TR－H－072　　0033

Object Recognition by Combining Paraperspective Images

Akihiro SUGIMOTO

1994. 3. 31

ATR人間情報通信研究所
〒619-02 京都府相楽郡精華町光台2-2　☎07749-5-1011

ATR Human Information Processing Research Laboratories
2-2, Hikaridai, Seika-cho, Soraku-gun, Kyoto 619-02 Japan
Telephone: +81-7749-5-1011
Facsimile: +81-7749-5-1008

Object Recognition by Combining Paraperspective Images*

Akihiro SUGIMOTO

ATR Human Information Processing Research Laboratories,
Soraku-gun, Kyoto 619-02, Japan
e-mail: sugimoto@hip.atr.co.jp

Abstract

This paper provides a study on object recognition under paraperspective projection. Discussed is the problem of determining whether or not a given image was obtained from a 3-D object to be recognized. First it is clarified that paraperspective projection is the first-order approximation of perspective projection. Then it is shown that, if we represent an object as a set of its feature points, any paraperspective image can be expressed as a linear combination of three appropriate paraperspective images. We show that any paraperspective image of an object enjoys this property even if it undergoes not only a rigid transformation but also an affine transformation. Particularly in the case of a rigid transformation, the coefficients of the combination have to satisfy two conditions: orthogonality and norm equality. A simple algorithm to solve the above problem based on these properties is presented: a linear, single-shot algorithm. Some experimental results with artificial images are also given; it is found that the algorithm correctly solves the problem for perspective projection as well as for paraperspective projection. Our investigation shows that there exists a simple linear algorithm for recognizing a 3-D object. Namely, we only have to store three images and, whenever a new image is given, we simply determine whether it can be expressed as a combination of the three images.

Key Words: paraperspective projection, linear combination, representation of transformations, 3-D object recognition.

*This work was done while the author was with ATR Auditory and Visual Perception Research Laboratories.

Contents

1 Introduction

Human beings can easily recognize objects in 3-D through visual 2-D information. The appearance of an object shape varies significantly as the viewpoint changes. This results in numerous different images even for the same object. Fundamental difficulty in recognizing objects from images is how to deal with the images that were obtained from the same object. Hence it is a very important problem in object recognition to determine whether or not a given new image was obtained from the same object. A classical approach [3] to this problem is to construct object descriptors that are unaffected by a change in viewpoint. For example, we first recover the 3-D information of an object, and then describe the object with the object-centered coordinates. Though use of generalized cylinders [5] has been proposed with a view to constructing a viewpoint invariant descriptor, it is not an easy task to construct such a descriptor from images. Methods [4], [6], [17] that use geometric invariants have also been proposed. Geometric invariants are viewpoint invariant functions in terms of the coordinates of point images or the coefficients of equations that represent line (or curve) images. Attaching the values of geometric invariants to objects makes it easy to identify one object out of many. Several invariants [11], [12], [13] have been actively derived; however, we must have strong prior knowledge about objects to calculate the invariants. Accordingly, we can not make good use of geometric invariants.

On the other hand, Poggio–Edelman [9] proposed an approach where orthographically projected images, which, of course, depend on the viewpoint, are directly treated for object recognition. The approach does not explicitly recover 3-D information of an object. GRBF (Generalized Radial Basis Function) is used there for learning and also recognizing an object. The network obtains the coordinates of point images of an object to be recognized as its inputs and after its learning process, it establishes some template images of the object within its hidden units. By interpolating or extrapolating the template images, it deals with an image from a different viewpoint. However, we need many images to make the network learn an object; consequently we must get a number of images of an object in advance.

In contrast to this, Ullman–Basri [8], [16] showed that three images are sufficient to describe any other image of the same object under orthographical projection and that any image can be described as a linear combination of the three images. Their results led to an approach that recognizes the object if the new image can be expressed as a linear combination of the three stored images. Sugimoto–Murota [14] extended their results to the case of perspective projection, showing that four images are sufficient to recognize an object under perspective projection, and that an image can be described as a certain nonlinear combination of the four images. However, it is not an easy computational task to determine whether or not an image can be described as a nonlinear combination of the stored images; we will face a convergence problem and a local minimum trap.

Orthographical projection is convenient, being a very rough approximation of the projection of light on the retina. On the other hand, perspective projection, which is the true model of the projection on the retina, often leads to complicated equations for many problems and makes the subsequent analysis difficult. As a compromise, Ohta–Maenobu–Sakai [7] proposed a new model, termed paraperspective projection by Aloimonos [1], [2] to approximate the distortion of a texel pattern under perspective projection. Paraperspective projection stands in complexity between the orthographical and the perspective. It is a good approximation of perspective projection when the size of an object is sufficiently small, compared with the distance between the object and the viewpoint.

This paper is a study on the problem above under paraperspective projection, namely, the problem of determining whether or not a given paraperspective image was obtained from a 3-D object to be recognized. The mathematical meaning of paraperspective projection is clarified: paraperspective projection is the first-order approximation of perspective projection. Under the paraperspective projection, when we represent an object as a set of its feature points, the coordinates of the feature points of any image can be expressed as a certain combination of the coordinates of the feature points of several images of the same object, just as in the case of orthographical projection. Three images are found to be sufficient, though the number of images required for such descriptions depends upon the representation of admissible transformations. The problem under paraperspective projection is thus reduced to the problem of determining whether or not the image is described as a combination of the three stored images. Therefore, the approximation of perspective projection by paraperspective projection makes the problem solvable with a computationally simple procedure: a single-shot algorithm.

The outline of this paper is as follows. In Section 2 we introduce paraperspective projection and show that it is the first-order approximation of perspective projection. In Sections 3 and 4, we formulate the problem to solve, and give a mathematical description to paraperspective images. In Section 5 we consider three representations of admissible transformations. Here the motion for an object is assumed to be described by an affine transformation or by a rigid transformation. And we show that, in either case, any image of the same object can be described as a certain combination of several images. First, we discuss a simple representation, linear combination (in the ordinary sense) representation of admissible transformations, and then exploit the other representations to reduce the number of images required for the description of other images. We also show that, when rigid transformations are admissible, we have two conditions, orthogonality and norm equality, on the coefficients that appear in the combinations. In Section 6 we present an algorithm for recognition under paraperspective projection and show some experimental results with artificial images in Section 7.

2 Paraperspective projection

2.1 Definition of paraperspective projection

The notion of paraperspective projection was introduced by Y. Ohta, K. Maenobu and T. Sakai (see [7]) and named by J. Aloimonos (see [1], [2]). It globally preserves the properties of perspective projection and locally realizes orthographical projection. Suppose that the center of a lens whose focal length is f coincides with the origin and that the z axis is aligned with the optical axis. Let[1] $\boldsymbol{x}^{\mathrm{G}} = (x^{\mathrm{G}}, y^{\mathrm{G}}, z^{\mathrm{G}})^{\mathrm{T}}$ be the coordinates of a reference point[2] G under paraperspective projection. Then a point p (with coordinates $\boldsymbol{x}^{p}$) in 3-D space is paraperspectively projected to $\tilde{\boldsymbol{\pi}}^{p}$ in the image plane ($z = f$) as follows (see Fig. 1):

1. $\boldsymbol{x}^{p}$ is first projected to[3] $\tilde{\boldsymbol{x}}^{p}\,(\in \mathbf{R}^3)$ on the plane $z = z^{\mathrm{G}}$, which is parallel to the image plane. The projection is performed by using a ray that is parallel to ray OG going through the origin O and the reference point G.

2. $\tilde{\boldsymbol{x}}^{p}$ is then projected perspectively to $\tilde{\boldsymbol{\pi}}^{p}$ in the image plane ($z = f$), where $\tilde{\boldsymbol{\pi}}^{p} \in \mathbf{R}^2$.

[1] We use a column vector and denote by $\boldsymbol{x}^{\mathrm{T}}$ the transposition of a vector $\boldsymbol{x}$.

[2] We take it that the centroid of the feature points is a reference point (see (2.2)).

[3] $\mathbf{R}$ represents the set of real numbers.

For $\boldsymbol{x}^p = (x^p, y^p, z^p)^{\mathrm{T}}$, we get

$$\begin{aligned}
\tilde{\boldsymbol{\pi}}^p &= (\tilde{\pi}_1^p,\ \tilde{\pi}_2^p)^{\mathrm{T}} \\
&= \frac{f}{z^{\mathrm{G}}} \begin{pmatrix} 1 & 0 & -\frac{x^{\mathrm{G}}}{z^{\mathrm{G}}} \\ 0 & 1 & -\frac{y^{\mathrm{G}}}{z^{\mathrm{G}}} \end{pmatrix} \boldsymbol{x}^p + \frac{f}{z^{\mathrm{G}}} \begin{pmatrix} x^{\mathrm{G}} \\ y^{\mathrm{G}} \end{pmatrix}.
\end{aligned} \tag{2.1}$$

Since $\boldsymbol{x}^{\mathrm{G}}$ is the centroid of the feature points, we have

$$\boldsymbol{x}^{\mathrm{G}} = \frac{1}{P} \sum_{p=1}^{P} \boldsymbol{x}^p. \tag{2.2}$$

It is clear that paraperspective projection decomposes the image distortions into two parts: Step 1 captures the foreshortening distortion and part of the position effect, and Step 2 captures both the distance and the position effect. For points in the plane $z = z^{\mathrm{G}}$, paraperspective projection coincides with perspective projection.

