

Dynamic Multiple View Geometry



Cheng Wan

18517509

Department of Computer Science and Engineering

Nagoya Institute of Technology

Supervisor: Jun Sato, Professor

A thesis submitted for the degree of

Doctor of Engineering

January, 2011

名古屋工業大学博士論文
甲第769号(課程修了による)
平成23年3月23日授与

Contents

1	Introduction	1
1.1	Motivation	1
1.2	Objective and Approach	2
1.3	Thesis Outline	4
2	Multiple View Geometry	6
2.1	Epipolar Geometry	6
2.2	Three View Geometry	7
2.3	Four View Geometry	9
2.4	Intersections of Four Planes	10
2.5	Summary	12
3	Dynamic Multiple View Geometry	15
3.1	Camera Trajectory Modeled by Degree- n Bezier Curve	15
3.2	Projective Projections from $(n + 3)$ D to 2D	18
3.3	$(n + 3)$ -Dimension Multiple View Geometry	18
3.3.1	Multilinear Relationships	18
3.3.2	Counting Arguments	19
3.4	Dynamic Configurations for Dynamic Multiple View Geometry	19
3.4.1	Camera Motion Following a Degree-0 Bezier Curve	20
3.4.2	Camera Motion Following a Degree-1 Bezier Curve	21
3.4.3	Camera Motion Following a Degree-2 Bezier Curve	21
3.4.4	Camera Motion Following a Degree-3 Bezier Curve	22
3.5	Summary	23
4	Dynamic Multiple View Geometry in 4D Space	24
4.1	Dynamic Multiple View Geometry for Multiple Translational Cameras	24
4.2	Projective Projections from 4D to 2D	25
4.3	Projective Multiple View Geometry from 4D to 2D	26
4.3.1	Three View Geometry	27
4.3.2	Four View Geometry	30
4.3.3	Five View Geometry	30
4.4	Intersections of Five Hyperplanes	32

4.5	Experiments	33
4.5.1	Real Image Experiment	33
4.5.2	Stability Evaluation	36
4.6	Summary	37
5	Dynamic Multiple View Geometry in 6D Space	41
5.1	Dynamic Multiple View Geometry for Multiple Curvilinear Motion Cameras	41
5.2	Multiple View Geometry for Affine Curvilinear Motion Cameras	41
5.2.1	Projection from 6D to 2D	44
5.2.2	Affine Multiple View Geometry from 6D to 2D	45
5.2.2.1	Four View Affine Geometry	45
5.2.2.2	Five View, Six View and Seven View Affine Geometry	47
5.2.3	Applications on Multiple View Geometry of Curvilinear Motion Cameras	50
5.2.3.1	View Transfer	50
5.2.3.2	3D Reconstruction	50
5.2.4	Experiments	51
5.2.4.1	Approximate Relationship between Affine Camera and Projective Camera	52
5.2.4.2	Real Image Experiment	52
5.2.4.3	3D Reconstruction	55
5.2.4.4	Stability Evaluation	55
5.3	Multiple View Geometry for Projective Curvilinear Motion Cameras	59
5.3.1	Projective Projections from 6D to 2D	60
5.3.2	Projective Multiple View Geometry from 6D to 2D	60
5.3.3	Experiments	62
5.3.3.1	Real Image Experiment	62
5.3.3.2	Stability Evaluation	62
5.4	Summary	64
6	Dynamic Multiple View Geometry with B-Spline Curve Motion Cameras	67
6.1	Camera Trajectory Modeled by Degree- n B-Spline Curve	67
6.2	B-Spline Curve	67
6.3	Non-Rigid Object Motions viewed from B-Spline Curve Motion Cameras	68
6.3.1	Cubic B-Spline Curve	68
6.3.2	Non-Rigid Object Motions viewed from Cubic B-Spline Curve Motion Cameras	68
6.4	Multiple View Geometry with Cubic B-Spline Curve Motion Cameras	70
6.4.1	Experiment	71
6.4.1.1	View Transfer	71
6.4.1.2	Stability Evaluation	72

6.5	Summary	74
7	Computing Dynamic Multiple View Geometry in 4D space from Mutual Projections of Multiple Cameras	77
7.1	Mutual Projections of Multiple Cameras	77
7.2	Computing Dynamic Multiple View Geometry in 4D Space from Mutual Projections	77
7.2.1	Using One Epipole Pair	78
7.2.1.1	Using Epipole Pair $\{\mathbf{e}_{21}, \mathbf{e}_{31}\}$ or $\{\mathbf{e}_{12}, \mathbf{e}_{32}\}$	78
7.2.1.2	Using Epipole Pair $\{\mathbf{e}_{13}, \mathbf{e}_{23}\}$	82
7.2.2	Using Two Epipole Pairs	82
7.2.2.1	Using Epipole Pair $\{\mathbf{e}_{21}, \mathbf{e}_{31}\}$ and $\{\mathbf{e}_{12}, \mathbf{e}_{32}\}$	82
7.2.2.2	Using Epipole Pair $\{\mathbf{e}_{21}, \mathbf{e}_{31}\}$ and $\{\mathbf{e}_{13}, \mathbf{e}_{23}\}$, or $\{\mathbf{e}_{12}, \mathbf{e}_{32}\}$ and $\{\mathbf{e}_{13}, \mathbf{e}_{23}\}$	92
7.2.3	Using All Three Epipole Pairs	97
7.3	Experiments	97
7.3.1	Real Image Experiment	98
7.3.2	Stability Evaluation	100
7.4	Summary	102
8	Conclusion	104
	Publication List	107
	Acknowledgments	109
	References	117

Chapter 1

Introduction

1.1 Motivation

A computer vision system processes images acquired from an electronic camera, which is like the human vision system where the brain processes images derived from the eyes. Computer vision is a rich and rewarding topic for study and research for electronic engineers, computer scientists and many others. Increasingly, it has a commercial future.

The structure from motion problem (SFM) in computer vision is to extract the 3D shape of the scene as well as the camera motion from a set of images taken by a camera undergoing unknown motion. The traditional methods in SFM provide us solutions, if a moving camera observes a static scene or a set of static cameras observe a dynamic scene [25, 85]. In this paper, we consider SFM problem under dynamic environments, where both the set of cameras and the scene change non-rigidly. In particular, we consider multiple view geometry under non-rigid object motions viewed from multiple moving cameras.

Over the past decade there has been a rapid development in the understanding and modeling of the geometry of multiple views in computer vision. The multiple view geometry is very important for describing the relationship between images taken from multiple cameras and for recovering 3D geometry from images [16, 17, 21, 25, 29, 74, 79, 80]. In the traditional multiple view geometry, the projection from the 3D space to 2D images has been assumed [25]. However, the traditional multiple view geometry is limited for describing the case where enough number of corresponding points are visible from a static configuration of multiple cameras.

Recently, some efforts for extending the multiple view geometry for more general point-camera configurations have been made [24, 27, 28, 82, 97, 98]. From stationary configurations [16, 25, 26, 82] to dynamic configurations [2, 91, 92, 97, 98], the multiple view geometry has been extensively developed. However, previous multiple view geometry involving dynamic scenes are constrained from the motions of the cameras or points moving independently along some restricted trajectory, i.e., straight line path

and in some cases second-order [2, 97, 98].

In this thesis we investigate the multiple view geometry in an absolute dynamic environment with a dynamic scene and multiple moving cameras. Moreover, the newly proposed "Dynamic Multiple View Geometry" can also describe the traditional multiple view geometry in a static environment.

1.2 Objective and Approach

In this research, we introduce a newly defined multiple view geometry named dynamic multiple view geometry, in which points in 3D undergo non-rigid motion and the cameras do arbitrary motions modeled by Degree- n Bezier curve. We find that the projective projections of non-rigid 3D motion to Degree- n Bezier curve can be represented by a projection from $(n + 3)$ D to 2D. If 3D point motions are tracked by multiple arbitrary motion cameras, the multilinear relationship under the projection from $(n + 3)$ D to 2D can be derived. Then, we analyze the projective projections from $(n + 3)$ D to 2D and deduced the degree of freedom of the extended projective camera. $(n + 3)$ -Dimension multiple view geometry involving several such extended cameras and a dynamic scene was also addressed. Multilinear relationships and the maximal linear relationship in the $(n + 3)$ D space were derived from the multifocal point relations. The counting arguments are also executed. From the geometric degree of freedom of extended projective cameras and the degree of freedom of the points in $(n + 3)$ D and all the images, the minimum number of points required for computing the multifocal tensors were available.

We next take $n = 1$ and $n = 3$ as two instances to introduce the dynamic multiple view geometry in the cases of non-rigid arbitrary motions viewed from translational motion cameras and curvilinear motion cameras respectively.

We analyze the dynamic multiple view geometry under projective projections from 4D space to 2D space, and showed that it can represent multiple view geometry under space-time projections, in which the multilinear relationship for 5 views is the maximal linear relationship in the 4D space unlike the traditional multiple view geometry. The new trilinear, quadrilinear and quintilinear relationships were analyzed. We show that the newly defined multiple view geometry can be used for describing the relationship between images taken from non-rigid motions viewed from multiple translational cameras and is very useful for generating images of non-rigid object motions viewed from arbitrary translational cameras. Here, the multifocal tensors are computed from corresponding points. For instance, the trifocal tensor can be derived by using 13 corresponding points, which are not collinear and coplanar. The method is implemented and tested by using real image sequences. The stability of extracted trifocal tensors is also evaluated.

We also extend the theory of the multiple view geometry in space-time to a multiple view geometry of multiple cameras with arbitrary curvilinear motions. We use affine camera model and projective camera model to describe the multilinear relationship

under the projection from 6D to 2D respectively, which can represent the geometric relationship of multiple curvilinear motion cameras whose motions are represented by cubic Bezier curves. The multifocal tensors defined under 6D to 2D multilinear relationships can be computed from non-rigid object motions viewed from multiple cameras with arbitrary curvilinear motions. We also showed that the multilinear relationships are very useful for generating arbitrary view images and reconstructing 3D non-rigid object motions viewed from cameras with arbitrary curvilinear motions. The method is tested in real images, and the stability is also evaluated.

The dynamic multiple view geometry, in which the camera trajectories are modeled by Degree- n Bezier curves, is proposed. However, when n is large, the multiple view geometry will become very complex and uncomputable. On the other hand, the main problem with Bezier curves is their lack of local control. To overcome the problems, we consider degree- n B-Spline curve, a piecewise curve, to represent the camera trajectories. In the mathematical field of numerical analysis, B-spline curves are very useful for representing arbitrary 3D shapes with small number of control points. Thus, we can use low degree B-spline curve to describe a complex curve. We gave the definition of the B-spline curve and especially took cubic B-spline curve as an instance of to represent the trajectory of the cameras. Although the multiple view geometry corresponding to each segment of B-spline curve motions is same as the case of Bezier curve, the camera motions could be more complex and less control points described if the camera motions are represented by B-spline curves. For example, a 2-segment cubic B-spline curve is smooth, second-order differentiable and depends on 5 control points, while two cubic successive Bezier curves are not second-order differentiable and determined by 7 control points. The synthetic experiment shows that even if all the cameras undergo complex curvilinear motions, the view transfer still can be realized by using the dynamic multiple view geometry.

We also investigate efficient, computational methods for computing the multiple view geometry in space-time. One disadvantage of the multiple view geometry in space-time is that it requires more corresponding points than the traditional multiple view geometry. It is also more sensitive to the image noise. Recently, it has been shown that if some cameras are projected to the other cameras, the multiple view geometry can be computed more stably from less corresponding points [75]. This is called mutual projections of cameras. We investigate mutual projections of cameras in four-dimensional space, and show it enables us to reduce the number of corresponding points required for computing the new multiple view geometry. Surprisingly, we no longer need any corresponding points for computing the new multiple view geometry, if all the cameras are projected to the other cameras mutually for two time intervals. We also show that the stability of the computation of new multiple view geometry is drastically improved by considering the mutual projections of cameras.

1.3 Thesis Outline

This thesis is organized as follows:

Chapter 2

The traditional multiple view geometry is reviewed. We describe the epipolar geometry of two cameras, the trifocal geometry of three cameras and extend three view geometry to four views.

Chapter 3

We describe $(n + 3)$ -Dimension to 2-Dimension dynamic multiple view geometry. We model the camera trajectory by using degree- n Bezier Curve and show that the new multiple view geometry system can represent the multiple view geometry in the case where non-rigid arbitrary motions are viewed from multiple arbitrary moving cameras, and it can also be used to represent traditional multiple view geometry.

Chapter 4

We introduce the dynamic multiple view geometry in 4D space. We also call it the multiple view geometry in space-time. In this case, the relationship among images taken from non-rigid motions viewed from multiple translational cameras can be described by the newly defined multiple view geometry. And, the multilinear relationships is up to 5 views unlike the traditional multilinear relationships. The three view, four view and five view geometries are studied extensively and new trilinear, quadrilinear and quintilinear relationships under the projective projection from 4D to 2D are presented. By some experiments, we show that it is very useful for generating images of non-rigid object motions viewed from arbitrary translational cameras.

Chapter 5

We address the dynamic multiple view geometry in 6D space which is a tensorial representation of multiple cameras with arbitrary curvilinear motions. It enables us to define multilinear relationship among image points derived from non-rigid object motions viewed from multiple cameras with arbitrary curvilinear motions. We discuss the proposed multiple view geometry under affine projection and projective projection respectively. We show the new multilinear relationship is useful for generating images and reconstructing 3D non-rigid object motions viewed from cameras with arbitrary curvilinear motions. The method is tested in real image sequences.

Chapter 6

We change the model of camera trajectory into degree- n B-Spline Curve to discuss the dynamic multiple view geometry.

Chapter 7

The mutual projection method is proposed to be applied to the multiple view geometry in space-time, which makes it possible to derive more stable results on multiple view

geometry with much less corresponding points. The method is tested in real image sequences.

Chapter 8

The conclusion is given, which reviews what has been learned from this work, describes some natural extension of it, and presents a final summary and future research topics.

Chapter 2

Multiple View Geometry

In this chapter, we will review the traditional multiple view geometry: the epipolar geometry of two cameras, the trifocal geometry of three cameras and the quadrifocal geometry of four views.

2.1 Epipolar Geometry

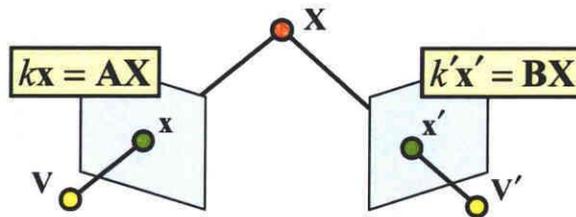


Figure 2.1: A 3D point \mathbf{X} is projected to two views as \mathbf{x} and \mathbf{x}' respectively.

Firstly, we consider the relationship that holds between the coordinates of a point seen in two separate views. Let \mathbf{x} and \mathbf{x}' be a pair of corresponding points which are the images of the same point \mathbf{X} in space as seen in the two separate views as shown in Figure 2.1. We represent the two camera matrices by \mathbf{A} and \mathbf{B} . The projection from space to image can now be expressed as $k\mathbf{x} = \mathbf{A}\mathbf{X}$ and $k'\mathbf{x}' = \mathbf{B}\mathbf{X}$ where k and k' are two undetermined constants. And we denote the i -th row of the matrix \mathbf{A} by \mathbf{a}^i , and similarly the i -th row of the matrix \mathbf{B} by \mathbf{b}^i . We also write $\mathbf{x} = [x^1, x^2, x^3]^\top$ and

$\mathbf{x}' = [x'^1, x'^2, x'^3]^\top$. That pair of equations may be written down as one equation:

$$\begin{bmatrix} \mathbf{a}^1 & x^1 \\ \mathbf{a}^2 & x^2 \\ \mathbf{a}^3 & x^3 \\ \mathbf{b}^1 & & x'^1 \\ \mathbf{b}^2 & & x'^2 \\ \mathbf{b}^3 & & x'^3 \end{bmatrix} \begin{bmatrix} \mathbf{X} \\ -k \\ -k' \end{bmatrix} = \mathbf{0}. \quad (2.1)$$

This is a 6×6 set of equations which by hypothesis has a non-zero solution, the vector $[\mathbf{X}^\top, -k, -k']^\top$. It follows that the matrix of coefficients in (2.1) must have zero determinant. It will be seen that this condition leads to a bilinear relationship between the entries of the vectors \mathbf{x} and \mathbf{x}' expressed by the fundamental matrix \mathbf{F} . We will now look specifically at the form of this relationship.

Consider the matrix appearing in (2.1). Denote it by \mathbf{M} . The determinant of \mathbf{M} may be written as an expression in terms of the quantities x^i and x'^i . Notice that the entries x^i and x'^i appear in only two columns of \mathbf{M} . This implies that the determinant of \mathbf{M} may be expressed as a quadratic expression in terms of the x^i and x'^i . In fact, since all the entries x^i appear in the same column, there can be no terms of the form $x^i x^j$ or $x'^i x'^j$. Briefly, in terms of the x^i and x'^i , the determinant of \mathbf{M} is a bilinear expression. The fact that the determinant is zero may be written as an equation

$$x^i x'^j \mathcal{F}_{ij} = 0 \quad (2.2)$$

where \mathcal{F} is a 6×6 matrix, the fundamental matrix. A way of writing the expression for \mathcal{F} makes use of the tensor ϵ_{rst} as follows:

$$\mathcal{F}_{ij} = \epsilon_{ipq} \epsilon_{jrs} \det \begin{bmatrix} \mathbf{a}^p \\ \mathbf{a}^q \\ \mathbf{b}^r \\ \mathbf{b}^s \end{bmatrix}. \quad (2.3)$$

Tensor ϵ_{rst} represents a sign based on even and odd permutation from $\{r, s, t\}$ to $\{1, 2, 3\}$ as follows:

$$\epsilon_{rst} = \begin{cases} 0, & \text{unless } r, s \text{ and } t \text{ are distinct} \\ +1, & \text{if } rst \text{ is an even permutation of } 123 \\ -1, & \text{if } rst \text{ is an odd permutation of } 123 \end{cases} \quad (2.4)$$

2.2 Three View Geometry

The determinant method of deriving the fundamental matrix can be used to derive relationships between the coordinates of points seen in three views. This analysis results in a formula for the trifocal tensor. Unlike the fundamental matrix, the trifocal

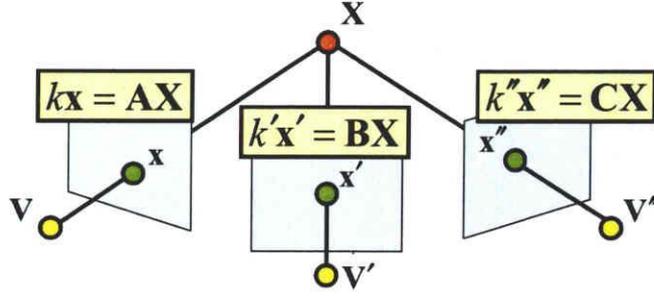


Figure 2.2: A 3D point \mathbf{X} is projected to three views as \mathbf{x} , \mathbf{x}' and \mathbf{x}'' respectively.

tensor relates both lines and points in the three images. We begin by describing the relationships for corresponding points.

Consider a point correspondence across three views: $x^i \leftrightarrow x'^i \leftrightarrow x''^i$ as shown in Figure 2.2. Let the third camera matrix be \mathbf{C} and let \mathbf{c}^i be its i -th row. Analogous to (2.1), we can write an equation describing the projection of a point \mathbf{X} into the three images as

$$\begin{bmatrix} \mathbf{A} & \mathbf{x} \\ \mathbf{B} & \mathbf{x}' \\ \mathbf{C} & \mathbf{x}'' \end{bmatrix} \begin{bmatrix} \mathbf{X} \\ -k \\ -k' \\ -k'' \end{bmatrix} = \mathbf{0}. \quad (2.5)$$

The leftmost matrix, \mathbf{M} , has 9 rows and 7 columns. From the existence of a solution to this set of equations, we deduce that its rank must be at most 6. Hence any 7×7 minor has zero determinant. This fact gives rise to the trilinear relationships that hold between the coordinates of the points x^i , x'^i and x''^i .

There are essentially two different types of 7×7 minors of \mathbf{M} . In choosing 7 rows of \mathbf{M} , only the case where three rows from one camera matrix and two rows from each of the two others is meaningful, since the other case is that one of the camera matrices only contributes one row, which leads to the bilinear relationship expressed by the fundamental matrix, as discussed in section 2.1. Then, we have the following trilinear relationship:

$$x^i x'^j x''^k \epsilon_{jqu} \epsilon_{krv} \mathcal{T}_i^{qr} = 0_{uv} \quad (2.6)$$

where u and v are free indices corresponding to the rows omitted from the matrices \mathbf{B} and \mathbf{C} , and \mathcal{T}_i^{qr} is the trifocal tensor and has the following form:

$$\mathcal{T}_i^{qr} = \epsilon_{ilm} \det \begin{bmatrix} \mathbf{a}^l \\ \mathbf{a}^m \\ \mathbf{b}^q \\ \mathbf{c}^r \end{bmatrix}. \quad (2.7)$$

Unlike the fundamental matrix, the trifocal tensor relates both lines and points in the three images, which are summarized here:

$$x^i x'^j l''_r \epsilon_{jqur} \mathcal{T}_i^{qr} = 0_u \quad (2.8)$$

$$x^i l'_q l''_r \mathcal{T}_i^{qr} = 0 \quad (2.9)$$

$$l'_p l'_q l''_r \epsilon^{piw} \mathcal{T}_i^{qr} = 0^w \quad (2.10)$$

2.3 Four View Geometry

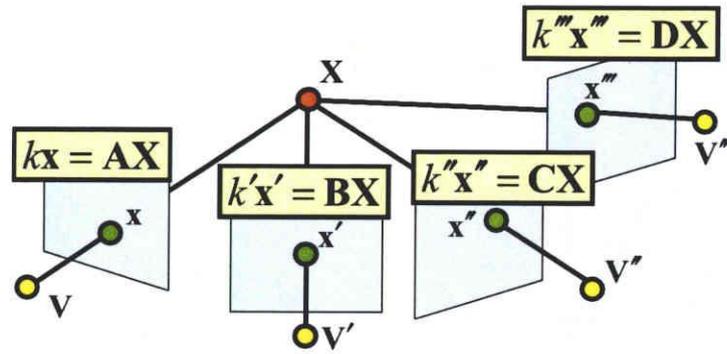


Figure 2.3: A 3D point \mathbf{X} is projected to four views as \mathbf{x} , \mathbf{x}' , \mathbf{x}'' and \mathbf{x}''' , respectively.

Similar arguments work in the case of four views. Once more, consider a point correspondence across 4 views: $x^i \leftrightarrow x'^i \leftrightarrow x''^i \leftrightarrow x'''^i$ as shown in Figure 2.3. With camera matrices \mathbf{A} , \mathbf{B} , \mathbf{C} and \mathbf{D} , the projection equations may be written as

$$\begin{bmatrix} \mathbf{A} & \mathbf{x} & & & \\ \mathbf{B} & & \mathbf{x}' & & \\ \mathbf{C} & & & \mathbf{x}'' & \\ \mathbf{D} & & & & \mathbf{x}''' \end{bmatrix} \begin{bmatrix} \mathbf{X} \\ -k \\ -k' \\ -k'' \\ -k''' \end{bmatrix} = \mathbf{0}. \quad (2.11)$$

Since this equation has a solution, the matrix \mathbf{M} on the left has rank at most 7, and so all 8×8 determinants are zero. As in the trilinear case, any determinant containing only one row from one of the camera matrices gives rise to a trilinear or bilinear relation between the remaining views. A different case occurs when we consider 8×8 determinants containing two rows from each of the camera matrices. Such a determinant leads to a new quadrilinear relationship of the form

$$x^i x'^j x''^k x'''^l \epsilon_{ipw} \epsilon_{jqx} \epsilon_{kry} \epsilon_{lsz} \mathcal{Q}^{pqrs} = 0_{wxyz} \quad (2.12)$$

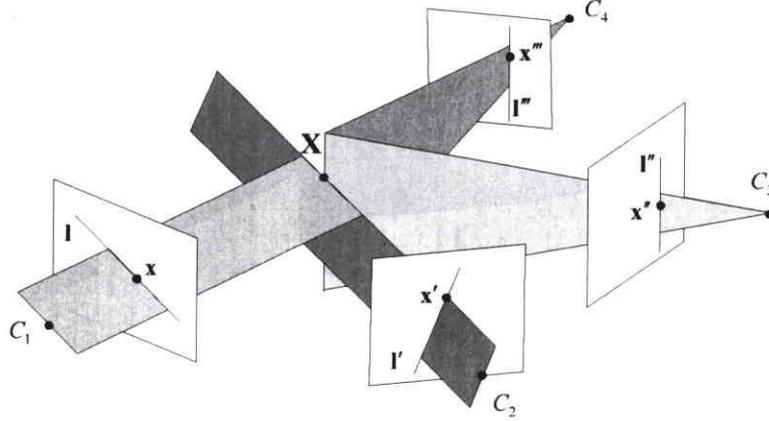


Figure 2.4: A line-line-line-line correspondence $l \leftrightarrow l' \leftrightarrow l'' \leftrightarrow l'''$ involving four images may be interpreted as: the four lines back-project to planes that meet in a point in space.

where each choice of the free variables w , x , y and z gives a different equation, and the 4-dimensional quadrifocal tensor Q^{pqrs} is defined by

$$Q^{pqrs} = \det \begin{bmatrix} \mathbf{a}^p \\ \mathbf{b}^q \\ \mathbf{c}^r \\ \mathbf{d}^s \end{bmatrix}. \quad (2.13)$$

As in the case of the trifocal tensor, there are also relations between lines and points in the case of the four-view tensor:

$$x^i x'^j x''^k l_s''' \epsilon_{ipw} \epsilon_{jqx} \epsilon_{kry} Q^{pqrs} = 0_{wxy} \quad (2.14)$$

$$x^i x'^j l_r'' l_s''' \epsilon_{ipw} \epsilon_{jqx} Q^{pqrs} = 0_{wx} \quad (2.15)$$

$$x^i l_q' l_r'' l_s''' \epsilon_{ipw} Q^{pqrs} = 0_w \quad (2.16)$$

$$l_p l_q' l_r'' l_s''' Q^{pqrs} = 0. \quad (2.17)$$

2.4 Intersections of Four Planes

The multi-view tensors may be given a different derivation, which sheds a little more light on their meaning. In this interpretation, the basic geometric property is the intersection of four planes. Four planes in space will generally not meet in a common point. A necessary and sufficient condition for them to do so is that the determinant of the 4×4 matrix formed from the vectors representing the planes should vanish.

In this section only we shall represent the determinant of a 4×4 matrix with rows a , b , c and d by $a \wedge b \wedge c \wedge d$. In a more general context, the symbol \wedge represents

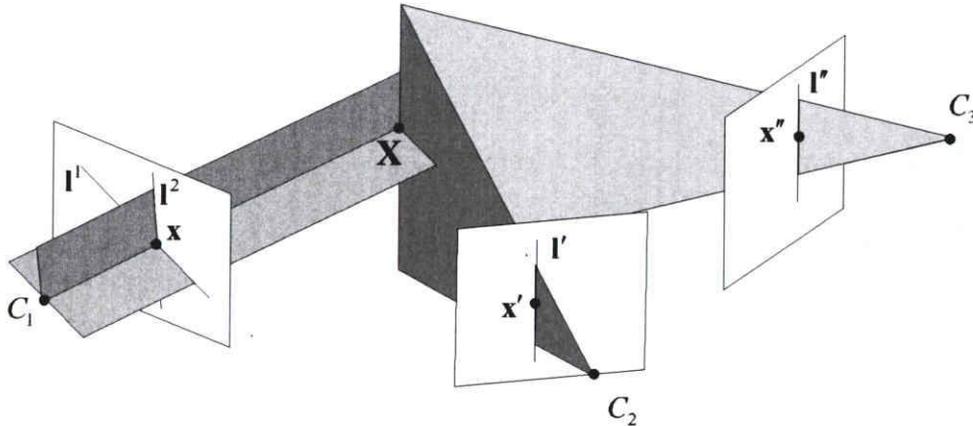


Figure 2.5: A point-line-line correspondence $\mathbf{x} \leftrightarrow l' \leftrightarrow l''$ involving three images may be interpreted as follows. Two arbitrary lines are chosen to pass through the point \mathbf{x} in the first image. The four lines then back-project to planes that meet in a point in space.

the meet (or intersection) operator in the double algebra. However, for the present purposes the reader need only consider it as a shorthand for the determinant.

We start with the quadrifocal tensor for which the derivation is easiest. Consider four lines l, l', l'', l''' in images formed from four cameras with camera matrices $\mathbf{A}, \mathbf{B}, \mathbf{C}$ and \mathbf{D} . The back projection of a line l through camera \mathbf{A} is written as the plane $l_i a^i$. The condition that these four planes are coincident shown as Figure 2.4 may be written as

$$(l_p \mathbf{a}^p) \wedge (l'_q \mathbf{b}^q) \wedge (l''_r \mathbf{c}^r) \wedge (l'''_s \mathbf{d}^s) = 0. \quad (2.18)$$

However, since the determinant is linear in each row, this may be written as

$$0 = l_p l'_q l''_r l'''_s (\mathbf{a}^p \wedge \mathbf{b}^q \wedge \mathbf{c}^r \wedge \mathbf{d}^s) \stackrel{\text{def}}{=} l_p l'_q l''_r l'''_s Q^{pqrs} \quad (2.19)$$

This corresponds to the definition (2.13) and line relation (2.17) for the quadrifocal tensor. The basic geometric property is the intersection of the four planes in space.

Trifocal tensor derivation. Consider now a point-line-line relationship $x^i \leftrightarrow l'_j \leftrightarrow l''_k$, for three views and let l^1_p and l^2_q be two lines in the first image that pass through the image point \mathbf{x} . The planes back-projected from the four lines meet in a point (see Figure 2.5). So we can write:

$$l^1_i l^2_m l'_q l''_r (\mathbf{a}^i \wedge \mathbf{a}^m \wedge \mathbf{b}^q \wedge \mathbf{c}^r) = 0. \quad (2.20)$$

The next step is an algebraic trick-to multiply this equation by the $\epsilon^{ilm} \epsilon_{ilm}$, which is a scalar value. The result after regrouping is

$$(l^1_i l^2_m \epsilon^{ilm}) l'_q l''_r \epsilon_{ilm} (\mathbf{a}^i \wedge \mathbf{a}^m \wedge \mathbf{b}^q \wedge \mathbf{c}^r) = 0. \quad (2.21)$$

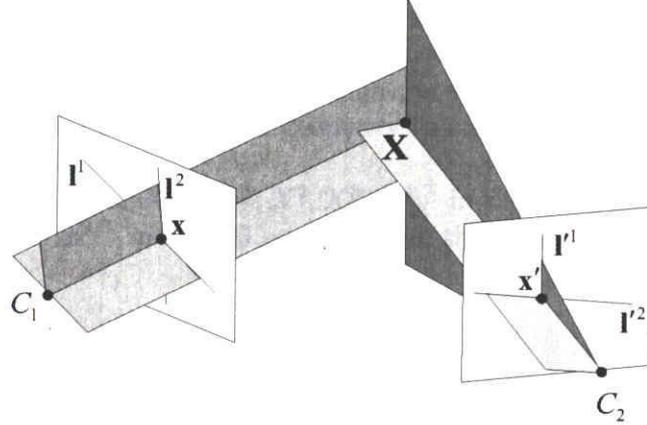


Figure 2.6: A point-point correspondence $\mathbf{x} \leftrightarrow \mathbf{x}'$ involves two images when two arbitrary lines are chosen to pass through the point \mathbf{x} in the first image and other two arbitrary lines are chosen to pass through the point \mathbf{x}' in the second image. The four lines then back-project to planes that meet in a point in space.

Now the expression $l_l^1 l_m^2 \epsilon^{ilm}$ is simply the cross-product of the two lines \mathbf{l}^1 and \mathbf{l}^2 , in other words their intersection point \mathbf{x} . Thus finally we can write

$$0 = x^i l_q^1 l_r^2 (\epsilon_{ilm} (\mathbf{a}^l \wedge \mathbf{a}^m \wedge \mathbf{b}^q \wedge \mathbf{c}^r)) \stackrel{\text{def}}{=} x^i l_q^1 l_r^2 \mathcal{T}_i^{qr} \quad (2.22)$$

which are the definition (2.7) and basic incidence relation (2.9) for the trifocal tensor.

Fundamental matrix. We can derive the fundamental matrix in the same manner. Given a correspondence $\mathbf{x} \leftrightarrow \mathbf{x}'$ select pairs of lines l_p^1 and l_q^2 passing through \mathbf{x} , and l_r^1 and l_s^2 passing through \mathbf{x}' . The back-projected planes all meet in a point as Figure 2.6, so we write

$$l_p^1 l_q^2 l_r^1 l_s^2 (\mathbf{a}^p \wedge \mathbf{a}^q \wedge \mathbf{b}^r \wedge \mathbf{b}^s) = 0. \quad (2.23)$$

Multiplying by $(\epsilon^{ipq} \epsilon_{ipq})(\epsilon^{jrs} \epsilon_{jrs})$ and proceeding as before leads to the coplanarity constraint

$$0 = x^i x'^j (\epsilon_{ipq} \epsilon_{jrs} (\mathbf{a}^p \wedge \mathbf{a}^q \wedge \mathbf{b}^r \wedge \mathbf{b}^s)) \quad (2.24)$$

which can be compared with (2.3).

2.5 Summary

In this chapter, we introduced the multiple view geometry for two, three and four views respectively, and analysed the fundamental projective relations over multiple views arise from the intersection of lines and planes. These intersection properties are

represented by the vanishing of determinants formed from the camera matrices of the views. The fundamental matrix, the trifocal tensor and the quadrifocal tensor arise naturally from these determinants as the multiple view tensors. Here, we give more general properties of the multiple view geometry for readers to see why the multilinear relationship is up to 4 views.

We first consider a point $\mathbf{X} = [X^1, X^2, X^3, X^4]^\top$ in the 3D space and it is projected to a point $\mathbf{x} = [x^1, x^2, x^3]^\top$ in the 2D space. Then, the projection from \mathbf{X} to \mathbf{x} can be described as follows:

$$\lambda \mathbf{x} = \mathbf{P}\mathbf{X} \quad (2.25)$$

where \mathbf{P} denotes the 3×4 camera matrix and has 11 degrees of freedom.

From (2.25), we have the following equation for N cameras:

$$\begin{bmatrix} \mathbf{P} & \mathbf{x} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{P}' & \mathbf{0} & \mathbf{x}' & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{P}'' & \mathbf{0} & \mathbf{0} & \mathbf{x}'' & \cdots & \mathbf{0} \\ \vdots & & & & & \vdots \end{bmatrix} \begin{bmatrix} \mathbf{X} \\ \lambda \\ \lambda' \\ \lambda'' \\ \vdots \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \vdots \end{bmatrix} \quad (2.26)$$

where, the leftmost matrix \mathbf{M} in (2.26) is $3N \times (N + 4)$, and the $(N + 4) \times (N + 4)$ minors \mathbf{Q} of \mathbf{M} constitute multilinear relationships as follows:

$$\det \mathbf{Q} = 0. \quad (2.27)$$

We can choose any $N + 4$ rows from \mathbf{M} to constitute \mathbf{Q} , but we have to take at least 2 rows from each camera for deriving meaningful N view relationships. Thus, the following condition must hold for defining multilinear relationships for N view geometry:

$$2N \leq N + 4. \quad (2.28)$$

Thus, we find that the multilinear relationship for 4 views is the maximal linear relationship.

The geometric DOF of N cameras is $11N - 15$, since each camera has 11 DOF and these N cameras are in a single 3D projective space whose DOF is 15. Therefore, the DOF of bifocal, trifocal and quadrifocal tensor is 7, 18 and 29 respectively.

The bilinear, trilinear and quadrilinear relationship have been derived and have the following forms:

$$x^i x'^j \mathcal{F}_{ij} = 0 \quad (2.29)$$

$$x^i x'^j x''^k \epsilon_{jqv} \epsilon_{krv} \mathcal{T}_i^{qr} = 0_{uv} \quad (2.30)$$

$$x^i x'^j x''^k x'''^l \epsilon_{ipw} \epsilon_{jqx} \epsilon_{kry} \epsilon_{lsz} \mathcal{Q}^{pqrs} = 0_{wxyz} \quad (2.31)$$

If we have N corresponding points in two, three and four views, the multilinear relationships gives us N , $9N$ and $81N$ equations respectively, but only N , $4N$ and $16N - N\mathcal{C}_2$ of them are linearly independent. On the other hand, \mathcal{F}_{ij} , \mathcal{T}_i^{qr} and \mathcal{Q}^{pqrs} has 9, 27 and 81 entries respectively, but except a scale ambiguity, they has 8, 26 and 80 free parameters respectively. Thus, minimum of 8, 7 and 6 corresponding points are required to compute the multifocal tensors linearly.

Chapter 3

Dynamic Multiple View Geometry

3.1 Camera Trajectory Modeled by Degree- n Bezier Curve

The traditional multiple view geometry as introduced in Chapter 2 is limited for describing the case where enough number of corresponding points are visible from a static configuration of multiple cameras. In this chapter, we consider the multiple view geometry in a dynamic environment, in which the point motion in 3D space is non-rigid and the camera trajectory is modeled by the degree- n Bezier curve.

A Bezier curve is a parametric curve frequently used in computer graphics and related fields. In vector graphics, Bezier curves are used to model smooth curves. Bezier curves are also used in animation as a tool to control motion. Degree- n Bezier curve is defined as follows:

$$\mathbf{B} = \sum_{i=0}^n \mathbf{b}_{i,n}(t) \mathbf{G}_i, \quad t \in [0, 1] \quad (3.1)$$

where \mathbf{G}_i is the i th control point and the polynomials $\mathbf{b}_{i,n}(t)$ known as Bernstein basis polynomials of degree n is written as:

$$\begin{aligned} \mathbf{b}_{i,n}(t) &= \binom{n}{i} t^i (1-t)^{n-i} \\ &= \binom{n}{i} t^i \sum_{j=0}^{n-i} \binom{n-i}{n-i-j} (-t)^{n-i-j} \\ &= \sum_{j=0}^{n-i} \binom{n}{i} \binom{n-i}{j} (-1)^{n-i-j} t^{n-j} \\ &= \sum_{j=0}^{n-i} C(n, i, j) t^{n-j} \end{aligned} \quad (3.2)$$

Here, $\binom{n}{i}$ is the binomial coefficient and has the alternative notation,

$${}^nC_i = \binom{n}{i} = \frac{n!}{i!(n-i)!}, \quad (3.3)$$

and $C(n, i, j)$ denotes the following function:

$$C(n, i, j) = \binom{n}{i} \binom{n-i}{j} (-1)^{n-i-j} \quad (3.4)$$

Suppose T denotes time and T_a represents the total time of the camera motion. Then, the relationship among parameter t , time T and total time T_a can be described like this:

$$t = \frac{T}{T_a} \quad (3.5)$$

Then, Bezier curve \mathbf{B} which we utilize to model the trajectory of camera motion can be rewritten as follows:

$$\begin{aligned} \mathbf{B} &= \sum_{i=0}^n \mathbf{b}_{i,n}(t) \mathbf{G}_i \\ &= \sum_{i=0}^n \mathbf{G}_i \sum_{j=0}^{n-i} C(n, i, j) t^{n-j} \\ &= [\mathbf{G}_0 \quad \mathbf{G}_1 \quad \cdots \quad \mathbf{G}_n] \mathbf{A} \begin{bmatrix} t^n \\ t^{n-1} \\ \vdots \\ 1 \end{bmatrix} \\ &= \mathbf{GAE} \begin{bmatrix} T^n \\ T^{n-1} \\ \vdots \\ 1 \end{bmatrix} \end{aligned} \quad (3.6)$$

where,

$$\begin{aligned} \mathbf{G} &= [\mathbf{G}_0 \quad \mathbf{G}_1 \quad \cdots \quad \mathbf{G}_n], \\ \mathbf{A} &= \begin{bmatrix} A_{00} & A_{01} & A_{02} & \cdots & A_{0n-1} & A_{0n} \\ A_{10} & A_{11} & A_{12} & \cdots & A_{1n-1} & 0 \\ A_{20} & A_{21} & A_{22} & \cdots & 0 & 0 \\ \vdots & & & & & \vdots \\ A_{n0} & 0 & 0 & \cdots & 0 & 0 \end{bmatrix}, \\ A_{ij} &= C(n, i, j), \quad (i = 0, \dots, n, \quad j = 0, \dots, n-i), \\ \mathbf{E} &= \text{diag}\left[\frac{1}{T_a^n}, \frac{1}{T_a^{n-1}}, \dots, 1\right]. \end{aligned}$$

Consider a usual projective camera which projects points in 3D to 2D images. The motions of a point in the 3D space can be represented by homogeneous coordinates, $\mathbf{X}(T) = [X(T), Y(T), Z(T), 1]^T$. The motions are projected to images, and can be observed as a set of points, $\mathbf{x}(T) = [x(T), y(T), 1]^T$. Thus, point motions are projected to the Bezier curve motion camera as follows:

$$\lambda \mathbf{x}(T) = \mathbf{P}(\mathbf{X}(T) - \mathbf{B}) \quad (3.7)$$

where \mathbf{P} denotes a 3×4 projective camera matrix, and λ denotes a scalar which represents a scale ambiguity. By substituting (3.6) into (3.7), we have the following equations:

$$\begin{aligned} \lambda \mathbf{x}(T) &= \mathbf{P}(\mathbf{X}(T) - \mathbf{GAE} \begin{bmatrix} T^n \\ T^{n-1} \\ \vdots \\ 1 \end{bmatrix}) \\ &= \mathbf{P}[\mathbf{I}, -\mathbf{GAE}] \begin{bmatrix} \mathbf{X}(T) \\ T^n \\ T^{n-1} \\ \vdots \\ 1 \end{bmatrix} \\ &= \mathbf{P}_a \begin{bmatrix} \mathbf{X}(T) \\ T^n \\ T^{n-1} \\ \vdots \\ 1 \end{bmatrix} \\ &= \mathbf{P}_a \begin{bmatrix} X(T) \\ Y(T) \\ Z(T) \\ 1 \\ T^n \\ T^{n-1} \\ \vdots \\ 1 \end{bmatrix} \\ &= \mathbf{P}' \begin{bmatrix} X(T) \\ Y(T) \\ Z(T) \\ T^n \\ T^{n-1} \\ \vdots \\ 1 \end{bmatrix} \end{aligned} \quad (3.8)$$

where

$$\mathbf{P}_a = \mathbf{P}[\mathbf{I}, -\mathbf{GAE}] \quad (3.9)$$

represents a $3 \times (n+5)$ matrix and \mathbf{P}' denotes a $3 \times (n+4)$ extended camera matrix. The $(n+4)$ th column of \mathbf{P}' is derived by merging the 4th column and the $(n+5)$ th column of \mathbf{P}_a . We therefore find that, from (3.8), the projections of point motions to multiple cameras with arbitrary motions can be described by the multilinear relationship under the projection from $(n+3)$ D to 2D. In the next section, the geometry of such projections will be analyzed in more detail.

