A Paraperspective Factorization Method for Shape and Motion Recovery

Conrad J. Poelman and Takeo Kanade, *Fellow, IEEE*

Abstract—The factorization method, first developed by Tomasi and Kanade, recovers both the shape of an object and its motion from a sequence of images, using many images and tracking many feature points to obtain highly redundant feature position information. The method robustly processes the feature trajectory information using singular value decomposition (SVD), taking advantage of the linear algebraic properties of orthographic projection. However, an orthographic formulation limits the range of motions the method can accommodate. Paraperspective projection, first introduced by Ohta, is a projection model that closely approximates perspective projection by modeling several effects not modeled under orthographic projection, while retaining linear algebraic properties. Our paraperspective factorization method can be applied to a much wider range of motion scenarios, including image sequences containing motion toward the camera and aerial image sequences of terrain taken from a low-altitude airplane.

Index Terms—Motion analysis, shape recovery, factorization method, three-dimensional vision, image sequence analysis, singular value decomposition.

1 Introduction

Recovering the geometry of a scene and the motion of the camera from a stream of images is an important task in a variety of applications, including navigation, robotic manipulation, and aerial cartography. While this is possible in principle, traditional methods have failed to produce reliable results in many situations [2].

Tomasi and Kanade [13], [14] developed a robust and efficient method for accurately recovering the shape and motion of an object from a sequence of images, called the *factorization method*. It achieves its accuracy and robustness by applying a well-understood numerical computation, the singular value decomposition (SVD), to a large number of images and feature points, and by directly computing shape without computing the depth as an intermediate step. The method was tested on a variety of real and synthetic images, and was shown to perform well even for distant objects, where traditional triangulation-based approaches tend to perform poorly.

The Tomasi-Kanade factorization method, however, assumed an orthographic projection model. The applicability of the method is therefore limited to image sequences created from certain types of camera motions. The orthographic model contains no notion of the distance from the camera to the object. As a result, shape reconstruction from image sequences containing large translations toward or away from the camera often produces deformed object shapes, as the method tries to explain the size differences in the images by creating size differences in the object. The method also supplies no estimation of translation along the camera's optical axis, which limits its usefulness for certain tasks.

There exist several perspective approximations which capture more of the effects of perspective projection while remaining linear. Scaled orthographic projection, sometimes referred to as "weak perspective" [5], accounts for the scaling effect of an object as it moves towards and away from the camera. Paraperspective projection, first introduced by Ohta [6] and named by Aloimonos [1], accounts for the scaling effect as well as the different angle from which an object is viewed as it moves in a direction parallel to the image plane.

In this paper, we present a factorization method based on the paraperspective projection model. The paraperspective factorization method is still fast, and robust with respect to noise. It can be applied to a wider realm of situations than the original factorization method, such as sequences containing significant depth translation or containing objects close to the camera, and can be used in applications where it is important to recover the distance to the object in each image, such as navigation.

We begin by describing our camera and world reference frames and introduce the mathematical notation that we use. We review the original factorization method as defined in [13], presenting it in a slightly different manner in order to make its relation to the paraperspective method more apparent. We then present our paraperspective factorization method, followed by a description of a perspective refinement step. We conclude with the results of several experiments which demonstrate the practicality of our system.

2 Problem Description

In a shape-from-motion problem, we are given a sequence of F images taken from a camera that is moving relative to an object. Assume for the time being that we locate P prominent feature points in the first image, and track these

- *C.J. Poelman is with the Satellite Assessment Center (WSAT), USAF Phillips Laboratory, Albuquerque, NM 87117-5776. E-mail: poelmanc@plk.af.mil.*
- *T. Kanade is with the School of Computer Science, Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA 15213-3890. E-mail: tk@cs.cmu.edu.*

Manuscript received June 15, 1994; revised Jan. 10, 1996. Recommended for acceptance by S. Peleg.
For information on obtaining reprints of this article, please send e-mail to: transpami@computer.org, and reference IEEECS Log Number P97001.

points from each image to the next, recording the coordinates (u_{fp}, v_{fp}) of each point p in each image f. Each feature point p that we track corresponds to a single world point, located at position $\mathbf{s}_p$ in some fixed world coordinate system. Each image f was taken at some camera orientation, which we describe by the orthonormal unit vectors $\mathbf{i}_f$, $\mathbf{j}_f$, and $\mathbf{k}_f$, where $\mathbf{i}_f$ and $\mathbf{j}_f$ correspond to the x and y axes of the camera's image plane, and $\mathbf{k}_f$ points along the camera's line of sight. We describe the position of the camera in each frame f by the vector $\mathbf{t}_f$ indicating the camera's focal point. This formulation is illustrated in Fig. 1.

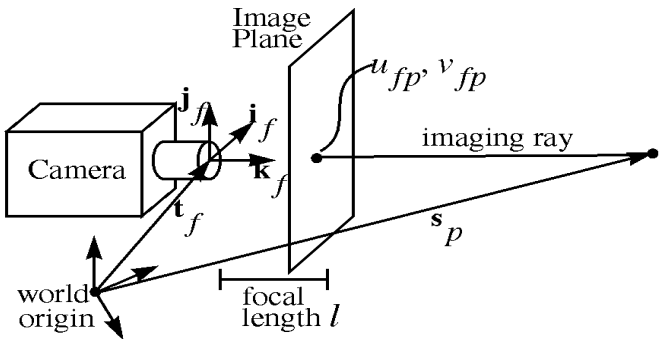


Fig. 1. Coordinate system.

The result of the feature tracker is a set of P feature point coordinates (u_{fp}, v_{fp}) for each of the F frames of the image sequence. From this information, our goal is to estimate the shape of the object as $\hat{\mathbf{s}}_p$ for each object point, and the motion of the camera as $\hat{\mathbf{i}}_f$, $\hat{\mathbf{j}}_f$, $\hat{\mathbf{k}}_f$, and $\hat{\mathbf{t}}_f$ for each frame in the sequence.

3 The Orthographic Factorization Method

This section presents a summary of the orthographic factorization method developed by Tomasi and Kanade. A more detailed description of the method can be found in [13].

3.1 Orthographic Projection

The orthographic projection model assumes that rays are projected from an object point along the direction parallel to the camera's optical axis, so that they strike the image plane orthogonally, as illustrated in Fig. 2. A point p whose location is $\mathbf{s}_p$ will be observed in frame f at image coordinates (u_{fp}, v_{fp}), where

$$u_{fp} = \mathbf{i}_f \cdot (\mathbf{s}_p - \mathbf{t}_f) \qquad v_{fp} = \mathbf{j}_f \cdot (\mathbf{s}_p - \mathbf{t}_f) \tag{1}$$

These equations can be rewritten as

$$u_{fp} = \mathbf{m}_f \cdot \mathbf{s}_p + x_f \qquad v_{fp} = \mathbf{n}_f \cdot \mathbf{s}_p + y_f \tag{2}$$

where

$$x_f = -(\mathbf{t}_f \cdot \mathbf{i}_f) \qquad y_f = -(\mathbf{t}_f \cdot \mathbf{j}_f) \tag{3}$$

$$\mathbf{m}_f = \mathbf{i}_f \qquad \mathbf{n}_f = \mathbf{j}_f \tag{4}$$

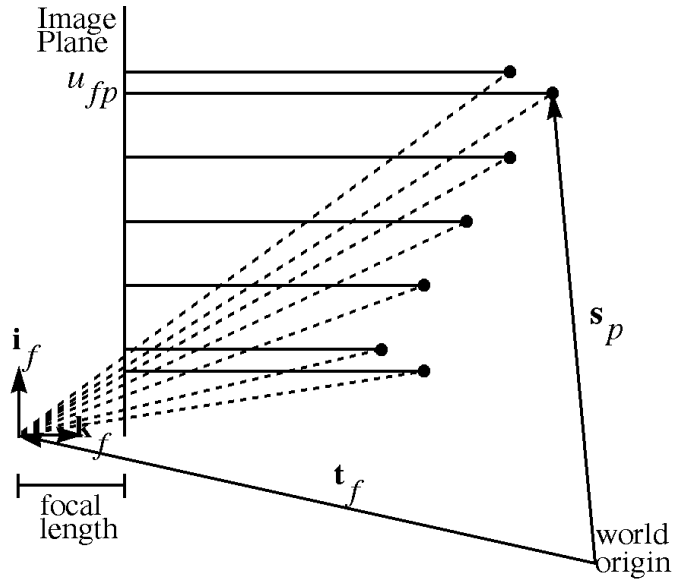


Fig. 2. Orthographic projection in two dimensions. Dotted lines indicate perspective projection.

3.2 Decomposition

All of the feature point coordinates (u_{fp}, v_{fp}) are entered in a $2F \times P$ *measurement matrix* W.

$$W = \begin{bmatrix} u_{11} & \cdots & u_{1P} \\ \cdots & \cdots & \cdots \\ u_{F1} & \cdots & u_{FP} \\ v_{11} & \cdots & v_{1P} \\ \cdots & \cdots & \cdots \\ v_{F1} & \cdots & v_{FP} \end{bmatrix} \tag{5}$$

Each column of the measurement matrix contains the observations for a single point, while each row contains the observed u-coordinates or v-coordinates for a single frame. Equation (2) for all points and frames can now be combined into the single matrix equation

$$W = MS + T[1 \ldots 1] \tag{6}$$

where M is the $2F \times 3$ motion matrix whose rows are the $\mathbf{m}_f$ and $\mathbf{n}_f$ vectors, S is the $3 \times P$ shape matrix whose columns are the $\mathbf{s}_p$ vectors, and T is the $2F \times 1$ translation vector whose elements are the x_f and y_f.

