

Chapter 1

LATEST PROGRESS OF 3-D RECONSTRUCTION FROM MULTIPLE CAMERA IMAGES

Kenichi Kanatani

Abstract

This chapter summarizes recent progress of the theories and techniques for 3-D reconstruction from multiple images taken by multiple cameras. We start with the camera imaging geometry in terms of homogeneous coordinates and the intrinsic and extrinsic parameters. Next, we describe the epipolar geometry for two, three, and four cameras, introducing such concepts as the fundamental matrix, epipolar lines, epipoles, the trifocal tensor, and the quadrifocal tensor. Then, we present the self-calibration technique using the absolute dual quadric constraint. Finally, we give the definition of the affine camera model and a procedure for 3-D reconstruction based on it. The detailed algorithms are listed in the Appendix.

1. Introduction

Analyzing camera or video images for understanding the 3-D meaning of the captured scene is generally known as *computer vision* (also *machine vision*, *robot vision*, or *image understanding*, depending on the emphasis of the researchers), which is one of the most crucial elements of autonomous robotic operations. In general terms, the procedure consists of the following three stages:

- Image processing for detecting, extracting, and matching *features*, which can be points, lines, regions, or anything that is characteristic to that scene.
- Acquiring *metric* information such as locations, orientations, distances, sizes, and motions of the objects in the scene.

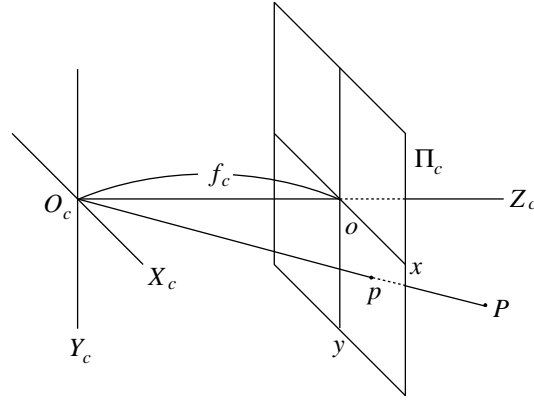


Figure 1. Perspective projection.

- Obtaining *semantic* information such as classification, recognition, labeling, indexing, and retrieval of specific objects in the scene.

These three stages roughly correspond to what has historically been known as *early* (or *low-level*) *vision*, *intermediate-level vision*, and *high-level vision*, respectively [22]. However, these are not necessarily treated separately. In fact, these stages are closely and interactively interwoven in most real computer vision systems.

One of the essential techniques for the second stage is to compute the 3-D shape of the scene or objects from multiple images, known as *3-D reconstruction* or *structure from motion (SFM)*. This computation critically depends on the *camera imaging geometry*, i.e., the geometric relationship between a 3-D scene and its projection onto a 2-D image. In contrast, analysis for the third stage crucially relies on the *domain knowledge* specific to individual applications such as faces, gestures, gaits, traffic, aerial photographs, and medical images.

Although the third stage is the ultimate goal of computer vision, it is still a very challenging task, and no universally satisfactory technologies have yet been established. However, the 3-D reconstruction technique for the second stage has been extensively studied in the last few decades to arrive at almost definitive conclusions. The aim of this chapter is to present thus established latest technologies of 3-D reconstruction from multiple images. Standard textbooks on this subject are, for example, [4, 5, 6, 8, 13, 14, 15, 23, 44].

2. Camera Imaging Geometry

2.1. Perspective Projection

We identify an image, or a photograph, with a mapping from a 3-D scene onto a 2-D plane and call this mapping the *camera model*. The standard model is *perspective projection* (Fig. 1): we imagine in the scene a point O_c , called the *viewpoint*, and a plane Π_c , called the *image plane* or *retina*, and assume that a point P in the scene is mapped to the intersection p of the image plane Π_c with the line O_cP , called the *line of sight*. This models an ideal *pin-hole camera* and is known to describe real cameras with sufficient accuracy.

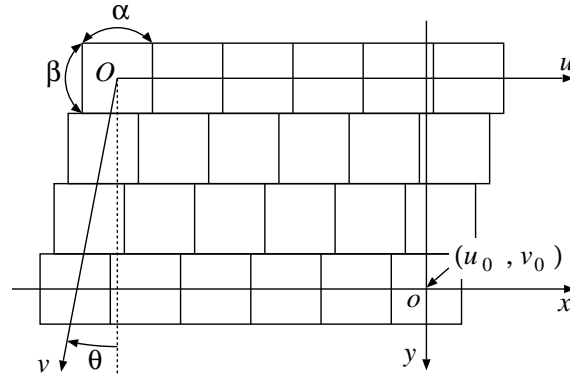


Figure 2. Pixels and the image coordinate system.

The line starting from the viewpoint O_c and perpendicularly passing through the image plane Π_c is called the *optical axis*. We define an $X_c Y_c Z_c$ coordinate system with the origin at the viewpoint O_c and the Z_c -axis along the optical axis. The intersection o of the optical axis with the image plane Π_c is called the *principal point*. We define an xy coordinate system with the origin at the principal point o and the x - and the y -axes parallel to the X_c - and the Y_c -axes, respectively (Fig. 1). Then, a point (X_c, Y_c, Z_c) in the scene is projected onto a point (x, y) in the image plane given by

$$x = f_c \frac{X_c}{Z_c}, \quad y = f_c \frac{Y_c}{Z_c}, \quad (1)$$

where f_c , called the *focal length*, is the distance from the viewpoint O_c to the image plane Π_c .

2.2. Pixel Coordinates

In real cameras, the image plane corresponds to the array of photo-cells, or *pixels*. The physical photo-cell configuration, in particular the configuration of the R-G-B (red, green, and blue) photocells, may differ depending on the type of the camera. Conceptually, however, we can think of pixels capable of perceiving R, G, and B placed in parallel rows at equal intervals in horizontal and vertical directions, but the vertical columns of pixels are not necessarily orthogonal to the horizontal rows. Also, the inter-pixel distance may not be the same in horizontal and vertical directions. Labeling the upper-left pixel $(u, v) = (0, 0)$, we count the pixels $u = 1, 2, \dots$ rightward and $v = 1, 2, \dots$ downward. Thus, the integer pair (u, v) is identified with the position at the center of that pixel. Inter-pixel, or *subpixel*, positions are specified with real number pairs (u, v) by linear interpolation. This defines a continuous *pixel coordinate system* of the image plane (Fig. 2).

If the xy coordinate system is oriented so that the x -axis is directed rightward in parallel to the horizontal pixel rows and the y -axis downward, the pixel coordinates (u, v) and the image coordinates (x, y) are related by

$$u = \frac{x}{\alpha} + \frac{y}{\alpha} \tan \theta + u_0, \quad v = \frac{y}{\beta} + v_0, \quad (2)$$

where (u_0, v_0) are the pixel coordinates of the principal point o , and α and β are, respectively, the distances between consecutive pixels in the horizontal and vertical directions. We define the angle between the horizontal and vertical pixel directions to be $\pi/2 + \theta$ and call θ the *skew angle*.

Remark 1 The xy coordinate system as defined above is “reversed” as compared with the usual sense. This convention originates from the human intuition that a hypothetical z -axis extends “away” from the viewer toward the scene, making the x -, y - and z -axes a right-handed system.

Remark 2 In most textbooks, the angle between the horizontal and vertical pixel directions is defined to be θ . Then, the first of Eqs. (2) becomes $u = x/\alpha + (y/\beta) \cot \theta + u_0$. We prefer our convention, because the skewless camera corresponds to $\theta = 0$ rather than $\theta = \pi/2$.

2.3. Intrinsic Parameters

Combining Eqs. (1) and (2), we have

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} \simeq \mathbf{K} \begin{pmatrix} X_c \\ Y_c \\ Z_c \end{pmatrix}, \quad (3)$$

where and throughout this chapter the symbol \simeq means that one side is a multiple of the other by a nonzero constant. The matrix \mathbf{K} is defined by

$$\mathbf{K} = \begin{pmatrix} f\gamma & f\gamma \tan \theta & u_0 \\ 0 & f & v_0 \\ 0 & 0 & 1 \end{pmatrix}, \quad (4)$$

where we put $f = f_c/\beta$, the normalized focal length so that the vertical distance between pixel rows is 1. Customarily, it is simply called the “focal length”. We also define $\gamma = \beta/\alpha$, called the *aspect ratio*. The constants f , γ , θ , u_0 , and v_0 are called the *intrinsic parameters* of the camera, and the matrix \mathbf{K} the *intrinsic parameter matrix*.

Remark 3 For digital cameras today, we can expect $\gamma \approx 1$ and $\theta \approx 0$ with high precision and the principal point (u_0, v_0) is nearly at the center of the photo-cell array.

Remark 4 In some textbooks, the vertical interval β is defined not as the distance between consecutive “rows” but as the distance between consecutive “pixels” in the vertical direction. In that case, the second of Eqs. (2) becomes $v = y/\beta \cos \theta + v_0$, so the (22) element of the matrix \mathbf{K} in Eq. (4) is $f/\cos \theta$. If we use the skew angle convention mentioned in Remark 2, $\cos \theta$ is replaced by $\sin \theta$. However, precise interpretation of the matrix \mathbf{K} is not essential. Many recent textbooks simply write

$$\mathbf{K} = \begin{pmatrix} f_1 & s & u_0 \\ 0 & f_2 & v_0 \\ 0 & 0 & 1 \end{pmatrix}, \quad (5)$$

emphasizing the fact that it is an *upper triangular matrix with 1 in the (33) element*.

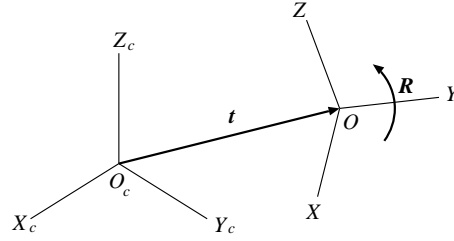


Figure 3. The camera coordinate system and the world coordinate system.

2.4. Motion Parameters

Since the $X_c Y_c Z_c$ coordinate system is defined with respect to the camera (i.e., the viewpoint O_c and the optical axis), it is called the *camera coordinate system*. We also define an XYZ coordinate system fixed to the scene and call it the *world coordinate system*. Let \mathbf{t} be its origin described with respect to the camera coordinate system. If the world coordinate system is rotated by \mathbf{R} relative to the camera coordinate system, a point in the scene with world coordinates (X, Y, Z) has the following camera coordinates (X_c, Y_c, Z_c) (Fig. 3):

$$\begin{pmatrix} X_c \\ Y_c \\ Z_c \end{pmatrix} = \mathbf{R} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} + \mathbf{t}. \quad (6)$$

We call $\{\mathbf{R}, \mathbf{t}\}$ the *motion parameters* or the *extrinsic parameters* of the camera.

Remark 5 The above motion parameters $\{\mathbf{R}, \mathbf{t}\}$ are a description with respect to the *camera* coordinate system. Alternatively, they can be described with respect to the *world* coordinate system. Let \mathbf{t}_c be the origin of the camera coordinate system described with respect to the world coordinate system. If the camera coordinate system is rotated by \mathbf{R}_c relative to the world coordinate system, we obtain instead of Eq. (6)

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \mathbf{R}_c \begin{pmatrix} X_c \\ Y_c \\ Z_c \end{pmatrix} + \mathbf{t}_c, \quad (7)$$

and the two descriptions $\{\mathbf{R}, \mathbf{t}\}$ and $\{\mathbf{R}_c, \mathbf{t}_c\}$ are related by

$$\mathbf{R} = \mathbf{R}_c^\top, \quad \mathbf{t} = -\mathbf{R}_c^\top \mathbf{t}_c. \quad (8)$$

2.5. Projection Matrix

From Eqs. (3) and (6), we can see that the pixel coordinates (u, v) are related to the world coordinates (X, Y, Z) in the form

$$\mathbf{u} \simeq \mathbf{P}\mathbf{X}, \quad (9)$$

where we put

$$\mathbf{u} = \begin{pmatrix} u \\ v \\ 1 \end{pmatrix}, \quad \mathbf{X} = \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}, \quad (10)$$

and

$$\mathbf{P} = \mathbf{K} \begin{pmatrix} \mathbf{R} & \mathbf{t} \end{pmatrix}. \quad (11)$$

This 3×4 matrix \mathbf{P} is called the *projection matrix* or the *camera matrix*. The vectors in Eqs. (10) represent the *homogeneous coordinates* of the point (u, v) in the image and the point (X, Y, Z) in the scene. Hereafter, we refer to points represented by vectors \mathbf{u} and \mathbf{X} simply as “point \mathbf{u} ” and “point \mathbf{X} ”, respectively.

Remark 6 Homogeneous coordinates are used not only for points in 2-D and 3-D but also for lines in 2-D and planes in 3-D, as we will see later. They are the description of points, lines, and planes with a set of real numbers, not all zero, defined up to a nonzero multiplier. For example, triples x^1, x^2, x^3 and cx^1, cx^2, cx^3 for an arbitrary $c \neq 0$ describe the same point in 2-D (the superscripts are indices, not powers). If $x^3 \neq 0$, the usual coordinates, or the *inhomogeneous coordinates*, are

$$x = \frac{x^1}{x^3}, \quad y = \frac{x^2}{x^3}. \quad (12)$$

If $x^3 = 0$, the point is interpreted to be at infinity; such a point is called an *ideal point*. Similarly, quadruples X^1, X^2, X^3, X^4 and cX^1, cX^2, cX^3, cX^4 for an arbitrary $c \neq 0$ describe the same point in 3-D. If $X^4 \neq 0$, its inhomogeneous coordinates are

$$X = \frac{X^1}{X^4}, \quad Y = \frac{X^2}{X^4}, \quad Z = \frac{X^3}{X^4}. \quad (13)$$

If $X^4 = 0$, the point is an *ideal point* at infinity. The symbol \simeq in Eqs. (3) and (9) reflects the indeterminacy of the absolute scale of homogeneous coordinates.

Remark 7 If we use the motion parameters $\{\mathbf{R}_c, \mathbf{t}_c\}$ described with respect to the world coordinate system, Eq. (11) becomes

$$\mathbf{P} = \mathbf{K} \begin{pmatrix} \mathbf{R}_c^\top & -\mathbf{R}_c^\top \mathbf{t} \end{pmatrix} = \mathbf{K} \mathbf{R}_c^\top (\mathbf{I} \quad -\mathbf{t}). \quad (14)$$

(\mathbf{I} denotes the unit matrix.) In this chapter, we adopt the description with respect to the camera coordinate system. Generally, the expressions become simpler if described with respect to the camera coordinate system, because the camera imaging geometry is usually defined with respect to the camera.

2.6. Absolute Conic

Since Eq. (9) is a relationship between homogeneous coordinates, it also holds for ideal points. In other words, Eq. (9) defines a mapping from the 3-D *projective space* \mathcal{P}^3 obtained by adding all ideal points in 3-D to \mathcal{R}^3 onto the 2-D *projective space* \mathcal{P}^2 obtained by adding all ideal points in 2-D to \mathcal{R}^2 .

The set Π_∞ of points X^1, X^2, X^3, X^4 in \mathcal{P}^3 with $X^4 = 0$ is called the *ideal plane*. The set Ω_∞ of (imaginary) points in Π_∞ that satisfy

$$(X^1)^2 + (X^2)^2 + (X^3)^2 = 0 \quad (15)$$

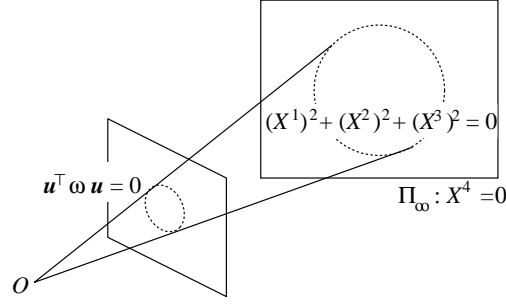


Figure 4. The absolute conic and its projection.

is called the *absolute conic*. It can be shown that any projection \mathbf{u} of a point $\mathbf{X} \in \Omega_\infty$ in the form of Eq. (9) satisfies, *irrespective* of the motion parameters $\{\mathbf{R}, \mathbf{t}\}$,

$$\mathbf{u}^\top \boldsymbol{\omega} \mathbf{u} = 0, \quad \boldsymbol{\omega} \equiv (\mathbf{K}^{-1})^\top \mathbf{K}^{-1}. \quad (16)$$

The set of (imaginary) points \mathbf{u} that satisfy this equation is interpreted to be the camera projection of the absolute conic Ω_∞ (Fig. 4).

Remark 8 If we are given camera images of objects in the scene with known 3-D information, we can determine the intrinsic parameters and the motion parameters of the camera in many different ways, depending on the type of the available 3-D information about the scene. Such a procedure is called *camera calibration*, and most known calibration procedures can be given projective geometric interpretations in terms of the absolute conic [45].

3. Epipolar Geometry

3.1. Multilinear Constraints

When geometric primitives such as points, lines, and planes in the scene are viewed by multiple cameras located in different positions, description of the relationships among their projection images is called *epipolar geometry* (typically for two cameras) or *multilinear geometry* (typically for more than two cameras).

Suppose we observe a point \mathbf{X} in the scene by M cameras. Let \mathbf{u}_κ be its projection onto the κ th image, $\kappa = 1, \dots, M$, and \mathbf{P}_κ the projection matrix of the κ th camera. For each camera, the relationship of Eq. (9) holds. If we introduce an indeterminate nonzero constant λ_κ instead of the relation \simeq , we have

$$\lambda_\kappa \mathbf{u}_\kappa = \mathbf{P}_\kappa \mathbf{X}. \quad (17)$$

The constant λ_κ is called the *projective depth*. Rearranging all the equations of this form for $\kappa = 1, \dots, M$ in a matrix form, we obtain

$$\begin{pmatrix} \mathbf{P}_1 & \mathbf{u}_1 & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{P}_2 & \mathbf{0} & \mathbf{u}_2 & \cdots & \mathbf{0} \\ \vdots & \mathbf{0} & \mathbf{0} & \ddots & \vdots \\ \mathbf{P}_M & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{u}_M \end{pmatrix} \begin{pmatrix} \mathbf{X} \\ -\lambda_1 \\ \vdots \\ -\lambda_M \end{pmatrix} = \begin{pmatrix} \mathbf{0} \\ 0 \\ \vdots \\ 0 \end{pmatrix}. \quad (18)$$

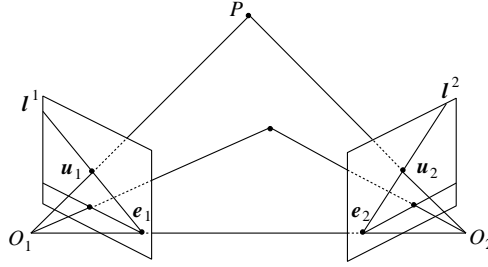


Figure 5. Epipolar lines and epipoles.

Since some \mathbf{X} ($\neq \mathbf{0}$) and λ_κ , $\kappa = 1, \dots, M$, that satisfy this equation should exist, the $3M \times (M+4)$ matrix on the left-hand side has at most rank $M+3$. Hence, all $(M+4) \times (M+4)$ minors should vanish. This leads to constraints on projection images in M ($= 2, 3, 4$) images [12].

Remark 9 It is easy to see that unless the chosen $(M+4) \times (M+4)$ minor contains two or more columns of \mathbf{P}_κ , we cannot obtain a meaningful constraint on the projection in the κ th image. In fact, if only one column of \mathbf{P}_κ is included, the resulting minor is linear in its elements, so its vanishing does not give any information about \mathbf{P}_κ . Hence, if M projection matrices are to be constrained by the vanishing of a $(M+4) \times (M+4)$ minor, we need $2M \leq M+4$, or $M \leq 4$. Thus, we can obtain constraints on only two, three, and four images.

