

Doctoral Dissertation

**Reconstruction of Texture Objects
without knowing
Image Correspondences**

Saba Batool Miyan

2015

Department of Computer Science & Engineering

Nagoya Institute of Technology

A Doctoral Dissertation
submitted to Department of Computer Science and Engineering
Nagoya Institute of Technology
In partial fulfillment of the requirements for the degree of
Doctor of ENGINEERING

Supervisor: Jun Sato, Professor

Contents

1	Introduction	1
1.1	Motivation	1
1.2	Sequential Patterns	2
1.3	Correspondence Problem	4
1.4	Approach and Assumption	6
1.5	Abbreviations	7
1.6	Symbols	8
1.7	Thesis Overview	10
2	Classical 3D Reconstruction	12
2.1	Object and Image Space	12
2.2	Camera	13
2.2.1	Camera Parameters	14
2.2.2	Camera Matrix Properties	15
	Camera Center	16
	Camera Matrix Columns	17
	Camera Matrix Rows	17
	Principal Axis and Principal Point	18
2.3	Affine Camera	18
2.3.1	Affine Camera Approximation	19
	Formation of Affine Camera	19
	Affine Approximation Error	21
	Anatomy of Affine Camera	24
	Affine Camera Properties	25
2.4	Back Projection from 2D images	26
2.5	Multiview Relations	27
2.5.1	Multiple view geometry	27
2.5.2	Multiview Constraints	28
2.5.3	Algebraic verification	31
2.5.4	Fundamental Matrix	32
	Computation of Fundamental Matrix	33

2.5.5	Minimum No. of Points	33
2.5.6	Maximum No. of Cameras	34
2.6	3D Reconstruction	34
2.6.1	Retrieval of Camera Matrices	34
2.6.2	Computation of Epipoles	35
2.6.3	Extraction of Homography	36
2.7	Affine Reconstruction and Rectification	37
3	Reconstructing Grayscale Objects without knowing Image Correspondences	39
3.1	Object and Image Spaces	40
3.1.1	4D World	40
3.1.2	3D Image	41
3.1.3	Object and Image Spaces considering Sequential Patterns	41
3.1.4	Object and Image Spaces considering Texture Patterns	42
3.2	Affine Camera from 4D to 3D Space	43
3.3	Projection in Frequency Domain	43
3.3.1	Projection in Frequency Domain for Sequential Patterns (1D-DFT)	45
3.3.2	Projection in Frequency Domain for Texture Patterns (2D-DFT)	46
3.4	Back Projection from 3D images	48
3.5	Multiview relations for Sequential Patterns	48
3.5.1	Geometric derivation	49
3.5.2	Algebraic verification	51
3.5.3	Generalization to Texture Patterns	53
3.5.4	Bifocal tensor	53
3.5.5	Computation of tensor	54
3.5.6	Minimum No. of Points for Tensor Computation	55
3.5.7	Maximum No. of Cameras	56
3.6	Sampling shift	56
3.6.1	Sampling shift for Sequential Patterns	57
3.6.2	Sampling shift for Texture Patterns	58
3.7	4D Reconstruction	59
3.7.1	Computation of Epipoles	60
3.7.2	Extraction of Homography	61
3.7.3	4D Reconstruction and Rectification	61
3.8	Experiments	62
3.8.1	Sequential Patterns	63
	Synthetic Image Experiment	63
	Real Image Experiment	66
	Stability Evaluation	68
3.8.2	Texture Patterns	70

	Synthetic Image Experiment	70
	Real Image Experiment	73
	Accuracy Evaluation	74
3.9	Summary	75
4	Reconstructing Color Objects without knowing Image Correspondences	78
4.1	Object and Image Spaces	78
4.1.1	6D World	78
4.1.2	5D Image	79
4.1.3	Color Sequential Patterns	80
4.1.4	Color Texture Patterns	80
4.2	Affine camera from 6D to 5D space	80
4.3	Projection in Frequency Domain	82
4.3.1	Projection in Frequency Domain for Sequential Patterns (1D-DFT)	82
4.3.2	Projection in Frequency Domain for Texture Patterns (2D-DFT)	84
4.4	Back Projection from 5D images	85
4.5	Multiview relations for Sequential Patterns	86
4.5.1	Geometric derivation	86
4.5.2	Algebraic verification	89
4.5.3	Bifocal Tensor	91
	Bifocal Tensor - I	92
	Bifocal Tensor - II	95
4.5.4	Computation of Bifocal Tensor	98
4.5.5	Independent Equations	100
4.5.6	Minimum Number of Points for Tensor Computation	101
	Bifocal Tensor - I	101
	Bifocal Tensor - II	101
4.5.7	Maximum No. of Cameras	102
4.6	Sampling shift	103
4.6.1	Sampling shift for Sequential Patterns	103
4.6.2	Sampling shift for Texture Patterns	104
4.7	6D Reconstruction	104
4.7.1	Retrieval of Camera Matrices	105
4.7.2	Computation of Epipoles	105
4.7.3	Extraction of Homography	106
4.7.4	Affine Reconstruction and Rectification	107
4.8	Experiments	108
4.8.1	Color Sequential Pattern	108
	Synthetic Image Experiment	108
	Real Image Experiment	111

	Stability Evaluation	113
4.8.2	Color Texture Patterns	114
	Synthetic Image Experiment	114
	Real Image Experiment	117
	Stability Evaluation	118
4.9	Summary	119
5	Conclusion	121
A	Theory	123
A.1	Homogeneous Coordinates	123
A.2	Matrix operations	124
A.3	Discrete Fourier Transform	125
A.4	Tensor and Tensor notation	126
A.5	Fundamental Matrix vs. Essential Matrix	127
B	Experiments	130
B.1	Normalization and Denormalization	130
	B.1.1 Normalization of image data	130
	B.1.2 Denormalization of Tensor	131
B.2	Camera Gain, Offset and Crosstalk	132
B.3	Gamma Transformation and Correction	135
	B.3.1 Gamma Transformation	135
	B.3.2 Gamma Correction	136
B.4	Blob detection using mathematica	137
B.5	Connecting image points	139
	B.5.1 Connecting image points for sequences	140
	B.5.2 Connecting image points for textures	140
	Acknowledgements	142
	List of Publications	143
	List of Figures	144
	List of Tables	150
	References	151

Chapter 1

Introduction

1.1 Motivation

The word reconstruction in classical multiple view geometry (MVG) [1, 17, 19, 21, 30] refers to generation of a 3 dimensional (3D) world from 2 dimensional (2D) images. Usually, structure or geometric information is the subject of study. Many methods exist to perform the task. Common examples are "Structure from X ", where X could be motion [1, 2, 3, 7, 19, 21], texture [35], shading [36], etc.

Reconstruction has 3 main steps, correspondence, motion and structure. Correspondence means finding the 2D location of a 3D point in multiple images. Motion refers to estimating the camera parameters. Structure means recovering scene geometry i.e. 3D reconstruction. After reconstruction arbitrary views can be generated from different viewpoints. Figure 1.1 shows the process of correspondence establishment, reconstruction and arbitrary view generation.

It would be worth mentioning that traditional multiview relations are in spatial domain and are capable of reconstructing geometric information of the 3D world. The accuracy of reconstruction depends on the accuracy of image points and the correctness of correspondence among these image points. In particular, wrong image correspondence causes completely wrong 3D reconstruction [4, 26]. Therefore, the correctness of image correspondence is quite important in 3D reconstruction.

If we have a large variety in image intensity, it is not so difficult to obtain accurate image correspondence in multiple images. However, if the object has repetitive tex-

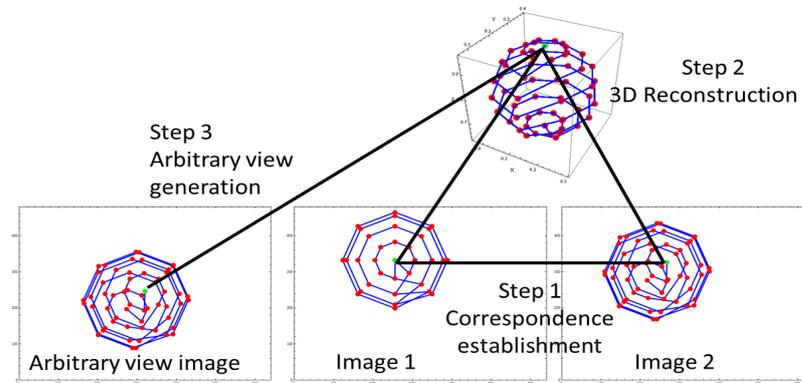


Figure 1.1: The process of reconstruction and arbitrary view synthesis. Corresponding points, reconstructed 3D point and its projection in arbitrary view are joined by black lines.

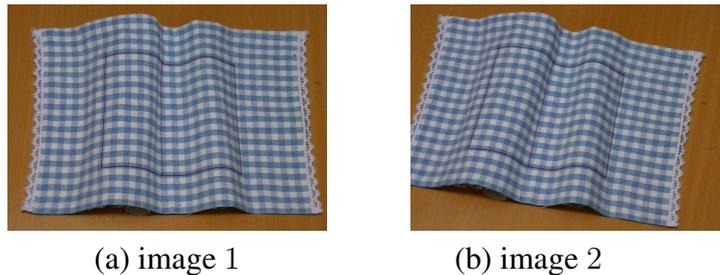


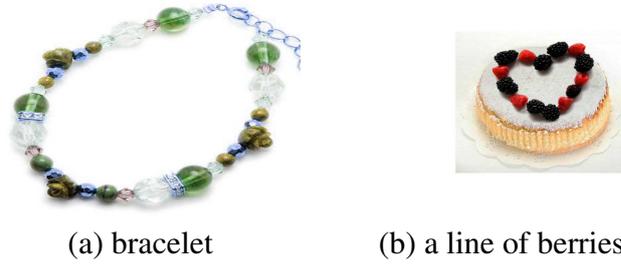
Figure 1.2: A texture patterns from two different viewpoints.

ture pattern as shown in Fig. 1.2, it is quite difficult to obtain correct correspondence in multiple images, since we have many similar patterns in images. This is called correspondence problem, and is still a major problem in computer vision. In this thesis, we tackle the correspondence problem in repetitive patterns, and propose methods for reconstructing texture objects while relaxing the correspondence problem.

1.2 Sequential Patterns

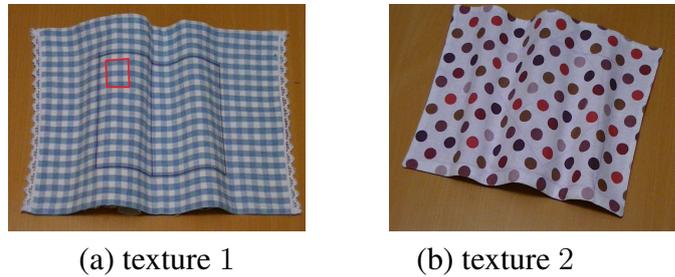
In this thesis, we consider the reconstruction of the following 2 types of objects:

1D Sequential Patterns: A series of grayscale/color 3D points is termed as a 1D sequence. It can also be referred to as 1D sequential texture. When we have multiple



(a) bracelet (b) a line of berries

Figure 1.3: Example of sequential patterns



(a) texture 1 (b) texture 2

Figure 1.4: Example of texture patterns.

objects in the 3D scene, we often have the order of objects and their order is visible in images. For example, a bracelet shown in Fig. 1.3 (a) consists of some small 3D objects of various colors, and the order of these objects is visible in the image. The line of berries shown in Fig. 1.3 (b) is another example. In this research, we propose a reconstruction method for such sequential patterns from multiple images. Two kinds of sequential patterns are considered, grayscale and color sequential patterns depending upon the camera used. Elements of a grayscale sequence are 4 dimensional and elements of a color sequence are 6 dimensional.

2D Sequential Patterns (Texture patterns): 2D Sequential patterns or texture patterns are considered as combination of geometric and photometric information on 3D curved surfaces. We assume that texture elements are connected in a 2D sampling grid which is visible in images. Figures 1.4 (a) and (b) show examples of texture patterns. A texture patch is defined to be a 2D array/grid of points on a 3D surface as shown by red in Fig. 1.4 (a). Two types of texture patterns are considered, grayscale texture patterns are 4 dimensional and color texture patterns are 6 dimensional.

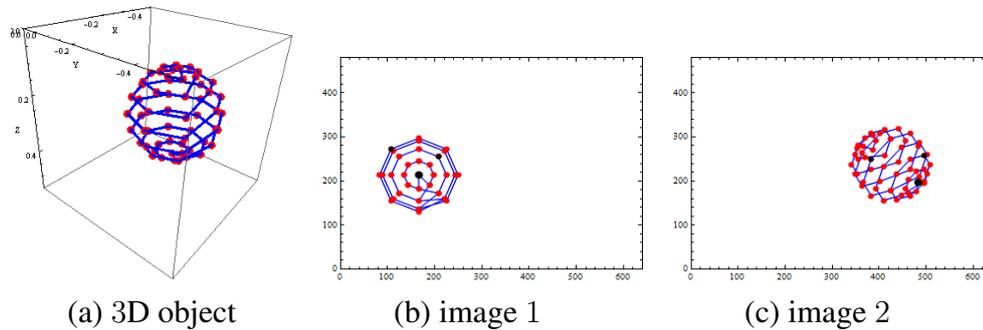


Figure 1.5: 3D object projected to 2 images. Black points in image 1 and image 2 show position of 1st, 15th and 29th points.

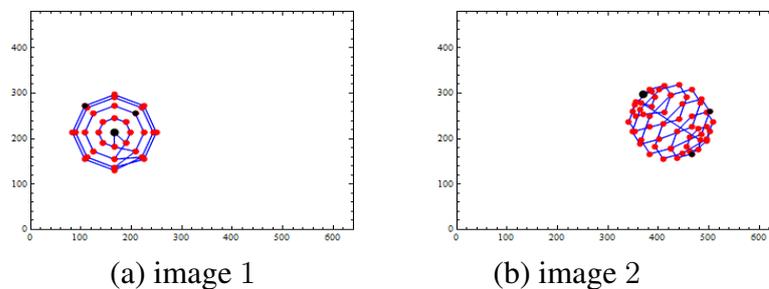


Figure 1.6: Sampling of image 2 is shifted by 5 units. Black points show position of 1st, 15th and 29th points.

1.3 Correspondence Problem

Traditionally, a set of discrete points in two or more than two images are used for reconstructing a 3D object. These points will be referred to as **samples** or **sample points** and their sequence in an image as **sampling order** or **sequence order**. Figure 1.5 (b) and (c) show sample points obtained from a 3D object shown in Fig. 1.5 (a). Let us consider an example when the two images view the same set of 3D points but the sequence order is different as shown in Fig. 1.6. In other words the sampling order of image 2 is shifted. Then, reconstruction would be terribly distorted as shown in Fig. 1.7, even if the sampling order is shifted only by 5 units. This is known as correspondence problem due to sampling shift, and it easily occurs in sequential patterns such as Fig. 1.3.

Finding correspondence in the sequential pattern is difficult due to two factors. The sequential patterns are often repetitive and set of similar intensities/color repeat in a single sequential pattern. Hence, finding the correspondence of points among multiple

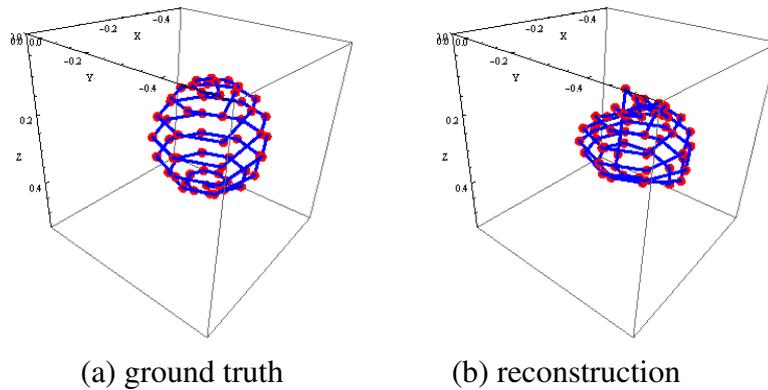


Figure 1.7: 3D reconstruction under sampling shift. The sequence order of image 2 was shifted with 5 units, therefore reconstruction is distorted.



Figure 1.8: The same scene is observed differently in each camera image because of the difference in intensity parameters of camera.

images is not easy in general. Second is the intensity calibration problem. The image intensity changes according to the intensity parameters of camera, and these parameters are different in each camera. As a result, the same scene is observed differently in each camera image as shown in Fig. 1.8. To cope with the problem, camera intensity parameters are often calibrated before using image intensity information. However, the calibration of intensity parameters is not easy and is time consuming. Also, the intensity calibration is not always available.

2D sequential patterns i.e. texture patterns have the same difficulties. Repetitive texture patterns contain many areas (**patches**) with similar intensity/color and distinguishing these areas is not easy. Figure 1.9 shows one such example of a texture object imaged from different viewpoints. In these images the patches outlined in red are cor-

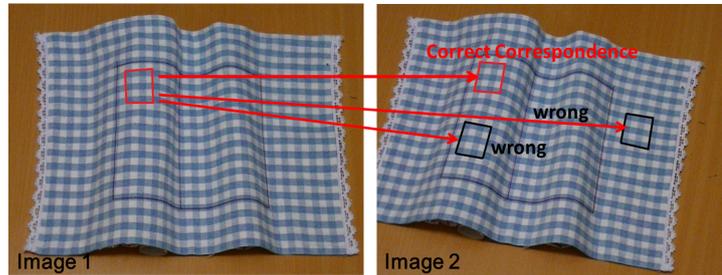


Figure 1.9: Corresponding texture patches in two images. Red patches are corresponding. Black patches in image 2 are false corresponding patches.

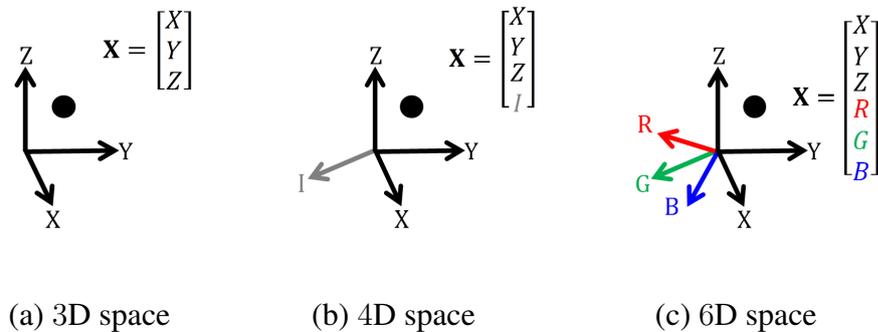


Figure 1.10: World space.

responding to each other. However, a sound criteria must be established to find these corresponding patches, as there are many patches which look nearly the same. For instance, the black patches in image 2 have different geometric positions but have similar visual appearance with the red patch.

1.4 Approach and Assumption

To cope with the correspondence problem we extend the classical 3D reconstruction method in 2 folds. Firstly, the reconstruction of texture objects is performed in higher dimensional space in this research. For grayscale texture, we consider 4D space which consists of the usual 3D space and a 1D intensity space, as shown in Fig. 1.10 (b). For color texture, we consider 6D space, which consists of the usual 3D space and RGB color space as shown in Fig. 1.10 (c). A projection from 4D to 3D space is considered

for grayscale texture and a projection from 6D to 5D is considered for color texture.

The unification of geometrical and photometrical information in frequency space has following benefits:

1. First it saves us from calibration of camera geometry and camera sensitivity parameters, which is required in the existing 3D reconstruction method.
2. It provides us a method for reconstructing the geometrical and photometrical information of an object simultaneously without knowing image correspondences.

Secondly, we consider camera projection and reconstruction not in the usual space, but in the frequency space, whose coordinates consists of frequencies. The use of frequency space enables us to reconstruct sequential patterns without knowing point correspondences.

For deriving our new method, we introduce the following assumptions in this research:

1. Cameras are affine.
2. Cameras observe same spatial domain, in which the sampling grid or line is given.
3. Surfaces are Lambertian.

1.5 Abbreviations

Table 1.5 summarizes abbreviations used in this thesis.

Table 1.1: Abbreviations used in text.

No.	Abbreviation	Term
1	D	dimensions
2	DOF	degrees of freedom

cont.

3	COP	center of projection
4	LOS	line of sight
5	TCC	tensor consistency check
6	SIFT	scale invariant feature transform
7	SURF	speeded up robust features
8	MVG	multiple view geomtry
9	SD	spatial domain
10	FD	Fourier domain
11	DFT	discrete Fourier transform
12	1D-DFT	1 dimensional discrete Fourier transform
13	2D-DFT	2 dimensional discrete Fourier transform
14	RGB	red green blue (color channels)
15	no.	number
16	s.t.	such that
17	w.r.t.	with respect to
18	cont.	continued
19	conf.	configuration
20	col./cols.	columns
21	def.	definition
22	ref.	refer
23	vs.	versus

1.6 Symbols

Following is a list of important symbols and their tensor representations.

Table 1.2: Important symbols and their tensor representations.

No.	Symbol	Tensor Representation	Quantity
1	\mathbf{X}	X^i	world point in SD
2	$\mathbf{X}(k)$	$X^i(k)$	sequential point SD
3	$\mathbf{X}(k, l)$	$X^i(k, l)$	texture point in SD
4	\mathbf{Z}	Z^i	world point in FD
5	$\mathbf{Z}(n)$	$Z^i(n)$	sequential point FD
6	$\mathbf{Z}(n, m)$	$Z^i(n, m)$	texture point in FD
7	\mathbf{S}	S_i	plane/hyperplane in world
8	\mathbf{x}	x^i	image point in SD
9	\mathbf{z}	z^i	image point in FD
10	$\mathbf{z}(n)$	$z^i(n)$	sequential image point in FD
11	$\mathbf{z}(n, m)$	$z^i(n, m)$	texture image point in FD
12	\mathbf{e}	e^i	epipole in image
13	\mathbf{l}	l_i, l_{ij}, l_{ijkl}	line in image
14	\mathbf{s}	s_i	plane/hyperplane in image
15	$\lambda(n)$	$\lambda(n)$	sampling shift in 1D sequential points
16	$\lambda(n, m)$	$\lambda(n, m)$	sampling shift in texture element
17	\mathbf{P}	P_j^i	camera matrix, i is col., j is row
18	\mathbf{H}	H^{ij}, H_j^i	homography between images
19	$\mathbf{0}$	$0_i, 0_{ij}$	zero vector and matrix
20	ϵ	$\epsilon_{ijk}, \epsilon^{ijk}, \epsilon_{ijkl}, \dots$	Levi-Civita tensor (def. in Appendix A.4)
21	\mathbf{F}	F_{ij}	2 view tensor of classical MVG

cont.

22	\mathcal{T}	\mathcal{T}_{fij}	2 view tensor of extended MVG (grayscale images)
23	\mathcal{T}	\mathcal{T}_{ijkl}	2 view tensor of extended MVG (color images)

1.7 Thesis Overview

The thesis consists of 5 chapters and 2 appendices. Experiments are provided at the end of each chapter.

Chapter 1 introduces the research, main terms of the text and assumptions.

Chapter 2 summarizes the traditional 2 view geometry using affine camera. Affine cameras are a useful approximation of general projective cameras. Multiview relations are derived in spatial domain and traditional reconstruction is shown.

Chapter 3 describes the theory for 4 dimensional to 3 dimensional affine projection. It is used for reconstruction of grayscale objects. One dimensional Fourier transform is used for sequences elements and 2 dimensional Fourier transform is used for texture elements. Multiview constraints are derived in frequency space and 4D reconstruction is explained. Method to establish correspondences across images is also discussed. Evaluation of new multiview tensor is shown using stability criteria.

Chapter 4 shows the case of 6 dimensional to 5 dimensional affine projection. It is used for reconstruction of color objects. Multiview relations are derived in frequency space and 6D reconstruction is shown. Method to establish correspondences across images is also discussed. Stability of tensor is also explained.

Appendix A has material supplementing Chap. 2 to Chap. 4.

Appendix B contains some details of the preliminary steps for experiments performed.

Chapter 2

Classical 3D Reconstruction

In this chapter traditional two view geometry and 3D reconstruction for affine cameras is explained. The classical reconstruction method is explained in many existing articles such as [1, 17, 19, 21, 30].

2.1 Object and Image Space

In classical multiple view geometry a point \mathbf{X} in world is defined in 3D space X , Y and Z , and its projection in image is described in 2D space x and y . Using homogeneous coordinates a 3D point and its 2D projection are expressed as vectors in matrix notation as follows;

$$\mathbf{X} = \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (2.1)$$

$$\mathbf{x} = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (2.2)$$

Next, we consider special point forms.

- In 3D world. For any arbitrary scalar W ,

- points of form $[X, 0, 0, W]^\top$ lie on X –axis
 - points of form $[0, Y, 0, W]^\top$ lie on Y –axis
 - points of form $[0, 0, Z, W]^\top$ lie on Z –axis
 - world origin (intersection of three axes) is of form $[0, 0, 0, 1]^\top$.
- In 2D image. For any arbitrary scalar w ,
 - points of form $[x, 0, w]^\top$ are present on x –axis
 - points of form $[0, y, w]^\top$ are present on y –axis

In tensor notation the points are represented with contravariant indices. For example, a 3D point and a 2D point have representations X^i and x^i respectively. Further details on tensor notation can be found in Appendix A.4. For homogeneous coordinates kindly refer Appendix A.1.

2.2 Camera

A traditional camera projects a 3D point in world to a 2D image point, as illustrated in Fig. 2.1. This projection is expressed mathematically as Eq. (2.3). The image geometric characteristics are dependent on camera internal characteristics such as focal length, etc. and the position of camera w.r.t world origin (external characteristics). These parameters determine the type of camera and are consolidated in a 3×4 matrix. Equation (2.4) shows the general camera matrix \mathbf{P} .

$$\mathbf{x} = \mathbf{P}\mathbf{X} \tag{2.3}$$

$$\mathbf{P} = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \end{bmatrix} \tag{2.4}$$

A general camera has 12 elements and 11 degrees of freedom (DOF), except a scale.

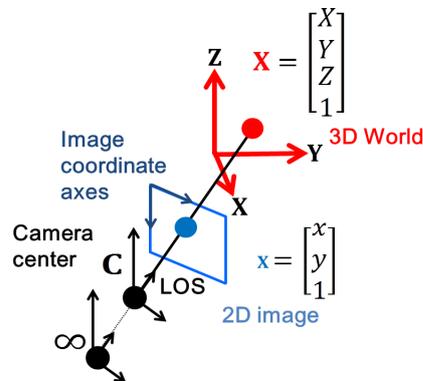


Figure 2.1: A general camera projection. The line joining the camera center C and image point x is termed as line of sight (LOS). The figure shows the camera center at a finite position for general cameras and at infinity for affine cameras.

2.2.1 Camera Parameters

The camera defined in Eq. (2.4) can be defined as a composition of intrinsic parameters K and extrinsic parameters M_{ext} .

$$P = KM_{\text{ext}} \quad (2.5)$$

Internal parameters include scale factors, skew and principal point. The five internal parameters are written as a 3×3 upper triangular matrix as shown in Eq. (2.6).

$$K = \begin{bmatrix} m_x & 0 & 0 \\ 0 & m_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} f & 0 & p_x \\ 0 & f & p_y \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \alpha_x & s & x_0 \\ 0 & \alpha_y & y_0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.6)$$

where,

- $[x_0, y_0]^T$ is the **principal point** (image center).
- α_x and α_y are the **scale factors** in x and y coordinates, α_y/α_x is termed as the **aspect ratio**.
- s is the **skew**. Generally s is zero, leading to 4 DOF. This means the pixels of image are square. In cases, where s is non-zero the image axes x and y are non-perpendicular and pixels are non-square. An example is taking image of an

image.

The matrix $\text{diag}(m_x, m_y, 1)$ converts the image coordinates to pixel form, m_x, m_y represents the no. of pixels per unit distance in x and y axes respectively, f is **focal length** and $[p_x, p_y]^\top$ represents the image center or principal point in metric dimensions.

Commonly, it is assumed that $\alpha_x = \alpha_y$ and $s = 0$. Equation (2.7) shows an example of camera internal parameter matrix.

$$\mathbf{K} = \begin{bmatrix} 500 & 0 & 250 \\ 0 & 500 & 250 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.7)$$

External parameters are rotation and translation. Rotation is represented as a 3×3 matrix, a product of 3 matrices R_x, R_y and R_z . These matrices represent rotation along X, Y and Z axes as shown in Eq. (2.9). Translation is represented as a 3×1 vector. Rotation and translation are often combined together to form a 3×4 matrix, as shown in Eq. (2.8).

$$\mathbf{M}_{\text{ext}} = \begin{bmatrix} R_{11} & R_{12} & R_{13} & T_X \\ R_{21} & R_{22} & R_{23} & T_Y \\ R_{31} & R_{32} & R_{33} & T_Z \end{bmatrix} \quad (2.8)$$

where the matrix \mathbf{R} is the orthogonal **Rotation matrix** defined as follows:

$$\mathbf{R} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta_X & -\sin\theta_X \\ 0 & \sin\theta_X & \cos\theta_X \end{bmatrix} \begin{bmatrix} \cos\theta_Y & 0 & \sin\theta_Y \\ 0 & 1 & 0 \\ -\sin\theta_Y & 0 & \cos\theta_Y \end{bmatrix} \begin{bmatrix} \cos\theta_Z & -\sin\theta_Z & 0 \\ \sin\theta_Z & \cos\theta_Z & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.9)$$

2.2.2 Camera Matrix Properties

A camera matrix defined in Eq. (2.4) can be written as follows:

$$\mathbf{P} = \begin{bmatrix} \mathbf{M} | \mathbf{p}_4 \end{bmatrix} \quad (2.10)$$

where \mathbf{M} represents the 3×3 left matrix portion and \mathbf{p}_4 last column. Matrix \mathbf{M} is an important element that classifies two major camera categories, namely finite and infinite

cameras. If the matrix rank of M is 3 the camera is finite, if it is 2 the camera is infinite. A camera at infinity is also known as affine.

A general camera matrix P has following 4 properties:

- Camera center is the null vector of matrix P
- Matrix columns representing images of special points
- Matrix rows representing special planes
- Principal axis/ray and principal point can be defined from matrix M

These properties are explained in the following part of this section.

Camera Center

The matrix P has 4 columns, however its rank is 3. This shows existence of 1 dimensional right null space, which is the camera center.

$$PC = 0 \quad (2.11)$$

Representing the camera center as $C = [\tilde{C}^T, 1]^T$, we may formulate Eq. (2.5) and Eq. (2.10) as follows:

$$P = KR[I | -\tilde{C}] = M[I | M^{-1}p_4] \quad (2.12)$$

Equation (2.12) confirms that a camera has 11 DOF, 5 for K , 3 for R and 3 for C . If M is non-singular, camera center has following form:

$$C = \begin{bmatrix} -M^{-1}p_4 \\ 1 \end{bmatrix} \quad (2.13)$$

In case M is singular, camera center has following form:

$$C = \begin{bmatrix} d \\ 0 \end{bmatrix} \quad (2.14)$$

where d is the right null vector of M .

Camera Matrix Columns

The four columns of camera matrix \mathbf{P} represent image points as follows:

- \mathbf{p}_1 represents the vanishing point of world X -axis, since $\mathbf{p}_1 = \mathbf{PD}$, where $\mathbf{D} = [1, 0, 0, 0]^\top$ is the direction of X -axis.
- \mathbf{p}_2 represents the vanishing point of world Y -axis, since $\mathbf{p}_2 = \mathbf{PD}$, where $\mathbf{D} = [0, 1, 0, 0]^\top$ is the direction of Y -axis.
- \mathbf{p}_3 represents the vanishing point of world Z -axis, since $\mathbf{p}_3 = \mathbf{PD}$, where $\mathbf{D} = [0, 0, 1, 0]^\top$ is the direction of Z -axis.
- \mathbf{p}_4 represents the image of world origin, since $\mathbf{p}_4 = \mathbf{PD}$, where $\mathbf{D} = [0, 0, 0, 1]^\top$ is the world origin.

Camera Matrix Rows

The three rows of camera matrix \mathbf{P} represent special planes. These planes can be considered as join of a point and a line.

- first row, $\mathbf{P}^{1\top}$: represents a plane, which goes through the camera center \mathbf{C} and image y -axis. The plane $\mathbf{P}^{1\top}$ goes through the camera center \mathbf{C} since $\mathbf{P}^{1\top}\mathbf{C} = 0$. Also, all points on \mathbf{P}^1 , s.t. $\mathbf{P}^{1\top}\mathbf{X} = 0$ are projected to image points $[0, y, w]^\top$, which forms the image y -axis. Thus, $\mathbf{P}^{1\top}$ is a plane which goes through \mathbf{C} and y -axis.
- second row, $\mathbf{P}^{2\top}$: represents a plane, which goes through the camera center \mathbf{C} and image x -axis. The plane $\mathbf{P}^{2\top}$ goes through the camera center \mathbf{C} since $\mathbf{P}^{2\top}\mathbf{C} = 0$. Also, all points on \mathbf{P}^2 , s.t. $\mathbf{P}^{2\top}\mathbf{X} = 0$ are projected to image points $[x, 0, w]^\top$, which forms the image x -axis. Thus, $\mathbf{P}^{2\top}$ is a plane which goes through \mathbf{C} and x -axis.
- third row, $\mathbf{P}^{3\top}$: represents a plane, which goes through the camera center \mathbf{C} and is parallel to image xy -axis. The plane $\mathbf{P}^{3\top}$ goes through the camera center \mathbf{C} since $\mathbf{P}^{3\top}\mathbf{C} = 0$. Also, all points on \mathbf{P}^3 , s.t. $\mathbf{P}^{3\top}\mathbf{X} = 0$ are projected to image points $[x, y, 0]^\top$, which forms the plane parallel to image xy -plane. Thus, $\mathbf{P}^{3\top}$ is a plane which goes through \mathbf{C} and is parallel to xy -plane.

The first two planes are known as axes planes, while third plane is called principal plane.

Principal Axis and Principal Point

From Eq. (2.10) and Eq. (2.12) and $s = 0$ we may define the left 3×3 portion of \mathbf{P} as follows:

$$\mathbf{M} = \mathbf{KR} = \begin{bmatrix} \alpha_x & 0 & x_0 \\ 0 & \alpha_y & y_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{r}^1 \\ \mathbf{r}^2 \\ \mathbf{r}^3 \end{bmatrix} = \begin{bmatrix} \alpha_x \mathbf{r}^1 + x_0 \mathbf{r}^3 \\ \alpha_y \mathbf{r}^2 + y_0 \mathbf{r}^3 \\ \mathbf{r}^3 \end{bmatrix} \quad (2.15)$$

\mathbf{r}^i represent the row vectors of \mathbf{R} , and \mathbf{m}^i represent the row vectors of \mathbf{M} . The third row of \mathbf{M} , \mathbf{m}^3 represents the vector normal to principal plane, it is known as the principal axis or principal ray. It is also normal to image plane.

Next, we consider **principal point** $\mathbf{x}_0 = \mathbf{Mm}^3$. From Eq. (2.15) we may write:

$$\mathbf{Mm}^3 = \begin{bmatrix} (\alpha_x \mathbf{r}^1 + x_0 \mathbf{r}^3) \mathbf{r}^{3\top} \\ (\alpha_y \mathbf{r}^2 + y_0 \mathbf{r}^3) \mathbf{r}^{3\top} \\ (\mathbf{r}^3) \mathbf{r}^{3\top} \end{bmatrix} = \begin{bmatrix} x_0 \\ y_0 \\ 1 \end{bmatrix} \quad (2.16)$$

Since $(\mathbf{r}^3) \mathbf{r}^{3\top} = 1$ and $(\mathbf{r}^i) \mathbf{r}^{3\top} = 0$. The distance from camera center to image center (principal point) is the focal length of camera. Similarly, we may obtain α_x and α_y as the 1st and the 2nd elements of vectors \mathbf{Mm}^1 and \mathbf{Mm}^2 respectively.

2.3 Affine Camera

In this research we are concerned with affine cameras only. Generally, the word "affine" in classical geometry characterizes **preservation of collinearity**. Equation (2.17) shows the affine camera projecting a 3D point to a 2D point in homogeneous coordinates.

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (2.17)$$

The affine camera has 8 degrees of freedom. The camera matrix rank is 3. And rank of upper 3x2 matrix part is 2. The part $(p_{14}, p_{24})^\top$ is the image of the world origin. The reasons for being termed affine and mathematical details for matrix structure will be clarified in subsequent sections.

2.3.1 Affine Camera Approximation

Affine camera is a useful approximation of general projective camera, when the scene depth is small compared to the distance between the scene and the camera, and the distance of the point from principal ray is small. Cameras with long focal length and small field of view satisfy these conditions.

General cameras available are perspective. Therefore, for experiments affine camera approximation can be achieved by placing the camera far from the scene, then zooming it appropriately. This technique is known as **track back and zoom in**. Long distance/focal length significantly minimizes the perspective effects, while zooming has the effect of scale transformation, which is also an affine operation.

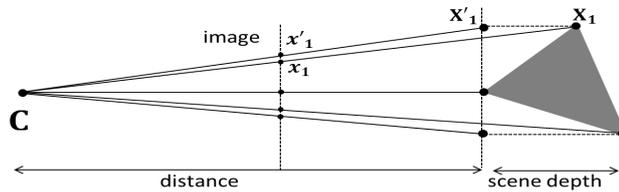
Figure 2.2 shows a perspective camera being approximated to affine. Compare point \mathbf{X}_1 in (a) and point \mathbf{X}_2 in (b) being projected to image points \mathbf{x}_1 and \mathbf{x}_2 respectively. The image points \mathbf{x}'_1 and \mathbf{x}'_2 are their affine projections. As shown in these figures, the approximation error $\Delta\mathbf{x}_2 = \|\mathbf{x}_2 - \mathbf{x}'_2\|$ is smaller than $\Delta\mathbf{x}_1 = \|\mathbf{x}_1 - \mathbf{x}'_1\|$. This means that we have better approximation under small scene depth.

It must be noted when the scene has many points with widely varying depths affine camera approximation is prone to errors. In such cases, the scene should be divided into multiple regions, with affine model for each region.

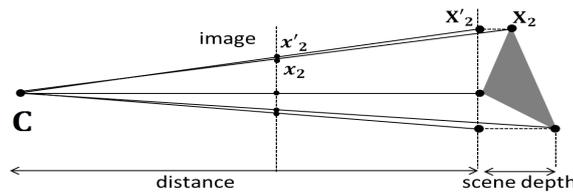
Formation of Affine Camera

A general projective camera can be approximated to affine. A finite projective camera can be written as follows:

$$\mathbf{P}_0 = \mathbf{KR}[\mathbf{I} | -\tilde{\mathbf{C}}] = \mathbf{K} \begin{bmatrix} \mathbf{r}^{1\top} & -\mathbf{r}^{1\top} \tilde{\mathbf{C}} \\ \mathbf{r}^{2\top} & -\mathbf{r}^{2\top} \tilde{\mathbf{C}} \\ \mathbf{r}^{3\top} & -\mathbf{r}^{3\top} \tilde{\mathbf{C}} \end{bmatrix} \quad (2.18)$$



(a) affine approximation



(b) affine approximation with smaller scene depth

Figure 2.2: Affine camera approximation.

