PAPER

Uncalibrated Factorization Using a Variable Symmetric Affine Camera

Kenichi KANATANI[†a)], **Yasuyuki SUGAYA**[††], ***Members***, ***and*** **Hanno ACKERMANN**[†], ***Nonmember***

SUMMARY In order to reconstruct 3-D Euclidean shape by the Tomasi-Kanade factorization, one needs to specify an affine camera model such as orthographic, weak perspective, and paraperspective. We present a new method that does not require any such specific models. We show that a minimal requirement for an affine camera to mimic perspective projection leads to a unique camera model, called *symmetric affine camera*, which has two free functions. We determine their values from input images by linear computation and demonstrate by experiments that an appropriate camera model is automatically selected.

key words: *factorization, structure from motion, affine camera, self-calibration, video image analysis*

1. Introduction

One of the best known techniques for 3-D reconstruction from feature point tracking through a video stream is the Tomasi-Kanade *factorization* [17], which computes the 3-D shape of the scene by approximating the camera imaging by an affine transformation. The computation consists of linear calculus alone without involving iterations (see [8] for the computational details). The solution is sufficiently accurate for many practical purposes and is also used as an initial solution for iterative reconstruction based on perspective projection [3].

If the camera model is not specified, other than being affine, the 3-D shape is computed only up to an affine transformation, known as *affine reconstruction*. For computing the correct shape (*Euclid reconstruction*), we need to specify the camera model. For this, *orthographic*, *weak perspective*, and *paraperspective* projections have been used [10]. However, the reconstruction accuracy does not necessarily follow that order [2]. To find the best camera models in a particular circumstance, one needs to choose the best one *a posteriori*. Is there any method for automatically selecting an appropriate camera model? This is the motivation of this paper.

Basri [1] pointed out that any affine camera can be regarded as paraperspective projection if the scene and the reference point are appropriately transformed, and Sugimoto [16] exploited this fact for object recognition from a single image based on affine invariants. Shapiro et al. [12] described the epipolar geometry for affine cameras and 3-D reconstruction methods based on it. Quan [11] showed that a generic affine camera has three intrinsic parameters and that they can be determined by self-calibration if the same camera is moved (i.e., the three intrinsic parameters are unchanged).

This paper extends Quan's result to variable intrinsic parameters. However, they cannot be determined if the camera is completely arbitrary from frame to frame. The situation is similar to the *dual absolute quadric constraint* [3] for upgrading projective reconstruction to Euclidean, which cannot be imposed unless something is known about the camera (e.g., zero skew).

In this paper, we show that minimal requirements for the general affine camera to mimic perspective projection leads to a unique camera model, which we call *symmetric affine camera*, having two free functions of motion parameters; their specific choices result in the orthographic, weak perspective, and paraperspective models.

Here, however, we do not specify such function forms. We determine their values directly from input images. All the computation is linear just as in the case of the traditional factorization method, and an appropriate model is automatically selected.

Section 2 summarizes fundamentals of affine cameras, and Sect. 3 summarizes the metric constraint. In Sect. 4, we derive our symmetric affine camera model. Section 5 describes the procedure for 3-D reconstruction using our model. Section 6 shows experiments, and Sect. 7 concludes this paper.

2. Affine Cameras

Consider a camera-based XYZ coordinate system with the origin O at the projection center and the Z axis along the optical axis. *Perspective projection* maps a point (X, Y, Z) in the scene onto a point in the image with coordinates (x, y) such that

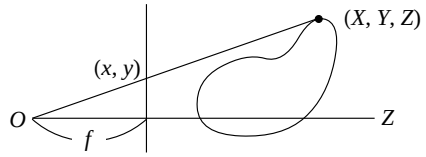
$$x = f\frac{X}{Z}, \qquad y = f\frac{Y}{Z}, \tag{1}$$

where f is a constant called the *focal length* (Fig. 1(a)).

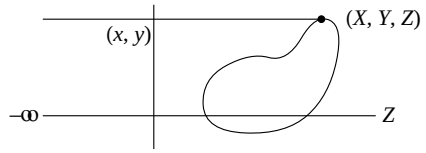
Consider a world coordinate system fixed to the scene, and let $\boldsymbol{t}$ and $\{\boldsymbol{i}, \boldsymbol{j}, \boldsymbol{k}\}$ be its origin and the orthonormal basis vectors with respect to the camera coordinate system. For convenience (with some risk

Manuscript received February 16, 2006.
Manuscript revised October 12, 2006.

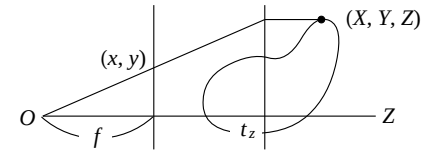
[†]The authors are with the Department of Computer Science, Okayama University, Okayama-shi, 700-8530 Japan.
[††]The author is with the Department of Information and Computer Sciences, Toyohashi University of Technology, Toyohashi-shi, 441-8580 Japan.
a) E-mail: kanatani@suri.it.okayama-u.ac.jp
DOI: 10.1093/ietisy/e90-d.5.851



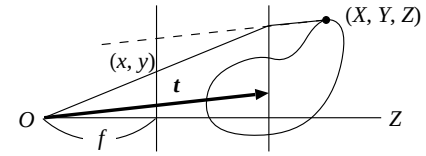
(a) Perspective projection.



(b) Orthographic projection.



(c) Weak perspective projection.



(d) Paraperspective projection.

Fig. 1 Camera models.

of confusion), we call $\boldsymbol{t}$ the *translation*, the matrix $\boldsymbol{R} = (\boldsymbol{i}\ \ \boldsymbol{j}\ \ \boldsymbol{k})$ having $\{\boldsymbol{i}, \boldsymbol{j}, \boldsymbol{k}\}$ as columns the *rotation*, and $\{\boldsymbol{t}, \boldsymbol{R}\}$ the *motion parameters*. Unlike the traditional formulation [10], [17], all the subsequent descriptions are based on the camera coordinate system.