3.2. Fundamental Matrix

For $M = 2$ (two images), the matrix on the left-hand side of Eq. (18) is 6×6 , so we obtain only one constraint: the matrix has determinant 0. This is rewritten as

$$\mathbf{u}_1^\top \mathbf{F} \mathbf{u}_2 = 0, \quad (19)$$

where \mathbf{F} is a 3×3 matrix called the *fundamental matrix*. Its (ij) element is

$$F_{ij} = \sum_{k,l,m,n=1}^3 \epsilon_{ikl} \epsilon_{jmn} \det \mathbf{P}_{1122}^{klmn}, \quad (20)$$

where \mathbf{P}_{1122}^{klmn} is the 4×4 matrix consisting of the k th row of \mathbf{P}_1 , the l th row of \mathbf{P}_1 , the m th row of \mathbf{P}_2 , and the n th row of \mathbf{P}_2 . From Eq. (20), it can be shown that the fundamental matrix \mathbf{F} has rank 2.

Remark 10 The symbol ϵ_{ijk} denotes the signature of the permutation (ijk) . Namely, it takes on 1 if (ijk) is an even permutation of (123) , -1 if it is an odd permutation, and 0 otherwise. This symbol is called the *Levi-Civita* (or *Eddington*) *epsilon*.

3.3. Epipolar Constraint

The line starting from the viewpoint O_1 of the first camera and passing through the point \mathbf{u}_1 in the image plane of the first camera is called the *line of sight* of \mathbf{u}_1 . The line of sight

of \mathbf{u}_2 is similarly defined. Geometrically, Eq. (19) describes the requirement that the line of sight of \mathbf{u}_1 and the line of sight of \mathbf{u}_2 should intersect at a point (it may be at infinity) in the scene. (Fig. 5). The set of points \mathbf{u} that satisfy $\mathbf{l}^\top \mathbf{u} = 0$ for some \mathbf{l} defines a line in the image. The vector \mathbf{l} labels this line up to a nonzero multiplier (i.e., \mathbf{l} and $c\mathbf{l}$ defines the same line for $c \neq 0$). The three components of \mathbf{l} define the homogeneous coordinates of this line. Henceforth, we abbreviate the line represented by vector \mathbf{l} simply as “line \mathbf{l} ”.

Eq. (19) implies that the point \mathbf{u}_1 is on the line $\mathbf{l}^1 = \mathbf{F}\mathbf{u}_2$, which is called the *epipolar line* of point \mathbf{u}_2 . Eq. (19) also implies that the point \mathbf{u}_2 is on the line $\mathbf{l}^2 = \mathbf{F}^\top \mathbf{u}_1$, called the *epipolar line* of point \mathbf{u}_1 . Thus, Eq. (19) states that *a point in one image should be on the epipolar line of the corresponding point in the other image*. This requirement is called the *epipolar constraint*. It follows that if the fundamental matrix \mathbf{F} is known, one can find point correspondence by searching the other image along the epipolar line of \mathbf{u} (Fig. 5).

3.4. Epipoles

Since the fundamental matrix \mathbf{F} has rank 2, it has a null vector. So does \mathbf{F}^\top , too. In other words, there exist vectors \mathbf{e}_1 and \mathbf{e}_2 such that $\mathbf{F}^\top \mathbf{e}_1 = \mathbf{0}$ and $\mathbf{F}\mathbf{e}_2 = \mathbf{0}$. Identifying \mathbf{e}_1 and \mathbf{e}_2 with homogeneous coordinates of points in the image, we call them the *epipoles*. Geometrically, the epipole \mathbf{e}_1 is *the projection of the viewpoint O_2 of the second camera onto the first image*, and the epipole \mathbf{e}_2 is *the projection of the viewpoint O_1 of the first camera onto the second image* (Fig. 5). From Eq. (19), we can see that in the first image the epipolar line $\mathbf{l}^1 = \mathbf{F}\mathbf{u}_2$ of any point \mathbf{u}_2 passes through the epipole \mathbf{e}_1 , i.e., $\mathbf{l}^{1\top} \mathbf{e}_1 = 0$. Similarly, in the second image, the epipolar line $\mathbf{l}^2 = \mathbf{F}^\top \mathbf{u}_1$ of any point \mathbf{u}_1 passes through the epipole \mathbf{e}_2 , i.e., $\mathbf{l}^{2\top} \mathbf{e}_2 = 0$.

It follows that epipolar lines of all points in the other image pass through the epipole, defining a *pencil of lines* (Fig. 5). This is easily understood if we note that the epipolar line of a point \mathbf{u}_2 of the second image is nothing but the intersection of the first image plane with the plane defined by \mathbf{u}_2 and the viewpoints O_1 and O_2 of the two cameras. This plane is called the *epipolar plane* of \mathbf{u}_2 (and hence of the corresponding point \mathbf{u}_1). The line connecting the two viewpoints O_1 and O_2 is called the *baseline*. All epipolar planes contain the baseline, defining a *pencil of planes* (Fig. 5).

3.5. Three-View Geometry

For $M = 3$ (three images), we obtain from Eq. (18) the following *trilinear constraint*:

$$\sum_{i,j,k,l,m=1}^3 \epsilon_{jlp} \epsilon_{kmq} T_i^{jk} u_1^i u_2^l u_3^m = 0. \quad (21)$$

Here, u_κ^i denotes the i th component of \mathbf{u}_κ , and

$$T_i^{jk} = \sum_{l,m=1}^3 \epsilon_{ilm} \det \mathbf{P}_{1123}^{lmjk} \quad (22)$$

is called the *trifocal tensor*.

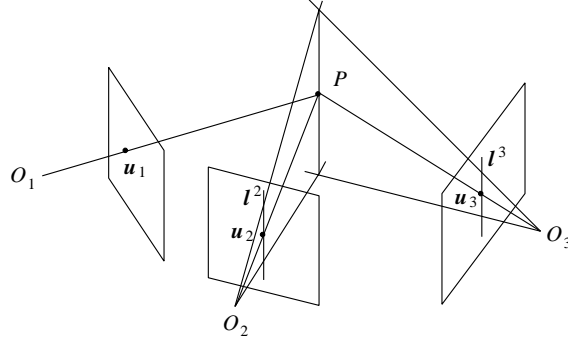


Figure 6. Trifocal constraint.

Given a line l in the image plane, the plane Π_1 defined by the viewpoint O_c and the line l is called the *back projection* of the line l . Let Π_{12} be the back projection of an arbitrary line l^2 passing through u_2 in the second image, and Π_{13} the back projection of an arbitrary line l^3 passing through u_3 in the third image. Geometrically, Eq. (21) describes the requirement that *the line of sight of u_1 in the first image should meet the intersection of the two planes Π_{12} and Π_{13} at a single point* (it may be at infinity) (Fig. 6).

Remark 11 Take an arbitrary point v_2 ($\neq u_2$) in the second image and an arbitrary point v_3 ($\neq u_3$) in the third image. Multiplying Eq. (21) by $v_2^p v_3^q$ and summing it over p and q , we obtain

$$\sum_{i,j,k=1}^3 T_i^{jk} u_1^i \left(\sum_{l,p=1}^3 \epsilon_{jlp} u_2^l v_2^p \right) \left(\sum_{m,q=1}^3 \epsilon_{kmq} u_3^m v_3^q \right) = 0. \quad (23)$$

If we define lines

$$l^2 = u_2 \times v_2, \quad l^3 = u_3 \times v_3, \quad (24)$$

Eq. (23) is rewritten as

$$\sum_{i,j,k=1}^3 T_i^{jk} u_1^i l_j^2 l_k^3 = 0, \quad (25)$$

which describe the geometric relationship mentioned earlier.

3.6. Four-View Geometry

For $M = 4$ (four images), we obtain from Eq. (18) the *quadrilinear constraint*

$$\sum_{i,j,k,l,m,n,p,q=1}^3 \epsilon_{ima} \epsilon_{jnb} \epsilon_{kpc} \epsilon_{lqd} Q^{ijkl} u_1^m u_2^n u_3^p u_4^q = 0, \quad (26)$$

where

$$Q^{ijkl} = \det \mathbf{P}_{1234}^{ijkl}, \quad (27)$$

is called the *quadrifocal tensor*. Geometrically, Eq. (26) describes the requirement that *the back projections $\Pi_{11}, \dots, \Pi_{14}$ of arbitrary lines l^1, \dots, l^4 in each image passing through points u_1, \dots, u_4 , respectively, should meet at a single point* (Fig. 7).

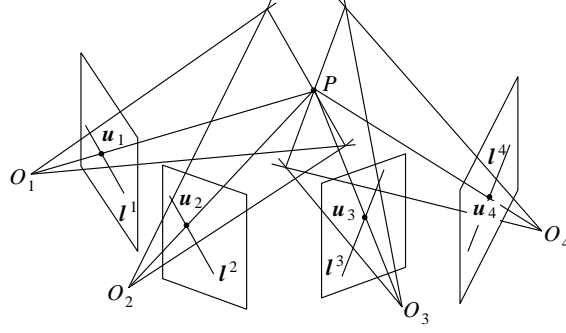


Figure 7. Quadrifocal constraint.

Remark 12 Take an arbitrary point \mathbf{v}_κ ($\neq \mathbf{u}_\kappa$) in the κ th image, $\kappa = 1, 2, 3, 4$. Multiplying Eq. (26) with $v_1^a v_2^b v_3^c v_4^d$ and summing it over a, b, c , and d , we obtain

$$\sum_{i,j,k,l=1}^3 Q^{ijkl} \left(\sum_{m,a=1}^3 \epsilon_{ima} u_1^m v_1^a \right) \left(\sum_{n,b=1}^3 \epsilon_{jnb} u_2^n v_2^b \right) \left(\sum_{p,c=1}^3 \epsilon_{kpc} u_3^p v_3^c \right) \left(\sum_{q,d=1}^3 \epsilon_{lqd} u_4^q v_4^d \right) = 0. \quad (28)$$

If we define lines

$$\begin{aligned} \mathbf{l}^1 &= \mathbf{u}_1 \times \mathbf{v}_1, & \mathbf{l}^2 &= \mathbf{u}_2 \times \mathbf{v}_2, \\ \mathbf{l}^3 &= \mathbf{u}_3 \times \mathbf{v}_3, & \mathbf{l}^4 &= \mathbf{u}_4 \times \mathbf{v}_4, \end{aligned} \quad (29)$$

Eq. (28) is rewritten as

$$\sum_{i,j,k,l=1}^3 Q^{ijkl} l_i^1 l_j^2 l_k^3 l_l^4 = 0, \quad (30)$$

which describe the geometric relationship mentioned earlier.

4. 3-D Reconstruction from Images

4.1. Classification of the Problem

Suppose we observe N points \mathbf{X}_α , $\alpha = 1, \dots, N$, in the scene by M cameras having projection matrices \mathbf{P}_κ , $\kappa = 1, \dots, M$. Equivalently, we may move one camera, changing its parameters and taking pictures at M different instances, which is also equivalent to fix the camera position and move the scene relative to it. In whichever interpretation, let $\mathbf{u}_{\kappa\alpha}$ the projection of point \mathbf{X}_α onto the κ th image. For each point and each image, we have the relationship described in the form of Eq. (9):

$$\mathbf{u}_{\kappa\alpha} \simeq \mathbf{P}_\kappa \mathbf{X}_\alpha. \quad (31)$$

Given projection images $\mathbf{u}_{\kappa\alpha}$, $\kappa = 1, \dots, M$, $\alpha = 1, \dots, N$, the task of computing \mathbf{X}_α , $\alpha = 1, \dots, N$, is called *3-D reconstruction* or *structure from motion (SFM)*. The problem is classified into the following three cases (we adopt the multiple camera interpretation for simplicity):

- (i) The projection matrix P of each camera is known.
- (ii) The intrinsic parameter matrix K of each camera is known (but the motion parameters $\{R, t\}$ are not).
- (iii) The projection matrix P of each camera is unknown.

In Case (i), Eq. (31) determines the 3-D coordinates $(X_\alpha, Y_\alpha, Z_\alpha)$ of point X_α up to one degree of freedom, which corresponds to the *depth* of the point X_α along the line of sight. In order to determine it uniquely, we need to observe two or more images. Computing the depths of points in the scene in this way is called (*multi-camera stereo vision*).

In Case (ii), the cameras are said to be *calibrated*. In this case, we first compute the fundamental matrix F from point correspondences between two images. Then, the motion parameters $\{R, t\}$ are determined by solving Eq. (20), and the problem reduces to stereo vision of Case (i).

In Case (iii), the cameras are said to be *uncalibrated*. 3-D reconstruction in this case is called *self-calibration* or *autocalibration*.

Remark 13 In Cases (ii) and (iii), the positions of the points in the scene and the camera motion parameters are determined only up to an unknown scale factor. This is because small camera motions relative to a small object located nearby cannot be distinguished from large camera motions relative to a large object located far away, as long as projection images are the only available information.

Remark 14 For calibrated cameras (Case (ii)), the motion parameters computed from the fundamental matrix F has ambiguity of “mirror image”. This is because we only require the 3-D positions of observed points to be on the lines of sight that they defines. As a result, the reconstructed shape can be a mirror image “behind” the camera. Mirror image solutions can be removed by imposing the constraint that observed points be in front of the cameras, which Hartley [7] called *cheirality* (or *chirality*) (see [14, 15] for the actual procedure).

4.2. Self-calibration

In Case (iii) (self-calibration), the projection matrices P_κ and the 3-D points X_α in Eq. (31) are both unknown. It is immediately seen from Eq. (31) that the solution is indeterminate if there is no constraint on the cameras or the 3-D points. In fact, if X_α and P_κ are a solution, we have another solution

$$\tilde{X}_\alpha \simeq H X_\alpha, \quad \tilde{P}_\kappa \simeq P_\kappa H^{-1} \quad (32)$$

for an arbitrary nonsingular 4×4 matrix H .

The first of Eqs. (32) can be regarded as applying a *projective transformation* (or a *homography*) H to the 3-D projective space \mathcal{P}^3 (Fig. 8). Accordingly, the points X_α and \tilde{X}_α have the same *projective structure*. For example, collinear points are mapped to collinear points, coplanar points are mapped to coplanar points, and their *incidence relationships*, such as “on ...”, “passing through ...” and “meeting at ...”, are preserved. However, *metric*

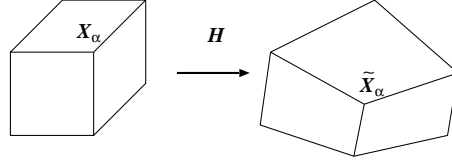


Figure 8. Projective transformation.

properties such as lengths and angles are not preserved. 3-D reconstruction determined up to an arbitrary projective transformation is called *projective reconstruction*.

In order to select a correct solution, one needs some constraint on either the cameras or the points. Selecting a unique solution by imposing such constraint is termed *upgrading* of projective reconstruction into *Euclidean* (or *metric*) *reconstruction*.

Note that Eqs. (32) are rewritten as

$$\mathbf{X}_\alpha \simeq \mathbf{H}^{-1} \tilde{\mathbf{X}}_\alpha, \quad \mathbf{P}_\kappa \simeq \tilde{\mathbf{P}}_\kappa \mathbf{H}. \quad (33)$$

If, for example, we know the true 3-D positions \mathbf{X}_α of five (or more) points in general position, we can uniquely determine the projective transformation \mathbf{H} that maps, or “rectifies”, the five points $\tilde{\mathbf{X}}_\alpha$ to their true positions \mathbf{X}_α . Applying the computed \mathbf{H} to the remaining points, we obtain the Euclidean reconstruction \mathbf{X}_α of all points. If no such five points are known, we need to assume some constraints on cameras and find an appropriate projective transformation \mathbf{H} such that the projection matrices \mathbf{P}_κ rectified by the second of Eqs. (33) satisfy the assumed constraints. This approach is called the *stratified reconstruction*.

Remark 15 Points in 3-D are said to be *in general position* if no three of them are coplanar. If we are given five (or more) points in general position for which we only know their relative configuration up to a scale factor, we can reconstruct the 3-D shape up to position, orientation, and scale by arbitrarily normalizing the position, the orientation, and the scale.

Remark 16 If no 3-D information is given about the scene, the absolute scale cannot be determined from images alone, as pointed out in Remark 13. Hence, all that can be obtained is, strictly speaking, “similarity” reconstruction rather than “Euclidean” or “metric”. However, the terms “Euclidean” and “metric” are commonly used to mean “up to similarity”.

4.3. Stratified Reconstruction

Eliminating the rotation \mathbf{R} from Eq. (11) by using the identity $\mathbf{R}\mathbf{R}^\top = \mathbf{I}$, we obtain for each image

$$\mathbf{P}_\kappa \text{diag}(1, 1, 1, 0) \mathbf{P}_\kappa^\top = \boldsymbol{\omega}_\kappa^*, \quad (34)$$

where $\text{diag}(a, b, c, \dots)$ denotes the diagonal matrix with diagonal elements a, b, c, \dots in that order. The 3×3 matrix $\boldsymbol{\omega}_\kappa^*$ is defined by

$$\boldsymbol{\omega}_\kappa^* \equiv \mathbf{K}_\kappa \mathbf{K}_\kappa^\top. \quad (35)$$

Substituting \mathbf{P}_κ in the second of Eqs. (33) into Eq. (34), we obtain

$$\tilde{\mathbf{P}}_\kappa \boldsymbol{\Omega}_\infty^* \tilde{\mathbf{P}}_\kappa^\top \simeq \boldsymbol{\omega}_\kappa^*, \quad (36)$$

where we define the 4×4 matrix Ω_∞^* by

$$\Omega_\infty^* \equiv \mathbf{H} \text{diag}(1, 1, 1, 0) \mathbf{H}^\top. \quad (37)$$

If the intrinsic parameter matrix \mathbf{K}_κ is known (i.e., the camera is calibrated), we can determine ω_κ^* from Eq. (35). Even if ω_κ^* is not completely known, we can obtain constraints on the elements of Ω_∞^* from Eq. (36) if we have some knowledge about ω_κ^* , such as a particular element being 0 or two particular elements being equal (we are assuming that $\tilde{\mathbf{P}}_\kappa$ are given). If the number M of images is sufficiently large to give a sufficient number of such constraints on Ω_∞^* , we can determine Ω_∞^* . Frequently used assumptions about the cameras are:

- All cameras have the same intrinsic parameters.
- The location of the principal point is known for all cameras.
- The skew angle θ is 0 for all cameras.
- The aspect ratio γ is 1 for all cameras.

For example, if all cameras have the same intrinsic parameters (i.e., one camera is moved to take multiple pictures without changing its parameters), the unknown is one intrinsic parameter matrix \mathbf{K} , so $\omega_1^* = \dots = \omega_M^* = \omega^* (\equiv \mathbf{K} \mathbf{K}^\top)$. Hence, Eq. (36) gives $5(M-1)$ equations of Ω_∞^* . If the principal point is known, we can translate the coordinate system so that $u_0 = v_0 = 0$. Then, the (13) and (23) elements of \mathbf{K} in Eq. (4) are 0, and hence the (13) and (23) elements of $\omega_\kappa^* = \mathbf{K}_\kappa \mathbf{K}_\kappa^\top$ are also 0. In this case, Eq. (36) gives $2M$ equations of Ω_∞^* . If the skew angle is zero in addition, the (12) element of ω_κ^* is also zero, so we obtain $3M$ equations of Ω_∞^* . If furthermore the aspect ratio γ is 1, the (11) element and the (22) element are equal, giving M additional equations. If we obtain nine or more such equations, we can solve them for Ω_∞^* up to a scale factor. If Ω_∞^* is determined, ω_κ^* is determined from Eq. (36). Then, the projective transformation \mathbf{H} is determined from Eq. (37). The intrinsic parameter matrix \mathbf{K}_κ is obtained by solving Eq. (35).

Remark 17 From Eq. (4), the matrix ω_κ^* in Eq. (35) has the form

$$\omega_\kappa^* = \begin{pmatrix} f_\kappa^2 \gamma_\kappa^2 + s_\kappa^2 + u_{0\kappa}^2 & f_\kappa s_\kappa^2 + u_{0\kappa} v_{0\kappa} & u_{0\kappa} \\ f_\kappa s_\kappa^2 + u_{0\kappa} v_{0\kappa} & f_\kappa^2 + v_{0\kappa}^2 & v_{0\kappa} \\ u_{0\kappa} & v_{0\kappa} & 1 \end{pmatrix}, \quad (38)$$

where we put $s_\kappa = f_\kappa \gamma_\kappa \tan \theta$. This is a 3×3 symmetric matrix with six different elements. Hence, if all the intrinsic parameters are known, Eq. (36) gives five constraints for each κ (one degree of freedom is lost for the indeterminate scale factor). The unknown is the 4×4 symmetric matrix Ω_∞^* with ten independent elements, but it has scale indeterminacy. Hence, two views are sufficient.

