Recognition by Linear Combinations of Models

Shimon Ullman and Ronen Basri

Abstract— **Visual object recognition requires the matching of an image with a set of models stored in memory. In this paper, we propose an approach to recognition in which a 3-D object is represented by the linear combination of 2-D images of the object. If $\mathcal{M}=\{M_1,\cdots,M_k\}$ is the set of pictures representing a given object and P is the 2-D image of an object to be recognized, then P is considered to be an instance of M if $P=\sum_{i=1}^{k}\alpha_i M_i$ for some constants α_i. We show that this approach handles correctly rigid 3-D transformations of objects with sharp as well as smooth boundaries and can also handle nonrigid transformations. The paper is divided into two parts. In the first part, we show that the variety of views depicting the same object under different transformations can often be expressed as the linear combinations of a small number of views. In the second part, we suggest how this linear combination property may be used in the recognition process.**

Index Terms—**Alignment, linear combinations, object recognition, recognition, 3-D object recognition, visual recognition.**

I. Modeling Objects by the Linear Combination of Images

A. Recognition by Alignment

VISUAL OBJECT recognition requires the matching of an image with a set of models stored in memory. Let $\mathcal{M} = \{M_1,\cdots,M_n\}$ be the set of stored models and P be the image to be recognized. In general, the viewed object, depicted by P, may differ from all the previously seen images of the same object. It may be, for instance, the image of a three-dimensional object seen from a novel viewing position. To compensate for these variations, we may allow the models (or the viewed object) to undergo certain compensating transformations during the matching stage. If $\mathcal{T}$ is the set of allowable transformations, the matching stage requires the selection of a model $M_i \in \mathcal{M}$ and a transformation $T \in \mathcal{T}$, such that the viewed object P and the transformed model TM_i will be as close as possible. The general scheme is called the alignment approach since an alignment transformation is applied to the model (or to the viewed object) prior to, or during, the matching stage. Such an approach is used in [5], [7], [8], [12], [16], [20], and [23]. Key problems that arise in any alignment scheme are how to represent the set of different models $\mathcal{M}$, what is the set of allowable transformations $\mathcal{T}$, and for a given model $M_i \in \mathcal{M}$, how to determine the transformation $T \in \mathcal{T}$ to minimize the difference between P and TM_i. For example, in the scheme proposed by Basri and Ullman [3], a model is represented by a set of 2-D contours, with associated depth and curvature values at each contour point. The set of allowed transformations includes 3-D rotation, translation, and scaling, followed by an orthographic projection. The transformation is determined as in [12] and [23] by identifying at least three corresponding features (points or lines) in the image and the object.

In this paper, we suggest a different approach, in which each model is represented by the linear combination of 2-D images of the object. The new approach has several advantages. First, it handles all the rigid 3-D transformations, but it is not restricted to such transformations. Second, there is no need in this scheme to explicitly recover and represent the 3-D structure of objects. Third, the computations involved are often simpler than in previous schemes.

The paper is divided into two parts. In the first (Section I), we show that the variety of views depicting the same object under different transformations can often be expressed as the linear combinations of a small number of views. In the second part (Section II), we suggest how this linear combination property may be used in the recognition process.

B. Using Linear Combinations of Images to Model Objects and Their Transformations

The modeling of objects using linear combinations of images is based on the following observation. For many continuous transformations of interest in recognition, such as 3-D rotation, translation, and scaling, all the possible views of the transforming object can be expressed simply as the linear combination of other views of the same object. The coefficients of these linear combinations often follow in addition to certain functional restrictions. In the next two sections, we show that the set of possible images of an object undergoing rigid 3-D transformations and scaling is embedded in a linear space and spanned by a small number of 2-D images.

The images we will consider are 2-D edge maps produced in the image by the (orthographic) projection of the bounding contours and other visible contours on 3-D objects. We will make use of the following definitions. Given an object and a viewing direction, the *rim* is the set of all the points on the object's surface whose normal is perpendicular to the viewing direction [13]. This set is also called the *contour generator* [17]. A *silhouette* is an image generated by the orthographic projection of the rim. In the analysis below, we assume that every point along the silhouette is generated by a single rim

Manuscript received November 1, 1990; revised December 30, 1990. This work was supported by an Office of Naval Research University Research Initiative grant under contract N00014-86-K-0685, the Advanced Research Projects Agency of the Department of Defense under Army contract DAC A76-85-C-0010, Office of Naval Research contract N00014-85-K-0124, and by NSF Grant IRI-8900267.

The authors are with the Department of Brain and Cognitive Science and the Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, MA 02139 and the Department of Applied Mathematics, The Weizmann Institute of Science, Rehovot, Israel.

IEEE Log Number 9102644.

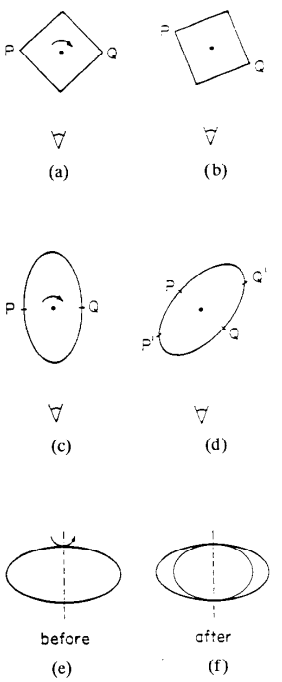

Fig. 1. Changes in the rim during rotation: (a) Bird's eye view of a cube; (b) cube after rotation. In both (a) and (b) points p, Q lie on the rim; (c) bird's eye view of an ellipsoid; (d) ellipsoid after rotation. The rim points p, Q in (c) are replaced by p', Q' in (d); (e) ellipsoid in a frontal view; (f) rotated ellipsoid (outer), superimposed on the appearance of the rim as a planar space curve after rotation by the same amount (inner) (from [3]).

point. An edge map of an object usually contains the silhouette, which is generated by its rim.

We will examined two cases below: the case of objects with sharp edges and the case of objects with smooth boundary contours. The difference between these two cases is illustrated in Fig. 1. For an object with sharp edges, such as the cube in Fig. 1(a) and (b), the rim is stable on the object as long as the edge is visible. In contrast, a rim that is generated by smooth bounding surfaces, such as in the ellipsoid in Fig. 1(c) and (d) is not fixed on the object but changes continuously with the viewpoint.

In both cases, a small number of images M_i, with known correspondence, will constitute the object's model. Given a new image P, the problem is then to determine whether P belongs to the same object represented by the M_i's. We will not directly address here the problem of segmentation, i.e., separating an object or a part of it from the image of a scene.

C. Objects with Sharp Edges

In the discussion below, we examine the case of objects with sharp edges undergoing different transformations followed by an orthographic projection. In each case, we show how the image of an object obtained by the transformation in question can be expressed as the linear combination of a small number of pictures. The coefficients of this combination may be different for the x and y coordinates, that is, the intermediate view of the object may be given by two linear combinations: one for the x coordinates and the other for the y coordinates. In addition, certain functional restrictions may hold among the different coefficients.

To introduce the scheme, we first apply it to the restricted case of rotation about the vertical axis and then examine more general transformations.

1) 3-D Rotation Around the Vertical Axis: Let P_1 and P_2 be two images of an object O rotating in depth around the vertical axis (Y axis). P_2 is obtained from P_1 following a rotation by an angle α ($\alpha \neq k\pi$). Let $\hat{P}$ be a third image of the same object obtained from P_1 by a rotation of an angle θ around the vertical axis. The projections of a point $p = (x, y, z) \in O$ in the three images are given by

$$\begin{aligned} p_1 &= (x_1, y_1) = (x, y) && \in P_1 \\ p_2 &= (x_2, y_2) = (x\cos\alpha + z\sin\alpha, y) && \in P_2 \\ \hat{p} &= (\hat{x}, \hat{y}) \quad = (x\cos\theta + z\sin\theta, y) && \in \hat{P}. \end{aligned}$$

Claim: Two scalars a and b exist, such that for every point $p \in O$

$$\hat{x} = ax_1 + bx_2.$$

The coefficients a and b are the same for all the points, with

$$a^2 + b^2 + 2ab\cos\alpha = 1.$$

Proof: The scalars a and b are given explicitly by

$$\begin{aligned} a &= \frac{\sin(\alpha - \theta)}{\sin\alpha} \\ b &= \frac{\sin\theta}{\sin\alpha}. \end{aligned}$$

Then

$$\begin{aligned} ax_1 + bx_2 &= \frac{\sin(\alpha - \theta)}{\sin\alpha}x + \frac{\sin\theta}{\sin\alpha}(x\cos\alpha + z\sin\alpha) \\ &= x\cos\theta + z\sin\theta = \hat{x}. \end{aligned}$$

Therefore, an image of an object rotating around the vertical axis is always a linear combination of two model images. It is straightforward to verify that the coefficients a and b satisfy the above constraint. It is worth noting that the new view $\hat{P}$ is not restricted to be an intermediate view (that is, the rotation angle θ may be larger than α). Finally, it should be noted that we do not deal at this stage with occlusion; we assume here that the same set of points is visible in the different views. The issue of occlusion and self-occlusion will be discussed further below.

