Memoirs of the Faculty of Engineering, Okayama University, Vol. 40, pp. 64–77, January 2006

Overview of 3-D Reconstruction from Images

Kenichi KANATANI*
Department of Computer Science, Okayama University
Okayama 700-8530 Japan

(Received December 7, 2005)

This article summarizes recent advancements of the theories and techniques for 3-D reconstruction from multiple images. We start with the description of the camera imaging geometry as perspective projection in terms of homogeneous coordinates and the definition of the intrinsic and extrinsic parameters of the camera. Next, we described the epipolar geometry for two, three, and four cameras, introducing such concepts as the fundamental matrix, epipolars, epipoles, the trifocal tensor, and the quadrifocal tensor. Then, we present the self-calibration technique based on the stratified reconstruction approach, 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.

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 basically consists of the following three stages:

- Image processing for detecting, extracting, and matching *features*, which can be points, lines, regions, or anything that is characteristics to that scene.
- Acquiring *metric* information such as locations, orientations, distances, sizes, and motions of the objects in the scene.
- 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 been historically known as *early* (or *low-level*) *vision*, *intermediate-level vision*, and *high-level vision*, respectively [21]. However, they are not necessarily treated separately; they are closely and interactively combined in most real computer vision systems.

The second stage, today called *3-D reconstruction* or *structure from motion* (*SFM*), 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, while 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. Currently, a lot of research is going on for it, mostly based on a trial-and-error basis combined with various heuristics.

In contrast, the 3-D reconstruction stage has extensively studied in the last few decades to arrive at almost definitive conclusions. The aim of this article is to present an overview of thus established latest technologies of 3-D reconstruction from multiple images. Standard textbooks on this subjects are, for example, [4, 5, 6, 7, 12, 13, 14, 22, 43].

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 a point O_c, called the *viewpoint*, and a plane Π_c, called the *image plane* or *retina*, in the scene 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.

We call the line starting from the viewpoint O_c and perpendicularly passing through the image plane

*E-mail kanatani@suri.it.okayama-u.ac.jp

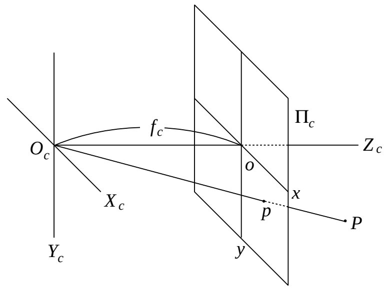


Figure 1: Perspective projection.

Π_c the *optical axis*. We define an $X_cY_cZ_c$ 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

$$x = f_c\frac{X_c}{Z_c}, \qquad y = f_c\frac{Y_c}{Z_c}, \tag{1}$$

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*. They are placed in parallel rows at equal intervals in horizontal and vertical directions. However, the vertical columns of pixels are not necessarily orthogonal to the horizontal rows, and the inter-pixel distance may not be the same in the horizontal and vertical directions. Labeling the upper-left pixel $(u, v) = (0, 0)$, we count the pixels u = 1, 2, ... rightward and v = 1, 2, ... downward. We identify the integer pair (u, v) with the position at the center of that pixel and specify inter-pixel, or *subpixel*, positions 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

$$u = \frac{x}{\alpha} + \frac{y}{\alpha}\tan\theta + u_0, \qquad v = \frac{y}{\beta} + v_0, \tag{2}$$

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

Remark 1. Note that 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

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} \simeq \boldsymbol{K}\begin{pmatrix} X_c \\ Y_c \\ Z_c \end{pmatrix}, \tag{3}$$

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

$$\boldsymbol{K} = \begin{pmatrix} f\gamma & f\gamma\tan\theta & u_0 \\ & f & v_0 \\ & & 1 \end{pmatrix}, \tag{4}$$

where we put $f = f_c/\beta$, which is also called the *focal length* but strictly the focal length measured in the unit defined so that the vertical distance between pixel rows is 1. We define $\gamma = \beta/\alpha$, which we call the *aspect ratio*. The constants f, γ, θ, u_0, and v_0 are called the *intrinsic parameters* of the camera, and the matrix $\boldsymbol{K}$ 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.

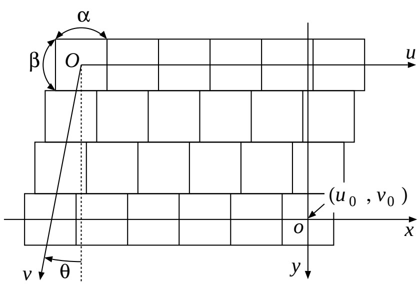


Figure 2: Pixels and the image coordinate system.

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. Then, the second of eqs. (2) becomes $v = y/\beta \cos\theta + v_0$, so the (22) element of the matrix $\boldsymbol{K}$ 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 $\boldsymbol{K}$ is not essential. Many recent textbooks simply write, e.g.,

$$\boldsymbol{K} = \begin{pmatrix} f_1 & s & u_0 \\ & f_2 & v_0 \\ & & 1 \end{pmatrix}, \tag{5}$$

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

2.4 Motion Parameters

Since the $X_cY_cZ_c$ 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 $\boldsymbol{t}$ be its origin described with respect to the camera coordinate system. If the world coordinate system is rotated by $\boldsymbol{R}$ 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):

$$\begin{pmatrix} X_c \\ Y_c \\ Z_c \end{pmatrix} = \boldsymbol{R} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} + \boldsymbol{t}. \tag{6}$$

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

Remark 5. Note that the motion parameters $\{\boldsymbol{R}, \boldsymbol{t}\}$ are description with respect to the *camera* coordinate system. We can also describe this with respect to the world coordinate system. Let $\boldsymbol{t}_c$ be the origin of the camera coordinate system described with respect to the world coordinate system. If the camera coordinate system is rotated by $\boldsymbol{R}_c$ relative to the world coordinate system, we obtain instead of eq. (6)

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \boldsymbol{R}_c \begin{pmatrix} X_c \\ Y_c \\ Z_c \end{pmatrix} + \boldsymbol{t}_c, \tag{7}$$

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

$$\boldsymbol{R} = \boldsymbol{R}_c^\top, \qquad \boldsymbol{t} = -\boldsymbol{R}_c^\top \boldsymbol{t}_c. \tag{8}$$

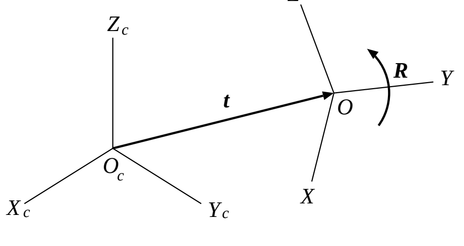


Figure 3: The camera coordinate system and the world coordinate system.

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

$$\boldsymbol{u} \simeq \boldsymbol{P}\boldsymbol{X}, \tag{9}$$

where we put

$$\boldsymbol{u} = \begin{pmatrix} u, \\ v \\ 1 \end{pmatrix}, \qquad \boldsymbol{X} = \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}, \tag{10}$$

and

$$\boldsymbol{P} = \boldsymbol{K} \begin{pmatrix} \boldsymbol{R} & \boldsymbol{t} \end{pmatrix}. \tag{11}$$

This 3×4 matrix $\boldsymbol{P}$ 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, respectively. Hereafter, we refer to points represented by vectors $\boldsymbol{u}$ and $\boldsymbol{X}$ simply as "point $\boldsymbol{u}$" and "point $\boldsymbol{X}$", 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 constant. 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

$$x = \frac{x^1}{x_3}, \qquad y = \frac{x^2}{x_3}. \tag{12}$$

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

$$X = \frac{X^1}{X_4}, \quad Y = \frac{X^2}{X_4}, \quad Z = \frac{X^3}{X_4}. \tag{13}$$

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 $\{\boldsymbol{R}_c, \boldsymbol{t}_c\}$ described with respect to the world coordinate system, eq. (11) becomes

$$\boldsymbol{P} = \boldsymbol{K}\begin{pmatrix}\boldsymbol{R}_c^\top & -\boldsymbol{R}_c^\top\boldsymbol{t}\end{pmatrix} = \boldsymbol{K}\boldsymbol{R}_c^\top\begin{pmatrix}\boldsymbol{I} & -\boldsymbol{t}\end{pmatrix}. \tag{14}$$

($\boldsymbol{I}$ denotes the unit matrix.) In this article, 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 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

$$(X^1)^2 + (X^2)^2 + (X^3)^2 = 0 \tag{15}$$

is called the *absolute conic*. Eq. (9) implies that any projection $\boldsymbol{u}$ of a point of Ω_∞ satisfies, *irrespective* of the motion parameters $\{\boldsymbol{R}, \boldsymbol{t}\}$,

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

The set of (imaginary) points $\boldsymbol{u}$ 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 [44].

