actual depth computation.

We have implemented the proposed methods and performed extensive experiments on real data sets. Our experiments show that the method synthesizes realistic images.

Finally, we have analyzed the degenerate scene and camera configurations for our parameterization. Degenerate scene configurations include the case where a point lies on the plane spanned by the three reference points. The parameterization still works for this case. For lines, a degeneracy occurs when the line is parallel to the reference plane. We have also analyzed the degenerate camera configurations. One of the degeneracies arises when the camera moves along a line which is parallel to the line joining the center of the camera and the reference point.

The parameterization is based on weak perspective and paraperspective projections. Although our approach can in principle be extended to the full perspective case, this would require eliminating three variables (the last row of the camera rotation) among five quadratic equations (the Euclidean constraints associated with perspective images), a formidable task in elimination theory [69].

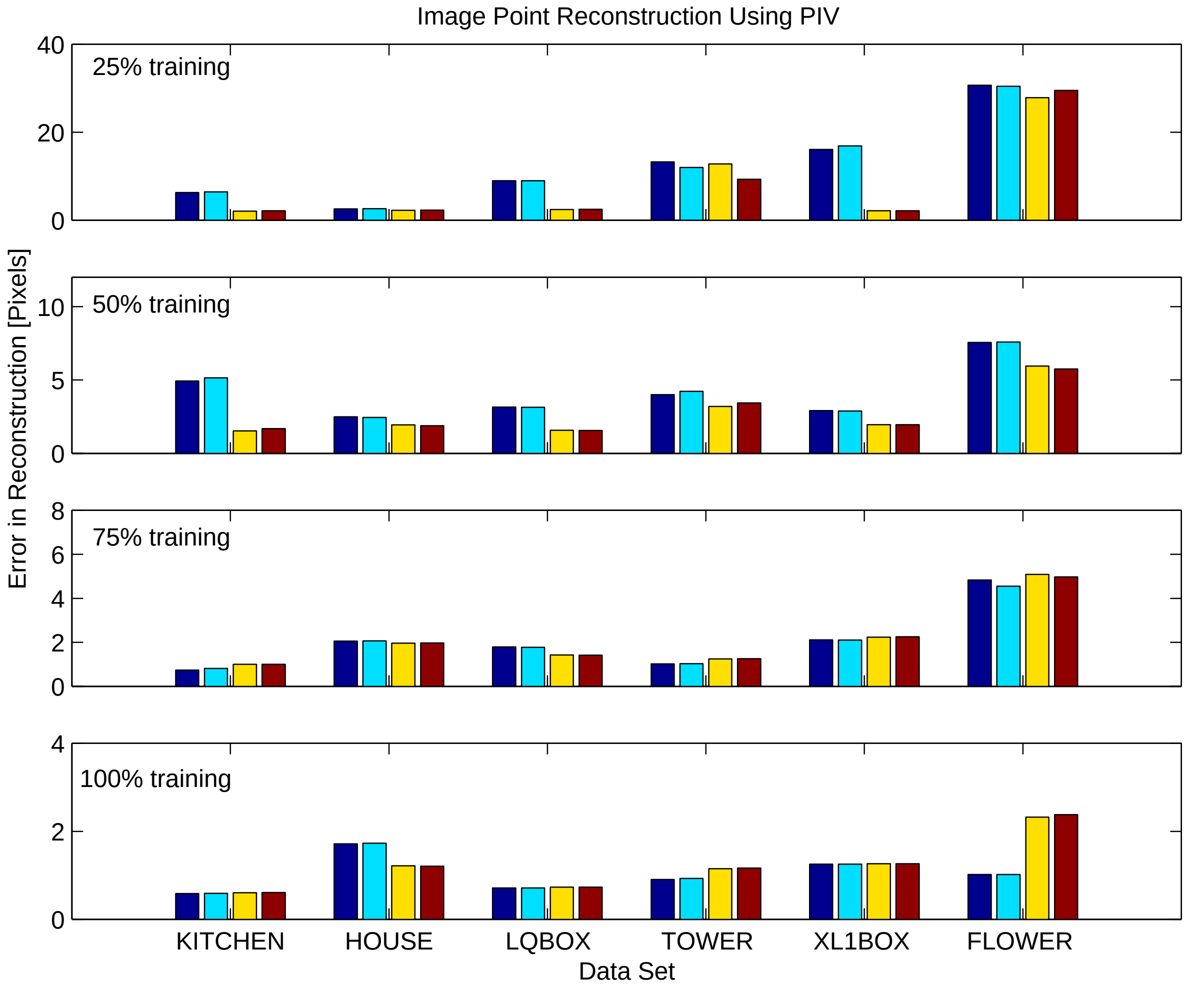


Figure 5.11 Image point reconstruction on real data sets: for each data the bars from left to right represents the SW1, SP1, SW2 and SP2 methods. In training, from top to bottom, the first 25%, 50%, 75% and 100% of the images were used. And, in testing, the remaining images for the first three plots and all of the images for the last plot were used and the average error in prediction were recorded.

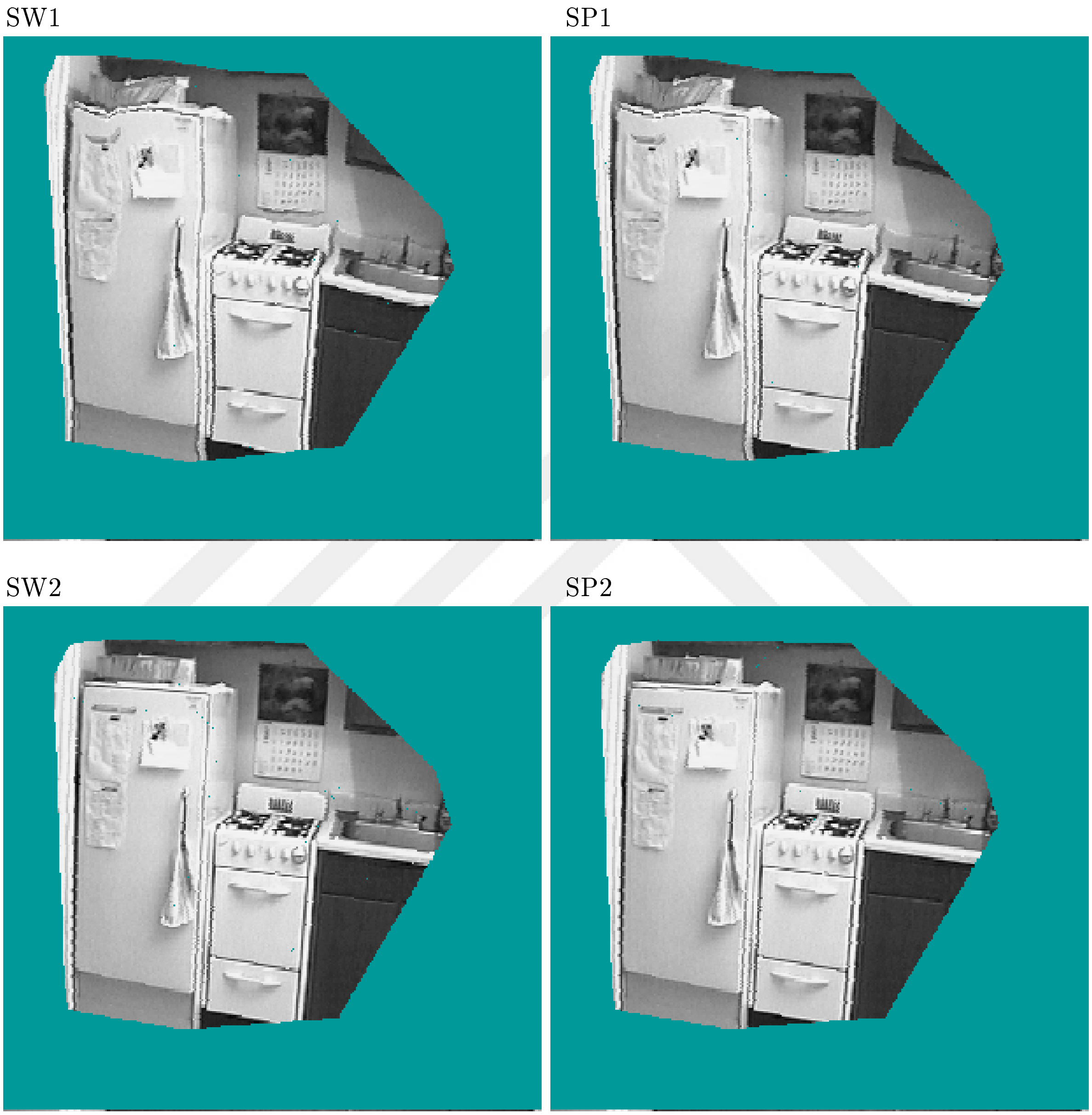


Figure 5.12 Image synthesis for the last image in the KITCHEN data set using the first half of the images in training.

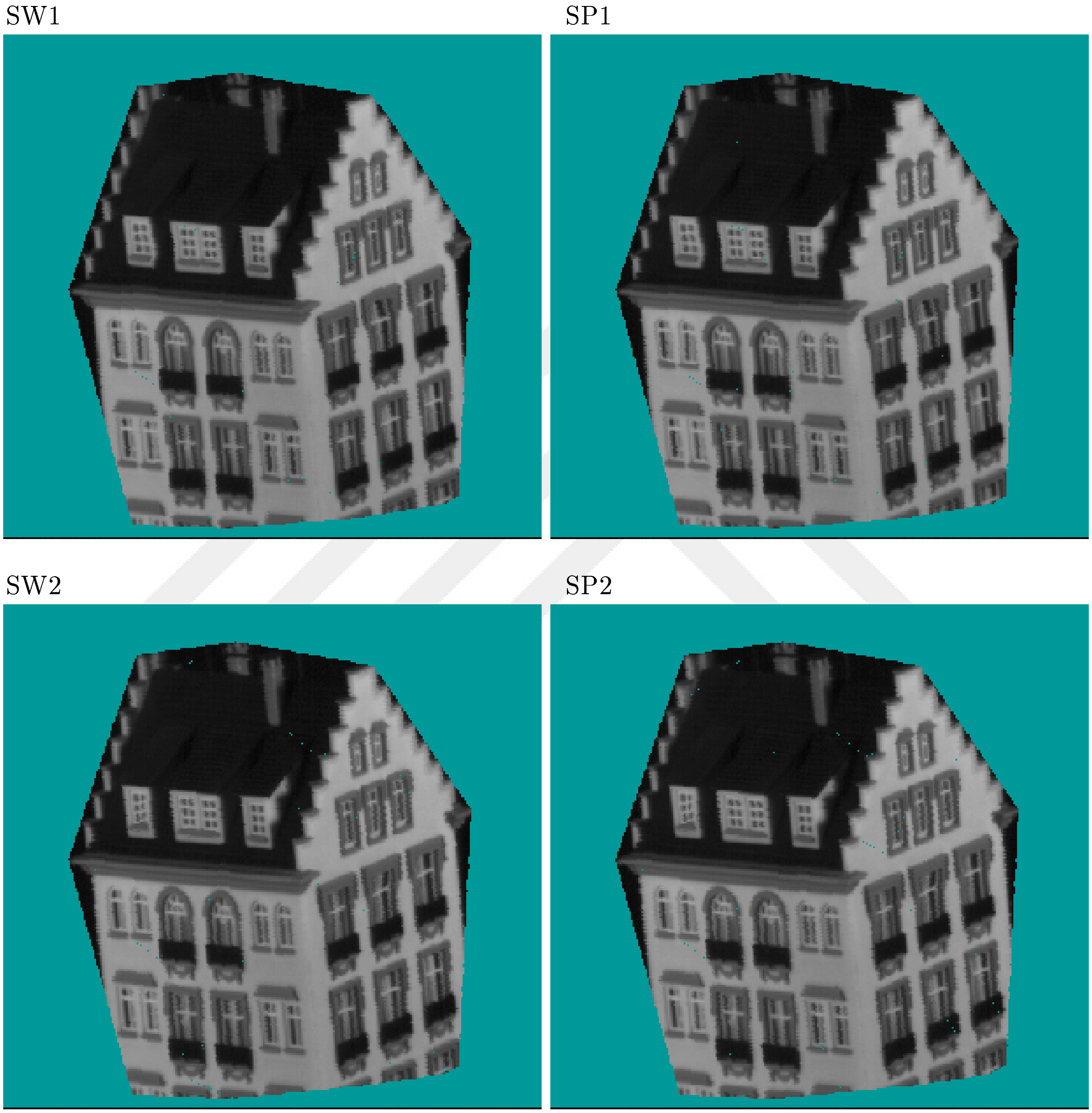


Figure 5.13 Image synthesis for the last image in the HOUSE data set using the first half of the images in training.

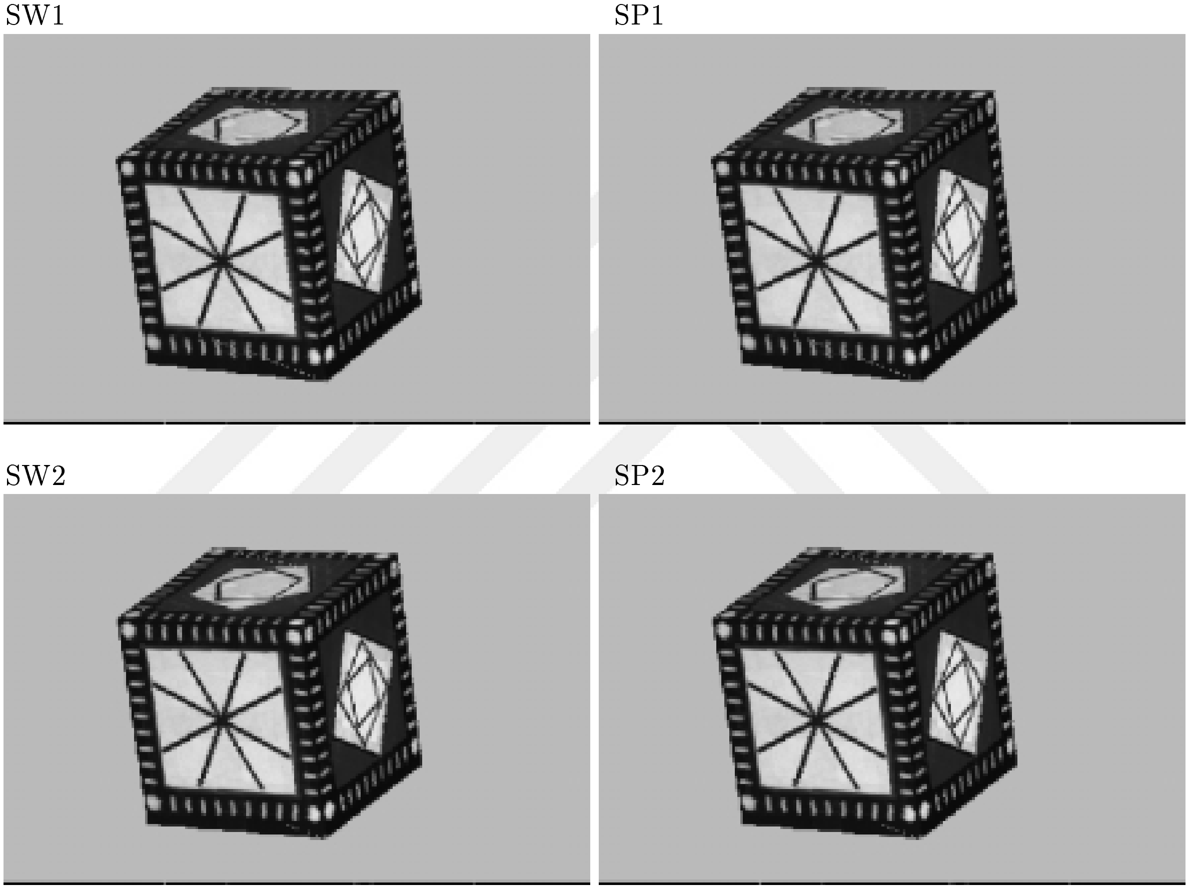


Figure 5.14 Image synthesis for the last image in the LQBOX data set using the first half of the images in training.

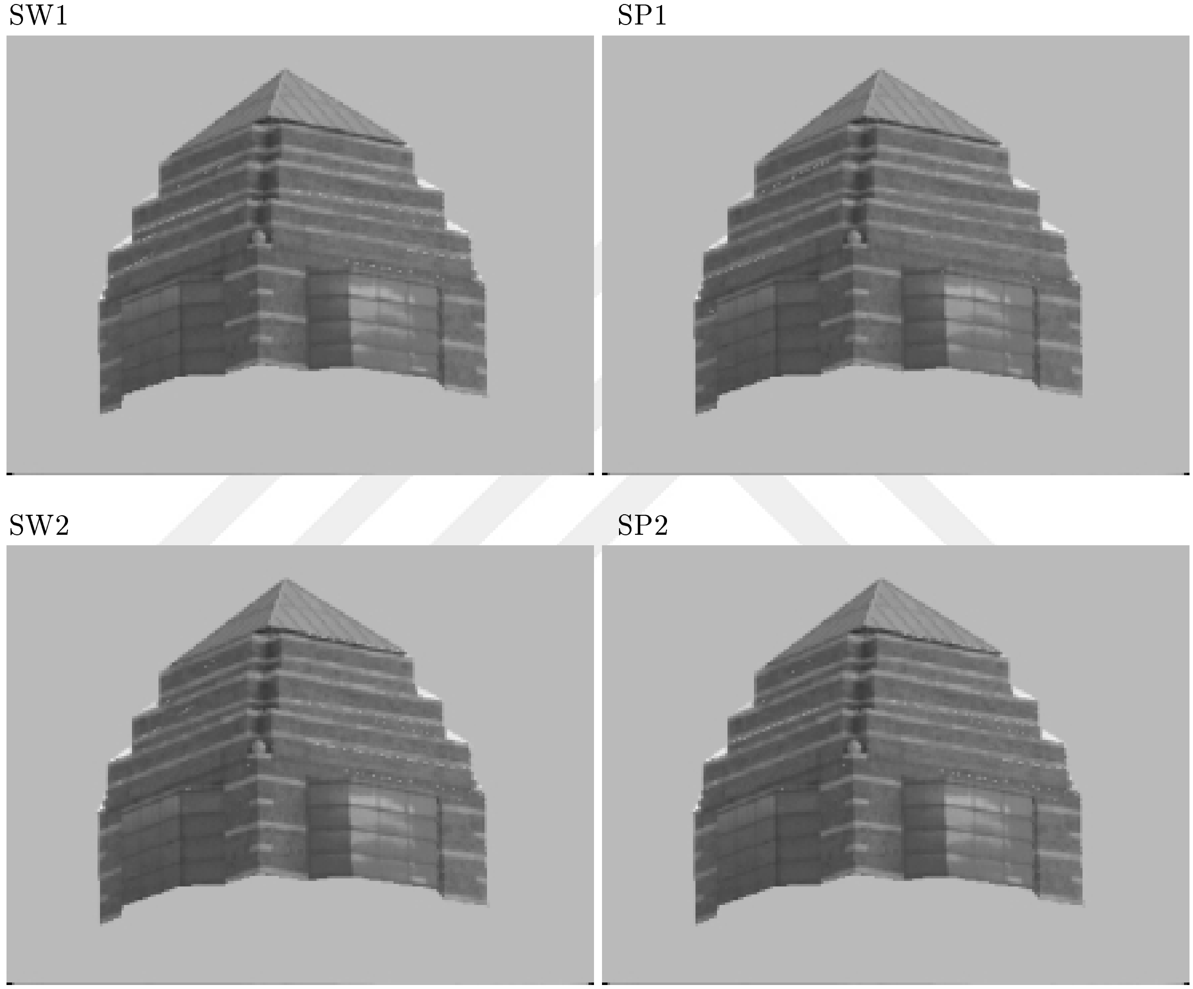


Figure 5.15 Image synthesis for the last image in the TOWER data set using the first half of the images in training.

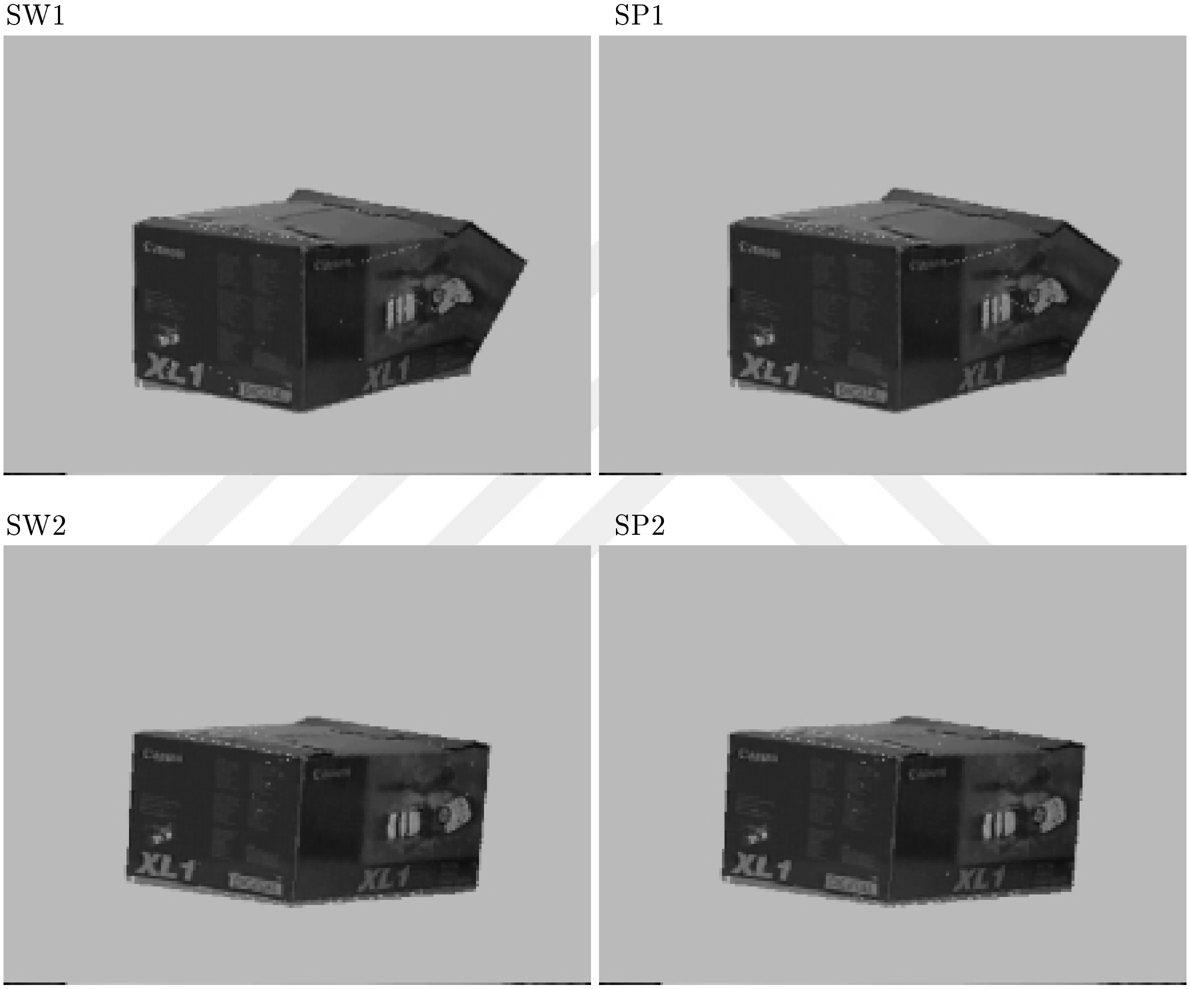


Figure 5.16 Image synthesis for the last image in the XL1BOX data set using the first half of the images in training.