2) Linear Transformations in 3-D Space: Let O be a set of object points. Let P_1, P_2, and P_3 be three images of O obtained by applying 3×3 matrices R, S, and T to O, respectively. (In particular, R can be the identity matrix, and S, T can be two rotations producing the second and third views.) Let $\hat{P}$ be a fourth image of the same object obtained by applying a different 3×3 matrix U to O. Let $\boldsymbol{r}_1$, $\boldsymbol{s}_1$, $\boldsymbol{t}_1$, and $\boldsymbol{u}_1$ be the first row vectors of R, S, T, and U, respectively, and let $\boldsymbol{r}_2$, $\boldsymbol{s}_2$, $\boldsymbol{t}_2$ and $\boldsymbol{u}_2$ be the second row vectors of R, S, T, and

U, respectively. The positions of a point $p \in O$ in the four images are given by

$$p_1 = (x_1, y_1) = (\boldsymbol{r}_1 p, \boldsymbol{r}_2 p)$$
$$p_2 = (x_2, y_2) = (\boldsymbol{s}_1 p, \boldsymbol{s}_2 p)$$
$$p_3 = (x_3, y_3) = (\boldsymbol{t}_1 p, \boldsymbol{t}_2 p)$$
$$\hat{p} = (\hat{x}, \hat{y}) \quad = (\boldsymbol{u}_1 p, \boldsymbol{u}_2 p).$$

Claim: If both sets $\{\boldsymbol{r}_1, \boldsymbol{s}_1, \boldsymbol{t}_1\}$ and $\{\boldsymbol{r}_2, \boldsymbol{s}_2, \boldsymbol{t}_2\}$ are linearly independent, then there exist scalars a_1, a_2, a_3, and b_1, b_2, b_3 such that for every point $p \in O$, it holds that

$$\hat{x} = a_1 x_1 + a_2 x_2 + a_3 x_3$$
$$\hat{y} = b_1 y_1 + b_2 y_2 + b_3 y_3.$$

Proof: $\{\boldsymbol{r}_1, \boldsymbol{s}_1, \boldsymbol{t}_1\}$ are linearly independent. Therefore, they span $\mathcal{R}^3$, and there exist scalars a_1, a_2, and a_3 such that

$$\boldsymbol{u}_1 = a_1 \boldsymbol{r}_1 + a_2 \boldsymbol{s}_1 + a_3 \boldsymbol{t}_1.$$

Since

$$\hat{x} = \boldsymbol{u}_1 p$$

it follows that

$$\hat{x} = a_1 \boldsymbol{r}_1 p + a_2 \boldsymbol{s}_1 p + a_3 \boldsymbol{t}_1 p.$$

Therefore

$$\hat{x} = a_1 x_1 + a_2 x_2 + a_3 x_3.$$

In a similar way, we obtain that

$$\hat{y} = b_1 y_1 + b_2 y_2 + b_3 y_3.$$

Therefore, an image of an object undergoing a linear transformation in 3-D space is a linear combination of three model images.

3) General Rotation in 3-D Space: Rotation is a nonlinear subgroup of the linear transformations. Therefore, an image of a rotating object is still a linear combination of three model images. However, not every point in this linear space represents a pure rotation of the object. Indeed, we can show that only points that satisfy the following three constraints represent images of a rotating object.

Claim: The coefficients of an image of a rotating object must satisfy the three following constraints:

$$\|a_1 \boldsymbol{r}_1 + a_2 \boldsymbol{s}_1 + a_3 \boldsymbol{t}_1\| = 1$$
$$\|b_1 \boldsymbol{r}_2 + b_2 \boldsymbol{s}_2 + b_3 \boldsymbol{t}_2\| = 1$$
$$(a_1 \boldsymbol{r}_1 + a_2 \boldsymbol{s}_1 + a_3 \boldsymbol{t}_1)(b_1 \boldsymbol{r}_2 + b_2 \boldsymbol{s}_2 + b_3 \boldsymbol{t}_2) = 0$$

Proof: U is a rotation matrix. Therefore

$$\|\boldsymbol{u}_1\| = 1$$
$$\|\boldsymbol{u}_2\| = 1$$
$$\boldsymbol{u}_1 \boldsymbol{u}_2 = 0$$

and the required terms are obtained directly by substituting $\boldsymbol{u}_1$ and $\boldsymbol{u}_2$ with the appropriate linear combinations. It also follows immediately that if the constraints are met, then the new view represents a possible rotation of the object, that is, the linear combination condition together with the constraints provide necessary and sufficient conditions for the novel view to be a possible projection of the same 3-D object.

These functional constraints are second-degree polynomials in the coefficients and therefore span a nonlinear manifold within the linear subspace. In order to check whether a specific set of coefficients represents a rigid rotation, the values of the matrices R, S, and T can be used. These can be retrieved by applying methods of "structure from motion" to the model views. Ullman [21] showed that in case of rigid transformations, four corresponding points in three views are sufficient for this purpose. An algorithm that can be used to recover the rotation matrices using mainly linear equations has been suggested by Huang and Lee [11]. (The same method can be extended to deal with scale changes in addition to the rotation).

It should be noted that in some cases, the explicit computation of the rotation matrices will not be necessary. First, if the set of allowable object transformations includes the entire set of linear 3-D transformations (including nonrigid stretch and shear), then no additional test of the coefficients is required. Second, if the transformations are constrained to be rigid but the test of the coefficient is not performed, then the penalty may be some "false positives" misidentifications. If the image of one object happens to be identical to the projection of a (nonrigid) linear transformation applied to another object, then the two will be confuseable. If the objects contain a sufficient number of points (five or more), the likelihood of such an ambiguity becomes negligible. Finally, it is worth noting that it is also possible to determine the coefficient of the constraint equations above without computing the rotation matrices, by using a number of additional views (see also Section I-C-5).

Regarding the independence condition mentioned above, for many triplets of rotation matrices R, S, and T both $\{\boldsymbol{r}_1, \boldsymbol{s}_1, \boldsymbol{t}_1\}$ and $\{\boldsymbol{r}_2, \boldsymbol{s}_2, \boldsymbol{t}_2\}$ will in fact be linearly independent. It will therefore be possible to select a nondegenerate triplet of views (P_1, P_2, and P_3) in terms of which intermediate views are expressible as linear combinations. Note, however, that in the special case that R is the identity matrix, S is a pure rotation about the X axis, and T is a pure rotation about the Y axis, the independent condition does not hold.

4) Rigid Transformations and Scaling in 3-D Space: We have considered above the case of rigid rotation in 3-D space. If, in addition, the object is allowed to undergo translation and a scale change, novel views will still be the linear combinations of three 2-D views of the object. More specifically, let O be a set of object points, and let P_1, P_2, and P_3 be three images of O, which are obtained by applying the 3×3 rotation matrices R, S, and T to O, respectively. Let $\hat{P}$ be a fourth image of the same object obtained by applying a 3×3 rotation matrix U to O, scaling by a scale factor s, and translating by a vector (t_x, t_y). Let $\boldsymbol{r}_1$, $\boldsymbol{s}_1$, $\boldsymbol{t}_1$, and $\boldsymbol{u}_1$ again be the first row vectors of R, S, T, and U and $\boldsymbol{r}_2$, $\boldsymbol{s}_2$, $\boldsymbol{t}_2$, and $\boldsymbol{u}_2$ the second row vectors of R, S, T, and U, respectively. For any point $p \in O$,

its positions in the four images are given by

$$\begin{aligned} p_1 &= (x_1, y_1) = (\boldsymbol{r}_1 p, \boldsymbol{r}_2 p) \\ p_2 &= (x_2, y_2) = (\boldsymbol{s}_1 p, \boldsymbol{s}_2 p) \\ p_3 &= (x_3, y_3) = (\boldsymbol{t}_1 p, \boldsymbol{t}_2 p) \\ \hat{p} &= (\hat{x}, \hat{y}) \quad = (s\boldsymbol{u}_1 p + t_x, s\boldsymbol{u}_2 p + t_y). \end{aligned}$$

Claim: If both sets $\{\boldsymbol{r}_1, \boldsymbol{s}_1, \boldsymbol{t}_1\}$ and $\{\boldsymbol{r}_2, \boldsymbol{s}_2, \boldsymbol{t}_2\}$ are linearly independent, then there exist scalars a_1, a_2, a_3, a_4 and b_1, b_2, b_3, b_4, such that for every point $p \in O$, it holds that

$$\begin{aligned} \hat{x} &= a_1 x_1 + a_2 x_2 + a_3 x_3 + a_4 \\ \hat{y} &= b_1 y_1 + b_2 y_2 + b_3 y_3 + b_4 \end{aligned}$$

with the coefficient satisfying the two constraints

$$\begin{aligned} \|a_1\boldsymbol{r}_1 + a_2\boldsymbol{s}_1 + a_3\boldsymbol{t}_1\| &= \|b_1\boldsymbol{r}_2 + b_2\boldsymbol{s}_2 + b_3\boldsymbol{t}_2\| \\ (a_1\boldsymbol{r}_1 + a_2\boldsymbol{s}_1 + a_3\boldsymbol{t}_1)(b_1\boldsymbol{r}_2 + b_2\boldsymbol{s}_2 + b_3\boldsymbol{t}_2) &= 0. \end{aligned}$$

We can view each of the above equations as a linear combination of three images and a fourth constant vector. (Instead of a constant vector, one can take a fourth image generated internally by shifting one of the three images.) The proof is almost identical to the one in Section I-C-3 and therefore will not be detailed. As for the constraints on the coefficients, since U is a rotation matrix

$$\begin{aligned} \|\boldsymbol{u}_1\| &= 1 \\ \|\boldsymbol{u}_2\| &= 1 \\ \boldsymbol{u}_1\boldsymbol{u}_2 &= 0 \end{aligned}$$

it follows that

$$\begin{aligned} \|s\boldsymbol{u}_1\| &= \|s\boldsymbol{u}_2\| \\ (s\boldsymbol{u}_1)(s\boldsymbol{u}_2) &= 0 \end{aligned}$$

and the constraints are obtained directly by substituting the appropriate linear combinations for $s\boldsymbol{u}_1$ and $s\boldsymbol{u}_2$.

5) Using Two Views Only: In the scheme described above, any image of a given object (within a certain range of rotations) is expressed as the linear combination of three fixed views of the object. For general linear transformations, it is also possible to use instead just two views of the object. (This observation was made independently by T. Poggio and R. Basri.)

Let O be again a rigid object (a collection of 3-D points). P_1 in a 2-D image of O, and let P_2 be the image of O following a rotation by R (a 3×3 matrix). We will denote by $\boldsymbol{r}_1, \boldsymbol{r}_2, \boldsymbol{r}_3$ the three rows of R and by $\boldsymbol{e}_1, \boldsymbol{e}_2, \boldsymbol{e}_3$ the three rows of the identity matrix. For a given 3-D point $\boldsymbol{p}$ in O, its coordinates (x_1, y_1) in the first image view are $x_1 = \boldsymbol{e}_1\boldsymbol{p}$, $y_1 = \boldsymbol{e}_2\boldsymbol{p}$. Its coordinates (x_2, y_2) in the second view are given by $x_2 = \boldsymbol{r}_1\boldsymbol{p}$, $y_2 = \boldsymbol{r}_2\boldsymbol{p}$.

Consider now any other view obtained by applying another 3×3 matrix $\boldsymbol{U}$ to the points of O. The coordinates $(\hat{x}, \hat{y})$ of $\boldsymbol{p}$ in this new view will be

$$\hat{x} = \boldsymbol{u}_1\boldsymbol{p}, \qquad \hat{y} = \boldsymbol{u}_2\boldsymbol{p}$$

(where $\boldsymbol{u}_1$, $\boldsymbol{u}_2$, are the first second rows of $\boldsymbol{U}$, respectively).