If the intrinsic parameters are all unknown but are the same for all cameras (or one camera is moved), we need to observe M views such that $5(M-1) \geq 9$, or $M \geq 3$. If the principal point $(u_{0\kappa}, v_{0\kappa})$ is known but other parameters can vary from frame to frame, the number M of necessary views is such that $2M \geq 9$, or $M \geq 5$. If the skew s_κ is 0 in addition, this is relaxed to $3M \geq 9$, or $M \geq 3$ views. If furthermore the aspect ratio γ_κ is 1, this becomes $4M \geq 9$, so we still need to observe $M \geq 3$ views.

Remark 18 If we have more equations than the number of unknowns, inconsistencies arise among these equations in the presence of noise in the data. Theoretically, we can determine the unknowns in a statistically optimal ways [15], but this is too complicated to carry out. So, a simple least-squares minimization is conducted in practice. This, however, causes another problem: Ω_∞^* should have rank 3 from the definition of Eq. (37), but it has generally rank 4 if computed by least squares. Ad-hoc treatments, such as computing the singular value decomposition (SVD) of the obtained Ω_∞^* and replacing the smallest singular value by 0, are widely employed.

Remark 19 If Ω_∞^* is obtained, Eq. (37) does not completely determine the projective transformation H : it has rotational ambiguity, and its fourth column is arbitrary. This corresponds to the fact that the orientation and the location of the world coordinate system can be arbitrarily defined. The details of the computation is given in Appendix A.

Remark 20 From the computed Ω_∞^* , Eq. (36) determines ω_κ^* up to a scale factor. Then, Eq. (35) must be solved for K_κ , which should be an upper triangular matrix. A standard procedure, called the *Cholesky factorization*, is well known for decomposing a given positive semi-definite symmetric matrix into the product of an upper triangular matrix and its transpose. The indeterminate scale of K_κ is fixed so that its (33) element becomes 1.

Remark 21 The stratified reconstruction approach was proposed by Faugeras [4] and others. First, the constant camera constraint was used by many researchers. Later, Heyden and Åström [9, 10] showed that Euclidean reconstruction is possible using as few constraints as zero skew alone if a sufficient number of images and point correspondences are available. The constraint in the form of Eq. (36) was first formulated by Triggs [42]. Pollefeys et al. [28] demonstrated that accurate reconstruction is indeed possible by this approach. Since then, various modifications and simplifications have been devised for imposing the constraint. Many researchers used nonlinear optimization in one form or another, but later simple formulations using linear computations have been found in many forms; see [30, 31, 32]. The actual procedure of one such approach is given in Appendix A.

4.4. Dual Absolute Quadric Constraint

Comparing the second of Eqs. (16) and Eq. (35), we can see that the matrix ω_κ^* is the inverse of ω_κ , which represents the projection, onto the κ th image, of the absolute conic Ω_∞ . This means that the set of lines l that satisfy $l^\top \omega_\kappa^* l = 0$ is the *envelope* of, or the set of tangent lines to, the (imaginary) conic defined by the first of Eqs. (16). In projective geometry, this is called the *line pencil of second class* dual to the conic $u^\top \omega_\kappa u = 0$.

Eq. (36) states that the line pencil of second class represented by ω_κ^* is the projection, onto the κ th image, of the *plane pencil of second class* represented by Ω_∞^* , i.e., the set of planes with homogeneous coordinates π that satisfy $\pi^\top \Omega_\infty^* \pi = 0$. This is the envelope of, or the set of tangent planes to, the absolute conic Ω_∞ regarded as a degenerate (imaginary) quadric surface (a 2-D “disk”) (Fig. 9). This envelope is called the *dual absolute quadric*. From this projective geometric interpretation, Eq. (36) is called the *dual absolute quadric constraint*.

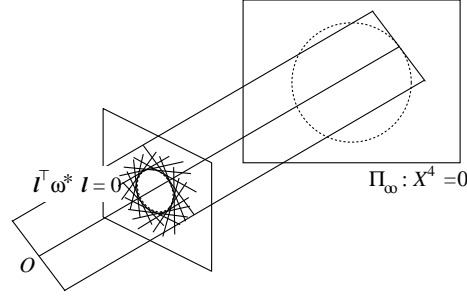


Figure 9. Dual absolute quadric constraint.

Remark 22 The fact that the constraint for Euclidean reconstruction can be given a projective geometric interpretation in terms of the dual absolute quadric is one of the greatest theoretical advances of 3-D reconstruction from images. For this reason, almost all papers, articles and books on 3-D reconstruction now start with theorems of projective geometry involving the absolute conic. At the cost of this elegance, however, this projective geometric interpretation makes the reconstruction procedure incomprehensible to average computer vision researchers, who tend to shy away from such mathematical sophistication involving imaginary quantities. In reality, the actual reconstruction procedure can be described without any reference to projective geometry, as we showed in Section 4.3. It is still being debated among researchers whether the projective geometric interpretation helps or prevents people's understanding of this method.

4.5. Projective Reconstruction

In order to start stratified reconstruction, we need an initial projective reconstruction. The most frequently used method for it is called *factorization*. If the projective depth $\lambda_{\kappa\alpha}$ is introduced as in Eq. (17), Eq. (31) is rewritten as the following equality:

$$\lambda_{\kappa\alpha} \mathbf{u}_{\kappa\alpha} = \mathbf{P}_{\kappa} \mathbf{X}_{\alpha}. \quad (39)$$

Let $\tilde{\mathbf{u}}_{\alpha}$ be the $3M$ -D vector obtained by vertically stacking $\lambda_{1\alpha} \mathbf{u}_{1\alpha}, \lambda_{2\alpha} \mathbf{u}_{2\alpha}, \dots, \lambda_{M\alpha} \mathbf{u}_{M\alpha}$, and $\tilde{\mathbf{p}}_i$ the $3M$ -D vector obtained by vertically stacking the i th columns of $\mathbf{P}_1, \mathbf{P}_2, \dots, \mathbf{P}_M$. Then, Eq. (39) is expressed in the form

$$\tilde{\mathbf{u}}_{\alpha} = X_{\alpha}^1 \tilde{\mathbf{p}}_1 + X_{\alpha}^2 \tilde{\mathbf{p}}_2 + X_{\alpha}^3 \tilde{\mathbf{p}}_3 + X_{\alpha}^4 \tilde{\mathbf{p}}_4, \quad (40)$$

where X_{α}^i is the i th component of the vector \mathbf{X}_{α} . Eq. (40) states that the N vectors $\tilde{\mathbf{u}}_{\alpha}$ are all constrained to be in the 4-D subspace \mathcal{L} of \mathcal{R}^{3M} spanned by $\{\tilde{\mathbf{p}}_1, \tilde{\mathbf{p}}_2, \tilde{\mathbf{p}}_3, \tilde{\mathbf{p}}_4\}$. This fact is called the *subspace constraint*.

We can see that Eq. (39) holds if we multiply both the projective depth $\lambda_{\kappa\alpha}$ and the homogeneous coordinates \mathbf{X}_{α} by a common nonzero constant c_{α} . As a result, the vector $\tilde{\mathbf{u}}_{\alpha}$ is multiplied by c_{α} . In order to remove this indeterminacy, we normalize $\tilde{\mathbf{u}}_{\alpha}$ to be a unit vector: $\|\tilde{\mathbf{u}}_{\alpha}\| = 1$. Then, we obtain the following iterative procedure for computing \mathbf{X}_{α} :

1. Give initial values for the projective depths $\lambda_{\kappa\alpha}$.

2. Compute the $3M$ -D vectors $\tilde{\mathbf{u}}_\alpha$ and fit a 4-D subspace \mathcal{L} to the resulting $\tilde{\mathbf{u}}_\alpha$ by least squares.
3. Adjust the projective depths $\lambda_{\kappa\alpha}$ so that the square distance from each $\tilde{\mathbf{u}}_\alpha$ to the fitted subspace \mathcal{L} is minimized.
4. Go back to Step 2, and repeat this until the computation converges.
5. Letting an arbitrary orthonormal basis of the converged subspace \mathcal{L} be $\tilde{\mathbf{p}}_i$, determine \mathbf{X}_α by expanding $\tilde{\mathbf{u}}_\alpha$ in the form of Eq. (40) by least squares.

Remark 23 In Step 1, the initial values of the projective depths $\lambda_{\kappa\alpha}$ can be set to 1. If all the cameras are “affine cameras” (to be defined in the next section), it can be shown that a solution such that $\lambda_{\kappa\alpha} = 1$ exists.

Remark 24 The least-squares solution in Step 2 can be immediately obtained by solving an eigenvalue problem. In fact, if we let

$$\mathbf{C} = \sum_{\alpha=1}^N \tilde{\mathbf{u}}_\alpha \tilde{\mathbf{u}}_\alpha^\top, \quad (41)$$

the subspace \mathcal{L} is spanned by the eigenvectors of \mathbf{C} for the largest four (positive) eigenvalues; the rest of the eigenvalues should vanish if the solution is exact. Alternatively, we may compute the singular value decomposition (SVD) in the form

$$(\tilde{\mathbf{u}}_1 \ \cdots \ \tilde{\mathbf{u}}_N) = \mathbf{U} \mathbf{\Lambda} \mathbf{V}^\top, \quad (42)$$

where \mathbf{U} is a $3M \times 3M$ orthogonal matrix, \mathbf{V} is a $N \times N$ orthogonal matrix, and $\mathbf{\Lambda}$ is a diagonal matrix. The diagonal elements of $\mathbf{\Lambda}$ consist of singular values in descending order; only four are nonzero if the solution is exact. The basis of the \mathcal{L} is given by the first four columns of \mathbf{U} . Usually, the use of SVD is computationally more efficient than the eigenvalue computation of Eq. (41).

Remark 25 The factorization approach to projective reconstruction was first introduced by Sturm and Triggs [34] and Triggs [41] with the observation that Eq. (39) for all κ and α can be rearranged in the form

$$\begin{pmatrix} \lambda_{11} \mathbf{u}_{11} & \cdots & \lambda_{1N} \mathbf{u}_{1N} \\ \vdots & \ddots & \vdots \\ \lambda_{M1} \mathbf{u}_{M1} & \cdots & \lambda_{MN} \mathbf{u}_{MN} \end{pmatrix} = \begin{pmatrix} \mathbf{P}_1 \\ \vdots \\ \mathbf{P}_M \end{pmatrix} (\mathbf{X}_1 \ \cdots \ \mathbf{X}_N). \quad (43)$$

In our notation, the vector $\tilde{\mathbf{u}}_\alpha$ is the α th column of the matrix on the left-hand side, and $\tilde{\mathbf{p}}_i$ is the i th column of the first matrix on the right-hand side. Sturm and Triggs [34] and Triggs [41] determined the projective depths $\lambda_{\kappa\alpha}$ so that the matrix on the left-hand side of Eq. (43) can be factorized into two matrices, hence the name “factorization”. To do this, they determined the projective depths $\lambda_{\kappa\alpha}$ by using the epipolar constraints (Section 3.3) on pairwise images, computing the fundamental matrices of image pairs in advance. See

Deguchi [2] for more details. Ueshiba and Tomita [43] did direct numerical search for $\lambda_{\kappa\alpha}$ based on the perturbation theorem of SVD. It was Heyden et al. [11] who explicitly stated the subspace constraint and reduced the problem to eigenvalue problem solving. However, they considered the space of the vectors constructed from “all projected points in each image”, rather than the vectors constructed from “each projected point in all images”, as in the above formulation. In this sense, their method is “dual” to the above treatment, which is based on Mahumud and Herbert [25]. Mahumud et al. [26] also presented an alternative update strategy.

Remark 26 In Step 3, it is easy to see that the square distance is a quadratic form in $\lambda_{\kappa\alpha}$ [25]. So, the solution that minimizes this subject to the normalization $\|\tilde{\mathbf{u}}_\alpha\|^2 = \sum_{\kappa=1}^M \|\mathbf{u}_{\kappa\alpha}\|^2 \lambda_{\kappa\alpha}^2 = 1$ is directly obtained by solving a generalized eigenvalue problem [15]. In Appendix B, the detailed procedure of Steps 1 – 5 (“primal method”) is described together with its dual form (“dual method”).

Remark 27 Iterations of Steps 2 – 4 are guaranteed to converge, because the sum of square distances of $\tilde{\mathbf{u}}_\alpha$ to the fitted subspace \mathcal{L} monotonically decreases due to the minimization in Step 3. This type of iteration is a special variant of the *EM algorithm* [3]. However, the convergence is, though guaranteed, very slow in general.

5. 3-D Reconstruction from Affine Cameras

5.1. Affine Cameras

In terms of homogeneous coordinates, perspective projection can be written as a linear equation in the form of Eq. (9), but this is in appearance only; the relationship is essentially nonlinear, as can be seen from Eq. (3), which makes the subsequent analysis very difficult. The analysis is made much easier if Eq. (3) is approximated by a linear relationship in the form

$$\begin{pmatrix} u \\ v \end{pmatrix} = \mathbf{\Pi} \begin{pmatrix} X_c \\ Y_c \\ Z_c \end{pmatrix} + \boldsymbol{\pi}, \quad (44)$$

where $\mathbf{\Pi}$ is a 2×3 matrix, $\boldsymbol{\pi}$ is a 2-D vector, and (X_c, Y_c, Z_c) is a point in the scene described with respect to the camera coordinate system. This approximation holds up to reasonable accuracy if

1. the object of our interest is localized around the world coordinate origin \mathbf{t} , and
2. the size of the object is small as compared with $\|\mathbf{t}\|$.

The approximate imaging geometry in the form of Eq. (44) is called an *affine camera*. Unlike the perspective camera model, the elements of the matrix $\mathbf{\Pi}$ and the vector $\boldsymbol{\pi}$ in Eq. (44) are now some functions of the motion parameters $\{\mathbf{R}, \mathbf{t}\}$. In order that Eq. (44) well mimic the perspective projection of Eq. (1), we require the following:

- (i) The camera imaging is symmetric around the Z -axis.

- (ii) The camera imaging does not depend on \mathbf{R} .
- (iii) The frontal parallel plane passing through the world coordinate origin is projected as if by perspective projection.

Requirement (i) states that if the scene is rotated around the optical axis by an angle θ , the resulting image should also rotate around the image origin by the same angle θ , a very natural requirement. Requirement (ii) is also natural, since the orientation of the world coordinate system can be defined arbitrarily, and such indeterminate parameterization should not affect the actual observation. Requirement (iii) corresponds to the assumption that the object of our interest is small and localized around the world coordinate origin \mathbf{t} . It can be shown that in order that Requirements (i) – (iii) be satisfied, Eq. (44) must have the following form [20]:

$$\begin{pmatrix} u \\ v \end{pmatrix} = \frac{1}{\zeta} \left(\begin{pmatrix} X_c \\ Y_c \end{pmatrix} + \beta(t_z - Z_c) \begin{pmatrix} t_x \\ t_y \end{pmatrix} \right). \quad (45)$$

Here, t_x , t_y , and t_z are the three components of \mathbf{t} , and $\{\zeta, \beta\}$ are arbitrary functions of $\sqrt{t_x^2 + t_y^2}$ and t_z ; function ζ determines the size of the projected image, while function β describes the deformation of the projection image as the point moves away from the plane $Z_c = t_z$. Typical examples are the following three (Fig. 10):

Orthographic projection

$$\zeta = 1, \quad \beta = 0. \quad (46)$$

Weak perspective (or scaled orthographic) projection [27, 40]

$$\zeta = \frac{t_z}{f_c}, \quad \beta = 0. \quad (47)$$

Paraperspective projection [27]

$$\zeta = \frac{t_z}{f_c}, \quad \beta = \frac{1}{t_z}. \quad (48)$$

Remark 28 The concept of affine camera and its epipolar geometry were presented by Shapiro et al. [33]. It was also shown that any affine camera can be interpreted to be paraperspective projection by appropriately adjusting the scale, the position, and the orientation of the world coordinate system [1]. This fact was exploited for object recognition from a single image [39]. The weak perspective and paraperspective models were introduced by Tomasi and Kanade [40] and Poelman and Kanade [27]. The generic form of Eq. (45) was derived by Kanatani et al. [20].

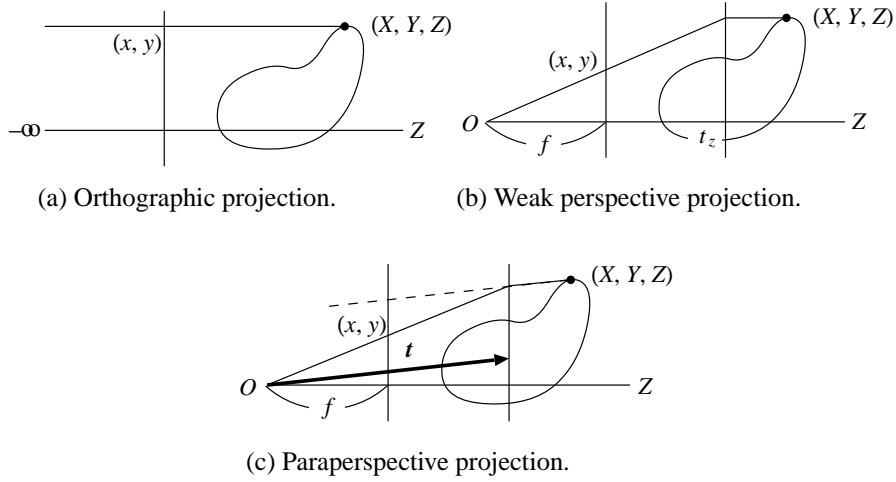


Figure 10. Affine camera models.

5.2. Affine Space Constraint

If a point in the scene is represented by a vector \mathbf{X}_α of homogeneous coordinates with the fourth component 1, Eqs. (6) and (44) imply that its projection onto the κ th image is represented by a vector $\mathbf{u}_{\kappa\alpha}$ with the third component 1 in the form

$$\mathbf{u}_{\kappa\alpha} = \begin{pmatrix} \mathbf{\Pi}_\kappa \mathbf{R}_\kappa & \mathbf{\Pi}_\kappa \mathbf{t}_\kappa + \boldsymbol{\pi}_\kappa \\ 0 & 0 & 0 & 1 \end{pmatrix} \mathbf{X}_\alpha, \quad (49)$$

where $\mathbf{\Pi}_\kappa$ and $\boldsymbol{\pi}_\kappa$ are, respectively, the values of the matrix $\mathbf{\Pi}$ and the vector $\boldsymbol{\pi}$ in Eq. (44) for the κ th image, and $\{\mathbf{R}_\kappa, \mathbf{t}_\kappa\}$ are the motion parameters of the κ th camera. Eq. (49) shows that an affine camera is a special case of the general projection in the form of Eq. (39) with the conditions that

- the third row of the projection matrix \mathbf{P}_κ is $(0 \ 0 \ 0 \ 1)$, and
- the projective depths $\lambda_{\kappa\alpha}$ are all 1 (Remark 23).

It follows that, corresponding to Eq. (40), the following relationship holds:

$$\tilde{\mathbf{u}}_\alpha = X_\alpha \tilde{\mathbf{p}}_1 + Y_\alpha \tilde{\mathbf{p}}_2 + Z_\alpha \tilde{\mathbf{p}}_3 + \tilde{\mathbf{p}}_4. \quad (50)$$

As in Section 4.5, $\tilde{\mathbf{u}}_\alpha$ is a vector, which we call the *trajectory* of the α th point, obtained by vertically stacking $\mathbf{u}_{1\alpha}, \mathbf{u}_{2\alpha}, \dots, \mathbf{u}_{M\alpha}$, while $\tilde{\mathbf{p}}_i$ is a vector obtained by vertically stacking the i th columns of the matrix on the right-hand side of Eq. (49) for $\kappa = 1, \dots, M$. We call $\tilde{\mathbf{p}}_1, \dots, \tilde{\mathbf{p}}_4$ the *motion vectors*. We can see that every third component of the vector equation in Eq. (50) gives the identity $1 = 1$, so they can be removed. As a result, all the trajectories $\tilde{\mathbf{u}}_\alpha$ and the motion vectors $\tilde{\mathbf{p}}_i$ become $2M$ -D vectors. Eq. (50) states that all the trajectories $\tilde{\mathbf{u}}_\alpha$ are constrained to be in the *3-D affine space* \mathcal{A} of \mathcal{R}^{2M} passing through $\tilde{\mathbf{p}}_4$ and spanned by the motion vectors $\{\tilde{\mathbf{p}}_1, \tilde{\mathbf{p}}_2, \tilde{\mathbf{p}}_3\}$. This fact is called the *affine space constraint*.