As is well known, the imaging can be approximated by an *affine camera* [12] in the form

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} \Pi_{11} & \Pi_{12} & \Pi_{13} \\ \Pi_{21} & \Pi_{22} & \Pi_{23} \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} + \begin{pmatrix} \pi_1 \\ \pi_2 \end{pmatrix}, \tag{2}$$

if (i) the object of our interest is localized around the world coordinate origin $\boldsymbol{t}$, and (ii) the size of that object is small as compared with $\|\boldsymbol{t}\|$. We call the 2×3 matrix $\boldsymbol{\Pi} = (\Pi_{ij})$ and the 2-D vector $\boldsymbol{\pi} = (\pi_i)$ the *projection matrix* and the *projection vector*, respectively; their elements are "functions" of the motion parameters $\{\boldsymbol{t}, \boldsymbol{R}\}$. Unlike Quan [11], we do not separate "intrinsic" parameters from the motion parameters (or "extrinsic" parameters); the intrinsic parameters are *implicitly* defined via the *functional forms* of $\{\boldsymbol{\Pi}, \boldsymbol{\pi}\}$ on $\{\boldsymbol{t}, \boldsymbol{R}\}$. Typical examples are:

Orthographic projection (Fig. 1(b))

$$\boldsymbol{\Pi} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}, \qquad \boldsymbol{\pi} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}. \tag{3}$$

Weak perspective projection (Fig. 1(c))

$$\boldsymbol{\Pi} = \begin{pmatrix} f/t_z & 0 & 0 \\ 0 & f/t_z & 0 \end{pmatrix}, \qquad \boldsymbol{\pi} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}. \tag{4}$$

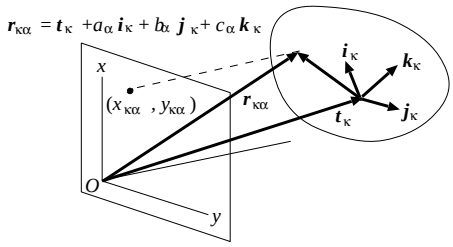


Fig. 2 Camera-based description of the world coordinate system.

Paraperspective projection (Fig. 1(d)†)

$$\boldsymbol{\Pi} = \begin{pmatrix} f/t_z & 0 & -ft_x/t_z^2 \\ 0 & f/t_z & -ft_x/t_z^2 \end{pmatrix}, \boldsymbol{\pi} = \begin{pmatrix} ft_x/t_z \\ ft_y/t_z \end{pmatrix}. \tag{5}$$

We say that the affine camera is *uncalibrated* if $\{\boldsymbol{\Pi}, \boldsymbol{\pi}\}$ contain unknown parameters.

Suppose we track N feature points over M frames. Identifying the frame number κ with "time", let $\boldsymbol{t}_\kappa$ and $\{\boldsymbol{i}_\kappa, \boldsymbol{j}_\kappa, \boldsymbol{k}_\kappa\}$ be the origin and the basis vectors of the world coordinate system at time κ (Fig. 2). The 3-D position of the αth point at time κ has the form

$$\boldsymbol{r}_{\kappa\alpha} = \boldsymbol{t}_\kappa + a_\alpha \boldsymbol{i}_\kappa + b_\alpha \boldsymbol{j}_\kappa + c_\alpha \boldsymbol{k}_\kappa. \tag{6}$$

Under the affine camera of Eq. (2), its image coordinates $(x_{\kappa\alpha}, y_{\kappa\alpha})$ are given by

$$\begin{pmatrix} x_{\kappa\alpha} \\ y_{\kappa\alpha} \end{pmatrix} = \tilde{\boldsymbol{t}}_\kappa + a_\alpha \tilde{\boldsymbol{i}}_\kappa + b_\alpha \tilde{\boldsymbol{j}}_\kappa + c_\alpha \tilde{\boldsymbol{k}}_\kappa, \tag{7}$$

where $\tilde{\boldsymbol{t}}_\kappa$, $\tilde{\boldsymbol{i}}_\kappa$, $\tilde{\boldsymbol{j}}_\kappa$, and $\tilde{\boldsymbol{k}}_\kappa$ are 2-D vectors defined by

$$\begin{aligned} \tilde{\boldsymbol{t}}_\kappa &= \boldsymbol{\Pi}_\kappa \boldsymbol{t}_\kappa + \boldsymbol{\pi}_\kappa, & \tilde{\boldsymbol{i}}_\kappa &= \boldsymbol{\Pi}_\kappa \boldsymbol{i}_\kappa, \\ \tilde{\boldsymbol{j}}_\kappa &= \boldsymbol{\Pi}_\kappa \boldsymbol{j}_\kappa, & \tilde{\boldsymbol{k}}_\kappa &= \boldsymbol{\Pi}_\kappa \boldsymbol{k}_\kappa. \end{aligned} \tag{8}$$

Here, $\boldsymbol{\Pi}_\kappa$ and $\boldsymbol{\pi}_\kappa$ are the projection matrix and the projective vector, respectively, at time κ. The motion history of the αth point is represented by a vector

$$\boldsymbol{p}_\alpha = \begin{pmatrix} x_{1\alpha} & y_{1\alpha} & x_{2\alpha} & y_{2\alpha} & \cdots & x_{M\alpha} & y_{M\alpha} \end{pmatrix}^\top, \tag{9}$$

which we simply call the *trajectory* of that point. Using Eq. (7), we can write

$$\boldsymbol{p}_\alpha = \boldsymbol{m}_0 + a_\alpha \boldsymbol{m}_1 + b_\alpha \boldsymbol{m}_2 + c_\alpha \boldsymbol{m}_3, \tag{10}$$

where $\boldsymbol{m}_0$, $\boldsymbol{m}_1$, $\boldsymbol{m}_2$, and $\boldsymbol{m}_3$ are the following $2M$-D vectors, respectively:

$$\begin{pmatrix} \tilde{\boldsymbol{t}}_1 \\ \tilde{\boldsymbol{t}}_2 \\ \vdots \\ \tilde{\boldsymbol{t}}_M \end{pmatrix}, \begin{pmatrix} \tilde{\boldsymbol{i}}_1 \\ \tilde{\boldsymbol{i}}_2 \\ \vdots \\ \tilde{\boldsymbol{i}}_M \end{pmatrix}, \begin{pmatrix} \tilde{\boldsymbol{j}}_1 \\ \tilde{\boldsymbol{j}}_2 \\ \vdots \\ \tilde{\boldsymbol{j}}_M \end{pmatrix}, \begin{pmatrix} \tilde{\boldsymbol{k}}_1 \\ \tilde{\boldsymbol{k}}_2 \\ \vdots \\ \tilde{\boldsymbol{k}}_M \end{pmatrix}. \tag{11}$$

†Some authors arbitrarily locate on the plane $Z = t_z$ a "reference point", with respect to which paraperspective projection is defined (e.g., [1]). In this paper, the reference point is always at the world coordinate origin. See Sect. 3 and 4 for this reason.