Remark 2.1 When we let $f \to \infty$ in (2.1), $\tilde{\boldsymbol{\pi}}^p$ does not tend to the orthographical image of p. Instead, we should first shift the coordinate system by $-f$ along the z axis and then take the limit $f \to \infty$. We choose in this paper a coordinate system in which the viewpoint and the origin coincide so that a rotation around the viewpoint can be expressed as a 3×3 orthogonal matrix. □

Remark 2.2 Under perspective projection, $\boldsymbol{x}^p = (x^p, y^p, z^p)^{\mathrm{T}}$ is projected to $\boldsymbol{\pi}^p = (\pi_1^p,\ \pi_2^p)^{\mathrm{T}}$ as follows:

$$\pi_1^p = \frac{x^p}{z^p} f, \qquad \pi_2^p = \frac{y^p}{z^p} f. \tag{2.3}$$

The coordinates of a point are not linearly related to the coordinates of its perspective image, whereas they are linear for its paraperspective image (see (2.1)). □

2.2 Meaning of paraperspective projection

Paraperspective projection is defined by the procedure explained above. Here we show that paraperspective projection is the first-order approximation of perspective projection. In accordance with the notation introduced above, suppose a point p, with coordinates $\boldsymbol{x}^p$, is paraperspectively projected to $\tilde{\boldsymbol{\pi}}^p$ and perspectively projected to $\boldsymbol{\pi}^p$. The coordinates of the reference point under paraperspective projection are denoted as $\boldsymbol{x}^{\mathrm{G}}$. Let $\delta\boldsymbol{x}^p = (\delta x^p, \delta y^p, \delta z^p)^{\mathrm{T}}$ be defined by

$$\begin{pmatrix} \delta x^p \\ \delta y^p \\ \delta z^p \end{pmatrix} := \begin{pmatrix} x^p - x^{\mathrm{G}} \\ y^p - y^{\mathrm{G}} \\ z^p - z^{\mathrm{G}} \end{pmatrix}.$$

From (2.3) we have

$$\pi_1^p = \frac{x^{\mathrm{G}} + \delta x^p}{z^{\mathrm{G}} + \delta z^p} f, \qquad \pi_2^p = \frac{y^{\mathrm{G}} + \delta y^p}{z^{\mathrm{G}} + \delta z^p} f. \tag{2.4}$$

We assume

$$\left|x^{\mathrm{G}}\right| \gg |\delta x^p|, \; \left|y^{\mathrm{G}}\right| \gg |\delta y^p|, \; \left|z^{\mathrm{G}}\right| \gg |\delta z^p| \tag{2.5}$$

and take up to the first-order terms in the Taylor expansion of (2.4) around $\boldsymbol{x}^{\mathrm{G}}$, then we get

$$\begin{aligned} \pi_1^p &= f\left[\frac{x^{\mathrm{G}}}{z^{\mathrm{G}}} + \frac{1}{z^{\mathrm{G}}}\delta x^p - \frac{x^{\mathrm{G}}}{(z^{\mathrm{G}})^2}\delta z^p + \cdots\right], \\ \pi_2^p &= f\left[\frac{y^{\mathrm{G}}}{z^{\mathrm{G}}} + \frac{1}{z^{\mathrm{G}}}\delta y^p - \frac{y^{\mathrm{G}}}{(z^{\mathrm{G}})^2}\delta z^p + \cdots\right]. \end{aligned}$$

On the other hand, from (2.1) we obtain

$$\begin{aligned} \tilde{\pi}_1^p &= f\left[\frac{x^{\mathrm{G}} + \delta x^p}{z^{\mathrm{G}}} - \frac{x^{\mathrm{G}}}{(z^{\mathrm{G}})^2}\delta z^p\right], \\ \tilde{\pi}_2^p &= f\left[\frac{y^{\mathrm{G}} + \delta y^p}{z^{\mathrm{G}}} - \frac{y^{\mathrm{G}}}{(z^{\mathrm{G}})^2}\delta z^p\right]. \end{aligned}$$

It is clear that $\tilde{\boldsymbol{\pi}}^p$ is the first-order approximation of $\boldsymbol{\pi}^p$. Therefore, when the distance between the object and the viewpoint is sufficiently large, compared with the size of the object (see (2.5)), paraperspective projection will be a good approximation of perspective projection.

Remark 2.3 Sugimoto–Murota [15] and Poelman–Kanade [10], independently, clarified the property that paraperspective projection is the first-order approximation of perspective projection. As we can see, we do not use (2.2) in this subsection. Accordingly, we can take any point G that satisfies (2.5) as a reference point under paraperspective projection in order to show this property. In contrast to this, Poelman–Kanade [10] assumes that the centroid of the feature points is the reference point throughout the paper. □

3 Formulation of the problem

In this section we formulate our problem in a well-defined form. The following are assumed:

- An object moves around a fixed viewpoint, and a motion is described by an affine transformation or by a rigid transformation.
- Any image is paraperspectively obtained.
- Feature points in an image are correctly extracted.
- The set of points of which the object consists has a one-to-one correspondence to the set of the feature points in the images, and the correspondence remains invariant under any transformation of the object.
- The set of the feature points in the images is fixed, and the correspondence of the feature points among the images is known.

Now suppose that a point p (with coordinates $\boldsymbol{x}^p$) moves to $\boldsymbol{x}_i^p$ with a transformation i and that it is paraperspectively projected to $\tilde{\pi}_i^p$ (see Fig. 2). When p is subject to an affine transformation, the transformation i is characterized as follows:

$$\boldsymbol{x}_i^p = R_i \boldsymbol{x}^p + \boldsymbol{t}_i,$$

where[4]

$$R_i \in \mathrm{GL}(3), \quad \boldsymbol{t}_i \in \mathbf{R}^3. \tag{3.1}$$

When p is subject to a rigid transformation, (3.1) is to be replaced by

$$R_i \in \mathrm{SO}(3), \quad \boldsymbol{t}_i \in \mathbf{R}^3.$$

It is clear that

$$\boldsymbol{x}_i^{\mathrm{G}} = R_i \boldsymbol{x}^{\mathrm{G}} + \boldsymbol{t}_i$$

[4]GL(3) denotes the general linear group of degree 3 over $\mathbf{R}$, and SO(3) denotes the special orthogonal group of degree 3 over $\mathbf{R}$.

follows from (2.2). Put

$$U_i := f \begin{pmatrix} 1/z_i^{\mathrm{G}} & 0 & -x_i^{\mathrm{G}}/(z_i^{\mathrm{G}})^2 \\ 0 & 1/z_i^{\mathrm{G}} & -y_i^{\mathrm{G}}/(z_i^{\mathrm{G}})^2 \end{pmatrix},$$

$$V_i := f \begin{pmatrix} x_i^{\mathrm{G}}/z_i^{\mathrm{G}} \\ y_i^{\mathrm{G}}/z_i^{\mathrm{G}} \end{pmatrix},$$

then we obtain

$$\tilde{\boldsymbol{\pi}}_i^p = U_i \boldsymbol{x}_i^p + V_i.$$

We represent an object as a set of its feature points. An image of an object is accordingly represented as a set of the projected feature points in the image plane. Let $\{\tilde{\boldsymbol{\pi}}_i^p\}_{p=1}^P$ denote the image of an object to be recognized with a transformation $i\,(i \in \{1, 2, \ldots, I\})$ where the number of the feature points is assumed to be P; hence we have I images of the object. We denote a new image by $\{\tilde{\boldsymbol{\pi}}_*^p\}_{p=1}^P$. The problem we investigate here is to determine whether or not the image $\{\tilde{\boldsymbol{\pi}}_*^p\}_{p=1}^P$ is obtained from the same object with a certain transformation. We assume that a class of admissible transformations of a 3-D object is specified. This is because the decision whether or not the new image was obtained from the same object depends upon the class of admissible transformations. In this paper, we consider two classes of admissible transformations: affine transformations $\mathcal{A}_{\mathrm{a}}$ and rigid transformations $\mathcal{A}_{\mathrm{r}}$. Both form a group and are expressed, respectively, as follows:

$$\begin{aligned} \mathcal{A}_{\mathrm{a}} &= \{(R,\ \boldsymbol{t}) \mid R \in \mathrm{GL}(3),\ \boldsymbol{t} \in \mathbf{R}^3\}, \\ \mathcal{A}_{\mathrm{r}} &= \{(R,\ \boldsymbol{t}) \mid R \in \mathrm{SO}\,(3)\,,\ \boldsymbol{t} \in \mathbf{R}^3\}. \end{aligned} \tag{3.2}$$