Assuming that $\boldsymbol{e}_1$, $\boldsymbol{e}_2$ and $\boldsymbol{r}_1$ span $\mathcal{R}^3$ (see below), then

$$\boldsymbol{u}_1 = a_1\boldsymbol{e}_1 + a_2\boldsymbol{e}_2 + a_3\boldsymbol{r}_1$$

for some scalars a_1, a_2, a_3. Therefore

$$\begin{aligned} \hat{x} = \boldsymbol{u}_1\boldsymbol{p} &= (a_1\boldsymbol{e}_1 + a_2\boldsymbol{e}_2 + a_3\boldsymbol{r}_1)\boldsymbol{p} \\ &= a_1 x_1 + a_2 y_1 + a_3 x_2. \end{aligned}$$

This equality holds for every point $\boldsymbol{p}$ in O. Let $\boldsymbol{x}_1$ be the vector of all the x coordinates of the points in the first view, $\boldsymbol{x}_2$ in the second, $\hat{\boldsymbol{x}}$ in the third, and $\boldsymbol{y}_1$ the vector of y coordinates in the first view. Then

$$\hat{\boldsymbol{x}} = a_1\boldsymbol{x}_1 + a_2\boldsymbol{y}_1 + a_3\boldsymbol{x}_2.$$

Here $\boldsymbol{x}_1$, $\boldsymbol{y}_1$, and $\boldsymbol{x}_2$ are used as a basis for all of the views. For any other image of the same object, its vector $\hat{\boldsymbol{x}}$ of x coordinates is the linear combination of these basis vectors.

Similarly, for the y coordinates

$$\hat{\boldsymbol{y}} = b_1\boldsymbol{x}_1 + b_2\boldsymbol{y}_1 + b_3\boldsymbol{x}_2.$$

The vector $\hat{\boldsymbol{y}}$ of y coordinates in the new image is therefore also the linear combination of the same three basis vectors. In this version, the basis vectors are the same for the x and y coordinates, and they are obtained from two rather then three views. One can view the situation as follows. Within an n-dimensional space, the vectors $\boldsymbol{x}_1$, $\boldsymbol{y}_1$, $\boldsymbol{x}_2$ span a three-dimensional subspace. For all the images of the object in question, the vectors of both the x and y coordinates must reside within this three-dimensional subspace.

Instead of using $(\boldsymbol{e}_1, \boldsymbol{e}_2, \boldsymbol{r}_1)$ as the basis for $\mathcal{R}^3$, we could also use $(\boldsymbol{e}_1, \boldsymbol{e}_2, \boldsymbol{r}_2)$. One of these bases spans $\mathcal{R}^3$, unless the rotation R is a pure rotation around the line of sight.

The use of two views described above is applicable to general linear transformations of the object, and without additional constraints, it is impossible to distinguish between rigid and linear but not rigid transformations of the object. To impose rigidity (with possible scaling), the coefficients $(a_1, a_2, a_3, b_1, b_2, b_3)$ must meet two simple constraints. Since $\boldsymbol{U}$ is now a rotation matrix (with possible scaling)

$$\begin{aligned} \boldsymbol{u}_1\boldsymbol{u}_2 &= 0 \\ \|\boldsymbol{u}_1\| &= \|\boldsymbol{u}_2\|. \end{aligned}$$

In terms of the coefficients a_i, b_i, $\boldsymbol{u}_1, \boldsymbol{u}_2 = 0$ implies

$$\begin{aligned} a_1b_1 + a_2b_2 + a_3b_3 + (a_1b_3 + a_3b_1)r_{11} + \\ (a_2b_3 + a_3b_2)r_{12} = 0. \end{aligned}$$

The second constraint implies

$$\begin{aligned} a_1^2 + a_2^2 + a_3^2 - b_1^2 - b_2^2 - b_3^2 = 2(b_1b_3 - a_1a_3)r_{11} \\ + 2(b_2b_3 - a_2a_3)r_{12}. \end{aligned}$$

A third view can therefore be used to recover, using two linear equations, the values of r_{11} and r_{12}. (r_{11} and r_{12} can in fact be determined to within a scale factor from the first two views; only one additional equation is required.) The full scheme for rigid objects is, then, the following. Given an image, determine whether the vectors $\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}$, are linear combinations of $\boldsymbol{x}_1$, $\boldsymbol{y}_1$ and $\boldsymbol{x}_2$. Only two views are required for this stage.

Using the values of r_{11} and r_{12}, test whether the coefficients $a_i, b_i, (i = 1, 2, 3)$ satisfy the two constraints above.

It is of interest to compare this use of two views to structure-from-motion (SFM) techniques for recovering 3-D structure from orthographic projections. It is well known that three distinct views are required; two are insufficient [21]. Given only two views and an infinitesimal rotation (the velocity field), the 3-D structure can be recovered to within depth-scaling [22]. It is also straightforward to establish that if the two views are separated by a general affine transformation of the 3-D object (rather than a rigid one), then the structure of the object can be recovered to within an affine transformation.

Our use of two views above for the purpose of recognition is thus related to known results regarding the recovery of structure from motion. Two views are sufficient to determine the object's structure to within an affine transformation, and three are required to recover the full 3-D structure of a rigidly moving object. Similarly, the linear combination scheme uses in the match two (for general linear transformation) or three views (for rigid rotation and scaling). The matching does not require, however, the full 3-D model. Instead, linear combinations of the 2-D images are used directly.

Finally, it can also be observed that an extension of the scheme above can be used to recover structure from motion. It was shown how the scheme can be used to recover r_{11} and r_{12}. r_{21} and r_{22} can be recovered in a similar manner. Consequently, it becomes possible to recover 3-D structure and motion in space based on three orthographic views, using linear equations. (For alternative methods that use primarily linear equations, see [15] and [11]).

6) Summary: In this section, we have shown that an object with sharp contours, undergoing rigid transformations and scaling in 3-D space followed by an orthographic projection, can be expressed as the linear combination of three images of the same object. In this scheme, the model of a 3-D object consists of a number of 2-D pictures of it. The pictures are in correspondence in the sense that it is known which are the corresponding points in the different pictures. Two images are sufficient to represent general linear transformations of the object; three images are required to represent rigid transformations in 3-D space.

The linear combination scheme assumes that the same object points are visible in the different views. When the views are sufficiently different, this will no longer hold due to self-occlusion. To represent an object from all possible viewing directions (e.g., both "front" and "back"), a number of different models of this type will be required. This notion is similar to the use of different object aspects suggested by Koenderink and Van Doorn [13]. (Other aspects of occlusion are examined in the final discussion and Appendix C.)

The linear combination scheme described above was implemented and applied first to artificially created images. Fig. 2 shows examples of object models and their linear combinations. The figure shows how 3-D similarity transformations can be represented by the linear combinations of four images.

D. Objects with Smooth Boundaries

The case of objects with smooth boundaries is identical to the case of objects with sharp edges as long as we deal with translation, scaling, and image rotation. The difference arises when the object rotates in 3-D space. This case is discussed in [3], where we have suggested a method for predicting the appearance of such objects following 3-D rotations. This method, called "the curvature method," is summarized briefly below.

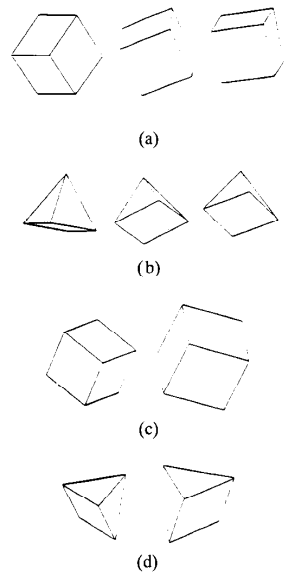

Fig. 2. (a) Three model pictures of a cube. The second picture was obtained by rotating the cube by 30° around the X axis and then by 30° around the Y axis. The third picture was obtained by rotating the cube by 30° around the Y axis and then by 30° around the X axis; (b) three model pictures of a pyramid taken with the same transformations as the pictures in (a); (c) two linear combinations of the cube model. The left picture was obtained using the following parameters: the x coefficients are $(0.343, -2.618, 2.989, 0)$, and the y coefficients are $(0.630, -2.533, 2.658, 0)$, which correspond to a rotation of the cube by 10, 20, and 45° around the X, Y, and Z axes, respectively. The right picture was obtained using the following parameter: x coefficients $(0.455, 3.392, -3.241, 0.25)$ and y coefficients $(0.542, 3.753, -3.343, -0.15)$. These coefficients correspond to a rotation of the cube by 20, 10, and $-45°$ around the X, Y and Z axes, respectively, followed by a scaling factor 1.2 and a translation of $(25, -15)$ pixels; (d) two linear combinations of the pyramid model taken with the same parameters as the picture in (c).

A model is represented by a set of 2-D contours. Each point $p = (x, y)$ along the contours is labeled with its depth value z and a curvature value r. The curvature value is the length of a curvature vector at p, $r = \|(r_x, r_y)\|$. (r_x is the surface's radius of curvature at p in a planar section in the X direction, r_y in the Y direction.) This vector is normal to the contour at p. Let V_ϕ be an axis lying in the image plane and forming an angle ϕ with the positive X direction, and let $\boldsymbol{r}_\phi$ be a vector of length $r_\phi = r_y \cos\phi - r_x \sin\phi$ and perpendicular to V_ϕ. When the object is rotated around V_ϕ, we approximate the new position of the point p in the image by

$$p' = R(p - \boldsymbol{r}_\phi) + \boldsymbol{r}_\phi \tag{1}$$

where R is the rotation matrix. The equation has the following meaning. When viewed in a cross section perpendicular to the rotation axis V_ϕ, the surface at p can be approximated by a

circular arc with radius r_ϕ and center at $p - r_\phi$. The new rim point p' is obtained by first applying R to this center of curvature $(p - r_\phi)$, then adding the radius of curvature r_ϕ. This expression is precise for circular arcs and gives a good approximation for other surfaces provided that the angle of rotation is not too large (see [3] for details). The depth and the curvature values were estimated in [3], using three pictures of the object, and the results were improved using five pictures. In this section, we show how the curvature method can also be replaced by linear combinations of a small number of pictures. In particular, three images are required to represent rotations around a fixed axis and five images for general rotations in 3-D space.