3.2 Projective Projections from $(n+3)$ D to 2D

We first consider a projection from $(n+3)$ D space to 2D space. Let $\mathbf{X} = [X, Y, Z, T^n, T^{n-1}, \dots, 1]^\top$ be the homogeneous coordinates of a $(n+3)$ D space point projected to a point in the 2D space, whose homogeneous coordinates are represented by $\mathbf{x} = [x^1, x^2, x^3]^\top$. Then, the extended affine projection from \mathbf{X} to \mathbf{x} can be described as follows:

$$\mathbf{x} \sim \mathbf{P}\mathbf{X} \quad (3.10)$$

where (\sim) denotes equality up to a scale, and \mathbf{P} denotes the following $3 \times (n+4)$ matrix:

$$\mathbf{P} = \begin{bmatrix} p_{11} & p_{12} & \cdots & p_{1(n+4)} \\ p_{21} & p_{22} & \cdots & p_{2(n+4)} \\ p_{31} & p_{32} & \cdots & p_{3(n+4)} \end{bmatrix} \quad (3.11)$$

From (3.11), we find that the extended projective camera, \mathbf{P} , has $3 \times (n+4) - 1 = 3n + 11$ DOF except a scale. In the next section, we consider the dynamic multiple view geometry of the extended projective cameras.

3.3 $(n+3)$ -Dimension Multiple View Geometry

3.3.1 Multilinear Relationships

From (3.10), we have the following equation for K extended projective cameras:

$$\begin{bmatrix} \mathbf{P} & \mathbf{x} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{P}' & \mathbf{0} & \mathbf{x}' & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{P}'' & \mathbf{0} & \mathbf{0} & \mathbf{x}'' & \cdots & \mathbf{0} \\ \vdots & & & & & \vdots \end{bmatrix} \begin{bmatrix} \mathbf{X} \\ \lambda \\ \lambda' \\ \lambda'' \\ \vdots \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \vdots \end{bmatrix} \quad (3.12)$$

where, the leftmost matrix, \mathbf{M} , in (3.12) is $3K \times (K+n+4)$. From the existence of a solution to this set of equations, we deduce that its rank must be at most $K+n+3$.

Hence any $(K + n + 4) \times (K + n + 4)$ minors \mathbf{Q} of \mathbf{M} has zero determinant, that arises the constitute multilinear relationships under the extended projection as follows:

$$\det \mathbf{Q} = 0. \quad (3.13)$$

We can choose any $K + n + 4$ rows from \mathbf{M} to constitute \mathbf{Q} , but we have to take at least 2 rows from each camera for deriving meaningful K view relationships (note, each camera has 3 rows in \mathbf{M}). Thus, the following inequality must hold for defining multilinear relationships for K view geometry in the $(n + 3)$ D space:

$$K + n + 4 \geq 2K \quad (3.14)$$

Thus, we find that, the multilinear relationship for $n + 4$ views is the maximal linear relationship in the $(n + 3)$ D space.

3.3.2 Counting Arguments

We next consider the minimum number of points required for computing the multifocal tensors. The geometric DOF S of K extended projective cameras is as follows:

$$S = (3n + 11)K - (n + 4)^2 + 1, \quad (3.15)$$

since each extended projective camera has $(3n + 11)$ DOF and these K cameras are in a single $(n + 3)$ D projective space whose DOF is $(n + 4)^2 - 1$. Meanwhile, if we are given M points in the $(n + 3)$ D space, and let them be projected to K cameras defined in (3.10). Then, we derive $2MK$ measurements from images, while we have to compute $(n + 3)M + S$ components for fixing all the geometry in the $(n + 3)$ D space. Thus, the following condition must hold for computing the multifocal tensors from images:

$$2MK \geq (n + 3)M + S. \quad (3.16)$$

Then, we have the following inequality:

$$M \geq \frac{S}{2K - n - 3} \quad (3.17)$$

Thus, we find that minimum of $\frac{S}{2K - n - 3}$ points are required to compute multifocal tensors in dynamic multiple view geometry. The minimum number of points required in the cases of 3D, 4D, 5D and 6D multiple view geometry are illustrated in Table 3.1.

3.4 Dynamic Configurations for Dynamic Multiple View Geometry

In our dynamic multiple view geometry theory, it has different dynamic configurations in different dimension space. We list several typical and basic examples of dynamic configurations to demonstrate this property.

Table 3.1: The minimum number of points required for computing the multifocal tensors with non-linear method in $(n + 3)$ -Dimension multiple view geometry.

views	2	3	4	5	6	7
3D	7	6	6	-	-	-
4D	-	9	8	8	-	-
5D	-	16	11	10	10	-
6D	-	-	16	13	12	12

3.4.1 Camera Motion Following a Degree-0 Bezier Curve



Figure 3.1: Camera trajectory. \mathbf{G}_i denotes the i th control point of the Bezier curve.

We first consider the case of degree-0 Bezier curve ($n = 0$). By substitute $n = 0$ into (3.1), we find that the camera motion can be represented as follows:

$$\begin{aligned}
 \mathbf{B} &= \sum_{i=0}^0 \mathbf{b}_{i,0}(t) \mathbf{G}_i \\
 &= \mathbf{b}_{0,0}(t) \mathbf{G}_0 \\
 &= \mathbf{G}_0
 \end{aligned} \tag{3.18}$$

In this case, the camera is not moving but static as shown in Figure 3.1 which is a special case and just coincides with the traditional multiple view geometry as introduced in Chapter 2. Therefore, our dynamic multiple view geometry theory can also be used to describe the case of the traditional multiple view geometry.

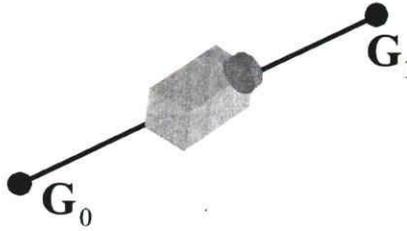


Figure 3.2: Camera trajectory. \mathbf{G}_i denotes the i th control point of the Bezier curve.

3.4.2 Camera Motion Following a Degree-1 Bezier Curve

When the camera moves following a degree-1 Bezier curve, the motions of camera is written like this:

$$\begin{aligned}
 \mathbf{B} &= \sum_{i=0}^1 \mathbf{b}_{i,1}(t) \mathbf{G}_i \\
 &= \mathbf{b}_{0,1}(t) \mathbf{G}_0 + \mathbf{b}_{1,1}(t) \mathbf{G}_1 \\
 &= (1-t) \mathbf{G}_0 + t \mathbf{G}_1
 \end{aligned} \tag{3.19}$$

The trajectory of camera is a line which goes through \mathbf{G}_0 and \mathbf{G}_1 as shown in Figure 3.2, that means the cameras are translational. The dynamic multiple view geometry defined here can represents the relationship among several translational motion cameras. This case will be analyzed in depth in Chapter 4.

3.4.3 Camera Motion Following a Degree-2 Bezier Curve

If the camera is moving following a Degree-2 Bezier curve:

$$\begin{aligned}
 \mathbf{B} &= \sum_{i=0}^2 \mathbf{b}_{i,2}(t) \mathbf{G}_i \\
 &= \mathbf{b}_{0,2}(t) \mathbf{G}_0 + \mathbf{b}_{1,2}(t) \mathbf{G}_1 + \mathbf{b}_{2,2}(t) \mathbf{G}_2 \\
 &= (1-t)^2 \mathbf{G}_0 + 2(1-t)t \mathbf{G}_1 + t^2 \mathbf{G}_2
 \end{aligned} \tag{3.20}$$

as shown in (3.20) and Figure 3.3, the camera motion is a quadratic curve. The geometry among such curvilinear motion cameras can also be described by the dynamic multiple view geometry.

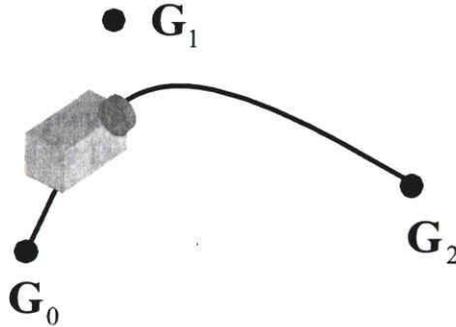


Figure 3.3: Camera trajectory. G_i denotes the i th control point of the Bezier curve.

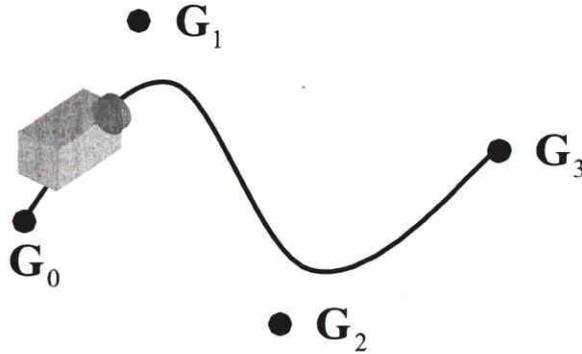


Figure 3.4: Camera trajectory. G_i denotes the i th control point of the Bezier curve.

3.4.4 Camera Motion Following a Degree-3 Bezier Curve

We next consider the case where $n = 3$. By substituting $n = 3$ into (3.1), we have the following camera motions:

$$\begin{aligned}
 \mathbf{B} &= \sum_{i=0}^3 \mathbf{b}_{i,3}(t) \mathbf{G}_i \\
 &= \mathbf{b}_{0,3}(t) \mathbf{G}_0 + \mathbf{b}_{1,3}(t) \mathbf{G}_1 + \mathbf{b}_{2,3}(t) \mathbf{G}_2 + \mathbf{b}_{3,3}(t) \mathbf{G}_3 \\
 &= (1-t)^3 \mathbf{G}_0 + 3(1-t)^2 t \mathbf{G}_1 + 3(1-t) t^2 \mathbf{G}_2 + t^3 \mathbf{G}_3
 \end{aligned} \tag{3.21}$$

As shown in (3.21) and Figure 3.4, the camera motion is cubic curve. This case will be studied in Chapter 5 extensively.

Even if the cameras undergo more complex curvilinear motion, the dynamic multiple view geometry is still competent.

3.5 Summary

This chapter introduced a newly defined multiple view geometry named dynamic multiple view geometry, in which points in 3D undergo non-rigid motion and the cameras do arbitrary motions modeled by Degree- n Bezier curve. We found that the projective projections of non-rigid 3D motion to Degree- n Bezier curve can be represented by a projection from $(n + 3)$ D to 2D. If 3D point motions are tracked by multiple arbitrary motion cameras, the multilinear relationship under the projection from $(n + 3)$ D to 2D can be derived. Then, we analyzed the projective projections from $(n + 3)$ D to 2D and deduced the degree of freedom of the extended projective camera. $(n + 3)$ -Dimension multiple view geometry involving several such extended cameras and a dynamic scene was also addressed. Multilinear relationships and the maximal linear relationship in the $(n + 3)$ D space were derived from the multifocal point relations. Finally, counting arguments were executed. From the geometric degree of freedom of extended projective cameras and the degree of freedom of the points in $(n + 3)$ D and all the images, the minimum number of points required for computing the multifocal tensors were derived.

Chapter 4

Dynamic Multiple View Geometry in 4D Space

4.1 Dynamic Multiple View Geometry for Multiple Translational Cameras

When cameras undergo Degree-1 Bezier curves as shown in Figure 4.1, the motion of the cameras are translational and the relationship among these cameras is just the case of the dynamic multiple view geometry in 4D space as discussed in the former chapter. We next show that this camera model can be used for describing non-rigid object motions viewed from multiple cameras with translational motions of constant speed.

Substituting $n = 1$ into (3.6), we have the following equation of Degree-1 Bezier curve:

$$\begin{aligned} \mathbf{B} &= \mathbf{GAE} \begin{bmatrix} T \\ 1 \end{bmatrix} \\ &= [\mathbf{G}_0 \quad \mathbf{G}_1] \begin{bmatrix} -1 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \frac{1}{T_a} & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} T \\ 1 \end{bmatrix} \end{aligned} \quad (4.1)$$

where, \mathbf{G}_i denotes the i th control point of the Bezier curve, T_a is the total time of the camera motion and T denotes time.

Let us consider a usual projective camera which projects points in 3D to 2D images. If the motions of the projective camera are translational constrained by Degree-1 Bezier

curve, non-rigid motions $\mathbf{X}(T)$ are projected to images $\mathbf{x}(T)$ as:

$$\begin{aligned}
\lambda \mathbf{x}(T) &= \mathbf{P}(\mathbf{X}(T) - \mathbf{B}) \\
&= \mathbf{P}(\mathbf{X}(T) - \mathbf{GAE} \begin{bmatrix} T \\ 1 \end{bmatrix}) \\
&= \mathbf{P}[\mathbf{I}, -\mathbf{GAE}] \begin{bmatrix} \mathbf{X}(T) \\ T \\ 1 \end{bmatrix} \\
&= \mathbf{P}' \begin{bmatrix} X(T) \\ Y(T) \\ Z(T) \\ T \\ 1 \end{bmatrix}
\end{aligned} \tag{4.2}$$

where \mathbf{P} is a 3×4 projection matrix of a usual camera, and $X(T)$, $Y(T)$, $Z(T)$ denote coordinates of a 3D point at time T ,

$$\mathbf{G} = \begin{bmatrix} \mathbf{G}_0 & \mathbf{G}_1 \end{bmatrix} \tag{4.3}$$

$$\mathbf{A} = \begin{bmatrix} -1 & 1 \\ 1 & 0 \end{bmatrix} \tag{4.4}$$

$$\mathbf{E} = \begin{bmatrix} \frac{1}{T_a} & 0 \\ 0 & 1 \end{bmatrix} \tag{4.5}$$

and, \mathbf{P}' is a 3×5 projection matrix of an extended projective camera. In the next sections, we will describe the dynamic multiple view geometry in 4D space.

4.2 Projective Projections from 4D to 2D

We first consider projective projections from 4D space to 2D space. This projection is used to describe the relationship between the real space-time and 2D images, and for analyzing the multiple view geometry under space-time projections. Let $\mathbf{X} = [X^1, X^2, X^3, X^4, X^5]^\top$ be the homogeneous coordinates of a 4D space point projected to a point in the 2D space, whose homogeneous coordinates are represented by $\mathbf{x} = [x^1, x^2, x^3]^\top$. Then, the extended projective projection from \mathbf{X} to \mathbf{x} can be described as follows:

$$\mathbf{x} \sim \mathbf{P}\mathbf{X} \tag{4.6}$$

where (\sim) denotes equality up to a scale, and \mathbf{P} denotes the following 3×5 matrix:

$$\mathbf{P} = \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} & m_{15} \\ m_{21} & m_{22} & m_{23} & m_{24} & m_{25} \\ m_{31} & m_{32} & m_{33} & m_{34} & m_{35} \end{bmatrix} \tag{4.7}$$

From (4.6), we find that the extended projective camera, \mathbf{P} , has 14 DOF. In the next section, we consider the multiple view geometry of the extended projective cameras.

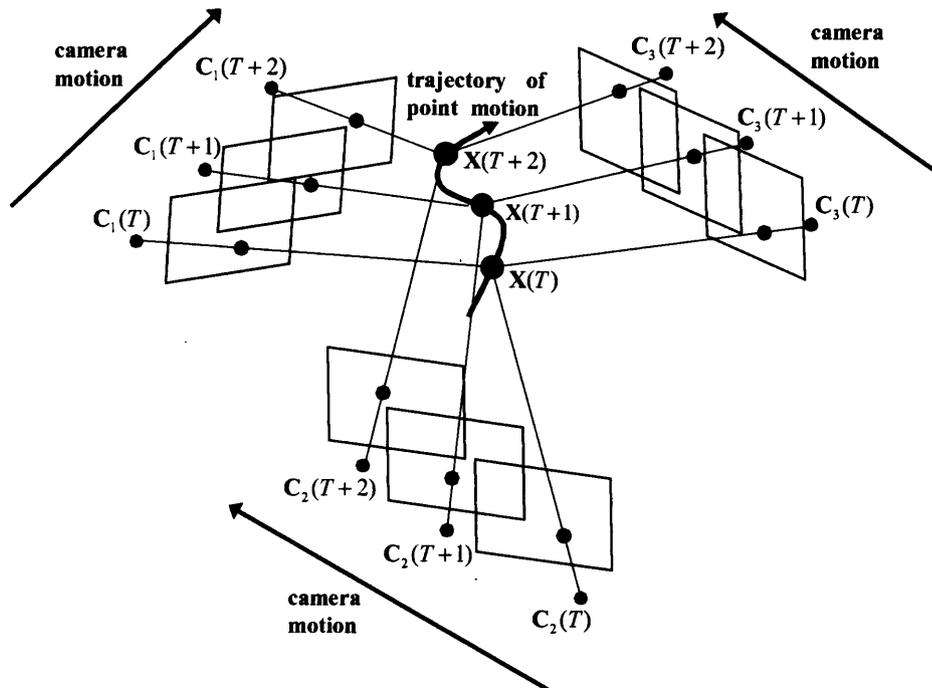


Figure 4.1: A moving point in 3D space and its projections in three translational projective cameras. The multifocal tensor defined under space-time projections can describe the relationship between these image projections.

4.3 Projective Multiple View Geometry from 4D to 2D

From (4.6), we have the following equation for N extended projective cameras:

$$\begin{bmatrix} \mathbf{P} & \mathbf{x} & 0 & 0 & \cdots & 0 \\ \mathbf{P}' & 0 & \mathbf{x}' & 0 & \cdots & 0 \\ \mathbf{P}'' & 0 & 0 & \mathbf{x}'' & \cdots & 0 \\ \vdots & & & & & \vdots \end{bmatrix} \begin{bmatrix} \mathbf{X} \\ \lambda \\ \lambda' \\ \lambda'' \\ \vdots \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \end{bmatrix} \quad (4.8)$$

where, the leftmost matrix, \mathbf{M} , in (4.8) is $3N \times (5+N)$, and the $(5+N) \times (5+N)$ minors \mathbf{Q} of \mathbf{M} constitute multilinear relationships under the extended projective projection as follows:

$$\det \mathbf{Q} = 0 \quad (4.9)$$

We can choose any $5 + N$ rows from \mathbf{M} to constitute \mathbf{Q} , but we have to take at least 2 rows from each camera for deriving meaningful N view relationships (note,

Table 4.1: The number of corresponding points required for computing multifocal tensors in three, four and five views with nonlinear method and linear method .

views	nonlinear mothod	linear mothod
three	9	13
four	8	10
five	8	9

each camera has 3 rows in \mathbf{M}). Thus, the following condition must hold for defining multilinear relationships for N view geometry in the 4D space:

$$5 + N \geq 2N \quad (4.10)$$

Thus, we find that, unlike the traditional multiple view geometry, the multilinear relationship for 5 views is the maximal linear relationship in the 4D space.

We next consider the minimum number of points required for computing the multifocal tensors. The geometric DOF of N extended projective cameras is $14N - 24$, since each extended projective camera has 14 DOF and these N cameras are in a single 4D projective space whose DOF is 24. Meanwhile, if we are given M points in the 4D space, and let these points be projected to N projective cameras defined in (4.6). Then, we derive $2MN$ measurements from images, while we have to compute $14N - 24 + 4M$ components for fixing all the geometry in the 4D space. Thus, the following condition must hold for computing the multifocal tensors from images:

$$2MN \geq 14N - 24 + 4M \quad (4.11)$$

From (4.11), we find that minimum of 9, 8, 8 points are required to compute multifocal tensors in three, four and five views (see Table 4.1).

4.3.1 Three View Geometry

We next introduce the multiple view geometry of three extended projective cameras. For three views, the square submatrix \mathbf{Q} is 8×8 . From $\det \mathbf{Q} = 0$, we have the following trilinear relationship under extended projective camera projections:

$$x^i x^j x^k \epsilon_{krv} \mathcal{T}_{ij}^r = 0_v \quad (4.12)$$

where ϵ_{ijk} denotes a tensor, which represents a sign based on permutation from $\{i,j,k\}$ to $\{1,2,3\}$. \mathcal{T}_{ij}^r is the trifocal tensor for the extended cameras and has the following form:

$$\mathcal{T}_{ij}^r = \epsilon_{ilm} \epsilon_{jqv} \det \begin{bmatrix} \mathbf{a}^l \\ \mathbf{a}^m \\ \mathbf{b}^q \\ \mathbf{b}^u \\ \mathbf{c}^r \end{bmatrix} \quad (4.13)$$

Table 4.2: Trilinear relations between point and line coordinates in three views. The final column denotes the number of linearly independent equations.

correspondence	relation	# of equations
three points	$x^i x'^j x''^k \epsilon_{krv} \mathcal{T}_{ij}^r = 0_v$	2
two points, one line	$x^i x'^j l_r'' \mathcal{T}_{ij}^r = 0$	1
one point, two lines	$x^i l_q'' l_r'' \epsilon^{qjt} \mathcal{T}_{ij}^r = 0^t$	2
three lines	$l_p l_q'' l_r'' \epsilon^{pis} \epsilon^{qjt} \mathcal{T}_{ij}^r = 0^{st}$	4

where \mathbf{a}^i denotes the i th row of \mathbf{P} , \mathbf{b}^i denotes the i th row of \mathbf{P}' and \mathbf{c}^i denotes the i th row of \mathbf{P}'' respectively. The trifocal tensor \mathcal{T}_{ij}^r is $3 \times 3 \times 3$ and has 27 entries. If the extended cameras are projective as shown in (4.6), we have only 26 free parameters in \mathcal{T}_{ij}^r except a scale ambiguity. On the other hand, (4.12) provides us 3 linear equations on \mathcal{T}_{ij}^r , but only 2 of them are linearly independent. Thus, at least 13 corresponding points are required to compute \mathcal{T}_{ij}^r from images linearly.

Why only 2 equations of (4.12) are linearly independent? Let us consider it.

Let the equations in (4.12) be written as $\mathbf{A}\mathbf{t} = 0$ where \mathbf{A} is a 3×27 matrix and \mathbf{t} is a vector containing the entries of \mathcal{T}_{ij}^r . Then the matrix \mathbf{A} may be written as:

$$\mathbf{A}_{(v)(ij)} = x^i x'^j x''^k \epsilon_{krv} \quad (4.14)$$

where (v) indexes the row and $\binom{ij}{r}$ index the column of \mathbf{A} . We may write $x''^k \epsilon_{krv} = S_{rv}$. Then the matrix \mathbf{A} in (4.14) may be written as follows:

$$\mathbf{A}_{(v)(ij)} = x^i x'^j S_{rv}. \quad (4.15)$$

It is known that a 3×3 skew-symmetric matrix has two equal non-zero singular values. Since $x''^k \epsilon_{krv} = -x''^k \epsilon_{kvr}$ we see that S_{rv} is a 3×3 skew-symmetric matrix, and hence has two equal singular values. Therefore, by using the SVD, we have:

$$S_{rv} = U_v^a D_{ab} V_r^b \quad (4.16)$$

with tenor notation. The matrix D_{ab} is diagonal with two equal non-zero diagonal entries. By substituting (4.16) into (4.15), we have

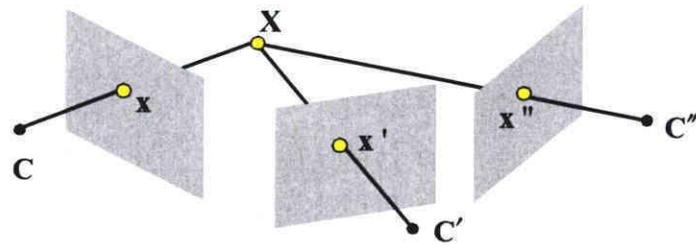
$$\mathbf{A}_{(v)(ij)} = x^i x'^j U_v^a D_{ab} V_r^b. \quad (4.17)$$

Let us consider a 3×27 matrix \mathbf{V}' as follows:

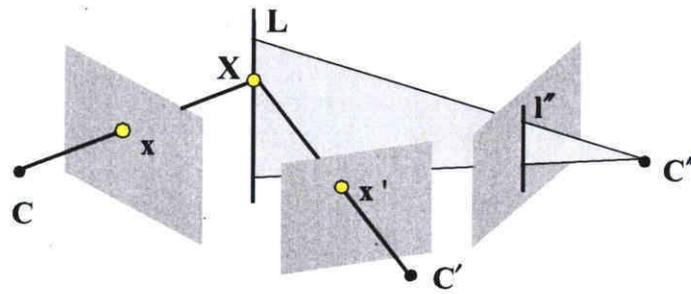
$$V_{(ij)}'^b = x^i x'^j V_r^b. \quad (4.18)$$

Then (4.17) can be written as:

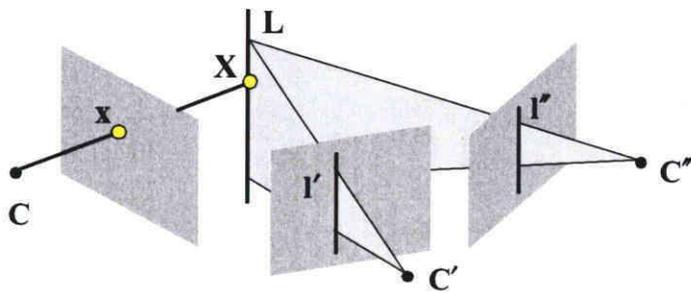
$$\mathbf{A}_{(v)(ij)} = U_v^a D_{ab} V_{(ij)}'^b. \quad (4.19)$$



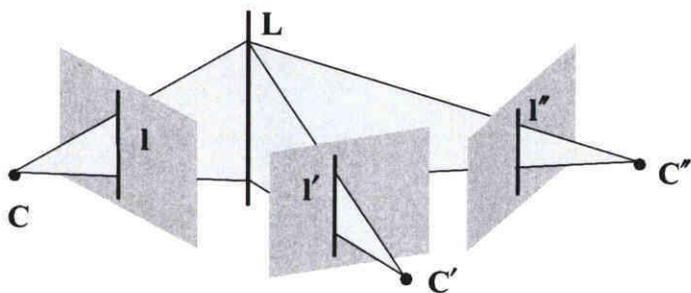
(a) Three points



(b) Two points and one line



(c) One point and two lines



(d) Three lines

Figure 4.2: Trilinear relations among points and lines.

Since V_r^b is orthogonal, $V_{(ij)}^b$ is also orthogonal. Hence, (4.19) is the SVD of the matrix \mathbf{A} . It means that \mathbf{A} has rank 2, and thus, only 2 equations of (4.12) are linearly independent.

Up to now, we considered the trilinear relationship of points in three views. However, we can also consider the trilinear constraints among points and lines. The incidence relations among points and lines are shown in Figure 4.2. A complete set of the trilinear equations among points and lines are given in Table 4.2. All of these equations are linear in the entries of the trifocal tensor \mathcal{T}_{ij}^r .

4.3.2 Four View Geometry

We next introduce the multiple view geometry of four extended projective cameras. The quadrilinear relationship under extended projective projection is:

$$x^i x'^j x''^k x'''^s \epsilon_{jlu} \epsilon_{kmv} \epsilon_{snw} \mathcal{Q}_i^{lmn} = 0_{uvw} \quad (4.20)$$

\mathcal{Q}_i^{lmn} is the quadrifocal tensor whose form is described as:

$$\mathcal{Q}_i^{lmn} = \epsilon_{ipq} \det \begin{bmatrix} \mathbf{a}^p \\ \mathbf{a}^q \\ \mathbf{b}^l \\ \mathbf{c}^m \\ \mathbf{d}^n \end{bmatrix} \quad (4.21)$$

where \mathbf{a}^i , \mathbf{b}^i , \mathbf{c}^i and \mathbf{d}^i denote the i th row of \mathbf{P} , \mathbf{P}' , \mathbf{P}'' and \mathbf{P}''' respectively. The quadrifocal tensor \mathcal{Q}_i^{lmn} has 81 entries. Excluding a scale ambiguity, it has 80 free parameters. Similar arguments in the three-view case hold here, and we can see that 27 linear equations are derived from (4.20) but only 8 of them are linearly independent. Therefore, minimum of 10 corresponding points are required to compute \mathcal{Q}_i^{lmn} from images linearly.

The quadrilinear relationships involving the quadrifocal tensor are summarized in Table 4.3.

4.3.3 Five View Geometry

Similarly, the five view geometry can also be derived for the extended projective cameras. The quintilinear constraint is expressed as follows:

$$x^i x'^j x''^k x'''^s x''''^t \epsilon_{ila} \epsilon_{jmb} \epsilon_{knc} \epsilon_{sfd} \epsilon_{tge} \mathcal{R}^{lmnfg} = 0_{abcde} \quad (4.22)$$

where \mathcal{R}^{lmnfg} is the quintifocal tensor (five view tensor) whose form is represented as follows:

$$\mathcal{R}^{lmnfg} = \det \begin{bmatrix} \mathbf{a}^l \\ \mathbf{b}^m \\ \mathbf{c}^n \\ \mathbf{d}^f \\ \mathbf{e}^g \end{bmatrix} \quad (4.23)$$

Table 4.3: Quadrilinear relations between point and line coordinates in four views. The final column denotes the number of linearly independent equations.

correspondence	relation	# of equations
four points	$x^i x'^j x''^k x'''^s \epsilon_{jlu} \epsilon_{kmv} \epsilon_{snw} Q_i^{lmn} = 0_{uvw}$	8
three points, one line	$x^i x'^j x''^k l_n''' \epsilon_{jlu} \epsilon_{kmv} Q_i^{lmn} = 0_{uv}$	4
two points, two lines	$x^i x'^j l_m'' l_n''' \epsilon_{jlu} Q_i^{lmn} = 0_u$	2
one point, three lines	$x^i l_l'' l_m''' l_n'''' Q_i^{lmn} = 0$	1
four lines	$l_k l_l'' l_m''' l_n'''' \epsilon^{kiw} Q_i^{lmn} = 0^w$	2

Table 4.4: Quintilinear relations between point and line coordinates in five views. The final column denotes the number of linearly independent equations.

correspondence	relation	eq.
five points	$x^i x'^j x''^k x'''^s x''''^t \epsilon_{ila} \epsilon_{jmb} \epsilon_{knc} \epsilon_{sfd} \epsilon_{tge} \mathcal{R}^{lmnfg} = 0_{abcde}$	32
four points, one line	$x^i x'^j x''^k x'''^s l_g'''' \epsilon_{ila} \epsilon_{jmb} \epsilon_{knc} \epsilon_{sfd} \mathcal{R}^{lmnfg} = 0_{abcd}$	16
three points, two lines	$x^i x'^j x''^k l_f''' l_g'''' \epsilon_{ila} \epsilon_{jmb} \epsilon_{knc} \mathcal{R}^{lmnfg} = 0_{abc}$	8
two points, three lines	$x^i x'^j l_n'' l_f''' l_g'''' \epsilon_{ila} \epsilon_{jmb} \mathcal{R}^{lmnfg} = 0_{ab}$	4
one point, four lines	$x^i l'_m l_n'' l_f''' l_g'''' \epsilon_{ila} \mathcal{R}^{lmnfg} = 0_a$	2
five lines	$l_l' l_m'' l_n''' l_f'''' l_g'''' \mathcal{R}^{lmnfg} = 0$	1

where \mathbf{a}^i , \mathbf{b}^i , \mathbf{c}^i , \mathbf{d}^i and \mathbf{e}^i denote the i th row of \mathbf{P} , \mathbf{P}' , \mathbf{P}'' , \mathbf{P}''' and \mathbf{P}'''' respectively. The quintifocal tensor \mathcal{R}^{lmnfg} has 243 entries. If the extended cameras are projective as shown in (4.6), we have only 242 free parameters in \mathcal{R}^{lmnfg} except a scale. On the other hand, (4.22) provides us 243 linear equations on \mathcal{R}^{lmnfg} , but only 32 of them are linearly independent, that can be proved in the same manner with the three-view case. However, it turns out that there exists a linear dependency between the 64 constraints obtained for two different corresponding points. Therefore, the set of equations (4.22) derived from a set of N general point correspondences across five views has rank $32N - \mathcal{N}C_2$. Thus for 8 points there are only 228 independent equations, which are not enough to solve for \mathcal{R}^{lmnfg} . For $N = 9$ points, the rank is $32N - \mathcal{N}C_2 = 252$, and we have enough equations to solve for the 243 entries of \mathcal{R}^{lmnfg} linearly.

The number of corresponding points required for computing multifocal tensors is summarized in Table 4.1. The quintilinear relationships are given in Table 4.4.

4.4 Intersections of Five Hyperplanes

The multiple view tensors may be given a different derivation, that is the basic geometric property is the intersection of five hyperplanes.

We start with the quintifocal tensor for which the derivation is easiest. Consider five lines l, l', l'', l''' and l'''' in images formed from five cameras with camera matrices $\mathbf{P}, \mathbf{P}', \mathbf{P}'', \mathbf{P}'''$ and \mathbf{P}'''' . The back projection of a line \mathbf{l} through camera \mathbf{P} is written as the hyperplane $l_i \mathbf{a}^i$, where \mathbf{a}^i denotes the i th row of \mathbf{P} . The condition that these five hyperplanes are coincident may be written as

$$(l_i \mathbf{a}^i) \wedge (l'_m \mathbf{b}^m) \wedge (l''_n \mathbf{c}^n) \wedge (l'''_f \mathbf{d}^f) \wedge (l''''_g \mathbf{e}^g) = 0. \quad (4.24)$$

where $\mathbf{b}^i, \mathbf{c}^i, \mathbf{d}^i$ and \mathbf{e}^i are i th row of $\mathbf{P}', \mathbf{P}'', \mathbf{P}'''$ and \mathbf{P}'''' . However, since the determinant is linear in each row, (4.24) may be written as:

$$0 = l_l l'_m l''_n l'''_f l''''_g (\mathbf{a}^l \wedge \mathbf{b}^m \wedge \mathbf{c}^n \wedge \mathbf{d}^f \wedge \mathbf{e}^g) \stackrel{\text{def}}{=} l_l l'_m l''_n l'''_f l''''_g \mathcal{R}^{lmnfg} \quad (4.25)$$

This corresponds to the definition (4.23) and five lines relation for the quintifocal tensor. The basic geometric property is the intersection of the five hyperplanes in 4D space.

Quadrifocal tensor derivation. Consider now a point-line-line-line relationship $x^i \leftrightarrow l'_i \leftrightarrow l''_m \leftrightarrow l'''_n$, for four views and let l^1_s, l^2_t be two lines in the first image that pass through the image point \mathbf{x} . The hyperplanes back-projected from the five lines meet in a point in 4D space. So we can write:

$$l^1_s l^2_t l'_i l''_m l'''_n (\mathbf{a}^s \wedge \mathbf{a}^t \wedge \mathbf{b}^i \wedge \mathbf{c}^m \wedge \mathbf{d}^n) = 0. \quad (4.26)$$

Next, multiply this equation by a scalar $\epsilon^{ist} \epsilon_{ist}$. The result after regrouping is

$$(l^1_s l^2_t \epsilon^{ist}) l'_i l''_m l'''_n \epsilon_{ist} (\mathbf{a}^s \wedge \mathbf{a}^t \wedge \mathbf{b}^i \wedge \mathbf{c}^m \wedge \mathbf{d}^n) = 0. \quad (4.27)$$

Now the expression $l^1_s l^2_t \epsilon^{ist}$ is simply the cross-product of the two lines \mathbf{l}^1 and \mathbf{l}^2 , in other words their intersection point \mathbf{x} . Thus finally we can write

$$0 = x^i l'_i l''_m l'''_n (\epsilon_{ist} (\mathbf{a}^s \wedge \mathbf{a}^t \wedge \mathbf{b}^i \wedge \mathbf{c}^m \wedge \mathbf{d}^n)) \stackrel{\text{def}}{=} x^i l'_i l''_m l'''_n \mathcal{Q}_i^{lmn} \quad (4.28)$$

which are the definition (4.21) and a point-line-line-line relationship for the quadrifocal tensor.

Trifocal tensor derivation. We can derive the trifocal tensor in the same manner. Given a correspondence $\mathbf{x} \leftrightarrow \mathbf{x}' \leftrightarrow \mathbf{l}''$ select pairs of lines l^1_p and l^2_q passing through \mathbf{x} , and l^1_r and l^2_s passing through \mathbf{x}' . The back-projected planes all meet in a point, so we write

$$l^1_s l^2_t l^1_u l^2_v l''_r (\mathbf{a}^s \wedge \mathbf{a}^t \wedge \mathbf{b}^u \wedge \mathbf{b}^v \wedge \mathbf{c}^r) = 0. \quad (4.29)$$

Multiplying by $(\epsilon^{ist} \epsilon_{ist})(\epsilon^{juv} \epsilon_{juv})$ and proceeding as before leads to the coplanarity constraint

$$0 = x^i x'^j l''_r (\epsilon^{ist} \epsilon^{juv} (\mathbf{a}^s \wedge \mathbf{a}^t \wedge \mathbf{b}^u \wedge \mathbf{b}^v \wedge \mathbf{c}^r)) \stackrel{\text{def}}{=} x^i x'^j l''_r \mathcal{T}_{ij}^r \quad (4.30)$$

which can be compared with (4.13).