Remark 29 The affine space constraint is not only a basis for 3-D reconstruction from affine camera images but also the core principle of *multibody motion segmentation* from images.

This is because if we observe multiple objects that are moving independently in the scene, the affine space constraint should hold for each rigid motion. Hence, if we track feature points that belong to multiple objects, classifying them into different motions is equivalent to classifying their trajectories, regarded as $2M$ -D vectors, into different affine spaces in \mathcal{R}^{2M} . See [16, 17, 18, 35, 36, 37, 38] for actual applications.

5.3. Affine Reconstruction and Metric Constraints

The standard procedure for 3-D reconstruction based on the affine space constraint is called *factorization* for the reason explained shortly.

First, we fit a 3-D affine space \mathcal{A} to the trajectories $\tilde{\mathbf{u}}_\alpha$ by least squares. It is specified by a particular point $\tilde{\mathbf{p}}_C \in \mathcal{A}$ and orthonormal vectors $\{\tilde{\mathbf{q}}_1, \tilde{\mathbf{q}}_2, \tilde{\mathbf{q}}_3\}$ that span \mathcal{A} at $\tilde{\mathbf{p}}_C$. If we identify $\{\tilde{\mathbf{q}}_1, \tilde{\mathbf{q}}_2, \tilde{\mathbf{q}}_3\}$ with $\{\tilde{\mathbf{p}}_1, \tilde{\mathbf{p}}_2, \tilde{\mathbf{p}}_3\}$ in Eq. (50), we can determine $(X_\alpha, Y_\alpha, Z_\alpha)$ by expanding each $\tilde{\mathbf{u}}_\alpha$ over them in the same way we did in Section 4.5. However, $\tilde{\mathbf{p}}_C$ can be anywhere in \mathcal{A} , and $\{\tilde{\mathbf{q}}_1, \tilde{\mathbf{q}}_2, \tilde{\mathbf{q}}_3\}$ can be any three linearly independent vectors, not necessarily orthonormal. Hence, the 3-D shape reconstructed from $\tilde{\mathbf{p}}_C$ and $\{\tilde{\mathbf{q}}_1, \tilde{\mathbf{q}}_2, \tilde{\mathbf{q}}_3\}$ has ambiguity up to an affine transformation. Such a reconstruction is called *affine reconstruction*. In order to upgrade the solution to Euclidean, we need to rectify the basis correctly by an affine transformation in the form

$$\tilde{\mathbf{p}}_i = \sum_{j=1}^3 A_{ji} \tilde{\mathbf{q}}_j. \quad (51)$$

The translational ambiguity due to the arbitrariness of $\tilde{\mathbf{p}}_C$ has no effect on the reconstructed 3-D shape. The rectifying transformation matrix $\mathbf{A} = (A_{ij})$ is determined by the condition that each $\tilde{\mathbf{p}}_i$ consists of coordinates of points in the scene viewed by an affine camera that has the form of Eq. (49). This condition, known as the *metric constraint*, is obtained, as in the case of the dual absolute quadric constraint, by eliminating \mathbf{R}_κ from the projection relation of Eq. (49) by using the identity $\mathbf{R}_\kappa \mathbf{R}_\kappa^\top = \mathbf{I}$.

Let \mathbf{Q} be the $2M \times 3$ matrix with columns $\tilde{\mathbf{q}}_1, \tilde{\mathbf{q}}_2,$ and $\tilde{\mathbf{q}}_3$ in that order. Let $\mathbf{q}_{\kappa(1)}^\dagger$ and $\mathbf{q}_{\kappa(2)}^\dagger$ be the $(2\kappa - 1)$ th and the 2κ th columns of \mathbf{Q}^\top , respectively. We define the 3×2 matrix $\mathbf{Q}_\kappa^\dagger$ by

$$\mathbf{Q}_\kappa^\dagger = \begin{pmatrix} \mathbf{q}_{\kappa(1)}^\dagger & \mathbf{q}_{\kappa(2)}^\dagger \end{pmatrix}. \quad (52)$$

It can be shown (see Appendix C for the derivation) that if we let

$$\mathbf{T} = \mathbf{A} \mathbf{A}^\top, \quad (53)$$

the metric constraint is written in the following form [20]:

$$\mathbf{Q}_\kappa^{\dagger\top} \mathbf{T} \mathbf{Q}_\kappa^\dagger = \mathbf{\Pi}_\kappa \mathbf{\Pi}_\kappa^\top. \quad (54)$$

As in the stratified reconstruction, we can obtain from Eq. (54) a set of equations for \mathbf{T} from the knowledge about the camera model, i.e., the relationships among the elements of the matrix $\mathbf{\Pi}_\kappa \mathbf{\Pi}_\kappa^\top$ on the right-hand side of Eq. (54). After that, we can obtain the rectifying matrix \mathbf{A} by decomposing the computed \mathbf{T} in the form of Eq. (53). The computational details for the typical models of Eqs. (46) – (48) and the general affine camera model of Eq. (45) are described in Appendix C.

Remark 30 As mentioned in Section 5.1, the affine camera model is a good approximation when the object of our interest is localized around the world coordinate origin. In such a situation, the world coordinate origin (which can be defined arbitrarily) can be located at the centroid of points $(X_\alpha, Y_\alpha, Z_\alpha)$, which means

$$\sum_{\alpha=1}^N X_\alpha = \sum_{\alpha=1}^N Y_\alpha = \sum_{\alpha=1}^N Z_\alpha = 0. \quad (55)$$

Let $\tilde{\mathbf{u}}_C$ be the centroid of the trajectories $\tilde{\mathbf{u}}_\alpha$:

$$\tilde{\mathbf{u}}_C = \frac{1}{N} \sum_{\alpha=1}^N \tilde{\mathbf{u}}_\alpha. \quad (56)$$

From Eqs. (50) and (55), we can see that the centroid $\tilde{\mathbf{u}}_C$ coincide with $\tilde{\mathbf{p}}_4$: $\tilde{\mathbf{u}}_C = \tilde{\mathbf{p}}_4$. As in the case of stratified reconstruction, the basis of the affine space \mathcal{A} that optimally fits the trajectories $\tilde{\mathbf{u}}_\alpha$ and passes through their centroid $\tilde{\mathbf{u}}_C$ is given by the eigenvectors of the matrix

$$\mathbf{C} = \sum_{\alpha=1}^N (\tilde{\mathbf{u}}_\alpha - \tilde{\mathbf{u}}_C)(\tilde{\mathbf{u}}_\alpha - \tilde{\mathbf{u}}_C)^\top, \quad (57)$$

for the largest three eigenvalues. Alternatively, we may compute the singular value decomposition (SVD) in the form

$$(\tilde{\mathbf{u}}_1 - \tilde{\mathbf{u}}_C \quad \cdots \quad \tilde{\mathbf{u}}_N - \tilde{\mathbf{u}}_C) = \mathbf{U} \mathbf{\Lambda} \mathbf{V}^\top, \quad (58)$$

where \mathbf{U} is a $2M \times 2M$ orthogonal matrix, \mathbf{V} is a $N \times N$ orthogonal matrix, and $\mathbf{\Lambda}$ is a diagonal matrix. The basis of the \mathcal{A} is given by the first three columns \mathbf{U} .

Remark 31 If we let $\tilde{\mathbf{u}}'_\alpha = \tilde{\mathbf{u}}'_\alpha - \tilde{\mathbf{u}}_C$, Eq. (50) for $\alpha = 1, \dots, N$ can be rearranged in the following form:

$$(\tilde{\mathbf{u}}'_1 \quad \cdots \quad \tilde{\mathbf{u}}'_N) = (\tilde{\mathbf{p}}_1 \quad \tilde{\mathbf{p}}_2 \quad \tilde{\mathbf{p}}_3) \begin{pmatrix} X_1 & Y_1 & Z_1 \\ \vdots & \vdots & \vdots \\ X_N & Y_N & Z_N \end{pmatrix}. \quad (59)$$

Hence, computing the solution $\{X_\alpha, Y_\alpha, Z_\alpha\}$ can be given the interpretation that we are *factorizing the measurement (or observation) matrix* $\mathbf{W} = (\tilde{\mathbf{u}}'_1 \quad \cdots \quad \tilde{\mathbf{u}}'_N)$ into the product of two matrices: the first describes the motion; the second the shape. This is the origin of the term *factorization*, named by Tomasi and Kanade [40], and the subsequent papers [24, 27] adopt this interpretation. Sturm and Triggs [34] and Triggs [41] presented a projective reconstruction procedure in a similar formalism, and this lead to the term ‘‘factorization’’ also for the approach described in Section 4.5 (Remark 25).

Remark 32 Since the factorization gives the solution by linear computation alone without any iterative search, it is widely used for many applications, such as object recognition and classification, which do not require so very high accuracy of the 3-D shape. Also, this method can be used to obtain a good initial guess of projective reconstruction for the stratified reconstruction.

Remark 33 When we say that we obtain “affine reconstruction” if the metric constraint is not imposed, we must keep in mind that an affine camera is a hypothetical concept; it only approximates existing cameras, which are modeled as perspective projection. Hence, if we use perspectively projected images as input, the resulting shape is not exactly affine reconstruction and is not exactly Euclidean even if the metric constraint is imposed.

Remark 34 The 3-D shape reconstructed by factorization is not unique, having the following ambiguity:

- (i) The absolute scale is indeterminate.
- (ii) The orientation of the world coordinate system is indeterminate.
- (iii) Mirror image ambiguity exists.

The absolute scale indeterminacy is unavoidable as long as images are only available information (Remark 13). In fact, we can see from Eq. (50) that multiplying $\{\tilde{\mathbf{p}}_1, \tilde{\mathbf{p}}_2, \tilde{\mathbf{p}}_3\}$ by a nonzero constant c gives rise to the same effect as dividing $\{X_\alpha, Y_\alpha, Z_\alpha\}$ by c . The orientation of the world coordinate system is indeterminate, because it can be arbitrarily defined in the scene. The mirror image ambiguity arises from the fact that the rectifying matrix \mathbf{A} is determined by Eq. (53), which can be rewritten as $\mathbf{T} = (\pm\mathbf{AR})(\pm\mathbf{AR})^\top$ for an arbitrary rotation matrix \mathbf{R} . The indeterminacy of the rotation \mathbf{R} corresponds to the orientation ambiguity; the indeterminacy of the sign corresponds to the mirror image ambiguity.

6. Concluding Remarks

This chapter has summarized recent advancements of the theories and techniques for 3-D reconstruction from multiple images. We started with the description of the camera imaging geometry as perspective projection in terms of homogeneous coordinates. We defined the intrinsic and extrinsic (motion) parameters of the camera by introducing the camera coordinate system and the world coordinate system.

It was shown that the camera imaging is regarded as a mapping from the 3-D projective space \mathcal{P}^3 onto the 2-D projective space \mathcal{P}^2 and that the absolute conic is invariant to camera motions, providing projective geometric interpretations to camera calibration procedures. Next, we described the epipolar geometry for two, three, and four cameras, introducing such concepts as the fundamental matrix, epipolar lines, epipoles, the trifocal tensor, and the quadrifocal tensor.

We then described the self-calibration technique based on the stratified reconstruction approach, using the absolute dual quadric constraint. We showed that an elegant projective geometric interpretation can be given but that it is not essential or even necessary for actually doing 3-D reconstruction computations. We also described the procedure for computing a projective reconstruction by the factorization based on the subspace constraint.

Finally, we gave the definition of the affine camera model and a procedure for 3-D reconstruction based on it. We discussed possible forms of the affine camera, described the affine space constraint, and introduced the metric constraint that is necessary for Euclidean reconstruction. The detailed procedures for 3-D reconstruction are given in the Appendix.

A. Euclidean Upgrading from Projective Reconstruction

Here, we describe the computational procedure for computing Ω_∞^* in Eq. (35) and H in Eq. (37), given the projection matrices \tilde{P}_κ of projective reconstruction (the projective reconstruction procedure is given in Appendix B). We then give the procedure for computing the 3-D shape X_α and the motion parameters $\{\mathbf{R}_\kappa, \mathbf{t}_\kappa\}$ using the computed projective transformation H .

The following is a modification of the scheme proposed by the method of Seo and Heyden [31], to which several techniques are introduced for increasing robustness. The basic assumption here is that the skew angles θ_κ are all 0 and the aspect ratios γ_κ are all 1 (Sections 2.2 and 2.3). Hence, unknown camera parameters are the focal lengths f_κ and the principal points $(u_{\kappa 0}, v_{\kappa 0})$ for M frames.

A.1. Computation of Ω

Substituting Eq. (35) into Eq. (36), we have

$$\tilde{P}_\kappa \Omega_\infty^* \tilde{P}_\kappa^\top \simeq \mathbf{K}_\kappa \mathbf{K}_\kappa^\top. \quad (60)$$

Suppose we have an estimate f_κ of the focal length and an estimate $(u_{\kappa 0}, v_{\kappa 0})$ of the principal point for each frame. We tentatively let

$$\mathbf{K}_\kappa = \begin{pmatrix} f_\kappa & 0 & u_{\kappa 0} \\ 0 & f_\kappa & v_{\kappa 0} \\ 0 & 0 & 1 \end{pmatrix}. \quad (61)$$

(See Eq. (4). Keep in mind that we are assuming that the skew angle is 0 and the aspect ratio is 1). Multiplying Eq. (60) by \mathbf{K}_κ^{-1} from left and $\mathbf{K}_\kappa^{-1\top}$ from right, we have

$$\mathbf{Q}_\kappa \Omega_\infty^* \mathbf{Q}_\kappa^\top \approx \text{scalar} \times \mathbf{I}, \quad (62)$$

where we define

$$\mathbf{Q}_\kappa \equiv \mathbf{K}_\kappa^{-1} \tilde{P}_\kappa. \quad (63)$$

Eq. (62) implies that the (11) and (22) elements of $\mathbf{Q}_\kappa \Omega_\infty^* \mathbf{Q}_\kappa^\top$ are approximately equal, and its (12), (23), and (31) elements are approximately 0. Namely,

$$\sum_{i,j=1}^4 Q_{\kappa(1i)} Q_{\kappa(1j)} \Omega_{\infty^*(ij)}^* - \sum_{i,j=1}^4 Q_{\kappa(2i)} Q_{\kappa(2j)} \Omega_{\infty^*(ij)}^* \approx 0, \quad (64)$$

$$\sum_{i,j=1}^4 Q_{\kappa(1i)} Q_{\kappa(2j)} \Omega_{\infty^*(ij)}^* \approx 0, \quad (65)$$

$$\sum_{i,j=1}^4 Q_{\kappa(2i)} Q_{\kappa(3j)} \Omega_{\infty^*(ij)}^* \approx 0, \quad (66)$$

$$\sum_{i,j=1}^4 Q_{\kappa(3i)} Q_{\kappa(1j)} \Omega_{\infty}^*(ij) \approx 0, \quad (67)$$

where $Q_{\kappa(ij)}$ and $\Omega_{\infty}^*(ij)$ are the (ij) elements of the matrices \mathbf{Q}_{κ} and Ω_{∞}^* , respectively. We determine Ω_{∞}^* by minimizing

$$\begin{aligned} K &= \sum_{\kappa=1}^M W_{\kappa} \left((\text{Eq. (64)})^2 + (\text{Eq. (65)})^2 + (\text{Eq. (66)})^2 + (\text{Eq. (64)})^2 \right) \\ &= \sum_{i,j,k,l=1}^4 A_{ijkl} \Omega_{\infty}^*(ij) \Omega_{\infty}^*(kl), \end{aligned} \quad (68)$$

where W_{κ} is an appropriate weight (initially we set $W_{\kappa} = 1$). The $3 \times 3 \times 3 \times 3$ tensor $\mathcal{A} = (A_{ijkl})$ has the form

$$\begin{aligned} A_{ijkl} &= \sum_{\kappa=1}^M W_{\kappa} \left(Q_{\kappa(1i)} Q_{\kappa(1j)} Q_{\kappa(1k)} Q_{\kappa(1l)} - Q_{\kappa(1i)} Q_{\kappa(1j)} Q_{\kappa(2k)} Q_{\kappa(2l)} \right. \\ &\quad - Q_{\kappa(2i)} Q_{\kappa(2j)} Q_{\kappa(1k)} Q_{\kappa(1l)} + Q_{\kappa(2i)} Q_{\kappa(2j)} Q_{\kappa(2k)} Q_{\kappa(2l)} + \frac{1}{4} (Q_{\kappa(1i)} Q_{\kappa(2j)} Q_{\kappa(1k)} Q_{\kappa(2l)} \\ &\quad + Q_{\kappa(2i)} Q_{\kappa(1j)} Q_{\kappa(1k)} Q_{\kappa(2l)} + Q_{\kappa(1i)} Q_{\kappa(2j)} Q_{\kappa(2k)} Q_{\kappa(1l)} + Q_{\kappa(2i)} Q_{\kappa(1j)} Q_{\kappa(2k)} Q_{\kappa(1l)}) \\ &\quad + \frac{1}{4} (Q_{\kappa(2i)} Q_{\kappa(3j)} Q_{\kappa(2k)} Q_{\kappa(3l)} + Q_{\kappa(3i)} Q_{\kappa(2j)} Q_{\kappa(2k)} Q_{\kappa(3l)} + Q_{\kappa(2i)} Q_{\kappa(3j)} Q_{\kappa(3k)} Q_{\kappa(2l)} \\ &\quad + Q_{\kappa(3i)} Q_{\kappa(2j)} Q_{\kappa(3k)} Q_{\kappa(2l)}) + \frac{1}{4} (Q_{\kappa(3i)} Q_{\kappa(1j)} Q_{\kappa(3k)} Q_{\kappa(1l)} + Q_{\kappa(1i)} Q_{\kappa(3j)} Q_{\kappa(3k)} Q_{\kappa(1l)} \\ &\quad \left. + Q_{\kappa(3i)} Q_{\kappa(1j)} Q_{\kappa(1k)} Q_{\kappa(3l)} + Q_{\kappa(1i)} Q_{\kappa(3j)} Q_{\kappa(1k)} Q_{\kappa(3l)}) \right). \end{aligned} \quad (69)$$

The absolute scale of Ω_{∞}^* cannot be determined from Eq. (62), so we tentatively adopt normalization $\sum_{i,j=1}^4 \Omega_{\infty}^{*2}(ij) = 1$. Since Ω_{∞}^* is a symmetric matrix, we can write

$$\Omega_{\infty}^* = \begin{pmatrix} w_1 & w_5/\sqrt{2} & w_6/\sqrt{2} & w_7/\sqrt{2} \\ w_5/\sqrt{2} & w_2 & w_8/\sqrt{2} & w_9/\sqrt{2} \\ w_6/\sqrt{2} & w_8/\sqrt{2} & w_3 & w_{10}/\sqrt{2} \\ w_7/\sqrt{2} & w_9/\sqrt{2} & w_{10}/\sqrt{2} & w_4 \end{pmatrix}. \quad (70)$$