Since $\mathrm{SO}(3) \subset \mathrm{GL}(3)$, elements of $\mathcal{A}_{\mathrm{r}}$ are characterized as those elements of $\mathcal{A}_{\mathrm{a}}$ that satisfy certain conditions. Therefore, (3.2) can be rewritten as

$$\mathcal{A}_{\mathrm{r}} = \{(R,\ \boldsymbol{t}) \in \mathcal{A}_{\mathrm{a}} \mid R \in \mathrm{SO}(3)\}.$$

In this paper, we regard $\mathcal{A}_{\mathrm{r}}$ as part of $\mathcal{A}_{\mathrm{a}}$ with the conditions. We write $i \in \mathcal{A}$ as a shorthand notation for $(R_i,\ \boldsymbol{t}_i) \in \mathcal{A}$. For a class of admissible transformations $\mathcal{A}$, put

$$\tilde{\Pi}^p := \{\tilde{\boldsymbol{\pi}}_i^p \mid \tilde{\boldsymbol{\pi}}_i^p = U_i \boldsymbol{x}_i^p + V_i,\ \exists i \in \mathcal{A}\}.$$

$\tilde{\Pi}^p$ is the set of possible paraperspective images of point p for a class of admissible transformations $\mathcal{A}$. The problem is formulated as follows.

Problem 3.1 Suppose a class $\mathcal{A}$ of admissible transformations is specified. Find a procedure which, treating directly $\{\tilde{\pi}_i^p\}_{p=1}^P$ ($i \in \{1, 2, \ldots, I\}$), determines whether or not $\tilde{\pi}_*^p \in \tilde{\Pi}^p$ for all $p \in \{1, 2, \ldots, P\}$ every time $\{\tilde{\pi}_*^p\}_{p=1}^P$ is given. □

We assume a representation of admissible transformations for further investigation, because a procedure to be constructed depends on the representation of admissible transformations.

In Section 5, we consider three representations, all of which are linear equations in the elements of admissible transformations.

4 Mathematical description of images

4.1 Coordinates in the image plane

Since

$$\tilde{\pi}_i^{\mathrm{G}} = \frac{1}{P} \sum_{p=1}^{P} \tilde{\pi}_i^p \tag{4.1}$$

follows from (2.1) and (2.2), we can calculate $\tilde{\pi}_i^{\mathrm{G}}$ easily from $\tilde{\pi}_i^p$ ($p = 1, 2, \ldots, P$). We denote the increment of $\tilde{\pi}_i^p$ from $\tilde{\pi}_i^{\mathrm{G}}$ by

$$\rho_i^p := \tilde{\pi}_i^p - \tilde{\pi}_i^{\mathrm{G}}, \tag{4.2}$$

and as a counterpart of $\tilde{\Pi}^p$ we put

$$P^p := \left\{ \rho_i^p \mid \rho_i^p = \tilde{\pi}_i^p - \tilde{\pi}_i^{\mathrm{G}},\ \tilde{\pi}_i^p \in \tilde{\Pi}^p \right\}.$$

Since ρ_i^p is easily calculated from $\tilde{\pi}_i^p$, we may concentrate on ρ_i^p instead of $\tilde{\pi}_i^p$. In other words, we may regard $\{\rho_i^p\}_{p=1}^P$ ($i = 1, 2, \ldots, I$) and $\{\rho_*^p\}_{p=1}^P$ as the stored image and a new image respectively.

From (4.2) we have

$$\rho_i^p = \frac{f}{(z_i^{\mathrm{G}})^2} Q\,[\boldsymbol{x}_i^{\mathrm{G}}]\, \delta \boldsymbol{x}_i^p, \tag{4.3}$$

where

$$Q := \begin{pmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \end{pmatrix}$$

and for $\boldsymbol{x} = (x_1, x_2, x_3)^{\mathrm{T}}$ in general, $[\,\boldsymbol{x}\,]$ is defined by

$$[\,\boldsymbol{x}\,] := \begin{pmatrix} 0 & -x_3 & x_2 \\ x_3 & 0 & -x_1 \\ -x_2 & x_1 & 0 \end{pmatrix}. \tag{4.4}$$

The operation $[\,\boldsymbol{x}\,]$ has the following properties.

Lemma 4.1 Let[5] $\boldsymbol{x}$, $\boldsymbol{y} \in \mathrm{R}^3$ and $R \in \mathrm{GL}(3)$.

$$[\,\boldsymbol{x}+\boldsymbol{y}\,] = [\,\boldsymbol{x}\,]+[\,\boldsymbol{y}\,], \tag{4.5}$$

$$R^{\mathrm{T}}\,[\,R\boldsymbol{x}\,]\,R = \det R \cdot [\,\boldsymbol{x}\,]. \tag{4.6}$$

Proof: (4.5) is trivial from the definition of $[\,\cdot\,]$ (see (4.4)). Putting

$$R^{\mathrm{T}} = \left(\begin{array}{c|c|c} \boldsymbol{r}_1 & \boldsymbol{r}_2 & \boldsymbol{r}_3 \end{array}\right),$$

we have

$$\det R \cdot R^{-1} = \left(\begin{array}{c|c|c} [\,\boldsymbol{r}_2\,]\,\boldsymbol{r}_3 & [\,\boldsymbol{r}_3\,]\,\boldsymbol{r}_1 & [\,\boldsymbol{r}_1\,]\,\boldsymbol{r}_2 \end{array}\right).$$

It is easy to see[6]

$$\begin{aligned} (\det R)^2 \cdot \left(R^{-\mathrm{T}}\,[\,\boldsymbol{x}\,]\,R^{-1}\right)_{12} &= ([\,\boldsymbol{r}_2\,]\,\boldsymbol{r}_3)^{\mathrm{T}}\,[\,\boldsymbol{x}\,]\,[\,\boldsymbol{r}_3\,]\,\boldsymbol{r}_1 \\ &= \boldsymbol{r}_3^{\mathrm{T}}\,[\,\boldsymbol{r}_2\,]\,[\,[\,\boldsymbol{r}_3\,]\,\boldsymbol{r}_1]\,\boldsymbol{x} \\ &= \boldsymbol{r}_3^{\mathrm{T}}\,[\,\boldsymbol{r}_2\,]\,\{\,(\boldsymbol{r}_3\cdot\boldsymbol{x})\,\boldsymbol{r}_1 - (\boldsymbol{r}_1\cdot\boldsymbol{x})\,\boldsymbol{r}_3\,\} \\ &= \boldsymbol{r}_3^{\mathrm{T}}\,[\,\boldsymbol{r}_2\,]\,\boldsymbol{r}_1\,\boldsymbol{r}_3^{\mathrm{T}}\,\boldsymbol{x} \\ &= \det R \cdot [\,R\boldsymbol{x}\,]_{12}. \end{aligned} \tag{4.7}$$

---

[5] $\det R$ is the determinant of a square matrix R.

[6] $R^{-\mathrm{T}}$ stands for $(R^{\mathrm{T}})^{-1}$ or equivalently $(R^{-1})^{\mathrm{T}}$. M_{ij} is the (i,j) component of a matrix M.

In the same way, we obtain

$$(\det R)^2 \cdot \left(R^{-\mathrm{T}}\,[\,\boldsymbol{x}\,]\,R^{-1}\right)_{13} \;=\; \det R \cdot [\,R\boldsymbol{x}\,]_{13}, \tag{4.8}$$

$$(\det R)^2 \cdot \left(R^{-\mathrm{T}}\,[\,\boldsymbol{x}\,]\,R^{-1}\right)_{23} \;=\; \det R \cdot [\,R\boldsymbol{x}\,]_{23}. \tag{4.9}$$

Since both $R^{-\mathrm{T}}\,[\,\boldsymbol{x}\,]\,R^{-1}$ and $[\,R\boldsymbol{x}\,]$ are alternating, it follows

$$\det R \cdot R^{-\mathrm{T}}\,[\,\boldsymbol{x}\,]\,R^{-1} \;=\; [\,R\boldsymbol{x}\,]$$

from (4.7), (4.8) and (4.9), which yields (4.6). □

4.2 Images

Assume that, for $i \in \{1, 2, \ldots, I\}$, the stored image $\{\rho_i^p\}_{p=1}^P$ was obtained from $\{\boldsymbol{x}_i^p\}_{p=1}^P$ that satisfies

$$\boldsymbol{x}_i^p = R_i\boldsymbol{x}^p + \boldsymbol{t}_i \quad (p \in \{1, 2, \ldots, P\}).$$