The camera is at position $\tilde{\mathbf{C}}$ with rotation \mathbf{R} w.r.t world coordinate frame. For simplicity let us assume \mathbf{r}^3 is in direction of world Z -axis. From previous discussion, we know that the principal axis is in direction of \mathbf{r}^3 . This means the camera is allowed to rotate only about Z -axis, under such circumstances the rotation matrix is $\mathbf{R} = \mathbf{R}_z$ (right most matrix in Eq. (2.9)) and $\mathbf{r}^{3\top} = [0, 0, 1]^\top$. The distance of the world origin from the camera center in the direction of the principal ray is given by $d_0 = -\mathbf{r}^{3\top} \tilde{\mathbf{C}}$. The first step for affine camera is **tracking back** i.e. moving the camera backward along the direction of principal ray with translation $\mathbf{t} = [0, 0, t]^\top$. The new position of camera center is $\tilde{\mathbf{C}} - \mathbf{t}$, as shown in Fig. 2.3.

$$\mathbf{P}_t = \mathbf{K} \begin{bmatrix} \mathbf{r}^{1\top} & -\mathbf{r}^{1\top} (\tilde{\mathbf{C}} - \mathbf{t}) \\ \mathbf{r}^{2\top} & -\mathbf{r}^{2\top} (\tilde{\mathbf{C}} - \mathbf{t}) \\ \mathbf{r}^{3\top} & -\mathbf{r}^{3\top} (\tilde{\mathbf{C}} - \mathbf{t}) \end{bmatrix} = \mathbf{K} \begin{bmatrix} \mathbf{r}^{1\top} & -\mathbf{r}^{1\top} \tilde{\mathbf{C}} \\ \mathbf{r}^{2\top} & -\mathbf{r}^{2\top} \tilde{\mathbf{C}} \\ \mathbf{r}^{3\top} & -\mathbf{r}^{3\top} \tilde{\mathbf{C}} + t \end{bmatrix} \quad (2.19)$$

Since \mathbf{t} is in direction of \mathbf{r}^3 and rotation matrix is orthogonal, $-\mathbf{r}^{i\top} \mathbf{t}$ are zero for $i \neq 3$ and $d_t = -\mathbf{r}^{3\top} \tilde{\mathbf{C}} + t$ is the depth of the world origin w.r.t camera center. The next step is **zooming**. The effect of zooming is right multiplication of calibration matrix by $\text{diag}(k, k, 1)$. To keep the size fixed, the magnification factor can be defined as $k =$

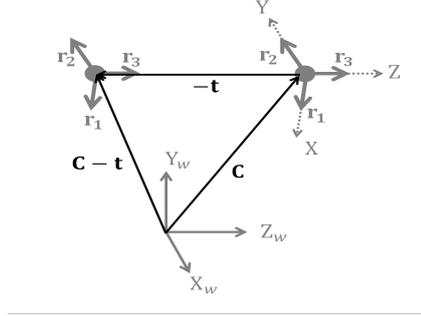


Figure 2.3: Backward translation of camera in direction of principal axis. The principal axis of camera is in direction of world Z -axis. The position vector \mathbf{C} shows original position of camera center and $\mathbf{C} - \mathbf{t}$ shows position of camera center after translation $-\mathbf{t}$.

d_t/d_0 . Equation (2.20) shows the effect of zooming on a tracked back camera.

$$\mathbf{P}_t = \mathbf{K} \begin{bmatrix} d_t/d_0 & 0 & 0 \\ 0 & d_t/d_0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{r}^{1\top} & -\mathbf{r}^{1\top} \tilde{\mathbf{C}} \\ \mathbf{r}^{2\top} & -\mathbf{r}^{2\top} \tilde{\mathbf{C}} \\ \mathbf{r}^{3\top} & d_t \end{bmatrix} = \frac{d_t}{d_0} \mathbf{K} \begin{bmatrix} \mathbf{r}^{1\top} & -\mathbf{r}^{1\top} \tilde{\mathbf{C}} \\ \mathbf{r}^{2\top} & -\mathbf{r}^{2\top} \tilde{\mathbf{C}} \\ \mathbf{r}^{3\top} d_0/d_t & d_0 \end{bmatrix} \quad (2.20)$$

The multiplicative factor d_t/d_0 can be ignored. Equation (2.20) shows the general camera matrix with combined effects of tracking and zooming. At $t = 0$, matrix \mathbf{P}_t is same as a finite projective camera \mathbf{P}_0 , shown in Eq. (2.18). If the camera is tracked backwards infinitely, implying $d_t \rightarrow \infty \Rightarrow d_0/d_t \rightarrow 0$ and the projective camera turns into affine as follows:

$$\mathbf{P}_\infty = \lim_{t \rightarrow \infty} \mathbf{P}_t = \mathbf{K} \begin{bmatrix} \mathbf{r}^{1\top} & -\mathbf{r}^{1\top} \tilde{\mathbf{C}} \\ \mathbf{r}^{2\top} & -\mathbf{r}^{2\top} \tilde{\mathbf{C}} \\ \mathbf{0}^\top & d_0 \end{bmatrix} \quad (2.21)$$

Affine Approximation Error

Next, we study the details of error introduced by affine approximation compared to projective camera. This will help us to understand the scene configurations required for best approximation. First, the points on plane through world origin and perpendicular to direction of principal axis are invariant to camera motion and zooming. With respect

to camera coordinates the points on this plane have following form:

$$\mathbf{X} = \begin{bmatrix} \alpha \mathbf{r}^1 + \beta \mathbf{r}^2 \\ 1 \end{bmatrix} \quad (2.22)$$

Since $\mathbf{P}_0 \mathbf{X} = \mathbf{P}_t \mathbf{X} = \mathbf{P}_\infty \mathbf{X} = [\alpha - \mathbf{r}^{1\top} \mathbf{C}, \beta - \mathbf{r}^{2\top} \mathbf{C}, 0 - \mathbf{r}^{3\top} \mathbf{C}]^\top \forall t$, and \mathbf{R} is orthogonal matrix i.e. $\mathbf{r}^{3\top}(\alpha \mathbf{r}^1 + \beta \mathbf{r}^2) = 0$. However, the images of points not on principal plane differ under projection of projective (\mathbf{P}_0) and affine camera (\mathbf{P}_∞), such points have following form.

$$\mathbf{X} = \begin{bmatrix} \alpha \mathbf{r}^1 + \beta \mathbf{r}^2 + \Delta \mathbf{r}^3 \\ 1 \end{bmatrix} \quad (2.23)$$

where Δ represents the perpendicular distance from principal plane. The projective and affine camera maps this point to following image points.

$$\mathbf{x}_{proj} = \mathbf{P}_0 \mathbf{X} = \mathbf{K} \begin{bmatrix} \tilde{x} \\ \tilde{y} \\ d_0 + \Delta \end{bmatrix} \quad (2.24)$$

$$\mathbf{x}_{aff} = \mathbf{P}_\infty \mathbf{X} = \mathbf{K} \begin{bmatrix} \tilde{x} \\ \tilde{y} \\ d_0 \end{bmatrix} \quad (2.25)$$

where $\tilde{x} = \alpha - \mathbf{r}^{1\top} \tilde{\mathbf{C}}$, $\tilde{y} = \beta - \mathbf{r}^{2\top} \tilde{\mathbf{C}}$. We can write the calibration matrix in following form:

$$\mathbf{K} = \begin{bmatrix} \mathbf{K}_{2 \times 2} & \tilde{\mathbf{x}}_0 \\ \tilde{\mathbf{0}}^\top & 1 \end{bmatrix} \quad (2.26)$$

$\mathbf{K}_{2 \times 2}$ is a 2×2 upper triangular matrix and $\tilde{\mathbf{x}}_0 = [x_0, y_0]^\top$. Equation (2.24) and (2.25) become,

$$\mathbf{x}_{proj} = \begin{bmatrix} \mathbf{K}_{2 \times 2} \tilde{\mathbf{x}} + (d_0 + \Delta) \tilde{\mathbf{x}}_0 \\ d_0 + \Delta \end{bmatrix} \quad (2.27)$$

$$\mathbf{x}_{aff} = \begin{bmatrix} \mathbf{K}_{2 \times 2} \tilde{\mathbf{x}} + d_0 \tilde{\mathbf{x}}_0 \\ d_0 \end{bmatrix} \quad (2.28)$$

Dehomogenizing the above images points,

$$\tilde{\mathbf{x}}_{proj} = \tilde{\mathbf{x}}_0 + \frac{\mathbf{K}_{2 \times 2} \tilde{\mathbf{x}}}{d_0 + \Delta} \quad (2.29)$$

$$\tilde{\mathbf{x}}_{aff} = \tilde{\mathbf{x}}_0 + \frac{\mathbf{K}_{2 \times 2} \tilde{\mathbf{x}}}{d_0} \quad (2.30)$$

From Eq. (2.29) and (2.30) we can find the relation between $\tilde{\mathbf{x}}_{proj}$ and $\tilde{\mathbf{x}}_{aff}$ as follows:

$$\tilde{\mathbf{x}}_{aff} - \tilde{\mathbf{x}}_0 = \frac{d_0 + \Delta}{d_0} (\tilde{\mathbf{x}}_{proj} - \tilde{\mathbf{x}}_0) \quad (2.31)$$

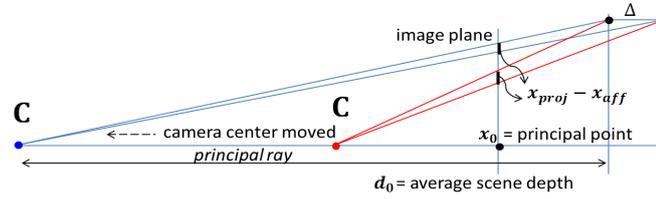
Equation (2.31) shows that the difference between affine approximated camera and projective camera is $\frac{d_0 + \Delta}{d_0} = 1 + \frac{\Delta}{d_0}$. We may rewrite Eq. (2.31) as follows:

$$\tilde{\mathbf{x}}_{aff} - \tilde{\mathbf{x}}_{proj} = \frac{\Delta}{d_0} (\tilde{\mathbf{x}}_{proj} - \tilde{\mathbf{x}}_0) \quad (2.32)$$

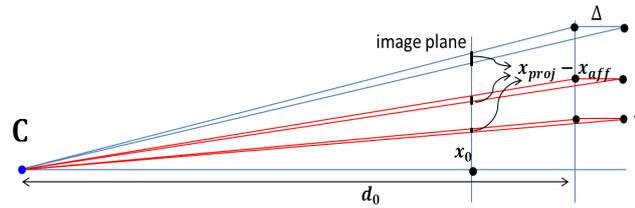
Using Eq. (2.32) we can consider the following three conditions under which a projective camera can be safely considered as affine. This is possible if the right hand side of Eq. (2.32) is close to zero.

- First, $\Delta \approx 0$ i.e. the scene depth is nearly equal to zero. Conversely, all points are clustered around average depth.
- Second, $\Delta \ll d_0$ or $d_0 \approx \infty$. This means that either the camera has long focal length or the distance between camera and scene is sufficiently large.
- Third, $\tilde{\mathbf{x}}_{proj} \approx \tilde{\mathbf{x}}_0$. This occurs under the condition when all 3D points are close to the principal axis.

For pictorial description of these conditions refer Fig. 2.4. The first condition is self-evident in Fig. 2.4. For the second condition the camera center is moved backward, as it can be seen that difference between $\tilde{\mathbf{x}}_{aff}$ and $\tilde{\mathbf{x}}_{proj}$ reduces. Fig. 2.4 (b) shows the effect of moving the points near to principal axis.



(a) effect of moving camera center farther



(b) effect of moving points closed to principal axis

Figure 2.4: Affine approximation conditions.

Anatomy of Affine Camera

Equation (2.21) shows the general form of affine camera, we write it as follows:

$$\mathbf{P}_\infty = \begin{bmatrix} \mathbf{K}_{2 \times 2} & \tilde{\mathbf{x}}_0 \\ \hat{\mathbf{0}}^\top & 1 \end{bmatrix} \begin{bmatrix} \hat{\mathbf{R}} & \hat{\mathbf{t}} \\ \mathbf{0}^\top & d_0 \end{bmatrix} \quad (2.33)$$

where,

- $\hat{\mathbf{0}} = [0, 0]^\top$ and $\mathbf{0} = [0, 0, 0]^\top$
- $\hat{\mathbf{R}}$ contains first two rows of rotation matrix \mathbf{R}
- $\hat{\mathbf{t}} = [-\mathbf{r}^{1\top} \tilde{\mathbf{C}}, -\mathbf{r}^{2\top} \tilde{\mathbf{C}}]^\top$

Following equation is equivalent to Eq. (2.33):

$$\mathbf{P}_\infty = \begin{bmatrix} \frac{1}{d_0} \mathbf{K}_{2 \times 2} & \tilde{\mathbf{x}}_0 \\ \hat{\mathbf{0}}^\top & 1 \end{bmatrix} \begin{bmatrix} \hat{\mathbf{R}} & \hat{\mathbf{t}} \\ \mathbf{0}^\top & 1 \end{bmatrix} \quad (2.34)$$

Assuming $d_0 = 1$ gives the definition of an affine camera.

$$\mathbf{P}_\infty = \begin{bmatrix} \mathbf{K}_{2 \times 2} \hat{\mathbf{R}} & \mathbf{K}_{2 \times 2} \tilde{\mathbf{x}}_0 + \hat{\mathbf{t}} \\ \hat{\mathbf{0}}^\top & 1 \end{bmatrix} \quad (2.35)$$

Thus following two decompositions are equivalent,

$$\mathbf{P}_\infty = \begin{bmatrix} \mathbf{K}_{2 \times 2} & \hat{\mathbf{0}} \\ \hat{\mathbf{0}}^\top & 1 \end{bmatrix} \begin{bmatrix} \hat{\mathbf{R}} & \hat{\mathbf{t}} \\ \mathbf{0}^\top & 1 \end{bmatrix} = \begin{bmatrix} \mathbf{K}_{2 \times 2} & \tilde{\mathbf{x}}_0 \\ \hat{\mathbf{0}}^\top & 1 \end{bmatrix} \begin{bmatrix} \hat{\mathbf{R}} & \hat{\mathbf{0}} \\ \mathbf{0}^\top & 1 \end{bmatrix} \quad (2.36)$$

Since $\hat{\mathbf{t}}$ is an external parameter, we cannot fix it in camera matrix. Other choice is $\tilde{\mathbf{x}}_0$ must be $[0, 0]^\top$. Therefore, affine camera can be decomposed as follows:

$$\mathbf{P}_{aff} = \mathbf{P}_\infty = \begin{bmatrix} \mathbf{K}_{2 \times 2} & \hat{\mathbf{0}} \\ \hat{\mathbf{0}}^\top & 1 \end{bmatrix} \begin{bmatrix} \hat{\mathbf{R}} & \hat{\mathbf{t}} \\ \mathbf{0}^\top & 1 \end{bmatrix} \quad (2.37)$$

Affine Camera Properties

Affine cameras are characterized by the following important properties:

- The Camera center is at infinity. Thus, principal plane for affine camera is at infinity.
- The last row of camera matrix is $[0, 0, 0, 1]^\top$.
- Calibration matrix \mathbf{K} in Eq. (2.6) takes the following form:

$$\mathbf{K} = \begin{bmatrix} \mathbf{K}_{2 \times 2} & \hat{\mathbf{0}} \\ \hat{\mathbf{0}}^\top & 1 \end{bmatrix} \quad (2.38)$$

- Weak perspective camera is one of the affine cameras, in which all points in world are first projected orthographically to average scene depth s.t. $Z = Z_0$ followed by perspective projection. Thus the non-linear component is tuned to a scalar multiplication, as follows:

$$x = \frac{1}{Z_0}X \quad y = \frac{1}{Z_0}Y \quad (2.39)$$

if $Z_0 = 1$, we have:

$$x = X \quad y = Y \quad (2.40)$$

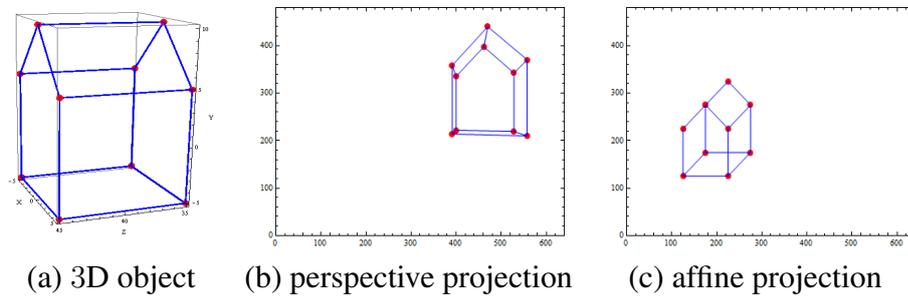


Figure 2.5: A 3D object with several parallel lines. Parallel lines do not remain parallel after projection in perspective projection whereas, affine preserves parallelism.

- The plane at infinity is projected to image points at infinity. As the point coordinates present on the plane at infinity have $(X, Y, Z, 0)^\top$ form, their projections have image coordinates $(x, y, 0)^\top$.
- Parallel lines remain parallel after projection. Since the points at infinity are invariant under affine projection and parallel lines intersect at infinity, parallelism is maintained. This is shown in Fig. 2.5. A 3D object with several parallel lines is projected to images using perspective camera (non-linear) in Fig. 2.5 (b) and an affine camera (linear) in Fig. 2.5 (c). Note, original parallel lines are no longer parallel in the perspective camera image, while they are parallel in the affine camera images. Other invariants under affine projection are volume ratio, centroid and plane at infinity.

2.4 Back Projection from 2D images

In 3D to 2D projection we consider back projection of a line to form a plane. Equation (2.3) shows the projection of a 3D point to a 2D point. Let us consider a line l passing through a point x in 2D space as follows:

$$l^\top \mathbf{x} = 0 \quad (2.41)$$

From Eq. (2.3) and Eq. (2.41) following expression is obtained:

$$l^\top \mathbf{P} \mathbf{X} = 0 \quad (2.42)$$

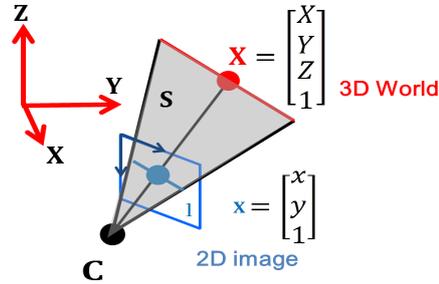


Figure 2.6: A general camera projection. A back projected plane S from a line l in image. The reader may observe that 3D points projected onto line l form a plane defined by camera matrix and line.

In the above equation, we define:

$$S = P^T l \quad (2.43)$$

where S is the back projected plane from a 2D point to a 3D point along the line of sight. This plane meets the 3D point X as follows:

$$S^T X = 0 \quad (2.44)$$

The plane S joins four entities, 3D point X , 2D point x and line l in image and camera center C . This phenomenon is shown in Fig. 2.6.

2.5 Multiview Relations

We next consider, the multiview relations, which describes the relationship among multiple cameras.

2.5.1 Multiple view geometry

Let us consider two cameras whose camera centers are C and C' as shown in Fig. 2.7. Let a 3D point X be projected to x in the first camera, and projected to x' in the second camera. Then, X , C and C' form a plane in the 3D space, which is called epipolar plane.

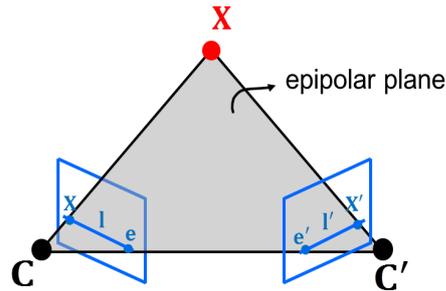


Figure 2.7: Epipolar geometry.

The intersections of the epipolar plane and two image planes form epipolar lines l and l' as shown in Fig. 2.7. The 3D line which goes through C and C' is called baseline, and the intersections of the baseline and two image planes are called epipoles e and e' . The epipoles are considered as the projection of the camera in the image of another camera. As shown in Fig. 2.7, the epipolar lines go through the epipoles e and e' and image points x and x' . Therefore, all the epipolar lines go through epipoles in each image.

2.5.2 Multiview Constraints

When we have multiple cameras, there exist special constraints among these cameras, which are called multiview constraints. In the 3D space, the multiview constraints are obtained by intersection of 4 planes at a 3D point $\mathbf{X} = [X, Y, Z, 1]^T$. Let us consider 2 cameras C and C' , and let a 3D point \mathbf{X} be projected to x and x' in the images of these cameras as shown in Fig. 2.8:

$$\begin{aligned} x &= \mathbf{P}\mathbf{X} \\ x' &= \mathbf{P}'\mathbf{X} \end{aligned} \tag{2.45}$$

Suppose we have 2 lines l and l' which go through x , and 2 lines l'_1 and l'_2 which go through x' respectively. If we consider back projected planes S^1 and S^2 from l_1 and l_2 , and S'^1 and S'^2 from l'_1 and l'_2 , these 4 planes intersect at \mathbf{X} in the 3D space as shown

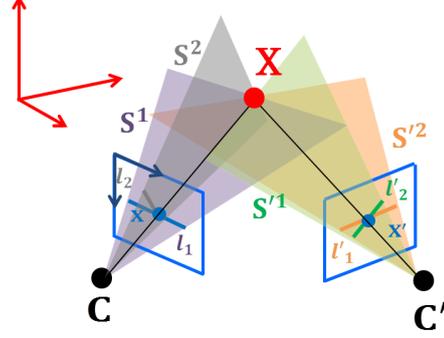


Figure 2.8: Intersection of four planes at a single 3D point \mathbf{X} . For two view geometry, planes \mathbf{S}^1 and \mathbf{S}^2 are back projected from image 1. Planes \mathbf{S}'^1 and \mathbf{S}'^2 are back projected from image 2.

in Fig. 2.8.

$$\mathbf{S}^1 = \mathbf{P}^\top \mathbf{l}_1, \quad \mathbf{S}^2 = \mathbf{P}^\top \mathbf{l}_2, \quad \mathbf{S}'^1 = \mathbf{P}'^\top \mathbf{l}'_1, \quad \mathbf{S}'^2 = \mathbf{P}'^\top \mathbf{l}'_2 \quad (2.46)$$

The condition of 4 planes $\mathbf{S}^1, \mathbf{S}^2, \mathbf{S}'^1, \mathbf{S}'^2$ intersecting at a point \mathbf{X} is stated mathematically as follows:

$$\begin{bmatrix} \mathbf{S}^{1\top} \\ \mathbf{S}^{2\top} \\ \mathbf{S}'^{1\top} \\ \mathbf{S}'^{2\top} \end{bmatrix} \mathbf{X} = \mathbf{0} \quad (2.47)$$

Since the linear equation Eq. (2.47) has a solution i.e. \mathbf{X} , the following equation must hold:

$$\det[\mathbf{S}^{1\top} \mathbf{S}^{2\top} \mathbf{S}'^{1\top} \mathbf{S}'^{2\top}] = 0 \quad (2.48)$$

In tensor notation Eq. (2.48) can be rewritten as follows:

$$\epsilon^{klmn} S_k^1 S_l^2 S_m'^1 S_n'^2 = 0 \quad (2.49)$$

where ϵ^{klmn} is known as the **Levi-Civita** tensor and defined as follows:

$$\epsilon^{klmn} = \begin{cases} +1 & \text{For even permutations of } (k, l, m, n) \\ -1 & \text{For odd permutations of } (k, l, m, n) \\ 0 & \text{otherwise} \end{cases} \quad (2.50)$$

where k, l, m and n have values from 1 to 4. Substituting Eq. (2.46) into Eq. (2.49) we have:

$$\epsilon^{klmn} P_k^p l_p^1 P_l^q l_q^2 P_m^r l_r^1 P_n^s l_s^2 = 0 \quad (2.51)$$

Since two lines l_1 and l_2 intersect at a point x , and two lines l'_1 and l'_2 intersect at a point x' , we may write:

$$\epsilon^{ipq} l_p^1 l_q^2 = x^i \quad (2.52)$$

$$\epsilon^{jrs} l_r^1 l_s^2 = x'^j \quad (2.53)$$

From Eq. (2.51), Eq. (2.52) and Eq. (2.53) we have,

$$\epsilon^{klmn} \epsilon_{ipq} \epsilon_{jrs} x^i x'^j P_k^p P_l^q P_m^r P_n^s = 0 \quad (2.54)$$

Then, we may define:

$$F_{ij} = \epsilon^{klmn} \epsilon_{ipq} \epsilon_{jrs} P_k^p P_l^q P_m^r P_n^s \quad (2.55)$$

where it may be noted that k, l, m and n range from 1 to 4 and represent columns of camera matrix. Variables p, q, r and s range from 1 to 3 and symbolize the rows of camera matrix. The entity F_{ij} is known as the **fundamental matrix** or **bifocal tensor**. It is a 3 x 3 matrix (**tensor of order 2**). Details of fundamental matrix and its computation are explained in Sec. 2.5.4. By using F_{ij} Eq. 2.54 can be described as follows:

$$x^i x'^j F_{ij} = 0 \quad (2.56)$$

This is the multiview constraints for two cameras known as bilinear constraints or epipolar constraints.

2.5.3 Algebraic verification

Another way to obtain the multiview relations is through matrix operations. Once again consider the affine projection equations of two cameras, combined into a single one as follows:

$$\begin{bmatrix} \mathbf{P} & \mathbf{x} & \mathbf{0} \\ \mathbf{P}' & \mathbf{0} & \mathbf{x}' \end{bmatrix} \begin{bmatrix} \mathbf{X} \\ -1 \\ -1 \end{bmatrix} = \mathbf{0} \quad (2.57)$$

Expanding above equation, we obtain:

$$\begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} & x^1 & 0 \\ p_{21} & p_{22} & p_{23} & p_{24} & x^2 & 0 \\ p_{31} & p_{32} & p_{33} & p_{34} & x^3 & 0 \\ \hline p'_{11} & p'_{12} & p'_{13} & p'_{14} & 0 & x'^1 \\ p'_{21} & p'_{22} & p'_{23} & p'_{24} & 0 & x'^2 \\ p'_{31} & p'_{32} & p'_{33} & p'_{34} & 0 & x'^3 \end{bmatrix} \begin{bmatrix} \mathbf{X} \\ -1 \\ -1 \end{bmatrix} = \mathbf{0} \quad (2.58)$$

Since Eq. (2.58) has non-trivial solution, the determinant of the left most 6×6 matrix must be zero. This condition is expressed as a bilinear equation in terms of x^i and x'^j . The easiest way is to eliminate one row of camera 1 and one row of camera 2, the columns to be eliminated are always the ones containing the image coordinates i.e. 5th column and 6th column. Remaining part is a 4×4 determinant. An example case is shown in Eq. (2.58) for computation of F_{33} with blue elements indicating the eliminated part. The bilinear relation obtained is:

$$x^i x'^j \epsilon_{ipq} \epsilon_{jrs} \det \begin{bmatrix} p^p \\ p^q \\ p^r \\ p^s \end{bmatrix} = 0 \quad (2.59)$$

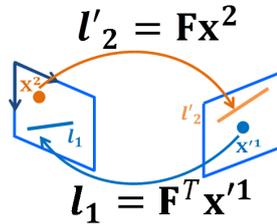


Figure 2.9: Fundamental matrix transferring a point into a line in the other image.

Equation (2.59) is the same two view constraint obtained in Eq. (2.56) with the fundamental matrix defined as in Eq. (2.55).

2.5.4 Fundamental Matrix

We next consider the fundamental matrix in detail. The **fundamental matrix** is the algebraic representation of two view geometry. It represents relative geometry (i.e. rotation and translation) between two cameras. The rank of \mathbf{F} is 2 and has 9 elements. Its degrees of freedom depend on the type of camera used. For projective cameras it has 7 degrees of freedom. If cameras are affine the last row of camera matrices is $[0, 0, 0, 1]^T$, therefore only 5 elements of \mathbf{F} are non-zero, leading to 4 degrees of freedom due to overall scale ambiguity. The non-zero \mathbf{F} elements are F_{13} , F_{23} , F_{31} , F_{32} and F_{33} . The non-zero elements of \mathbf{F} can be found by analyzing Eq. (2.55), the value of \mathbf{F} is determined by the selections of camera rows. All rows participate in computation of \mathbf{F} except those indicated by i and j . Thus for affine cameras, all elements of \mathbf{F} are zero except five elements stated afore.

Fundamental matrix is responsible for the transfer of image information between images. For a point in one image, it gives a line in the other on which the corresponding point may be found. Therefore it reduces the search space from 2 dimensional image to a line. This is shown in Fig. 2.9.

If the internal camera matrices are calibrated, the fundamental matrix can be modified into **essential matrix** whose DOF is just 5. Appendix A.5 shows the relationship between the two matrices.

Computation of Fundamental Matrix

Equation (2.56) can be written in matrix form as follows;

$$\mathbf{x}^\top \mathbf{F} \mathbf{x}' = 0 \quad (2.60)$$

where $\mathbf{x} = [x, y, 1]^\top$ and $\mathbf{x}' = [x', y', 1]^\top$. Then Eq. (2.60) can be expressed as follows:

$$x'x f_{11} + x'y f_{12} + x' f_{13} + y'x f_{21} + y'y f_{22} + y' f_{23} + x f_{31} + y f_{32} + f_{33} = 0 \quad (2.61)$$

In the case of affine cameras only 5 elements of F_{ij} are non-zero. Therefore Eq. (2.61) is reduced to the following form:

$$x' f_{13} + y' f_{23} + x f_{31} + y f_{32} + f_{33} = 0 \quad (2.62)$$

Let us define $\mathbf{f} = [f_{13}, f_{23}, f_{31}, f_{32}, f_{33}]^\top$. Then in matrix form we may write Eq. (2.62) as follows;

$$[x', y', x, y, 1] \mathbf{f} = 0 \quad (2.63)$$

For n corresponding points the matrix of image coordinates, say \mathbf{M} is of dimension $n \times 5$. Eigenvalue decomposition of $\mathbf{M}^\top \mathbf{M}$ i.e. eigenvector corresponding to the smallest eigenvalue will give the coefficients of \mathbf{f} .

2.5.5 Minimum No. of Points

Equation (2.62) shows each image correspondence gives one independent bilinear constraint (equation). Therefore a minimum of 4 point correspondences in two views is sufficient to compute the affine fundamental matrix up to an overall scale ambiguity. However, these 4 points must have good 3D variation i.e. 4 points must not be coplanar and no 3 points should be collinear.

2.5.6 Maximum No. of Cameras

In classical multiple view geometry we can have a maximum of 4 cameras, owing to the fact that the no. of planes used for the generation of multilinear constraints is 4. The geometric entities that describe the multiview relations for 3 and 4 cameras are **trifocal tensor** and **quadrifocal tensor**. Considering Eq. (2.47) for trifocal tensor the first two planes are back projected from image 1 and second and third planes are back projected from image 2 and image 3 respectively. For quadrifocal tensor each image provides one plane. The multiview relations for point correspondences across three and four images are given by following equations [1, 2, 19, 21, 30].

$$x^i x'^j x''^k \epsilon_{jru} \epsilon_{ksv} \tau_i^{rs} = 0_{uv} \quad (2.64)$$

$$x^i x'^j x''^k x'''^l \epsilon_{ipu} \epsilon_{jqv} \epsilon_{krw} \epsilon_{lsx} Q^{pqrs} = 0_{uvwx} \quad (2.65)$$

where tensors, τ_i^{rs} and Q^{pqrs} are defined by using the projection matrices of four cameras \mathbf{P} , \mathbf{P}' , \mathbf{P}'' and \mathbf{P}''' as follows:

$$\tau_i^{rs} = \epsilon_{ipq} \epsilon^{abcd} P_a^p P_b^q P_c^r P_d^s \quad (2.66)$$

$$Q^{pqrs} = \epsilon^{abcd} P_a^p P_b^q P_c^r P_d^s \quad (2.67)$$

2.6 3D Reconstruction

Once a multiview entity is computed, such as fundamental matrix for two views. 3D reconstruction can be achieved as shown in the following sections:

2.6.1 Retrieval of Camera Matrices

It is known that the canonical camera pair can be defined in the affine case as follows:

$$\mathbf{P} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.68)$$

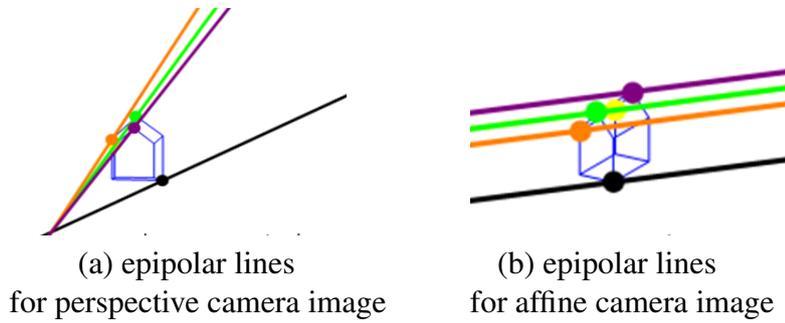


Figure 2.10: Comparison of epipolar lines. The epipolar lines in perspective camera intersect at a finite epipole. The epipolar lines in affine camera intersect at infinity (epipole), thus parallel.

$$\mathbf{P}' = [\mathbf{H} | \mathbf{e}'] \quad (2.69)$$

\mathbf{H} is called the homography, which represents 2D projective transformation and \mathbf{e}' represents the epipole in the second image. Thus, if we compute \mathbf{H} and \mathbf{e}' , a pair of cameras can be calibrated.

2.6.2 Computation of Epipoles

As shown in Sec. 2.5.1, all the epipolar lines go through epipoles. This means that the epipole in one image corresponds to all the points in other another image. Hence the following equation must hold:

$$e^i F_{ij} = 0_j \quad (2.70)$$

$$e'^j F_{ij} = 0_i \quad (2.71)$$

Thus, the epipoles can be computed as the left and right null vector of fundamental matrix. The camera centers for affine camera are at infinity. Points at infinity are mapped to points at infinity in affine camera images. Therefore, the epipoles for affine cameras are at infinity, and epipolar lines are parallel. This is another characteristic of affine cameras. This is shown in Fig. 2.10.

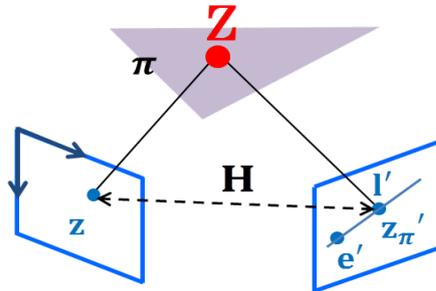


Figure 2.11: Image points \mathbf{x} and \mathbf{x}' of a 3D point \mathbf{X} on an arbitrary plane π are related through a homography \mathbf{H} . The homography \mathbf{H} can be computed from fundamental matrix \mathbf{F} .

2.6.3 Extraction of Homography

Projections of coplanar 3D points in one image are related to projections in another image via a homography. Consider a point \mathbf{X} in 3D world on an arbitrary plane π . The point \mathbf{X} is projected to a point \mathbf{x} in image 1 and \mathbf{x}'_{π} in image 2. The corresponding points \mathbf{x} and \mathbf{x}'_{π} are related by a 2D homography \mathbf{H} , as shown in Fig. 2.11.

$$x_{\pi}^l = H_i^l x^i \quad (2.72)$$

Next, an epipolar line l' in image 2 can be defined by joining epipole e' and point \mathbf{x}'_{π} .

$$l'_j = \epsilon_{jlr} x_{\pi}^l e'^r \quad (2.73)$$

From bilinear constraint Eq. (2.56) the line l'_j can also be defined as follows:

$$l'_j = F_{ij} x^i \quad (2.74)$$

Equations (2.72), (2.73) and (2.74) lead to following result.

$$H_i^l = e'^{lr} e'_r F_{ij} \quad (2.75)$$

Therefore, homography \mathbf{H} can be computed from a bifocal tensor, which is fundamental matrix \mathbf{F} in this case. Once the homography \mathbf{H} and the epipole e' are computed, camera

matrices of two cameras can be recovered as shown in Sec. 2.6.1.

2.7 Affine Reconstruction and Rectification

Once the camera matrices \mathbf{P} and \mathbf{P}' are retrieved, 3D reconstruction is possible. Since the 3D point \mathbf{X} is projected to \mathbf{x} and \mathbf{x}' as shown in Eq. (2.45), we have the following equations:

$$[\mathbf{x}]_{\times} \mathbf{P} \mathbf{X} = \mathbf{0} \quad (2.76)$$

$$[\mathbf{x}']_{\times} \mathbf{P}' \mathbf{X} = \mathbf{0} \quad (2.77)$$

where, $[\cdot]_{\times}$ denotes a skew symmetric matrix for vector product. From Eq. (2.76) and Eq. (2.77) we have:

$$\mathbf{M} \mathbf{X} = \mathbf{0} \quad (2.78)$$

where \mathbf{M} is a 6×4 matrix as follows:

$$\mathbf{M} = \begin{bmatrix} [\mathbf{x}]_{\times} \mathbf{P} \\ [\mathbf{x}']_{\times} \mathbf{P}' \end{bmatrix} \quad (2.79)$$

By computing \mathbf{X} from Eq. (2.78), we can reconstruct the 3D point \mathbf{X} . The least squares solution of Eq. (2.78) is obtained by computing an eigenvector which corresponds to the minimum eigenvalue of $\mathbf{M}^{\top} \mathbf{M}$.

The reconstructed object is correct up to an affine transformation. It is still different from ground truth, hence a geometric correction is required. To obtain the original object a transformation \mathbf{H}_a is computed to rectify the reconstruction $\mathbf{X} = \mathbf{H}_a \mathbf{X}'$, where \mathbf{H}_a is the affine transformation between reconstruction \mathbf{X}' and ground truth \mathbf{X} . It is described using 4 corresponding points in homogeneous coordinates $\mathbf{X}_i, \mathbf{X}'_i (i = 1, \dots, 4)$ which act as basis vectors.

$$\mathbf{H}_a = \begin{bmatrix} \mathbf{X}_1 & \mathbf{X}_2 & \mathbf{X}_3 & \mathbf{X}_4 \end{bmatrix} \begin{bmatrix} \mathbf{X}'_1 & \mathbf{X}'_2 & \mathbf{X}'_3 & \mathbf{X}'_4 \end{bmatrix}^{-1} \quad (2.80)$$

Figure 2.12 shows 4 points forming basis vectors .

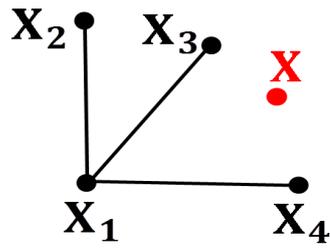


Figure 2.12: Four points forming affine basis vectors for the rectification of reconstruction.

Chapter 3

Reconstructing Grayscale Objects without knowing Image Correspondences

As shown in the previous chapter, the MVG enables us to describe the relationship between multiple cameras, and reconstruct 3D objects in the scene. However, the classical MVG explained in the previous chapter can only describe the geometric relationship among multiple cameras, and photometric properties, such as image intensity, cannot be treated.

In this chapter, we extend the classical MVG into high dimensional space, and show new MVG which enables us to describe photometric relationship as well as geometric relationship among multiple images. Based on the new MVG, we propose a method for reconstructing sequential patterns without knowing image correspondences.

A general imaging process can be described as follows. Suppose we have an object, a source of light and a camera. The light source emits the light and the object reflects it (obeying some reflection models), the camera captures the reflections (obeying some transformation model) and maps them onto an image plane.

When a ray of light strikes a surface, it is reflected and scattered on the surface. This scattering is different in different directions and depends on the surface properties. Usually under normal circumstances, reflection pattern is a combination of specular (or specular lobe) and diffuse reflection. To make things simpler **Johan Lambert** proposed

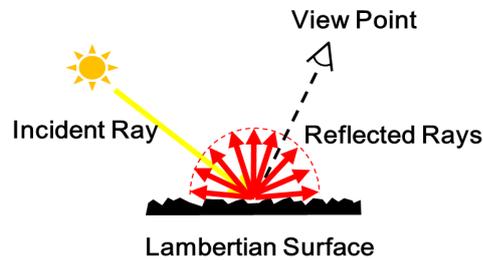


Figure 3.1: The uniform reflection behavior of a Lambertian surface.

a special case of light reflection, where the reflection is uniform in all directions, like a hemisphere. This is a good approximation of light reflection in many cases, such as matte paper or unfinished wood etc. In this case, the specular part is zero. A reflection behavior of such kind is termed as **diffuse reflection** or **Lambertian**. Figure 3.1 shows the case of Lambertian reflectance. In Lambertian reflectance the apparent surface brightness is constant, even if the viewing direction is different. The benefit of Lambertian surface assumption is that we can choose any view point without any difference in surface radiosity.

3.1 Object and Image Spaces

For considering the relationship between the real world and its projections in images, we in this section explain points in the real world and in images.

3.1.1 4D World

For deriving a reconstruction method without image correspondences, we consider that the real world is in 4 dimensions. Three dimensions are for geometric information and one dimension is for intensity / gray level. It is assumed that Lambertian reflectance model is obeyed for the intensity information. A 4D point in homogeneous form is

represented as a vector of following form:

$$\mathbf{X} = \begin{bmatrix} X \\ Y \\ Z \\ I \\ 1 \end{bmatrix} \quad (3.1)$$

where X , Y and Z denote the coordinates of a point in the usual 3D space, and I denotes the intensity of the 3D point.

3.1.2 3D Image

An image is considered to be in 3 dimensions. Two dimensions are for geometric information and 1 dimension is reserved for intensity. A 3D image point in homogeneous coordinates is defined in vector form as follows:

$$\mathbf{x} = \begin{bmatrix} x \\ y \\ i \\ 1 \end{bmatrix} \quad (3.2)$$

where x and y denote the coordinates of an image point, and i denotes the intensity of the image point. In tensor notation the points are represented with contravariant indices. For example, a 4D point and a 3D point have representations X^i and x^i respectively. Further details on tensor notation can be found in Appendix A.4. For homogeneous coordinates kindly refer Appendix A.1.

3.1.3 Object and Image Spaces considering Sequential Patterns

In sequential patterns the 4D points in world and 3D points in image are considered to be connected in a sequence, as shown in Fig. 1.3. In spatial domain sequential and texture patterns differ depending on how the elements are connected. An example of sequential pattern with different sequence ordering is shown in Fig. 3.2. As shown in this figure, the set of coordinates of sequential pattern is completely different if the sampling order

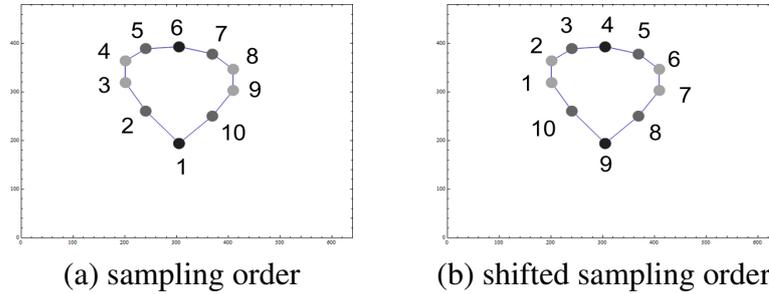


Figure 3.2: Example of a sequence in an image with sampling order shifted.