Up to this point, Tomasi and Kanade placed no restrictions on the location of the world origin, except that it be stationary with respect to the object. Without loss of generality, they position the world origin at the center of mass of the object, denoted by $\mathbf{c}$, so that

$$\mathbf{c} = \frac{1}{P}\sum_{p=1}^{P} \mathbf{s}_p = 0 \tag{7}$$

Because the sum of any row of S is zero, the sum of any row i of W is PT_i. This enables them to compute the ith element of the translation vector T directly from W, simply by averaging the ith row of the measurement matrix. The translation is the subtracted from W, leaving a "registered" measurement matrix $W^* = W - T[1 \ldots 1]$. Because W^* is the product of a $2F \times 3$ motion matrix M and a $3 \times P$ shape

matrix S, its rank is at most three. When noise is present in the input, the W^* will not be exactly of rank three, so the Tomasi-Kanade factorization method uses the SVD to find the best rank three approximation to W^*, factoring it into the product

$$W^* = \hat{M}\hat{S} \tag{8}$$

3.3 Normalization

The decomposition of (8) is only determined up to a linear transformation. Any non-singular 3×3 matrix A and its inverse could be inserted between $\hat{M}$ and $\hat{S}$, and their product would still equal W^*. Thus the actual motion and shape are given by

$$M = \hat{M}A \quad S = A^{-1}\hat{S} \tag{9}$$

with the appropriate 3×3 invertible matrix A selected. The correct A can be determined using the fact that the rows of the motion matrix M (which are the $\mathbf{m}_f$ and $\mathbf{n}_f$ vectors) represent the camera axes, and therefore they must be of a certain form. Since $\mathbf{i}_f$ and $\mathbf{j}_f$ are unit vectors, we see from (4) that

$$\left|\mathbf{m}_f\right|^2 = 1 \quad \left|\mathbf{n}_f\right|^2 = 1 \tag{10}$$

and because they are orthogonal,

$$\mathbf{m}_f \cdot \mathbf{n}_f = 0 \tag{11}$$

Equations (10) and (11) give us $3F$ equations which we call the *metric constraints*. Using these constraints, we solve for the 3×3 matrix A which, when multiplied by $\hat{M}$, produces the motion matrix M that best satisfies these constraints. Once the matrix A has been found, the shape and motion are computed from (9).

4 The Paraperspective Factorization Method

The Tomasi-Kanade factorization method was shown to be computationally inexpensive and highly accurate, but its use of an orthographic projection assumption limited the method's applicability. For example, the method does not produce accurate results when there is significant translation along the camera's optical axis, because orthography does not account for the fact that an object appears larger when it is closer to the camera. We must model this and other perspective effects in order to successfully recover shape and motion in a wider range of situations. We choose an approximation to perspective projection known as paraperspective projection, which was introduced by Ohta et al. [6] in order to solve a shape from texture problem. Although the paraperspective projection equations are more complex than those for orthography, their basic form is the same, enabling us develop a method analogous to that developed by Tomasi and Kanade.

4.1 Paraperspective Projection

Paraperspective projection closely approximates perspective projection by modeling both the scaling effect (closer objects appear larger than distant ones) and the position effect (objects in the periphery of the image are viewed from a different angle than those near the center of projection [1]) while retaining the linear properties of orthographic projection. Paraperspective projection is related to, but distinct from, the affine camera model, as described in Appendix A. The paraperspective projection of an object onto an image, illustrated in Fig. 3, involves two steps.

1) An object point is projected along the direction of the line connecting the focal point of the camera to the object's center of mass, onto a hypothetical image plane parallel to the real image plane and passing through the object's center of mass.
2) The point is then projected onto the real image plane using perspective projection. Because the hypothetical plane is parallel to the real image plane, this is equivalent to simply scaling the point coordinates by the ratio of the camera focal length and the distance between the two planes.[1]

In general, the projection of a point $\mathbf{p}$ along direction $\mathbf{r}$, onto the plane with normal $\mathbf{n}$ and distance from the origin d, is given by the equation

$$\mathbf{p}' = \mathbf{p} - \frac{\mathbf{p} \cdot \mathbf{n} - d}{\mathbf{r} \cdot \mathbf{n}} \mathbf{r} \tag{12}$$

In frame f, each object point $\mathbf{s}_p$ is projected along the direction $\mathbf{c} - \mathbf{t}_f$ (which is the direction from the camera's focal point to the object's center of mass) onto the plane defined by normal $\mathbf{k}_f$ and distance from the origin $\mathbf{c} \cdot \mathbf{k}_f$. The result $\mathbf{s}'_{fp}$ of this projection is

$$\mathbf{s}'_{fp} = \mathbf{s}_p - \frac{\left(\mathbf{s}_p \cdot \mathbf{k}_f\right) - \left(\mathbf{c} \cdot \mathbf{k}_f\right)}{\left(\mathbf{c} - \mathbf{t}_f\right) \cdot \mathbf{k}_f} \left(\mathbf{c} - \mathbf{t}_f\right) \tag{13}$$

The perspective projection of this point onto the image plane is given by subtracting $\mathbf{t}_f$ from $\mathbf{s}'_{fp}$ to give the position of the point in the camera's coordinate system, and then scaling the result by the ratio of the camera's focal length l to the depth to the object's center of mass z_f. Adjusting for the aspect ratio a and projection center $\left(o_x, o_y\right)$ yields the coordinates of the projection in the image plane,

$$\begin{aligned} u_{fp} &= \frac{l\mathbf{i}_f}{z_f}\left(\mathbf{s}'_{fp} - \mathbf{t}_f\right) + o_x \\ v_{fp} &= \frac{la\mathbf{j}_f}{z_f}\left(\mathbf{s}'_{fp} - \mathbf{t}_f\right) + o_y \\ \text{where } z_f &= \left(\mathbf{c} - \mathbf{t}_f\right) \cdot \mathbf{k}_f \end{aligned} \tag{14}$$

Substituting (13) into (14) and simplifying gives the general paraperspective equations for u_{fp} and v_{fp}

1. The scaled orthographic projection model (also known as "weak perspective") is similar to paraperspective projection, except that the direction of the initial projection in Step 1 is parallel to the camera's optical axis rather than parallel to the line connecting the object's center of mass to the camera's focal point. This model captures the scaling effect of perspective projection, but not the position effect, as explained in Appendix B.

$$u_{fp} = \frac{1}{z_f}\left\{\left[\mathbf{i}_f - \frac{\mathbf{i}_f \cdot (\mathbf{c}-\mathbf{t}_f)}{z_f}\mathbf{k}_f\right]\cdot(\mathbf{s}_p-\mathbf{c}) + (\mathbf{c}-\mathbf{t}_f)\cdot\mathbf{i}_f\right\} + o_x$$

$$v_{fp} = \frac{la}{z_f}\left\{\left[\mathbf{j}_f - \frac{\mathbf{j}_f \cdot (\mathbf{c}-\mathbf{t}_f)}{z_f}\mathbf{k}_f\right]\cdot(\mathbf{s}_p-\mathbf{c}) + (\mathbf{c}-\mathbf{t}_f)\cdot\mathbf{j}_f\right\} + o_y \quad (15)$$

We simplify these equations by assuming unit focal length, unit aspect ratio, and (0, 0) center of projection. This requires that the image coordinates (u_{fp}, v_{fp}) be adjusted to account for these camera parameters before commencing shape and motion recovery.

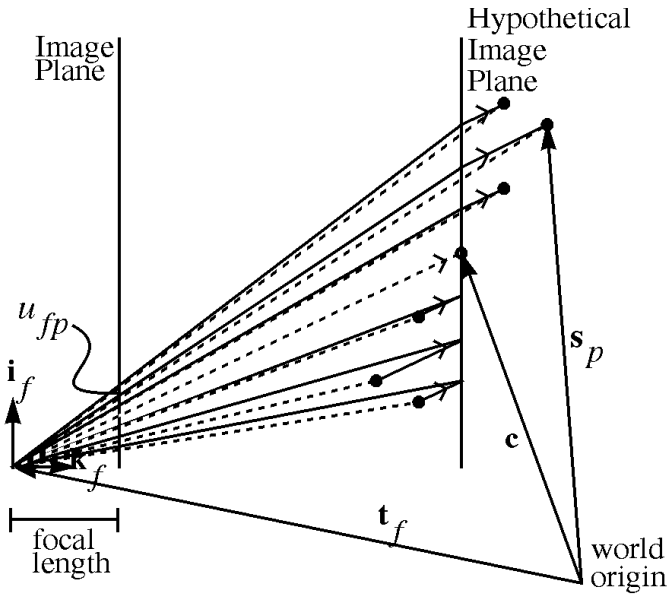


Fig. 3. Paraperspective projection in two dimensions. Dotted lines indicate perspective projection. → indicates parallel lines.