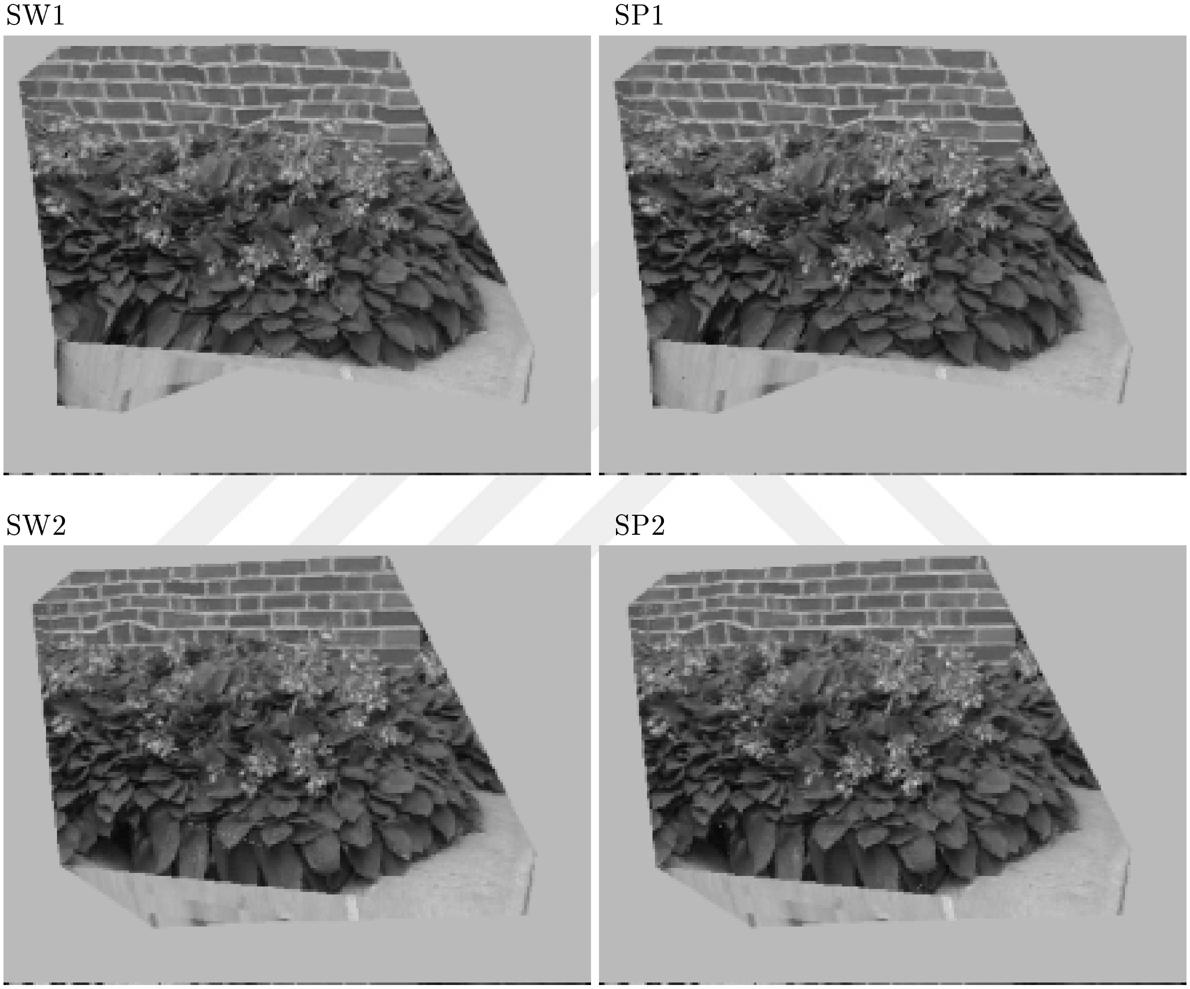


Figure 5.17 Image synthesis for the last image in the FLOWER data set using the first half of the images in training.

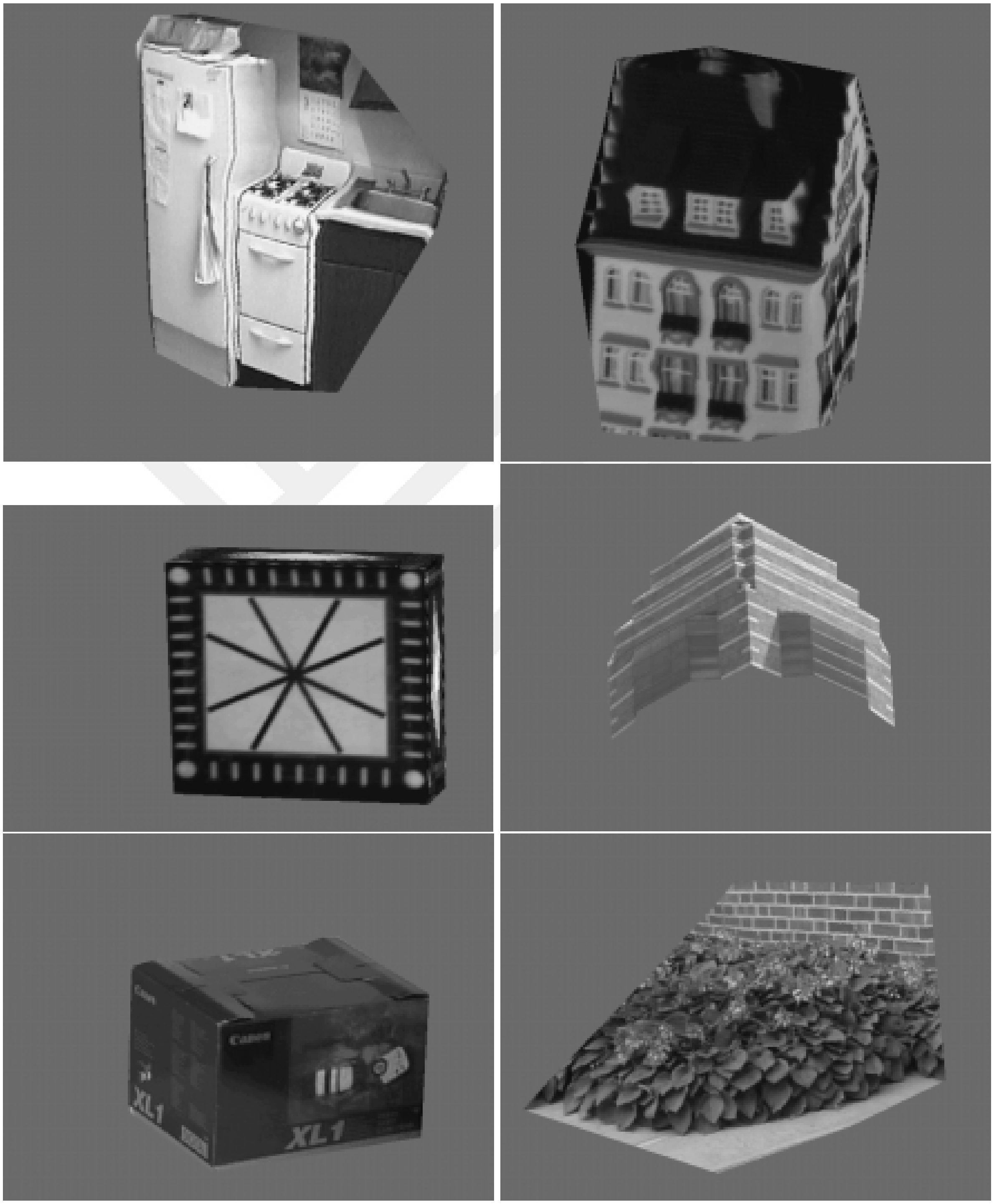


Figure 5.18 Image synthesis for novel views using point features for the data sets KITCHEN, HOUSE, LQBOX, TOWER, Xl1BOX and FLOWER.

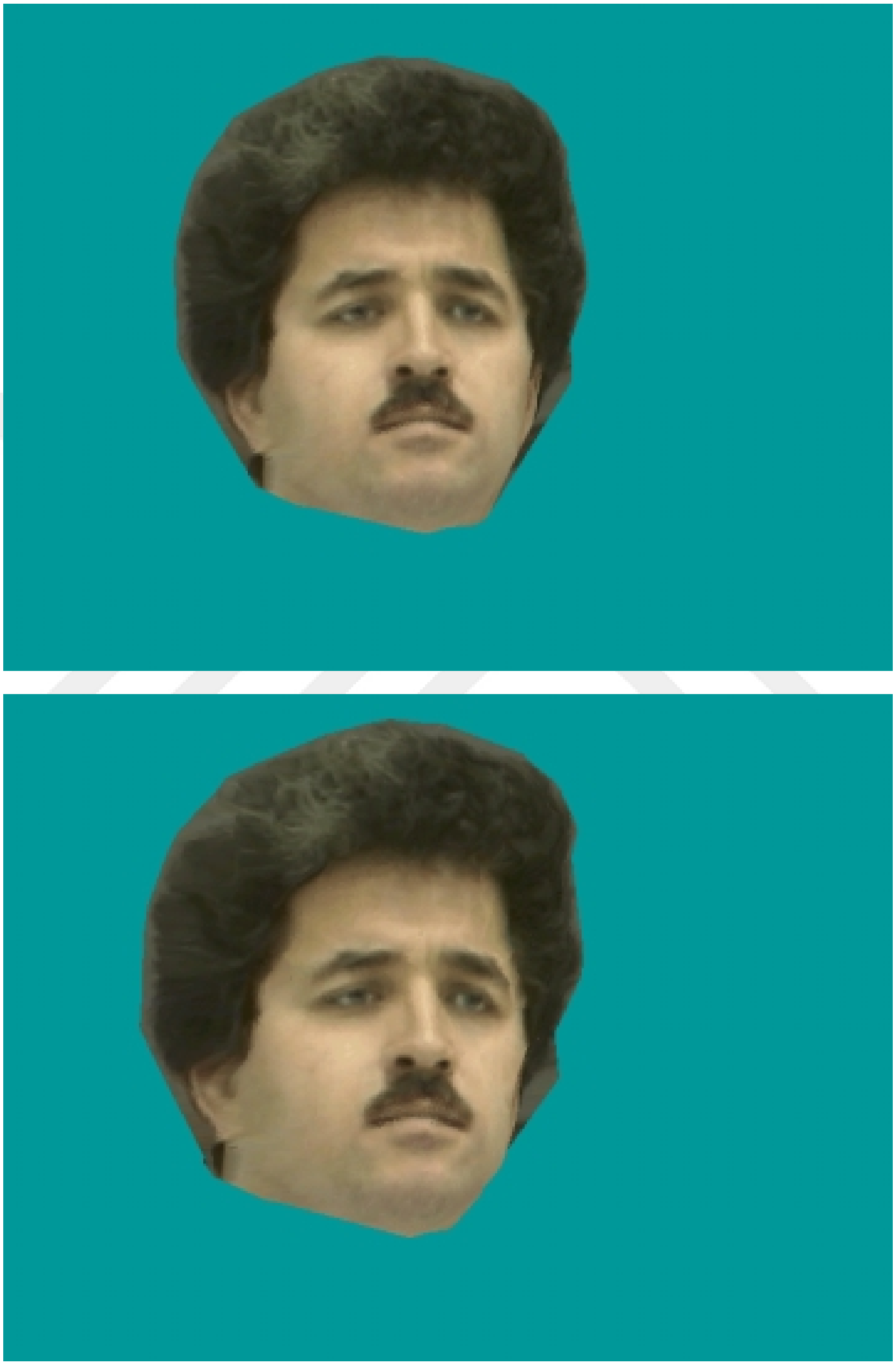


Figure 5.19 Image synthesis for novel views using point features for the FACE data set.

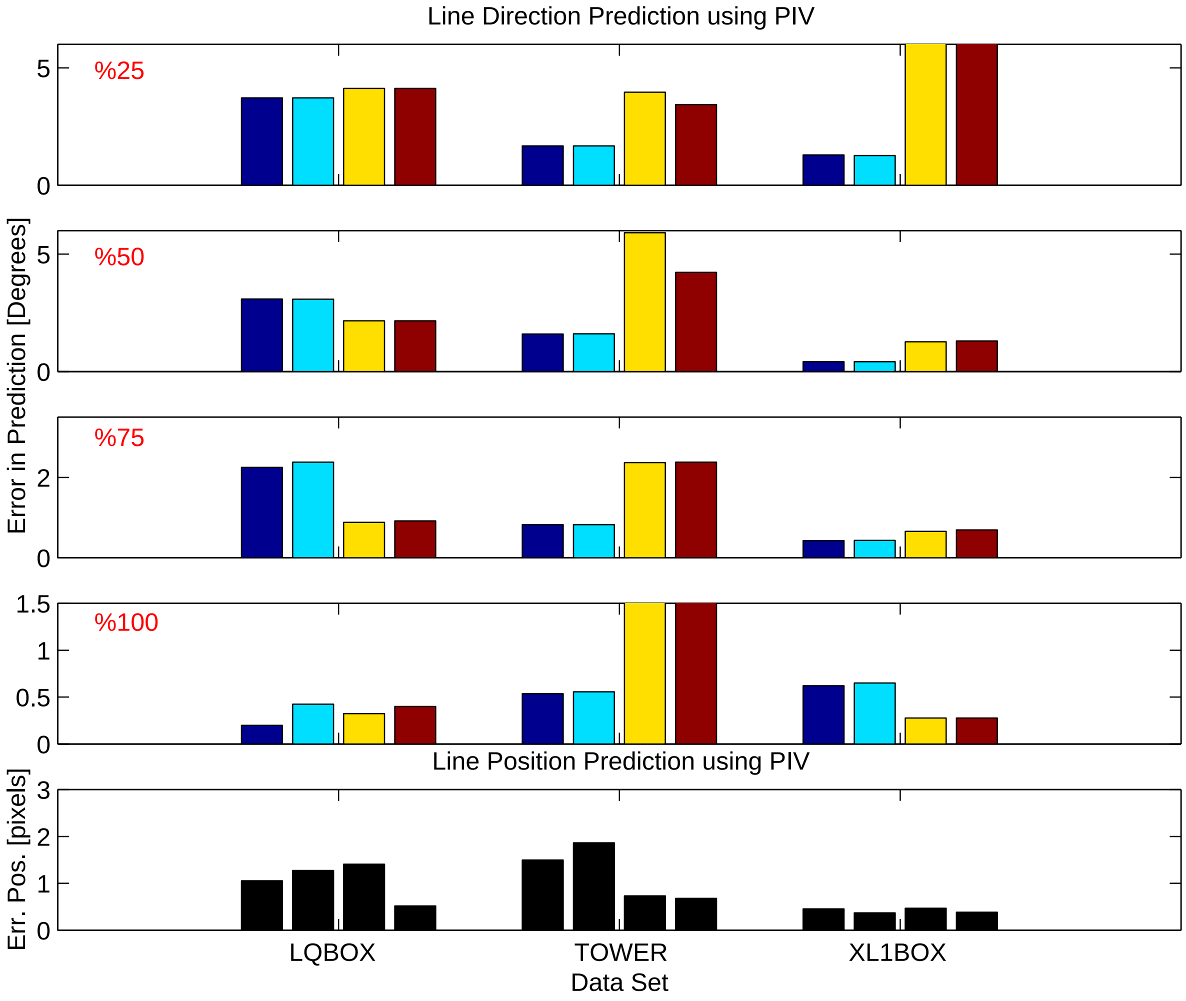


Figure 5.20 Image line reconstruction on real data sets. The upper four graphs show the error in line directions where the bars from left to right represents the W1, P1, W2 and P2 methods. From top to bottom, the first 25%, 50%, 75% and 100% of the images are used in training. The last graph shows the error in line position (from left to right the first 25%, 50%, 75% and 100% of the images are used in training. Note that the four graphs are not to the same scale.

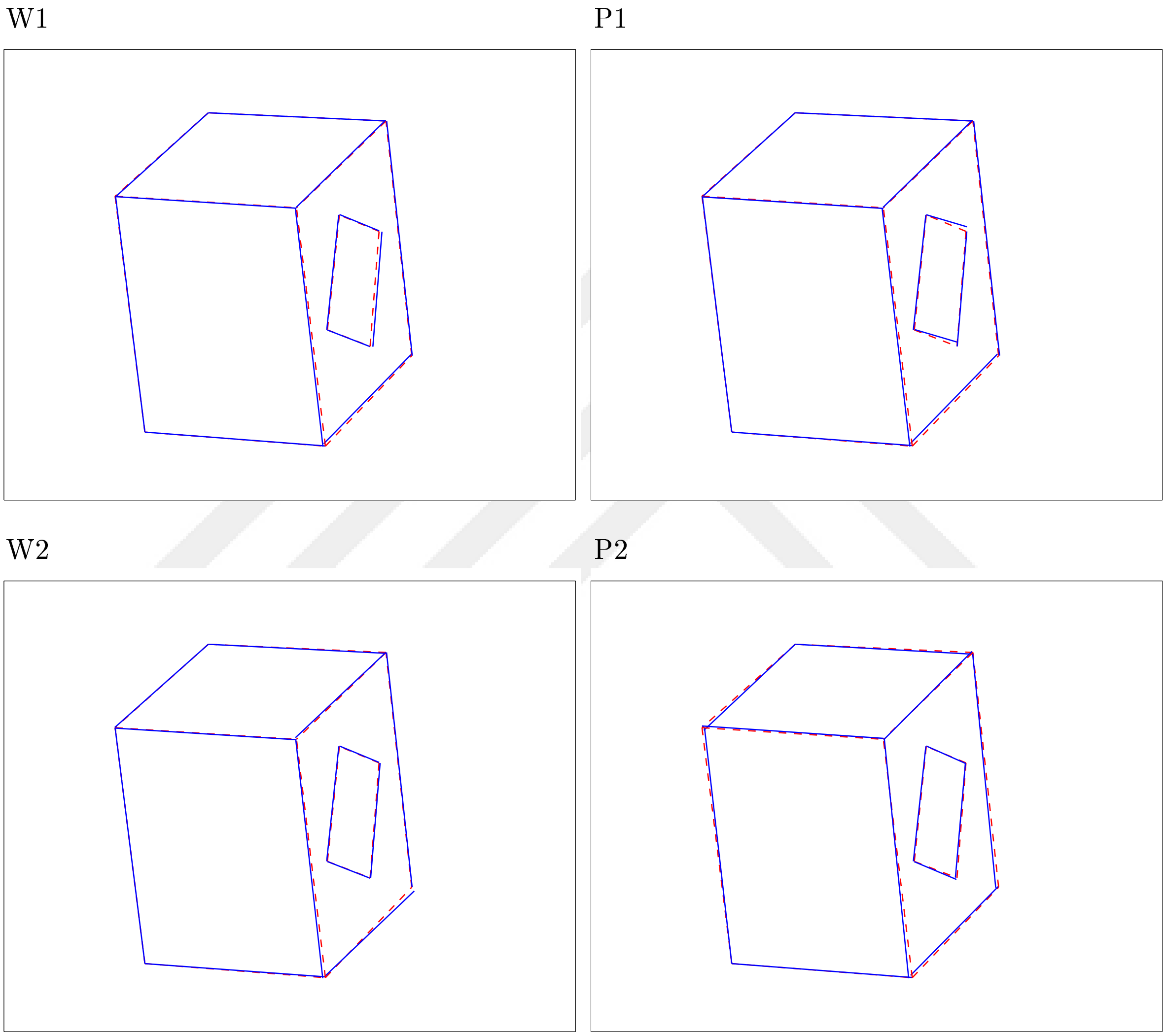


Figure 5.21 Reconstructed lines (solid) together with the original lines (dashed) for the last image in the LQBOX data set.

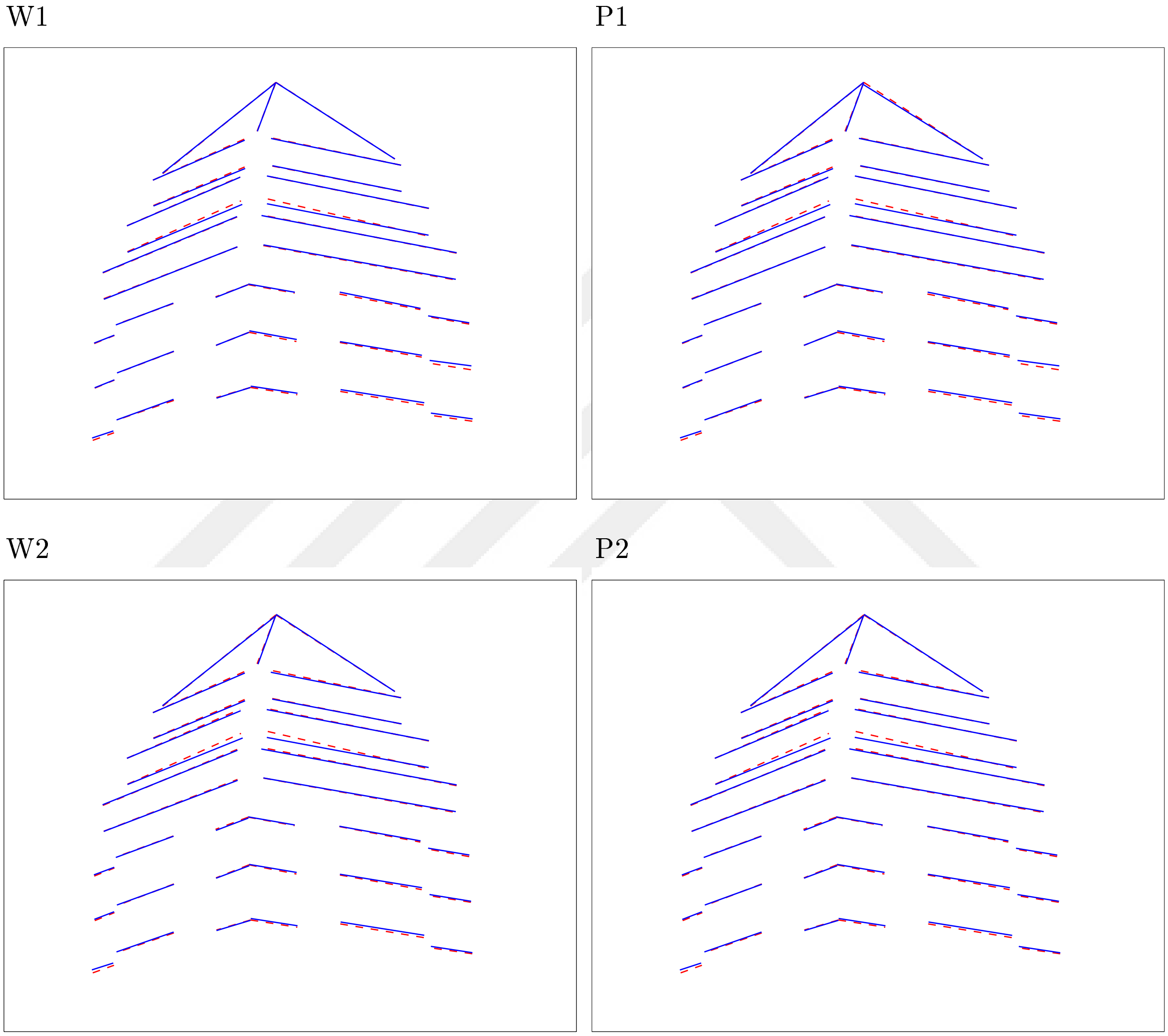


Figure 5.22 Reconstructed lines (solid) together with the original lines (dashed) for the last image in the TOWER data set.

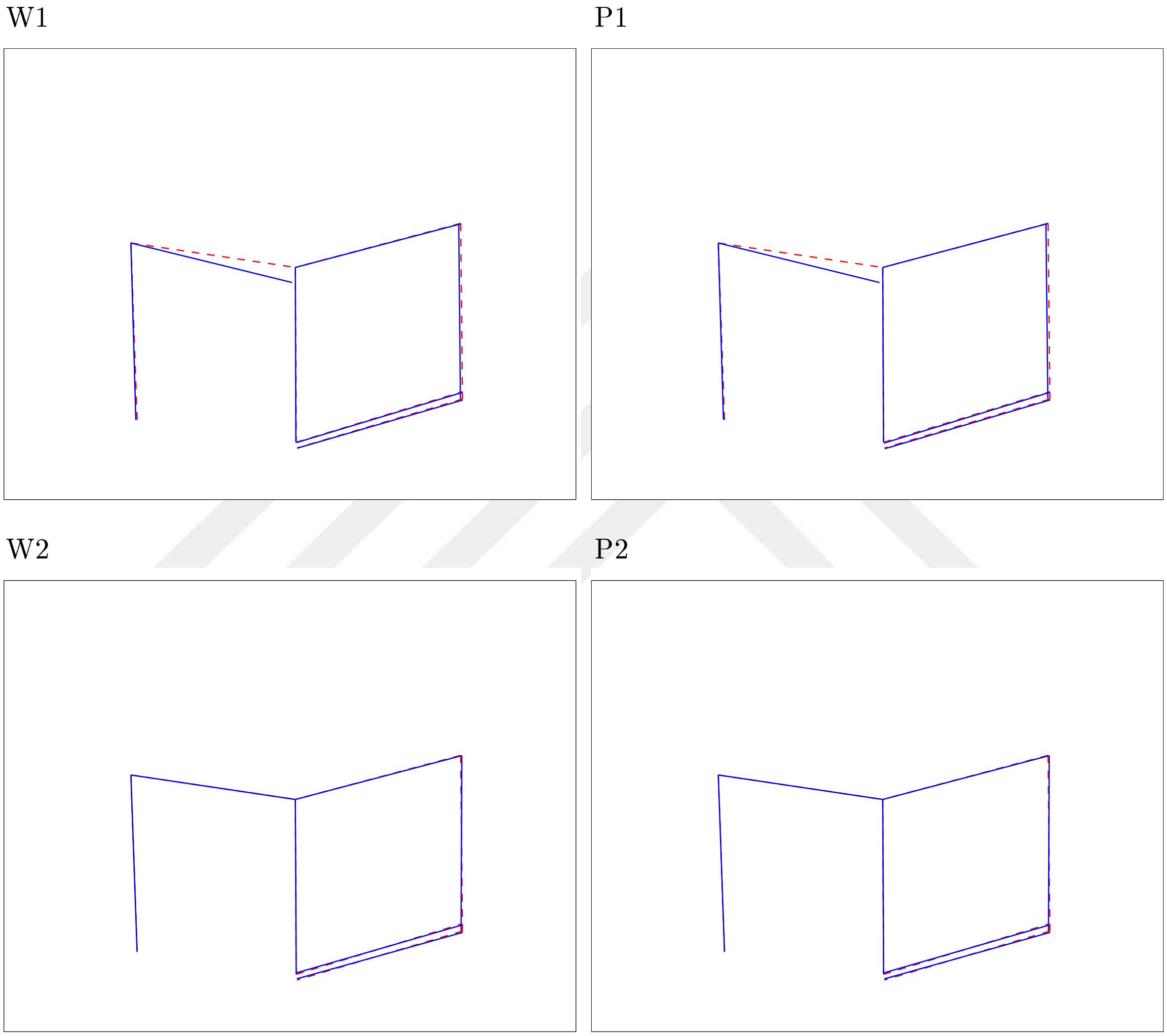


Figure 5.23 Reconstructed lines (solid) together with the original lines (dashed) for the last image in the XL1BOX data set.

