Model-based invariants for $3D$ vision

Daphna Weinshall*
IBM T.J. Watson Research Center, H1-B47
P.O.Box 704
Yorktown Heights, NY 10598

Abstract

Invariance under a group of $3D$ transformations seems a desirable component of an efficient $3D$ shape representation. We propose representations which are invariant under weak perspective to either rigid or affine $3D$ transformations, and we show how they can be computed efficiently from a sequence of images with a linear and incremental algorithm. We show simulated results with perspective projection and noise, and the results of model acquisition from a real sequence of images. The use of linear computation, together with the integration through time of invariant representations, offers improved robustness and stability. Using these invariant representations, we derive model-based projective invariant functions of general $3D$ objects. We discuss the use of the model-based invariants with existing recognition strategies: alignment without transformation, and constant time indexing from $2D$ images of general $3D$ objects.

1 Introduction

Efficient object representation is a key to a successful application of pattern recognition methods. Representations which are invariant to the actions of a group of transformations, describing any aspect of the imaging process, are of particular interest in vision. Such representations are not sensitive to changes in the appearance of objects, which are due solely to the imaging process and not to the quality of the objects. One group is given by the possible transformations of the camera, e.g., $3D$ rotations and translations. Other transformations describe the change of illumination.

In this paper we are interested in the group of $3D$ transformations. We describe a hierarchy of invariant representations, produced by the particular selection of the group of $3D$ transformations (cf. [11]):

Complete: The complete representation of a scene includes the $3D$ coordinates of each point, possibly as a depth map, and the pose of the camera relative to the object in each image. This representation is typically sought in reconstruction algorithms in computer vision.

Rigid invariant: A description of the $3D$ shape of objects which is invariant to the action of the group of similarity transformations in $3D$, which includes rotations, translations, and isotropic scaling. Rigid representations are unique.

*Present address: Department of Computer Science, The Hebrew University of Jerusalem, 91904 Jerusalem, Israel; email: daphna@cs.huji.ac.il

Affine invariant: A description of the $3D$ shape of objects which is invariant to the action of the group of linear transformations in $3D$ and translations. The group of affine transformations includes the group of similarity transformations. Affine representations are *not* unique.

A *complete* representation describes uniquely the object and its orientation relative to the camera. It includes the largest amount of information, and therefore not surprisingly it is the most difficult to compute. Moreover, it has been demonstrated many times that errors introduced in the pose computation, which is rather sensitive to noise, often lead to large errors in the shape computation. Nevertheless, most Structure From Motion algorithms attempt to compute a *complete* representation of the scene, namely, structure in the form of a depth map and pose, using a sequence of $2D$ images (see, for example, [18, 7]).

The *invariant* representations do not require the computation of camera calibration (or pose), and therefore they should be easier to compute robustly. Moreover, an invariant representation is a natural frame of reference for the integration of information across time. Therefore when the pose information is not needed, which is typically the case in recognition tasks, the use of invariant representations promises significant computational advantages. *Affine* invariant representations in particular, and their computational advantages, were discussed in [11]. Affine representations were initially used for recognition applications, in geometric hashing [12] and linear combination [24]. The cost of using an affine representation is more false positive matches, since affine representations are not unique.

$3D$ shape representations, which are invariant to projective $3D$ transformations, are of particular interest for the recognition of $3D$ objects from $2D$ images. Invariance with respect to projective transformation is harder to accomplish. Recently, Burns et al. [2] (as well as Clemens & Jacobs [3] and Moses & Ullman [16]) have shown that unconstrained projective invariant functions do not exist. Most of the research in the area of projective invariance has concentrated on the identification of computable invariances in some special cases [26]. In fact, most of the published results address planar collections of points or curves. One elegant example of the use of such invariants, e.g. pairs of plane conics, has been recently given in [5]. More generally, Moses & Ullman [16] studied the existence of projective invariants for specific classes of objects, such as bilaterally symmetric objects.

In this paper we describe a particular hierarchy of affine and rigid invariant representations. We describe the transformations between different representations, and show some object symmetries that are readily mediated by these representations. The justification for these representations is computational, both in model acquisition and model-based recognition:

Model acquisition: To compute these representations we discuss a hierarchical shape from motion algorithm which, given correspondence, requires solving two linear systems of equations, one for affine shape and one for rigid shape. The algorithm requires at least three frames for the rigid shape, and at least two frames for the affine shape. Additional frames are incorporated into the linear systems to increase robustness. This algorithm appears simpler and easier to use in an incremental way (as additional data is accumulated) than existing structure from motion algorithms. Moreover, the computation of the "usual" depth map and pose can be obtained with an additional, almost trivial, non-linear step.

Model-based recognition: Since unconstrained projective invariants do not exist, we study the existence of model-based projective invariants. Using our hierarchy of invariant representations, we describe two model-based projective invariant functions for any non-planar $3D$ object. It immediately follows that a projective invariant function can be defined for any finite set of $3D$ objects. The computational complexity of these invariant functions depends linearly on the number of objects. We also show how model-based invariant functions can be used for recognition with constant computational complexity, providing a constant time index into a lookup table (cf. [12]). The cost is the need to use large multi-dimensional lookup tables (two- and four-dimensional in our examples) instead of a one-dimensional table.

The rest of this paper is organized as follows: in Section 2 we describe affine and rigid invariant representations of $3D$ shape. In Section 3 we describe affine and rigid model-based invariant functions. In Section 4 we discuss model acquisition, namely, the linear computation of invariant shape from motion. In Section 5 we apply the model-based invariant functions to existing model-based recognition techniques: we describe alignment without transformation, and the generalization of geometric hashing to $2D$ views of general $3D$ objects.

2 Invariant representations of $3D$ shape

We first describe a hierarchy of invariant representations which can be computed hierarchically with a linear algorithm, introducing an affine-invariant representation of objects in Section 2.1.1 and a rigid-invariant representation in Section 2.1.2. We discuss the linear computation of these representations in Section 2.2. Alternative representations, and the transformation between different representations, are described in Section 2.3. Finally we describe some symmetries, which the rigid invariant representation readily reveals, in Section 2.4.

2.1 Definitions:

Let $\mathcal{D} : \Omega \rightarrow \mathcal{R}^n$ denote a representation function from the space of objects to $\mathcal{R}^n$, where for every object $\omega \in \Omega$, $\mathcal{D}(\omega)$ is a description of the $3D$ shape of the object. Let $\mathcal{G}$ be a group of transformations acting on the space of objects Ω such that an element $g \in \mathcal{G}$ is a transformation $g : \Omega \rightarrow \Omega$, together with the composition of transformations as the product rule. A representation function $\mathcal{D}$ is invariant with respect to the group of transformations $\mathcal{G}$ if $\mathcal{D}(g(\omega)) = \mathcal{D}(\omega), \forall g \in \mathcal{G}$.

Generally, the camera's motion is characterized by the rigid group of transformations $\mathcal{G}_{rig}$, which includes $3D$ rotations and translations. When weak perspective is assumed, $\mathcal{G}_{rig}$ is the similarity group, which includes rotations, translations, and isotropic scaling. The weak perspective approximation, which is assumed in this paper, is valid when the relative distances between points in the object are much smaller than their distances to the camera.

A representation function $\mathcal{D}$, which is invariant to $\mathcal{G}_{rig}$, will be called a **rigid-invariant representation**. During recognition, the camera's position with respect to objects is usually unknown. A rigid-invariant representation is therefore desirable since it eliminates the need to compute the orientation of the object.

For computational convenience, we also consider the group of transformations $\mathcal{G}_{aff}$, which includes $3D$ translations and linear (or affine) transformations [11]. A representation function $\mathcal{D}$, which is invariant to $\mathcal{G}_{aff}$, will be called an **affine-invariant representation.** Affine representations cannot be unique: objects which are related by a linear transformation will have the same affine representation. Note that an affine-invariant representation is also rigid-invariant.

Let $\langle P \rangle = \{P_l\}_{l=0}^{n}$, $P_l \in \mathcal{R}^3$, denote the $3D$ coordinates of an object ω composed of $n+1$ features. To account for translations, we fix the origin at point $P_0 = (0,0,0)$. Let $\{\mathbf{p}_l\}_{l=1}^{n}$ denote the $3D$ vectors corresponding to the remaining n points.

2.1.1 Affine shape

Let $\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3$ denote three independent vectors in $\mathcal{R}^3$. The vectors $\{\mathbf{v}_l\}_{l=1}^{3}$ are the basis of an affine coordinate system $\mathcal{V}$ (a system whose basis is not necessarily orthonormal). Every vector $\mathbf{p}_l \in \mathcal{R}^3$ can be written as a linear combination of this basis:

$$\mathbf{p}_l = b_1^l \mathbf{v}_1 + b_2^l \mathbf{v}_2 + b_3^l \mathbf{v}_3$$

The vector $\mathbf{b}^l = (b_1^l, b_2^l, b_3^l)$ is an affine invariant of point P_l (a related representation is discussed in [12]). Given a particular basis of three independent vectors $\{\mathbf{v}_l\}_{l=1}^{3}$ defining the coordinate system $\mathcal{V}$, the affine coordinates $\{\mathbf{b}^l\}_{l=1}^{n}$ of the set of points $\langle P \rangle$ are unique, providing an affine-invariant representation of the object.