1) General Rotation in 3-D Space: In this section, we first derive an expression for the image deformation of an object with smooth boundaries under general 3-D rotation. We then use this expression to show that the deformed image can be expressed as the linear combination of five images.

a) Computing the transformed image: Using the curvature method, we can predict the appearance of an object undergoing a general rotation in 3-D space as follows. A rotation in 3-D space can be decomposed into the following three successive rotations: a rotation around the Z axis, a subsequent rotation around the X axis and a final rotation around the Z axis by angles α, β, and γ, respectively. Since the Z axis coincides with the line of sight, a rotation around the Z axis is simply an image rotation. Therefore, only the second rotation deforms the object, and the curvature method must be applied to it. Suppose that the curvature vector at a given point $p = (x, y)$ before the first Z rotation is (r_x, r_y). Following the rotation by α, it becomes $r'_x = r_x \cos\alpha - r_y \sin\alpha$ and $r'_y = r_x \sin\alpha + r_y \cos\alpha$. The second rotation is around the X axis, and therefore, the appropriate r_ϕ to be used in (1) becomes $r'_y = r_x \sin\alpha + r_y \cos\alpha$. The complete rotation (all three rotations) therefore takes a point $p = (x, y)$ through the following sequence of transformations:

$$
\begin{aligned}
&(x, y) \rightarrow (x\cos\alpha - y\sin\alpha, x\sin\alpha + y\cos\alpha) \rightarrow \\
&(x\cos\alpha - y\sin\alpha, (x\sin\alpha + y\cos\alpha)\cos\beta - z\sin\beta \\
&\quad + (r_x\sin\alpha + r_y\cos\alpha)(1 - \cos\beta)) \rightarrow \\
&\big((x\cos\alpha - y\sin\alpha)\cos\gamma - ((x\sin\alpha + y\cos\alpha)\cos\beta \\
&\quad - z\sin\beta + (r_x\sin\alpha + r_y\cos\alpha)(1 - \cos\beta))\sin\gamma, \\
&(x\cos\alpha - y\sin\alpha)\sin\gamma + ((x\sin\alpha + y\cos\alpha)\cos\beta - z\sin\beta \\
&\quad + (r_x\sin\alpha + r_y\cos\alpha)(1 - \cos\beta))\cos\gamma\big).
\end{aligned}
$$

(The first of these transformations is the first Z rotation, the second is the deformation caused by the X rotation, and the third is the final Z rotation).

This is an explicit expression of the final coordinates of a point on the object's contour. This can also be expressed more compactly as follows. Let $R = \{r_{ij}\}$ be a 3×3 rotation matrix. Let α, β, and γ be the angles of the Z-X-Z rotations represented by R. We construct a new matrix $R' = \{r'_{ij}\}$ of size 2×5 as follows:

$$
R' = \begin{pmatrix} r_{11} & r_{12} & r_{13} & -\sin\alpha(1-\cos\beta)\sin\gamma \\ r_{21} & r_{22} & r_{23} & \sin\alpha(1-\cos\beta)\cos\gamma \\ & & & -\cos\alpha(1-\cos\beta)\sin\gamma \\ & & & \cos\alpha(1-\cos\beta)\cos\gamma \end{pmatrix}.
$$

Let $p = (x, y)$ be a contour point with depth z and curvature vector (r_x, r_y), and let $\tilde{p} = (x, y, z, r_x, r_y)$. Then, the new appearance of p after a rotation R is applied to the object is described by

$$
p' = R'\tilde{p}. \tag{2}
$$

This is true because (2) is equivalent to (1) in Section I-D with the appropriate values for r_ϕ.

b) Expressing the transformed image as a linear combination: Let O be a set of points of an object rotating in 3-D space. Let P_1, P_2, P_3, P_4, and P_5 be five images of O, which are obtained by applying a rotation matrix $R_1, \cdots, R_5$ respectively. $\hat{P}$ is an image of the same object obtained by applying a rotation matrix $\hat{R}$ to O. Let $R'_1, \cdots, R'_5, \hat{R}'$ be the corresponding 2×5 matrices representing the transformations applied to the contour points according to the curvature method. Finally, let $\boldsymbol{r}_1, \cdots, \boldsymbol{r}_5, \hat{\boldsymbol{r}}$ denote the first row vectors of $R'_1, \cdots, R'_5, \hat{R}'$ and $\boldsymbol{s}_1, \cdots, \boldsymbol{s}_5, \hat{\boldsymbol{s}}$ the second row vectors $R'_1, \cdots, R'_5, \hat{R}'$, respectively. The positions of a point $p = (x, y) \in O$, $\tilde{p} = (x, y, z, r_x, r_y)$ in the six pictures is then given by

$$
\begin{aligned}
p_i = (x_i, y_i) &= (\boldsymbol{r}_i\tilde{p}, \boldsymbol{s}_i\tilde{p}) \in P_i, \qquad 1 \le i \le 5 \\
\hat{p} = (\hat{x}, \hat{y}) &= (\hat{\boldsymbol{r}}\tilde{p}, \hat{\boldsymbol{s}}\tilde{p}) \quad \in \hat{P}.
\end{aligned}
$$

Claim: If both sets $\{\boldsymbol{r}_1, \cdots, \boldsymbol{r}_5\}$ and $\{\boldsymbol{s}_1, \cdots, \boldsymbol{s}_5\}$ are linearly independent vectors, then there exist scalars $a_1, \cdots, a_5$ and $b_1, \cdots, b_5$ such that for every point $p \in O$, it holds that

$$
\hat{x} = \sum_{i=1}^{5} a_i x_i
$$

$$
\hat{y} = \sum_{i=1}^{5} b_i y_i.
$$

Proof: $\{\boldsymbol{r}_1, \cdots, \boldsymbol{r}_5\}$ are linearly independent. Therefore, they span $\mathcal{R}^5$, and there exist scalars $a_1, \cdots, a_5$ such that

$$
\hat{\boldsymbol{r}} = \sum_{i=1}^{5} a_i \boldsymbol{r}_i.
$$

Since

$$
\hat{x} = \hat{\boldsymbol{r}}\tilde{p}.
$$

Then

$$
\hat{x} = \sum_{i-1}^{5} a_i \boldsymbol{r}_i \tilde{p}
$$

that is

$$
\hat{x} = \sum_{i=1}^{5} a_i x_i.
$$

In a similar way, we obtain that

$$\hat{y} = \sum_{i=1}^{5} b_i y_i.$$

In addition, for pure rotation, the coefficients of these linear combinations satisfy seven functional constraints. These constraints, which are second-degree polynomials, are given in Appendix A. The coefficients of these polynomials can be found (by linear equations) using additional views.

Again, one may or may not actually test for these additional constraints. Assuming that different objects in memory differ by more than just a linear transformation, if the test is omitted, the probability of a false-positive misidentification is slightly increased.

As in our case of sharp boundaries, it is possible to use mixed x and y coordinates to reduce the number of basic views for general linear transformations (Section I-C-5). For example, one can use five basis vectors $(x_1, x_2, x_3, y_1, y_2)$ taken from these distinct views as the basis for the x and y coordinates in all other views.

2) Rigid Transformation and Scaling in 3-D Space: So far, we have shown that an object with smooth boundaries represented by the curvature scheme and undergoing a rotation in 3-D space can be represented as a linear combination of 2-D views. The method can again be easily extended to handle translation and scaling. The linear combination scheme for objects with smooth bounding contours is thus a direct extension of the scheme in Section I-C for objects with sharp boundaries. In both cases, object views are expressed as the linear combination of a small number of pictures. The scheme for objects with sharp boundaries can be viewed as a special case of the more general one when r, which the radius of curvature, vanishes. In practice, we found that it is also possible to use the scheme for sharp boundaries that uses a smaller number of views in each model for general objects, provided that r is not too large (and at the price of increasing the number of models).

3) Summary: In this section, we have shown that an object with smooth boundaries undergoing rigid transformations and scaling in 3-D space followed by an orthographic projection can be expressed (within the approximation of the curvature method) as the linear combination of six images of the object. Five images are used to represent rotations in 3-D space, and one additional image (or, alternatively, a constant vector) is required to represent translations. (In fact, although the coordinates are expressed in terms of five basis vectors, only three distinct views are needed for a general linear transformation.) The scaling does not require any additional image since it is represented by a scaling of the coefficients. This scheme was implemented and applied to images of 3-D objects.

Figs. 3 and 4 show the application of the linear combination (LC) method to complex objects with smooth bounding contours. Since the rotation was about the vertical axis, three 2-D views were used for each model. The models were created by taking three images and producing their edge maps (only edges that appeared in all three images were maintained). Since the rotation was around the vertical axis, a simple correspondence scheme was used to match points along the same scan line. The matching accuracy was sufficient for unambiguous discrimination in the presence of unavoidable noise, e.g., in image formation, edge detection, and correspondence. The figure shows a good agreement between the actual image and the appropriate linear combination. Although the objects are similar, they are easily discriminable by the LC method within the entire 60° rotation range.

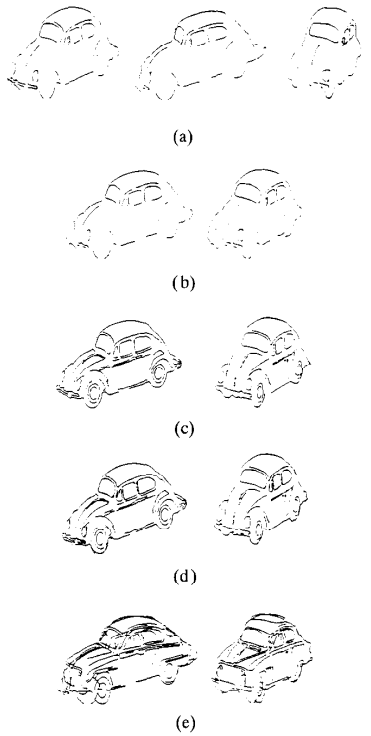

Fig. 3. (a) Three model pictures of a VW car for rotations around the vertical axis. The second and the third pictures were obtained from the first by rotations of $\pm 30°$ around the Y axis; (b) two linear combinations of the VW model. The x coefficients are $(0.556, 0.463, -0.018)$ and $(0.582, -0.065, 0.483)$, which correspond to a rotation of the first model picture by $\pm 15°$. These artificial images, which are created by linear combinations of the first three views rather than actual views; (c) real images of a VW car; (d) matching the linear combinations to the real images. Each contour image is a linear combination superimposed on the actual image. The agreement is good within the entire range of $\pm 30°$; (e) matching the VW model to pictures of the Saab car.

