Linear and Incremental Acquisition of Invariant Shape Models from Image Sequences

Daphna Weinshall*
Department of Computer Science
The Hebrew University of Jerusalem
91904 Jerusalem
ISRAEL

Carlo Tomasi†
Department of Computer Science
Cornell University
Ithaca, NY 14853
USA

Abstract

We show how to automatically acquire similarity-invariant shape representations of objects from noisy image sequences under weak perspective. The proposed method is linear and incremental, requiring no more than pseudo-inverse. It is based on the observation that the trajectories that points on the object form in weak-perspective image sequences are linear combinations of three of the trajectories themselves, and that the coefficients of the linear combinations represent shape in an affine-invariant basis. A nonlinear but numerically sound preprocessing stage is added to improve the accuracy of the results even further. Experiments show that attention to noise and computational techniques improve the shape results substantially with respect to previous methods proposed for ideal images.

1 Introduction

In model-based recognition, images are matched against stored libraries of three-dimensional object representations, so that a good match implies recognition of the object. The recognition process is greatly simplified if the quality of the match can be determined without camera calibration, namely, without having to compute the pose of each candidate object in the reference system of the camera. For this purpose, three-dimensional object representations have been proposed [34] that are invariant with respect to *similarity* transformations, that is, rotations, translations, and isotropic scaling. These are exactly the transformations that occur in the *weak perspective* projection model [1], where images are scaled orthographic projections of rotated and translated objects. Because of its linearity, weak perspective strikes a good balance between mathematical tractability and model generality.

In this paper, we propose a method for *acquiring* a similarity-invariant representation from a sequence of images of the objects themselves. Automatic acquisition from images avoids the tedious and error prone process of typing three-dimensional coordinates of points on the objects, and makes expensive three-dimensional sensors such as laser rangefinders unnecessary. However, model recognition techniques such as geometric hashing have been shown [9] to produce false positive

*This work was done at IBM T.J. Watson Research Center, Yorktown Heights, NY.
†This work was supported by Grant No. IRI-9201751 from the NSF.

matches with even moderate levels of error in the representations or in the images. Consequently, we pay close attention to accuracy and numerical soundness of the algorithms employed, and derive a computationally robust and efficient counterpart to the schemes that previous papers discuss under ideal circumstances.

To be sure, several systems have been proposed for computing depth or shape information from image sequences. For instance, [28, 32, 23, 18, 27, 16] identify the minimum number of points necessary to recover motion and structure from two or three frames, [2, 21] recover depth from many frames when motion is known, [17, 14, 3, 6, 11] consider restricted or partially known motion, [30] solves the complete multiframe problem under orthographic projection, and [26, 13] propose multiframe solutions under perspective projection.

Conceivably, one could use one of these algorithms to determine the complete three-dimensional shape and pose of the object in an Euclidean reference system, and process the results to achieve similarity invariance. However, a similarity-invariant representation is *weaker* than a full representation with pose, since it does not include the orientation of the camera relative to the object. Consequently, the invariant representation contains less information, and ought to be easier to compute. This intuition is supported by experiments with complete calibration and reconstruction algorithms, which, given a good initial guess of the shape of the object, spend a large number of iterations modifying the parameters of the calibration and pose matrices, without affecting the shape by much[1].

In this paper, we show that this is indeed the case. (We assume weak perspective projection; A few recent papers described structure from motion algorithms without camera calibration and with perspective projection [7, 22].) Specifically, we compute a similarity-invariant representation of shape both *linearly* and *incrementally* from a sequence of weak perspective images. This is a very important gain. In fact, a linear multiframe algorithm avoids both the instability of two- or three-frame recovery methods and the danger of local minima that nonlinear multiframe methods must face. Moreover, the incremental nature of our method makes it possible to process images one at a time, moving away from the storage-intensive batch methods of the past.

Our acquisition method is based on the observation that the trajectories that points on the object form in weak perspective image sequences can be written as linear combinations of three of the trajectories themselves, and that the coefficients of the linear combinations represent shape in an affine-invariant basis. This result is closely related to, but different from, the statement that any image in the sequence is a linear combination of three of its images [31].

In this paper, we also show that the optional addition of a nonlinear but numerically sound stage, which selects the most suitable basis trajectories, improves the accuracy of the representation even further. This leads to an image-to-model matching criterion that better discriminates between new images that depict the model object and those that do not. In order to compare our method to existing model acquisition (or structure from motion) methods, we describe a simple transformation, by which we compute a depth representation from the similarity-invariant representation computed by our algorithm.

In the following, we first define the weak perspective imaging model (Section 2). We review the similarity-invariant shape representation and the image-to-model matching measure (Section 3). We then introduce our linear and incremental acquisition algorithm, as well as the nonlinear preprocessing procedure (Section 4). Finally, we evaluate performance with some experiments on real

[1] B. Boufama, personal communication.

image sequences (Section 5).

2 Multiframe Weak Perspective

Under weak perspective, a point $\mathbf{p}_n = (X_n, Y_n, Z_n)^T$ on an object can be related to the corresponding image point $\mathbf{w}_{mn} = (\xi_{mn}, \eta_{mn})^T$ in frame m by a scaling, a rotation, a translation, and a projection:

$$\mathbf{w}_{mn} = \mathcal{P}(s_m R_m \mathbf{p}_n + \tau_m) \tag{1}$$

where R_m is an orthonormal 3×3 matrix, τ_m is a three-dimensional translation vector, s_m is a scalar and $\mathcal{P}$ is the projection operator that simply selects the first two rows of its argument. The two components of $\mathbf{w}_{mn}$ are thus:

$$\begin{aligned} \xi_{mn} &= s_m \mathbf{i}_m^T \mathbf{p}_n + a_m \\ \eta_{mn} &= s_m \mathbf{j}_m^T \mathbf{p}_n + b_m \end{aligned} \tag{2}$$

where the orthonormal vectors $\mathbf{i}_m^T$, $\mathbf{j}_m^T$ are the first two rows of R_m, and a_m, b_m are the first two components of τ_m. In a sequence of images, feature points can be extracted and tracked with one of the methods described in [24, 29, 8] (see the experiments in section 5 below). If N points are tracked in M frames, the equations (2) are repeated MN times, and can be written in matrix form as follows:

$$\begin{bmatrix} \xi_{11} & \cdots & \xi_{1N} \\ \vdots & & \vdots \\ \xi_{M1} & \cdots & \xi_{MN} \\ \eta_{11} & \cdots & \eta_{1N} \\ \vdots & & \vdots \\ \eta_{M1} & \cdots & \eta_{MN} \end{bmatrix} = \begin{bmatrix} s_1 \mathbf{i}_1^T \\ \vdots \\ s_M \mathbf{i}_M^T \\ s_1 \mathbf{j}_1^T \\ \vdots \\ s_M \mathbf{j}_M^T \end{bmatrix} \begin{bmatrix} \mathbf{p}_1 & \cdots \mathbf{p}_N \end{bmatrix} + \begin{bmatrix} a_1 \\ \vdots \\ a_M \\ b_1 \\ \vdots \\ b_M \end{bmatrix} \begin{bmatrix} 1 & \cdots & 1 \end{bmatrix}$$

that is,

$$\hat{W} = RP + \mathbf{t}\mathbf{1}^T \tag{3}$$

where $\mathbf{1}$ is a vector of N ones. Thus, $\hat{W}$ collects the image measurements, R represents both scaling and rotation, P is shape, and $\mathbf{t}$ is translation. In Section 4, we show that R and P need not in fact be computed explicitly in order to compute a similarity-invariant representation.

3 Review of the Similarity-Invariant Representation

Starting with Eq. (3) as a multiframe imaging model, we now describe how to define a shape representation that is invariant with respect to similarity transformations, that is, rigid transformations and isotropic scalings [34]. Specifically, we work towards similarity invariance in three steps:

1. invariance to translation (Section 3.1);
2. invariance to affine transformations (Section 3.2);

3. invariance to similarity transformations (Section 3.3).