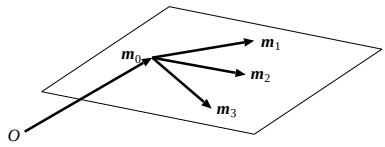


Fig. 3 Affine space constraint.

Thus, all the trajectories $\{\boldsymbol{p}_\alpha\}$ are constrained to be in the 3-D affine space $\mathcal{A}$ in $\mathcal{R}^{2M}$ passing through $\boldsymbol{m}_0$ and spanned by $\boldsymbol{m}_1$, $\boldsymbol{m}_2$, and $\boldsymbol{m}_3$ (Fig. 3). This fact is known as the *affine space constraint* [7], which plays a central role in many applications including outlier removal [13], missing data recovery [14], and multibody motion segmentation [6], [15].

3. Metric Constraint

In order to approximate perspective projection by an affine camera, we place the origin of the world coordinate system at the centroid of the N feature points. They should concentrate in a small region around the world coordinate origin for the affine camera modeling to be valid†. We assume this hereafter.

By the definition of the world coordinate origin, we have $\sum_{\alpha=1}^N a_\alpha = \sum_{\alpha=1}^N b_\alpha = \sum_{\alpha=1}^N c_\alpha = 0$, so we have from Eq. (10)

$$\frac{1}{N}\sum_{\alpha=1}^{N}\boldsymbol{p}_\alpha = \boldsymbol{m}_0, \tag{12}$$

i.e., $\boldsymbol{m}_0$ is the centroid of the trajectories $\{\boldsymbol{p}_\alpha\}$ in $\mathcal{R}^{2M}$. It follows that the deviation $\boldsymbol{p}'_\alpha$ of $\boldsymbol{p}_\alpha$ from the centroid $\boldsymbol{m}_0$ is written as††

$$\boldsymbol{p}'_\alpha = \boldsymbol{p}_\alpha - \boldsymbol{m}_0 = a_\alpha\boldsymbol{m}_1 + b_\alpha\boldsymbol{m}_2 + c_\alpha\boldsymbol{m}_3, \tag{13}$$

which means that $\{\boldsymbol{p}'_\alpha\}$ are constrained to be in the 3-D subspace $\mathcal{L}$ in $\mathcal{R}^{2M}$. Hence, the matrix

$$\boldsymbol{C} = \sum_{\alpha=1}^{N}\boldsymbol{p}'_\alpha\boldsymbol{p}'^{\top}_\alpha \tag{14}$$

has rank 3, having three nonzero eigenvalues. The corresponding unit eigenvectors $\{\boldsymbol{u}_1, \boldsymbol{u}_2, \boldsymbol{u}_3\}$ constitute an orthonormal basis of the subspace $\mathcal{L}$, and $\boldsymbol{m}_1$, $\boldsymbol{m}_2$, and $\boldsymbol{m}_3$ are expressed as a linear combination of them in the form

$$\boldsymbol{m}_i = \sum_{j=1}^{3} A_{ji}\boldsymbol{u}_j. \tag{15}$$

Let $\boldsymbol{M}$ and $\boldsymbol{U}$ be the $2M \times 3$ matrices consisting of $\{\boldsymbol{m}_1, \boldsymbol{m}_2, \boldsymbol{m}_3\}$ and $\{\boldsymbol{u}_1, \boldsymbol{u}_2, \boldsymbol{u}_3\}$ as columns:

$$\boldsymbol{M} = \begin{pmatrix}\boldsymbol{m}_1 & \boldsymbol{m}_2 & \boldsymbol{m}_3\end{pmatrix}, \quad \boldsymbol{U} = \begin{pmatrix}\boldsymbol{u}_1 & \boldsymbol{u}_2 & \boldsymbol{u}_3\end{pmatrix}. \tag{16}$$

From Eq. (15), $\boldsymbol{M}$ and $\boldsymbol{U}$ are related by the matrix $\boldsymbol{A} = (A_{ij})$ in the form†††:

$$\boldsymbol{M} = \boldsymbol{U}\boldsymbol{A}. \tag{17}$$

The rectifying matrix $\boldsymbol{A} = (A_{ij})$ is determined so that $\boldsymbol{m}_1$, $\boldsymbol{m}_2$ and $\boldsymbol{m}_3$ in Eq. (11) are projections of the orthonormal basis vectors $\{\boldsymbol{i}_\kappa, \boldsymbol{j}_\kappa, \boldsymbol{k}_\kappa\}$ in the form of Eqs. (8). From Eqs. (8), we obtain

$$\begin{pmatrix}\tilde{\boldsymbol{i}}_\kappa & \tilde{\boldsymbol{j}}_\kappa & \tilde{\boldsymbol{k}}_\kappa\end{pmatrix} = \boldsymbol{\Pi}_\kappa\begin{pmatrix}\boldsymbol{i}_\kappa & \boldsymbol{j}_\kappa & \boldsymbol{k}_\kappa\end{pmatrix} = \boldsymbol{\Pi}_\kappa\boldsymbol{R}_\kappa, \tag{18}$$

where $\boldsymbol{R}_\kappa$ is the rotation at time κ. Let $\boldsymbol{m}^\dagger_{\kappa(a)}$ be the $(2(\kappa-1)+a)$th column of the transpose $\boldsymbol{M}^\top$ of the matrix $\boldsymbol{M}$ in Eqs. (16), $\kappa = 1, ..., M$, $a = 1, 2$. The transpose on both sides of Eq. (18) yields

$$\boldsymbol{R}^\top_\kappa\boldsymbol{\Pi}^\top_\kappa = \begin{pmatrix}\boldsymbol{m}^\dagger_{\kappa(1)} & \boldsymbol{m}^\dagger_{\kappa(2)}\end{pmatrix}. \tag{19}$$

Equation (17) implies $\boldsymbol{M}^\top = \boldsymbol{A}^\top\boldsymbol{U}^\top$, so if we let $\boldsymbol{u}^\dagger_{\kappa(a)}$ be the $(2(\kappa-1)+a)$th column of the transpose $\boldsymbol{U}^\top$ of the matrix $\boldsymbol{U}$ in Eqs. (16), we obtain

$$\boldsymbol{m}^\dagger_{\kappa(a)} = \boldsymbol{A}^\top\boldsymbol{u}^\dagger_{\kappa(a)}. \tag{20}$$