Finally, it is worth noting that the modeling of objects by linear combinations of stored pictures is not limited only to rigid objects. The method can also be used to deal with various types of nonrigid transformations, such as articulations and nonrigid stretching. For example, in the case of an articulated object, the object is composed of a number of rigid parts linked together by joints that constraint the relative movement of the parts. We saw that the x and y coordinates of a rigid part are constrained to a 4-D subspace. Two rigid parts reside within an 8-D subspace, but because of the constraints at the joints, they usually occupy a smaller subspace (e.g., 6-D for a planar joint).

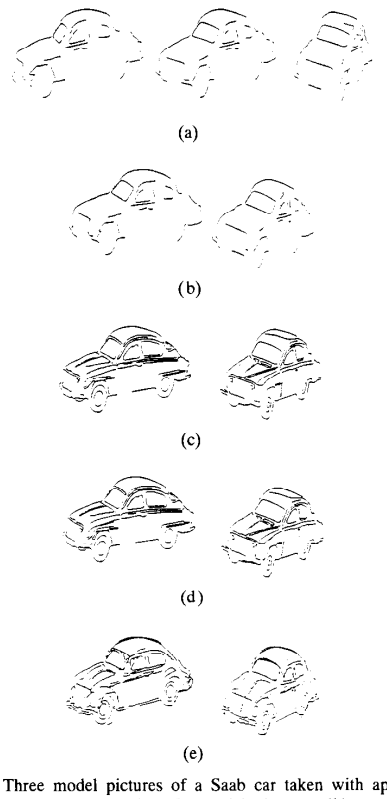

Fig. 4. (a) Three model pictures of a Saab car taken with approximately the same transformations as the VW model pictures; (b) two linear combinations of the Saab model. The x coefficients are $(0.601, 0.471, -0.072)$ and $(0.754, -0.129, 0.375)$, which correspond to a rotation of the first model picture by $\pm 15°$; (c) real images of a Saab car; (d) matching the linear combinations to the real images; (e) matching the Saab model to pictures of the VW car.

II. Determining the Alignment Coefficients

In the previous section, we have shown that the set of possible views of an object can often be expressed as the linear combination of a small number of views. In this section, we examine the problem of determining the transformation between a model and a viewed object. The model is given in this scheme as a set of k corresponding 2-D images $\{M_1, \cdots, M_k\}$. A viewed project P is an instance of this model if there exists a set of coefficients $\{a_1, \cdots, a_k\}$ (with a possible set of restrictions $F(a_1, \cdots, a_k) = 0$) such that

$$P = a_1 M_1 + \cdots + a_k M_k. \tag{3}$$

In practice, we may not obtain a strict equality. We will attempt to minimize, therefore, the difference between P and $a_1 M_1 + \cdots + a_k M_k$. The problem we face is how to determine the coefficients $\{a_1, \cdots, a_k\}$. In the following subsections, we will discuss three alternative methods for approaching this problem.

A. Minimal Alignment: Using a Small Number of Corresponding Features

The coefficients of the linear combination that align the model to the image can be determined using a small number of features, which are identified in both the model and the image to be recognized. This is similar to previous work in the framework of the alignment approach [8], [12], [16], [23]. It has been shown that three corresponding points or lines are usually sufficient to determine the transformation that aligns a 3-D model to a 2-D image [23], [12], [19], assuming the object can undergo only rigid transformations and uniform scaling. In previous methods, 3-D models of the object were stored. The corresponding features (lines and points) were then used to recover the 3-D transformation separating the viewed object from the stored model.

The coefficients of the linear combination required to align the model views with the image can be derived in principle, as in previous methods, by first recovering the 3-D transformations. They can also be derived directly, however, by simply solving a set of linear equations. This method requires k points to align a model of k pictures to a given image. Therefore, four points are required to determine the transformation for objects with sharp edges and six points for objects with smooth boundaries. In this way, we can deal with any transformation that can be approximated by linear combinations of pictures without recovering the 3-D transformations explicitly.

The coefficients of the linear combination are determined by solving the following equations. We assume that a small number of corresponding points (the "alignment points") have been identified in the image and the model. Let X be the matrix of the x coordinates of the alignment points in the model, that is, x_{ij} is the x coordinate of the jth point in the ith model-picture. $\boldsymbol{p}_x$ is the vector of x coordinates of the alignment points in the image, and $\boldsymbol{a}$ is the vector of unknown alignment parameters. The linear system to be solved is then $X\boldsymbol{a} = \boldsymbol{p}_x$. The alignment parameters are given by $\boldsymbol{a} = X^{-1}\boldsymbol{p}_x$ if an exact solution exists. We may use an overdetermined system (by using additional points), in which case, $\boldsymbol{a} = X^{+}\boldsymbol{p}_x$ (where X^{+} denotes the pseudo-inverse of X). The matrix X^{+} does not depend on the image and can be precomputed for the model. The recovery of the coefficients therefore requires only a multiplication of $\boldsymbol{p}_x$ by a known matrix. Similarly, we solve for $Y\boldsymbol{b} = \boldsymbol{p}_y$ to extract the alignment parameters $\boldsymbol{b}$ in the y direction from Y (the matrix of y coordinates in the model) and $\boldsymbol{p}_y$ (the corresponding y coordinates in the image). The stability of the computation in the face of noise will depend on the condition number of the matrices XX^T and YY^T. These matrices depend on the model images only, and this raises the possibility of selecting the model images in a manner that will increase the stability of the computation during matching.

It is also worth noting that the computation can proceed in a similar fashion in the basis of correspondence between straight-line segments rather than points. In this case, due to the "aperture problem" [18], only the perpendicular component (to the contour) of the displacement can be measured. This component can be used, however, in the equations above. In this case, each contour segment contributes a single equation (as opposed to a point correspondence, which gives two equations).

As a possible model of object recognition by a human being, one question that may arise in this context is whether the

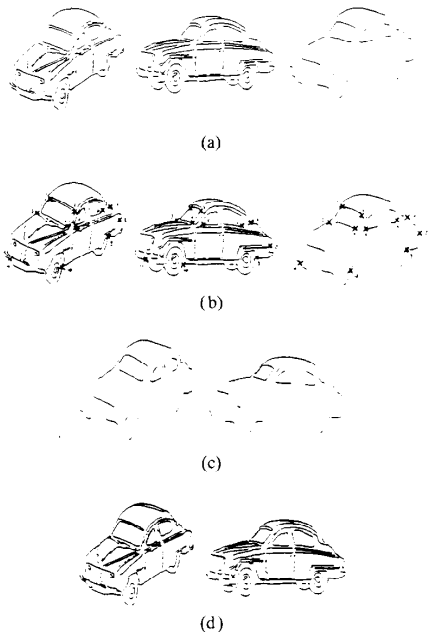

Fig. 5. Aligning a model to images using corresponding features: (a) Two images of a Saab car and one of the six model pictures; (b) corresponding points used to align the model to the images. The correspondence was determined using apparent motion, as explained in the text; (c) transformed model; (d) transformed model superimposed on the original images.

visual system can be expected to reliably extract a sufficient number of alignment features. Two comments are noteworthy. First, this difficulty is not specific to the linear combination scheme but applies to other alignment schemes as well. Second, although the task is not simple, the phenomenon of apparent motion suggests that mechanisms for establishing feature correspondence do, in fact, exist in the visual system.

It is interesting to note in this regard that the correspondence established during apparent motion appears to provide sufficient information for the purpose of recognition by linear combinations. For example, when the car pictures in Fig. 5(a) are shown in apparent motion, the points marked on the right in Fig. 5(b) appear perceptually to move and match the corresponding points marked on the pictures on the right. These points, with the perceptually established match, were used to align the model and images in Fig. 5, that is, the coordinates of these points were used in the equations above to recover the alignment coefficients. The model contained six pictures of a Saab car in order to cover all rigid transformations for an object with smooth boundaries. As can be seen, a close agreement was obtained between the image and the transformed model. (The model contained only a subset of the contours: the ones that were clearly visible in all of the different pictures.)

B. Searching for the Coefficients

An alternative method to determine the best linear combination is by a search in the space of possible coefficients. In this method, we choose some initial values for the set $\{a_1, \cdots, a_k\}$ of coefficients, we then apply a linear combination to the model using this set of coefficients. We repeat this process using a different set of coefficients and take the coefficient values that produced the best match of the model to the image.

The most problematic aspect of this method is that the domain of coefficients might be large; therefore, the search might be prohibitive. We can reduce the search space by first performing a rough alignment of the model to the image. The identification of general features in both the image and the model, such as a dominant orientation, the center of gravity, and a measurement of the overall size of the imaged object, can be used for compensating roughly for image rotation, translation, and scaling. Assuming that this process compensates for these transformations up to a bounded error and that the rotations in 3-D space covered by the model are also restricted, then we could restrict the search for the best coefficients to a limited domain. Moreover, the search can be guided by an optimization procedure. We can define an error measure (for instance, the area enclosed between the transformed model and the image) that must be minimized and use minimization techniques such as gradient descent to make the search more efficient. The preliminary stage of rough alignment may help preventing such methods from reaching a local minimum instead of the global one.

C. Linear Mappings

The linear combination scheme is based on the fact that a 3-D object can be modeled by the linear combination of a small number of pictures, that is, the set of possible views of an object is embedded in a linear space of a low dimensionality. We can use this property to construct a linear operator that maps each member of such a space to a predefined vector, which identifies the object. This method is different from the previous two in that we do not explicitly recover the coefficients $(a_1, \cdots, a_k)$ of the linear combination. Instead, we assume that a full correspondence has been established between the viewed object and the stored model. We then use a linear mapping to test whether the viewed object is a linear combination of the model views.