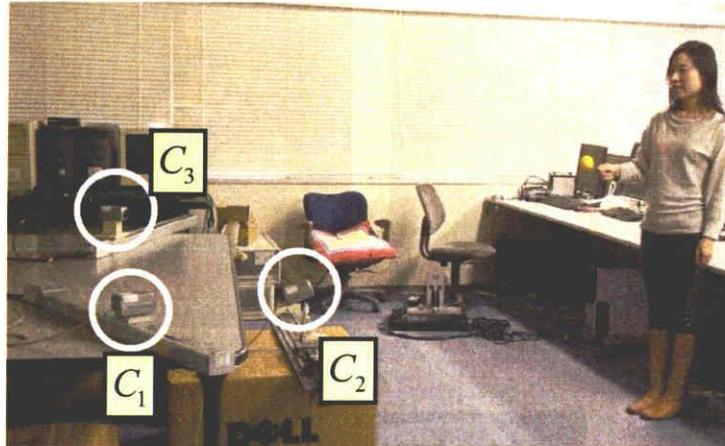


Figure 4.3: Experimental circumstance. Three cameras are fixed on three single axis robots respectively to capture the 3D motions.

4.5 Experiments

We next show the results of experiments. We first show the results from real images that the trifocal tensor for extended projective cameras can be computed from image motions viewed from arbitrary translational cameras, and can be used for generating the third view from the first view and the second view of moving cameras. We next evaluate the stability of extracted trifocal tensors for extended projective cameras.

4.5.1 Real Image Experiment

In this section, we show the results from single point motion and multiple point motion experiments.

The experimental circumstance is shown as Figure 4.3. Three cameras (Sony DFW-VL 500) are fixed on three single axis robots (Oriental Motor Electric Actuator EZS6) respectively to control the 3D motions of cameras. These three cameras are translating with different constant speed and different direction.

In the first experiment, we used three cameras to compute trifocal tensors between these cameras by using a single moving point in the 3D space. Since multiple cameras are dynamic, we can not compute the traditional trifocal tensor of these cameras from a single moving point. Nonetheless we can compute the extended trifocal tensor and can generate image motions in one of three views from the other two views. In this experiment we generated image motions in camera 3 by using image motions in camera 1 and camera 2. Figure 4.4 (a), (b) and (c) show image motions of a single moving point in translational camera 1, camera 2 and camera 3 respectively. The trifocal tensor is computed from 13 points on the image motions in three views. These are shown by green points in (a), (b) and (c). The extracted trifocal tensor is used for

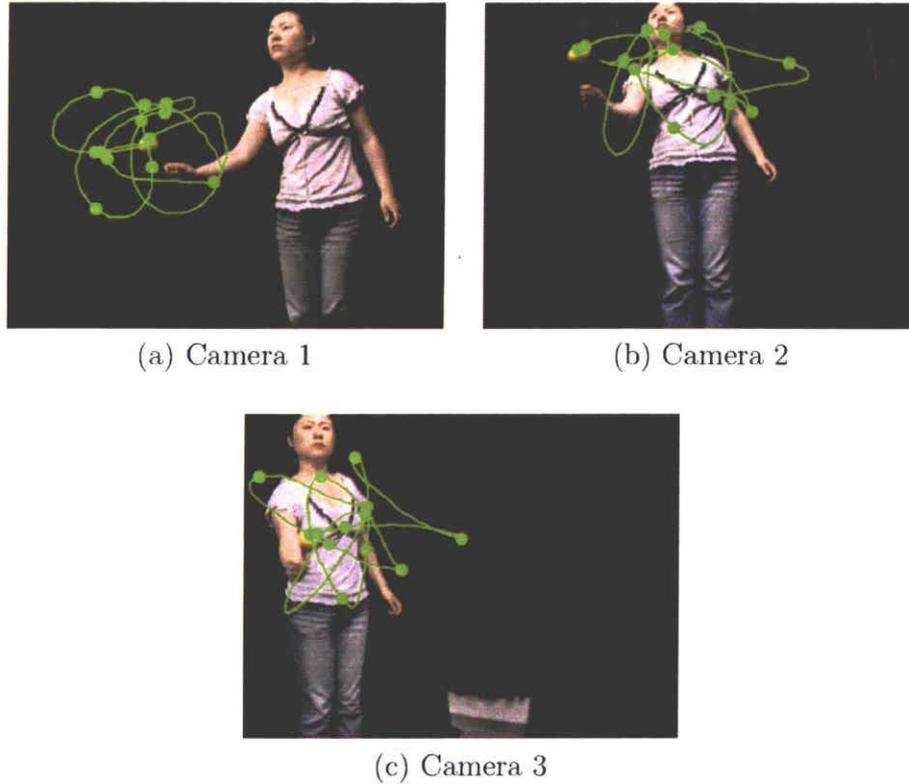
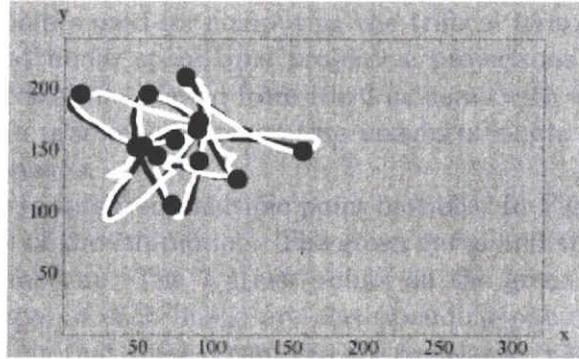
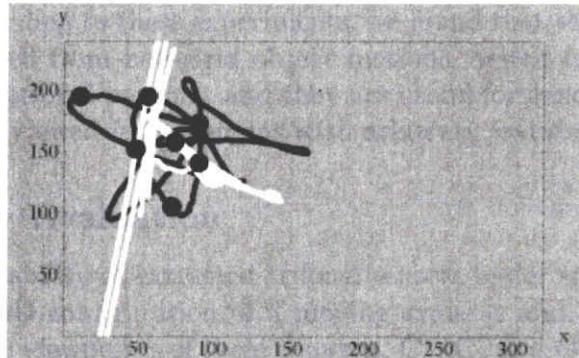


Figure 4.4: Single point motion experiment. (a), (b) and (c) show image motions of a single point viewed from camera 1, 2 and 3. The 13 green points in each image are corresponding points used for computing the trifocal tensor. Note that these 3 cameras are translating with different constant speed and different direction.

generating image motions in camera 3 from image motions in camera 1 and 2. The white curve in Figure 4.5 (a) shows image motions in camera 3 generated from the extended trifocal tensor, and the black curve shows the real image motions viewed from camera 3. As shown in Figure 4.5 (a), the generated image motions almost recovered the original complex image motions even if these 3 cameras have unknown translational motions. To show the advantage of the extended trifocal tensor, we also show image motions generated from the traditional trifocal tensor, that is, trifocal tensor defined for projections from 3D space to 2D space. 7 points taken from the former 13 points are used as corresponding points in three views for computing the traditional projective trifocal tensor. The image motion in camera 3 generated from the image motions in camera 1 and 2 by using the extracted traditional trifocal tensor is shown by white curve in Figure 4.5 (b). As shown in Figure 4.5 (b), the generated image motion is very different from the real image motion shown by black curve as we expected, and



(a) Image motions recovered from the extended trifocal tensor



(b) Image motions recovered from the traditional trifocal tensor

Figure 4.5: Image motion in camera 3 recovered from the extended trifocal tensor and the traditional trifocal tensor. The white curve in (a) shows image motions recovered from the extended trifocal tensor, and the black curve shows real image motions observed in camera 3. (b) shows those recovered from the traditional trifocal tensor. The 13 black points in (a) and 7 black points in (b) show points used for computing the trifocal tensors.

thus we find that the traditional multiple view geometry cannot describe such general situations, while the proposed multiple view geometry can as shown in Figure 4.5 (a).

The results from other 3 single point motions are also given. In Figure 4.6, (a*i*), (b*i*) and (c*i*) show three views of the *i*th motion. The 13 green points in each image are corresponding points used for computing the trifocal tensor. Note that these 3 cameras are translating with different speed and different direction. The white curve in (d*i*) shows image motions recovered from the extended trifocal tensor in camera 3, and the black curve shows real image motions observed in camera 3. The 13 black

points in (di) show points used for computing the trifocal tensor. As we can see, the trifocal tensor defined under space-time projective projections can be derived from arbitrary single point motions viewed from the 3 cameras with arbitrary translational motions, and they are practical for generating images of single point motions viewed from translational cameras.

Next we show the results from multiple point motions. In Figure 4.7, (ai), (bi) and (ci) show three views of the i th motion. The green curve and the red curve represent two different image motion. The 7 green points on the green curve and the 6 red points on the red curve in each image are corresponding points used for computing the trifocal tensor. Note that these 3 cameras are translating with different speed and different direction. The white curve in (di) shows image motions recovered from the extended trifocal tensor in camera 3, and the black curve shows real image motions observed in camera 3. The 13 black points in (di) show points used for computing the trifocal tensor. According to these experiments, we found that the extended multifocal tensors can be derived from non-rigid object motions viewed from multiple cameras with arbitrary translational motions, and they are useful for generating images of non-rigid object motions viewed from cameras with arbitrary translational motions.

4.5.2 Stability Evaluation

We next show the stability of extracted trifocal tensors under space-time projections. Figure 4.8 shows a 3D configuration of 3 moving cameras and a moving point. The black points show the viewpoints of three cameras, \mathbf{C}_1 , \mathbf{C}_2 and \mathbf{C}_3 , before translational motions, and the white points show their viewpoints after the translational motions. The translational motions of these three cameras are different and unknown. The black curve shows a locus of a freely moving point. For evaluating the extracted trifocal tensors, we computed reprojection errors derived from the trifocal tensors. The reprojection error is defined as follows:

$$\frac{1}{N} \sum_{i=1}^N d(\mathbf{m}_i, \hat{\mathbf{m}}_i)^2 \quad (4.31)$$

where $d(\mathbf{m}_i, \hat{\mathbf{m}}_i)$ denotes a distance between a true point \mathbf{m}_i and a point $\hat{\mathbf{m}}_i$ recovered from the trifocal tensor.

We increased the number of corresponding points used for computing trifocal tensors in three views from 13 to 25, and evaluated the reprojection errors. Gaussian noise of standard deviation of 1 pixel is added to each image. Figure 4.9 shows the relationship between the number of corresponding points and the reprojection errors. As we can see, the stability is obviously improved by using a few more points than the minimum number of corresponding points.

4.6 Summary

In this chapter, we analyzed multiple view geometry under projective projections from 4D space to 2D space, and showed that it can represent multiple view geometry under space-time projections. In particular, we showed that multifocal tensors defined under space-time projective projections can be computed from non-rigid object motions viewed from multiple cameras with arbitrary translational motions. We also showed that they are very useful for generating images of non-rigid motions viewed from projective cameras with arbitrary translational motions. The method was implemented and tested by using real image sequences. The stability of extracted trifocal tensors was also evaluated.

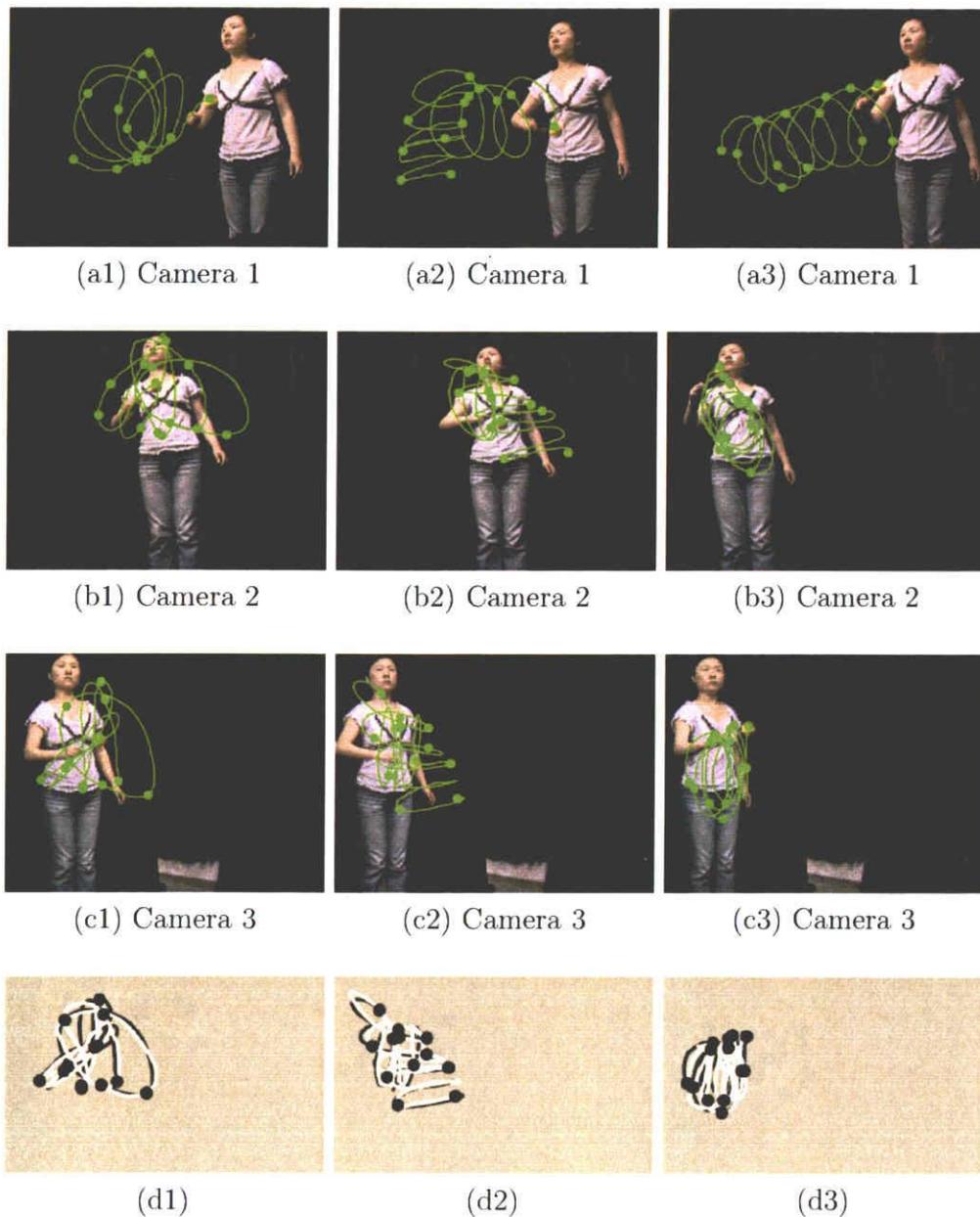


Figure 4.6: Other 3 single point motion experiments. (a_i) , (b_i) and (c_i) show three views of the i th motion. The 13 green points in each image are corresponding points used for computing the trifocal tensor. Note that these 3 cameras are translating with different speed and different direction. The white curve in (d_i) shows image motions recovered from the extended trifocal tensor, and the black curve shows real image motions observed in camera 3. The 13 black points in (d_i) show points used for computing the trifocal tensor.

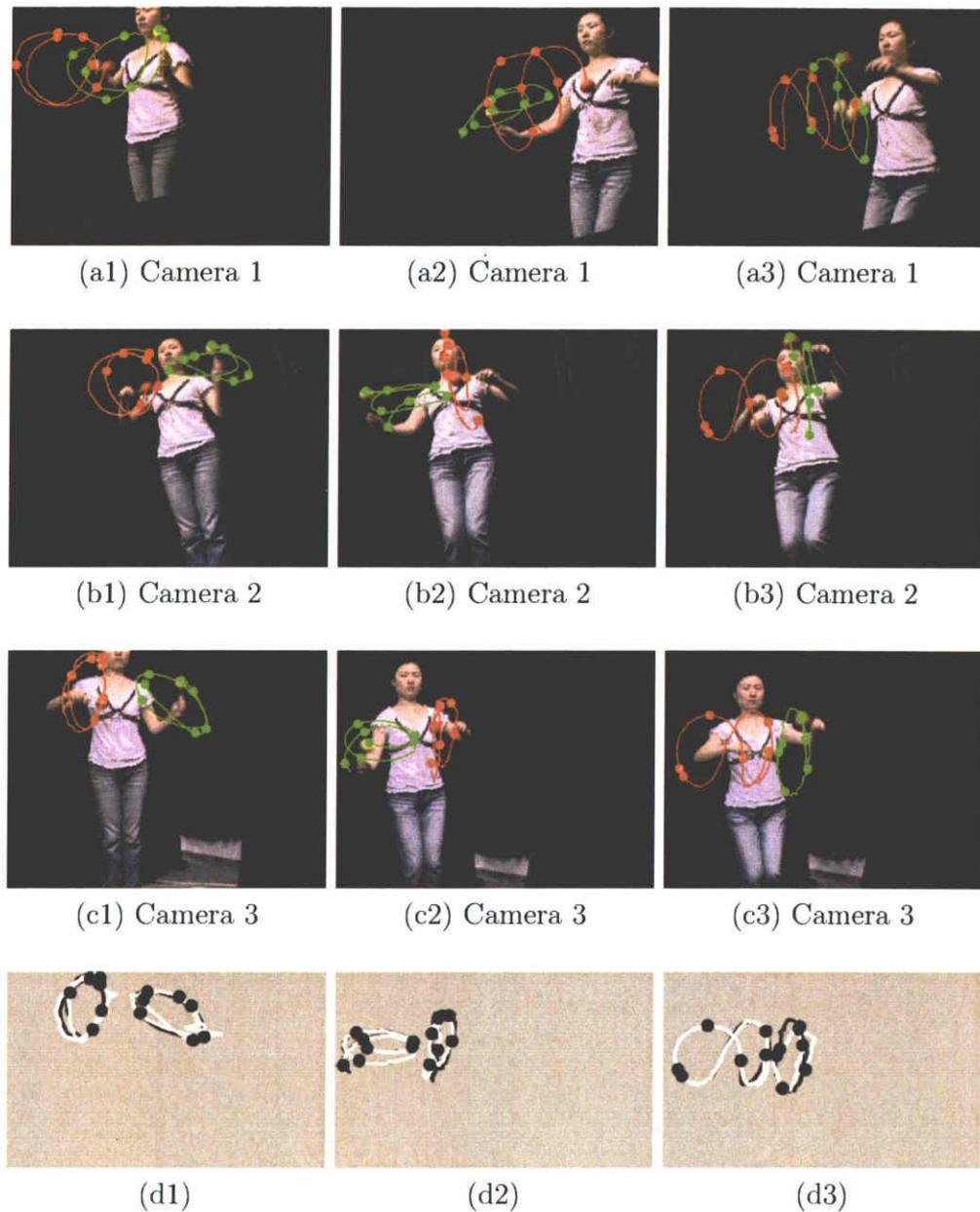


Figure 4.7: Multiple point motion experiments. (a_i) , (b_i) and (c_i) show three views of the i th motion. The green curve and the red curve represent two different image motion. The 7 green points on the green curve and the 6 red points on the red curve in each image are corresponding points used for computing the trifocal tensor. Note that these 3 cameras are translating with different speed and different direction. The white curve in (d_i) shows image motions recovered from the extended trifocal tensor, and the black curve shows real image motions observed in camera 3. The 13 black points in (d_i) show points used for computing the trifocal tensor.

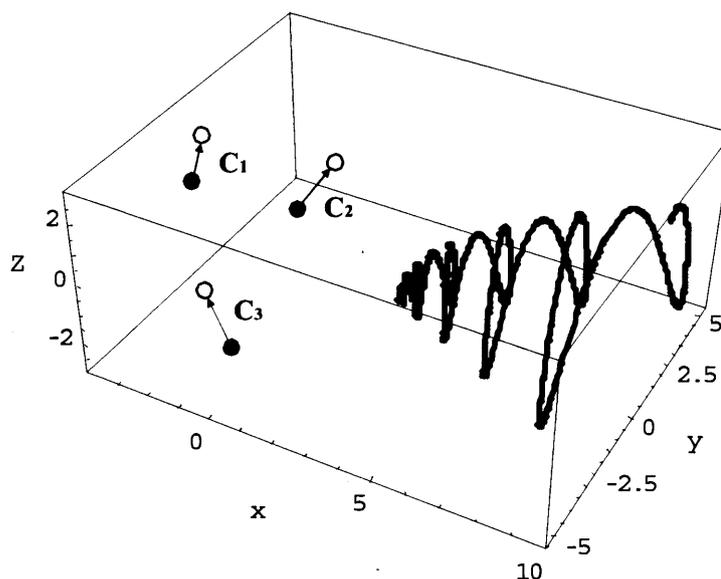


Figure 4.8: Three translating cameras and a moving point in the 3D space. The black points show the viewpoints of three cameras before translational motions, and the white points show those after the translational motions.

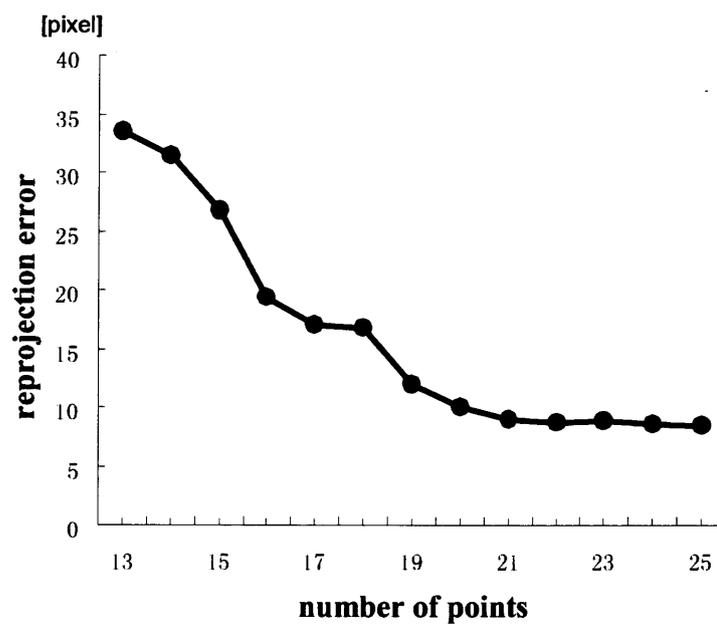


Figure 4.9: The relationship between the number of corresponding points used for computing trifocal tensors and the reprojection errors.

Chapter 5

Dynamic Multiple View Geometry in 6D Space

5.1 Dynamic Multiple View Geometry for Multiple Curvilinear Motion Cameras

In chapter 4, we considered the multiple view geometry of multiple cameras with translational motions. In this chapter, we extend the theory discussed in chapter 4, and introduces a multiple view geometry of multiple cameras with arbitrary curvilinear motions. The curvilinear motion means curved motion without rotation. The multiple view geometry analyzed in this chapter enables us to define multilinear relationship among image points derived from non-rigid object motions viewed from multiple cameras with arbitrary curvilinear motions as shown in Figure 5.1. We show the new multilinear relationship is useful for generating arbitrary view images and reconstructing 3D non-rigid object motions viewed from cameras with arbitrary curvilinear motions. The method is tested in real image sequences.

In this chapter, we derive multiple view geometry under two different camera models. One is an affine camera model and the other is a projective model. In Section 5.2, we show multiple view geometry of affine cameras, and in Section 5.3, we introduce the multiple view geometry of projective cameras.

5.2 Multiple View Geometry for Affine Curvilinear Motion Cameras

Let us consider a single moving point in the 3D space. If the multiple cameras are stationary or translational, we can compute the multifocal tensors with the methods proposed in chapter 2 and chapter 3 to figure out the multiple view geometry. However, if these cameras have independent curvilinear motions, the mentioned multifocal tensors cannot be computed from the image motion of the point. Therefore, we in this

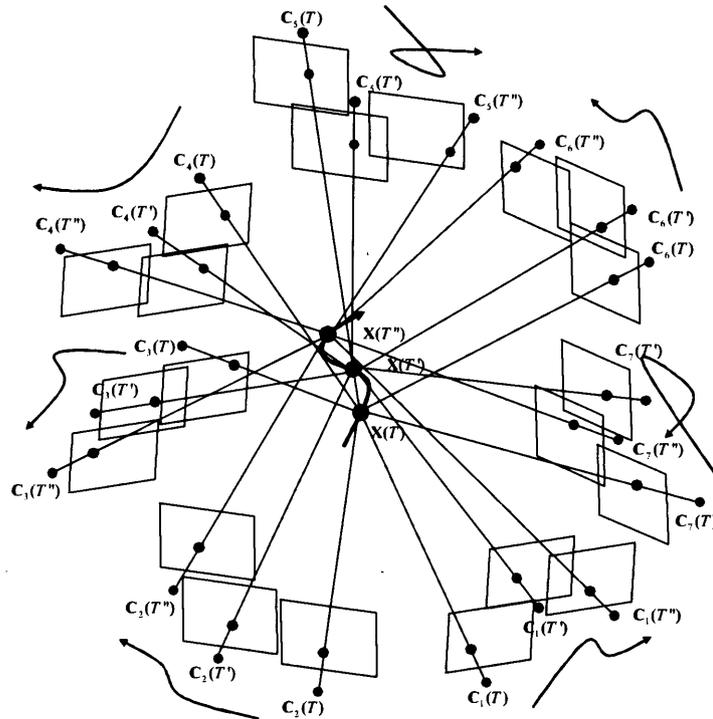


Figure 5.1: A moving point in 3D space and its projections in seven curvilinear motion cameras. The multifocal tensor defined under projections from P^6 to P^2 can describe the relationship among these image projections.

section show that if the camera motions are curvilinear as shown in Figure 5.1, the multiple view geometry under extended projections can be computed from the image motion of the point, and they can be used to, for example, generate image motions viewed from arbitrary curvilinear motion cameras.

Consider a usual affine camera which projects points in 3D to 2D images. The motions of a point in the 3D space can be represented by homogeneous coordinates, $\mathbf{X}(T) = [X(T), Y(T), Z(T), 1]^T$, where T denotes time. The motions are projected to images, and can be observed as a set of points, $\mathbf{x}(T) = [x(T), y(T), 1]^T$. Here, we make use of cubic Bezier curves to describe the arbitrary 3D motions of cameras $\Delta\mathbf{X} = [\Delta X, \Delta Y, \Delta Z, \Delta W]^T$ in homogeneous coordinates in this paper. The camera motion is relative to the camera initial position, and hence its fourth entry is equal to 0, and thus it is represented as $\Delta\mathbf{X} = [\Delta X, \Delta Y, \Delta Z, 0]^T$. The cubic Bezier curve is

defined as written in (3.6):

$$\Delta \mathbf{X} = \mathbf{GAE} \begin{bmatrix} T^3 \\ T^2 \\ T \\ 1 \end{bmatrix} \quad (5.1)$$

where,

$$\mathbf{G} = [\mathbf{G}_0 \quad \mathbf{G}_1 \quad \mathbf{G}_2 \quad \mathbf{G}_3],$$

$$\mathbf{A} = \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \quad (5.2)$$

$$\mathbf{E} = \text{diag}[\frac{1}{T_a^3}, \frac{1}{T_a^2}, \frac{1}{T_a}, 1]. \quad (5.3)$$

$\mathbf{G}_0, \mathbf{G}_1, \mathbf{G}_2, \mathbf{G}_3$ denote four control points and T_a is the total time of camera's motion. Thus, point motions are projected to affine camera as follows:

$$\begin{aligned} \begin{bmatrix} x(T) \\ y(T) \\ 1 \end{bmatrix} &= \mathbf{P}_a(\mathbf{X}(T) - \Delta \mathbf{X}_i) \\ &= \mathbf{P}_a(\mathbf{X}(T) - \mathbf{GAE} \begin{bmatrix} T^3 \\ T^2 \\ T \\ 1 \end{bmatrix}) \\ &= \mathbf{P}_a[\mathbf{I}, -\mathbf{GAE}] \begin{bmatrix} \mathbf{X}(T) \\ T^3 \\ T^2 \\ T \\ 1 \end{bmatrix} \\ &= \mathbf{P}_a[\mathbf{I}, -\mathbf{GAE}] \begin{bmatrix} X(T) \\ Y(T) \\ Z(T) \\ 1 \\ T^3 \\ T^2 \\ T \\ 1 \end{bmatrix} \end{aligned}$$

$$\begin{aligned}
&= \mathbf{P}_b \begin{bmatrix} X(T) \\ Y(T) \\ Z(T) \\ 1 \\ T^3 \\ T^2 \\ T \\ 1 \end{bmatrix} \\
&= \mathbf{P} \begin{bmatrix} X(T) \\ Y(T) \\ Z(T) \\ T^3 \\ T^2 \\ T \\ 1 \end{bmatrix}
\end{aligned} \tag{5.4}$$

where \mathbf{P}_a denotes a 3×4 affine camera matrix, whose third row is $[0,0,0,1]$, and \mathbf{P}_b represents a 3×8 matrix. In (5.4), \mathbf{P} denotes a 3×7 extended affine camera matrix, whose third row is $[0,0,0,0,0,0,1]$. The 7th column of \mathbf{P} is derived by merging the 4th column and the 8th column of \mathbf{P}_b . We therefore find that, from (5.4), the projections of point motions to multiple cameras with curvilinear motions can be described by the multilinear relationship under the projection from 6D to 2D. In the next sections, the geometry of such projections will be given in more detail.

5.2.1 Projection from 6D to 2D

We first consider a projection from 6D space to 2D space. Let $\mathbf{X} = [X^1, X^2, X^3, X^4, X^5, X^6, X^7]^\top$ be the homogeneous coordinates of a 6D space point projected to a point in the 2D space, whose homogeneous coordinates are represented by $\mathbf{x} = [x^1, x^2, x^3]^\top$. Then, the extended affine projection from \mathbf{X} to \mathbf{x} can be described as follows:

$$\mathbf{x} \sim \mathbf{P}\mathbf{X} \tag{5.5}$$

where (\sim) denotes equality up to a scale, and \mathbf{P} denotes the following 3×7 matrix:

$$\mathbf{P} = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} & p_{15} & p_{16} & p_{17} \\ p_{21} & p_{22} & p_{23} & p_{24} & p_{25} & p_{26} & p_{27} \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{5.6}$$

From (5.5), we find that the extended affine camera, \mathbf{P} , has 14 DOF. In the next section, we consider the multiple view geometry of the extended affine cameras.

Table 5.1: Quadilinear relations between point and line coordinates in four views. The final column denotes the number of linearly independent equations.

correspondence	relation	eq.
four points	$x^i x'^j x''^k x'''^h \epsilon_{hvd} Q_{ijk}^v = 0_d$	2
three points, one line	$x^i x'^j x''^k l_v''' Q_{ijk}^v = 0$	1
two points, two lines	$x^i x'^j l_r''' l_v''' \epsilon^{rku} Q_{ijk}^v = 0^u$	2
one point, three lines	$x^i l_q''' l_r''' l_v''' \epsilon^{qjt} \epsilon^{rku} Q_{ijk}^v = 0^{tu}$	4
four lines	$l_p l_q''' l_r''' l_v''' \epsilon^{pis} \epsilon^{qkt} \epsilon^{rku} Q_{ijk}^v = 0^{stu}$	8

5.2.2 Affine Multiple View Geometry from 6D to 2D

From (5.5), we have the following equation for N extended affine cameras:

$$\begin{bmatrix} \mathbf{P} & \mathbf{x} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{P}' & \mathbf{0} & \mathbf{x}' & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{P}'' & \mathbf{0} & \mathbf{0} & \mathbf{x}'' & \cdots & \mathbf{0} \\ \vdots & & & & & \vdots \end{bmatrix} \begin{bmatrix} \mathbf{X} \\ \lambda \\ \lambda' \\ \lambda'' \\ \vdots \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \vdots \end{bmatrix} \quad (5.7)$$

where, the leftmost matrix, \mathbf{M} , in (5.7) is $3N \times (7 + N)$, and the $(7 + N) \times (7 + N)$ minors \mathbf{Q} of \mathbf{M} constitute multilinear relationships under the extended projection as: $\det \mathbf{Q} = 0$. We can choose any $7 + N$ rows from \mathbf{M} to constitute \mathbf{Q} , but we have to take at least 2 rows from each camera for deriving meaningful N view relationships (note, each camera has 3 rows in \mathbf{M}). Thus, $7 + N \geq 2N$ must hold for defining multilinear relationships for N view geometry in the 6D space. Thus, we find that, the multilinear relationship for 7 views is the maximal linear relationship in the 6D space.

We next consider the minimum number of points required for computing the multifocal tensors. The geometric DOF of N extended affine cameras is $14N - 42$, since each extended affine camera has 14 DOF and these N cameras are in a single 6D affine space whose DOF is 42. Meanwhile, if we are given M points in the 6D space, and let them be projected to N cameras defined in (5.5). Then, we derive $2MN$ measurements from images, while we have to compute $14N - 42 + 6M$ components for fixing all the geometry in the 6D space. Thus, the following condition must hold for computing the multifocal tensors from images: $2MN \geq 14N - 42 + 6M$. We find that minimum of 7 points are required to compute multifocal tensors in four, five, six and seven views.

5.2.2.1 Four View Affine Geometry

We next introduce the multiple view geometry of four extended cameras. For four views, the sub square matrix \mathbf{Q} is 11×11 . From $\det \mathbf{Q} = 0$, we have the following

quadrilinear relationship under extended camera projections:

$$x^i x'^j x''^k x'''^h \epsilon_{hvd} \mathcal{Q}_{ijk}^v = 0_d \quad (5.8)$$

where ϵ_{hvd} (or its contravariant counterpart, ϵ^{hvd}) denotes a tensor, which represents a sign based on permutation from $\{h,v,d\}$ to $\{1,2,3\}$. \mathcal{Q}_{ijk}^v is the quadrifocal tensor for the extended cameras and has the following form:

$$\mathcal{Q}_{ijk}^v = \epsilon_{ipq} \epsilon_{jrs} \epsilon_{ktu} \det \begin{bmatrix} \mathbf{a}^p \\ \mathbf{a}^q \\ \mathbf{b}^r \\ \mathbf{b}^s \\ \mathbf{c}^t \\ \mathbf{c}^u \\ \mathbf{d}^v \end{bmatrix} \quad (5.9)$$

where \mathbf{a}^i denotes the i th row of \mathbf{P} , \mathbf{b}^i denotes the i th row of \mathbf{P}' , \mathbf{c}^i denotes the i th row of \mathbf{P}'' and \mathbf{d}^i denotes the i th row of \mathbf{P}''' respectively. The quadrifocal tensor \mathcal{Q}_{ijk}^v is $3 \times 3 \times 3 \times 3$ and has 81 entries. Since all the third rows of the extended affine camera matrices are $[0,0,0,0,0,1]$, many zero entries arise in \mathcal{Q}_{ijk}^v . As a result, $\mathcal{Q}_{133}^1, \mathcal{Q}_{133}^2, \mathcal{Q}_{233}^1, \mathcal{Q}_{233}^2, \mathcal{Q}_{313}^1, \mathcal{Q}_{313}^2, \mathcal{Q}_{323}^1, \mathcal{Q}_{323}^2, \mathcal{Q}_{331}^1, \mathcal{Q}_{331}^2, \mathcal{Q}_{332}^1, \mathcal{Q}_{332}^2, \mathcal{Q}_{333}^1, \mathcal{Q}_{333}^2, \mathcal{Q}_{333}^3$ are non-zero entries and thus we have only 14 free parameters in \mathcal{Q}_{ijk}^v except a scale ambiguity. On the other hand, (5.8) provides us 3 linear equations on \mathcal{Q}_{ijk}^v , but only 2 of them are linearly independent. Thus, at least 7 corresponding points are required to compute \mathcal{Q}_{ijk}^v from images linearly.

Since corresponding points with time marks induce linear constraints, for computing quadrifocal tensor, we reformulate (5.8) as follows:

$$\mathbf{E}(t)\mathbf{q} = \mathbf{0} \quad (5.10)$$

where $\mathbf{q} = [\mathcal{Q}_{133}^1, \mathcal{Q}_{133}^2, \mathcal{Q}_{233}^1, \mathcal{Q}_{233}^2, \mathcal{Q}_{313}^1, \mathcal{Q}_{313}^2, \mathcal{Q}_{323}^1, \mathcal{Q}_{323}^2, \mathcal{Q}_{331}^1, \mathcal{Q}_{331}^2, \mathcal{Q}_{332}^1, \mathcal{Q}_{332}^2, \mathcal{Q}_{333}^1, \mathcal{Q}_{333}^2, \mathcal{Q}_{333}^3]^\top$, and $\mathbf{E}(t)$ is a 3×15 matrix whose elements are calculated from the corresponding points $\mathbf{x}(t)$, $\mathbf{x}'(t)$, $\mathbf{x}''(t)$ and $\mathbf{x}'''(t)$. Then, if we have N corresponding points, \mathbf{q} can be computed by solving the following linear equations.

$$\begin{aligned} \mathbf{U}\mathbf{q} &= \mathbf{0} \\ \mathbf{U} &= [\mathbf{E}(t_1)^\top, \dots, \mathbf{E}(t_N)^\top]^\top \end{aligned} \quad (5.11)$$

where $N \geq 7$. The solution on \mathbf{q} is the eigenvector corresponding to the smallest eigenvalue of $\mathbf{U}^\top \mathbf{U}$.

Since two points x'''^d and x'''^h in the forth view can be used to represent a line l_v''' which goes through x'''^d and x'''^h as: $x'''^h x'''^d \epsilon_{hvd} = l_v'''$, (5.8) becomes

$$x^i x'^j x''^k x'''^h x'''^d \epsilon_{hvd} \mathcal{Q}_{ijk}^v = x^i x'^j x''^k l_v''' \mathcal{Q}_{ijk}^v = 0 \quad (5.12)$$

Table 5.2: Quintilinear relations between point and line coordinates in five views. The final column denotes the number of linearly independent equations.

relation	# of eq.
$x^i x'^j x''^k x'''^h x''''^m \epsilon_{ktc} \epsilon_{hud} \epsilon_{mve} \mathcal{R}_{ij}^{tuv} = 0_{cde}$	8
$x^i x'^j x''^k x'''^h l_v'''' \epsilon_{ktc} \epsilon_{hud} \mathcal{R}_{ij}^{tuv} = 0_{cd}$	4
$x^i x'^j x''^k l_u'''' l_v'''' \epsilon_{ktc} \mathcal{R}_{ij}^{tuv} = 0_c$	2
$x^i x'^j l_t'''' l_u'''' l_v'''' \mathcal{R}_{ij}^{tuv} = 0$	1
$x^i l_q'''' l_t'''' l_u'''' l_v'''' \epsilon_{qjn} \mathcal{R}_{ij}^{tuv} = 0^n$	2
$l_p'''' l_q'''' l_t'''' l_u'''' l_v'''' \epsilon_{pim} \epsilon_{qjn} \mathcal{R}_{ij}^{tuv} = 0^{mn}$	4

by multiplying x'''^d on both sides. Then, (5.12) shows the connection of the quadrifocal tensor with three points and one line. Furthermore, if multiplying x''^u , a point in the third view, to (5.12), we can derive:

$$\begin{aligned}
 x^i x'^j x''^k x''^u l_v'''' Q_{ijk}^v &= \frac{1}{6} x^i x'^j x''^k x''^u \epsilon_{rku} \epsilon^{rku} l_v'''' Q_{ijk}^v \\
 &= \frac{1}{6} x^i x'^j l_r'''' l_v'''' \epsilon^{rku} Q_{ijk}^v = 0^u
 \end{aligned} \tag{5.13}$$

where l_r'''' is a line in the third view going through x''^k and x''^u . (5.13) is the correspondence on point-point-line-line. The other correspondences may be obtained in the same manner.

A complete set of the quadrilinear equations involving the quadrifocal tensor are given in Table 5.1. All of these equations are linear in the entries of the quadrifocal tensor Q_{ijk}^v .

As described in Section 5.2, this multiple view geometry can be applied to multiple affine cameras with curvilinear motions. Meanwhile, since the position of points in our research includes the information of time, we can derive the multiple view geometry from fewer time instants if we observe more than one point. For example, in the case of four views, we need 7 time instants, if we observe a single point in the space. However, if we observe 2 point motions in 3D, we only need to observe them 4 time instants to figure out the multiple view geometry.