For performance evaluation only, we will also discuss the

4. computation of depth (Section 3.4).

Using the invariant representations, we describe an image-to-model matching measure (Section 3.5). For completeness, we outline how to compute the pose of the camera (Section 3.6), although this is not used in this paper.

3.1 Translation Invariance

Invariance with respect to translation is easily achieved by measuring the coordinates in P using some linear combination of the points themselves as a reference origin: if the object translates, so does the reference point, and the representation does not change. In [16], it is suggested to pick an object point as the reference origin. In [30], the centroid of all the points is used instead (this is the optimal translation in a least-square sense [11]). In practice, our experiments show that there is little difference between these two choices.

Let now $\mathbf{t}$ be a $2M$-dimensional vector that collects the two coordinates of the projection of the reference point in each frame. In a suitable reference system where the reference point is the origin, this vector is the same as the vector $\mathbf{t}$ of Eq. (3). The image measurements in the matrix $\hat{W}$ can now be centered by subtracting $\mathbf{t}$ from each column to yield

$$W = \hat{W} - \mathbf{t}\mathbf{1}^T \; , \tag{4}$$

so that Eq. (3) becomes

$$W = RP \; . \tag{5}$$

3.2 Affine Transformation Invariance

The $M \times 3$ matrix R in Eq. (5) is built from 3×3 orthonormal matrices and isotropic scaling factors (see Eq. (2)). Therefore, corresponding rows in the upper and lower halves of R (that is, rows m and $m + M$ for $m = 1, \ldots, M$) must be mutually orthogonal and have the same norm s_m. If these *orthogonality* constraints are satisfied, we say that R, $\mathbf{t}$ represent full Euclidean motion, and the corresponding P represents Euclidean shape. In particular, the columns of P are the three-dimensional coordinates of the object points with respect to some orthonormal reference basis.

Invariance with respect to affine transformations is achieved by replacing this basis by one that is more intimately related to the shape of the object. Specifically, the basis is made by three of the object points themselves, that is, by the vectors from the reference origin to the three points, assumed not to be coplanar with the origin. This basis is no more orthonormal. The new coordinates were called *affine* in [16]. If now the object undergoes some affine transformation, so do the basis points, and the affine coordinates of the N object points do not change.

The choice of the three basis points can be important. In fact, the requirement that the points be noncoplanar with the origin is not an all-or-nothing proposition. Four points can be *almost* coplanar, and with noisy data this is almost as bad as having *exactly* coplanar points. We discuss this issue in Section 4, where we propose a method that selects a basis as far away as possible from being coplanar with the origin. In the experiments of Section 5, this additional nonlinear stage is shown to improve markedly the quality of the shape representation.

Notice that in the new affine basis the three selected basis points have coordinates $(1,0,0)$, $(0,1,0)$, and $(0,0,1)$, so that the new $3 \times N$ matrix A of affine coordinates is related to the Euclidean matrix P of Eq. (5) by the 3×3 linear transformation:

$$P_b A = P \tag{6}$$

where $P_b = [\mathbf{p}_i \; \mathbf{p}_j \; \mathbf{p}_k]$ is the submatrix of P that collects the three selected basis points.

If we substitute Eq. (6) into the centered projection Eq. (5), we obtain

$$W = W_b A \tag{7}$$

where $W_b = RP_b$. However, because the submatrix $A_b = [\mathbf{a}_i \; \mathbf{a}_j \; \mathbf{a}_k]$ is the identity matrix, we see that W_b is a submatrix of W:

$$W_b = \left[\begin{array}{ccc} \mathbf{w}_i & \mathbf{w}_j & \mathbf{w}_k \end{array} \right] .$$

In more geometric terms, Eq. (7) expresses the following key result:

> all the image trajectories (W) of the object points can be written as a linear combination of the image trajectories (W_b) of three of the points. The coefficients (A) of the linear combinations are the three-dimensional coordinates of the corresponding points in space in the affine three-dimensional basis of the points themselves.

Notice the analogy and difference between this result and the statement, made in [31], that under weak perspective any image of an object is a linear combination of three of its views. We are saying that any trajectory is a linear combination of three trajectories, while they are saying that any snapshot is a linear combination of three snapshots. The concise matrix equation in (7) contains these two statements in a symmetric form: Ullman and Basri read the equation by rows, we read it by columns. When we discuss the recognition stage in Section 3.5, we consider the case in which new rows (unfamiliar views) are added to W, and so we revert to Ullman and Basri's reading of Eq. (7).

3.3 Similarity Invariance

Although conceptually a useful intermediate step, affine transformations are too general. Affine-invariant representations are also invariant with respect to similarity transformations, since the latter are a special case of the former. However, a representation that is invariant with respect to similarity and not, say, shearing (an affine transformation) will be able to discriminate among more models: for instance, the roman letter 'A' will not be confused with an italic '*A*'. In other words, it is best to have representations that are invariant *exactly* with respect to the class of possible transformations between model and viewer reference systems. Under weak perspective, this is the class of similarity transformations, a proper subset of affine transformations.

To achieve invariance with respect to similarity transformations, we augment the affine representation introduced above with metric information about the three basis points. Of course, we cannot simply list the coordinates of the three basis points in a fixed reference system, since these coordinates would not be invariant with respect to rotation and scaling. Instead, we introduce the *Gramian* matrix of the three basis points, defined as follows [34]:

$$G = \begin{bmatrix} \mathbf{p}_i^T\mathbf{p}_i & \mathbf{p}_i^T\mathbf{p}_j & \mathbf{p}_i^T\mathbf{p}_k \\ \mathbf{p}_j^T\mathbf{p}_i & \mathbf{p}_j^T\mathbf{p}_j & \mathbf{p}_j^T\mathbf{p}_k \\ \mathbf{p}_k^T\mathbf{p}_i & \mathbf{p}_k^T\mathbf{p}_j & \mathbf{p}_k^T\mathbf{p}_k \end{bmatrix} . \tag{8}$$

In Section 4.2 we normalize G to make it invariant to scaling.

The Gramian is a symmetric matrix, and is defined in terms of the *Euclidean* coordinates of the basis points (see Eq. (1)). However, we show in Section 4 that G can be computed *linearly* from the images, without first computing the depth or pose of the object.

The pair of matrices (A, G) is our target representation. Because of centering and the appropriate selection of the affine basis, A is invariant with respect to translations and affine transformations. The Gramian G, on the other hand, can be normalized for scale invariance. Furthermore, except for scale, its entries are cosines of the angles between the vectors of P_b (see Eq. (8)). This makes G invariant to rotations and, because of centering, to translations, but sensitive to affine transformations that change angles between points. In the next subsection we show constructively that the pair (A, G) contains complete information about the object's shape, but not directly about its pose in each image.

3.4 Depth Map

Determining the depth of the object requires to express its shape in an orthonormal system of reference, that is, to compute the matrix P of Eq. (5). We now show that the shape Gramian G of Eq. (8) contains all the necessary information. In fact, let W_b be the matrix of the basis trajectories introduced in Eq. (7), and let P_b be the coordinates of the corresponding basis points in space in an orthonormal reference system (see Eq. (6)). Then, the definition (8) of the Gramian can be rewritten as

$$G = P_b^T P_b . \tag{9}$$

Suppose now that T is the Cholesky factor of the Gramian G. We recall [10] that the Cholesky factor of a symmetric positive definite matrix G is the unique upper triangular matrix T with positive diagonal entries such that

$$G = T^T T . \tag{10}$$

Eq. (9) and Eq. (10) are formally similar factorizations of G. We claim that P_b can differ from T only by a rotation or a mirror transformation, so that T is in fact the representation of the three selected basis points on the object in an orthonormal frame of reference. The projection equation (5) does not specify the particular orientation of the orthonormal axes of the underlying reference system, so P_b and T can be taken to be the same matrix:

$$P_b = T .$$

The remaining ambiguity due to the possibility that T is a mirror reflection of the "true" P_b is intrinsic in the weak perspective model (Necker reversal), and cannot be eliminated. Similarly, the overall scale cannot be determined. In fact, we will see in Section 8 that G is the solution of a *homogeneous* system of linear equations.