In [3] the factorization approach is extended to handle multiple objects moving separately, which requires each object to be projected based on its own mass center. However, since this paper addresses the single object case, we can further simplify our equations by placing the world origin at the object's center of mass so that by definition

$$\mathbf{c} = \frac{1}{P}\sum_{p=1}^{P}\mathbf{s}_p = 0 \quad (16)$$

This reduces (15) to

$$u_{fp} = \frac{1}{z_f}\left\{\left[\mathbf{i}_f + \frac{\mathbf{i}_f \cdot \mathbf{t}_f}{z_f}\mathbf{k}_f\right]\cdot\mathbf{s}_p - (\mathbf{t}_f\cdot\mathbf{i}_f)\right\}$$

$$v_{fp} = \frac{1}{z_f}\left\{\left[\mathbf{j}_f + \frac{\mathbf{j}_f \cdot \mathbf{t}_f}{z_f}\mathbf{k}_f\right]\cdot\mathbf{s}_p - (\mathbf{t}_f\cdot\mathbf{j}_f)\right\} \quad (17)$$

These equations can be rewritten as

$$u_{fp} = \mathbf{m}_f\cdot\mathbf{s}_p + x_f \qquad v_{fp} = \mathbf{n}_f\cdot\mathbf{s}_p + y_f \quad (18)$$

where

$$z_f = -\mathbf{t}_f\cdot\mathbf{k}_f \quad (19)$$

$$x_f = -\frac{\mathbf{t}_f\cdot\mathbf{i}_f}{z_f} \qquad y_f = -\frac{\mathbf{t}_f\cdot\mathbf{j}_f}{z_f} \quad (20)$$

$$\mathbf{m}_f = \frac{\mathbf{i}_f - x_f\mathbf{k}_f}{z_f} \qquad \mathbf{n}_f = \frac{\mathbf{j}_f - y_f\mathbf{k}_f}{z_f} \quad (21)$$

Notice that (18) has a form identical to its counterpart for orthographic projection, (2), although the corresponding definitions of x_f, y_f, $\mathbf{m}_f$, and $\mathbf{n}_f$ differ. This enables us to perform the basic decomposition of the matrix in the same manner that Tomasi and Kanade did for orthographic projection.

4.2 Paraperspective Decomposition

We can combine (18), for all points p from 1 to P, and all frames f from 1 to F, into the single matrix equation

$$\begin{bmatrix} u_{11} & \cdots & u_{1P} \\ \cdots & \cdots & \cdots \\ u_{F1} & \cdots & u_{FP} \\ v_{11} & \cdots & v_{1P} \\ \cdots & \cdots & \cdots \\ v_{F1} & \cdots & v_{FP} \end{bmatrix} = \begin{bmatrix} \mathbf{m}_1 \\ \cdots \\ \mathbf{m}_F \\ \mathbf{n}_1 \\ \cdots \\ \mathbf{n}_F \end{bmatrix}\begin{bmatrix}\mathbf{s}_1 \cdots \mathbf{s}_P\end{bmatrix} + \begin{bmatrix} x_1 \\ \cdots \\ x_F \\ y_1 \\ \cdots \\ y_F \end{bmatrix}\begin{bmatrix}1 \cdots 1\end{bmatrix} \quad (22)$$

or in short

$$W = MS + T\begin{bmatrix}1 \cdots 1\end{bmatrix} \quad (23)$$

where W is the $2F \times P$ measurement matrix, M is the $2F \times 3$ motion matrix, S is the $3 \times P$ shape matrix, and T is the $2F \times 1$ translation vector.

Using (16) and (18), we can write

$$\sum_{p=1}^{P} u_{fp} = \sum_{p=1}^{P}(\mathbf{m}_f\cdot\mathbf{s}_p + x_f) = \mathbf{m}_f\cdot\sum_{p=1}^{P}\mathbf{s}_p + Px_f = Px_f$$

$$\sum_{p=1}^{P} v_{fp} = \sum_{p=1}^{P}(\mathbf{n}_f\cdot\mathbf{s}_p + y_f) = \mathbf{n}_f\cdot\sum_{p=1}^{P}\mathbf{s}_p + Py_f = Py_f \quad (24)$$

Therefore we can compute x_f and y_f, which are the elements of the translation vector T, immediately from the image data as

$$x_f = \frac{1}{P}\sum_{p=1}^{P} u_{fp} \qquad y_f = \frac{1}{P}\sum_{p=1}^{P} v_{fp} \quad (25)$$

Once we know the translation vector T, we subtract it from W, giving the registered measurement matrix

$$W^* = W - T\begin{bmatrix}1 \ \ldots \ 1\end{bmatrix} = MS \quad (26)$$

Since W^* is the product of two matrices each of rank at most three, W^* has rank at most three, just as it did in the orthographic projection case. If there is noise present, the rank of W^* will not be exactly three, but by computing the SVD of

W^* and only retaining the largest three singular values, we can factor it into

$$W^* = \hat{M}\hat{S} \tag{27}$$

where $\hat{M}$ is a $2F \times 3$ matrix and $\hat{S}$ is a $3 \times P$ matrix. Using the SVD to perform this factorization guarantees that the product $\hat{M}\hat{S}$ is the best possible rank three approximation to W^*, in the sense that it minimizes the sum of squares difference between corresponding elements of W^* and $\hat{M}\hat{S}$.

4.3 Paraperspective Normalization

Just as in the orthographic case, the decomposition of W^* into the product of $\hat{M}$ and $\hat{S}$ by (27) is only determined up to a linear transformation matrix A. Again, we determine this matrix A by observing that the rows of the motion matrix M (the $\mathbf{m}_f$ and $\mathbf{n}_f$ vectors) must be of a certain form. Taking advantage of the fact that $\mathbf{i}_f$, $\mathbf{j}_f$, and $\mathbf{k}_f$ are unit vectors, from (21) we observe that

$$\left|\mathbf{m}_f\right|^2 = \frac{1+x_f^2}{z_f^2} \quad \left|\mathbf{n}_f\right|^2 = \frac{1+y_f^2}{z_f^2} \tag{28}$$

We know the values of x_f and y_f from our initial registration step, but we do not know the value of the depth z_f. Thus we cannot impose individual constraints on the magnitudes of $\mathbf{m}_f$ and $\mathbf{n}_f$ as was done in the orthographic factorization method. However, we can adopt the following constraint on the magnitudes of $\mathbf{m}_f$ and $\mathbf{n}_f$

$$\frac{\left|\mathbf{m}_f\right|^2}{1+x_f^2} = \frac{\left|\mathbf{n}_f\right|^2}{1+y_f^2} \quad \left(= \frac{1}{z_f^2}\right) \tag{29}$$

In the case of orthographic projection, one constraint on $\mathbf{m}_f$ and $\mathbf{n}_f$ was that they each have unit magnitude, as required by (10). In the above paraperspective case, we simply require that their magnitudes be in a certain ratio.

There is also a constraint on the angle relationship of $\mathbf{m}_f$ and $\mathbf{n}_f$. From (21), and the knowledge that $\mathbf{i}_f$, $\mathbf{j}_f$, and $\mathbf{k}_f$ are orthogonal unit vectors,

$$\mathbf{m}_f \cdot \mathbf{n}_f = \frac{\mathbf{i}_f - x_f\mathbf{k}_f}{z_f} \cdot \frac{\mathbf{j}_f - y_f\mathbf{k}_f}{z_f} = \frac{x_f y_f}{z_f^2} \tag{30}$$

The problem with this constraint is that, again, z_f is unknown. We could use either of the two values given in (29) for $1 / z_f^2$, but in the presence of noisy input data the two will not be exactly equal, so we use the average of the two quantities. We choose the arithmetic mean over the geometric mean or some other measure in order to keep the solution of these constraints linear. Thus our second constraint becomes

$$\mathbf{m}_f \cdot \mathbf{n}_f = x_f y_f \frac{1}{2}\left(\frac{\left|\mathbf{m}_f\right|^2}{1+x_f^2} + \frac{\left|\mathbf{n}_f\right|^2}{1+y_f^2}\right) \tag{31}$$

This is the paraperspective version of the orthographic constraint given by (11), which required that the dot product of $\mathbf{m}_f$ and $\mathbf{n}_f$ be zero.

Equations (29) and (31) are homogeneous constraints, which could be trivially satisfied by the solution $\forall f\, \mathbf{m}_f = \mathbf{n}_f = 0$, or $M = 0$. To avoid this solution, we impose the additional constraint

$$\left|\mathbf{m}_1\right| = 1 \tag{32}$$

This does not effect the final solution except by a scaling factor.

Equations (29), (31), and (32) give us $2F + 1$ equations, which are the paraperspective version of the *metric constraints*. We compute the 3×3 matrix A such that $M = \hat{M}A$ best satisfies these metric constraints in the least sum-of-squares error sense. This is a simple problem because the constraints are linear in the six unique elements of the symmetric 3×3 matrix $Q = A^T A$. We use the metric constraints to compute Q, compute its Jacobi Transformation $Q = L\Lambda L^T$, where Λ is the diagonal eigenvalue matrix, and as long as Q is positive definite, $A = \left(L\Lambda^{1/2}\right)^T$. A non-positive-definite Q indicates that unmodeled distortion has overwhelmed the third singular value of the measurement matrix, due possibly to noise, perspective effects, insufficient rotational motion, a planar object shape, or a combination of these effects.