Since the above affine-invariant representation depends on the basis, and in order to obtain a representation which depends only on the object, we choose the basis from the set of vectors $\{\mathbf{p}_l\}_{l=1}^{n}$. Let $\mathcal{P}$ denote the affine coordinate system whose basis is $\mathbf{p}_i, \mathbf{p}_j, \mathbf{p}_k$. We have:

$$\mathbf{p}_l = b_1^l \mathbf{p}_i + b_2^l \mathbf{p}_j + b_3^l \mathbf{p}_k, \quad \forall l \tag{1}$$

Note that in $\mathcal{P}$ $\mathbf{b}^i = (1,0,0)$, $\mathbf{b}^j = (0,1,0)$, and $\mathbf{b}^k = (0,0,1)$.

$\mathcal{D}_{aff} = \{\mathbf{b}^l\}_{l=1}^{n}$ is an affine-invariant representation of object ω since any linear transformation of the coordinate system does not change the values of the coefficients $\mathbf{b}^l$ in Eq (1). Because of the invariance to linear transformations, the computation of $\mathcal{D}_{aff}$ does not require knowledge of the aspect-ratio of the camera. $\mathbf{b}$ can be computed from (at least) two images (or one image and a perpendicular optical flow) by solving a linear system of equations. $\mathbf{b}$ does not contain $3D$ information on the five points (such as depth).

2.1.2 Rigid shape

We represent the Euclidean metric information on the basis points using their inverse Gramian matrix $\mathbf{B} = \mathbf{G}^{-1}$, which is invariant to rotations of the coordinate system but not to general linear transformations. The Gramian of the basis points P_i, P_j, P_k is the following 3×3 symmetric matrix $\mathbf{G}$:

$$\mathbf{G} = \begin{pmatrix} \mathbf{p}_i^T \mathbf{p}_i & \mathbf{p}_i^T \mathbf{p}_j & \mathbf{p}_i^T \mathbf{p}_k \\ \mathbf{p}_i^T \mathbf{p}_j & \mathbf{p}_j^T \mathbf{p}_j & \mathbf{p}_j^T \mathbf{p}_k \\ \mathbf{p}_i^T \mathbf{p}_k & \mathbf{p}_j^T \mathbf{p}_k & \mathbf{p}_k^T \mathbf{p}_k \end{pmatrix} \tag{2}$$

The elements of $\mathbf{G}$ contain all the $3D$ information on the geometry of the four basis points, such as angles and length. Thus, given a particular choice of coordinate system, the depth of all the points can be computed from $\mathbf{G}$ (see discussion in Section 2.3.2).

$\mathcal{D}_{rig} = [\mathbf{B}, \mathcal{D}_{aff}]$ is a hierarchical rigid-invariant representation of object ω. The computation of $\mathbf{B}$ (and $\mathcal{D}_{aff}$) is linear and incremental, requiring at least three images, as described in Section 2.2.

2.2 Computation

2.2.1 Affine shape:

Given the image coordinates of five points (x_0, y_0), (x_1, y_1), (x_2, y_2), (x_3, y_3), (x_4, y_4), assume without loss of generality that $x_0 = 0, y_0 = 0$, namely, the origin is defined to be at point 0. (Otherwise, replace any x_l in the following expressions by $x_l - x_0$, and y_l by $y_l - y_0$.) From the definition of $\mathbf{b}$, it follows that in any image:

$$x_4 = \sum_{l=1}^{3} b_l x_l, \quad \text{and} \quad y_4 = \sum_{l=1}^{3} b_l y_l. \tag{3}$$

It follows that each image gives two equations that are linear in the elements of $\mathbf{b}$. There are three unknown elements in $\mathbf{b}$, thus we need three equations, or at least two views, to compute $\mathbf{b}$. (Note that the computations of $\mathbf{b}$ does not require knowledge of the aspect-ratio of the camera.)

Two views give us 4 equations, 1 too many. Indeed, it can be readily shown that it is sufficient to use one view and the perpendicular component of the $2D$ motion field at each point. (This result has also been shown by Shashua in [19].) This is useful when the aperture problem constrains the computation of the $2D$ motion field, and for the generalization of this analysis to continuous optical flow.

More specifically, if the perpendicular optical flow vector at point (x_l, y_l) is $(x'_l - x_l, y'_l - y_l)$, and let m_l denote the slope of the line perpendicular to $(x'_l - x_l, y'_l - y_l)$, then $\mathbf{b}$ can be computed from the two equations in (3) and the following equation:

$$y'_4 + m_4 x'_4 = \sum_{l=1}^{3} b_l (y'_l + m_l x'_l).$$

2.2.2 Rigid shape:

Given the image coordinates of P_0 and the three basis points (x_0, y_0), (x_i, y_i), (x_j, y_j), and (x_k, y_k), assume without loss of generality $x_0 = 0, y_0 = 0$. (Otherwise, replace any x_l in the following expressions by $x_l - x_0$, and y_l by $y_l - y_0$.) Let $\mathbf{x} = (x_i, x_j, x_k)$ and $\mathbf{y} = (y_i, y_j, y_k)$.

By fixing the origin in the first point $P_0 = (0, 0, 0)$, we account for the translation and can ignore it in further discussion. Let $P_i = (X_i, Y_i, Z_i)$, $P_j = (X_j, Y_j, Z_j)$, and $P_k = (X_k, Y_k, Z_k)$ denote the $3D$ coordinates of the three basis points in some fixed frame of reference with origin at P_0. Any image is the orthographic projection of the object rotated from this absolute frame by a rotation matrix $\mathbf{R} = \{r_{m,n}\}_{m,n=1}^{3}$ and scaled by s.

First note that:

$$\begin{pmatrix} X_i & Y_i & Z_i \\ X_j & Y_j & Z_j \\ X_k & Y_k & Z_k \end{pmatrix} \begin{pmatrix} sr_{11} \\ sr_{12} \\ sr_{13} \end{pmatrix} = \begin{pmatrix} x_i \\ x_j \\ x_k \end{pmatrix}, \quad \begin{pmatrix} X_i & Y_i & Z_i \\ X_j & Y_j & Z_j \\ X_k & Y_k & Z_k \end{pmatrix} \begin{pmatrix} sr_{21} \\ sr_{22} \\ sr_{23} \end{pmatrix} = \begin{pmatrix} y_i \\ y_j \\ y_k \end{pmatrix}.$$

Using the notation

$$\mathbf{A} = \begin{pmatrix} X_i & Y_i & Z_i \\ X_j & Y_j & Z_j \\ X_k & Y_k & Z_k \end{pmatrix}, \quad \mathbf{r}_1 = \begin{pmatrix} r_{11} \\ r_{12} \\ r_{13} \end{pmatrix}, \quad \text{and } \mathbf{r}_2 = \begin{pmatrix} r_{21} \\ r_{22} \\ r_{23} \end{pmatrix} \tag{4}$$

we can rewrite the above equations as follows:

$$\mathbf{r}_1 = \frac{1}{s}\mathbf{A}^{-1}\mathbf{x} , \qquad \mathbf{r}_2 = \frac{1}{s}\mathbf{A}^{-1}\mathbf{y} . \tag{5}$$

From the orthonormality of the rotation matrix we have:

$$\mathbf{x}^T(\mathbf{A}\mathbf{A}^T)^{-1}\mathbf{y} = 0 , \qquad \mathbf{x}^T(\mathbf{A}\mathbf{A}^T)^{-1}\mathbf{x} - \mathbf{y}^T(\mathbf{A}\mathbf{A}^T)^{-1}\mathbf{y} = 0 . \tag{6}$$

We denote $\mathbf{B} = (\mathbf{A}\mathbf{A}^T)^{-1}$, and rewrite $\mathbf{B}$ as follows:

$$\mathbf{B}^{-1} = \begin{pmatrix} X_i & Y_i & Z_i \\ X_j & Y_j & Z_j \\ X_k & Y_k & Z_k \end{pmatrix} \begin{pmatrix} X_i & X_j & X_k \\ Y_i & Y_j & Y_k \\ Z_i & Z_j & Z_k \end{pmatrix} = \begin{pmatrix} \mathbf{p}_i^T\mathbf{p}_i & \mathbf{p}_i^T\mathbf{p}_j & \mathbf{p}_i^T\mathbf{p}_k \\ \mathbf{p}_i^T\mathbf{p}_j & \mathbf{p}_j^T\mathbf{p}_j & \mathbf{p}_j^T\mathbf{p}_k \\ \mathbf{p}_i^T\mathbf{p}_k & \mathbf{p}_j^T\mathbf{p}_k & \mathbf{p}_k^T\mathbf{p}_k \end{pmatrix} \tag{7}$$

$\mathbf{B}$ is the inverse Gramian of the basis points. It follows from Eq (6) that:

$$\mathbf{x}^T\mathbf{B}\mathbf{y} = 0, \quad \text{and} \quad \mathbf{x}^T\mathbf{B}\mathbf{x} - \mathbf{y}^T\mathbf{B}\mathbf{y} = 0. \tag{8}$$

Note that we could have started the derivation from any initial $3D$ configuration of the four points, rotated from the original configuration by matrix $\mathbf{R}'$. Let $\mathbf{A}'$ denote another initial configuration, where $\mathbf{A}' = \mathbf{A}\mathbf{R}'$. It follows that $\mathbf{B}'^{-1} = \mathbf{A}'\mathbf{A}'^T = \mathbf{A}\mathbf{R}'\mathbf{R}'^T\mathbf{A}^T = \mathbf{A}\mathbf{A}^T = \mathbf{B}^{-1}$. Thus $\mathbf{B}$ is invariant of the particular viewpoint, or the orientation of the initial coordinate system. The invariance of $\mathbf{B}$ makes it possible to compute it from a few images, and to use it in a structure from motion algorithm. (Note that the computation of $\mathbf{B}$ requires knowledge of the aspect-ratio of the camera.)

