Duality of multi-point and multi-frame Geometry: Fundamental Shape Matrices and Tensors*

Daphna Weinshall[1], Michael Werman[1] and Amnon Shashua[2]

[1] Inst. of Computer Science, Hebrew University, Jerusalem 91904, Israel
{werman,daphna}@cs.huji.ac.il
[2] Dept. of Computer Science, Technion, 32000 Haifa, Israel

Abstract. We provide a complete analysis of the geometry of N points in 1 image, employing a formalism in which multi-frame and multi-point geometries appear in symmetry: points and projections are interchangeable. We derive bilinear equations for 6 points, trilinear equations for 7 points, and quadrilinear equations for 8 points. The new equations are used to design new algorithms for the reconstruction of projective shape from many frames. Shape is represented by shape descriptors, which are sufficient for object recognition, and for the simulation of new images of the object. We further propose a linear shape reconstruction scheme which uses all the available data - all points and all frames - simultaneously. Unlike previous approaches, the equations developed here lead to *direct* and *linear* computation of shape, without going through the cameras' geometry.

1 Introduction

The geometry of multiple primitives in multiple frames, where a $3D$ model consisting of many primitives is projected to a sequences of images via unknown cameras, has two inherent unknowns: the camera geometry, and the shape geometry. We do not know in advance the parameters of the projection from $3D$ to $2D$ in each image, and we do not know the parameters of the $3D$ shape (position) of each point in the model. Depending on the application at hand, we may want to compute shape, or camera geometry, or both. For example, tracking requires the knowledge of camera geometry, whereas object recognition requires the knowledge of shape.

Very often, the sequence of computations had been argued to be the following: First, compute the camera geometry (by camera geometry we refer here to both explicit representations, e.g., rotation and translation matrices, or implicit representations, e.g., the fundamental matrix). Second, compute the shape using the known projection geometry. This order of events makes particular sense

* This research was supported by the Israeli Ministry of Science under Grant 032.7568, and by the U.S. Office of Naval Research under Grant N00014-93-1-1202, R&T Project Code 4424341—01.

when the first task that one needs to solve is the tracking of features across frames.

But the reverse order of computation is also sensible: first compute shape, then compute camera geometry if necessary. This is the logical order when only shape is needed, for example, in a pure recognition task. If this is the case, it makes sense to compute the shape first, rather than rely in the shape computation on an otherwise unnecessary intermediate computation (the computation of camera geometry), which may not always be reliable.

This line of reasoning lead us to consider representations of shape which differ from the traditional Cartesian coordinates (whether in a Euclidean, affine, or projective basis). We seek shape representations that can be computed from images directly and robustly, and which include sufficient detail to unambiguously identify and simulate new images of the same shape. These are necessary requirements for a shape representation to be "good". Thus we propose below new shape descriptors, describing the shape of 6-8 points, which can be directly and linearly computed from at least 2-4 images. The computation of these shape descriptors does not require the computation of camera parameters (camera calibration, see also [13]).

Most of the literature on the subject followed the first path, namely, computing camera calibration first using techniques derived from multi-frame geometry (see, e.g., [6, 7, 10, 4]). Much less is known about multi-point geometry under perspective projection with uncalibrated cameras, and this gap is filled by our paper. We present here a complete analysis of the multi-point geometry, describing relations between the projections of many points in an image (each relation has a dual relation in multi-camera geometry). This analysis provides the foundation for a computation where shape is computed first, and camera geometry second. A similar analysis was presented by Carlsson in [1] (see also [15]).

More specifically, in Section 2 we show dual results to the ones obtained by computing camera geometry first. We first observe that a projection matrix from a $3D$ projective world to a $2D$ projective image is really a point in $\mathcal{P}^3$ (a geometrical interpretation of this point is given in [1]). We then observe that the relations between models, projections, and images can be written in a symmetrical form where models and projections are interchangeable. Using these observations, for every known relation between images and projection matrices we can derive a dual relation between images and models.

We use the dual results to develop new algorithms for the direct computation of shape, without first computing camera geometry. In particular, we describe a linear algorithm to compute the fundamental shape matrix of 6 points from at least 4 images, a linear algorithm to compute the fundamental shape tensor of 7 points from at least 3 images, and a linear algorithm to compute the fundamental shape tensor of 8 points from at least 2 images. Experiments with real images are described in Section 3. In Section 4 we show how to enhance the shape computation to include many points simultaneously. The computed shape descriptions are sufficient for the identification of novel images of the same object, and for the prediction of new images, as described in Section 5.

2 Algebraic and Geometrical derivation of Results

In this section we derive bilinear, trilinear, and quadrilinear relations using a formalism analogous (but dual) to [3]. We show duality between multi-frame and multi-point geometries: every relation between the vector coordinates of one point in many images has an almost identical equivalent here, a relation between the vector coordinates of many points in 1 image. This duality follows from the employment of a symmetrical formalism to describe multi-point and multi-camera geometry, where $3D$ coordinates of points and projection parameters are interchangeable (see Section 2.2).