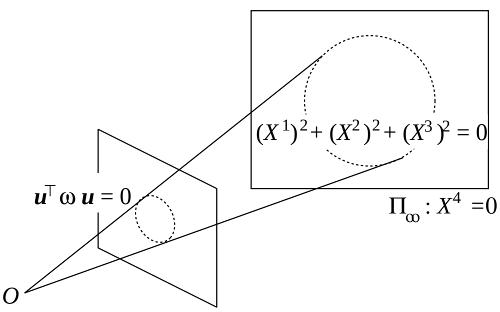


Figure 4: The absolute conic and its projection.

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 $\boldsymbol{X}$ in the scene by M cameras, and let $\boldsymbol{u}_\kappa$ be its projection in the κth image, $\kappa = 1, ..., M$. Let $\boldsymbol{P}_\kappa$ be the projection matrix of the κth camera. For each camera, we have the relationship of eq. (9). If we use an indeterminate nonzero constant λ_κ instead of the relation symbol $\simeq$, we have

$$\lambda_\kappa\boldsymbol{u}_\kappa = \boldsymbol{P}_\kappa\boldsymbol{X}. \tag{17}$$

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

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

Since some $\boldsymbol{X}$ ($\neq \boldsymbol{0}$) and λ_κ, $\kappa = 1, ..., M$, 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 [11].

Remark 9. It is easy to see that unless the chosen $(M+4)\times(M+4)$ minor contains two or more columns of $\boldsymbol{P}_\kappa$, we cannot obtain a meaningful constraint on the projection in the κth image. In fact, if only one column of $\boldsymbol{P}_\kappa$ is included, it simply multiplies the minor by a constant, so its vanishing does not give any information about $\boldsymbol{P}_\kappa$. 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

$$\boldsymbol{u}_1^\top\boldsymbol{F}\boldsymbol{u}_2 = 0, \tag{19}$$

where $\boldsymbol{F}$ is a 3×3 matrix whose (ij) element is

$$F_{ij} = \sum_{k,l,m,n=1}^{3}\epsilon_{ikl}\epsilon_{jmn}\det\boldsymbol{P}_{1122}^{klmn}. \tag{20}$$

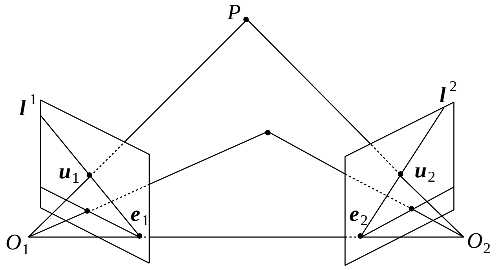


Figure 5: Epipolars and epipoles.

Here, $\boldsymbol{P}^{klmn}_{1122}$ denotes the 4×4 matrix consisting of the kth row of $\boldsymbol{P}_1$, the lth row of $\boldsymbol{P}_1$, the mth row of $\boldsymbol{P}_2$, and the nth row of $\boldsymbol{P}_2$. The matrix $\boldsymbol{F}$ is called the *fundamental matrix*. From eq. (20), it can be shown that the fundamental matrix $\boldsymbol{F}$ 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 $\boldsymbol{u}_1$ in the image plane of the first camera is called the *line of sight* of $\boldsymbol{u}_1$. The line of sight of $\boldsymbol{u}_2$ is defined similarly. Geometrically, eq. (19) describes the requirement that the line of sight of $\boldsymbol{u}_1$ and the line of sight of $\boldsymbol{u}_2$ should intersect at a point (it may be at infinity) in the scene. (Fig. 5).

The set of points $\boldsymbol{u}$ that satisfy $\boldsymbol{l}^\top\boldsymbol{u} = 0$ for some $\boldsymbol{l}$ defines a line in the image. The vector $\boldsymbol{l}$ labels this line up to a nonzero multiplier (i.e., $\boldsymbol{l}$ and $c\boldsymbol{l}$ defines the same line for an arbitrary $c \neq 0$). The three components of $\boldsymbol{l}$ define the homogeneous coordinates of this line. Henceforth, we abbreviate the line represented by vector $\boldsymbol{l}$ simply as "line $\boldsymbol{l}$".

Eq. (19) implies that the point $\boldsymbol{u}_1$ is on the line $\boldsymbol{l}^1 = \boldsymbol{F}\boldsymbol{u}_2$. This line is called the *epipolar line* or simply the *epipolar* of point $\boldsymbol{u}_2$. Eq. (19) also implies that the point $\boldsymbol{u}_2$ is on the line $\boldsymbol{l}^2 = \boldsymbol{F}^\top\boldsymbol{u}_1$, called the *epipolar* (*line*) of point $\boldsymbol{u}_1$.

Thus, eq. (19) states that *a point in one image should be on the epipolar of the corresponding point in the other image*. This requirement is called the *epipolar constraint*. If follows that if the fundamental matrix $\boldsymbol{F}$ is known, one can find point correspondence easily: given a point $\boldsymbol{u}$ in one image, one only needs to search the other image along the epipolar of $\boldsymbol{u}$ (Fig. 5).

3.4 Epipoles

Since the fundamental matrix $\boldsymbol{F}$ has rank 2, it has a null vector. So does $\boldsymbol{F}^\top$, too. Hence, there exist a vector $\boldsymbol{e}_1$ such that $\boldsymbol{F}^\top\boldsymbol{e}_1 = \boldsymbol{0}$ and a vector $\boldsymbol{e}_2$ such that $\boldsymbol{F}\boldsymbol{e}_2 = \boldsymbol{0}$. Identifying $\boldsymbol{e}_1$ and $\boldsymbol{e}_2$ with homogeneous coordinates of points in the image, we call these points the *epipoles*. Geometrically, the epipole $\boldsymbol{e}_1$ is *the projection of the viewpoint O_2 of the second camera onto the first image*, and the epipole $\boldsymbol{e}_2$ 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 $\boldsymbol{F}\boldsymbol{u}_2$ of any point $\boldsymbol{u}_2$ in the second image passes through the epipole $\boldsymbol{e}_1$. Similarly, in the second image, the epipolar $\boldsymbol{F}^\top\boldsymbol{u}_1$ of any point $\boldsymbol{u}_1$ in the first image passes through the epipole $\boldsymbol{e}_2$.

It follows that epipolars 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 of a point $\boldsymbol{u}_2$ of the second image is nothing but the intersection of the first image plane with the plane passing through $\boldsymbol{u}$ and the viewpoints O_1 and O_2 of the two cameras. This plane is called the *epipolar plane* of $\boldsymbol{u}_2$ (and hence of the corresponding point $\boldsymbol{u}_1$).

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*:

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

Here, u^i_κ denotes the ith component of $\boldsymbol{u}_\kappa$, and T_i^{jk} is defined by

$$T_i^{jk} = \sum_{l,m=1}^{3} \epsilon_{ilm} \det \boldsymbol{P}^{lmjk}_{1123}, \tag{22}$$

and called the *trifocal tensor*.