5.2.2.2 Five View, Six View and Seven View Affine Geometry

Similarly, the five view, six view and seven view geometry can also be derived for the extended cameras. The quintilinear relationship under extended projection is:

$$x^i x'^j x''^k x'''^h x''''^m \epsilon_{ktc} \epsilon_{hud} \epsilon_{mve} \mathcal{R}_{ij}^{tuv} = 0_{cde} \tag{5.14}$$

Table 5.3: Sextilinear relations between point and line coordinates in six views. The final column denotes the number of linearly independent equations.

relation	eq.
$x^i x^j x^k x^h x^m x^n \epsilon_{jrb} \epsilon_{ksc} \epsilon_{htd} \epsilon_{mue} \epsilon_{nvf} \mathcal{S}_i^{rstuv} = 0_{bcdef}$	32
$x^i x^j x^k x^h x^m l^u l^v \epsilon_{jrb} \epsilon_{ksc} \epsilon_{htd} \epsilon_{mue} \mathcal{S}_i^{rstuv} = 0_{bcde}$	16
$x^i x^j x^k x^h l^u l^v \epsilon_{jrb} \epsilon_{ksc} \epsilon_{htd} \mathcal{S}_i^{rstuv} = 0_{bcd}$	8
$x^i x^j x^k l^u l^v \epsilon_{jrb} \epsilon_{ksc} \mathcal{S}_i^{rstuv} = 0_{bc}$	4
$x^i x^j l^u l^v \epsilon_{jrb} \mathcal{S}_i^{rstuv} = 0_b$	2
$x^i l^u l^v \mathcal{S}_i^{rstuv} = 0$	1
$l^m l^r l^s l^t l^u l^v \epsilon^{miv} \mathcal{S}_i^{rstuv} = 0_w$	2

Table 5.4: Septilinear relations between point and line coordinates in seven views. The final column denotes the number of linearly independent equations.

relation	# of eq.
$x^i x^j x^k x^h x^m x^n x^o \epsilon_{ipa} \epsilon_{jqb} \epsilon_{krc} \epsilon_{hsd} \epsilon_{mte} \epsilon_{nuf} \epsilon_{ovg} \mathcal{H}^{pqrstuv} = 0_{bcdefg}$	128
$x^i x^j x^k x^h x^m l^u l^v \epsilon_{ipa} \epsilon_{jqb} \epsilon_{krc} \epsilon_{hsd} \epsilon_{mte} \epsilon_{nuf} \mathcal{H}^{pqrstuv} = 0_{bcdef}$	64
$x^i x^j x^k x^h x^m l^u l^v \epsilon_{ipa} \epsilon_{jqb} \epsilon_{krc} \epsilon_{hsd} \mathcal{H}^{pqrstuv} = 0_{abcde}$	32
$x^i x^j x^k x^h l^u l^v \epsilon_{ipa} \epsilon_{jqb} \epsilon_{krc} \mathcal{H}^{pqrstuv} = 0_{abcd}$	16
$x^i x^j x^k l^u l^v \epsilon_{ipa} \epsilon_{jqb} \epsilon_{krc} \mathcal{H}^{pqrstuv} = 0_{abc}$	8
$x^i x^j l^u l^v \epsilon_{ipa} \epsilon_{jqb} \mathcal{H}^{pqrstuv} = 0_{ab}$	4
$x^i l^u l^v \epsilon_{ipa} \mathcal{H}^{pqrstuv} = 0_a$	2
$l^p l^q l^r l^s l^t l^u l^v \mathcal{H}^{pqrstuv} = 0$	1

\mathcal{R}_{ij}^{tuv} is the quintifocal tensor whose form is described as:

$$\mathcal{R}_{ij}^{tuv} = \epsilon_{ipq} \epsilon_{jrs} \det \begin{bmatrix} \mathbf{a}^p \\ \mathbf{a}^q \\ \mathbf{b}^r \\ \mathbf{b}^s \\ \mathbf{c}^t \\ \mathbf{d}^u \\ \mathbf{e}^v \end{bmatrix} \quad (5.15)$$

where \mathbf{a}^i , \mathbf{b}^i , \mathbf{c}^i , \mathbf{d}^i and \mathbf{e}^i denote the i th row of five camera matrices. The quintifocal tensor \mathcal{R}_{ij}^{tuv} has 243 entries. Excluding 191 zero entries and a scale ambiguity, it has 51 free parameters. And 27 linear equations are given from (5.14) but only 8 of them are linearly independent. Therefore, minimum of 7 corresponding points are required to compute \mathcal{R}_{ij}^{tuv} from images linearly. The quintilinear relationships involving the quintifocal tensor are summerized in Table 5.2.

We next introduce the multiple view geometry of six extended cameras. The sextilinear constraint is expressed as follows:

$$x^i x^j x^k x^l x^m x^n \epsilon_{jtb} \epsilon_{ksc} \epsilon_{ltd} \epsilon_{rme} \epsilon_{twf} \mathcal{S}_i^{rstuv} = 0_{bcdef} \quad (5.16)$$

where \mathcal{S}_i^{rstuv} is the sextifocal tensor (six view tensor) whose form is represented as follows:

$$\mathcal{S}_i^{rstuv} = \epsilon_{ipq} \det \begin{bmatrix} \mathbf{a}^p \\ \mathbf{a}^q \\ \mathbf{b}^r \\ \mathbf{c}^s \\ \mathbf{d}^t \\ \mathbf{e}^u \\ \mathbf{f}^v \end{bmatrix} \quad (5.17)$$

where \mathbf{a}^i , \mathbf{b}^i , \mathbf{c}^i , \mathbf{d}^i , \mathbf{e}^i and \mathbf{f}^i denote the i th row of six camera matrices. The sextifocal tensor \mathcal{S}_i^{rstuv} has 729 entries. If the extended cameras are affine as shown in (5.5), we have only 175 free parameters in \mathcal{S}_i^{rstuv} except zero entries and a scale. On the other hand, (5.16) shows one set of corresponding points provides us 243 linear equations on \mathcal{S}_i^{rstuv} , but only 32 of them are linearly independent. Furthermore, the constraints between multiple sets of points are not independent. As a result, at least 7 corresponding points are required to compute \mathcal{S}_i^{rstuv} from images linearly. The quintilinear relationships are given in Table 5.3.

Finally, let us have a look at the multiple view geometry of seven extended cameras. The septilinear constraint is described as:

$$x^i x^j x^k x^l x^m x^n x^o \epsilon_{ipa} \epsilon_{jqb} \epsilon_{krc} \epsilon_{hsd} \epsilon_{mte} \epsilon_{nuf} \epsilon_{ovg} \mathcal{H}^{pqrstuv} = 0_{abcdefg} \quad (5.18)$$

where $\mathcal{H}^{pqrstuv}$ is the septifocal tensor (seven view tensor) whose form is represented as follows:

$$\mathcal{H}^{pqrstuv} = \det \begin{bmatrix} \mathbf{a}^p \\ \mathbf{b}^q \\ \mathbf{c}^r \\ \mathbf{d}^s \\ \mathbf{e}^t \\ \mathbf{f}^u \\ \mathbf{g}^v \end{bmatrix} \quad (5.19)$$

where \mathbf{a}^i , \mathbf{b}^i , \mathbf{c}^i , \mathbf{d}^i , \mathbf{e}^i , \mathbf{f}^i and \mathbf{g}^i denote the i th row of seven camera matrices. The septifocal tensor $\mathcal{H}^{pqrstuv}$ has 2187 entries, including 576 non-zero entries. Then we have 575 free parameters in $\mathcal{H}^{pqrstuv}$ except a scale. On the other hand, (5.18) provides us 2187 linear equations on $\mathcal{H}^{pqrstuv}$, but only 128 of them are linearly independent. Excluding the dependences between the corresponding points, 7 sets of corresponding points are enough to compute $\mathcal{H}^{pqrstuv}$ from images linearly. The septilinear relationships are given in Table 5.4.

5.2.3 Applications on Multiple View Geometry of Curvilinear Motion Cameras

5.2.3.1 View Transfer

The constraints between corresponding points and multifocal tensors have been derived (see (5.8), (5.14), (5.16), (5.18)), and multifocal tensors can be computed by 7 corresponding points in 4, 5, 6 and 7 views. Thus, if we have the image motions in $N - 1$ images, the image motion in the remaining image can be calculated from N view tensor. It realizes the view transfer from $N - 1$ views to the other view.

5.2.3.2 3D Reconstruction

From (5.5), if image points and extended camera matrix are given, the coordinates of points in 3D can be obtained. Therefore, computing the extended camera matrix is very important.

Assuming that the first viewpoint is at the origin, the camera matrices may now be written as:

$$\begin{aligned}\mathbf{P}_1 &= [\mathbf{I} \mid \mathbf{0}] \\ \mathbf{P}_n &= [\mathbf{H}_{1n} \mid \mathbf{e}_{n1}]\end{aligned}$$

where \mathbf{H}_{1n} denotes the 3×3 homography from the first view to the n th view, and \mathbf{e}_{n1} denotes a 3×4 matrix which represents the epipole, the projection of the first viewpoint in the n th view. Since we consider the affine mapping, the third row of \mathbf{P}_n is $[0, 0, 1, 0, 0, 0, 0]$ as same as \mathbf{P}_1 . Although the order of 1 in the third row is different from that of (5.4), \mathbf{P}_1 and \mathbf{P}_n are still extended affine camera matrices. Note, from \mathbf{P}_1 and \mathbf{P}_n , $[X(T), Y(T), 1, Z(T), T^3, T^2, T]^T$ can be recovered. The epipole of projection from 6D to 2D is a 3D space, and the four column vectors in \mathbf{e}_{n1} are four basis points in this space [98].

Take four views for instance. In (5.12), $x^i x'^j x''^k Q_{ijk}^v$ can be considered as a point, p'''^v . Then p'''^v and l_v''' have the following relation:

$$p'''^v l_v''' = 0. \quad (5.20)$$

That is, p'''^v is a point on the line l_v''' in the fourth view. If x^i , x'^j and x''^k are corresponding points, then p'''^v is also a corresponding point x'''^v in the fourth view. Thus,

(5.12) may be rewritten as:

$$x^i x'^j x''^k Q_{ijk}^v = x'''^v. \quad (5.21)$$

Then, the following equations can be derived:

$$x'''^v = H_{14i}^v x^i \quad (5.22)$$

$$H_{14i}^v = x'^j x''^k Q_{ijk}^v \quad (5.23)$$

H_{14i}^v denotes a homography from the first view to the fourth view. If we have two pairs of x'^j and x''^k , two H_{14i}^v can be obtained:

$$H_{14i}^v = x_1'^j x_1''^k Q_{ijk}^v \quad (5.24)$$

$$H_{14i}^v = x_2'^j x_2''^k Q_{ijk}^v \quad (5.25)$$

Thus, we have the following constraints:

$$\mathbf{e}_{41} = \mathbf{H}_{14} \mathbf{e}_{14} \quad (5.26)$$

$$\mathbf{e}_{41} = \mathbf{H}'_{14} \mathbf{e}_{14} \quad (5.27)$$

If \mathbf{H}_{14} and \mathbf{H}'_{14} are independent, we can obtain:

$$(\mathbf{H}_{14} - \mathbf{H}'_{14}) \mathbf{e}_{14} = \mathbf{0} \quad (5.28)$$

Since \mathbf{H}_{14} and \mathbf{H}'_{14} have been figured out, epipole \mathbf{e}_{14} can also be derived. However, here we only can derive one column vector in \mathbf{e}_{14} . For obtaining the other three column vectors, we need other three homography pairs. Once \mathbf{e}_{14} and \mathbf{H}_{14} are known, \mathbf{e}_{41} can be calculated from (5.26). Thus, the camera matrix \mathbf{P}_4 can be computed from \mathbf{H}_{14} and \mathbf{e}_{41} . \mathbf{P}_2 and \mathbf{P}_3 can also be derived in the same manner. Then, using \mathbf{P}_1 , \mathbf{P}_2 , \mathbf{P}_3 , \mathbf{P}_4 and a set of corresponding points in these camera images, we can reconstruct \mathbf{X} in (5.5), and hence the point in 3D space and time T .

5.2.4 Experiments

We next show the results of some experiments. We at first discuss the approximate relationship between affine cameras and projective cameras. We next show that the quadrifocal tensor for extended affine cameras can be computed from image motions viewed from arbitrary curvilinear motion cameras by the results from real images, and can be used for generating one view from the others and for recovering 3D motions. We finally evaluate the stability of extracted quadrifocal tensors for extended affine cameras.

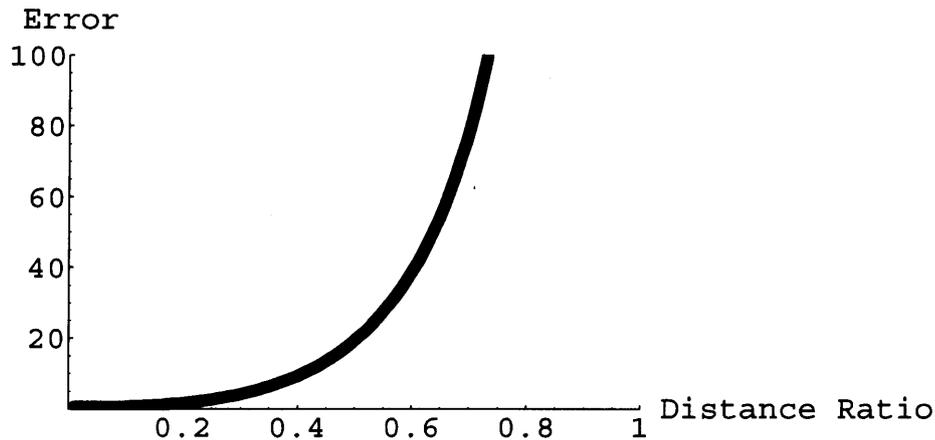


Figure 5.2: The relationship between the distance ratio and the reprojection errors under the projective camera model.

5.2.4.1 Approximate Relationship between Affine Camera and Projective Camera

Affine camera is an ideal model whose optical center is at infinity. It does not exist in the real world. Therefore, we here desire to find some clue to the approximate relationship between affine camera and the most general camera model, projective camera.

We consider a ratio between the “radius” of the 3D motion (the average distance between the center and the boundary of the motion) and the distance between motion’s center and projective camera, which we call *distance ratio*. The relationship between distance ratio and reprojection error (its definition is same to stability evaluation) is shown in Figure 5.2. The image size is 640×480 . As we can see, when distance ratio ≤ 0.4 , reprojection error is less than 10.

5.2.4.2 Real Image Experiment

We next show the result from a real image experiment.

In this experiment, we used four cameras (Sony DFW-VL500), one of which is static (Camera 4) and three of which (Camera 1, Camera 2 and Camera 3) are controlled by 3-axis robots (Originalmind 3-Axis Robot) respectively to undergo different curvilinear motions as shown in Figure 5.3. We computed quadrifocal tensors among these four cameras by using two moving points in the 3D space. The experimental circumstance is shown in Figure 5.4. Figure 5.5(a), (b), (c) and (d) show image motions of two 3D



Figure 5.3: Originalmind 3-Axis Robot.

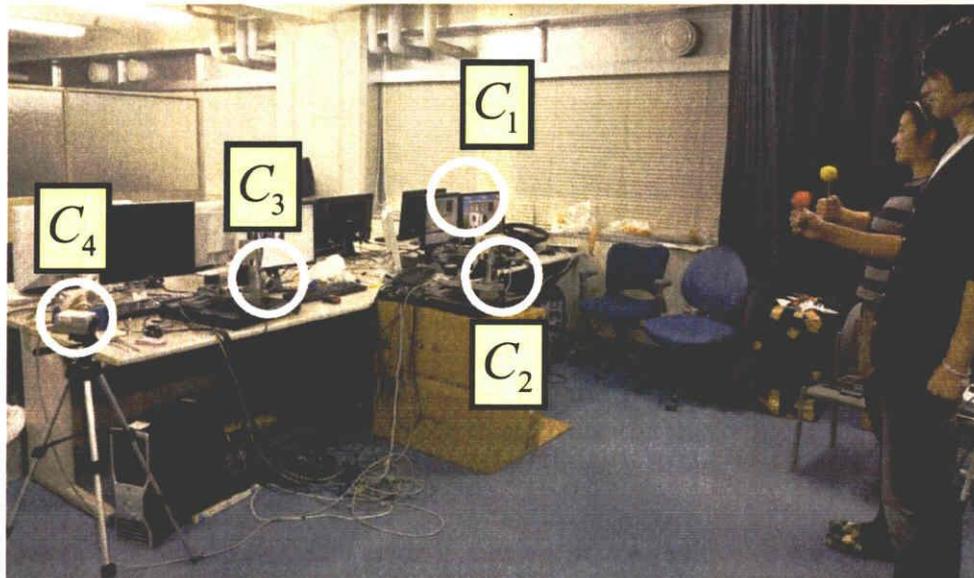


Figure 5.4: Experimental Circumstance.

points viewed from Camera 1, 2, 3 and 4 respectively. Here, distance ratio is about 0.25. Such configuration could be considered approximating with affine camera models as addressed.

The green and red curves in Figure 5.5 represent two different image motions. The 7 white points on the two curves in each image are corresponding points used for computing the quadrifocal tensor. The curves in Figure 5.6(b) show image motions computed from the extended quadrifocal tensor in camera 2. The average error of the recovered

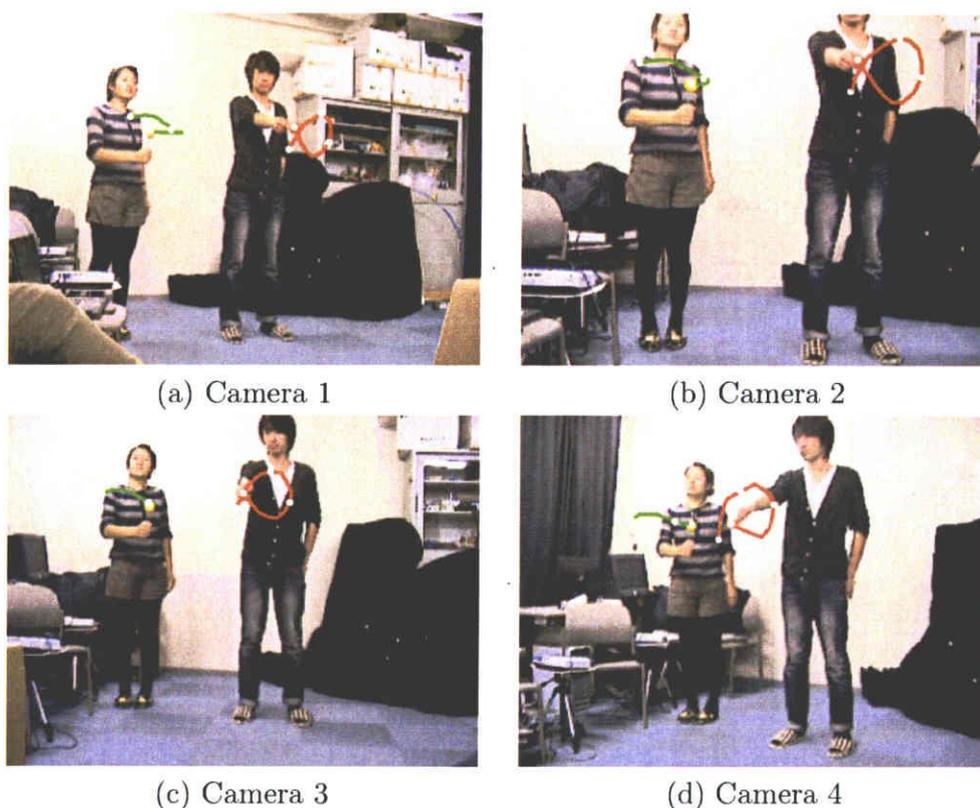


Figure 5.5: Multiple point motion experiment. Figures (a), (b), (c) and (d) show four views of the motion in camera 1, 2, 3 and 4. The green and red curves show two different image motions in each view. The 7 white points on the two curves in each image are corresponding points used for computing the quadrifocal tensor. Note that Camera 1, Camera 2 and Camera 3 have curvilinear motion.

image motion is 6.0 pixels. The error is caused by the following reasons according to our analysis: (1) camera motion error. It is difficult for controlling four cameras to do rigorous spline curve motions; (2) approximate error. We used projective cameras to approximate affine cameras; (3) selection of corresponding points. The correct results are derived from corresponding points which have 6D variety. For example, coplanar corresponding points may arise degeneration.

As we can see, the quadrifocal tensor defined under extended projection can be derived from multiple point motions viewed from three curvilinear motion cameras and one static camera, and they are practical for generating images of multiple point motions viewed from curvilinear motion camera.

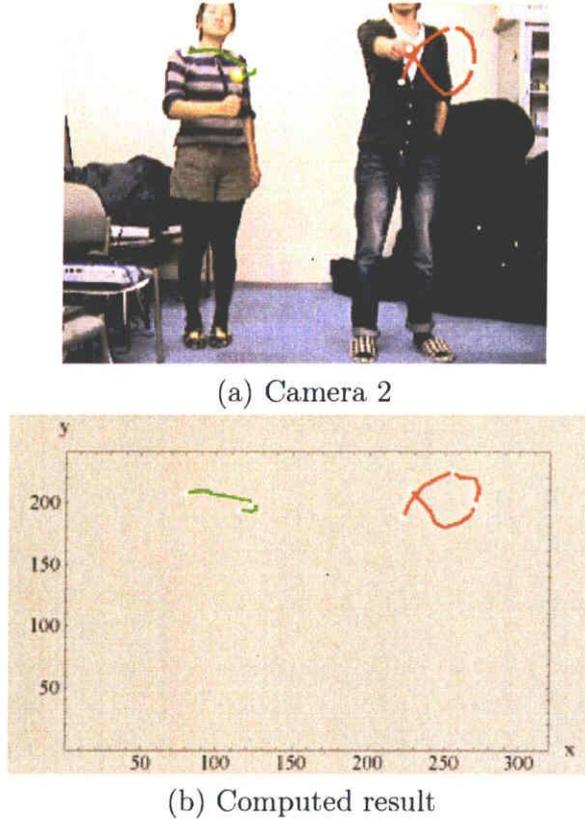


Figure 5.6: Computed result of the multiple point motion experiment. The curves in (b) show image motions computed by the extended quadrifocal tensor in camera 2.

5.2.4.3 3D Reconstruction

We next show the results of 3D reconstruction using the 3D configurations shown in Figure 5.7 to verify another application, 3D reconstruction. For convenience, we assumed camera C_1 a static camera in this experiment. The non-rigid 3D motion is projected to four cameras as shown in Figure 5.8. Figure 5.9(a) shows the real 3D motion. The corresponding points with Gaussian noise of standard deviation of 1 pixel in the four images were used to figure out the coordinates of each point in the 3D space by using the method addressed in section 5.2.3.2. The reconstructed result is shown in Figure 5.9(b). We can see the shape of the 3D motion is recovered properly.

5.2.4.4 Stability Evaluation

We next show the stability of extracted quadrifocal tensors under extended projections. For evaluating the extracted quadrifocal tensors, we computed reprojection errors derived from the quadrifocal tensors. The reprojection error is defined as:

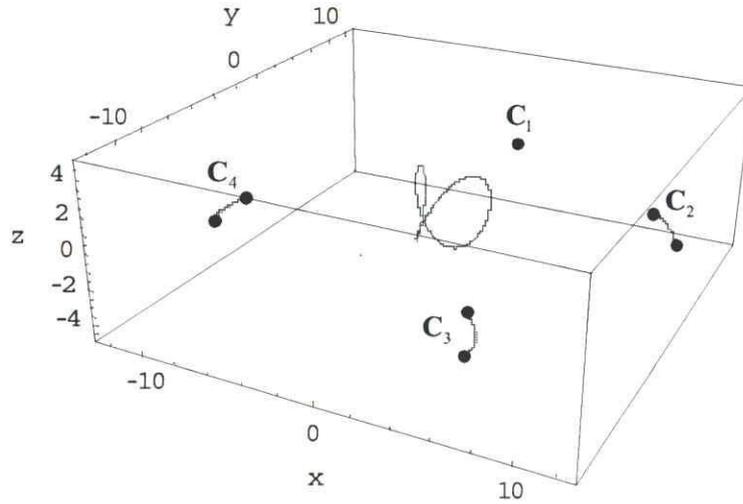


Figure 5.7: One static camera, three curvilinear motion cameras and a moving point in the 3D space.

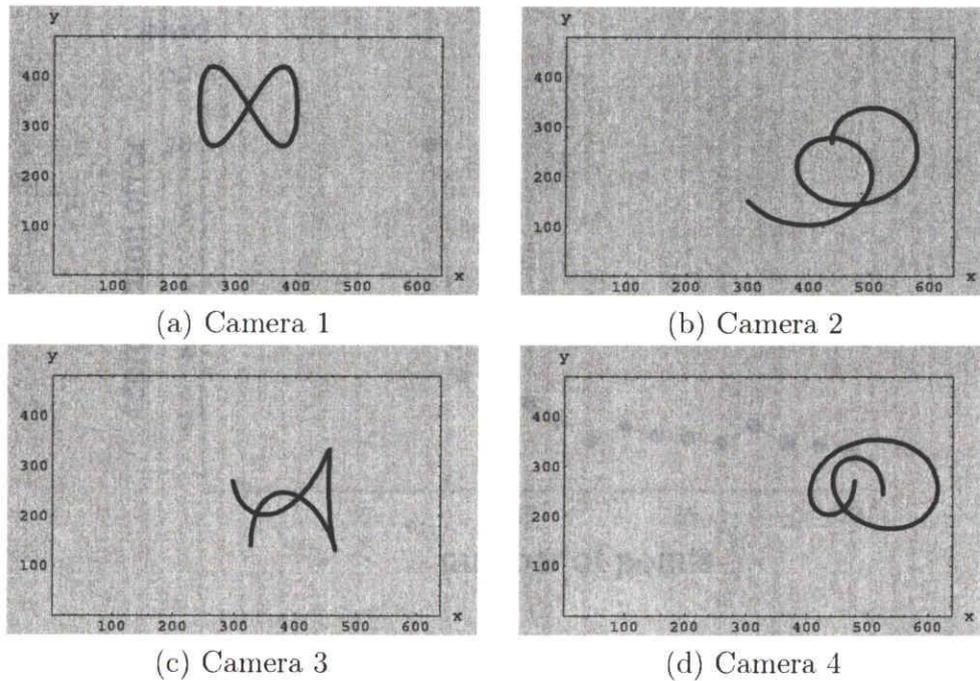


Figure 5.8: Non-rigid motion in the 3D space is projected to four cameras.

$\frac{1}{N} \sum_{i=1}^N d(\mathbf{m}_i, \hat{\mathbf{m}}_i)$, where $d(\mathbf{m}_i, \hat{\mathbf{m}}_i)$ denotes a distance between a true point \mathbf{m}_i and a point $\hat{\mathbf{m}}_i$ recovered from the quadrifocal tensor. We increased the number of corre-

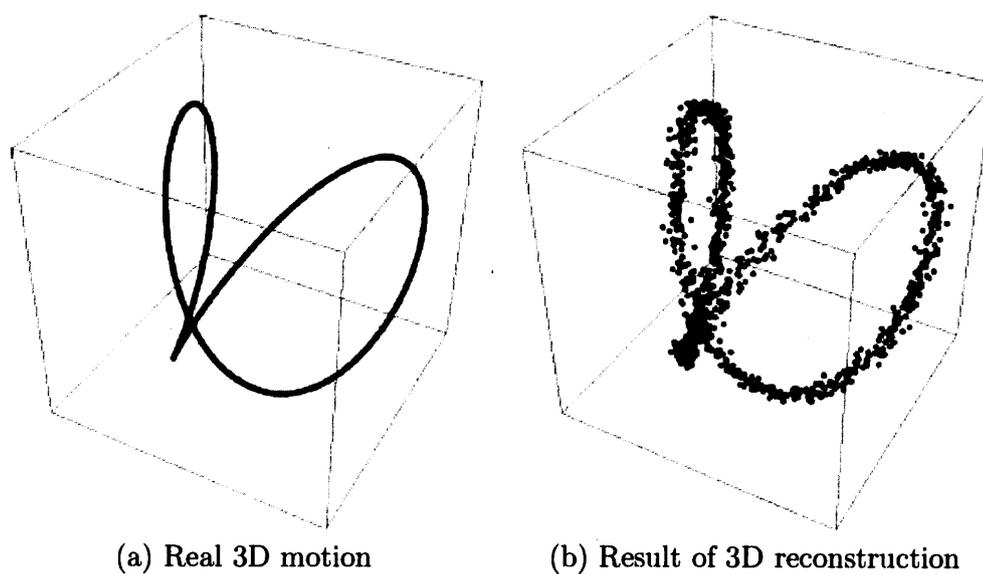


Figure 5.9: 3D reconstruction.

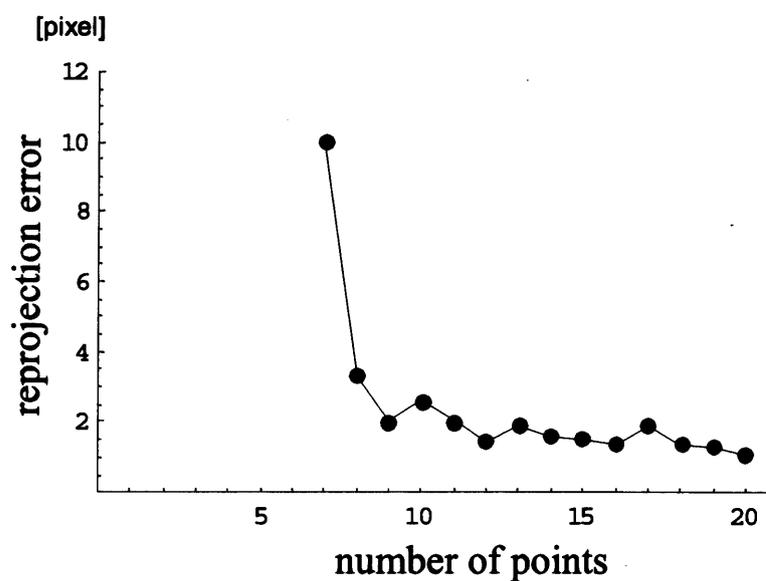


Figure 5.10: The relationship between the number of corresponding points used for computing quadrifocal tensors and the reprojection errors. Camera trajectories and 3D point motions are randomly generated for 1000 times by changing the control points of the Bezier curves.

sponding points used for computing quadrifocal tensors in four views from 7 to 20, and evaluated the reprojection errors. Camera trajectories and 3D point motions are ran-

domly generated for 1000 times by changing the control points of the B-spline curves. For each camera trajectory and 3D point, the images motions are generated 100 times by adding Gaussian noise with the standard deviation of 1 pixel. Figure 5.10 shows the relationship between the number of corresponding points and the reprojection errors. As we can see, the stability is obviously improved by using a few more points than the minimum number of corresponding points.

5.3 Multiple View Geometry for Projective Curvilinear Motion Cameras

If all the cameras are projective cameras and the cameras are moving following cubic Bezier curves, point motions are projected to projective camera as follows:

$$\begin{aligned}
 \lambda \begin{bmatrix} x(T) \\ y(T) \\ 1 \end{bmatrix} &= \mathbf{P}_a(\mathbf{X}(T) - \Delta\mathbf{X}_i) \\
 &= \mathbf{P}_a(\mathbf{X}(T) - \mathbf{GAE} \begin{bmatrix} T^3 \\ T^2 \\ T \\ 1 \end{bmatrix}) \\
 &= \mathbf{P}_a[\mathbf{I}, -\mathbf{GAE}] \begin{bmatrix} \mathbf{X}(T) \\ T^3 \\ T^2 \\ T \\ 1 \end{bmatrix} \\
 &= \mathbf{P}_a[\mathbf{I}, -\mathbf{GAE}] \begin{bmatrix} X(T) \\ Y(T) \\ Z(T) \\ 1 \\ T^3 \\ T^2 \\ T \\ 1 \end{bmatrix} \\
 &= \mathbf{P}_b \begin{bmatrix} X(T) \\ Y(T) \\ Z(T) \\ 1 \\ T^3 \\ T^2 \\ T \\ 1 \end{bmatrix} \\
 &= \mathbf{P} \begin{bmatrix} X(T) \\ Y(T) \\ Z(T) \\ T^3 \\ T^2 \\ T \\ 1 \end{bmatrix}
 \end{aligned} \tag{5.29}$$

where \mathbf{P}_a denotes a 3×4 projective camera matrix, and \mathbf{P} denotes a 3×7 extended camera matrix. We therefore find that, from (5.29), the projections of point motions to multiple cameras with curvilinear motions can be described by the multilinear relationship under the projective projection from 6D to 2D. In the next sections, we will discuss the geometry of projective projections.

5.3.1 Projective Projections from 6D to 2D

We first consider a projection from 6D space to 2D space. Let $\mathbf{X} = [X^1, X^2, X^3, X^4, X^5, X^6, X^7]^\top$ be the homogeneous coordinates of a 6D space point projected to a point in the 2D space, whose homogeneous coordinates are represented by $\mathbf{x} = [x^1, x^2, x^3]^\top$. Then, the extended projection from \mathbf{X} to \mathbf{x} can be described as follows:

$$\mathbf{x} \sim \mathbf{P}\mathbf{X} \quad (5.30)$$

where (\sim) denotes equality up to a scale, and \mathbf{P} denotes the following 3×7 matrix:

$$\mathbf{P} = \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} & m_{15} & m_{16} & m_{17} \\ m_{21} & m_{22} & m_{23} & m_{24} & m_{25} & m_{26} & m_{27} \\ m_{31} & m_{32} & m_{33} & m_{34} & m_{35} & m_{36} & m_{37} \end{bmatrix} \quad (5.31)$$

From (5.30), we find that the extended camera, \mathbf{P} , has 20 DOF except a scale. In the next section, we consider the multiple view geometry of the extended cameras.

5.3.2 Projective Multiple View Geometry from 6D to 2D

From (5.30), we have the following equation for N extended projective cameras:

$$\begin{bmatrix} \mathbf{P} & \mathbf{x} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{P}' & \mathbf{0} & \mathbf{x}' & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{P}'' & \mathbf{0} & \mathbf{0} & \mathbf{x}'' & \cdots & \mathbf{0} \\ \vdots & & & & & \vdots \end{bmatrix} \begin{bmatrix} \mathbf{X} \\ \lambda \\ \lambda' \\ \lambda'' \\ \vdots \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \vdots \end{bmatrix} \quad (5.32)$$

where, the leftmost matrix, \mathbf{M} , in (5.32) is $3N \times (7 + N)$, and the $(7 + N) \times (7 + N)$ minors \mathbf{Q} of \mathbf{M} constitute multilinear relationships under the extended projection as: $\det \mathbf{Q} = 0$. We can choose any $7 + N$ rows from \mathbf{M} to constitute \mathbf{Q} , but we have to take at least 2 rows from each camera for deriving meaningful N view relationships (note, each camera has 3 rows in \mathbf{M}). Thus, $7 + N \geq 2N$ must hold for defining multilinear relationships for N view geometry in the 6D space. Thus, we find that, the multilinear relationship for 7 views is the maximal linear relationship in the 6D space.

We next consider the minimum number of points required for computing the multifocal tensors. The geometric DOF of N extended projective cameras is $20N - 48$, since

Table 5.5: Quadilinear relations between point and line coordinates in four views. The final column denotes the number of linearly independent equations.

relation	# of eq.
$x^i x'^j x''^k x'''^l \epsilon_{lvd} Q_{ijk}^v = 0_d$	2
$x^i x'^j x''^k l_v''' Q_{ijk}^v = 0$	1
$x^i x'^j l_r'' l_v''' \epsilon^{rku} Q_{ijk}^v = 0^u$	2
$x^i l_q'' l_r''' l_v'''' \epsilon^{qjt} \epsilon^{rku} Q_{ijk}^v = 0^{tu}$	4
$l_p l_q'' l_r''' l_v'''' \epsilon^{pis} \epsilon^{qkt} \epsilon^{rku} Q_{ijk}^v = 0^{stu}$	8

each extended projective camera has 20 DOF and these N cameras are in a single 6D projective space whose DOF is 48.

Meanwhile, if we are given M points in the 6D space, and let them be projected to N cameras defined in (5.30). Then, we derive $2MN$ measurements from images, while we have to compute $20N - 48 + 6M$ components for fixing all the geometry in the 6D space. Thus, this condition must hold for computing the multifocal tensors from images: $2MN \geq 20N - 48 + 6M$. We find that minimum of 16,13,12,12 points are required to compute multifocal tensors in four, five, six and seven views nonlinearly.

We next introduce the multiple view geometry of four extended projective cameras. For four views, the sub square matrix \mathbf{Q} is 11×11 . From $\det \mathbf{Q} = 0$, we have the following quadrilinear relationship under extended camera projections:

$$x^i x'^j x''^k x'''^l \epsilon_{lvd} Q_{ijk}^v = 0_d \quad (5.33)$$

where ϵ_{lvd} denotes a tensor, which represents a sign based on permutation from $\{l,v,d\}$ to $\{1,2,3\}$. Q_{ijk}^v is the quadrifocal tensor for the extended cameras and has the following form:

$$Q_{ijk}^v = \epsilon_{ipq} \epsilon_{jrs} \epsilon_{ktu} \det \begin{bmatrix} \mathbf{a}^p \\ \mathbf{a}^q \\ \mathbf{b}^r \\ \mathbf{b}^s \\ \mathbf{c}^t \\ \mathbf{c}^u \\ \mathbf{d}^v \end{bmatrix} \quad (5.34)$$

where \mathbf{a}^i denotes the i th row of \mathbf{P} , \mathbf{b}^i denotes the i th row of \mathbf{P}' , \mathbf{c}^i denotes the i th row of \mathbf{P}'' and \mathbf{d}^i denotes the i th row of \mathbf{P}''' respectively. The quadrifocal tensor Q_{ijk}^v is $3 \times 3 \times 3 \times 3$ and has 81 entries. If the extended cameras are projective as shown in (5.30), we have only 80 free parameters in Q_{ijk}^v except a scale ambiguity. On the other hand, (5.33) provides us 3 linear equations on Q_{ijk}^v , but only 2 of them are linearly independent. Thus, at least 40 corresponding points are required to compute Q_{ijk}^v from images linearly.

A complete set of the quadrilinear equations involving the quadrifocal tensor are given in Table 5.5. All of these equations are linear in the entries of the quadrifocal tensor \mathcal{Q}_{ijk}^v .

As described in Section 5.3, this multiple view geometry can be applied to multiple projective cameras with curvilinear motions. Meanwhile, since the position of points in our research includes the information of time, we can derive the multiple view geometry from less time intervals if we observe more than one point. For example, in the case of four views, we need 40 time intervals, if we observe a single point in the space. However, if we observe 4 point motions in 3D, we only need to observe them 10 time intervals to figure out the multiple view geometry.

5.3.3 Experiments

We next show the results of some experiments. We first show the results from real image experiment, in which the quadrifocal tensor for extended cameras is computed from image motions viewed from arbitrary curvilinear motion cameras, and is used to generate a view from the other three views of moving cameras. We next evaluate the stability of extracted quadrifocal tensors for extended cameras.

5.3.3.1 Real Image Experiment

In this experiment, we used four cameras, three of which have curvilinear motions and one of which is static. The experimental setup is same as Section 5.2.4.2. We computed quadrifocal tensors among these 4 cameras by using two moving points in the 3D space. Figure 5.11(a), (b), (c) and (d) show image motions of two points viewed from three curvilinear motion cameras and one static camera. The green curves and red curves represent two different image motions. The 20 white points on each curve are corresponding points used for computing the quadrifocal tensor. The extracted tensor is next used to generate the image motion of camera 2 from the image motions of camera 1, 3 and 4. The green and red curve in Figure 5.12(b) shows image motions computed from the extended quadrifocal tensor in camera 2. The 40 points in (e) show points used for computing the quadrifocal tensor. Comparing Figure 5.12(a) and (b), we find that the computed motions are reasonably accurate. Thus, we can see that the quadrifocal tensor defined under extended projection can be derived from arbitrary multiple point motions viewed from 4 cameras, even if three of them undergo arbitrary curvilinear motion, and it is practical for generating images of multiple point motions viewed from curvilinear motion camera.