This implies

$$\begin{aligned}
\delta\boldsymbol{x}_i^p &= R_i\delta\boldsymbol{x}^p, \\
\boldsymbol{x}_i^{\mathrm{G}} &= R_i\boldsymbol{x}^{\mathrm{G}} + \boldsymbol{t}_i,
\end{aligned}$$

from which it follows

$$[\,\boldsymbol{x}_i^{\mathrm{G}}\,]\,\delta\boldsymbol{x}_i^p \;=\; (\det R_i)\,R_i^{-\mathrm{T}}\,[\,\boldsymbol{x}^{\mathrm{G}} + R_i^{-1}\boldsymbol{t}_i\,]\,\delta\boldsymbol{x}^p$$

by Lemma 4.1. Substituting this into (4.3) we obtain

$$\rho_i^p \;=\; f\left(A_i\,[\,\boldsymbol{x}^{\mathrm{G}}\,] + B_i\right)\delta\boldsymbol{x}^p, \tag{4.10}$$

where

$$\begin{aligned}
A_i &:= \frac{\det R_i}{(z_i^{\mathrm{G}})^2}\,Q\,R_i^{-\mathrm{T}}, \\
B_i &:= \frac{\det R_i}{(z_i^{\mathrm{G}})^2}\,Q\,R_i^{-\mathrm{T}}\,[\,R_i^{-1}\boldsymbol{t}_i\,].
\end{aligned}$$

Here A_i and B_i are both 2×3 matrices. The stored images are expressed as (4.10).

Remark 4.1 In the case of $R_i \in \mathrm{GL}(3)$, both A_i and B_i can be any 2×3 matrix, whereas in the case of $R_i \in \mathrm{SO}(3)$, the conditions

$$\det R_i = 1, \qquad R_i^{-\mathrm{T}} = R_i$$

are satisfied. Hence the first-row vector of A_i is equal to the second-row vector of R_i multiplied by a constant c, and the second-row vector of A_i is equal to the first-row vector of R_i multiplied by $-c$. In other words, the two row vectors of A_i are orthogonal and have the same norm. We call these orthogonality and norm equality. □

As for a new image $\{\rho_*^p\}_{p=1}^P$, suppose similarly that

$$\boldsymbol{x}_*^p = R_* \boldsymbol{x}^p + \boldsymbol{t}_* \quad (p \in \{1, 2, \ldots, P\})$$

is satisfied. Then by putting

$$\begin{aligned} A_* &:= \frac{\det R_*}{(z_*^{\mathrm{G}})^2}\, Q\, R_*^{-\mathrm{T}}, \\ B_* &:= \frac{\det R_*}{(z_*^{\mathrm{G}})^2}\, Q\, R_*^{-\mathrm{T}}\, [\, R_*^{-1} \boldsymbol{t}_* \,], \end{aligned}$$

we obtain

$$\boldsymbol{\rho}_*^p = f\left(A_* [\, \boldsymbol{x}^{\mathrm{G}} \,] + B_*\right) \delta \boldsymbol{x}^p. \tag{4.11}$$

5 Representations of images

Here we consider three representations of admissible transformations and investigate how an image can be described by a combination of the stored images. We define the following 2×6 matrices:

$$C_i := (A_i \mid B_i), \qquad C_* := (A_* \mid B_*). \tag{5.1}$$

5.1 Linear combination I

The matrix C_* in (5.1), being a 2×6 matrix, can also be thought of as a vector in $\mathbf{R}^{12}$. Therefore, if $\{C_i\}_{i=1}^I$ spans $\mathbf{R}^{12}$, any C_* can be expressed as

$$C_* = \sum_{i=1}^{I} \lambda_i C_i$$

in terms of the coefficient set $\{\lambda_i\}_{i=1}^{I}$. This is equivalent to

$$A_* = \sum_{i=1}^{I} \lambda_i A_i, \qquad B_* = \sum_{i=1}^{I} \lambda_i B_i, \tag{5.2}$$

which yields a representation of A_* and B_*. Substituting (5.2) into (4.11) we obtain

$$\begin{aligned} \rho_*^p &= f\left\{\sum_{i=1}^{I} \lambda_i A_i [\boldsymbol{x}^{\mathrm{G}}] + \sum_{i=1}^{I} \lambda_i B_i\right\} \delta \boldsymbol{x}^p \\ &= \sum_{i=1}^{I} \lambda_i f\left(A_i [\boldsymbol{x}^{\mathrm{G}}] + B_i\right) \delta \boldsymbol{x}^p \\ &= \sum_{i=1}^{I} \lambda_i \rho_i^p. \end{aligned} \tag{5.3}$$

Theorem 5.1 Suppose $\mathcal{A}_{\mathrm{a}}$ (affine transformations) is the class of admissible transformations and that $\{C_i\}_{i=1}^{12}$ is linearly independent. Then for $\forall \rho_*^p \in P^p$, there exists $\{\lambda_i\}_{i=1}^{12}$, independent of p, such that

$$\rho_*^p = \sum_{i=1}^{12} \lambda_i \rho_i^p. \tag{5.4}$$

□

When $\mathcal{A}_{\mathrm{r}}$ (rigid transformations) is the class of admissible transformations, the two row vectors of A_* are orthogonal and have the same norm (see Remark 4.1). Putting

$$A_i^{\mathrm{T}} = \left(\begin{array}{c|c} & \\ \boldsymbol{a}_1^i & \boldsymbol{a}_2^i \\ & \end{array} \right), \tag{5.5}$$

we see the conditions[7]

$$\left(\sum_{i=1}^{12} \lambda_i \boldsymbol{a}_1^i\right) \cdot \left(\sum_{i=1}^{12} \lambda_i \boldsymbol{a}_2^i\right) = 0, \tag{5.6}$$

$$\left\| \sum_{i=1}^{12} \lambda_i \boldsymbol{a}_1^i \right\| = \left\| \sum_{i=1}^{12} \lambda_i \boldsymbol{a}_2^i \right\| \tag{5.7}$$

on $\{\lambda_i\}_{i=1}^{12}$ in Theorem 5.1, which are orthogonality and norm equality.

Theorem 5.2 When $\mathcal{A}_{\mathrm{r}}$ is the class of admissible transformations, $\{\lambda_i\}_{i=1}^{12}$ in Theorem 5.1 is subject to (5.6) and (5.7). □

---

[7] $\|\boldsymbol{x}\|$ is the Euclidean norm of a vector $\boldsymbol{x}$.

Remark 5.1 Theorem 5.1 states that all the feature points in the images obtained from the same object should satisfy (5.4) with a common coefficient set $\{\lambda_i\}_{i=1}^{I}$. The converse is not true, namely, an image in which all the feature points satisfy (5.4) is not necessarily obtained from the same object. However, when the number of the feature points is sufficiently large, it will not be expected that all the feature points of an image of a different object happen to satisfy (5.4). This remark applies also to the theorems below. □

5.2 Linear combination II

Put

$$C_i^{\mathrm{T}} = \left(\begin{array}{c|c} & \\ \boldsymbol{c}_1^i & \boldsymbol{c}_2^i \\ & \end{array}\right).$$

If both $\{\boldsymbol{c}_1^i\}_{i=1}^{I}$ and $\{\boldsymbol{c}_2^i\}_{i=1}^{I}$ span $\mathbf{R}^6$ respectively, there exists $D_i := \begin{pmatrix} \mu_i & 0 \\ 0 & \nu_i \end{pmatrix}$ $(i = 1, 2, \ldots, I)$ such that

$$C_* = \sum_{i=1}^{I} D_i \, C_i,$$

which is equivalent to

$$A_* = \sum_{i=1}^{I} D_i \, A_i, \qquad B_* = \sum_{i=1}^{I} D_i \, B_i. \tag{5.8}$$

This gives another representation of A_* and B_*. Then we obtain

$$\begin{aligned} \rho_*^p &= f\left\{\sum_{i=1}^{I} D_i \, A_i [\boldsymbol{x}^{\mathrm{G}}] + \sum_{i=1}^{I} D_i \, B_i\right\} \delta\boldsymbol{x}^p \\ &= \sum_{i=1}^{I} D_i \, \rho_i^p. \end{aligned}$$

Theorem 5.3 Suppose that $\mathcal{A}_{\mathrm{a}}$ is the class of admissible transformations and that $\{\boldsymbol{c}_1^i\}_{i=1}^{6}$ and $\{\boldsymbol{c}_2^i\}_{i=1}^{6}$ are linearly independent, respectively. Then for $\forall \rho_*^p \in P^p$, there exists $\{\mu_i, \nu_i\}_{i=1}^{6}$, independent of p, such that

$$\rho_*^p = \sum_{i=1}^{6} \begin{pmatrix} \mu_i & 0 \\ 0 & \nu_i \end{pmatrix} \rho_i^p.$$

□

Theorem 5.4 When $\mathcal{A}_r$ is the class of admissible transformations, $\{\mu_i, \nu_i\}_{i=1}^6$ in Theorem 5.3 is subject to the following two conditions:

$$\left(\sum_{i=1}^{6} \mu_i\, \boldsymbol{a}_1^i\right) \cdot \left(\sum_{i=1}^{6} \nu_i\, \boldsymbol{a}_2^i\right) = 0,$$

$$\left\| \sum_{i=1}^{6} \mu_i\, \boldsymbol{a}_1^i \right\| = \left\| \sum_{i=1}^{6} \nu_i\, \boldsymbol{a}_2^i \right\|.$$

□

We have shown that any image can be described as a combination of six appropriate images under the representation (5.8).