The equations in (8), which are linear in $\mathbf{B}$, can be used to compute $\mathbf{B}$ directly from images by solving an overdetermined linear system of equations, as will be discussed in detail in Section 4.1. This computation requires correspondence and the use of at least three images. Since the linear system in Eq (8) is homogeneous, $\mathbf{B}$ is only determined up to a scaling factor. Therefore there are only five independent unknown elements in Eq (8).

From Eq (8) it also follows that each image gives two linear equations in the elements of $\mathbf{B}$. Since there are five independent unknowns, we need at least three views to compute $\mathbf{B}$[1]. This is the theoretically minimal information required for structure from motion [22, 1]. Note, however, that three views give us six equations, enough to compute $\mathbf{B}$ and to verify that the three views of the four points indeed come from a rigid object and a rigid transformation. Thus we extend the minimal structure from motion theorem to the following:

Extended minimal structure from motion theorem: *Assuming weak perspective projection, three views of four points give enough equations to compute the structure of the points and to verify that they are moving rigidly.*

[1]Under orthographic projection without scale, three views are still needed, as is shown in the appendix.

2.3 Alternative representations

We first describe the transformation between two representations of the type $\mathcal{D}_{rig}$, with different basis points (Section 2.3.1). We then describe the transformation from any basis to an orthonormal basis (Section 2.3.2). Finally, we describe alternative invariant representations to $\mathcal{D}_{rig}$ (Section 2.3.3).

2.3.1 Change of basis:

Let $\mathbf{B}$ denote the 3×3 Gramian matrix of the four original basis points, and let $\mathbf{b}^s$ denote the affine coordinates of point P_s in the original basis. Let $\mathbf{D}$ denote the Gramian matrix of another basis P_0, P_l, P_m, P_n, and let $\mathbf{d}^s$ denote the affine coordinates of point P_s in the new basis. Let $\mathbf{S} = (\, \mathbf{b}^l \quad \mathbf{b}^m \quad \mathbf{b}^n \,)^T$ denote the 3×3 matrix whose rows are the affine coordinates of the new basis points in the original basis. It can be readily shown that:

$$\begin{aligned} \mathbf{D} &= (\mathbf{S}^{-1})^T \mathbf{B} \mathbf{S}^{-1} \\ \mathbf{d}^s &= (\mathbf{S}^{-1})^T \mathbf{b}^s \end{aligned}$$

2.3.2 Change to orthonormal basis, or depth:

It may be necessary to represent points or surfaces by their coordinates in a Cartesian coordinate system (i.e., a coordinate system with orthonormal axes). This may be useful when differential properties of surfaces, such as the Gaussian curvature, are required. We therefore define an invariant Cartesian coordinate system $\mathcal{Q} = \{\mathbf{X}, \mathbf{Y}, \mathbf{Z}\}$ with orthonormal axes. Let $\mathcal{P}$ denote the affine invariant coordinate system defined in Section 2.1.1. We obtain $\mathcal{Q}$ by orthonormalizing the basis of $\mathcal{P}$, in a similar way to Gram-Schmidt: we choose $\mathbf{X}$ to parallel $\mathbf{p}_i$, $\mathbf{Y} \in span\{\mathbf{p}_i, \mathbf{p}_j\}$ is perpendicular to $\mathbf{X}$, and $\mathbf{Z}$ is perpendicular to the $\mathbf{X} - \mathbf{Y}$ plane.

Let $\mathbf{q}_l$ denote the coordinates of point P_l in system $\mathcal{Q}$. Let $\mathbf{Q}$ denote the upper triangular 3×3 matrix whose columns are $\mathbf{q}_i, \mathbf{q}_j, \mathbf{q}_k$, the coordinates of the basis points. It follows from the definition in Eq (1) that $\mathbf{q}_l = \mathbf{Q}\mathbf{b}^l$, $\forall l$. Since any rigid transformation of the coordinate system does not change the values of $\{\mathbf{q}_l\}_{l=1}^n$, $\mathcal{D}_{cart} = \{\mathbf{Q}\mathbf{b}^l\}_{l=1}^n$ is a rigid-invariant representation of object ω. $\mathcal{D}_{cart}$ is equivalent to a depth map.

It follows from Eq (2) that $\mathbf{B}^{-1} = \mathbf{Q}^T\mathbf{Q}$. $\mathbf{Q}$, which is the root of $\mathbf{B}^{-1}$, can be easily computed using a decomposition known as **Choleski** factorization. Since $\mathbf{B}$ is positive definite, the computation of $\mathbf{Q}$ from $\mathbf{B}$ is straightforward and very fast. Thus the following lemma immediately follows:

Lemma 1: *[Transformation between coordinate systems:] Given a 3D point whose coordinates in the affine system $\mathcal{P}$ are $\mathbf{b} = (b_1, b_2, b_3)$, its coordinates in the Cartesian system $\mathcal{Q}$ are $(X, Y, Z) = \mathbf{Q}\mathbf{b}$, where $\mathbf{B}^{-1} = \mathbf{Q}^T\mathbf{Q}$.*

The transformation to the new orthonormal coordinate system requires a non-linear step, taking the root (or Cholesky decomposition) of the original Gramian $\mathbf{B}^{-1}$. Although this root is simple to compute, it only exists for positive-definite matrices. When the inverse Gramian $\mathbf{B}$ is computed from noisy data, it may not turn out to be positive definite, and the root may not exist. Thus it is computationally more robust to use the rigid invariant representation $\mathcal{D}_{rig}$ rather than a depth map representation such as $\mathcal{D}_{cart}$.

2.3.3 Other representations:

It is possible to represent the set of points $\langle P \rangle$ by the set of invariant representations of subsets of four points, thus capturing only local properties of the object. This can be implemented by storing up to $O(n^4)$ 3×3 matrices (see [14]).

It is also possible to represent $\langle P \rangle$ by a $n \times n$ symmetric matrix $\tilde{\mathbf{B}}$, the extension of the inverse Gramian $\mathbf{B}$ to $n+1$ points. Given four points, matrix $\mathbf{B}$ was defined above, using the 3×3 coordinates matrix $\mathbf{A}$ given in Eq (4), as $\mathbf{B} = (\mathbf{A}^{-1})^T \mathbf{A}^{-1}$. Given $n+1$ points, and a $n \times 3$ coordinate matrix $\tilde{\mathbf{A}}$, we define[2] $\tilde{\mathbf{B}} = (\tilde{\mathbf{A}}^{+})^T \tilde{\mathbf{A}}^{+}$. Applying similar considerations, it can be readily shown that the quadratic invariant equations in Eq (8) hold for $\tilde{\mathbf{B}}$ as well.

$\tilde{\mathbf{B}}$ is of rank 3, and therefore cannot be computed directly from images. It can, however, be obtained from $\mathcal{D}_{rig}$ directly, since, if we let $\mathbf{S}$ denote the matrix whose l-th row is $\mathbf{b}^l$:

$$\tilde{\mathbf{B}} = (\mathbf{S}^{+})^T \mathbf{B} \mathbf{S}^{+}$$

2.4 Symmetry:

In $\mathcal{D}_{rig}$ we represent the $3D$ geometrical information on four non-coplanar points by their inverse-Gramian matrix $\mathbf{B}$. Since $\mathbf{B}$ contains $3D$ Euclidean invariant information on the four points, some symmetries in the $3D$ configuration are immediately apparent from it. For example, if points 1,2 (e.g., two eyes) have a reflection (bilateral) symmetry with respect to the line going through points 0,3 (e.g., the nose and mouth), then $\mathbf{B}$ shows this symmetry as $\mathbf{B}_{13} = \mathbf{B}_{23}$ and $\mathbf{B}_{11} = \mathbf{B}_{22}$. If the points' configuration is a box with sides u, v, w, then

$$\mathbf{B} \propto \begin{pmatrix} \frac{1}{u^2} & 0 & 0 \\ 0 & \frac{1}{v^2} & 0 \\ 0 & 0 & \frac{1}{w^2} \end{pmatrix}$$

Thus $\mathbf{B}$ can be used to classify $3D$ objects by their symmetries.