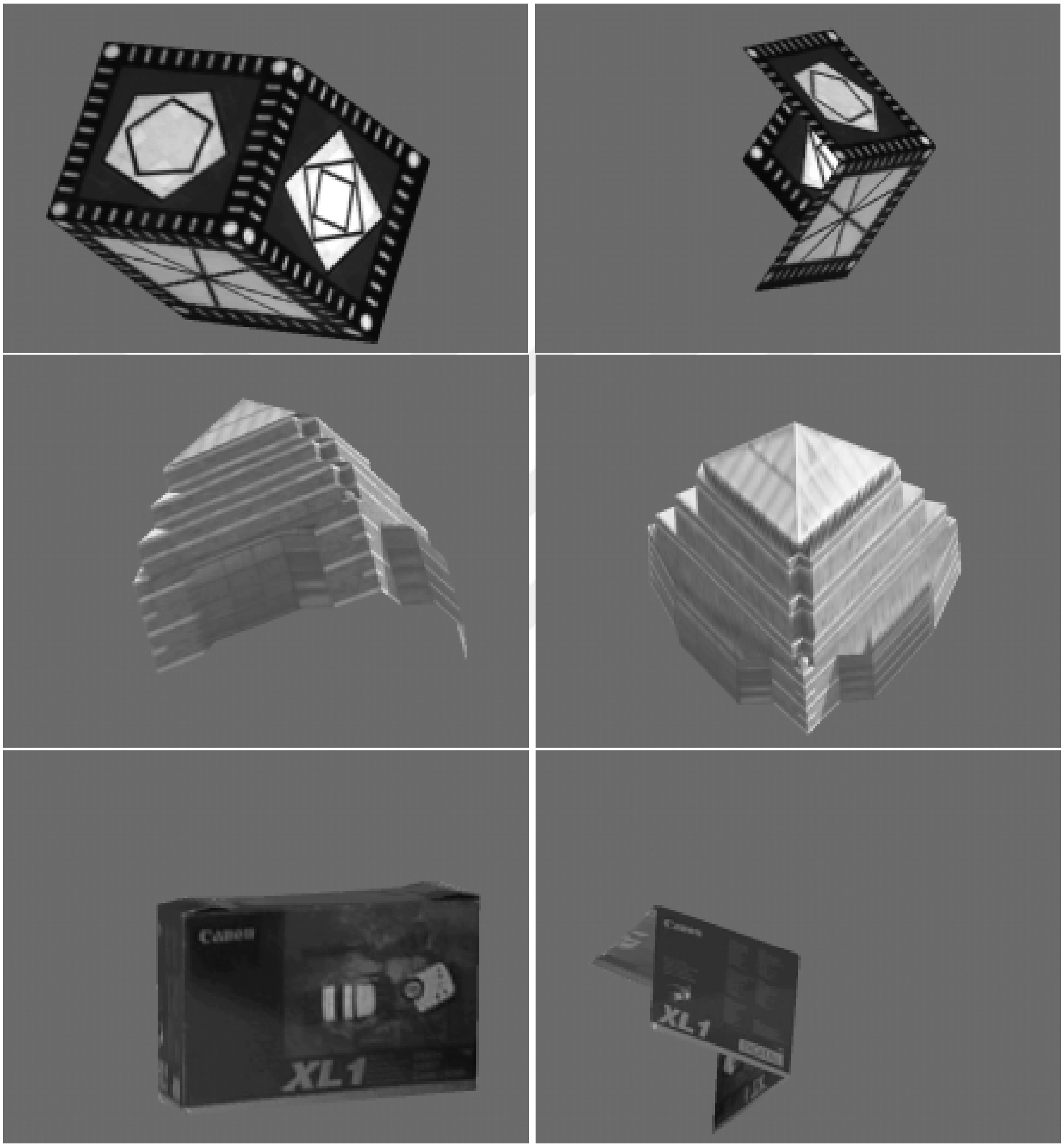


Figure 5.24 Image synthesis for novel views using both line and point features using the second pass of paraperspective case.

CHAPTER 6

BILINEAR ESTIMATION

As described in Chapter 5, the computation of the point-based parameterized image varieties for weak perspective cameras involves estimating parameters from a set of constraints which are linear in the parameters and bilinear in the image observations. For paraperspective cameras, the constraints are not bilinear but higher-order functions of the image observations. In the previous chapter, we have used linear least squares based on singular value decomposition to estimate the parameters in both cases. The underlying assumption is that the residual errors are linear, but for higher-order data dependencies, the residual errors are non-linear and, therefore, using linear least squares will bias the solution.

Optimal solutions to the parameter fitting problems of this type have been recently studied by Leedan and Meer in [59, 60] in the context of epipolar geometry estimation and ellipse fitting, and significant performance improvements over linear least squares have been reported. Their parameter estimation method takes into account the non-linear nature of the errors in the image observations. In this chapter, we will adapt this method to estimate the point PIVs in the context of image-based rendering.

The rest of this chapter is organized as follows. After presenting some background on the parameter fitting problems for higher-order data dependencies, we restate the parameter estimation problem from Section 5.2.1 and show in Section 6.2 how to apply the method described in [59] to the problem. For simplicity we first consider the weak perspective case, then give the solution for the paraperspective case. We present results of our experiments in Section 6.4.

6.1 Background

In this section, we will define a statistical model (*Errors-In-Variables* model) that best describes the parameter estimation problem encountered in Chapter 5 and state the reasons why we choose this model. We also provide some background necessary to follow the discussion in later sections (we largely follow [25]).

A statistical *model* is a specification of [25, p. 9]:

- the variables and the parameters of interest,
- the relationships among the variables,
- the assumptions about the statistical properties of the random variables.

In a parameter estimation problem the aim is to estimate the parameters of a model from the samples generated by this model. The variables are the data points and the noise terms. The actual observations are the data points perturbed by the noise terms. The model also defines the relationships (or constraints) among the variables and the parameters along with the assumptions about the stochastic properties of the random variables (e.g., noise).

A *functional, zero-residual, symmetrical, Errors-In-Variables (EIV)* model can be described by (for $i = 1, ..., n$, where n is the number of samples that the model generates):

- $\boldsymbol{\omega}_{io}$: the error-free observation vectors,
- $\boldsymbol{\theta}_o$: the true parameter vectors,
- $\Delta\boldsymbol{\omega}_i$: the noise vectors,
- $\boldsymbol{\omega}_i$: the noisy observations which are related to $\boldsymbol{\omega}_{io}$ with

$$\boldsymbol{\omega}_i = \boldsymbol{\omega}_{io} + \Delta\boldsymbol{\omega}_i, \tag{6.1}$$

- the continuous constraint function

$$\varphi(\omega_{io}, \theta_o) = 0, \tag{6.2}$$

- the assumption on the statistical properties of the noise, i.e.,

$$\Delta\boldsymbol{\omega}_i \sim \mathrm{GI}(0, \mathcal{C}_i). \tag{6.3}$$

This model assumes that the random noise vectors $\Delta\boldsymbol{\omega}_i$ are independently distributed with a general probability density function as described above. In particular, the noise is from a distribution with zero mean and different covariance matrices for each observation. Furthermore, it is assumed that the noise vectors are independent of each other.

In a *functional* model, the error-free observations $\boldsymbol{\omega}_{io}$ are treated as fixed. For this model, given the noisy observations, the error-free observations will also be estimated along with the parameters. The alternative is a *structural* model where the error-free observation vectors are considered as random variables with some unknown probability density functions and these functions are estimated as well. We have chosen the functional model because the error-free observations (i.e., the image points) actually correspond to a fixed value.

In a *symmetrical* model, there is no distinction between the observations. In other words, we do not seek any dependency among the data although such dependencies might exist. This model fits our problem since we do not emphasize a particular observation and seek no dependencies among the data.

A *zero-residual* model means that the constraint function has no error in it. We have chosen this model because every true data point obeys a model governed by a geometrical constraint (i.e., the Euclidean constraints).

Depending on the constraint function, a model can be either *linear* or *non-linear*. A *linear* model has a constraint function that is both linear in the observations and the parameters. A *non-linear* model, on the other hand, has functions that are non-linear in the observed variables. Note that the type of estimation problems we are considering in this chapter have constraint functions that are non-linear in the observations and we will show below how to linearize them to find an optimal solution.

6.2 The Errors-in-Variables Model

We propose in this section the errors-in-variables (or EIV) model to estimate the structure parameters as an alternative to the solution given in Section 5.2.1.1 and show that computing the PIV parameters reduces to a constrained optimization problem after a suitable linearization. In turn, using Lagrange multipliers reduces the constrained optimization to an unconstrained one, that can be solved iteratively using generalized singular value decomposition. The presentation is an adaptation of the method proposed by Leedan and Meer in [59].

6.2.1 Problem Statement

When f images have been observed, we can rewrite the constraints (5.14) defining the parameterized image varieties as

$$\boldsymbol{s}_i^T \boldsymbol{\xi} = 0, \quad \text{for} \quad i = 1, \dots, 2f, \tag{6.4}$$

where, for $j = 1, \dots, f$, $\boldsymbol{s}_{2j-1} \stackrel{\text{def}}{=} \boldsymbol{d}_j$ and $\boldsymbol{s}_{2j} \stackrel{\text{def}}{=} \boldsymbol{e}_j$.

Let us introduce

$$\boldsymbol{w}_j \stackrel{\text{def}}{=} \begin{pmatrix} \boldsymbol{u}_j \\ \boldsymbol{v}_j \end{pmatrix}, \quad \text{for} \quad j = 1, \dots, f, \tag{6.5}$$

where $\boldsymbol{u}_j$ and $\boldsymbol{v}_j$ are the values of the vectors $\boldsymbol{u}$ and $\boldsymbol{v}$ associated with image number j as defined in Section 5.2.1. Note that $\boldsymbol{s}_i$ is a non-linear function of the actual observation $\boldsymbol{w}_j$. This function is bilinear for the weak perspective cameras and a higher-order polynomial for the paraperspective cameras.

Since image measurements are not exact, the observation vector $\boldsymbol{w}_j$ is noisy and we assume that

$$\boldsymbol{w}_j = \boldsymbol{w}_{jo} + \Delta \boldsymbol{w}_{jo}, \tag{6.6}$$

where $\boldsymbol{w}_{jo}$ is the exact or error-free observation and $\Delta \boldsymbol{w}_{jo}$ is the noise distributed independently with a general, symmetric probability density function with zero mean and positive semidefinite matrix $\mathcal{C}_\omega$, i.e.,

$$\Delta \boldsymbol{w}_{jo} \sim \mathrm{GI}(0, \mathcal{C}_\omega). \tag{6.7}$$

Note that the image measurements are assumed to have been translated so the image $\boldsymbol{p}_0$ of the first reference point is at the origin. This does not change the noise distribution for the observations since the difference of two normally distributed random variables with zero mean and equal variances is also normal with zero mean and twice the original variance.

We are going to assume as in [59] that $\mathcal{C}_\omega = \sigma^2 \mathcal{C}_\omega^*$ is known up to the common multiple σ^2. As described in the previous section, this error process defines a non-linear, functional, zero-residual, symmetrical, errors-in-variables (EIV) model [25, 59]. The optimal solution to the non-linear EIV model estimation problem is obtained by minimizing the Mahalanobis distances between noisy and correct data, i.e., computing

$$[\hat{\boldsymbol{\xi}}, \hat{\boldsymbol{w}}_j] = \arg \min_{\boldsymbol{\xi}_o, \boldsymbol{w}_{jo}} \sum_{j=1}^{f} (\boldsymbol{w}_j - \boldsymbol{w}_{jo})^T \mathcal{C}_\omega^\dagger (\boldsymbol{w}_j - \boldsymbol{w}_{jo}), \tag{6.8}$$

subject to

$$\boldsymbol{s}_{io}^T \boldsymbol{\xi}_o = 0, \quad \text{for} \quad i = 1, \dots, 2f,$$

where, for $j = 1, \dots, f$, the vectors $\boldsymbol{s}_{2j-1,o}$ and $\boldsymbol{s}_{2j,o}$ are the values of $\boldsymbol{s}_{2j-1}$ and $\boldsymbol{s}_{2j}$ associated with $\boldsymbol{w}_{jo}$, and $\mathcal{C}_\omega^\dagger$ denotes the pseudo-inverse of $\mathcal{C}_\omega$.

Several solutions to (6.8) have been proposed, including total least squares [48] and the renormalization method of Kanatani [53]. These methods all involve approximations of the objective function in (6.8).

The total-least-squares solution [48] is obtained by assuming that the error in linearized observation vectors are from a normal distribution with zero mean and a common variance, i.e.,

$$\Delta \boldsymbol{s}_{io} \sim \mathrm{GI}(0, \sigma^2 \mathcal{I}). \tag{6.9}$$

Clearly this assumption may not hold in general since the observation is a non-linear function of the original observations with the same assumed error statistics. Since linear least squares is a specific case of total least squares, its solution is not optimal.

An effective (if potentially sub-optimal) solution can be found by solving instead a *linearized* version of the original problem as proposed in [59]. The idea is to use $\boldsymbol{s}_i$ as observable instead of $\boldsymbol{w}_j$: let us assume that for $j = 1, \dots, f$,

$$\begin{cases} \boldsymbol{d}_j = \boldsymbol{d}_{jo} + \Delta \boldsymbol{d}_{jo}, & \Delta \boldsymbol{d}_{jo} \sim \mathrm{GI}(0, \mathcal{C}_{dj}), \\ \boldsymbol{e}_j = \boldsymbol{e}_{jo} + \Delta \boldsymbol{e}_{jo}, & \Delta \boldsymbol{e}_{jo} \sim \mathrm{GI}(0, \mathcal{C}_{ej}), \end{cases} \tag{6.10}$$

where $\mathcal{C}_{dj} = \sigma^2 \mathcal{C}_{dj}^*$ is the covariance matrix (with σ^2 being the common multiple and $\mathcal{C}_{dj}^*$ being the normalized covariance matrix) and similarly $\mathcal{C}_{ej} = \sigma^2 \mathcal{C}_{ej}^*$. We can restate the errors-in-variables model of the estimation process as the computation of

$$[\hat{\boldsymbol{\xi}}, \hat{\boldsymbol{s}}_i] = \arg \min_{\boldsymbol{\xi}_o, \boldsymbol{s}_{io}} \sum_{i=1}^{2f} (\boldsymbol{s}_i - \boldsymbol{s}_{io})^T \mathcal{C}_i^\dagger (\boldsymbol{s}_i - \boldsymbol{s}_{io}), \tag{6.11}$$

subject to

$$\boldsymbol{s}_{io}^T \boldsymbol{\xi}_o = 0 \quad \text{for} \quad i = 1, \ldots, 2f,$$

where

$$\mathcal{C}_{2j-1} \overset{\text{def}}{=} \mathcal{C}_{dj} \quad \text{and} \quad \mathcal{C}_{2j} \overset{\text{def}}{=} \mathcal{C}_{ej} \quad \text{for} \quad j = 1, \ldots, f.$$

Note that the covariance matrices $\mathcal{C}_i$ are by definition symmetric positive semidefinite.

6.2.2 Lagrange Multipliers Formulation

The constrained minimization problem in (6.11) can be solved by assuming that the covariance matrices $\mathcal{C}_{dj}^*$ and $\mathcal{C}_{ej}^*$ (hence $\mathcal{C}_i^*$) are known and converting the problem into an unconstrained minimization problem. To do this, we introduce Lagrange multipliers and replace the true parameters with the estimated ones. This reduces the original optimization problem to the problem of minimizing

$$L \overset{\text{def}}{=} \frac{1}{2} \sum_{i=1}^{2f} (\boldsymbol{s}_i - \hat{\boldsymbol{s}}_i)^T \mathcal{C}_i^{\dagger} (\boldsymbol{s}_i - \hat{\boldsymbol{s}}_i) + \sum_{i=1}^{2f} \eta_i \hat{\boldsymbol{s}}_i^T \hat{\boldsymbol{\xi}} \tag{6.12}$$

with respect to the original unknowns plus the Lagrange multipliers η_i associated with the constraints for $i = 1, \ldots, 2f$.

Differentiating L with respect to the unknowns $\hat{\boldsymbol{\xi}}$, $\hat{\boldsymbol{s}}_i$ and η_i allows us (after some simple algebraic manipulations) to reformulate our original problem as the solution the following eigenproblem in the unknown $\hat{\boldsymbol{\xi}}$:

$$\mathcal{P}\hat{\boldsymbol{\xi}} - \mathcal{Q}\hat{\boldsymbol{\xi}} = 0, \tag{6.13}$$

where

$$\mathcal{P} \overset{\text{def}}{=} \sum_{i=1}^{2f} \frac{\boldsymbol{s}_i \boldsymbol{s}_i^T}{\hat{\boldsymbol{\xi}}^T \mathcal{C}_i^* \hat{\boldsymbol{\xi}}} \tag{6.14}$$

is the *weighted, centered moment matrix* of the observation and

$$\mathcal{Q} \overset{\text{def}}{=} \sum_{i=1}^{2f} \left(\frac{\boldsymbol{s}_i^T \hat{\boldsymbol{\xi}}}{\hat{\boldsymbol{\xi}}^T \mathcal{C}_i^* \hat{\boldsymbol{\xi}}} \right)^2 \mathcal{C}_i^* \tag{6.15}$$

is the *weighted error covariance matrix*. Note that $\mathcal{P}$ and $\mathcal{Q}$ also depend on the unknown $\hat{\boldsymbol{\xi}}$. As an intermediate step of elimination of the Lagrange multipliers, we have

$$\hat{\boldsymbol{s}}_i = \boldsymbol{s}_i - \frac{\hat{\boldsymbol{\xi}}\boldsymbol{s}_i}{\hat{\boldsymbol{\xi}}^T \mathcal{C}_i \hat{\boldsymbol{\xi}}} \mathcal{C}_i \hat{\boldsymbol{\xi}} \tag{6.16}$$

which can be used to estimate $\boldsymbol{s}_{io}$.

This formulation is of course very similar (if a bit simpler, due to the absence of a data-independent term) to the general formulation obtained in [59].

The approach described in this section assumes that the normalized covariance matrices are known. Expressions for the covariance matrices $\mathcal{C}^*_{dj}$ and $\mathcal{C}^*_{ej}$ for $j = 1, ..., f$, are derived in the next section.

6.2.3 Computing the Covariance Matrices

Let us first rewrite the linearized random variables $\Delta\boldsymbol{d}_o$ and $\Delta\boldsymbol{e}_o$ as a function of the true and noisy image measurements (ignoring the subscript j for convenience):

$$\Delta\boldsymbol{d}_o = \begin{pmatrix} u_1^2 - v_1^2 - u_{1o}^2 + v_{1o}^2 \\ 2(u_1u_2 - v_1v_2 - u_{1o}u_{2o} + v_{1o}v_{2o}) \\ u_2^2 - v_2^2 - u_{2o}^2 + v_{2o}^2 \\ 2(u_1u - v_1v - u_{1o}u_o + v_{1o}v_o) \\ 2(u_2u - v_2v - u_{2o}u_o + v_{2o}v_o) \\ u^2 - v^2 - u_o^2 + v_o^2 \end{pmatrix} \tag{6.17}$$

and

$$\Delta\boldsymbol{e}_o = \begin{pmatrix} u_1v_1 - u_{1o}v_{1o} \\ u_1v_2 + u_2v_1 - u_{1o}v_{2o} - u_{2o}v_{1o} \\ u_2v_2 - u_{2o}v_{2o} \\ u_1v + uv_1 - u_{1o}v_o - u_ov_{1o} \\ u_2v + uv_2 - u_{2o}v_o - u_ov_{2o} \\ uv - u_ov_o \end{pmatrix}. \tag{6.18}$$

We have assumed that the image noise has zero mean and a variance equal to σ^2, in other words

$$\Delta\boldsymbol{w}_o \sim \mathrm{NI}(0, \sigma^2\mathcal{I}).$$

We first eliminate the noisy image observations u and v in $\Delta \boldsymbol{d}_o$ and $\Delta \boldsymbol{e}_o$ by replacing their equivalents in terms of the noise-free observations and the noise (i.e., $u = u_0 + \Delta u_0$ and $v = v_0 + \Delta v_0$). Now, when we take the expectations of the vectors $\Delta \boldsymbol{d}_o$ and $\Delta \boldsymbol{e}_o$ with respect to the actual random variables Δu_0 and Δv_0, it is easy to show that the means for $\Delta \boldsymbol{d}_o$ and $\Delta \boldsymbol{e}_o$ are zero and the associated covariance matrices are

$$\mathcal{C}_d^* = \begin{pmatrix} a_{11} & a_7 & 0 & a_8 & 0 & 0 \\ a_7 & a_{22} & a_7 & a_9 & a_8 & 0 \\ 0 & a_7 & a_{33} & 0 & a_9 & 0 \\ a_8 & a_9 & 0 & a_{44} & a_7 & a_8 \\ 0 & a_8 & a_9 & a_7 & a_{55} & a_9 \\ 0 & 0 & 0 & a_8 & a_9 & a_{66} \end{pmatrix} \quad \text{and} \quad \mathcal{C}_e^* = 4\mathcal{C}_d^*, \tag{6.19}$$

with

$$\begin{cases} a_{11} = \sigma^2 + {v_{1o}}^2 + {u_{1o}}^2, \\ a_{22} = 2\,\sigma^2 + {v_{2o}}^2 + {u_{1o}}^2 + {v_{1o}}^2 + {u_{2o}}^2, \\ a_{33} = \sigma^2 + {v_{2o}}^2 + {u_{2o}}^2, \\ a_{44} = 2\,\sigma^2 + {v_o}^2 + {v_{1o}}^2 + {u_o}^2 + {u_{1o}}^2, \\ a_{55} = 2\,\sigma^2 + {v_o}^2 + {u_{2o}}^2 + {v_{2o}}^2 + {u_o}^2, \\ a_{66} = \sigma^2 + {v_o}^2 + {u_o}^2, \\ a_7 = v_{1o}v_{2o} + u_{1o}u_{2o}, \\ a_8 = v_o v_{1o} + u_{1o}u_o, \\ a_9 = v_{2o}v_o + u_{2o}u_o. \end{cases}$$

Note that the entries of the matrices $\mathcal{C}_d^*$ and $\mathcal{C}_e^*$ depend on the noise variance σ^2 and the error-free observation vector $\boldsymbol{w}_o$ that are both a priori unknown. A method for estimating these quantities will be discussed below.