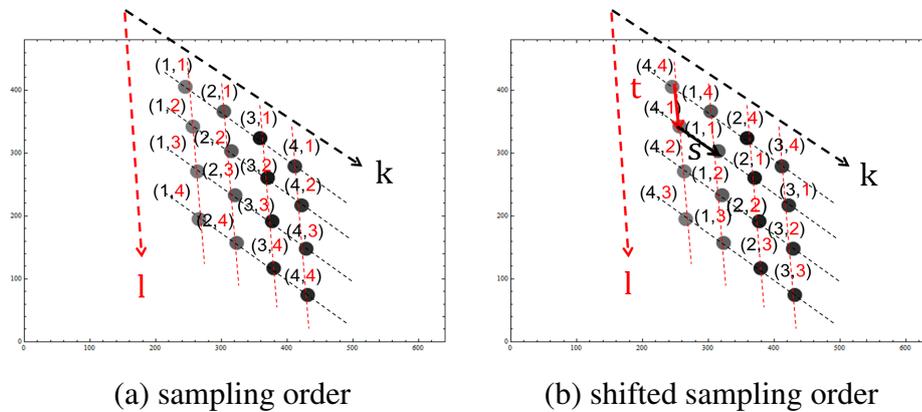


Figure 3.3: An example of texture sampling. (b) shows the sampling shift in horizontal and vertical direction using the numbering convention of (a). k and l denote a 2D sampling grid, and t and s represent the shift in horizontal and vertical directions of the sampling grid.

is shifted. Thus, finding the correct ordering is essential in the classical MVG and for 3D reconstruction.

3.1.4 Object and Image Spaces considering Texture Patterns

In texture pattern, the sample points vary in two directions. The sample points are connected in horizontal and vertical directions to form a 2D sampling grid on the texture elements. Fig. 3.3 shows the texture elements of an image connected to form a grid with different sampling order. Again, the set of coordinates of texture pattern is completely different if the sampling order is shifted as shown in Fig. 3.3.

3.2 Affine Camera from 4D to 3D Space

The 4D world point defined in Sec. 3.1.1 is projected to an image point defined in Sec. 3.1.2 by an affine camera as follows:

$$\begin{bmatrix} x \\ y \\ i \\ 1 \end{bmatrix} = \begin{bmatrix} p_{11} & p_{12} & p_{13} & 0 & p_{15} \\ p_{21} & p_{22} & p_{23} & 0 & p_{25} \\ 0 & 0 & 0 & p_{34} & p_{35} \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ I \\ 1 \end{bmatrix} \quad (3.3)$$

This is the extension of the classical affine camera shown in Sec. 2.3 , and we call it **extended affine camera** model in this thesis. The red part in the matrix denotes rotation and the blue part represents translation. Intensity is projected through p_{34} and p_{35} . We assume image intensity depends on two camera factors i.e. intensity gain (p_{34}) and intensity offset (p_{35}). They are known as camera linear sensitivity parameters. In general, these parameters have different values in different cameras. Details of these parameters can be found in Appendix B. Equation (3.3) shows, the extended camera model can not only capture geometric information, but also capture photometric information of the scene. It should be noted that geometric and photometric information are independent of each other. Therefore, the projection is simultaneous yet independent. A conceptual figure of 4D to 3D camera projection is shown in Fig. 3.4.

Center of projection (COP) in 4D to 3D projection is a tensor of order $4 - 3 = 1$, i.e. it is 4D point. COP is usually the null space of camera matrix (4×5 in this case). The line of sight (LOS) is the join/union of 2 higher dimensional points (tensor of order 1). Therefore, it is a line (tensor of order $1 + 1 = 2$). In tensor notation image points and lines are represented as x^i and l_{ij} respectively.

3.3 Projection in Frequency Domain

For deriving the reconstruction method without image correspondence, we introduce image projection in frequency domain in this thesis. We first, consider the general concept of affine projection in frequency domain. Equation (3.3) shows the projection

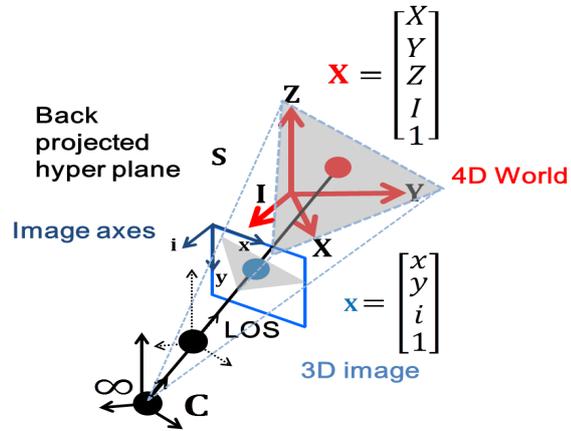


Figure 3.4: An extended camera projection from 4D world to 3D image. The line joining the camera center C and image point x is termed as line of sight (LOS). The figure shows the camera center is at infinity for affine cameras. A plane in the 3D image is back projected to a hyperplane in the 4D world space.

of point \mathbf{X} to image point \mathbf{x} in spatial domain. Let us denote, the discrete Fourier transform by \mathcal{F} . To change the domain from spatial to frequency we apply \mathcal{F} to both sides of Eq. (3.3) as follows:

$$\mathcal{F}(\mathbf{x}) = \mathcal{F}(\mathbf{P}\mathbf{X}) \quad (3.4)$$

Affine projection and Fourier transform are both linear operations, the product is commutative and order of these operations can be swapped as follows:

$$\mathcal{F}(\mathbf{x}) = \mathbf{P}\mathcal{F}(\mathbf{X}) \quad (3.5)$$

Let \mathbf{z} be the Fourier transform of \mathbf{x} , and \mathbf{Z} be the Fourier transform of \mathbf{X} . Then, Eq. (3.5) can be rewritten as follows:

$$\mathbf{z} = \mathbf{P}\mathbf{Z} \quad (3.6)$$

Thus, the Fourier transformation of a series of points being affine projected has same effect of projecting frequencies (points in frequency domain) under affine transformation.

3.3.1 Projection in Frequency Domain for Sequential Patterns (1D-DFT)

To consider the transformation of the sequential pattern in spatial domain to frequency domain, suppose two cameras \mathbf{C} and \mathbf{C}' projecting a series of 3D points $\mathbf{X}(k)$ ($k = 1, \dots, N$). Assuming the point correspondences in two cameras are unknown but the order of the points is maintained in each view, the k th 3D point $\mathbf{X}(k)$ is observed as k th image point $\mathbf{x}(k)$ in image 1. The same point is observed as k' th image point $\mathbf{x}'(k')$ in image 2 as follows:

$$\mathbf{x}(k) = \mathbf{P}\mathbf{X}(k) \quad (3.7)$$

$$\mathbf{x}'(k') = \mathbf{P}'\mathbf{X}(k) \quad (3.8)$$

where k and k' are different in general. As the order is preserved in each image, the following relationship holds:

$$k' = k + s \quad (3.9)$$

where, s is the shift in sampling of image 2 with respect to sampling of image 1. In this research, we assume that it is unknown but it is constant. Therefore, $\mathbf{x}(k)$ and $\mathbf{x}'(k')$ are not corresponding points. To solve with the correspondence problem, the projection model in the frequency space described in Eq. (3.6) is used. Suppose there are N image points in image 1 and image 2 respectively. Then, by applying discrete Fourier transform to both sides of Eq. (3.7):

$$\begin{aligned} \mathbf{z}(n) &= \frac{1}{N} \sum_{k=0}^{N-1} \mathbf{P}\mathbf{X}(k) e^{-\frac{j2\pi nk}{N}} \\ &= \mathbf{P}\mathbf{Z}(n) \end{aligned} \quad (3.10)$$

where, $\mathbf{z}(n) = [x_f(n), y_f(n), i_f(n), \delta(n)]^\top$ is an image point, and $\mathbf{Z}(n) = [X_f(n), Y_f(n), Z_f(n), I_f(n), \delta(n)]^\top$ is a 4D object point in frequency space. The reader may notice we are using homogeneous coordinates. $\delta(n)$ is the delta function, whose value is equal to 1 for $n = 0$ and 0 for others. Similarly, applying discrete Fourier transform

to Eq. (3.8):

$$\begin{aligned} \mathbf{z}'(n) &= \frac{1}{N} \sum_{k'=0}^{N-1} \mathbf{P}' \mathbf{X}(k' - s) e^{-\frac{j2\pi nk'}{N}} \\ &= \mathbf{P}' \mathbf{Z}(n) e^{-\frac{j2\pi ns}{N}} \end{aligned} \quad (3.11)$$

where $\mathbf{z}'(n) = [x'_f(n), y'_f(n), i'_f(n), \delta(n)]^\top$ is a point in second image in frequency space. We rewrite Eq. (3.11) as follows:

$$\lambda(n) \mathbf{z}'(n) = \mathbf{P}' \mathbf{Z}(n) \quad (3.12)$$

where $\lambda(n) = e^{\frac{j2\pi ns}{N}}$ represents the phase shift of sampling in camera \mathbf{C}' relative to camera \mathbf{C} , and it is unknown. From Eq. (3.10) and Eq. (3.11), we find that in the frequency space both cameras project the same 4D point $\mathbf{Z}(n)$ and thus it is possible to consider the correspondence of image data, even if the sampling order of two images is different.

3.3.2 Projection in Frequency Domain for Texture Patterns (2D-DFT)

For the case of texture surface consider a 2D sampling grid on it, as shown in Fig. 3.3, where the 2D grid is indexed by k and l . Then, each point on the texture pattern is defined using two variables k and l of the sampling grid. Suppose two cameras \mathbf{C} and \mathbf{C}' project a texture pattern of 4D points $\mathbf{X}(k, l)$ ($k = 1, \dots, N, l = 1, \dots, M$) to image 1 and image 2 respectively. It is assumed that the point correspondences in these two cameras are unknown, but a grid of points and their order is obtained in each view. Thus, the (k, l) th 3D point $\mathbf{X}(k, l)$ is observed as (k, l) th image point $\mathbf{x}(k, l)$ in image 1 and is observed as (k', l') th image point $\mathbf{x}'(k', l')$ in image 2 as follows:

$$\mathbf{x}(k, l) = \mathbf{P} \mathbf{X}(k, l) \quad (3.13)$$

$$\mathbf{x}'(k', l') = \mathbf{P}' \mathbf{X}(k, l) \quad (3.14)$$

k, l and k', l' are different in general. Since order is preserved in each image, the following relationship holds:

$$k' = k + s \quad (3.15)$$

$$l' = l + t \quad (3.16)$$

where, s and t are the horizontal and vertical shifts in the sampling of image 2 with respect to the sampling of image 1. In this thesis, we assume that s and t are unknown but are constant. Thus, $\mathbf{x}(k, l)$ and $\mathbf{x}'(k', l')$ are non-corresponding points. We solve this problem by considering camera projection in the frequency space. Suppose there are $N \times M$ image points in image 1 and image 2 respectively. Applying Fourier transform to Eq. (3.13):

$$\begin{aligned} \mathbf{z}(n, m) &= \frac{1}{N} \frac{1}{M} \sum_{k=0}^{N-1} \sum_{l=0}^{M-1} \mathbf{P}\mathbf{X}(k, l) e^{-\frac{j2\pi nk}{N}} e^{-\frac{j2\pi ml}{M}} \\ &= \mathbf{P}\mathbf{Z}(n, m) \end{aligned} \quad (3.17)$$

where, $\mathbf{z}(n, m) = [x_f(n, m), y_f(n, m), i_f(n, m), \delta(n, m)]^\top$ represents the 3D image point in frequency space, and $\mathbf{Z}(n, m) = [X_f(n, m), Y_f(n, m), Z_f(n, m), I_f(n, m), \delta(n, m)]^\top$ represents the 4D point in the frequency space. $\delta(n, m)$ is the delta function, whose value is equal to 1 for $n = m = 0$ and 0 for others. Similarly, applying Fourier transform to Eq. (3.14) we have:

$$\begin{aligned} \mathbf{z}'(n, m) &= \frac{1}{N} \frac{1}{M} \sum_{k'=0}^{N-1} \sum_{l'=0}^{M-1} \mathbf{P}'\mathbf{X}(k' - s, l' - t) e^{-\frac{j2\pi nk'}{N}} e^{-\frac{j2\pi ml'}{M}} \\ &= \mathbf{P}'\mathbf{Z}(n, m) e^{-\frac{j2\pi ns}{N}} e^{-\frac{j2\pi mt}{M}} \end{aligned} \quad (3.18)$$

where $\mathbf{z}'(n, m) = [x'_f(n, m), y'_f(n, m), i'_f(n, m), \delta(n, m)]^\top$ is a point in the second image in frequency space. Equation (3.18) can be rewritten as follows:

$$\lambda(n, m) \mathbf{z}'(n, m) = \mathbf{P}'\mathbf{Z}(n, m) \quad (3.19)$$

where $\lambda(n, m) = e^{-\frac{j2\pi ns}{N}} e^{-\frac{j2\pi mt}{M}}$ represents the 2D phase shift of sampling in camera C' relative to camera C . Equation (3.17) and (3.19) show that in the frequency space both

cameras project the same 4D point $\mathbf{Z}(n, m)$, even if the original 4D points \mathbf{X} projected to these two cameras are different from each other. Thus, it is possible to consider the correspondence of image data, even if the sampling order of image 2 is shifted in 2 directions of the 2D sampling grid w.r.t the sampling order of image 1.

3.4 Back Projection from 3D images

We next consider back projection from 3D images to 4D space in frequency domain, which is the extension of the concept shown in Sec. 2.4. The projection in frequency domain is described in Eq. (3.6) as follows:

$$\mathbf{z} = \mathbf{P}\mathbf{Z} \quad (3.20)$$

The following equations shows a plane containing a point \mathbf{z} in 3D image.

$$\mathbf{s}^\top \mathbf{z} = 0 \quad (3.21)$$

From above two equations, following result is straight forward:

$$\mathbf{s}^\top \mathbf{P}\mathbf{Z} = 0 \quad (3.22)$$

In Eq. (3.22), a hyperplane \mathbf{S} can be defined such that $\mathbf{S} = \mathbf{P}^\top \mathbf{s}$. Then, we have:

$$\mathbf{S}^\top \mathbf{Z} = 0 \quad (3.23)$$

Equation (3.23) asserts important results. That is, a hyperplane \mathbf{S} back projected from the plane \mathbf{s} in 3D image goes through a 4D point \mathbf{Z} in the frequency space. Equation (3.23) is agreeable in spatial domain as well. Figure 3.4 portrays the back projection concept.

3.5 Multiview relations for Sequential Patterns

Following sections describe the method of deriving multiview relations in 4D extended space. For ease of understanding, a quick look of Appendix A.4 would be helpful for

tensor notation.

3.5.1 Geometric derivation

Multiview relations in 4D space can be considered as the intersection of 5 hyperplanes meeting at a single point [1]. Let us consider two cameras in the 4D frequency space, and let a 4D point \mathbf{Z} be projected to \mathbf{z} and \mathbf{z}' in the image of these cameras, as shown in Fig. 3.5.

$$\mathbf{z} = \mathbf{P}\mathbf{Z} \quad (3.24)$$

$$\lambda\mathbf{z}' = \mathbf{P}'\mathbf{Z} \quad (3.25)$$

where λ represents the phase shift of the sampling between two cameras. Suppose we have three planes $\mathbf{s}_1, \mathbf{s}_2$ and \mathbf{s}_3 which go through \mathbf{z} , and two planes \mathbf{s}'_1 and \mathbf{s}'_2 which go through \mathbf{z} . If we consider back projected hyperplanes $\mathbf{S}^1, \mathbf{S}^2$ and \mathbf{S}^3 from $\mathbf{s}^1, \mathbf{s}^2$ and \mathbf{s}^3 , and \mathbf{S}'^1 and \mathbf{S}'^2 from \mathbf{s}'^1 and \mathbf{s}'^2 , these hyperplanes intersect at \mathbf{Z} in the 4D frequency space as shown in Fig. 3.5.

$$\mathbf{S}^1 = \mathbf{P}^\top \mathbf{s}^1 \quad (3.26)$$

$$\mathbf{S}^2 = \mathbf{P}^\top \mathbf{s}^2 \quad (3.27)$$

$$\mathbf{S}^3 = \mathbf{P}^\top \mathbf{s}^3 \quad (3.28)$$

$$\mathbf{S}'^1 = \mathbf{P}'^\top \mathbf{s}'^1 \quad (3.29)$$

$$\mathbf{S}'^2 = \mathbf{P}'^\top \mathbf{s}'^2 \quad (3.30)$$

A necessary and sufficient condition for these five hyperplanes to meet in a common point in 4D space is that the determinant of the matrix formed from the vectors representing these hyperplanes should vanish. Mathematically, this condition is written as follows:

$$\det[\mathbf{P}^\top \mathbf{s}^1, \mathbf{P}^\top \mathbf{s}^2, \mathbf{P}^\top \mathbf{s}^3, \mathbf{P}'^\top \mathbf{s}'^1, \mathbf{P}'^\top \mathbf{s}'^2] = 0 \quad (3.31)$$

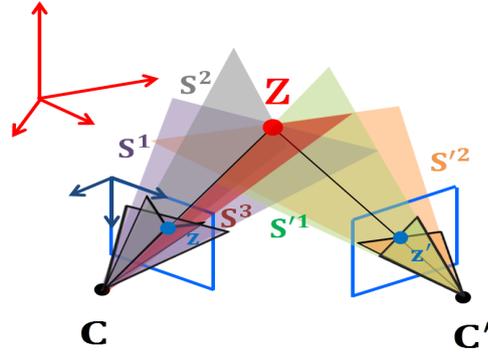


Figure 3.5: Intersection of five planes at a single 4D point Z . For two view geometry, hyperplanes S^1 , S^2 and S^3 are back projected from image 1 and S'^1 and S'^2 are back projected from image 2.

Note, these planes are considered in analysis for deriving the multilinear relationship between z and z' , and determining these planes is not required. Equation (3.31) can be rewritten in tensor format as follows:

$$\epsilon^{pqrst} s_a^1 P_p^a s_b^2 P_q^b s_c^3 P_r^c s'_d{}^1 P'_s{}^d s'_e{}^2 P'_t{}^e = 0 \quad (3.32)$$

Since Eq. (3.32) holds for any $s'_f{}^3$ we have:

$$\epsilon^{pqrst} s_a^1 P_p^a s_b^2 P_q^b s_c^3 P_r^c s'_d{}^1 P'_s{}^d s'_e{}^2 P'_t{}^e s'_f{}^3 = 0_f \quad (3.33)$$

The planes s_a^1 , s_b^2 and s_c^3 intersect at a point z^i in image 1, and $s'_d{}^1$, $s'_e{}^2$ and $s'_f{}^3$ intersect at a point z'^j in image 2 as follows:

$$\epsilon^{abc i} s_a^1 s_b^2 s_c^3 = z^i \quad (3.34)$$

$$\epsilon^{def j} s'_d{}^1 s'_e{}^2 s'_f{}^3 = z'^j \quad (3.35)$$

Thus, the following equations hold:

$$s_a^1 s_b^2 s_c^3 = \epsilon_{abc i} z^i \quad (3.36)$$

$$s'_d{}^1 s'_e{}^2 s'_f{}^3 = \epsilon_{def j} z'^j \quad (3.37)$$

By substituting Eq. (3.36) and Eq. (3.37) into Eq. (3.33), following relationship is obtained:

$$z^i z'^j \epsilon_{abci} \epsilon_{defj} \epsilon^{pqrst} P_p^a P_q^b P_r^c P_s^d P_t^e = 0_f \quad (3.38)$$

We define a bifocal tensor \mathcal{T}_{fij} as follows:

$$\mathcal{T}_{fij} = \epsilon_{abci} \epsilon_{defj} \epsilon^{pqrst} P_p^a P_q^b P_r^c P_s^d P_t^e \quad (3.39)$$

Then Eq. (3.38) results in following bilinear constraint in frequency space:

$$z^i z'^j \mathcal{T}_{fij} = 0_f \quad (3.40)$$

where $\mathbf{z} = [z^1, z^2, z^3, z^4]^\top$ and $\mathbf{z}' = [z'^1, z'^2, z'^3, z'^4]^\top$ are 3D image points in the frequency space, ϵ_{abci} is a $4 \times 4 \times 4 \times 4$ tensor, which takes 1 for even permutation, -1 for odd permutation and 0 for others. Also, ϵ^{pqrst} is a $5 \times 5 \times 5 \times 5 \times 5$ tensor which takes -1 , 1 and 0 depending on its permutation. The **bifocal tensor** \mathcal{T}_{fij} defined in Eq. (3.39) is a $4 \times 4 \times 4$ tensor, which consists of camera projection matrices only. Thus, the computation of \mathcal{T}_{fij} is same as the calibration of two cameras. Equation (3.40) shows by varying f a set of 4 equations is obtained, each equation having 16 terms. The details of \mathcal{T}_{fij} are postponed till Sec. 3.5.4.

3.5.2 Algebraic verification

Next is the algebraic verification of the bifocal tensor. Considering the frequency domain projection of two cameras and combining Eq. (3.10) and Eq. (3.13):

$$\begin{bmatrix} \mathbf{P} & \mathbf{z} & \mathbf{0} \\ \mathbf{P}' & \mathbf{0} & \mathbf{z}' \end{bmatrix} \begin{bmatrix} \mathbf{Z} \\ -1 \\ -\lambda \end{bmatrix} = \mathbf{0} \quad (3.41)$$

Expanding Eq. (3.41), we have:

$$\begin{bmatrix} p_{11} & p_{12} & p_{13} & 0 & p_{15} & z^1 & 0 \\ p_{21} & p_{22} & p_{23} & 0 & p_{25} & z^2 & 0 \\ 0 & 0 & 0 & p_{34} & p_{35} & z^3 & 0 \\ 0 & 0 & 0 & 0 & 1 & z^4 & 0 \\ \hline p'_{11} & p'_{12} & p'_{13} & 0 & p'_{15} & 0 & z'^1 \\ p'_{21} & p'_{22} & p'_{23} & 0 & p'_{25} & 0 & z'^2 \\ 0 & 0 & 0 & p'_{34} & p'_{35} & 0 & z'^3 \\ 0 & 0 & 0 & 0 & 1 & 0 & z'^4 \end{bmatrix} \begin{bmatrix} \mathbf{Z} \\ -1 \\ -\lambda \end{bmatrix} = \mathbf{0} \quad (3.42)$$

Again, to form multiview relations in such a case is to start eliminating the rows until a square minor is obtained. For this case, a 7x7 minor has a zero determinant. For ease of notation let us name the rows of camera matrices. The 4 rows of camera 1 are \mathbf{p}^a , \mathbf{p}^b , \mathbf{p}^c and \mathbf{p}^i . The four row vectors of camera 2 will be referred to as \mathbf{p}'^d , \mathbf{p}'^e , \mathbf{p}'^f and \mathbf{p}'^j . Then expansion of the determinant by cofactors results in following constraint:

$$z^i z'^j \epsilon_{abci} \epsilon_{defj} \det \begin{bmatrix} \mathbf{p}^a \\ \mathbf{p}^b \\ \mathbf{p}^c \\ \mathbf{p}'^d \\ \mathbf{p}'^e \end{bmatrix} = 0_f \quad (3.43)$$

The tensor \mathcal{T}_{fij} is described as follows:

$$\mathcal{T}_{fij} = \epsilon_{abci} \epsilon_{defj} \det \begin{bmatrix} \mathbf{p}^a \\ \mathbf{p}^b \\ \mathbf{p}^c \\ \mathbf{p}'^d \\ \mathbf{p}'^e \end{bmatrix} \quad (3.44)$$

From Eq. (3.43) and (3.44), we have the bilinear constraints shown in Eq. (3.40). A part of \mathcal{T}_{443} is shown as an example in Eq. (3.42), with blue colored elements indicating the eliminated matrix part.

3.5.3 Generalization to Texture Patterns

Equation (3.40) represents two view constraints for sequential patterns in frequency space. For clarity, it can be rewritten in the following form:

$$z^i(n)z'^j(n)\mathcal{T}_{fij} = 0_f \quad (3.45)$$

For the case of texture patterns i.e. 2D sequential patterns, the two view constraint can be generalized to following form:

$$z^i(n, m)z'^j(n, m)\mathcal{T}_{fij} = 0_f \quad (3.46)$$

The bifocal tensor \mathcal{T}_{fij} in Eq. (3.46) is identical with that in Eq. (3.45), since it is irrelevant to objects observed by the cameras. Eq. (3.45) and Eq. (3.46) are two view relations in 4D space, which are important for recovering 4D points from 3D images.

3.5.4 Bifocal tensor

Tensor \mathcal{T}_{fij} is the algebraic representation of extended two view geometry. It represents the relative geometric and photometric relationship between two cameras. From Eq. (3.40), it can be seen that the tensor \mathcal{T}_{fij} can be computed from corresponding points in frequency space, \mathbf{z} and \mathbf{z}' . Camera matrix Eq. (3.3) and Eq. (3.39) gives an insight of \mathcal{T}_{fij} tensor structure. First, affine camera structure shows presence of zero terms in tensor. Hence only 18 components are non-zero out of 64 elements of the tensor \mathcal{T}_{fij} . It also shows the elements will be repeated twice due to presence of f and j in one ϵ term. The components of \mathcal{T}_{fij} are non-zero when all the following conditions hold;

1. The two fourth rows of both camera matrices should not be present for computation of \mathcal{T}_{fij} in Eq. (3.39).
2. If one fourth row exists, two third rows of camera matrices should not be present in Eq. (3.39).
3. Third row for at least one camera must exist in Eq. (3.39).
4. $f \neq j$. Otherwise $\epsilon_{defj} = 0$, and \mathcal{T}_{fij} vanishes.

In tensor \mathcal{T}_{fij} , there are 18 non-zero components. However, 9 of them have the same values with the other 9 components, and only the sign is different. Therefore, only 9 elements are unique and 8 are independent due to scale ambiguity. The unique elements are \mathcal{T}_{144} , \mathcal{T}_{244} , \mathcal{T}_{341} , \mathcal{T}_{342} , \mathcal{T}_{344} , \mathcal{T}_{413} , \mathcal{T}_{423} , \mathcal{T}_{431} and \mathcal{T}_{432} . The other 9 non-zero components can be obtained by taking negative and reversing the indices as follows: $\mathcal{T}_{144} = -\mathcal{T}_{441}$, $\mathcal{T}_{244} = -\mathcal{T}_{442}$, $\mathcal{T}_{341} = -\mathcal{T}_{143}$, $\mathcal{T}_{342} = -\mathcal{T}_{243}$, $\mathcal{T}_{344} = -\mathcal{T}_{443}$, $\mathcal{T}_{413} = -\mathcal{T}_{314}$, $\mathcal{T}_{423} = -\mathcal{T}_{324}$, $\mathcal{T}_{431} = -\mathcal{T}_{134}$, $\mathcal{T}_{432} = -\mathcal{T}_{234}$.

Next, let us ponder on uniqueness of 9 components of \mathcal{T}_{fij} . In Eq. (3.39) apart from the ϵ terms (Levi-Civita tensors) to be non-zero, the selection of camera rows is also important for tensor elements to be non-zero. Right hand side of Eq. (3.39) shows the variables a, b, c, d and e are used for selecting the camera rows whereas the variable i, j and f only contribute to the value of ϵ term. Also, if ϵ_{defj} is 1 then ϵ_{dejf} is always -1 . Therefore, for elements of tensor \mathcal{T}_{fij} whenever the values of variables f and j switch such that the value of f in one element is j in the other, and value of j in one element is f in the other, those two tensor elements will have the same value with only difference in sign. For example, let us consider two elements \mathcal{T}_{134} and \mathcal{T}_{431} . Observe that for \mathcal{T}_{134} , $f=1$ and $j=4$ and for \mathcal{T}_{431} , $f=4$ and $j=1$. Expansion of Eq. (3.39) for \mathcal{T}_{134} , shows that the combinations of a, b, c, i and d, e, f, j for non-zero terms are: $\{1243, 2314\}$, $\{1243, 3214\}$, $\{1423, 2314\}$, $\{1423, 3214\}$, $\{2143, 2314\}$, $\{2143, 3214\}$, $\{2413, 2314\}$, $\{2413, 3214\}$, $\{4123, 2314\}$, $\{4123, 3214\}$, $\{4213, 2314\}$, $\{4213, 3214\}$. Similarly, to compute \mathcal{T}_{431} the combinations of a, b, c, i and d, e, f, j for non-zero terms include: $\{1243, 2341\}$, $\{1243, 3241\}$, $\{1423, 2341\}$, $\{1423, 3241\}$, $\{2143, 2341\}$, $\{2143, 3241\}$, $\{2413, 2341\}$, $\{2413, 3241\}$, $\{4123, 2341\}$, $\{4123, 3241\}$, $\{4213, 2341\}$, $\{4213, 3241\}$. Comparison of these combinations shows the difference is only in the values of f and j . This means that the numerical value of \mathcal{T}_{134} and \mathcal{T}_{431} is same, and only sign is different. The same is true for all 18 elements. Hence, we have only 9 unique tensor components.

3.5.5 Computation of tensor

In this section computation of tensor \mathcal{T}_{fij} from point correspondences in frequency space is shown. From Eq. (3.40) we obtain following 4 equations, in case of $n = 0$.

$$-z^3 z'^4 \mathcal{T}_{431} - z^4 z'^3 \mathcal{T}_{341} + z^4 z'^4 \mathcal{T}_{144} = 0 \quad (3.47)$$

$$-z^3 z'^4 \mathcal{T}_{432} - z^4 z'^3 \mathcal{T}_{342} + z^4 z'^4 \mathcal{T}_{244} = 0 \quad (3.48)$$

$$\begin{aligned} -z^1 z'^4 \mathcal{T}_{413} - z^2 z'^4 \mathcal{T}_{423} + z^4 z'^1 \mathcal{T}_{341} + \\ z^4 z'^2 \mathcal{T}_{342} + z^4 z'^4 \mathcal{T}_{344} = 0 \end{aligned} \quad (3.49)$$

$$\begin{aligned} z^1 z'^3 \mathcal{T}_{413} + z^2 z'^3 \mathcal{T}_{423} + z^3 z'^1 \mathcal{T}_{431} + z^3 z'^2 \mathcal{T}_{432} - \\ z^4 z'^1 \mathcal{T}_{144} - z^4 z'^2 \mathcal{T}_{244} - z^4 z'^3 \mathcal{T}_{344} = 0 \end{aligned} \quad (3.50)$$

For correspondence at $n \neq 0$, $z^4 = z'^4 = 0$ in frequency space. This results in only 1 independent equation as follows:

$$z^1 z'^3 \mathcal{T}_{413} + z^2 z'^3 \mathcal{T}_{423} + z^3 z'^1 \mathcal{T}_{431} + z^3 z'^2 \mathcal{T}_{432} = 0 \quad (3.51)$$

We may stack these equations in a matrix form as follows:

$$\mathbf{MT} = \mathbf{0} \quad (3.52)$$

where $\mathbf{T} = [\mathcal{T}_{144}, \mathcal{T}_{244}, \mathcal{T}_{341}, \mathcal{T}_{342}, \mathcal{T}_{344}, \mathcal{T}_{413}, \mathcal{T}_{423}, \mathcal{T}_{431}, \mathcal{T}_{432}]^\top$ and \mathbf{M} is 5×9 matrix.

$$\mathbf{M} = \begin{bmatrix} z^4 z'^4 & 0 & -z^4 z'^3 & 0 & 0 & 0 & 0 & -z^3 z'^4 & 0 \\ 0 & z^4 z'^4 & 0 & -z^4 z'^3 & 0 & 0 & 0 & 0 & -z^3 z'^4 \\ 0 & 0 & z^4 z'^1 & z^4 z'^2 & z^4 z'^4 & -z^1 z'^4 & -z^2 z'^4 & 0 & 0 \\ -z^4 z'^1 & -z^4 z'^2 & 0 & 0 & -z^4 z'^3 & z^1 z'^3 & z^2 z'^3 & z^3 z'^1 & z^3 z'^2 \\ 0 & 0 & 0 & 0 & 0 & z^1 z'^3 & z^2 z'^3 & z^3 z'^1 & z^3 z'^2 \end{bmatrix} \quad (3.53)$$

The 9 elements of tensor \mathcal{T}_{fij} can be found by computing the least square solution of Eq. (3.52). Although we have 4 equations in $n = 0$, only three of them are linearly independent. This can be observed in Eq. (3.47)– Eq. (3.50). Equation (3.50) is a linear combination of Eq. (3.47), Eq. (3.48) and Eq. (3.49).

3.5.6 Minimum No. of Points for Tensor Computation

The elements $\mathcal{T}_{144}, \mathcal{T}_{244}, \mathcal{T}_{341}, \mathcal{T}_{342}$ and \mathcal{T}_{344} can be computed only from the point correspondence at $n = 0$, as it can be seen from Eqs. (3.47)–(3.51). The other tensor elements $\mathcal{T}_{413}, \mathcal{T}_{423}, \mathcal{T}_{431}$ and \mathcal{T}_{432} can be obtained from corresponding points at $n \neq 0$ or $n = 0$. Accordingly, at least 4 corresponding points (2 from $n = 0$ and 2 from $n \neq 0$) are required for the computation of bifocal tensor \mathcal{T}_{fij} . Since 2 correspondence at $n = 0$ are required, at least two sequential patterns are required for computing \mathcal{T}_{fij} .

It should be noted that for 4 corresponding frequencies, the matrix of image coordinates in Eq. (3.53) would be of size 10×9 with a matrix rank of 8, which is sufficient to compute 8 elements of scaled \mathcal{T}_{fij} . If we have k more point correspondences for $n \neq 0$ the size of \mathbf{M} increases as $(8 + k) \times 9$.

The minimum no. of points for computing the tensor from sequential patterns in Eq. (3.45) and texture patterns in Eq. (3.46) is same. However, for texture patterns at least three frequencies should be used e.g. 2 correspondences at $n = 0$, and 1 correspondence at $n = 1$ and $n = 2$ make up for rank 8 matrix \mathbf{M} in Eq. (3.53).

3.5.7 Maximum No. of Cameras

Since the number of hyperplanes to form a multilinear constraint is 5. The maximum number of cameras/images for 4D to 3D projection is five. If we denote corresponding points of image 1, 2, ..., 5 in frequency space as \mathbf{z} , \mathbf{z}' , \mathbf{z}'' , \mathbf{z}''' and \mathbf{z}'''' and the camera matrices as \mathbf{P} , \mathbf{P}' , \mathbf{P}'' , \mathbf{P}''' and \mathbf{P}'''' . The multiview relations in frequency domain for 3, 4 and 5 cameras are shown below:

$$z^i z'^j z''^k \epsilon_{defj} \epsilon_{ghlk} \mathcal{A}_i^{dg} = 0_{efhl} \quad (3.54)$$

$$z^i z'^j z''^k z''''^l \epsilon_{defj} \epsilon_{ghmk} \epsilon_{nopl} \mathcal{D}_{ci}^{dgn} = 0_{cefhmop} \quad (3.55)$$

$$z^i z'^j z''^k z''''^p z''''^q \epsilon_{abci} \epsilon_{defj} \epsilon_{ghlk} \epsilon_{mnop} \epsilon_{qrst} \mathcal{C}^{adgmq} = 0_{bcefhlnors} \quad (3.56)$$

where tensors \mathcal{A}_i^{dg} , \mathcal{D}_{ci}^{dgn} and \mathcal{C}^{adgmq} have following definitions:

$$\mathcal{A}_i^{dg} = \epsilon_{abci} \epsilon^{pqrst} P_p^a P_q^b P_r^c P_s^d P_t^g \quad (3.57)$$

$$\mathcal{D}_{ci}^{dgn} = \epsilon_{abci} \epsilon^{uvwxyz} P_u^a P_v^b P_w^d P_x^g P_y^n \quad (3.58)$$

$$\mathcal{C}^{adgmq} = \epsilon^{uvwxyz} P_u^a P_v^d P_w^g P_x^m P_y^q \quad (3.59)$$

3.6 Sampling shift

Sampling shifts for sequential patterns and texture patterns can be found using the following methods.

3.6.1 Sampling shift for Sequential Patterns

In this section, a method to find the relative sampling shift is explained. For sequential patterns the sampling shift is one dimensional as shown in Eq. (3.9), since all points in sequence are connected in a single line. Considering upper 3×5 part of each camera matrix and upper 3×1 part of each image frequency z and z' in Eq. (3.10) and Eq. (3.12), we have following equations:

$$\begin{bmatrix} p_{11} & p_{12} & p_{13} & 0 & p_{15} & z^1 \\ p_{21} & p_{22} & p_{23} & 0 & p_{25} & z^2 \\ 0 & 0 & 0 & p_{34} & p_{35} & z^3 \\ p'_{11} & p'_{12} & p'_{12} & 0 & p'_{15} & \lambda z'^1 \\ p'_{21} & p'_{22} & p'_{23} & 0 & p'_{25} & \lambda z'^2 \\ 0 & 0 & 0 & p'_{34} & p'_{35} & \lambda z'^3 \end{bmatrix} \begin{bmatrix} \mathbf{Z} \\ -1 \end{bmatrix} = \mathbf{0} \quad (3.60)$$

where $\lambda = e^{jn\theta}$ and $\theta = \frac{2\pi s}{N}$. It is the phase shift in image 2, as shown in Eq. (3.12). Since Eq. (3.60) has a solution the determinant of the left most matrix must be zero. To obtain a valid constraint between image coordinates, the last column must be eliminated as follows:

$$z^1 Q_1 + z^2 Q_2 + z^3 Q_3 + \lambda(z'^1 Q_4 + z'^2 Q_5 + z'^3 Q_6) = 0 \quad (3.61)$$

where Q_i are the determinants of the 5×5 minors obtained by eliminating the last column and i th row as follows:

$$Q_i = (-1)^{i+6} \det[\sim m^i] \quad (3.62)$$

where $\sim m^i$ denotes the eliminated row. Separating the phase part in Eq. (3.61) we have:

$$\lambda(n) = e^{jn\theta} = -\frac{z^1 Q_1 + z^2 Q_2 + z^3 Q_3}{z'^1 Q_4 + z'^2 Q_5 + z'^3 Q_6} \quad (3.63)$$

From Eq. (3.63), it seems possible to compute shift s from a single λ of any n ($n \neq 0$). However, this is not the case and only λ of $n = 1$ can be used. When $n \geq 2$ multiple solutions of s are obtained, and because of aliasing correct value of s cannot be

distinguished from wrong ones. Since the complex exponentials vary from 0 to 2π and any value greater than 2π is again mapped to this range. For example, let $N = 5$ then we cannot distinguish between $e^{\frac{4\pi j}{5}}$ and $e^{\frac{14\pi j}{5}}$. This phenomenon is termed as aliasing. Thus, only λ of $n = 1$ can be used. On the contrary if the ratio of two consecutive frequencies is used, say $\lambda(n_1)$ and $\lambda(n_2)$, we do not have the aliasing problem in Eq. (3.64), and all the n frequency components can be used for computing s .

$$\frac{\lambda(n_2)}{\lambda(n_1)} = \frac{e^{jn_2\theta}}{e^{jn_1\theta}} = e^{j(n_2-n_1)\theta} \quad (3.64)$$

where $n_2 - n_1 = 1$. In this case, the sampling shift s can be computed as follows:

$$s = \frac{N}{2\pi j(n_2 - n_1)} \log \frac{\lambda(n_2)}{\lambda(n_1)} \quad (3.65)$$

Equation (3.65) describes the case when $\lambda(n_1)$ and $\lambda(n_2)$ have unit norm. In the presence of noise, $\lambda(n_1)$ and $\lambda(n_2)$ are divided by their magnitude, so that they have unit norm. Since s obtained from Eq. (3.65) is a complex number only the real part is taken, and rounded off to obtain an integer. After obtaining s for each frequency, the most repeated value of s is considered as the sampling shift.