Given a line $\boldsymbol{l}$ in the image plane, the plane Π_{l} defined by the viewpoint O_c and the line $\boldsymbol{l}$ is called the *back projection* of the line $\boldsymbol{l}$. Let Π_{l^2} be the back projection of an arbitrary line $\boldsymbol{l}^2$ passing through $\boldsymbol{u}_2$ in the second image, and Π_{l^3} the back projection of an arbitrary line $\boldsymbol{l}^3$ passing through $\boldsymbol{u}_3$ in the third image. Geometrically, eq. (21) describes the requirement that *the line of sight of $\boldsymbol{u}_1$ in the first image should meet the intersection of the two planes Π_{l^2} and Π_{l^3} at a single point* (it may be at infinity) (Fig. 6).

Remark 11. Take an arbitrary point $\boldsymbol{v}_2$ ($\neq \boldsymbol{u}_2$) in the second image and an arbitrary point $\boldsymbol{v}_3$ ($\neq \boldsymbol{u}_3$) in the third image. Multiplying eq. (21) by $v_2^p v_3^q$ and

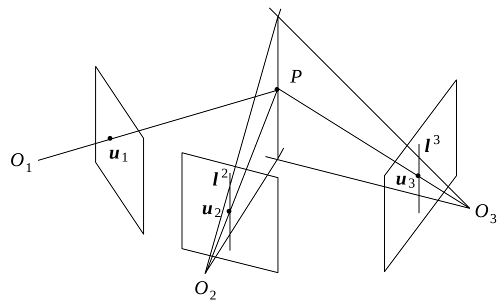


Figure 6: Trifocal constraint.

summing it over p and q, we obtain

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

If we define lines

$$\boldsymbol{l}^2 = \boldsymbol{u}_2 \times \boldsymbol{v}_2, \qquad \boldsymbol{l}^3 = \boldsymbol{u}_3 \times \boldsymbol{v}_3, \tag{24}$$

eq. (23) is rewritten as

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

which describe the geometric relationship mentioned earlier.

3.6 Four-View Geometry

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

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

Here, Q^{ijkl} is a tensor defined by

$$Q^{ijkl} = \det \boldsymbol{P}_{1234}^{ijkl}, \tag{27}$$

called the *quadrifocal tensor*.

Geometrically. eq. (26) describes the requirement that *the back projections* $\Pi_{\mathbf{l}^1} \sim \Pi_{\mathbf{l}^4}$ *of arbitrary lines* $\boldsymbol{l}^1 \sim \boldsymbol{l}^4$ *in each image passing through points* $\boldsymbol{u}_1 \sim \boldsymbol{u}_4$*, respectively, should meet at a single point* (Fig. 7).

Remark 12. Take an arbitrary point $\boldsymbol{v}_\kappa$ ($\neq \boldsymbol{u}_\kappa$) in the κth 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

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

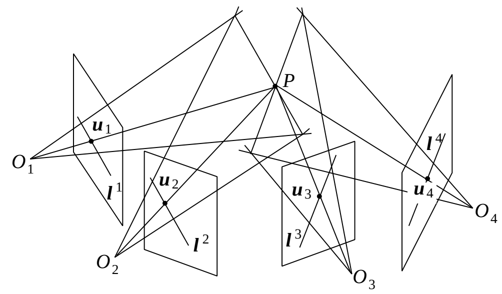


Figure 7: Quadrifocal constraint.

If we define lines

$$\boldsymbol{l}^1 = \boldsymbol{u}_1 \times \boldsymbol{v}_1, \qquad \boldsymbol{l}^2 = \boldsymbol{u}_2 \times \boldsymbol{v}_2,$$
$$\boldsymbol{l}^3 = \boldsymbol{u}_3 \times \boldsymbol{v}_3, \qquad \boldsymbol{l}^4 = \boldsymbol{u}_4 \times \boldsymbol{v}_4, \tag{29}$$

eq. (28) is rewritten as

$$\sum_{i,j,k,l=1}^{3} Q^{ijkl} l_i^1 l_j^2 l_k^3 l_m^4 = 0, \tag{30}$$

which describe the geometric relationship mentioned earlier.

4. 3-D Reconstruction from Images

4.1 Classification of the Problem

Suppose we observe N points $\boldsymbol{X}_\alpha$, $\alpha = 1, ..., N$, in the scene by M cameras having projection matrices $\boldsymbol{P}_\kappa$, $\kappa = 1, ..., M$. Let $\boldsymbol{u}_{\kappa\alpha}$ the projection of point $\boldsymbol{X}_\alpha$ in the κth image. For each point and each camera, we have the relationship described in the form of eq. (9):

$$\boldsymbol{u}_{\kappa\alpha} \simeq \boldsymbol{P}_\kappa \boldsymbol{X}_\alpha. \tag{31}$$

Given projection images $\boldsymbol{u}_{\kappa\alpha}$, $\kappa = 1, ..., M$, $\alpha = 1, ..., N$, the task of computing $\boldsymbol{X}_\alpha$, $\alpha = 1, ..., N$, from them is called *3-D reconstruction* or *structure from motion*. The problem can be classified into the following three cases:

(i) The projection matrix $\boldsymbol{P}$ of each camera is known.

(ii) The intrinsic parameter matrix $\boldsymbol{K}$ of each camera is known (but the motion parameters $\{\boldsymbol{R}, \boldsymbol{t}\}$ are not).

(iii) The projection matrix $\boldsymbol{P}$ of each camera is unknown.

In Case (i), the 3-D coordinates $(X_\alpha, Y_\alpha, Z_\alpha)$ of point $\boldsymbol{X}_\alpha$ can be determined from eq. (31) except for one degree of freedom, which corresponds to the *depth* of the point $\boldsymbol{X}_\alpha$ along the line of sight. Hence, if we have two or more images, we can uniquely determine the depths of all points. 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 $\boldsymbol{F}$ from point correspondences between two images (the trifocal tensor T_i^{jk} from point correspondences over three images, or the quadrifocal tensor Q^{ijkl} from point correspondences over four images). Then, we can compute the motion parameters $\{\boldsymbol{R}, \boldsymbol{t}\}$ by solving eq. (20) (eq. (22), or eq. (27)). Hence, the problem reduces to stereo vision of Case (ii).

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)), one can compute the motion parameters from the trifocal tensor T_i^{jk} over three images or the quadrifocal tensor Q^{ijkl} over four images in principle, but in practice the use of the fundamental matrix $\boldsymbol{F}$ over two images is sufficient. The solution is obtained up to sign and mirror image reflection. However, we can select a unique solution if we impose the constraint that observed points are *in front of* all the cameras with a positive depth from each camera [13, 14].

4.2 Self-calibration

In Case (iii) (self-calibration), the camera matrices $\{\boldsymbol{P}_\kappa\}$ and the 3-D points $\{\boldsymbol{X}_\alpha\}$ in eq. (31) are both unknowns. 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 $\{\boldsymbol{X}_\alpha\}$ and $\{\boldsymbol{P}_\kappa\}$ are a solution, we have another solution

$$\tilde{\boldsymbol{X}}_\alpha \simeq \boldsymbol{H}\boldsymbol{X}_\alpha, \qquad \tilde{\boldsymbol{P}}_\kappa \simeq \boldsymbol{P}_\kappa \boldsymbol{H}^{-1} \tag{32}$$

for an arbitrary nonsingular 4×4 matrix $\boldsymbol{H}$.

The first of eqs. (32) can be regarded as applying a *projective transformation* (or a *homography*) $\boldsymbol{H}$ to the 3-D projective space $\mathcal{P}^3$ (Fig. 8). Accordingly, the points $\{\boldsymbol{X}_\alpha\}$ and $\{\tilde{\boldsymbol{X}}_\alpha\}$ 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 properties* such as lengths and angles are not preserved. 3-D reconstruction determined up to an arbitrary projective transformation is called *projective reconstruction*.