5.3.3.2 Stability Evaluation

We next show the stability of extracted quadrifocal tensors under extended projections. Figure 5.13 shows a 3D configuration of 4 curvilinear motion cameras and a moving point used in this experiment. The black points show the viewpoints of four cameras,

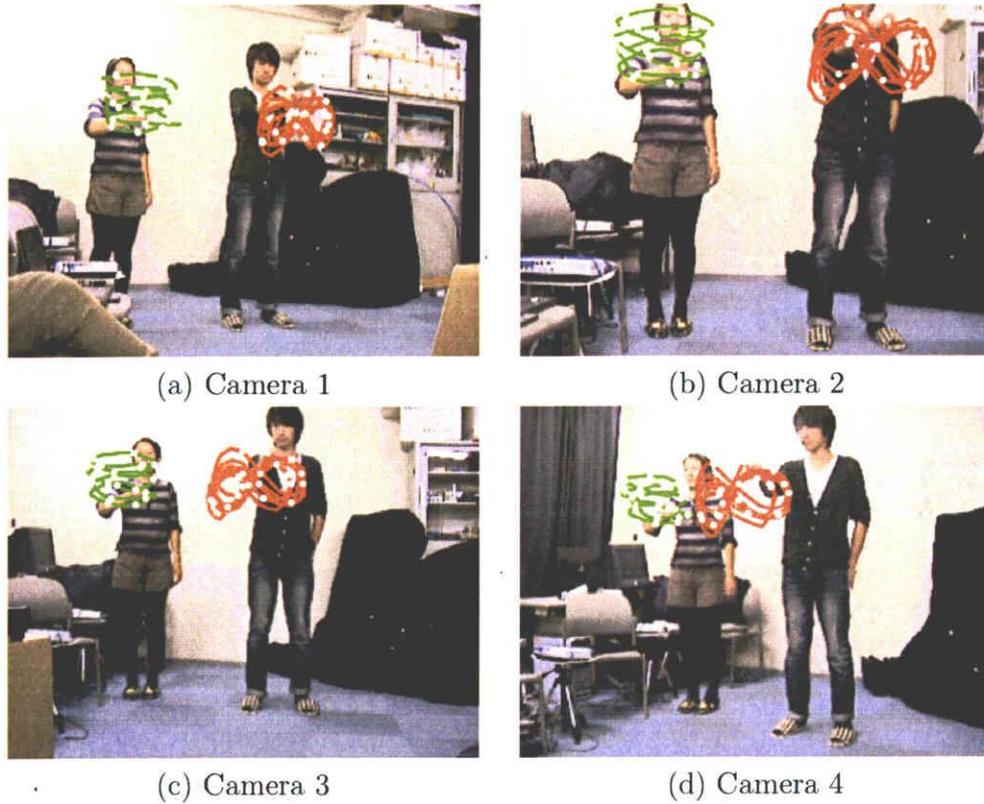
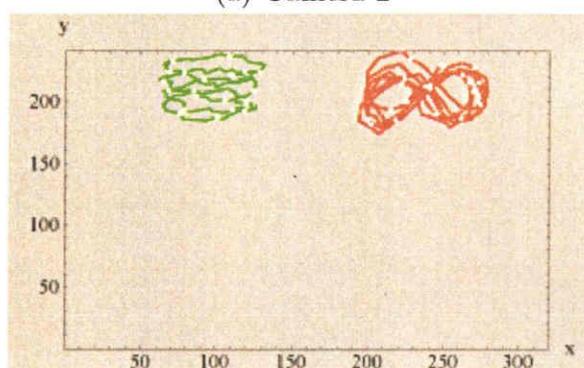


Figure 5.11: Multiple point motion experiment. Figures (a), (b), (c) and (d) show image motions of two points viewed from three curvilinear motion cameras (Camera 1, Camera 2 and Camera 3) and one static camera (Camera 4). The green curves and red curves represent two different image motions. The 20 points on each curve are corresponding points used for computing the quadrifocal tensor.

\mathbf{C}_1 , \mathbf{C}_2 , \mathbf{C}_3 and \mathbf{C}_4 , before motions, and the white points show those after the Bezier curve motions. The motions of these four cameras are different and unknown. The black curve shows a locus of a freely moving point, which is projected to four curvilinear motion cameras as shown in Figure 5.14. For evaluating the extracted quadrifocal tensors, we computed reprojection errors derived from the quadrifocal tensors. The reprojection error is defined as: $\frac{1}{N} \sum_{i=1}^N d(\mathbf{m}_i, \hat{\mathbf{m}}_i)^2$, where $d(\mathbf{m}_i, \hat{\mathbf{m}}_i)$ denotes a distance between a true point \mathbf{m}_i and a point $\hat{\mathbf{m}}_i$ recovered from the quadrifocal tensor. We increased the number of corresponding points used for computing quadrifocal tensors in four views from 40 to 60, and evaluated the reprojection errors. The Gaussian noise with the standard deviation of 1 pixel is added to every point on the locus. Figure 5.15 shows the relationship between the number of corresponding points and the reprojection errors. As we can see, the stability is drastically improved by using a few more points than the minimum number of corresponding points.



(a) Camera 2



(b) Computed result

Figure 5.12: Computed result of the multiple point motion experiment. The green and red curve in (b) shows image motions computed from the extended quadrifocal tensor in Camera 2.

5.4 Summary

In this chapter, we used affine camera model and projective camera model to describe the multilinear relationship under the projection from 6D to 2D respectively, which can represent the geometric relationship of multiple curvilinear motion cameras whose motions are represented by cubic Bezier curves. The multifocal tensors defined under 6D to 2D multilinear relationships can be computed from non-rigid object motions viewed from multiple cameras with arbitrary curvilinear motions. We also showed that the multilinear relationships are very useful for generating images of non-rigid motions viewed from cameras with arbitrary curvilinear motions. The method was implemented and tested by using real image sequences. The stability of extracted quadrifocal tensors was also evaluated.

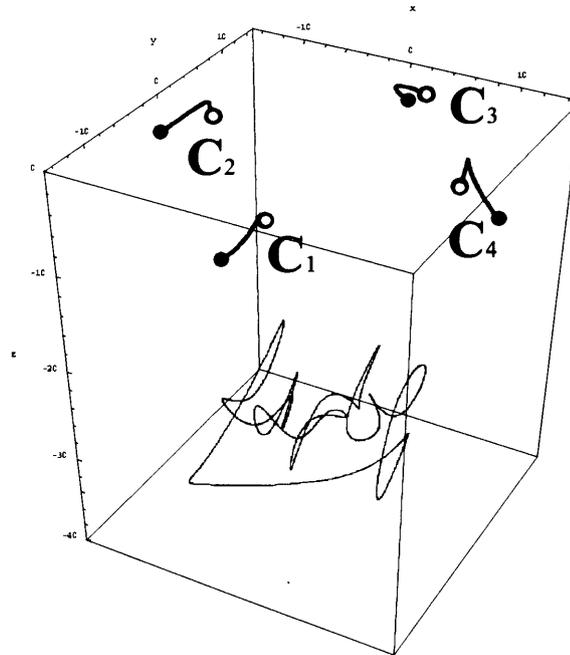


Figure 5.13: 3D configuration of four curvilinear motion cameras and one moving point.

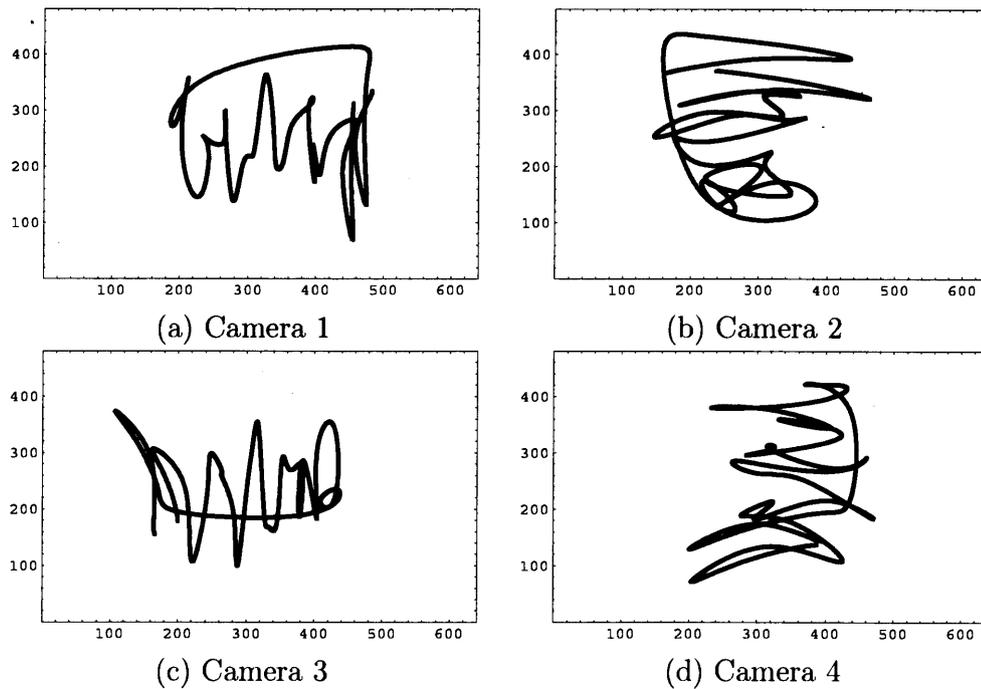


Figure 5.14: Non-rigid motion in the 3D space is projected to four curvilinear motion cameras.

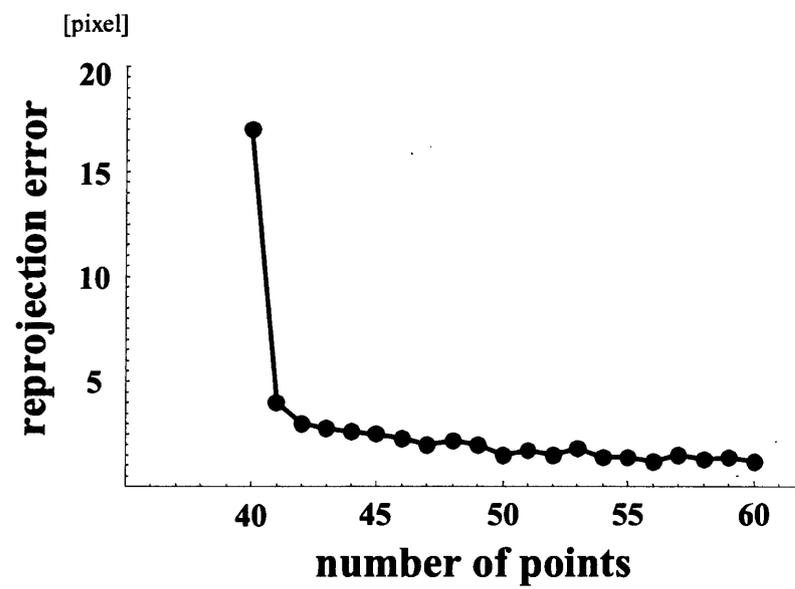


Figure 5.15: The relationship between the number of corresponding points used for computing quadrifocal tensors and the reprojection errors.

Chapter 6

Dynamic Multiple View Geometry with B-Spline Curve Motion Cameras

6.1 Camera Trajectory Modeled by Degree- n B-Spline Curve

In chapter 3, 4 and 5, we introduced the dynamic multiple view geometry, in which the camera trajectories are modeled by Degree- n Bezier curves. However, when n is large, the multiple view geometry will become very complex, uncomputable. Moreover, the main problem with Bezier curves is their lack of local control. To overcome the problems, we consider degree- n B-Spline curve, a piecewise curve, to represent the camera trajectories. Thus, we can use low degree B-spline curve to describe a complex curve.

6.2 B-Spline Curve

Given m knots t_i with

$$t_1 \leq t_2 \leq \dots \leq t_m \quad (6.1)$$

A B-spline of degree- n is a parametric curve composed of a linear combination of basis B-splines $b_{i,n}$ of degree- n

$$\mathbf{S} = \sum_{i=0}^{m-n-1} \mathbf{P}_i b_{i,n}(t), \quad t \in [t_n, t_{m-n}] \quad (6.2)$$

The basis B-splines of degree- n is defined as follows:

$$b_{j,0}(t) = \begin{cases} 1, & t_j \leq t < t_{j+1} \\ 0, & \text{else} \end{cases} \quad j = 0, \dots, m-1 \quad (6.3)$$

$$b_{j,n}(t) = \frac{t - t_j}{t_{j+n} - t_j} b_{j,n-1}(t) + \frac{t_{j+n+1} - t}{t_{j+n+1} - t_{j+1}} b_{j+1,n-1}(t), \quad j = 0, \dots, m - n - 1 \quad (6.4)$$

If the knots are equidistant the B-spline curve is called to be **uniform**, otherwise **non-uniform**.

6.3 Non-Rigid Object Motions viewed from B-Spline Curve Motion Cameras

6.3.1 Cubic B-Spline Curve

Cubic (degree-3) B-spline curve with uniform knot-vector is the most commonly used form of B-spline curve. In this section, we will make use of uniform cubic B-spline curve to represent the trajectory of the camera.

The i th segment of a cubic B-spline curve is defined using four control points, \mathbf{Q}_i , \mathbf{Q}_{i+1} , \mathbf{Q}_{i+2} , \mathbf{Q}_{i+3} , a basis matrix \mathbf{B} and a parameter t as follows:

$$\mathbf{S}_i = [\mathbf{Q}_i, \mathbf{Q}_{i+1}, \mathbf{Q}_{i+2}, \mathbf{Q}_{i+3}] \mathbf{B} \begin{bmatrix} t^3 \\ t^2 \\ t \\ 1 \end{bmatrix}, \quad (6.5)$$

$$t \in [0, 1], \quad i = 0, 1, \dots, n - 1$$

where, the fourth entries of $\mathbf{Q}_i, \mathbf{Q}_{i+1}, \mathbf{Q}_{i+2}, \mathbf{Q}_{i+3}$ are equal to 0, and \mathbf{B} denotes the following 4×4 matrix:

$$\mathbf{B} = \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 0 & 4 \\ -3 & 3 & 3 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix} / 6,$$

6.3.2 Non-Rigid Object Motions viewed from Cubic B-Spline Curve Motion Cameras

Consider an affine camera which projects points in 3D to 2D images. The motions of a point in the 3D space can be represented by homogeneous coordinates, $\mathbf{X}(T) = [X(T), Y(T), Z(T), 1]^T$, where T denotes time. The motions are projected to images, and can be observed as a set of points, $\mathbf{x}(T) = [x(T), y(T), 1]^T$.

The camera motion is relative to the camera original position, and hence its fourth entry is equal to 0, and thus represented as $\mathbf{S}_i = [\Delta X, \Delta Y, \Delta Z, 0]^T$.

Assume each motion segment \mathbf{S}_i spends time T_a . Then $t = T/T_a - i$. Thus, the parameter vector can be written as:

$$\begin{aligned} \begin{bmatrix} t^3 \\ t^2 \\ t \\ 1 \end{bmatrix} &= \begin{bmatrix} 1/T_a^3 & -3i/T_a^2 & 3i^2/T_a & -i^3 \\ 0 & 1/T_a^2 & -2i/T_a & i^2 \\ 0 & 0 & 1/T_a & -i \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} T^3 \\ T^2 \\ T \\ 1 \end{bmatrix} \\ &= \mathbf{C}(i) \begin{bmatrix} T^3 \\ T^2 \\ T \\ 1 \end{bmatrix} \end{aligned} \quad (6.6)$$

Let $\mathbf{G}_i = [\mathbf{Q}_i, \mathbf{Q}_{i+1}, \mathbf{Q}_{i+2}, \mathbf{Q}_{i+3}]$. Then, \mathbf{S}_i can be rewritten as follows:

$$\mathbf{S}_i = \mathbf{G}_i \mathbf{BC}(i) \begin{bmatrix} T^3 \\ T^2 \\ T \\ 1 \end{bmatrix} \quad (6.7)$$

Thus, point motions are projected to affine camera as follows:

$$\begin{aligned} \begin{bmatrix} x(T) \\ y(T) \\ 1 \end{bmatrix} &= \mathbf{P}(\mathbf{X}(T) - \mathbf{S}_i) \\ &= \mathbf{P}(\mathbf{X}(T) - \mathbf{G}_i \mathbf{BC}(i) \begin{bmatrix} T^3 \\ T^2 \\ T \\ 1 \end{bmatrix}) \\ &= \mathbf{P}[\mathbf{I}, -\mathbf{G}_i \mathbf{BC}(i)] \begin{bmatrix} \mathbf{X}(T) \\ T^3 \\ T^2 \\ T \\ 1 \end{bmatrix} \\ &= \mathbf{P}'_i \begin{bmatrix} X(T) \\ Y(T) \\ Z(T) \\ T^3 \\ T^2 \\ T \\ 1 \end{bmatrix} \end{aligned} \quad (6.8)$$

where \mathbf{P} denotes a 3×4 affine camera matrix, and \mathbf{P}'_i denotes a 3×7 extended camera matrix for i th camera motion segment. We therefore find that, from (6.8), the projections of point motions to multiple cameras with cubic B-spline curve motions can be described by the multilinear relationship under the projection from 6D to 2D.

6.4 Multiple View Geometry with Cubic B-Spline Curve Motion Cameras

We first consider a projection from 6D space to 2D space. Let $\mathbf{X} = [X^1, X^2, X^3, X^4, X^5, X^6, X^7]^\top$ be the homogeneous coordinates of a 6D space point projected to a point in the 2D space, whose homogeneous coordinates are represented by $\mathbf{x} = [x^1, x^2, x^3]^\top$. Then, the extended projection from \mathbf{X} to \mathbf{x} can be described as follows:

$$\mathbf{x} \sim \mathbf{P}\mathbf{X} \quad (6.9)$$

where (\sim) denotes equality up to a scale, and \mathbf{P} denotes the following 3×7 matrix:

$$\mathbf{P} = \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} & m_{15} & m_{16} & m_{17} \\ m_{21} & m_{22} & m_{23} & m_{24} & m_{25} & m_{26} & m_{27} \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (6.10)$$

From (6.9), we find that the extended camera, \mathbf{P} , has 14 DOF. Next, we consider the multiple view geometry of the extended cameras.

From (6.9), we have the following equation for N extended projective cameras:

$$\begin{bmatrix} \mathbf{P} & \mathbf{x} & 0 & 0 & \cdots & 0 \\ \mathbf{P}' & 0 & \mathbf{x}' & 0 & \cdots & 0 \\ \mathbf{P}'' & 0 & 0 & \mathbf{x}'' & \cdots & 0 \\ \vdots & & & & & \vdots \end{bmatrix} \begin{bmatrix} \mathbf{X} \\ \lambda \\ \lambda' \\ \lambda'' \\ \vdots \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \end{bmatrix} \quad (6.11)$$

where, the leftmost matrix, \mathbf{M} , in (6.11) is $3N \times (7 + N)$, and the $(7 + N) \times (7 + N)$ minors \mathbf{Q} of \mathbf{M} constitute multilinear relationships under the extended projection as: $\det \mathbf{Q} = 0$. We can choose any $7 + N$ rows from \mathbf{M} to constitute \mathbf{Q} , but we have to take at least 2 rows from each camera for deriving meaningful N view relationships (note, each camera has 3 rows in \mathbf{M}). Thus, $7 + N \geq 2N$ must hold for defining multilinear relationships for N view geometry in the 6D space. Thus, we find that, the multilinear relationship for 7 views is the maximal linear relationship in the 6D space.

We next consider the minimum number of points required for computing the multifocal tensors. The geometric DOF of N extended affine cameras is $14N - 42$, since each extended affine camera has 14 DOF and these N cameras are in a single 6D affine space whose DOF is 42. Meanwhile, if we are given M points in the 6D space, and let them be projected to N cameras defined in (6.9). Then, we derive $2MN$ measurements from images, while we have to compute $14N - 42 + 6M$ components for fixing all the geometry in the 6D space. Thus, the following condition must hold for computing the multifocal tensors from images: $2MN \geq 14N - 42 + 6M$. We find that minimum of 7 points are required to compute multifocal tensors in four, five, six and seven views.

We next introduce the multiple view geometry of four extended cameras. For four views, the sub square matrix \mathbf{Q} is 11×11 . From $\det \mathbf{Q} = 0$, we have the following

quadrilinear relationship under extended camera projections:

$$x^i x'^j x''^k x'''^h \epsilon_{hvd} \mathcal{Q}_{ijk}^v = 0_d \quad (6.12)$$

where ϵ_{hvd} (or its contravariant counterpart, ϵ^{hvd}) denotes a tensor, which represents a sign based on permutation from $\{h,v,d\}$ to $\{1,2,3\}$. \mathcal{Q}_{ijk}^v is the quadrifocal tensor for the extended cameras and has the following form:

$$\mathcal{Q}_{ijk}^v = \epsilon_{ipq} \epsilon_{jrs} \epsilon_{ktu} \det \begin{bmatrix} \mathbf{a}^p \\ \mathbf{a}^q \\ \mathbf{b}^r \\ \mathbf{b}^s \\ \mathbf{c}^t \\ \mathbf{c}^u \\ \mathbf{d}^v \end{bmatrix} \quad (6.13)$$

where \mathbf{a}^i denotes the i th row of \mathbf{P} , \mathbf{b}^i denotes the i th row of \mathbf{P}' , \mathbf{c}^i denotes the i th row of \mathbf{P}'' and \mathbf{d}^i denotes the i th row of \mathbf{P}''' respectively. The quadrifocal tensor \mathcal{Q}_{ijk}^v is $3 \times 3 \times 3 \times 3$ and has 81 entries. Since all the third rows of the extended affine camera matrices are $[0,0,0,0,0,1]$, many zero entries arise in \mathcal{Q}_{ijk}^v . As a result, $\mathcal{Q}_{133}^1, \mathcal{Q}_{133}^2, \mathcal{Q}_{233}^1, \mathcal{Q}_{233}^2, \mathcal{Q}_{313}^1, \mathcal{Q}_{313}^2, \mathcal{Q}_{323}^1, \mathcal{Q}_{323}^2, \mathcal{Q}_{331}^1, \mathcal{Q}_{331}^2, \mathcal{Q}_{332}^1, \mathcal{Q}_{332}^2, \mathcal{Q}_{333}^1, \mathcal{Q}_{333}^2, \mathcal{Q}_{333}^3$ are non-zero entries and thus we have only 14 free parameters in \mathcal{Q}_{ijk}^v except a scale ambiguity. On the other hand, (6.12) provides us 3 linear equations on \mathcal{Q}_{ijk}^v , but only 2 of them are linearly independent. Thus, at least 7 corresponding points are required to compute \mathcal{Q}_{ijk}^v from images linearly.

6.4.1 Experiment

6.4.1.1 View Transfer

We next show two view transfer experiments by using synthetic images.

Figure 6.1 shows a 3D configuration of 4 moving cameras and a moving point. The black points show the viewpoints of four cameras, $\mathbf{C}_1, \mathbf{C}_2, \mathbf{C}_3$ and \mathbf{C}_4 , with B-spline motions which consist of two B-spline segments. The curvilinear motions of these four cameras are different and unknown. The black curve shows a locus of a moving point \mathbf{S} . Figure 6.2 (a), (b),(c) and (d) show image motions of \mathbf{S} viewed from $\mathbf{C}_1, \mathbf{C}_2, \mathbf{C}_3$ and \mathbf{C}_4 respectively. Note, the original locus of \mathbf{S} is closed in the 3D space as shown in Figure 6.1, but its loci in images are not closed as shown in Figure 6.2 because of the camera motions. We added Gaussian image noises with the standard deviation of 1 pixel to all the points on the loci in images. The 7 black points on the black loci and the 7 white points on the white loci in Figure 6.2 (a), (b), (c) and (d) are used to compute the two quadrifocal tensors on these four moving cameras with two B-spline motions. The quadrifocal tensors are used to recover the image motion in \mathbf{C}_1 from image motions in $\mathbf{C}_2, \mathbf{C}_3$ and \mathbf{C}_4 . Figure 6.3 (b) shows the recovered result. The black

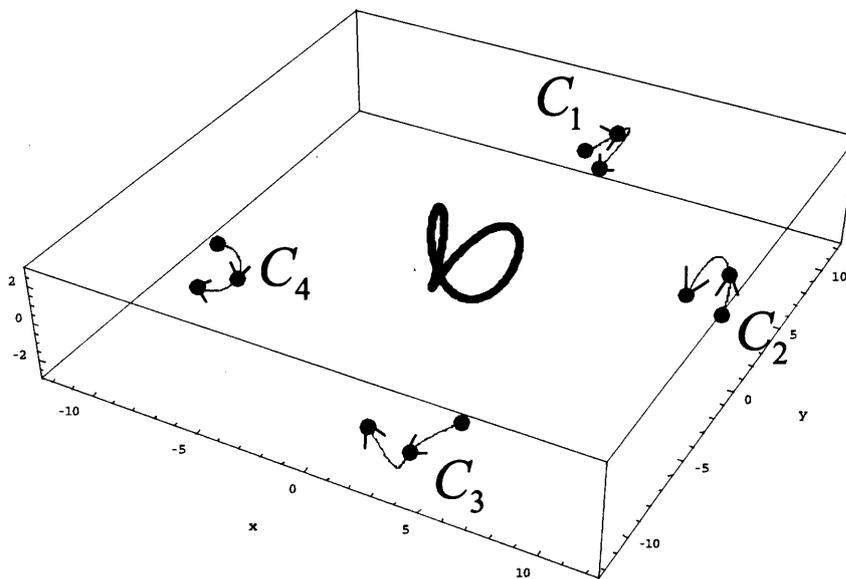


Figure 6.1: Four curvilinear motion cameras and a moving point in the 3D space. Each camera motion consists of two B-spline segments.

curve shows the real trajectory, and the white curve shows the computed motion. The average error of the recovered image motion is 6.03 pixels.

Another view transfer experiment is also done. Figure 6.4 shows the 3D configuration of 4 moving cameras and a moving point. The black points show the viewpoints of four cameras, C_1 , C_2 , C_3 and C_4 . Each camera undergoes two-segment B-spline curve motion. The curvilinear motions of these four cameras are different and unknown. The black curve shows the trajectory of a moving point. Figure 6.5 (a), (b), (c) and (d) show image motions viewed from C_1 , C_2 , C_3 and C_4 respectively. We added Gaussian image noises with the standard deviation of 1 pixel to all the image motions. The 7 black points on the black loci and the 7 white points on the white loci in Figure 6.5 (a), (b), (c) and (d) are used to compute the two quadrifocal tensors on these four moving cameras with two B-spline motions. The quadrifocal tensors are used to recover the image motion in C_1 from image motions in C_2 , C_3 and C_4 . Figure 6.6 (b) shows the recovered result. The black curve shows the real trajectory, and the white curve shows the computed motion. The average error of the recovered image motion is 4.51 pixels.

6.4.1.2 Stability Evaluation

We next show the stability of extracted quadrifocal tensors under extended projections. For evaluating the extracted quadrifocal tensors, we computed reprojection errors derived from the quadrifocal tensors. The reprojection error is defined as: $\frac{1}{N} \sum_{i=1}^N d(\mathbf{m}_i, \hat{\mathbf{m}}_i)$, where $d(\mathbf{m}_i, \hat{\mathbf{m}}_i)$ denotes a distance between a true point \mathbf{m}_i and

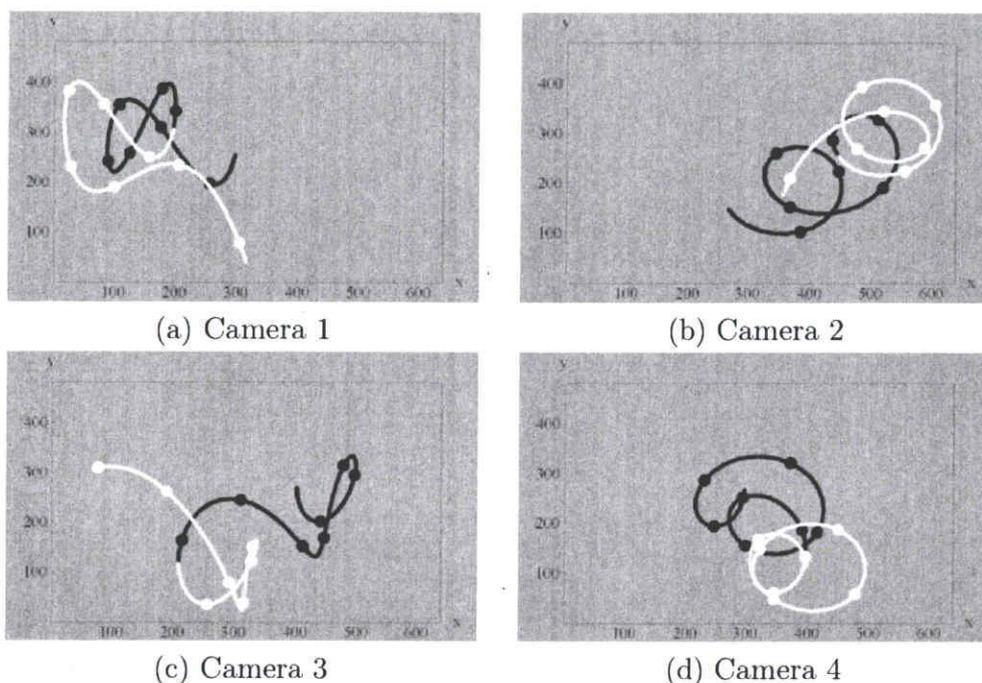


Figure 6.2: Figure (a), (b), (c) and (d) show four views of the motion in camera 1, 2, 3 and 4. The black curves represent the image motions when the cameras undergo the first curvilinear motions. The white curves correspond to the second camera motions. The 7 black points on each black loci and the 7 white points on each white loci in Figure (a), (b), (c) and (d) are used to compute the two quadrifocal tensors.

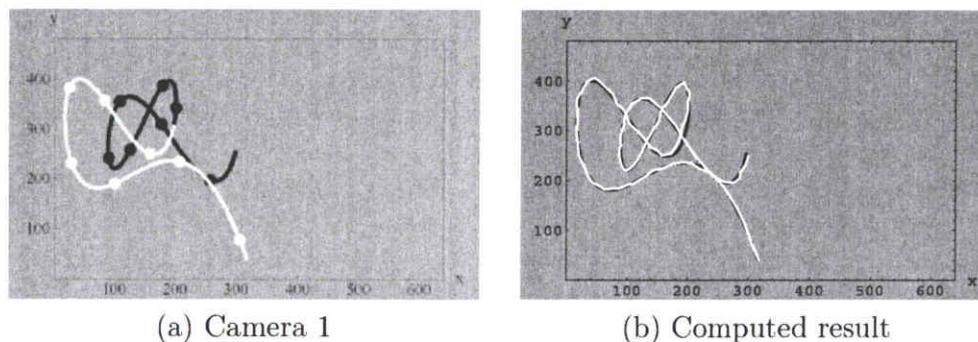


Figure 6.3: View transfer experiment 1. Figure (a) show the view of the motion in camera 1. The black curves represent the image motions when the cameras undergo the first curvilinear motions. The white curves correspond to the second camera motions. The white curve in (b) shows image motions computed by the extended quadrifocal tensors in the image plane of camera 1. The black curve is the true value.

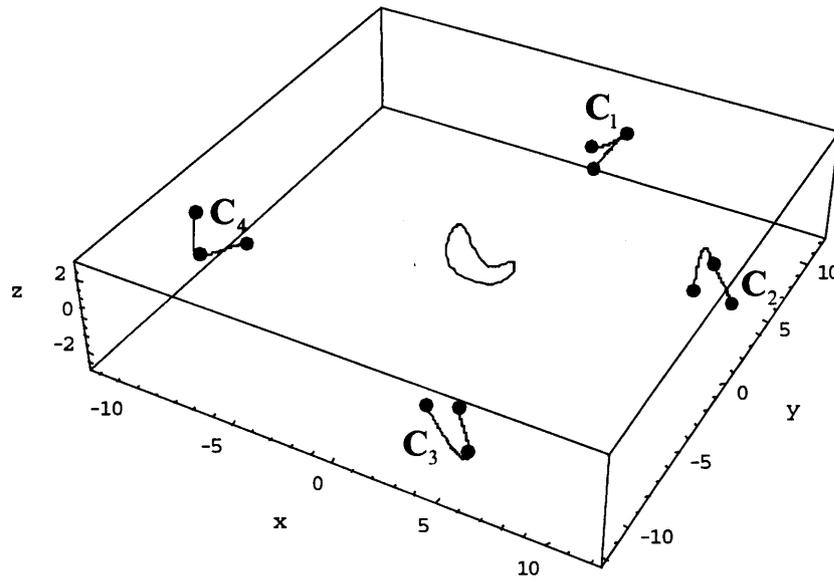


Figure 6.4: Four curvilinear motion cameras and a moving point in the 3D space. Each camera motion consists of two B-spline segments.

a point $\hat{\mathbf{m}}_i$ recovered from the quadrifocal tensor. We increased the number of corresponding points used for computing quadrifocal tensors in four views from 7 to 20, and evaluated the reprojection errors. Camera trajectories and 3D point motions are randomly generated for 1000 times by changing the control points of the B-spline curves. For each camera trajectory and 3D point, the images motions are generated 100 times by adding Gaussian noise with the standard deviation of 1 pixel. Figure 6.7 shows the relationship between the number of corresponding points and the reprojection errors. As we can see, the stability is obviously improved by using a few more points than the minimum number of corresponding points.

6.5 Summary

In this chapter, we introduced B-Spline curve for representing camera motions and derived the multiple view geometry for multiple cameras, whose motion trajectories are described by B-Spline curves.

As we know, in the mathematical field of numerical analysis, B-spline curves are very useful for representing arbitrary 3D shapes with small number of control points. And it can overcome the main problem with Bezier curves, that is their lack of local control, and, when the degree of Bezier curve is large, the multiple view geometry will become very complex, uncomputable. Hence, in this chapter we make use of cubic B-spline curves to describe the arbitrary 3D motions of cameras.

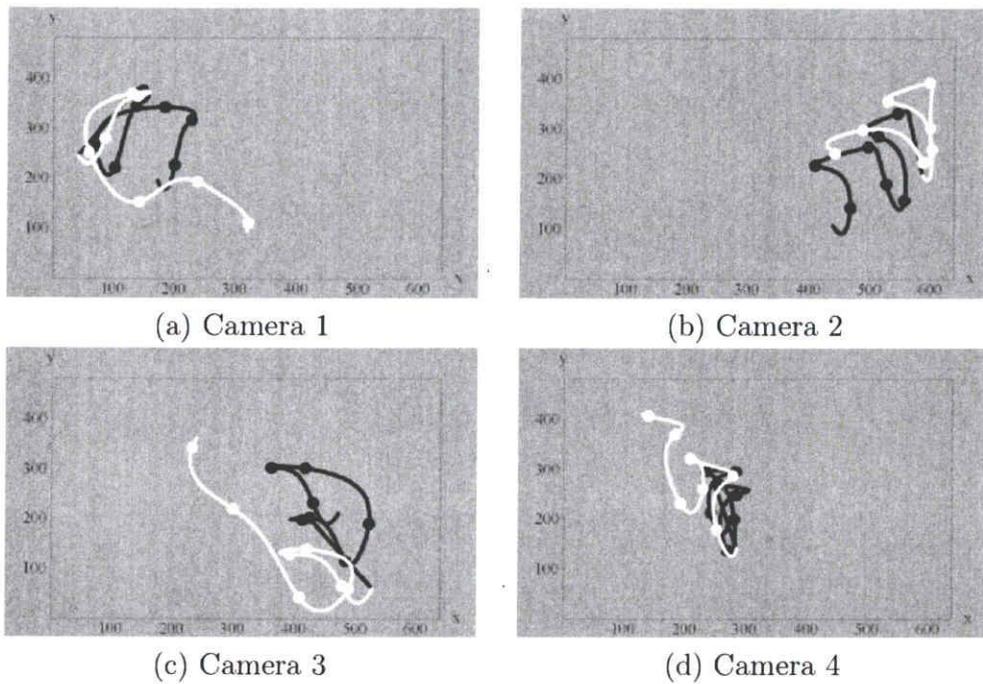


Figure 6.5: Figure (a), (b), (c) and (d) show four views of the motion in camera 1, 2, 3 and 4. The black curves represent the image motions when the cameras undergo the first curvilinear motions. The white curves correspond to the second camera motions. The 7 black points on each black loci and the 7 white points on each white loci in Figure (a), (b), (c) and (d) are used to compute the two quadrifocal tensors.

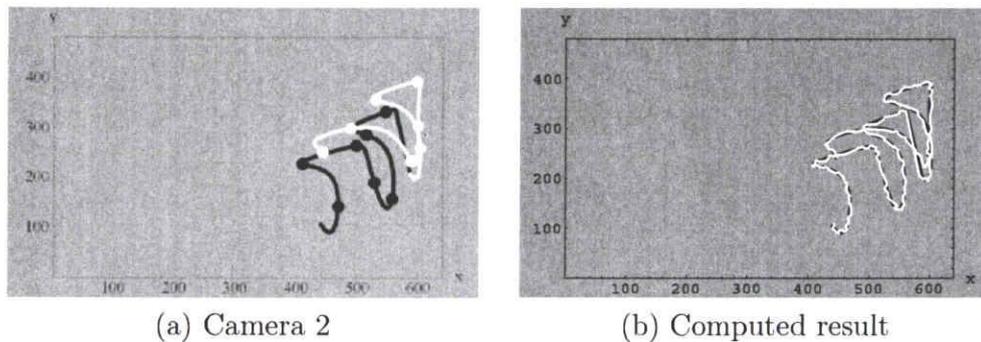


Figure 6.6: View transfer experiment 2. Figure (a) show the view of the motion in camera 2. The black curves represent the image motions when the cameras undergo the first curvilinear motions. The white curves correspond to the second camera motions. The white curve in (b) shows image motions computed by the extended quadrifocal tensors in the image plane of camera 1. The black curve is the true value.

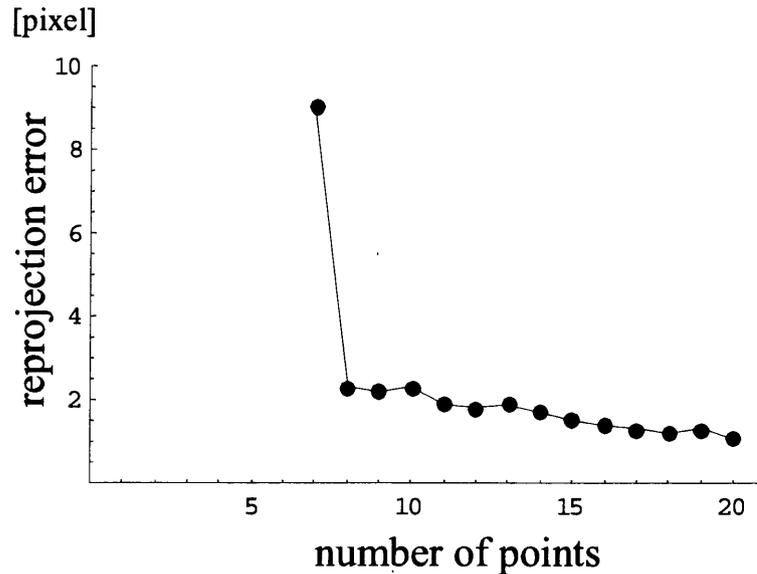


Figure 6.7: The relationship between the number of corresponding points used for computing quadrifocal tensors and the reprojection errors. Camera trajectories and 3D point motions are randomly generated for 1000 times by changing the control points of the B-spline curves.

We provided the definition of the B-spline curve. Especially, we took cubic B-spline curve as an instance to represent the trajectory of the cameras. B-spline curve is a kind of piecewise curve. Although the multiple view geometry corresponding to each segment of cubic B-spline curve motions is same as the case of Bezier curve described in chapter 5, the camera motions could be more complex and less control points required if the camera motions are represented by B-spline curves. For example, a 2-segment cubic B-spline curve is smooth, second-order differentiable and is defined by 5 control points, while two cubic successive Bezier curves are not second-order differentiable and determined by 7 control points.

We showed the result from the synthetic experiment. We can see that even if all the cameras undergo complex curvilinear motions, the view transfer still can be realized by using the dynamic multiple view geometry.

Chapter 7

Computing Dynamic Multiple View Geometry in 4D space from Mutual Projections of Multiple Cameras

7.1 Mutual Projections of Multiple Cameras

So-called mutual projections of multiple cameras describe the case where some cameras are projected to other cameras. Under such configuration, we can directly use the actual projection of cameras as the approximation of epipoles and derive the multiple view geometry more stably with less corresponding points [75].

However, the known mutual projection method is applied to the traditional multiple view geometry, in which all the cameras are assumed to be static. That means all the projections of cameras, the epipoles, are single points.

Whereas the dynamic multiple view geometry in 4D space proposed in chapter 3 describes the relationship between images viewed from multiple translational cameras. In this case, a set of epipoles derived from a translational camera is no longer a single point but a line. Thus, the problem becomes much more complex than the case of traditional mutual projections of multiple cameras. Then, how about the case of mutual projections in 4D space? In this chapter, we show the answer to this question.