3.6.2 Sampling shift for Texture Patterns

Sampling shift for texture can be computed in the same way as for sequential patterns. Although the shift is in two dimensions, knowing the nature of Fourier basis matrix, the coefficients of shifts can be considered separately. The shift in case of textures is of the form $e^{jn\theta}e^{jm\phi} = e^{\frac{2\pi js}{N}}e^{\frac{2\pi jt}{M}}$, as shown in Eq. (3.19). For example, when $N = M = 5$, $\lambda(n, m)$ is as follows:

$$\lambda(n, m) = \begin{bmatrix} 1 & e^{\frac{2j\pi s}{5}} & e^{\frac{4j\pi s}{5}} & e^{\frac{6j\pi s}{5}} & e^{\frac{8j\pi s}{5}} \\ e^{\frac{2j\pi t}{5}} & e^{\frac{2j\pi s}{5} + \frac{2j\pi t}{5}} & e^{\frac{4j\pi s}{5} + \frac{2j\pi t}{5}} & e^{\frac{6j\pi s}{5} + \frac{2j\pi t}{5}} & e^{\frac{8j\pi s}{5} + \frac{2j\pi t}{5}} \\ e^{\frac{4j\pi t}{5}} & e^{\frac{2j\pi s}{5} + \frac{4j\pi t}{5}} & e^{\frac{4j\pi s}{5} + \frac{4j\pi t}{5}} & e^{\frac{6j\pi s}{5} + \frac{4j\pi t}{5}} & e^{\frac{8j\pi s}{5} + \frac{4j\pi t}{5}} \\ e^{\frac{6j\pi t}{5}} & e^{\frac{2j\pi s}{5} + \frac{6j\pi t}{5}} & e^{\frac{4j\pi s}{5} + \frac{6j\pi t}{5}} & e^{\frac{6j\pi s}{5} + \frac{6j\pi t}{5}} & e^{\frac{8j\pi s}{5} + \frac{6j\pi t}{5}} \\ e^{\frac{8j\pi t}{5}} & e^{\frac{2j\pi s}{5} + \frac{8j\pi t}{5}} & e^{\frac{4j\pi s}{5} + \frac{8j\pi t}{5}} & e^{\frac{6j\pi s}{5} + \frac{8j\pi t}{5}} & e^{\frac{8j\pi s}{5} + \frac{8j\pi t}{5}} \end{bmatrix} \quad (3.66)$$

The phase shift $e^{\frac{2\pi js}{N}}$ is constant in vertical direction and $e^{\frac{2\pi jt}{M}}$ is constant in horizon-

tal direction. Therefore, taking ratio of two consecutive frequencies in vertical direction $\frac{\lambda(n_1, m_1)}{\lambda(n_1, m_2)}$ eliminates the horizontal shift and gives us vertical shift value t . Similarly, taking ratio of two consecutive frequencies in horizontal direction $\frac{\lambda(n_2, m_1)}{\lambda(n_1, m_1)}$ would provide us horizontal shift value s . Working same as Eq. (3.60) – Eq. (3.63) following equation is obtained:

$$\lambda(n, m) = -\frac{-z^1 Q_1 + z^2 Q_2 - z^3 Q_3}{-z'^1 Q_4 + z'^2 Q_5 - z'^3 Q_6} \quad (3.67)$$

where Q_i is defined as Eq. (3.62). The horizontal and vertical shifts can be computed as follows:

$$s = \frac{N}{2\pi j(n_2 - n_1)} \log \frac{\lambda(n_2, m_1)}{\lambda(n_1, m_1)} \quad (3.68)$$

$$t = \frac{M}{2\pi j(m_2 - m_1)} \log \frac{\lambda(n_1, m_2)}{\lambda(n_1, m_1)} \quad (3.69)$$

where $n_2 - n_1 = m_2 - m_1 = 1$ for consecutive frequencies.

3.7 4D Reconstruction

We next consider a method for reconstructing 4D sequential patterns and texture patterns without knowing individual point correspondence. For reconstructing 4D points \mathbf{X} we first reconstruct 4D points \mathbf{Z} in frequency space. For this objective, we have to recover camera matrices. We consider the following canonical affine camera pair:

$$\mathbf{P} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.70)$$

$$\mathbf{P}' = [\mathbf{H}|\mathbf{e}'] \quad (3.71)$$

where \mathbf{H} represents the homography between a pair of images and \mathbf{e}' is the epipole in the second image. Then, \mathbf{H} and \mathbf{e}' can be computed from \mathcal{T}_{fij} as described in the following sections. Once the homography and the epipole are computed, camera matrices are

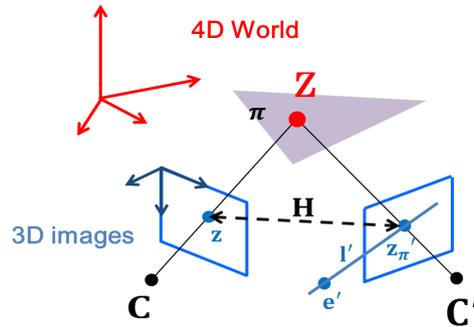


Figure 3.6: Homography induced by a plane.

recovered from Eq. (3.70) and Eq. (3.71), and the 4D points $\mathbf{Z}(i)$ ($n = 1, \dots, N$) can be reconstructed. Then the 4D point \mathbf{X} in the real space can be obtained by inverse Fourier transform of $\mathbf{Z}(i)$.

3.7.1 Computation of Epipoles

For a given bifocal tensor \mathcal{T}_{fij} , the epipole \mathbf{e} in image 1 can be computed as its left null space. Similarly, the epipole \mathbf{e}' in image 2 can be computed as the right null space of \mathcal{T}_{fij} , as follows:

$$e^i \mathcal{T}_{fij} = 0_{fj} \quad (3.72)$$

$$e'^j \mathcal{T}_{fij} = 0_{fi} \quad (3.73)$$

Equation (3.73) can be described in matrix form as follows:

$$\mathbf{M}\mathbf{e}' = \mathbf{0} \quad (3.74)$$

where, $\mathbf{e}' = [e'^1, e'^2, e'^3, e'^4]^\top$, and \mathbf{M} is a 16×4 matrix, which consists of the components of \mathcal{T}_{fij} . Thus \mathbf{e}' can be obtained by solving Eq. (3.74) as the eigenvector corresponding to the smallest eigenvalue of $\mathbf{M}^\top \mathbf{M}$.

3.7.2 Extraction of Homography

Let us consider an arbitrary but fixed plane π , not passing through any of the camera centers, as shown in Fig. 3.6. Then, a point \mathbf{z} in image 1 is related to a point \mathbf{z}'_π in image 2 via homography \mathbf{H} as follows:

$$z'^l_\pi = H^l_i z^i \quad (3.75)$$

An epipolar line l' in image 2 can be defined as the join of epipole e' and the point \mathbf{z}'_π as follows:

$$l'_{jf} = \epsilon_{jflr} e'^r z'^l_\pi \quad (3.76)$$

Since images are 3 dimensional the epipolar line l' is a second order tensor (i.e. matrix). From Eq. (3.40), epipolar line l' can also be defined as the corresponding line in image 2 for a point \mathbf{z} in image 1 as follows:

$$l'_{jf} = \mathcal{T}_{fij} z^i \quad (3.77)$$

Eq. (3.75), Eq. (3.76) and Eq. (3.77) result in the following equation:

$$H^l_i = \epsilon^{jflr} e'_r \mathcal{T}_{fij} \quad (3.78)$$

By using Eq. (3.78), we can compute the homography \mathbf{H} from the bifocal tensor \mathcal{T}_{fij} and the epipole e' in image 2. Thus, both \mathbf{H} and e' can be computed from tensor \mathcal{T}_{fij} , and camera matrices \mathbf{P} and \mathbf{P}' can be obtained from Eq. (3.70) and Eq. (3.71).

3.7.3 4D Reconstruction and Rectification

Once the camera matrices \mathbf{P} and \mathbf{P}' have been computed, 4D points can be recovered using the following method. Let us consider the camera projection equation in frequency space again:

$$\mathbf{z} = \mathbf{PZ} \quad (3.79)$$

$$\mathbf{z}' = \mathbf{P}'\mathbf{Z} \quad (3.80)$$

Taking the vector product between the left side and the right side of Eq. (3.79) and Eq. (3.80), we have:

$$[\mathbf{z}]_{\times}\mathbf{P}\mathbf{Z} = 0 \quad (3.81)$$

$$[\mathbf{z}']_{\times}\mathbf{P}'\mathbf{Z} = 0 \quad (3.82)$$

where, $[\cdot]_{\times}$ denotes a skew symmetric matrix for vector product. From Eq. (3.81) and Eq. (3.82), we have:

$$\mathbf{M}\mathbf{Z} = \mathbf{0} \quad (3.83)$$

where, \mathbf{M} is a 6×5 matrix as follows:

$$\mathbf{M} = \begin{bmatrix} [\mathbf{z}]_{\times}\mathbf{P} \\ [\mathbf{z}']_{\times}\mathbf{P}' \end{bmatrix} \quad (3.84)$$

Then, by solving Eq. (3.83), 4D point \mathbf{Z} in the frequency space can be recovered. The least squares solution of Eq. (3.83) can be obtained by computing an eigenvector which corresponds to the minimum eigenvalue of $\mathbf{M}^{\top}\mathbf{M}$. After all points $\mathbf{Z}(n)$ ($n = 1, \dots, N$) are reconstructed using Eq. (3.83) in frequency space, we apply inverse Fourier transform to \mathbf{Z} to get original points $\mathbf{X}(i)$ ($i = 1, \dots, N$) in spatial domain. The two objects, reconstructed and the original one still differ by an affine homography. This homography has 20 degrees of freedom and can be computed by choosing five points in general configuration for rectification of the reconstructed object. A case similar to Eq. (2.80).

3.8 Experiments

We next show the experimental results from the proposed reconstruction method. In particular, we show that the proposed method can reconstruct sequential patterns and texture patterns without knowing point correspondences. We first show the results from 1D sequential patterns, followed by the results from 2D sequential patterns, i.e. texture

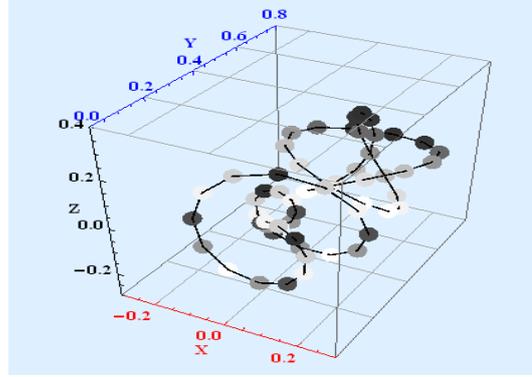


Figure 3.7: Sequential objects used in synthetic experiment. There exist two series of 4D points, which are connected by lines.

patterns.

3.8.1 Sequential Patterns

This section will show reconstruction and arbitrary view generation of 1D sequential patterns.

Synthetic Image Experiment

To ensure that the proposed method works correctly, we first show results from synthetic image experiments.

Figure 3.7 shows a synthetic scene considered in this experiment. There exist two sequences of points connected by lines. As shown in this figure, each sequences has 3D geometric and 1D intensity information, making them 4D objects.

These objects are then projected into two grayscale cameras at different positions with different camera gains as shown in Fig. 3.8 (a) and Fig. 3.8 (b). The 3D images have point position and intensity. Image 1 in Fig. 3.8 (a) is a high gain image, whereas the image 2 in Fig. 3.8 (b) is a low gain image. The image size is 640×480 and intensity ranges from 0 to 255, therefore geometric and photometric coordinates vary in same range i.e. $0 \sim 10^2$. Moreover, we shifted the sampling order of points in image 2 by 9 units to demonstrate the correspondence freeness of our method. Figure 3.8 (c) and (d) show image 2 before and after the sampling shift. Using the proposed method we computed the tensor \mathcal{T}_{fij} from images shown in Fig. 3.8 (a) and (d). Two corresponding

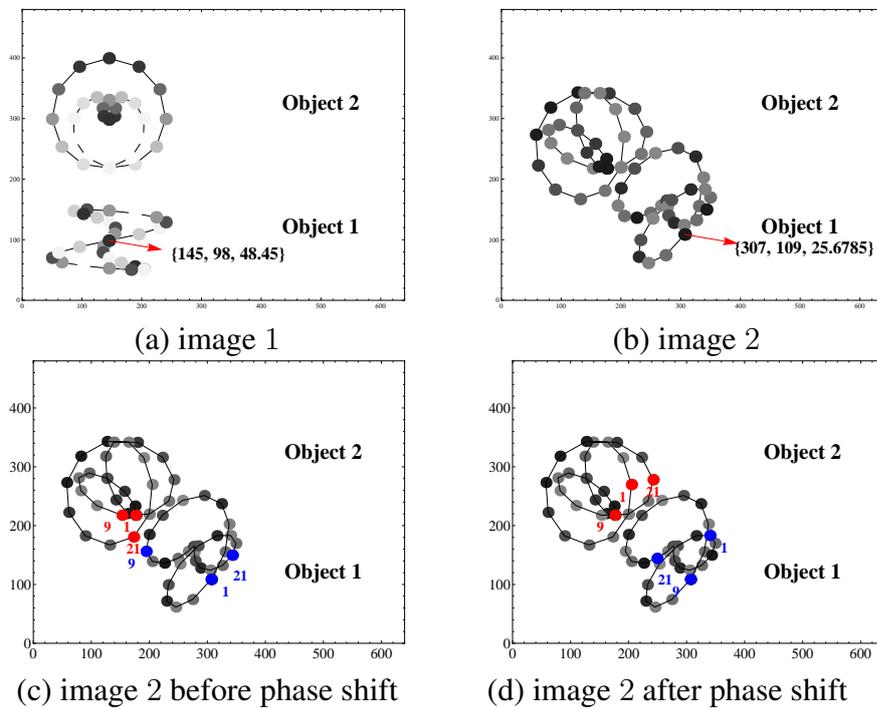


Figure 3.8: 3D Images taken from different viewpoints. Index numbers in red and blue points signify the sampling order.

frequencies at $n = 0$ and two at $n = 1$ were used for tensor computation. Next, the estimated bifocal tensor was used for computing camera matrices and for reconstructing 4D object.

Figure 3.9 (a) and Fig. 3.9 (b) show that, the reconstruction from the proposed method is correct both geometrically and photometrically, while the reconstruction from the classical method is completely wrong because of the wrong correspondence as shown in Fig. 3.9 (c). This shows that unlike the traditional reconstruction method in spatial domain, the proposed method can reconstruct objects, even in the absence of exact corresponding points. It also shows that proposed method is independent of camera sensitivity parameters. This implies that off line calibration of camera intensity parameters is not necessary if the camera sensitivity is approximated linear. Next, we discuss reconstruction for three special cases:

1. Reconstruction is possible for planar sequences as shown in Fig. 3.10 (a) and (b). However for coplanar sequences, there should be a difference in plane heights e.g. if 1 sequence is at plane $Z = 0.3$, the other sequence should be at plane

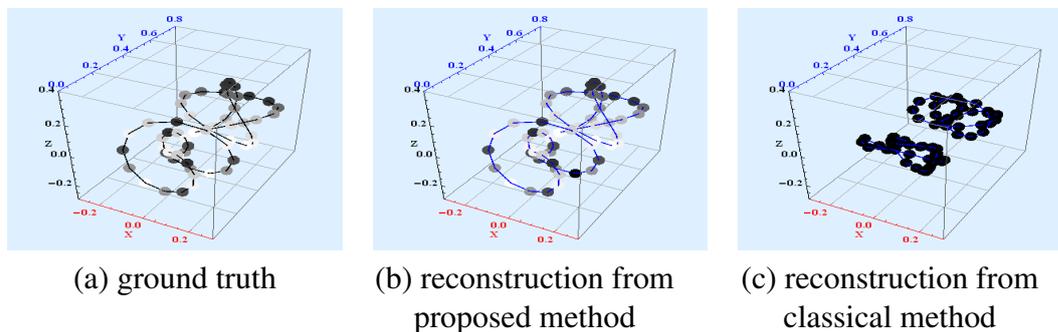


Figure 3.9: Original and reconstructed 4D object. The reconstructed object has correct 3D shape and point intensity values.

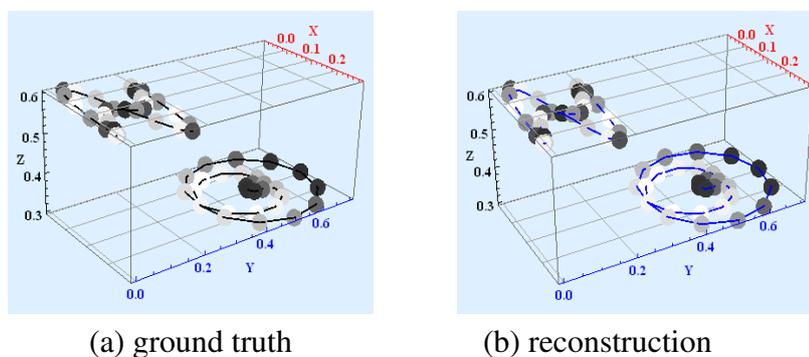


Figure 3.10: Coplanar 4D sequences.

$Z = 0.6$. If both sequences are at the same level, the matrix M in Eq. (3.53) is rank deficient (less than 8), and proper reconstruction is not possible. Another valid configuration of planar sequences is when the sequences are planar perpendicularly, as shown in Fig. 3.11 (a) and (b).

2. Sequences lack intensity variation, as shown in Fig. 3.12 (a). From Eq. (3.53) at least 1 sequence must have intensity variation for reconstruction Fig. 3.12 (b).
3. Sequences are planar and lack intensity variation, as shown in Fig. 3.13. For reconstruction, the rank of matrix M should be 8, therefore the coplanar sequences should differ in plane heights and at least 1 sequence should have intensity variation.

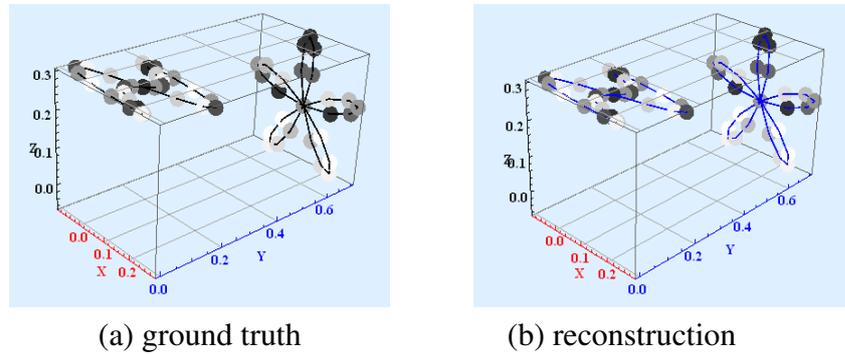


Figure 3.11: Perpendicularly planar 4D sequences.

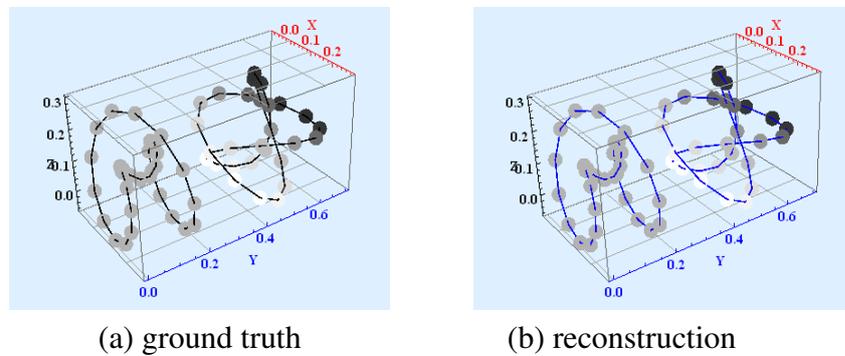


Figure 3.12: 4D sequences. One sequence has constant intensity.

Real Image Experiment

We next show the results from real image experiments. Figure 3.14 (a) and (b) show two real images. Image 1 in Fig. 3.14 (a) is a high gain image, taken by a high gain camera, while image 2 in Fig. 3.14 (b) is a low gain image.

In these images, necklace beads and patchwork design provide us two sequences. We consider the end points of these sequences to be connected to form closed loops. The blobs in the sequential pattern were detected using a method described in Appendix B.4. The centroid and average intensity of each blob were used for computing the tensor and for reconstructing the sequential pattern. The detected blobs are shown Fig. 3.15. Index numbers show sampling order. It may be observed that the order in image 2 is shifted by 3 with respect to that in image 1. Note, we regarded the last blob in the sequence is connected with the first blob in the sequence, so that the blobs constitute a closed sequence.

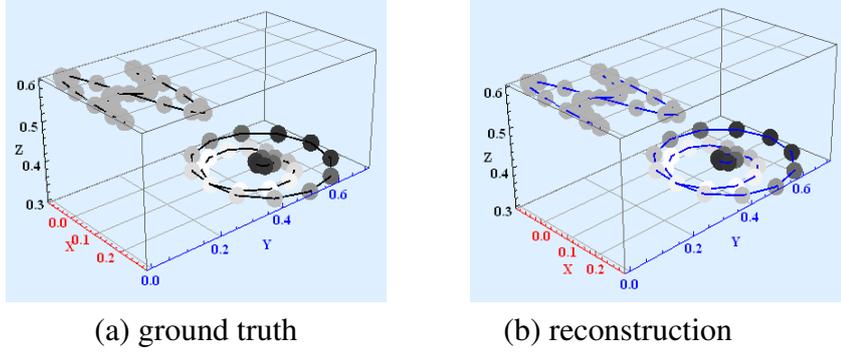


Figure 3.13: Coplanar 4D sequences. One sequence has constant intensity.

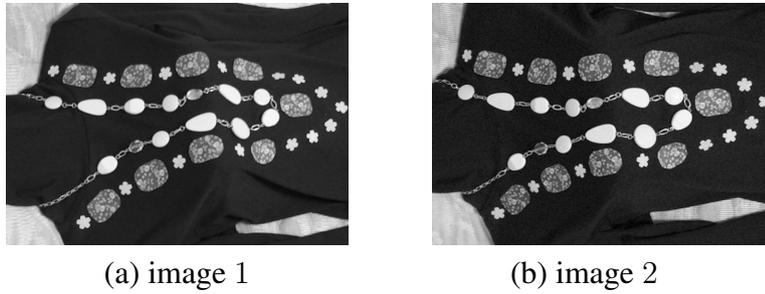


Figure 3.14: Images of sequential objects used in real image experiment.

Using the images shown in Fig. 3.14 (a) and Fig. 3.14 (b) the bifocal tensor \mathcal{T}_{fij} was computed, and the original objects were reconstructed in frequency domain. Two corresponding frequencies at $n = 0$, one corresponding frequency at $n = 1$ and one corresponding frequency at $n = 2$ obtained from two sequential patterns were used for tensor computation. The phase shift was also estimated. The estimated phase shift was 3 as we expected. Inverse discrete Fourier transform was applied to obtain reconstruction in spatial domain. The 3D shape of mannequin was measured to obtain the ground truth shape of sequential patterns. Figure 3.16 (a) and (b) show the ground truth and reconstructed object from the proposed method, while Fig. 3.16 (c) shows that from classical reconstruction method. These results confirm following important results (i) correspondence freeness of our multiview relations (ii) camera sensitivity parameter freeness of our linear camera model. It shows that our method is independent of camera intensity parameters, and we can freely use it for images taken by cameras with different camera gains. Figure 3.17 (a) and (b) show reprojected images of two arbitrary views of the sequential objects reconstructed in Fig. 3.16 (a) and (b). The actual image points

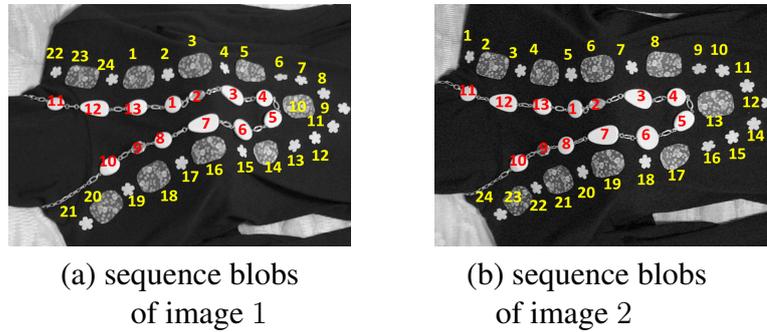


Figure 3.15: The extracted blobs of two sequences are indexed in red and yellow. Sampling order of image 2 sequences is shifted by 3 w.r.t the sampling order of image 1 sequences.

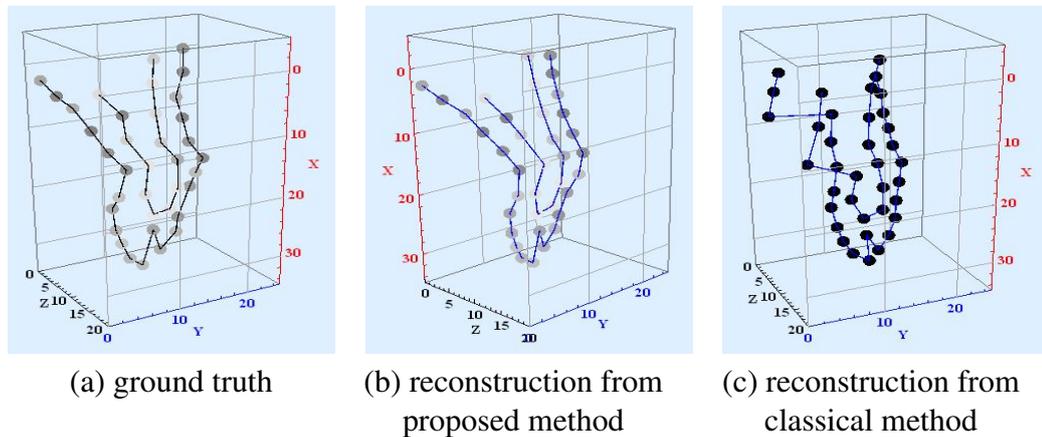


Figure 3.16: Reconstruction result of real image experiment.

are connected by the green lines whereas the reprojected image points are joined by the red lines. These results show the proposed method is able to generate arbitrary views of a reconstructed sequential object properly. Other examples of sequential object reconstruction can be found in [37].

Stability Evaluation

We next show the stability of 4D points reconstructed by the proposed method. For stability analysis a Gaussian noise with standard deviation of 1 pixel was added to each image point, and a Gaussian noise with standard deviation of 1 was added to each image point intensity. The image size is 640×480 and the image intensity varies from 0 to 255.

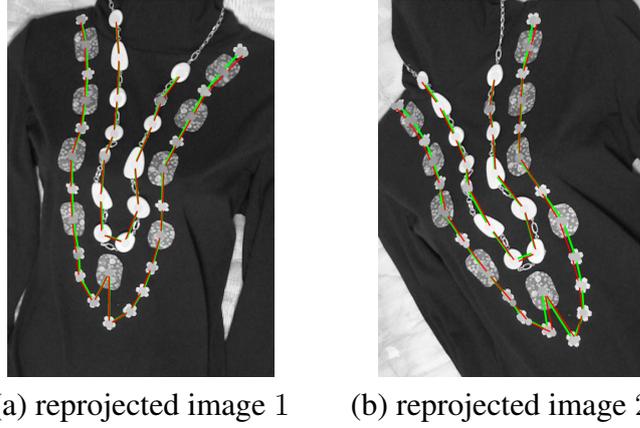


Figure 3.17: Arbitrary views of real image experiment. Green and red lines connect the actual and reprojected image points respectively.

Then, the following reconstruction error was computed from N reconstructed points:

$$E = \frac{1}{N} \sum_{i=1}^N d(\mathbf{Q}(i) - \hat{\mathbf{Q}}_r(i))^2 \quad (3.85)$$

where $d(\mathbf{Q}(i) - \hat{\mathbf{Q}}_r(i))^2$ represents the squared distance between the true quantity and reconstructed quantity in spatial domain. \mathbf{Q} is 3D geometric coordinates for Fig. 3.18 (a) and 1D intensity information for Fig. 3.18 (b). N is total no. of points. By varying the experimental conditions, such as 4D configuration, camera position, and sensitivity, and noise magnitude, we increased the number of corresponding frequencies, i.e. number of corresponding points in the frequency space, for computing the bifocal tensor from 4 (minimum number required) to 20 and evaluated the reconstruction errors. These frequencies were chosen, so that they were the lowest N frequencies. The result were averaged for 500 iterations. Figure 3.18 (a) shows the relationship between the number of corresponding frequencies used for tensor computation and the reconstruction error in 3D shape; the horizontal axis represents the the number of corresponding frequencies for tensor computation and the vertical axis shows the reconstruction error in spatial domain. Figure 3.18 (b) shows the graph between the number of frequencies for bifocal tensor computation and reconstruction error in intensity; the horizontal axis shows the the number of frequencies used for tensor computation and the vertical axis represents the reconstruction error in intensity.

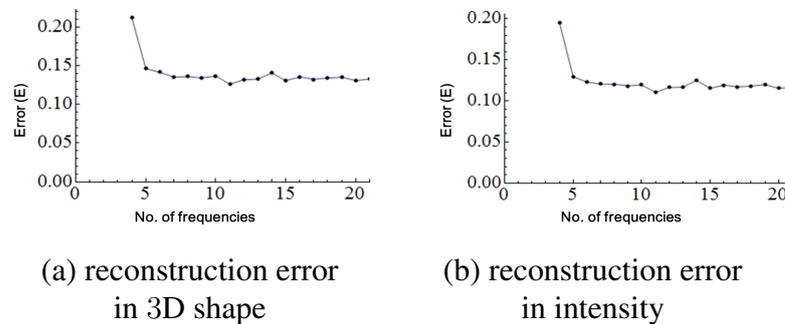


Figure 3.18: Stability of reconstruction. The horizontal axes represent the number of frequencies tensor computation. The vertical axes show errors in shape and intensity.

Figure 3.18 (a) and (b) show that as we increase the number of corresponding frequencies, the magnitude of reconstruction error is reduced resulting in better reconstruction. The graph is the generic behavior of tensor stability i.e. sharp reduction of error at the beginning followed by gradual decay. However, we find that the error does not decrease monotonically. This is because some frequencies are close to zero and lack variation in the frequency space. Thus, the choice of frequencies which have enough variation is important for stable computation.

3.8.2 Texture Patterns

We next show experimental results from 2D sequential patterns, i.e. texture patterns.

Synthetic Image Experiment

Figure. 3.19 shows the texture object considered. The object has 3D geometric information and repetitive intensity information, and therefore it has a repetitive texture surface with each texture element being 4D point. Then this object was projected into two cameras with different parameters. Fig. 3.20(a) shows high intensity image and (b) shows low intensity image. The sampling of two images is unknown. Therefore, we consider different sampling start points, as shown in Fig. 3.20. As it can be seen that the start point of image 2 is shifted w.r.t sampling start point of image 1. Using the proposed method, we computed the bifocal tensor \mathcal{T}_{fij} , and reconstructed the texture pattern. In this experiment, two corresponding frequencies at $n = 0$, two corresponding

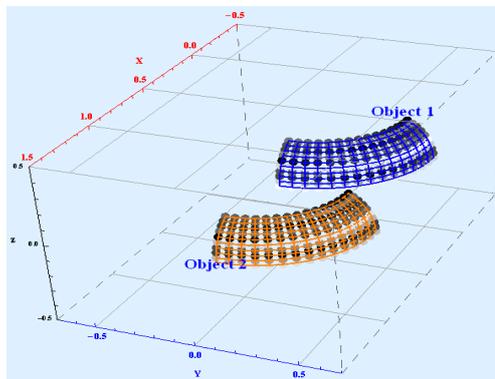


Figure 3.19: Grayscale texture object used in synthetic experiment. There exist two objects. The texture elements are connected in a 2D grid.

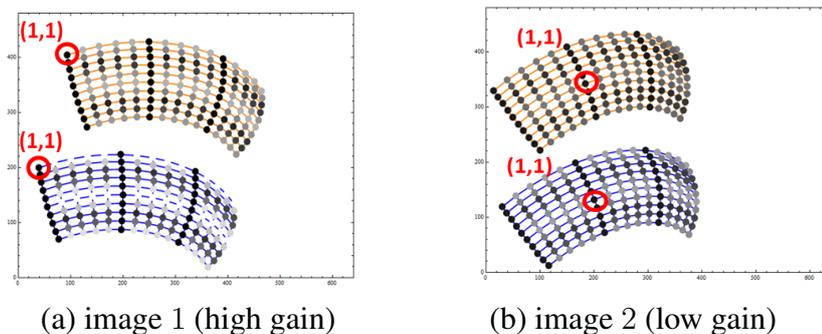


Figure 3.20: 3D Images of a grayscale texture. The first sampling start point is shown in each image.

frequencies at $n = 1$ and one corresponding frequency at $n = 2$ obtained from two texture objects were used for tensor computation. The reconstruction result is shown in Fig. 3.21 which shows the object is reconstructed accurately by using the proposed method. Both the 3D geometric and 1D photometric information are correct. The texture object was then reprojected to virtual cameras with different camera gains. The arbitrary views are shown in Fig. 3.22.

Next, we discuss reconstruction for three special cases:

1. Textures are planar Fig. 3.23 (a). From Eq. (3.53), reconstruction requires rank of matrix M to be 8. Therefore, the planar textures should vary in plane heights, as shown in Fig. 3.23. Another valid configuration for planar textures is to be perpendicular, as shown in Fig. 3.24.

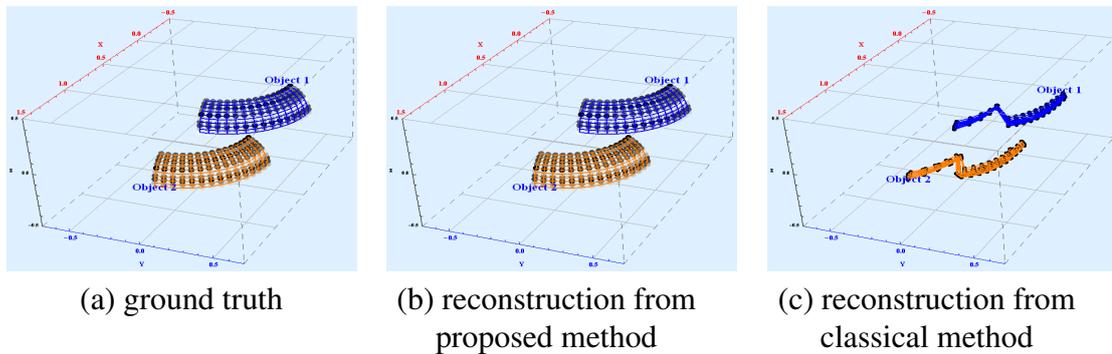


Figure 3.21: Ground truth and reconstructed grayscale texture. The reconstructed texture object has correct 3D shape and point intensity values.

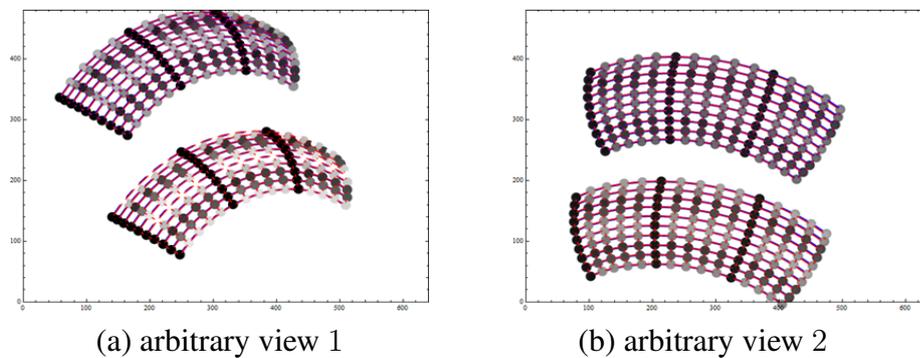


Figure 3.22: Arbitrary views of reconstructed texture pattern.

2. Textures lack intensity variation Fig. 3.25. For reconstruction, at least one texture pattern should have varying intensity.
3. Textures are planar and lack intensity variation Fig. 3.26. For similar reasons as above, the planar textures should differ in plane heights and at least 1 texture pattern should have varying intensity.

Lastly, we discuss the case of mixed objects. When sequences and textures both are present. Sequences are considered using 1D Fourier transform and textures using 2D Fourier transform. Reconstruction is similar (1D-DFT and 2D-DFT) and tensor requires a minimum of 4 correspondences as shown in Fig. 3.27.

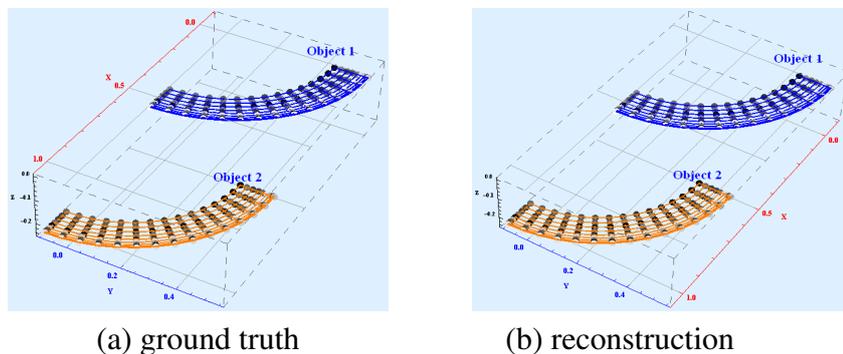


Figure 3.23: Planar texture patterns.

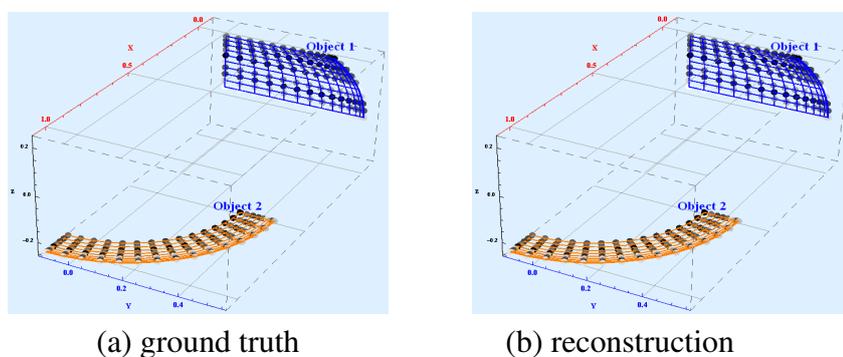


Figure 3.24: Perpendicularly planar texture patterns.

Real Image Experiment

Next, we evaluate our theory using real images. Fig. 3.28 shows texture considered in real image experiments. Image 1 is a high gain image taken by a high intensity gain camera, while image 2 is taken by a low gain camera. The colored check boxes connected by red grid are considered as the texture elements. The images were first processed for blob detection (refer Appendix B.4). These blobs were then connected in a 2D grid pattern. In this experiment, we assumed that the end points of the 2D grid are connected to each other, so that the 2D grid is closed. Although image domain is same, image correspondences are unknown. Therefore sampling start points in image 1 and image 2 are different from each other. After tensor computation, phase was estimated. Horizontal shift was 1 and vertical shift was 2. Two correspondences at $n = 0$ and two correspondences at $n = 1$ were used for tensor computation. Then reconstruction was performed, Fig. 3.29 (a) shows ground truth and (b) shows reconstruction from the

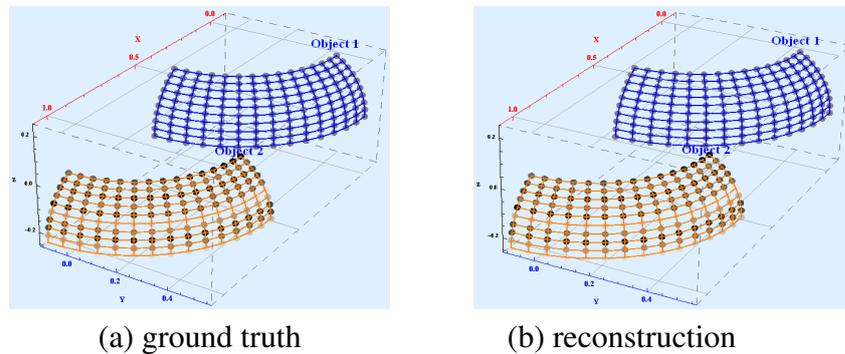


Figure 3.25: Planar texture patterns. One texture pattern has constant intensity.

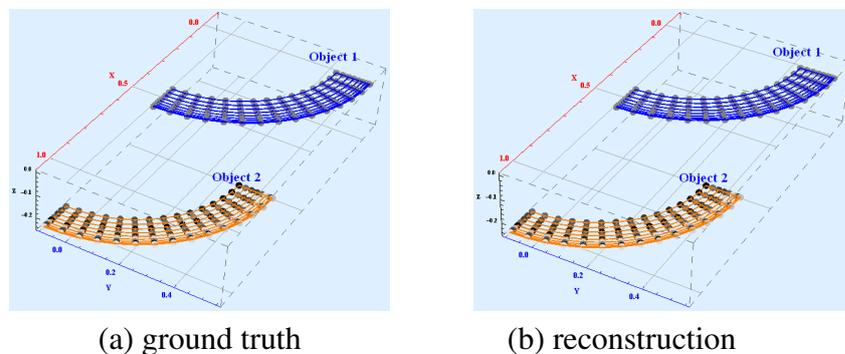


Figure 3.26: Planar texture patterns. One texture pattern has constant intensity.

proposed method, while (c) shows reconstruction from the classical method. Arbitrary view synthesized from reconstruction is shown in Fig. 3.30. Green lines connect the ground truth and red lines connect the reprojected points. These lines are overlapping showing that the reprojected texture overlaps the ground truth image texture. Therefore the arbitrary view has been generated properly. The synthetic and real image experiments show that the proposed method can reconstruct the texture objects correctly, even if there are no exact image correspondences, while the classical reconstruction method cannot reconstruct correctly. Arbitrary views of the texture object can also be accurately generated.