On the other hand, a known symmetry of the object can reduce the amount of images required to compute $\mathbf{B}$. For example, with known bilateral symmetry there are only four unknown elements in $\mathbf{B}$, thus two views are sufficient to compute the rigid structure of the four points and to verify their rigidity. This complements the results discussed in [17, 13], where it was shown that a single view of a bilaterally symmetric object is equivalent to having two views of the object.

3 Model Based Invariant functions

We now define model-based projective invariant functions (Section 3.1), and describe a rigid and an affine model-based invariant functions (Section 3.2).

[2] $\tilde{\mathbf{A}}^{+} = (\tilde{\mathbf{A}}^T \tilde{\mathbf{A}})^{-1} \tilde{\mathbf{A}}^T$ denotes the pseudo-inverse of $\tilde{\mathbf{A}}$.

3.1 Definitions:

Let an object be a set of n points in $\mathcal{R}^3$, to be denoted $\omega = \{P_l\}_{l=1}^n$, $\omega \in \Omega$. The image of the object is a set of n $2D$ points $\{p_l\}_{l=1}^n$ (disregarding occlusion), produced by a $3D$ transformation $g \in \mathcal{G}$ of the object, and followed by a projection π which is a $3D$ to $2D$ mapping:

$$p_l = \pi(g(P_l)) \qquad p_l \in \mathcal{R}^2, \quad \pi : \mathcal{R}^3 \to \mathcal{R}^2, \quad g \in \mathcal{G}, \quad P_l \in \mathcal{R}^3$$

A **projective view-invariant function** $f : \pi(\omega) \to \mathcal{R}$ with respect to some group of transformations $\mathcal{G}$ is a real function on the viewed set of points $\{p_l\}$, which has the property that it is invariant to the action of the group $\mathcal{G}$, namely, $f(\pi(g(\omega))) = f(\pi(\omega))$. More precisely, it is the projection of a scalar invariant description $\mathcal{D}$ on the $3D$ points $\{P_l\}_{l=1}^n$. f is a **universal invariant function** if there is no restriction on the domain Ω, the set of possible objects (namely, $\Omega = \mathcal{R}^{3n}$).

More precise definitions, as well as the enumeration of the number of invariants under different transformation groups $\mathcal{G}$ and object sets Ω, are given in [5, 6]. As discussed in the introduction, a few recent papers showed that **universal invariant functions** (which are not constant) do not exist [2, 3, 16] for perspective and weak perspective projections. Special-case invariants for special sets of objects Ω are discussed in [16, 5], for example.

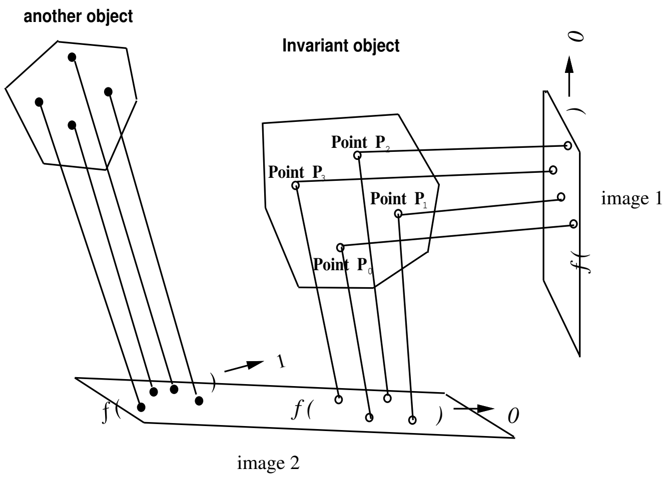

Figure 1: An illustration of a **model-based invariant** function whose invariant value is 0, $n = 4$.

A projective invariant function f is **model-based invariant** if its domain of objects is a single general $3D$ object, namely, $\Omega = \{\omega\}$, $\omega \in \mathcal{R}^{3n}$ (see illustration in Fig. 1). The parameters of the invariant function depend on the choice of object (thus it is model-based). A **composite model-based invariant** function returns a different view-independent value for each object in a set Ω which contains a finite number of general $3D$ objects. (A composite model-based invariant function for a set of objects is like a discriminant over that set, with properties undefined outside the set.)

Let $\mathcal{G}$ be the similarity group (rigid transformations and scale), and let f be a model-based invariant function for $\Omega = \{\omega\}$. It is possible that for another object ω', $f(\pi(\omega)) = f(\pi(g(\omega')))$ for some transformation g. Such false identifications are called **accidental matches**. If, however,

$f(\pi(\omega)) = f(\pi(g(\omega')))$ for all $g \in \mathcal{G}$, these false identifications are called **non-accidental matches**. It follows from the definition that affine invariants lead to non-accidental matches.

3.2 Examples of model based invariant functions:

Rigid invariant:

Consider an object composed of four non-coplanar $3D$ points. (If all points are coplanar, there exist a few other projective invariants, e.g., the affine invariant discussed in [12]). Let $\{P_l\}_{l=0}^3$ denote the $3D$ coordinates of the four points in some frame of reference. Assume $P_0 = (0,0,0)$ without loss of generality, and let $\{\mathbf{p}_l\}_{l=1}^3$ denote the $3D$ vectors corresponding to the three remaining points (see illustration in Fig. 2). Let $\mathbf{B}$ denote the inverse Gramian matrix of the four points, defined in Eq (2).

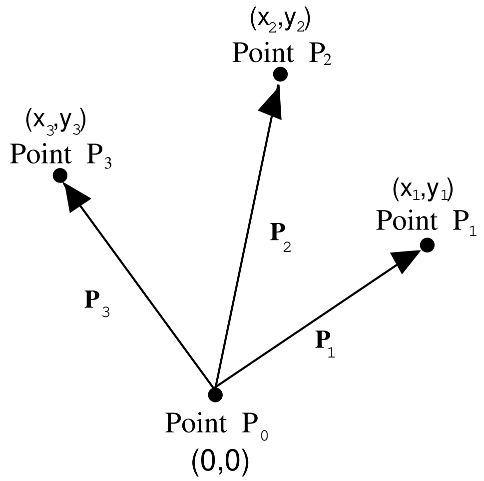

Figure 2: Four $3D$ points P_l, whose image coordinates are (x_l, y_l).

Given the image coordinates of the four points (x_0, y_0), (x_1, y_1), (x_2, y_2), (x_3, y_3), assume without loss of generality $x_0 = 0, y_0 = 0$. (Otherwise, replace any x_l in the following expressions by $x_l - x_0$, and y_l by $y_l - y_0$.) Let $\mathbf{x} = (x_1, x_2, x_3)$ and $\mathbf{y} = (y_1, y_2, y_3)$.

Definition 1 (Quadratic) *Given four non-coplanar $3D$ points, let*

$$f_{\mathbf{B}}(\mathbf{x}, \mathbf{y}) = \frac{|\mathbf{x}^T\mathbf{B}\mathbf{y}| + |\mathbf{x}^T\mathbf{B}\mathbf{x} - \mathbf{y}^T\mathbf{B}\mathbf{y}|}{|\mathbf{x}| \parallel \mathbf{B} \parallel |\mathbf{y}|} \tag{9}$$

$f_{\mathbf{B}}$ is a rigid model-based projective invariant function, which allows accidental matches only. $f_{\mathbf{B}}$ returns 0 for all the views of a single object, and possibly for accidental views of other objects.

The model-based invariance of $f_{\mathbf{B}}$ follows from Eq (8), since $f_{\mathbf{B}}(\mathbf{x}, \mathbf{y}) = 0$ for any image of the object described by $\mathbf{B}$. $f_{\mathbf{B}}$ is normalized, so that its value does not depend on the distance from the camera to the object (or the scale factor). The quadratic invariant function may operate on less than four points if the transformation of the object is restricted. For example, the $3D$ rotations group requires only three points.

The equations in (8) correspond to two orthonormality constraints on the columns of the rotation matrix $\mathbf{R}$ (see Section 2.2.2). For $\mathbf{r}_3 = (r_{31}, r_{32}, r_{33})^T$, the constraints $\mathbf{r}_3^T\mathbf{r}_3 = 1$, $\mathbf{r}_3^T\mathbf{r}_2 = 0$, and $\mathbf{r}_3^T\mathbf{r}_1 = 0$ cannot be verified from a single $2D$ image. This explains why accidental matches, where the invariant function defined in Eq (9) returns 0 for isolated views of different objects, are possible. Non-accidental matches are not possible. This is guaranteed by the satisfaction of a few of the orthonormality constraints.

Affine invariant:

Consider an object composed of (at least) five $3D$ points, and assume w.l.g. that the first four points are not coplanar. Let $\{P_l\}_{l=0}^4$ denote the $3D$ coordinates of the five points in some frame of reference. Assume $P_0 = (0, 0, 0)$ without loss of generality, and let $\{\mathbf{p}_l\}_{l=1}^4$ denote the $3D$ vectors corresponding to the four remaining points. Let $\mathbf{b} = (b_1, b_2, b_3)$ denote the affine invariant vector representation of the five points, as defined in Eq (1).