From the triangular structure of T we can easily determine where the new orthonormal basis vectors are with respect to the points in space: the first basis vector points from the origin to point 1, because the first entry of column 1 in T is its only nonzero entry. Since the third entry of the second column of T is zero, the second basis vector, orthogonal to the first, must be in the plane of the origin and points 1 and 2. Finally, the third basis vector is orthogonal to the first two.

To prove our claim that P_b and T can only differ by orthonormal transformations, we first observe that $G = T^T T = P_b^T P_b$, which implies that T and P_b must be related by a linear transformation. In other words, there must be a 3×3 matrix Q such that $T = QP_b$. However, by replacing this expression in $T^T T = P_b^T P_b$, we obtain $Q^T Q = I$ (the 3×3 identity), so Q must be orthonormal.

In summary, we have the following method for computing the shape matrix P of Eq. (5): determine the Gramian G by the linear method of Section 4, take its Cholesky factorization $T = P_b$, and let T be the transformation of the affine shape matrix A into the new orthonormal basis. Namely:

$$P = TA$$

Notice that the three basis points i, j, k, whose coordinates in A are the identity matrix, are transformed into the columns of T.

We emphasize once more that this last decomposition stage need not be performed for the computation of the similarity-invariant shape representation. Furthermore, this stage can fail in the presence of noise. In fact, the matrix G can be Cholesky-decomposed only if it is positive definite. Bad data can cause this condition to be violated.

3.5 Recognition

Once the similarity-invariant representation (A, G) has been determined from a given sequence of images, it can be used on new, unfamiliar views to determine whether they contain the object represented by (A, G). In fact, from Eq. (5) and Eq. (9) we obtain

$$W_b G^{-1} W_b^T = R P_b (P_b^T P_b)^{-1} P_b^T R^T = R R^T \ .$$

If we write out the relevant terms of this equation, we have $\forall m$:

$$\begin{aligned} \mathbf{x}_m^T G^{-1} \mathbf{x}_m &= s_m^2 \\ \mathbf{y}_m^T G^{-1} \mathbf{y}_m &= s_m^2 \\ \mathbf{x}_m^T G^{-1} \mathbf{y}_m &= 0 \end{aligned}$$

so that in particular

$$\begin{aligned} \mathbf{x}_m^T G^{-1} \mathbf{x}_m &= \mathbf{y}_m^T G^{-1} \mathbf{y}_m \\ \mathbf{x}_m^T G^{-1} \mathbf{y}_m &= 0 \end{aligned} \tag{11}$$

where the vectors $\mathbf{x}_m^T = (x_{m1}, x_{m2}, x_{m3})$ and $\mathbf{y}_m^T = (y_{m1}, y_{m2}, y_{m3})$ are the rows of the upper and lower half of the centered image measurement matrix W_b. Namely, $\mathbf{x}_m$ and $\mathbf{y}_m$ are the centered image measurement of the basis points in frame m.

Eq. (11) provide strong constraints, capturing all the information that can be obtained from a single image, since all the images that satisfy Eq. (11) are a possible instance of the object represented by G. The two equations in Eq. (11) can be used in two ways: during recognition, the given G can be used to check whether new image measurements $\mathbf{x}_m^T$, $\mathbf{y}_m^T$ represent the same three basis points as in the familiar views, thus yielding a key for indexing into the object library. During acquisition of the shape representation, on the other hand, G is the unknown, and Eq. (11) can be solved for G. In this section, we review the use of G for recognition (as discussed at length in [34]). We discuss its computation, the major point of the present paper, in Section 4.

Essentially, the recognition scheme checks the residues between the left- and the righthand sides of the equations in (11). The residues, however, must be properly normalized to make the matching criterion independent of scale. The most straightforward definition of matching measure for the three basis points is the following [34]:

$$g(\mathbf{x}_m, \mathbf{y}_m) = \frac{|\mathbf{x}_m^T G^{-1} \mathbf{y}_m| + |\mathbf{x}_m^T G^{-1} \mathbf{x}_m - \mathbf{y}_m^T G^{-1} \mathbf{y}_m|}{|\mathbf{x}_m^T G^{-1} \mathbf{x}_m| + |\mathbf{y}_m^T G^{-1} \mathbf{y}_m|} \; . \tag{12}$$

The function (12) is a quadratic image invariant that can be used to check the basis points.

The matching of all the other points, on the other hand, can be verified by using the affine part A of the similarity-invariant shape representation. In fact, all the terms of Eq. (7) are known once a new frame

$$W = \begin{bmatrix} \mathbf{x}_m^T \\ \mathbf{y}_m^T \end{bmatrix}$$

becomes available, so the following matching criterion suggests itself naturally:

$$a(\mathbf{x}_m, \mathbf{y}_m) = \sum_{l \neq i,j,k} \left(\frac{|x_l - (\mathbf{x}_b)_m^T \mathbf{a}_l|}{|(\mathbf{x}_b)_m^T \mathbf{a}_l|} + \frac{|y_l - (\mathbf{y}_b)_m^T \mathbf{a}_l|}{|(\mathbf{y}_b)_m^T \mathbf{a}_l|} \right) \tag{13}$$

where $(\mathbf{x}_b)_m = [x_{mi} \; x_{mj} \; x_{mk}]$ and $(\mathbf{y}_b)_m = [y_{mi} \; y_{mj} \; y_{mk}]$ are the new image coordinates of the basis points, and $\mathbf{a}_1, \ldots, \mathbf{a}_N$ are the columns of A.

3.6 Pose

The computation of pose is beyond the scope of the present paper. For completeness, we now briefly outline methods to compute pose.

Since the Gramian matrix G defined in Eq. (8) contains the complete $3D$ information on the basis points, and given the x and y coordinates of the basis points in frame m $(\mathbf{x}_b)_m$ and $(\mathbf{y}_b)_m$, it is straightforward to obtain their depth z in the coordinate system $\mathcal{M}$ corresponding to the camera's orientation in frame m. Namely, the coordinate system whose $X - Y$ plane is parallel to the image plane in frame m, and whose depth axis Z is perpendicular to the image plane. The known coordinates of the basis points in system $\mathcal{M}$ and in the orthonormal system defined in Section 3.4 can be used to compute the transformation between the two coordinates system, e.g., using the method described in [11]. This transformation gives the pose of the camera in frame m.

Alternatively, it is possible to compute pose directly from the image measurements, using Eq. (5). Tomasi & kanade [30] described a relatively simple non-linear method, which computes pose from the Singular Value Decomposition of the centered measurement matrix W.

4 The Algorithm

In this section, we show how to compute the affine shape matrix A (Section 4.1) and the Gramian G of the basis points (Section 4.2) linearly and incrementally from a sequence of images. We then show how to choose three good basis points i, j, k (Section 4.3). This algorithm can use as little as two frames and five points for computing matrix A, and as little as three frames and four points for computing matrix G. More data can be added to the computation incrementally, if and when available.

4.1 The Affine Shape Matrix

The affine shape matrix A is easily computed as the solution of the overconstrained linear system (7), which we repeat for convenience:

$$W = W_b A \ .$$

Recall that W is the matrix of centered image measurements, and W_b is the matrix of centered image measurements of the basis points.