Remark 5.2 The representation (5.8) is similar to that of Ullman–Basri [16]. □

5.3 Linear combination III

If $\{\boldsymbol{c}_1^i, \boldsymbol{c}_2^i\}_{i=1}^3$ spans $\mathbf{R}^6$, there exists $\{\alpha_i, \beta_i, \gamma_i, \delta_i\}_{i=1}^3$ which satisfies

$$C_* = \sum_{i=1}^{3} \begin{pmatrix} \alpha_i & \beta_i \\ \gamma_i & \delta_i \end{pmatrix} C_i. \tag{5.9}$$

Putting

$$M_i := \begin{pmatrix} \alpha_i & \beta_i \\ \gamma_i & \delta_i \end{pmatrix},$$

we rewrite (5.9) as

$$A_* = \sum_{i=1}^{3} M_i\, A_i, \qquad B_* = \sum_{i=1}^{3} M_i\, B_i, \tag{5.10}$$

which gives a third representation of A_* and B_*. Then,

$$\begin{aligned} \rho_*^p &= f\left\{\sum_{i=1}^{3} (M_i A_i)\, [\,\boldsymbol{x}^{\mathrm{G}}\,] + \sum_{i=1}^{3} (M_i B_i)\right\} \delta \boldsymbol{x}^p \\ &= \sum_{i=1}^{3} M_i\, \rho_i^p. \end{aligned}$$

Theorem 5.5 Suppose that $\mathcal{A}_a$ is the class of admissible transformations and that $\{\boldsymbol{c}_1^i, \boldsymbol{c}_2^i\}_{i=1}^3$ is linearly independent. Then for $\forall \rho_*^p \in P^p$, there exists $\{\alpha_i, \beta_i, \gamma_i, \delta_i\}_{i=1}^3$, independent of p, such that

$$\rho_*^p = \sum_{i=1}^{3} \begin{pmatrix} \alpha_i & \beta_i \\ \gamma_i & \delta_i \end{pmatrix} \rho_i^p. \tag{5.11}$$

□

Theorem 5.6 When $\mathcal{A}_r$ is the class of admissible transformations, $\{\alpha_i, \beta_i, \gamma_i, \delta_i\}_{i=1}^{3}$ in Theorem 5.5 is subject to the following two conditions:

$$\sum_{i=1}^{3} \left(\alpha_i\, \boldsymbol{a}_1^i + \beta_i\, \boldsymbol{a}_2^i\right) \cdot \sum_{i=1}^{3} \left(\gamma_i\, \boldsymbol{a}_1^i + \delta_i\, \boldsymbol{a}_2^i\right) = 0, \tag{5.12}$$

$$\left\| \sum_{i=1}^{3} (\alpha_i\, \boldsymbol{a}_1^i + \beta_i\, \boldsymbol{a}_2^i) \right\| = \left\| \sum_{i=1}^{3} (\gamma_i\, \boldsymbol{a}_1^i + \delta_i\, \boldsymbol{a}_2^i) \right\|. \tag{5.13}$$

□

We have demonstrated that for the two classes of admissible transformations (affine and rigid), any image can be expressed as a linear combination of three appropriate images under the representation (5.10). Furthermore, we need at least six points to determine $\{\alpha_i, \beta_i, \gamma_i, \delta_i\}_{i=1}^{3}$ since we have two independent equations for each point (see (5.11)).

6 Algorithm

In this section we describe an algorithm for object recognition, i.e., a single-shot algorithm of linear least-squares type, which is based on the third representation considered in Section 5. This is because the representation has the advantage of requiring the smallest number of images. Similar algorithms could be made for the other representations.

6.1 Degeneracy of the coefficient matrix

In Subsection 5.3 we proved that $\boldsymbol{\rho}_*^p$ can be expressed as a combination of $\{\boldsymbol{\rho}_i^p\}_{i=1}^{3}$ for all p ($p = 1, 2, \ldots, P$) (see (5.11)). When a new image $\{\tilde{\boldsymbol{\pi}}_*^p\}_{p=1}^{P}$ is given, we first calculate $\{\boldsymbol{\rho}_*^p\}_{p=1}^{P}$ and then regard (5.11) as an overdetermined system of linear equations in $\{\alpha_i, \beta_i, \gamma_i, \delta_i\}_{i=1}^{3}$. (5.11) shows independency of $\{\alpha_i, \beta_i\}_{i=1}^{3}$ and $\{\gamma_i, \delta_i\}_{i=1}^{3}$: the 1st component of $\boldsymbol{\rho}_*^p$ is expressed as the linear equation in only α_i and β_i; whereas the 2nd one is expressed as the linear equation in only γ_i and δ_i. Hence, we can recover $\{\alpha_i, \beta_i\}_{i=1}^{3}$ and $\{\gamma_i, \delta_i\}_{i=1}^{3}$, separately.

Let $\boldsymbol{\rho}_i^p = (\rho_{i1}^p, \rho_{i2}^p)^{\mathrm{T}}$, $\boldsymbol{\rho}_*^p = (\rho_{*1}^p, \rho_{*2}^p)^{\mathrm{T}}$ and also define

$$\begin{aligned} \boldsymbol{\rho}_k &:= (\rho_{*k}^1,\, \rho_{*k}^2, \ldots, \rho_{*k}^P)^{\mathrm{T}} \qquad (k = 1, 2), \qquad (6.1)\\ \boldsymbol{\tau}_1 &:= (\alpha_1,\, \alpha_2,\, \alpha_3,\, \beta_1,\, \beta_2,\, \beta_3)^{\mathrm{T}},\\ \boldsymbol{\tau}_2 &:= (\gamma_1,\, \gamma_2,\, \gamma_3,\, \delta_1,\, \delta_2,\, \delta_3)^{\mathrm{T}}, \end{aligned}$$

$$H \;:=\; \begin{pmatrix} \rho_{11}^1 & \rho_{21}^1 & \rho_{31}^1 & \rho_{12}^1 & \rho_{22}^1 & \rho_{32}^1 \\ \rho_{11}^2 & \rho_{21}^2 & \rho_{31}^2 & \rho_{12}^2 & \rho_{22}^2 & \rho_{32}^2 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \rho_{11}^P & \rho_{21}^P & \rho_{31}^P & \rho_{12}^P & \rho_{22}^P & \rho_{32}^P \end{pmatrix}.$$

Then (5.11) is rewritten as

$$\rho_k \;=\; H\tau_k \qquad (k = 1, 2). \tag{6.2}$$

Therefore, we first store H in advance for an object to be recognized. Then when a new image (ρ_k ($k = 1, 2$)) is given, if affine transformations are concerned, the problem of determining whether or not it was obtained from the same object is equivalent to determining whether or not there exists τ_k that satisfies (6.2); for the case of rigid transformations, the problem is equivalent to determining whether or not there exists τ_k that satisfies all of (6.2), (5.12) and (5.13). We should remark again that $P \geq 6$ must be satisfied to recover τ_k. $P \geq 6$ is assumed in the following argument. The coefficient matrix H of linear equation (6.2) in τ_k has the following property.

Theorem 6.1 For the coefficient matrix H in (6.2), we have

$$\mathrm{rank} H \;\leq\; 3, \tag{6.3}$$

which is generally satisfied with equality.