Suppose that a pattern P is represented by a vector $\boldsymbol{p}$ of its coordinates (e.g., $(x_1, y_1, x_2, y_2, \cdots, x_n y_n)$). Let P_1 and P_2 be two different patterns representing the same object. We can now construct a matrix L that maps both $\boldsymbol{p}_1$ and $\boldsymbol{p}_2$ to the same output vector $\boldsymbol{q}$, that is, $L\boldsymbol{p}_1 = L\boldsymbol{p}_2 = \boldsymbol{q}$. Any linear combination $a\boldsymbol{p}_1 + b\boldsymbol{p}_2$ will then be mapped to the same output vector $\boldsymbol{q}$ multiplied by the scalar $a + b$. We can choose, for example $\boldsymbol{q} = \boldsymbol{p}_1$, in which case any view of the object will be mapped by L to a selected "canonical view" of it.

We have seen above that different views of the same object can usually be expressed as linear combinations $\sum a_i \boldsymbol{p}_i$ of a small number of representative views P_i. If the mapping matrix L is constructed in such a manner that $L\boldsymbol{p}_i = \boldsymbol{q}$ for all the views P_i in the same model, then any combined view $\hat{\boldsymbol{p}} = \sum a_i \boldsymbol{p}_i$ will be mapped by L to the same $\boldsymbol{q}$ (up to a scale) since $L\hat{\boldsymbol{p}} = (\sum a_i)\boldsymbol{q}$.

L can be constructed as follows. Let $\{p_1, \cdots, p_k\}$ be k linearly independent vectors representing the model pictures (we can assume that they are all linearly independent since a picture that is not is obviously redundant). Let $\{p_{k+1}, \cdots, p_n\}$ be a set of vectors such that $\{p_1, \cdots, p_n\}$ are all linearly independent. We define the following matrices:

$$P = (p_1, \cdots, p_k, p_{k+1}, \cdots, p_n)$$
$$Q = (q, \cdots, q, p_{k+1}, \cdots, p_n)$$

We require that

$$LP = Q.$$

Therefore

$$L = QP^{-1}.$$

Note that since P is composed of n linearly independent vectors, the inverse matrix P^{-1} exists; therefore, L can always be constructed.

By this definition, we obtain a matrix L that maps any linear combination of the set of vectors $\{p_1, \cdots, p_k\}$ to a scaled pattern αq. Furthermore, it maps any vector orthogonal to $\{p_1, \cdots, p_k\}$ to itself. Therefore, if $\hat{p}$ is a linear combination of $\{p_1, \cdots, p_k\}$ with an additional orthogonal noise component, it would be mapped by L to q combined with the same amount of noise.

In constructing the matrix L, one may use more than just k vectors p_i, particularly if the input data is noisy. In this case, a problem arises of estimating the best k dimensional linear subspace spanned by a larger collection of vectors. This problem is treated in Appendix B.

In our implementation, we have used $Lp_i = 0$ for all the view vectors p_i of a given object. The reason is that if a new view of the object $\hat{p}$ is given by $\sum a_i p_i$ with $\sum a_i = 0$, then $L\hat{p} = 0$. This means that the linear mapping L may send a legal view to the zero vector, and it is therefore convenient to choose the zero vector as the common output for all the object's views. If it is desirable to obtain at the output level a canonical view of the object such as p_1 rather than the zero vector, then one can use as the final output the vector $p_1 - L\hat{p}$.

The decision regarding whether or not $\hat{p}$ is a view of the object represented by L can be based on comparing $\|L\hat{p}\|$ with $\|\hat{p}\|$. If $\hat{p}$ is indeed a view of the object, then this ratio will be small (exactly 0 in the noise-free condition). If the view is "pure noise" (in the space orthogonal to the span of $(p_1, \cdots, p_k)$), then this ratio will be equal to 1.

Fig. 6 shows the application of the linear mapping to two models of simple geometrical structures: a cube (a) and a pyramid (b). For each model, we have constructed a matrix that maps any linear combination of the model pictures to the first model picture that serves as its 'canonical view.' Consider the cube images in Fig. 6(a) first. The left column depicts two different views of the cube. Applying the cube matrix to these views yields in both cases the canonical view, as shown in the middle column. When the input to the cube matrix was a pyramid rather than a cube, the output was different from the canonical view (right column). In this manner, different views of the cubes can be identified by comparing the output to the canonical cube. Fig. 6(b) shows similar results obtained for the pyramid.

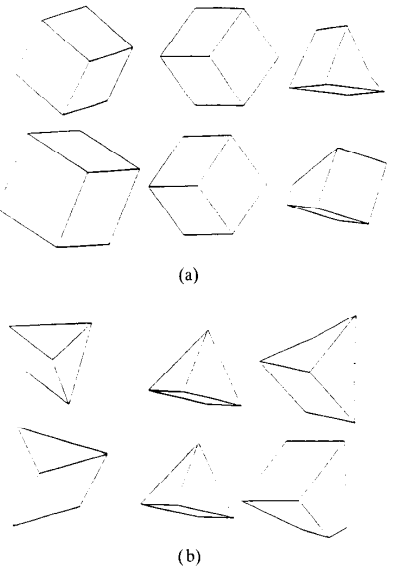

Fig. 6. (a) Applying cube and pyramid matrices to the cubes of Fig. 2; (b) applying pyramid and cube matrices to the pyramids of Fig. 2. Left column of pictures—the input images. Middle column—the result of applying the appropriate matrix to the images (these results are identical to the first model pictures, which serve as canonical views). Right column—the result of applying the wrong matrix to the images (these results are not similar to the canonical views).

III. General Discussion

We have proposed above a method for recognizing 3-D objects from 2-D images. In this method, an object-model is represented by the linear combinations of several 2-D views of the object. It was shown that for objects with sharp edges as well as with smooth bounding contours, the set of possible images of a given object is embedded in a linear space spanned by a small number of views. For objects with sharp edges, the linear combination representation is exact. For objects with smooth boundaries, it is an approximation that often holds over a wide range of viewing angles. Rigid transformations (with or without scaling) can be distinguished from more general linear transformations of the object by testing certain constraints placed on the coefficients of the linear combinations.

We have proposed three alternative methods for determining the transformation that matches a model to a given image. The first method uses a small set of corresponding features identified in both the model and the image. Alternatively, the coefficients can be determined using a search. The third method uses a linear mapping as the main step in a scheme that maps the different views of the same object into a common representation.

The development of the scheme so far has been primarily theoretical, and initial testing on a small number of objects shows good results. Future work should include more exten-

sive testing using natural objects, as well as the advancement of the theoretical issues discussed below.

In the concluding section, we discuss three issues. First, we place the current scheme within the framework of alignment methods in general. Second, we discuss possible extensions. Finally, we list a number of general conclusions that emerge from this study.

A. Classes of Alignment Schemes

The schemes discussed in this paper fall into the general class of alignment recognition methods. Other alignment schemes have been proposed by [2], [5], [7], [8], [10], [16], and [20]. In an alignment scheme, we seek a transformation T_α out of a set of allowed transformations and a model M from a given set of models that minimizes a distance measure $d(M, T_\alpha, p)$ (where p is the image of the object). T_α is called the alignment transformation because it is supposed to bring the model M and the viewed object p into an optimal agreement.

The distance measure d typically contains two contributions:

$$d(M, T_\alpha, p) = d_1(T_\alpha, M, p) + d_2(T_\alpha).$$

The first term $d_1(T_\alpha, M, p)$ measure the residual distance between the picture p and the transformed model $T_\alpha M$ following the alignment, and $d_2(T_\alpha)$ penalizes for the transformation T_α that was required to bring M into a close agreement with p. For example, it may be possible to bring M into a close agreement with p by stretching it considerably. In this case $d_1(T_\alpha, M, p)$ will be small, but if large stretches of the object are unlikely, $d_2(T_\alpha)$ will be large. We will see below that different classes of alignment schemes differ in the relative emphasis they place on d_1 and d_2.

Alignment approaches can be subdivided according to the method used for determining the aligning transformation T_α. The main approaches used in the past can be summarized by the following three categories.

1) Minimal Alignment: In this approach, T_α is determined by a small number of corresponding features in the model and the image. Methods using this approach assume that the set of possible transformations is restricted (usually to rigid 3-D transformations with possible scaling or a Lie transformation group [6]) so that the correct transformation can be recovered using a small number of constraints.

This approach has been used by Faugeras and Hebert [7], Fischler and Bolles [8], Huttenlocher and Ullman [12], Shoham and Ullman [19], Thompson and Mundy [20], and Ullman [23]. In these schemes, the term d_2 above is usually ignored since there is no reason to penalize for a rigid 3-D aligning transformation, and the match is therefore evaluated by d_1 only.

The correspondence between features may be guided in these schemes by the labeling of different types of features, such as cusps, inflections, blob-centers, etc. [12], [23] by using pairwise constraints between features [10] or by a more exhaustive search (as in [14], where possible transformations are precomputed and hashed).

Minimal alignment can be used in the context of the linear combination scheme discussed in this paper. This method was discussed in Section II-A. A small number of corresponding features is used to determine the coefficients of the linear combination. The linear combination is then computed, and the result is compared with the viewed image.

2) Full Alignment: In this approach, a full correspondence is established between the model and the image. This correspondence defines a distortion transformation that takes M into P. The set of transformations is not restricted in this approach to rigid transformations. Complex nonrigid distortions are included as well. In contrast with minimal alignment, in the distance measure d above, the first term $d_1(T_\alpha, M, P)$ does not play an important role since the full correspondence forces $T_\alpha M$ and P to be in close agreement. The match is therefore evaluated by the plausibility of the required transformation T_α. Our linear mapping scheme in Section II-C is a full alignment scheme. A full correspondence is established to produce a vector that the linear mapping can then act upon.

3) Alignment Search: In contrast with the previous approaches, this method does not use feature correspondence to recover the transformation. Instead, a search is conducted in the space of possible transformations. The set of possible transformations $\{T_\alpha\}$ is parametrized by a parameter vector α, and a search is performed in the parameter space to determine the best value of α. The deformable template method [25] is an example for this approach. Section II-B described the possibility of performing such a search in the linear combination approach to determine the value of the required coefficients.

B. Extensions

The LC recognition scheme is restricted in several ways. It will be of interest to extend it in the future in at least three directions: relaxing the constraints, dealing effectively with occlusions, and dealing with large libraries of objects. We limit the discussion below of brief comments on these three issues.

1) Relaxing the Constraints: The scheme as presented assumes rigid transformation and an orthogaphic projection. Under these conditions, all the views of a given object are embedded in a low-dimensional linear subspace of a much larger space. What happens if the projection is perspective rather than orthographic or if the transformations are not entirely rigid? The effect of perspecitivity appears to be quite limited. We have applied the LC scheme to objects with ratio of distance-to-camera to object-size down to 4:1 with only minor effects on the results (less than 3% deviation from the orthographic projection for rotations up to 45°).