4.4 Paraperspective Motion Recovery

Once the matrix A has been determined, we compute the shape matrix $S = A^{-1}\hat{S}$ and the motion matrix $M = \hat{M}A$. For each frame f, we now need to recover the camera orientation vectors $\hat{\mathbf{i}}_f$, $\hat{\mathbf{j}}_f$, and $\hat{\mathbf{k}}_f$ from the vectors $\mathbf{m}_f$ and $\mathbf{n}_f$, which are the rows of the matrix M. From (21) we see that

$$\hat{\mathbf{i}}_f = z_f\mathbf{m}_f + x_f\hat{\mathbf{k}}_f \quad \hat{\mathbf{j}}_f = z_f\mathbf{n}_f + y_f\hat{\mathbf{k}}_f \tag{33}$$

From this and the knowledge that $\hat{\mathbf{i}}_f$, $\hat{\mathbf{j}}_f$, and $\hat{\mathbf{k}}_f$ must be orthonormal, we determine that

$$\begin{aligned} &\hat{\mathbf{i}}_f \times \hat{\mathbf{j}}_f = \left(z_f\mathbf{m}_f + x_f\hat{\mathbf{k}}_f\right) \times \left(z_f\mathbf{n}_f + y_f\hat{\mathbf{k}}_f\right) = \hat{\mathbf{k}}_f \\ &\left|\hat{\mathbf{i}}_f\right| = \left|z_f\mathbf{m}_f + x_f\hat{\mathbf{k}}_f\right| = 1 \\ &\left|\hat{\mathbf{j}}_f\right| = \left|z_f\mathbf{n}_f + y_f\hat{\mathbf{k}}_f\right| = 1 \end{aligned} \tag{34}$$

Again, we do not know a value for z_f, but using the relations specified in (29) and the additional knowledge that $\left|\hat{\mathbf{k}}_f\right| = 1$, (34) can be reduced to

$$G_f\hat{\mathbf{k}}_f = H_f \tag{35}$$

where

$$G_f = \begin{bmatrix} \left(\tilde{\mathbf{m}}_f \times \tilde{\mathbf{n}}_f\right) \\ \tilde{\mathbf{m}}_f \\ \tilde{\mathbf{n}}_f \end{bmatrix} \quad H_f = \begin{bmatrix} 1 \\ -x_f \\ -y_f \end{bmatrix} \tag{36}$$

$$\tilde{\mathbf{m}}_f = \sqrt{1+x_f^2}\,\frac{\mathbf{m}_f}{|\mathbf{m}_f|} \quad \tilde{\mathbf{n}}_f = \sqrt{1+y_f^2}\,\frac{\mathbf{n}_f}{|\mathbf{n}_f|} \tag{37}$$

We compute $\hat{\mathbf{k}}_f$ simply as

$$\hat{\mathbf{k}}_f = G_f^{-1} H_f \tag{38}$$

and then compute

$$\hat{\mathbf{i}}_f = \tilde{\mathbf{n}}_f \times \hat{\mathbf{k}}_f \quad \hat{\mathbf{j}}_f = \hat{\mathbf{k}}_f \times \tilde{\mathbf{m}}_f \tag{39}$$

There is no guarantee that the $\hat{\mathbf{i}}_f$ and $\hat{\mathbf{j}}_f$ given by this equation will be orthonormal, because $\mathbf{m}_f$ and $\mathbf{n}_f$ may not have exactly satisfied the metric constraints. Therefore we actually use the orthonormals which are closest to the $\hat{\mathbf{i}}_f$ and $\hat{\mathbf{j}}_f$ vectors given by (39). We further refine these values using a non-linear optimization step to find the orthonormal $\hat{\mathbf{i}}_f$ and $\hat{\mathbf{j}}_f$, as well as depth z_f, which provide the best fit to (33). Due to the arbitrary world coordinate orientation, to obtain a unique solution we then rotate the computed shape and motion to align the world axes with the first frame's camera axes, so that $\hat{\mathbf{i}}_1 = [1\,0\,0]^T$ and $\hat{\mathbf{j}}_1 = [0\,1\,0]^T$.

All that remain to be computed are the translations for each frame. We calculate the depth z_f from (29). Since we know z_f, x_f, y_f, $\hat{\mathbf{i}}_f$, $\hat{\mathbf{j}}_f$, and $\hat{\mathbf{k}}_f$, we can calculate $\hat{\mathbf{t}}_f$ using (19) and (20).

5 Perspective Refinement of Paraperspective Solution

This section presents an iterative method used to recover the shape and motion using a perspective projection model. The object shape and camera motion provided by paraperspective factorization are refined alternately. This is a simpler and more efficient solution than the method of [11] in which all parameters are refined simultaneously, but this method may converge more slowly if the initial values are inaccurate. Although our algorithm was developed independently and handles the full three dimensional case, this method is quite similar to a two dimensional algorithm reported in [12].

5.1 Perspective Projection

Under perspective projection, often referred to as the pinhole camera model, object points are projected directly towards the focal point of the camera. An object point's image coordinates are determined by the position at which the line connecting the object point with the camera's focal point intersects the image plane, as illustrated in Fig. 4.

Simple geometry using similar triangles produces the perspective projection equations

$$u_{fp} = l\frac{\mathbf{i}_f \cdot (\mathbf{s}_p - \mathbf{t}_f)}{\mathbf{k}_f \cdot (\mathbf{s}_p - \mathbf{t}_f)} \quad v_{fp} = l\frac{\mathbf{j}_f \cdot (\mathbf{s}_p - \mathbf{t}_f)}{\mathbf{k}_f \cdot (\mathbf{s}_p - \mathbf{t}_f)} \tag{40}$$

Assuming unit focal length, we rewrite the equations in the form

$$u_{fp} = \frac{\mathbf{i}_f \cdot \mathbf{s}_p + x_f}{\mathbf{k}_f \cdot \mathbf{s}_p + z_f} \quad v_{fp} = \frac{\mathbf{j}_f \cdot \mathbf{s}_p + y_f}{\mathbf{k}_f \cdot \mathbf{s}_p + z_f} \tag{41}$$

where

$$x_f = -\mathbf{i}_f \cdot \mathbf{t}_f \quad y_f = -\mathbf{j}_f \cdot \mathbf{t}_f \quad z_f = -\mathbf{k}_f \cdot \mathbf{t}_f \tag{42}$$

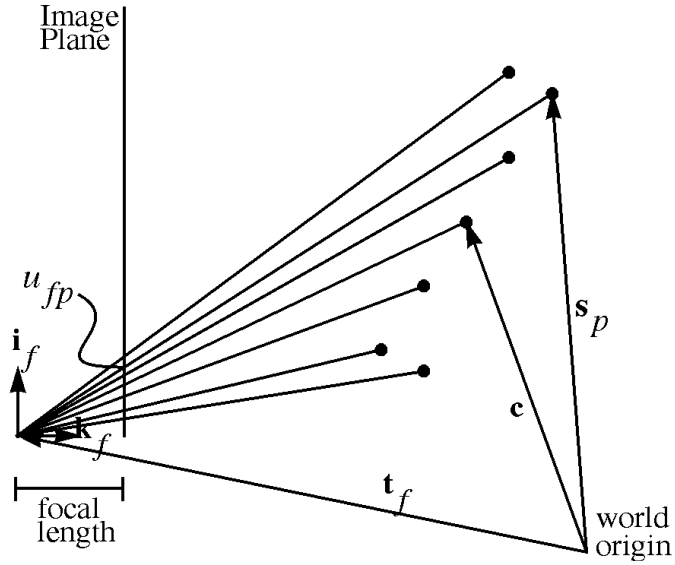


Fig. 4. Perspective projection in two dimensions.

5.2 Iterative Minimization Method

Equation (41) defines two equations relating the predicted and observed positions of each point in each frame, for a total of $2FP$ equations. We formulate the problem as a non-linear least squares problem in the motion and shape variables, in which we seek to minimize the error

$$\varepsilon = \sum_{f=1}^{F}\sum_{p=1}^{P}\left\{\left(u_{fp} - \frac{\mathbf{i}_f \cdot \mathbf{s}_p + x_f}{\mathbf{k}_f \cdot \mathbf{s}_p + z_f}\right)^2 + \left(v_{fp} - \frac{\mathbf{j}_f \cdot \mathbf{s}_p + y_f}{\mathbf{k}_f \cdot \mathbf{s}_p + z_f}\right)^2\right\} \tag{43}$$

In the above formulation, there appear to be 12 motion variables for each frame, since each image frame is defined by three orientation vectors and a translation vector. However, we can enforce the constraint that $\mathbf{i}_f$, $\mathbf{j}_f$, and $\mathbf{k}_f$ are orthogonal unit vectors by writing them as functions of three independent rotational parameters α_f, β_f, and γ_f.

$$\left[\mathbf{i}_f \mathbf{j}_f \mathbf{k}_f\right] = \begin{bmatrix} \cos\alpha_f \cos\beta_f & \left(\cos\alpha_f \sin\beta_f \sin\gamma_f - \sin\alpha_f \cos\gamma_f\right) & \left(\cos\alpha_f \sin\beta_f \cos\gamma_f + \sin\alpha_f \sin\gamma_f\right) \\ \sin\alpha_f \cos\beta_f & \left(\sin\alpha_f \sin\beta_f \sin\gamma_f + \cos\alpha_f \cos\gamma_f\right) & \left(\sin\alpha_f \sin\beta_f \cos\gamma_f - \cos\alpha_f \sin\gamma_f\right) \\ -\sin\beta_f & \cos\beta_f \sin\gamma_f & \cos\beta_f \cos\gamma_f \end{bmatrix} \tag{44}$$

This gives six motion parameters for each frame (x_f, y_f, z_f, α_f, β_f, and γ_f) and three shape parameters for each point $\left(s_p = [s_{p1} s_{p2} s_{p3}]\right)$ for a total of $6F + 3P$ variables.

We could apply any one of a number of non-linear techniques to minimize the error ε as a function of these $6F + 3P$ variables. Such methods begin with a set of initial variable values, and iteratively refine those values to reduce the er-

ror. Our method takes advantage of the particular structure of the equations by separately refining the shape and motion parameters. First the shape is held constant while solving for the motion parameters which minimize the error. Then the motion is held constant while solving for the shape parameters which minimize the error. This process is repeated until an iteration produces no significant reduction in the total error ε.