It is well known from the literature of Kalman filtering that linear systems can be solved incrementally (see for instance [20]) one row at a time. The idea is to realize that the expression for the solution

$$A = W_b^+ W$$

where W_b^+ is the pseudoinverse of W_b:

$$W_b^+ = (W_b^T W_b)^{-1} W_b^T$$

is composed of two parts whose size is independent of the number of image frames, namely, the so-called covariance matrix

$$Q = (W_b^T W_b)^{-1}$$

of size 3×3, and the $3 \times M$ matrix

$$S = W_b^T W \ .$$

Both Q and S can be updated incrementally every time a new row $\mathbf{w}^T$ is added to W (so the corresponding row $\mathbf{w}_b^T$ is also added to W_b). Specifically, the matrices Q_+ and S_+ after the update are given by

$$\begin{aligned} Q_+ &= (I - \frac{Q\mathbf{w}_b\mathbf{w}_b^T}{1 + \mathbf{w}_b^T Q \mathbf{w}_b})Q \\ S_+ &= S + \mathbf{w}_b \mathbf{w}^T \end{aligned}$$

where I is the 3×3 identity matrix. For added efficiency, this pair of equations can be manipulated into the following update rule for A [20]:

$$A_+ = A + \frac{Q\mathbf{w}_b}{1 + \mathbf{w}_b^T Q \mathbf{w}_b} \left(\mathbf{w}^T - \mathbf{w}_b^T A \right) \ .$$

Note that the computation of A requires at least two frames.

4.2 The Gramian

For each frame m, the two equations in (11) define linear constraints on the entries of the inverse Gramian $H = G^{-1}$, so H can be computed as the solution of a linear system. This system, however, is homogeneous, so H can only be computed up to a scale factor.

To write this linear system in the more familiar form $C\mathbf{h} = 0$, we first notice that H is a symmetric 3×3 matrix, so it has six distinct entries h_{ij}, $1 \leq i \leq j \leq 3$. Let us gather those entries in the vector

$$\mathbf{h} = \begin{bmatrix} h_{11} \\ h_{12} \\ h_{13} \\ h_{22} \\ h_{23} \\ h_{33} \end{bmatrix} .$$

Furthermore, given two 3-vectors $\mathbf{a}$ and $\mathbf{b}$, define the operator

$$\mathbf{z}^T(\mathbf{a}, \mathbf{b}) = \begin{bmatrix} a_1b_1 & a_1b_2 + a_2b_1 & a_1b_3 + a_3b_1 & a_2b_2 & a_2b_3 + a_3b_2 & a_3b_3 \end{bmatrix} .$$

Then, the equations in (11) are readily verified to be equivalent to the $2M \times 6$ system

$$C\mathbf{h} = 0 \tag{14}$$

where

$$C = \begin{bmatrix} \mathbf{z}^T(\mathbf{x}_1, \mathbf{x}_1) - \mathbf{z}^T(\mathbf{y}_1, \mathbf{y}_1) \\ \vdots \\ \mathbf{z}^T(\mathbf{x}_M, \mathbf{x}_M) - \mathbf{z}^T(\mathbf{y}_M, \mathbf{y}_M) \\ \mathbf{z}^T(\mathbf{x}_1, \mathbf{y}_1) \\ \vdots \\ \mathbf{z}^T(\mathbf{x}_M, \mathbf{y}_M) \end{bmatrix} .$$

A unit norm solution to this linear system is reliably and efficiently obtained from the singular value decomposition of $C = U_C \Sigma_C V_C^T$ as

$$\mathbf{h} = \mathbf{v}_{C6} ,$$

the sixth column of V_C. Because this linear system is overconstrained as soon as $M \geq 3$, the computation of H, and therefore of the Gramian $G = H^{-1}$, can be made insensitive to noise if sufficiently many frames are used. Notice that the fact that the vector $\mathbf{h}$ has unit norm automatically normalizes the Gramian.

Alternatively, in order to obtain an incremental algorithm for the computation of the Gramian G, Eq. (14) can be solved with pseudo-inverse. (The incremental implementation of pseudo-inverse was discussed in Section 4.1.) However, the method of choice for solving a homogeneous linear system, which avoids rare singularities, is the method outlined above using SVD.

4.3 Selecting a Good Basis

The computation of the similarity-invariant representation (A, G) is now complete. However, no criterion has yet been given to select the three basis points i, j, k. The only requirement so far has been that the selected points should not be coplanar with the origin. However, a basis can be very close to coplanar without being strictly coplanar, and in the presence of noise this is almost equally troublesome.

To make this observation more quantitative, we define a basis to be good if for any vector $\mathbf{v}$ the coordinates $\mathbf{a}$ in that basis do not change much when the basis is slightly perturbed. Quantitatively, we can measure the quality of the basis by the norm of the largest perturbation of $\mathbf{a}$ that is obtained as $\mathbf{v}$ ranges over all unit-norm vectors. The size of this largest perturbation turns out to be equal to the *condition number* of W_b, that is, to the ratio between its largest and smallest singular values [10].

The problem of selecting three columns W_b of W that are as good as possible in this sense is known as the *subset selection problem* in the numerical analysis literature [10]. In the following, we summarize the solution to this problem proposed in [10]:

1. compute the singular value decomposition of W,
$$W = U \Sigma V^T \ ;$$
2. apply QR factorization with column pivoting to the right factor V^T:
$$V^T = \hat{Q} \hat{R} \Pi^T \ .$$

The first three columns of the permutation matrix Π are all zero, except for one entry in each column, which is equal to one. The row subscripts of those three nonzero entries are the desired subscripts i, j, k.

The rationale of this procedure is that singular value decomposition preconditions the shape matrix, and then QR factorization with column pivoting brings a well conditioned submatrix in front of $\hat{Q}\hat{R}$. (See [10] for more details, as well as for definitions and algorithms for singular value decomposition and QR factorization.)

Although heuristic in nature, this procedure has proven to work well in all the cases we considered (see also [5] for a discussion of possible alternatives). Both the singular value decomposition and the QR factorization of a $M \times N$ matrix can be performed in time $O(MN^2)$, so this heuristical algorithm is much more efficient than the $O(MN^3)$ brute-force approach of computing the condition numbers of all the possible bases.

4.4 Summary of the Algorithm

The following steps summarize the algorithm for the acquisition of the similarity-invariant representation (A, G) from a sequence $\hat{W}$ of images under weak perspective (see Eq. (3)).

1. Center the measurement matrix with respect to one of its columns or the centroid of all its columns:
$$W = \hat{W} - \mathbf{t}\mathbf{1}^T$$
where $\mathbf{t}$ is either the first column of W or the average of all its columns.

2. (optional) Find a good basis i, j, k for the columns of W as follows:

   (a) compute the singular value decomposition of W,

   $$W = U\Sigma V^T \; ;$$

   (b) apply QR factorization with column pivoting to the right factor V^T:

   $$V^T = \hat{Q}\hat{R}\Pi^T \; .$$

   The row subscripts of the three nonzero entries in the first three columns of Π are i, j, k. These are the indices of the chosen basis points.

3. Compute the solution A to the overconstrained system $W = W_b A$ by adding one row at a time. Specifically, initialize A to a $3 \times N$ matrix of zeros. Let $\mathbf{w}^T$ be a new row, let $\mathbf{w}_b^T$ collect entries i, j, k of $\mathbf{w}$, and let $Q = (W_b^T W_b)^{-1}$. The matrix A is updated to

   $$A_+ = A + \frac{Q\mathbf{w}_b}{1 + \mathbf{w}_b^T Q \mathbf{w}_b} \left(\mathbf{w}^T - \mathbf{w}_b^T A \right) \; .$$

4. Determine the Gramian G as follows:

   (a) construct the $2M \times 6$ matrix

   $$C = \begin{bmatrix} \mathbf{z}^T(\mathbf{x}_1, \mathbf{x}_1) - \mathbf{z}^T(\mathbf{y}_1, \mathbf{y}_1) \\ \vdots \\ \mathbf{z}^T(\mathbf{x}_M, \mathbf{x}_M) - \mathbf{z}^T(\mathbf{y}_M, \mathbf{y}_M) \\ \mathbf{z}^T(\mathbf{x}_1, \mathbf{y}_1) \\ \vdots \\ \mathbf{z}^T(\mathbf{x}_M, \mathbf{y}_M) \end{bmatrix}$$

   where

   $$\mathbf{z}^T(\mathbf{a}, \mathbf{b}) = \begin{bmatrix} a_1 b_1 & a_1 b_2 + a_2 b_1 & a_1 b_3 + a_3 b_1 & a_2 b_2 & a_2 b_3 + a_3 b_2 & a_3 b_3 \end{bmatrix} \; ;$$

   (b) solve the system

   $$C\mathbf{h} = 0$$

   which yields the distinct entries of the symmetric matrix H. Compute G as the inverse of H.