7.2 Computing Dynamic Multiple View Geometry in 4D Space from Mutual Projections

Let us consider the mutual projections of three moving cameras in sequential images. Then, we can derive at most 3 pairs of epipoles: $\{\mathbf{e}_{21}, \mathbf{e}_{31}\}$, $\{\mathbf{e}_{12}, \mathbf{e}_{32}\}$ and $\{\mathbf{e}_{13}, \mathbf{e}_{23}\}$ at an instance. Here, $\{\mathbf{e}_{21}, \mathbf{e}_{31}\}$, for example, denote epipoles which can be approximately regarded as the projections of camera 1 to camera 2 and 3.

At time t , the epipole $\mathbf{e}_{21}(t)$ is a point in view 2, which corresponds to any point

in view 1. Also, the epipole $\mathbf{e}_{31}(t)$ at time t is a point in view 3, which corresponds to any point in view 1. This means, by substituting $\mathbf{e}_{21}(t)$ and $\mathbf{e}_{31}(t)$ into \mathbf{x}' and \mathbf{x}'' in (4.12), we have the following trilinear relationship which must hold for any point \mathbf{m} in view 1.

$$m^i e_{21}^j(t) e_{31}^k(t) \epsilon_{krv} \mathcal{T}_{ij}^r = 0_v \quad \forall \mathbf{m} \quad (7.1)$$

Since (7.1) must hold for any \mathbf{m} , the remaining part, $e_{21}^j(t) e_{31}^k(t) \epsilon_{krv} \mathcal{T}_{ij}^r$, must be zero tensor. The similar discussions hold for the pairs of epipoles $\{\mathbf{e}_{12}, \mathbf{e}_{32}\}$ and $\{\mathbf{e}_{13}, \mathbf{e}_{23}\}$. Thus, we have the following relationships between epipoles and trifocal tensors:

$$e_{21}^j(t) e_{31}^k(t) \epsilon_{krv} \mathcal{T}_{ij}^r = 0_{iv} \quad (7.2)$$

$$e_{12}^i(t) e_{32}^k(t) \epsilon_{krv} \mathcal{T}_{ij}^r = 0_{jv} \quad (7.3)$$

$$e_{13}^i(t) e_{23}^j(t) \mathcal{T}_{ij}^r = 0^r \quad (7.4)$$

Although (7.2) provides us 9 linear equations on trifocal tensor, only 6 of them are linearly independent. And if we use $\{\mathbf{e}_{21}, \mathbf{e}_{31}\}$ at N different time, the number of independent equations derived from (7.2) is not always $6N$. The same thing happens to other two pairs of epipoles. Furthermore, if we combine some pairs of epipoles and use N of them respectively to compute trifocal tensor, the results are very different. In the following subsections, let us consider the number of independent equations and the minimum number of corresponding points required for computing trifocal tensors under mutual projection of cameras in 4D space by using one, two and all three epipole pairs respectively.

7.2.1 Using One Epipole Pair

7.2.1.1 Using Epipole Pair $\{\mathbf{e}_{21}, \mathbf{e}_{31}\}$ or $\{\mathbf{e}_{12}, \mathbf{e}_{32}\}$

We first consider why (7.2) has only 6 independent equations. In general, a tensor $m^i \epsilon_{ijk}$ represents three lines which go through a point \mathbf{m} . Thus $e_{31}^k(t) \epsilon_{krv}$ in (7.2) represents three epipolar lines in view 3, which go through an epipole $\mathbf{e}_{31}(t)$. So, (7.2) describes relationships between epipole $\mathbf{e}_{21}(t)$ in view 2 and epipolar lines $l''(t)$ which go through $\mathbf{e}_{31}(t)$ in view 3 as follows:

$$e_{21}^j(t) l_k''(t) \mathcal{T}_{ij}^k = 0_i \quad (7.5)$$

Since (7.5) must hold for any point \mathbf{m} in view 1, (7.5) is considered as a point-point-line incidence on any point \mathbf{m} in view 1, epipole $\mathbf{e}_{21}(t)$ and any epipolar line $l''(t)$ which goes through $\mathbf{e}_{31}(t)$ as follows:

$$m^i e_{21}^j(t) l_k''(t) \mathcal{T}_{ij}^k = 0 \quad (7.6)$$

For finding the number of independent equations in (7.2) in sequential images, we need to count the number of independent incidence relations described by (7.6) when we use N pairs of $\{\mathbf{e}_{21}(t), \mathbf{e}_{31}(t)\}$, $(t = t_1, \dots, t_N)$.

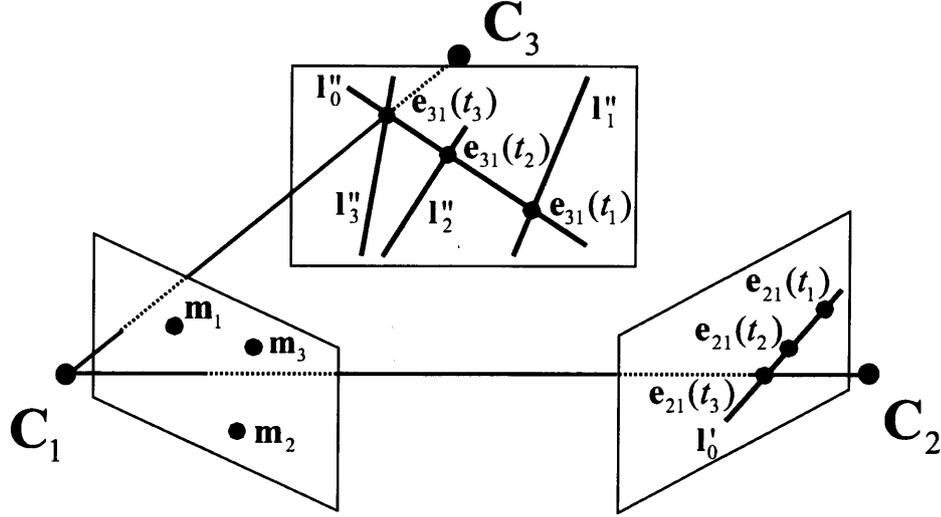


Figure 7.1: The basis points, basis lines and epipole lines for representing incidence relations in three views. $\{\mathbf{m}_1, \mathbf{m}_2, \mathbf{m}_3\}$ show three basis points in view 1. l'_0 shows epipole line which goes through \mathbf{e}_{21} in view 2. l''_0 shows epipole line going through \mathbf{e}_{31} in view 3. $\{l''_1, l''_2, l''_3\}$ show three basis lines in view 3.

We note that any point \mathbf{m} in view 1 can be described by a linear combination of three basis points \mathbf{m}_1 , \mathbf{m}_2 and \mathbf{m}_3 . And any epipolar line $l''(t)$ which goes through $\mathbf{e}_{31}(t)$ in view 3 can be described by a linear combination of two basis lines. Since $\mathbf{e}_{31}(t)$ are collinear, one of these two basis lines can be a line which goes through all the epipoles $\mathbf{e}_{31}(t)$ ($t = t_1, \dots, t_N$) as shown in Figure 7.1. We call the line epipole line and denote it by l''_0 . Suppose l''_1 , l''_2 and l''_3 go through $\mathbf{e}_{31}(t_1)$, $\mathbf{e}_{31}(t_2)$ and $\mathbf{e}_{31}(t_3)$ respectively as shown in Figure 7.1. Then, if we have one pair of epipoles $\{\mathbf{e}_{21}(t), \mathbf{e}_{31}(t)\}$ at t_1 , any incidence relation represented by (7.5) at time t_1 can be described by a linear combination of the following 6 basis incidence relations:

$$\begin{aligned}
 m_1^i e_{21}^j(t_1) l''_{0k} \mathcal{T}_{ij}^k &= 0 & m_1^i e_{21}^j(t_1) l''_{1k} \mathcal{T}_{ij}^k &= 0 \\
 m_2^i e_{21}^j(t_1) l''_{0k} \mathcal{T}_{ij}^k &= 0 & m_2^i e_{21}^j(t_1) l''_{1k} \mathcal{T}_{ij}^k &= 0 \\
 m_3^i e_{21}^j(t_1) l''_{0k} \mathcal{T}_{ij}^k &= 0 & m_3^i e_{21}^j(t_1) l''_{1k} \mathcal{T}_{ij}^k &= 0
 \end{aligned} \tag{7.7}$$

Therefore, we have only 6 linearly independent equations in (7.2).

For simplification of (7.7), we define a new notation for describing all the 6 equations as follows:

$$\left\{ \begin{array}{c} \mathbf{m}_1 \\ \mathbf{m}_2 \\ \mathbf{m}_3 \end{array} \right\} \{ \mathbf{e}_{21}(t_1) \} \left\{ \begin{array}{c} l''_0 \\ l''_1 \end{array} \right\} \tag{7.8}$$

The number of equations is the product of the number of rows of each column. So, the number of equations in (7.8) is $3 \times 1 \times 2 = 6$.

Thus, if we have a pair of epipoles $\{\mathbf{e}_{21}(t), \mathbf{e}_{31}(t)\}$ at two different time, t_1 and t_2 , then we can derive the following equations:

$$\begin{Bmatrix} \mathbf{m}_1 \\ \mathbf{m}_2 \\ \mathbf{m}_3 \end{Bmatrix} \{\mathbf{e}_{21}(t_1)\} \begin{Bmatrix} \mathbf{l}_0'' \\ \mathbf{l}_1'' \end{Bmatrix}, \quad \begin{Bmatrix} \mathbf{m}_1 \\ \mathbf{m}_2 \\ \mathbf{m}_3 \end{Bmatrix} \{\mathbf{e}_{21}(t_2)\} \begin{Bmatrix} \mathbf{l}_0'' \\ \mathbf{l}_2'' \end{Bmatrix} \quad (7.9)$$

Then, the number of equations in (7.9) is $3 \times 1 \times 2 + 3 \times 1 \times 2 = 12$. It means that by using a pair of epipoles $\{\mathbf{e}_{21}(t), \mathbf{e}_{31}(t)\}$ at two different time, there exist 12 independent equations.

How about the case of 3 different time, t_1 , t_2 and t_3 ? At time t_3 , we have:

$$\begin{Bmatrix} \mathbf{m}_1 \\ \mathbf{m}_2 \\ \mathbf{m}_3 \end{Bmatrix} \{\mathbf{e}_{21}(t_3)\} \begin{Bmatrix} \mathbf{l}_0'' \\ \mathbf{l}_3'' \end{Bmatrix} \quad (7.10)$$

Since $\mathbf{e}_{21}(t_1)$, $\mathbf{e}_{21}(t_2)$ and $\mathbf{e}_{21}(t_3)$ are collinear, $\mathbf{e}_{21}(t_3)$ can be written by the linear combination of $\mathbf{e}_{21}(t_1)$ and $\mathbf{e}_{21}(t_2)$ as:

$$\mathbf{e}_{21}(t_3) = c_1 \mathbf{e}_{21}(t_1) + c_2 \mathbf{e}_{21}(t_2) \quad (7.11)$$

Then (7.10) can be described as follows:

$$\begin{Bmatrix} \mathbf{m}_1 \\ \mathbf{m}_2 \\ \mathbf{m}_3 \end{Bmatrix} \{(c_1 \mathbf{e}_{21}(t_1) + c_2 \mathbf{e}_{21}(t_2))\} \begin{Bmatrix} \mathbf{l}_0'' \\ \mathbf{l}_3'' \end{Bmatrix} \quad (7.12)$$

Now, since

$$\begin{Bmatrix} \mathbf{m}_1 \\ \mathbf{m}_2 \\ \mathbf{m}_3 \end{Bmatrix} \{(c_1 \mathbf{e}_{21}(t_1) + c_2 \mathbf{e}_{21}(t_2))\} \{\mathbf{l}_0''\} \quad (7.13)$$

can be described by a linear combination of

$$\begin{Bmatrix} \mathbf{m}_1 \\ \mathbf{m}_2 \\ \mathbf{m}_3 \end{Bmatrix} \{\mathbf{e}_{21}(t_1)\} \{\mathbf{l}_0''\} \quad \text{and} \quad \begin{Bmatrix} \mathbf{m}_1 \\ \mathbf{m}_2 \\ \mathbf{m}_3 \end{Bmatrix} \{\mathbf{e}_{21}(t_2)\} \{\mathbf{l}_0''\},$$

it is linearly dependent with (7.9). Therefore, only

$$\begin{Bmatrix} \mathbf{m}_1 \\ \mathbf{m}_2 \\ \mathbf{m}_3 \end{Bmatrix} \{(c_1 \mathbf{e}_{21}(t_1) + c_2 \mathbf{e}_{21}(t_2))\} \{\mathbf{l}_3''\} \quad (7.14)$$

is linearly independent, and then we have only 3 independent equations in (7.10). Thus, we find that from a pair of $\{\mathbf{e}_{21}(t), \mathbf{e}_{31}(t)\}$ at 3 different time, 15 independent equations can be derived.

At time t_4 , it seems that we have another independent equation as follows:

$$\begin{pmatrix} \mathbf{m}_1 \\ \mathbf{m}_2 \\ \mathbf{m}_3 \end{pmatrix} \{\mathbf{e}_{21}(t_4)\} \{\mathbf{l}_4''\} \quad (7.15)$$

However, this is not the case. Since any line on the plane can be described by a set of three basis lines, \mathbf{l}_3'' can be described by \mathbf{l}_0'' , \mathbf{l}_1'' and \mathbf{l}_2'' as follows:

$$\mathbf{l}_3'' = d_1\mathbf{l}_0'' + d_2\mathbf{l}_1'' + d_3\mathbf{l}_2''$$

Then, at t_3 , (7.14) can be represented as:

$$\begin{pmatrix} \mathbf{m}_1 \\ \mathbf{m}_2 \\ \mathbf{m}_3 \end{pmatrix} \{c_1\mathbf{e}_{21}(t_1) + c_2\mathbf{e}_{21}(t_2)\} \{d_1\mathbf{l}_0'' + d_2\mathbf{l}_1'' + d_3\mathbf{l}_2''\} \quad (7.16)$$

Simplifying (7.16), we have:

$$\begin{pmatrix} \mathbf{m}_1 \\ \mathbf{m}_2 \\ \mathbf{m}_3 \end{pmatrix} \left\{ \mathbf{e}_{21}(t_2)\mathbf{l}_1'' + \frac{c_1d_3}{c_2d_2}\mathbf{e}_{21}(t_1)\mathbf{l}_2'' \right\} \quad (7.17)$$

Similarly, at t_4 , (7.15) can also be described as:

$$\begin{pmatrix} \mathbf{m}_1 \\ \mathbf{m}_2 \\ \mathbf{m}_3 \end{pmatrix} \{c_3\mathbf{e}_{21}(t_1) + c_4\mathbf{e}_{21}(t_2)\} \{d_4\mathbf{l}_0'' + d_5\mathbf{l}_1'' + d_6\mathbf{l}_2''\} \quad (7.18)$$

and their simplified forms are:

$$\begin{pmatrix} \mathbf{m}_1 \\ \mathbf{m}_2 \\ \mathbf{m}_3 \end{pmatrix} \left\{ \mathbf{e}_{21}(t_2)\mathbf{l}_1'' + \frac{c_3d_6}{c_4d_5}\mathbf{e}_{21}(t_1)\mathbf{l}_2'' \right\} \quad (7.19)$$

We find that (7.17) and (7.19) are very similar, only the coefficient $\frac{c_1d_3}{c_2d_2}$ and $\frac{c_3d_6}{c_4d_5}$ are different, but in fact they are equal and relate to the initial position of camera 1, camera motions, camera matrices, t_1 and t_2 , all of which are constants. As a result, there is no independent equation at t_4 . The same thing happens at time t_5 , t_6 , \dots . Thus when we use 3 or more pairs of $\{\mathbf{e}_{21}(t), \mathbf{e}_{31}(t)\}$, the number of independent equations we can derive is 15.

Table 7.1: The number of independent equations derived by using $\{\mathbf{e}_{21}(t), \mathbf{e}_{31}(t)\}$ or $\{\mathbf{e}_{12}(t), \mathbf{e}_{32}(t)\}$ for N time ($t = t_1, \dots, t_N$), and the number of corresponding points required for computing trifocal tensors in each case of mutual projections of cameras. 3^* denotes 3 or greater than 3.

$N \times \{\mathbf{e}_{21}(t), \mathbf{e}_{31}(t)\}$ or $\{\mathbf{e}_{12}(t), \mathbf{e}_{32}(t)\}$	# of independent equations	# of corresponding points required
1	6	10
2	12	7
3^*	15	6

Since N sets of corresponding points provide us $2N$ linearly independent equations, the following inequality must hold for computing 26 free parameters of the trifocal tensor \mathcal{T}_{ij}^k , if we have a pair of epipoles at time t_1 :

$$2N + 6 \geq 26$$

Thus we need 10 corresponding points. Similarly, if we have a pair of $\{\mathbf{e}_{21}(t), \mathbf{e}_{31}(t)\}$ at time t_1 and t_2 , the number of corresponding points required for computing \mathcal{T}_{ij}^k is $(26 - 12)/2 = 7$, and if we have a pair of $\{\mathbf{e}_{21}(t), \mathbf{e}_{31}(t)\}$ at time t_1, t_2 and t_3 , we require 6 corresponding points.

The case of epipole pair $\{\mathbf{e}_{12}, \mathbf{e}_{32}\}$ is almost same as $\{\mathbf{e}_{21}, \mathbf{e}_{31}\}$. We summarize the number of corresponding points required for computing trifocal tensors in each case of mutual projections of cameras in Table 7.1.

7.2.1.2 Using Epipole Pair $\{\mathbf{e}_{13}, \mathbf{e}_{23}\}$

The case of epipole pair $\{\mathbf{e}_{13}, \mathbf{e}_{23}\}$ is much simpler than other two epipole pairs and the analysis process can refer to the previous section. The number of independent equations by using N pairs of $\{\mathbf{e}_{13}, \mathbf{e}_{23}\}$, and the number of corresponding points required for computing trifocal tensors in each case of mutual projections of cameras are summarized in Table 7.2.

7.2.2 Using Two Epipole Pairs

7.2.2.1 Using Epipole Pair $\{\mathbf{e}_{21}, \mathbf{e}_{31}\}$ and $\{\mathbf{e}_{12}, \mathbf{e}_{32}\}$

(1) $1 \times \{\mathbf{e}_{21}(t), \mathbf{e}_{31}(t)\} + 1 \times \{\mathbf{e}_{12}(t), \mathbf{e}_{32}(t)\}$

Suppose at time t_1 and t_2 we have one pair of $\{\mathbf{e}_{21}(t), \mathbf{e}_{31}(t)\}$ and $\{\mathbf{e}_{12}(t), \mathbf{e}_{32}(t)\}$ respectively:

$$e_{21}^j(t_1)e_{31}^k(t_1)\epsilon_{krv}\mathcal{T}_{ij}^r = 0_{iv} \quad (7.20)$$

$$e_{12}^i(t_2)e_{32}^k(t_2)\epsilon_{krv}\mathcal{T}_{ij}^r = 0_{jv} \quad (7.21)$$

Table 7.2: The number of independent equations derived by using $\{\mathbf{e}_{13}(t), \mathbf{e}_{23}(t)\}$ for N time ($t = t_1, \dots, t_N$), and the number of corresponding points required for computing trifocal tensors in each case of mutual projections of cameras. 3^* denotes 3 or greater than 3.

$N \times \{\mathbf{e}_{13}(t), \mathbf{e}_{23}(t)\}$	# of independent eq.	# of points required
1	3	12
2	6	10
3^*	9	9

We have known that (7.20) can be written as:

$$m^i e_{21}^j(t_1) l_k''(t_1) \mathcal{T}_{ij}^k = 0 \quad (7.22)$$

where \mathbf{m} denotes any point in view 1, and $l''(t_1)$ denotes any epipolar line which goes through $\mathbf{e}_{31}(t_1)$ in view 3. Since (7.21) is quite similar to (7.20), (7.21) can also be described as follows:

$$e_{12}^i(t_2) m'^j l_k''(t_2) \mathcal{T}_{ij}^k = 0 \quad (7.23)$$

where \mathbf{m}' denotes any point in view 2, and $l''(t_2)$ denotes any epipolar line which goes through $\mathbf{e}_{32}(t_2)$ in view 3. As shown in Figure 7.2, l_1'' and l_2'' go through $\mathbf{e}_{31}(t_1)$. l_2'' and l_3'' go through $\mathbf{e}_{32}(t_2)$. The three lines do not coincide. Then (7.22) can be written as:

$$\left\{ \begin{array}{c} \mathbf{m}_1 \\ \mathbf{m}_2 \\ \mathbf{m}_3 \end{array} \right\} \{ \mathbf{e}_{21}(t_1) \} \left\{ \begin{array}{c} l_1'' \\ l_2'' \end{array} \right\} \quad (7.24)$$

which denotes 6 independent equations. Similarly, (7.23) can also be written as:

$$\{ \mathbf{e}_{12}(t_2) \} \left\{ \begin{array}{c} \mathbf{m}'_1 \\ \mathbf{m}'_2 \\ \mathbf{m}'_3 \end{array} \right\} \left\{ \begin{array}{c} l_3'' \\ l_2'' \end{array} \right\} \quad (7.25)$$

However, (7.25) provides us less than 6 independent equations because of (7.24). Let us explain it in detail.

(7.25) can be described as two parts:

$$\{ \mathbf{e}_{12}(t_2) \} \left\{ \begin{array}{c} \mathbf{m}'_1 \\ \mathbf{m}'_2 \\ \mathbf{m}'_3 \end{array} \right\} \{ l_3'' \} \quad (7.26)$$

$$\{ \mathbf{e}_{12}(t_2) \} \left\{ \begin{array}{c} \mathbf{m}'_1 \\ \mathbf{m}'_2 \\ \mathbf{m}'_3 \end{array} \right\} \{ l_2'' \} \quad (7.27)$$

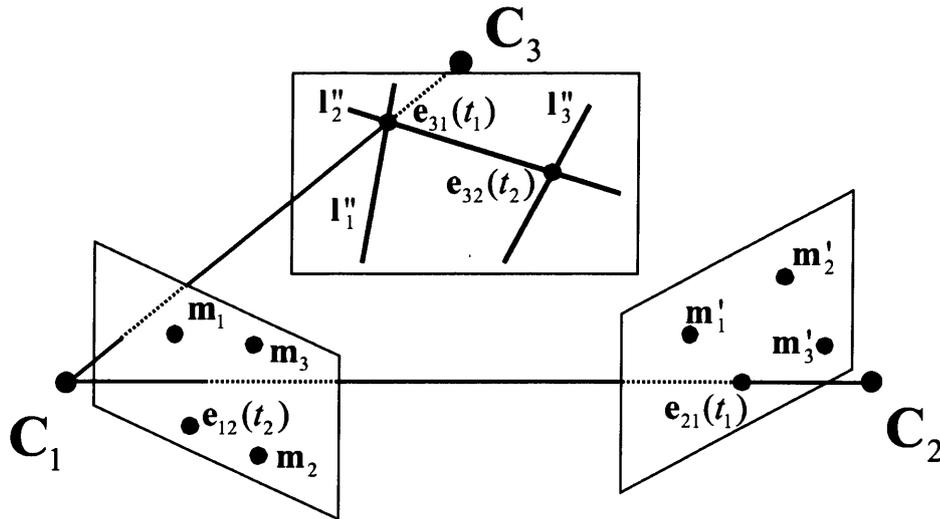


Figure 7.2: The basis points and lines for representing incidence relations in three views. $\{\mathbf{m}_1, \mathbf{m}_2, \mathbf{m}_3\}$ show three basis points in view 1. $\{\mathbf{m}'_1, \mathbf{m}'_2, \mathbf{m}'_3\}$ show three basis points in view 2. $\{\mathbf{l}''_1, \mathbf{l}''_2, \mathbf{l}''_3\}$ show three basis lines in view 3.

Since (7.26) is independent relative to (7.24), (7.26) brings us 3 independent equations. But (7.27) is dubious.

If we consider one of the basis points \mathbf{m}'_3 as $\mathbf{e}_{21}(t_1)$, (7.27) becomes to:

$$\{\mathbf{e}_{12}(t_2)\} \left\{ \begin{array}{c} \mathbf{m}'_1 \\ \mathbf{m}'_2 \\ \mathbf{e}_{21}(t_1) \end{array} \right\} \{\mathbf{l}''_2\} \quad (7.28)$$

where

$$\{\mathbf{e}_{12}(t_2)\} \left\{ \begin{array}{c} \mathbf{m}'_1 \\ \mathbf{m}'_2 \end{array} \right\} \{\mathbf{l}''_2\} \quad (7.29)$$

are independent with (7.24). However, the third equation

$$\{\mathbf{e}_{12}(t_2)\} \{\mathbf{e}_{21}(t_1)\} \{\mathbf{l}''_2\} \quad (7.30)$$

is dependent, since $\mathbf{e}_{12}(t_2)$ can be represented by the basis points $\{\mathbf{m}_1, \mathbf{m}_2, \mathbf{m}_3\}$ in view 1. Then (7.30) can be written as:

$$\{c_1\mathbf{m}_1 + c_2\mathbf{m}_2 + c_3\mathbf{m}_3\} \{\mathbf{e}_{21}(t_1)\} \{\mathbf{l}''_2\} \quad (7.31)$$

which is just the linear combination of three equations in (7.24). Then, (7.25) provides us $3 + 2 = 5$ independent equations. Thus, 1 pair of $\{\mathbf{e}_{21}(t), \mathbf{e}_{31}(t)\}$ and 1 pair of $\{\mathbf{e}_{12}(t), \mathbf{e}_{32}(t)\}$ provide us $6 + 5 = 11$ independent equations.

$$(2) \mathbf{1} \times \{\mathbf{e}_{21}(t), \mathbf{e}_{31}(t)\} + \mathbf{2} \times \{\mathbf{e}_{12}(t), \mathbf{e}_{32}(t)\}$$

If we have a pair of $\{\mathbf{e}_{21}(t), \mathbf{e}_{31}(t)\}$ at time t_1 , and a pair of $\{\mathbf{e}_{12}(t), \mathbf{e}_{32}(t)\}$ at time t_2 and t_3 :

$$e_{21}^j(t_1)e_{31}^k(t_1)\epsilon_{krv}\mathcal{I}_{ij}^r = 0_{iv} \quad (7.32)$$

$$e_{12}^i(t_2)e_{32}^k(t_2)\epsilon_{krv}\mathcal{I}_{ij}^r = 0_{jv} \quad (7.33)$$

$$e_{12}^i(t_3)e_{32}^k(t_3)\epsilon_{krv}\mathcal{I}_{ij}^r = 0_{jv} \quad (7.34)$$

their simplified forms we can obtain are as follows:

$$\left\{ \begin{array}{c} \mathbf{m}_1 \\ \mathbf{m}_2 \\ \mathbf{m}_3 \end{array} \right\} \{\mathbf{e}_{21}(t_1)\} \left\{ \begin{array}{c} \mathbf{l}_2'' \\ \mathbf{l}_3'' \end{array} \right\} \quad (7.35)$$

$$\{\mathbf{e}_{12}(t_2)\} \left\{ \begin{array}{c} \mathbf{m}'_1 \\ \mathbf{m}'_2 \\ \mathbf{m}'_3 \end{array} \right\} \left\{ \begin{array}{c} \mathbf{l}_1'' \\ \mathbf{l}_2'' \end{array} \right\} \quad (7.36)$$

$$\{\mathbf{e}_{12}(t_3)\} \left\{ \begin{array}{c} \mathbf{m}'_1 \\ \mathbf{m}'_2 \\ \mathbf{m}'_3 \end{array} \right\} \left\{ \begin{array}{c} \mathbf{l}_1'' \\ \mathbf{l}_3'' \end{array} \right\} \quad (7.37)$$

where \mathbf{l}_2'' and \mathbf{l}_3'' go through $\mathbf{e}_{31}(t_1)$, \mathbf{l}_1'' and \mathbf{l}_2'' go through $\mathbf{e}_{32}(t_2)$, and, \mathbf{l}_1'' and \mathbf{l}_3'' go through $\mathbf{e}_{32}(t_3)$ as shown in Figure 7.3.

The former discussions also hold here, so we can see that (7.35) provides us 6 independent equations, (7.36) and (7.37) brings us 5 independent equations respectively. Thus, a pair of $\{\mathbf{e}_{21}(t), \mathbf{e}_{31}(t)\}$ at **1** time and a pair of $\{\mathbf{e}_{12}(t), \mathbf{e}_{32}(t)\}$ at **2** different time provide us $6 + 5 + 5 = 16$ independent equations.

$$(3) \mathbf{1} \times \{\mathbf{e}_{21}(t), \mathbf{e}_{31}(t)\} + \mathbf{3}^* \times \{\mathbf{e}_{12}(t), \mathbf{e}_{32}(t)\}$$

Here, 3^* denotes 3 or greater than 3. We first consider the case of $\mathbf{1} \times \{\mathbf{e}_{21}(t), \mathbf{e}_{31}(t)\} + \mathbf{3} \times \{\mathbf{e}_{12}(t), \mathbf{e}_{32}(t)\}$, which is the case of $\mathbf{1} \times \{\mathbf{e}_{21}(t), \mathbf{e}_{31}(t)\} + \mathbf{2} \times \{\mathbf{e}_{12}(t), \mathbf{e}_{32}(t)\}$, (7.35), (7.36) and (7.37), plus the following equations:

$$\{\mathbf{e}_{12}(t_4)\} \left\{ \begin{array}{c} \mathbf{m}'_1 \\ \mathbf{m}'_2 \\ \mathbf{m}'_3 \end{array} \right\} \left\{ \begin{array}{c} \mathbf{l}_1'' \\ \mathbf{l}_4'' \end{array} \right\} \quad (7.38)$$

where \mathbf{l}_4'' goes through $\mathbf{e}_{31}(t_1)$ and $\mathbf{e}_{32}(t_4)$ as shown in Figure 7.4.

$\mathbf{1} \times \{\mathbf{e}_{21}(t), \mathbf{e}_{31}(t)\} + \mathbf{2} \times \{\mathbf{e}_{12}(t), \mathbf{e}_{32}(t)\}$ provide us 16 independent equations. Then how many independent equations does (7.38) include?

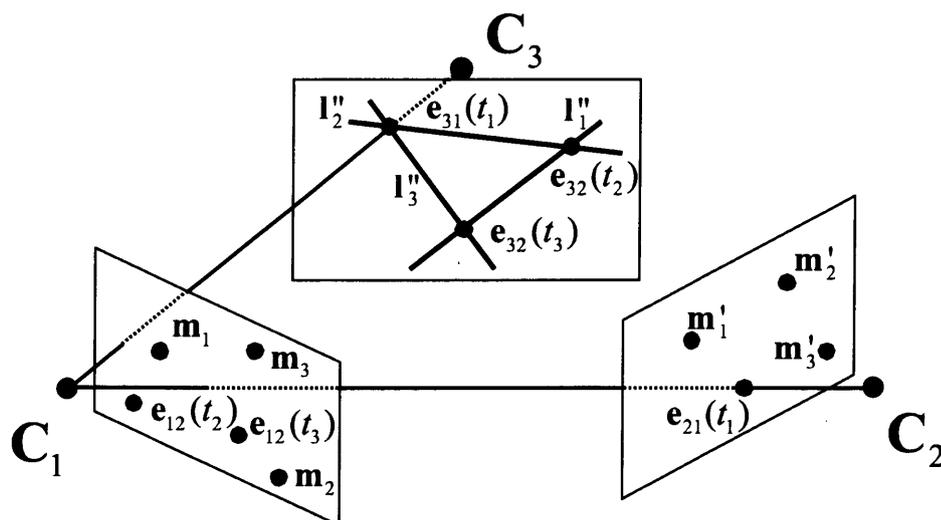


Figure 7.3: The basis points, basis lines and epipole line for representing incidence relations in three views. $\{m_1, m_2, m_3\}$ show three basis points in view 1. $\{m'_1, m'_2, m'_3\}$ show three basis points in view 2. $\{l''_2, l''_3\}$ show two basis lines in view 3. l''_1 shows epipole line which goes through e_{32} .

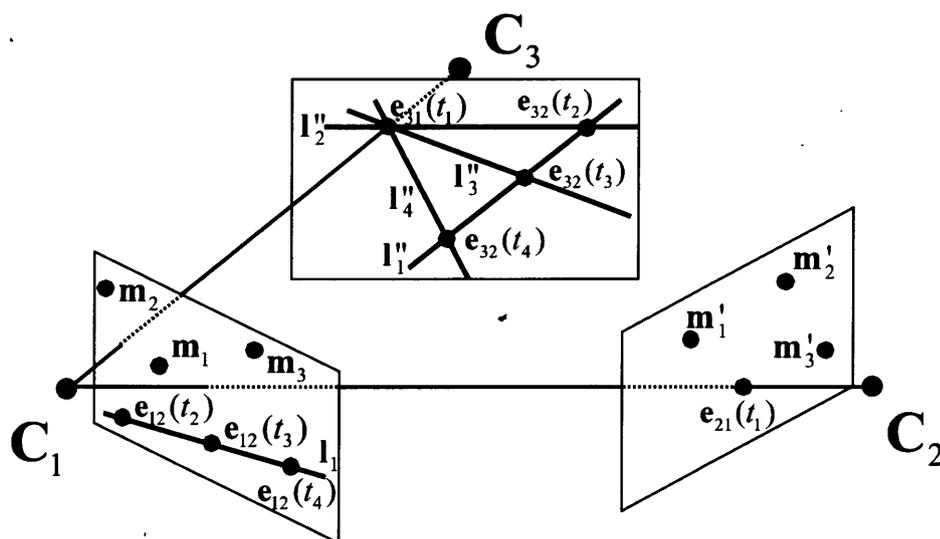


Figure 7.4: The basis points, basis lines and epipole lines for representing incidence relations in three views. $\{m_1, m_2, m_3\}$ show three basis points in view 1. $\{m'_1, m'_2, m'_3\}$ show three basis points in view 2. $\{l''_2, l''_3, l''_4\}$ show three basis lines in view 3. l''_1 shows epipole line which goes through e_{12} . l''_1 shows epipole line which goes through e_{32} .

(7.38) can be written into two parts:

$$\{\mathbf{e}_{12}(t_4)\} \left\{ \begin{array}{c} \mathbf{m}'_1 \\ \mathbf{m}'_2 \\ \mathbf{m}'_3 \end{array} \right\} \{\mathbf{l}''_1\} \quad (7.39)$$

$$\{\mathbf{e}_{12}(t_4)\} \left\{ \begin{array}{c} \mathbf{m}'_1 \\ \mathbf{m}'_2 \\ \mathbf{m}'_3 \end{array} \right\} \{\mathbf{l}''_4\} \quad (7.40)$$

Since $\mathbf{e}_{12}(t_2)$, $\mathbf{e}_{12}(t_3)$ and $\mathbf{e}_{12}(t_4)$ are collinear, $\mathbf{e}_{12}(t_4)$ can be described by $\mathbf{e}_{12}(t_2)$ and $\mathbf{e}_{12}(t_3)$. Then, (7.39) can be changed to:

$$\{c_1\mathbf{e}_{12}(t_2) + c_2\mathbf{e}_{12}(t_3)\} \left\{ \begin{array}{c} \mathbf{m}'_1 \\ \mathbf{m}'_2 \\ \mathbf{m}'_3 \end{array} \right\} \{\mathbf{l}''_1\} \quad (7.41)$$

which can be represented by the combination of some equations from (7.36) and (7.37). So (7.39) does not bring us any independent equation. Then how about (7.40)? If we consider one of the basis points \mathbf{m}'_3 as $\mathbf{e}_{21}(t_1)$, (7.40) can be written as:

$$\{\mathbf{e}_{12}(t_4)\} \left\{ \begin{array}{c} \mathbf{m}'_1 \\ \mathbf{m}'_2 \\ \mathbf{e}_{21}(t_1) \end{array} \right\} \{\mathbf{l}''_4\} \quad (7.42)$$

The first two equations are independent to others, but the third equation is not. Since $\mathbf{e}_{12}(t_4)$ and \mathbf{l}''_4 can be described by $\{\mathbf{m}_1, \mathbf{m}_2, \mathbf{m}_3\}$ and $\{\mathbf{l}''_2, \mathbf{l}''_3\}$ respectively, the third equation can be represented as follows:

$$\{a_1\mathbf{m}_1 + a_2\mathbf{m}_2 + a_3\mathbf{m}_3\} \{\mathbf{e}_{21}(t_1)\} \{d_1\mathbf{l}''_2 + d_2\mathbf{l}''_3\} \quad (7.43)$$

which is the combination of 6 equations in (7.35). Therefore, (7.40) involves 2 independent equations. Thus, using one more pair of $\{\mathbf{e}_{12}(t), \mathbf{e}_{32}(t)\}$, we can derive 2 more independent equations than $1 \times \{\mathbf{e}_{21}(t), \mathbf{e}_{31}(t)\} + 2 \times \{\mathbf{e}_{12}(t), \mathbf{e}_{32}(t)\}$, that is, $1 \times \{\mathbf{e}_{21}(t), \mathbf{e}_{31}(t)\} + 3 \times \{\mathbf{e}_{12}(t), \mathbf{e}_{32}(t)\}$ can bring us $16 + 2 = 18$ independent equations.