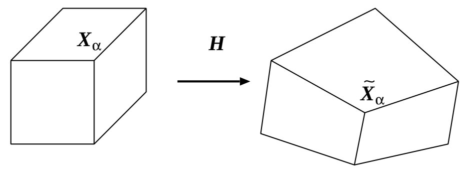


Figure 8: Projective transformation.

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

$$\boldsymbol{X}_\alpha \simeq \boldsymbol{H}^{-1}\tilde{\boldsymbol{X}}_\alpha, \qquad \boldsymbol{P}_\kappa \simeq \tilde{\boldsymbol{P}}_\kappa \boldsymbol{H}. \tag{33}$$

If, for example, we know the true 3-D positions $\boldsymbol{X}_\alpha$ of five (or more) points in general position among $\tilde{\boldsymbol{X}}_\alpha$, we can uniquely determine the projective transformation $\boldsymbol{H}$ that maps the five points $\tilde{\boldsymbol{X}}_\alpha$ to their true positions $\boldsymbol{X}_\alpha$. Then, we can obtain by the first of eqs. (33) the Euclidean reconstruction $\{\boldsymbol{X}_\alpha\}$ from an arbitrary projective reconstruction $\{\tilde{\boldsymbol{X}}_\alpha\}$.

If no such five points are known, we need to assume some constrains on cameras and find an appropriate projective transformation $\boldsymbol{H}$ such that the projectively reconstructed camera matrices $\{\tilde{\boldsymbol{P}}_\kappa\}$ are mapped by the second of eqs. (33) to camera matrices $\{\boldsymbol{P}_\kappa\}$ that satisfy the assumed constraints. Such an 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, we cannot recover the absolute scale from images, as pointed out in Remark 13. Hence, all we can obtain is, strictly speaking, "similarity" reconstruction rather than "Euclidean" or "metric" reconstruction. However, the terms "Euclidean" and "metric" are commonly used to mean "up to similarity".

4.3 Stratified Reconstruction

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

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

where $\mathrm{diag}(a, b, c, ...)$ denotes the diagonal matrix with diagonal elements a, b, c, ... in that order, and we define the 3×3 matrices $\boldsymbol{\omega}_\kappa^*$ by

$$\boldsymbol{\omega}_\kappa^* \equiv \boldsymbol{K}_\kappa \boldsymbol{K}_\kappa^\top. \tag{35}$$

Substituting $\boldsymbol{P}_\kappa$ in the second of eqs. (33) into eq. (34), we obtain

$$\tilde{\boldsymbol{P}}_\kappa \boldsymbol{\Omega}^*_\infty \tilde{\boldsymbol{P}}_\kappa^\top \simeq \boldsymbol{\omega}^*_\kappa, \tag{36}$$

where we define the 4×4 matrix $\boldsymbol{\Omega}^*_\infty$ by

$$\boldsymbol{\Omega}^*_\infty \equiv \boldsymbol{H}\mathrm{diag}(1,1,1,0)\boldsymbol{H}^\top. \tag{37}$$

If the intrinsic parameter matrix $\boldsymbol{K}_\kappa$ is known (i.e., the camera is calibrated), we can determine $\boldsymbol{\omega}^*_\kappa$ from eq. (35). Even if $\boldsymbol{\omega}^*_\kappa$ is not completely known, we can obtain constraints on the elements of $\boldsymbol{\Omega}^*_\infty$ from eq. (36) if we have some knowledge about $\boldsymbol{\omega}^*_\kappa$, such as a particular element being 0 or particular two elements being equal (we are assuming that $\tilde{\boldsymbol{P}}_\kappa$ are given). If the number M of images is sufficiently large to give sufficiently many such constraints on $\boldsymbol{\Omega}^*_\infty$, we can solve them for $\boldsymbol{\Omega}^*_\infty$.

Frequently used assumptions about the camera are:

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

For example, if all the cameras have the same intrinsic parameters (i.e., one camera is moved to take multiple pictures), there is only one intrinsic parameter matrix $\boldsymbol{K}$ for which $\boldsymbol{\omega}^*_1 = ... = \boldsymbol{\omega}^*_M = \boldsymbol{\omega}^*$ ($\equiv \boldsymbol{K}\boldsymbol{K}^\top$). Hence, eq. (36) gives $5(M-1)$ equations of $\boldsymbol{\Omega}^*_\infty$.

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 $\boldsymbol{K}$ in eq. (4) are 0, and hence the (13) and (23) elements of all $\boldsymbol{\omega}^*_\kappa = \boldsymbol{K}_\kappa\boldsymbol{K}_\kappa^\top$ are also 0. In this case, eq. (36) gives $2M$ equations of $\boldsymbol{\Omega}^*_\infty$. If the skew angle is zero in addition, the (12) element of $\boldsymbol{\omega}^*_\kappa$ is also zero, so we obtain $3M$ equations of $\boldsymbol{\Omega}^*_\infty$. 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 $\boldsymbol{\Omega}^*_\infty$ up to a scale factor. If $\boldsymbol{\Omega}^*_\infty$ is determined, $\boldsymbol{\omega}^*_\kappa$ is determined from eq. (36). Then, the projective transformation $\boldsymbol{H}$ is determined from eq. (37). The intrinsic parameter matrix $\boldsymbol{K}_\kappa$ is obtained by solving eq. (35).

Remark 17. From eq. (4), the matrix $\boldsymbol{\omega}^*_\kappa$ in eq. (35) has the form

$$\boldsymbol{\omega}^*_\kappa = \begin{pmatrix} f_\kappa^2\gamma_\kappa^2 + s_\kappa^2 + u_{0\kappa}^2 & f_\kappa s_\kappa^2 + u_{0\kappa}v_{0\kappa} & u_{0\kappa} \\ f_\kappa s_\kappa^2 + u_{0\kappa}v_{0\kappa} & f_\kappa^2 + v_{0\kappa}^2 & v_{0\kappa} \\ u_{0\kappa} & v_{0\kappa} & 1 \end{pmatrix}, \tag{38}$$

where we put $s_\kappa = f_\kappa\gamma_\kappa \tan\theta$. Since this is a 3×3 symmetric matrix with six different elements, 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 $\boldsymbol{\Omega}^*_\infty$ with ten independent elements, but it has scale indeterminacy. So, if we move one camera, we need to have $5(M-1) \geq 9$, or $M \geq 3$. If the principal point $(u_{0\kappa}, u_{0\kappa})$ is known but other parameters can vary from frame to frame, we need to have $2M \geq 9$, or $M \geq 5$. If the skew s_κ is 0 in addition, this is relaxed to $3M \geq 9$, or $M \geq 3$. If furthermore the aspect ratio γ_κ is 1, this becomes $4M \geq 9$, so we still need $M \geq 3$ images.

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 [14], but this is too complicated to carry out. So, a simple least-squares minimization is conducted in practice. There is another problem about the intrinsic constraint on $\boldsymbol{\Omega}^*_\infty$: it should have rank 3 from the definition of eq. (37). If computed by least squares, however, the resulting $\boldsymbol{\Omega}^*_\infty$ is generally rank 4. Since it is difficult to incorporate the rank constraint in the least-squares computation, an ad-hoc treatment, such as computing the singular value decomposition (SVD) of the obtained $\boldsymbol{\Omega}^*_\infty$ and replacing the smallest singular value by 0, is widely employed.

Remark 19. If $\boldsymbol{\Omega}^*_\infty$ is obtained, the projective transformation $\boldsymbol{H}$ is not completely determined by eq. (37); the fourth column of $\boldsymbol{H}$ is arbitrary. However, it can be defined arbitrarily as long as the resulting $\boldsymbol{H}$ is nonsingular. This indeterminacy corresponds to the fact that the absolute orientation of the world coordinate system is indeterminate (recall that $\boldsymbol{R}_\kappa$ specifies the orientation of the world coordinate system relative to the κth camera).