Accuracy Evaluation

Next important step is to consider the stability of the tensor (computed using Eq. (3.46)) used for reconstruction of texture patterns. The experiments were iterated for 3000 iter-

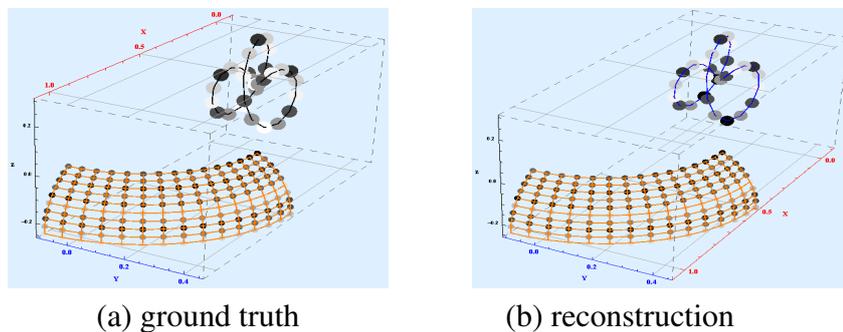


Figure 3.27: Reconstruction of mixed 4D objects.

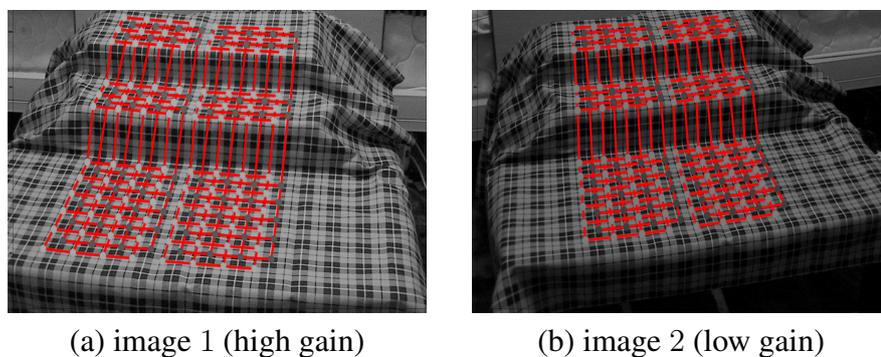


Figure 3.28: Grayscale texture object from different viewpoints.

ations adding image noises and changing conditions such as texture patterns and camera parameters. The reconstruction errors in shape and intensity were separately evaluated. Graph in Fig. 3.31 shows the reconstruction error in shape and intensity. Horizontal axes show the no.of frequencies used for the computation of tensor and vertical axes show the error in shape and intensity. The graph reveals that error reduces by increasing the no. of frequencies leading to stable tensors and better reconstruction results. The graph steepness can be explained in the way similar to Sec. 3.8.1. The graph also shows that tensor is capable of handling different texture objects and varying imaging conditions.

3.9 Summary

In this chapter two view geometry for 4D to 3D affine projection was elaborated. Components of grayscale affine cameras were described. Projection in frequency space and two view constraints for 3D images were discussed and used for reconstructing sequen-

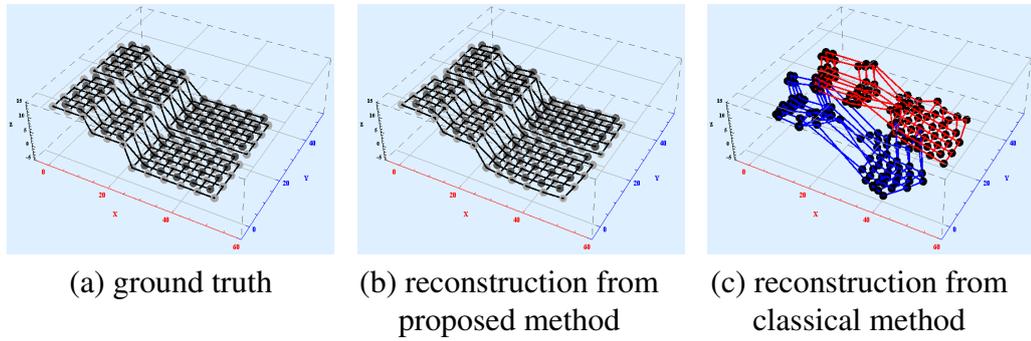


Figure 3.29: Reconstruction of grayscale texture pattern.

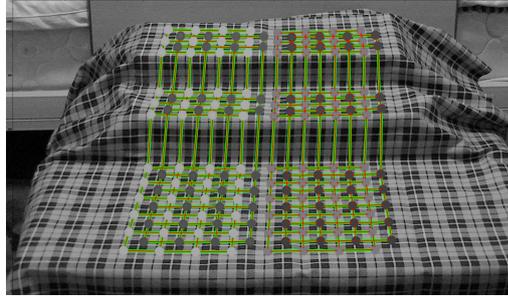
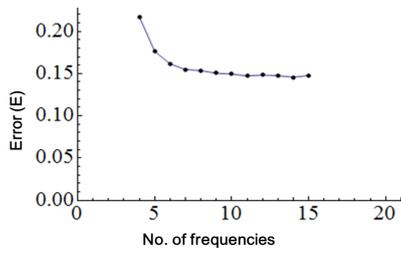
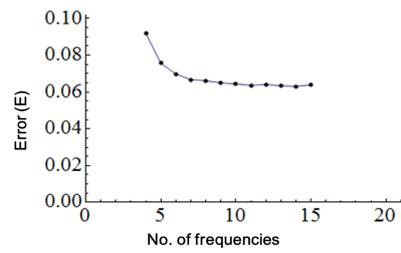


Figure 3.30: Arbitrary view of grayscale reconstructed texture.

tial patterns without knowing point correspondence. The comparison with the classical reconstruction method shows that the proposed reconstruction method outperforms the classical method, since the proposed method can reconstruct objects accurately even if the image correspondence is wrong. In next chapter we will show how to expand this theory to color images.



(a) reconstruction error in shape



(b) reconstruction error in intensity

Figure 3.31: Stability of grayscale texture tensor \mathcal{T}_{fij} . Horizontal axes show no. of point (in frequency domain) used for tensor computation. Vertical axes show reconstruction error in (a) shape and (b) intensity.

Chapter 4

Reconstructing Color Objects without knowing Image Correspondences

The previous chapter showed the reconstruction of texture objects. But, it is restricted to the case of grayscale sequences and textures. In this chapter we consider the case of reconstructing color sequential and texture patterns.

4.1 Object and Image Spaces

4.1.1 6D World

For color objects we consider that the world is 6 dimensional as shown in Fig. 4.1 (a). 3 dimensions are for geometric information and 3 for photometric information. Geometric information consists of X , Y and Z axes. Photometric information comprises of red, green and blue color channels. Equation (4.1) shows the homogeneous coordinates of a

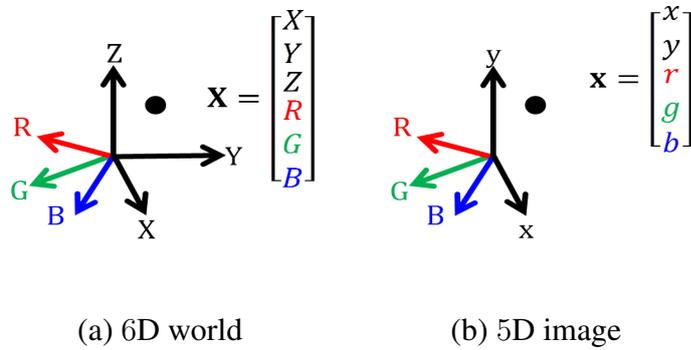


Figure 4.1: 6D world and 5D image. Three dimensions are for geometric information and three for color information in 6D. Two dimensions are for geometric and three for color information in 5D image.

6D point \mathbf{X} .

$$\mathbf{X} = \begin{bmatrix} X \\ Y \\ Z \\ R \\ G \\ B \\ 1 \end{bmatrix} \quad (4.1)$$

4.1.2 5D Image

The Dimensionality of image space is also increased to 5 as shown in Fig. 4.1. Two dimensions are for image geometric information (x, y) and three dimensions are for red, green and blue color channels. Equation (4.2) shows the homogeneous coordinates

of a 5D point \mathbf{x} .

$$\mathbf{x} = \begin{bmatrix} x \\ y \\ r \\ g \\ b \\ 1 \end{bmatrix} \quad (4.2)$$

In tensor notation the points are represented with contravariant indices. In spatial domain, a 6D point and a 5D point have representations X^i and x^i respectively. Further details on tensor notation can be found in Appendix A.4.

4.1.3 Color Sequential Patterns

In 6D space multiple points $\mathbf{X}(k)$ ($k = 1, \dots, N$) are connected in a curve to form 1D sequential patterns. In image 5D points $\mathbf{x}(k)$ ($k = 1, \dots, N$) join to form a sequential pattern. Sampling of color sequences is same as described in Sec. 3.1.3.

4.1.4 Color Texture Patterns

For color textures 6D points $\mathbf{X}(k, l)$ ($k = 1, \dots, N, l = 1, \dots, M$) are connected in a 2D grid to form a color texture. Image textures join 5D points $\mathbf{x}(k, l)$ ($k = 1, \dots, N, l = 1, \dots, M$) horizontally and vertically to form a grid. Sampling concept is same as explained in Sec. 3.1.4. Fig. 3.3 gives the idea of sampling. The difference is the photometric information is 3 dimensional in this case.

4.2 Affine camera from 6D to 5D space

We next consider an affine camera which projects sequential patterns defined in 6D space into 5D space. We call it 6D-5D affine camera. In this chapter we consider two types of affine camera matrices. Equation (4.3) represents a simple case when there is no color channel crosstalk, thus camera matrix and resulting tensor have fewer DOF. Equation (4.4) shows the complex case when inter channel crosstalk exists, it may be

noticed that the camera matrix has more non-zero, components and it will be shown in the next sections that the resulting tensor will also have greater DOF:

$$\begin{bmatrix} x \\ y \\ r \\ g \\ b \\ 1 \end{bmatrix} = \begin{bmatrix} p_{11} & p_{12} & p_{13} & 0 & 0 & 0 & p_{15} \\ p_{21} & p_{22} & p_{23} & 0 & 0 & 0 & p_{25} \\ 0 & 0 & 0 & p_{34} & 0 & 0 & p_{37} \\ 0 & 0 & 0 & 0 & p_{45} & 0 & p_{47} \\ 0 & 0 & 0 & 0 & 0 & p_{56} & p_{57} \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ R \\ G \\ B \\ 1 \end{bmatrix} \quad (4.3)$$

$$\begin{bmatrix} x \\ y \\ r \\ g \\ b \\ 1 \end{bmatrix} = \begin{bmatrix} p_{11} & p_{12} & p_{13} & 0 & 0 & 0 & p_{15} \\ p_{21} & p_{22} & p_{23} & 0 & 0 & 0 & p_{25} \\ 0 & 0 & 0 & p_{34} & p_{35} & p_{36} & p_{37} \\ 0 & 0 & 0 & p_{44} & p_{45} & p_{46} & p_{47} \\ 0 & 0 & 0 & p_{54} & p_{55} & p_{56} & p_{57} \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ R \\ G \\ B \\ 1 \end{bmatrix} \quad (4.4)$$

The matrix part in red denotes rotation and the blue one represents translation. Color is projected using a 3×3 transformation matrix shown in green, where diagonal components p_{34} , p_{45} and p_{56} are gain in red, green and blue color channels, p_{37} , p_{47} and p_{57} are offset in each color channel. Components p_{35} , p_{36} , p_{44} , p_{46} , p_{54} and p_{55} are crosstalk among red, green and blue color channels. For discussion on gain and offset kindly refer Appendix B.2. Equation (4.3) and (4.4), shows the extended camera projection model can capture both geometric information and photometric (RGB) information of the scene. The reader may notice that geometric and photometric information is independent of each other. The projection is dealt independently although it is simultaneous. A general projection of 6D to 5D camera is shown in Fig. 4.2.

Center of projection (COP) in 6D to 5D projection is a tensor of order $6 - 5 = 1$, i.e. it is a 6D point. COP is usually the null space of camera matrix (6×7 in this case). The line of sight (LOS) is the join/union of 2 higher dimensional points (tensor of step 1).

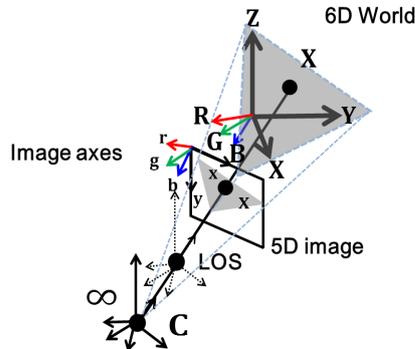


Figure 4.2: Camera projection for 6D-5D affine camera. The line joining the camera center C and image point x is termed as line of sight (LOS). The figure shows the camera center is at infinity for affine cameras.

4.3 Projection in Frequency Domain

We next consider the 6D-5D camera projection in frequency domain. Discussion about linearity of projection in Sec. 3.3 is also valid here.

4.3.1 Projection in Frequency Domain for Sequential Patterns (1D-DFT)

Suppose two cameras C and C' and projection of a series of 6D points $\mathbf{X}(k)$ ($k = 1, \dots, N$). Assume the point correspondences in these two cameras are unknown but the order of the points is maintained. Thus, a k th 6D point $\mathbf{X}(k)$ is observed as k th 5D image point $\mathbf{x}(k)$ in image 1 and is observed as k' th 5D image point $\mathbf{x}'(k')$ in image 2 as follows:

$$\mathbf{x}(k) = \mathbf{P}\mathbf{X}(k) \quad (4.5)$$

$$\mathbf{x}'(k') = \mathbf{P}'\mathbf{X}(k) \quad (4.6)$$

k and k' are different in general. Since we assume each image preserves the point order, the following relationship holds:

$$k' = k + s \quad (4.7)$$

where, s is the shift in sampling of image 2 with respect to image 1. s is unknown but it constant. Since s is unknown, we do not know the correspondence between $\mathbf{x}(k)$ and $\mathbf{x}'(k')$. To solve this problem, we consider camera projection in the frequency space. Suppose we have N image points in image 1 and image 2. By applying discrete Fourier transform to both sides of Eq. (4.5), we have the following representation for image projection in frequency space:

$$\begin{aligned}\mathbf{z}(n) &= \frac{1}{N} \sum_{k=0}^{N-1} \mathbf{P}\mathbf{X}(k) e^{-\frac{j2\pi nk}{N}} \\ &= \mathbf{P}\mathbf{Z}(n)\end{aligned}\quad (4.8)$$

where, $\mathbf{z}(n) = [x_f(n), y_f(n), r_f(n), g_f(n), b_f(n), \delta(n)]^\top$ is an image point and $\mathbf{Z}(n) = [X_f(n), Y_f(n), Z_f(n), R_f(n), G_f(n), B_f(n), \delta(n)]^\top$ is a 6D object point in frequency space, which is represented in homogeneous coordinates. $\delta(n)$ is the delta function, whose value is equal to 1 for $n = 0$ and 0 for others. Similarly, we apply discrete Fourier transform to Eq. (4.6) as follows:

$$\begin{aligned}\mathbf{z}'(n) &= \frac{1}{N} \sum_{k'=0}^{N-1} \mathbf{P}'\mathbf{X}(k' - s) e^{-\frac{j2\pi nk'}{N}} \\ &= \mathbf{P}'\mathbf{Z}(n) e^{-\frac{j2\pi ns}{N}}\end{aligned}\quad (4.9)$$

where $\mathbf{z}'(n) = [x'_f(n), y'_f(n), r'_f(n), g'_f(n), b'_f(n), \delta(n)]^\top$ is a point in the second image in frequency space. We may rewrite Eq. (4.9) as follows:

$$\lambda(n)\mathbf{z}'(n) = \mathbf{P}'\mathbf{Z}(n)\quad (4.10)$$

where $\lambda(n) = e^{\frac{j2\pi ns}{N}}$ represents the phase shift of sampling in camera \mathbf{C}' relative to camera \mathbf{C} . Equation (4.8) and Eq. (4.10) show both cameras project the same 6D point $\mathbf{Z}(n)$, therefore it is possible to consider the point correspondence of image data in frequency domain, even if the relative sampling of images is shifted as in the spatial domain.

4.3.2 Projection in Frequency Domain for Texture Patterns (2D-DFT)

For color texture pattern we consider a 2D sampling grid on the texture patterns. The concept is same as shown in Fig. 3.3, the position of a point on texture is defined by k and l . Suppose two cameras \mathbf{C} and \mathbf{C}' project texture elements $\mathbf{X}(k, l)$ ($k = 1, \dots, N$ and $l = 1, \dots, M$). The correspondences in these two cameras are unknown. Thus, the (k, l) th 6D point $\mathbf{X}(k, l)$ is observed as (k, l) th 5D image point $\mathbf{x}(k, l)$ in image 1 and is observed as (k', l') th 5D image point $\mathbf{x}'(k', l')$ in image 2.

$$\mathbf{x}(k, l) = \mathbf{P}\mathbf{X}(k, l) \quad (4.11)$$

$$\mathbf{x}'(k', l') = \mathbf{P}'\mathbf{X}(k, l) \quad (4.12)$$

k, l and k', l' are generally different. Assuming the order of points is preserved, we have the following relationship:

$$k' = k + s \quad (4.13)$$

$$l' = l + t \quad (4.14)$$

where, s and t are the relative horizontal and vertical shifts in the sampling of image 2 with respect to the sampling of image 1. Thus, s and t are unknown but are constant. This means $\mathbf{x}(k, l)$ and $\mathbf{x}'(k', l')$ are non corresponding points. To find correspondence it is necessary to determine the shifts s and t . This is possible by considering projection in the frequency space. Suppose N image points in image 1 and image 2. Applying Fourier transform to Eq. (4.11) yields following equation:

$$\begin{aligned} \mathbf{z}(n, m) &= \frac{1}{N} \frac{1}{M} \sum_{k=0}^{N-1} \sum_{l=0}^{M-1} \mathbf{P}\mathbf{X}(k, l) e^{-\frac{j2\pi nk}{N}} e^{-\frac{j2\pi ml}{M}} \\ &= \mathbf{P}\mathbf{Z}(n, m) \end{aligned} \quad (4.15)$$

where, $\mathbf{z}(n, m) = [x_f(n, m), y_f(n, m), r_f(n, m), g_f(n, m), b_f(n, m), \delta(n, m)]^\top$ represents the image point in the frequency space, and $\mathbf{Z}(n, m) = [X_f(n, m), Y_f(n, m), Z_f(n, m), R_f(n, m), G_f(n, m), B_f(n, m), \delta(n, m)]^\top$ represents the 6D point in the frequency space. $\delta(n, m)$ is the delta function, and its value is 1 for $n = m = 0$ and 0

for others. Similarly, applying Fourier transform to Eq. (4.14):

$$\begin{aligned}\mathbf{z}'(n, m) &= \frac{1}{N} \frac{1}{M} \sum_{k'=0}^{N-1} \sum_{l'=0}^{M-1} \mathbf{P}' \mathbf{X}(k' - s, l' - t) e^{-\frac{j2\pi nk'}{N}} e^{-\frac{j2\pi ml'}{M}} \\ &= \mathbf{P}' \mathbf{Z}(n, m) e^{-\frac{j2\pi ns}{N}} e^{-\frac{j2\pi mt}{M}}\end{aligned}\quad (4.16)$$

where $\mathbf{z}'(n, m) = [x'_f(n, m), y'_f(n, m), r'_f(n, m), g'_f(n, m), b'_f(n, m), \delta(n, m)]^\top$ is a point in the image 2 in frequency space. Rewriting Eq. (4.16) as follows:

$$\lambda(n, m) \mathbf{z}'(n, m) = \mathbf{P}' \mathbf{Z}(n, m) \quad (4.17)$$

where $\lambda(n, m) = e^{\frac{j2\pi ns}{N}} e^{\frac{j2\pi mt}{M}}$ represents the 2D phase shift of sampling in camera C' relative to camera C . Equation (4.15) and Eq. (4.17) shows that both cameras project the same 6D point $\mathbf{Z}(n, m)$. Therefore, it is possible to consider the correspondence of image data even if the sampling order of two images is different.

4.4 Back Projection from 5D images

Usually a higher dimensional entity is projected to a lower dimensional entity by application of camera. Back projection involves projection of lower dimensional entity in image space to be back projected to a higher dimensional entity in world. For instance, a point in image to a line in world, or a line in image to a plane in world etc. In this case, it is projected from a lower dimensional hyperplane to a higher dimensional hyperplane. Equation (4.5) shows projection in frequency domain as follows:

$$\mathbf{z} = \mathbf{P} \mathbf{Z} \quad (4.18)$$

A hyperplane containing a point \mathbf{z} in 5D image can be described as follows:

$$\mathbf{s}^\top \mathbf{z} = 0 \quad (4.19)$$

Combining Eq. (4.18) and Eq. (4.19):

$$\mathbf{s}^\top \mathbf{P} \mathbf{Z} = 0 \quad (4.20)$$

A hyperplane \mathbf{S} in 6D space can be defined as $\mathbf{S} = \mathbf{P}^\top \mathbf{s}$. Then, we have:

$$\mathbf{S}^\top \mathbf{Z} = 0 \quad (4.21)$$

Equation (4.21) shows the hyperplane \mathbf{S} contains 6D point \mathbf{Z} . It also shows that back projection of a hyperplane in 5D image is a hyperplane in 6D image. The back projection of a hyperplane can be observed in Fig. 4.3 where hyperplanes are back projected from 2 cameras.

4.5 Multiview relations for Sequential Patterns

We next derive multiview relations in 6D space, which is important for reconstructing color sequential patterns without knowing point correspondences.

4.5.1 Geometric derivation

Multiview relations in 6D space can be considered as the intersection of 7 hyperplanes, as shown in Fig. 4.3. This is the extension of the existing multiview relations [1]. Let us consider two cameras in 6D frequency space and let a 6D point be projected to \mathbf{z} and \mathbf{z}' in the images of these cameras.

$$\mathbf{z} = \mathbf{P}\mathbf{Z} \quad (4.22)$$

$$\lambda \mathbf{z}' = \mathbf{P}'\mathbf{Z} \quad (4.23)$$

where λ represents the phase shift of sampling between two cameras. Suppose we have 5 planes $\mathbf{s}^1, \mathbf{s}^2, \mathbf{s}^3, \mathbf{s}^4$ and \mathbf{s}^5 which go through \mathbf{z} and two planes \mathbf{s}'^1 and \mathbf{s}'^2 , which go through \mathbf{z}' . Note, these planes are considered for deriving the multilinear constraints and it is not important to determine these planes. If we consider back projected hyperplanes $\mathbf{S}^1, \mathbf{S}^2, \mathbf{S}^3, \mathbf{S}^4$ and \mathbf{S}^5 , from $\mathbf{s}^1, \mathbf{s}^2, \mathbf{s}^3, \mathbf{s}^4$ and \mathbf{s}^5 and \mathbf{S}'^1 and \mathbf{S}'^2 from \mathbf{s}'^1 and \mathbf{s}'^2 , these 7 hyper planes intersect at \mathbf{Z} in 6D frequency space as shown in Fig. 4.3.

$$\mathbf{S}^1 = \mathbf{P}^\top \mathbf{s}^1 \quad (4.24)$$

$$\mathbf{S}^2 = \mathbf{P}^\top \mathbf{s}^2 \quad (4.25)$$

$$\mathbf{S}^3 = \mathbf{P}^\top \mathbf{s}^3 \quad (4.26)$$

$$\mathbf{S}^4 = \mathbf{P}^\top \mathbf{s}^4 \quad (4.27)$$

$$\mathbf{S}^5 = \mathbf{P}^\top \mathbf{s}^5 \quad (4.28)$$

$$\mathbf{S}'^1 = \mathbf{P}'^\top \mathbf{s}'^1 \quad (4.29)$$

$$\mathbf{S}'^2 = \mathbf{P}'^\top \mathbf{s}'^2 \quad (4.30)$$

The condition for these seven hyperplanes to meet at a single point in 6D space is considered as the determinant of the matrix of hyperplanes vanishes as follows:

$$\det[\mathbf{P}^\top \mathbf{s}^1, \mathbf{P}^\top \mathbf{s}^2, \mathbf{P}^\top \mathbf{s}^3, \mathbf{P}^\top \mathbf{s}^4, \mathbf{P}^\top \mathbf{s}^5, \mathbf{P}'^\top \mathbf{s}'^1, \mathbf{P}'^\top \mathbf{s}'^2] = 0 \quad (4.31)$$

Equation (4.31) can be written in tensor format as follows:

$$\epsilon^{mnpqrs} s_a^1 P_m^a s_b^2 P_n^b s_c^3 P_o^c s_d^4 P_p^d s_e^5 P_q^e s'_f{}^1 P'_r{}^f s'_g{}^2 P'_s{}^g = 0 \quad (4.32)$$

The five planes $s_a^1, s_b^2, s_c^3, s_d^4$ and s_e^5 intersect at a point z^i in image 1, and planes $s'_f{}^1, s'_g{}^2, s'_h{}^3, s'_k{}^4$ and $s'_l{}^5$ intersect at a point z'^j in image 2 as follows:

$$\epsilon^{abcde} s_a^1 s_b^2 s_c^3 s_d^4 s_e^5 = z^i \quad (4.33)$$

$$\epsilon^{jklfg} s'_h{}^1 s'_k{}^2 s'_l{}^3 s'_f{}^4 s'_g{}^5 = z'^j \quad (4.34)$$

Thus, following equations hold:

$$s_a^1 s_b^2 s_c^3 s_d^4 s_e^5 = \epsilon_{abcde} z^i \quad (4.35)$$

$$s'_h{}^1 s'_k{}^2 s'_l{}^3 s'_f{}^4 s'_g{}^5 = \epsilon_{jklfg} z'^j \quad (4.36)$$

Substituting Eq. (4.35) and Eq. (4.36) into Eq. (4.32), we have the following relationship:

$$z^i z'^j \epsilon_{abcde} \epsilon_{jklfg} \epsilon^{mnpqrs} P_m^a P_n^b P_o^c P_p^d P_q^e P'_r{}^f P'_s{}^g = 0_{hkl} \quad (4.37)$$

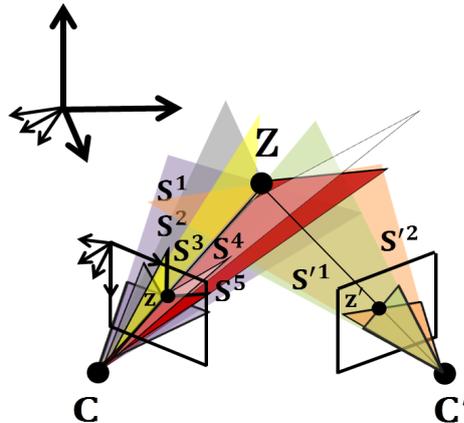


Figure 4.3: Intersection of seven hyperplanes at a 6D point Z . The hyperplanes S^1 , S^2 , S^3 , S^4 and S^5 are back projected from image 1. Hyperplanes S'^1 and S'^2 are back projected from image 2.

We define the bifocal tensor \mathcal{T}_{ijkl} as follows:

$$\mathcal{T}_{ijkl} = \epsilon_{iabcde} \epsilon_{jghklf} \epsilon^{mnopqrs} P_m^a P_n^b P_o^c P_p^d P_q^e P_r^f P_s^g \quad (4.38)$$

Then, the following bilinear constraint for 6D-5D camera is obtained in frequency space:

$$z^i z'^j \mathcal{T}_{ijkl} = 0_{hkl} \quad (4.39)$$

where $\mathbf{z} = [z^1, z^2, z^3, z^4, z^5, z^6]^\top$ and $\mathbf{z}' = [z'^1, z'^2, z'^3, z'^4, z'^5, z'^6]^\top$. ϵ_{iabcde} is a $6 \times 6 \times 6 \times 6 \times 6 \times 6$ tensor, which takes 1 for even permutation, -1 for odd permutation and 0 for others. Also, $\epsilon^{mnopqrs}$ is a $7 \times 7 \times 7 \times 7 \times 7 \times 7 \times 7$ tensor which takes -1 , 1 and 0 depending on its permutation. The **bifocal tensor** \mathcal{T}_{ijkl} is a $6 \times 6 \times 6 \times 6 \times 6$ tensor. Equation (4.39) shows for 1 point correspondence in frequency domain, a set of 216 equations is obtained (by varying the variables h, k and l) and each equation contains 36 terms. The details of \mathcal{T}_{ijkl} are postponed till Sec. 4.5.3.

In the case of 1D color sequential patterns, Eq. (4.39) can be specified as follows:

$$z^i(n)z'^j(n) \mathcal{T}_{ijhkl} = 0_{hkl} \quad (4.40)$$

For the case of 2D color sequential patterns, i.e. color texture patterns, the projection in frequency space is defined by Eq. (4.15) and Eq. (4.17), and Eq. (4.39) is modified to the following form:

$$z^i(n, m)z'^j(n, m) \mathcal{T}_{ijhkl} = 0_{hkl} \quad (4.41)$$

Equation (4.40) and Eq. (4.41) are two view relations in 6D space, which are essential for recovering 6D points from 5D images.

4.5.2 Algebraic verification

We next show, the algebraic verification of the bifocal tensor. Equation (4.3) or (4.4) can be used for computation of tensor. The only difference would be their DOF. Thus, in this section only Eq. (4.4) is considered. Let us consider the projection of two cameras in frequency domain, Eq. (4.8) and Eq. (4.10). These two equations can be described by a single matrix equation as follows:

$$\begin{bmatrix} \mathbf{P} & \mathbf{z} & \mathbf{0} \\ \mathbf{P}' & \mathbf{0} & \mathbf{z}' \end{bmatrix} \begin{bmatrix} \mathbf{Z} \\ -1 \\ -\lambda \end{bmatrix} = \mathbf{0} \quad (4.42)$$

Expanding Eq. (4.42), we obtain:

$$\begin{bmatrix}
 p_{11} & p_{12} & p_{13} & 0 & 0 & 0 & p_{17} & z^1 & 0 \\
 p_{21} & p_{22} & p_{23} & 0 & 0 & 0 & p_{27} & z^2 & 0 \\
 0 & 0 & 0 & p_{34} & p_{35} & p_{36} & p_{37} & z^3 & 0 \\
 0 & 0 & 0 & p_{44} & p_{45} & p_{46} & p_{47} & z^4 & 0 \\
 0 & 0 & 0 & p_{54} & p_{55} & p_{56} & p_{57} & z^5 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 1 & z^6 & 0 \\
 \hline
 p'_{11} & p'_{12} & p'_{13} & 0 & 0 & 0 & p'_{17} & 0 & z'^1 \\
 p'_{21} & p'_{22} & p'_{23} & 0 & 0 & 0 & p'_{27} & 0 & z'^2 \\
 0 & 0 & 0 & p'_{34} & p'_{35} & p'_{36} & p'_{37} & 0 & z'^3 \\
 0 & 0 & 0 & p'_{44} & p'_{45} & p'_{46} & p'_{47} & 0 & z'^4 \\
 0 & 0 & 0 & p'_{54} & p'_{55} & p'_{56} & p'_{57} & 0 & z'^5 \\
 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & z'^6
 \end{bmatrix}
 \begin{bmatrix}
 \mathbf{Z} \\
 -1 \\
 -\lambda
 \end{bmatrix}
 = \mathbf{0} \tag{4.43}$$

To form multiview relations eliminate rows until a square minor is obtained. In this case, a 7x7 minor has a zero determinant. Let us name the rows of camera matrices. The 6 row vectors of camera 1 are \mathbf{p}^a , \mathbf{p}^b , \mathbf{p}^c , \mathbf{p}^d , \mathbf{p}^e and \mathbf{p}^i . The six row vectors of camera 2 will be referred to as \mathbf{p}'^f , \mathbf{p}'^g , \mathbf{p}'^h , \mathbf{p}'^k , \mathbf{p}'^l and \mathbf{p}'^j . Then, cofactor expansion of determinant gives following bilinear constraint:

$$z^i z'^j \epsilon_{iabcde} \epsilon_{jhklfg} \det \begin{bmatrix} \mathbf{p}^a \\ \mathbf{p}^b \\ \mathbf{p}^c \\ \mathbf{p}^d \\ \mathbf{p}^e \\ \mathbf{p}'^f \\ \mathbf{p}'^g \end{bmatrix} = 0_{hkl} \tag{4.44}$$

The tensor \mathcal{T}_{ijkl} is defined as follows:

$$\mathcal{T}_{ijkl} = \epsilon_{abcde} \epsilon_{jhklfg} \det \begin{bmatrix} \mathbf{p}^a \\ \mathbf{p}^b \\ \mathbf{p}^c \\ \mathbf{p}^d \\ \mathbf{p}^e \\ \mathbf{p}'^f \\ \mathbf{p}'^g \end{bmatrix} \quad (4.45)$$

From Eq. (4.44) and Eq. (4.45), we have the bilinear constraints shown in Eq. (4.39). A part of tensor \mathcal{T}_{66345} is shown as an example in Eq. (4.43), where the eliminated matrix part is shown in blue.

4.5.3 Bifocal Tensor

In this section, we consider the detail of the bifocal tensor for 6D-5D cameras. It represents the relative geometric and photometric relationship between the two cameras which exist in the 6D space. Equation (4.39) shows that the tensor \mathcal{T}_{ijkl} can be computed from corresponding points in frequency space \mathbf{z} and \mathbf{z}' . Camera matrices and Eq. (4.38) exhibit features of tensor structure. Because of the structure of affine cameras many zero elements exist in the bifocal tensor. Equation (4.38) also shows that each element will be repeated $24(= 4!)$ times (12 times positively and negatively) owing to the presence of 4 variables of tensor \mathcal{T}_{ijkl} present in one ϵ term. Depending on the structure of camera matrices shown in Eq. (4.3) and (4.4), the resulting tensors have different DOF. Thus, the minimum no. of frequencies required for its computation is also different. However, the maximum no. of cameras and reconstruction procedure remains the same.

The bifocal tensor defined in Eq. (4.38) has 2 types depending upon which projection model in Eq. (4.3) and Eq. (4.4) is used.

Bifocal Tensor - I

Considering the bifocal tensor of camera matrices defined in Eq. (4.3), \mathcal{T}_{ijhkl} has 7776 total elements and 408 components are non-zero. This tensor would termed as **bifocal tensor - I** or **tensor - I**.

The tensor \mathcal{T}_{ijhkl} , contains 408 non-zero components. However, 17 of them are unique and 16 independent. Each element is repeated 12 times with same polarity and 12 times with reversed polarity such that $17 \times (12 \times 2) = 408$. The unique elements are $\mathcal{T}_{13456}, \mathcal{T}_{23456}, \mathcal{T}_{31456}, \mathcal{T}_{32456}, \mathcal{T}_{41356}, \mathcal{T}_{42356}, \mathcal{T}_{51346}, \mathcal{T}_{52346}, \mathcal{T}_{61345}, \mathcal{T}_{61346}, \mathcal{T}_{61356}, \mathcal{T}_{61456}, \mathcal{T}_{62345}, \mathcal{T}_{62356}, \mathcal{T}_{62456}, \mathcal{T}_{62346}$ and \mathcal{T}_{63456} . A complete list of 408 non-zero components of tensor \mathcal{T}_{ijhkl} classified into 17 unique component groups is given in Table 4.1.

Table 4.1: List of all \mathcal{T}_{ijhkl} non-zero elements (tensor - I).