This is the complete acquisition algorithm. When new images become available, the criteria (12) and (13) can be used to decide whether the new images depict the same object as the old ones.

In order to compute a depth map P from the similarity-invariant representation (A, G), another optional step is added to the algorithm:

5. (optional) Take the Cholesky factorization of matrix $G = T^T T$, and let T be the transformation of the affine shape matrix A into an orthonormal basis. Namely:

   $$P = TA$$

5 Experiments

We first illustrate the ideas presented in this paper with some experiments performed on a real image sequence (Section 5.1).

We applied our algorithm, including the depth computation, to two other sequences of images, originally taken by Rakesh Kumar and Harpreet Singh Sawhney at UMASS-Amherst. The data was provided by J. Inigo Thomas from UMass, who also provided the solution to the correspondence problem (namely, a list of the coordinates of the tracked points in all the frames).

For comparison, we received the $3D$ coordinates of the points in the first frame as ground truth. We used the algorithm described in [11] to compute the optimal similarity transformation between the invariant depth map representation computed by our algorithm (step 5), and the given data in the coordinate system of the first frame. We applied the transformation to our depth reconstruction to obtain z_{est} at each point, and compared this output with the ground truth data z_{real}. We report the relative error at each point, namely, $\frac{z_{est}-z_{real}}{z_{real}}$.

We evaluated the affine shape reconstruction separately. We computed the optimal affine transformation between the invariant affine representation computed by our algorithm (matrix A computed in step 3), and the given depth data in the coordinate system of the first frame. We applied the transformation to the affine shape representation to obtain z_{est}^{aff} at each point, and compared this output with the ground truth data z_{real}.

5.1 The Ping Pong Ball:

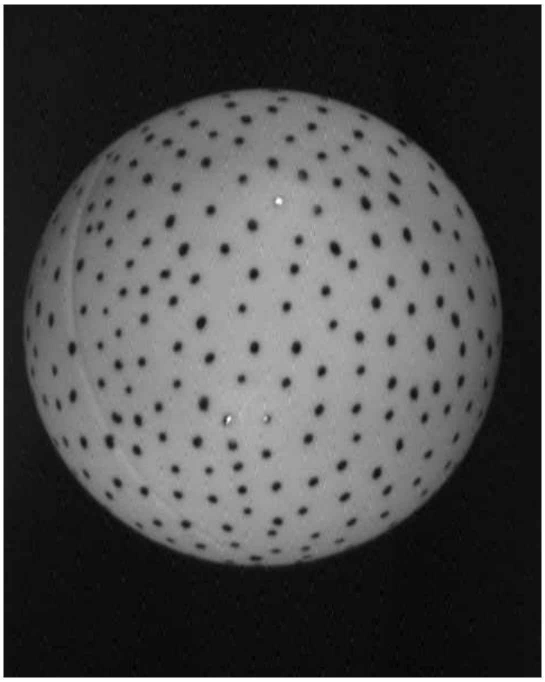
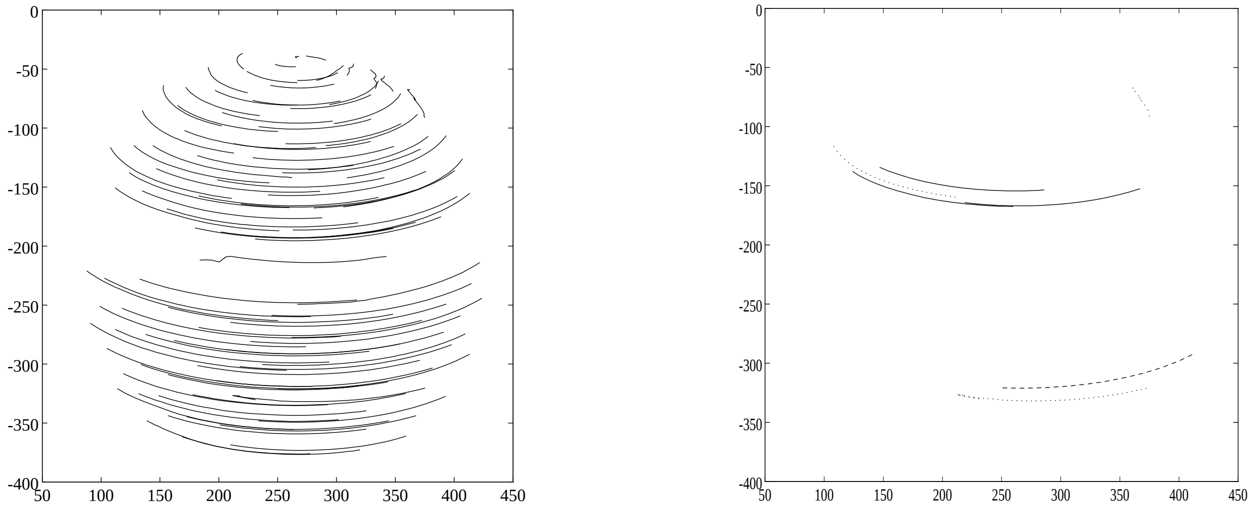


Figure 1: The first frame of the ping pong ball sequence.

A ping pong ball with clearly visible marks was rotated in front of a Sony camera. Thirty frames were used in these experiments, and the ball rotated 2 degrees per frame. Fig. 1 shows the first frame of the sequence. A number of frames (five in some cases, fifteen in others) were used for the acquisition of the similarity-invariant representation. The quadratic and linear matching criteria (12) and (13) were then applied to each frame of the entire sequence, old frames and new alike. In every experiment, the two criteria were also applied to a random sequence. The ratio between the value of the criterion applied to the actual sequence over that for the random sequence was used as a performance measure: if the ratio is very small, the algorithm discriminates well between the "true" object and other "false" comparison objects.

The tracker described in [29] was used to both select and track features from frame to frame. Only those features that were visible throughout the thirty frames were used in these experiments. Ninety points on the ball satisfied this requirement. Fig. 2a shows the trajectories of those 90 points. Fig. 2b shows the trajectories of points used as basis and origin. More specifically, the dashed trajectory is that of point 1, used for centering the measurement matrix. The three solid trajectories (two of them overlap) correspond to the basis points found by the basis selection procedure. The three dotted trajectories are a very poor choice, corresponding to three points that happen to be close to coplanar with the origin. The condition number of the good basis is about 19, that of the poorly selected basis is about 300.



(a) (b)

Figure 2: (a) The ninety tracks automatically selected and tracked through thirty frames. (b) The dashed track was used as the origin; The three solid tracks are the ones chosen by the selection algorithm, while the three dotted tracks yield a poor basis, since the corresponding points in space are almost coplanar with the origin.