Substituting this, we can rewrite Eq. (19) as

$$\boldsymbol{R}^\top_\kappa\boldsymbol{\Pi}^\top_\kappa = \boldsymbol{A}^\top\begin{pmatrix}\boldsymbol{u}^\dagger_{\kappa(1)} & \boldsymbol{u}^\dagger_{\kappa(2)}\end{pmatrix}. \tag{21}$$

Let $\boldsymbol{U}^\dagger_\kappa$ be the 3×2 matrix having $\boldsymbol{u}^\dagger_{\kappa(1)}$ and $\boldsymbol{u}^\dagger_{\kappa(2)}$ as columns:

$$\boldsymbol{U}^\dagger_\kappa = \begin{pmatrix}\boldsymbol{u}^\dagger_{\kappa(1)} & \boldsymbol{u}^\dagger_{\kappa(2)}\end{pmatrix}. \tag{22}$$

From Eq. (21), we have $\boldsymbol{U}^{\dagger\top}_\kappa\boldsymbol{A}\boldsymbol{A}^\top\boldsymbol{U}^\dagger_\kappa = \boldsymbol{\Pi}_\kappa\boldsymbol{R}_\kappa\boldsymbol{R}^\top_\kappa\boldsymbol{\Pi}^\top_\kappa$. Since $\boldsymbol{R}_\kappa$ is a rotation matrix, we have

$$\boldsymbol{U}^{\dagger\top}_\kappa\boldsymbol{T}\boldsymbol{U}^\dagger_\kappa = \boldsymbol{\Pi}_\kappa\boldsymbol{\Pi}^\top_\kappa, \tag{23}$$

where we define the *metric matrix* $\boldsymbol{T}$ as follows:

$$\boldsymbol{T} = \boldsymbol{A}\boldsymbol{A}^\top. \tag{24}$$

given by Quan [11]. If we take out the elements on both sides, we have the following three expressions:

$$\begin{aligned}(\boldsymbol{u}^\dagger_{\kappa(1)}, \boldsymbol{T}\boldsymbol{u}^\dagger_{\kappa(1)}) &= \sum_{i=1}^{3}\Pi^2_{1i\kappa},\\ (\boldsymbol{u}^\dagger_{\kappa(2)}, \boldsymbol{T}\boldsymbol{u}^\dagger_{\kappa(2)}) &= \sum_{i=1}^{3}\Pi^2_{2i\kappa},\\ (\boldsymbol{u}^\dagger_{\kappa(1)}, \boldsymbol{T}\boldsymbol{u}^\dagger_{\kappa(2)}) &= \sum_{i=1}^{3}\Pi_{1i\kappa}\Pi_{2i\kappa}.\end{aligned} \tag{25}$$

---

[a] For this reason, we do not allow arbitrary rigid motions of the scene as in [1].

[b] In the traditional formulation [10], [17], vectors $\{\boldsymbol{p}'_\alpha\}$ are combined into the *measurement* (or *observation*) *matrix*, $\boldsymbol{W} = \begin{pmatrix}\boldsymbol{p}'_1 & \cdots & \boldsymbol{p}'_N\end{pmatrix}$, and the object coordinates $\{(a_\alpha, b_\alpha, c_\alpha)\}$ are combined into the *shape matrix*, $\boldsymbol{S} = \begin{pmatrix}a_1 & \cdots & a_N\\ b_1 & \cdots & b_N\\ c_1 & \cdots & c_N\end{pmatrix}$. Then, Eq. (13) is written as $\boldsymbol{W} = \boldsymbol{M}\boldsymbol{S}$, where $\boldsymbol{M}$, the *motion matrix*, is defined by the first of Eqs. (16).

[c] In the traditional formulation [10], [17], the measurement matrix $\boldsymbol{W}$ is decomposed by the singular value decomposition into $\boldsymbol{W} = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{V}^\top$, and the motion and the shape matrices $\boldsymbol{M}$ and $\boldsymbol{S}$ are set to $\boldsymbol{M} = \boldsymbol{U}\boldsymbol{A}$ an $\boldsymbol{S} = \boldsymbol{A}^{-1}\boldsymbol{\Lambda}\boldsymbol{V}^\top$ via a nonsingular matrix $\boldsymbol{A}$.

If we let, instead of Eq. (15), simply $\boldsymbol{m}_i = \boldsymbol{u}_i$, $i = 1, 2, 3$, we can still reconstruct the 3-D shape, but it is a deformation of the true shape by some affine transformation, known as *affine reconstruction*[†]. In order to restore the true shape (*Euclidean reconstruction*[††]), one needs to rectify the basis $\{\boldsymbol{u}_1, \boldsymbol{u}_2, \boldsymbol{u}_3\}$ of the subspace $\mathcal{L}$ by some linear transformation $\boldsymbol{A}$, and Eq. (23) gives the constraint on it. In this sense, Eq. (23) corresponds to the *dual absolute quadric constraint* [3] on the homography that rectifies the basis of projective reconstruction to Euclidean.

We now show that (i) we can restrict the camera model so that it has *two* free functions and that (ii) they can be *linearly* estimated.

4. Symmetric Affine Cameras

We impose minimal requirements that Eq. (2) mimic perspective projection.

Requirement 1. The camera imaging does not depend on $\boldsymbol{R}$.

Requirement 2. The camera imaging is symmetric around the Z-axis.

Requirement 3. The frontal parallel plane passing through the world coordinate origin is projected as if by perspective projection.

Requirement 1 is a logical consequence of the fact that the orientation of the world coordinate system can be defined arbitrarily, since such indeterminate parameterization should not affect the actual observation. Requirement 2 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 3 is a minimal requirement that the image look like perspective within the affine camera framework.