On the other hand, (7.40) can also be written into:

$$\{c_1\mathbf{e}_{12}(t_2) + c_2\mathbf{e}_{12}(t_3)\} \left\{ \begin{array}{c} \mathbf{m}'_1 \\ \mathbf{m}'_2 \\ \mathbf{m}'_3 \end{array} \right\} \{d_1\mathbf{l}''_2 + d_2\mathbf{l}''_3\} \quad (7.44)$$

Simplifying it, we have:

$$\left\{ \begin{array}{c} c_2d_1 \\ c_1d_2 \end{array} \right\} \left\{ \begin{array}{c} \mathbf{e}_{12}(t_3)\mathbf{l}''_2 + \mathbf{e}_{12}(t_2)\mathbf{l}''_3 \end{array} \right\} \left\{ \begin{array}{c} \mathbf{m}'_1 \\ \mathbf{m}'_2 \\ \mathbf{m}'_3 \end{array} \right\} \quad (7.45)$$

At time $t_n, n > 4$, we can derive similar equations in which only the coefficient $\frac{c_2 d_1}{c_1 d_2}$ is different, but it can be proved that this coefficient is a constant. That means more pairs of $\{\mathbf{e}_{12}(t), \mathbf{e}_{32}(t)\}$ do not bring new independent equations. Thus, $\mathbf{1} \times \{\mathbf{e}_{21}(t), \mathbf{e}_{31}(t)\} + \mathbf{3}^* \times \{\mathbf{e}_{12}(t), \mathbf{e}_{32}(t)\}$ still provide us 18 independent equations.

$$(4) \mathbf{2} \times \{\mathbf{e}_{21}(t), \mathbf{e}_{31}(t)\} + \mathbf{2} \times \{\mathbf{e}_{12}(t), \mathbf{e}_{32}(t)\}$$

In this case, we have the following simplified equations:

$$\begin{Bmatrix} \mathbf{m}_1 \\ \mathbf{m}_2 \\ \mathbf{m}_3 \end{Bmatrix} \{\mathbf{e}_{21}(t_1)\} \begin{Bmatrix} \mathbf{l}_1'' \\ \mathbf{l}_2'' \end{Bmatrix} \quad (7.46)$$

$$\begin{Bmatrix} \mathbf{m}_1 \\ \mathbf{m}_2 \\ \mathbf{m}_3 \end{Bmatrix} \{\mathbf{e}_{21}(t_2)\} \begin{Bmatrix} \mathbf{l}_1'' \\ \mathbf{l}_3'' \end{Bmatrix} \quad (7.47)$$

$$\{\mathbf{e}_{12}(t_3)\} \begin{Bmatrix} \mathbf{m}'_1 \\ \mathbf{m}'_2 \\ \mathbf{m}'_3 \end{Bmatrix} \begin{Bmatrix} \mathbf{l}_4'' \\ \mathbf{l}_2'' \end{Bmatrix} \quad (7.48)$$

$$\{\mathbf{e}_{12}(t_4)\} \begin{Bmatrix} \mathbf{m}'_1 \\ \mathbf{m}'_2 \\ \mathbf{m}'_3 \end{Bmatrix} \begin{Bmatrix} \mathbf{l}_4'' \\ \mathbf{l}_3'' \end{Bmatrix} \quad (7.49)$$

where \mathbf{l}_1'' and \mathbf{l}_2'' go through $\mathbf{e}_{31}(t_1)$, \mathbf{l}_1'' and \mathbf{l}_3'' go through $\mathbf{e}_{31}(t_2)$, \mathbf{l}_4'' and \mathbf{l}_2'' go through $\mathbf{e}_{32}(t_3)$, and \mathbf{l}_4'' and \mathbf{l}_3'' go through $\mathbf{e}_{32}(t_4)$ as shown in Figure 7.5. (7.46) and (7.47) first provides us 6 independent equations respectively. Next, let us consider (7.48). It can be separated into:

$$\{\mathbf{e}_{12}(t_3)\} \begin{Bmatrix} \mathbf{m}'_1 \\ \mathbf{m}'_2 \\ \mathbf{m}'_3 \end{Bmatrix} \{\mathbf{l}_4''\} \quad (7.50)$$

$$\{\mathbf{e}_{12}(t_3)\} \begin{Bmatrix} \mathbf{m}'_1 \\ \mathbf{m}'_2 \\ \mathbf{m}'_3 \end{Bmatrix} \{\mathbf{l}_2''\} \quad (7.51)$$

Since (7.50) is independent with (7.46) and (7.47), (7.50) brings us 3 independent equations. Moreover, it can be written as:

$$\{(c_1 \mathbf{m}_1 + c_2 \mathbf{m}_2 + c_3 \mathbf{m}_3)\} \begin{Bmatrix} \mathbf{m}'_1 \\ \mathbf{m}'_2 \\ \mathbf{e}_{21}(t_2) \end{Bmatrix} \{\mathbf{l}_4''\} \quad (7.52)$$

Then (7.51) can also be rewritten into:

$$\{\mathbf{e}_{12}(t_3)\} \begin{Bmatrix} \mathbf{m}'_1 \\ \mathbf{l}_2'' \end{Bmatrix} \quad (7.53)$$

$$\{c_1 \mathbf{m}_1 + c_2 \mathbf{m}_2 + c_3 \mathbf{m}_3\} \begin{Bmatrix} \mathbf{e}_{21}(t_1) \\ \mathbf{l}_2'' \end{Bmatrix} \quad (7.54)$$

$$\{c_1 \mathbf{m}_1 + c_2 \mathbf{m}_2 + c_3 \mathbf{m}_3\} \begin{Bmatrix} \mathbf{e}_{21}(t_2) \\ d_1 \mathbf{l}_1'' + d_2 \mathbf{l}_3'' + d_3 \mathbf{l}_4'' \end{Bmatrix} \quad (7.55)$$

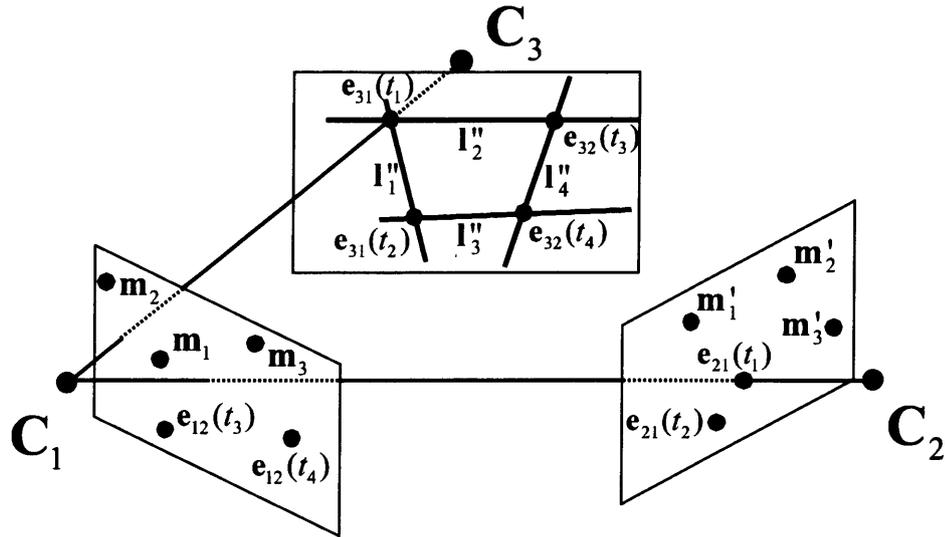


Figure 7.5: The basis points, basis lines and epipole lines for representing incidence relations in three views. $\{\mathbf{m}_1, \mathbf{m}_2, \mathbf{m}_3\}$ show three basis points in view 1. $\{\mathbf{m}'_1, \mathbf{m}'_2, \mathbf{m}'_3\}$ show three basis points in view 2. $\{\mathbf{l}''_2, \mathbf{l}''_3\}$ show two basis lines in view 3. \mathbf{l}''_1 shows epipole line which goes through \mathbf{e}_{31} . \mathbf{l}''_4 shows epipole line which goes through \mathbf{e}_{32} .

(7.53) is independent to (7.46), (7.47) and (7.50), but (7.54) can be described by (7.46), in addition, (7.55) can be represented by (7.47) and (7.52). Therefore, (7.51) contributes only 1 independent equation. Thus, (7.48) provides us $3 + 1 = 4$ independent equation. For the same reason, (7.49) also brings us 4 independent equations. Then, $2 \times \{\mathbf{e}_{21}(t), \mathbf{e}_{31}(t)\} + 2 \times \{\mathbf{e}_{12}(t), \mathbf{e}_{32}(t)\}$ provides us $6 + 6 + 4 + 4 = 20$ independent equation.

(5) $2 \times \{\mathbf{e}_{21}(t), \mathbf{e}_{31}(t)\} + 3^* \times \{\mathbf{e}_{12}(t), \mathbf{e}_{32}(t)\}$

We have derived 20 independent equations from the case $2 \times \{\mathbf{e}_{21}(t), \mathbf{e}_{31}(t)\} + 2 \times \{\mathbf{e}_{12}(t), \mathbf{e}_{32}(t)\}$. What will happen when we use one more pair of $\{\mathbf{e}_{12}(t), \mathbf{e}_{32}(t)\}$? We can add the following equations to (7.46)~(7.49) to consider this case:

$$\{\mathbf{e}_{12}(t_5)\} \left\{ \begin{array}{c} \mathbf{m}'_1 \\ \mathbf{m}'_2 \\ \mathbf{m}'_3 \end{array} \right\} \left\{ \begin{array}{c} \mathbf{l}''_4 \\ \mathbf{l}''_5 \end{array} \right\} \quad (7.56)$$

where \mathbf{l}''_4 is a epipole line which goes through \mathbf{e}_{32} , and \mathbf{l}''_5 denotes a line going through

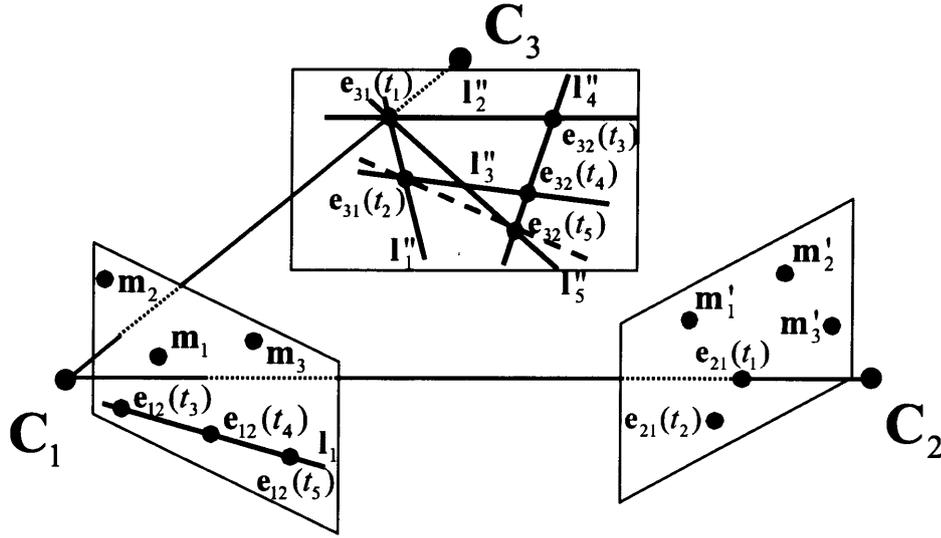


Figure 7.6: The basis points, basis lines and epipole lines for representing incidence relations in three views. $\{\mathbf{m}_1, \mathbf{m}_2, \mathbf{m}_3\}$ show three basis points in view 1. $\{\mathbf{m}'_1, \mathbf{m}'_2, \mathbf{m}'_3\}$ show three basis points in view 2. $\{\mathbf{l}''_2, \mathbf{l}''_3, \mathbf{l}''_5\}$ show three basis lines in view 3. \mathbf{l}_1 shows epipole line which goes through \mathbf{e}_{12} . \mathbf{l}'_1 shows epipole line which goes through \mathbf{e}_{31} . \mathbf{l}'_4 shows epipole line which goes through \mathbf{e}_{32} .

Table 7.3: The number of independent equations derived by using $\{\mathbf{e}_{21}(t), \mathbf{e}_{31}(t)\}$ for N_1 time ($t = t_1, \dots, t_{N_1}$), and $\{\mathbf{e}_{12}(t), \mathbf{e}_{32}(t)\}$ for N_2 time ($t = t_1, \dots, t_{N_2}$), and the number of corresponding points required for computing trifocal tensors in each case of mutual projections of cameras. 3^* denotes 3 or greater than 3.

$N_1 \times \{\mathbf{e}_{21}(t), \mathbf{e}_{31}(t)\}$ + $N_2 \times \{\mathbf{e}_{12}(t), \mathbf{e}_{32}(t)\}$	# of independent eq.	# of points required
1 + 1	11	8
1 + 2	16	5
1 + 3*	18	4
2 + 2	20	3
2 + 3*	21	3
3* + 3*	22	2

$\mathbf{e}_{32}(t_5)$ as shown in Figure 7.6. Then (7.56) can be rewritten into:

$$\{a_1 \mathbf{e}_{12}(t_3) + a_2 \mathbf{e}_{12}(t_4)\} \left\{ \begin{array}{c} \mathbf{m}'_1 \\ \mathbf{m}'_2 \\ \mathbf{m}'_3 \end{array} \right\} \{\mathbf{l}'_4\} \quad (7.57)$$

$$\{c_1 \mathbf{m}_1 + c_2 \mathbf{m}_2 + c_3 \mathbf{m}_3\} \left\{ \begin{array}{c} \mathbf{m}'_1 \\ \mathbf{e}_{21}(t_1) \\ \mathbf{e}_{21}(t_2) \end{array} \right\} \{\mathbf{l}'_5\} \quad (7.58)$$

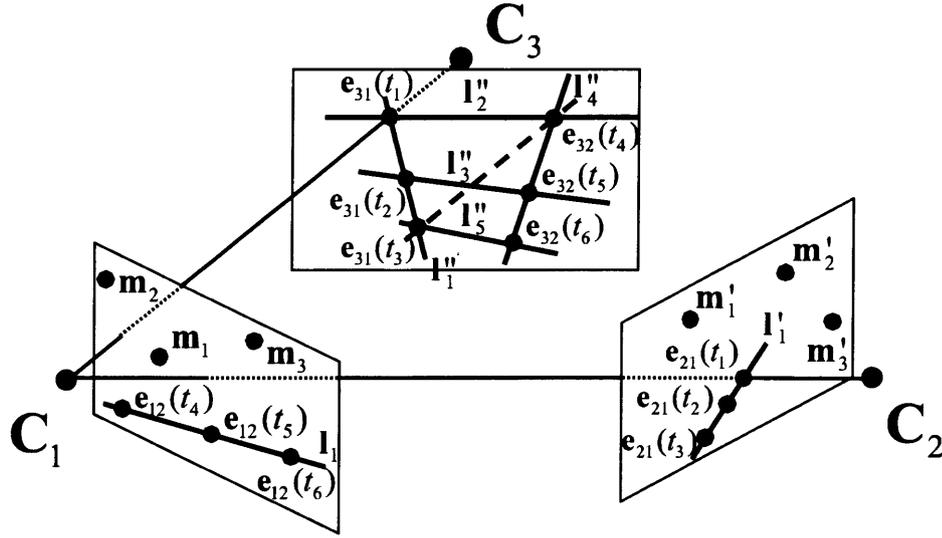


Figure 7.7: The basis points, basis lines and epipole lines for representing incidence relations in three views. $\{\mathbf{m}_1, \mathbf{m}_2, \mathbf{m}_3\}$ show three basis points in view 1. $\{\mathbf{m}'_1, \mathbf{m}'_2, \mathbf{m}'_3\}$ show three basis points in view 2. $\{\mathbf{l}''_2, \mathbf{l}''_3, \mathbf{l}''_5\}$ show three basis lines in view 3. \mathbf{l}_1 shows epipole line which goes through \mathbf{e}_{12} . \mathbf{l}'_1 shows epipole line which goes through \mathbf{e}_{21} . \mathbf{l}''_1 shows epipole line which goes through \mathbf{e}_{31} . \mathbf{l}''_4 shows epipole line which goes through \mathbf{e}_{32} .

(7.57) can be described by (7.48) and (7.49), so it does not provide any independent equation. In addition, \mathbf{l}''_5 goes through $\mathbf{e}_{32}(t_5)$. Then it can be a line going through not only $\mathbf{e}_{31}(t_1)$ and $\mathbf{e}_{32}(t_5)$, but $\mathbf{e}_{31}(t_2)$ and $\mathbf{e}_{32}(t_5)$ as shown in Figure 7.6. Thus, (7.58) can be written as:

$$\{c_1\mathbf{m}_1 + c_2\mathbf{m}_2 + c_3\mathbf{m}_3\} \quad \{\mathbf{m}'_1\} \quad \{\mathbf{l}''_5\} \quad (7.59)$$

$$\{c_1\mathbf{m}_1 + c_2\mathbf{m}_2 + c_3\mathbf{m}_3\} \quad \{\mathbf{e}_{21}(t_1)\} \quad \{d_1\mathbf{l}''_1 + d_2\mathbf{l}''_2\} \quad (7.60)$$

$$\{c_1\mathbf{m}_1 + c_2\mathbf{m}_2 + c_3\mathbf{m}_3\} \quad \{\mathbf{e}_{21}(t_2)\} \quad \{d_3\mathbf{l}''_1 + d_4\mathbf{l}''_3\} \quad (7.61)$$

(7.59) is independent to other equations, but (7.60) and (7.61) can be represented by (7.46) and (7.47) respectively. Then (7.56) provides us 1 independent equation. Thus, in the case of $2 \times \{\mathbf{e}_{21}(t), \mathbf{e}_{31}(t)\} + 3 \times \{\mathbf{e}_{12}(t), \mathbf{e}_{32}(t)\}$, we have $20 + 1 = 21$ independent equations.

On the other hand, (7.56) can also be described by:

$$\{a_1\mathbf{e}_{12}(t_3) + a_2\mathbf{e}_{12}(t_4)\} \left\{ \begin{array}{l} \mathbf{m}'_1 \\ \mathbf{m}'_2 \\ \mathbf{m}'_3 \end{array} \right\} \{b_1\mathbf{l}''_2 + b_2\mathbf{l}''_3 + b_3\mathbf{l}''_4\} \quad (7.62)$$

Modifying it, we have:

$$\left\{ \frac{a_2 b_1}{a_1 b_2} \mathbf{e}_{12}(t_4) \mathbf{l}_2'' + \mathbf{e}_{12}(t_3) \mathbf{l}_3'' \right\} \begin{Bmatrix} \mathbf{m}'_1 \\ \mathbf{m}'_2 \\ \mathbf{m}'_3 \end{Bmatrix} \quad (7.63)$$

Since $\frac{a_2 b_1}{a_1 b_2}$ is a constant, even if we use more $\{\mathbf{e}_{12}(t), \mathbf{e}_{32}(t)\}$, the number of independent equations could not increase for the same reason mentioned before. Therefore, $2 \times \{\mathbf{e}_{21}(t), \mathbf{e}_{31}(t)\} + 3^* \times \{\mathbf{e}_{12}(t), \mathbf{e}_{32}(t)\}$ provides us 21 independent equations.

(6) $3^* \times \{\mathbf{e}_{21}(t), \mathbf{e}_{31}(t)\} + 3^* \times \{\mathbf{e}_{12}(t), \mathbf{e}_{32}(t)\}$

The discussion on the number of independent equations in this case is very similar to the previous case, $2 \times \{\mathbf{e}_{21}(t), \mathbf{e}_{31}(t)\} + 3^* \times \{\mathbf{e}_{12}(t), \mathbf{e}_{32}(t)\}$, so we do not give the explainer here, only the configuration (see Figure 7.7).

Up to now, we have considered all the cases of using epipole pair $\{\mathbf{e}_{21}(t), \mathbf{e}_{31}(t)\}$ and $\{\mathbf{e}_{12}(t), \mathbf{e}_{32}(t)\}$. They are summarized in Table 7.3.

7.2.2.2 Using Epipole Pair $\{\mathbf{e}_{21}, \mathbf{e}_{31}\}$ and $\{\mathbf{e}_{13}, \mathbf{e}_{23}\}$, or $\{\mathbf{e}_{12}, \mathbf{e}_{32}\}$ and $\{\mathbf{e}_{13}, \mathbf{e}_{23}\}$

In such combinations, the number of independent equations and corresponding points required in all the cases are summarized in Table 7.4. Most of them can be obtained by Table 7.1 and Table 7.2 directly. Only the following two cases need to be explained:

No.	$N_1 \times \{\mathbf{e}_{21}(t), \mathbf{e}_{31}(t)\}$ or $\{\mathbf{e}_{12}(t), \mathbf{e}_{32}(t)\}$ + $N_2 \times \{\mathbf{e}_{13}(t), \mathbf{e}_{23}(t)\}$	# of independent eq.	# of points required
1	$2 + 3^*$	20	3
2	$3^* + 3^*$	22	2

We first consider $2 \times \{\mathbf{e}_{21}(t), \mathbf{e}_{31}(t)\} + 3 \times \{\mathbf{e}_{13}(t), \mathbf{e}_{23}(t)\}$ in No.1.

Table 7.4: The number of independent equations derived by using of $\{\mathbf{e}_{21}(t), \mathbf{e}_{31}(t)\}$ or $\{\mathbf{e}_{12}(t), \mathbf{e}_{32}(t)\}$ for N_1 time ($t = t_1, \dots, t_{N_1}$), and $\{\mathbf{e}_{13}(t), \mathbf{e}_{23}(t)\}$ for N_2 time ($t = t_1, \dots, t_{N_2}$), and the number of corresponding points required for computing trifocal tensors in each case of mutual projections of cameras. 3^* denotes 3 or greater than 3.

$N_1 \times \{\mathbf{e}_{21}(t), \mathbf{e}_{31}(t)\}$ or $\{\mathbf{e}_{12}(t), \mathbf{e}_{32}(t)\}$ + $N_2 \times \{\mathbf{e}_{13}(t), \mathbf{e}_{23}(t)\}$	# of independent eq.	# of points required
1 + 1	9	9
1 + 2	12	7
1 + 3*	15	6
2 + 1	15	6
3* + 1	18	4
2 + 2	18	4
2 + 3*	20	3
3* + 2	21	3
3* + 3*	22	2

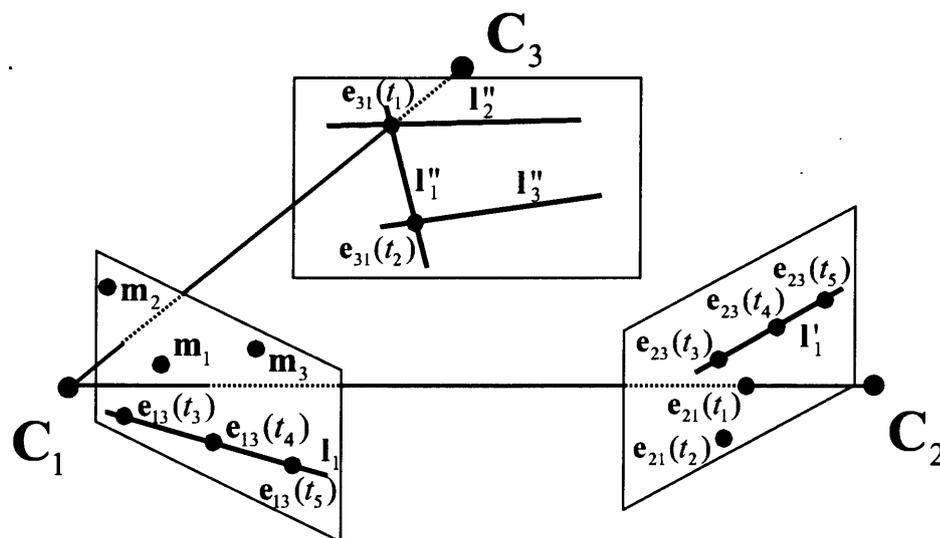


Figure 7.8: The basis points, basis lines and epipole lines for representing incidence relations in three views. $\{\mathbf{m}_1, \mathbf{m}_2, \mathbf{m}_3\}$ show three basis points in view 1. $\{l''_2, l''_3\}$ show three basis lines in view 3. l_1 shows epipole line which goes through \mathbf{e}_{13} . l'_1 shows epipole line which goes through \mathbf{e}_{23} . l''_1 shows epipole line which goes through \mathbf{e}_{31} .

In this case, we have 5 sets of simplified equations:

$$\begin{Bmatrix} \mathbf{m}_1 \\ \mathbf{m}_2 \\ \mathbf{m}_3 \end{Bmatrix} \{ \mathbf{e}_{21}(t_1) \} \begin{Bmatrix} \mathbf{l}_1'' \\ \mathbf{l}_2'' \end{Bmatrix} \quad (7.64)$$

$$\begin{Bmatrix} \mathbf{m}_1 \\ \mathbf{m}_2 \\ \mathbf{m}_3 \end{Bmatrix} \{ \mathbf{e}_{21}(t_2) \} \begin{Bmatrix} \mathbf{l}_1'' \\ \mathbf{l}_3'' \end{Bmatrix} \quad (7.65)$$

$$\{ \mathbf{e}_{13}(t_3) \} \{ \mathbf{e}_{23}(t_3) \} \begin{Bmatrix} \mathbf{l}_1'' \\ \mathbf{l}_2'' \\ \mathbf{l}_3'' \end{Bmatrix} \quad (7.66)$$

$$\{ \mathbf{e}_{13}(t_4) \} \{ \mathbf{e}_{23}(t_4) \} \begin{Bmatrix} \mathbf{l}_1'' \\ \mathbf{l}_2'' \\ \mathbf{l}_3'' \end{Bmatrix} \quad (7.67)$$

$$\{ \mathbf{e}_{13}(t_5) \} \{ \mathbf{e}_{23}(t_5) \} \begin{Bmatrix} \mathbf{l}_1'' \\ \mathbf{l}_2'' \\ \mathbf{l}_3'' \end{Bmatrix} \quad (7.68)$$

where \mathbf{l}_1'' and \mathbf{l}_2'' go through $\mathbf{e}_{31}(t_1)$, and, \mathbf{l}_1'' and \mathbf{l}_3'' go through $\mathbf{e}_{31}(t_2)$ as shown in Figure 7.8. (7.64)~(7.67) represent the case of $\mathbf{2} \times \{ \mathbf{e}_{21}(t), \mathbf{e}_{31}(t) \} + \mathbf{2} \times \{ \mathbf{e}_{13}(t), \mathbf{e}_{23}(t) \}$, which provides us 18 independent equations. Then based on it, how many independent equations can we derive from (7.68)?

Since $\mathbf{e}_{13}(t_4)$ and $\mathbf{e}_{23}(t_4)$ can be described by $\{ \mathbf{m}_1, \mathbf{m}_2, \mathbf{m}_3 \}$ and $\{ \mathbf{e}_{21}(t_1), \mathbf{e}_{21}(t_2), \mathbf{e}_{23}(t_3) \}$ respectively, the first equation in (7.67) has the following calculations:

$$\{ \mathbf{e}_{13}(t_4) \} \{ \mathbf{e}_{23}(t_4) \} \{ \mathbf{l}_1'' \} \quad (7.69)$$

$$= \{ c_1 \mathbf{m}_1 + c_2 \mathbf{m}_2 + c_3 \mathbf{m}_3 \} \{ d_1 \mathbf{e}_{21}(t_1) + d_2 \mathbf{e}_{21}(t_2) + d_3 \mathbf{e}_{23}(t_3) \} \{ \mathbf{l}_1'' \} \quad (7.70)$$

$$= \{ c_1 \mathbf{m}_1 + c_2 \mathbf{m}_2 + c_3 \mathbf{m}_3 \} \{ \mathbf{e}_{23}(t_3) \} \{ \mathbf{l}_1'' \} \quad (7.71)$$

$$= \{ \mathbf{e}_{13}(t_4) \} \{ \mathbf{e}_{23}(t_3) \} \{ \mathbf{l}_1'' \} \quad (7.72)$$

For the same reason, the first equation in (7.68) can also be rewritten as:

$$\{ \mathbf{e}_{13}(t_5) \} \{ \mathbf{e}_{23}(t_3) \} \{ \mathbf{l}_1'' \} \quad (7.73)$$

and since it can be described by:

$$\{ c_4 \mathbf{e}_{13}(t_3) + c_5 \mathbf{e}_{13}(t_4) \} \{ \mathbf{e}_{23}(t_3) \} \{ \mathbf{l}_1'' \} \quad (7.74)$$

which can be represented by (7.66) and (7.72), the first equation in (7.68) is not independent. Whereas the other 2 equations in (7.68) are independent to all the others. Therefore, (7.68) brings us only 2 independent equations. Even if one more pair of $\{ \mathbf{e}_{13}(t), \mathbf{e}_{23}(t) \}$ is given, it will not provide us more independent constrains.

Next, let us discuss the case of $\mathbf{3} \times \{ \mathbf{e}_{21}(t), \mathbf{e}_{31}(t) \} + \mathbf{3} \times \{ \mathbf{e}_{13}(t), \mathbf{e}_{23}(t) \}$ in No.2.

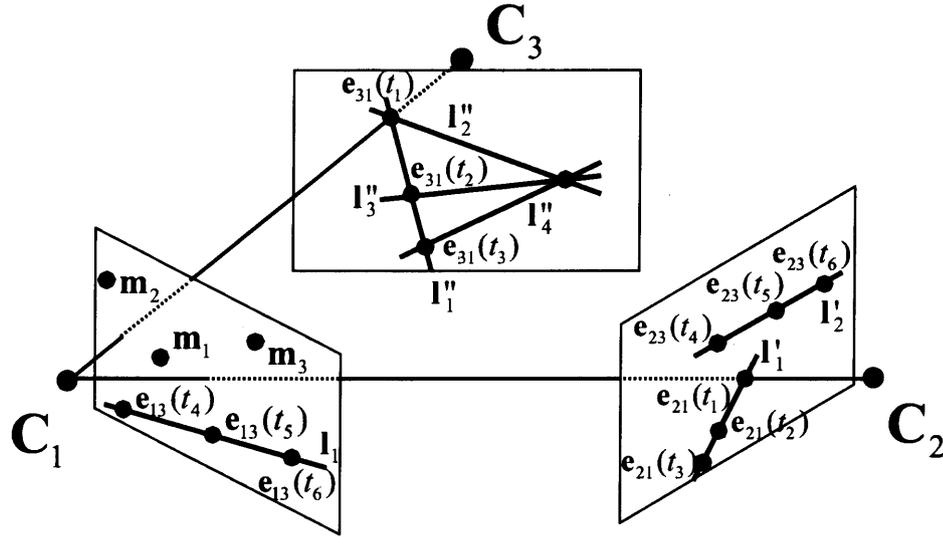


Figure 7.9: The basis points, basis lines and epipole lines for representing incidence relations in three views. $\{\mathbf{m}_1, \mathbf{m}_2, \mathbf{m}_3\}$ show three basis points in view 1. $\{\mathbf{l}''_1, \mathbf{l}''_2, \mathbf{l}''_3, \mathbf{l}''_4\}$ show three basis lines in view 3. \mathbf{l}_1 shows epipole line which goes through \mathbf{e}_{13} . \mathbf{l}'_1 shows epipole line which goes through \mathbf{e}_{21} . \mathbf{l}'_2 shows epipole line which goes through \mathbf{e}_{23} . \mathbf{l}''_1 shows epipole line which goes through \mathbf{e}_{31} .

All the equations with simplified forms are as follows:

$$\begin{Bmatrix} \mathbf{m}_1 \\ \mathbf{m}_2 \\ \mathbf{m}_3 \end{Bmatrix} \{ \mathbf{e}_{21}(t_1) \} \begin{Bmatrix} \mathbf{l}''_1 \\ \mathbf{l}''_2 \end{Bmatrix} \quad (7.75)$$

$$\begin{Bmatrix} \mathbf{m}_1 \\ \mathbf{m}_2 \\ \mathbf{m}_3 \end{Bmatrix} \{ \mathbf{e}_{21}(t_2) \} \begin{Bmatrix} \mathbf{l}''_1 \\ \mathbf{l}''_3 \end{Bmatrix} \quad (7.76)$$

$$\begin{Bmatrix} \mathbf{m}_1 \\ \mathbf{m}_2 \\ \mathbf{m}_3 \end{Bmatrix} \{ \mathbf{e}_{21}(t_3) \} \begin{Bmatrix} \mathbf{l}''_1 \\ \mathbf{l}''_4 \end{Bmatrix} \quad (7.77)$$

$$\{ \mathbf{e}_{13}(t_4) \} \{ \mathbf{e}_{23}(t_4) \} \begin{Bmatrix} \mathbf{l}''_1 \\ \mathbf{l}''_2 \\ \mathbf{l}''_3 \end{Bmatrix} \quad (7.78)$$

$$\{ \mathbf{e}_{13}(t_5) \} \{ \mathbf{e}_{23}(t_5) \} \begin{Bmatrix} \mathbf{l}''_1 \\ \mathbf{l}''_2 \\ \mathbf{l}''_3 \end{Bmatrix} \quad (7.79)$$

$$\{ \mathbf{e}_{13}(t_6) \} \{ \mathbf{e}_{23}(t_6) \} \begin{Bmatrix} \mathbf{l}''_1 \\ \mathbf{l}''_2 \\ \mathbf{l}''_3 \end{Bmatrix} \quad (7.80)$$

The configuration of them is shown in Figure 7.9. If we only focus on (7.75)~(7.79), we know that they describe the case of $\mathbf{3} \times \{\mathbf{e}_{21}(t), \mathbf{e}_{31}(t)\} + \mathbf{2} \times \{\mathbf{e}_{13}(t), \mathbf{e}_{23}(t)\}$ which brings us 21 independent equations. How about (7.80)?

Since (7.79) can be represented by:

$$\{c_1 \mathbf{m}_1 + c_2 \mathbf{m}_2 + c_3 \mathbf{m}_3\} \{d_1 \mathbf{e}_{21}(t_1) + d_2 \mathbf{e}_{21}(t_2) + d_3 \mathbf{e}_{23}(t_4)\} \left\{ \begin{array}{l} \mathbf{l}_1'' \\ \mathbf{l}_2'' \\ \mathbf{l}_3'' \end{array} \right\} \quad (7.81)$$

Expanding and simplifying them, we obtain:

$$\{\mathbf{e}_{13}(t_5)\} \{\mathbf{e}_{23}(t_4)\} \{\mathbf{l}_1''\} \quad (7.82)$$

$$\{\mathbf{e}_{13}(t_5)\} \{d_2 \mathbf{e}_{21}(t_2) + d_3 \mathbf{e}_{23}(t_4)\} \{\mathbf{l}_2''\} \quad (7.83)$$

$$\{\mathbf{e}_{13}(t_5)\} \{d_1 \mathbf{e}_{21}(t_1) + d_3 \mathbf{e}_{23}(t_4)\} \{\mathbf{l}_3''\} \quad (7.84)$$

For the same reason, (7.80) also equals to:

$$\{\mathbf{e}_{13}(t_6)\} \{\mathbf{e}_{23}(t_4)\} \{\mathbf{l}_1''\} \quad (7.85)$$

$$\{\mathbf{e}_{13}(t_6)\} \{d_2' \mathbf{e}_{21}(t_2) + d_3' \mathbf{e}_{23}(t_4)\} \{\mathbf{l}_2''\} \quad (7.86)$$

$$\{\mathbf{e}_{13}(t_6)\} \{d_1' \mathbf{e}_{21}(t_1) + d_3' \mathbf{e}_{23}(t_4)\} \{\mathbf{l}_3''\} \quad (7.87)$$

(7.85) can be described by (7.78) and (7.82), so in (7.80), only 2 independent equation candidates exit.

On the other hand,

$$\left\{ \begin{array}{l} \mathbf{m}_1 \\ \mathbf{m}_2 \\ \mathbf{m}_3 \end{array} \right\} \{\mathbf{e}_{21}(t_3)\} \{\mathbf{l}_4''\} \quad (7.88)$$

in (7.77) can be rewritten into:

$$\left\{ \begin{array}{l} \mathbf{e}_{13}(t_5) \\ \mathbf{e}_{13}(t_6) \\ \mathbf{m}_3 \end{array} \right\} \{a_1 \mathbf{e}_{21}(t_1) + a_2 \mathbf{e}_{21}(t_2)\} \{b_1 \mathbf{l}_2'' + b_2 \mathbf{l}_3''\} \quad (7.89)$$

So we have these 2 equations:

$$\frac{a_2 b_1}{a_1 b_2} \{\mathbf{e}_{13}(t_5)\} \{\mathbf{e}_{21}(t_2)\} \{\mathbf{l}_2''\} + \{\mathbf{e}_{13}(t_5)\} \{\mathbf{e}_{21}(t_1)\} \{\mathbf{l}_3''\} \quad (7.90)$$

$$\frac{a_2 b_1}{a_1 b_2} \{\mathbf{e}_{13}(t_6)\} \{\mathbf{e}_{21}(t_2)\} \{\mathbf{l}_2''\} + \{\mathbf{e}_{13}(t_6)\} \{\mathbf{e}_{21}(t_1)\} \{\mathbf{l}_3''\} \quad (7.91)$$

Combining (7.87) and (7.91) we have:

$$-\frac{a_2 b_1 d_1'}{a_1 b_2} \{\mathbf{e}_{13}(t_6)\} \{\mathbf{e}_{21}(t_2)\} \{\mathbf{l}_2''\} + \{d_3' \mathbf{e}_{13}(t_6)\} \{\mathbf{e}_{23}(t_4)\} \{\mathbf{l}_3''\} \quad (7.92)$$

Then, by substituting (7.86) into (7.92), we obtain:

$$-\frac{a_2 b_1 d_1'}{a_1 b_2 d_2'} \{\mathbf{e}_{13}(t_6)\} \{\mathbf{e}_{23}(t_4)\} \{\mathbf{l}_2''\} + \{\mathbf{e}_{13}(t_6)\} \{\mathbf{e}_{23}(t_4)\} \{\mathbf{l}_3''\} \quad (7.93)$$

Similarly, the following equation can also be derived:

$$-\frac{a_2 b_1 d_1}{a_1 b_2 d_2} \{\mathbf{e}_{13}(t_5)\} \{\mathbf{e}_{23}(t_4)\} \{\mathbf{l}_2''\} + \{\mathbf{e}_{13}(t_5)\} \{\mathbf{e}_{23}(t_4)\} \{\mathbf{l}_3''\} \quad (7.94)$$

Since

$$-\frac{a_2 b_1 d_1}{a_1 b_2 d_2} = -\frac{a_2 b_1 d_1'}{a_1 b_2 d_2'}, \quad (7.95)$$

we denote them as A . For $\mathbf{e}_{13}(t_6) = c_4 \mathbf{e}_{13}(t_4) + c_5 \mathbf{e}_{13}(t_5)$, (7.93) can be written as follows:

$$A c_4 \{\mathbf{e}_{13}(t_4)\} \{\mathbf{e}_{23}(t_4)\} \{\mathbf{l}_2''\} + c_4 \{\mathbf{e}_{13}(t_4)\} \{\mathbf{e}_{23}(t_4)\} \{\mathbf{l}_3''\} \\ + c_5 (A \{\mathbf{e}_{13}(t_5)\} \{\mathbf{e}_{23}(t_4)\} \{\mathbf{l}_2''\} + \{\mathbf{e}_{13}(t_5)\} \{\mathbf{e}_{23}(t_4)\} \{\mathbf{l}_3''\})$$

which is the pure combination of (7.78) and (7.94). Therefore, (7.87) can be described by other equations. Thus, (7.80) only provides us 1 independent equation. Then, from the case of $\mathbf{3} \times \{\mathbf{e}_{21}(t), \mathbf{e}_{31}(t)\} + \mathbf{3} \times \{\mathbf{e}_{13}(t), \mathbf{e}_{23}(t)\}$, we derive $21 + 1 = 22$ independent equations. Increasing the number of $\{\mathbf{e}_{21}(t), \mathbf{e}_{31}(t)\}$ or $\{\mathbf{e}_{13}(t), \mathbf{e}_{23}(t)\}$ will not bring more independent constrains. So, in case No.2, we still have 22 independent equations. And if we change the epipole pair $\{\mathbf{e}_{21}, \mathbf{e}_{31}\}$ to $\{\mathbf{e}_{12}, \mathbf{e}_{32}\}$, the same results will be derived.

7.2.3 Using All Three Epipole Pairs

From Table 7.1~Table 7.4, we can deduce the number of independent equations and corresponding points required by using all three epipole pairs. The results are summarized in Table 7.5. The most interesting thing is when we have 2 or more samples of each epipole pair, we do not need any corresponding point to derive trifocal tensors anymore.

7.3 Experiments

We next show the results of experiments and discuss the efficiency of making use of mutual projections of cameras in the computation of trifocal tensors in 4D space.

We first show the results from real images that the trifocal tensor in 4D space can be computed from three epipole pairs at different time viewed from arbitrary translational cameras with no corresponding points, and can be used for generating the third view from the first view and the second view of moving cameras. We next evaluate the stability of extracted trifocal tensors in this brand new case and compare it with traditional method.

Table 7.5: The number of independent equations derived by using $\{\mathbf{e}_{21}(t), \mathbf{e}_{31}(t)\}$ for N_1 time ($t = t_1, \dots, t_{N_1}$), $\{\mathbf{e}_{12}(t), \mathbf{e}_{32}(t)\}$ for N_2 time ($t = t_1, \dots, t_{N_2}$) and $\{\mathbf{e}_{13}(t), \mathbf{e}_{23}(t)\}$ for N_3 time ($t = t_1, \dots, t_{N_3}$), and the number of corresponding points required for computing trifocal tensors in each case of mutual projections of cameras. x^* denotes x or greater than x .

$N_1 \times \{\mathbf{e}_{21}(t), \mathbf{e}_{31}(t)\}$ + $N_2 \times \{\mathbf{e}_{12}(t), \mathbf{e}_{32}(t)\}$ + $N_3 \times \{\mathbf{e}_{13}(t), \mathbf{e}_{23}(t)\}$	# of independent eq.	# of points required
1 + 1 + 1	14	6
1 + 1 + 2	17	5
1 + 2 + 1	19	4
1 + 1 + 3*	20	3
1 + 3* + 1	21	3
1 + 2 + 2	22	2
2 + 2 + 1	23	2
1 + 2 + 3*	24	1
1 + 3* + 2	24	1
2 + 3* + 1	24	1
1 + 3* + 3*	25	1
3* + 3* + 1	25	1
2* + 2* + 2*	26	0

7.3.1 Real Image Experiment

In this section, we show the results from the following case:

$$2 \times \{\mathbf{e}_{21}(t), \mathbf{e}_{31}(t)\} + 2 \times \{\mathbf{e}_{12}(t), \mathbf{e}_{32}(t)\} + 2 \times \{\mathbf{e}_{13}(t), \mathbf{e}_{23}(t)\}.$$

That is, we use each epipole pair at two different time respectively to compute trifocal tensor, but do not use any corresponding point.