Fig. 3 shows the ratios described above for the quadratic criterion (12) and for the linear criterion (13), respectively. In particular, the dotted curves correspond to the dotted basis of Fig. 2b, and

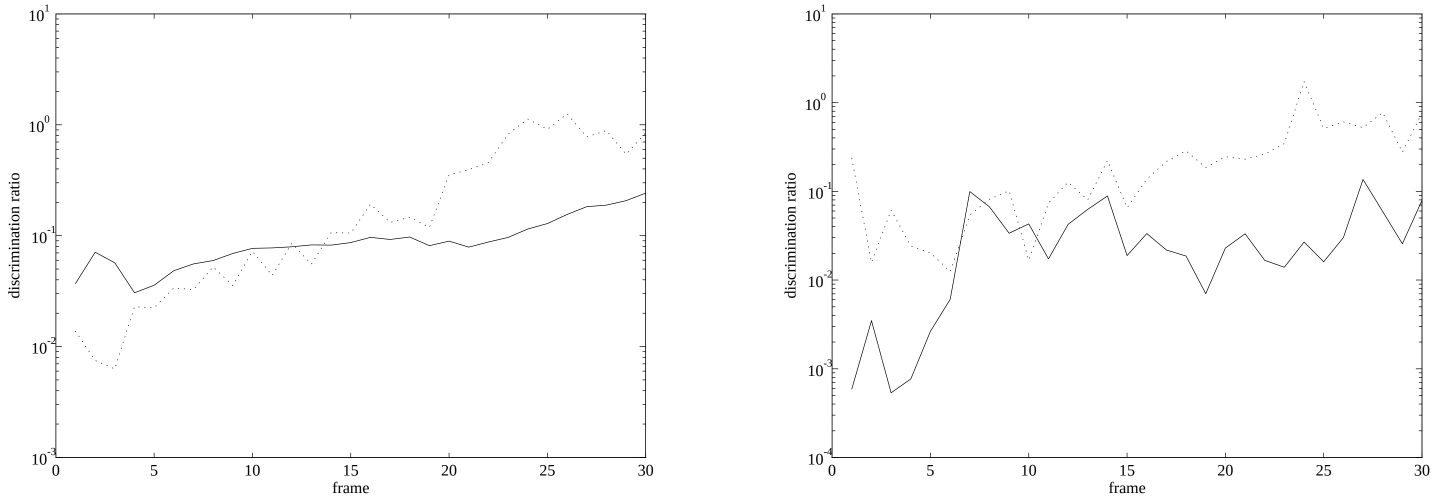


Figure 3: Discrimination ratio (true/random sequence) for the quadratic (left) and linear (right) criterion, plotted in logarithmic scale. The dotted curves use the poor basis, the solid curves use the good basis.

the solid curves correspond to the solid basis.

For the poor basis (dotted curves in Fig. 3), only the first five frames were used for acquisition. Both the quadratic and the linear criterion functions are good (small) for those five frames, and then deteriorate gradually for new unfamiliar views (frames 5-30). In particular, the quadratic criterion becomes basically useless after frame 23 or so, that is, after more than 30 degrees of rotation away from the familiar views. In fact, the value that the criterion returns for a random sequence is about the same as the one it returns for the actual object. Up to frame 15 or so, however, the quadratic criterion has at least a 10:1 discrimination ratio. Similar considerations, but with different numbers and a more erratic trend, hold for the linear criterion.

The situation changes substantially with the good basis chosen by the selection algorithm. The trajectories are the solid lines in Fig. 2b, and the corresponding plots are the solid curves in Fig. 3. For the linear criterion, the situation is undoubtedly improved: the discrimination ratio becomes an order of magnitude better in most cases. The results with the quadratic criterion are less crisp. However, the effect on unfamiliar views are clearly beneficial, and make the criterion useful throughout the sixty degrees of visual angle spanned by the sequence.

Fig. 4 shows the effect of changing the number or distribution of frames used for acquisition. The good basis is used for all the curves. The solid curves, labeled '1-5', are the same as the solid curves in Fig. 3: the first five frames were used for acquisition. The dashed curves, labeled '1-15', are the result of using the first fifteen frames for acquisition. For the quadratic criterion, the results are at some points two orders of magnitude better, and consistently better throughout the sequence. The bad effect of moving away from familiar viewing angles, however, is even more marked: the discrimination ratio is small (good) for the fifteen frames used for acquisition, but increases rapidly for unfamiliar frames. Even there, however, the value of the quadratic criterion

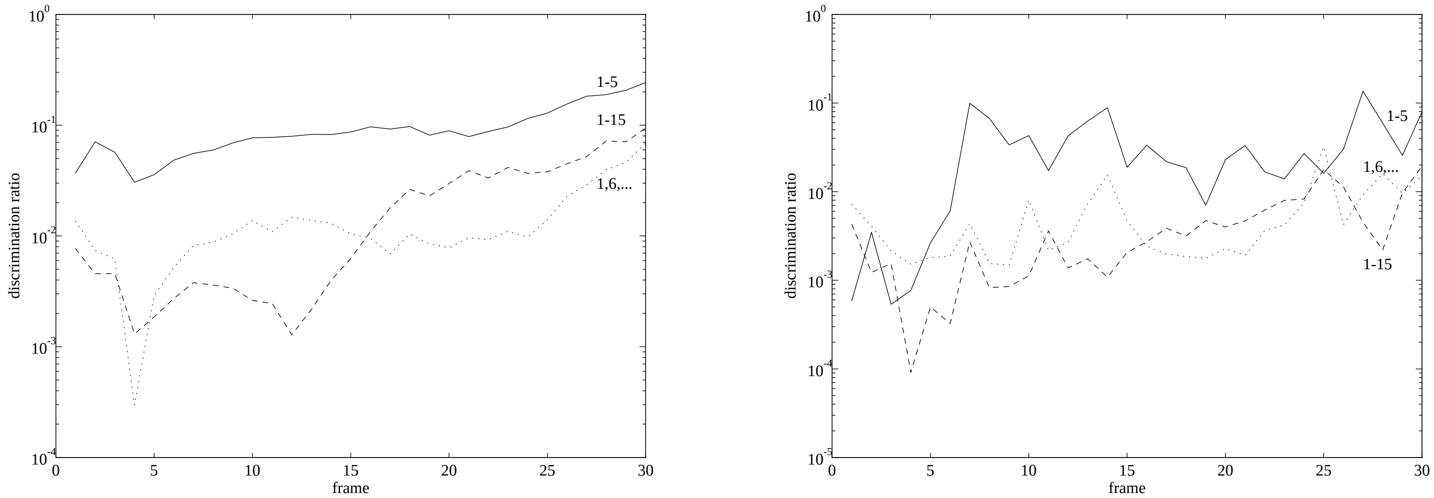


Figure 4: Discrimination ratio (true/random sequence) for the quadratic (left) and linear (right) criteria, plotted in logarithmic scale. The solid curve uses the first five frames for acquisition, the dashed curve uses the first fifteen frames, the dotted curve uses five frames spread throughout the sequence.

is more than ten times smaller for the real object than it is for a random sequence (that is, the discrimination ratio is smaller than 0.1).

The dotted lines in Fig. 4 ('1,6,...') use again five frames during acquisition, but the five frames are spread throughout the viewing positions. Specifically, frames 1,6,11,16,21 were used. Not surprisingly, the results are typically worse over the fifteen frames used for the dashed curve, but they are much better for unfamiliar views. In other words, interpolating between unfamiliar views is better than extrapolating from them. Similar considerations hold again for the linear criterion.

5.2 Box sequence:

This sequence includes 8 images of a rectangular chequered box rotating around a fixed axis (one frame is shown in Fig. 5). 40 corner-like points on the box were tracked. The depth values of the points in the first frame ranged from 550 to 700 mms, therefore weak perspective provided a good approximation to this sequence. (See a more detailed description of the sequence in [25] Fig. 5, or [15] Fig. 2.)

We compared the relative errors of our algorithm to the errors reported in [25]. Three results were reported in [25] and copied to Table 1: column "Rot." – depth computation with their algorithm, which assumes perspective projection and rotational motion only; column "2-frm" – depth computation using the algorithm described in [12], which uses 2-frames only; and column "2-frm, Ave." – depth computation using the 2-frames algorithm, where the depth estimates were averages over six pairs of frames. Table 1 summarized these results, as well as the results using our affine algorithm (column "Aff. Invar.") and similarity algorithm (column "Rigid Invar.").