Given the image coordinates of the five points $(x_0, y_0), (x_1, y_1), (x_2, y_2), (x_3, y_3), (x_4, y_4)$, assume without loss of generality that $x_0 = 0, y_0 = 0$, namely, the origin is defined to be at point 0. (Otherwise, replace any x_l in the following expressions by $x_l - x_0$, and y_l by $y_l - y_0$.) Let $\mathbf{x} = (x_1, x_2, x_3)$ and $\mathbf{y} = (y_1, y_2, y_3)$.

Definition 2 (Linear) *Given five non-coplanar* $3D$ *points, let*

$$f_{\mathbf{b}}(\mathbf{x}, \mathbf{y}) = \frac{|x_4 - \sum_{i=1}^3 b_i x_i|}{|\mathbf{x}||\mathbf{b}|} + \frac{|y_4 - \sum_{i=1}^3 b_i y_i|}{|\mathbf{y}||\mathbf{b}|}. \tag{10}$$

$f_{\mathbf{b}}$ *is an affine model-based projective invariant function for the five points.* $f_{\mathbf{b}}$ *returns 0 for non-accidental views of a family of related objects.*

The model-based invariance of $f_{\mathbf{b}}$ follows from Eq (3), since $f_{\mathbf{b}}(\mathbf{x}, \mathbf{y}) = 0$ for any image of the object described by $\mathbf{b}$. $f_{\mathbf{b}}$ is normalized, so that its value does not depend on the distance from the camera to the object (or the scale factor). The linear invariant function may operate on less than five points if the transformation of the object is restricted. For example, the $3D$ rotations group requires only four points.

A composite invariant:

Given $M > 1$ objects with four to n points, let $f_j(\mathbf{x}, \mathbf{y})$ denote the model-based invariant function of the $j-$th object. Depending on the value of n and other preferences, $f_j()$ can be one of the functions given in Eq (9) or Eq (10), or a combination of them. Let ϵ denote a small threshold, the maximal allowed deviation from 0 of the model-based invariant functions due to expected noise. A function that returns j given object j, and 0 otherwise, is the following variation on a winner-takes-all function:

$$F(f_1, ..., f_M) = \begin{cases} j & f_j < \epsilon \text{ and } f_j \leq f_k \ \forall\, 1 \leq k \leq M \\ 0 & \text{otherwise} \end{cases} \tag{11}$$

Result 1 (Model-based invariants) *For every M objects with four to n $3D$ points, there exists a composite model-based invariant function, given in Eq (11). The time complexity of the evaluation of this function is $O(M)$.*

4 Model acquisition

Invariant representations make it possible to integrate information across different frames in a natural way. This integration offers robustness and stability in building the $3D$ description of objects from multiple images. In Section 4.1 we use this property to outline a new structure from motion algorithm. In Section 4.2 we show results with simulated and real data. A more complete and efficient algorithm is described in [25], with additional tests and a quantitative comparison to other algorithms.

4.1 Linear structure from motion algorithm given correspondence

We have shown that the rigid representation $\mathcal{D}_{rig}$ can be computed from as few as three images of four points with a linear algorithm. We have also shown in Section 2.3.2 how to obtain the depth map representation $\mathcal{D}_{cart}$ from $\mathcal{D}_{rig}$. We will now use these results to describe a linear structure from motion algorithm for n points from $m \geq 3$ views. This algorithm assumes weak perspective, of which orthographic projection is a special case, and it requires correspondence

The proposed algorithm differs from other SFM algorithms in that it does not compute structure in the usual sense, and it does not compute the transformation between the images. Thus it is a shape from motion algorithm without depth and without transformation. Instead, the algorithm computes an invariant $3D$ representation of the object. Depth can be optionally computed from this representation by computing the square root of a 3×3 matrix.

The point to be made here is that by computing an invariant representation, rather than depth, and by not computing the motion of the camera, the problem becomes simpler and linear. The non-invariant quantities (such as depth and transformation) can be later computed from the invariant function, but they need not always be computed, e.g. they need not be computed at recognition. This direct approach may therefore save computation time and increase robustness.

The algorithm:

Initialization: Select the origin P_0 and subtract its coordinates from the image measurements of all the other points. Select three independent basis points.

Affine: For all but the basis points, compute their affine representation $\mathbf{b}$ by solving the overdetermined linear system defined in Eq (3).

Rigid: For the three basis points, compute their inverse Gramian $\mathbf{B}$ by solving the overdetermined linear system defined in Eq (8).
Verify with principal component analysis that the homogeneous linear system has a solution, namely, that the basis points move rigidly.

This algorithm is completely linear. In order to compare its results to other algorithms which compute depth, we add the following step:

Depth (optional): Compute $\mathbf{Q}$, the root of $\mathbf{B}^{-1}$, and multiply the affine vector $\mathbf{b}$ at each point by $\mathbf{Q}$.

This step is non-linear and may not have a closed-form solution.

We now describe in detail the computation of the inverse Gramian $\mathbf{B}$, the second step of the algorithm:

The computation of B:

First, we rewrite Eq (8) as 2 linear equations in the elements of $\mathbf{B}$. $\mathbf{B}$ is a 3×3 symmetric matrix and therefore has 6 unknown elements. Since the equations in (8) are homogeneous, we can only compute $\mathbf{B}$ up to a scaling factor, and therefore we arbitrarily set $\mathbf{B}_{33} = 1$. We define a 5-dimensional vector $\mathbf{h} = (\,\mathbf{B}_{11} \quad \mathbf{B}_{12} \quad \mathbf{B}_{13} \quad \mathbf{B}_{22} \quad \mathbf{B}_{23}\,)^T$, where $\mathbf{h}$ includes the remaining unknown elements of matrix $\mathbf{B}$. We can now rewrite (8) as follows:

$$\mathbf{u}_1^l \mathbf{h} = v_1^l, \qquad \mathbf{u}_2^l \mathbf{h} = v_2^l \tag{12}$$

where

$$\mathbf{u}_1^l = \begin{pmatrix} x_i y_i \\ x_i y_j + x_j y_i \\ x_i y_k + x_k y_i \\ x_j y_j \\ x_j y_k + x_k y_j \end{pmatrix}^T, \qquad \mathbf{u}_2^l = \begin{pmatrix} x_i x_i - y_i y_i \\ 2(x_i x_j - y_i y_j) \\ 2(x_i x_k - y_i y_k) \\ x_j x_j - y_j y_j \\ 2(x_j x_k - y_j y_k) \end{pmatrix}^T$$

and

$$v_1^l = -(x_k y_k), \qquad v_2^l = -(x_k x_k - y_k y_k)$$

Each image l, $1 \leq l \leq m$, provides 2 linear constraints on $\mathbf{h}$. We define a constraint matrix $\mathbf{U}$ of dimensions $(2m) \times 5$, where rows $(2l-1)$ and $2l$ are $\mathbf{u}_1^l$ and $\mathbf{u}_2^l$ of Eq (12) respectively. We denote the solution of the linear system $\mathbf{v}$, a $2m$-dimensional vector whose elements $(2l-1)$ and $2l$ are v_1^l and v_2^l of Eq (12) respectively. Given $m \geq 3$ frames, $\mathbf{h}$ can be computed by solving the following overdetermined linear system of equations:

$$\mathbf{U}\mathbf{h} = \mathbf{v}$$

We obtain the minimal least square solution of this system by using the pseudo-inverse:

$$\mathbf{h} = (\mathbf{U}^T\mathbf{U})^{-1} \cdot \mathbf{U}^T\mathbf{v} \tag{13}$$

The solution of $\mathbf{h}$ above is described in terms of a 5×5 matrix multiplied by a 5-dimensional vector. It can be readily seen that both the inverse of the matrix, $\mathbf{U}^T\mathbf{U}$, and the vector, $\mathbf{U}^T\mathbf{v}$, are additive in the number of images (number of rows in $\mathbf{U}$ and $\mathbf{v}$), and therefore this algorithm can be implemented in an incremental way as follows:

- $\mathbf{U}^T\mathbf{U}$ and $\mathbf{U}^T\mathbf{v}$ are computed from three images or more and stored.

- $\mathbf{h}$ is computed from Eq (13).
- When additional data is obtained, $\mathbf{U}^T\mathbf{U}$ and $\mathbf{U}^T\mathbf{v}$ are computed from the new data, and their new values are added to their stored estimate.
- $\mathbf{h}$ is computed from Eq (13).

Note that if the basis points move rigidly, the five columns of matrix $\mathbf{U}$ and vector $\mathbf{v}$ are linearly dependent. An indication that the points do not move rigidly, or that the weak perspective approximation is not appropriate, can be obtained from an analysis showing that the vectors are independent.

Discussion

The invariant structure from motion algorithm described above belongs to a small group of recent algorithms which rely on many images to compute structure (e.g., [21]; see also [23] for an earlier work). This is an important feature, since motion disparities are typically noisy with low signal to noise ratios. Our algorithm is particularly simple, requiring only the closed-form solution of over-determined linear systems of equations.

Unlike most algorithms, our algorithm computes shape without computing explicit depth or the transformation between images. Like [23] 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 perspective distortions) approaches a Gaussian distribution with mean value 0.