Remark 20. If $\boldsymbol{\Omega}^*_\infty$ is obtained, we can determine $\boldsymbol{\omega}^*_\kappa$ up to a scale factor by eq. (36). Then, we can solve eq. (35) to obtain $\boldsymbol{K}_\kappa$, which is 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 $\boldsymbol{K}_\kappa$ is determined so that its (33) element is 1.

Remark 21. The stratified reconstruction approach was proposed by Faugeras [4] and others. First, the constant cameras constraint was used by many researchers. Later, Heyden and Åström [8, 9] showed that Euclidean reconstruction is possible using as few constraints as zero skew alone if a sufficient number of images and point correspondences over them are available. The dual absolute quadric constraint was formulated by Triggs [41]. Pollefeys et al. [27] demonstrated that exact and detailed reconstruction is indeed possible by this approach. Since then, various modifications and simplifications have been devised for imposing the dual absolute quadric constraint. Many researchers used nonlinear optimization in one form or another, but later formulations in terms of linear computations have been found in many forms; see [29, 30, 31].

4.4 Dual Absolute Quadric Constraint

Comparing the second of eqs. (16) and eq. (35), we can see that the matrix $\boldsymbol{\omega}_\kappa^*$ is the inverse of $\boldsymbol{\omega}_\kappa$, which represents the projection, on the κth image, of the absolute conic Ω_∞. Hence, the set of lines $\boldsymbol{l}$ that satisfy $\boldsymbol{l}^\top \boldsymbol{\omega}_\kappa^* \boldsymbol{l} = 0$ 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 $\boldsymbol{u}^\top \boldsymbol{\omega}_\kappa \boldsymbol{u} = 0$.

Eq. (36) states that the line pencil of second class represented by $\boldsymbol{\omega}_\kappa^*$ is the projection, on the κth image, of the *plane pencil of second class* represented by $\boldsymbol{\Omega}_\infty^*$, i.e., the set of planes with homogeneous coordinates $\boldsymbol{\pi}$ that satisfy $\boldsymbol{\pi}^\top \boldsymbol{\Omega}_\infty^* \boldsymbol{\pi} = 0$. This is the envelope of, or the set of tangent planes to, the absolute conic Ω_∞ regarded as a degenerate quadric surface (a 2-D "disk") (Fig. 9). This envelop is called the *dual absolute quadric*. From this projective geometric interpretation, eq. (36) is called the *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 of explanation, 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.

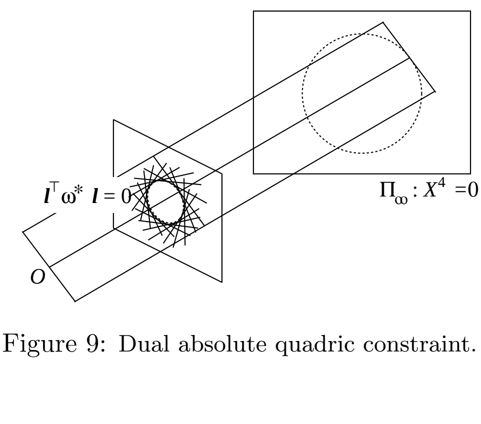


Figure 9: Dual absolute quadric constraint.

4.5 Projective Reconstruction

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

$$\lambda_{\kappa\alpha} \boldsymbol{u}_{\kappa\alpha} = \boldsymbol{P}_\kappa \boldsymbol{X}_\alpha. \tag{39}$$

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

$$\tilde{\boldsymbol{u}}_\alpha = X_\alpha^1 \tilde{\boldsymbol{p}}_1 + X_\alpha^2 \tilde{\boldsymbol{p}}_2 + X_\alpha^3 \tilde{\boldsymbol{p}}_3 + X_\alpha^4 \tilde{\boldsymbol{p}}_4, \tag{40}$$

where X_α^i is the ith component of the vector $\boldsymbol{X}_\alpha$. Eq. (40) states that the N vectors $\{\tilde{\boldsymbol{u}}_\alpha\}$ are all constrained to be in the *4-D subspace* $\mathcal{L}$ of $\mathcal{R}^{3M}$ spanned by $\{\tilde{\boldsymbol{p}}_1, \tilde{\boldsymbol{p}}_2, \tilde{\boldsymbol{p}}_3, \tilde{\boldsymbol{p}}_4\}$. 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 $\boldsymbol{X}_\alpha$ by a common nonzero constant c_α. As a result, the vector $\tilde{\boldsymbol{u}}_\alpha$ is multiplied by c_α. In order to remove this indeterminacy, we normalize $\tilde{\boldsymbol{u}}_\alpha$ to be a unit vector: $\|\tilde{\boldsymbol{u}}_\alpha\| = 1$. Then, we obtain the following iterative procedure:

1. Give initial values for the projective depths $\{\lambda_{\kappa\alpha}\}$.
2. Compute the $3M$-D vectors $\{\tilde{\boldsymbol{u}}_\alpha\}$ and fit a 4-D subspace $\mathcal{L}$ to the resulting $\{\tilde{\boldsymbol{u}}_\alpha\}$ by least squares.
3. Adjust the projective depths $\{\lambda_{\kappa\alpha}\}$ so that the square distance J_α from each $\tilde{\boldsymbol{u}}_\alpha$ 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 $\{\tilde{\boldsymbol{p}}_i\}$, determine $\boldsymbol{X}_\alpha$ by expanding $\tilde{\boldsymbol{u}}_\alpha$ 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" as 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

$$\boldsymbol{C} = \sum_{\alpha=1}^{N} \tilde{\boldsymbol{u}}_\alpha \tilde{\boldsymbol{u}}_\alpha^\top, \tag{41}$$

the subspace $\mathcal{L}$ is spanned by the eigenvectors of $\boldsymbol{C}$ 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

$$\begin{pmatrix} \tilde{\boldsymbol{u}}_1 & \cdots & \tilde{\boldsymbol{u}}_N \end{pmatrix} = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{V}^\top, \tag{42}$$

where $\boldsymbol{U}$ is a $3M \times 3M$ orthogonal matrix, $\boldsymbol{V}$ is a $N \times N$ orthogonal matrix. The matrix $\boldsymbol{\Lambda}$ consists of 0 elements except along the diagonal, where the (positive) singular values appear in descending order: only four are nonzero if the solution is exact. Then, the basis of the $\mathcal{L}$ is simply the first four columns of $\boldsymbol{U}$.

Remark 25. Historically, this problem was first solved by Sturm and Triggs [33] and Triggs [40], using SVD in analogy with the factorization for affine reconstruction to be described in the next section, and hence this method has been known as the method of *factorization*. However, they did not employ the adjustment procedure of Step 3. Rather, they updated the initial guess by using the epipolar constraints on pairwise images, computing the fundamental matrices of image pairs in advance. See Deguchi [2] for more details. The subspace constraint was implicitly used by Ueshiba and Tomita [42], who updated the projective depths by numerical search based on the perturbation theorem of SVD. It was Heyden et al. [10] who explicitly used 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 our treatment, which is based on Mahumud and Herbert [24]. Mahumud et al. [25] also presented an alternative update strategy.

Remark 26. In Step 3, it is easy to see that the square distance J_α is a quadratic form in $\{\lambda_{\kappa\alpha}\}$ [24]. So, the solution that minimizes this subject to the normalization $\|\tilde{\boldsymbol{u}}_\alpha\|^2 = \sum_{\kappa=1}^{M} \|\boldsymbol{u}_{\kappa\alpha}\|^2 \lambda_{\kappa\alpha}^2 = 1$ is directly obtained by solving a generalized eigenvalue problem [14].