A point on the plane $Z = t_z$ is written as (X, Y, t_z), so Requirement 3 implies

$$\begin{pmatrix} \phi X/t_z \\ \phi Y/t_z \end{pmatrix} = \begin{pmatrix} \Pi_{11} & \Pi_{12} \\ \Pi_{21} & \Pi_{22} \end{pmatrix} \begin{pmatrix} X \\ Y \end{pmatrix} + t_z \begin{pmatrix} \Pi_{13} \\ \Pi_{23} \end{pmatrix} + \begin{pmatrix} \pi_1 \\ \pi_2 \end{pmatrix}, \quad (26)$$

for some ϕ, which is a function of $\boldsymbol{t}$ due to Requirement 1. Since this should hold for arbitrary X and Y, we obtain

$$\begin{aligned} &\Pi_{11} = \Pi_{22} = \frac{\phi}{t_z}, \qquad \Pi_{12} = \Pi_{21} = 0, \\ &t_z\Pi_{13} + \pi_1 = 0, \qquad t_z\Pi_{23} + \pi_2 = 0, \end{aligned} \quad (27)$$

which reduces Eq. (2) to

$$\begin{pmatrix} x \\ y \end{pmatrix} = \frac{\phi}{t_z} \begin{pmatrix} X \\ Y \end{pmatrix} - (t_z - Z) \begin{pmatrix} \Pi_{13} \\ \Pi_{23} \end{pmatrix}, \quad (28)$$

where Π_{13} and Π_{23} are some functions of $\boldsymbol{t}$.

Let $\mathcal{R}(\theta)$ be the 2-D rotation matrix by angle θ:

$$\mathcal{R}(\theta) = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}. \quad (29)$$

Requirement 2 is written as

$$\mathcal{R}(\theta) \begin{pmatrix} x \\ y \end{pmatrix} = \frac{\phi'}{t_z} \mathcal{R}(\theta) \begin{pmatrix} X \\ Y \end{pmatrix} - (t_z - Z) \begin{pmatrix} \Pi'_{13} \\ \Pi'_{23} \end{pmatrix}, \quad (30)$$

where ϕ', Π'_{13}, and Π'_{23} are the values of the functions ϕ, Π_{13}, and Π_{23}, respectively, obtained by replacing t_x and t_y by $t_x \cos\theta - t_y \sin\theta$ and $t_x \sin\theta + t_y \cos\theta$, respectively. Multiplying Eq. (28) by $\mathcal{R}(\theta)$ on both sides, we obtain

$$\mathcal{R}(\theta) \begin{pmatrix} x \\ y \end{pmatrix} = \frac{\phi}{t_z} \mathcal{R}(\theta) \begin{pmatrix} X \\ Y \end{pmatrix} - (t_z - Z) \mathcal{R}(\theta) \begin{pmatrix} \Pi_{13} \\ \Pi_{23} \end{pmatrix}. \quad (31)$$

Comparing Eqs. (30) and (31), we conclude that the equalities

$$\phi' = \phi, \qquad \begin{pmatrix} \Pi'_{13} \\ \Pi'_{23} \end{pmatrix} = \mathcal{R}(\theta) \begin{pmatrix} \Pi_{13} \\ \Pi_{23} \end{pmatrix} \quad (32)$$

should hold identically for an arbitrary θ. According to the theory of invariants [4], this implies that ϕ is a function of $t_x^2 + t_y^2$ and t_z only and that

$$\begin{pmatrix} \Pi_{13} \\ \Pi_{23} \end{pmatrix} = c \begin{pmatrix} t_x \\ t_y \end{pmatrix}, \quad (33)$$

where c is an arbitrary function of $t_x^2 + t_y^2$ and t_z.

Now, if we define

$$\zeta = \frac{t_z}{\phi}, \qquad \beta = -\frac{ct_z}{\phi}, \quad (34)$$

Eq. (28) is written as

$$\begin{pmatrix} x \\ y \end{pmatrix} = \frac{1}{\zeta} \left(\begin{pmatrix} X \\ Y \end{pmatrix} + \beta(t_z - Z) \begin{pmatrix} t_x \\ t_y \end{pmatrix} \right). \quad (35)$$

The corresponding projection matrix $\boldsymbol{\Pi}$ and the projection vector $\boldsymbol{\pi}$ are

$$\boldsymbol{\Pi} = \begin{pmatrix} 1/\zeta & 0 & -\beta t_x/\zeta \\ 0 & 1/\zeta & -\beta t_y/\zeta \end{pmatrix}, \quad \boldsymbol{\pi} = \begin{pmatrix} \beta t_x t_z/\zeta \\ \beta t_y t_z/\zeta \end{pmatrix}. \quad (36)$$

Since c and ϕ are arbitrary functions of $t_x^2 + t_y^2$ and t_z, so are ζ and β. We observe:

- Equation (35) reduces to the paraperspective projection of Eqs. (5) if we choose

$$\zeta = \frac{t_z}{f}, \qquad \beta = \frac{1}{t_z}. \quad (37)$$

- Equation (35) reduces to the weak perspective projection of Eqs. (4) if we choose

$$\zeta = \frac{t_z}{f}, \qquad \beta = 0. \quad (38)$$

[†]We are assuming an affine camera model. If we use perspective images, the resulting shape may not be affine reconstruction, of course.

[††]Strictly, this should be called "similarity reconstruction", since the absolute scale is indeterminate. However, this term is widely used.

- Equation (35) reduces to the orthographic projection of Eqs. (3) if we choose

$$\zeta = 1, \qquad \beta = 0. \tag{39}$$

Thus, Eq. (35) includes the traditional affine camera models as special instances and is the *only possible form* that satisfies Requirements 1, 2, and 3.

However, we need not define the functions ζ and β in any particular form; we can regard them as *time varying unknowns* and determine their values by self-calibration. This is made possible by the fact that *at most two* time varying unknowns can be eliminated from the metric constraint of Eqs. (25).