As for nonrigid transformations, an interesting general extension to explore is where the set of views is no longer a linear subspace but still occupies a low-dimensional manifold within a much higher dimensional space. This manifold resembles locally a linear subspace, but it is no longer "globally straight." By analogy, one can visualize the simple linear combinations case in terms of a 3-D space, in which all the orthographic views of a rigid object are restricted to some 2-D plane. In the more general case, the plane will bend to become a curved 2-D manifold within the 3-D space.

This appears to be a general case of interest for recognition as well as for other learning tasks. For recognition to be feasible, the set of views $\{V\}$ corresponding to a given object cannot be arbitrary but must obey some constraints, e.g., in the form $F(V_i) = 0$. Under general conditions, these restrictions will define locally a manifold embedded in the larger space. Algorithms that can learn to classify efficiently sets that form low-dimensional manifolds embedded in high dimensional spaces will therefore be of general value.

2) Occlusion: In the linear combination scheme, we assumed that the same set of points is visible in the different views. What happens if some of the object's points are occluded by either self-occlusion or by other objects?

As we mentioned in Section I-C-5, self-occlusion is handled by representing an object not by a single model but by a number of models covering its different "aspects" [13] or "characteristic views" [24].

As for occlusion by other objects, this problem is handled in a different manner by the minimal alignment and the full alignment versions of the LC scheme. In the minimal alignment version, a small number of corresponding features are used to recover the coefficients of the linear combination. In this scheme, occlusion does not present a major special difficulty. After computing the linear combination, a good match will be obtained between the transformed model the visible part of the object, and recognition may proceed on the basis of this match. (Alignment search will behave in a similar manner.)

In the linear mapping version, an object's view is represented by a vector v_i of its coordinates. Due to occlusion, some of the coordinates will remain unknown. A way of evaluating the match in this case in an optimal manner is suggested in Appendix C.

Scene clutter also affects the computation by making the correspondence more difficult, that is, model features (points or lines) may be incorrectly matched with spurious data in the image. This effect of clutter on model-to-image correspondence is discussed e.g., in Grimson [9].

3) Multiple Models: We have considered above primarily the problem of matching a viewed object with a single model. If there are many candidate models, a question arises regarding the scaling of the computational load with the number of models.

In the LC scheme, the main problem is in the stage of performing the correspondence since the subsequent testing of a candidate model is relatively straightforward. The linear mapping scheme is particularly attractive in this regard: Once the correspondence is known, the testing of model requires only a multiplication of a matrix by a vector.

With respect to the correspondence stage, the question is how to perform correspondence efficiently with multiple models. This problem remains open for future study; we just comment here on a possible direction. The idea is to use prealignment to a prototype in the following manner. Suppose that $M_1, \cdots, M_k$ is a family of related models. A single model M will be used for representing this set for the purpose of alignment. The correspondence T_i between each M_i in the set and M is precomputed. Given an observed object P, a single correspondence T: $M \rightarrow P$ is computed. The individual transformations $M_i \rightarrow P$ are computed by the compositions $T \circ T_i$.

C. General Conclusions

In this section, we briefly summarize a number of general characteristics of the linear combinations scheme. In this scheme, as in some other alignment schemes, significant aspects of visual object recognition are more low-level in nature and more pictorial compared with structural description recognition approaches (e.g., [4]). The scheme uses directly 2-D views rather than an explicit 3-D model. The use of the 2-D views is different, however, from a simple associative memory [1], where new views are simply compared in parallel with all previously stored views. Rather than measuring the distance between the observed object and each of the stored views, a distance is measured from the observed object to the linear subspace (or a low-dimensional manifold) defined by previous views.

The linear combination scheme "reduces" the recognition problem in the sense to the problem of establishing a correspondence between the viewed object and candidate models. The method demonstrates that if a correspondence can be established, the remaining computation is relatively straightforward. Establishing a reliable correspondence between images is not an easy task, but it is a general task solved by the visual system (e.g., in motion measurement and stereoscopic vision), and related processes may also be involved in visual object recognition.

APPENDIX A

In Section I-D-1, we showed that the images of an object with smooth surfaces rotating in 3-D space can be represented as the linear combination of five views and mentioned that the coefficients for these linear combinations satisfy seven functional constraints. In this appendix, we list these constraints.

We use the same notation as in Section I-D-2. Let $R_1, \cdots, R_5$, $\hat{R}$, be 3×3 rotation matrices and $R'_1, \cdots, R'_5$, $\hat{R}'$ be the corresponding 2×5 matrices defined in Section I-D-2. Let $\boldsymbol{r}_1, \cdots, \boldsymbol{r}_5$, $\hat{\boldsymbol{r}}$ be the first row vectors and $\boldsymbol{s}_1, \cdots, \boldsymbol{s}_5$, $\hat{\boldsymbol{s}}$ the second row vectors of $R'_1, \cdots, R'_5$, $\hat{R}'$, respectively. In Section I-D-2, we showed that each of the two row vectors of $\hat{R}'$ is a linear combination of the corresponding row vectors of $R'_1, R'_2, \cdots, R'_5$, that is

$$\hat{\boldsymbol{r}} = \sum_{i=1}^{5} a_i \boldsymbol{r}_i$$

$$\hat{\boldsymbol{s}} = \sum_{i=1}^{5} b_i \boldsymbol{s}_i.$$

The functional constraints can be expressed as

$$\hat{r}_1^2 + \hat{r}_2^2 + \hat{r}_3^2 = 1$$

$$\hat{s}_1^2 + \hat{s}_2^2 + \hat{s}_3^2 = 1$$

$$\hat{r}_1 \hat{s}_1 + \hat{r}_2 \hat{s}_2 + \hat{r}_3 \hat{s}_3 = 0$$

$$\hat{r}_1 + \hat{r}_4 = \hat{s}_2 + \hat{s}_5$$
$$\hat{r}_2 + \hat{r}_5 = -(\hat{s}_1 + \hat{s}_4)$$
$$(\hat{r}_1 + \hat{r}_4)^2 + (\hat{r}_2 + \hat{r}_5)^2 = 1$$
$$\hat{r}_4\hat{s}_5 = \hat{s}_4\hat{r}_5.$$

(Constraints 1, 2, and 3, are immediate. Constraints 4, 5, 6, and 7 can be verified by expressing all the entries in terms of the rotation angles α, β, and γ.)

To express these constraints as a function of the coefficients, every occurrence of a term $\hat{r}_{ij}$ should be replaced by the appropriate linear combination as follows:

$$\hat{r}_j = \sum_{i=1}^{5} a_i (r_i)_j$$
$$\hat{s}_j = \sum_{i=1}^{5} b_i (s_i)_j.$$

In the case of a similarity transformations (i.e., with scale change), the first two constraints are substituted by

$$\hat{r}_1^2 + \hat{r}_2^2 + \hat{r}_3^2 = \hat{s}_1^2 + \hat{s}_2^2 + \hat{s}_3^2$$

and the sixth becomes

$$(\hat{r}_1 + \hat{r}_4)^2 + (\hat{r}_2 + \hat{r}_5)^2 = \hat{r}_1^2 + \hat{r}_2^2 + \hat{r}_3^2.$$

APPENDIX B

In this Appendix, we describe a method to find a space of a given dimension that lies as close as possible to a given set of points.

Let $\{\boldsymbol{p}_1, \boldsymbol{p}_2, \cdots, \boldsymbol{p}_m\}$ be a set of points in $\mathcal{R}^n$. We would like to find the $(n-k)$ dimensional space that lies as close as possible (in the least-square sense) to the points $\{\boldsymbol{p}_1, \boldsymbol{p}_2, \cdots, \boldsymbol{p}_m\}$. Let P be the $n \times m$ matrix given by $(\boldsymbol{p}_1, \boldsymbol{p}_2, \cdots, \boldsymbol{p}_m)$. Let $\{u_1, \cdots, u_n\}$ be a set of orthonormal vectors in $\mathcal{R}^n$, and define $\mathcal{U}_k = span\{u_{k+1}, u_n\}$. The sum of the distances (squared) of the points $\boldsymbol{p}_1, \boldsymbol{p}_2, \cdots, \boldsymbol{p}_m$ from $\mathcal{U}_k$ is given by

$$D^2(\mathcal{U}_k) = \sum_{i=1}^{k} \left\| P^t u_i \right\|^2.$$

(Since $\sum_{i=1}^{k} (\boldsymbol{p}_i u_i)^2$ is the squared distance of $\boldsymbol{p}_i$ from $\mathcal{U}_k$.)

Let $F = PP^t$. Then

$$D^2(\mathcal{U}_k) = \sum_{i=1}^{k} \left\| P^t u_i \right\|^2 = \sum_{i=1}^{k} (P^t u_i)^t (P^t u_i) = \sum_{i=1}^{k} u_i^t F u_i.$$

Any real matrix of the form XX^t is symmetric and nonnegative. Therefore, F has n eigenvectors and n real nonnegative eigenvalues. Assume that the $\{u_1, \cdots, u_n\}$ above are the eigenvectors of F with eigenvalues $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$, respectively, then $Fu_i = \lambda_i u_i$, and therefore

$$D^2(\mathcal{U}_k) = \sum_{i=1}^{k} \lambda_i.$$

Claim: Let $\{\lambda_1, \cdots, \lambda_k\}$ be the k smallest eigenvalues of F; then

$$\sum_{i=1}^{k} \lambda_i = \min_{\mathcal{V}_k} D^2(\mathcal{V}_k)$$

where the minimum is taken over all the linear subspaces of dimension $n-k$. Therefore, $span\{u_{k+1}, \cdots, u_n\}$ is the best $(n-k)$ dimensional space through $\boldsymbol{p}_1, \boldsymbol{p}_2, \cdots, \boldsymbol{p}_m$.