6.2.4 Solving the Eigenproblem

The difficulty in solving the eigenproblem (6.13) stems from the fact that the matrices $\mathcal{P}$ and $\mathcal{Q}$ depend on the unknown parameter vector $\hat{\boldsymbol{\xi}}$, which requires an iterative scheme to estimate $\hat{\boldsymbol{\xi}}$. Following [59], we repeatedly compute the generalized eigenvector $\hat{\boldsymbol{\xi}}^{(k)}$ associated with the smallest eigenvalue $\lambda_{\min}$ of

$$\mathcal{P}^{(k-1)}\hat{\boldsymbol{\xi}}^{(k)} = \lambda \mathcal{Q}^{(k-1)}\hat{\boldsymbol{\xi}}^{(k)}, \tag{6.20}$$

where $\mathcal{P}^{(k-1)}$ and $\mathcal{Q}^{(k-1)}$ are the matrices $\mathcal{P}$ and $\mathcal{Q}$ associated with the value $\hat{\boldsymbol{\xi}}^{(k-1)}$ of $\hat{\boldsymbol{\xi}}$. The iterative process is stopped when $|\lambda_{\min} - 1|$ or the variation in the objective function L becomes smaller than a preset threshold. For this eigenproblem, the Rayleigh quotient is

$$\lambda = \frac{\hat{\boldsymbol{\xi}}^{(k)^T} \mathcal{P}^{(k-1)}\hat{\boldsymbol{\xi}}^{(k)}}{\hat{\boldsymbol{\xi}}^{(k)^T} \mathcal{Q}^{(k-1)}\hat{\boldsymbol{\xi}}^{(k)}} \tag{6.21}$$

and when $\hat{\boldsymbol{\xi}}^{(k)} = \hat{\boldsymbol{\xi}}^{(k-1)}$ it can be shown that $\lambda = 1$, and the convergence can be measured from $|\lambda_{\min} - 1|$.

In the actual implementation, we also follow [59] and use generalized singular value decomposition [29] to improve the numerical behavior of the method. If $\mathcal{P}_c^{(k-1)}$ and $\mathcal{Q}_c^{(k-1)}$ are the Cholesky decompositions of the matrices $\mathcal{P}^{(k-1)}$ and $\mathcal{Q}^{(k-1)}$ respectively, the solution $\hat{\boldsymbol{\xi}}^{(k)}$ is obtained by computing the generalized singular value decomposition of the matrices $\mathcal{P}_c^{(k-1)}$ and $\mathcal{Q}_c^{(k-1)}$ and taking the solution associated with the smallest generalized singular value. Even though the convergence is not guaranteed, our experiments have shown that with a reliable initial estimate the algorithm converges after a few steps, in agreement with the experiments of [59].

This method requires an initial estimate of $\hat{\boldsymbol{\xi}}^{(0)}$ which can be obtained as follows [59]: we can approximate the known normalized covariance matrices with

$$\mathcal{C}_i^* = \gamma_i \mathcal{C}^*, \tag{6.22}$$

where $\mathcal{C}^*$ is not known. We can find $\mathcal{C}^*$ and γ_i via minimizing the Frobenius norm of the matrix $\mathcal{C}_i^* - \gamma_i \mathcal{C}^*$ by taking $\gamma_i = 1$ as the starting value. When we plug $\mathcal{C}_i^*$ in (6.20) we obtain

$$\mathcal{P}_0\hat{\boldsymbol{\xi}}^{(0)} - \lambda^* \mathcal{C}^* \hat{\boldsymbol{\xi}}^{(0)} = 0, \tag{6.23}$$

where

$$\mathcal{P}_0 = \sum_{i=1}^{2f} \frac{\boldsymbol{s}_i \boldsymbol{s}_i^T}{\gamma_i}. \tag{6.24}$$

As before we can solve for $\hat{\boldsymbol{\xi}}^{(0)}$ using generalized singular value decomposition on the Cholesky decompositions of matrices $\mathcal{P}_0$ and $\mathcal{C}^*$.

To compute the approximate covariance matrices as described in Section 6.2.3, we need to find the true observation vectors $\boldsymbol{\omega}_{io}$ and the noise variance σ^2. We can compute initial estimates for $\hat{\boldsymbol{\xi}}^{(0)}$ as described above, which is independent of $\hat{\sigma}^2$. However, estimation of $\hat{\boldsymbol{\xi}}^{(0)}$ still requires the covariance matrix $\mathcal{C}^*$. By neglecting the higher-order moments (i.e., assuming zero mean), we can take $\mathcal{C}^*$ as the weighted sum of $\mathcal{C}_i^*$ evaluated at $\boldsymbol{\omega}_{io} = \boldsymbol{\omega}_i$ in (6.19). Once we have $\hat{\boldsymbol{\xi}}^{(0)}$, $\hat{\sigma}^2$ can be estimated from

$$\hat{\sigma}^2 = \frac{\hat{\boldsymbol{\xi}}^{(0)^T} \mathcal{P}^{(0)} \hat{\boldsymbol{\xi}}^{(0)}}{2f - 6}. \tag{6.25}$$

Finally, an initial estimate of $\hat{\boldsymbol{s}}_i^{(0)}$ hence $\hat{\boldsymbol{\omega}}_i^{(0)}$ can be obtained using $\hat{\boldsymbol{\xi}}^{(0)}$ in (6.16).

6.2.5 Summary of the Algorithm

Figure 6.1 shows the various steps of the proposed approach to the parameter estimation problem for weak-perspective point PIVs. Note that we are mostly interested in the estimation of the parameters. However, the estimates for the error-free data can be obtained easily as the last step of the process.

6.3 The Paraperspective Case

The ideas described in the previous sections also apply to the paraperspective case. However, this time the constraints are no longer bilinear in the observations. In particular, for the linearized model we have

$$\boldsymbol{s}_i^T \boldsymbol{\xi} = 0, \quad \text{for} \quad i = 1, \ldots, 2f, \tag{6.26}$$

where, for $j = 1, \ldots, f$, $\boldsymbol{s}_{2j-1} \stackrel{\text{def}}{=} \boldsymbol{d}_j$ and $\boldsymbol{s}_{2j} \stackrel{\text{def}}{=} \boldsymbol{e}_j$. But this time

$$\boldsymbol{d}_j \stackrel{\text{def}}{=} (1 + u_{0j}^2)\boldsymbol{d}'_{1j} - (1 + v_{0j}^2)\boldsymbol{d}'_{2j} \tag{6.27}$$

and

$$\boldsymbol{e}_j \stackrel{\text{def}}{=} (1 + u_{0j}^2)(1 + v_{0j}^2)\boldsymbol{d}'_j - u_{0j}v_{0j}(1 + u_{0j}^2)\boldsymbol{d}'_{1j} - u_{0j}v_{0j}(1 + v_{0j}^2)\boldsymbol{d}'_{1j}, \tag{6.28}$$

1. Initialization
   (a) Compute $\mathcal{C}_i^{*(0)}$, γ_i and $\mathcal{C}^*$ as described in Section 6.2.4.
   (b) Estimate $\hat{\boldsymbol{\xi}}^{(0)}$ as described in Section 6.2.4.
   (c) Estimate $\hat{\sigma}^2$ from (6.25).
   (d) Estimate $\hat{\omega}_i^{(0)}$ from (6.16).
   (e) Compute the covariance matrices $\mathcal{C}_i^*$ from (6.19).
2. Repeat the following until $|\lambda_{\min} - 1|$ or the variation in the objective function L become smaller than a preset threshold.
   - Estimate $\hat{\boldsymbol{\xi}}^{(k)}$ as described in Section 6.2.2.
3. Compute $\hat{\omega}_i$ from (6.16).

Figure 6.1 The EIV parameter estimation algorithm.

where

$$\boldsymbol{d}'_{1j} \overset{\text{def}}{=} \begin{pmatrix} u_{1j}^2 \\ 2u_{1j}u_{2j} \\ u_{2j}^2 \\ 2u_{1j}u_j \\ 2u_{2j}u_j \\ u_j^2 \end{pmatrix}, \quad \boldsymbol{d}'_{2j} \overset{\text{def}}{=} \begin{pmatrix} v_{1j}^2 \\ 2v_{1j}v_{2j} \\ v_{2j}^2 \\ 2v_{1j}v_j \\ 2v_{2j}v_j \\ v_j^2 \end{pmatrix} \quad \text{and} \quad \boldsymbol{d}'_j \overset{\text{def}}{=} \begin{pmatrix} u_{1j}v_{1j} \\ u_{1j}v_{2j} + u_{2j}v_{1j} \\ u_{2j}v_{2j} \\ u_{1j}v_j + u_jv_{1j} \\ u_{2j}v_j + u_jv_{2j} \\ u_jv_j \end{pmatrix}. \tag{6.29}$$

In this case, it is not straightforward to find the covariance matrices for the linearized observation vectors. However, if we assume that the images of the first reference point are noise free, we can compute the covariance matrices associated with the linearized random variables $\Delta\boldsymbol{d}_j$ and $\Delta\boldsymbol{e}_j$ as we have done for the weak perspective case in Section 6.2.3. We have used MAPLE symbolic algebra package to find these covariance matrices and due to the complexity of their expressions we are not able to show them here. The results for the paraperspective case in the following section is based on this algorithm.

Table 6.1 Abbreviations used to denote the variants of the PIV algorithm.

	Estimation Algorithm	Projection Model	Second Pass
SW1	SVD	weak perspective	no
EW1	EIV	weak perspective	no
SW2	SVD	weak perspective	yes
EW2	EIV	weak perspective	yes
SP1	SVD	paraperspective	no
EP1	EIV	paraperspective	no
SP2	SVD	paraperspective	yes
EP2	EIV	paraperspective	yes

6.4 Implementation and Results

We have implemented the proposed parameter estimation algorithm in MATLAB and tested it on real as well as synthetic data sets. We have compared the performance of the new parameter estimation algorithm with the linear-least-squares method based on singular value decomposition. The variants of the algorithms and their abbreviations are shown in Table 6.1. Note that the second passes of the SVD- and EIV-based algorithms involve the same procedure, namely, using the estimated structure parameters and refining them as explained in Section 5.5.

6.4.1 Synthetic Experiments

We have tested the various algorithms discussed in this and the previous chapters for point-based PIVs on synthetic data sets. The synthetic images have been created as 512×512 weak-perspective images of 20 points inside the unit cube. Then the images of all points including those of the reference points have been perturbed by zero mean Gaussian noise with varying standard deviations. We have divided the noise-added synthetic data into training and testing images and estimated the structure parameters using the training images and then tested the computed parameters by reconstructing the images of the points in the test images. We have kept the size of the test images fixed (i.e., 50 images). Note that we have repeated each experiment for 50 times and recorded the average errors in reconstructions. Furthermore, we have chosen the three reference points at random to reduce any bias towards a particular choice.

The first experiment have been carried out to measure the performance of the algorithms with respect to the level of noise added to the images. For this we have used 10 randomly generated training images with added noise whose amount (standard deviation) is varied from 0 to 1.5 pixel with 0.1 pixel increments. Figure 6.2 shows the average errors in the reconstruction of images of points. We have identified two different cases for the training images. In the first case (the upper graph) the camera has no restrictions on its motion. In the second case (the lower graph) however we have restricted the camera so that mostly sees a fixed face of the unit cube. Clearly the second case is a more challenging image-based rendering problem. As it can be seen from these results, the EIV estimation method improves the results consistently. The performance improvement is more visible for the second case where we have less viewing variations in the training images.

The second experiment has been conducted to observe the performance of the algorithms with respect to the number of images in the training data. By adding noise with standard deviation of 0.5 pixels, we have changed the number of images in the training data from 3 to 20 images. The results are depicted in Figure 6.3. Once again, the EIV method improves the results consistently though the difference becomes small when we have more images in the training. Moreover, as before the difference is more visible for the second case where we have less viewing variations in the training images.

6.4.2 Experiments with Real Data

We have conducted experiments on the real data sets described earlier in Chapter 5. The first test has been done by using different amount of images in training. In particular, we have used the first 25%, 50% and 75% of the images in training to compute the parameters and used these parameters to predict the image positions of all data points in the rest of the images. We also used all the images in each data set to reconstruct the parameters and predict the image positions in all of the images. The average errors in the predictions are displayed in Figure 6.4. In this figure, we have four different training cases as indicated above. Also, for each data set, we have 8 bars representing the eight variants of the algorithm in the following order (from the left to the right): SW1, EW1, SW2, EW2, SP1, EP1, SP2 and EP2. Once again we see that the EIV method consistently gives better results when only the first pass is used. Interestingly, the second pass does not always improve the results.

Figures 6.5-6.16 show more comparisons, qualitative this time: in this case, the views corresponding to the last image in each data set generated by the various methods are shown. We have used first half of the images in training. Again, the EIV estimation algorithm has performed better especially for the KITCHEN and XL1BOX data sets where difference in image quality is clearly noticeable.

6.4.3 Comparisons with Factorization

In this section we compare our approach to the factorization method of Tomasi and Kanade [108] as explained in Chapter 4. We first compare both methods quantitatively as follows: we have used all of the images in the training to compute the point-based PIV for our approach and the Euclidean structure and the motion for the factorization method. We then have reconstructed all of the images using both methods and recorded the error statistics in the reconstructions. Figure 6.17 shows the results of this experiment. Note that, for the TOWER and FLOWER data sets, our implementation of normalization (i.e., enforcing the Euclidean constraints) for factorization had failed, therefore, we present the results of the experiments with the other four data sets. These experiments shows that the EW1 variant of our algorithm and the factorization method performed quite well.

Completely new views of the scenes are synthesized using the factorization approach and the EW1 variant of the PIV algorithm. The results are shown in Figures 6.18 and 6.18. The new views are obtained as follows: first the motion of the camera is picked (i.e., the translation and the rotation of the camera), then the scene is reconstructed using the structure obtained from the factorization algorithm, and finally using the images of the reference of this new view, the corresponding image is reconstructed using the EW1 algorithm.

6.5 Conclusions

We have applied the parameter estimation procedure for bilinear constraints described by Leedan and Meer in [59] to the estimation of the point-based parameterized image varieties in the context of image-based rendering. Our experiments with real and synthetic data demonstrated that the new parameter estimation technique consistently yields better results than those of the linear least squares approach based on singular value decomposition.

Both the weak perspective and paraperspective cases are considered. The weak perspective case yields very simple expressions for the linearized observation vectors as functions of the real image measurements. The mean of the corresponding random variable is zero and the normalized covariance matrices include second order moments. Unfortunately, the paraperspective case has more complex linearized observations vectors which makes it impossible to adapt the original algorithm. Instead, we have assumed that the first reference point is noise free resulting in a method similar to the weak perspective case.

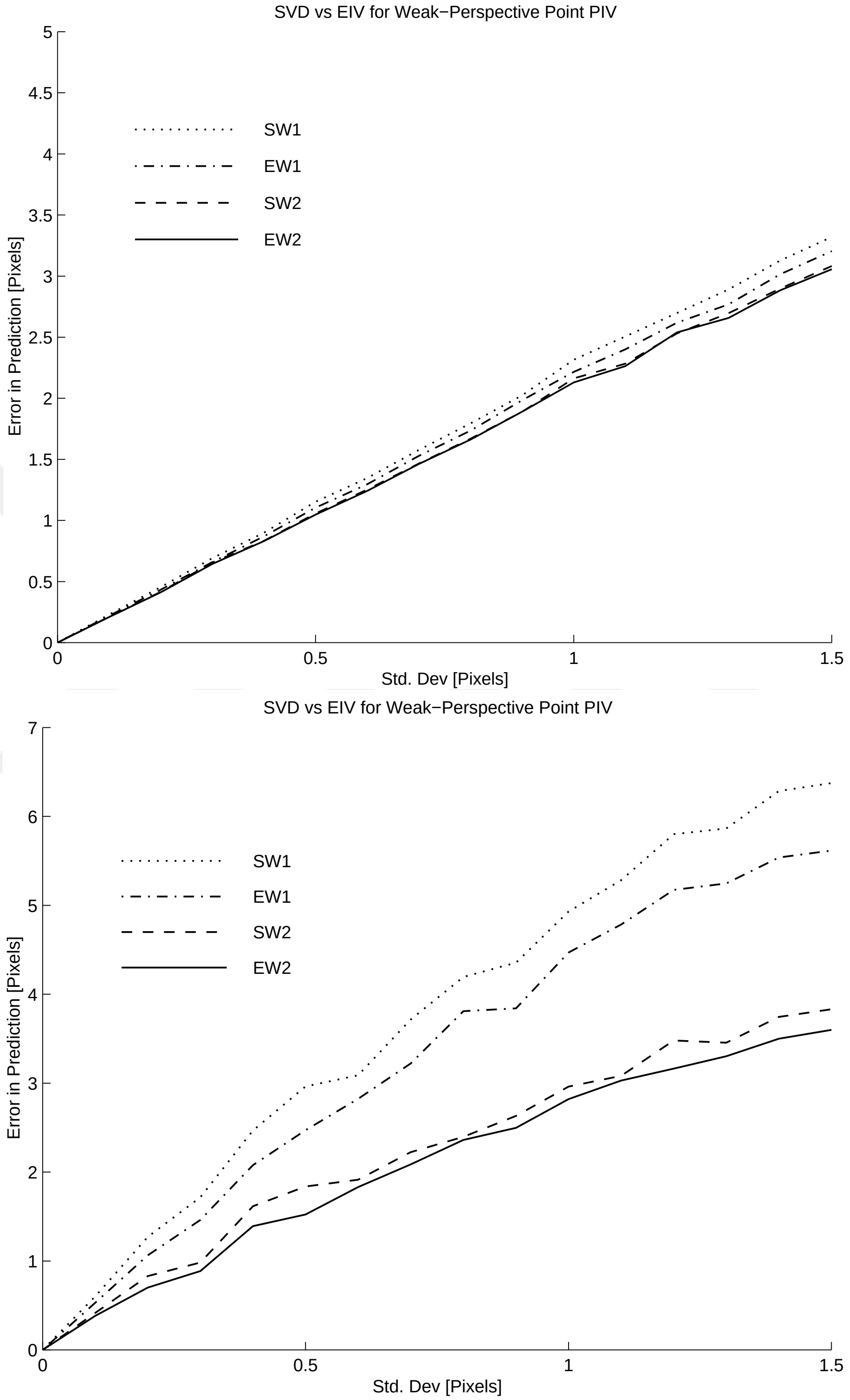


Figure 6.2 Performance of the weak perspective SVD- and EIV-based methods with respect to the amount of noise: the parameters have been computed using 10 images and the mean error in reconstruction of image point locations has been recorded for 50 test images.

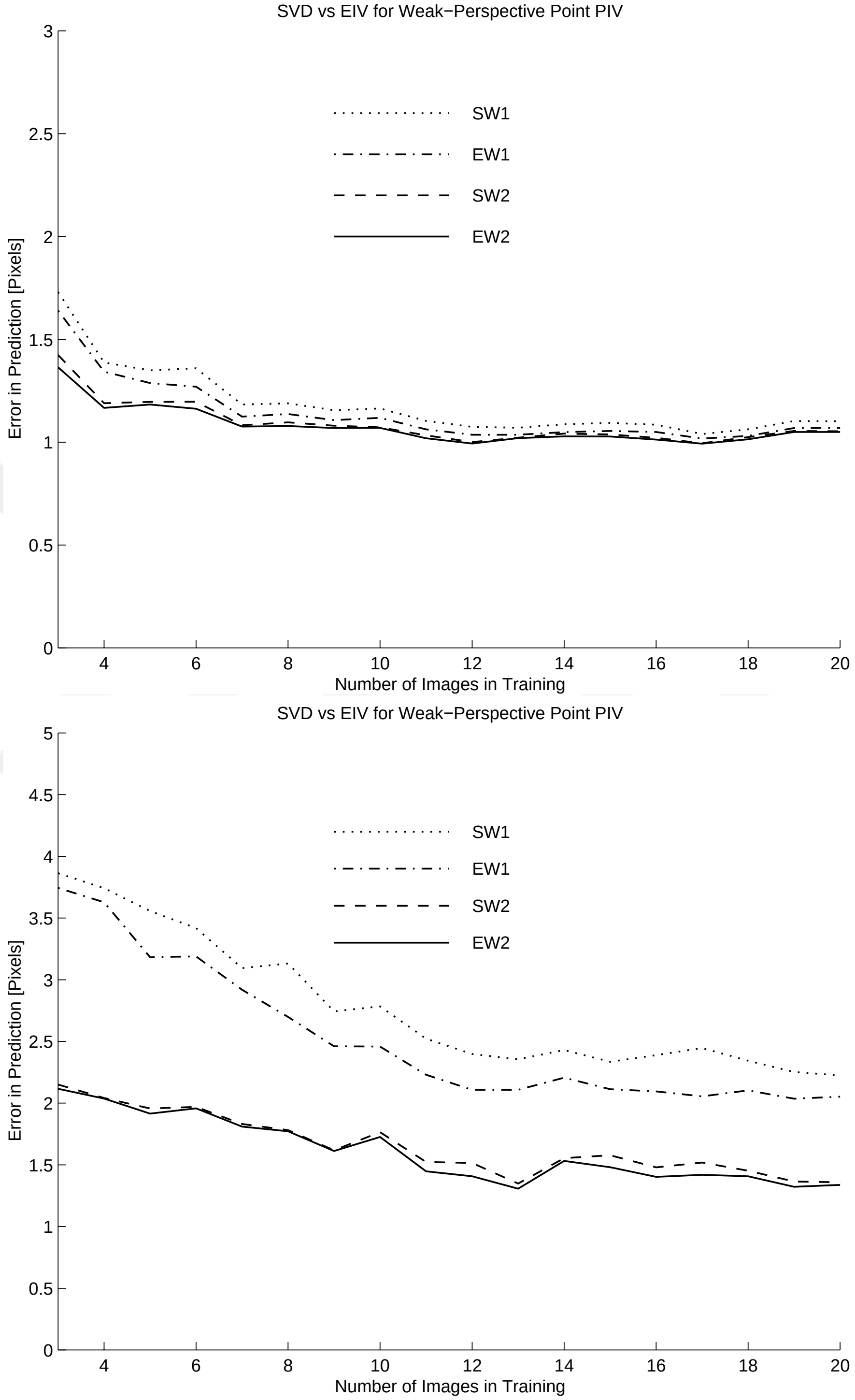


Figure 6.3 Performance of the weak perspective SVD- and EIV-based methods with respect to the number of images in training: all of the testing (50) and the training images have been perturbed with a Gaussian noise with $\sigma^2 = 0.5$ pixels.

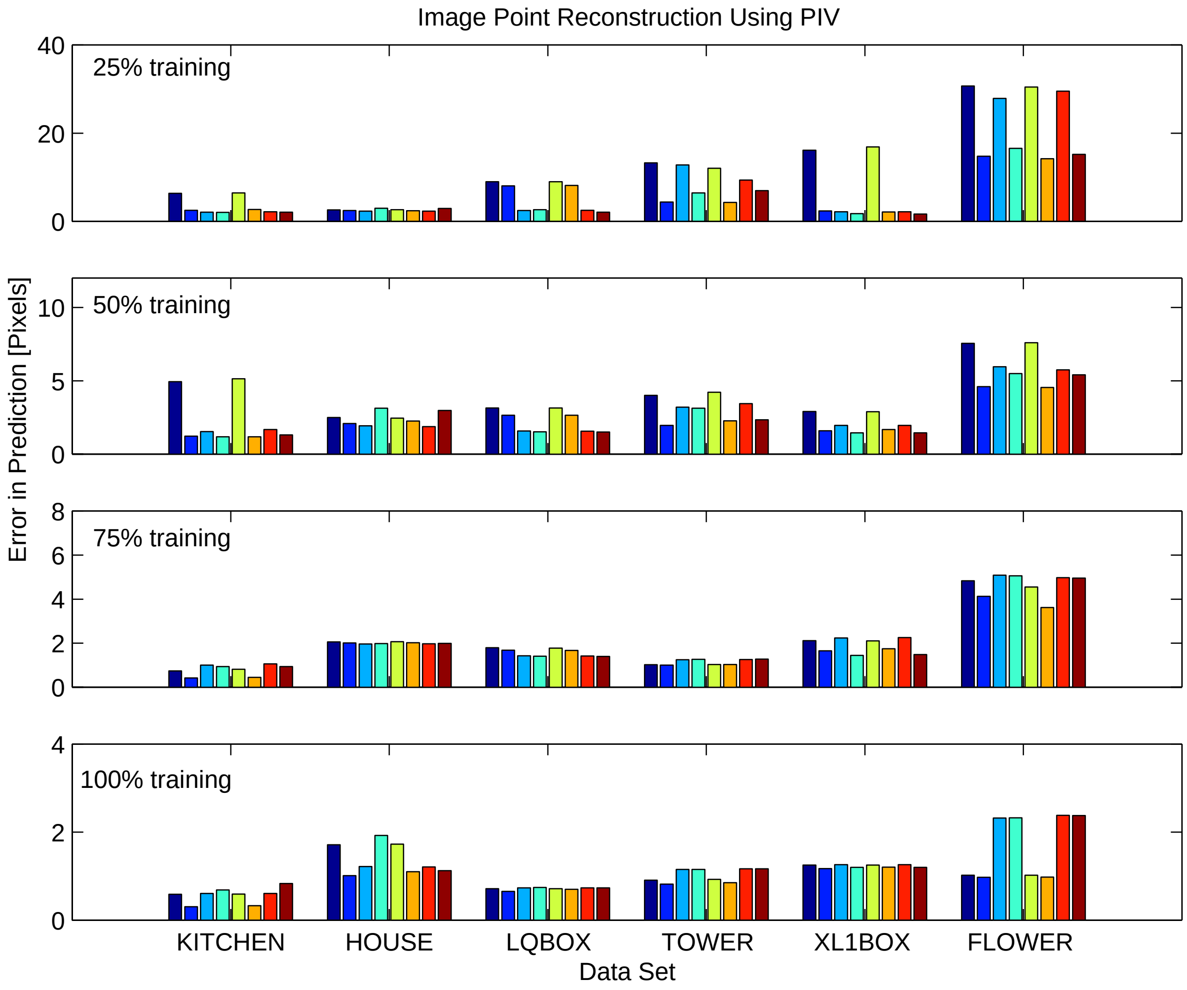


Figure 6.4 Quantitative comparisons of the SVD- and EIV-based methods on real data sets. All variants of the algorithm are tested (from left to right, SW1, EW1, SW2, EW2, SP1, EP1, SP2 and EP2).