4.2 Experiments with simulated and real data:

The discussion of model-based invariants and structure from motion above assumes weak perspective. The analysis and conclusions are therefore only approximately correct for real images, which are produced by perspective projection, and which are noisy. The simulations in Section 4.2.1 were designed to test the effects of perspective projection and noise on the SFM algorithm and the model-based invariant functions. In Section 4.2.2 we present results with a real matched sequence of images.

4.2.1 Simulated perspective projection and noise

In the following simulations we generated a test object of four points, for which we computed the invariant matrix $\mathbf{B}$. For the computation we used 20 simulated random views of the test object, with noise added to the image, and with either real or weak perspective projection. $\mathbf{B}$ was then used to define the model-based invariant function $f_{\mathbf{B}}$ (defined in Eq (9)) specific to the test object. We evaluated $f_{\mathbf{B}}$ on $10,000$ new random views of the test object. We also evaluated $f_{\mathbf{B}}$ on views of different objects.

We varied the distance from the camera to the object, which affected the relative size of the object in the image and the amount of perspective distortions. We present comparative plots of the value of $f_{\mathbf{B}}$ as a function of the varying distance between the camera and the object, using two different test objects:

- The first test object (graphs I in Fig 3) was composed of four points selected randomly in $[0,1]^3$. We compared it to two other types of objects: in type 1 the objects were composed of four points selected randomly in $[0,1]^3$; in type 2 the object was composed of four non-coplanar corners of the unity cube.

- The second test object (graphs II in Fig 3) was composed of four non-coplanar corners of the unity cube. We compared it to two slightly different types of objects: type 1 as above; in type 2 the object was composed of four non-coplanar corners of a box with edges of length 1,2,3.

We repeated the simulation 1000 times (each time generating a new random object), and for each set of objects we repeated the simulation 10 times (for 10 randomly generated views).

Each graph in Fig 3 shows three plots of the value of $f_{\mathbf{B}}$, where $\mathbf{B}$ represents one of the two test objects. The three plots were generated by applying $f_{\mathbf{B}}$ to the test object and to two other types of objects (as discussed above). For each test object, three conditions are shown: (a) weak perspective, (b) full perspective, (c) full perspective and Gaussian noise. In condition (c), we added 1% Gaussian noise to the data. The noise was added to the image $x-$ and $y-$ coordinates, and its mean value amounted to 1% of the actual radius of the $3D$ object. (Thus, in images obtained at 10 units away from the camera, this noise was equivalent to 10% Gaussian noise in the image plane.) The value of $f_{\mathbf{B}}$ was computed using simulated images at randomly selected viewpoints, and was averaged over 10,000 repetitions. The error bars at each data point indicate half the standard deviation of the distribution of the value of $f_{\mathbf{B}}$.

In all the graphs, for both test objects and under all conditions, the average value of the invariant operator, when applied to the test object, was always lower than its average value when applied to different objects of all the types tested. This was true even with high perspective distortions (when the distance to the camera was smaller than the size of the object), as can be seen in Fig 3(b). Thus the invariant operator can be used to distinguish the test object from most views of other objects at any distance, using a fixed threshold of 0.3. Moreover, a comparison of plots (a) and (b) shows that the dominant source of error was the variation in the appearance of objects, some of which appear similar to the test object at some viewpoints, rather than the distortion due to perspective projection.

4.2.2 Experiments with real data

We used a sequence of 226 images of a ping-pong ball with black marks on it, rotating 450 degrees in front of the camera. The sequence was provided by Carlo Tomasi, who also provided the coordinates of the tracked marks, which were obtained by his tracking algorithm described in [20]. One image of the sequence is shown in Fig 4. The object was relatively far from the camera, and therefore the weak perspective assumption is appropriate for this sequence.

Four points were selected to serve as basis points. The invariant matrix $\mathbf{B}$ was computed for the four points using half the frames in which all the basis points had appeared. The affine coordinates vectors $\mathbf{b}$, for each of the remaining points, were also computed using the same frames. $\mathbf{B}$ and $\mathbf{b}$ were used to compute the model-based invariant functions: $f_{\mathbf{B}}$ from Eq (9) and $f_{\mathbf{b}}$ from Eq (10). These functions were then evaluated on all the frames in which the relevant points appeared. As expected, they returned small values, distributed around 0 with a small variance. The average result and standard deviation were:

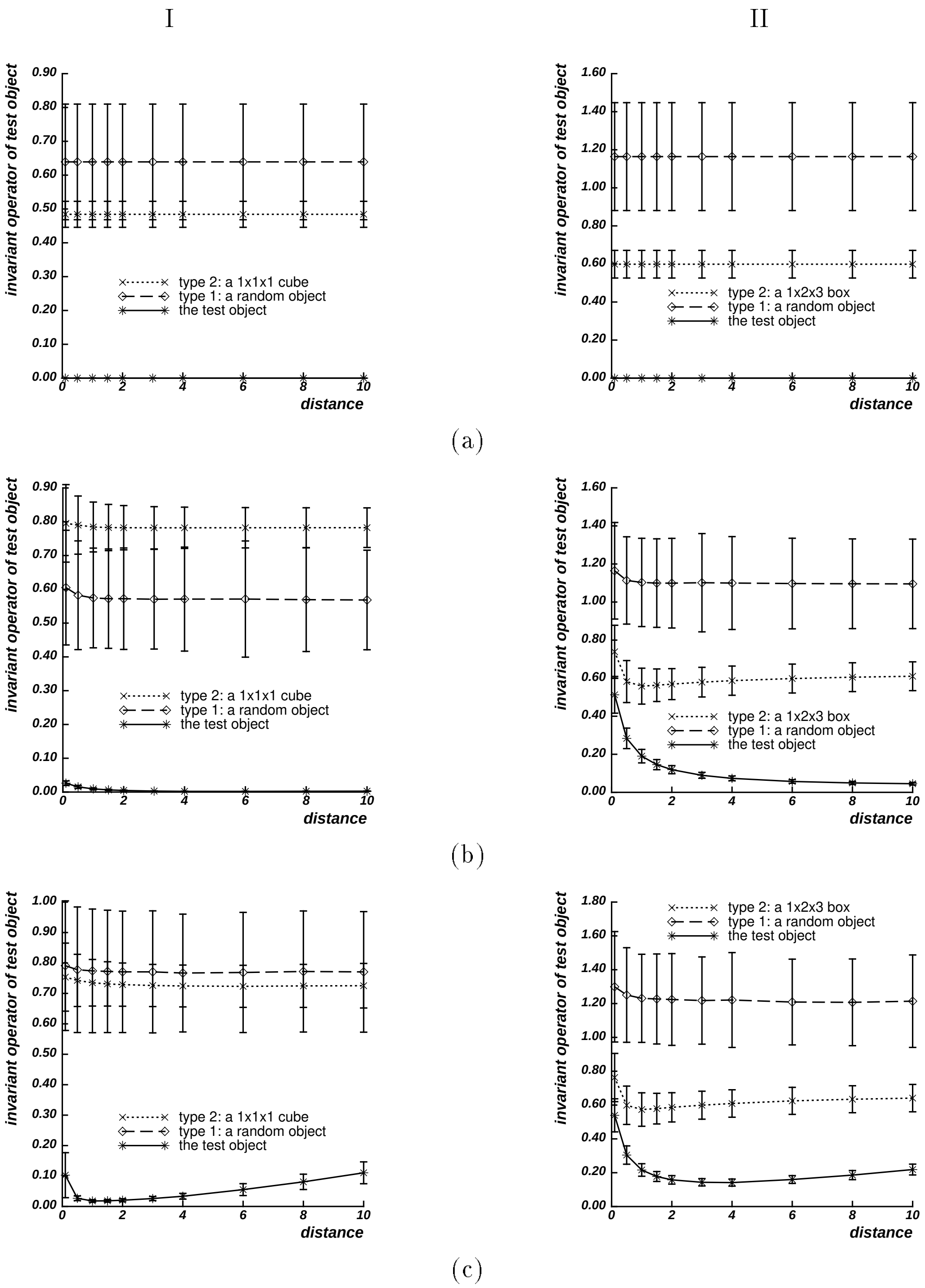

Figure 3: The invariant operator of a test object, applied to images of the same object and two other types of objects (see text): I- The test object was generated randomly, II- the test object was a cube. (a) The images were produced with weak perspective projection at different distances from the camera, (b) full perspective, (c) perspective projection and Gaussian noise.