5. Procedure for 3-D Reconstruction

3-D Euclidean reconstruction using Eq. (35) goes just as using the traditional camera models (see [9] for the details):

1. We fit a 3-D affine space $\mathcal{A}$ to the trajectories $\{\boldsymbol{p}_\alpha\}$ by least squares. Namely, we compute the centroid $\boldsymbol{m}_0$ by Eq. (12) and compute the unit eigenvectors $\{\boldsymbol{u}_1, \boldsymbol{u}_2, \boldsymbol{u}_3\}$ of the matrix $\boldsymbol{C}$ in Eq. (14) for the largest three eigenvalues[†].
2. We eliminate time varying unknowns from the the metric constraint of Eqs. (25) and solve for the metric matrix $\boldsymbol{T}$ by least squares. To be specific, substitution of Eqs. (36) into Eqs. (25) yields

$$\begin{aligned}
(\boldsymbol{u}^\dagger_{\kappa(1)}, \boldsymbol{T}\boldsymbol{u}^\dagger_{\kappa(1)}) &= \frac{1}{\zeta_\kappa^2} + \beta_\kappa^2 \tilde{t}_{x\kappa}^2, \\
(\boldsymbol{u}^\dagger_{\kappa(2)}, \boldsymbol{T}\boldsymbol{u}^\dagger_{\kappa(2)}) &= \frac{1}{\zeta_\kappa^2} + \beta_\kappa^2 \tilde{t}_{y\kappa}^2 \\
(\boldsymbol{u}^\dagger_{\kappa(1)}, \boldsymbol{T}\boldsymbol{u}^\dagger_{\kappa(2)}) &= \beta_\kappa^2 \tilde{t}_{x\kappa}\tilde{t}_{y\kappa},
\end{aligned} \tag{40}$$

where $\tilde{t}_{x\kappa}$ and $\tilde{t}_{y\kappa}$ are, respectively, the $(2(\kappa-1)+1)$th and the $(2(\kappa-1)+2)$th components of the centroid $\boldsymbol{m}_0$. Eliminating ζ_κ and β_κ, we obtain

$$\begin{aligned}
A_\kappa(\boldsymbol{u}^\dagger_{\kappa(1)}, \boldsymbol{T}\boldsymbol{u}^\dagger_{\kappa(1)}) - C_\kappa(\boldsymbol{u}^\dagger_{\kappa(1)}, \boldsymbol{T}\boldsymbol{u}^\dagger_{\kappa(2)}) \\
- A_\kappa(\boldsymbol{u}^\dagger_{\kappa(2)}, \boldsymbol{T}\boldsymbol{u}^\dagger_{\kappa(2)}) = 0,
\end{aligned} \tag{41}$$

where $A_\kappa = \tilde{t}_{x\kappa}\tilde{t}_{y\kappa}$ and $C_\kappa = \tilde{t}_{x\kappa}^2 - \tilde{t}_{y\kappa}^2$. This is a linear constraint on $\boldsymbol{T}$, so we can determine $\boldsymbol{T}$ by least squares. Then, we can determine $1/\zeta_\kappa^2$ and β_κ^2 from Eqs. (40) by least squares.
3. We decompose the metric matrix $\boldsymbol{T}$ into the rectifying matrix $\boldsymbol{A}$ in the form of Eq. (24), and compute the vectors $\boldsymbol{m}_1$, $\boldsymbol{m}_2$, and $\boldsymbol{m}_3$ from Eq. (15).
4. We compute the motion parameters $\{\boldsymbol{t}_\kappa, \boldsymbol{R}_\kappa\}$. The translation components $t_{x\kappa}$ and $t_{y\kappa}$ are given by the first of Eqs. (8) in the form of $t_{x\kappa} = \zeta_\kappa \tilde{t}_{x\kappa}$ and $t_{y\kappa} = \zeta_\kappa \tilde{t}_{y\kappa}$. The three rows $\boldsymbol{r}_{\kappa(1)}$, $\boldsymbol{r}_{\kappa(2)}$, and $\boldsymbol{r}_{\kappa(3)}$ of the rotation $\boldsymbol{R}_\kappa$ are given by solving the following linear equations:

$$\begin{aligned}
\boldsymbol{r}_{\kappa(1)} - \beta_\kappa t_{x\kappa}\boldsymbol{r}_{\kappa(3)} &= \zeta_\kappa \boldsymbol{m}^\dagger_{\kappa(1)}, \\
\boldsymbol{r}_{\kappa(2)} - \beta_\kappa t_{y\kappa}\boldsymbol{r}_{\kappa(3)} &= \zeta_\kappa \boldsymbol{m}^\dagger_{\kappa(2)},
\end{aligned}$$

$$\beta_\kappa t_{x\kappa}\boldsymbol{r}_{\kappa(1)} + \beta_\kappa t_{y\kappa}\boldsymbol{r}_{\kappa(2)} + \boldsymbol{r}_{\kappa(3)} = \zeta_\kappa^2 \boldsymbol{m}^\dagger_{\kappa(1)} \times \boldsymbol{m}^\dagger_{\kappa(2)}. \tag{42}$$

The resulting matrix $\begin{pmatrix} \boldsymbol{r}_{\kappa(1)} & \boldsymbol{r}_{\kappa(2)} & \boldsymbol{r}_{\kappa(3)} \end{pmatrix}$ may not be strictly orthogonal, so we compute its singular value decomposition $\boldsymbol{V}_\kappa \boldsymbol{\Lambda}_\kappa \boldsymbol{U}_\kappa^\top$ and let $\boldsymbol{R}_\kappa = \boldsymbol{U}_\kappa \boldsymbol{V}_\kappa^\top$ [5].
5. We recompute the vectors $\boldsymbol{m}_1$, $\boldsymbol{m}_2$, and $\boldsymbol{m}_3$ in the form of Eqs. (11) using the computed rotations $\boldsymbol{R}_\kappa = \begin{pmatrix} \boldsymbol{i}_\kappa & \boldsymbol{j}_\kappa & \boldsymbol{k}_\kappa \end{pmatrix}$.
6. We compute the object coordinates $(a_\alpha, b_\beta, c_\beta)$ of each point by least-squares expansion of $\boldsymbol{p}'_\alpha$ in the form of Eq. (13). The solution is given by $\boldsymbol{M}^- \boldsymbol{p}_\alpha$, using the pseudoinverse $\boldsymbol{M}^-$ of $\boldsymbol{M}$.

However, the following indeterminacy remains:

1. Another solution is obtained by multiplying all $\{\boldsymbol{t}_\kappa\}$ and $\{(a_\alpha, b_\alpha, c_\alpha)\}$ by a common constant.
2. Another solution is obtained by multiplying all $\{\boldsymbol{R}_\kappa\}$ by a common rotation. The object coordinates $\{(a_\alpha, b_\alpha, c_\alpha)\}$ are rotated accordingly.
3. Each solution has its mirror image solution. The mirror image rotation $\boldsymbol{R}'_\kappa$ is the rotation $\boldsymbol{R}_\kappa$ followed by a rotation around axis $(\beta_\kappa t_{x\kappa}, \beta_\kappa t_{y\kappa}, 1)$ by angle 2π. The object coordinates $\{(a_\alpha, b_\alpha, c_\alpha)\}$ change their signs.
4. *The absolute depth t_z of the world coordinate origin is indeterminate.*

Item 1 is the fundamental ambiguity of 3-D reconstruction from images, meaning that a large motion of a large object in the distance is indistinguishable from a small motion of a small object nearby. Item 2 reflects the fact that the orientation of the world coordinate system can be arbitrarily chosen. Item 3 is due to Eq. (24), which can be written as $\boldsymbol{T} = (\pm\boldsymbol{A}\boldsymbol{Q})(\pm\boldsymbol{A}\boldsymbol{Q})^\top$ for an arbitrary rotation $\boldsymbol{Q}$, and is inherent of all affine cameras [11], [12].