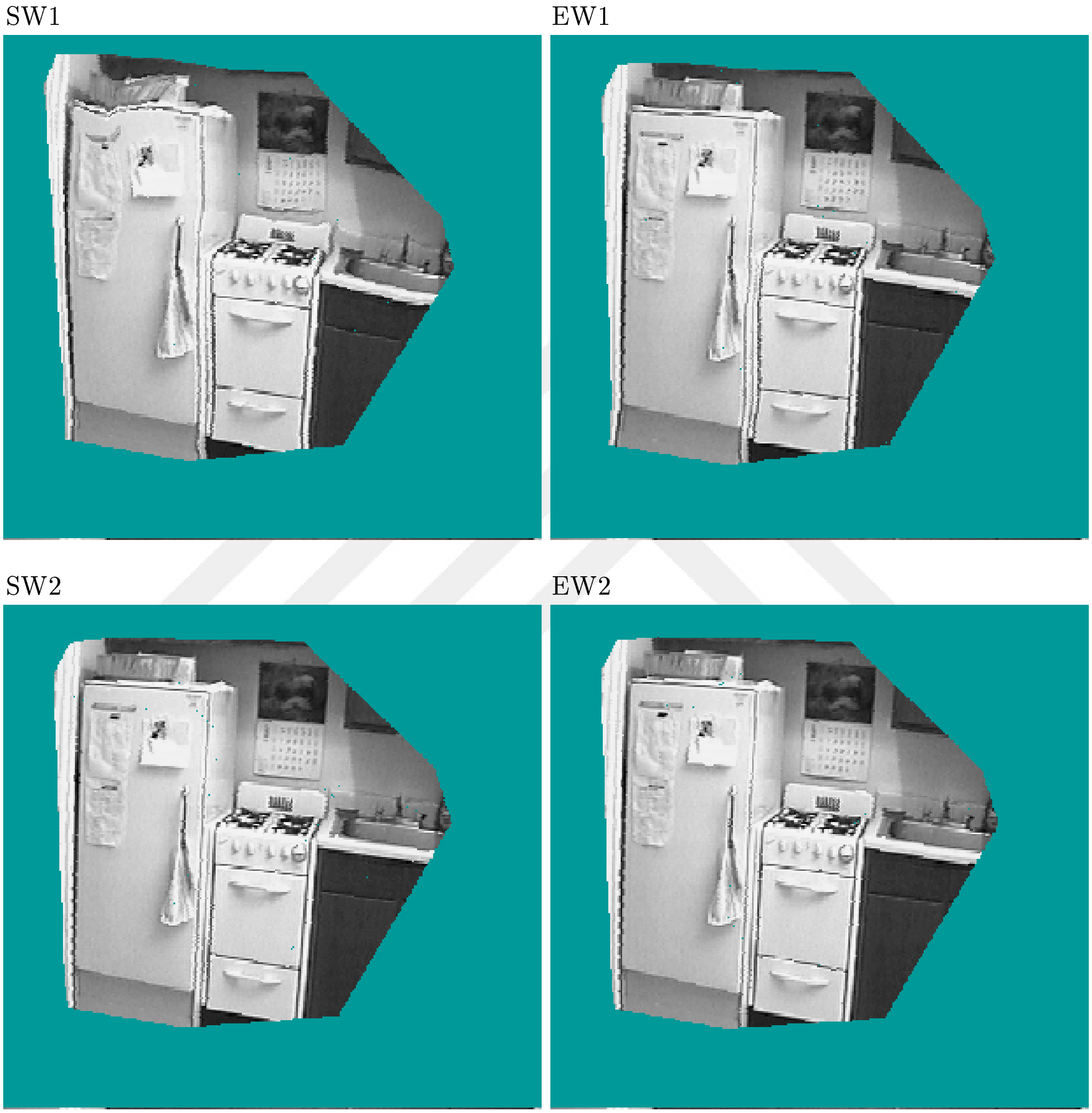
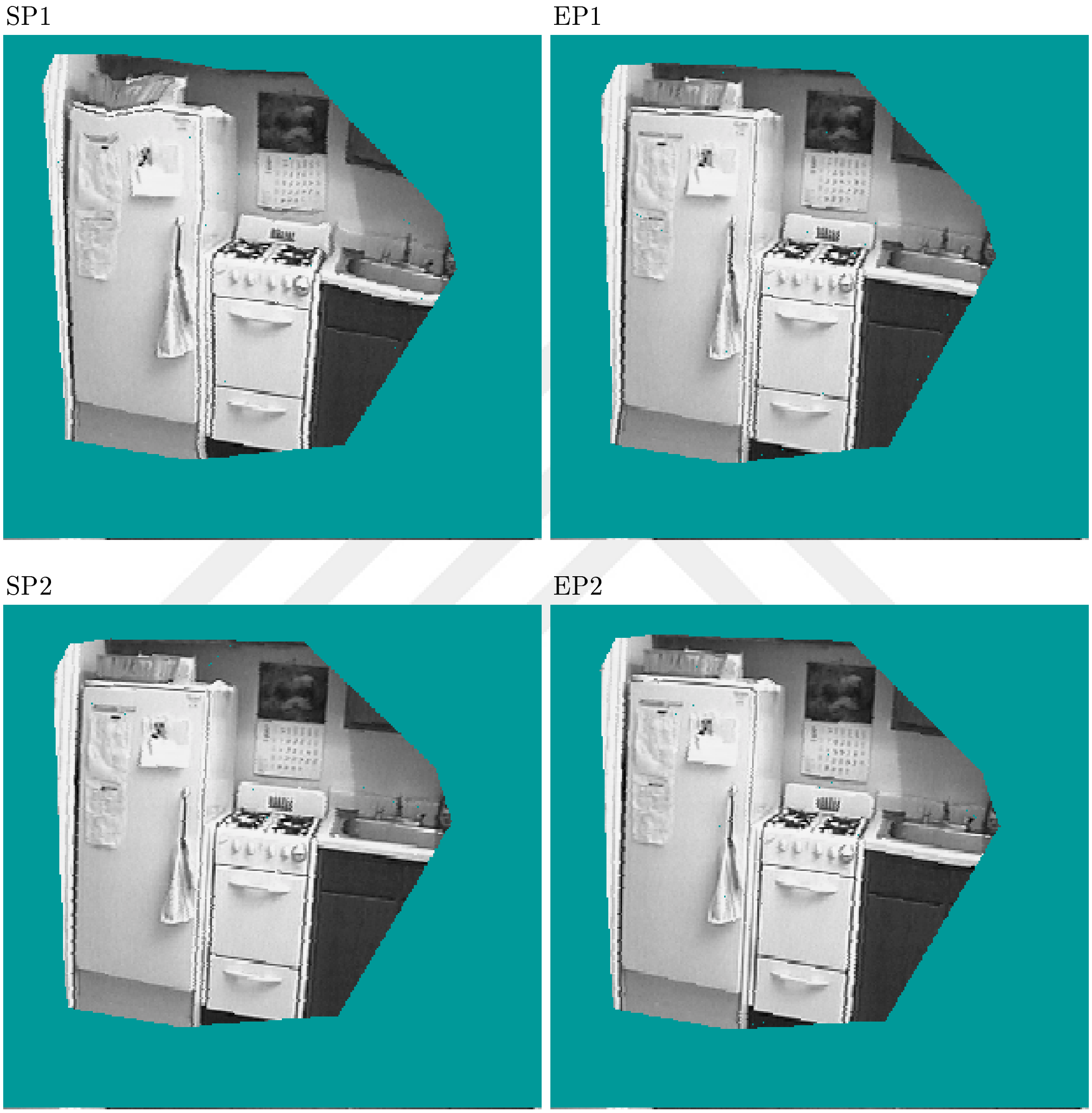


Figure 6.5 Qualitative comparison of the weak-perspective SVD- and EIV-based methods on the KITCHEN data set using the first half of the images in training. The synthesized images correspond to the last image in the data.



SP1

EP1

SP2

EP2

Figure 6.6 Qualitative comparison of the paraperspective SVD- and EIV-based methods on the KITCHEN data set using the first half of the images in training. The synthesized images correspond to the last image in the data.

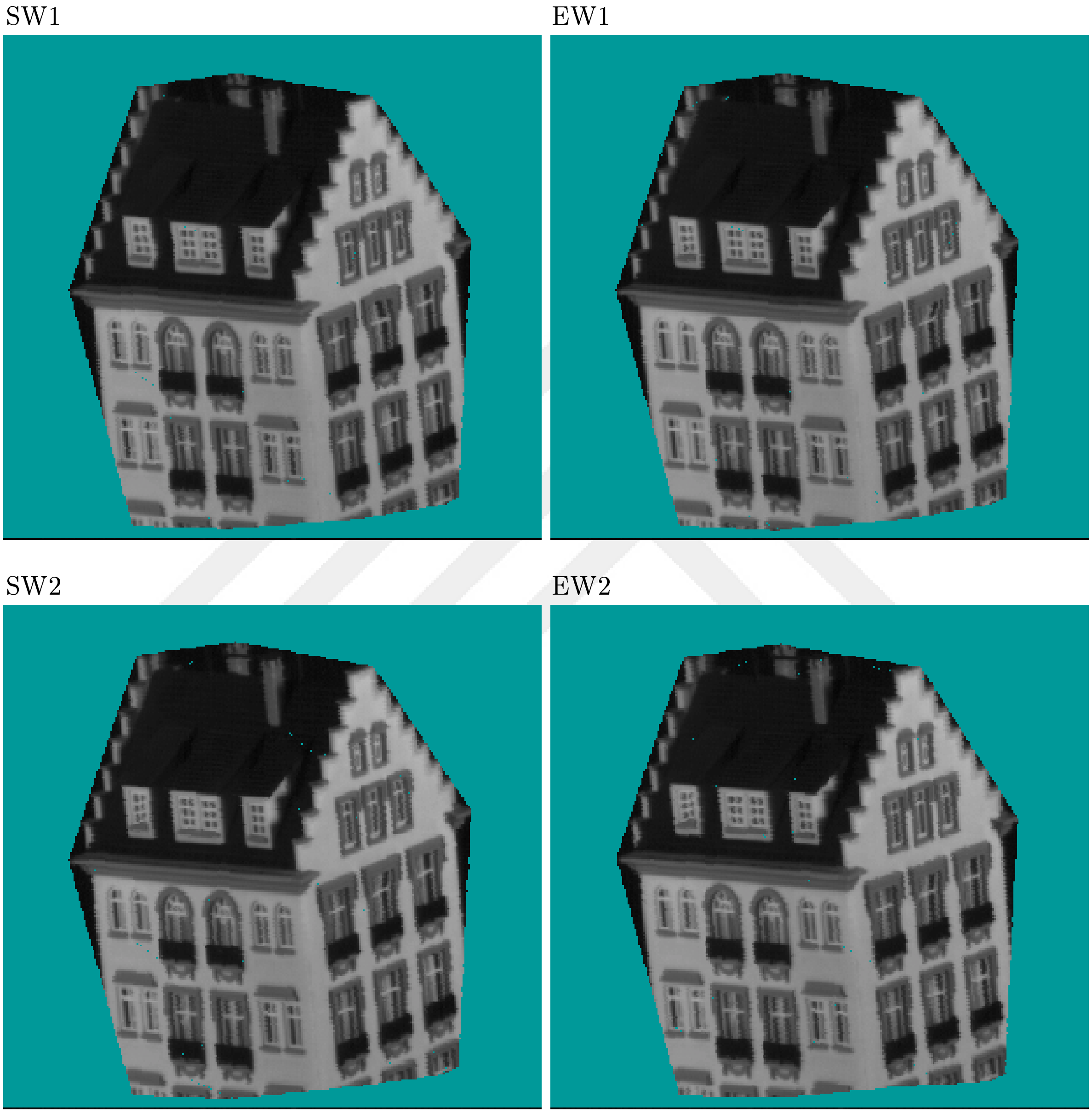


Figure 6.7 Qualitative comparison of the weak-perspective SVD- and EIV-based methods on the HOUSE data set using the first half of the images in training. The synthesized images correspond to the last image in the data.

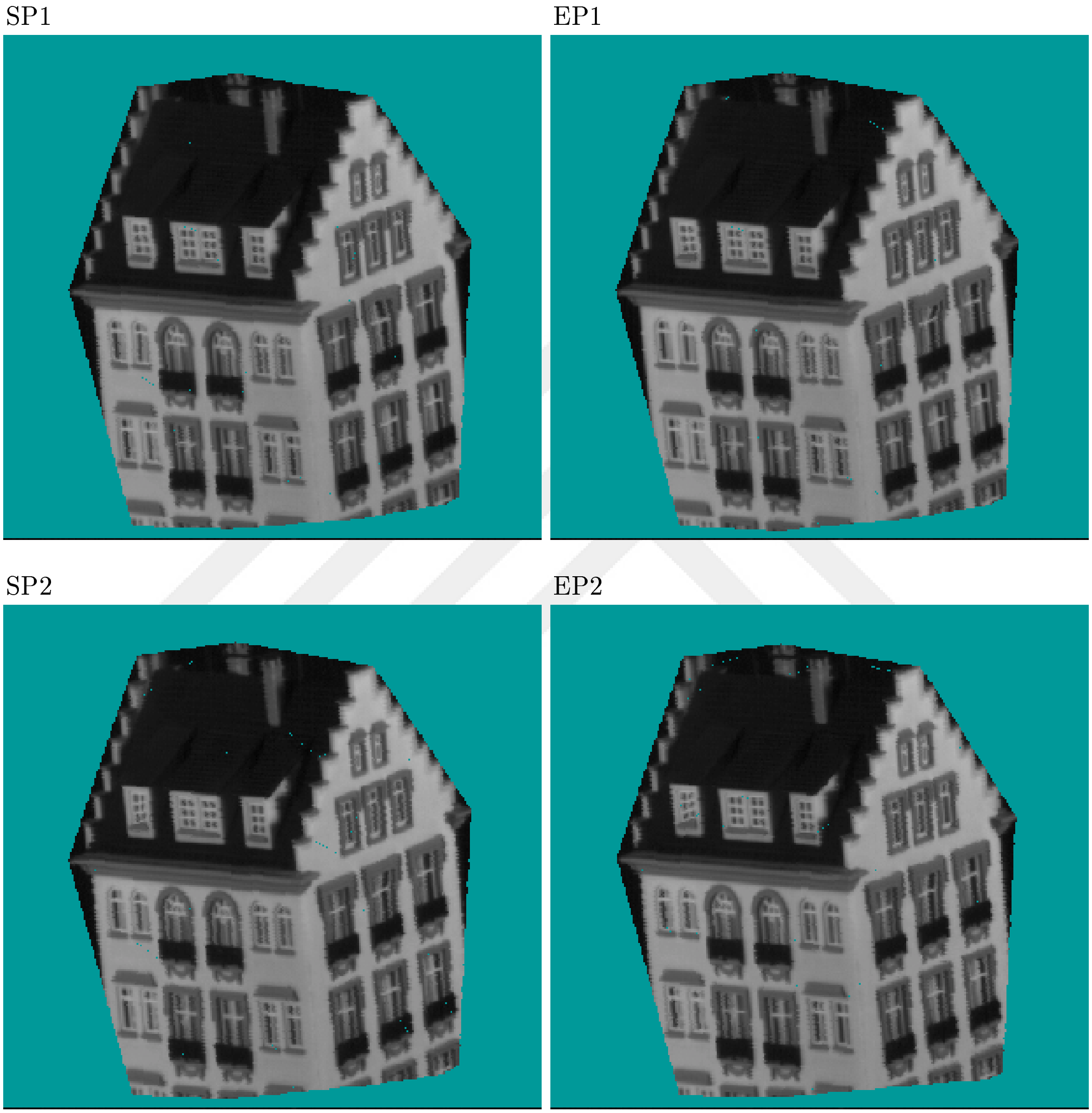


Figure 6.8 Qualitative comparison of the paraperspective SVD- and EIV-based methods on the HOUSE data set using the first half of the images in training. The synthesized images correspond to the last image in the data.

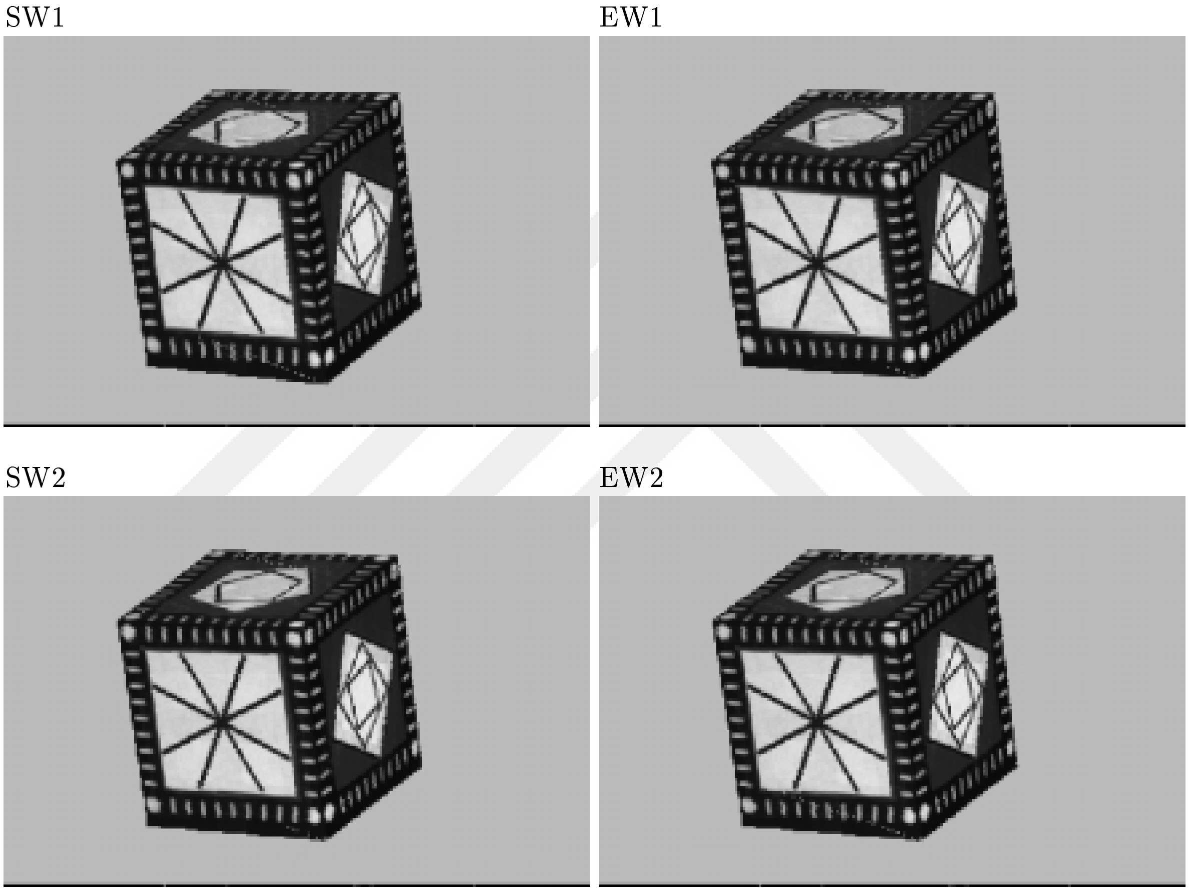


Figure 6.9 Qualitative comparison of the weak-perspective SVD- and EIV-based methods on the LQBOX data set using the first half of the images in training. The synthesized images correspond to the last image in the data.

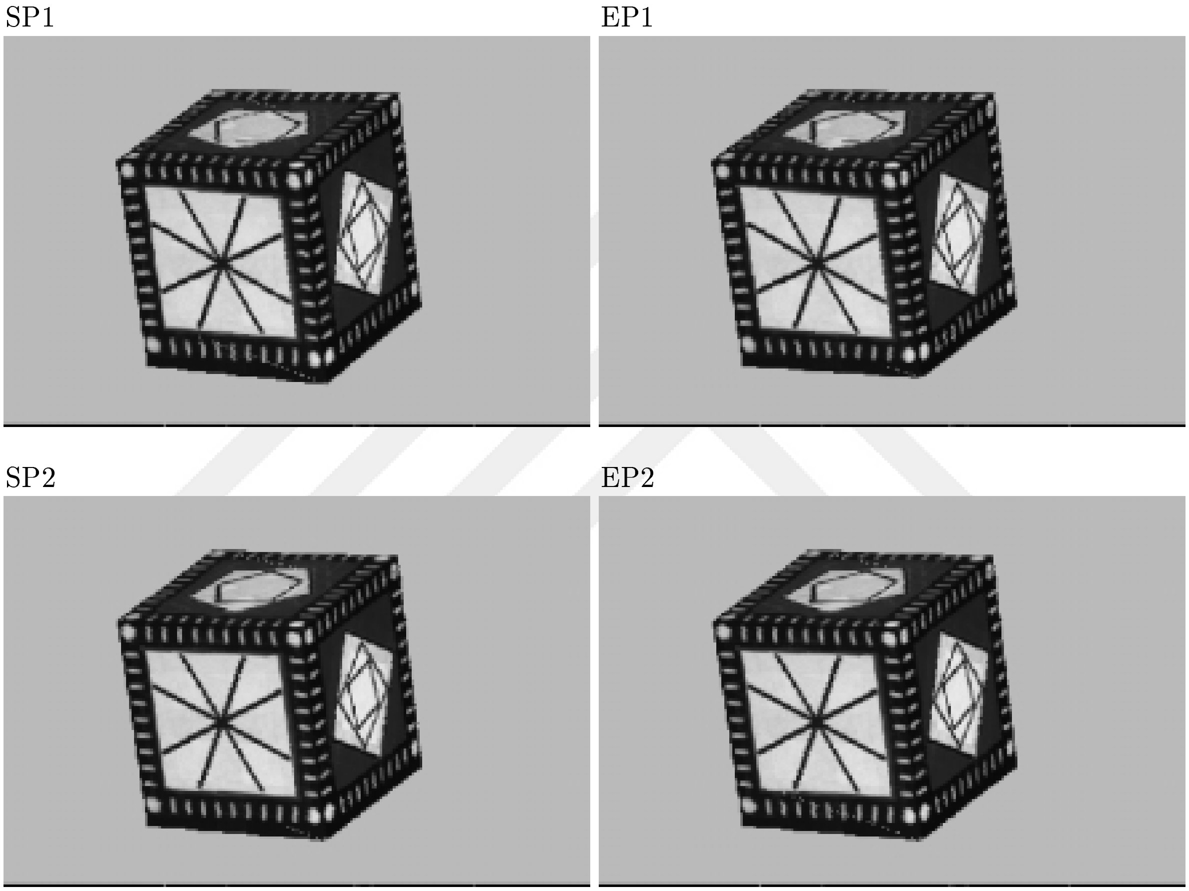


Figure 6.10 Qualitative comparison of the paraperspective SVD- and EIV-based methods on the LQBOX data set using the first half of the images in training. The synthesized images correspond to the last image in the data.