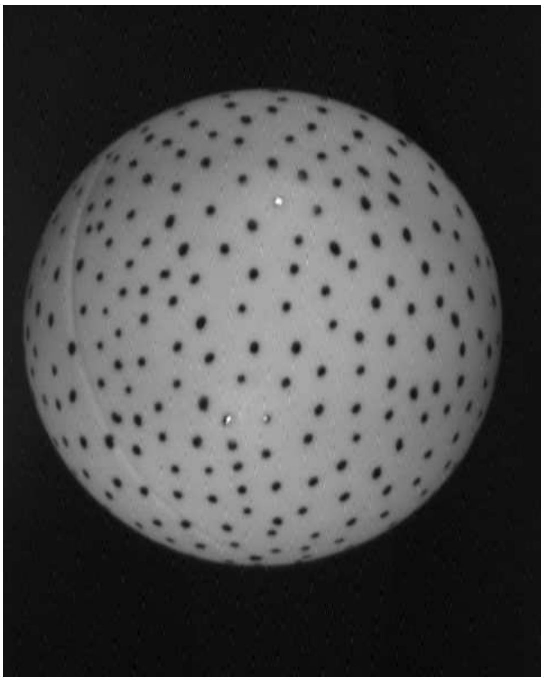

Figure 4: One frame from the ping-pong sequence.

$$
\begin{aligned}
f_{\mathbf{B}} &= 0.0264 \pm 0.0157 \\
f_{\mathbf{b}} &= 0.0194 \pm 0.0217
\end{aligned}
$$

Typically, the model-based invariant functions returned higher values when applied to the wrong points (higher by an order of magnitude or 2 orders of magnitude). As expected from the simulations above, the results varied around 1, with a large variance. One example of a wrong correspondence between the image and the model, where the model-based invariants strongly suggested rejection of the model, was the following:

$$
\begin{aligned}
f_{\mathbf{B}} &= 1.7073 \pm 0.1209 \\
f_{\mathbf{b}} &= 0.8034 \pm 0.2888
\end{aligned}
$$

Another example of a wrong correspondence, where the rejection was somewhat weaker, was the following:

$$
\begin{aligned}
f_{\mathbf{B}} &= 0.1708 \pm 0.0417 \\
f_{\mathbf{b}} &= 0.8015 \pm 0.2644
\end{aligned}
$$

4.2.3 Discussion

The results of the simulations, discussed in Section 4.2.1, show robustness to the effects of perspective distortions. Even when the depth variations between points were of the same order of magnitude as the depth of the points, the model-based invariant in Eq (9) could be used to successfully distinguish the "correct" object from most views of other objects with a fixed threshold.

When the scheme proposed in Section 3.2 is extended to full perspective projection, a non-linear algorithm is obtained. The extended algorithm depends on small variations in the data, variations which may not be reliably obtainable from a noisy image. Unless the perspective effects are sufficiently large, it may be better to use the robust linear scheme for weak perspective, rather than rely on small variations that may be unobservable in practice. To compensate, the weak perspective algorithm uses all the available images in order to minimize the errors due to noise and higher order effects of perspectivity. The weak perspective scheme requires at least three images. Algorithms that use full perspective (see [4, 15] for recent related work on structure without calibration) are useful under conditions typical to stereo viewing, namely, when there are only two images or when the differences between the images are large and observable.

The experiments with simulated and real data show that the model, matrix **B** for the four basis points and the affine coordinates for the remaining points, can be extracted reliably from a few images. Both in the simulated and real data, the rigid and affine model-based invariant functions effectively rejected most of the images of points whose $3D$ configuration was different from the configuration of the points in the learned model. The distribution of the values returned by the model-based invariant functions, when operating on images of the correct configuration, was reliably distinguishable from the distribution of values when operating on images of the wrong configuration.

When the correspondence between image and model is not known and the number of features in the image is large, the number of possible correspondences increases exponentially. If all correspondences are possible, the distributions of the invariant functions are not sufficiently separated to prevent false identifications (which happens when an image from the tail of the false distribution overlaps the tail of the correct distribution). Thus it is necessary to initially constrain the correspondence so that it can be reliably decided whether the model exists in the image or not. For example, it is possible to decrease the number of features by choosing more selective features or by using grouping.

5 Application to model-based recognition

A model-based invariant function can be viewed as a recognition operator that operates on a single image. This operator can distinguish images of a particular object from images of all other objects. We now discuss how the model-based invariant functions can be used with various existing recognition schemes. In Section 5.1 we discuss their use with alignment, a computationally-intense recognition strategy. In Section 5.2 we discuss their use with a memory-intense strategy, geometric hashing, introducing an indexing scheme into a database of $3D$ objects given a single frame ($2D$ image).

5.1 Alignment:

The following is based on the results discussed in Section 2. We showed that at recognition it is sufficient to verify the rigid structure of the four basis points with the quadratic invariant given in Eq (9), and the affine structure of the remaining points with the linear invariant given in Eq (10).

Alignment without transformation:

The alignment method for recognition [9] can be summarized as follows:

1. match three points in the image to the model;
2. compute the transformation between the model and the image;
3. verify recognition by predicting and matching additional model points using the computed transformation.

The model-based invariants can be used to implement alignment without computing the transformation, and with the same time complexity. The modified scheme can be described as follows (modified steps are marked with $*$):

1. match three points in the image to the model;
2. *compute from the appropriate $\mathbf{B}$ the image coordinates of the fourth basis point and verify its existence;
3. *verify recognition by predicting and matching additional model points using their affine coordinates.

This scheme involves one non-linear step, the prediction of the location of the fourth basis point from the locations of the matched three basis points. The worst-case time complexity of this algorithm is $O(Mm^3n^3)$, for M models, n image points and m model points.

The model-based invariants may be used more effectively for verification, rather than prediction. In this case, the recognition scheme does not involve any calculation other than the computation of the value of the quadratic and linear invariant functions, and comparing them to 0. The following linear alignment scheme has higher time complexity, but may be more robust and easier to implement:

1. match four points in the image to the model;
2. verify the match with the quadratic invariant $f_{\mathbf{B}}$;
3. verify recognition by predicting and matching additional model points using their affine coordinates.

The worst-case time complexity of this scheme is $O(Mm^4n^4)$ for M models, n image points and m model points.

5.2 How model based invariants can be used for indexing:

Result 1 can, in principle, be used as a recognition operator, which returns the serial number relative to the database of the object in the image. However, it requires $O(M)$ computations on a serial machine. We now describe a different use for recognition of the model-based invariant functions given in Eq (9) and Eq (10). The time complexity of the proposed computation does not depend on M. Another application for indexing is described in [14].

In this discussion we use a variation on the recognition scheme proposed by Lamdan & Wolfson in [12]. The idea there is to store as much information as possible on all the objects in the database, and all their possible appearances, in a lookup table. A set of features in the image provides an index into the table, pointing to a location where all the objects in the database, which could have possibly lead to the particular geometry of the observed set of features, are listed. The principle behind this approach is the conversion of on-line time-complexity at recognition time to space-complexity, at the cost of additional preprocessing computation time.

In the scheme proposed in [12], each ordered subset of the object features is represented by a single entry in the table. At recognition time, using $2D$ images of coplanar objects or using range data, the data provides a unique index into the table. However, when $2D$ images of general $3D$ objects are used, the index produced by the image is not unique. Therefore it is necessary to scan a line in the table, which increases the time complexity of recognition. Thus for each ordered set of image features, the complexity goes up from constant to $O(M)$, where M is the size of the table. This is unfortunate, and has lead to the limitation of most practical applications of this scheme to range data or images of coplanar objects.

To solve this problem, we construct the lookup table in a different way. Basically, we use different variables to fix the dimensions of the table, as will be defined shortly. These new dimensions bear less direct relationship to properties of the image features. The number of dimensions, denoted by l, depends on the particular invariant used to describe the geometry of sets of features.

In our scheme, a subset of ordered object features is represented by a $l-1$ dimensional hyperplane in the table, rather than by a single entry. Thus, for example, we will describe a 4-dimensional table where four ordered features of the object are represented by a 3-dimensional hyperplane in the table. A set of four ordered features in the image provides a *single* index at recognition time, so that the time complexity remains constant at the recognition of $2D$ views of general $3D$ objects.

The following discussion describes how the affine and rigid model-based invariant functions can be used to construct a table with the properties described above:

- Each of the model-based invariants requires a certain amount of features to operate on. For example, the quadratic invariant discussed in the previous section requires four features. We first find the necessary number of ordered features in the image.

- Each model-based invariant has a fixed number of unknown parameters, denoted $l+1$. For example, the quadratic invariant is determined by five parameters since $\mathbf{B}$ is a 3×3 scale-free symmetric matrix. Each image provides two constraints on the unknown parameters of the invariant, leaving us with two linear equations with l unknowns (one unknown is eliminated from each constraint). For example, l is 1 for a universal invariant (which does not exist), 2 for the affine invariant given in Eq (10), and 4 for the quadratic invariant given in Eq (9). We build two l-dimensional tables, one for each constraint. We use the coefficients of each linear constraint as an index into the table.

For anything but a universal invariant the image provides a vector index. Thus the price we pay for not being able to use a universal invariant is the size of the table, which is not one-dimensional anymore but multi-dimensional. We also pay in a higher frequency of inevitable false positive matches (or accidental matches).

As an example, we will derive the lookup table for the affine model-based invariant function (a similar scheme is described in [10]):

Let $\mathbf{b}$ denote the affine representation of five non-coplanar image points, as defined in Eq (3). There are three unknown parameters in this representation, b_1, b_2, b_3, while a single image gives only two constraints on these unknowns. We project the two constraints to a lower dimension by substituting one of the unknowns in each constraint. Thus, if we substitute b_1 for example, we are left with a linear constraint and two unknown parameters only:

$$b_2 + h_1(\mathbf{x}, \mathbf{y})b_3 = h_2(\mathbf{x}, \mathbf{y}) \tag{14}$$

We store a two-dimensional table to which the image provides a single index pair:
$[h_1(\mathbf{x}, \mathbf{y}), h_2(\mathbf{x}, \mathbf{y})]$. A second table is stored for the second linear constraint.