Group No.	Polarity	Element List
1	+	$\mathcal{T}_{13456}, \mathcal{T}_{13564}, \mathcal{T}_{13645}, \mathcal{T}_{14365}, \mathcal{T}_{14536}, \mathcal{T}_{14653},$ $\mathcal{T}_{15346}, \mathcal{T}_{15463}, \mathcal{T}_{15634}, \mathcal{T}_{16354}, \mathcal{T}_{16435}, \mathcal{T}_{16543}$
	-	$\mathcal{T}_{13465}, \mathcal{T}_{13546}, \mathcal{T}_{13654}, \mathcal{T}_{14356}, \mathcal{T}_{14563}, \mathcal{T}_{14635},$ $\mathcal{T}_{15364}, \mathcal{T}_{15436}, \mathcal{T}_{15643}, \mathcal{T}_{16345}, \mathcal{T}_{16453}, \mathcal{T}_{16534}$
2	+	$\mathcal{T}_{23465}, \mathcal{T}_{23546}, \mathcal{T}_{23654}, \mathcal{T}_{24356}, \mathcal{T}_{24563}, \mathcal{T}_{24635},$ $\mathcal{T}_{25364}, \mathcal{T}_{25436}, \mathcal{T}_{25643}, \mathcal{T}_{26345}, \mathcal{T}_{26453}, \mathcal{T}_{26534}$
	-	$\mathcal{T}_{23456}, \mathcal{T}_{23564}, \mathcal{T}_{23645}, \mathcal{T}_{24365}, \mathcal{T}_{24536}, \mathcal{T}_{24653},$ $\mathcal{T}_{25346}, \mathcal{T}_{25463}, \mathcal{T}_{25634}, \mathcal{T}_{26354}, \mathcal{T}_{26435}, \mathcal{T}_{26543}$
3	+	$\mathcal{T}_{31456}, \mathcal{T}_{31564}, \mathcal{T}_{31645}, \mathcal{T}_{34165}, \mathcal{T}_{34516}, \mathcal{T}_{34651},$ $\mathcal{T}_{35146}, \mathcal{T}_{35461}, \mathcal{T}_{35614}, \mathcal{T}_{36154}, \mathcal{T}_{36415}, \mathcal{T}_{36541}$

cont.

$$\begin{aligned}
& - \mathcal{T}_{31465}, \mathcal{T}_{31546}, \mathcal{T}_{31654}, \mathcal{T}_{34156}, \mathcal{T}_{34561}, \mathcal{T}_{34615}, \\
& \quad \mathcal{T}_{35164}, \mathcal{T}_{35416}, \mathcal{T}_{35641}, \mathcal{T}_{36145}, \mathcal{T}_{36451}, \mathcal{T}_{36514} \\
4 \quad & + \mathcal{T}_{32465}, \mathcal{T}_{32546}, \mathcal{T}_{32654}, \mathcal{T}_{34256}, \mathcal{T}_{34562}, \mathcal{T}_{34625}, \\
& \quad \mathcal{T}_{35264}, \mathcal{T}_{35426}, \mathcal{T}_{35642}, \mathcal{T}_{36245}, \mathcal{T}_{36452}, \mathcal{T}_{36524} \\
& - \mathcal{T}_{32456}, \mathcal{T}_{32564}, \mathcal{T}_{32645}, \mathcal{T}_{34265}, \mathcal{T}_{34526}, \mathcal{T}_{34652}, \\
& \quad \mathcal{T}_{35246}, \mathcal{T}_{35462}, \mathcal{T}_{35624}, \mathcal{T}_{36254}, \mathcal{T}_{36425}, \mathcal{T}_{36542} \\
5 \quad & + \mathcal{T}_{41365}, \mathcal{T}_{41536}, \mathcal{T}_{41653}, \mathcal{T}_{43156}, \mathcal{T}_{43561}, \mathcal{T}_{43615}, \\
& \quad \mathcal{T}_{45163}, \mathcal{T}_{45316}, \mathcal{T}_{45631}, \mathcal{T}_{46135}, \mathcal{T}_{46351}, \mathcal{T}_{46513} \\
& - \mathcal{T}_{41356}, \mathcal{T}_{41563}, \mathcal{T}_{41635}, \mathcal{T}_{43165}, \mathcal{T}_{43516}, \mathcal{T}_{43651}, \\
& \quad \mathcal{T}_{45136}, \mathcal{T}_{45361}, \mathcal{T}_{45613}, \mathcal{T}_{46153}, \mathcal{T}_{46315}, \mathcal{T}_{46531} \\
6 \quad & + \mathcal{T}_{42356}, \mathcal{T}_{42563}, \mathcal{T}_{42635}, \mathcal{T}_{43265}, \mathcal{T}_{43526}, \mathcal{T}_{43652}, \\
& \quad \mathcal{T}_{45236}, \mathcal{T}_{45362}, \mathcal{T}_{45623}, \mathcal{T}_{46253}, \mathcal{T}_{46325}, \mathcal{T}_{46532} \\
& - \mathcal{T}_{42365}, \mathcal{T}_{42536}, \mathcal{T}_{42653}, \mathcal{T}_{43256}, \mathcal{T}_{43562}, \mathcal{T}_{43625}, \\
& \quad \mathcal{T}_{45263}, \mathcal{T}_{45326}, \mathcal{T}_{45632}, \mathcal{T}_{46235}, \mathcal{T}_{46352}, \mathcal{T}_{46523} \\
7 \quad & + \mathcal{T}_{51346}, \mathcal{T}_{51463}, \mathcal{T}_{51634}, \mathcal{T}_{53164}, \mathcal{T}_{53416}, \mathcal{T}_{53641}, \\
& \quad \mathcal{T}_{54136}, \mathcal{T}_{54361}, \mathcal{T}_{54613}, \mathcal{T}_{56143}, \mathcal{T}_{56314}, \mathcal{T}_{56431} \\
& - \mathcal{T}_{51364}, \mathcal{T}_{51436}, \mathcal{T}_{51643}, \mathcal{T}_{53146}, \mathcal{T}_{53461}, \mathcal{T}_{53614}, \\
& \quad \mathcal{T}_{54163}, \mathcal{T}_{54316}, \mathcal{T}_{54631}, \mathcal{T}_{56134}, \mathcal{T}_{56341}, \mathcal{T}_{56413} \\
8 \quad & + \mathcal{T}_{52364}, \mathcal{T}_{52436}, \mathcal{T}_{52643}, \mathcal{T}_{53246}, \mathcal{T}_{53462}, \mathcal{T}_{53624}, \\
& \quad \mathcal{T}_{54263}, \mathcal{T}_{54326}, \mathcal{T}_{54632}, \mathcal{T}_{56234}, \mathcal{T}_{56342}, \mathcal{T}_{56423}
\end{aligned}$$

cont.

$$\begin{aligned}
& - \mathcal{T}_{52346}, \mathcal{T}_{52463}, \mathcal{T}_{52634}, \mathcal{T}_{53264}, \mathcal{T}_{53426}, \mathcal{T}_{53642}, \\
& \quad \mathcal{T}_{54236}, \mathcal{T}_{54362}, \mathcal{T}_{54623}, \mathcal{T}_{56243}, \mathcal{T}_{56324}, \mathcal{T}_{56432} \\
9 \quad & + \mathcal{T}_{61354}, \mathcal{T}_{61435}, \mathcal{T}_{61543}, \mathcal{T}_{63145}, \mathcal{T}_{63451}, \mathcal{T}_{63514}, \\
& \quad \mathcal{T}_{64153}, \mathcal{T}_{64315}, \mathcal{T}_{64531}, \mathcal{T}_{65134}, \mathcal{T}_{65341}, \mathcal{T}_{65413} \\
& - \mathcal{T}_{61345}, \mathcal{T}_{61453}, \mathcal{T}_{61534}, \mathcal{T}_{63154}, \mathcal{T}_{63415}, \mathcal{T}_{63541}, \\
& \quad \mathcal{T}_{64135}, \mathcal{T}_{64351}, \mathcal{T}_{64513}, \mathcal{T}_{65143}, \mathcal{T}_{65314}, \mathcal{T}_{65431} \\
10 \quad & + \mathcal{T}_{61364}, \mathcal{T}_{61436}, \mathcal{T}_{61643}, \mathcal{T}_{63146}, \mathcal{T}_{63461}, \mathcal{T}_{63614}, \\
& \quad \mathcal{T}_{64163}, \mathcal{T}_{64316}, \mathcal{T}_{64631}, \mathcal{T}_{66134}, \mathcal{T}_{66341}, \mathcal{T}_{66413} \\
& - \mathcal{T}_{61346}, \mathcal{T}_{61463}, \mathcal{T}_{61634}, \mathcal{T}_{63164}, \mathcal{T}_{63416}, \mathcal{T}_{63641}, \\
& \quad \mathcal{T}_{64136}, \mathcal{T}_{64361}, \mathcal{T}_{64613}, \mathcal{T}_{66143}, \mathcal{T}_{66314}, \mathcal{T}_{66431} \\
11 \quad & + \mathcal{T}_{61365}, \mathcal{T}_{61536}, \mathcal{T}_{61653}, \mathcal{T}_{63156}, \mathcal{T}_{63561}, \mathcal{T}_{63615}, \\
& \quad \mathcal{T}_{65163}, \mathcal{T}_{65316}, \mathcal{T}_{65631}, \mathcal{T}_{66135}, \mathcal{T}_{66351}, \mathcal{T}_{66513} \\
& - \mathcal{T}_{61356}, \mathcal{T}_{61563}, \mathcal{T}_{61635}, \mathcal{T}_{63165}, \mathcal{T}_{63516}, \mathcal{T}_{63651}, \\
& \quad \mathcal{T}_{65136}, \mathcal{T}_{65361}, \mathcal{T}_{65613}, \mathcal{T}_{66153}, \mathcal{T}_{66315}, \mathcal{T}_{66531} \\
12 \quad & + \mathcal{T}_{61465}, \mathcal{T}_{61546}, \mathcal{T}_{61654}, \mathcal{T}_{64156}, \mathcal{T}_{64561}, \mathcal{T}_{64615}, \\
& \quad \mathcal{T}_{65164}, \mathcal{T}_{65416}, \mathcal{T}_{65641}, \mathcal{T}_{66145}, \mathcal{T}_{66451}, \mathcal{T}_{66514} \\
& - \mathcal{T}_{61456}, \mathcal{T}_{61564}, \mathcal{T}_{61645}, \mathcal{T}_{64165}, \mathcal{T}_{64516}, \mathcal{T}_{64651}, \\
& \quad \mathcal{T}_{65146}, \mathcal{T}_{65461}, \mathcal{T}_{65614}, \mathcal{T}_{66154}, \mathcal{T}_{66415}, \mathcal{T}_{66541} \\
13 \quad & + \mathcal{T}_{62345}, \mathcal{T}_{62453}, \mathcal{T}_{62534}, \mathcal{T}_{63254}, \mathcal{T}_{63425}, \mathcal{T}_{63542}, \\
& \quad \mathcal{T}_{64235}, \mathcal{T}_{64352}, \mathcal{T}_{64523}, \mathcal{T}_{65243}, \mathcal{T}_{65324}, \mathcal{T}_{65432} \\
& - \mathcal{T}_{62354}, \mathcal{T}_{62435}, \mathcal{T}_{62543}, \mathcal{T}_{63245}, \mathcal{T}_{63452}, \mathcal{T}_{63524},
\end{aligned}$$

cont.

$$\begin{aligned}
& \mathcal{T}_{64253}, \mathcal{T}_{64325}, \mathcal{T}_{64532}, \mathcal{T}_{65234}, \mathcal{T}_{65342}, \mathcal{T}_{65423} \\
14 \quad & + \quad \mathcal{T}_{62346}, \mathcal{T}_{62463}, \mathcal{T}_{62634}, \mathcal{T}_{63264}, \mathcal{T}_{63426}, \mathcal{T}_{63642}, \\
& \quad \mathcal{T}_{64236}, \mathcal{T}_{64362}, \mathcal{T}_{64623}, \mathcal{T}_{66243}, \mathcal{T}_{66324}, \mathcal{T}_{66432} \\
& - \quad \mathcal{T}_{62364}, \mathcal{T}_{62436}, \mathcal{T}_{62643}, \mathcal{T}_{63246}, \mathcal{T}_{63462}, \mathcal{T}_{63624}, \\
& \quad \mathcal{T}_{64263}, \mathcal{T}_{64326}, \mathcal{T}_{64632}, \mathcal{T}_{66234}, \mathcal{T}_{66342}, \mathcal{T}_{66423} \\
15 \quad & + \quad \mathcal{T}_{62356}, \mathcal{T}_{62563}, \mathcal{T}_{62635}, \mathcal{T}_{63265}, \mathcal{T}_{63526}, \mathcal{T}_{63652}, \\
& \quad \mathcal{T}_{65236}, \mathcal{T}_{65362}, \mathcal{T}_{65623}, \mathcal{T}_{66253}, \mathcal{T}_{66325}, \mathcal{T}_{66532} \\
& - \quad \mathcal{T}_{62365}, \mathcal{T}_{62536}, \mathcal{T}_{62653}, \mathcal{T}_{63256}, \mathcal{T}_{63562}, \mathcal{T}_{63625}, \\
& \quad \mathcal{T}_{65263}, \mathcal{T}_{65326}, \mathcal{T}_{65632}, \mathcal{T}_{66235}, \mathcal{T}_{66352}, \mathcal{T}_{66523} \\
16 \quad & + \quad \mathcal{T}_{62456}, \mathcal{T}_{62564}, \mathcal{T}_{62645}, \mathcal{T}_{64265}, \mathcal{T}_{64526}, \mathcal{T}_{64652}, \\
& \quad \mathcal{T}_{65246}, \mathcal{T}_{65462}, \mathcal{T}_{65624}, \mathcal{T}_{66254}, \mathcal{T}_{66425}, \mathcal{T}_{66542} \\
& - \quad \mathcal{T}_{62465}, \mathcal{T}_{62546}, \mathcal{T}_{62654}, \mathcal{T}_{64256}, \mathcal{T}_{64562}, \mathcal{T}_{64625}, \\
& \quad \mathcal{T}_{65264}, \mathcal{T}_{65426}, \mathcal{T}_{65642}, \mathcal{T}_{66245}, \mathcal{T}_{66452}, \mathcal{T}_{66524} \\
17 \quad & + \quad \mathcal{T}_{63456}, \mathcal{T}_{63564}, \mathcal{T}_{63645}, \mathcal{T}_{64365}, \mathcal{T}_{64536}, \mathcal{T}_{64653}, \\
& \quad \mathcal{T}_{65346}, \mathcal{T}_{65463}, \mathcal{T}_{65634}, \mathcal{T}_{66354}, \mathcal{T}_{66435}, \mathcal{T}_{66543} \\
& - \quad \mathcal{T}_{63465}, \mathcal{T}_{63546}, \mathcal{T}_{63654}, \mathcal{T}_{64356}, \mathcal{T}_{64563}, \mathcal{T}_{64635}, \\
& \quad \mathcal{T}_{65364}, \mathcal{T}_{65436}, \mathcal{T}_{65643}, \mathcal{T}_{66345}, \mathcal{T}_{66453}, \mathcal{T}_{66534}
\end{aligned}$$

Bifocal Tensor - II

Next, bifocal tensor \mathcal{T}_{ijhkl} of camera matrices defined in Eq. (4.4) will be dealt. It will be referred to as **bifocal tensor-II** or **tensor - II**. The total no. of elements remains same, i.e. 7776 elements. But the no. of non-zero elements increase to 696. There are

29 unique elements and 28 independent elements except a scale. Each element repeated 12 times positively and 12 times negatively. The unique elements are, \mathcal{T}_{13456} , \mathcal{T}_{23456} , \mathcal{T}_{31346} , \mathcal{T}_{31356} , \mathcal{T}_{31456} , \mathcal{T}_{32346} , \mathcal{T}_{32356} , \mathcal{T}_{32456} , \mathcal{T}_{41346} , \mathcal{T}_{41356} , \mathcal{T}_{41456} , \mathcal{T}_{42346} , \mathcal{T}_{42356} , \mathcal{T}_{42456} , \mathcal{T}_{51346} , \mathcal{T}_{51356} , \mathcal{T}_{51456} , \mathcal{T}_{52346} , \mathcal{T}_{52356} , \mathcal{T}_{52456} , \mathcal{T}_{61345} , \mathcal{T}_{61346} , \mathcal{T}_{61356} , \mathcal{T}_{61456} , \mathcal{T}_{62345} , \mathcal{T}_{62346} , \mathcal{T}_{62356} , \mathcal{T}_{62456} and \mathcal{T}_{63456} . In addition to the non-zero elements of Table 4.1, the list of 12 additional non-zero components of tensor \mathcal{T}_{ijkl} is given in Table 4.2.

Table 4.2: List of additional 12 non-zero \mathcal{T}_{ijkl} elements (tensor - II).

Group No.	Polarity	Element List
18	+	$\mathcal{T}_{31364}, \mathcal{T}_{31436}, \mathcal{T}_{31643}, \mathcal{T}_{33146}, \mathcal{T}_{33461}, \mathcal{T}_{33614},$ $\mathcal{T}_{34163}, \mathcal{T}_{34316}, \mathcal{T}_{34631}, \mathcal{T}_{36134}, \mathcal{T}_{36341}, \mathcal{T}_{36413}$
	-	$\mathcal{T}_{31346}, \mathcal{T}_{31463}, \mathcal{T}_{31634}, \mathcal{T}_{33164}, \mathcal{T}_{33416}, \mathcal{T}_{33641},$ $\mathcal{T}_{34136}, \mathcal{T}_{34361}, \mathcal{T}_{34613}, \mathcal{T}_{36143}, \mathcal{T}_{36314}, \mathcal{T}_{36431}$
19	+	$\mathcal{T}_{31365}, \mathcal{T}_{31536}, \mathcal{T}_{31653}, \mathcal{T}_{33156}, \mathcal{T}_{33561}, \mathcal{T}_{33615},$ $\mathcal{T}_{35163}, \mathcal{T}_{35316}, \mathcal{T}_{35631}, \mathcal{T}_{36135}, \mathcal{T}_{36351}, \mathcal{T}_{36513}$
	-	$\mathcal{T}_{31356}, \mathcal{T}_{31563}, \mathcal{T}_{31635}, \mathcal{T}_{33165}, \mathcal{T}_{33516}, \mathcal{T}_{33651},$ $\mathcal{T}_{35136}, \mathcal{T}_{35361}, \mathcal{T}_{35613}, \mathcal{T}_{36153}, \mathcal{T}_{36315}, \mathcal{T}_{36531}$
20	+	$\mathcal{T}_{32346}, \mathcal{T}_{32463}, \mathcal{T}_{32634}, \mathcal{T}_{33264}, \mathcal{T}_{33426}, \mathcal{T}_{33642},$ $\mathcal{T}_{34236}, \mathcal{T}_{34362}, \mathcal{T}_{34623}, \mathcal{T}_{36243}, \mathcal{T}_{36324}, \mathcal{T}_{36432}$
	-	$\mathcal{T}_{32364}, \mathcal{T}_{32436}, \mathcal{T}_{32643}, \mathcal{T}_{33246}, \mathcal{T}_{33462}, \mathcal{T}_{33624},$ $\mathcal{T}_{34263}, \mathcal{T}_{34326}, \mathcal{T}_{34632}, \mathcal{T}_{36234}, \mathcal{T}_{36342}, \mathcal{T}_{36423}$

cont.

$$\begin{aligned}
21 \quad & + \mathcal{T}_{32356}, \mathcal{T}_{32563}, \mathcal{T}_{32635}, \mathcal{T}_{33265}, \mathcal{T}_{33526}, \mathcal{T}_{33652}, \\
& \mathcal{T}_{35236}, \mathcal{T}_{35362}, \mathcal{T}_{35623}, \mathcal{T}_{36253}, \mathcal{T}_{36325}, \mathcal{T}_{36532} \\
& - \mathcal{T}_{32365}, \mathcal{T}_{32536}, \mathcal{T}_{32653}, \mathcal{T}_{33256}, \mathcal{T}_{33562}, \mathcal{T}_{33625}, \\
& \mathcal{T}_{35263}, \mathcal{T}_{35326}, \mathcal{T}_{35632}, \mathcal{T}_{36235}, \mathcal{T}_{36352}, \mathcal{T}_{36523} \\
22 \quad & + \mathcal{T}_{41364}, \mathcal{T}_{41436}, \mathcal{T}_{41643}, \mathcal{T}_{43146}, \mathcal{T}_{43461}, \mathcal{T}_{43614}, \\
& \mathcal{T}_{44163}, \mathcal{T}_{44316}, \mathcal{T}_{44631}, \mathcal{T}_{46134}, \mathcal{T}_{46341}, \mathcal{T}_{46413} \\
& - \mathcal{T}_{41346}, \mathcal{T}_{41463}, \mathcal{T}_{41634}, \mathcal{T}_{43164}, \mathcal{T}_{43416}, \mathcal{T}_{43641}, \\
& \mathcal{T}_{44136}, \mathcal{T}_{44361}, \mathcal{T}_{44613}, \mathcal{T}_{46143}, \mathcal{T}_{46314}, \mathcal{T}_{46431} \\
23 \quad & + \mathcal{T}_{41465}, \mathcal{T}_{41546}, \mathcal{T}_{41654}, \mathcal{T}_{44156}, \mathcal{T}_{44561}, \mathcal{T}_{44615}, \\
& \mathcal{T}_{45164}, \mathcal{T}_{45416}, \mathcal{T}_{45641}, \mathcal{T}_{46145}, \mathcal{T}_{46451}, \mathcal{T}_{46514} \\
& - \mathcal{T}_{41456}, \mathcal{T}_{41564}, \mathcal{T}_{41645}, \mathcal{T}_{44165}, \mathcal{T}_{44516}, \mathcal{T}_{44651}, \\
& \mathcal{T}_{45146}, \mathcal{T}_{45461}, \mathcal{T}_{45614}, \mathcal{T}_{46154}, \mathcal{T}_{46415}, \mathcal{T}_{46541} \\
24 \quad & + \mathcal{T}_{42346}, \mathcal{T}_{42463}, \mathcal{T}_{42634}, \mathcal{T}_{43264}, \mathcal{T}_{43426}, \mathcal{T}_{43642}, \\
& \mathcal{T}_{44236}, \mathcal{T}_{44362}, \mathcal{T}_{44623}, \mathcal{T}_{46243}, \mathcal{T}_{46324}, \mathcal{T}_{46432} \\
& - \mathcal{T}_{42364}, \mathcal{T}_{42436}, \mathcal{T}_{42643}, \mathcal{T}_{43246}, \mathcal{T}_{43462}, \mathcal{T}_{43624}, \\
& \mathcal{T}_{44263}, \mathcal{T}_{44326}, \mathcal{T}_{44632}, \mathcal{T}_{46234}, \mathcal{T}_{46342}, \mathcal{T}_{46423} \\
25 \quad & + \mathcal{T}_{42456}, \mathcal{T}_{42564}, \mathcal{T}_{42645}, \mathcal{T}_{44265}, \mathcal{T}_{44526}, \mathcal{T}_{44652}, \\
& \mathcal{T}_{45246}, \mathcal{T}_{45462}, \mathcal{T}_{45624}, \mathcal{T}_{46254}, \mathcal{T}_{46425}, \mathcal{T}_{46542} \\
& - \mathcal{T}_{42465}, \mathcal{T}_{42546}, \mathcal{T}_{42654}, \mathcal{T}_{44256}, \mathcal{T}_{44562}, \mathcal{T}_{44625}, \\
& \mathcal{T}_{45264}, \mathcal{T}_{45426}, \mathcal{T}_{45642}, \mathcal{T}_{46245}, \mathcal{T}_{46452}, \mathcal{T}_{46524} \\
26 \quad & + \mathcal{T}_{51365}, \mathcal{T}_{51536}, \mathcal{T}_{51653}, \mathcal{T}_{53156}, \mathcal{T}_{53561}, \mathcal{T}_{53615},
\end{aligned}$$

cont.

$$\begin{aligned}
& \mathcal{T}_{55163}, \mathcal{T}_{55316}, \mathcal{T}_{55631}, \mathcal{T}_{56135}, \mathcal{T}_{56351}, \mathcal{T}_{56513} \\
- & \mathcal{T}_{51356}, \mathcal{T}_{51563}, \mathcal{T}_{51635}, \mathcal{T}_{53165}, \mathcal{T}_{53516}, \mathcal{T}_{53651}, \\
& \mathcal{T}_{55136}, \mathcal{T}_{55361}, \mathcal{T}_{55613}, \mathcal{T}_{56153}, \mathcal{T}_{56315}, \mathcal{T}_{56531} \\
27 \quad + & \mathcal{T}_{51456}, \mathcal{T}_{51564}, \mathcal{T}_{51645}, \mathcal{T}_{54165}, \mathcal{T}_{54516}, \mathcal{T}_{54651}, \\
& \mathcal{T}_{55146}, \mathcal{T}_{55461}, \mathcal{T}_{55614}, \mathcal{T}_{56154}, \mathcal{T}_{56415}, \mathcal{T}_{56541} \\
- & \mathcal{T}_{51465}, \mathcal{T}_{51546}, \mathcal{T}_{51654}, \mathcal{T}_{54156}, \mathcal{T}_{54561}, \mathcal{T}_{54615}, \\
& \mathcal{T}_{55164}, \mathcal{T}_{55416}, \mathcal{T}_{55641}, \mathcal{T}_{56145}, \mathcal{T}_{56451}, \mathcal{T}_{56514} \\
28 \quad + & \mathcal{T}_{52356}, \mathcal{T}_{52563}, \mathcal{T}_{52635}, \mathcal{T}_{53265}, \mathcal{T}_{53526}, \mathcal{T}_{53652}, \\
& \mathcal{T}_{55236}, \mathcal{T}_{55362}, \mathcal{T}_{55623}, \mathcal{T}_{56253}, \mathcal{T}_{56325}, \mathcal{T}_{56532} \\
- & \mathcal{T}_{52365}, \mathcal{T}_{52536}, \mathcal{T}_{52653}, \mathcal{T}_{53256}, \mathcal{T}_{53562}, \mathcal{T}_{53625}, \\
& \mathcal{T}_{55263}, \mathcal{T}_{55326}, \mathcal{T}_{55632}, \mathcal{T}_{56235}, \mathcal{T}_{56352}, \mathcal{T}_{56523} \\
29 \quad + & \mathcal{T}_{52465}, \mathcal{T}_{52546}, \mathcal{T}_{52654}, \mathcal{T}_{54256}, \mathcal{T}_{54562}, \mathcal{T}_{54625}, \\
& \mathcal{T}_{55264}, \mathcal{T}_{55426}, \mathcal{T}_{55642}, \mathcal{T}_{56245}, \mathcal{T}_{56452}, \mathcal{T}_{56524} \\
- & \mathcal{T}_{52456}, \mathcal{T}_{52564}, \mathcal{T}_{52645}, \mathcal{T}_{54265}, \mathcal{T}_{54526}, \mathcal{T}_{54652}, \\
& \mathcal{T}_{55246}, \mathcal{T}_{55462}, \mathcal{T}_{55624}, \mathcal{T}_{56254}, \mathcal{T}_{56425}, \mathcal{T}_{56542}
\end{aligned}$$

4.5.4 Computation of Bifocal Tensor

The unique elements of bifocal tensor \mathcal{T}_{ijhkl} can be computed by using the relationship shown in Eq. (4.39). By modifying Eq. (4.39), we have:

$$\mathbf{MT} = \mathbf{0} \tag{4.46}$$

For computing bifocal tensor - II, \mathbf{M} and \mathbf{T} are as follows:

$$\mathbf{M} = \begin{bmatrix}
 z_1 z'_3 & z_2 z'_3 & z_3 z'_1 & z_3 z'_2 & z_4 z'_1 & z_4 z'_2 & z_5 z'_1 & z_5 z'_2 \\
 z_1 z'_5 & z_2 z'_5 & 0 & 0 & 0 & 0 & 0 & 0 \\
 -z_1 z'_4 & -z_2 z'_4 & 0 & 0 & 0 & 0 & 0 & 0 \\
 -z_1 z'_6 & -z_2 z'_6 & 0 & 0 & 0 & 0 & 0 & 0 \\
 \\
 z_6 z'_1 & z_6 z'_2 & z_6 z'_3 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & z_6 z'_5 & z_3 z'_1 & z_3 z'_2 & z_4 z'_1 & z_4 z'_2 & z_5 z'_1 \\
 0 & 0 & -z_6 z'_4 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & -z_6 z'_6 & 0 & 0 & 0 & 0 & 0 \\
 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 z_5 z'_2 & z_6 z'_1 & z_6 z'_2 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & z_3 z'_1 & z_3 z'_2 & z_4 z'_1 & z_4 z'_2 & z_5 z'_1 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 z_5 z'_2 & z_6 z'_1 & z_6 z'_2 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & z_6 z'_1 & z_6 z'_2 & 0 & 0 & 0
 \end{bmatrix} \quad (4.49)$$

$$\mathbf{T} = [\mathcal{T}_{13456}, \mathcal{T}_{23456}, \mathcal{T}_{31346}, \mathcal{T}_{31356}, \mathcal{T}_{31456}, \mathcal{T}_{32346}, \mathcal{T}_{32356}, \mathcal{T}_{32456}, \mathcal{T}_{41346}, \mathcal{T}_{41356}, \\
 \mathcal{T}_{41456}, \mathcal{T}_{42346}, \mathcal{T}_{42356}, \mathcal{T}_{42456}, \mathcal{T}_{51346}, \mathcal{T}_{51356}, \mathcal{T}_{51456}, \mathcal{T}_{52346}, \mathcal{T}_{52356}, \mathcal{T}_{52456}, \\
 \mathcal{T}_{61345}, \mathcal{T}_{61346}, \mathcal{T}_{61356}, \mathcal{T}_{61456}, \mathcal{T}_{62345}, \mathcal{T}_{62346}, \mathcal{T}_{62356}, \mathcal{T}_{62456}, \mathcal{T}_{63456}]^T \quad (4.50)$$

4.5.5 Independent Equations

The number of independent equations is same for both the tensors, tensor - I and tensor - II. Equation (4.39) shows 1 correspondence in frequency domain gives 216 equations, however only 4 of them are independent for $n = 0$ as shown in Sec. 4.5.4, and 3 are independent for $n \neq 0$, since at $n \neq 0$, $z^6 = z'^6 = 0$ in frequency space.

4.5.6 Minimum Number of Points for Tensor Computation

We next consider the minimum number of points required for computing the bifocal tensor.

Bifocal Tensor - I

The 17 unique (16 independent) elements of tensor - I, \mathcal{T}_{ijkl} can be computed from a minimum of 5 corresponding frequencies. In frequency domain 3 point correspondences at $n = 0$ provide $3 \times 4 = 12$ independent equations, and 2 correspondences at $n \neq 0$ provide $3 \times 2 = 6$ independent equations. Therefore, a total of 16 independent equations is sufficient to compute 16 independent components of tensor. It may be noticed that elements \mathcal{T}_{61345} , \mathcal{T}_{61346} , \mathcal{T}_{61356} , \mathcal{T}_{61456} , \mathcal{T}_{62345} , \mathcal{T}_{62356} , \mathcal{T}_{62456} , \mathcal{T}_{62346} and \mathcal{T}_{63456} can be computed from frequencies at $n = 0$ only. A discussion similar to Sec. 3.5.4.

The minimum no. of frequencies for computation of tensor - I for texture patterns (Eq. (4.41)) is same as for sequences (Eq. (4.40)). However, for proper reconstruction, the rank of matrix M in Eq. (4.47) should be 16. Therefore for repetitive textures, the choice of frequencies is slightly different. Two dimensional fourier transform of color information produces several zero components in absence of noise and components of negligible magnitude in presence of noise. Therefore, 3 frequencies must be chosen at $n = 0$ and two frequencies must be chosen at $n \neq 0$ for which the RGB frequency transformed components are non-zero.

Bifocal Tensor - II

The number of independent equations for tensor -II is same as that for tensor - I, but the DOF for tensor - II is increased to 28 (29 minus overall scale). The minimum no. of points for the tensor -II computation is 9. Three frequencies at $n = 0$ provide $3 \times 4 = 12$ constraints. Six frequencies for $n \neq 0$ give $6 \times 3 = 18$ constraints. Thus, a total of 30 equations provide the least no. of constraints required for tensor - II computation. However, it must be noted that elements \mathcal{T}_{61345} , \mathcal{T}_{61346} , \mathcal{T}_{61356} , \mathcal{T}_{61456} , \mathcal{T}_{62345} , \mathcal{T}_{62346} , \mathcal{T}_{62356} , \mathcal{T}_{62456} , \mathcal{T}_{63456} can be computed from correspondences at $n = 0$ only.

4.5.7 Maximum No. of Cameras

The maximum no. of cameras/images for 6D to 5D projection is 7, since the number of hyperplanes intersecting to form a multilinear constraint is 7. Let us denote the points in image 1, . . . , image 7 in frequency space as \mathbf{z} , \mathbf{z}' , \mathbf{z}'' , \mathbf{z}''' , \mathbf{z}^{IV} , \mathbf{z}^V , \mathbf{z}^{VI} and camera matrices as \mathbf{P} , \mathbf{P}' , \mathbf{P}'' , \mathbf{P}''' , \mathbf{P}^{IV} , \mathbf{P}^V and \mathbf{P}^{VI} .

The multilinear constraint for 3 cameras is:

$$z^i z'^j z''^k \epsilon_{ja_6 \dots a_{10}} \epsilon_{ka_{11} \dots a_{15}} C_i^{a_6 a_{11}} = 0_{a_7 \dots a_{10} a_{12} \dots a_{15}} \quad (4.51)$$

The multilinear constraint for 4 cameras is:

$$z^i z'^j z''^k z'''^l \epsilon_{ja_6 \dots a_{10}} \epsilon_{ka_{11} \dots a_{15}} \epsilon_{la_{16} \dots a_{20}} D_{ia_5}^{a_6 a_{11} a_{16}} = 0_{a_5 a_7 \dots a_{10} a_{12} \dots a_{15} a_{17} \dots a_{20}} \quad (4.52)$$

The multilinear constraint for 5 cameras is:

$$z^i z'^j z''^k z'''^l z^{IVx} \epsilon_{ja_6 \dots a_{10}} \epsilon_{ka_{11} \dots a_{15}} \epsilon_{la_{16} \dots a_{20}} \epsilon_{xa_{21} \dots a_{25}} \mathcal{E}_{ia_4 a_5}^{a_6 a_{11} a_{16} a_{21}} = 0_{a_4 a_5 a_7 \dots a_{10} a_{12} \dots a_{15} a_{17} \dots a_{20} a_{22} \dots a_{25}} \quad (4.53)$$

The multilinear constraint for 6 cameras is:

$$z^i z'^j z''^k z'''^l z^{IVx} z^{Vy} \epsilon_{ja_6 \dots a_{10}} \epsilon_{ka_{11} \dots a_{15}} \epsilon_{la_{16} \dots a_{20}} \epsilon_{ma_{21} \dots a_{25}} \epsilon_{ya_{26} \dots a_{30}} \mathcal{F}_{ia_3 a_4 a_5}^{a_6 a_{11} a_{16} a_{21} a_{26}} = 0_{a_3 a_4 a_5 a_7 \dots a_{10} a_{12} \dots a_{15} a_{17} \dots a_{20} a_{22} \dots a_{25} a_{27} \dots a_{30}} \quad (4.54)$$

The multilinear constraint for 7 cameras is:

$$z^i z'^j z''^k z'''^l z^{IVx} z^{Vy} z^{VIw} \epsilon_{ja_6 \dots a_{10}} \epsilon_{ka_{11} \dots a_{15}} \epsilon_{la_{16} \dots a_{20}} \epsilon_{ma_{21} \dots a_{25}} \epsilon_{ya_{26} \dots a_{30}} \epsilon_{wa_{31} \dots a_{35}} \mathcal{G}^{a_1 a_6 a_{11} a_{16} a_{21} a_{26} a_{31}} = 0_{a_2 \dots a_5 a_7 \dots a_{10} a_{12} \dots a_{15} a_{17} \dots a_{20} a_{27} \dots a_{30} a_{32} \dots a_{35}} \quad (4.55)$$

where $C_i^{a_6 a_{11}}$, $D_{ia_5}^{a_6 a_{11} a_{16}}$, $\mathcal{E}_{ia_4 a_5}^{a_6 a_{11} a_{16} a_{21}}$, $\mathcal{F}_{ia_3 a_4 a_5}^{a_6 a_{11} a_{16} a_{21} a_{26}}$ and $\mathcal{G}^{a_1 a_6 a_{11} a_{16} a_{21} a_{26} a_{31}}$ are defined as follows:

$$C_i^{a_6 a_{11}} = \epsilon^{mnopqrs} \epsilon_{ia_1 \dots a_5} P_m^{a_1} P_n^{a_2} P_o^{a_3} P_p^{a_4} P_q^{a_5} P_r^{a_6} P_s^{a_{11}} \quad (4.56)$$

$$D_{ia_5}^{a_6 a_{11} a_{16}} = \epsilon^{mnopqrs} \epsilon_{ia_1 \dots a_5} P_m^{a_1} P_n^{a_2} P_o^{a_3} P_p^{a_4} P_q^{a_6} P_r^{a_{11}} P_s^{a_{16}} \quad (4.57)$$

$$\mathcal{E}_{ia_4 a_5}^{a_6 a_{11} a_{16} a_{21}} = \epsilon^{mnopqrs} \epsilon_{ia_1 \dots a_5} P_m^{a_1} P_n^{a_2} P_o^{a_3} P_p^{a_6} P_q^{a_{11}} P_r^{a_{16}} P_s^{a_{21}} \quad (4.58)$$

$$\mathcal{F}_{ia_3a_4a_5}^{a_6a_{11}a_{16}a_{21}a_{26}} = \epsilon^{mnopqrs} \epsilon_{ia_1\dots a_5} P_m^{a_1} P_n^{a_2} P_o^{a_6} P_p^{a_{11}} P_q^{a_{16}} P_r^{IV a_{21}} P_s^{V a_{26}} \quad (4.59)$$

$$\mathcal{G}_{ia_3a_4a_5}^{a_6a_{11}a_{16}a_{21}a_{26}} = \epsilon^{mnopqrs} \epsilon_{ia_1\dots a_5} P_m^{a_1} P_n^{a_2} P_o^{a_6} P_p^{a_{11}} P_q^{IV a_{16}} P_r^{V a_{21}} P_s^{VI a_{26}} \quad (4.60)$$

The notation a_i, \dots, a_j (in subscripts and superscripts) means elements ranging from a_i to a_j . The definition of tensors $\epsilon^{mnopqrs}$ etc. can be seen in Appendix A.4.

4.6 Sampling shift

Sampling shifts for color sequence and texture patterns can be computed using the following methods.

4.6.1 Sampling shift for Sequential Patterns

We next consider the method of finding the sampling shift for sequential patterns. For sequential patterns the phase shift is 1 dimensional as shown in Eq. (4.7) and Eq. (4.10), as all points in sequence are connected in a single line. Considering upper 5x7 part of each camera matrix and upper 5x1 part of each image frequency \mathbf{z} and \mathbf{z}' in Eq. (4.8) and Eq. (4.10), the following system is obtained:

$$\begin{bmatrix} p_{11} & p_{12} & p_{13} & 0 & 0 & 0 & p_{17} & z^1 \\ p_{21} & p_{22} & p_{23} & 0 & 0 & 0 & p_{27} & z^2 \\ 0 & 0 & 0 & p_{34} & p_{35} & p_{36} & p_{37} & z^3 \\ 0 & 0 & 0 & p_{44} & p_{45} & p_{46} & p_{47} & z^4 \\ 0 & 0 & 0 & p_{54} & p_{55} & p_{56} & p_{57} & z^5 \\ p'_{11} & p'_{12} & p'_{13} & 0 & 0 & 0 & p'_{17} & e^{jn\theta} z'^1 \\ p'_{21} & p'_{22} & p'_{23} & 0 & 0 & 0 & p'_{27} & e^{jn\theta} z'^2 \\ 0 & 0 & 0 & p'_{34} & p'_{35} & p'_{36} & p'_{37} & e^{jn\theta} z'^3 \\ 0 & 0 & 0 & p'_{44} & p'_{45} & p'_{46} & p'_{47} & e^{jn\theta} z'^4 \\ 0 & 0 & 0 & p'_{54} & p'_{55} & p'_{56} & p'_{57} & e^{jn\theta} z'^5 \end{bmatrix} \begin{bmatrix} \mathbf{Z} \\ -1 \end{bmatrix} = \mathbf{0} \quad (4.61)$$

where $e^{jn\theta} = \lambda$ is the phase shift in sampling of image 2. Since Eq. (4.61) has solution, the determinant of the left most matrix \mathbf{M} must be zero. The matrix \mathbf{M} is non-square (10×8) and thus by solving a quadratic equation derived from the determinant of $\mathbf{M}^\top \mathbf{M}$, phase shift λ can be determined. The quadratic equation would give two solutions for

λ . In absence of noise, the roots of the equation will be equal. However, in the presence of noise the roots will be slightly different, so any one of them can be chosen. Taking the ratio of two consecutive frequencies, we have:

$$\frac{\lambda(n_2)}{\lambda(n_1)} = \frac{e^{jn_2\theta}}{e^{jn_1\theta}} = e^{j(n_2-n_1)\theta} \quad (4.62)$$

where $n_2 - n_1 = 1$ for consecutive frequencies. Thus, the phase shift s can be computed as follows:

$$s = \frac{N}{2\pi j(n_2 - n_1)} \log \frac{\lambda(n_2)}{\lambda(n_1)} \quad (4.63)$$

where $\lambda(n_2)$ and $\lambda(n_1)$ represent the phase at frequencies n_2 and n_1 respectively. In the presence of noise k will be a complex number. Therefore, we consider the real part and most repeated integer.

4.6.2 Sampling shift for Texture Patterns

The sampling shift is 2 dimensional in texture patterns, as shown in Eq. (4.13) and Eq. (4.14). Following the discussion in Sec. 3.6.2 taking the ratio of two horizontal consecutive frequencies to gives horizontal phase shift. Similarly ratio of two vertical consecutive frequencies gives vertical phase shift as follows:

$$s = \frac{N}{2\pi j(n_2 - n_1)} \log \frac{\lambda(n_2, m_1)}{\lambda(n_1, m_1)} \quad (4.64)$$

$$t = \frac{M}{2\pi j(m_2 - m_1)} \log \frac{\lambda(n_1, m_2)}{\lambda(n_1, m_1)} \quad (4.65)$$

where (n_2, m_1) and (n_1, m_1) are horizontally consecutive frequencies, and (n_1, m_2) and (n_1, m_1) are vertically consecutive frequencies respectively.

4.7 6D Reconstruction

We next consider reconstruction of 6D points from 5D images for recovering color sequential patterns. The method does not require point correspondences on the sequential

patterns.

4.7.1 Retrieval of Camera Matrices

Reconstruction of 6D point \mathbf{Z} requires knowledge of camera matrices. Assuming a canonical camera pair, the two camera matrices are defined as follows:

$$\mathbf{P} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.66)$$

$$\mathbf{P}' = [\mathbf{H}|\mathbf{e}'] \quad (4.67)$$

where \mathbf{H} represents the homography between images and \mathbf{e}' is the epipole in image 2. \mathbf{H} and \mathbf{e}' can be computed from \mathcal{T}_{ijhkl} as described in the following sections. Once the homography and the epipole are computed, camera matrices are recovered from Eq. (4.66) and Eq. (4.67), and the 6D frequency points $\mathbf{Z}(n)$ ($n = 1, \dots, N$) can be reconstructed.

4.7.2 Computation of Epipoles

Epipole \mathbf{e} in image 1, can be computed as the left null space of \mathcal{T}_{ijhkl} . Similarly, epipole \mathbf{e}' in image 2 can be computed as the right null space of tensor \mathcal{T}_{ijhkl} .

$$e^i \mathcal{T}_{ijhkl} = 0_{jhkl} \quad (4.68)$$

$$e'^j \mathcal{T}_{ijhkl} = 0_{ihkl} \quad (4.69)$$

Equation (4.69) can be described in matrix form as follows:

$$\mathbf{M}\mathbf{e}' = \mathbf{0} \quad (4.70)$$

where, $\mathbf{e}' = [e'^1, e'^2, e'^3, e'^4, e'^5, e'^6]^T$, and \mathbf{M} is a 1296 x 6 matrix, which consists of the components of tensor \mathcal{T}_{ijhkl} . Then, \mathbf{e}' can be obtained by solving Eq. (4.70). In the presence of noise, the least square solution of \mathbf{e}' can be computed as an eigenvector corresponding to the smallest eigenvalue of $\mathbf{M}^T \mathbf{M}$.

4.7.3 Extraction of Homography

Let us consider an arbitrary but fixed plane π , not passing through any of the camera centers. Then, a point \mathbf{z} in image 1 is related to a point \mathbf{z}'_π in image 2 by the homography \mathbf{H} as follows:

$$z'^s_\pi = H^s_i z^i \quad (4.71)$$

An epipolar line l' in image 2 can be defined as the join of epipole \mathbf{e}' and the point \mathbf{z}'_π as follows:

$$l'_{jhlk} = \epsilon_{jhlkrs} e'^r z'^s_\pi \quad (4.72)$$

From Eq. (4.39), epipolar line l' can also be defined as the corresponding line in image 2 for a point in image 1.

$$l'_{jhlk} = \mathcal{T}_{ijhkl} z^i \quad (4.73)$$

Equations (4.71), (4.72) and (4.73) show the following relationship for homography:

$$H^s_i = \epsilon^{jhlkrs} e'_r \mathcal{T}_{ijhkl} \quad (4.74)$$

Fig. 3.6 showed the concept induced homography for 3D images. The same concept is generalized to 5D images in this section.

Thus, both \mathbf{H} and \mathbf{e}' can be computed from \mathcal{T}_{ijhkl} , and camera matrices \mathbf{P} and \mathbf{P}' can be obtained from Eq. (4.66) and Eq. (4.67).