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

5. 3-D Shape 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. (1), which makes the subsequent analysis very difficult. The analysis is made much simpler if this is approximated by a linear relationship in the form

$$\begin{pmatrix} u \\ v \end{pmatrix} = \boldsymbol{\Pi} \begin{pmatrix} X_c \\ Y_c \\ Z_c \end{pmatrix} + \boldsymbol{\pi}, \tag{43}$$

where $\boldsymbol{\Pi}$ 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. If

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

this approximation holds up to reasonable accuracy. Such an approximate imaging geometry is called an *affine camera*.

The elements of the matrix $\boldsymbol{\Pi}$ and the vector $\boldsymbol{\pi}$ in eq. (43) are some functions of the motion parameters $\{\boldsymbol{R}, \boldsymbol{t}\}$. In order that eq. (43) well mimic the perspective projection of eq. (1), we require the following:

(i) The frontal parallel plane passing through the world coordinate origin is projected as if by perspective projection.

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

(iii) The camera imaging does not depend on $\boldsymbol{R}$.

Requirement (i) corresponds to the assumption that the object of our interest is small and localized around the world coordinate origin $\boldsymbol{t}$. Requirement

(ii) 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 (iii) 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.

It can be shown that in order that Requirements (i)∼(iii) be satisfied, eq. (43) must have the following form [19]:

$$\begin{pmatrix} u \\ v \end{pmatrix} = \frac{1}{\zeta}\left(\begin{pmatrix} X_c \\ Y_c \end{pmatrix} + \beta(t_z - Z_c)\begin{pmatrix} t_x \\ t_y \end{pmatrix}\right). \tag{44}$$

Here, t_x, t_y, and t_z are the three components of $\boldsymbol{t}$, 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

$$\zeta = 1, \qquad \beta = 0 \tag{45}$$

Weak perspective (or **scaled orthographic**) **projection** [39, 26]

$$\zeta = \frac{t_z}{f_c}, \qquad \beta = 0 \tag{46}$$

Paraperspective projection [26]

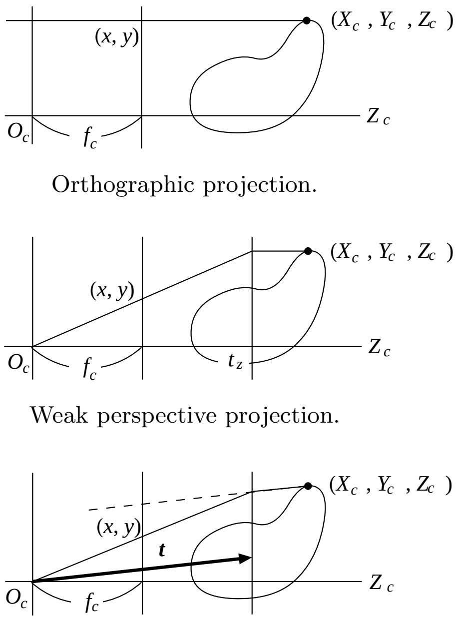
$$\zeta = \frac{t_z}{f_c}, \qquad \beta = \frac{1}{t_z} \tag{47}$$

Remark 28. The concept of affine camera and its epipolar geometry were presented by Shapiro et al. [32]. It was then 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 [38].

5.2 Affine Space Constraint

If we represent a point in the scene by the vector $\boldsymbol{X}_\alpha$ of homogeneous coordinates with the fourth component 1, eqs. (6) and (43) imply that its projection onto the κth image is represented by vector $\boldsymbol{u}_{\kappa\alpha}$ with the third component 1 in the following form:

$$\boldsymbol{u}_{\kappa\alpha} = \begin{pmatrix} \boldsymbol{\Pi}_\kappa \boldsymbol{R}_\kappa & \boldsymbol{\Pi}_\kappa \boldsymbol{t}_\kappa + \boldsymbol{\pi}_\kappa \\ 0 \quad 0 \quad 0 & 1 \end{pmatrix} \boldsymbol{X}_\alpha. \tag{48}$$



Orthographic projection.

Weak perspective projection.

Paraperspective projection.

Figure 10: Affine camera models.

Here, $\boldsymbol{\Pi}_\kappa$ and $\boldsymbol{\pi}_\kappa$ are, respectively, the values of the matrix $\boldsymbol{\Pi}$ and the vector $\boldsymbol{\pi}$ in eq. (43) for the κth image, while $\{\boldsymbol{R}_\kappa,\ \boldsymbol{t}_\kappa\}$ are the motion parameters of the κth camera.

Eq. (48) shows that an affine camera is a special case of the general projection in the form of eq. (39) with the conditions:

- the third row of the projection matrix $\boldsymbol{P}_\kappa$ is (0 0 0 1),
- the projective depths $\lambda_{\kappa\alpha}$ are all 1.

It follows that we have the following relationship corresponding to eq. (40):

$$\tilde{\boldsymbol{u}}_\alpha = X_\alpha \tilde{\boldsymbol{p}}_1 + Y_\alpha \tilde{\boldsymbol{p}}_2 + Z_\alpha \tilde{\boldsymbol{p}}_3 + \tilde{\boldsymbol{p}}_4. \tag{49}$$

As in Section 4.5, $\tilde{\boldsymbol{u}}_\alpha$ is a vector, which we simply call the *trajectory*, obtained by vertically stacking $\boldsymbol{u}_{1\alpha}$, $\boldsymbol{u}_{2\alpha}$, ..., $\boldsymbol{u}_{M\alpha}$, while $\tilde{\boldsymbol{p}}_i$ is a vector, which we call the *motion vectors*, obtained by vertically stacking the ith columns of the matrix on the right-hand side of eq. (48) for $\kappa = 1$, ..., M. However, every third component of the vector equation (49) means simply 1 = 1, so we can remove them. Then, all the trajectories $\{\tilde{\boldsymbol{u}}_\alpha\}$ and the motion vectors $\{\tilde{\boldsymbol{p}}_i\}$ become $2M$-D vectors.

Eq. (49) states that all the trajectories $\{\tilde{\boldsymbol{u}}_\alpha\}$ are constrained to be in the *3-D affine space* $\mathcal{A}$ of $\mathcal{R}^{2M}$ passing through $\tilde{\boldsymbol{p}}_4$ and spanned by the motion vectors $\{\tilde{\boldsymbol{p}}_1,\ \tilde{\boldsymbol{p}}_2,\ \tilde{\boldsymbol{p}}_3\}$. 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 [15, 16, 17, 34, 35, 36, 37] for actual applications.

5.3 Metric Constraint

3-D reconstruction based on the affine space constraint is called the *factorization method*. Let $\tilde{\boldsymbol{p}}_4$ be the centroid of the trajectories $\{\tilde{\boldsymbol{u}}_\alpha\}$, and fit to them by least squares a 3-D affine space $\mathcal{A}$ that passes through $\tilde{\boldsymbol{p}}_4$. If we let $\{\boldsymbol{q}_1, \boldsymbol{q}_2, \boldsymbol{q}_3\}$ an arbitrary orthonormal basis that span $\mathcal{A}$, we can compute $(X_\alpha, Y_\alpha, Z_\alpha)$ by identifying $\{\boldsymbol{q}_1, \boldsymbol{q}_2, \boldsymbol{q}_3\}$ with $\{\tilde{\boldsymbol{p}}_1, \tilde{\boldsymbol{p}}_2, \tilde{\boldsymbol{p}}_3\}$ and expanding each $\tilde{\boldsymbol{u}}_\alpha$ in the form of eq. (49) by least squares, just as we did in Section 4.5.