Item 4 is due to the fact that Eq. (35) involves only the *relative depth* of individual point from the world coordinate origin $\boldsymbol{t}_\kappa$. The absolute depth t_z is determined only if ζ and β are given as *specific functions of* t_z, as in the case of the traditional camera models. If we assume the weak perspective model (Eq. (4)) or the paraperspective model (Eq. (5)), for example, t_z is obtained because the parameter[††] f is known. However, our model does not specify their functional forms; we directly determine their values by self-calibration and leave t_z unspecified.

6. Experiments

Figure 4 shows four simulated image sequences of 600×600 pixels perspectively projected with focal length $f = 600$ pixels. Each consists of 11 frames; six decimated frames are shown here. We added Gaussian random

[†]This corresponds to the singular value decomposition $\boldsymbol{W} = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{V}^\top$ of the measurement matrix $\boldsymbol{W}$ in the traditional formulation [10], [17].

[††]The parameter f in Eqs. (4) and (5) should not be identified with the "focal length" f for perspective projection in Eq. (1).

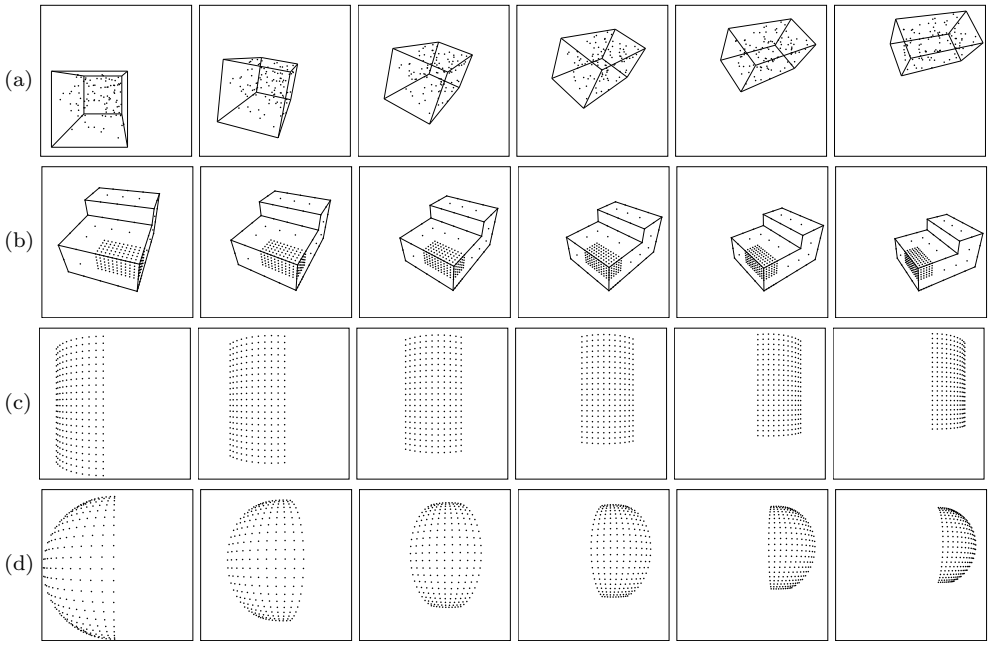
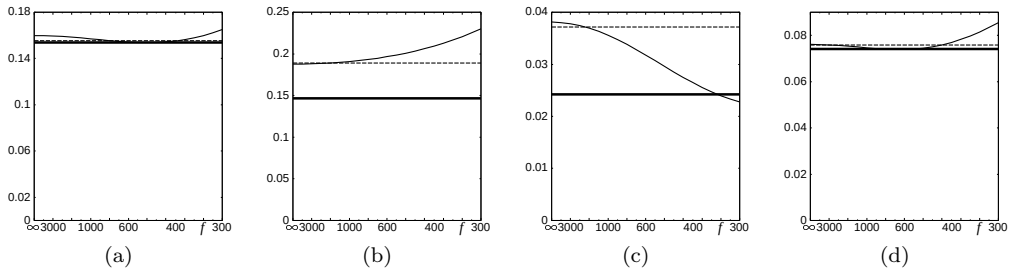


Fig. 4 Simulated image sequences (six decimated frames for each).



(a) (b) (c) (d)

Fig. 5 3-D reconstruction accuracy for the image sequences of Fig. 4(a)∼(d). The horizontal axis is scaled in proportion to $1/f$. Three models are compared: weak perspective (dashed lines), paraperspective (thin solid lines), and our generic model (thick solid lines).

noise of mean 0 and standard deviation 1 pixel independently to the x and y coordinates of the feature points and reconstructed their 3-D shape (the frames in Fig. 4(a), (b) are merely for visual ease).

From the resulting two mirror image shapes, we choose the correct one by comparing the depths of two points that are known be close to the camera. Since the absolute depth and scale are indeterminate, we translate the true and the reconstructed shapes so that their centroids are at the coordinate origin and scaled their sizes so that the root-mean-square distance of the feature points from the origin is 1. Then, we rotate the reconstructed shape so that the root-mean-square distances between the corresponding points of the two shapes is minimized. We adopt the resulting residual as the measure of reconstruction accuracy.

We compared three camera models: the weak perspective, the paraperspective, and our symmetric affine camera models. The orthographic model is omitted, since evidently good results cannot be obtained when the object moves in the depth direction. For using the weak perspective and paraperspective models, we need to specify the parameter f (see Eqs. (4) and (5)). If the size of the reconstructed shape is normalized as described earlier, the choice of f is irrelevant for the weak perspective model, because it only affects the object size as a whole. However, the paraperspective model depends on the value of f we use.