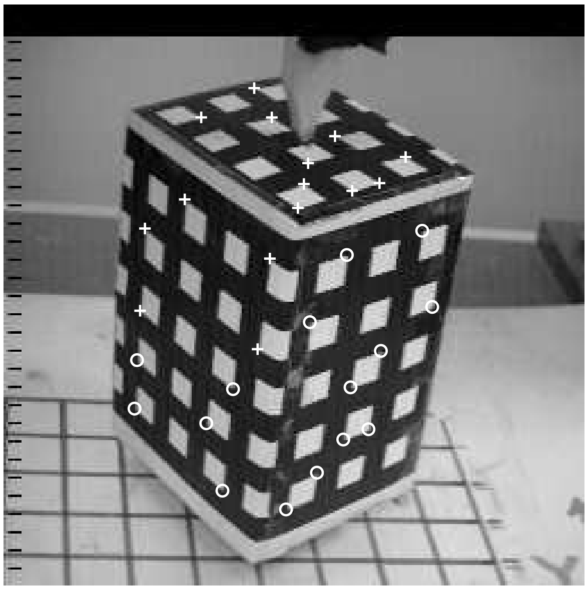


Figure 5: One frame from the box sequence.

Pt.	Pose	Rigid Invar.		Aff. Invar.		Rot.		2-frm		2-frm, Ave.	
#	Z	Z	E (%)	Z	E (%)	Z	E (%)	Z	E (%)	Z	E (%)
1	591.4	587.8	-0.6	591.3	-0.0	588.9	-0.4	613.9	3.8	591.7	0.1
2	666.3	669.7	0.5	662.0	-0.7	665.8	-0.1	694.4	4.2	666.4	0.0
3	621.8	618.3	-0.6	621.9	0.0	617.8	-0.6	648.4	4.3	624.9	0.5
4	640.7	642.2	0.2	640.2	-0.1	635.0	-0.9	667.5	4.2	641.5	0.1
5	637.7	633.7	-0.6	637.0	-0.1	637.7	0.0	665.0	4.3	639.6	0.3
6	647.9	647.6	-0.1	647.0	-0.1	650.9	0.5	679.2	4.8	651.7	0.6
7	656.6	656.5	-0.0	654.3	-0.3	661.9	0.8	687.5	4.7	658.8	0.3
8	640.0	640.2	0.0	639.7	-0.0	653.8	2.2	668.0	4.4	642.3	0.4
9	709.7	708.8	-0.1	706.5	-0.5	700.7	-1.3	744.8	5.0	714.4	0.7
10	614.8	615.4	0.1	618.2	0.5	603.6	-1.8	644.1	4.8	618.5	0.6
11	602.3	602.6	0.0	601.5	-0.1	606.2	0.6	626.9	4.1	604.8	0.4
12	628.9	631.2	0.4	626.3	-0.4	636.5	1.2	655.3	4.2	630.5	0.2
ave.			0.27%		0.23%		0.86%		4.4%		0.35%

Table 1: Comparison of the relative errors in depth computation using our algorithm (rigid and affine shape separately), with two other algorithms. The average of the absolute value of the relative errors is listed at the bottom for each algorithm.

5.3 Room sequence

This sequence, which was used in the 1991 motion workshop, includes 16 images of a robotics laboratory, obtained by rotating a robot arm 120^o. 32 corner-like points were tracked. The depth values of the points in the first frame ranged from 13 to 33 feet, therefore weak perspective does *not* provide a good approximation to this sequence. Moreover, a wide-lens camera was used, causing distortions at the periphery which were not compensated for. (See a more detailed description of a similar sequence in [25] Fig. 4, or [15] Fig. 3.)

Table 2 summarizes the results of our invariant algorithm for the last 8 points. Due to the noise in the data and the large perspective distortions, not all the frames were consistent with rigid motion. (Namely, when all the frames were used, the computed Gramian was not positive-definite). We therefore used only the last 8 frames from the available 16 frames.

Pt.	Pose	Rigid Invar.		Aff. Invar.	
#	Z	Z	Err (%)	Z	Err (%)
1	14.4	16.8	16.3	14.4	0.2
2	15.1	15.1	-0.0	15.1	-0.1
3	14.5	16.3	12.5	14.3	-1.1
4	13.5	16.0	18.4	12.3	-9.1
5	21.7	23.7	9.6	21.8	0.9
6	18.8	20.1	7.0	18.4	-2.3
7	21.5	20.7	-4.0	22.0	2.3
8	20.0	23.7	18.0	19.8	-1.3
9	21.6	21.5	-0.5	22.3	2.9
10	21.0	22.5	7.3	21.8	4.1
11	21.6	20.1	-7.0	22.7	4.9
12	21.0	21.0	0.3	22.2	6.0
ave.			8.4%		2.9%

Table 2: The relative errors in depth computation using our invariant algorithm, for affine and rigid shape.

We compared in Table 3 the average relative error of the results of our algorithm to the average relative error of a random set of $3D$ points, aligned to the ground truth data with the optimal similarity or affine transformation.

Rigid Invar.	Rigid random	Aff. Invar	Aff. random
8.4%	27.6%	2.9%	23.3%

Table 3: The mean relative errors in depth computation.

5.4 Discussion:

Not surprisingly, our results (Section 5.3 in particular) show that affine shape can be recovered more reliably than depth. We expect this to be the case since the computation of affine shape does not require knowledge of the aspect-ratio of the camera, and since it does not require the computation of the square root of the Gramian matrix G.

The results reported in Section 5.1 illustrate the importance of a good basis selection, the improved performance when more frames are used for the model acquisition, and the improved performance when the frames used for acquisition are well chosen (namely, spaced further apart). The only advantage for the simultaneous analysis of more than five points was the availability of a better (more independent) basis. A comparison with a SVD based algorithm for the computation of the decomposition in Eq. (7), which was described in [30], showed a comparable average performance of both our simple linear algorithm and the SVD based algorithm.

In our simulations of a recognition operator (Fig. 4), we see a marked difference between interpolation and extrapolation. More specifically, the discrimination ratio between images of the model and images of random objects is lower (and therefore better) for images that lie between images used for model acquisition (interpolation) than for images that lie outside the range of images used for model acquisition (extrapolation). We also see an average lower discrimination ratio for images used for model acquisition than for other images. This replicates the performance of human subjects in similar recognition tasks [4]. This demonstration is of particular interest since the results with human subjects were used to conclude that humans do not build an object-centered $3D$ representation, which is exactly what our algorithm is doing, but rather use a viewer-centered $2D$ representation (namely, storing a collection of $2D$ views of the object).

The sequence discussed in Section 5.2 was taken at a relatively large distance between the camera and the object (the depth values of the points varied from 550 to 700 mms). The weak perspective assumption therefore gave a good approximation. This sequence is typical of a recognition task. Under these conditions, which lend themselves favorably to the weak perspective approximation, our algorithm clearly performs very well.

The excellent results of all the algorithms with the box sequence are due in part to the reliable data, which was obtained by the particular tracking method described in [25]. Given this reliable correspondence, our algorithm gave the best results, although it relies on the weak perspective approximation. When compared with the other two algorithms, our algorithm is more efficient in its time complexity, it is simpler to implement, and it does not make any assumption on the type of motion (namely, it does not use the knowledge that the motion is rotational). A perspective projection algorithm should be used, however, if the scale of the object, or its actual distance from the camera, are sought.