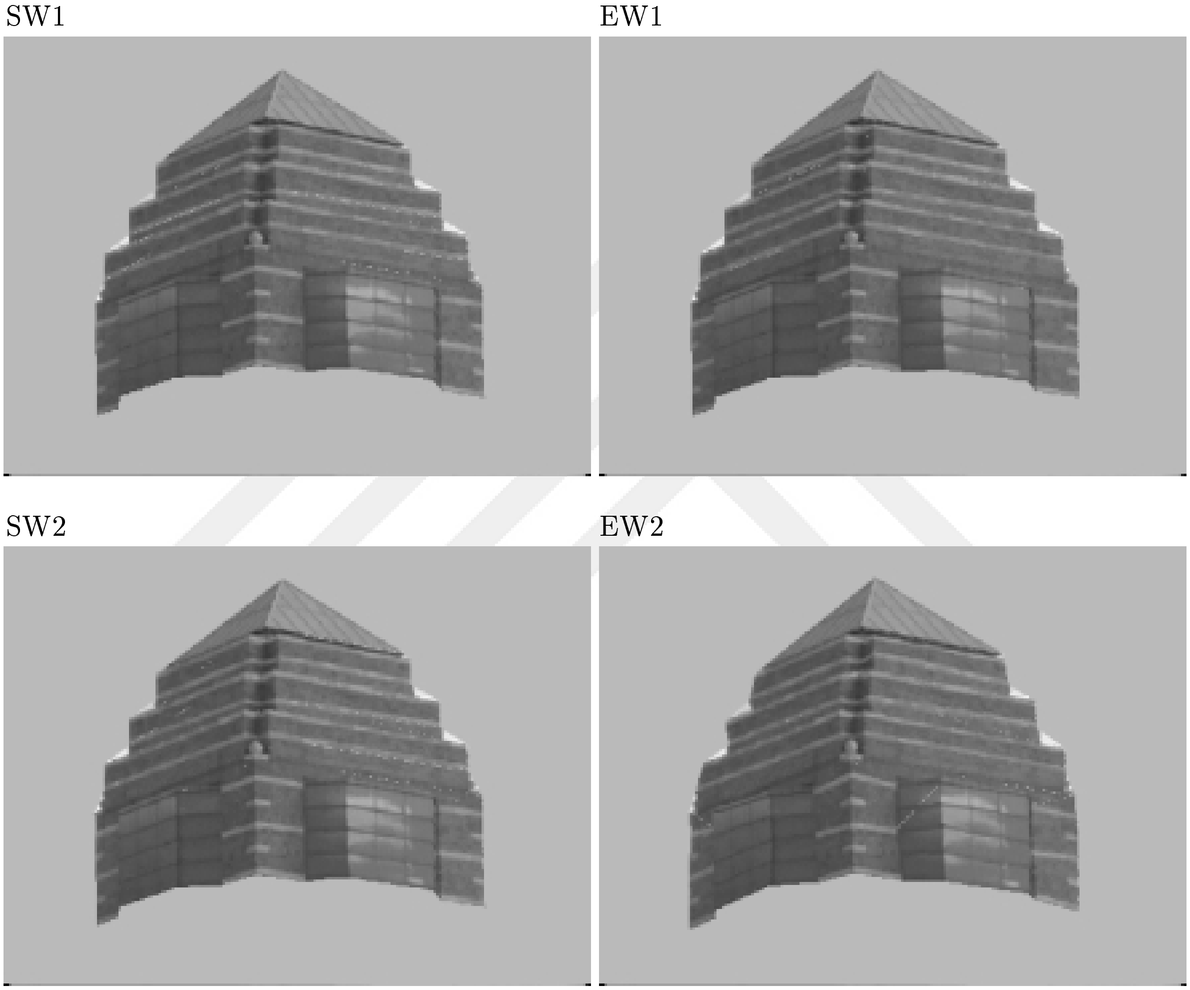


Figure 6.11 Qualitative comparison of the weak-perspective SVD- and EIV-based methods on the TOWER data set using the first half of the images in training. The synthesized images correspond to the last image in the data.

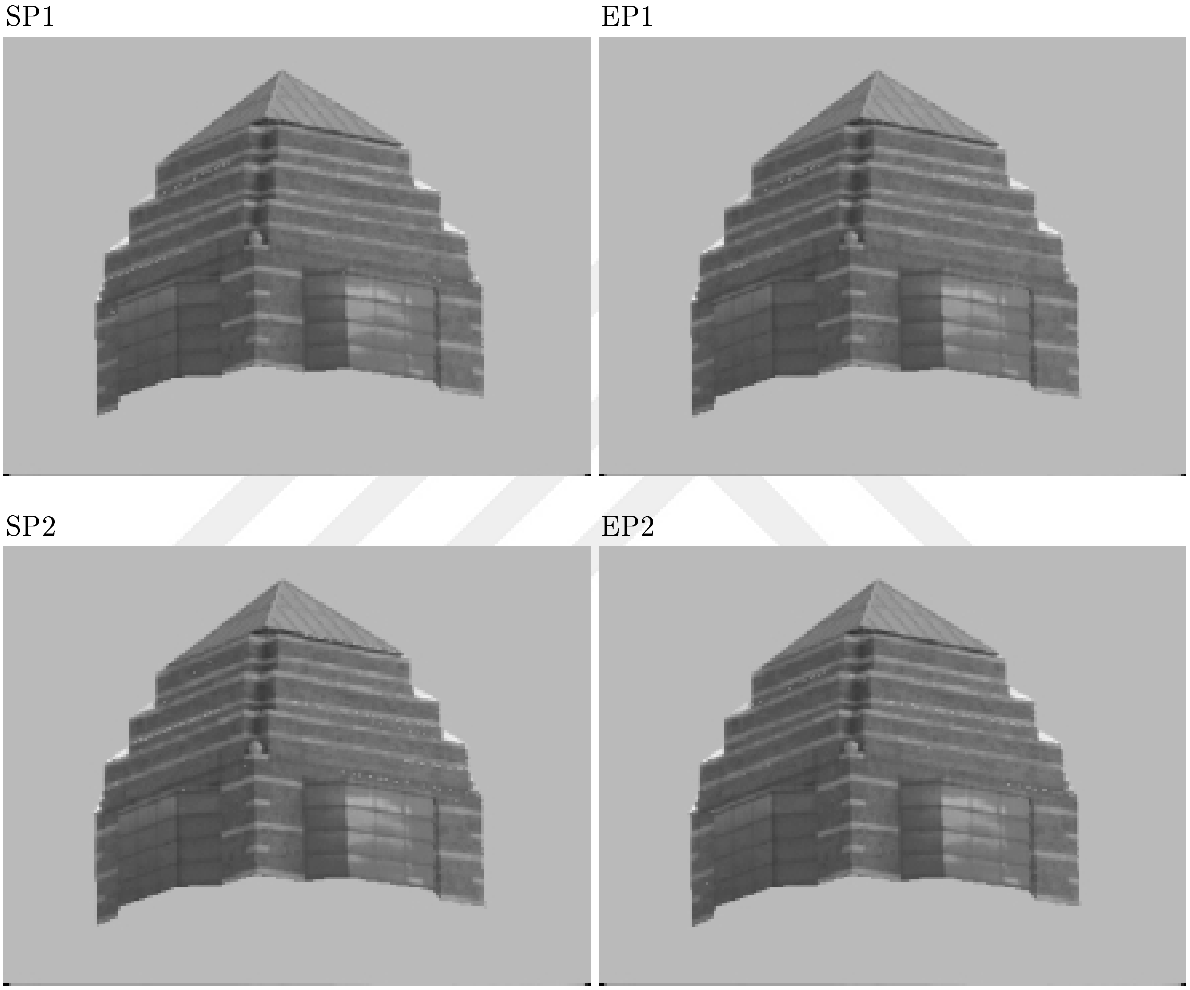


Figure 6.12 Qualitative comparison of the paraperspective SVD- and EIV-based methods on the TOWER data set using the first half of the images in training. The synthesized images correspond to the last image in the data.

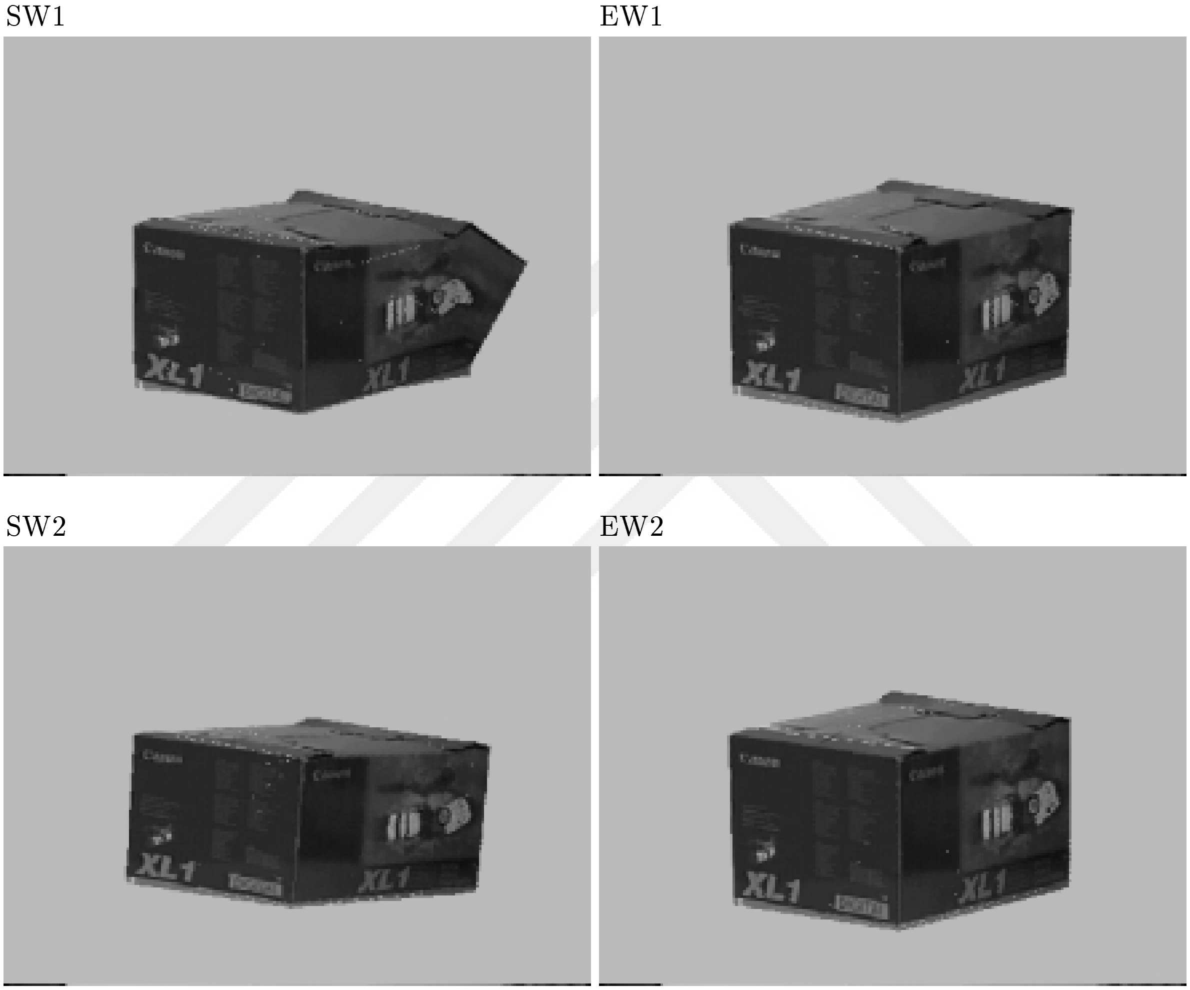
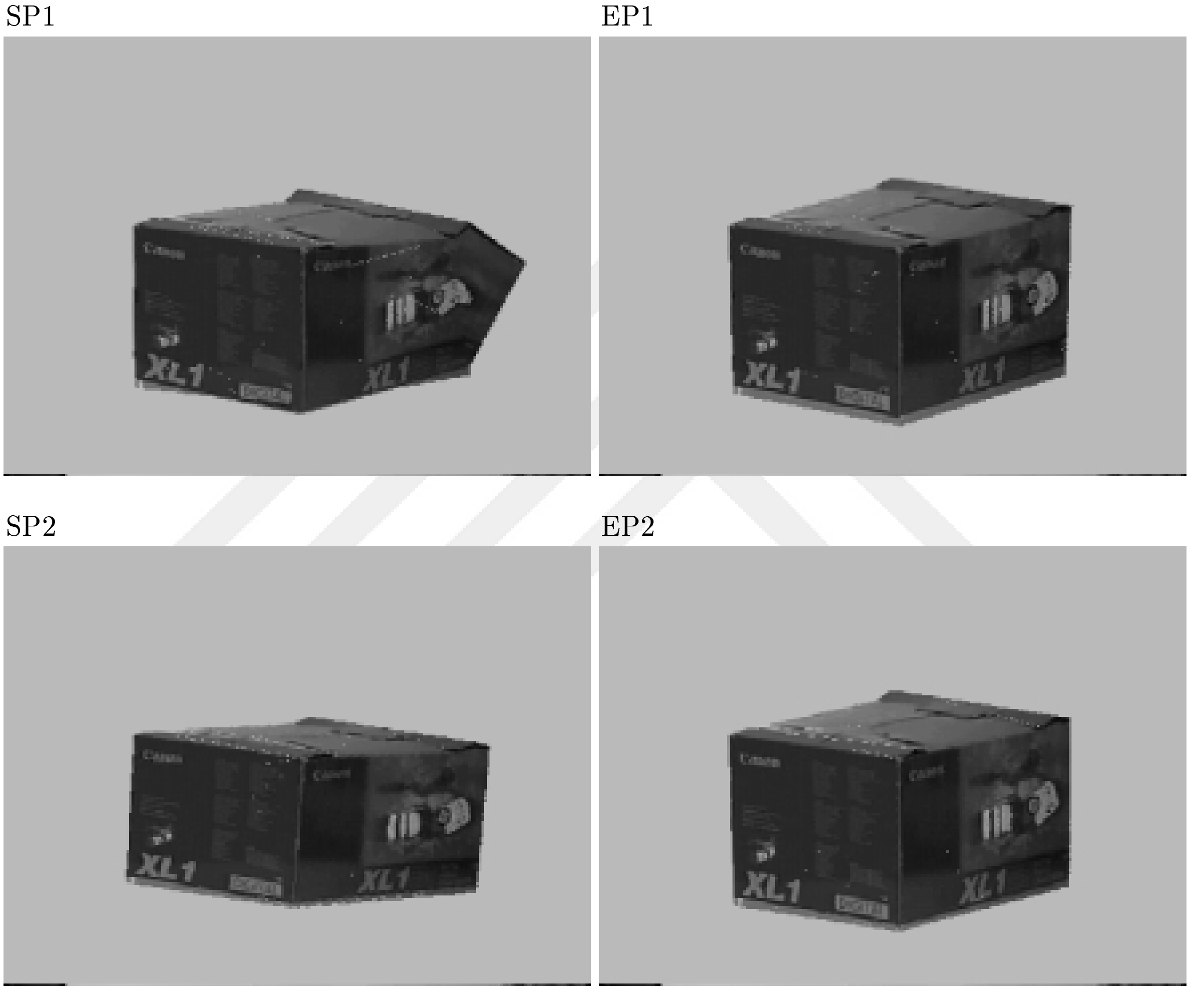


Figure 6.13 Qualitative comparison of the weak-perspective SVD- and EIV-based methods on the XL1BOX data set using the first half of the images in training. The synthesized images correspond to the last image in the data.



SP1

EP1

SP2

EP2

Figure 6.14 Qualitative comparison of the paraperspective SVD- and EIV-based methods on the XL1BOX data set using the first half of the images in training. The synthesized images correspond to the last image in the data.

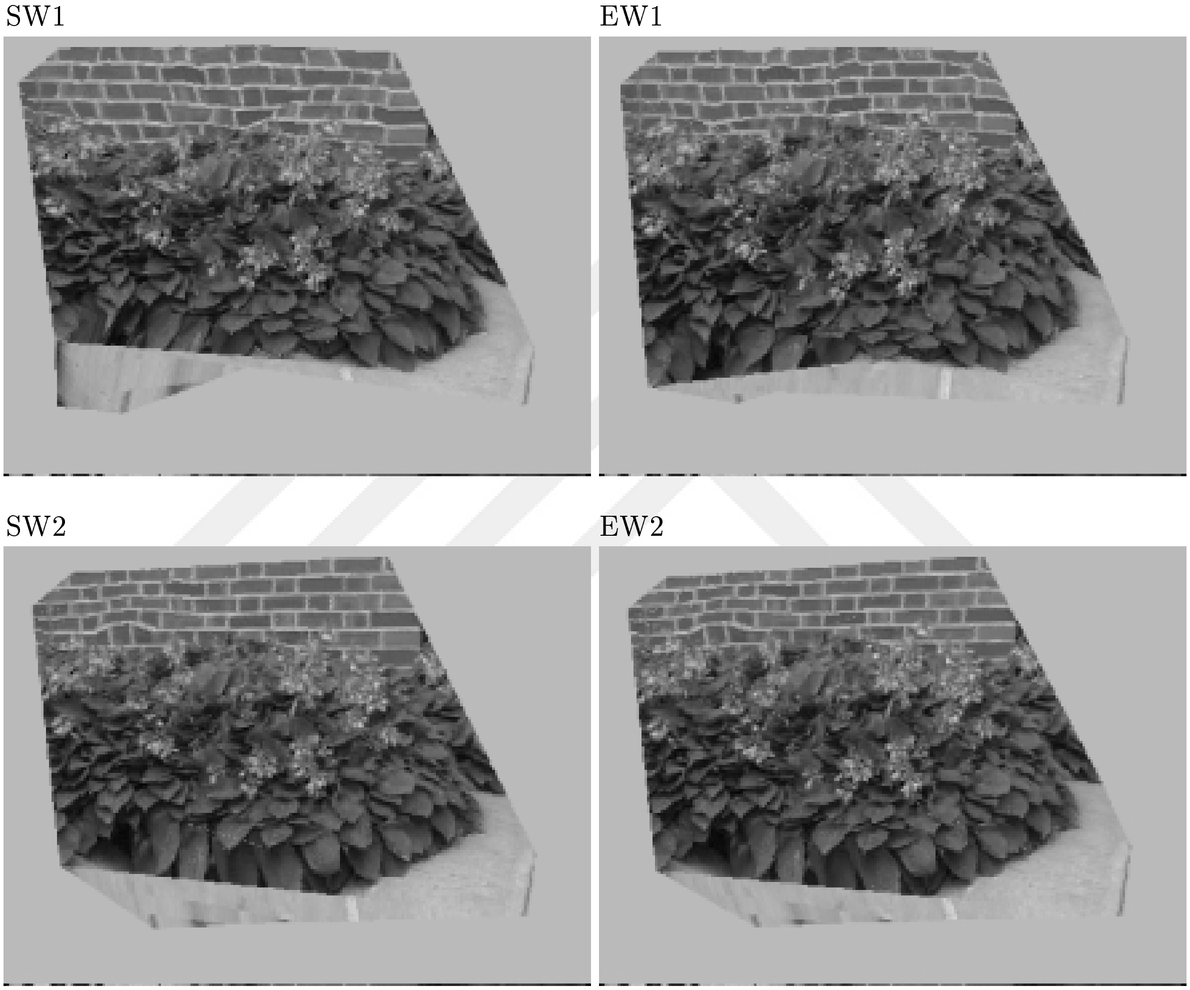


Figure 6.15 Qualitative comparison of the weak-perspective SVD- and EIV-based methods on the FLOWER data set using the first half of the images in training. The synthesized images correspond to the last image in the data.

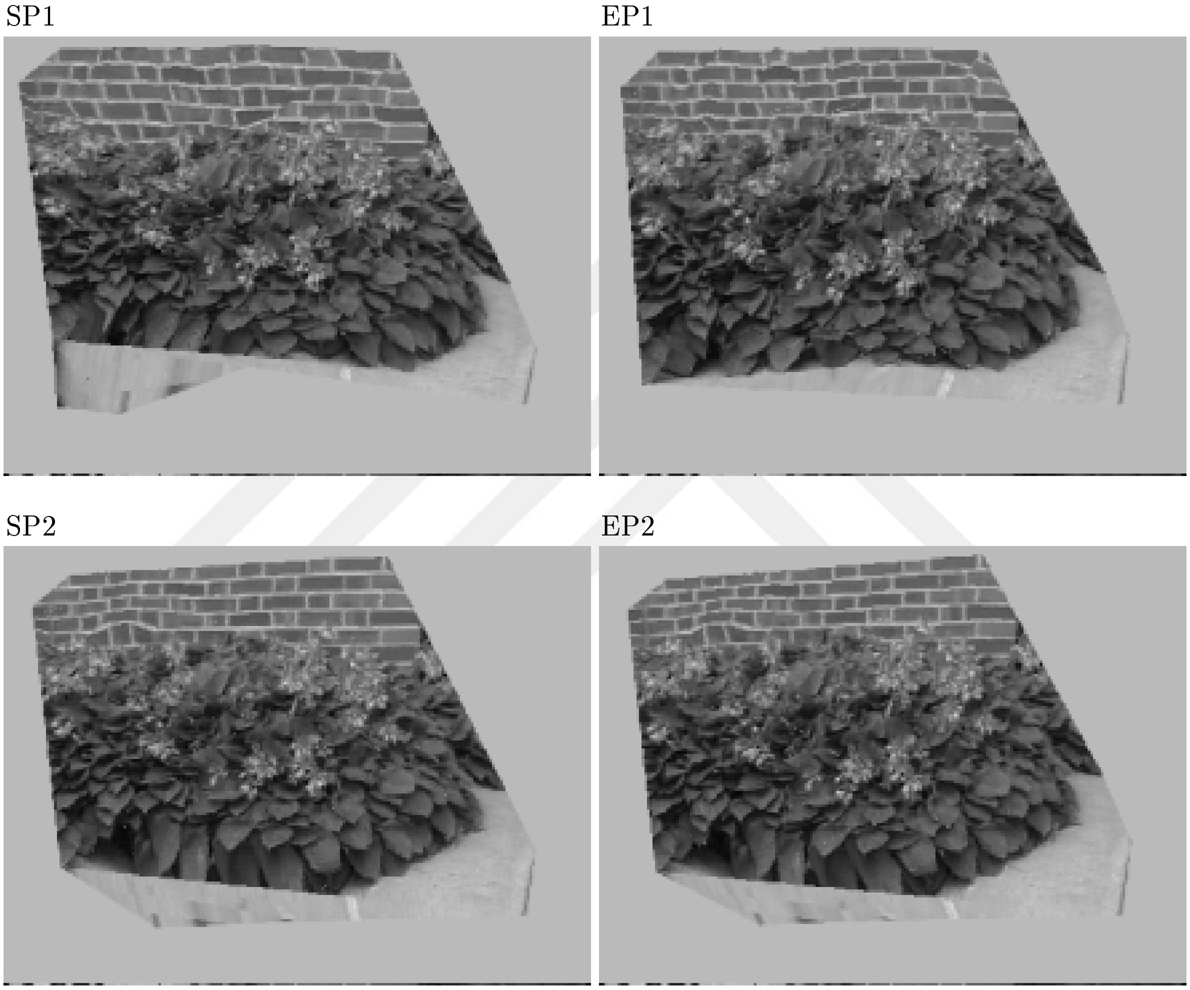


Figure 6.16 Qualitative comparison of the paraperspective SVD- and EIV-based methods on the FLOWER data set using the first half of the images in training. The synthesized images correspond to the last image in the data.