Proof: For $i = 1, 2, 3$, we denote by ω_1^i and ω_2^i the first-row vector and the second-row vector of the 2×3 matrix $f\,A_i\,[\,\boldsymbol{x}^{\mathrm{G}} + R_i^{-1}\boldsymbol{t}_i\,]$, respectively (see (4.10)). We also define a $3 \times P$ matrix S and a 6×3 matrix T as follows:

$$S := \left(\begin{array}{c|c|c|c} \delta\boldsymbol{x}^1 & \delta\boldsymbol{x}^2 & \cdots & \delta\boldsymbol{x}^P \end{array} \right), \qquad T := \left(\begin{array}{c|c|c|c|c|c} \omega_1^1 & \omega_1^2 & \omega_1^3 & \omega_2^1 & \omega_2^2 & \omega_2^3 \end{array} \right)^{\mathrm{T}}.$$

Then we have

$$H \;=\; (T\,S)^{\mathrm{T}}.$$

Hence

$$\mathrm{rank}H \quad \le \quad \min(\,\mathrm{rank}S,\ \mathrm{rank}T\,) \quad \le \quad 3.$$

Furthermore, in general, point p ($p = 1, 2, \ldots, P$) is randomly chosen in three dimensions and the three transformations for the stored images are also independent of one another. This indicates that $\{\delta \boldsymbol{x}^p\}_{p=1}^{P}$ and $\{\omega_1^i,\, \omega_2^i\}_{i=1}^{3}$ span R^3, respectively. Thus, in general, (6.3) is satisfied with equality. □

From here on, we assume (6.3) is satisfied with equality. Theorem 6.1 shows that a $P \times 6$ matrix H always degenerates. In other words, $\boldsymbol{\tau}_k$ ($k = 1, 2$) that satisfies (6.2) has three degrees of freedom; we can not uniquely recover $\boldsymbol{\tau}_k$. Accordingly, when $\mathcal{A}_{\mathrm{r}}$ (rigid transformations) is the class of admissible transformations, we should make use of this freedom so that $\boldsymbol{\tau}_1$ and $\boldsymbol{\tau}_2$ satisfy the conditions (5.12) and (5.13). We apply the method of singular value decomposition to H to recover $\boldsymbol{\tau}_1$ and $\boldsymbol{\tau}_2$. We can decompose H into

$$H \quad = \quad U\,\Sigma\,V^{\mathrm{T}}, \tag{6.4}$$

where U and V are respectively a $P \times P$ orthogonal matrix and a 6×6 orthogonal matrix; and Σ is a $P \times 6$ matrix such that

$$\begin{aligned} \Sigma \quad &= \quad \begin{pmatrix} D & O_{3,3} \\ O_{P-3,3} & O_{P-3,3} \end{pmatrix}, \\ D \quad &= \quad \mathrm{diag}(\sigma_1,\, \sigma_2,\, \sigma_3), \quad \sigma_1 \ge \sigma_2 \ge \sigma_3 > 0. \end{aligned}$$

Let $\boldsymbol{d}_k = U^{\mathrm{T}} \boldsymbol{\rho}_k$ ($k = 1, 2$), then the solutions of (6.2) are given by[8]

$$\left\{\boldsymbol{\tau}_k = V\,\boldsymbol{y}_k \mid y_{kj} = {}^{d_{kj}}/_{\sigma_j}\ (1 \le j \le 3), \quad y_{kj}\ (4 \le j \le 6) \text{ is arbitrary}\right\}. \tag{6.5}$$

Especially

$$\boldsymbol{\tau}_k \quad = \quad V\,\Sigma^{+}\,U^{\mathrm{T}} \boldsymbol{\rho}_k, \tag{6.6}$$

where

$$\Sigma^{+} \quad = \quad \begin{pmatrix} D^{-1} & O_{3,P-3} \\ O_{3,3} & O_{3,P-3} \end{pmatrix},$$

[8] d_{kj} is the j-th component of vector $\boldsymbol{d}_k$.

which is obtained by setting $y_{kj} = 0$ $(4 \leq j \leq 6)$ in (6.5), is the least-norm solution, i.e., the solution that has the minimum norm among the solutions of (6.2). Note that we need not know the transformations for the stored images to calculate singular values σ_j from H.

6.2 Simplification of orthogonality and norm equality

When rigid transformations are admissible, we have two conditions: orthogonality and norm equality. Here we show that two conditions (5.12) and (5.13) in Theorem 5.6 can be rewritten in a simpler fashion where singular value decomposition again plays an important role. And we give the least-norm solution of (6.2) that satisfies both orthogonality and norm equality.

We define a 3×6 matrix W as follows:

$$W := \left(\begin{array}{c|c|c|c|c|c} \\ a_1^1 & a_1^2 & a_1^3 & a_2^1 & a_2^2 & a_2^3 \\ \\ \end{array} \right),$$

which is derived from the three transformations of an object for the stored images (see (5.5)). By using W, we can rewrite (5.12) and (5.13) as

$$\tau_1^{\mathrm{T}} W^{\mathrm{T}} W \tau_2 = 0, \tag{6.7}$$

$$\tau_1^{\mathrm{T}} W^{\mathrm{T}} W \tau_1 = \tau_2^{\mathrm{T}} W^{\mathrm{T}} W \tau_2. \tag{6.8}$$

The number of singular values of W is three since the three transformations that are related to the stored images are generally independent: $\mathrm{rank} W = 3$. Hence, we apply the method of singular value decomposition to W to obtain

$$W = F \Psi G^{\mathrm{T}}. \tag{6.9}$$

Here F and G are, respectively, a 3×3 orthogonal matrix and a 6×6 orthogonal matrix; and Ψ is given by

$$\Psi = \begin{pmatrix} \psi_1 & 0 & 0 & 0 & 0 & 0 \\ 0 & \psi_2 & 0 & 0 & 0 & 0 \\ 0 & 0 & \psi_3 & 0 & 0 & 0 \end{pmatrix} \quad (\psi_1 \geq \psi_2 \geq \psi_3 > 0).$$

We define

$$\tilde{\tau}_k := G^{\mathrm{T}} \tau_k \quad (k = 1, 2), \tag{6.10}$$

and denote by $\tilde{\tau}_{kj}$ the j-th component of vector $\tilde{\boldsymbol{\tau}}_k$. Then (6.7) and (6.8), namely, the conditions on $\{\alpha_i, \beta_i, \gamma_i, \delta_i\}_{i=1}^{3}$ (or equivalently on $\boldsymbol{\tau}_1$ and $\boldsymbol{\tau}_2$) that are incurred for the case where rigid transformations are admissible, are rewritten as

$$\sum_{j=1}^{3} \psi_j^2\, \tilde{\tau}_{1j}\, \tilde{\tau}_{2j} = 0, \tag{6.11}$$

$$\sum_{j=1}^{3} \psi_j^2 \left(\tilde{\tau}_{1j}^2 - \tilde{\tau}_{2j}^2\right) = 0. \tag{6.12}$$

We should remark that ψ_j $(j = 1, 2, 3)$ can be calculated in advance if we know the transformations for the stored images. In other words, whereas we need not know the transformations to calculate σ_j, we must know them to calculate ψ_j. From (6.10) and (6.5), the solutions of (6.2) are given in terms of $\tilde{\boldsymbol{\tau}}_k$:

$$\left\{\tilde{\boldsymbol{\tau}}_k = G^{\mathrm{T}}\, V\, \boldsymbol{y}_k \mid y_{kj} = {}^{d_{kj}}\!/\!_{\sigma_j}\ (1 \le j \le 3),\quad y_{kj}\ (4 \le j \le 6) \text{ is arbitrary}\right\}. \tag{6.13}$$

Though y_{kj} $(j = 1, 2, 3)$ is already determined, y_{kj} $(j = 4, 5, 6)$ is arbitrary, which shows that $\tilde{\boldsymbol{\tau}}_k$ is expressed as the linear combination in y_{kj} $(j = 4, 5, 6)$. Therefore, (6.11) and (6.12) are quadratic equations in y_{kj} $(j = 4, 5, 6\,;\ k = 1, 2)$. The fact that the number of the constraint conditions is two indicates that we still have freedom in determining y_{kj} $(j = 4, 5, 6\,;\ k = 1, 2)$. Hence, we set $y_{kj} = 0$ $(j = 5, 6\,;\ k = 1, 2)$. Then when we solve (6.11) and (6.12), $\boldsymbol{y}_k$ gives us the least-norm solution of (6.2) that satisfies both (6.11) and (6.12) since $\|\boldsymbol{y}_k\| = \|\tilde{\boldsymbol{\tau}}_k\| = \|\boldsymbol{\tau}_k\|$ (see (6.5)).

6.3 Algorithm

In Subsection 6.1 we showed that the problem of determining whether a given new image was obtained from the same object is equivalent to determining whether there exists $\boldsymbol{\tau}_k$ $(k = 1, 2)$ that satisfies (6.2). And we proved that if the image was obtained from the same object, $\boldsymbol{\tau}_k$ is given by (6.5), where singular value decomposition to H plays an important role since H is of (at most) rank 3. Furthermore, when affine transformations are concerned, the least-norm solution is given by (6.6), which should be recovered. In Subsection 6.2, we proved that the conditions that are incurred for the case of rigid transformations are rewritten as (6.11) and (6.12). We also showed that the least-norm solution of (6.2) that satisfies the conditions, the solution for the rigid transformation case, is given by $\boldsymbol{\tau}_k = V\boldsymbol{y}_k$ where $\boldsymbol{y}_k$

satisfies $y_{kj} = 0$ ($j = 5, 6$; $k = 1, 2$) and is determined by solving (6.11) and (6.12) (cf. (6.13)). Therefore, when a new image is given, we regard (6.2) as an overdetermined system of linear equations in $\boldsymbol{\tau}_k$; and then we apply the method of least squares to see whether the residual is (almost) equal to zero.