When the table is precomputed, an object is represented by a straight line. This line is determined uniquely by the object, since all possible image measurement pairs (h_1, h_2) must satisfy $b_2 + h_1 b_3 = h_2$. (Note that at the time the table is constructed, b_2, b_3 are known, and h_1, h_2 are not known.) More specifically, an object is represented by a line which identifies all the possible pairs (h_1, h_2) which can be computed from an image of the given object.

Given n image features and using the affine invariant which operates on five features, the corresponding lookup table can be used for recognition with complexity $O(n^5)$. This follows an assumption that all features may be similar in the worst case, and therefore all possible combinations of five image features should be examined. If the rigid invariant, which operates on four features only, is used, the indexing complexity can be reduced to $O(n^4)$. This should be compared with time complexity of $O(Mn^5)$, where M is the size of the table, when the original geometric hashing scheme is used. However, to avoid the need in storing large hyper-planes in four-dimensional tables, it is possible to use the two-dimensional tables of the affine invariant, and add the verification of the rigid invariant at each entry.

6 Summary

We proposed representations which are invariant to either rigid or linear $3D$ transformations, and we showed how they can be computed efficiently from a sequence of images with a linear algorithm. The algorithm assumed weak perspective, and it required at least four points and at least two frames. We derived from these invariant representations model-based projective invariant functions of general $3D$ objects. We showed how these functions can be used with novel and existing recognition strategies, in particular alignment and geometric hashing.

Appendix:

Under orthographic projection, or weak perspective without scale ($s = 1$), each image gives three constraints on the parameters of the symmetric matrix $\mathbf{B}$. More specifically, (8) becomes:

$$\mathbf{x}^T\mathbf{B}\mathbf{y} = 0, \quad \mathbf{x}^T\mathbf{B}\mathbf{x} = 1, \quad \text{and} \quad \mathbf{y}^T\mathbf{B}\mathbf{y} = 1. \tag{15}$$

Since the relations in (15) are not homogeneous, $\mathbf{B}$ should be fully determined including its scale. Thus there are 6 unknown elements in $\mathbf{B}$, for which two images give three constraints each. If the equations were independent, two images of four points should have been sufficient to compute depth. This contradicts previous results [1], and therefore it is not surprising that the 6 equations are linearly dependent, as the following discussion shows.

Given four image points (x_0, y_0), (x_1, y_1), (x_2, y_2), (x_3, y_3), assume without loss of generality that $x_0 = 0, y_0 = 0$. Let $\mathbf{x} = (x_1, x_2, x_3)$ and $\mathbf{y} = (y_1, y_2, y_3)$. Let $\mathbf{X}$ and $\mathbf{Y}$ denote the coordinates of the three points in a second image, and let $\mathbf{R} = \{r_{i,j}\}_{i,j=1}^3$ denote the $3D$ rotation matrix between the two images. In addition to the relations in (15), which hold for both images, we know from [8] that:

$$\mathbf{Y} = \frac{r_{32}}{r_{13}}\mathbf{x} - \frac{r_{31}}{r_{13}}\mathbf{y} + \frac{r_{23}}{r_{13}}\mathbf{X} = c_1\mathbf{x} + c_2\mathbf{y} + c_3\mathbf{X}, \tag{16}$$

where

$$r_{32}^2 + r_{31}^2 = r_{23}^2 + r_{13}^2 \quad \Rightarrow \quad c_1^2 + c_2^2 - c_3^2 = 1. \tag{17}$$

Substituting (16) in one of the corresponding relations given in (15) for the second image, and using $\mathbf{X}^T\mathbf{B}\mathbf{X} = 1$, we get:

$$\mathbf{Y}^T\mathbf{B}\mathbf{X} = 0 \quad \Rightarrow \quad c_3 + c_1\mathbf{x}^T\mathbf{B}\mathbf{X} + c_2\mathbf{y}^T\mathbf{B}\mathbf{X} = 0. \tag{18}$$

Using the equations in (15) for the first image, $\mathbf{X}^T\mathbf{B}\mathbf{X} = 1$, (17), (18), and the symmetry of $\mathbf{B}$, we get:

$$\mathbf{Y}^T\mathbf{B}\mathbf{Y} = c_1^2 + 2c_1c_3\mathbf{x}^T\mathbf{B}\mathbf{X} + c_2^2 + 2c_2c_3\mathbf{y}^T\mathbf{B}\mathbf{X} + c_3^2 = c_1^2 + c_2^2 - c_3^2 = 1.$$

Thus the last constraint, $\mathbf{Y}^T\mathbf{B}\mathbf{Y} = 1$, follows from the first five constraints, and the 6 equations produced by the two images are not linearly independent.

Acknowledgements: I thank Misha Pavel, Scott Kirkpatrick, and Ronen Basri for many helpful discussions and suggestions. I am also grateful to Carlo Tomasi, who provided the data for the experiments with real images, to Larry Maloney for technical help, and to Richard Weiss, Roger Mohr, Amnon Shashua, and the referees for many useful comments regarding the manuscript.

References

[1] J. Y. Aloimonos and C. M. Brown. Perception of structure from motion: I: optic flow vs. discrete displacements, II: lower bound results. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition*, pages 510–517, Miami Beach, FL, 1986.

[2] J.B. Burns, R. Weiss, and E. Riseman. View variation of point-set and line segment features. In *Proceedings Image Understanding Workshop*, pages 650–659, April 1990.

[3] D. T. Clemens and D. W. Jacobs. Space and time bounds on indexing 3-D models from 2-D images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(10):1007–1017, 1991.

[4] 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.

[5] D. Forsyth, J. L. Mundy, A. Zisserman, C. Coelho, A. Heller, and C. Rothwell. Invariant descriptors for 3-D object recognition and pose. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13:971–991, 1991.

[6] P. Gros and L. Quan. Invariant theory: a practical introduction. RT 69–IMAG–7 LIFIA, Institut IMAG, University of Grenoble, France, 1991.

[7] D. Heeger and A. Jepson. Simple method for computing 3D motion and depth. In *Proceedings of the 3rd International Conference on Computer Vision*, pages 96–100, Osaka, Japan, 1990. IEEE, Washington, DC.

[8] T. S. Huang and C. H. Lee. Motion and structure from orthographic projections. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(5):536–540, 1989.

[9] D. P. Huttenlocher and S. Ullman. Object recognition using alignment. In *Proceedings of the 1st International Conference on Computer Vision*, pages 102–111, London, England, June 1987. IEEE, Washington, DC.

[10] D. W. Jacobs. Space efficient 3D model indexing. In *Proceedings Image Understanding Workshop*, January 1992.

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

[12] Y. Lamdan and H. Wolfson. Geometric hashing: a general and efficient recognition scheme. In *Proceedings of the 2nd International Conference on Computer Vision*, pages 238–251, Tarpon Springs, FL, 1988. IEEE, Washington, DC.

[13] H. Mitsumoto, S. Tamura, K. Okazaki, N. Kajimi, and Y. Fukui. 3-D reconstruction using mirror images based on a plane symmetry recovering method. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(9):941–946, 1992.

[14] R. Mohan, D. Weinshall, and R. R. Sarukkai. 3D object recognition by indexing structural invariants from multiple views. In *Proceedings of the 4th International Conference on Computer Vision*, pages 264–268, Berlin, Germany, 1993. IEEE, Washington, DC.

[15] 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.

[16] Y. Moses and S. Ullman. Limitations of non model-based schemes. A.I. Memo No. 1301, Artificial Intelligence Laboratory, Massachusetts Institute of Technology, 1991.

[17] T. Poggio and T. Vetter. Recognition of structure from one 2D model view: observations of prototypes, object classes and symmetries. A.I. Memo No. 1347, Artificial Intelligence Laboratory, Massachusetts Institute of Technology, 1992.

[18] 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.

[19] A. Shashua. Correspondence and affine shape from two orthographic view: motion and recognition. A.I. Memo No. 1327, Artificial Intelligence Laboratory, Massachusetts Institute of Technology, December 1991.

[20] C. Tomasi and T. Kanade. Shape and motion from image streams: a factorization method - 3. detection and tracking of point features. CMU-CS-91-132, School of Computer Science, CMU, 1991.

[21] 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.

[22] S. Ullman. Computational studies in the interpretation of structure and motion: summary and extension. In J. Beck, B. Hope, and A. Rosenfeld, editors, *Human and Machine Vision*. Academic Press, New York, 1983.

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

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

[25] D. Weinshall and C. Tomasi. Linear and incremental acquisition of invariant shape models from image sequences. In *Proceedings of the 4th International Conference on Computer Vision*, pages 675–682, Berlin, Germany, 1993. IEEE, Washington, DC.

[26] I. Weiss. Projective invariants of shapes. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition*, pages 291–297, June 1988.