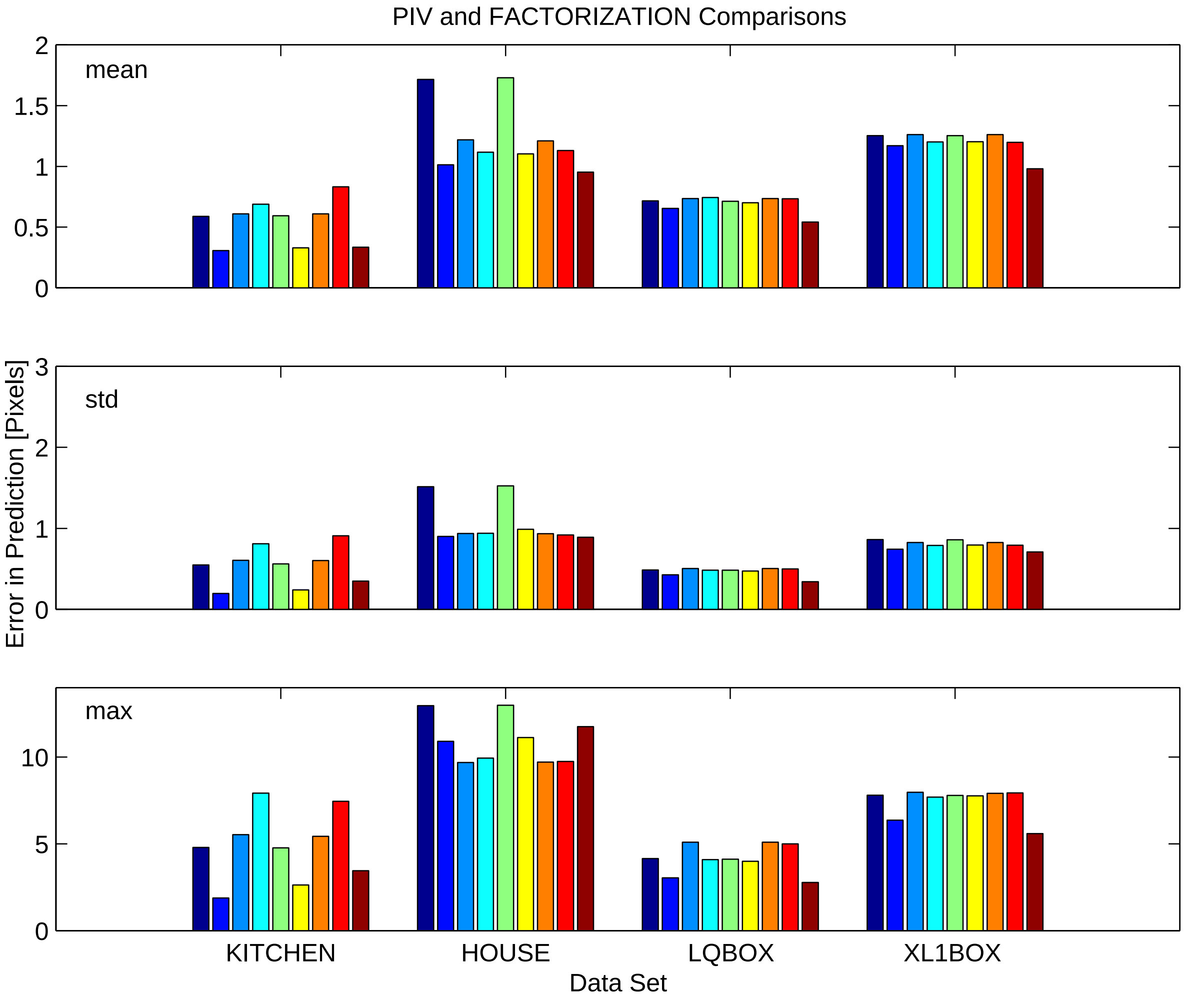
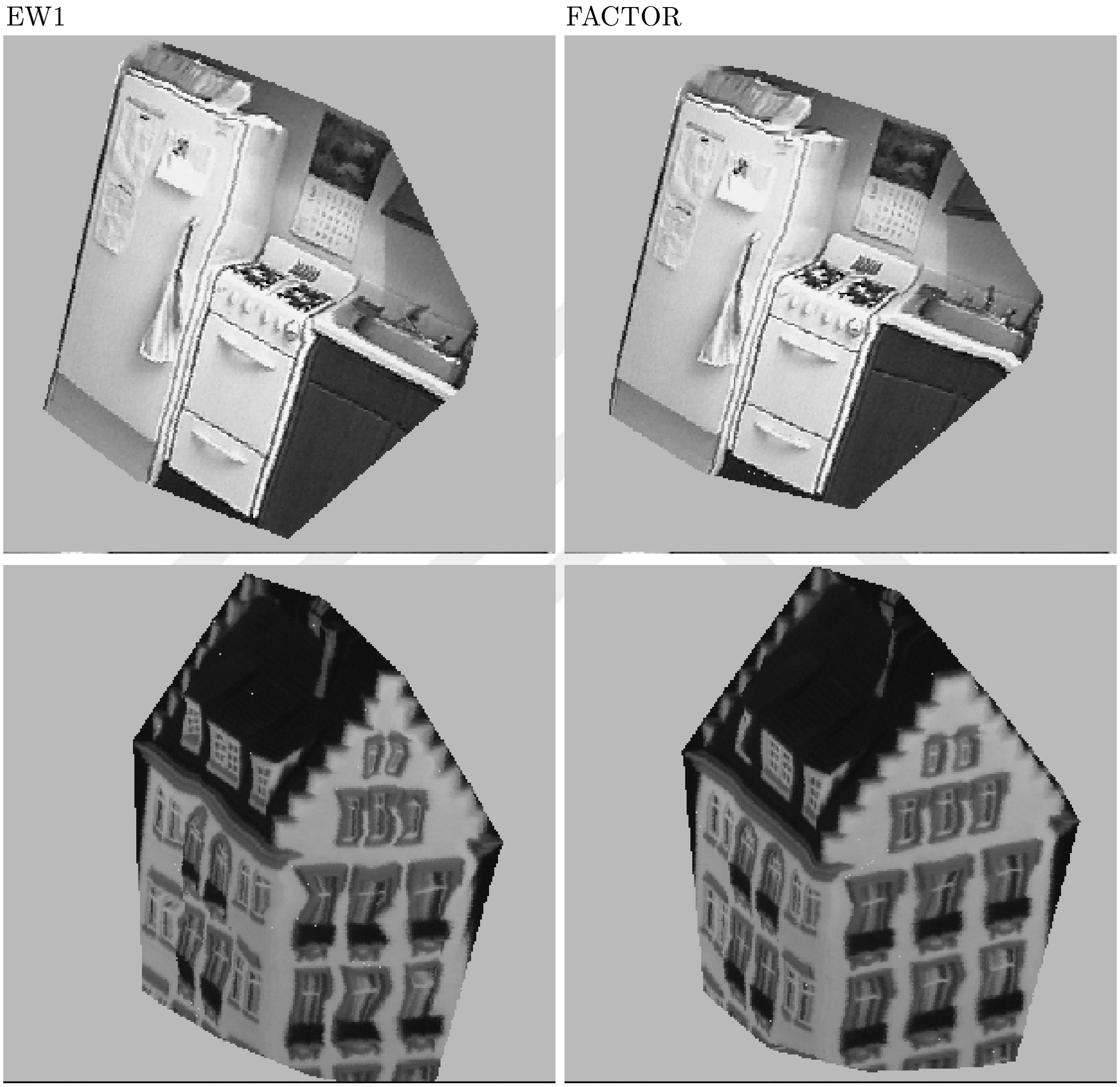


Figure 6.17 Comparisons of the variants of the PIV algorithm (from left to right, SW1, EW1, SW2, EW2, SP1, EP1, SP2 and EP2) and the factorization method of Tomasi and Kanade.



EW1

FACTOR

Figure 6.18 Synthesized images for novel views using the EW1 and the factorization algorithms for the HOUSE and KITCHEN data sets.

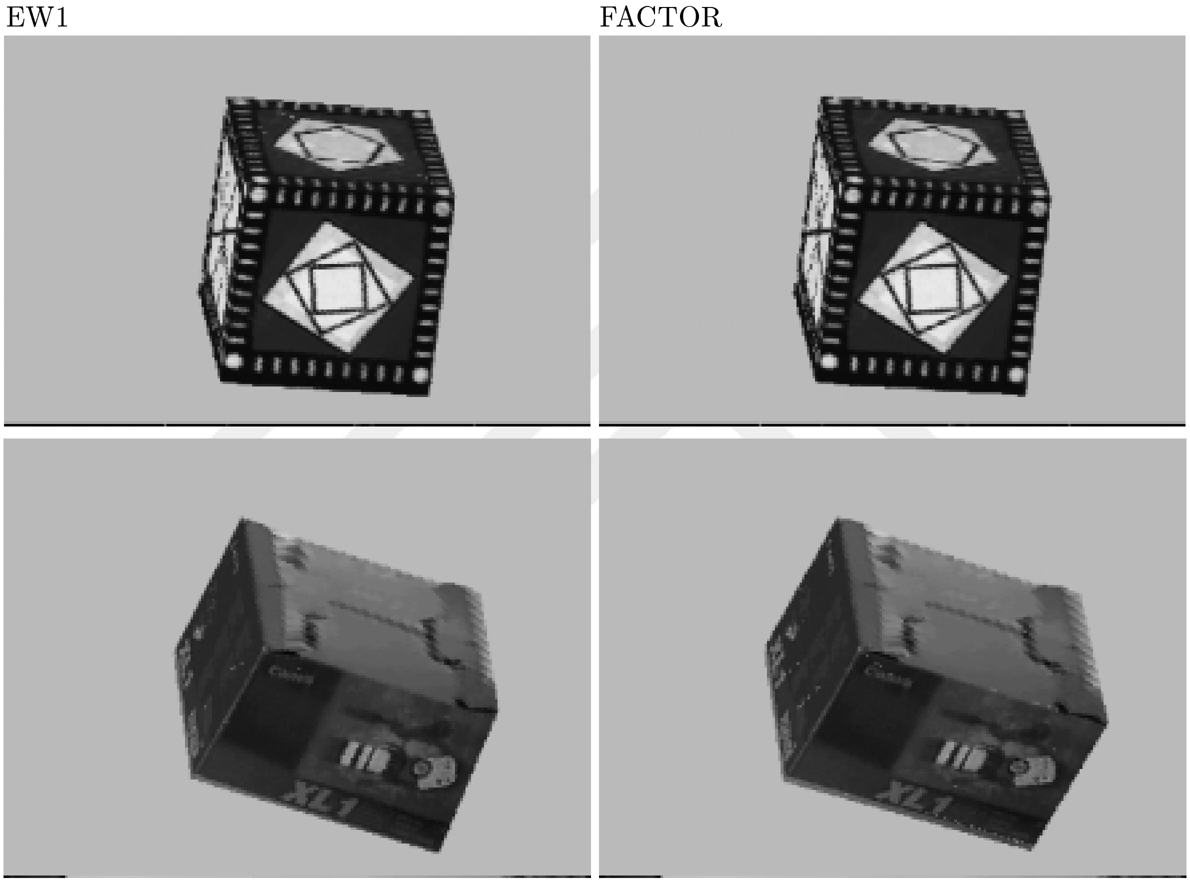


Figure 6.19 Synthesized images for novel views using the EW1 and the factorization algorithms for the LQBOX and XL1BOX data sets.

CHAPTER 7

CONCLUSIONS

7.1 Summary of Contributions

We have addressed in this thesis two computer vision problems related to the analysis and synthesis of image sequences, namely, the weak calibration of a stereo rig from point correspondences, and image-based rendering without explicit three-dimensional reconstruction. We have introduced novel algorithms for both problems.

Chapters 2 and 3 address the weak calibration problem. We have introduced a linear weak calibration algorithm by recasting the epipolar constraints in a projective setting with an appropriate basis choice. We have shown that Jepson's and Heeger's linear subspace algorithm for infinitesimal motion estimation can be generalized to finite motion case. With experiments on both real and synthetic images, and comparisons to other linear and non-linear weak calibration algorithms, we have demonstrated that our approach performs quite well. The main contributions of this part of the thesis can be summarized as follows:

- We have clarified the relationship between the linear subspace method of Jepson and Heeger for infinitesimal motion estimation and the Longuet-Higgins' characterization of epipolar geometry.

- We have derived a new formulation for the epipolar geometry in a projective setting.

- We have introduced a linear algorithm for estimating epipolar geometry based on this formulation.

Chapters 4, 5 and 6 address the image-based rendering problem. We have introduced a new algorithm that uses point and line features together and respects the Euclidean constraints associated with real cameras. After showing that the set of all images of a rigid scene taken by a Euclidean camera is a six-dimensional variety, we have described a method to parameterize this variety in terms of images of three reference points. We have used this parameterization to synthesize new images from point and line correspondences without actual three-dimensional reconstruction. We have verified our approach with extensive experiments. Furthermore, we have applied recent advances in statistically-unbiased least-squares methods to our image-based rendering approach resulting in significant improvements. In summary, our contributions include:

- We have introduced a parameterization of the set of all images of a scene (points and lines) based on the images of three reference points for weak-perspective and paraperspective projections.
- We have successfully used this parameterization to synthesize new images using point and line features in a general framework and presented a method to handle occlusions without actually recovering the depth.
- We have analyzed the degenerate scene and camera configurations for which our parameterization may fail.
- We have successfully adapted recent advances in statistically-unbiased least-squares methods of [59] to our algorithm and obtained significant performance improvements over the linear least-squares solution.

7.2 Future Research Directions

Here we sketch some future research directions.

In the weak calibration case:

- **Choice of basis points:** A key part of our approach is the choice of an appropriate basis for projective space, which assumes that four reference points can be measured accurately. An interesting question is whether it is possible to design a different linear subspace algorithm that takes every image measurement equally into account, hopefully averaging out individual errors.

- **Calibrated cameras:** Another interesting question is whether the linear subspace approach can be generalized to handle calibrated cameras directly and estimate the full motion in a linear manner.

In the image-based rendering case:

- **Applications of PIVs:** Let us first note that parameterized image varieties are potentially useful in motion segmentation and object recognition tasks: although the PIV is not a vector space but a variety, its parameters can be estimated through linear least squares. In particular, a point will be rigidly attached to the Euclidean frame defined by the three reference points of the PIV if and only if the space spanned by six polynomials in the coordinates of the projections of the four points is five-dimensional. Thus singular value decomposition can in principle be used to test this rank condition and perform motion segmentation of video sequences. It is also possible to use the PIV of an object as its model in object recognition tasks, replacing the costly manual construction of CAD models by the automatic learning of the PIV from an image sequence. In this scenario, point matches can be predicted and verified from the PIV as a simple variant of the image synthesis approach discussed earlier. This technique is similar to Basri's and Ullman's method for recognition by linear combination of model images [112], but it incorporates Euclidean constraints in the matching process.

- **Adaptive Triangulation:** A very interesting problem is the construction of better meshes from image sequences. This is a difficult issue for any image-based rendering technique that does not attempt to estimate the camera motion or the actual scene structure, and it is also a very important one in practice since rendering a scene from truly arbitrary viewpoints requires constructing a mesh covering its whole surface.

APPENDIX A

Elementary Notions of Analytical Projective Geometry

In this section we introduce some basic notions from analytical projective geometry. In particular, we define projective bases and coordinates in the real projective plane $\mathbb{P}^2$ and 3D space $\mathbb{P}^3$ (we largely follow [106]).

A projective basis for a plane is defined by a triangle of reference and a coplanar unit point, and a projective basis of 3D space is defined by a tetrahedron of reference and a unit point.

The non-homogeneous projective coordinates of a point can be defined geometrically in terms of *cross-ratios*: the cross-ratio of four collinear points $\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}, \boldsymbol{D}$ is

$$[\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}, \boldsymbol{D}] = \frac{\overline{\boldsymbol{CA}}}{\overline{\boldsymbol{CB}}} \times \frac{\overline{\boldsymbol{DB}}}{\overline{\boldsymbol{DA}}},$$

the cross-ratio of four lines passing through some point $\boldsymbol{O}$ (a pencil of lines) is defined as the cross-ratio of the intersections of these lines with any other line l' not passing through $\boldsymbol{O}$, and the cross-ratio of four planes intersecting along some line l (a pencil of planes) is the cross-ratio of the pencil of lines formed by the intersection of these planes with any other plane Π' not containing l. These cross-ratios are independent of l' and Π'.

In the plane, the non-homogeneous projective coordinates (k_1, k_2) of the point $\boldsymbol{P}$ in the basis $(\boldsymbol{A}_0, \boldsymbol{A}_1, \boldsymbol{A}_2, \boldsymbol{A})$ are defined by

$$\begin{cases} k_1 = [\boldsymbol{A}_1\boldsymbol{A}_0, \boldsymbol{A}_1\boldsymbol{A}_2, \boldsymbol{A}_1\boldsymbol{A}, \boldsymbol{A}_1\boldsymbol{P}], \\ k_2 = [\boldsymbol{A}_0\boldsymbol{A}_1, \boldsymbol{A}_0\boldsymbol{A}_2, \boldsymbol{A}_0\boldsymbol{A}, \boldsymbol{A}_0\boldsymbol{P}], \end{cases}$$

where $\boldsymbol{MN}$ denotes the line joining the points $\boldsymbol{M}$ and $\boldsymbol{N}$, and $[l_1, l_2, l_3, l_4]$ denotes the cross-ratio of the pencil of lines l_1, l_2, l_3, l_4. With these conventions, the homogeneous coordinates of the basis points $\boldsymbol{A}_0$, $\boldsymbol{A}_1$, $\boldsymbol{A}_2$ and $\boldsymbol{A}$ are respectively $(1,0,0)^T$, $(0,1,0)^T$, $(0,0,1)^T$ and $(1,1,1)^T$. The homogeneous coordinates of $\boldsymbol{P}$ are $(k_1, k_2, 1)^T$.

Similarly, the non-homogeneous projective coordinates (k_1, k_2, k_3) of the point $\boldsymbol{P}$ in the basis $(\boldsymbol{A}_0, \boldsymbol{A}_1, \boldsymbol{A}_2, \boldsymbol{A}_3, \boldsymbol{A})$ are defined by

$$\begin{cases} k_1 = [\boldsymbol{A}_1\boldsymbol{A}_2\boldsymbol{A}_0, \boldsymbol{A}_1\boldsymbol{A}_2\boldsymbol{A}_3, \boldsymbol{A}_1\boldsymbol{A}_2\boldsymbol{A}, \boldsymbol{A}_1\boldsymbol{A}_2\boldsymbol{P}]\,, \\ k_2 = [\boldsymbol{A}_2\boldsymbol{A}_0\boldsymbol{A}_1, \boldsymbol{A}_2\boldsymbol{A}_0\boldsymbol{A}_3, \boldsymbol{A}_2\boldsymbol{A}_0\boldsymbol{A}, \boldsymbol{A}_2\boldsymbol{A}_0\boldsymbol{P}]\,, \\ k_3 = [\boldsymbol{A}_0\boldsymbol{A}_1\boldsymbol{A}_2, \boldsymbol{A}_0\boldsymbol{A}_1\boldsymbol{A}_3, \boldsymbol{A}_0\boldsymbol{A}_1\boldsymbol{A}, \boldsymbol{A}_0\boldsymbol{A}_1\boldsymbol{P}]\,, \end{cases} \tag{A.1}$$

and $\boldsymbol{LMN}$ denotes the plane spanned by the three points $\boldsymbol{L}$, $\boldsymbol{M}$ and $\boldsymbol{N}$, and $[\Pi_1, \Pi_2, \Pi_3, \Pi_4]$ denotes the cross-ratio of the pencil of planes Π_1, Π_2, Π_3, Π_4. With these conventions, the homogeneous coordinates of the basis points $\boldsymbol{A}_0$, $\boldsymbol{A}_1$, $\boldsymbol{A}_2$, $\boldsymbol{A}_3$ and $\boldsymbol{A}$ are respectively $(1, 0, 0, 0)^T$, $(0, 1, 0, 0)^T$, $(0, 0, 1, 0)^T$, $(0, 0, 0, 1)^T$ and $(1, 1, 1, 1)^T$. The homogeneous coordinates of $\boldsymbol{P}$ are $(k_1, k_2, k_3, 1)^T$.

APPENDIX B

$\boldsymbol{\tau}(\boldsymbol{\xi}_{\boldsymbol{\chi}})$ and $\boldsymbol{\chi}(\boldsymbol{\xi}_{\boldsymbol{\tau}})$ are Orthogonal to $(1,1,1)^T$

The vectors $\boldsymbol{\tau}(\boldsymbol{\xi})$ and $\boldsymbol{\chi}(\boldsymbol{\xi})$ are defined by

$$\begin{cases} \boldsymbol{\tau}(\boldsymbol{\xi}) = \sum_{i=1}^{n} \xi_i(\boldsymbol{p}_i \times \boldsymbol{p}_i') = \sum_{i=1}^{n} \xi_i(v_i - v_i', u_i' - u_i, u_i v_i' - u_i' v_i)^T, \\ \boldsymbol{\chi}(\boldsymbol{\xi}) = \sum_{i=1}^{n} \xi_i \boldsymbol{q}_i = \sum_{i=1}^{n} \xi_i(v_i - v_i u_i', u_i v_i' - u_i, u_i' - v_i')^T. \end{cases}$$

Note that, *for any value* $\boldsymbol{\xi}_{\boldsymbol{\chi}}$ *such that* $\boldsymbol{\chi}(\boldsymbol{\xi}_{\boldsymbol{\chi}}) = 0$, we have

$$\boldsymbol{\tau}(\boldsymbol{\xi}_{\boldsymbol{\chi}}) = \sum_{i=1}^{n} \xi_{\boldsymbol{\chi},i}(v_i - u_i', u_i' - u_i, u_i - v_i)^T.$$

In particular, the sum of the coordinates of the vector $\boldsymbol{\tau}(\boldsymbol{\xi}_{\boldsymbol{\chi}})$ is zero, which implies that it is orthogonal to the vector $(1,1,1)^T$. On the other hand, for *any value* $\boldsymbol{\xi}_{\boldsymbol{\tau}}$ *such that* $\boldsymbol{\tau}(\boldsymbol{\xi}_{\boldsymbol{\tau}}) = 0$, we have

$$\boldsymbol{\chi}(\boldsymbol{\xi}_{\boldsymbol{\tau}}) = \sum_{i=1}^{n} \xi_{\boldsymbol{\tau},i}(v_i - u_i v_i', u_i v_i' - u_i, u_i - v_i) = 0.$$

Thus the sum of the coordinates of the vector $\boldsymbol{\chi}(\boldsymbol{\xi}_{\boldsymbol{\tau}})$ is zero, which implies that this vector is also orthogonal to $(1,1,1)^T$.

APPENDIX C

Paraperspective Constraints

Let C denote the optical center of the camera and $(\boldsymbol{i}, \boldsymbol{j}, \boldsymbol{k})$ denote the orthonormal basis vectors attached to the camera. Let R denote a scene reference point, Δ denote the line joining C to R, and Π denote the fronto-parallel plane passing through R. Paraperspective projection of a scene point P operates in two steps [82]: parallel projection in the direction of Δ is first used to map P onto a point Q of the plane Π; perspective projection is then used to map the point Q onto the image point p.

Using the above notation, paraperspective projection can be modeled by

$$\boldsymbol{p} = \begin{pmatrix} \frac{1}{t_z}(\boldsymbol{i}^T - \frac{t_x}{t_z}\boldsymbol{k}^T) \\ \frac{1}{t_z}(\boldsymbol{j}^T - \frac{t_y}{t_z}\boldsymbol{k}^T) \end{pmatrix} (\boldsymbol{P} - \boldsymbol{R}) + \begin{pmatrix} \frac{t_x}{t_z} \\ \frac{t_y}{t_z} \end{pmatrix},$$

where (t_x, t_y, t_z) are the coordinates of the vector $\boldsymbol{R} - \boldsymbol{C}$ in the coordinate system $(\boldsymbol{i}, \boldsymbol{j}, \boldsymbol{k})$.

If $\boldsymbol{p}_0$ is the projection of the origin of the world coordinate frame, we obtain

$$\boldsymbol{p} - \boldsymbol{p}_0 = \begin{pmatrix} \frac{1}{t_z}(\boldsymbol{i}^T - \frac{t_x}{t_z}\boldsymbol{k}^T) \\ \frac{1}{t_z}(\boldsymbol{j}^T - \frac{t_y}{t_z}\boldsymbol{k}^T) \end{pmatrix} \boldsymbol{P},$$

which is indeed an instance of (5.1) with

$$u_r = \frac{t_x}{t_z}, \quad v_r = \frac{t_y}{t_z}, \quad \boldsymbol{a} = \frac{1}{t_z}(\boldsymbol{i} - u_r\boldsymbol{k}) \quad \text{and} \quad \boldsymbol{b} = \frac{1}{t_z}(\boldsymbol{j} - v_r\boldsymbol{k}).$$

Note that $\boldsymbol{a}$ and $\boldsymbol{b}$ are by construction orthogonal to the vector $\boldsymbol{R}-\boldsymbol{C}$. In particular, they form a basis of the plane orthogonal to the line joining the optical center of the camera to the reference point.