4.7.4 Affine Reconstruction and Rectification

Once the camera matrices \mathbf{P} and \mathbf{P}' have been computed, 6D points can be recovered as follows: Let us consider the camera projection equation in frequency space again:

$$\mathbf{z} = \mathbf{P}\mathbf{Z} \quad (4.75)$$

$$\mathbf{z}' = \mathbf{P}'\mathbf{Z} \quad (4.76)$$

Taking the vector product between the left side and the right side of Eq. (4.75) and Eq. (4.76), we have:

$$[\mathbf{z}]_{\times}\mathbf{P}\mathbf{Z} = 0 \quad (4.77)$$

$$[\mathbf{z}']_{\times}\mathbf{P}'\mathbf{Z} = 0 \quad (4.78)$$

where, $[\cdot]_{\times}$ denotes a skew symmetric matrix for vector product. From Eq. (4.77) and Eq. (4.78), we have:

$$\mathbf{M}\mathbf{Z} = \mathbf{0} \quad (4.79)$$

where, \mathbf{M} is a 250×7 matrix as follows:

$$\mathbf{M} = \begin{bmatrix} [\mathbf{z}]_{\times}\mathbf{P} \\ [\mathbf{z}']_{\times}\mathbf{P}' \end{bmatrix} \quad (4.80)$$

Then, by solving Eq. (4.79), 6D point \mathbf{Z} in the frequency space can be recovered. The least squares solution of Eq. (4.79) can be obtained by computing an eigenvector which corresponds to the minimum eigenvalue of $\mathbf{M}^{\top}\mathbf{M}$. After all points $\mathbf{Z}(n)$ ($n = 1, \dots, N$) are reconstructed using Eq. (4.79) in frequency space, we apply inverse Fourier transform to \mathbf{Z} to get original points $\mathbf{X}(i)$ ($i = 1, \dots, N$) in spatial domain, as $\mathbf{X} = \mathcal{F}^{-1}(\mathbf{Z})$. The two objects, reconstructed and the original one still differ by an affine homography. This homography has 42 degrees of freedom and can be computed by choosing seven points in general configuration for rectification of the reconstructed object. A case similar to Eq. (2.80).

The original sequential patterns $\hat{\mathbf{X}}$ can be obtained by rectifying the reconstructed

pattern \mathbf{X} by the homography \mathbf{H} as follows:

$$\hat{\mathbf{X}} = \mathbf{H}\mathbf{X} \quad (4.81)$$

4.8 Experiments

We next show the results of reconstructing 6D points, that is color sequential patterns and color textures. We show that the proposed method can reconstruct these objects without knowing point correspondence. We also show that our method can reconstruct photometric information as well as geometric shape information.

4.8.1 Color Sequential Pattern

First, we consider color sequences.

Synthetic Image Experiment

Figure 4.4 shows the sequential patterns used. The sequential patterns have 3D geometric information and 3D color information, thus termed "color". The objects are 6 dimensional. Two cameras with different geometric and sensitivity parameters project these sequences to image 1 and image 2. Figure 4.5 (a) and (b) show the 5 dimensional images. It may be observed that, image 1 was taken by a high gain camera, and image 2 was taken by a low gain camera. Also, the correspondence is unknown between images. Therefore, we assume that sampling of image 2 is shifted w.r.t sampling of image 1.

Using the method described in Sec. 4.5.4 tensor was computed. Three frequencies at $n = 0$, three frequencies at $n = 1$ and three frequencies at $n = 2$ were obtained from three textures were used for computing the tensor. The sampling of image 2 was found to be shifted w.r.t image 1 by 5 units. And appropriate phase values were multiplied to the reconstructed points \mathbf{Z} before applying inverse Fourier transform(Sec. 4.7).

Figure 4.6 (a) shows the ground truth and (b) illustrates the reconstructed object, while (c) shows the result from the classical method. The reconstruction from the proposed method is exactly same as the ground truth in terms of shape and color, which shows the method works well. It also depicts the fact that the method is independent to camera sensitivity differences.

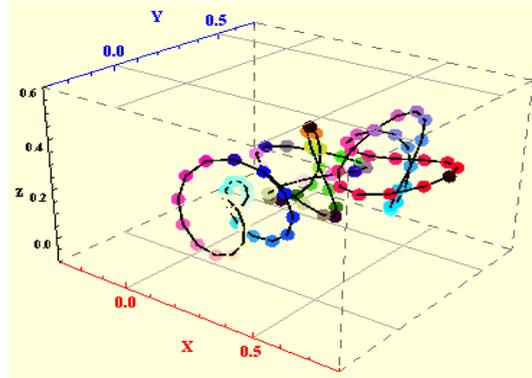


Figure 4.4: 6D color sequences used in synthetic experiment. There are three series of color points, which are connected by lines (object 1, object 2 and object 3).

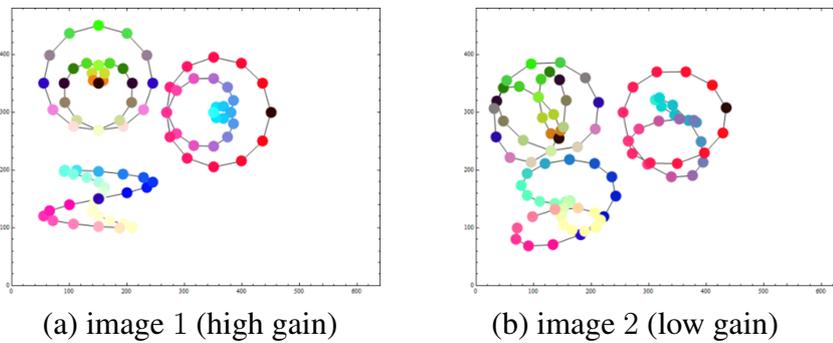


Figure 4.5: 5D Images of color sequences.

The last part is to synthesize arbitrary views of the reconstructed object. Figure 4.7 shows the arbitrary views. It can be observed that not only the viewpoints but the camera sensitivity parameters are also different for these images.

Nonetheless, the synthesized image is properly generated by using the proposed method. Next, we discuss reconstruction for three special cases:

1. Sequences are planar as shown in Fig. 4.8 (a).
2. Sequences lack color variation, two sequences have no color variation as shown in Fig. 4.9 (a).
3. Sequences are planar and lack color variation as shown in Fig. 4.10 (a).

Planar sequences can be reconstructed perfectly well. However, all sequences must not be on same plane. In this experiment, three correspondences at $n = 0$, three at $n = 1$

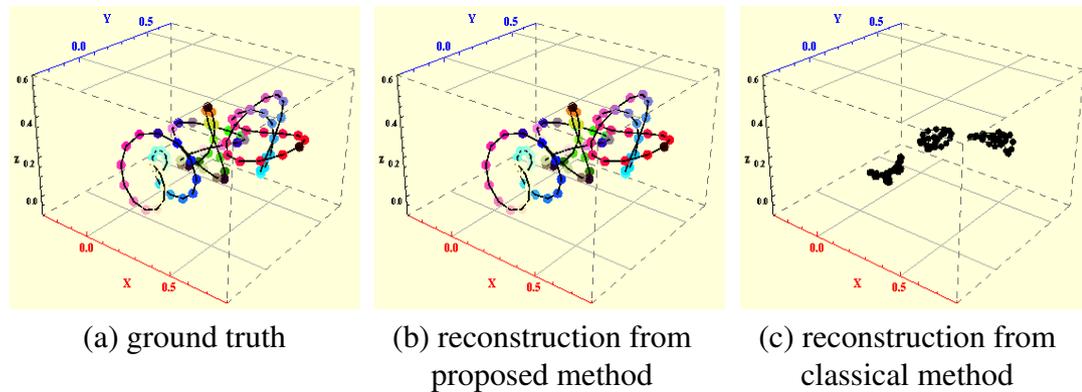


Figure 4.6: Original and reconstructed 6D object. The reconstruction has correct 3D shape and color (RGB) information.

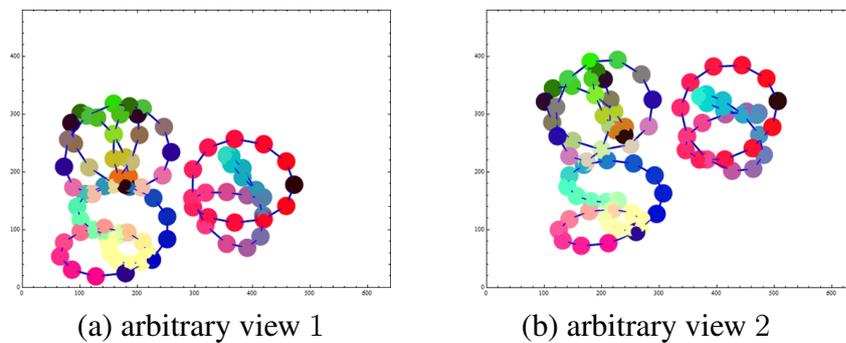


Figure 4.7: Arbitrary views of the 6D sequence reconstruction.

and three at $n = 2$ were used for tensor computation.

Next, if there is no color variation in all the sequences, reconstruction is not possible. This is because the rank of matrix M in Eq. (4.46) reaches at most 12, even if all the points are used. Therefore, at least one sequence must have color variation for determining the tensor as shown in Fig. 4.9 (b). Three points are chosen at $n = 0$ from each sequence, and other 6 points are chosen from the sequence having color variation.

Third, if the sequences are planar and lack color variation. Again, at least one sequence must have color variation. And correspondences are chosen in the similar way as for case above.

The experiments show the proposed method can reconstruct and produce the arbitrary views of the color sequences. These experiments were carried out in absence of image noise. Noise factor will be catered in stability analysis of tensor in Sec. 4.8.1.

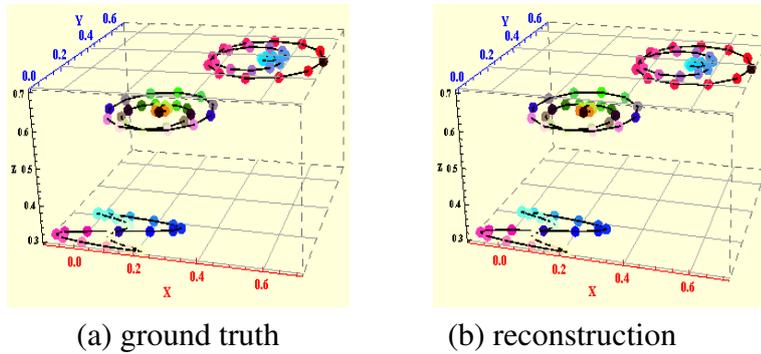


Figure 4.8: Planar 6D sequences.

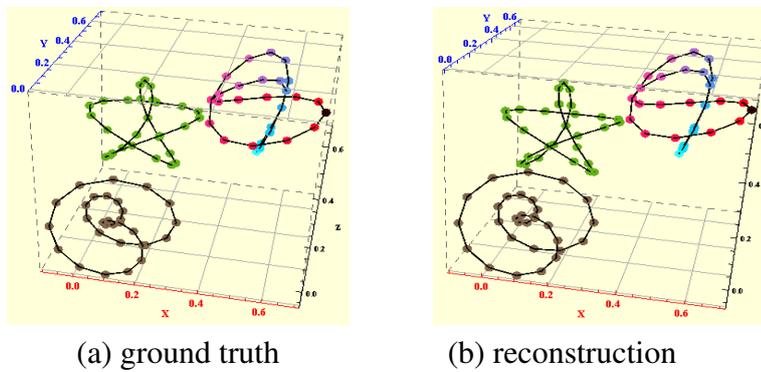


Figure 4.9: 6D sequences. Two sequences have no color variation.

Real Image Experiment

Figure 4.11 shows the images used for real image experiment. These images differ in viewpoint and camera gain. Concentric circles of blocks form the sequential patterns in this case. In this experiment, blobs are detected by using a method described in Appendix B.4, and their centroid and mean color are used for the coordinates of 5D image point Z . The correspondence of image points between image 1 and image 2 is not known.

From image 1 and image 2, tensor \mathcal{T}_{ijkl} and phase shift were estimated by using the proposed method. Three frequencies at $n = 0$, three frequencies at $n = 1$, three frequencies at $n = 2$ and three frequencies at $n = 3$ from three sequences were used for computing the tensor. The sampling of image 2 was found to be shifted by 5 units. After that, the proposed method was used to reconstruct the objects.

Figure 4.12 shows the comparison of (a) ground truth, (b) reconstruction from the

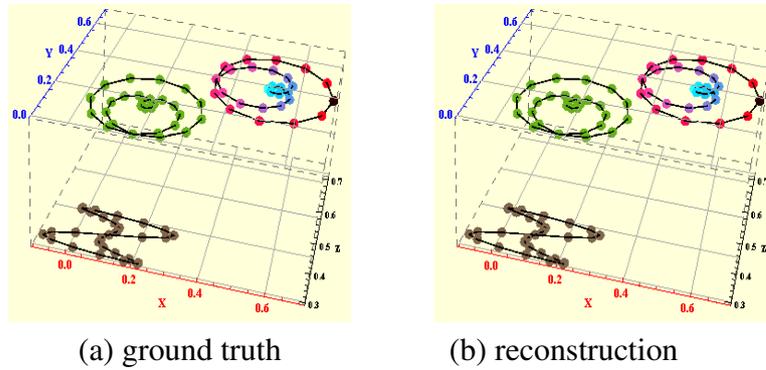


Figure 4.10: Planar 6D sequences. Two sequences have no color variation.

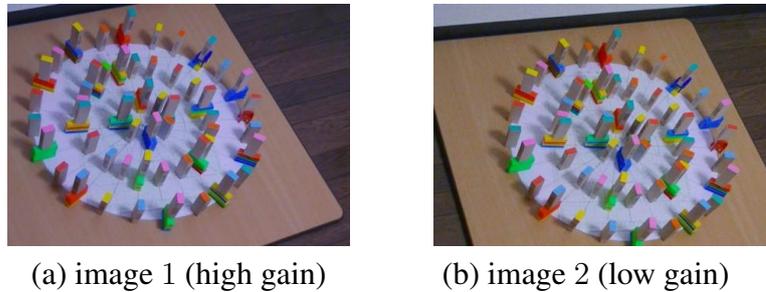


Figure 4.11: Images for real experiment. The color tops of blocks form the sequential patterns.

proposed method and (c) reconstruction from the classical method. As it may be seen, the reconstruction result of the proposed method is quite similar to the original object geometrically and photometrically, while the result from the classical method is completely wrong. This shows the proposed method works well for real images as well. It also shows the reconstruction of 3D shape and 3D color information is possible even if the sampling of images used is different. It also shows camera parameters need not to be same.

Next, the reconstruction was projected to arbitrary views as shown in Fig. 4.13, where arbitrary view 1 is high gain image. The original image points are joined using black lines and reprojected points are connected using red lines. As we may observe the difference is very minute. This shows that the arbitrary views are produced accurately. From above experiment we may state that the proposed method works for different kinds of color sequences and camera parameters.

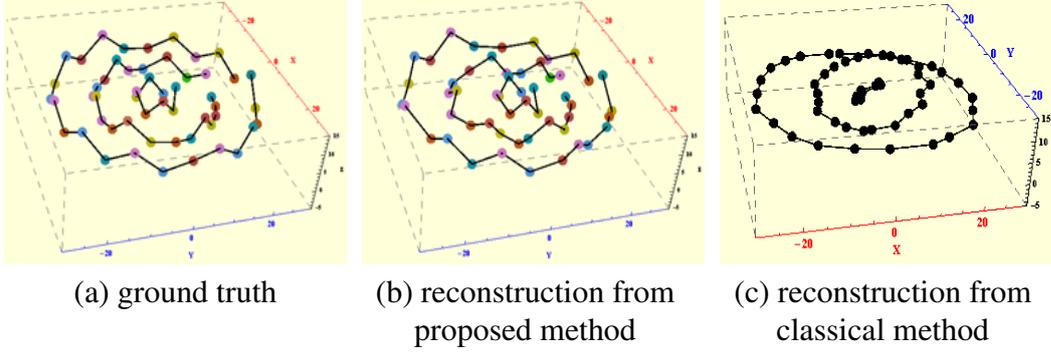


Figure 4.12: Original and reconstructed 6D object. The reconstruction has correct 3D shape and color values.

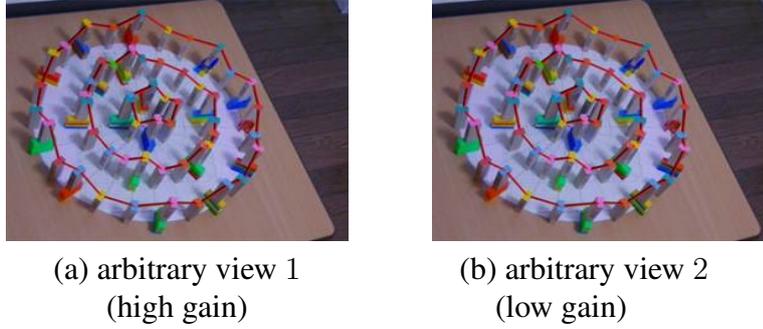


Figure 4.13: Arbitrary views of the 6D real color sequence. Black lines connect actual image points. Red lines connect the reprojected points.

Stability Evaluation

Next, we will look at its two important aspects of tensor \mathcal{T}_{ijkl} ; resistance against noise and increase in no. of frequencies for its computation. Reconstruction error Eq. (4.82) is used for gauging tensor stability.

$$E = \frac{1}{N} \sum_{i=1}^N d(\mathbf{Q}(i) - \hat{\mathbf{Q}}_r(i))^2 \quad (4.82)$$

where \mathbf{Q} and \mathbf{Q}_r represent the true quantity and reconstructed quantity. \mathbf{Q} is 3D geometric information for Fig. 4.14 (a) and it is 3D color information for Fig. 4.14 (b), N is the total no. of points present in sequences. The tensor was computed changing object shape and camera position. A random Gaussian noise of standard deviation of 1 was added to the image coordinates and color information. The result was averaged over

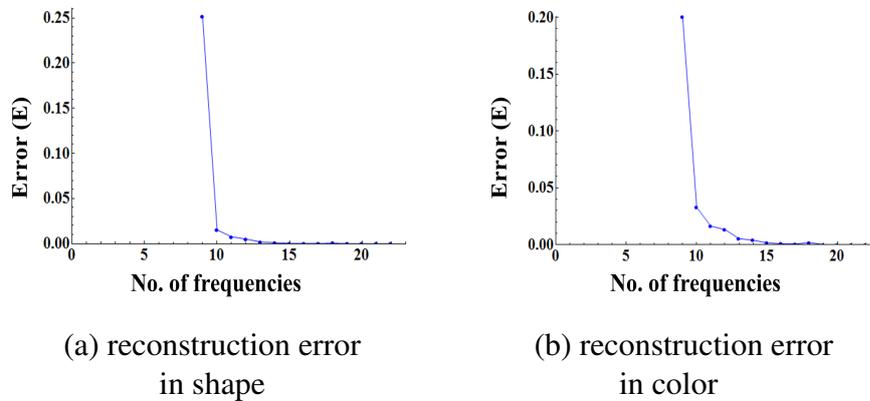


Figure 4.14: Stability of reconstruction. Horizontal axes show no. of frequencies for tensor computation. Vertical axes show reconstruction errors.

100 iterations. From the small magnitude of errors in the presence of noisy images and changing conditions we understand that the tensor is quite stable. Next, by increasing the no. of points in frequency domain for tensor computation, the reconstruction error is reduced and better reconstruction is obtained. This shows the stability of tensor is exponentially proportional to the no. of frequencies used. The graph also shows the fact that tensor is capable of handling different sequences under varying conditions.

4.8.2 Color Texture Patterns

This section will show 6D reconstruction and arbitrary view generation of color textures, i.e. 2D sequential patterns.

Synthetic Image Experiment

Figure 4.15 shows the synthetic texture used in this experiment. The textures have 3D geometric shape and 3D repetitive color information. It is easy to notice, the texture elements are connected vertically and horizontally to form a 2D sampling grid. The following experiments consider bifocal tensor - II for reconstruction. Two cameras at different locations and with different camera sensitivity values give image 1 and image 2, as shown in Fig. 4.16. Image 1 is a high gain image, and image 2 is a low gain image. To show the correspondence freeness of the method, image 2 was given a sampling shift. The horizontal sampling shift was 2 and the vertical sampling shift was 3. The

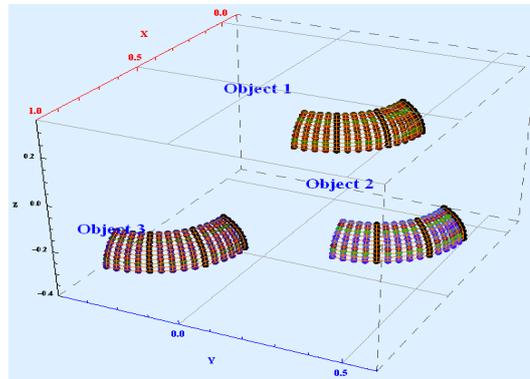


Figure 4.15: Color textures for synthetic experiment. The objects are 6 dimensional having 3D shape and repetitive 3D color information.

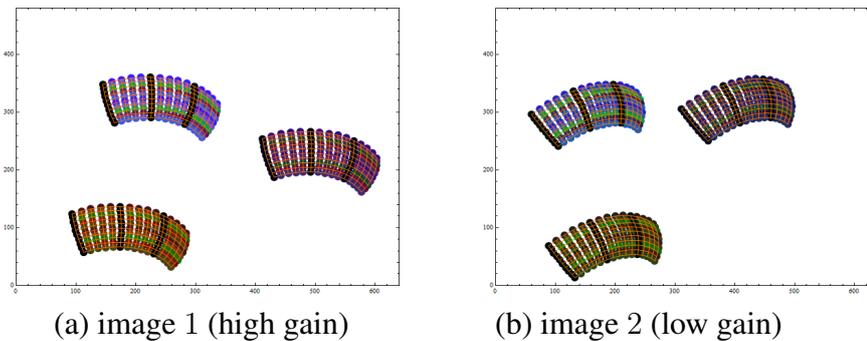


Figure 4.16: 5D images of color texture objects.

images were then Fourier transformed. Tensor \mathcal{T}_{ijkl} was computed for these images, since the color is repetitive 5 correspondences were chosen in frequency space with enough variation. Three correspondences were chosen at $n = 0$ and other two at $n = 4$. Figure 4.17 shows the ground truth with reconstruction results from the proposed method and the classical method. The result from the proposed method is identical to the ground truth both in shape and texture colors, highlighting that reconstruction of color textures is accurate, even in the absence of exact correspondences. It also shows that our method is free of camera sensitivity parameters and sampling differences. To show the reconstructed textures from different viewpoints with different camera parameters, arbitrary views Fig. 4.18 were generated. These images differ in viewpoint and color information from the actual images Fig. 4.16 were used. Next, we discuss reconstruction in the following three special cases:

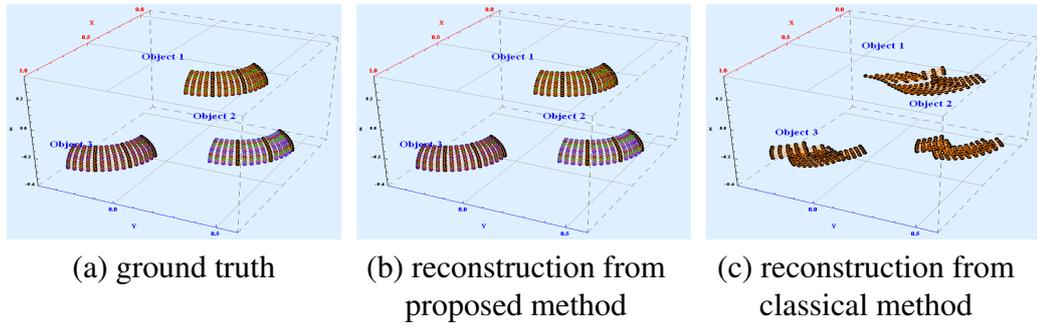


Figure 4.17: Original and reconstructed 6D texture objects. The reconstructed color texture has correct 3D shape and color values.

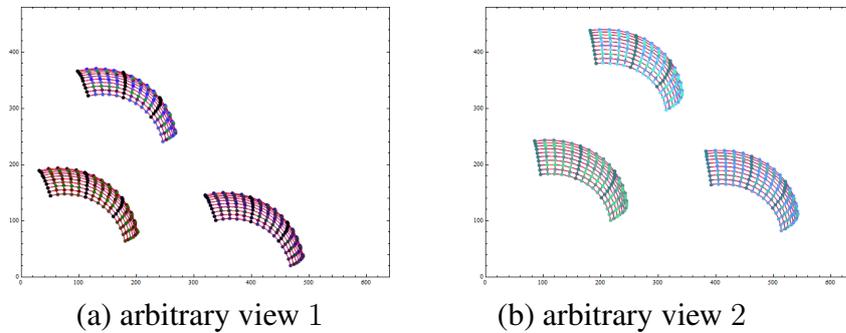


Figure 4.18: Arbitrary views of 6D reconstructed texture.

1. Textures are planar as shown in Fig. 4.19 (a). Reconstruction is possible with 9 correspondences. Three at $n = 0$ and 6 at $n \neq 0$. However, the three texture patterns should not be at same plane.
2. Textures lack color variation as shown in Fig. 4.20 (a). To meet the rank criteria of matrix M in Eq. (4.49), the color should vary for at least 1 object.
3. Textures are planar and also lack color variation as shown in Fig. 4.21 (a).

Next, we study the possibility of mixed objects, when both 1D sequences and 2D textures are present in the scene. Sequences are considered using 1D Fourier transform and textures using 2D Fourier transform. Reconstruction is similar (1D-DFT and 2D-DFT) and tensor requires a minimum of 9 correspondences as shown in Fig. 4.22. The 9 correspondences can be divided into two groups, 3 at $n = 0$ and 6 at $n \neq 0$.

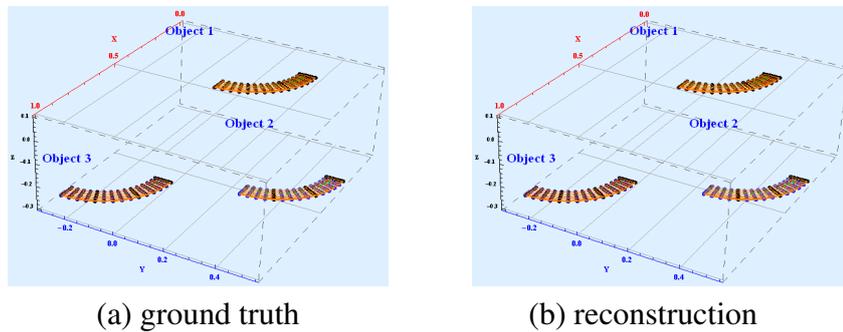


Figure 4.19: Planar 6D textures.

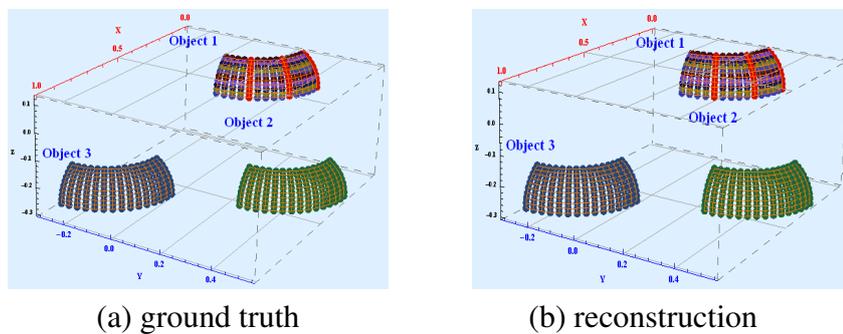


Figure 4.20: 6D textures. Two texture patterns have no color variation.

Real Image Experiment

To show the reconstruction of real image textures we consider the texture shown in Fig. 4.23. The color texture is viewed from cameras with different sensitivity parameters. Image 1 in Fig. 4.23 (a) was taken by a high gain camera, while image 2 in Fig. 4.23 (b) was taken by a low gain camera. Since, we do not know the true sampling order of images, it is assumed that the sampling order of 2 is relatively shifted horizontally and vertically. The proposed method was used to compute the tensor \mathcal{T}_{ijkl} and estimate the sampling shift. It was found correct. The sampling order of image 2 was shifted by 2 units vertically and 1 unit horizontally. Figure 4.24 shows the (a) ground truth and (b) reconstruction from the proposed method while (c) shows the result from the classical method. The reconstruction result from the proposed method is almost same as the ground truth, confirming the results of synthetic image experiments.

Next, color texture was reprojected to arbitrary views. Figure 4.25 show the re-projection result. Green and red lines link actual image points and reprojected image

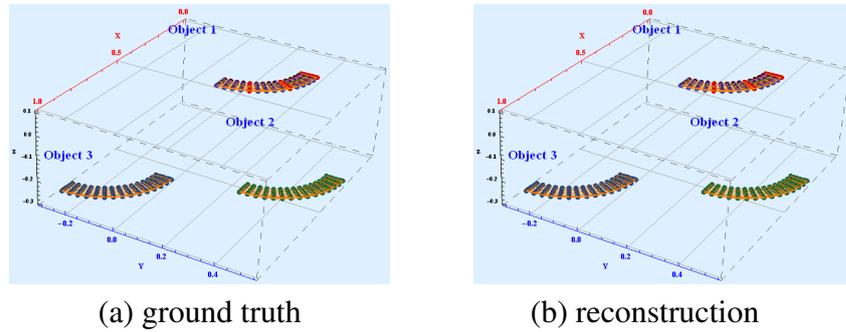


Figure 4.21: Planar 6D textures. Two texture patterns have no color variation.

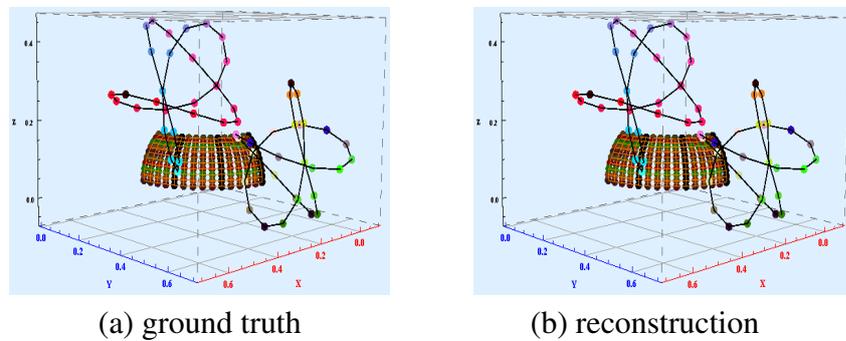


Figure 4.22: Reconstruction of mixed 6D objects.

points. We may observe the lines are almost coincident. This shows that arbitrary view is generated properly.

Stability Evaluation

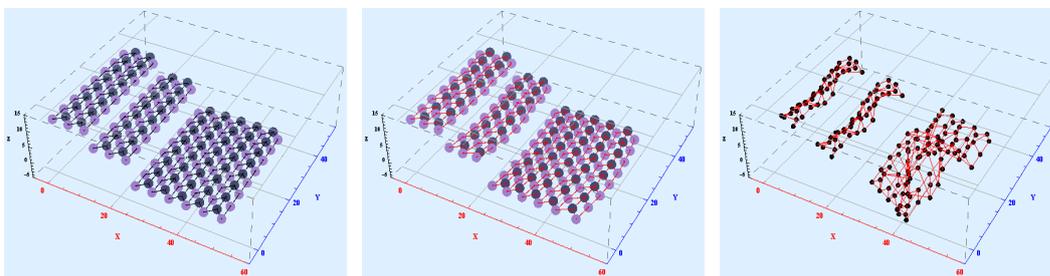
We next examine the stability of reconstruction of the color textures in the presence of random image noise and increase in no. of frequencies used for its computation. The texture shape, color, camera sensitivity parameters and positions were changed. Also, image points coordinates and color information were perturbed by a Gaussian noise of standard deviation of 1. As expected from Sec. 4.8.1 tensor showed same response Fig. 4.26, displaying robustness against noise and exponential decrease in reconstruction error. The result was an enhanced stability with increase in no. of frequencies.



(a) image 1 (high gain)

(b) image 2 (low gain)

Figure 4.23: Real images used for color texture reconstruction



(a) ground truth

(b) reconstruction from proposed method

(c) reconstruction from classical method

Figure 4.24: Original and reconstructed color texture. The reconstructed object has correct 3D texture shape and 3D color.

4.9 Summary

In this chapter, reconstruction of color sequences and textures was shown. The dimensions of object and image space were 6D and 5D respectively. Multiview constraints for 2 cameras were derived in frequency domain. One dimensional Fourier transform was used for sequences and two dimensional Fourier transform was used for textures. \mathcal{T}_{ijkl} is the algebraic entity between color sequences and textures of two images. Multiview relationship for 3, ..., 7 cameras were also shown. It was shown that by relaxing correspondence criteria, reconstruction from relatively shifted samples of color objects is possible.



Figure 4.25: Arbitrary view of the 6D texture. Green and red lines join the actual image points and reprojected points respectively.

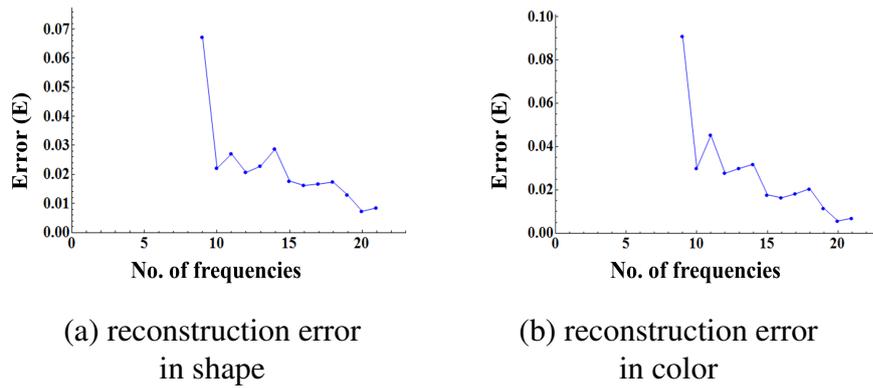


Figure 4.26: Stability of reconstruction. Horizontal axes are no. of frequencies for tensor computation, vertical axes represent reconstruction errors.

Chapter 5

Conclusion

Computer Vision is an interesting field with diverse applications in industry, medicine etc. However, the biggest advantage it offers is the visual practicality of linear algebra in education. It is quite interesting to apply the various concepts of physics, mathematics etc. to find a solution to a problem. Multiple view geometry is the core of computer vision. It's classical theory as well as its ability to expand and encompass other information is intriguing. It is also the feature used in this research.

The object of this research was to reconstruct texture objects and synthesize its arbitrary views. Two kinds of textures were considered 1D sequential and 2D patch texture. Each of the pattern was studied under the light of two cases grayscale and color information. The method to solve the problem consisted of expanding the scope of classical view geometry for mixed image information.

For reconstruction, finding accurate correspondences is inevitable. Therefore, another important aspect was deriving multiview constraints while relaxing the correspondence criteria. The approach was to work in Fourier domain, instead of spatial. With properties of Fourier domain such as linearity and translation the correspondence of individual points was solved.

Texture is usually characterized by the repetition of texture elements or group of points (patch correspondence). The proposed method to handle group correspondence is to utilize tensor consistency with minimization of reprojection errors. Thus far, static textures are reconstructed and being synthesized to produce arbitrary views.

Several extensions of this research can be considered of this research. One of them,

is to contemplate on dynamic textures i.e. reconstruction of moving textures.

Appendix A

Theory

A.1 Homogeneous Coordinates

Homogeneous coordinates can be defined as an extended coordinate system (usually of Euclidean in spatial domain). Generally in Euclidean coordinate system we describe a point in 2D and 3D as $\mathbf{x} = [x, y]^T$ and $\mathbf{X} = [X, Y, Z]^T$ respectively. In homogeneous coordinates we add an additional coordinate as follows:

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \lambda \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (\text{A.1})$$

$$\mathbf{X} = \begin{bmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \end{bmatrix} = \lambda \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (\text{A.2})$$

where λ denotes arbitrary scalar.

The benefits of using homogeneous coordinate system are:

1. It can define the point at infinity as $[x, y, 0]^T$ and differentiate between points at infinity in different directions e.g. $[2, 3, 1, 0]^T$ and $[-2, -3, 1, 0]^T$. Therefore, the intersection point of parallel lines can be described mathematically as a point at infinity.

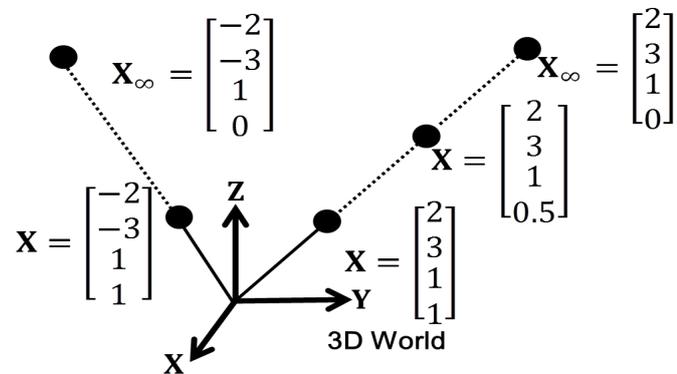


Figure A.1: 3D points in homogeneous coordinates.

2. It enables us to write non-linear equations linearly. For example, camera projection equations for perspective cameras can be described linearly as follows:

$$\mathbf{x} = \mathbf{P}\mathbf{X} \quad (\text{A.3})$$

where \mathbf{x} and \mathbf{X} are 2D and 3D homogeneous coordinates and \mathbf{P} is a camera matrix.

Fig. A.1 shows the 3D points with homogeneous coordinates.

A.2 Matrix operations

Some common matrix operations used are enumerated below.

1. A cross product of vector $\mathbf{X} = [X_1, X_2, X_3]^T$ and $\mathbf{Y} = [Y_1, Y_2, Y_3]^T$ is defined in matrix notation as follows:

$$\mathbf{X} \times \mathbf{Y} = [\mathbf{X}]_{\times} \mathbf{Y} \quad (\text{A.4})$$

where $[\mathbf{X}]_{\times}$ is defined as follows:

$$[\mathbf{X}]_{\times} = \begin{bmatrix} 0 & -X_3 & X_2 \\ X_3 & 0 & -X_1 \\ -X_2 & X_1 & 0 \end{bmatrix} \quad (\text{A.5})$$

2. A dot product is written as follows:

$$\mathbf{X} \cdot \mathbf{Y} = \mathbf{X}^{\top} \mathbf{Y} \quad (\text{A.6})$$

3. Pseudo Inverse: Let us suppose \mathbf{a}, \mathbf{b} and \mathbf{M} have dimensions $n \times 1$, $m \times 1$ and $n \times m$ respectively, and let $\mathbf{a} = \mathbf{M}\mathbf{b}$. Then, pseudo inverse of a non-square matrix \mathbf{M} is given as follows:

$$\mathbf{M}^{+} = (\mathbf{M}^{\top} \mathbf{M})^{-1} \mathbf{M}^{\top} \quad (\text{if } n > m) \quad (\text{A.7})$$

$$\mathbf{M}^{-} = \mathbf{M}^{\top} (\mathbf{M} \mathbf{M}^{\top})^{-1} \quad (\text{if } n < m) \quad (\text{A.8})$$

A.3 Discrete Fourier Transform

One dimensional discrete Fourier transform of a vector $\mathbf{X} = [X^0, X^1, \dots, X^N]^{\top}$ can be written in matrix notation as follows:

$$\begin{bmatrix} X_f^0 \\ X_f^1 \\ X_f^2 \\ \cdot \\ \cdot \\ \cdot \\ X_f^N \end{bmatrix} = \frac{1}{N} \begin{bmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & E & E^2 & \dots & E^{(N-1)} \\ 1 & E^2 & E^4 & \dots & E^{2(N-1)} \\ \cdot & \cdot & \cdot & \dots & \cdot \\ \cdot & \cdot & \cdot & \dots & \cdot \\ \cdot & \cdot & \cdot & \dots & \cdot \\ 1 & E^{N-1} & E^{2(N-1)} & \dots & E^{(N-1)^2} \end{bmatrix}_{N \times N} \begin{bmatrix} X^0 \\ X^1 \\ X^2 \\ \cdot \\ \cdot \\ \cdot \\ X^N \end{bmatrix} \quad (\text{A.9})$$

where $E = e^{-\frac{2\pi j}{N}}$. Since Fourier transform is a linear operation. Two dimensional Fourier transform of data can be considered as arranging the data into matrix form and applying 1D Fourier transform on rows followed by 1D Fourier transform on columns.

A.4 Tensor and Tensor notation

The essentials of tensor and tensor notation are discussed in this appendix.