Then, the normalization $\sum_{i,j=1}^4 \Omega_{\infty}^{*2}(ij) = 1$ is equivalent to $\sum_{i=1}^{10} w_i^2 = 1$. If we define the 10×10 matrix

$$\mathbf{A}^{\dagger} = \begin{pmatrix} A_{1111} & A_{1122} & A_{1133} & A_{1144} & \sqrt{2}A_{1112} \\ A_{2211} & A_{2222} & A_{2233} & A_{2244} & \sqrt{2}A_{2212} \\ A_{3311} & A_{3322} & A_{3333} & A_{3344} & \sqrt{2}A_{3312} \\ A_{4411} & A_{4422} & A_{4433} & A_{4444} & \sqrt{2}A_{4412} \\ \sqrt{2}A_{1211} & \sqrt{2}A_{1222} & \sqrt{2}A_{1233} & \sqrt{2}A_{1244} & 2A_{1212} \\ \sqrt{2}A_{1311} & \sqrt{2}A_{1322} & \sqrt{2}A_{1333} & \sqrt{2}A_{1344} & 2A_{1312} \\ \sqrt{2}A_{1411} & \sqrt{2}A_{1422} & \sqrt{2}A_{1433} & \sqrt{2}A_{1444} & 2A_{1412} \\ \sqrt{2}A_{2311} & \sqrt{2}A_{2322} & \sqrt{2}A_{2333} & \sqrt{2}A_{2344} & 2A_{2312} \\ \sqrt{2}A_{2411} & \sqrt{2}A_{2422} & \sqrt{2}A_{2433} & \sqrt{2}A_{2444} & 2A_{2412} \\ \sqrt{2}A_{3411} & \sqrt{2}A_{3422} & \sqrt{2}A_{3433} & \sqrt{2}A_{3444} & 2A_{3412} \end{pmatrix}$$

$$\begin{pmatrix} \sqrt{2}A_{1113} & \sqrt{2}A_{1114} & \sqrt{2}A_{1123} & \sqrt{2}A_{1124} & \sqrt{2}A_{1134} \\ \sqrt{2}A_{2213} & \sqrt{2}A_{2214} & \sqrt{2}A_{2223} & \sqrt{2}A_{2224} & \sqrt{2}A_{2234} \\ \sqrt{2}A_{3313} & \sqrt{2}A_{3314} & \sqrt{2}A_{3323} & \sqrt{2}A_{3324} & \sqrt{2}A_{3334} \\ \sqrt{2}A_{4413} & \sqrt{2}A_{4414} & \sqrt{2}A_{4423} & \sqrt{2}A_{4424} & \sqrt{2}A_{4434} \\ 2A_{1213} & 2A_{1214} & 2A_{1223} & 2A_{1224} & 2A_{1234} \\ 2A_{1313} & 2A_{1314} & 2A_{1323} & 2A_{1324} & 2A_{1334} \\ 2A_{1413} & 2A_{1414} & 2A_{1423} & 2A_{1424} & 2A_{1434} \\ 2A_{2313} & 2A_{2314} & 2A_{2323} & 2A_{2324} & 2A_{2334} \\ 2A_{2413} & 2A_{2414} & 2A_{2423} & 2A_{2424} & 2A_{2434} \\ 2A_{3413} & 2A_{3414} & 2A_{3423} & 2A_{3424} & 2A_{3434} \end{pmatrix}, \quad (71)$$

Eq. (68) is written as

$$K = \sum_{i,j=1}^{10} A_{ij}^\dagger w_i w_j. \quad (72)$$

Hence, minimization of Eq. (68) subject to $\sum_{i,j=1}^4 \Omega_\infty^{*2}(ij) = 1$ reduces to minimization of Eq. (72) subject to $\sum_{i=1}^{10} w_i^2 = 1$. The solution is given by the unit eigenvector $\mathbf{w} = (w_i)$ of the matrix $\mathbf{A}^\dagger = (A_{ij}^\dagger)$ (alternatively, we can use SVD, but explicit expressions are cumbersome to write down). The computed $\mathbf{w} = (w_i)$ is then converted to a 4×4 matrix in the form of Eq. (70). However, the sign of the eigenvector \mathbf{w} , hence of Ω_∞^* , is indeterminate. Also, Ω_∞^* must be positive-semi definite with rank 3. So, we redefine Ω_∞^* as follows. Let $\sigma_1 \geq \dots \geq \sigma_4$ be the eigenvalues of Ω_∞^* , and $\mathbf{u}_1, \dots, \mathbf{u}_4$ the corresponding unit eigenvectors. We let

$$\Omega_\infty^* = \begin{cases} \sigma_1 \mathbf{u}_1 \mathbf{u}_1^\top + \sigma_2 \mathbf{u}_2 \mathbf{u}_2^\top + \sigma_3 \mathbf{u}_3 \mathbf{u}_3^\top & \sigma_3 > 0 \\ -\sigma_4 \mathbf{u}_4 \mathbf{u}_4^\top - \sigma_3 \mathbf{u}_3 \mathbf{u}_3^\top - \sigma_2 \mathbf{u}_2 \mathbf{u}_2^\top & \sigma_2 < 0 \end{cases}. \quad (73)$$

A.2. Update of \mathbf{K}_κ

Suppose the left-hand side of Eq. (62) for the computed Ω_∞^* has the form

$$\mathbf{Q}_\kappa \Omega_\infty^* \mathbf{Q}_\kappa^\top = \begin{pmatrix} c_{\kappa(11)} & c_{\kappa(12)} & c_{\kappa(13)} \\ c_{\kappa(21)} & c_{\kappa(22)} & c_{\kappa(23)} \\ c_{\kappa(31)} & c_{\kappa(32)} & c_{\kappa(33)} \end{pmatrix}. \quad (74)$$

If this is not a scalar multiple of the unit matrix \mathbf{I} , we update \mathbf{K}_κ in the form of $\mathbf{K}_\kappa \leftarrow \delta \mathbf{K}_\kappa \mathbf{K}_\kappa$, where we let

$$\delta \mathbf{K}_\kappa = \begin{pmatrix} \delta f_\kappa & 0 & \delta u_{\kappa 0} \\ 0 & \delta f_\kappa & \delta v_{\kappa 0} \\ 0 & 0 & 1 \end{pmatrix}. \quad (75)$$

The increment $\delta \mathbf{K}_\kappa$ is determined in such a way that Eq. (74) is approximated by $\delta \mathbf{K}_\kappa \delta \mathbf{K}_\kappa^\top$. From Eqs. (74) and (75), we find that

$$\delta u_{\kappa 0} = \frac{c_{\kappa(13)}}{c_{\kappa(33)}}, \quad \delta v_{\kappa 0} = \frac{c_{\kappa(23)}}{c_{\kappa(33)}},$$

$$\delta f_\kappa = \sqrt{\frac{1}{2} \left(\frac{c_{\kappa(11)} + c_{\kappa(22)}}{c_{\kappa(33)}} - \delta u_{\kappa 0}^2 - \delta v_{\kappa 0}^2 \right)}. \quad (76)$$

Since the projection matrix $\tilde{\mathbf{P}}_\kappa$ can be defined only up to scalar multiplication (see Eq. (31)), the matrix \mathbf{Q}_κ in Eq. (63) also has scale indeterminacy. So, we normalize \mathbf{Q}_κ by dividing it by $\sqrt{c_{\kappa(33)}}$ so that Eq. (74) has approximately the same scale as \mathbf{I} for all κ . However, $c_{\kappa(33)}$ can be negative in the presence of extremely large noise, and the inside of the square root in Eqs. (76) may also become negative. In such a case, we skip that frame in the computation. To do this systematically, we make the weight W_κ reflect the closeness of Eq. (74) to a scalar multiple of \mathbf{I} . We also measure the goodness of estimation not by totaling the goodness measures of individual frames but by their “median” so that exceptional frames are not counted (see Section A.4).

A.3. Computation of \mathbf{H}

Since $\mathbf{\Omega}_\infty^*$ has the form of Eq. (73), a 4×4 matrix \mathbf{H} that satisfies Eq. (37) is given up to a rotation by $(\sqrt{\sigma_1}\mathbf{u}_1 \ \sqrt{\sigma_2}\mathbf{u}_2 \ \sqrt{\sigma_3}\mathbf{u}_3 \ \mathbf{v})$ for $\sigma_3 > 0$ and $(\sqrt{-\sigma_4}\mathbf{u}_4 \ \sqrt{-\sigma_3}\mathbf{u}_3 \ \sqrt{-\sigma_2}\mathbf{u}_2 \ \mathbf{v})$ for $\sigma_2 < 0$, where \mathbf{v} is an arbitrary vector. The indeterminate freedom of rotation and the arbitrariness of the vector \mathbf{v} correspond to the fact that the orientation and the location of the world coordinate system are arbitrary. However, the matrix \mathbf{H} must be nonsingular, which means that \mathbf{v} must be linearly independent of the first, the second, and the third columns of \mathbf{H} . So, we choose as \mathbf{v} a unit vector orthogonal to them. This means that we can take as \mathbf{v} the remaining unit eigenvector of $\mathbf{\Omega}_\infty^*$.

A.4. Computational Procedure

The above computation is summarized as follows:

Input:

- Approximate principal points $(u_{\kappa 0}, v_{\kappa 0})$ and the focal lengths f_κ , $\kappa = 1, \dots, M$.
- Projection matrices $\tilde{\mathbf{P}}_\kappa$, $\kappa = 1, \dots, M$.

Output:

- Rectifying projective transformation \mathbf{H} .
- Intrinsic parameter matrices \mathbf{K}_κ , $\kappa = 1, \dots, M$.

Computation:

1. Let

$$\hat{\mathbf{H}} = \mathbf{I}_{4 \times 4}, \quad \hat{\mathbf{K}} = \mathbf{I}_{3 \times 3}, \quad \hat{J}_{\text{med}} = \infty, \quad (77)$$

where the subscript of \mathbf{I} denotes its size (omitted if understood), and ∞ means a sufficiently large number.

2. Initialize \mathbf{K}_κ in the form of Eq. (61), and let $W_\kappa = 1$ and $\gamma_\kappa = 1$.

3. Let

$$\mathbf{Q}_\kappa = \gamma_\kappa \mathbf{K}_\kappa^{-1} \tilde{\mathbf{P}}_\kappa. \quad (78)$$

4. Compute the tensor $\mathcal{A} = (A_{ijkl})$ in Eq. (69).

5. Compute the 10-D unit eigenvector \mathbf{w} of the 10×10 matrix \mathbf{A} in Eq. (71) for the smallest eigenvalue.

6. Compute the tentative matrix $\mathbf{\Omega}_\infty^*$ in Eq. (70).

7. Compute the eigenvalues $\sigma_1 \geq \dots \geq \sigma_4$ of $\mathbf{\Omega}_\infty^*$ and the corresponding unit eigenvectors $\mathbf{u}_1, \dots, \mathbf{u}_4$.

8. Compute

$$\mathbf{H} = \begin{cases} \begin{pmatrix} \sqrt{\sigma_1} \mathbf{u}_1 & \sqrt{\sigma_2} \mathbf{u}_2 & \sqrt{\sigma_3} \mathbf{u}_3 & \mathbf{u}_4 \\ \sqrt{-\sigma_4} \mathbf{u}_4 & \sqrt{-\sigma_3} \mathbf{u}_3 & \sqrt{-\sigma_2} \mathbf{u}_2 & \mathbf{u}_1 \end{pmatrix} & \sigma_3 > 0 \\ \begin{pmatrix} \sqrt{-\sigma_4} \mathbf{u}_4 & \sqrt{-\sigma_3} \mathbf{u}_3 & \sqrt{-\sigma_2} \mathbf{u}_2 & \mathbf{u}_1 \end{pmatrix} & \sigma_2 < 0 \end{cases}. \quad (79)$$

9. Do the following computation for $\kappa = 1, \dots, M$:

(a) Compute $c_{\kappa(ij)}$ by Eq. (74), and let

$$F_\kappa = \frac{c_{\kappa(11)} + c_{\kappa(22)}}{c_{\kappa(33)}} - \left(\frac{c_{\kappa(13)}}{c_{\kappa(33)}} \right)^2 - \left(\frac{c_{\kappa(23)}}{c_{\kappa(33)}} \right)^2. \quad (80)$$

(b) If $c_{\kappa(33)} > 0$ and $F_\kappa > 0$, compute $\delta u_{\kappa 0}$, $\delta v_{\kappa 0}$, and δf_κ in Eqs. (76) and let

$$J_\kappa = \left(\frac{c_{\kappa(11)}}{c_{\kappa(33)}} - 1 \right)^2 + \left(\frac{c_{\kappa(22)}}{c_{\kappa(33)}} - 1 \right)^2 + 2 \frac{c_{\kappa(12)}^2 + c_{\kappa(23)}^2 + c_{\kappa(31)}^2}{c_{\kappa(33)}^2}. \quad (81)$$

Then, update \mathbf{K}_κ and γ_κ as follows:

$$\mathbf{K}_\kappa \leftarrow \mathbf{K}_\kappa \delta \mathbf{K}_\kappa, \quad \gamma_\kappa \leftarrow \frac{\gamma_\kappa}{\sqrt{c_{\kappa(33)}}}. \quad (82)$$

(c) Else, let $J_\kappa = \infty$.

10. Compute the following median:

$$J_{\text{med}} = \text{med}_{\kappa=1}^M J_\kappa. \quad (83)$$

11. If $J_{\text{med}} \approx 0$, return \mathbf{H} and \mathbf{K}_κ and stop.

12. If $J_{\text{med}} \geq \hat{J}_{\text{med}}$, return $\hat{\mathbf{H}}$ and $\hat{\mathbf{K}}_\kappa$ as \mathbf{H} and \mathbf{K}_κ and stop.

13. Go back to Step 3 after letting

$$\hat{J}_{\text{med}} \leftarrow J_{\text{med}}, \quad \hat{\mathbf{H}} \leftarrow \mathbf{H}, \quad \hat{\mathbf{K}}_\kappa \leftarrow \mathbf{K}_\kappa, \quad W_\kappa \leftarrow e^{-J_\kappa/J_{\text{med}}}. \quad (84)$$

Note that this algorithm does not compute the matrix $\mathbf{\Omega}_\infty^*$ in Eq. (73); it directly outputs the rectifying projective transformation \mathbf{H} and the intrinsic parameter matrix \mathbf{K}_κ .

A.5. 3-D Positions and Motion Parameters

Using the computed \mathbf{H} , we can rectify the projection matrices $\tilde{\mathbf{P}}_\kappa$ and the 3-D positions \mathbf{X}_α as follows:

$$\bar{\mathbf{P}}_\kappa = \mathbf{P}_\kappa \mathbf{H}, \quad \bar{\mathbf{X}}_\alpha = \mathbf{H}^{-1} \mathbf{X}_\alpha. \quad (85)$$

The 3-D coordinates $(X_\alpha, Y_\alpha, Z_\alpha)$ are given by Eqs. (13). From the computed \mathbf{K}_κ , the motion parameters $\{\mathbf{R}_\kappa, \mathbf{t}_\kappa\}$ are to be determined such that

$$\mathbf{K}_\kappa^{-1} \bar{\mathbf{P}}_\kappa \simeq (\mathbf{R}_\kappa \quad \mathbf{t}_\kappa). \quad (86)$$

So, we adjust the scale of $\mathbf{K}_\kappa^{-1} \bar{\mathbf{P}}_\kappa$ so that its first three columns are all unit vectors (in practice, their average norm is made 1). We choose the sign of $\mathbf{K}_\kappa^{-1} \bar{\mathbf{P}}_\kappa$ so that its first three columns define a rotation matrix \mathbf{R}_κ of determinant 1. Then, the fourth column gives the translation \mathbf{t}_κ . The resulting \mathbf{R}_κ may not be strictly orthonormal in the presence of noise, so we enforce the orthonormality by computing the singular value decomposition

$$\mathbf{R}_\kappa = \mathbf{U} \text{diag}(\lambda_1, \lambda_2, \lambda_3) \mathbf{V}^\top, \quad (87)$$

and letting $\mathbf{R}_\kappa = \mathbf{U} \mathbf{V}^\top$ [15].

A.6. Mirror Image Solution Removal

Now, we remove the mirror image solution (Remark 14). If a point is at $(X_\alpha, Y_\alpha, Z_\alpha)$, its coordinates $(X_{\kappa\alpha}^c, Y_{\kappa\alpha}^c, Z_{\kappa\alpha}^c)$ with respect to the κ th camera coordinate system are given by

$$\begin{pmatrix} X_{\kappa\alpha}^c \\ Y_{\kappa\alpha}^c \\ Z_{\kappa\alpha}^c \end{pmatrix} = \mathbf{t}_\kappa + \mathbf{R}_\kappa \begin{pmatrix} X_\alpha \\ Y_\alpha \\ Z_\alpha \end{pmatrix}. \quad (88)$$

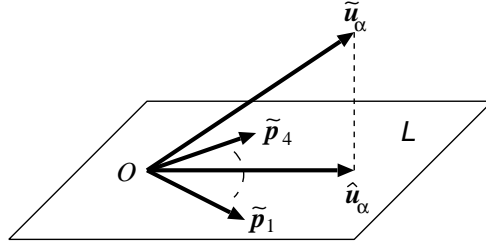
We can judge that it is in front of the camera if

$$\sum_{\alpha=1}^N \text{sgn}(Z_{1\alpha}^c) > 0, \quad (89)$$

where $\text{sgn}(x)$ returns 1, 0, and -1 for $x > 0$, $x = 0$, and $x < 0$, respectively. If Eq. (89) is not satisfied, we reverse the signs of X_α , Y_α , Z_α , and \mathbf{t}_κ . We introduce $\text{sgn}(x)$ because if we require $\sum_{\alpha=1}^N Z_{1\alpha}^c > 0$, the judgment may be reversed when a very large depth $Z_{1\alpha}^c \approx \infty$ may be computed to be $Z_{1\alpha}^c \approx -\infty$ in the presence of noise. Theoretically, we should require $\sum_{\kappa=1}^M \sum_{\alpha=1}^N \text{sgn}(Z_{\kappa\alpha}^c) > 0$, but considering the first camera alone is sufficient in practice.

B. Procedure for Projective Reconstruction

Here, we give two algorithms for projective reconstruction. One is the method of Mahamud and Hebert [25], which we call the *primal method*. The other, which we call the *dual method*, is based on Heyden et al. [11]. We modify these, using corresponding symbols and notations so that their mutual relationships become clear.

Figure 11. Orthogonal projection of $\tilde{\mathbf{u}}_\alpha$ onto \mathcal{L} .