While holding the shape constant, the minimization with respect to the motion variables can be performed independently for each frame. This minimization requires solving an overconstrained system of six variables in P equations. Likewise while holding the motion constant, we can solve for the shape separately for each point by solving a system of $2F$ equations in three variables. This not only reduces the problem to manageable complexity, but as pointed out in [12], it lends itself well to parallel implementation.

We perform the individual minimizations, fitting six motion variables to P equations or fitting three shape variables to $2F$ equations, using the Levenberg-Marquardt method [8]. This method uses steepest descent when far from the minimum and varies continuously towards the inverse-Hessian method as the minimum is approached. Since we know the mathematical form of the expression of ε, the Hessian matrix is easily computed by taking derivatives of ε with respect to each variable.

A single step of the Levenberg-Marquardt method requires a single inversion of a 6×6 matrix when refining a single frame of motion, or a single inversion of a 3×3 matrix when refining the position of a single point. Generally about six steps were required for convergence of a single point or frame refinement, so a complete refinement step requires $6P$ inversions of 3×3 matrices and $6F$ inversions of 6×6 matrices.

In theory we do not actually need to vary all $6F + 3P$ variables, since the solution is only determined up to a scaling factor, the world origin is arbitrary, and the world coordinate orientation is arbitrary. We could choose to arbitrarily fix each of the first frame's rotation variables at zero degrees, and similarly fix some shape or translation parameters to reduce the problem to $6F + 3P - 7$ variables. However, it was experimentally confirmed that the algorithm converged significantly faster when all shape and motion parameters are all allowed to vary. The final shape and translation are then adjusted to place the origin at the object's center of mass and scale the solution so that the depth in the first frame is one. This shape and the final motion are then rotated so that $\hat{\mathbf{i}}_1 = \begin{bmatrix}1 & 0 & 0\end{bmatrix}^T$ and $\hat{\mathbf{j}}_1 = \begin{bmatrix}0 & 1 & 0\end{bmatrix}^T$, or equivalently, so that $\alpha_1 = \beta_1 = \Gamma_1 = 0$.

A common drawback of iterative methods on complex non-linear error surfaces is that the final result can be highly dependent on the initial value. Taylor, Kriegman, and Anandan [12] require some basic odometry measurements as might be produced by a navigation system to use as initial values for their motion parameters, and use the 2D shape of the object in the first image frame, assuming constant depth, as their initial shape. To avoid the requirement for odometry measurements, which will not be available in many situations, we use the paraperspective factorization method to supply initial values to the iterative perspective refinement process.

6 Comparison of Methods Using Synthetic Data

In this section we compare the performance of the paraperspective factorization method with the previous orthographic factorization method. The comparison also includes a factorization method based on scaled orthographic projection (also known as "weak perspective"), which models the scaling effect of perspective projection but not the position effect, in order to demonstrate the importance of modeling the position effect for objects at close range.[2] Our results show that the paraperspective factorization method is a vast improvement over the orthographic method, and underscore the importance of modeling both the scaling and position effects. We further examine the results of perspectively refining the paraperspective solution. This confirms that modeling of perspective distortion is important primarily for accurate shape recovery of objects at close range.

6.1 Data Generation

The synthetic feature point sequences used for comparison were created by moving a known "object"—a set of 3D points—through a known motion sequence. We tested three different object shapes, each containing approximately 60 points. Each test run consisted of 60 image frames of an object rotating through a total of 30 degrees each of roll, pitch, and yaw. The "object depth"—the distance from the camera's focal point to the front of the object—in the first frame was varied from three to 60 times the object size. In each sequence, the object translated across the field of view by a distance of one object size horizontally and vertically, and translated away from the camera by half its initial distance from the camera. For example, when the object's depth in the first frame was 3.0, its depth in the last frame was 4.5. Each "image" was created by perspectively projecting the 3D points onto the image plane, for each sequence choosing the largest focal length that would keep the object in the field of view throughout the sequence. The coordinates in the image plane were perturbed by adding Gaussian noise, to model tracking imprecision. The standard deviation of the noise was two pixels (assuming a 512×512 pixel image), which we consider to be a rather high noise level from our experience processing real image sequences. For each combination of object, depth, and noise, we performed three tests, using different random noise each time.

6.2 Error Measurement

We ran each of the three factorization methods on each synthetic sequence and measured the rotation error, shape error, X-Y offset error, and Z offset (depth) error. The rota-

2. The scaled orthographic factorization method is very similar to the paraperspective factorization method; the metric constraints for the method are $\left|\mathbf{m}_f\right|^2 = \left|\mathbf{n}_f\right|^2$, $\mathbf{m}_f \cdot \mathbf{n}_f = 0$, and $\left|\mathbf{m}_1\right| = 1$. See Appendix B.

tion error is the root-mean-square (RMS) of the size in radians of the angle by which the computed camera coordinate frame must be rotated about some axis to produce the known camera orientation. The shape error is the RMS error between the known and computed 3D point coordinates. Since the shape and translations are only determined up to scaling factor, we first scaled the computed shape by the factor which minimizes this RMS error. The term "offset" refers to the translational component of the motion as measured in the camera's coordinate frame rather than in world coordinates; the X offset is $\hat{\mathbf{t}}_f \cdot \hat{\mathbf{i}}_f$, the Y offset is $\hat{\mathbf{t}}_f \cdot \hat{\mathbf{j}}_f$, and the Z offset is $\hat{\mathbf{t}}_f \cdot \hat{\mathbf{k}}_f$. The X-Y offset error and Z offset error are the RMS error between the known and computed offset; like the shape error, we first scaled the computed offset by the scale factor that minimized the RMS error. Note that the orthographic factorization method supplies no estimation of translation along the camera's optical axis, so the Z offset error cannot be computed for that method.

6.3 Discussion of Results

Fig. 5 shows the average errors in the solutions computed by the various methods, as a functions of object depth in the first frame. We see that the paraperspective method performs significantly better than the orthographic factorization method regardless of depth, because orthography cannot model the scaling effect that occurs due to the motion along the camera's optical axis. The figure also shows that at close range, the paraperspective method performs substantially better than the scaled orthographic method (discussed in Appendix B) while the errors from the two methods are nearly the same when the object is distant. This confirms the importance of modeling the position effect when objects are near the camera. Perspective refinement of the paraperspective results only marginally improves the recovered camera motion, while it significantly improves the accuracy of the computed shape, even up to fairly distant ranges.

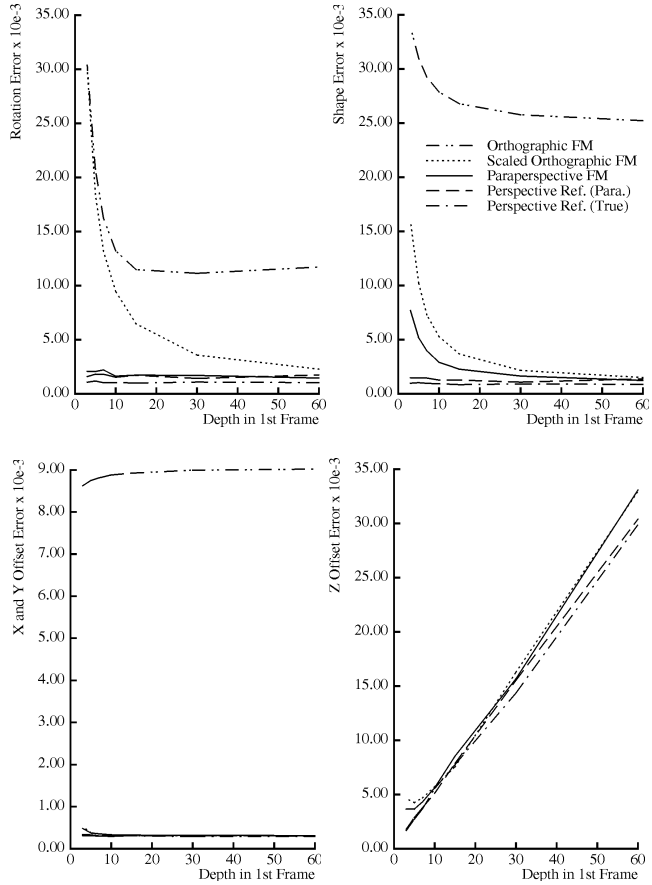


Fig. 5. Methods compared for a typical case. Noise standard deviation = two pixels.

We show the results of refining the known correct motion and shape only for comparison, as it indicates what is essentially the best one could hope to achieve using the least squares formulation without incorporating additional knowledge or constraints.

In other experiments in which the object was centered in the image and there was no translation across the field of view, the paraperspective method and the scaled orthographic method performed equally well, as we would expect since such image sequences contain no position effects. Similarly, we found that when the object remained centered in the image and there was no depth translation, the orthographic factorization method performed well, and the paraperspective factorization method provided no significant improvement since such sequences contain neither scaling effects nor position effects.

6.4 Analysis of Paraperspective Method Using Synthetic Data

Now that we have shown the advantages of the paraperspective factorization method over the previous method, we further analyze the performance of the paraperspective method to determine its behavior at various depths and its robustness with respect to noise. The synthetic sequences used in these experiments were created in the same manner as in the previous section, except that the standard deviation of the noise was varied from 0 to 4.0 pixels.