The sequence discussed in Section 5.3 had very large perspective distortions (the depth values of the points varied from 13 to 33 feet). Moreover, the sequence was obtained with a wide-lens camera, which lead to distortions in the image coordinates of points at the periphery. This sequence is more typical of a navigation task. Under these conditions, which do not lend themselves favorably to the weak perspective approximation, our algorithm is not accurate. The accuracy is sufficient for tasks which require only relative depth (e.g., obstacle avoidance), or less precise reconstruction of the environment. Note, however, that even algorithms which use the perspective projection model do not necessarily perform better with such sequences (compare with the results for a similar sequence reported in [25]).

In this last sequence, the computation of invariant shape using 8 frames or 16 frames lead to rather similar results for the affine shape matrix and the Gramian matrix. However, in the second case the computed Gramian matrix was not positive-definite, and therefore we could not compute depth. This demonstrates how the computation of depth is more sensitive to errors than the computation of the similarity invariant representation. For the same reason, the affine reconstruction was an order of magnitude closer to the ground truth values than a set of random points, whereas the depth reconstruction had an average error only 3 times smaller than a set of random points.

6 Summary and Conclusions

We described a simple linear algorithm to compute similarity-invariant shape from motion, requiring only the closed-form solution of an over-determined linear system of equations. Unlike most algorithms, our algorithm computes shape without computing explicit depth or the transformation between images. Depth can be optionally obtained by computing the square root of a 3×3 matrix. Like [33] and unlike most algorithms, it can be implemented in an incremental way, updating the results with additional data without storing all the previous data. Finally, the algorithm is guaranteed to converge as the number of images grow, as long as the distribution of the noise in the images (including errors due to perspectivity) approaches a Gaussian distribution with mean value 0.

Our analysis shows that by computing an invariant representation, rather than depth, and by ignoring the transformation of the camera between different frames (or camera calibration), the problem becomes simpler. The non-invariant quantities (such as depth) can be later computed from the invariant representation, but they need not always be computed, e.g., they need not be computed at recognition. Thus computing an invariant representation directly promises to save computation time and to increase robustness.

Acknowledgements: We thank J. Inigo Thomas, who gave us two of the real data sets.

References

[1] J. Aloimonos. Perspective approximations. *Image and Vision Computing*, 8(3):179–192, August 1990.

[2] R. C. Bolles, H. H. Baker, and D. H. Marimont. Epipolar-plane image analysis: An approach to determining structure from motion. *International Journal of Computer Vision*, 1(1):7–55, 1987.

[3] T.J. Broida, S. Chandrashekhar, and R. Chellappa. Recursive 3-d motion estimation from a monocular image sequence. *IEEE Transactions on Aerospace and Electronic Systems*, 26(4):639–656, July 1990.

[4] H. H. Bülthoff and S. Edelman. Psychophysical support for a 2D interpolation theory of object recognition. *Proceedings of the National Academy of Science*, 89:60–64, 1992.

[5] S. Chandrasekaran and I. Ipsen. On rank-revealing QR factorizations. Research report YALEU/DCS/RR-880, Yale University, Computer Science, December 1991.

[6] C. H. Debrunner and N. Ahuja. A direct data approximation based motion estimation algorithm. In *Proceedings of the 10th International Conference on Pattern Recognition*, pages 384–389, Atlantic City, NJ, June 1990.

[7] O. Faugeras. What can be seen in three dimensions with an uncalibrated stereo rig? In *Proceedings of the 2nd European Conference on Computer Vision*, pages 563–578, Santa Margherita Ligure, Italy, 1992. Springer-Verlag.

[8] C. Fuh and P. Maragos. Motion displacement estimation using and affine model for matching. *Optical Engineering*, 30(7):881–887, July 1991.

[9] W. E. L. Grimson, D. P. Huttenlocher, and D. W. Jacobs. A study of affine matching with bounded sensor error. In G. Sandini, editor, *Computer Vision – ECCV92*, pages 291–306, Berlin, May 1992. Springer-Verlag.

[10] G. H. Golub and C. F. Van Loan. *Matrix Computations.* The Johns Hopkins University Press, Baltimore, Md, 1989.

[11] B. K. P. Horn, H. M. Hilden, and S. Negahdaripour. Closed-form solution of absolute orientation using orthonormal matrices. *Journal of the Optical Society of America A*, 5(7):1127–1135, July 1988.

[12] B. K. P. Horn. Relative orientation. *International Journal of Computer Vision*, 4(1):59–78, 1990.

[13] D. J. Heeger and A. Jepson. Visual perception of three-dimensional motion. Technical Report 124, MIT Media Laboratory, Cambridge, Ma, December 1989.

[14] R. Jain. Direct computation of the focus of expansion. *IEEE Pattern Analysis and Machine Intelligence*, 5:58–63, 1983.

[15] R. Kumar and A. R. Hanson. Sensitivity of the pose refinement problem to accurate estimation of camera parameters. In *Proceedings of the 3rd International Conference on Computer Vision*, pages 365–369, Osaka, Japan, 1990. IEEE, Washington, DC.

[16] J. J. Koenderink and A. J. van Doorn. Affine structure from motion. *Journal of the Optical Society of America*, 8(2):377–385, 1991.

[17] D. T. Lawton. Processing translational motion sequences. *Computer Graphics and Image Processing*, 22:116–144, 1983.

[18] H. C. Longuet-Higgins. A computer algorithm for reconstructing a scene from two projections. *Nature*, 293:133–135, September 1981.

[19] Y. Lamdan, J. T. Schwartz, and H. J. Wolfson. Affine invariant model-based object recognition. *IEEE Transactions on Robotics and Automation*, 6:578–589, 1990.

[20] P. S. Maybeck. *Stochastic Models, Estimation, and Control.* Academic Press, Orlando, Fl, 1979.

[21] L. Matthies, T. Kanade, and R. Szeliski. Kalman filter-based algorithms for estimating depth from image sequences. *International Journal of Computer Vision*, 3(3):209–236, September 1989.

[22] R. Mohr, , L. Quan, F. Veillon, and B. Boufama. Relative 3D reconstruction using multiple uncalibrated images. RT 84–IMAG–12 LIFIA, Institut IMAG, University of Grenoble, France, 1992.

[23] J. W. Roach and J. K. Aggarwal. Computer tracking of objects moving in space. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-1(2):127–135, April 1979.

[24] J. Rehg and A. Witkin. Visual tracking with deformation models. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 844–850, Sacramento, CA, April 1991.

[25] H. S. Sawhney, J. Oliensis, and A. R. Hanson. Description and reconstruction from image trajectories of rotational motion. In *Proceedings of the 3rd International Conference on Computer Vision*, pages 494–498, Osaka, Japan, 1990. IEEE, Washington, DC.

[26] M. E. Spetsakis and J. Y. Aloimonos. Optimal motion estimation. In *Proceedings of the IEEE Workshop on Visual Motion*, pages 229–237, Irvine, California, March 1989.

[27] R. Y. Tsai and T. S. Huang. Uniqueness and estimation of three-dimensional motion parameters of rigid objects with curved surfaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-6(1):13–27, January 1984.

[28] E. H. Thompson. A rational algebraic formulation of the problem of relative orientation. *Photogrammetric Record*, 3(14):152–159, 1959.

[29] C. Tomasi and T. Kanade. Shape and motion from image streams: a factorization method - 3. detection and tracking of point features. Technical Report CMU-CS-91-132, Carnegie Mellon University, Pittsburgh, PA, April 1991.

[30] C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: a factorization method. *International Journal of Computer Vision*, 9(2):137–154, 1992.

[31] S. Ullman and R. Basri. Recognition by linear combinations of models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13:992–1006, 1991.

[32] S. Ullman. *The Interpretation of Visual Motion.* The MIT Press, Cambridge, MA, 1979.

[33] S. Ullman. Maximizing rigidity: the incremental recovery of 3D structure from rigid and rubbery motion. *Perception*, 13:255–274, 1984.

[34] D. Weinshall. Model-based invariants for 3D vision. *International Journal on Computer Vision*, 10(1):27–42, 1993.