To be more specific, we first assume that the given image satisfies (6.2) and then recover $\boldsymbol{\tau}_k$. Next we determine whether the sum of the distance between two vectors, $H\boldsymbol{\tau}_k$ and $\boldsymbol{\rho}_k$, for $k = 1$ and 2 is (almost) equal to zero. Here we define a function *dis* that calculates the distance between two vectors: for two vectors $\boldsymbol{x}$ and $\boldsymbol{y}$,

$$dis(\boldsymbol{x}, \boldsymbol{y}) \ := \ \|\boldsymbol{x} - \boldsymbol{y}\| .$$

We should note that we can calculate the two orthogonal matrices U and V as well as singular values σ_j from the three stored images in advance; and also note that we can calculate the orthogonal matrix G and ψ_j from the three transformations for the stored images beforehand (see (6.4) and (6.9)). We should again remark that in order to calculate them we need not know the three transformations for the stored images when affine transformations are admissible; whereas we must know them in the case of rigid transformations.

The following procedure determines, for a given paraperspective image, whether or not it was obtained from an object to be recognized.

Algorithm

1. Calculate $\tilde{\boldsymbol{\pi}}_*^{\mathrm{G}}$ (see (4.1)).
2. Calculate $\boldsymbol{\rho}_*^p$ for all p ($p \in \{1, 2, \ldots, P\}$) (see (4.2)).
3. $\boldsymbol{d}_k := U^{\mathrm{T}} \boldsymbol{\rho}_k \quad (k = 1, 2)$ (see (6.1)).
4. (a) If affine transformations are considered admissible, then

$$\boldsymbol{\tau}_k \ := \ V \Sigma^{+} \boldsymbol{d}_k \quad (k = 1, 2) \text{ (see (6.6))};$$

   (b) if rigid transformations are considered admissible, then

      i. $\boldsymbol{y}_k := (d_{k1}/\sigma_1, d_{k2}/\sigma_2, d_{k3}/\sigma_3, y_{k4}, 0, 0)^{\mathrm{T}} \quad (k = 1, 2)$.

      ii. $\tilde{\boldsymbol{\tau}}_k := G^{\mathrm{T}} V \boldsymbol{y}_k \quad (k = 1, 2)$.

iii. For $k = 1$ and 2, find y_{k4} that satisfies (6.11) and (6.12); let y_k^* be the solution.

iv. $\boldsymbol{y}_k^* := (d_{k1}/\sigma_1, d_{k2}/\sigma_2, d_{k3}/\sigma_3, y_k^*, 0, 0)^{\mathrm{T}} \quad (k = 1, 2)$.

v. $\boldsymbol{\tau}_k := V\boldsymbol{y}_k^* \quad (k = 1, 2)$.

5. Let cost function h be $h := \sum_{k=1}^{2} dis\,(\boldsymbol{\rho}_k,\ H\boldsymbol{\tau}_k)$.

6. If h is (almost) equal to zero, then the same object; otherwise, a different object. □

The algorithm is very simple as well as single-shot: just linear operations with no iteration.

Remark 6.1 From a theoretical point of view, the decision to be made in Step 6 should be "$h = 0$" whereas from a practical point of view, it should be "$h \approx 0$". This is due to the rounding errors in the numerical computation. □

7 Experimental results

On the basis of the algorithm above, our experimental results with artificial images are shown. And it is found that the algorithm correctly determines whether or not a given image was obtained from the same object. Note that we fixed the focal length at $f = 1.0$, and that the class of admissible transformations was set rigid.

First, paraperspective images were artificially generated from an object to be recognized. The object is a parallelepiped (see Fig. 3) with eight vertices: $(5.00, 6.50, 10.00)$, $(5.80, 5.96, 9.73)$, $(6.16, 7.04, 9.37)$, $(5.36, 7.58, 9.64)$, $(5.45, 6.05, 11.35)$, $(6.25, 5.51, 11.08)$, $(6.61, 6.59, 10.72)$, $(5.81, 7.13, 10.99)$. Then we regarded the seven visible vertices as the feature points. Hence an image of the parallelepiped is represented as the set of the paraperspective images of the seven feature points. Three stored images for the object are shown in Fig. 4. Each image was obtained with a transformation[9] in Table 1. Fig. 5 shows three new images, each of which should be determined whether or not it was obtained from the parallelepiped. The algorithm was applied to the three images. The results are shown in Table 2 (first column). Since the values of $\boldsymbol{\rho}_i^p$ are $\mathrm{O}(10^{-2})$, we set the threshold for the decision "$h \approx 0$" to 1.0×10^{-5} (rather arbitrarily). Then, Table 2 shows the algorithm determines that both (d) and (e) were obtained from the same object, and that (f) was obtained from a different object. Actually in Fig. 5,

[9]All the transformations consist of a rotation around the viewpoint followed by a translation.

(d) was obtained by rotating the parallelepiped by $30°$ around the x axis and then by $-30°$ around the y axis and then translating it by $(1.00, 4.50, -0.50)$; (e) was obtained by rotating the parallelepiped by $5°$, $20°$, and $30°$ around the x, the y and the z axes, respectively, and then translating it by $(-3.00, -5.00, 0.00)$; whereas (f) was obtained from a different object, i.e., a frustum of the pyramid.

On the other hand, in order to find our algorithm could be applied to perspective images, we experimented the case where images were perspectively obtained. Table 2 (second column) also gives the results of the algorithm applied to the perspective images under the same conditions. Note that the stored images and the new images are shown in Fig. 6 and in Fig. 7, respectively. Table 2 also shows our algorithm determines that both (d) and (e) were obtained from the same object, and that (f) was obtained from a different object under the threshold 1.0×10^{-5}. These decisions are again correct even though there are approximation errors in this case. This shows that the algorithm can be applied even to perspective images.

Our experimental results indicate that our algorithm correctly determines whether or not a given image was obtained from the same object under not only paraperspective projection but also perspective projection.

8 Conclusion

It was clarified that paraperspective projection is the first-order approximation of perspective projection. And it was shown that under paraperspective projection, when we represent an object as a set of its feature points, several images are sufficient to express any other image of the same object that has at least six feature points. Particularly when the class of admissible transformations for an object is affine or rigid, the coordinates of the feature points of any image can be described as the linear combination of the coordinates of the feature points of three appropriate images where the coefficients of the combination are parametrized and their number is independent of the number of the feature points (and accordingly, independent of an object to be recognized). In addition, when rigid transformations are considered admissible, the coefficients of the linear combination have to satisfy two conditions: orthogonality and norm equality.

Based on these properties, we proposed a single-shot algorithm for object recognition under paraperspective projection; it is an algorithm of linear least-squares type and makes use of singular value decomposition to recover the values of the coefficients. This indicates that, when a new image is given, we have only to determine whether or not the cost function can be almost nullified by a suitable set of parameter values. As pointed out, we have indeterminacy in recovering the values of the coefficients; hence we introduced a coefficient vector $\boldsymbol{\tau}_k$ by aligning the coefficients and set a criteria of least norm in order to eliminate the indeterminacy. We should note that this indeterminacy could cause an image obtained from a different object to happen to be determined as an image obtained from the same object. Theoretical considerations on such a case are left open in this paper.

We presented some experimental results with artificial paraperspective images and found that the algorithm correctly determines whether or not a given image was obtained from the same object. We also applied our algorithm to perspective images and found that it still works well for recognizing an object though there are approximation errors in addition to the rounding errors in the numerical computation.

Future investigations should include (1) the analysis of the errors incurred by the approximation of perspective projection by paraperspective projection, and (2) the analysis of the rounding errors in the actual implementation of the proposed algorithm.

Acknowledgments

The author is especially thankful to Kazuo Murota of Research Institute for Mathematical Science of Kyoto University for the discussion as well as comments on this work. He is also grateful to Eiji Yodogawa and Keiichi Ueno, who worked at ATR Auditory and Visual Perception Research Laboratories, for providing the opportunity for this research. Finally he appreciates the encouragement of Yoh'ichi Tohkura and Shigeru Akamatsu of ATR Human Information Processing Research Laboratories.

References

[1] J. Aloimonos: Shape from Texture, *Biological Cybernetics*, 58, 5, 345-360 (1989).

[2] J. Aloimonos and A. Basu: Shape and 3-D Motion from Contour without Point to Point Correspondence: General Principles, *Proc. of CVPR*, 518-527, 1986.

[3] P. J. Besl and R. C. Jain: Three-Dimensional Object Recognition, *ACM Computing Surveys*, 1, 17, 75-145 (1985).

[4] R. O. Duda and P. E. Hart: *Pattern Classification and Scene Analysis*, Wiley, 1973.

[5] D. Marr and H. K. Nishihara: Representation and Recognition of the Spatial Organization of Three-Dimensional Shapes, *Proc. of R. Soc. Lond.*, B 100, 269-294 (1978).