In Fig. 6, we see that at high depth values, the error in

the solution is roughly proportional to the level of noise in the input, while at low depths the error is inversely related to the depth. This occurs because at low depths, perspective distortion of the object's shape is the primary source of error in the computed results. At higher depths, perspective distortion of the object's shape is negligible, and noise becomes the dominant cause of error in the results. For example, at a noise level of one pixel, the rotation and XY-offset errors are nearly invariant to the depth once the object is farther from the camera than 10 times the object size. The shape results, however, appear sensitive to perspective distortion even at depths of 30 or 60 times the object size.

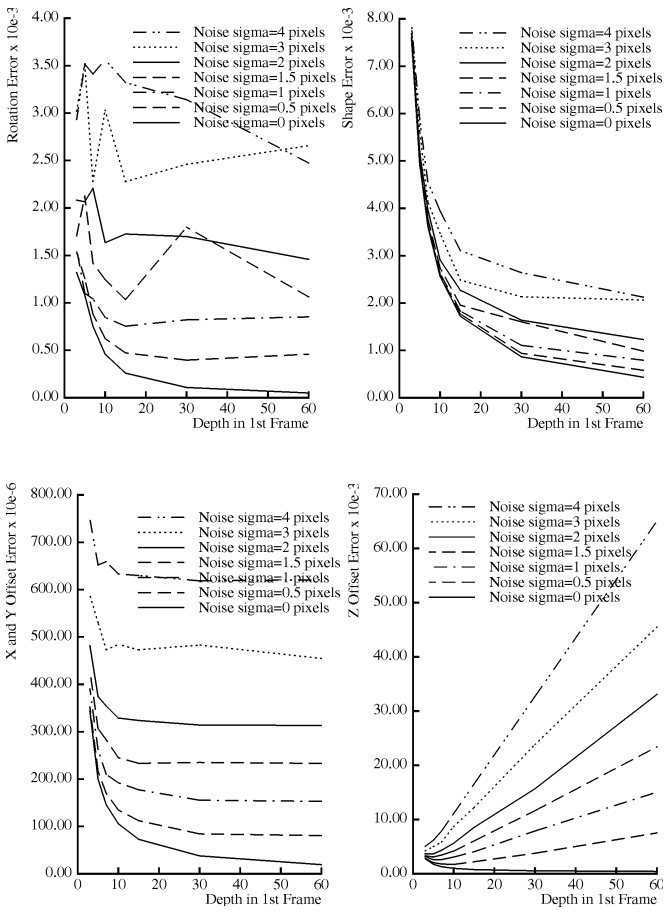


Fig. 6. Paraperspective shape and motion recovery by noise level.

7 Shape and Motion Recovery from Real Image Sequences

We tested the paraperspective factorization method on two real image sequences—a laboratory experiment in which a small model building was imaged, and an aerial sequence taken from a low-altitude plane using a hand-held video camera. Both sequences contain significant perspective effects, due to translations along the optical axis and across the field of view. We implemented a system to automatically identify and track features, based on [13] and [4]. This tracker computes the position of a square feature window by minimizing the sum of the squares of the intensity difference over the feature window from one image to the next.

7.1 Hotel Model Sequence

A hotel model was imaged by a camera mounted on a computer-controlled movable platform. The camera motion included substantial translation away from the camera and across the field of view (see Fig. 7). The feature tracker automatically identified and tracked 197 points throughout the sequence of 181 images.

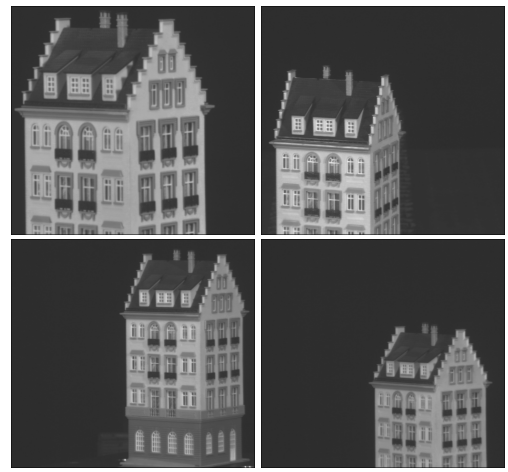


Fig. 7. Hotel model image sequence. (Top left) Frame 1, (top right) Frame 61, (bottom left) Frame 121, (bottom right) Frame 151.

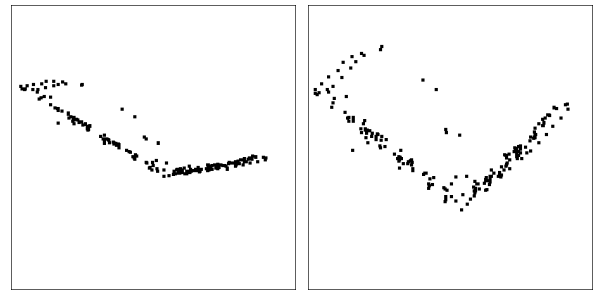


Fig. 8. Comparison of top views of orthographic (left) and paraperspective (right) shape results.

Both the paraperspective factorization method and the orthographic factorization method were tested with this sequence. The shape recovered by the orthographic factorization method was rather deformed (see Fig. 8) and the recovered motion incorrect, because the method could not

account for the scaling and position effects which are prominent in the sequence. The paraperspective factorization method, however, models these effects of perspective projection, and therefore produced an accurate shape and accurate motion.

Several features in the sequence were poorly tracked, and as a result their recovered 3D positions were incorrect. While they did not disrupt the overall solution greatly, we found that we could achieve improved results by automatically removing these features in the following manner. Using the recovered shape and motion, we computed the reconstructed measurement matrix W^{recon}, and then eliminated from those features for which the average error between the elements of W and W^{recon} was more than twice the average such error. We then ran the shape and motion recovery again, using only the remaining 179 features. Eliminating the poorly tracked features decreased errors in the recovered rotation about the camera's x-axis in each frame by an average of 0.5 degree, while the errors in the other rotation parameters were also slightly improved. The final rotation values are shown in Fig. 9, along with the values we measured using the camera platform. The computed rotation about the camera x-axis, y-axis, and z-axis was always within 0.29 degree, 1.78 degrees, and 0.45 degree of the measured rotation, respectively.

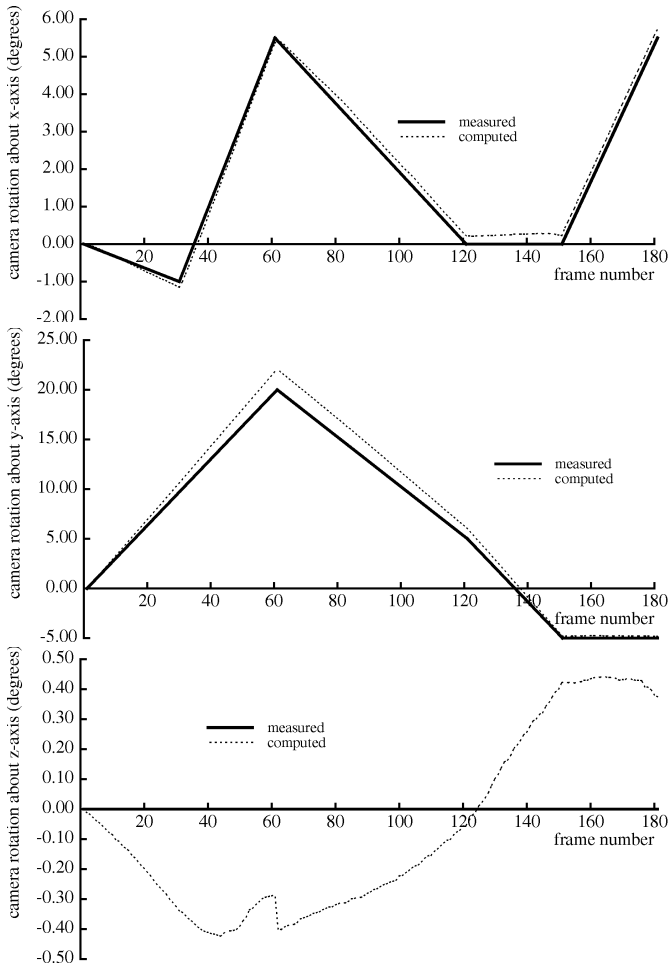


Fig. 9. Hotel model rotation results.

7.2 Aerial Image Sequence

An aerial image sequence was taken from a small airplane overflying a suburban Pittsburgh residential area adjacent to a steep, snowy valley, using a small hand-held video camera. The plane altered its altitude during the sequence and also varied its roll, pitch, and yaw slightly. Several images from the sequence are shown in Fig. 10.

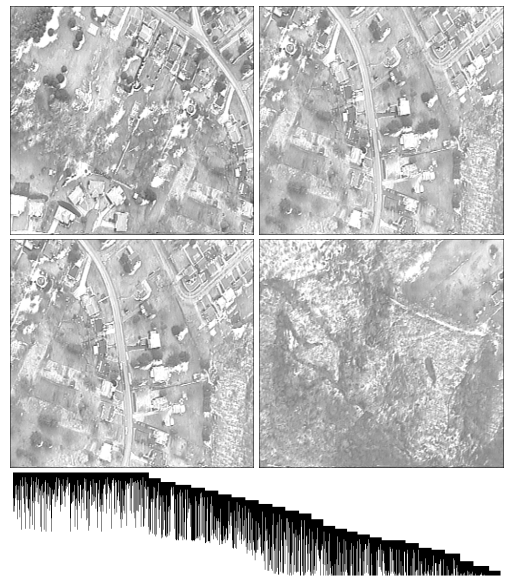


Fig. 10. Aerial image sequence. (Top left) Frame 1, (top right) Frame 35, (middle left) Frame 70, (middle right) Frame 108, (bottom) fill pattern indicating points visible in each frame.