B.1. Primal Method

Eq. (40) indicates that vectors $\tilde{\mathbf{u}}_\alpha$ (= the columns of the matrix on the left-hand side of Eq. (43)) are constrained to be in the 4-D subspace \mathcal{L} spanned by $\{\tilde{\mathbf{p}}_1, \tilde{\mathbf{p}}_2, \tilde{\mathbf{p}}_3, \tilde{\mathbf{p}}_4\}$ if the projective depths $z_{\kappa\alpha}$ are all correct. This does not hold if $z_{\kappa\alpha}$ are not correct, so we update $z_{\kappa\alpha}$ so that each $\tilde{\mathbf{u}}_\alpha$ is as close to \mathcal{L} as possible, identifying $\{\tilde{\mathbf{p}}_1, \tilde{\mathbf{p}}_2, \tilde{\mathbf{p}}_3, \tilde{\mathbf{p}}_4\}$ with the unit eigenvectors of \mathbf{C} in Eq. (41) for the largest four eigenvalues (or the first four columns of \mathbf{U} in Eq. (42)). The orthogonal projection of $\tilde{\mathbf{u}}_\alpha$ onto \mathcal{L} is (Fig. 11)

$$\hat{\mathbf{u}}_\alpha = \sum_{i=1}^4 (\tilde{\mathbf{u}}_\alpha, \tilde{\mathbf{p}}_i) \tilde{\mathbf{p}}_i, \quad (90)$$

where and hereafter we denote the inner product of vectors \mathbf{a} and \mathbf{b} by (\mathbf{a}, \mathbf{b}) . Since $\tilde{\mathbf{u}}_\alpha$ is normalized to unit norm (Section 4.5), the distance of $\tilde{\mathbf{u}}_\alpha$ from the subspace \mathcal{L} is

$$\sqrt{\|\tilde{\mathbf{u}}_\alpha\|^2 - \|\hat{\mathbf{u}}_\alpha\|^2} = \sqrt{1 - \sum_{i=1}^4 (\tilde{\mathbf{u}}_\alpha, \tilde{\mathbf{p}}_i)^2}. \quad (91)$$

Minimizing this is equivalent to maximizing

$$\begin{aligned} J_\alpha &= \sum_{i=1}^4 (\tilde{\mathbf{u}}_\alpha, \tilde{\mathbf{p}}_i)^2 = \sum_{i=1}^4 \left(\sum_{\kappa=1}^M (z_{\kappa\alpha} \mathbf{x}_{\kappa\alpha}, \mathbf{p}_{i\kappa}) \right)^2 \\ &= \sum_{\kappa, \lambda=1}^M \left(\sum_{i=1}^4 (\mathbf{x}_{\kappa\alpha}, \mathbf{p}_{i\kappa})(\mathbf{x}_{\lambda\alpha}, \mathbf{p}_{i\lambda}) \right) z_{\kappa\alpha} z_{\lambda\alpha}, \end{aligned} \quad (92)$$

where $\mathbf{p}_{i\kappa}$ is the 3-D vector consisting of the $3(\kappa-1)+1$ th, $3(\kappa-1)+2$ th, and $3(\kappa-1)+3$ th components of $\tilde{\mathbf{p}}_i$. Thus, Eq. (92) is to be maximized subject to

$$\|\tilde{\mathbf{u}}_\alpha\|^2 = \sum_{\kappa=1}^M z_{\kappa\alpha}^2 \|\mathbf{x}_{\kappa\alpha}\|^2 = 1. \quad (93)$$

Define new variables $\xi_{\kappa\alpha}$ by

$$\xi_{\kappa\alpha} = \|\mathbf{x}_{\kappa\alpha}\| z_{\kappa\alpha}, \quad (94)$$

and consider the M -D vector ξ_α with components $\xi_{1\alpha}, \dots, \xi_{M\alpha}$. Then, Eq. (93) means $\|\xi_\alpha\| = 1$, and Eq. (92) is rewritten as

$$J_\alpha = \sum_{\kappa, \lambda=1}^M A_{\kappa\lambda}^\alpha \xi_{\kappa\alpha} \xi_{\lambda\alpha} = (\xi_\alpha, \mathbf{A}^\alpha \xi_\alpha), \quad (95)$$

where we define the $M \times M$ matrix $\mathbf{A}^\alpha = (A_{\kappa\lambda}^\alpha)$ by

$$A_{\kappa\lambda}^\alpha = \frac{\sum_{i=1}^4 (\mathbf{x}_{\kappa\alpha}, \mathbf{p}_{i\kappa})(\mathbf{x}_{\lambda\alpha}, \mathbf{p}_{i\lambda})}{\|\mathbf{x}_{\kappa\alpha}\| \cdot \|\mathbf{x}_{\lambda\alpha}\|}. \quad (96)$$

Eq. (95) is maximized by the unit eigenvector ξ_α of the matrix \mathbf{A}^α for the largest eigenvalue. The sign is chosen so that

$$\sum_{\kappa=1}^M \xi_{\kappa\alpha} \geq 0. \quad (97)$$

The corresponding projective depths $z_{\kappa\alpha}$ are determined from Eq. (94). The procedure is summarized as follows:

Input: $\mathbf{x}_{\kappa\alpha}, \kappa = 1, \dots, M, \alpha = 1, \dots, N$.

Output: $\mathbf{P}_\kappa, \kappa = 1, \dots, M, \mathbf{X}_\alpha, \alpha = 1, \dots, N$.

Computation:

1. Initialize the projective depths to $z_{\kappa\alpha} = 1$ (Remark 23).
2. Compute $\tilde{\mathbf{u}}_\alpha$ and normalize them into unit norm.
3. Fit a 4-D subspace \mathcal{L} to $\tilde{\mathbf{u}}_\alpha$ by least squares (Remark 24).
4. Do the following computations for $\alpha = 1, \dots, N$.
 - (a) Compute the unit eigenvector ξ_α of the matrix \mathbf{A}^α defined by Eq. (96) for the largest eigenvalue, and choose the sign as in Eq. (97).
 - (b) Determine the projective depths $z_{\kappa\alpha}$ according to Eq. (94).
 - (c) Recompute the vector $\tilde{\mathbf{u}}_\alpha$.
5. Go back to Step 3, and repeat this until the iterations converge.
6. Compute $\mathbf{X}_\alpha = (X_\alpha^i)$ by

$$X_\alpha^i = (\tilde{\mathbf{u}}_\alpha, \tilde{\mathbf{p}}_i). \quad (98)$$

7. Determine the projection matrix \mathbf{P}_κ by

$$\mathbf{P}_\kappa = (\tilde{\mathbf{p}}_{1\kappa} \quad \tilde{\mathbf{p}}_{2\kappa} \quad \tilde{\mathbf{p}}_{3\kappa} \quad \tilde{\mathbf{p}}_{4\kappa}), \quad (99)$$

where $\tilde{\mathbf{p}}_{i\kappa}$ is a 3-D vector whose first, second, and third components are, respectively, the $(3(\kappa - 1) + 1)$ st, $(3(\kappa - 1) + 2)$ nd, and $(3(\kappa - 1) + 3)$ rd components of $\tilde{\mathbf{p}}_i$.

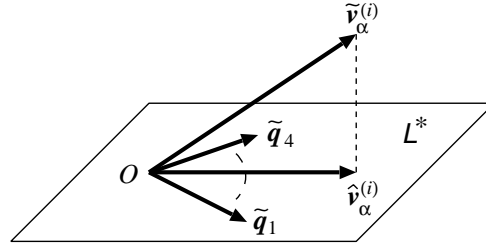


Figure 12. Orthogonal projection of $\tilde{\mathbf{v}}_\kappa^{(i)}$ onto \mathcal{L}^* .

B.2. Dual Method

Consider the following N -D vectors:

$$\tilde{\mathbf{v}}_\kappa^{(1)} = \begin{pmatrix} z_{\kappa 1} x_{\kappa 1} \\ z_{\kappa 2} x_{\kappa 2} \\ \vdots \\ z_{\kappa 1} x_{\kappa 1} \end{pmatrix}, \quad \tilde{\mathbf{v}}_\kappa^{(2)} = \begin{pmatrix} z_{\kappa 1} y_{\kappa 1} \\ z_{\kappa 2} y_{\kappa 2} \\ \vdots \\ z_{\kappa 1} y_{\kappa 1} \end{pmatrix}, \quad \tilde{\mathbf{v}}_\kappa^{(3)} = \begin{pmatrix} z_{\kappa 1} \\ z_{\kappa 2} \\ \vdots \\ z_{\kappa 1} \end{pmatrix}. \quad (100)$$

Note that the transpose $\tilde{\mathbf{v}}_\kappa^{(i)\top}$ is the $(3(\kappa-1)+i)$ th row of the matrix on the left-hand side of Eq. (43), which is written as $(\tilde{\mathbf{v}}_1^{(1)} \tilde{\mathbf{v}}_1^{(2)} \tilde{\mathbf{v}}_1^{(3)} \tilde{\mathbf{v}}_2^{(1)} \dots \tilde{\mathbf{v}}_M^{(3)})^\top$. For the scale normalization, we impose

$$\sum_{i=1}^3 \|\tilde{\mathbf{v}}_\kappa^{(i)}\|^2 = \sum_{\alpha=1}^N z_{\kappa\alpha}^2 \|\mathbf{x}_\kappa\|^2 = 1. \quad (101)$$

If we take out the i th component of Eq. (39) and vertically align it for $\alpha = 1, \dots, N$, we obtain

$$\tilde{\mathbf{v}}_\kappa^{(i)} = P_{\kappa(i1)} \mathbf{X}^1 + P_{\kappa(i2)} \mathbf{X}^2 + P_{\kappa(i3)} \mathbf{X}^3 + P_{\kappa(i4)} \mathbf{X}^4, \quad (102)$$

where $P_{\kappa(ij)}$ is the (ij) element of \mathbf{P}_κ , and \mathbf{X}^k is the N -D vector consisting of X_α^k (= the k th component of \mathbf{X}_α), $\alpha = 1, \dots, N$. Eq. (102) implies that the $3M$ vectors $\tilde{\mathbf{v}}_\kappa^{(i)}$ belong to the 4-D subspace \mathcal{L}^* spanned by \mathbf{X}^1 , \mathbf{X}^2 , \mathbf{X}^3 , and \mathbf{X}^4 . The orthonormal basis $\{\tilde{\mathbf{q}}_1, \dots, \tilde{\mathbf{q}}_4\}$ of the subspace \mathcal{L}^* is given by the first four columns of the matrix \mathbf{V} in Eq. (42). The orthogonal projection of $\tilde{\mathbf{v}}_\kappa^{(i)}$ onto \mathcal{L}^* is (Fig. 12)

$$\hat{\mathbf{v}}_\kappa^{(i)} = \sum_{k=1}^4 (\tilde{\mathbf{v}}_\kappa^{(i)}, \tilde{\mathbf{q}}_k) \tilde{\mathbf{q}}_k. \quad (103)$$

We update $z_{\kappa\alpha}$ so that the sum of squares of the distances of $\tilde{\mathbf{v}}_\kappa^{(1)}$, $\tilde{\mathbf{v}}_\kappa^{(2)}$, and $\tilde{\mathbf{v}}_\kappa^{(3)}$ from the subspace \mathcal{L}^*

$$\sum_{i=1}^3 \left(\|\tilde{\mathbf{v}}_\kappa^{(i)}\|^2 - \|\hat{\mathbf{v}}_\kappa^{(i)}\|^2 \right) = \sum_{i=1}^3 \|\tilde{\mathbf{v}}_\kappa^{(i)}\|^2 - \sum_{i=1}^3 \sum_{k=1}^4 (\tilde{\mathbf{v}}_\kappa^{(i)}, \tilde{\mathbf{q}}_k)^2 \quad (104)$$

is minimized for each α . Consider the N -D vector ξ_κ with components $\xi_{\kappa 1}, \dots, \xi_{\kappa N}$ defined by Eq. (94). Then, minimizing Eq. (104) is equivalent to maximizing

$$J_\kappa^* = \sum_{i=1}^3 \sum_{k=1}^4 (\tilde{v}_\kappa^{(i)}, \tilde{q}_k)^2 = (\xi_\kappa, \mathbf{B}^\kappa \xi_\kappa), \quad (105)$$

where we define the $N \times N$ matrix $\mathbf{B}^\kappa = (B_{\alpha\beta}^\kappa)$ by

$$B_{\alpha\beta}^\kappa = \frac{(\mathbf{q}_\alpha, \mathbf{q}_\beta)(\mathbf{x}_{\kappa\alpha}, \mathbf{x}_{\kappa\beta})}{\|\mathbf{x}_{\kappa\alpha}\| \cdot \|\mathbf{x}_{\kappa\beta}\|}. \quad (106)$$

Here, \mathbf{q}_α is the 4-D vector consisting of the α th components of the basis vectors $\tilde{q}_1, \dots, \tilde{q}_4$. Eq. (105) is maximized by the unit eigenvector ξ_κ of the matrix \mathbf{B}^κ for the largest eigenvalue. The sign is chosen so that

$$\sum_{\alpha=1}^N \xi_{\kappa\alpha} \geq 0, \quad (107)$$

and the corresponding projective depths $z_{\kappa\alpha}$ are determined from Eq. (94). The procedure is summarized as follows:

Input: $\mathbf{x}_{\kappa\alpha}$, $\kappa = 1, \dots, M$, $\alpha = 1, \dots, N$.

Output: P_κ , $\kappa = 1, \dots, M$, \mathbf{X}_α , $\alpha = 1, \dots, N$.

Computation:

1. Initialize the projective depths to $z_{\kappa\alpha} = 1$.
2. Compute the vectors $\tilde{v}_\kappa^{(i)}$ in Eqs. (100), and normalize them as in Eqs. (101).
3. Fit a 4-D subspace \mathcal{L}^* to $\tilde{v}_\kappa^{(i)}$ by least squares.
4. Do the following computations for $\kappa = 1, \dots, M$.
 - (a) Compute the unit eigenvector ξ_κ of the matrix \mathbf{B}^κ defined by Eq. (106) for the largest eigenvalue, and choose the sign as in Eq. (107).
 - (b) Determine the projective depths $z_{\kappa\alpha}$ according to Eq. (107).
 - (c) Recompute the vectors $\tilde{v}_\kappa^{(i)}$.
5. Go back to Step 3, and repeat this until the iterations converge.
6. Compute $\mathbf{X}_\alpha = (X_\alpha^i)$ by

$$X_\alpha^i = (\text{the } \alpha\text{th component of } \tilde{q}_i). \quad (108)$$

7. Determine the projection matrix $P_\kappa = (P_{\kappa(ij)})$ by

$$P_{\kappa(ij)} = (\tilde{v}_\kappa^{(i)}, \tilde{q}_j). \quad (109)$$

C. Affine Camera Factorization

Here, we give the details of the 3-D reconstruction procedure described in Section 5.3. The actual computation depends on what affine camera model we use, so we first describe the general framework that does not depend on specific camera models and then add details that depend on individual models, for which we consider (i) orthographic projection of Eqs. (46) (Fig. 10(a)), (ii) weak perspective projection of Eqs. (47) (Fig. 10(b)), (iii) paraperspective projection of Eqs. (48) (Fig. 10(c)), and (iv) the generic model of Eq. (45). Whichever model we use, we obtain “two” solutions that are mirror images of each other, which cannot be distinguished as long as we use affine camera modeling.

C.1. General Framework

Suppose we track N points over M frames. Let $(x_{\kappa\alpha}, y_{\kappa\alpha})$ be the image coordinates of the α th point in the κ th image. The algorithm for affine camera 3-D reconstruction has the following structure [19, 20]. Items with * depend on the camera model we use. The detailed procedure for them is given later.

Input:

- $2M$ -D trajectory vectors

$$\tilde{\mathbf{u}}_\alpha = (x_{1\alpha} \ y_{1\alpha} \ x_{2\alpha} \ y_{2\alpha} \ \cdots \ x_{M\alpha} \ y_{M\alpha})^\top, \quad \alpha = 1, \dots, N. \quad (110)$$

- Focal lengths f_κ , $\kappa = 1, \dots, M$ (arbitrary if unknown).

Output:

- Translations \mathbf{t}_κ (= the world coordinate origin for the κ th view).
- Shape vectors, i.e., 3-D positions \mathbf{s}_α and \mathbf{s}'_α (mirror images of each other) of the points relative to the world coordinate system centered on their centroid.
- Corresponding rotations \mathbf{R}_κ and \mathbf{R}'_κ that specify the world coordinate axis orientations.

Computation:

1. Compute the centroid $\tilde{\mathbf{u}}_C$ of the trajectory vectors $\tilde{\mathbf{u}}_\alpha$ by Eq. (56).
2. Let $\tilde{t}_{x\kappa}$ and $\tilde{t}_{y\kappa}$ be the $(2(\kappa - 1) + 1)$ th and $(2(\kappa - 1) + 2)$ th components of $\tilde{\mathbf{u}}_C$, respectively.
3. Fit a 3-D affine space to the trajectory vectors $\tilde{\mathbf{u}}_\alpha$, and let $\{\tilde{\mathbf{q}}_1, \tilde{\mathbf{q}}_2, \tilde{\mathbf{q}}_3\}$ be its basis.
4. Let \mathbf{Q} be the $2M \times 3$ matrix having $\tilde{\mathbf{q}}_1$, $\tilde{\mathbf{q}}_2$, and $\tilde{\mathbf{q}}_3$ as its columns, and let $\mathbf{q}_{\kappa(a)}^\dagger$ be the $(2(\kappa - 1) + a)$ th column of \mathbf{Q}^\top , $\kappa = 1, \dots, M$, $a = 1, 2$.
5. *Compute the 3×3 metric matrix \mathbf{T} .

6. Compute the eigenvalues $\{\lambda_1, \lambda_2, \lambda_3\}$ of \mathbf{T} and the corresponding orthonormal system $\{\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3\}$ of unit eigenvectors.
7. *Compute the translation vectors $\mathbf{t}_\kappa = (t_{x\kappa}, t_{y\kappa}, t_{z\kappa})^\top$.
8. Compute the following $2M$ -D vectors:

$$\mathbf{m}_i = \sqrt{\lambda_i} \begin{pmatrix} (\mathbf{q}_{1(1)}^\dagger, \mathbf{v}_i) \\ (\mathbf{q}_{1(2)}^\dagger, \mathbf{v}_i) \\ (\mathbf{q}_{2(1)}^\dagger, \mathbf{v}_i) \\ \vdots \\ (\mathbf{q}_{M(2)}^\dagger, \mathbf{v}_i) \end{pmatrix}, \quad i = 1, 2, 3. \quad (111)$$

9. Let \mathbf{M} be the $2M \times 3$ motion matrix having $\mathbf{m}_1, \mathbf{m}_2,$ and \mathbf{m}_3 as its columns, and let $\mathbf{m}_{\kappa(a)}^\dagger$ be the $(2(\kappa - 1) + a)$ th column of \mathbf{M}^\top , $\kappa = 1, \dots, M, a = 1, 2$.
10. *Compute the rotations \mathbf{R}_κ .
11. *Recompute the motion matrix \mathbf{M} by

$$\mathbf{M} = \sum_{\kappa=1}^M \Pi_\kappa^\top \mathbf{R}_\kappa, \quad (112)$$

where $\Pi_\kappa = (\Pi_{\kappa(ij)})$ is a $3 \times 2M$ matrix that depends on the assumed camera model.

12. Compute the 3-D shape vectors \mathbf{s}_α by

$$\mathbf{s}_\alpha = (\mathbf{M}^\top \mathbf{M})^{-1} \mathbf{M}^\top (\tilde{\mathbf{u}}_\alpha - \tilde{\mathbf{u}}_C). \quad (113)$$

13. *Compute \mathbf{s}'_α and \mathbf{R}'_κ by

$$\mathbf{s}'_\alpha = -\mathbf{s}_\alpha, \quad \mathbf{R}'_\kappa = \mathbf{\Omega}_\kappa \mathbf{R}_\kappa, \quad (114)$$

where $\mathbf{\Omega}_\kappa$ is a rotation matrix that depends on the assumed camera model.

C.2. Metric Constraint

The metric constraint of Eq. (54) is derived as follows. By definition, the three columns $\mathbf{i}_\kappa, \mathbf{j}_\kappa,$ and \mathbf{k}_κ of the rotation \mathbf{R}_κ are the world coordinate axis directions for the κ th view. Their homogeneous coordinate representations are $(1 \ 0 \ 0 \ 0)^\top, (0 \ 1 \ 0 \ 0)^\top,$ and $(0 \ 0 \ 1 \ 0)^\top,$ respectively (they define ‘‘orientations’’ in the projective space \mathcal{P}^3). Hence, according to Eq. (49), their image projections are represented by the first, the second, and the third columns of $\Pi_\kappa \mathbf{R}_\kappa,$ respectively, if the third components are removed, i.e., if expressed in inhomogeneous (or usual) coordinates. From Eq. (50), on the other hand, these vectors are, respectively,

$$\begin{pmatrix} \tilde{p}_{1(3(\kappa-1)+1)} \\ \tilde{p}_{1(3(\kappa-1)+2)} \end{pmatrix}, \quad \begin{pmatrix} \tilde{p}_{2(3(\kappa-1)+1)} \\ \tilde{p}_{2(3(\kappa-1)+2)} \end{pmatrix}, \quad \begin{pmatrix} \tilde{p}_{3(3(\kappa-1)+1)} \\ \tilde{p}_{3(3(\kappa-1)+2)} \end{pmatrix}, \quad (115)$$

where $\tilde{p}_{i(j)}$ is the j th component of $\tilde{\mathbf{p}}_i$. Thus, we have

$$\mathbf{\Pi}_\kappa \mathbf{R}_\kappa = \begin{pmatrix} \tilde{p}_{1(3(\kappa-1)+1)} & \tilde{p}_{2(3(\kappa-1)+1)} & \tilde{p}_{3(3(\kappa-1)+1)} \\ \tilde{p}_{1(3(\kappa-1)+2)} & \tilde{p}_{2(3(\kappa-1)+2)} & \tilde{p}_{3(3(\kappa-1)+2)} \end{pmatrix}. \quad (116)$$

The right-hand side equals $\mathbf{Q}_\kappa^{\dagger\top} \mathbf{A}$ from the definition of \mathbf{A} in Eq. (51) and $\mathbf{Q}_\kappa^\dagger$ in Eq. (52). Hence, we have

$$\mathbf{\Pi}_\kappa \mathbf{R}_\kappa = \mathbf{Q}_\kappa^{\dagger\top} \mathbf{A}. \quad (117)$$

It follows that

$$\mathbf{Q}_\kappa^{\dagger\top} \mathbf{A} \mathbf{A}^\top \mathbf{Q}_\kappa^\dagger = \mathbf{\Pi}_\kappa \mathbf{R}_\kappa \mathbf{R}_\kappa^\top \mathbf{\Pi}_\kappa^\top = \mathbf{\Pi}_\kappa \mathbf{\Pi}_\kappa^\top, \quad (118)$$

or Eq. (54) if the metric matrix \mathbf{T} is defined by Eq. (53).