Equivalently, we have $\boldsymbol{i} = t_z\boldsymbol{a} + u_r\boldsymbol{k}$ and $\boldsymbol{j} = t_z\boldsymbol{b} + v_r\boldsymbol{k}$, and using the orthonormality constraints

$$\boldsymbol{i}\cdot\boldsymbol{j} = \boldsymbol{j}\cdot\boldsymbol{k} = \boldsymbol{k}\cdot\boldsymbol{i} = 0, \quad \text{and} \quad |\boldsymbol{i}|^2 = |\boldsymbol{j}|^2 = |\boldsymbol{k}|^2$$

yields immediately the new constraints

$$u_r v_r|\boldsymbol{a}|^2 - (1+u_r^2)\boldsymbol{a}\cdot\boldsymbol{b} = 0 \quad \text{and} \quad u_r v_r|\boldsymbol{b}|^2 - (1+v_r^2)\boldsymbol{a}\cdot\boldsymbol{b} = 0.$$

These are then trivially rearranged in the more symmetric form

$$\begin{cases} \psi_1|\boldsymbol{a}|^2 = \psi_2|\boldsymbol{b}|^2, \\ 2\boldsymbol{a}\cdot\boldsymbol{b} = \phi_1|\boldsymbol{a}|^2 + \phi_2|\boldsymbol{b}|^2, \end{cases} \tag{C.1}$$

where

$$\psi_1 = 1+v_r^2, \quad \psi_2 = 1+u_r^2, \quad \phi_1 = \frac{1}{\psi_2}u_r v_r, \quad \text{and} \quad \phi_2 = \frac{1}{\psi_1}u_r v_r,$$

and (u_r, v_r) denote the coordinates of the image of the reference point associated with the scene.

REFERENCES

[1] E.H. Adelson and J.R. Bergen. The plenoptic function and the elements of early vision. In M. Landy and J.A. Movshon, editors, *Computational Models of Visual Processing*, Cambridge, Mass., 1991. The MIT Press.

[2] Y. Aloimonos. Perspective approximations. *Image and Vision Computing*, 8(3):177–192, August 1990.

[3] S. Avidan and A. Shashua. Novel view synthesis in tensor space. In *Proc. IEEE Conf. Comp. Vision Patt. Recog.*, pages 1034–1040, San Juan, Puerto Rico, 1997.

[4] S.T. Barnard and W.B. Thompson. Disparity analysis of images. *IEEE Trans. Patt. Anal. Mach. Intell.*, 2(4):333–340, July 1980.

[5] E.B. Barrett, M.H. Brill, N.N. Haag, and P.M. Payton. Invariant linear models in photogrammetry and model-matching. In J. Mundy and A. Zisserman, editors, *Geometric Invariance in Computer Vision*, pages 277–292. MIT Press, Cambridge, Mass., 1992.

[6] B. Boufama and R. Mohr. Epipole and fundamental matrix estimation using virtual parallax. In *Proc. Int. Conf. Comp. Vision*, pages 1030–1036, Boston, MA, 1995.

[7] S. Carlsson. The double algebra: An effective tool for computing invariants in computer vision. In *Applications of Invariance in Computer Vision*, volume Lecture Notes in Computer Science vol 825, pages 145–164. Springer-Verlag, 1993.

[8] M. Chasles. Question no. 296. *Nouv. Ann. Math.*, 14(50), 1855.

[9] S.E. Chen. Quicktime VR: An image-based approach to virtual environment navigation. In *SIGGRAPH*, pages 29–38, Los Angeles, CA, August 1995.

[10] S.E. Chen and L. Williams. View interpolation for image synthesis. In *SIGGRAPH*, pages 279–288, New York, NY, August 1993.

[11] P.G. Ciarlet and J.L. Lions. *Handbook of Numerical Analysis: Volume 1.* North Holland, Amsterdam, 1990.

[12] B. Couapel. *Stéréovision par Ordinateur, Géométrie et Expérimentation.* PhD thesis, Université de Rennes 1, 1994.

[13] P. Debevec, C.J. Taylor, and J. Malik. Modeling and rendering architecture from photographs: a hybrid geometry- and image-based approach. In *SIGGRAPH*, pages 11–20, New Orleans, LA, August 1996.

[14] S. Demey, A. Zisserman, and P.A. Beardsley. Affine and projective structure from motion. In D. Hogg, editor, *Proc. British Machine Vision Conference*, pages 49–58. Springer-Verlag, Leeds, September 1992.

[15] R. Deriche and G. Giraudon. A computational approach for corner and vertex detection. *Int. J. of Comp. Vision*, 10(2):101–124, 1993.

[16] R. Deriche, Z. Zhang, Q.-T. Luong, and O.D. Faugeras. Robust recovery of the epipolar geometry for an uncalibrated stereo rig. In J.-O. Eklundh, editor, *Proc. European Conf. Comp. Vision*, volume 800 of *Lecture Notes in Computer Science*, pages 567–576. Springer-Verlag, 1994.

[17] J.J. Dongarra, J.R. Bunch, C.B. Moler, and G.W. Stewart. *LINPACK User's Guide.* Society for Industrial and Applied Mathematics, Philadelphia, 1979.

[18] L. Dreschler and H.H. Nagel. On the selection of critical points and local curvature extrema of region boundaries for interframe matching. In *International Conference on Pattern Recognition*, pages 542–544, Munich, Germany, 1982.

[19] O.D. Faugeras. What can be seen in three dimensions with an uncalibrated stereo rig? In G. Sandini, editor, *Proc. European Conf. Comp. Vision*, volume 588 of *Lecture Notes in Computer Science*, pages 563–578, Santa Margherita, Italy, 1992. Springer-Verlag.

[20] O.D. Faugeras. *Three-Dimensional Computer Vision.* MIT Press, 1993.

[21] O.D. Faugeras, Q.-T. Luong, and S.J. Maybank. Camera self-calibration: theory and experiments. In G. Sandini, editor, *Proc. European Conf. Comp. Vision*, volume 588 of *Lecture Notes in Computer Science*, pages 321–334, Santa Margherita, Italy, 1992. Springer-Verlag.

[22] O.D. Faugeras and F. Lustman. Motion and structure from motion in a piecewise-planar environment. *Journal of Pattern Recognition and Artificial Intelligence*, 2(3):485–508, 1988.

[23] M. Fischler and R. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Comm. of the ACM*, 24:381–385, 1981.

[24] P. Fua. A parallel stereo algorithm that produces dense depth maps and preserves image features. *Machine Vision and Applications*, 6(1), 1993.

[25] W. A. Fuller. *Measurement Error Models.* Wiley, New York, 1987.

[26] Y. Genc and J. Ponce. Parameterized image varieties: A novel approach to the analysis and synthesis of image sequences. In *Proc. Int. Conf. Comp. Vision*, pages 11–16, Bombay, India, January 1998.

[27] Y. Genc, J. Ponce, Y. Leedan, and P. Meer. Parameterized image varieties and estimation with bilinear constraints. In *Proc. IEEE Conf. Comp. Vision Patt. Recog.*, Fort Collins, CO, June 1999.

[28] D.B. Gennery. *Modelling the environment of an exploring vehicle by means of stereo vision.* PhD thesis, Stanford University, Stanford, CA, 1980.

[29] G.H. Golub and C.F. Van Loan. *Matrix Computation.* John Hopkins University Press, Baltimore, MD, 1993.

[30] S.J. Gortler, R. Grzeszczuk, R. Szeliski, and M. Cohen. The lumigraph. In *SIGGRAPH*, pages 43–54, New Orleans, LA, August 1996.

[31] P. Gros and L. Quan. Projective invariants for vision. Technical Report RT 90-IMAG 15-LIFIA, LIFIA-IRIMAG, December 1992.

[32] C. Harris and M. Stephens. A combined edge and corner detector. In 4th *Alvey Vision Conference*, pages 189–192, Manchester, UK, 1988.

[33] R.I. Hartley. Estimation of relative camera positions for uncalibrated cameras. In G. Sandini, editor, *Proc. European Conf. Comp. Vision*, volume 588 of *Lecture Notes in Computer Science*, pages 579–587, Santa Margherita, Italy, 1992. Springer-Verlag.

[34] R.I. Hartley. Cheirality invariants. In *Proc. DARPA Image Understanding Workshop*, pages 745–753, Washington, D.C., 1993.

[35] R.I. Hartley. Invariants of lines in space. In *Proc. DARPA Image Understanding Workshop*, pages 737–744, Washington, D.C., 1993.

[36] R.I. Hartley. An algorithm for self calibration from several views. In *Proc. IEEE Conf. Comp. Vision Patt. Recog.*, pages 908–912, Seattle, WA, June 1994.

[37] R.I. Hartley. In defence of the 8-point algorithm. In *Proc. Int. Conf. Comp. Vision*, pages 1064–1070, Boston, MA, 1995.

[38] R.I. Hartley and R. Gupta. Computing matched-epipolar projections. In *Proc. IEEE Conf. Comp. Vision Patt. Recog.*, pages 549–555, New York City, NY, June 1993.

[39] R.I. Hartley, R. Gupta, and T. Chang. Stereo from uncalibrated cameras. In *Proc. IEEE Conf. Comp. Vision Patt. Recog.*, pages 761–764, Champaign, IL, 1992.

[40] P. Havaldar, M.S. Lee, and G. Medioni. View synthesis from unregistered 2D images. In *Graphics Interface'96*, pages 61–69, 1996.

[41] D. Hearn and M.P. Baker. *Computer Graphics.* Prentice Hall, New Jersey, second edition, 1994.

[42] P.S. Heckbert. Survey of texture mapping. *IEEE Computer Graphics and Applications*, 6(11):56–67, November 1986.

[43] D.J. Heeger and A.D. Jepson. Subspace methods for recovering rigid motion I: Algorithm and implementation. *Int. J. of Comp. Vision*, 7(2):95–117, January 1992.

[44] O. Hesse. Die cubische Gleichung, von welcher die Lösung des Problems der Homographie von M. Chasles abhängt. *J. Reine Angew. Math.*, 62:188–192, 1863.

[45] R. Horaud and T. Skordas. Stereo correspondences through feature grouping and maximal cliques. *IEEE Trans. Patt. Anal. Mach. Intell.*, 11(11):1168–1180, December 1989.

[46] B.K.P. Horn. *Computer Vision.* MIT Press, Cambridge, Mass., 1986.

[47] T.S. Huang and O.D. Faugeras. Some properties of the E-matrix in two-view motion estimation. *IEEE Trans. Patt. Anal. Mach. Intell.*, 11(12):1310–1312, December 1989.

[48] S. Van Huffel and J. Vandewalle. *Total Least Squares Problems.* PA:SIAM, Philadelphia, 1991.

[49] K. Ikeuchi and B.K.P. Horn. Numerical shape from shading and occluding boundaries. *Artificial Intelligence*, 17:141–184, 1981.

[50] A.D. Jepson and D.J. Heeger. A fast subspace algorithm for recovering rigid motion. In *IEEE Workshop on Visual Motion*, pages 124–131, Princeton, NJ, 1991.

[51] A.D. Jepson and D.J. Heeger. Linear subspace methods for recovering translational direction. In L. Harris and M. Jenkin, editors, *Spatial Vision in Humans and Robots*, pages 39–62. Cambridge University Press, 1992.

[52] T. Kanade, P.J. Narayanan, and P.W. Rander. Virtualized reality: Concepts and early results. In *IEEE Workshop on the Representation of Visual Scenes*, Boston, June 1995.

[53] K. Kanatani. *Statistical Optimization for Geometric Computation: Theory and Practice.* North Holland, Amsterdam, The Netherlands, 1996.

[54] M. Kilgard. An OpenGL Toolkit. *The X Journal*, SIGS Publications, November/December 1994.

[55] J.J. Koenderink and A.J. Van Doorn. Affine structure from motion. *J. Opt. Soc. Am. A*, 8:377–385, 1990.

[56] K.N. Kutulakos and J. Vallino. Non-euclidean object representations for calibration-free video overlay. In J. Ponce, A. Zisserman, and M. Hebert, editors, *Object Representation in Computer Vision II*, number 1144 in Lecture Notes in Computer Sciences, pages 381–401. Springer-Verlag, 1996.

[57] S. Laveau and O.D. Faugeras. 3D scene representation as a collection of images and fundamental matrices. Technical Report 2205, INRIA Sophia-Antipolis, 1994.

[58] J. Lawn and R. Cipolla. Reliable extraction of the camera matrix using constraint on the epipole. In B. B. and R. Cipolla, editors, *Proc. 4rd ECCV, LNCS 1065*, pages 161–173. Springer-Verlag, Cambridge, 1996.

[59] Y. Leedan and P. Meer. Estimation with bilinear constraints in computer vision. In *Proc. Int. Conf. Comp. Vision*, pages 733–738, Bombay, India, January 1998.

[60] Yoram Leedan. *Statistical Analysis of Quadratic Problems in Computer Vision.* PhD thesis, Rutgers University, New Brunsick, NJ, October 1997.

[61] M. Levoy and P. Hanrahan. Light field rendering. In *SIGGRAPH*, pages 31–42, New Orleans, LA, August 1996.

[62] A. Lippman. Movie-maps: An application of the optical videodisc to computer graphics. In *SIGGRAPH*, volume 14(3), pages 32–42, July 1980.

[63] H.C. Longuet-Higgins. A computer algorithm for reconstructing a scene from two projections. *Nature*, 293:133–135, 1981.

[64] Q.-T. Luong. *Matrice fondamentale et calibration visuelle sur l'environnement: vers une plus grande autonomie des systèmes robotiques.* PhD thesis, University of Paris XI, Orsay, France, 1992.

[65] Q.-T. Luong, R. Deriche, O.D. Faugeras, and T. Papadopoulo. On determining the fundamental matrix: analysis of different methods and experimental results. Technical Report 1894, INRIA Sophia-Antipolis, 1993.

[66] Q.-T. Luong and O.D. Faugeras. Determining the fundamental matrix with planes: instability and new algorithms. In *Proc. IEEE Conf. Comp. Vision Patt. Recog.*, pages 489–494, New York City, NY, 1993.

[67] Q.-T. Luong and O.D. Faugeras. A stability analysis of the fundamental matrix. In J.-O. Eklundh, editor, *Proc. European Conf. Comp. Vision*, volume 800 of *Lecture Notes in Computer Science*, pages 577–588. Springer-Verlag, 1994.

[68] Q.-T. Luong and O.D. Faugeras. The fundamental matrix: theory, algorithms, and stability analysis. *Int. J. of Comp. Vision*, 17(1):43–76, January 1996.

[69] F.S. Macaulay. *The Algebraic Theory of Modular Systems.* Cambridge University Press, 1916.

[70] D. Marr and T. Poggio. A cooperative computation of stereo disparity. *Science*, 194:283–287, 1976.

[71] S.J. Maybank. *Theory of reconstruction from image motion.* Springer-Verlag, 1992.

[72] S.J. Maybank and O.D. Faugeras. A theory of self-calibration of a moving camera. *Int. J. of Comp. Vision*, 8(2):123–151, 1992.

[73] L. McMillan. *An image-based approach to three-dimensional computer graphics.* PhD thesis, University of North Carolina, Chapel Hill, NC, 1997.

[74] L. McMillan and G. Bishop. Plenoptic modelling: An image-based rendering system. In *SIGGRAPH*, pages 39–46, Los Angeles, CA, August 1995.

[75] R. Mohr, L. Quan, F. Veillon, and B. Boufama. Relative 3D reconstruction using multiple uncalibrated images. Technical Report RT 84-IMAG 12-LIFIA, LIFIA-IRIMAG, June 1992.

[76] H.P. Moravec. Toward automatic visual obstacle avoidance. In *Proc. Int. Joint Conference on Artificial Intelligence*, page 584, Cambridge, MA, 1977.

[77] J.J. Moré, B.S. Garbow, and K.E. Hillstrom. User guide for MINPACK-1. ANL-80-74, Argonne National Laboratories, 1980.

[78] P.J. Narayanan, P. Rander, and T. Kanade. Constructing virtual worlds using dense stereo. In *Proc. Int. Conf. Comp. Vision*, pages 3–10, Bombay, India, January 1998.

[79] H.K. Nishihara. PRISM, a practical real-time imaging stereo matcher. AI Memo 780, MIT, 1984.

[80] E. Nishimura, G. Xu, and S. Tsuji. Motion segmentation and correspondence using epipolar constraint. In *First Asian Conference on Computer Vision*, pages 199–204, Osaka, Japan, 1993.

[81] J.A. Noble. Finding corners. *Image and Vision Computing*, 6(2):121–128, 1988.

[82] Y. Ohta, K. Maenobu, and T. Sakai. Obtaining surface orientation from texels under perspective projection. In *Proc. Int. Joint Conference on Artificial Intelligence*, pages 746–751, 1981.

[83] S.L. Olsen. Epipolar line estimation. In G. Sandini, editor, *Proc. European Conf. Comp. Vision*, volume 588 of *Lecture Notes in Computer Science*, pages 307–311, Santa Margherita, Italy, 1992. Springer-Verlag.

[84] A.P. Pentland. Local shading analysis. *IEEE Trans. Patt. Anal. Mach. Intell.*, 6(2):170–187, 1984.

[85] C.J. Poelman and T. Kanade. A paraperspective factorization method for shape and motion recovery. *IEEE Trans. Patt. Anal. Mach. Intell.*, 19(3):206–218, March 1997.

[86] J. Ponce and Y. Genc. Epipolar geometry and linear subspace methods: A new approach to weak calibration. In *Proc. IEEE Conf. Comp. Vision Patt. Recog.*, pages 776–781, San Francisco, CA, June 1996.

[87] J. Ponce and Y. Genc. Epipolar geometry and linear subspace methods: A new approach to weak calibration. *Int. J. of Comp. Vision*, 28(3):223–243, 1998.

[88] J. Ponce, D.H. Marimont, and T.A. Cass. Analytical methods for uncalibrated stereo and motion reconstruction. In J.-O. Eklundh, editor, *Proc. European Conf. Comp. Vision*, volume 800 of *Lecture Notes in Computer Science*, pages 463–470. Springer-Verlag, 1994.

[89] A.R. Pope and D.G. Lowe. Vista: A software environment for computer vision research. In *Proc. IEEE Conf. Comp. Vision Patt. Recog.*, pages 768–772, 1994.

[90] L. Quan and T. Kanade. Affine structure from line correspondences with uncalibrated affine cameras. *IEEE Trans. Patt. Anal. Mach. Intell.*, 19(8):834–845, August 1997.

[91] L. Robert and O.D. Faugeras. Relative 3D positioning and 3D convex hull computation from a weakly calibrated stereo pair. In *Proc. Int. Conf. Comp. Vision*, pages 540–544, Berlin, Germany, 1993.

[92] K. Rohr. Modelling and identification of characteristic intensity variations. *Image and Vision Computing*, 10(2):66–76, 1992.

[93] M. Segal and K. Akeley. *The OpenGL Graphics System: A specification.* Version 1.0, Silicon Graphics, 1992.

[94] S.M. Seitz and C.R. Dyer. Complete scene structure from four point correspondences. In *Proc. Int. Conf. Comp. Vision*, pages 330–337, Boston, MA, 1995.

[95] S.M. Seitz and C.R. Dyer. Physically-valid view synthesis by image interpolation. In *Workshop on Representations of Visual Scenes*, Boston, MA, 1995.

[96] S.M. Seitz and C.R. Dyer. View morphing. In *SIGGRAPH*, pages 21–30, August 1996.

[97] L. Shapiro and M. Brady. Rejecting outliers and estimating errors in an orthogonal regression framework. Technical Report OUEL 1974/93, Dept. of Engineering Science, Oxford University, February 1993.

[98] L.G. Shapiro and R.M. Haralick. Structural descriptions and inexact matching. *IEEE Trans. Patt. Anal. Mach. Intell.*, 3:514–519, September 1981.

[99] A. Shashua. Projective depth: a geometric invariant for 3D reconstruction from two perspective/orthographic views and for visual recognition. In *Proc. Int. Conf. Comp. Vision*, pages 583–590, Berlin, Germany, 1993.

[100] A. Shashua. Trilinearity in visual recognition by alignment. In J.-O. Eklundh, editor, *Proc. European Conf. Comp. Vision*, volume 800 of *Lecture Notes in Computer Science*, pages 479–484. Springer-Verlag, 1994.

[101] J.R. Shewchuk. Triangle: Engineering a 2D quality mesh generator and Delaunay triangulator. In *ACM Workshop on Applied Computational Geometry*, pages 124–133, Philadelphia, PA, May 1996. The Triangle software is available from Netlib and http://www.cs.cmu.edu/afs/cs/project/quake/public/www/triangle.html.

[102] D. Sinclair, A. Blake, S. Smith, and C. Rothwell. Planar region detection and motion recovery. In *Proc. British Machine Vision Conference*, pages 59–68, 1992.

[103] R. Sturm. Das Problem der Projektivität und seine Anwendung auf die Flächen zweiten Grades. *Math. Annal.*, 1:533–574, 1869.

[104] S. Sullivan and J. Ponce. Automatic model construction, pose estimation, and object recognition from photographs using triangular splines. In *Proc. Int. Conf. Comp. Vision*, 1997. Accepted for publication.

[105] C.J. Taylor and D.J. Kriegman. Structure and motion from line segments in multiple images. *IEEE Trans. Patt. Anal. Mach. Intell.*, 17(10):1021–1032, November 1995.

[106] J.A. Todd. *Projective and Analytical Geometry.* Pitman Publishing Corporation, New York – Chicago, 1946.

[107] C. Tomasi. Pictures and trails: a new framework for the computation of shape and motion from perspective image sequences. In *Proc. IEEE Conf. Comp. Vision Patt. Recog.*, pages 913–918, Seattle, WA, June 1994.

[108] C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: a factorization method. *Int. J. of Comp. Vision*, 9(2):137–154, 1992.

[109] P.H. Torr, A. Zisserman, and S.J. Maybank. Robust detection of degenerate configurations for the fundamental matrix. In *Proc. Int. Conf. Comp. Vision*, pages 1037–1042, Boston, MA, 1995.

[110] P.H.S. Torr and D.W. Murray. Outlier detection and motion segmentation. In P.S. Schenker, editor, *Sensor Fusion VI*, pages 432–443. SPIE Volume 2059, Boston, September 1993.

[111] R.Y. Tsai and T.S. Huang. Uniqueness and estimation of 3D motion parameters of rigid bodies with curved surfaces. *IEEE Trans. Patt. Anal. Mach. Intell.*, 6:13–27, 1984.

[112] S. Ullman and R. Basri. Recognition by linear combination of models. *IEEE Trans. Patt. Anal. Mach. Intell.*, 13(10):992–1006, 1991.

[113] T. Viéville and O. Faugeras. Motion analysis with a camera with unknown, and possibly varying intrinsic parameters. In *Proc. Int. Conf. Comp. Vision*, pages 750–756, Boston, MA, 1995.

[114] D. Weinshall and C. Tomasi. Linear and incremental acquisition of invariant shape models from image sequences. *IEEE Trans. Patt. Anal. Mach. Intell.*, 17(5), May 1995.

[115] J. Weng, N. Ahuja, and T. Huang. Matching two perspective views. *IEEE Trans. Patt. Anal. Mach. Intell.*, 14(8):806–825, August 1992.

[116] J.H. Wilkinson and C. Reinsch. *Linear Algebra - Vol. II of Handbook for Automatic Computation.* Springer-Verlag, New York, 1971. Chapter I.10 by G.H. Golub and C. Reinsch.

[117] G. Xu, E. Nishimura, and S. Tsuji. Image correspondences and segmentation by epipolar lines: theory, algorithm and applications. Technical report, Dept. of Systems Engineering, Osaka University, Japan, July 1993.

[118] Z. Zhang. Token tracking in a cluttered scene. *Int. J. of Comp. Vision*, 12(2):110–120, 1994.

[119] Z. Zhang. Determining the epipolar geometry and its uncertainity: A review. *Int. J. of Comp. Vision*, 27(2):161–195, 1998.

[120] Z. Zhang, R. Deriche, O.D. Faugeras, and Q.-T. Luong. A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry. *Artificial Intelligence Journal*, 78:87–119, October 1995.

VITA

Yakup Genc was born in Giresun, Turkey, on October 1, 1969. He received a B.S. degree with honors (*cum laude*) in Computer Engineering in 1991 from Istanbul Technical University, Istanbul, Turkey. In 1992, he was awarded a scholorship by the Ministry of Education of Turkey to pursue his graduate studies. He continued his study at the Department of Computer Science at the University of Illinois at Urbana-Champaign in 1993 and completed his M.S. and Ph.D. programs in 1995 and 1999 respectively.

His published articles include:

- Y. Genc, J. Ponce, Y. Leedan, and P. Meer. Parameterized image varieties and estimation with bilinear constraints. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, Fort Collins, CO, June 1999.

- Y. Genc and J. Ponce. Parameterized image varieties: A novel approach to the analysis and synthesis of image sequences. In *Proc. International Conference on Computer Vision*, pages 11-16, Bombay, India, January 1998.

- J. Ponce and Y. Genc. Epipolar geometry and linear subspace methods: A new approach to weak calibration. *International Journal of Computer Vision*, 28(3):223–243, 1998.

- J. Ponce and Y. Genc. Epipolar geometry and linear subspace methods: A new approach to weak calibration. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 776–781, San Francisco, CA, June 1996.