Figure 5 plots the reconstruction accuracy vs. the value f we used; the horizontal axis is scaled in proportion to $1/f$. The dashed line is for weak perspective, the thin solid line is for paraperspective, and the thick solid line is for our model. We observe that the paraperspective model does not necessarily give the highest accuracy when f coincides with the focal length (600 pixels) of the perspective images. The error is indeed minimum around $f = 600$ for Fig. 5(a), (d), but the error decreases as f increases for Fig. 5(b) and as f decreases for Fig. 5(c).

We conclude that our model achieves the accuracy comparable to paraperspective projection given an appropriate value of f, which is unknown in advance. This means that our model automatically chooses appropriate parameter values without any knowledge about f.

However, the difference is very small as seen from Fig. 5.

We conducted real video experiments, using the KLT[†] for tracking feature points. We observed that our method can reconstruct 3-D shapes very similar to those obtained by the traditional models; the differences are difficult to see visually. The computation time is almost the same whichever model is used.

It is to be noted that sometimes *degeneracy* occurs; the matrix $\boldsymbol{A}$ becomes rank deficient so that the resulting vectors $\{\boldsymbol{m}_i\}$ are linearly dependent (see Eq. (15)). As a result, the reconstructed shape is "flat" (see Eq. (13)). This occurs when the smallest eigenvalue of $\boldsymbol{T}$ computed by least squares is negative. In such a case, we replaced the negative eigenvalue by zero, resulting in degeneracy. This type of degeneracy occurs whichever affine model we use.

In principle, we could avoid degeneracy by parameterizing $\boldsymbol{T}$ so that it is guaranteed to be positive definite [11]. However, this would require nonlinear optimization, and the merit of the factorization approach (i.e., linear computation only) would be lost. Moreover, if we look at the images that cause degeneracy, they really look as if a planar object is moving. Since the information is insufficient in the first place, any methods may not be able to solve such degeneracy.

7. Conclusions

We showed that minimal requirements for an affine camera to mimic perspective projection leads to a unique camera model, which we call "symmetric affine camera", having two free functions, whose specific choices would result in the traditional camera models. We regarded them as time varying parameters and determined their values by self-calibration, using linear computation alone, so that an appropriate model is automatically selected. We demonstrated by simulation that the reconstruction accuracy is comparable to the paraperspective model given an appropriate value of the parameter f.

Overall, however, the difference is very small. In practical applications, the weak perspective model, for which the value of f does not affect the result, is sufficient; for higher accuracy, we need to do iterative nonlinear computations based on perspective projection [3].

Acknowledgments

The authors thank Seitaro Asahara of Recruit Staffing, Co., Ltd., Japan, for participating in this project. 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).

References

[1] R. Basri, "Paraperspective ≡ affine," Int. J. Comput. Vis., vol.19, no.2, pp.169–179, Aug. 1996.

[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., vol.E79-D, no.9, pp.1329–1336, Sept. 1996.

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

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

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

[6] K. Kanatani, "Motion segmentation by subspace separation: Model selection and reliability evaluation," Int. J. Image Graphics, vol.2, no.2, pp.179–197, April 2002.

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

[8] K. Kanatani and Y. Sugaya, "Factorization without factorization: complete recipe,"' Mem. Fac. Eng. Okayama Univ., vol.38, nos.1/2, pp.61-72, March 2004.

[9] K. Kanatani, Y. Sugaya and H. Ackermann, "Uncalibrated factorization using a variable symmetric affine camera," Mem. Fac. Eng. Okayama Univ., vol.40, pp.53–63, Jan. 2006.

[10] C.J. Poelman and T. Kanade, "A paraperspective factorization method for shape and motion recovery," IEEE Trans. Pattern Anal. Mach. Intell., vol.19, no.3, pp.206–218, March 1997.

[11] L. Quan, "Self-calibration of an affine camera from multiple views," Int. J. Comput. Vis., vol.19, no.1, pp.93–105, July 1996.

[12] L.S. Shapiro, A. Zisserman, and M. Brady, "3D motion recovery via affine epipolar geometry," Int. J. Comput. Vis., vol.16, no.2, pp.147–182, Oct. 1995.

[13] Y. Sugaya and K. Kanatani, "Outlier removal for motion tracking by subspace separation," IEICE Trans. Inf. & Syst., vol.E86-D, no.6, pp.1095–1102, June 2003.

[14] Y. Sugaya and K. Kanatani, "Extending interrupted feature point tracking for 3-D affine reconstruction," IEICE Trans. Inf. & Syst., vol.E87-D, no.4, pp.1031–1038, April 2004.

[15] Y. Sugaya and K. Kanatani, "Multi-stage optimization for multi-body motion segmentation," IEICE Trans. Inf. & Syst., vol.E87-D, no.7, pp. 1935–1942, July 2004.

[16] A. Sugimoto, "Object recognition by combining paraperspective images," Int. J. Comput. Vis., vol.19, no.2, pp.181–201, Aug. 1996.

[17] C. Tomasi and T. Kanade, "Shape and motion from image streams under orthography—A factorization method," Int. J. Comput. Vis., vol.9, no.2, pp.137–154, Oct. 1992.

[†]http://vision.stanford.edu/~birch/klt/

Kenichi Kanatani received his B.E., M.S., and Ph.D. in applied mathematics from the University of Tokyo in 1972, 1974 and 1979, respectively. After serving as Professor of computer science at Gunma University, Gunma, Japan, he is currently Professor of computer science at Okayama University, Okayama, Japan. He is the author of many books on computer vision and received many awards including the best paper awards from IPSJ (1987) and IEICE (2005). He is an IEEE Fellow.

Yasuyuki Sugaya received his B.E., M.S., and Ph.D. in computer science from the University of Tsukuba, Ibaraki, Japan, in 1996, 1998, and 2001, respectively. From 2001 to 2006, he was Assistant Professor of computer science at Okayama University, Okayama, Japan. Currently, he is Lecturer of information and computer sciences at Toyohashi University of Technology, Toyohashi, Aichi, Japan. His research interests include image processing and computer vision. He received the IEICE best paper award in 2005.

Hanno Ackermann finished the undergraduate and graduate courses of computer engineering at the University of Mannheim, Mannheim, Germany, in 2000 and 2004, respectively, and obtained his Diplom (M.S.) in 2004. He is currently a Ph.D. candidate at the Department of Computer Science, Okayama University, Okayama, Japan. His research interests include image processing and computer vision.