However, we can choose as $\{\tilde{\boldsymbol{p}}_1, \tilde{\boldsymbol{p}}_2, \tilde{\boldsymbol{p}}_3\}$ any other (not necessarily orthonormal) basis of the fitted space $\mathcal{A}$. This means that the reconstructed shape is related to the true shape by 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 in the form

$$\tilde{\boldsymbol{p}}_i = \sum_{j=1}^{3} A_{ji}\boldsymbol{q}_j. \tag{50}$$

The rectifying transformation matrix $\boldsymbol{A} = (A_{ij})$ is determined by the condition that each $\tilde{\boldsymbol{p}}_i$ consists of coordinates of points in the scene viewed by an affine camera in the form of eq. (48). This condition, called the *metric constraint*, is obtained, as in the case of the dual absolute quadric constraint, by eliminating $\boldsymbol{R}_\kappa$ from the projection relation of eq. (48) by using the identity $\boldsymbol{R}_\kappa\boldsymbol{R}_\kappa^\top = \boldsymbol{I}$. If we let

$$\boldsymbol{T} = \boldsymbol{A}\boldsymbol{A}^\top, \tag{51}$$

the metric constraint is written in the following form [19]:

$$\boldsymbol{Q}_\kappa^{\dagger\top}\boldsymbol{T}\boldsymbol{Q}_\kappa^\dagger = \boldsymbol{\Pi}_\kappa\boldsymbol{\Pi}_\kappa^\top. \tag{52}$$

Let $\boldsymbol{Q}$ be the $2M \times 3$ matrix with columns $\boldsymbol{q}_1$, $\boldsymbol{q}_2$, and $\boldsymbol{q}_3$ in that order. The matrix $\boldsymbol{Q}_\kappa^\dagger$ in eq. (52) is a 3×2 matrix given by

$$\boldsymbol{Q}_\kappa^\dagger = \begin{pmatrix} \boldsymbol{q}_{\kappa(1)}^\dagger & \boldsymbol{q}_{\kappa(2)}^\dagger \end{pmatrix}, \tag{53}$$

where $\boldsymbol{q}_{\kappa(1)}^\dagger$ and $\boldsymbol{q}_{\kappa(2)}^\dagger$ are the $(2\kappa-1)$th and the 2κth columns of $\boldsymbol{Q}^\top$, respectively.

As in the stratified reconstruction, we can obtain from eq. (52) equations of $\boldsymbol{T}$ by using some knowledge about the camera model, i.e., some relationships among the elements of the matrix $\boldsymbol{\Pi}_\kappa\boldsymbol{\Pi}_\kappa^\top$ on the right-hand side. Then, we can obtain the rectifying matrix $\boldsymbol{A}$ by decomposing the computed $\boldsymbol{T}$ in the form of eq. (51).

See [18] for the computational details of 3-D reconstruction based on the typical camera models of eqs. (45)∼(47). For the method directly using the general form of eq. (44), see [19].

Remark 30. As in the stratified reconstruction, the basis of the affine space $\mathcal{A}$ that optimally fits the trajectories $\{\tilde{\boldsymbol{u}}_\alpha\}$ and passes through their centroid $\tilde{\boldsymbol{p}}_4$ squares is given by the eigenvectors of the matrix

$$\boldsymbol{C} = \sum_{\alpha=1}^{N}(\tilde{\boldsymbol{u}}_\alpha - \tilde{\boldsymbol{p}}_4)(\tilde{\boldsymbol{u}}_\alpha - \tilde{\boldsymbol{p}}_4)^\top, \tag{54}$$

for the largest three (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

$$\begin{pmatrix} \tilde{\boldsymbol{u}}_1 - \tilde{\boldsymbol{p}}_4 & \cdots & \tilde{\boldsymbol{u}}_N - \tilde{\boldsymbol{p}}_4 \end{pmatrix} = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{V}^\top, \tag{55}$$

where $\boldsymbol{U}$ is a $2M \times 2M$ orthogonal matrix, $\boldsymbol{V}$ is a $N \times N$ orthogonal matrix. The matrix $\boldsymbol{\Lambda}$ consists of 0 elements except along the diagonal, where the (positive) singular values appear in descending order: only four are nonzero if the solution is exact. Then, the basis of the $\mathcal{A}$ is given by the first three columns $\boldsymbol{U}$.

Remark 31. If we let $\tilde{\boldsymbol{u}}'_\alpha = \tilde{\boldsymbol{u}}'_\alpha - \tilde{\boldsymbol{p}}_4$, eq. (49) for $\alpha = 1, ..., N$ can be rearranged in the following form:

$$\begin{pmatrix} \tilde{\boldsymbol{u}}'_1 & \cdots & \tilde{\boldsymbol{u}}'_N \end{pmatrix} = \begin{pmatrix} \tilde{\boldsymbol{p}}_1 & \tilde{\boldsymbol{p}}_2 & \tilde{\boldsymbol{p}}_3 \end{pmatrix} \begin{pmatrix} X_1 & Y_1 & Z_1 \\ \vdots & \vdots & \vdots \\ X_N & Y_N & Z_N \end{pmatrix}. \tag{56}$$

Hence, computing the solution $\{X_\alpha, Y_\alpha, Z_\alpha\}$ can be viewed as *factorizing* the *measurement* (or *observation*) *matrix* $\boldsymbol{W} = \begin{pmatrix} \tilde{\boldsymbol{u}}'_1 & \cdots & \tilde{\boldsymbol{u}}'_N \end{pmatrix}$ 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 [39], and almost all subsequent papers [23, 26] on this method adopt the above matrix factorization formulation: the basis vectors $\{\boldsymbol{q}_i\}$ are computed as the first three columns of the matrix $\boldsymbol{U}$ obtained by the SVD in the form of eq. (55), and the rectifying matrix $\boldsymbol{A} = (A_{ij})$ in eq. (50) is determined from the metric constraint of eq. (52). This convention has spread the misconception that the factorization method is the method of "matrix factorization by SVD". Because of this misconception, the projective reconstruction procedure described in Section 4.5 is also called the "factorization method".

Remark 32. Since the factorization method 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 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 are assuming that the input images are taken by affine cameras. However, 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 the factorization method is not unique. First, the absolute scale is indeterminate, which is unavoidable for any methods based on images alone (see Remark 13). The orientation of the world coordinate system is indeterminate, too, because it can be arbitrarily defined in the scene. In addition, mirror image ambiguity remains. The scale indeterminacy is evident from eq. (49): multiplying $\{\tilde{\boldsymbol{p}}_1, \tilde{\boldsymbol{p}}_2, \tilde{\boldsymbol{p}}_3\}$ by a nonzero constant c gives rise to the same effect as dividing $\{X_\alpha, Y_\alpha, Z_\alpha\}$ by c. The orientation and mirror image ambiguity arises from the fact that the rectifying matrix $\boldsymbol{A}$ is determined by eq. (51), which can be rewritten as $\boldsymbol{T} = (\pm\boldsymbol{AR})(\pm\boldsymbol{AR})^\top$ for an arbitrary rotation matrix $\boldsymbol{R}$. The indeterminacy of the rotation $\boldsymbol{R}$ corresponds to the orientation ambiguity; the indeterminacy of the sign corresponds to the mirror image ambiguity.

6. Concluding Remarks

This article 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, epipolars, epipoles, the trifocal tensor, and the quadrifocal tensor.

Then, we 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, which is necessary as an initial value for the stratified approach, by the factorization method 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.

In this article, we have described various types of 3-D reconstruction techniques, separately treated in the past, in a single framework, using unified disciplines and notations.

Acknowledgments: This work was supported in part by the Ministry of Education, Culture, Sports, Science and Technology, Japan, under a Grant in Aid for Scientific Research C (No. 17500112). The author thanks Toshio Ueshiba of the National Institute of Advanced Industrial Science and Technology, Japan, for helpful discussions about this article.

References

[1] R. Basri, Paraperspective ≡ affine, *Int. J. Comput. Vision*, **19**-2 (1996-8), 169–179.

[2] K. Deguchi, T. Sasano, H. Arai, and H. Yoshikawa, 3-D shape reconstruction from endoscope image sequences by the factorization method, *IEICE Trans. Inf. & Syst.*, **E79-D**-9 (1996-9), 1329–1336.

[3] A. P. Dempster, N. M. Laird and D. B. Rubin, Maximum likelihood from incomplete data via the EM algorithm, *J. Roy. Statist. Soc.*, **B-39** (1977), 1–38.