C.3. Orthographic Projection

If the orthographic projection model of Eqs. (46) is assumed, (Fig. 10(a)), the metric constraint of Eq. (54) takes the following form [19]:

$$(\mathbf{q}_{\kappa(1)}^\dagger, \mathbf{T} \mathbf{q}_{\kappa(1)}^\dagger) = (\mathbf{q}_{\kappa(2)}^\dagger, \mathbf{T} \mathbf{q}_{\kappa(2)}^\dagger) = 1, \quad (\mathbf{q}_{\kappa(1)}^\dagger, \mathbf{T} \mathbf{q}_{\kappa(2)}^\dagger) = 0. \quad (119)$$

From these, we determine the metric matrix \mathbf{T} by least squares. The computation of Step 5 goes as follows [19]. First, we define the $3 \times 3 \times 3 \times 3$ tensor $\mathcal{B} = (B_{ijkl})$ by

$$B_{ijkl} = \sum_{\kappa=1}^M \left[(\mathbf{q}_{\kappa(1)}^\dagger)_i (\mathbf{q}_{\kappa(1)}^\dagger)_j (\mathbf{q}_{\kappa(1)}^\dagger)_k (\mathbf{q}_{\kappa(1)}^\dagger)_l + (\mathbf{q}_{\kappa(2)}^\dagger)_i (\mathbf{q}_{\kappa(2)}^\dagger)_j (\mathbf{q}_{\kappa(2)}^\dagger)_k (\mathbf{q}_{\kappa(2)}^\dagger)_l \right. \\ \left. + \frac{1}{4} \left((\mathbf{q}_{\kappa(1)}^\dagger)_i (\mathbf{q}_{\kappa(2)}^\dagger)_j + (\mathbf{q}_{\kappa(2)}^\dagger)_i (\mathbf{q}_{\kappa(1)}^\dagger)_j \right) \left((\mathbf{q}_{\kappa(1)}^\dagger)_k (\mathbf{q}_{\kappa(2)}^\dagger)_l + (\mathbf{q}_{\kappa(2)}^\dagger)_k (\mathbf{q}_{\kappa(1)}^\dagger)_l \right) \right], \quad (120)$$

where $(\mathbf{q}_{\kappa(a)}^\dagger)_i$ denotes the i th component of the 3-D vector $\mathbf{q}_{\kappa(a)}^\dagger$. We define the 6×6 symmetric matrix \mathbf{B} and the 6-D vector \mathbf{c} by

$$\mathbf{B} = \begin{pmatrix} B_{1111} & B_{1122} & B_{1133} & \sqrt{2}B_{1123} & \sqrt{2}B_{1131} & \sqrt{2}B_{1112} \\ B_{2211} & B_{2222} & B_{2233} & \sqrt{2}B_{2223} & \sqrt{2}B_{2231} & \sqrt{2}B_{2212} \\ B_{3311} & B_{3322} & B_{3333} & \sqrt{2}B_{3323} & \sqrt{2}B_{3331} & \sqrt{2}B_{3312} \\ \sqrt{2}B_{2311} & \sqrt{2}B_{2322} & \sqrt{2}B_{2333} & 2B_{2323} & 2B_{2331} & 2B_{2312} \\ \sqrt{2}B_{3111} & \sqrt{2}B_{3122} & \sqrt{2}B_{3133} & 2B_{3123} & 2B_{3131} & 2B_{3112} \\ \sqrt{2}B_{1211} & \sqrt{2}B_{1222} & \sqrt{2}B_{1233} & 2B_{1223} & 2B_{1231} & 2B_{1212} \end{pmatrix}, \quad (121)$$

$$\mathbf{c} = (1 \ 1 \ 1 \ 0 \ 0 \ 0)^\top, \quad (122)$$

and solve the following simultaneous linear equations for $\boldsymbol{\tau} = (\tau_i)$:

$$\mathbf{B} \boldsymbol{\tau} = \mathbf{c}. \quad (123)$$

The metric matrix \mathbf{T} is given by

$$\mathbf{T} = \begin{pmatrix} \tau_1 & \tau_6/\sqrt{2} & \tau_5/\sqrt{2} \\ \tau_6/\sqrt{2} & \tau_2 & \tau_4/\sqrt{2} \\ \tau_5/\sqrt{2} & \tau_4/\sqrt{2} & \tau_3 \end{pmatrix}. \quad (124)$$

For the translation computation in Step 5, we simply let $t_{x\kappa} = \tilde{t}_{x\kappa}$ and $t_{y\kappa} = \tilde{t}_{y\kappa}$, $\kappa = 1, \dots, 2M$. The third components $t_{z\kappa}$ are left indeterminate. For the rotation computation in Step 10, we compute the SVD

$$\begin{pmatrix} \mathbf{m}_{\kappa(1)}^\dagger & \mathbf{m}_{\kappa(2)}^\dagger & \mathbf{0} \end{pmatrix} = \mathbf{V}_\kappa \mathbf{\Lambda}_\kappa \mathbf{U}_\kappa^\top. \quad (125)$$

Then, the \mathbf{R}_κ is given by

$$\mathbf{R}_\kappa = \mathbf{U}_\kappa \text{diag}(1, 1, \det(\mathbf{V}_\kappa \mathbf{U}_\kappa^\top)) \mathbf{V}_\kappa^\top. \quad (126)$$

The matrix Π_κ in Step 11 is given by

$$\Pi_\kappa = \begin{pmatrix} 0 & \cdots & 0 & \overset{(2\kappa-1)}{1} & \overset{(2\kappa)}{0} & 0 & \cdots & 0 \\ 0 & \cdots & 0 & 0 & 1 & 0 & \cdots & 0 \\ 0 & \cdots & 0 & 0 & 0 & 0 & \cdots & 0 \end{pmatrix}. \quad (127)$$

and the matrix $\mathbf{\Omega}_\kappa$ in Step 13 is simply $\mathbf{\Omega}_\kappa = \text{diag}(-1, -1, 1)$.

C.4. Weak Perspective Projection

If the weak perspective projection model of Eqs. (47) is assumed (Fig. 10(b)), the metric constraint of Eq. (54) takes the following form [19]:

$$(\mathbf{q}_{\kappa(1)}^\dagger, \mathbf{T} \mathbf{q}_{\kappa(1)}^\dagger) = (\mathbf{q}_{\kappa(2)}^\dagger, \mathbf{T} \mathbf{q}_{\kappa(2)}^\dagger) = \frac{f_\kappa^2}{t_{z\kappa}^2}, \quad (\mathbf{q}_{\kappa(1)}^\dagger, \mathbf{T} \mathbf{q}_{\kappa(2)}^\dagger) = 0. \quad (128)$$

Dropping the term $f_\kappa^2/t_{z\kappa}^2$, we determine the metric matrix \mathbf{T} from the resulting two equations by least squares. The computation of Step 5 goes as follows [19]. We define the $3 \times 3 \times 3 \times 3$ tensor $\mathcal{B} = (B_{ijkl})$ by

$$\begin{aligned} B_{ijkl} = & \sum_{\kappa=1}^M \left[(\mathbf{q}_{\kappa(1)}^\dagger)_i (\mathbf{q}_{\kappa(1)}^\dagger)_j (\mathbf{q}_{\kappa(1)}^\dagger)_k (\mathbf{q}_{\kappa(1)}^\dagger)_l - (\mathbf{q}_{\kappa(1)}^\dagger)_i (\mathbf{q}_{\kappa(1)}^\dagger)_j (\mathbf{q}_{\kappa(2)}^\dagger)_k (\mathbf{q}_{\kappa(2)}^\dagger)_l \right. \\ & - (\mathbf{q}_{\kappa(2)}^\dagger)_i (\mathbf{q}_{\kappa(2)}^\dagger)_j (\mathbf{q}_{\kappa(1)}^\dagger)_k (\mathbf{q}_{\kappa(1)}^\dagger)_l + (\mathbf{q}_{\kappa(2)}^\dagger)_i (\mathbf{q}_{\kappa(2)}^\dagger)_j (\mathbf{q}_{\kappa(2)}^\dagger)_k (\mathbf{q}_{\kappa(2)}^\dagger)_l \\ & + \frac{1}{4} \left((\mathbf{q}_{\kappa(1)}^\dagger)_i (\mathbf{q}_{\kappa(2)}^\dagger)_j (\mathbf{q}_{\kappa(1)}^\dagger)_k (\mathbf{q}_{\kappa(2)}^\dagger)_l + (\mathbf{q}_{\kappa(2)}^\dagger)_i (\mathbf{q}_{\kappa(1)}^\dagger)_j (\mathbf{q}_{\kappa(1)}^\dagger)_k (\mathbf{q}_{\kappa(2)}^\dagger)_l \right. \\ & \left. \left. + (\mathbf{q}_{\kappa(1)}^\dagger)_i (\mathbf{q}_{\kappa(2)}^\dagger)_j (\mathbf{q}_{\kappa(2)}^\dagger)_k (\mathbf{q}_{\kappa(1)}^\dagger)_l + (\mathbf{q}_{\kappa(2)}^\dagger)_i (\mathbf{q}_{\kappa(1)}^\dagger)_j (\mathbf{q}_{\kappa(2)}^\dagger)_k (\mathbf{q}_{\kappa(1)}^\dagger)_l \right) \right], \quad (129) \end{aligned}$$

and compute the 6×6 symmetric matrix \mathbf{B} in Eq. (121). Let $\boldsymbol{\tau} = (\tau_i)$ be the 6-D unit eigenvector of \mathbf{B} for the smallest eigenvalue. Then, the metric matrix \mathbf{T} is given by Eq. (124) if $\det \mathbf{T} \geq 0$. If $\det \mathbf{T} < 0$, we change the sing of \mathbf{T} . For the translation computation in Step 5, we first compute

$$t_{z\kappa} = f_\kappa \sqrt{\frac{2}{(\mathbf{q}_{\kappa(1)}^\dagger, \mathbf{T} \mathbf{q}_{\kappa(1)}^\dagger) + (\mathbf{q}_{\kappa(2)}^\dagger, \mathbf{T} \mathbf{q}_{\kappa(2)}^\dagger)}}. \quad (130)$$

Next, we let

$$t_{x\kappa} = \frac{t_{z\kappa} \tilde{t}_{x\kappa}}{f_\kappa}, \quad t_{y\kappa} = \frac{t_{z\kappa} \tilde{t}_{y\kappa}}{f_\kappa}. \quad (131)$$

For the rotation computation in Step 10, we compute the SVD

$$\frac{t_{z\kappa}}{f_\kappa} \begin{pmatrix} \mathbf{m}_{\kappa(1)}^\dagger & \mathbf{m}_{\kappa(2)}^\dagger & \mathbf{0} \end{pmatrix} = \mathbf{V}_\kappa \mathbf{\Lambda}_\kappa \mathbf{U}_\kappa^\top, \quad (132)$$

and determine \mathbf{R}_κ by Eq. (126). The matrix Π_κ in Step 11 is given by

$$\Pi_\kappa = \frac{f_\kappa}{t_{z\kappa}} \begin{pmatrix} & & & \overset{(2\kappa-1)}{0} & \overset{(2\kappa)}{1} & 0 & 0 & \cdots & 0 \\ 0 & \cdots & 0 & 0 & 0 & 1 & 0 & \cdots & 0 \\ 0 & \cdots & 0 & 0 & 0 & 0 & 0 & \cdots & 0 \end{pmatrix}, \quad (133)$$

and the matrix $\mathbf{\Omega}_\kappa$ in Step 13 is simply $\mathbf{\Omega}_\kappa = \text{diag}(-1, -1, 1)$.

C.5. Paraperspective Projection

If the weak paraperspective projection model of Eqs. (48) is assumed (Fig. 10(c)), the metric constraint of Eq. (54) takes the following form [19]:

$$(\mathbf{q}_{\kappa(1)}^\dagger, \mathbf{T} \mathbf{q}_{\kappa(1)}^\dagger) = \frac{f_\kappa^2}{\alpha_\kappa t_{z\kappa}^2}, \quad (\mathbf{q}_{\kappa(2)}^\dagger, \mathbf{T} \mathbf{q}_{\kappa(2)}^\dagger) = \frac{f_\kappa^2}{\beta_\kappa t_{z\kappa}^2}, \quad (\mathbf{q}_{\kappa(1)}^\dagger, \mathbf{T} \mathbf{q}_{\kappa(2)}^\dagger) = \frac{\gamma_\kappa f_\kappa^2}{t_{z\kappa}^2}, \quad (134)$$

where

$$\alpha_\kappa = \frac{1}{1 + \tilde{t}_{x\kappa}^2 / f_\kappa^2}, \quad \beta_\kappa = \frac{1}{1 + \tilde{t}_{y\kappa}^2 / f_\kappa^2}, \quad \gamma_\kappa = \frac{\tilde{t}_{x\kappa} \tilde{t}_{y\kappa}}{f_\kappa^2}. \quad (135)$$

We eliminate $f_\kappa^2 / t_{z\kappa}^2$ from Eqs. (134) and determine the metric matrix \mathbf{T} from the resulting two equations by least squares. The computation of Step 5 goes as follows [19]. We define the $3 \times 3 \times 3 \times 3$ tensor $\mathcal{B} = (B_{ijkl})$ by

$$\begin{aligned} B_{ijkl} = \sum_{\kappa=1}^M & \left[(\gamma_\kappa^2 + 1) \alpha_\kappa^2 (\mathbf{q}_{\kappa(1)}^\dagger)_i (\mathbf{q}_{\kappa(1)}^\dagger)_j (\mathbf{q}_{\kappa(1)}^\dagger)_k (\mathbf{q}_{\kappa(1)}^\dagger)_l \right. \\ & + (\gamma_\kappa^2 + 1) \beta_\kappa^2 (\mathbf{q}_{\kappa(2)}^\dagger)_i (\mathbf{q}_{\kappa(2)}^\dagger)_j (\mathbf{q}_{\kappa(2)}^\dagger)_k (\mathbf{q}_{\kappa(2)}^\dagger)_l + (\mathbf{q}_{\kappa(1)}^\dagger)_i (\mathbf{q}_{\kappa(2)}^\dagger)_j (\mathbf{q}_{\kappa(1)}^\dagger)_k (\mathbf{q}_{\kappa(2)}^\dagger)_l \\ & + (\mathbf{q}_{\kappa(1)}^\dagger)_i (\mathbf{q}_{\kappa(2)}^\dagger)_j (\mathbf{q}_{\kappa(2)}^\dagger)_k (\mathbf{q}_{\kappa(1)}^\dagger)_l + (\mathbf{q}_{\kappa(2)}^\dagger)_i (\mathbf{q}_{\kappa(1)}^\dagger)_j (\mathbf{q}_{\kappa(1)}^\dagger)_k (\mathbf{q}_{\kappa(2)}^\dagger)_l \\ & + (\mathbf{q}_{\kappa(2)}^\dagger)_i (\mathbf{q}_{\kappa(1)}^\dagger)_j (\mathbf{q}_{\kappa(2)}^\dagger)_k (\mathbf{q}_{\kappa(1)}^\dagger)_l - \alpha_\kappa \gamma_\kappa (\mathbf{q}_{\kappa(1)}^\dagger)_i (\mathbf{q}_{\kappa(1)}^\dagger)_j (\mathbf{q}_{\kappa(1)}^\dagger)_k (\mathbf{q}_{\kappa(2)}^\dagger)_l \\ & - \alpha_\kappa \gamma_\kappa (\mathbf{q}_{\kappa(1)}^\dagger)_i (\mathbf{q}_{\kappa(1)}^\dagger)_j (\mathbf{q}_{\kappa(2)}^\dagger)_k (\mathbf{q}_{\kappa(1)}^\dagger)_l - \alpha_\kappa \gamma_\kappa (\mathbf{q}_{\kappa(1)}^\dagger)_i (\mathbf{q}_{\kappa(2)}^\dagger)_j (\mathbf{q}_{\kappa(1)}^\dagger)_k (\mathbf{q}_{\kappa(1)}^\dagger)_l \\ & - \alpha_\kappa \gamma_\kappa (\mathbf{q}_{\kappa(2)}^\dagger)_i (\mathbf{q}_{\kappa(1)}^\dagger)_j (\mathbf{q}_{\kappa(1)}^\dagger)_k (\mathbf{q}_{\kappa(1)}^\dagger)_l - \beta_\kappa \gamma_\kappa (\mathbf{q}_{\kappa(2)}^\dagger)_i (\mathbf{q}_{\kappa(2)}^\dagger)_j (\mathbf{q}_{\kappa(1)}^\dagger)_k (\mathbf{q}_{\kappa(2)}^\dagger)_l \\ & - \beta_\kappa \gamma_\kappa (\mathbf{q}_{\kappa(2)}^\dagger)_i (\mathbf{q}_{\kappa(2)}^\dagger)_j (\mathbf{q}_{\kappa(2)}^\dagger)_k (\mathbf{q}_{\kappa(1)}^\dagger)_l - \beta_\kappa \gamma_\kappa (\mathbf{q}_{\kappa(1)}^\dagger)_i (\mathbf{q}_{\kappa(2)}^\dagger)_j (\mathbf{q}_{\kappa(2)}^\dagger)_k (\mathbf{q}_{\kappa(2)}^\dagger)_l \\ & - \beta_\kappa \gamma_\kappa (\mathbf{q}_{\kappa(2)}^\dagger)_i (\mathbf{q}_{\kappa(1)}^\dagger)_j (\mathbf{q}_{\kappa(2)}^\dagger)_k (\mathbf{q}_{\kappa(2)}^\dagger)_l \\ & + (\gamma_\kappa^2 - 1) \alpha_\kappa \gamma_\kappa (\mathbf{q}_{\kappa(1)}^\dagger)_i (\mathbf{q}_{\kappa(1)}^\dagger)_j (\mathbf{q}_{\kappa(2)}^\dagger)_k (\mathbf{q}_{\kappa(2)}^\dagger)_l \\ & \left. + (\gamma_\kappa^2 - 1) \alpha_\kappa \gamma_\kappa (\mathbf{q}_{\kappa(2)}^\dagger)_i (\mathbf{q}_{\kappa(2)}^\dagger)_j (\mathbf{q}_{\kappa(1)}^\dagger)_k (\mathbf{q}_{\kappa(1)}^\dagger)_l \right]. \quad (136) \end{aligned}$$

Then, we compute the 6×6 symmetric matrix \mathbf{B} in Eq. (121), and let $\boldsymbol{\tau} = (\tau_i)$ be the 6-D unit eigenvector of \mathbf{B} for the smallest eigenvalue. The metric matrix \mathbf{T} is given by Eq. (124) if $\det \mathbf{T} \geq 0$. If $\det \mathbf{T} < 0$, we change the sing of \mathbf{T} . For the translation computation in Step 5, we first compute

$$t_{z\kappa} = f_\kappa \sqrt{\frac{2}{\alpha_\kappa(\mathbf{q}_{\kappa(1)}^\dagger, \mathbf{T}\mathbf{q}_{\kappa(1)}^\dagger) + \beta_\kappa(\mathbf{q}_{\kappa(2)}^\dagger, \mathbf{T}\mathbf{q}_{\kappa(2)}^\dagger)}}. \quad (137)$$