Due to the bumpy motion of the plane and the instability of the hand-held camera, features often moved by as much as 30 pixels from one image to the next. The original feature tracker could not track motions of more than approximately three pixels, so we implemented a coarse-to-fine tracker. The tracker first estimated the translation using low resolution images, and then refined that value using the same methods as the initial tracker.

The sequence covered a long sweep of terrain, so none of the features were visible throughout the entire sequence. As some features left the field of view, new features were automatically detected and added to the set of features being tracked. A vertical bar in the fill pattern (shown in Fig. 10) indicates the range of frames through which a feature was successfully tracked. Each observed data measurement was assigned a confidence value based on the gradient of the feature and the tracking residue. A total of 1,026 points were tracked in the 108 image sequence, with each point being visible for an average of 30 frames of the sequence.

Because not all entries of the $2F \times P$ measurement matrix W were known, it was not possible to compute its SVD. Instead, a confidence-weighted decomposition step, described in [7], was used to decompose the measurement matrix W into $\hat{S}$, $\hat{M}$, and T. Paraperspective factorization was then used to recover the final shape of the terrain and motion of the airplane. Two views of the reconstructed terrain map are shown in Fig. 11. While no ground-truth was available for the shape or the motion, we observed

that the terrain was qualitatively correct, capturing the flat residential area and the steep hillside as well, and that the recovered positions of features on buildings were elevated from the surrounding terrain.

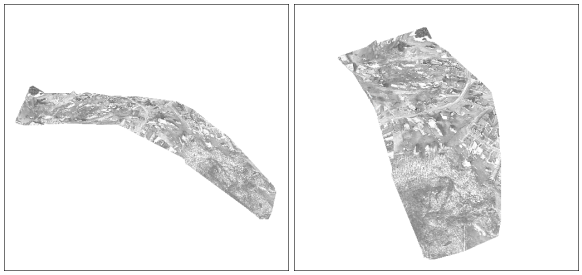


Fig. 11. Two views of reconstructed terrain.

8 Conclusions

The principle that the measurement matrix has rank three, as put forth by Tomasi and Kanade in [14], was dependent on the use of an orthographic projection model. We have shown in this paper that this important result also holds for the case of paraperspective projection, which closely approximates perspective projection. We have devised a paraperspective factorization method based on this model, which uses different metric constraints and motion recovery techniques, but retains many of the features of the original factorization method.

In image sequences in which the object being viewed translates significantly toward or away from the camera or across the camera's field of view, the paraperspective factorization method performs significantly better than the orthographic method. The paraperspective factorization method also computes the distance from the camera to the object in each image and can accommodate missing or uncertain tracking data, which enables its use in a variety of applications. Furthermore, even at close range when perspective distortion is significant, paraperspective factorization produces accurate motion results, and errors in the shape result due to perspective distortion can be largely reduced using a simple iterative perspective refinement step.

The C implementation of the paraperspective factorization method required about 20-24 seconds to solve a system of 60 frames and 60 points on a Sun 4/65, with most of this time spent computing the singular value decomposition of the measurement matrix. Perspective refinement of the solution required longer, but significant improvement of the shape results was achieved in a comparable amount of time.

Appendix A
Relation of Paraperspective to Affine Models

In an unrestricted affine camera, the image coordinates are given by

$$\begin{bmatrix} u_{fp} \\ v_{fp} \end{bmatrix} = \begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \end{bmatrix} \begin{bmatrix} s_{p1} \\ s_{p2} \\ s_{p3} \end{bmatrix} + \begin{bmatrix} x_f \\ y_f \end{bmatrix} \tag{45}$$

where the m_{ij} are free to take on any values. In motion applications, this matrix is commonly decomposed into a scaling factor, a 2×2 camera calibration matrix, and a 2×3 rotation matrix. The calibration matrix is considered to remain constant throughout the sequence, while the rotation matrix and scaling factor are allowed to vary with each image.

$$\begin{bmatrix} u_{fp} \\ v_{fp} \end{bmatrix} = \frac{1}{z_f} \begin{bmatrix} 1 & 0 \\ s & a \end{bmatrix} \begin{bmatrix} i_{f1} & i_{f2} & i_{f3} \\ j_{f1} & j_{f2} & j_{f3} \end{bmatrix} \begin{bmatrix} s_{p1} \\ s_{p2} \\ s_{p3} \end{bmatrix} + \begin{bmatrix} x_f \\ y_f \end{bmatrix} \tag{46}$$

These parameters have the following physical interpretations: the $\mathbf{i}_f$ and $\mathbf{j}_f$ vectors represent the camera rotation in each frame, x_f, y_f, and z_f represent the object translation (z_f is scaled by the camera focal length, x_f and y_f are offset by the image center), a is the camera aspect ratio, and s is a skew parameter. The skew parameter is non-zero only if the projection rays, while still parallel, do not strike the image plane orthogonally.

The paraperspective projection equations can be rewritten, retaining the camera parameters, as

$$\begin{bmatrix} u_{fp} \\ v_{fp} \end{bmatrix} = \frac{1}{z_f} \begin{bmatrix} 1 & 0 & \frac{(o_x - x_f)}{l} \\ 0 & a & \frac{(o_y - y_f)}{l} \end{bmatrix} \begin{bmatrix} i_{f1} & i_{f2} & i_{f3} \\ j_{f1} & j_{f2} & j_{f3} \\ k_{f1} & k_{f2} & k_{f3} \end{bmatrix} \begin{bmatrix} s_{p1} \\ s_{p2} \\ s_{p3} \end{bmatrix} + \begin{bmatrix} x_f \\ y_f \end{bmatrix} \tag{47}$$

This can be reduced by Householder transformation, in the manner shown by [9], to a form identical to that of the fixed-intrinsic affine camera,

$$\begin{bmatrix} u_{fp} \\ v_{fp} \end{bmatrix} = \frac{\sqrt{1+b_f^2}}{z_f} \begin{bmatrix} 1 & 0 \\ a\frac{b_f c_f}{1+b_f^2} & a\frac{\sqrt{1+b_f^2+c_f^2}}{1+b_f^2} \end{bmatrix} \begin{bmatrix} i'_{f1} & i'_{f2} & i'_{f3} \\ j'_{f1} & j'_{f2} & j'_{f3} \end{bmatrix} \begin{bmatrix} s_{p1} \\ s_{p2} \\ s_{p3} \end{bmatrix} + \begin{bmatrix} x_f \\ y_f \end{bmatrix} \tag{48}$$

where $b_f = \frac{o_x - x_f}{l}$, $c_f = \frac{o_y - y_f}{la}$, and $\mathbf{i}'_f$ and $\mathbf{j}'_f$ are orthonormal unit vectors.

Both the fixed-intrinsic-parameter affine camera and the paraperspective models are specializations of the unrestricted affine camera model, yet they are different from each other. The former projects all rays onto the image plane at the same angle throughout the sequence, which can be an accurate model if the object does not translate in the image or if the angle is non-perpendicular due to a lens misalignment. Under paraperspective, the direction of image projection and the axis scaling parameters change with each image in a physically realistic manner tied to the translation of the object in the image relative to the image center. This allows it to accurately model the position effect, unlike the fixed-intrinsic affine camera, while enforcing the constraint that the camera calibration parameters remain constant, unlike the unrestricted affine camera.

APPENDIX B
SCALED ORTHOGRAPHIC FACTORIZATION

Scaled orthographic projection, also known as "weak perspective" [5], is a closer approximation to perspective projection than orthographic projection, yet not as accurate as paraperspective projection. It models the scaling effect of perspective projection, but not the position effect. The scaled orthographic factorization method can be used when the object remains centered in the image, or when the distance to the object is large relative to the size of the object.

B.1 Scaled Orthographic Projection

Under scaled orthographic projection, object points are orthographically projected onto a hypothetical image plane parallel to the actual image plane but passing through the object's center of mass **c**. This image is then projected onto the image plane using perspective projection (see Fig. 12).

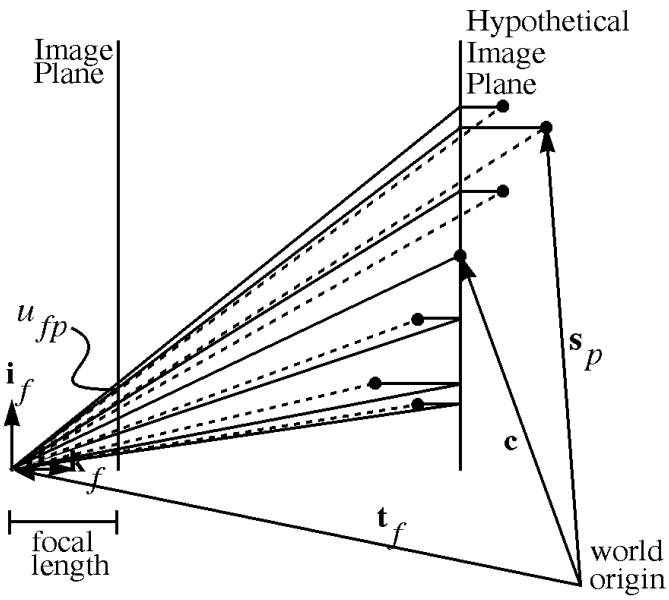


Fig. 12. Scaled orthographic projection in two dimensions. Dotted lines indicate perspective projection.