In the first experiment, we used 2 omnidirectional cameras and 1 general camera. These 3 cameras are translating with different constant speed and different direction. We computed the trifocal tensor between these three cameras by using 3 epipole pairs. We can compute the extended trifocal tensor and can generate arbitrary image motions in one of three views from the other two views. In this experiment we generated image motions in camera 3 by using image motions in camera 1 and camera 2. Figure 7.10 (a), (b) and (c) show image motions of a single moving point and 6 epipole lines in translational camera 1, camera 2 and camera 3 respectively. The trifocal tensor is computed from 3 epipole pairs, each of which is sampled at two

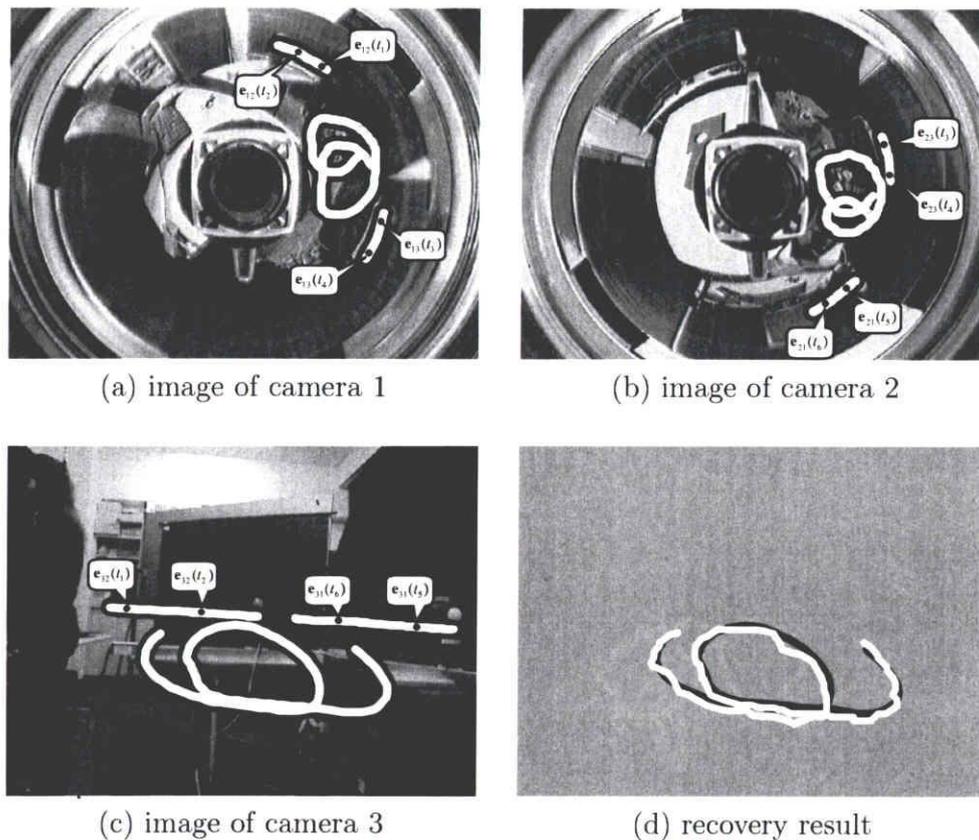


Figure 7.10: Real image experiment 1. (a), (b) and (c) show epipole lines and image motions of a single point viewed from camera 1, 2 and 3. The black points on epipole lines in each image are used for computing the trifocal tensor. The white curve in (d) shows image motions in camera 3 generated from the extended trifocal tensor, and the black curve shows the real image motions.

different time, $\{\mathbf{e}_{12}(t_1), \mathbf{e}_{32}(t_1)\}$, $\{\mathbf{e}_{12}(t_2), \mathbf{e}_{32}(t_2)\}$, $\{\mathbf{e}_{13}(t_3), \mathbf{e}_{23}(t_3)\}$, $\{\mathbf{e}_{13}(t_4), \mathbf{e}_{23}(t_4)\}$, $\{\mathbf{e}_{21}(t_5), \mathbf{e}_{31}(t_5)\}$, $\{\mathbf{e}_{21}(t_6), \mathbf{e}_{31}(t_6)\}$, which are shown by black points in (a), (b) and (c). The extracted trifocal tensor is used for generating image motions in camera 3 from image motions in camera 1 and 2. The white curve in Figure 7.10 (d) shows image motions in camera 3 generated from the extended trifocal tensor, and the black curve shows the real image motions viewed from camera 3. As shown in Figure 7.10 (d), the generated image motions almost recovered the original image motions even if these 3 cameras have unknown translational motions.

The other experiment is also given. Figure 7.11, (a), (b) and (c) show three views of the epipole lines and image motions. The 2 black points on each epipole line are used for computing the trifocal tensor. Note that these 3 cameras are translating with

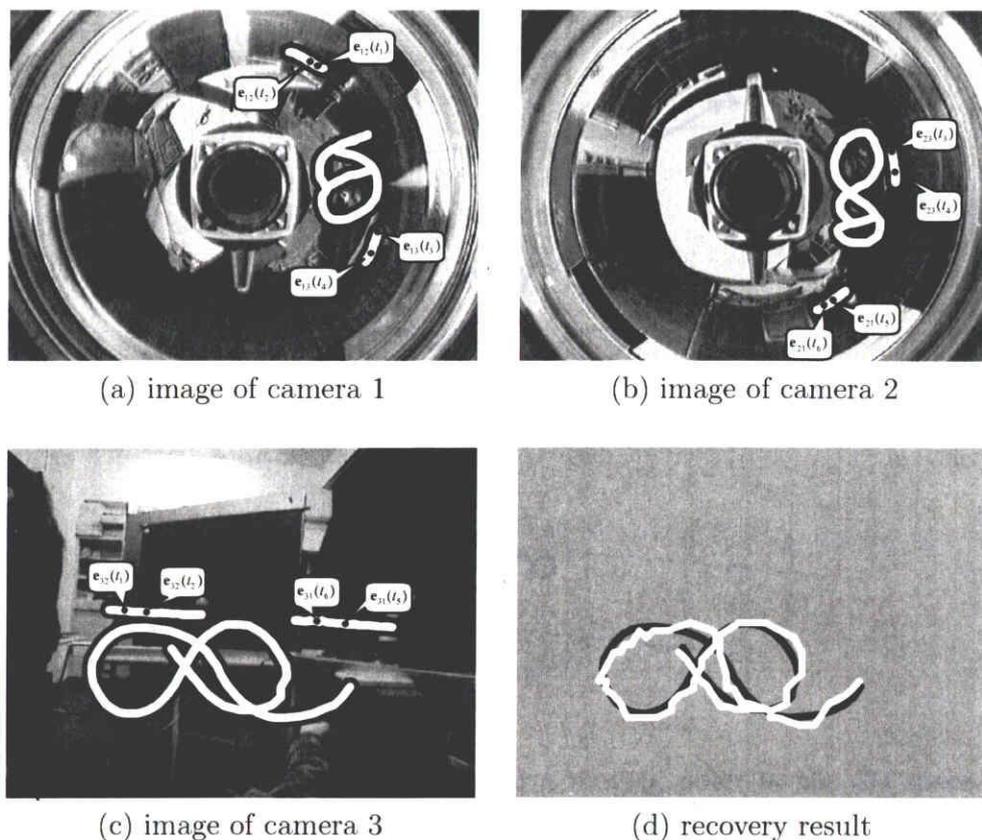


Figure 7.11: Real image experiment 2. (a), (b) and (c) show epipole lines and image motions of a single point viewed from camera 1, 2 and 3. The black points on epipole lines in each image are used for computing the trifocal tensor. The white curve in (d) shows image motions in camera 3 generated from the extended trifocal tensor, and the black curve shows the real image motions.

different speed and different direction. The white curve in (d) shows image motions recovered from the extended trifocal tensor in camera 3, and the black curve shows real image motions observed in camera 3. As we can see, the trifocal tensor defined under 4D to 2D projections can be derived only from 2 samples of the projection of each camera with arbitrary translational motion, and it is practical for generating images of arbitrary motions viewed from translational cameras.

7.3.2 Stability Evaluation

We next show the stability of extracted trifocal tensors under 4D to 2D projections with 13 point method and mutual projection method.

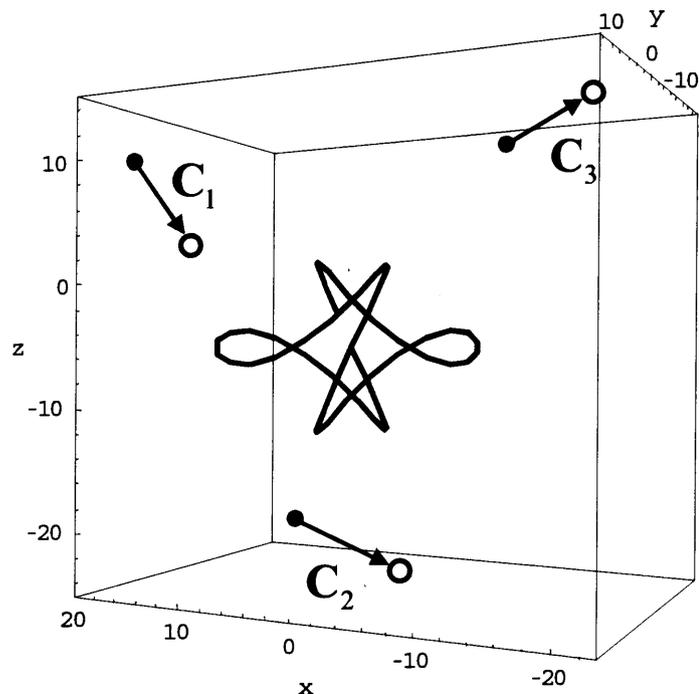


Figure 7.12: Three translating cameras and a moving point in the 3D space. The black points show the viewpoints of three cameras before translational motions, and the white points show those after the translational motions.

Figure 7.12 shows a 3D configuration of 3 moving cameras and a moving point. The black points show the position of three cameras, C_1 , C_2 and C_3 , before translational motions, and the white points show those after the translational motions. The translational motions of these three cameras are different and unknown. The black curve shows a locus of a freely moving point. For evaluating the extracted trifocal tensors, we computed reprojection errors derived from the trifocal tensors. The reprojection error is defined as follows:

$$\frac{1}{N} \sum_{i=1}^N d(\mathbf{m}_i, \hat{\mathbf{m}}_i)^2 \quad (7.96)$$

where $d(\mathbf{m}_i, \hat{\mathbf{m}}_i)$ denotes a distance between a true point \mathbf{m}_i and a point $\hat{\mathbf{m}}_i$ recovered from the trifocal tensor.

The case of mutual projections is still $2 \times \{\mathbf{e}_{21}(t), \mathbf{e}_{31}(t)\} + 2 \times \{\mathbf{e}_{12}(t), \mathbf{e}_{32}(t)\} + 2 \times \{\mathbf{e}_{13}(t), \mathbf{e}_{23}(t)\}$. We increased the number of corresponding points used for computing trifocal tensors in three views from 0 to 25, and evaluated the reprojection errors. In the same way, we also evaluated the 13 point method with same corresponding points from 13 to 25. Gaussian noise of standard deviation of 1 pixel is added to each

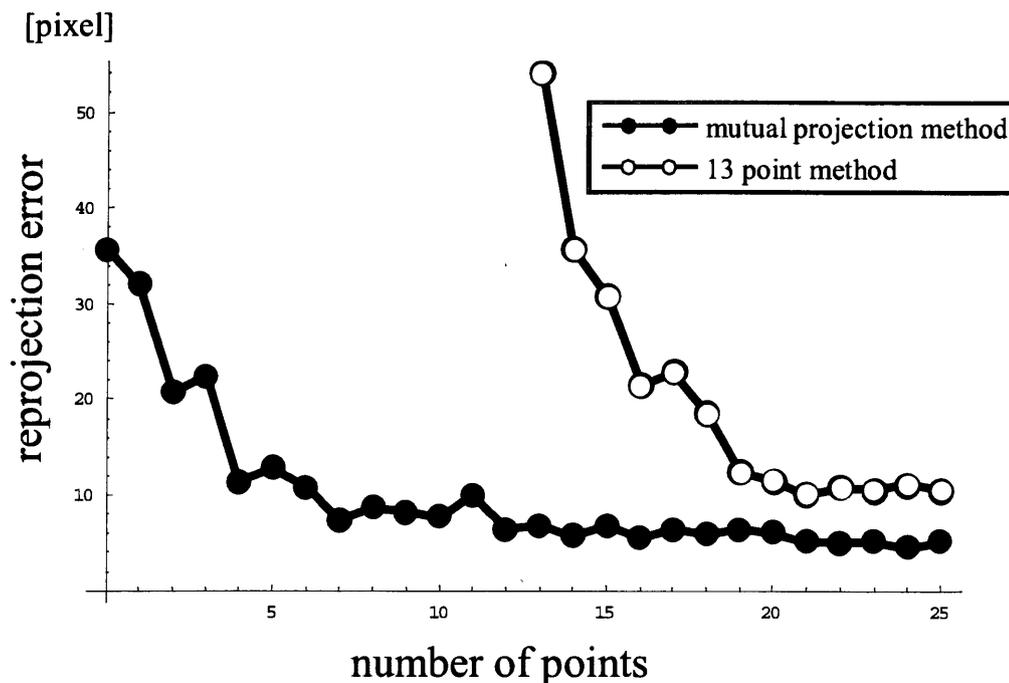


Figure 7.13: The relationship between the number of corresponding points used for computing trifocal tensors and the reprojection errors. The black points show the result from mutual projection method, and the white points show that from 13 point method.

image. Figure 7.13 shows the relationship between the number of corresponding points and the reprojection errors. The black points show the result from mutual projection method, and the white points show that from 13 point method. As we can see, the stability is obviously improved by using a few more points than the minimum number of corresponding points. Moreover, with less or even no corresponding points, the mutual projection method can derive more stable trifocal tensors than the 13 point method.

7.4 Summary

In this chapter, we analyzed the computation of dynamic multiple view geometry in 4D space from mutual projections of multiple cameras. Taking three translational cameras for example, we discussed using one, two and all three epipole pairs at different time how many independent equations we can derive and then how many corresponding points are required to compute the trifocal tensor. As a result, with one epipole pair at 3 different time we need 6 corresponding points, with two epipole pairs we at least

require 2 corresponding points, and when we use three epipole pairs at 2 different time respectively, we do not need any corresponding point to figure out the trifocal tensor. That means arbitrary image motions tracked by moving cameras can be recovered even if they are coplanar or collinear, as long as we have the projections of cameras. The method was implemented and tested by using real image sequences. The stability of extracted trifocal tensors was also evaluated.

Chapter 8

Conclusion

In this thesis, we presented the dynamic multiple view geometry which includes the traditional multiple view geometry and the high-dimension multiple view geometry. We modeled the camera trajectory by Degree- n Bezier curves and made points in 3D undergo non-rigid motions. We found that the projective projections of non-rigid 3D motion to Degree- n Bezier curve can be represented by a projection from $(n + 3)$ D to 2D. If 3D point motions are tracked by multiple arbitrary motion cameras, the multilinear relationship under the projection from $(n + 3)$ D to 2D can be derived. Then, we analyzed the projective projections from $(n + 3)$ D to 2D and deduced the degree of freedom of the extended projective camera. $(n + 3)$ -Dimension multiple view geometry involving several such extended cameras and a dynamic scene was also addressed. Multilinear relationships and the maximal linear relationship in the $(n+3)$ D space were derived from the multifocal point relations. Finally, counting arguments were executed. From the geometric degree of freedom of extended projective cameras and the degree of freedom of the points in $(n+3)$ D and all the images, the minimum number of points required for computing the multifocal tensors were derived.

We next analyzed the dynamic multiple view geometry under projective projections from 4D space to 2D space, and showed that it can represent multiple view geometry under space-time projections, in which the multilinear relationship for 5 views is the maximal linear relationship in the 4D space unlike the traditional multiple view geometry. The new trilinear, quadrilinear and quintilinear relationships were analyzed. We showed that the newly defined multiple view geometry can be used for describing the relationship between images taken from non-rigid motions viewed from multiple translational cameras and is very useful for generating images of non-rigid object motions viewed from arbitrary translational cameras. Here, the multifocal tensors are computed from corresponding points. For instance, the trifocal tensor can be derived by using 13 corresponding points, which are not collinear and coplanar. The method was implemented and tested by using real image sequences. The stability of extracted trifocal tensors was also evaluated.

We also extended the theory of multiple view geometry in space-time, and introduced a multiple view geometry of multiple cameras with arbitrary curvilinear motions.

We used affine camera model and projective camera model to describe the multilinear relationship under the projection from 6D to 2D respectively, which can represent the geometric relationship of multiple curvilinear motion cameras whose motions are represented by cubic Bezier curves. The multifocal tensors defined under 6D to 2D multilinear relationships can be computed from non-rigid object motions viewed from multiple cameras with arbitrary curvilinear motions. We also showed that the multilinear relationships are very useful for generating arbitrary view images and reconstructing 3D non-rigid object motions viewed from cameras with arbitrary curvilinear motions. The method was tested by real images. We also evaluated the stability of extracted quadrifocal tensors.

We have introduced the dynamic multiple view geometry, in which the camera trajectories are modeled by Degree- n Bezier curves. However, when n is large, the multiple view geometry will become very complex and uncomputable. On the other hand, the main problem with Bezier curves is their lack of local control. To overcome the problems, we consider degree- n B-Spline curve, a piecewise curve, to represent the camera trajectories. In the mathematical field of numerical analysis, B-spline curves are very useful for representing arbitrary 3D shapes with small number of control points. Thus, we can use low degree B-spline curve to describe a complex curve. We gave the definition of the B-spline curve and especially took cubic B-spline curve as an instance of to represent the trajectory of the cameras. Although the multiple view geometry corresponding to each segment of B-spline curve motions is same as the case of Bezier curve, the camera motions could be more complex and less control points described if the camera motions are represented by B-spline curves. For example, a 2-segment cubic B-spline curve is smooth, differentiable and depends on 5 control points, while two successive Bezier curves are not differentiable and determined by 7 control points. The synthetic experiment showed that even if all the cameras undergo complex curvilinear motions, the view transfer still can be realized by using the dynamic multiple view geometry.

We also proposed the computation of multiple view geometry in space-time under the case where cameras are projected each other and epipoles are given as the projections of cameras in images. Since all the cameras are dynamic, an epipole at different time has different value. Making use of those values, we worked out many interesting results, such as, in computing the trifocal tensor, if we use one epipole pair at 3 different time, we require at least 6 corresponding points; using two epipole pairs at 3 time respectively, we need only 2 corresponding points; if we have all the epipole pairs at 2 time respectively, the corresponding points are no longer required. In the last two cases, computing the trifocal tensor is not restricted by the relative positions of the corresponding points any more and arbitrary image motions can be recovered. Furthermore, the mutual projection method enables us to obtain the multiple view geometry much more stably.

The dynamic multiple view geometry proposed in this thesis describes the relationship among a dynamic scene and multiple moving cameras, which is a more general configuration than the traditional multiple view geometry. By using it, we can recover

not only the non-rigid object motions in one of the camera views, but also their original shapes in the 3D space. Moreover, it can also be used for measuring and modeling the real objects, recognizing human actions, navigating robots, etc. In the future, we will try to apply the new theory to some fields and further extend it. On the other hand, the proposed theory also has its limitations. For example, the cameras are assumed to have no rotations and the motions of the cameras are constrained by parametric equations, such as Bezier curve or B-spline curve. These problems will be solved in our future works.

Publication List

• Journal

1. C. Wan and J. Sato, Computing Spatio-temporal Multiple View Geometry from Mutual Projections of Multiple Cameras, *IEICE Transactions on Information and Systems*, Vol.E93-D, No.9, pp.2602-2613, Sep. 2010.[93]
2. C. Wan and J. Sato, Multiple View Geometry under Projective Projection in Space-Time, *IEICE Transactions on Information and Systems*, vol.91-D, No.9, pp.2353-2359, 2008.[92]
3. K. Kozuka, C. Wan and J. Sato, Rectification of 3D Data Obtained from Moving Range Sensor by Using Extended Projective Multiple View Geometry, *International Journal of Automation & Computing*, vol.5, No3, pages 268-275, Springer, 2008.[48]

• International Conference

4. C. Wan and J. Sato, Computing Multiple View Geometry for Non-Rigid Motions Viewed from Curvilinear Motion Projective Cameras, *Proc.20th International Conference on Pattern Recognition*, pp.181-184, 2010.[94]
5. C. Wan and J. Sato, Computing multiple view geometry in space-time from mutual projections of multiple cameras, *Proc.19th International Conference on Pattern Recognition*, pp.1-4, 2008.[91]
6. C. Wan, K. Kozuka, and J. Sato. Multiple view geometry for non-rigid motions viewed from translational cameras. *Proc. 8th Asian Conference on Computer Vision*, 4844 of *LNCIS*, pages 342–352. Springer, 2007.[89]
7. K. Kozuka, C. Wan and J. Sato. Rectification of 3D data obtained from moving range sensor by using multiple view geometry under projective projections in space-time. *ACCV'07 Workshop on Multi-Dimensional and Multi-View Image Processing*, 2007.[47]

• Domestic Conference

8. C. Wan and J. Sato. Multiple view geometry under projective projections in space-time. In Proc. 10th Meeting on Image Recognition and Understanding, pages 205–211, 2007.[90] (**Special Session Award**)

Acknowledgments

The work described in this dissertation has been carried out at the Department of Computer Science and Engineering of Nagoya Institute of Technology under the supervision of Professor Jun Sato. I would like to express my sincere thanks to my supervisor Professor Jun Sato for his guidance, expertise, inspiration and patience. I appreciate his vast knowledge and skill in many areas, and his assistance in writing works.

I would also like to thank Professor Hidekata Hontani and Dr Fumihiko Sakaue for fruitful discussion and comments on my research. I wish to thank the other members of our Sato–Hontani Laboratory for their various assistance and friendliness. Although some of them were graduated, I will never forget the scene when I came to Japan, to this laboratory. They gave me a good impression of Japan and this laboratory.

Finally, I am indebted to my family for the support they provided me through my entire life and in particular, I must acknowledge my husband and best friend, Lin, without whose love, encouragement and technical assistance, I would not have finished this dissertation.

References

- [1] D.G. ALIAGA, T. FUNKHOUSER, D. YANOVSKY, AND I. CARLBOM. Sea of images: A dense sampling approach for rendering large indoor environments. *IEEE Computer Graphics and Applications*, **23**(6):22–30, November 2003.
- [2] SHAI AVIDAN AND AMNON SHASHUA. Trajectory triangulation: 3d reconstruction of moving points from a monocular image sequence. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **22**:200–0, 2000.
- [3] E.P. BALTSAVIAS. Multiphoto geometrically constrained matching. In *Ph.D. Thesis*, page Zurich, 1991.
- [4] R. BASRI, A.J. GROVE, AND D.W. JACOBS. Efficient determination of shape from multiple images containing partial information. *Pattern Recognition*, **31**(11):1691–1703, November 1998.
- [5] P.A. BEARDSLEY, J.M. BRADY, AND D.W. MURRAY. Prediction of stereo disparity using optical flow. In *BMVC90*, 1990.
- [6] M. BERTHOD, H. SHEKARFOROUSH, M. WERMAN, AND J.B. ZERUBIA. Reconstruction of high resolution 3d visual information using sub-pixel camera displacements. In *CVPR94*, pages 654–657, 1994.
- [7] G. BRADSKI AND T.E. BOULT. Guest editorial: Stereo and multi-baseline vision. *International Journal of Computer Vision*, **47**(1-3):5–5, April 2002.
- [8] S.S. BRANDT. Conditional solutions for the affine reconstruction of n-views. *Image and Vision Computing*, **23**(7):619–630, July 2005.
- [9] Y.Q. CHENG, X.G. WANG, R.T. COLLINS, E.M. RISEMAN, AND A.R. HANSON. Three-dimensional reconstruction of points and lines with unknown correspondence across images. *International Journal of Computer Vision*, **45**(2):129–156, November 2001.
- [10] C.I. CONNOLLY AND J.R. STENSTROM. Construction of polyhedral models from multiple range views. In *ICPR86*, pages 85–87, 1986.
- [11] C.I. CONNOLLY AND J.R. STENSTROM. 3-d scene reconstruction from multiple intensity images. In *3DWS89*, pages 124–130, 1989.
- [12] CARL DE BOOR. *A Practical Guide to Splines*. Springer-Verlag, 1978.
- [13] M.A. DROUIN, M. TRUDEAU, AND S. ROY. Geo-consistency for wide multi-camera stereo. In *CVPR05*, pages I: 351–358, 2005.
- [14] M.A. DROUIN, M. TRUDEAU, AND S. ROY. Non-uniform hierarchical geo-consistency for multi-baseline stereo. In *CRV07*, pages 208–215, 2007.

- [15] M.A. DROUIN, M. TRUDEAU, AND S. ROY. Improving border localization of multi-baseline stereo using border-cut. *International Journal of Computer Vision*, **83**(3):233–247, July 2009.
- [16] O.D. FAUGERAS AND Q.T. LUONG. *The Geometry of Multiple Images*. MIT Press, 2001.
- [17] O.D. FAUGERAS AND B. MOURRAIN. On the geometry and algebra of the point and line correspondences between n images. In *Proc. 5th International Conference on Computer Vision*, pages 951–956, 1995.
- [18] O.D. FAUGERAS AND T. PAPADOPOULOU. A nonlinear method for estimating the projective geometry of three views. In *ICCV98*, pages 477–484, 1998.
- [19] G. FLOROU AND R. MOHR. What accuracy for 3d measurements with cameras? In *ICPR96*, pages I: 354–358, 1996.
- [20] M.L. GONG AND Y.H. YANG. Genetic-based stereo algorithm and disparity map evaluation. *International Journal of Computer Vision*, **47**(1-3):63–77, April 2002.
- [21] R.I. HARTLEY. Multilinear relationship between coordinates of corresponding image points and lines. *Proc. International Workshop on Computer Vision and Applied Geometry*, 1995.
- [22] R.I. HARTLEY. In defence of the 8-point algorithm. *IEEE Trans. Pattern Analysis and Machine Intelligence*, **19**(6):580–593, 1997.
- [23] R.I. HARTLEY. Minimizing algebraic error in geometric estimation problems. In *Proc. 6th International Conference on Computer Vision*, **1**, pages 469–476, 1998.
- [24] R.I. HARTLEY AND F. SCHAFFALITZKY. Reconstruction from projections using grassman tensors. In *Proc. 8th European Conference on Computer Vision*, **1**, pages 363–375, 2004.
- [25] R.I. HARTLEY AND A. ZISSERMAN. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000.
- [26] RICHARD I. HARTLEY AND FREDERIK SCHAFFALITZKY. Reconstruction from projections using grassmann tensors. *Int. J. Comput. Vision*, **83**(3):274–293, 2009.
- [27] K. HAYAKAWA AND J. SATO. Multiple view geometry in the space-time. In *Proc. 7th Asian Conference on Computer Vision*, pages 437–446, 2006.

- [28] A. HEYDEN. A common framework for multiple view tensors. In *Proc. 5th European Conference on Computer Vision*, **1**, pages 3–19, 1998.
- [29] A. HEYDEN. Tensorial properties of multiple view constraints. *Mathematical Methods in the Applied Sciences*, **23**:169–202, 2000.
- [30] X.Y. HU AND P. MORDOHAJ. Evaluation of stereo confidence indoors and outdoors. In *CVPR10*, pages 1466–1473, 2010.
- [31] K. HUANG, R.M. FOSSUM, AND Y. MA. Generalized rank conditions in multiple view geometry with applications to dynamical scenes. In *ECCV02*, page II: 201 ff., 2002.
- [32] M. IRANI, P. ANANDAN, AND M. COHEN. Direct recovery of planar-parallax from multiple frames. *IEEE Trans. Pattern Analysis and Machine Intelligence*, **24**(11):1528–1534, November 2002.
- [33] H. ISHIGURO, T. SOGO, AND M. BARTH. Baseline detection and localization for invisible omnidirectional cameras. *International Journal of Computer Vision*, **58**(3):209–226, July 2004.
- [34] T.S. HUANG J. WENG AND N. AHUJA. Motion and structure from two perspective views: Algorithms, error analysis and error estimation. *IEEE Trans. Pattern Analysis and Machine Intelligence*, **11**(5), 1989.
- [35] D.W. JACOBS, A.J. GROVE, AND R. BASRI. Efficient determination of shape from multiple images containing partial information. In *ICPR96*, pages I: 268–274, 1996.
- [36] F. KAHL, A. HEYDEN, AND L. QUAN. Minimal projective reconstruction including missing data. *IEEE Trans. Pattern Analysis and Machine Intelligence*, **23**(4):418–424, April 2001.
- [37] T. KANADE AND M. OKUTOMI. A multiple-baseline stereo. In *CVPR91*, pages 63–69, 1991.
- [38] S.B. KANG, R.S. SZELISKI, AND J.X. CHAI. Handling occlusions in dense multi-view stereo. In *CVPR01*, pages I:103–110, 2001.
- [39] J.H. KANNALA AND S.S. BRANDT. Quasi-dense wide baseline matching using match propagation. In *CVPR07*, pages 1–8, 2007.
- [40] J.H. KANNALA, E. RAHTU, S.S. BRANDT, AND J. HEIKKILA. Object recognition and segmentation by non-rigid quasi-dense matching. In *CVPR08*, pages 1–8, 2008.

- [41] A.K. KATSAGGELOS AND C.J. TSAI. Sequential construction of 3-d based scene description. In *ICIP99*, pages II:510–514, 1999.
- [42] V. KOLMOGOROV AND C. ROTHER. Comparison of energy minimization algorithms for highly connected graphs. In *ECCV06*, pages II: 1–15, 2006.
- [43] V. KOLMOGOROV AND R. ZABIH. Computing visual correspondence with occlusions via graph cuts. In *ICCV01*, pages II: 508–515, 2001.
- [44] V. KOLMOGOROV AND R. ZABIH. Multi-camera scene reconstruction via graph cuts. In *ECCV02*, page III: 82 ff., 2002.
- [45] V. KOLMOGOROV AND R. ZABIH. What energy functions can be minimized via graph cuts? *IEEE Trans. Pattern Analysis and Machine Intelligence*, **26**(2):147–159, February 2004.
- [46] P. KOSKENKORVA, J.H. KANNALA, AND S.S. BRANDT. Quasi-dense wide baseline matching for three views. In *ICPR10*, pages 806–809, 2010.
- [47] K. KOZUKA, C. WAN, AND J. SATO. Rectification of 3d data obtained from moving range sensor by using multiple view geometry under projective projections in space-time. *ACCV'07 Workshop on Multi-Dimensional and Multi-View Image Processing*, 2007.
- [48] K. KOZUKA, C. WAN, AND J. SATO. Rectification of 3d data obtained from moving range sensor by using extended projective multiple view geometry. *International Journal of Automation and Computing*, **5**(3):268–275, 2008.
- [49] S. LAZEBNIK, Y. FURUKAWA, AND J. PONCE. Projective visual hulls. *International Journal of Computer Vision*, **74**(2):137–165, August 2007.
- [50] B. LIU, M. YU, D. MAIER, AND R. MANNER. An efficient and accurate method for 3d-point reconstruction from multiple views. *International Journal of Computer Vision*, **65**(3):175–188, December 2005.
- [51] Q.T. LUONG AND O.D. FAUGERAS. The fundamental matrix: Theory, algorithm and stability analysis. *International Journal of Computer Vision*, **17**(1):43–76, 1996.
- [52] Y. MA. A differential geometric approach to multiple view geometry in spaces of constant curvature. *International Journal of Computer Vision*, **58**(1):37–53, June 2004.
- [53] S. MANN AND R.W. PICARD. Virtual bellows: Constructing high quality stills from video. In *ICIP94*, pages I: 363–367, 1994.

- [54] S. MANN AND R.W. PICARD. Video orbits of the projective group: A simple approach to featureless estimation of parameters. *IEEE Trans. Image Processing*, **6**(9):1281–1295, September 1997.
- [55] S.J. MAYBANK. *Theory of Reconstruction from Image Motion*. Springer-Verlag, 1993.
- [56] J.E.W. MAYHEW, J. PORRILL, AND S.B. POLLARD. Recovering partial 3d wire frame descriptions from stereo data. In *BMVC90*, 1990.
- [57] J.E.W. MAYHEW AND N.A. THACKER. Optimal combination of stereo camera calibration from arbitrary stereo images. In *BMVC90*, 1990.
- [58] R. MOHR, B.S. BOUFAMA, AND P. BRAND. Understanding positioning from multiple images. *Artificial Intelligence*, **78**(1-2):213–238, October 1995.
- [59] P. MORDOHAI. The self-aware matching measure for stereo. In *ICCV09*, pages 1841–1848, 2009.
- [60] P. MORDOHAI AND G. MEDIONI. Perceptual grouping for multiple view stereo using tensor voting. In *ICPR02*, pages III: 639–644, 2002.
- [61] P. MORDOHAI AND G. MEDIONI. Stereo using monocular cues within the tensor voting framework. *IEEE Trans. Pattern Analysis and Machine Intelligence*, **28**(6):968–982, June 2006.
- [62] D.W. MURRAY AND P.A. BEARDSLEY. Range recovery using virtual multi-camera stereo. In *BMVC92*, pages 29–38, 1992.
- [63] K.C. NG, M. TRIVEDI, AND H. ISHIGURO. Generalized multiple baseline stereo and direct virtual view synthesis using range-space search, match, and render. *International Journal of Computer Vision*, **47**(1-3):131–147, April 2002.
- [64] M. OKUTOMI AND T. KANADE. A multiple-baseline stereo. *IEEE Trans. Pattern Analysis and Machine Intelligence*, **15**(4):353–363, April 1993.
- [65] F. PEDERSINI, P. PIGAZZINI, A. SARTI, AND S. TUBARO. Multicamera motion estimation for high-accuracy 3d reconstruction. *Signal Processing*, **80**(1):1–21, January 2000.
- [66] F. PEDERSINI, A. SARTI, AND S. TUBARO. 3d motion estimation of a trinocular system for a full-3d object reconstruction. In *ICIP96*, pages II: 867–870, 1996.
- [67] F. PEDERSINI, A. SARTI, AND S. TUBARO. Egomotion estimation of a multicamera system through line correspondences. In *ICIP97*, pages II: 175–178, 1997.

- [68] F. PEDERSINI, A. SARTI, AND S. TUBARO. Accurate feature detection and matching for the tracking of calibration parameters in multi-camera acquisition systems. In *ICIP98*, pages II: 598–602, 1998.
- [69] F. PEDERSINI, A. SARTI, AND S. TUBARO. Multi-camera parameter tracking. *IEE Proceedings-Vision Image and Signal Processing*, **148**(1):70–77, February 2001.
- [70] L. QUAN, A. HEYDEN, AND F. KAHL. Minimal projective reconstruction with missing data. In *CVPR99*, pages II: 210–216, 1999.
- [71] S. ROY. Stereo without epipolar lines: A maximum-flow formulation. *International Journal of Computer Vision*, **34**(2-3):147–161, August 1999.
- [72] S. ROY AND I.J. COX. A maximum-flow formulation of the n-camera stereo correspondence problem. In *ICCV98*, pages 492–499, 1998.
- [73] S. ROY, M. TRUDEAU, AND M.A. DROUIN. Improving border localization of multi-baseline stereo using border-cut. In *CVPR06*, pages I: 511–518, 2006.
- [74] J. SATO. *Computer Vision—Visual Geometry*. Corona Ltd, 1999.
- [75] J. SATO. Recovering multiple view geometry from mutual projections of multiple cameras. *International Journal of Computer Vision*, **66**(2):123–140, 2006.
- [76] D. SCHARSTEIN AND C.J. PAL. Learning conditional random fields for stereo. In *CVPR07*, pages 1–8, 2007.
- [77] M. SCHRAMECK, R. VOYLES, T. MYERS, R. BODOR, AND O. MASOUD. Automatic euclidean reconstruction for turn-table sequences by indirect epipolar search between pairs of views. *Image and Vision Computing*, **24**(7):693–708, July 2006.
- [78] A. SHASHUA. Trilinearity in visual recognition by alignment. In *Proc. 4th European Conference on Computer Vision*, **1**, pages 479–484, 1994.
- [79] A. SHASHUA AND L. WOLF. Homography tensors: On algebraic entities that represent three views of static or moving planar points. In *Proc. 6th European Conference on Computer Vision*, **1**, pages 507–521, 2000.
- [80] A. SHASHUA AND M. WERMAN. Trilinearity of three perspective views and its associated tensor. In *Proc. 4th European Conference on Computer Vision*, **1**, pages 920–925, 1995.
- [81] J.R. STENSTROM AND C.I. CONNOLLY. Constructing object models from multiple images. *International Journal of Computer Vision*, **9**(3):185–212, 1992.

- [82] P. STURM. Multi-view geometry for general camera models. In *Proc. Conference on Computer Vision and Pattern Recognition*, pages 206–212, 2005.
- [83] A. SUGIMOTO. Object recognition by combining paraperspective images. *International Journal of Computer Vision*, **19**(2):181–201, August 1996.
- [84] R.S. SZELISKI, R. ZABIH, D. SCHARSTEIN, O. VEKSLER, V. KOLMOGOROV, A. AGARWALA, M. TAPPEN, AND C. ROTHER. A comparative study of energy minimization methods for markov random fields with smoothness-based priors. *IEEE Trans. Pattern Analysis and Machine Intelligence*, **30**(6):1068–1080, June 2008.
- [85] CARLO TOMASI. Shape and motion from image streams under orthography: a factorization method. *International Journal of Computer Vision*, **9**:137–154, 1992.
- [86] P. TORR. *Motion segmentation and outlier detection*. PhD thesis, University of Oxford, 1995.
- [87] Y. TSIN, S.B. KANG, AND R.S. SZELISKI. Stereo matching with reflections and translucency. In *CVPR03*, pages I: 702–709, 2003.
- [88] Y. TSIN, S.B. KANG, AND R.S. SZELISKI. Stereo matching with linear superposition of layers. *IEEE Trans. Pattern Analysis and Machine Intelligence*, **28**(2):290–301, February 2006.
- [89] C. WAN, K. KOZUKA, AND J. SATO. Multiple view geometry for non-rigid motions viewed from translational cameras. In *Proc. 8th Asian Conference on Computer Vision*, **4844** of *LNCS*, pages 342–352. Springer, 2007.
- [90] C. WAN AND J. SATO. Multiple view geometry under projective projections in space-time. In *Proc. 10th Meeting on Image Recognition and Understanding*, pages 205–211, 2007.
- [91] C. WAN AND J. SATO. Computing multiple view geometry in space-time from mutual projections of multiple cameras. In *Proc. 19th International Conference on Pattern Recognition*, pages 1–4, 2008.
- [92] C. WAN AND J. SATO. Multiple view geometry under projective projection in space-time. *IEICE Transactions on Information and Systems*, **91-D**(9):2353–2359, 2008.
- [93] C. WAN AND J. SATO. Computing spatio-temporal multiple view geometry from mutual projections of multiple cameras. *IEICE Transactions on Information and Systems*, **E93-D**(9):2602–2613, September 2010.

- [94] CHENG WAN AND JUN SATO. Multiple view geometry for non-rigid motions viewed from curvilinear motion projective cameras. In *Proc. 20th International Conference on Pattern Recognition*, pages 181–184, 2010.
- [95] X.G. WANG, Y.Q. CHENG, R.T. COLLINS, AND A.R. HANSON. Determining correspondences and rigid motion of 3d point sets with missing data. In *CVPR96*, pages 252–257, 1996.
- [96] J.J. WEINMAN, L. TRAN, AND C.J. PAL. Efficiently learning random fields for stereo vision with sparse message passing. In *ECCV08*, pages I: 617–630, 2008.
- [97] Y. WEXLER AND A. SHASHUA. On the synthesis of dynamic scenes from reference views. In *Proc. Conference on Computer Vision and Pattern Recognition*, pages 576–581, 2000.
- [98] L. WOLF AND A. SHASHUA. On projection matrices $p^k \rightarrow p^2$, $k = 3, \dots, 6$, and their applications in computer vision. In *Proc. 8th International Conference on Computer Vision*, 1, pages 412–419, 2001.
- [99] G. XU, S. TSUJI, AND M. ASADA. Coarse to fine control strategy for matching motion stereo pairs. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 9(2):332–336, March 1987.
- [100] Z. ZHANG. Determining the epipolar geometry and its uncertainty: A review. *International Journal of Computer Vision*, 27(2):161–195, 1987.