Next, we compute $t_{x\kappa}$ and $t_{y\kappa}$ by Eqs. (131). For the rotation computation in Step 10, we compute

$$\begin{aligned} \mathbf{r}_{\kappa(3)}^\dagger &= \frac{t_{z\kappa}/f_\kappa}{1 + (t_{x\kappa}/t_{z\kappa})^2 + (t_{y\kappa}/t_{z\kappa})^2} \left(\frac{t_{z\kappa}}{f_\kappa} \mathbf{m}_{\kappa(1)}^\dagger \times \mathbf{m}_{\kappa(2)}^\dagger - \frac{t_{x\kappa}}{t_{z\kappa}} \mathbf{m}_{\kappa(1)}^\dagger - \frac{t_{y\kappa}}{t_{z\kappa}} \mathbf{m}_{\kappa(2)}^\dagger \right), \\ \mathbf{r}_{\kappa(1)}^\dagger &= \frac{t_{z\kappa}}{f_\kappa} \mathbf{m}_{\kappa(1)}^\dagger + \frac{t_{x\kappa}}{t_{z\kappa}} \mathbf{r}_{\kappa(3)}^\dagger, \quad \mathbf{r}_{\kappa(2)}^\dagger = \frac{t_{z\kappa}}{f_\kappa} \mathbf{m}_{\kappa(2)}^\dagger + \frac{t_{y\kappa}}{t_{z\kappa}} \mathbf{r}_{\kappa(3)}^\dagger. \end{aligned} \quad (138)$$

Then, we compute the SVD

$$\left(\mathbf{r}_{\kappa(1)}^\dagger \mathbf{r}_{\kappa(2)}^\dagger \mathbf{r}_{\kappa(3)}^\dagger \right) = \mathbf{V}_\kappa \boldsymbol{\Lambda}_\kappa \mathbf{U}_\kappa^\top. \quad (139)$$

The rotation matrices \mathbf{R}_κ are given by Eq. (126). The matrix Π_κ in Step 11 is given by

$$\Pi_\kappa = \frac{f_\kappa}{t_{z\kappa}} \begin{pmatrix} 0 & \cdots & 0 & \overset{(2\kappa-1)}{1} & \overset{(2\kappa)}{0} & 0 & \cdots & 0 \\ 0 & \cdots & 0 & 0 & 1 & 0 & \cdots & 0 \\ 0 & \cdots & 0 & -t_{x\kappa}/t_{z\kappa} & -t_{y\kappa}/t_{z\kappa} & 0 & \cdots & 0 \end{pmatrix}, \quad (140)$$

and the matrix $\boldsymbol{\Omega}_\kappa$ in Step 13 is given by

$$\boldsymbol{\Omega}_\kappa = \frac{2\mathbf{t}_\kappa \mathbf{t}_\kappa^\top}{\|\mathbf{t}_\kappa\|^2} - \mathbf{I}. \quad (141)$$

C.6. Generic Model

If the generic model of Eqs. (45) is assumed, the metric constraint of Eq. (54) takes the following form [20]:

$$\begin{aligned} (\mathbf{q}_{\kappa(1)}^\dagger, \mathbf{T}\mathbf{q}_{\kappa(1)}^\dagger) &= \frac{1}{\zeta_\kappa^2} + \beta_\kappa^2 \tilde{t}_{x\kappa}^2, & (\mathbf{q}_{\kappa(2)}^\dagger, \mathbf{T}\mathbf{q}_{\kappa(2)}^\dagger) &= \frac{1}{\zeta_\kappa^2} + \beta_\kappa^2 \tilde{t}_{y\kappa}^2, \\ (\mathbf{q}_{\kappa(1)}^\dagger, \mathbf{T}\mathbf{q}_{\kappa(2)}^\dagger) &= \beta_\kappa^2 \tilde{t}_{x\kappa} \tilde{t}_{y\kappa}, \end{aligned} \quad (142)$$

We eliminate $1/\zeta_\kappa^2$ and β_κ^2 from Eqs. (142) and determine the metric matrix \mathbf{T} from the resulting two equations by least squares. The computation of Step 5 goes as follows [20]. We let

$$A_\kappa = \tilde{t}_{x\kappa} \tilde{t}_{y\kappa}, \quad C_\kappa = \tilde{t}_{x\kappa}^2 - \tilde{t}_{y\kappa}^2, \quad (143)$$

and define the $3 \times 3 \times 3 \times 3$ tensor $\mathcal{B} = (B_{ijkl})$ by

$$\begin{aligned}
B_{ijkl} = & \sum_{\kappa=1}^M \left[A_{\kappa}^2 \left((\mathbf{q}_{\kappa(1)}^{\dagger})_i (\mathbf{q}_{\kappa(1)}^{\dagger})_j (\mathbf{q}_{\kappa(1)}^{\dagger})_k (\mathbf{q}_{\kappa(1)}^{\dagger})_l + (\mathbf{q}_{\kappa(2)}^{\dagger})_i (\mathbf{q}_{\kappa(2)}^{\dagger})_j (\mathbf{q}_{\kappa(2)}^{\dagger})_k (\mathbf{q}_{\kappa(2)}^{\dagger})_l \right. \right. \\
& - (\mathbf{q}_{\kappa(1)}^{\dagger})_i (\mathbf{q}_{\kappa(1)}^{\dagger})_j (\mathbf{q}_{\kappa(2)}^{\dagger})_k (\mathbf{q}_{\kappa(2)}^{\dagger})_l - (\mathbf{q}_{\kappa(2)}^{\dagger})_i (\mathbf{q}_{\kappa(2)}^{\dagger})_j (\mathbf{q}_{\kappa(1)}^{\dagger})_k (\mathbf{q}_{\kappa(1)}^{\dagger})_l \Big) \\
& + \frac{1}{4} C_{\kappa}^2 \left((\mathbf{q}_{\kappa(1)}^{\dagger})_i (\mathbf{q}_{\kappa(2)}^{\dagger})_j (\mathbf{q}_{\kappa(1)}^{\dagger})_k (\mathbf{q}_{\kappa(2)}^{\dagger})_l + (\mathbf{q}_{\kappa(2)}^{\dagger})_i (\mathbf{q}_{\kappa(1)}^{\dagger})_j (\mathbf{q}_{\kappa(1)}^{\dagger})_k (\mathbf{q}_{\kappa(2)}^{\dagger})_l \right. \\
& + (\mathbf{q}_{\kappa(1)}^{\dagger})_i (\mathbf{q}_{\kappa(2)}^{\dagger})_j (\mathbf{q}_{\kappa(2)}^{\dagger})_k (\mathbf{q}_{\kappa(1)}^{\dagger})_l + (\mathbf{q}_{\kappa(2)}^{\dagger})_i (\mathbf{q}_{\kappa(1)}^{\dagger})_j (\mathbf{q}_{\kappa(2)}^{\dagger})_k (\mathbf{q}_{\kappa(1)}^{\dagger})_l \Big) \\
& - \frac{1}{2} A_{\kappa} C_{\kappa} \left((\mathbf{q}_{\kappa(1)}^{\dagger})_i (\mathbf{q}_{\kappa(1)}^{\dagger})_j (\mathbf{q}_{\kappa(1)}^{\dagger})_k (\mathbf{q}_{\kappa(2)}^{\dagger})_l + (\mathbf{q}_{\kappa(1)}^{\dagger})_i (\mathbf{q}_{\kappa(1)}^{\dagger})_j (\mathbf{q}_{\kappa(2)}^{\dagger})_k (\mathbf{q}_{\kappa(1)}^{\dagger})_l \right. \\
& + (\mathbf{q}_{\kappa(1)}^{\dagger})_i (\mathbf{q}_{\kappa(2)}^{\dagger})_j (\mathbf{q}_{\kappa(1)}^{\dagger})_k (\mathbf{q}_{\kappa(1)}^{\dagger})_l + (\mathbf{q}_{\kappa(2)}^{\dagger})_i (\mathbf{q}_{\kappa(1)}^{\dagger})_j (\mathbf{q}_{\kappa(1)}^{\dagger})_k (\mathbf{q}_{\kappa(1)}^{\dagger})_l \\
& - (\mathbf{q}_{\kappa(1)}^{\dagger})_i (\mathbf{q}_{\kappa(2)}^{\dagger})_j (\mathbf{q}_{\kappa(2)}^{\dagger})_k (\mathbf{q}_{\kappa(2)}^{\dagger})_l - (\mathbf{q}_{\kappa(2)}^{\dagger})_i (\mathbf{q}_{\kappa(1)}^{\dagger})_j (\mathbf{q}_{\kappa(2)}^{\dagger})_k (\mathbf{q}_{\kappa(2)}^{\dagger})_l \\
& \left. \left. - (\mathbf{q}_{\kappa(2)}^{\dagger})_i (\mathbf{q}_{\kappa(2)}^{\dagger})_j (\mathbf{q}_{\kappa(1)}^{\dagger})_k (\mathbf{q}_{\kappa(2)}^{\dagger})_l - (\mathbf{q}_{\kappa(2)}^{\dagger})_i (\mathbf{q}_{\kappa(2)}^{\dagger})_j (\mathbf{q}_{\kappa(2)}^{\dagger})_k (\mathbf{q}_{\kappa(1)}^{\dagger})_l \right) \right]. \quad (144)
\end{aligned}$$

Then, we compute the 6×6 symmetric matrix \mathbf{B} in Eq. (121), and let $\boldsymbol{\tau} = (\tau_i)$ be the 6-D unit eigenvector of \mathbf{B} for the smallest eigenvalue. The metric matrix \mathbf{T} is given by Eq. (124) if $\det \mathbf{T} \geq 0$. If $\det \mathbf{T} < 0$, we change the sing of \mathbf{T} . For the translation computation in Step 5, we solve the following simultaneous linear equations for $1/\zeta_{\kappa}$ and β_{κ}^2 :

$$\begin{aligned}
& \begin{pmatrix} 2 & \tilde{t}_{x\kappa}^2 + \tilde{t}_{y\kappa}^2 \\ \tilde{t}_{x\kappa}^2 + \tilde{t}_{y\kappa}^2 & \tilde{t}_{x\kappa}^4 + \tilde{t}_{y\kappa}^4 \end{pmatrix} \begin{pmatrix} 1/\zeta_{\kappa}^2 \\ \beta_{\kappa}^2 \end{pmatrix} \\
& = \begin{pmatrix} (\mathbf{q}_{\kappa(1)}^{\dagger}, \mathbf{T} \mathbf{q}_{\kappa(1)}^{\dagger}) + (\mathbf{q}_{\kappa(2)}^{\dagger}, \mathbf{T} \mathbf{q}_{\kappa(2)}^{\dagger}) \\ \tilde{t}_{x\kappa}^2 (\mathbf{q}_{\kappa(1)}^{\dagger}, \mathbf{T} \mathbf{q}_{\kappa(1)}^{\dagger}) + \tilde{t}_{y\kappa}^2 (\mathbf{q}_{\kappa(2)}^{\dagger}, \mathbf{T} \mathbf{q}_{\kappa(2)}^{\dagger}) + \tilde{t}_{x\kappa} \tilde{t}_{y\kappa} (\mathbf{q}_{\kappa(1)}^{\dagger}, \mathbf{T} \mathbf{q}_{\kappa(2)}^{\dagger}) \end{pmatrix}. \quad (145)
\end{aligned}$$

Next, we let

$$\begin{pmatrix} t_{x\kappa} \\ t_{y\kappa} \end{pmatrix} = \zeta_{\kappa} \begin{pmatrix} \tilde{t}_{x\kappa} \\ \tilde{t}_{y\kappa} \end{pmatrix}. \quad (146)$$

The third components $t_{z\kappa}$ are left indeterminate. For the rotation computation in Step 10, we compute

$$\begin{aligned}
\mathbf{r}_{\kappa(3)}^{\dagger} &= \zeta_{\kappa} \left(\frac{\zeta_{\kappa} \mathbf{m}_{\kappa(1)}^{\dagger} \times \mathbf{m}_{\kappa(2)}^{\dagger} - \beta_{\kappa} (t_{x\kappa} \mathbf{m}_{\kappa(1)}^{\dagger} + t_{y\kappa} \mathbf{m}_{\kappa(2)}^{\dagger})}{1 + \beta_{\kappa}^2 (t_{x\kappa}^2 + t_{y\kappa}^2)} \right), \\
\mathbf{r}_{\kappa(1)}^{\dagger} &= \zeta_{\kappa} \mathbf{m}_{\kappa(1)}^{\dagger} + \beta_{\kappa} t_{x\kappa} \mathbf{r}_{\kappa(3)}^{\dagger}, \quad \mathbf{r}_{\kappa(2)}^{\dagger} = \zeta_{\kappa} \mathbf{m}_{\kappa(2)}^{\dagger} + \beta_{\kappa} t_{y\kappa} \mathbf{r}_{\kappa(3)}^{\dagger}. \quad (147)
\end{aligned}$$

Then, we compute the SVD of Eq. (139), and \mathbf{R}_{κ} are given by Eq. (126). The matrix Π_{κ} in Step 11 is given by

$$\Pi_{\kappa} = \begin{pmatrix} 0 & \cdots & 0 & \overset{(2\kappa-1)}{1/\zeta_{\kappa}} & \overset{(2\kappa)}{0} & 0 & \cdots & 0 \\ 0 & \cdots & 0 & 0 & 1/\zeta_{\kappa} & 0 & \cdots & 0 \\ 0 & \cdots & 0 & -\beta_{\kappa} t_{x\kappa}/\zeta_{\kappa} & -\beta_{\kappa} t_{y\kappa}/\zeta_{\kappa} & 0 & \cdots & 0 \end{pmatrix}, \quad (148)$$

and the matrix Ω_κ in Step 13 is given by

$$\Omega_\kappa = \frac{2\mathbf{n}_\kappa\mathbf{n}_\kappa^\top}{\|\mathbf{n}_\kappa\|^2} - \mathbf{I}, \quad \mathbf{n}_\kappa = \begin{pmatrix} 1 \\ 0 \\ -\beta_\kappa t_{x\kappa} \end{pmatrix} \times \begin{pmatrix} 0 \\ 1 \\ -\beta_\kappa t_{y\kappa} \end{pmatrix}. \quad (149)$$

References

- [1] Basri, R. *Int. J. Comput. Vision* 1996, 19, 169–179.
- [2] Deguchi, K.; Sasano, T.; Arai, H.; Yoshikawa, H. *IEICE Trans. Inf. & Syst.* 1996, E79-D, 1329–1336.
- [3] Dempster, A. P.; Laird, N. M.; Rubin, D. B. *J. Roy. Statist. Soc.* 1977, B-39, 1–38.
- [4] Faugeras, O. *Three-Dimensional Computer Vision: A Geometric Viewpoint*; MIT Press, Cambridge, MA, 1993.
- [5] Faugeras, O.; Luong, Q.-T. *The Geometry of Multiple Images*; MIT Press, Cambridge, MA, 2001.
- [6] Forsyth, D. A.; Ponce, J. *Computer Vision: A Modern Approach*; Prentice Hall, Upper Saddle River, NJ, 2003.
- [7] Hartley, R. I. *Int. J. Comput. Vision.* 1998, 21-6, 41–61.
- [8] Hartley, R.; Zisserman, A. *Multiple View Geometry in Computer Vision*; Cambridge University Press, Cambridge, 2000.
- [9] Heyden, A.; Åström, K. In *Proc. IEEE Conf. Comput. Vision Patt. Recog.*, June 1997, Puerto Rico, pp. 438–443.
- [10] Heyden A.; Åström, K. In *Proc. 7th Int. Conf. Computer. Vision*; September 1999, Kerkyra, Greece, Vol. 1, pp. 350–355.
- [11] Heyden, A.; Berthilsson, R.; Sparr, G. *Image Vision Comput.* 1999, 17, 981–991.
- [12] Heyden A.; Pollefeys, M. In *Emerging Topics in Computer Vision*; G. Medioni, G.; Kang, S. B.; Eds.; Prentice Hall, Upper Saddle River, NJ, pp. 44–106.
- [13] Kanatani, K. *Group-Theoretical Methods in Image Understanding*; Springer, Berlin, 1990.
- [14] Kanatani, K. *Geometric Computation for Machine Vision*; Oxford University Press, Oxford, 1993.
- [15] Kanatani, K. *Statistical Optimization for Geometric Computation: Theory and Practice*; Elsevier Science, Amsterdam, 1996; reprinted by Dover Publications, New York, NY, 2005.
- [16] Kanatani, K. In *Proc. 8th Int. Conf. Comput. Vision*; July 2001, Vancouver, Canada, Vol. 2, pp. 301–306.
- [17] Kanatani, K. *Int. J. Image Graphics* 2002, 2, 179–197.
- [18] Kanatani, K. In *Proc. 7th Euro. Conf. Comput. Vision*; May 2002, Copenhagen, Denmark, Vol. 3, pp. 335–349.

-
- [19] Kanatani, K.; Sugaya, Y. *Mem. Fac. Eng. Okayama Univ.* 2004, 38, 61–72.
- [20] Kanatani, K.; Sugaya, Y.; Ackermann, H. *IEICE Trans. Inf. & Syst.* 2007, E90-D, 851–858.
- [21] Ma, Y.; Soatto, S.; Košecká, J.; Sastry, S. S. *An Invitation to 3-D Vision: From Images to Geometric Models*; Springer, New York, NY, 2004.
- [22] Marr, D. *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information*; W. H. Freeman, San Francisco, CA, 1982.
- [23] Medioni, G.; Kang, S. B.; Eds. *Emerging Topics in Computer Vision*, Prentice Hall, Upper Saddle River, NJ, 2004.
- [24] Morita, M.; Kanade, T. *IEEE Trans. Patt. Anal. Mach. Intell.* 1997, 19-8, 858–867.
- [25] Mahamud, S.; Hebert, M.; In *Proc. IEEE Conf. Comput. Vision Patt. Recog.*; June 2000, Hilton Head Island, SC, Vol. 2, pp. 430–437.
- [26] Mahamud, S.; Hebert, H.; Omori, Y.; Ponce, J. In *Proc. IEEE Conf. Comput. Vision Patt. Recog.*; December 2001, Kauai, HI, Vol. 1, pp. 1018–1025.
- [27] Poelman, C. J.; and Kanade, T. *IEEE Trans. Patt. Anal. Mach. Intell.* 1997, 19, 206–218.
- [28] Pollefeys, M.; Koch, R.; Van Gool, L. *Int. J. Comput. Vision* 1999, 32, 145–150.
- [29] Quan, L. *Int. J. Comput. Vision* 1996, 19, 93–105.
- [30] Seo, Y.; Heyden, A. In *Proc. 15th Int. Conf. Patt. Recog.*; September 2000, Barcelona, Spain, Vol. 1, pp. 69–71.
- [31] Seo Y.; Heyden, H. *Image and Vision Computing* 2004, 22, 919–926.
- [32] Seo, Y.; K.-S. Hong, K.-S. *IEICE Trans. Inf. & Syst.* 2001, E84-D, 1626–1632.
- [33] Shapiro, L. S.; Zisserman, A.; Brady, M. *Int. J. Comput. Vision* 1995, 16, 147–182.
- [34] Sturm, P.; Triggs, B. In *Proc. 4th Euro. Conf. Comput. Vision*; April 1996, Cambridge, Vol. 2, pp. 709–720.
- [35] Sugaya Y.; Kanatani, K. *IEICE Trans. Inf. & Syst.* 2003, E86-D, 1095–1102.
- [36] Sugaya, Y.; Kanatani, K. *IEICE Trans. Inf. & Syst.* 2004, E87-D, 1031–1038.
- [37] Sugaya, Y.; Kanatani, K. *IEICE Trans. Inf. & Syst.* 2004, E87-D, 1935–1942.
- [38] Sugaya, Y.; Kanatani, K. *Mem. Fac. Eng. Okayama Univ.* 2005, 39, 56–62.
- [39] Sugimoto, A. *Int. J. Comput. Vision* 1996, 19, 181–201.
- [40] Tomasi, C.; Kanade, T. *Int. J. Comput. Vision* 1992, 9, 137–154.
- [41] Triggs, B. In *Proc. IEEE Conf. Comput. Vision Patt. Recog.*; June 1996, San Francisco, CA, pp. 845–851.
- [42] Triggs, B. In *Proc. IEEE Conf. Compt. Vision Patt. Recog.*; June 1997, San Juan, Puerto Rico, pp. 609–614.
- [43] Ueshiba, T.; Tomita, F. *Proc. 5th Euro. Conf. Comput. Vision*; June 1998, Freiburg, Germany, Vol. 1, pp. 296–310.

-
- [44] Xu, B.; Zhang, Z. *Epipolar Geometry in Stereo, Motion and Object Recognition*; Kluwer, Dordrecht, The Netherlands, 1996.
- [45] Zhang, Z. In *Emerging Topics in Computer Vision*, Medioni, G.; Kang, S. B.; Eds.; Prentice Hall, Upper Saddle River, NJ, pp. 4–40.