Because the perspectively projected points all lie on a plane parallel to the image plane, they all lie at the same depth

$$z_f = \left(\mathbf{c} - \mathbf{t}_f\right) \cdot \mathbf{k}_f \tag{49}$$

Thus the scaled orthographic projection equations are very similar to the orthographic projection equations, except that the image plane coordinates are scaled by the ratio of the focal length to the depth z_f.

$$u_{fp} = \frac{l}{z_f}\left(\mathbf{i}_f \cdot \left(\mathbf{s}_p - \mathbf{t}_f\right)\right)$$
$$v_{fp} = \frac{l}{z_f}\left(\mathbf{j}_f \cdot \left(\mathbf{s}_p - \mathbf{t}_f\right)\right) \tag{50}$$

To simplify the equations we assume unit focal length, $l = 1$. The world origin is arbitrary, so we fix it at the object's center of mass, so that $\mathbf{c} = 0$, and rewrite the above equations as

$$u_{fp} = \mathbf{m}_f \cdot \mathbf{s}_p + x_f \quad v_{fp} = \mathbf{n}_f \cdot \mathbf{s}_p + y_f \tag{51}$$

where

$$z_f = -\mathbf{t}_f \cdot \mathbf{k}_f \tag{52}$$

$$x_f = -\frac{\mathbf{t}_f \cdot \mathbf{i}_f}{z_f} \quad y_f = -\frac{\mathbf{t}_f \cdot \mathbf{j}_f}{z_f} \tag{53}$$

$$\mathbf{m}_f = \frac{\mathbf{i}_f}{z_f} \quad \mathbf{n}_f = \frac{\mathbf{j}_f}{z_f} \tag{54}$$

B.2 Decomposition

Because (51) is identical to (2), the measurement matrix W can still be written as $V = MS + T$ just as in the orthographic and paraperspective cases. We still compute x_f and y_f immediately from the image data using (25), and use singular value decomposition to factor the registered measurement matrix W^* into the product of $\hat{M}$ and $\hat{S}$.

B.3 Normalization

Again, the decomposition is not unique and we must determine the 3×3 matrix A which produces the actual motion matrix $M = \hat{M}A$ and the shape matrix $S = A^{-1}\hat{S}$. From (54),

$$\left|\mathbf{m}_f\right|^2 = \frac{1}{z_f^2} \quad \left|\mathbf{n}_f\right|^2 = \frac{1}{z_f^2} \tag{55}$$

We do not know the value of the depth z_f, so we cannot impose individual constraints on $\mathbf{m}_f$ and $\mathbf{n}_f$ as we did in the orthographic case. Instead, we combine the two equations as we did in the paraperspective case, to impose the constraint

$$\left|\mathbf{m}_f\right|^2 = \left|\mathbf{n}_f\right|^2 \tag{56}$$

Because $\mathbf{m}_f$ and $\mathbf{n}_f$ are just scalar multiples of $\mathbf{i}_f$ and $\mathbf{j}_f$, we can still use the constraint that

$$\mathbf{m}_f \cdot \mathbf{n}_f = 0 \tag{57}$$

As in the paraperspective case, (56) and (57) are homogeneous constraints, which could be trivially satisfied by the solution $M = 0$, so to avoid this solution we add the constraint that

$$\left|\mathbf{m}_1\right| = 1 \tag{58}$$

Equations (56), (57), and (58) are the scaled orthographic version of the *metric constraints*. We can compute the 3×3 matrix A which best satisfies them very easily, because the constraints are linear in the six unique elements of the symmetric 3×3 matrix $Q = A^T A$.

B.4 Shape and Motion Recovery

Once the matrix A has been found, the shape is computed as $S = A^{-1}\hat{S}$. We compute the motion parameters as

$$\hat{\mathbf{i}}_f = \frac{\mathbf{m}_f}{\left|\mathbf{m}_f\right|} \quad \hat{\mathbf{j}}_f = \frac{\mathbf{n}_f}{\left|\mathbf{n}_f\right|} \tag{59}$$

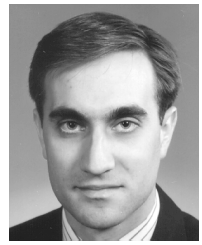
Unlike the orthographic case, we can now compute z_f, the component of translation along the camera's optical axis, from (55).

ACKNOWLEDGMENT

This research was partially supported by the Avionics Laboratory, Wright Research and Development Center, Aeronautical Systems Division (AFSC), U.S. Air Force, Wright-Patterson Air Force Base, OH 45433-6543 under Contract F33615-90-C1465, ARPA Order No. 7597.

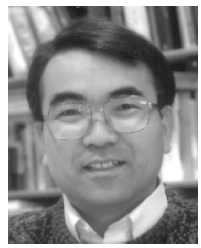
REFERENCES

[1] J.Y. Aloimonos, "Perspective Approximations," *Image and Vision Computing*, vol. 8, no. 3, pp. 177-192, Aug. 1990.
[2] T. Broida, S. Chandrashekhar, and R. Chellappa, "Recursive 3-D Motion Estimation from a Monocular Image Sequence," *IEEE Trans. Aerospace and Electronic Systems*, vol. 26, no. 4, pp. 639-656, July 1990.
[3] J. Costeira and T. Kanade, "A Multi-Body Factorization Method for Motion Analysis," Technical Report CMU-CS-TR-94-220, Carnegie Mellon Univ., Pittsburgh PA, Sept. 1994.
[4] B.D. Lucas and T. Kanade, "An Iterative Image Registration Technique with an Application to Stereo Vision," *Proc. Seventh Int'l Joint Conf. Artificial Intelligence*, 1981.
[5] J.L. Mundy and A. Zisserman, *Geometric Invariance in Computer Vision*. MIT Press, 1992, p. 512.
[6] Y. Ohta, K. Maenobu, and T. Sakai, "Obtaining Surface Orientation from Texels Under Perspective Projection," *Proc. Seventh Int'l Joint Conf. Artificial Intelligence*, pp. 746-751, Aug. 1981.
[7] C.J. Poelman and T. Kanade, "A Paraperspective Factorization Method for Shape and Motion Recovery," Technical Report CMU-CS-93-219, Carnegie Mellon Univ., Pittsburgh, PA, Dec. 1993.
[8] W.H. Press, B.P. Flannery, S.A. Teukolsky, and W.T. Vetterling, *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge Univ. Press, 1988.
[9] L. Quan, "Self-Calibration of an Affine Camera from Multiple Views," Technical Report R.T. Imag-Lifia 26, LIFIA-CNRS-INRIA, Grenoble, France, Nov. 1994.
[10] A. Ruhe and P.A. Wedin, "Algorithms for Separable Nonlinear Least Squares Problems," *SIAM Review*, vol. 22, no. 3, July 1980.
[11] R. Szeliski and S.B. Kang, "Recovering 3D Shape and Motion from Image Streams Using Non-Linear Least Squares," Technical Report 93/3, Digital Equipment Corporation, Cambridge Research Lab, Mar. 1993.
[12] C. Taylor, D. Kriegman, and P. Anandan, "Structure and Motion from Multiple Images: A Least Squares Approach," *IEEE Workshop on Visual Motion*, pp. 242-248, Oct. 1991.
[13] C. Tomasi, "Shape and Motion from Image Streams: A Factorization Method," Technical Report CMU-CS-91-172, Carnegie Mellon University, Pittsburgh, PA, Sept. 1991.
[14] C. Tomasi and T. Kanade, "Shape and Motion from Image Streams Under Orthography: A Factorization Method," *Int'l J. Computer Vision*, vol. 9, no. 2, pp. 137-154, Nov. 1992.
[15] R. Tsai and T. Huang, "Uniqueness and Estimation of Three-Dimensional Motion Parameters of Rigid Objects with Curved Surfaces," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 6, no. 1, pp. 13-27, Jan. 1984.
[16] D. Weinshall and C. Tomasi, "Linear and Incremental Acquisition of Invariant Shape Models from Image Sequences," *Proc. Fourth Int'l Conf. Computer Vision*, Berlin, Germany, pp. 675-682, 1993.



Conrad J. Poelman received BS degrees in computer science and aerospace engineering from the Massachusetts Institute of Technology in 1990 and the PhD degree in computer science from Carnegie Mellon University in 1995.

Dr. Poelman is currently a researcher at the U.S. Air Force Phillips Laboratory in Albuquerque, New Mexico. His research interests include image sequence analysis, model-based motion estimation, radar imagery analysis, satellite imagery analysis, and dynamic neural networks.



Takeo Kanade received his doctoral degree in electrical engineering from Kyoto University, Kyoto, Japan, in 1974. After holding a faculty position at the Department of Information Science, Kyoto University, he joined Carnegie Mellon University in 1980, where he is currently U.A. Helen Whitaker Professor of Computer Science and director of the Robotics Institute. Dr. Kanade has made technical contributions in multiple areas of robotics: vision, manipulators, autonomous mobile robots, and sensors. He has written more than 150 technical papers and reports in these areas. He has been the principal investigator of several major vision and robotics projects at Carnegie Mellon. In the area of education, he was a founding chairperson of Carnegie Mellon University's robotics PhD program, probably the first of its kind.

Dr. Kanade is a Fellow of the IEEE, a Founding Fellow of the American Association of Artificial Intelligence, and the founding editor of the *International Journal of Computer Vision*. He has received several awards, including the Joseph Engelberger Award in 1995 and the Marr Prize Award in 1990. Dr. Kanade has served for many government, industry, and university advisory or consultant committees, including Aeronautics and Space Engineering Board (ASEB) of National Research Council, NASA's Advanced Technology Advisory Committee (congressional mandate committee), and the Advisory Board of the Canadian Institute for Advanced Research.