[4] O. Faugeras, *Three-Dimensional Computer Vision: A Geometric Viewpoint*, MIT Press, Cambridge, MA, U.S.A., 1993.

[5] O. Faugeras and Q.-T. Luong, *The Geometry of Multiple Images*, MIT Press, Cambridge, MA, U.S.A., 2001.

[6] D. A. Forsyth and J. Ponce, *Computer Vision: A Modern Approach*, Prentice Hall, Upper Saddle River, NJ, U.S.A., 2003.

[7] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, Cambridge University Press, Cambridge, U.K., 2000.

[8] A. Heyden and K. Åström, Euclidean reconstruction from image sequences with varying and unknown focal length and principal point, *Proc. IEEE Conf.*

Comput. Vision Patt. Recog., June 1997, Puerto Rico, pp. 438–443.

[9] A. Heyden and K. Åström, Flexible calibration: Minimal cases for auto-calbration, *Proc. 7th Int. Conf. Computer. Vision*, September 1999, Kerkyra, Greece, Vol. 1, pp. 350–355.

[10] A. Heyden, R. Berthilsson, and G. Sparr, An iterative factorization method for projective structure and motion from image sequences, *Image Vision Comput.*, **17**-13 (1999-11), 981–991.

[11] A. Heyden and M. Pollefeys, Multiple view geometry, in G. Medioni, S. B. Kang (Eds.), *Emerging Topics in Computer Vision*, Prentice Hall, Upper Saddle River, NJ, U.S.A., pp. 44–106.

[12] K. Kanatani, *Group-Theoretical Methods in Image Understanding*, Springer, Berlin, Germany, 1990.

[13] K. Kanatani, *Geometric Computation for Machine Vision*, Oxford University Press, Oxford, U.K., 1993.

[14] K. Kanatani, *Statistical Optimization for Geometric Computation: Theory and Practice*, Elsevier Science, Amsterdam, The Netherlands, 1996; reprinted by Dover Publications, New York, U.S.A., 2005.

[15] K. Kanatani, Motion segmentation by subspace separation and model selection, *Proc. 8th Int. Conf. Comput. Vision* July 2001, Vancouver, Canada, Vol. 2, pp. 301–306.

[16] K. Kanatani, Motion segmentation by subspace separation: Model selection and reliability evaluation, *Int. J. Image Graphics*, *2*-2 (2002-4), 179–197.

[17] K. Kanatani, Evaluation and selection of models for motion segmentation, *Proc. 7th Euro. Conf. Comput. Vision*, May 2002, Copenhagen, Denmark, Vol. 3, pp. 335–349.

[18] K. Kanatani and Y. Sugaya Factorization without factorization: complete recipe, *Mem. Fac. Eng. Okayama Univ.*, **38**-1/2 (2004-3), 61–72.

[19] K. Kanatani, Y. Sugaya and H. Ackermann, Uncalibrated factorization using a variable symmetric affine camera, *Mem. Fac. Eng. Okayama Univ.*, **40** (2006-1), to appear.

[20] Y. Ma, S. Soatto, J. Košecká, and S. S. Sastry, *An Invitation to 3-D Vision: From Images to Geometric Models*, Springer, New York, NY, U.S.A., 2004.

[21] D. Marr, *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information*, W. H. Freeman, San Francisco, CA, U.S.A., 1982

[22] G. Medioni, S. B. Kang (Eds.), *Emerging Topics in Computer Vision*, Prentice Hall, Upper Saddle River, NJ, U.S.A.

[23] T. Morita and T. Kanade, A sequential factorization method for recovering shape and motion from image sequence, *IEEE Trans. Patt. Anal. Mach. Intell.*, *19*-8 (1997-8), 858–867.

[24] S. Mahamud and M. Hebert, Iterative projective reconstruction from multiple views, *Proc. IEEE Conf. Comput. Vision Patt. Recog.*, June 2000, Hilton Head Island, SC, U.S.A., Vol. 2, pp. 430-437.

[25] S. Mahamud, M. Hebert, Y. Omori, and J. Ponce Provably-convergent iterative methods for projective structure from motion, *Proc. IEEE Conf. Comput. Vision Patt. Recog.*, December 2001, Kauai, HI, U.S.A., Vol.1, pp. 1018–1025.

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

[27] M. Pollefeys, R. Koch, and L. Van Gool, Self-calibration and metric reconstruction in spite of varying and unknown internal camera parameters, *Int. J. Comput. Vision*, **32**-2 (1999), 145–150.

[28] L. Quan, Self-calibration of an affine camera from multiple views, *Int. J. Comput. Vision*, **19**-1 (1996-7), 93–105.

[29] Y. Seo and A. Heyden, Auto-calibration from the orthogonality constraints *Proc. 15th Int. Conf. Patt. Recog.*, September 2000, Barcelona, Spain, Vol. 1, pp. 69-71.

[30] Y. Seo and H. Heyden, Auto-calibration by linear iteration using the DAC equation, *Image and Vision Computing*, **22**-11 (2004-9), 919-926

[31] Y. Seo and K.-S. Hong, A linear metric reconstruction by complex eigen-decomposition, *IEICE Trans. Inf. & Syst.*, **E84-D**-12 (2001-12), 1626-1632.

[32] L. S. Shapiro, A. Zisserman, and M. Brady, 3D motion recovery via affine epipolar geometry, *Int. J. Comput. Vision*, **16**-2 (1995-10), 147–182.

[33] P. Sturm and B. Triggs, A factorization based algorithm for multi-image projective structure and motion, *Proc. 4th Euro. Conf. Comput. Vision*, April 1996, Cambridge, U.K., Vol. 2, pp. 709–720.

[34] Y. Sugaya and K. Kanatani, Outlier removal for motion tracking by subspace separation, *IEICE Trans. Inf. & Syst.*, *E86-D*-6 (2003-6), 1095–1102.

[35] Y. Sugaya and K. Kanatani, Extending interrupted feature point tracking for 3-D affine reconstruction, *IEICE Trans. Inf. & Syst.*, *E87-D*-4 (2004-4), 1031–1038.

[36] Y. Sugaya and K. Kanatani, Multi-stage optimization for multi-body motion segmentation, *IEICE Trans. Inf. & Syst.*, *E87-D*-7 (2004-7), 1935–1942.

[37] Y. Sugaya and K. Kanatani, Extracting moving objects from a moving camera video sequence, *Mem. Fac. Eng. Okayama Univ. 39* (2005-1), 56–62.

[38] A. Sugimoto, Object recognition by combining paraperspective images, *Int. J. Comput. Vision*, **19**-2 (1996-8), 181–201.

[39] C. Tomasi and T. Kanade, Shape and motion from image streams under orthography—A factorization method, *Int. J. Comput. Vision*, **9**-2 (1992-10), 137–154.

[40] B. Triggs, Factorization methods for projective structure and motion, *Proc. IEEE Conf. Comput. Vision Patt. Recog.*, June 1996, San Francisco, CA, U.S.A., pp. 845–851.

[41] B. Triggs, Autocalibration and the absolute quadric, *Proc. IEEE Conf. Compt. Vision Patt. Recog.*, June 1997, San Juan, Puerto Rico, pp. 609–614.

[42] T. Ueshiba and F. Tomita, A factorization method for perspective and Euclidean reconstruction from multiple perspective views via iterative depth estimation, *Proc. 5th Euro. Conf. Comput. Vision*, June 1998, Freiburg, Germany, Vol. 1, pp. 296–310.

[43] G. Xu and Z. Zhang, *Epipolar Geometry in Stereo, Motion and Object Recognition*, Kluwer, Dordrecht, The Netherlands, 1996.

[44] Z. Zhang, Camera calibration, in G. Medioni, S. B. Kang (Eds.), *Emerging Topics in Computer Vision*, Prentice Hall, Upper Saddle River, NJ, U.S.A., pp. 4–40.