Proof: Let $\mathcal{V}_k$ be a linear subspace of dimension $(n-k)$. We must establish that

$$D^2(\mathcal{V}_k) \geq D^2(\mathcal{U}_k).$$

Let $\{v_1, \cdots, v_n\}$ be a set of orthonormal vectors in $\mathcal{R}^n$ such that $\mathcal{V}_k = span\{v_{k+1}, \cdots, v_n\}$. $V = (v_1, \cdots, v_n)$ and $U = (u_1, \cdots, u_n)$ are $n \times n$ orthonormal matrices. Let

$$R = U^t V.$$

Then

$$UR = V$$

that is

$$v_j = \sum_{i=1}^{n} r_{ij} u_i.$$

R is also orthonormal, therefore

$$\sum_{i=1}^{n} r_{ij}^2 = \sum_{j=1}^{n} r_{ij}^2 = 1.$$

Now

$$Fv_j = F\left(\sum_{i=1}^{n} r_{ij} u_i\right) = \sum_{i=1}^{n} r_{ij} \lambda_i u_i$$

and therefore

$$v_j^t F v_j = \left(\sum_{i=1}^{n} r_{ij} u_i\right)\left(\sum_{i=1}^{n} r_{ij} \lambda_i u_i\right).$$

Since $u_i^t u_j = \delta_{ij}$, we obtain that

$$v_j^t F v_j = \sum_{i=1}^{n} r_{ij}^2 \lambda_i.$$

Therefore

$$D^2(\mathcal{V}_k) = \sum_{j=1}^{k} v_j^t F v_j = \sum_{j=1}^{k} \sum_{i=1}^{n} r_{ij}^2 \lambda_i = \sum_{i=1}^{n} \left(\sum_{j=1}^{k} r_{ij}^2\right) \lambda_i.$$

Let

$$\alpha_i = \sum_{j=1}^{k} r_{ij}^2.$$

Then

$$D^2(\mathcal{V}_k) = \sum_{i=1}^{n} \alpha_i \lambda_i.$$

where $0 \leq \alpha_i \leq 1$ and $\sum_{i=1}^{n} \alpha_i = k$.

The claim we wish to establish is that the minimum is obtained when $\alpha_i = 1$ for $i = 1 \ldots k$, and $\alpha_i = 0$ for $i = k+1 \ldots n$. Assume that for $\mathcal{V}_k$, there exists $1 \leq m \leq k$ such that $\alpha_m < 1$, and $k+1 < l < n$ such that $\alpha_l > 0$. We can decrease α_l and increase α_m (by $\min(\alpha_l, 1-\alpha_m)$), and this cannot increase the value of $D^2(\mathcal{V}_k)$. By repeating this process, we will eventually reach the value of $D^2(\mathcal{U}_k)$. Since during this process the value cannot increase, we obtain that

$$D^2(\mathcal{U}_k) \leq D^2(\mathcal{V}_k)$$

and therefore

$$\sum_{i=1}^{k} \lambda_i = \min_{\mathcal{V}_k} D^2(\mathcal{V}_k).$$

APPENDIX C

In the linear mapping method, a matrix L was constructed that maps every legal view v of the object to a constant output vector. If the common output is chosen to be the zero vector, then $Lv = \mathbf{0}$ for any legal view of the object.

In this Appendix, we briefly consider the case where the object is only partially visible. We model this situation by assuming that we are given a partial vector $\boldsymbol{p}$. In this vector, the first k coordinates are unknown due to the occlusion, and only the last $n-k$ coordinates are observable. (A partial correspondence between the occluded object and the model is assumed to be known.)

In the vector $\boldsymbol{p}$, we take the first k coordinates to be zero. We try to construct from $\boldsymbol{p}$ a new vector $\boldsymbol{p}'$ by supplementing the missing coordinates so as to minimize $\|L\boldsymbol{p}'\|$. The relation between $\boldsymbol{p}$ and $\boldsymbol{p}'$ is

$$\boldsymbol{p}' = \boldsymbol{p} + \sum_{i=1}^{k} a_i \boldsymbol{u}_i$$

where the a_i are unknown constants, and the $\boldsymbol{u}_i$ are unit vectors along the first k coordinates.

In matrix notation, we seek to complement the occluded view by minimizing:

$$\min_{\boldsymbol{a}} \|L\boldsymbol{p} + LU\boldsymbol{a}\|$$

where the columns of the matrix U are the vectors $\boldsymbol{u}_i$, and $\boldsymbol{a}$ is the vector on the unknown a_i's.

The solution to this minimization problem is

$$\boldsymbol{a} = -[LU]^{+} L_p$$

(where H^{+} denotes the pseudo inverse of the matrix H). This means that the pseudo inverse $(LU)^{+}$ will have to be computed. The matrix L if fixed, but U depends on the points that are actually visible.

This optimal value of $\boldsymbol{a}$ can also be used to determine the output vector of the recognition process $L\boldsymbol{p}'$:

$$L\boldsymbol{p}' = \left(I - [LU][LU]^{+}\right) L\boldsymbol{p}.$$

$\boldsymbol{p}$ is then recognized as a legal view if this output is sufficiently close to zero.

ACKNOWLEDGMENT

We wish to thank E. Grimson, S. Edelman, T. Poggio, and A. Yuille for helpful comments; we also wish to thank T. Poggio for his suggestions regarding the use of two views and A. Yuille for Appendix B.

REFERENCES

[1] Y. S. Abu-Mostafa and D. Pslatis, "Optical neural computing," *Sci. Amer.*, vol. 256, pp. 66–73, 1987.
[2] R. Bajcsy and F. Solina, "Three dimensional object representation revisited," in *Proc. 1st ICCV Conf.* (London), 1987, pp. 231–240.
[3] R. Basri and S. Ullman, "The alignment of objects with smooth surfaces," in *Proc. 2nd ICCV Conf.* 1988, pp. 482–488.
[4] I. Biederman, "Human image understanding: Recent research and a theory," *Comput. Vision Graphics Image Processing*, vol. 32, pp. 29–73, 1985.
[5] C. H. Chien and J. K. Aggarwal, "Shape recognition from single silhouette," in *Proc. ICCV Conf.* (London), 1987, pp. 481–490.
[6] R. W. Brockett, "Least squares matching problems," in *Linear Algebra Appl.*, pp. 1–17, 1989.
[7] O. D. Faugeras and M. Hebert, "The representation, recognition and location of 3-D objects," *Int. J. Robotics Res.*, vol. 5, no. 3, pp. 27–52, 1986.
[8] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with application to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395.
[9] W. E. L. Grimson, "The combinatorics of heuristic search termination for object recognition in cluttered environments," O. Faugeras (Ed.), *Proceedings ECCV 1990* (Berlin).
[10] W. E. L. Grimson and T. Lozano-Perez, "Model-based recognition and localization from sparse data," *Int. J. Robotics Res.*, vol. 3, pp. 3–35, 1984.
[11] T. S. Huang and C. H. Lee, "Motion and structure from orthographic projections," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. 12, no. 5, pp. 536–540, 1989.
[12] D. P. Huttenlocher and S. Ullman, "Object recognition using alignment," in *Proc. ICCV Conf.* (London), 1987, pp. 102–111.
[13] J. J. Koenderink and A. J. Van Doorn, "The internal representation of solid shape with respect to vision," *Biol. Cybernetics*, vol. 32, pp. 211–216, 1989; in G. E. Hinton and J. A. Anderson, *Parallel Models of Associative Memory*. Hillsdale, NJ: Lawrence Erlbaum, pp. 105–143.
[14] Y. Lamdan, J. T. Schwartz, and H. Wolfson, "On recognition of 3-D objects from 2-D images," *Courant Inst. Math. Sci.*, Robotics Tech. Rep. 122, 1987.
[15] C. H. Longuet-Higgins, "A computer algorithm for reconstructing a scene from two projections," *Nature*, vol. 293, pp. 133–135, 1981.
[16] D. G. Lowe, *Perceptual Organization and Visual Recognition*. Boston: Kluwer Academic, 1985.
[17] D. Marr, "Analysis of occluding contour," *Phil. Trans. R. Soc. Lond. B*, vol. 275, pp. 483–524, 1977.
[18] D. Marr and S. Ullman, "Directional selectivity and its use in early visual processing," in *Proc. R. Soc. Lond. B*, vol. 211, pp. 151–180, 1981.
[19] D. Shoham and U. Ullman, "Aligning a model to an image using minimal information," in *Proc. 2nd ICCV Conf.*, 1988, pp. 259–263.
[20] D. W. Thompson and J. L. Mundy, "Three dimensional model matching from an unconstrained viewpoint," in *Proc. IEEE Int. Conf. Robotics Automat.*, (Rayleigh, NC), 1987, pp. 208–220.
[21] S. Ullman, *The Interpretation of Visual Motion*. Cambridge, MA: MIT Press, 1979.
[22] ——, "Recent computational studies in the interpretation of structure from motion," in A. Rosenfeld and J. Beck (Eds.), *Human and Machine Vision*. New York: Academic, 1983.
[23] ——, "Aligning pictorial descriptions: An approach to object recognition: *Cognition*, vol. 32, no. 3, pp. 193–254, 1989; A.I. Memo 931, Artificial Intell. Lab., Mass. Inst. Technol., 1986.
[24] H. Freeman and Chakravarty, "The use of characteristic views in the recognition of three-dimensional objects," in E. Gelsema and L. Kanal (Eds.), *Pattern Recognition in Practice*. Amsterdam: North-Holland, 1980.
[25] A. L. Yuille, D. S. Cohen, and P. W. Hllinan, "Feature extraction from faces using deformable templates," in *Proc. Comput. Vision Patt. Recogn.*, (San Diego), 1988, pp. 104–109.
[26] D. Zipser and R. A. Andersen, "A back-propagation programmed network that simulates response properties of a subset of posterior paricta neurons," *Nature*, vol. 331, pp. 679–684, 1988.

Shimon Ullman received the B.Sc. degree (summa cum laude) in mathematics, physics, and biology from the Hebrew University, Jerusalem, Israel, in 1973 and the Ph.D. degree in artificial intelligence from Massachusetts Institute of Technology in 1977.

He is currently a Professor of Brain and Cognitive Science at M.I.T. and is a member of the Artificial Intelligence Laboratory there. He is also the Samy and Ruth Cohn Professor of Computer Science at the Weizmann Institute of Science, Rehovot, Israel. His areas of research include computational vision and models of human vision.

Ronen Basri was born in Tel Aviv, Israel, in 1960. He received the B.Sc. degree (summa cum laude) in mathematics and computer science from Tel Aviv University, Israel, in 1985 and the Ph.D. degree from The Weizmann Institute of Science, Rehovot, Israel, in 1990.

He is currently a post-doctoral fellow in the Department of Brain and Cognitive Sciences at The Massachusetts Institute of Technology, Cambridge, MA.