[6] J. L. Mundy and A. Zisserman eds.: *Geometric Invariance in Computer Vision*, MIT Press, Cambridge, Massachusetts, 1992.

[7] Y. Ohta, K. Maenobu and T. Sakai: Obtaining Surface Orientation from Texels under Perspective Projection, *Proc. of the 7th IJCAI*, 746-751, 1981.

[8] T. Poggio: *3D Object Recognition: On a Result of Basri and Ullman*, IRST Technical Report, 9005-03, Trento, Italy, 1990.

[9] T. Poggio and S. Edelman: A Network that Learns to Recognize Three-Dimensional Objects, *Nature*, 343, 6225, 263-266, 1990.

[10] C. J. Poelman and T. Kanade: *A Paraperspective Factorization Methods for Shape and Motion Recovery*, CMU-CS-92-208, School of Computer Science, Carnegie Mellon Univ., 1992.

[11] C. A. Rothwell, D. A. Forsyth, A. Zisserman and J. L. Mundy: Extracting Projective Structure from Single Perspective Views of 3D Point Sets, *Proc. of ICCV4*, 573-582, 1993.

[12] A. Sugimoto: *Projective Invariant of Lines on Adjacent Planar Regions in a Single View*, ATR Technical Report, TR-H-034, Kyoto, Japan, 1993.

[13] A. Sugimoto: *Geometric Invariant of Noncoplanar Lines in a Single View*, ATR Technical Report, TR-H-046, Kyoto, Japan, 1993.

[14] A. Sugimoto and K. Murota: 3D Object Recognition by Combination of Perspective Images, *Proc. of SPIE Conf.* Vol. 1904, 183-195, 1993.

[15] A. Sugimoto and K. Murota: Recognition by Combination of Paraperspective Images, *Proc. of The 8th Scandinavian Conf. on Image Analysis*, Vol.2, 1161-1169, 1993.

[16] S. Ullman and R. Basri: Recognition by Linear Combinations of Models, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, PAMI-13, 10, 992-1006 (1991).

[17] I. Weiss: Geometric Invariants and Object Recognition, *Int. J. of Computer Vision*, 10, 3, 207-231 (1993).

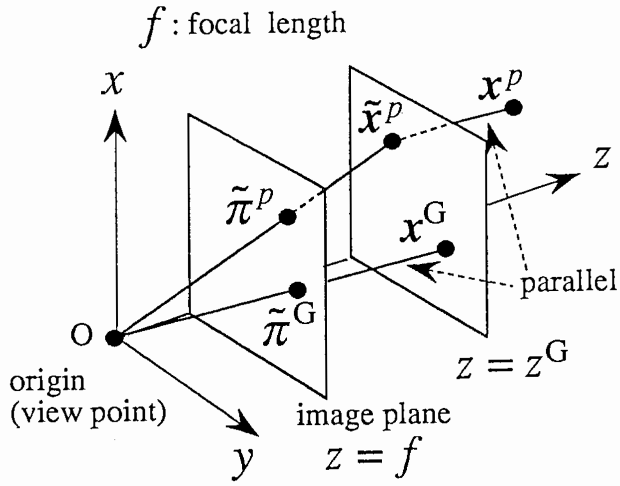


Fig. 1: Principle of paraperspective projection

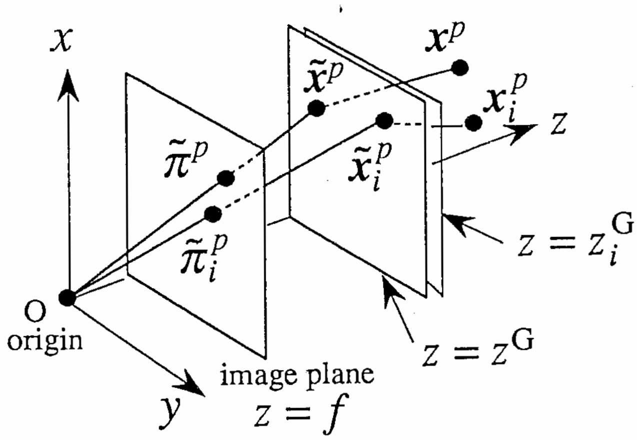


Fig. 2: A transformation i of a point p

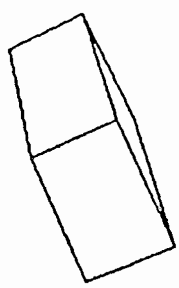
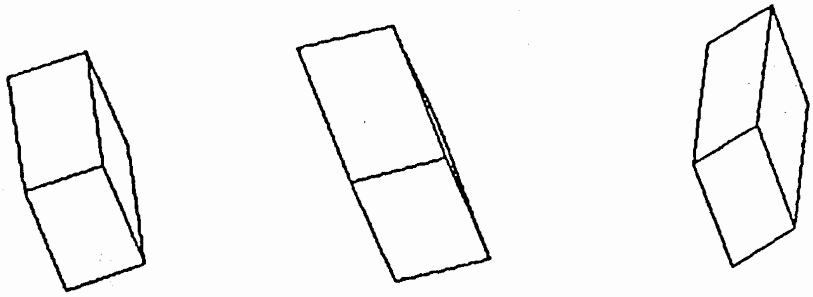
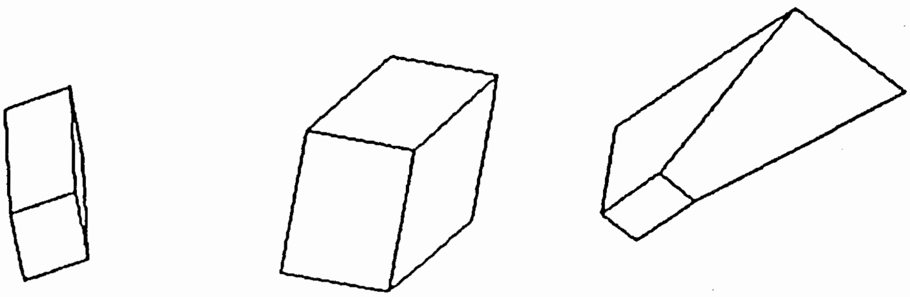


Fig. 3: The parallelepiped to be recognized



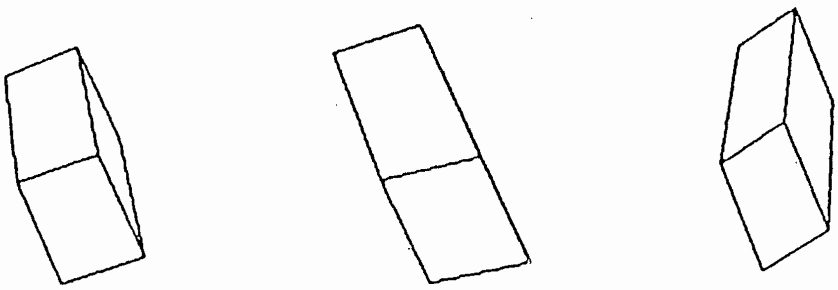
(a) (b) (c)

Fig. 4: The stored images of the object in Fig. 3



(d) (e) (f)

Fig. 5: New images



(a) (b) (c)

Fig. 6: The stored images of the object in Fig. 3 (perspective)

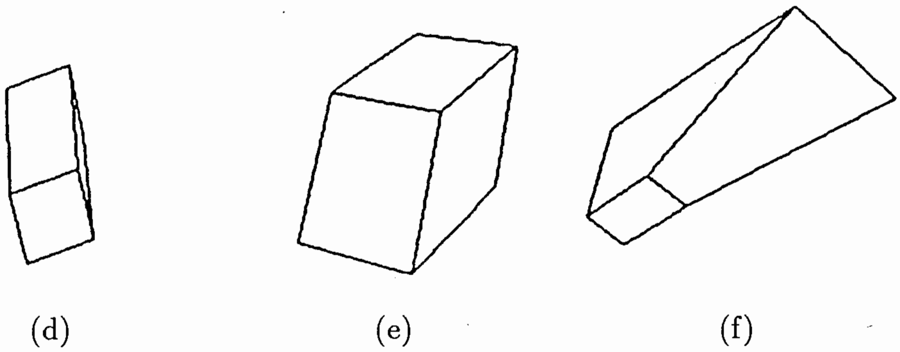


Fig. 7: New images (perspective)

Table 1: Transformations for the stored images

	rotation		translation
	axis	degree	(x, y, z)
(a)	x	10°	(−0.50, 2.00, 0.00)
(b)	x	15°	(2.50, 4.00, −1.00)
(c)	y	−10°	(−4.50, 2.50, 0.50)

Table 2: Minimum values of the cost function h

	h (paraperspective)	h (perspective)
(d)	5.63×10^{-14}	2.86×10^{-7}
(e)	5.99×10^{-13}	4.75×10^{-6}
(f)	1.14×10^{-2}	2.32×10^{-3}