1. The tensor is a multi-dimensional array of quantities, such as scalar, vector and matrix. The dimension of quantities is called order/step/valency of the tensor. For example, a scalar is order 0 tensor, a vector is order 1 tensor, while a matrix is order 2 tensor.
2. The tensor also helps in conceptualizing the higher order algebraic entities, specially the ones that can not be described pictorially.
3. There are two types of tensor, that is covariant and contravariant tensors. If the quantity of the tensor changes in proportion to the basis, it is called covariant tensor. If the quantity changes in inverse proportion to the basis, it is called covariant tensor. The lines and planes are covariant tensors, while the 2D points and 3D points are contravariant tensors.
4. The tensor notation is a way to represent tensors compactly using indices. The upper index is used for contravariant tensor, while the lower index is used for covariant tensor. Thus, a point \mathbf{x} is represented by x^i , and a line l is represented by l_i . For the case of matrices upper index is for matrix rows and lower index is for matrix columns.
5. For the case of camera matrices e.g. in Eq. (2.51) and Eq. (3.32) the superscript is for matrix rows and subscript is for matrix columns.
6. In the tensor notation, summation sign is often omitted and shown as repeated indices in subscript of one variable and superscript of the other. It is known as "Einstein's Notation", "contraction" or "tensor summation". For instance, $\sum_i l_i x^i$ is described as $l_i x^i$. Thus, Eq. (3.40) represents following 4 equations:

$$\sum_i \sum_j z^i z'^j \mathcal{T}_{1ij} = 0 \quad (\text{A.10})$$

$$\sum_i \sum_j z^i z'^j \mathcal{T}_{2ij} = 0 \quad (\text{A.11})$$

$$\sum_i \sum_j z^i z'^j \mathcal{T}_{3ij} = 0 \quad (\text{A.12})$$

$$\sum_i \sum_j z^i z'^j \mathcal{T}_{4ij} = 0 \quad (\text{A.13})$$

7. ϵ_{ijk} is a special tensor known as "Levi-Civita tensor". It has value of 1 for even permutation of i, j and k , -1 for odd permutation and 0 otherwise. Usually, it separates out non-zero components and describes the polarity of the elements associated e.g. in multiview constraints Eq. (2.55), Eq. (3.39) etc. Some important properties of this tensor are:

- $\epsilon_{ijk\dots n} \epsilon^{ijk\dots n} = n!$ e.g. $\epsilon_{ijk} \epsilon^{ijk} = 6$
- cross product of two vectors \mathbf{a} and \mathbf{b} can be described as follows:

$$\mathbf{a} \times \mathbf{b} = \epsilon_{ijk} a^i b^j \quad (\text{A.14})$$

- skew symmetric matrix $[\cdot]_{\times}$ for cross product can be described as follows:

$$[\mathbf{a}]_{\times} = \epsilon_{ijk} a^i \quad (\text{A.15})$$

A.5 Fundamental Matrix vs. Essential Matrix

While deriving the essential matrix it is assumed that camera internal parameter matrix is known. Therefore, essential matrix has less DOF i.e. 5 compared to fundamental matrix i.e. 7. Let us consider the world coordinate is at the 1st camera. Then we may write the projection equations as follows:

$$\mathbf{x} = \mathbf{P}\mathbf{X} = [\mathbf{I}|\mathbf{0}]\mathbf{X} = \tilde{\mathbf{X}} \quad (\text{A.16})$$

$$\mathbf{x}' = \mathbf{P}'\mathbf{X} = [\mathbf{R}|\mathbf{T}]\mathbf{X} = \mathbf{R}\tilde{\mathbf{X}} + \mathbf{T} \quad (\text{A.17})$$

where $\tilde{\cdot}$ denotes non-homogeneous coordinates. Combining Eq. (A.16) and Eq. (A.17),

$$\mathbf{x}' = \mathbf{R}\mathbf{x} + \mathbf{T} \quad (\text{A.18})$$

Equation (A.18) shows the three vectors are coplanar, hence the volume enclosed must be zero. In vector notation this condition is expressed as follows:

$$\mathbf{x}' \cdot (\mathbf{T} \times (\mathbf{R}\mathbf{x})) = 0 \quad (\text{A.19})$$

In matrix notation we may write,

$$\mathbf{x}'^{\top} [\mathbf{T}]_{\times} \mathbf{R}\mathbf{x} = 0 \quad (\text{A.20})$$

where, essential matrix \mathbf{E} has following definition:

$$\mathbf{E} = [\mathbf{T}]_{\times} \mathbf{R} \quad (\text{A.21})$$

By using \mathbf{E} matrix, Eq. (A.20) takes the following form:

$$\mathbf{x}'^{\top} \mathbf{E}\mathbf{x} = 0 \quad (\text{A.22})$$

To mark the difference between fundamental and essential matrix clearly, let us describe \mathbf{x} and \mathbf{x}' in pixel based coordinates \mathbf{m} and \mathbf{m}' as follows:

$$\mathbf{m} = \mathbf{K}\mathbf{x} \quad (\text{A.23})$$

$$\mathbf{m}' = \mathbf{K}'\mathbf{x}' \quad (\text{A.24})$$

where \mathbf{K} and \mathbf{K}' are intrinsic parameter matrices. From Eq. (A.22), Eq. (A.23) and Eq. (A.24) we have:

$$\mathbf{m}'^{\top} \mathbf{K}'^{-\top} \mathbf{E}\mathbf{K}^{-1} \mathbf{m} = 0 \quad (\text{A.25})$$

where fundamental matrix is defined as follows:

$$\mathbf{F} = \mathbf{K}'^{-\top} \mathbf{E}\mathbf{K}^{-1} \quad (\text{A.26})$$

By using \mathbf{F} matrix, Eq. (A.25) is modified as follows:

$$\mathbf{m}'^{\top} \mathbf{F}\mathbf{m} = 0 \quad (\text{A.27})$$

Equations (A.22) and (A.27) are known as **Epipolar equations**. The difference being Eq. (A.22) is defined using physical image coordinates and Eq. (A.27) is generalized for correspondences given in pixel coordinates. Essential matrix has fewer DOF (5) than fundamental matrix (7). Three for rotation and 3 for translation equals 6, minus one for over all scale reduces to 5. The rank of \mathbf{E} is 2.

Appendix B

Experiments

B.1 Normalization and Denormalization

B.1.1 Normalization of image data

A typical image has dimensions 640×480 . Therefore the homogeneous coordinates of image point $\mathbf{x} = [x, y, 1]^\top$ ranges in $1 \sim 10^2$. Therefore, for computation of tensors involved in multiview constraints, we often have to work with data ranging from $1 \sim 10^8$ (e.g. terms like xy , $x'y$ in Eq. (2.61) and Eq. (3.53) range in $1 \sim 10^4 \sim 1$ and entries in matrix $\mathbf{M}^\top \mathbf{M}$ for computing the least squares solution range in $1 \sim 10^8$). This can lead to significant numerical inaccuracies in the presence of noise while estimating an approximate solution. To avoid such a problem image data is normalized. Normalization of image data refers to the task of scaling the image data so that it ranges from around $-\sqrt{2}$ to $\sqrt{2}$ and is centered around origin. It can be achieved by following equation.

$$A_i^j x^i = w^j \quad (\text{B.1})$$

where x^i and w^j are unnormalized and normalized image points. Matrix \mathbf{A} is defined as below:

$$\mathbf{A} = \begin{bmatrix} \frac{\sqrt{2}}{d} & 0 & \frac{-\sqrt{2}\tilde{x}}{d} \\ 0 & \frac{\sqrt{2}}{d} & \frac{-\sqrt{2}\tilde{y}}{d} \\ 0 & 0 & 1 \end{bmatrix} \quad (\text{B.2})$$

where \tilde{x} and \tilde{y} are the x -coordinate and y -coordinate of mean point and d is the average of distance of all image points to center/mean point. By application of normalization the images used for computation of tensor and subsequent calculation may have different coordinate origin and sizes. Normalization brings the performance and accuracy of linear algorithms for tensor computation near the iterative algorithms. Details on numerical accuracy due to normalization can be found in [32]. It is understood in all experiments (synthetic and real) in this thesis that images used for tensor computations are normalized, and denormalization is performed afterwards.

B.1.2 Denormalization of Tensor

Since the matrix/tensor is estimated from normalized image data, denormalization is necessary to revert the change in data range. We first consider denormalization of **fundamental matrix** F_{ij} (Sec. 2.5.4). Let us change the notation in Eq. (2.56) as equation below, where w^i and w'^j are the normalized image points.

$$w^i w'^j F_{ij} = 0 \quad (\text{B.3})$$

Then defining $w^i = A_p^i x^p$ and $w'^j = A_q'^j x'^q$, where x^p and x'^q are the un-normalized image points, we may write:

$$(A_p^i x^p)(A_q'^j x'^q) F_{ij} = 0 \quad (\text{B.4})$$

$$x^p x'^q (A_p^i A_q'^j F_{ij}) = 0 \quad (\text{B.5})$$

$$x^p x'^q \hat{F}_{pq} = 0 \quad (\text{B.6})$$

Therefore, we have

$$\hat{F}_{pq} = A_p^i A_q'^j F_{ij} \quad (\text{B.7})$$

where \hat{F}_{pq} is the denormalized fundamental matrix.

We next consider denormalization of **bifocal tensor** \mathcal{T}_{fij} (Sec. 3.5.4). The tensor is

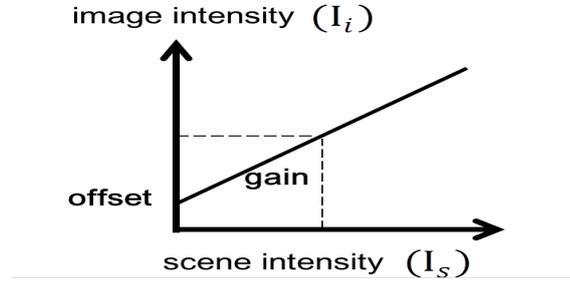


Figure B.1: Camera linear sensitivity parameters (gain and offset).

computed from normalized image data using following equation (refer Eq. (3.40)).

$$w^i w^j \mathcal{T}_{fij} = 0_f \quad (\text{B.8})$$

Since the relationship between normalized points w^i, w^j and un-normalized points z^p, z^q in frequency space are $w^i = A_p^i z^p$ and $w^j = A_q^j z^q$, we have:

$$z^p z^q (A_p^i A_q^j \mathcal{T}_{fij}) = 0_f \quad (\text{B.9})$$

$$z^p z^q \mathcal{T}_{fpq} = 0_f \quad (\text{B.10})$$

Therefore, we obtain:

$$\hat{\mathcal{T}}_{fpq} = A_p^i A_q^j \mathcal{T}_{fij} \quad (\text{B.11})$$

where $\hat{\mathcal{T}}_{fpq}$ is the denormalized tensor.

B.2 Camera Gain, Offset and Crosstalk

Two linear camera sensitivity parameters considered in this research are gain and offset. In Equation (3.3), p_{34} represents intensity gain and p_{35} intensity offset. These parameters are used for the affine projection of intensity information of world point and determine the brightness of the projected image point.

Intensity gain is a multiplicative factor and results in signal amplification. The

typical value is 1. Any value of gain < 1 will result in a darker intensity value. A high gain increases brightness, over all contrast and image noise.

Intensity offset is the value of the image intensity when the intensity of the original scene is zero. Any offset > 0 increases brightness and naturally decreases the dynamic range of intensity from $0 \sim 255$ to $offset \sim 255$. Also, any signal above the upper limit (255) due to offset addition is cut off.

Figure B.1 shows the roles of gain and offset. The image intensity I_i obtained by the camera and the original scene intensity I_s have the following relationship.

$$I_i = gain \times I_s + offset \quad (B.12)$$

Figure B.2 show the original image and Fig. B.3 show the effects of applying varying camera gains and offset.

Crosstalk: An image from a color camera consists of 3D color information that is red, green and blue color channels e.g. $[1, 0, 0]^T$ represents red. These color channels are projections in color space of the RGB values of the scene, and are independent of each other i.e. red color channel of an image depends on the red color of the scene and is independent of the green and the blue colors of the scene. Similarly, green color channel of an image is independent of the red and the blue colors of the scene and the blue color channel of an image is independent of the red and the green colors. However, in real cameras inter channel leakage occurs and color channels of the image are affected by one another, this is known as crosstalk. Under such circumstances, the image RGB values slightly differ from the true values as shown in Eq. (B.13).

$$\begin{bmatrix} r \\ g \\ b \end{bmatrix} = \begin{bmatrix} gain_r & c_{gr} & c_{br} \\ c_{rg} & gain_g & c_{bg} \\ c_{rb} & c_{gb} & gain_b \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (B.13)$$

where the diagonal components $gain_r$, $gain_g$ and $gain_b$ are the gains in the red, green and blue color channels and the off diagonal components c_{rb} , c_{rg} , c_{gr} , c_{gb} , c_{br} and c_{bg} are crosstalk components.

Although, this leakage/crosstalk is small compared to actual channel value, it affects the image color quality, so that the image appears to be color tinted e.g. bluish. We next show the effect of crosstalk by considering an example shown in Eq. (B.14).



Figure B.2: Original image.

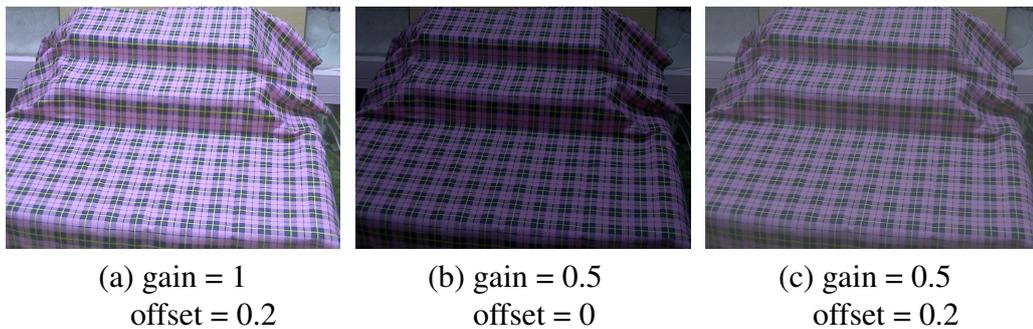


Figure B.3: Image with varying camera gains and offset.

Figure B.2 shows original image. Fig. B.4 (a) shows the result of the crosstalk applied and Fig. B.4 (b) shows the result of crosstalk with offset.

$$\begin{bmatrix} r \\ g \\ b \end{bmatrix} = \begin{bmatrix} 1 & 0.02 & 0.03 \\ 0.5 & 1 & 0.03 \\ 0.5 & 0.02 & 1 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (\text{B.14})$$

Camera gain, offset and crosstalk calibrations are usually performed to have radiometrically same images from different cameras. These images can be used for various applications e.g. correspondence search, reconstruction etc.

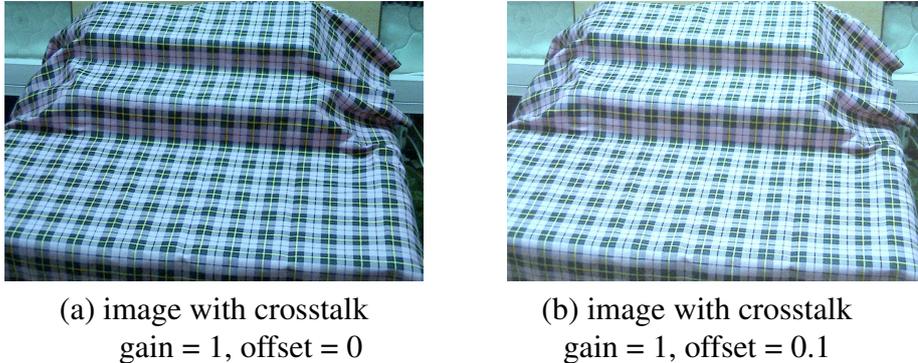


Figure B.4: Image with crosstalk and offset.

B.3 Gamma Transformation and Correction

Let's briefly review one more imaging parameters.

B.3.1 Gamma Transformation

Today many devices used for image capture or display work according to **gamma/power law transformation**. It is defined mathematically as follows:

$$Y = X^\gamma \quad (\text{B.15})$$

where Y and X are the output and input (original) image intensity values, γ is the transformation factor. If $\gamma < 1$, then according to Eq. (B.15) a narrow range of dark input values are mapped to a wide range of output values. Also, a wide range of high input values are mapped to a narrow range of dark input values. The foremost advantage is that range of middle graylevels/tones is expanded. These are the intensity levels our eyes are more sensitive to.

It is different from camera gain. In the case of gain enhancement gray levels of all pixels are brighten equally (linear response). For an image containing dark and bright areas. The darker areas are brightened but the bright areas are over brightened. To avoid such a case, gamma correction is applied. The result is although the dark areas get brighter, the bright areas remain relatively same (non-linear response). The value of gamma is always positive and usually in the range of $\frac{1}{5} \sim 5$, depending on device. Therefore, before working on images a **gamma correction** is applied to revert

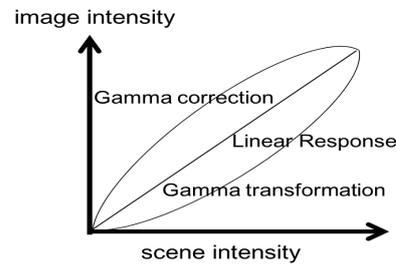


Figure B.5: Gamma transformation and gamma correction plot.

the changes in gray levels applied by camera according to gamma law. The result is output intensity levels as a linear function of input signal values. A conceptual graph of gamma transformation and gamma correction is given in Fig. B.5. In this graph it is assumed that image under consideration is far too bright, therefore the gamma transformation produces a darker one. Gamma correction reverses the transformation and produces the original image.

B.3.2 Gamma Correction

Before working with real image data gamma correction is applied to pixels intensity values. Taking logarithm of Eq. (B.15) yields following equation:

$$\log \mathbf{Y} = \gamma \log \mathbf{X} \quad (\text{B.16})$$

For i no. of input intensities, finding a least squares fit to minimize the error ϵ we obtain following equation:

$$\epsilon = \sum_i (\gamma \log \mathbf{X}_i - \log \mathbf{Y}_i)^2 \quad (\text{B.17})$$

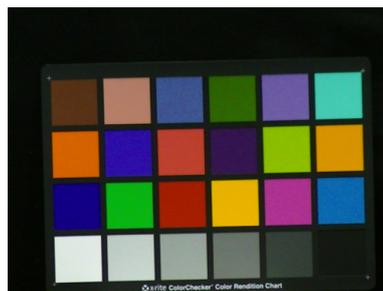
Equating the derivative w.r.t γ to 0 gives us the expression for γ as follows:

$$\gamma = \frac{\sum_i \log(\mathbf{X}_i) \sum_i \log(\mathbf{Y}_i)}{(\sum_i \log(\mathbf{X}_i))^2} \quad (\text{B.18})$$

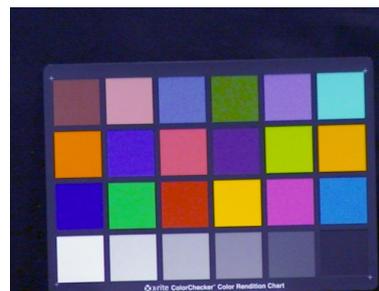
To find the γ values of red, green and blue color channels **X-rite color checker** was used in our experiments. Images of color checker were taken with low gain and high

Table B.1: γ values of the camera used for real image experiments.

Color	γ	γ
	(low gain)	(high gain)
Red	0.19	0.77
Green	0.76	0.69
Blue	0.52	0.55



(a) color checker



(b) gamma corrected image

Figure B.6: Image of x-rite color checker and its gamma corrected image.

gain. A chart of true color values \mathbf{X} is provided with it, and average observation values \mathbf{Y} were obtained after blob detection of images, denoted by \mathbf{Y} . Thus, Eq. (B.18) was used for finding γ value. Table B.1 shows γ values of the camera used in our real image experiments. Fig. B.6 shows the color checker with gamma corrected image. All real images used in this research are gamma corrected.

B.4 Blob detection using mathematica

To perform real image experiments of sequences and textures we need to sieve the required information. This information consists of blobs (texture elements). In our experiments, the blob detection was carried out in 2 main steps by using mathematica. A sample image after gamma correction is shown in Fig. B.7. Generally, the blobs (sequence/texture elements) are of various shapes and color. The first step is to make the required image features as distinctive as possible and to delete the unrequired features. Following are the steps performed:

- **Color separate** the image into red, green and blue as shown in Fig. B.8. Some blobs are detected more efficiently in one color than the other.



Figure B.7: Gamma corrected image used for blob detection.



(a) red

(b) green

(c) blue

Figure B.8: Image color channels separated into red, green and blue.

- **Binarize** each color separately. Most appropriate values for image thresholds ranged from 0.2 to 0.65. For this example binary image of blue color channel is considered.
- Following steps are used to reduce the size of the unnecessary features with locations near the blobs.
 - Apply **Erosion** (to reduce small light features) if the required blob is black
 - Apply **Dilation** (to remove tiny / thin dark features) if the required blob is colored white. Mostly erosion and dilation were performed with Diamond matrix. This is shown in Fig. B.9 (a).

Next, **Morphological Component** function of Mathematica 7.0 was used for labeling the foreground blobs using appropriate thresholds. Most appropriate values of image thresholds ranged from 0.4 ~ 0.75. Fig. B.9 (c) shows the blobs separated from background. It uses connected component analysis and has option for 4 pixel adjacency or 8

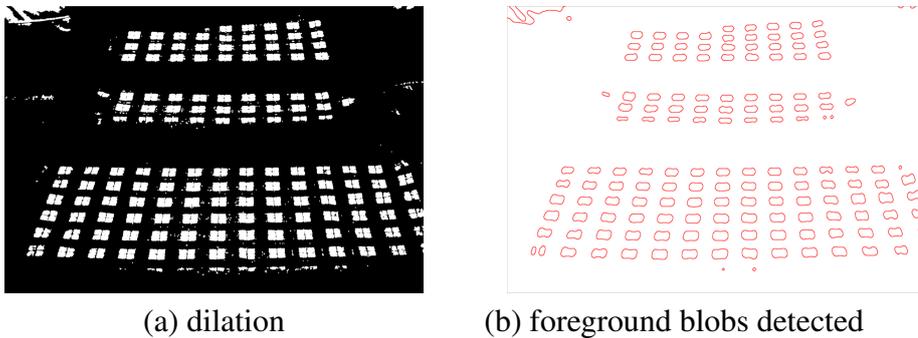


Figure B.9: Image dilation, and detection of blobs.



Figure B.10: Center points of (colored) blobs shown after detection.

pixel adjacency, latter is preferred. Related pixels are accumulated into N groups. For each group/blob find the center and average color/intensity, which gives N blob set for sequence/texture pattern per image. These center points however are not ordered in row, column form as shown in Fig. B.10. Finding the sequence of these points is established by the method shown in B.5.

B.5 Connecting image points

Blob detection shown in B.4 gives the image points. Next, these points can be connected either to form a sequence or texture.

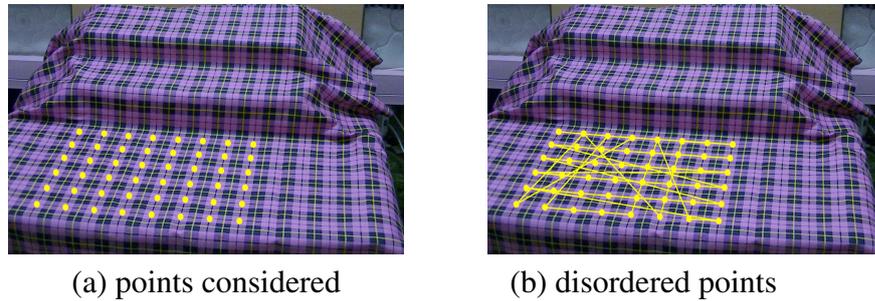


Figure B.11: Image for connecting the points in 2D grid. The considered points are shown in yellow.

B.5.1 Connecting image points for sequences

A set of points in an image can be connected in a closed sequence by finding the optimum path i.e. a sequence that results in overall minimum distance. In Mathematica 7.0 such an ordering of sequence points can be obtained using function `FindShortestTour[{ e_1, e_2, \dots, e_n }]`, where e_1, e_2, \dots, e_n represents the list of sequence points.

B.5.2 Connecting image points for textures

To connect points in a 2D grid, given the no. of rows (r) and columns (c), consider a set of points in an image, as shown in Fig. B.11 (a) by yellow. In this example $r = 6$ and $c = 8$. At the initial stage, these points are not arranged correctly as shown in Fig. B.11 (b). Then, we apply the following algorithm for arranging these points.

1. The points are first sorted w.r.t y - coordinates to get a rough arrangement.
2. Then the set of coordinates is divided into r subsets, each having c elements.
3. A line l is formed using 2 points, 1st and last points in the subset.
4. The points x are tested using $x^\top l < t$ and count the no. of points which lie on the line by setting a threshold t . In our experiments $t = 0.1$. Let this no. of points on the line be n .
5. The subset r_i is selected if the no. of points in step 4 equals the no. of columns i.e. $n = c$. It is then sorted w.r.t x - coordinates.



Figure B.12: The points are connected to form rows.



Figure B.13: 2D grid of points.

6. If $n < c$ in step 4, the line is rejected. A new line l is formed by selecting 2nd and 2nd last points, and steps 4 and 5 are repeated until $r = c$.
7. The points connected in rows is shown in Fig. B.12. Then, points in selected subsets r_i were concatenated to form a sorted list, where r_i form rows, as shown in Fig. B.12. The extracted 2D grid of blobs is shown in Fig. B.13.

Acknowledgements

Thank you very much Sato Sensei, Sakaue Sensei, Hontani Sensei, Kitamura Sensei
and Mori Sensei.

Thank you every one (members of Sato & Hontani Laboratory, Nagoya Institute of
Technology, MEXT, friends, family and many others) for everything.

List of Publications

Following papers were published during my doctoral study.

1. S. B. Miyan and J. Sato, “Reconstructing sequential patterns without knowing image correspondences”, *Proc. of Asian Conference on Computer Vision (ACCV)*, vol. 7727, pp. 484–496, 2012.
2. S. B. Miyan and J. Sato, “Finding Corresponding Patches in Texture Images using Tensor Consistency Check”, *Proc. of International Conference on Pattern Recognition (ICPR)*, pp. 4021–4026, 2014.
3. S. B. Miyan and J. Sato, “Reconstructing sequential patterns without knowing image correspondences”, *Proc. of IEICE Transaction on Information and Systems*, vol. E98-D, July 2015.

List of Figures

1.1	The process of reconstruction and arbitrary view synthesis. Corresponding points, reconstructed 3D point and its projection in arbitrary view are joined by black lines.	2
1.2	A texture patterns from two different viewpoints.	2
1.3	Example of sequential patterns	3
1.4	Example of texture patterns.	3
1.5	3D object projected to 2 images. Black points in image 1 and image 2 show position of 1st, 15th and 29th points.	4
1.6	Sampling of image 2 is shifted by 5 units. Black points show position of 1st, 15th and 29th points.	4
1.7	3D reconstruction under sampling shift. The sequence order of image 2 was shifted with 5 units, therefore reconstruction is distorted.	5
1.8	The same scene is observed differently in each camera image because of the difference in intensity parameters of camera.	5
1.9	Corresponding texture patches in two images. Red patches are corresponding. Black patches in image 2 are false corresponding patches. . .	6
1.10	World space.	6
2.1	A general camera projection. The line joining the camera center C and image point x is termed as line of sight (LOS). The figure shows the camera center at a finite position for general cameras and at infinity for affine cameras.	14
2.2	Affine camera approximation.	20

2.3	Backward translation of camera in direction of principal axis. The principal axis of camera is in direction of world Z -axis. The position vector C shows original position of camera center and $C - t$ shows position of camera center after translation $-t$	21
2.4	Affine approximation conditions.	24
2.5	A 3D object with several parallel lines. Parallel lines do not remain parallel after projection in perspective projection whereas, affine preserves parallelism.	26
2.6	A general camera projection. A back projected plane S from a line l in image. The reader may observe that 3D points projected onto line l form a plane defined by camera matrix and line.	27
2.7	Epipolar geometry.	28
2.8	Intersection of four planes at a single 3D point X . For two view geometry, planes S^1 and S^2 are back projected from image 1. Planes S'^1 and S'^2 are back projected from image 2.	29
2.9	Fundamental matrix transferring a point into a line in the other image.	32
2.10	Comparison of epipolar lines. The epipolar lines in perspective camera intersect at a finite epipole. The epipolar lines in affine camera intersect at infinity (epipole), thus parallel.	35
2.11	Image points x and x' of a 3D point X on an arbitrary plane π are related through a homography H . The homography H can be computed from fundamental matrix F	36
2.12	Four points forming affine basis vectors for the rectification of reconstruction.	38
3.1	The uniform reflection behavior of a Lambertian surface.	40
3.2	Example of a sequence in an image with sampling order shifted.	42
3.3	An example of texture sampling. (b) shows the sampling shift in horizontal and vertical direction using the numbering convention of (a). k and l denote a 2D sampling grid, and t and s represent the shift in horizontal and vertical directions of the sampling grid.	42

3.4	An extended camera projection from 4D world to 3D image. The line joining the camera center C and image point x is termed as line of sight (LOS). The figure shows the camera center is at infinity for affine cameras. A plane in the 3D image is back projected to a hyperplane in the 4D world space.	44
3.5	Intersection of five planes at a single 4D point Z . For two view geometry, hyperplanes S^1 , S^2 and S^3 are back projected from image 1 and S'^1 and S'^2 are back projected from image 2.	50
3.6	Homography induced by a plane.	60
3.7	Sequential objects used in synthetic experiment. There exist two series of 4D points, which are connected by lines.	63
3.8	3D Images taken from different viewpoints. Index numbers in red and blue points signify the sampling order.	64
3.9	Original and reconstructed 4D object. The reconstructed object has correct 3D shape and point intensity values.	65
3.10	Coplanar 4D sequences.	65
3.11	Perpendicularly planar 4D sequences.	66
3.12	4D sequences. One sequence has constant intensity.	66
3.13	Coplanar 4D sequences. One sequence has constant intensity.	67
3.14	Images of sequential objects used in real image experiment.	67
3.15	The extracted blobs of two sequences are indexed in red and yellow. Sampling order of image 2 sequences is shifted by 3 w.r.t the sampling order of image 1 sequences.	68
3.16	Reconstruction result of real image experiment.	68
3.17	Arbitrary views of real image experiment. Green and red lines connect the actual and reprojected image points respectively.	69
3.18	Stability of reconstruction. The horizontal axes represent the number of frequencies tensor computation. The vertical axes show errors in shape and intensity.	70
3.19	Grayscale texture object used in synthetic experiment. There exist two objects. The texture elements are connected in a 2D grid.	71
3.20	3D Images of a grayscale texture. The first sampling start point is shown in each image.	71

3.21	Ground truth and reconstructed grayscale texture. The reconstructed texture object has correct 3D shape and point intensity values.	72
3.22	Arbitrary views of reconstructed texture pattern.	72
3.23	Planar texture patterns.	73
3.24	Perpendicularly planar texture patterns.	73
3.25	Planar texture patterns. One texture pattern has constant intensity. . . .	74
3.26	Planar texture patterns. One texture pattern has constant intensity. . . .	74
3.27	Reconstruction of mixed 4D objects.	75
3.28	Grayscale texture object from different viewpoints.	75
3.29	Reconstruction of grayscale texture pattern.	76
3.30	Arbitrary view of grayscale reconstructed texture.	76
3.31	Stability of grayscale texture tensor \mathcal{T}_{fij} . Horizontal axes show no. of point (in frequency domain) used for tensor computation. Vertical axes show reconstruction error in (a) shape and (b) intensity.	77
4.1	6D world and 5D image. Three dimensions are for geometric information and three for color information in 6D. Two dimensions are for geometric and three for color information in 5D image.	79
4.2	Camera projection for 6D-5D affine camera. The line joining the camera center C and image point x is termed as line of sight (LOS). The figure shows the camera center is at infinity for affine cameras.	82
4.3	Intersection of seven hyperplanes at a 6D point Z . The hyperplanes S^1 , S^2 , S^3 , S^4 and S^5 are back projected from image 1. Hyperplanes S'^1 and S'^2 are back projected from image 2.	88
4.4	6D color sequences used in synthetic experiment. There are three series of color points, which are connected by lines (object 1, object 2 and object 3).	109
4.5	5D Images of color sequences.	109
4.6	Original and reconstructed 6D object. The reconstruction has correct 3D shape and color (RGB) information.	110
4.7	Arbitrary views of the 6D sequence reconstruction.	110
4.8	Planar 6D sequences.	111
4.9	6D sequences. Two sequences have no color variation.	111

4.10	Planar 6D sequences. Two sequences have no color variation.	112
4.11	Images for real experiment. The color tops of blocks form the sequential patterns.	112
4.12	Original and reconstructed 6D object. The reconstruction has correct 3D shape and color values.	113
4.13	Arbitrary views of the 6D real color sequence. Black lines connect actual image points. Red lines connect the reprojected points.	113
4.14	Stability of reconstruction. Horizontal axes show no. of frequencies for tensor computation. Vertical axes show reconstruction errors.	114
4.15	Color textures for synthetic experiment. The objects are 6 dimensional having 3D shape and repetitive 3D color information.	115
4.16	5D images of color texture objects.	115
4.17	Original and reconstructed 6D texture objects. The reconstructed color texture has correct 3D shape and color values.	116
4.18	Arbitrary views of 6D reconstructed texture.	116
4.19	Planar 6D textures.	117
4.20	6D textures. Two texture patterns have no color variation.	117
4.21	Planar 6D textures. Two texture patterns have no color variation.	118
4.22	Reconstruction of mixed 6D objects.	118
4.23	Real images used for color texture reconstruction	119
4.24	Original and reconstructed color texture. The reconstructed object has correct 3D texture shape and 3D color.	119
4.25	Arbitrary view of the 6D texture. Green and red lines join the actual image points and reprojected points respectively.	120
4.26	Stability of reconstruction. Horizontal axes are no. of frequencies for tensor computation, vertical axes represent reconstruction errors.	120
A.1	3D points in homogeneous coordinates.	124
B.1	Camera linear sensitivity parameters (gain and offset).	132
B.2	Original image.	134
B.3	Image with varying camera gains and offset.	134
B.4	Image with crosstalk and offset.	135
B.5	Gamma transformation and gamma correction plot.	136

B.6	Image of x-rite color checker and its gamma corrected image.	137
B.7	Gamma corrected image used for blob detection.	138
B.8	Image color channels separated into red, green and blue.	138
B.9	Image dilation, and detection of blobs.	139
B.10	Center points of (colored) blobs shown after detection.	139
B.11	Image for connecting the points in 2D grid. The considered points are shown in yellow.	140
B.12	The points are connected to form rows.	141
B.13	2D grid of points.	141

List of Tables

1.1	Abbreviations used in text.	7
1.2	Important symbols and their tensor representations.	9
4.1	List of all \mathcal{T}_{ijkl} non-zero elements (tensor - I).	92
4.2	List of additional 12 non-zero \mathcal{T}_{ijkl} elements (tensor - II).	96
B.1	γ values of the camera used for real image experiments.	137

Bibliography

- [1] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, Cambridge University Press, 2000.
- [2] L. Wolf and A. Shashua, “On projection matrices $\mathbf{P}^k \rightarrow \mathbf{P}^2$, $k = 3, \dots, 6$, and their applications in computer vision,” *Int’l. Journal of Computer Vision*, vol.48, pp.53–67, 2002.
- [3] C. Wan and J. Sato, “Multiple view geometry under projective projection in space-time,” *Trans. IEICE Inf.&Syst.*, vol.E91-D, pp.2353–2359, 2008.
- [4] H. Matsumoto, J. Sato, and F. Sakaue, “Multiview constraints in frequency space and camera calibration from unsynchronized images,” *IEEE Conf. on Computer Vision and Pattern Recognition (ICPR)*, pp.1601–1608, 2010.
- [5] S. Thirthala and M. Pollefeys, “Multi-view geometry of 1d radial cameras and its application to omnidirectional camera calibration,” *IEEE Int’l. Conf. on Computer Vision (ICCV)*, 2005.
- [6] R.A. Newcombe, S.J. Lovegrove, and A.J. Davison, “Dtam: dense tracking and mapping in real-time,” *IEEE Int’l. Conf. on Computer Vision (ICCV)*, 2011.
- [7] S. Agarwal, N. Snavely, I. Simon, S.M. Seitz, and R. Szeliski, “Building rome in a day,” *Int’l Conf. on Computer Vision (ICCV)*, 2009.
- [8] A.S. Ogale and Y. Aloimonos, “Shape and the stereo correspondence problem,” *Int’l Journal of Computer Vision*, pp.147–162, 2005.
- [9] G. Li and S.W. Zucker, “A differential geometrical model for contour-based stereo correspondence,” *Int’l. Conf. on Computer Vision (ICCV)*, 2003.

- [10] A.A. Efros and W.T. Freeman, "Image quilting for texture synthesis and transfer," ACM SIGGRAPH. Computer Graphics Proceedings, pp.341–346, 2001.
- [11] A. Hertzmann, C.E. Jacobs, N. Oliver, B. Curless, and D.H. Salesin, "Image analogies," Proc. ACM SIGGRAPH, *Proc. Computer Graphics*, pp.327–340, 2001.
- [12] S.C. Zhu, C.E. Guo, Y. Wang, and Z. Xu, "What are textons?," Int'l Journal of Computer Vision, Vol.62, pp.121-143, 2005.
- [13] N. Ahuja, S. Todorovic, "Extracting Texels in 2.1D Natural Textures," *Proc. of Int'l Conf. on Computer Vision (ACCV)*, pp.1-8, 2007.
- [14] T. Lindeberg, *Scale-Space Theory in Computer Vision*, Kluwer Academic Publishers, 1993.
- [15] D.G. Lowe, *Perceptual organization and visual recognition*, Kluwer Academic Publishers, 1985.
- [16] L.L. Loss, G. Bebis, M. Nicolescu, A. Skurikhin, "An iterative multi-scale tensor voting scheme for perceptual grouping of natural shapes in cluttered backgrounds," *Computer Vision and Image Understanding*, Vol.113, No.1, pp.126-149, 2009.
- [17] J. Sato, *Computer Vision - Geometry of views*, Corona Press, 1999.
- [18] P. R. Rao, *Signals and Systems*, McGraw Hill Education Private Limited, 2008.
- [19] D. A. Forsyth and J. Ponce, *Computer Vision - A Modern Approach*, Pearson Prentice Hall, 2003.
- [20] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, Pearson Prentice Hall, 2009.
- [21] R. Szeliski, *Computer Vision: Algorithms and Applications*, Springer, 2010.
- [22] C. Barnes, E. Shechtman, D. B. Goldman, and A. Finkelstein, "The generalized patchmatch correspondence algorithm," *Proc. of ECCV*, pp. 29–43, 2010.
- [23] F. Dornaika and R. Chung, "Stereo correspondence from motion correspondence," *Proc. of CVPR*, pp. 1070–1075, 1999.

- [24] D. G. Lowe, "Object recognition from local scale invariant features," *Proc. of ICCV*, vol. 2, pp. 1150–1157, 1999.
- [25] H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool, "Surf: Speeded up robust features," *Computer Vision and Image Understanding (CVIU)*, vol. 110, no. 3, pp. 346–359, 2008.
- [26] Y. Kakumu, F. Sakaue, J. Sato, K. Ishimaru and M. Imanishi, "High Frequency 3D Reconstruction from Unsynchronized Multiple Cameras", *Proc. of the British Machine Vision Conference (BMVC)*, 2013.
- [27] K. Hayakawa and J. Sato, "Multiple View Geometry in the Space-time", *Proc. of Asian Conference on Computer Vision (ACCV)*, pp. 437–446, 2006.
- [28] N. VISHVJIT. A Guided Tour of Computer Vision. Addison-Wesley, 1993.
- [29] K. Kozuka and J. Sato, "Multiple view geometry for mixed dimensional cameras", *In Proc. of VISAPP*, 2008.
- [30] R. Kurazume, W. Ishikawa, K. Takekazu and J. Sato, *Computer Vision and Image Media, CVIM Tutorial Series 1*, 2008.
- [31] K. Kozuka, C. Wan, J. Sato, "Rectification of 3D data obtained from moving range sensor by using extended projective multiple view geometry", *Int'l. Journal of Automation and Computing*, vol. 5, pp. 268–275, 2008.
- [32] R. I. Hartley, "In Defense of the Eight-Point Algorithm", *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 9, pp.580–593, 1997.
- [33] D. C. Lay, "Linear Algebra and its Applications", Addison Wesley, 2000.
- [34] Z. Wei and J. Kosecka, "Generalized RANSAC Framework for Relaxed Correspondence Problems", in '3DPVT', IEEE Computer Society, pp. 854–860, 2006.
- [35] N. Payet and S. Todorovic, "Scene shape from texture of objects", *CVPR*, IEEE Computer Society, pp. 2017–2024, 2011.
- [36] S. Bakshi and Y. Yang, "Shape from shading for non-lambertian surfaces", *ICIP* 94, vol. 94, pp. 130–134, 1994.

- [37] S. B. Miyan and J. Sato, “Reconstructing sequential patterns without knowing image correspondences”, *Proc. of ACCV*, vol. 7727, pp. 484–496, 2012.
- [38] S. B. Miyan and J. Sato, “Finding Corresponding Patches in Texture Images using Tensor Consistency Check”, *Proc. of ICPR*, pp. 4021–4026, 2014.