2.1 Problem definition

Consider a model which includes more than 5 $3D$ points in $\mathcal{P}^3$ (all 5 point sets are projectively equivalent). W.l.o.g. (assuming uncalibrated cameras) we choose a coordinate system where the first 5 points are the standard projective basis, and $\mathbf{M}_i = [X_i, Y_i, Z_i, W_i]$ denotes the coordinates of the (i+5) point. Thus the shape matrix of the model $\mathtt{M}$, a $4 \times n$ matrix, is:

$$\mathtt{M} = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & X_1 & \ldots & X_n \\ 0 & 1 & 0 & 0 & 1 & Y_1 & \ldots & Y_n \\ 0 & 0 & 1 & 0 & 1 & Z_1 & \ldots & Z_n \\ 0 & 0 & 0 & 1 & 1 & W_1 & \ldots & W_n \end{pmatrix} \tag{1}$$

Similarly and again w.l.o.g, we choose the image coordinates of the first 4 points to be the standard projective basis in $\mathcal{P}^2$. Let $\mathbf{m}_i = [a_i, b_i, c_i] \in \mathcal{P}^2$ denote the vector of homogeneous coordinates of the $(i+5)$ point in the image. The projected shape matrix $\mathtt{m}$, a $3 \times n$ matrix, is:

$$\mathtt{m} = \begin{pmatrix} 1 & 0 & 0 & 1 & a_0 & \ldots & a_n \\ 0 & 1 & 0 & 1 & b_0 & \ldots & b_n \\ 0 & 0 & 1 & 1 & c_0 & \ldots & c_n \end{pmatrix}$$

Since the image shape matrix $\mathtt{m}$ is a projection of the shape matrix $\mathtt{M}$, there exists a 3×4 projection matrix $\mathtt{P}$ such that the following equality holds in $\mathcal{P}^2$:

$$\mathtt{P} \cdot \mathtt{M} = \mathtt{m} \tag{2}$$

Given our particular selection of projective bases, and using the fact that the first 4 points are transformed from a basis in $\mathcal{P}^3$ to a basis in $\mathcal{P}^2$, the projection matrix $\mathtt{P}$ is of the form [2]:

$$\mathtt{P} = \begin{bmatrix} \alpha \; 0 \; 0 \; \delta \\ 0 \; \beta \; 0 \; \delta \\ 0 \; 0 \; \gamma \; \delta \end{bmatrix}$$

We define a corresponding projection vector in $\mathcal{P}^3$ (a geometrical interpretation of this vector can be found in [1]):

$$\mathbf{p} = \begin{pmatrix} \alpha & \beta & \gamma & \delta \end{pmatrix}$$

2.2 Multi-frame and multi-point geometry

Using Eq. (2) with the $5+i$ model point $\mathbf{M}_i$ and the j-th frame obtained by the projection camera $\mathbf{p}_j$, producing the image point $(x_{ji}, y_{ji}, 1)$ gives:

$$x_{ji} \;=\; \frac{\alpha_j X_i + \delta_j W_i}{\gamma_j Z_i + \delta_j W_i} \;,\quad y_{ji} \;=\; \frac{\beta_j Y_i + \delta_j W_i}{\gamma_j Z_i + \delta_j W_i}$$

Clearly these equations are completely symmetrical with respect to $\mathbf{M}_i$ and $\mathbf{p}_j$: if we interchange the 2 vectors, we will get exactly the same image point. With k frames and $n+4$ points, we get the $2k \times n$ measurements matrix $\mathtt{W}$ whose ji element is x_{ji} for $j \leq k$, and $y_{(j-k)i}$ for $k < j \leq 2k$ (cf. [11, 14]). Now if we read $\mathtt{W}$ by columns, the i-th column gives us multi-camera geometry; if we read it by rows, the j-th and $(j+k)-th$ rows give us multi-point geometry in a single image.

We use this symmetry to obtain dual relations to those obtained for multi-camera geometry, by reading the data-matrix by rows instead of by columns; there are the following role changes:

- The solution vector in $\mathcal{P}^3$ is a projection operator $\mathbf{p}_j$ in multi-point geometry, and a $3D$ point $\mathbf{M}_i$ in multi-camera geometry.
- In multi-point geometry the data vectors are the 2 row-vectors $[x_{1j}, \ldots, x_{nj}]$ and $[y_{1j}, \ldots, y_{nj}]$ - the image coordinates in the j-th frame of all the points. In multi-frame geometry, the data vector is the column-vector $[x_{i1}, \ldots, x_{ik}, y_{i1}, \ldots, y_{ik}]$ - the trajectory of the $i+5$ point in k frames.

2.3 Multi-point geometry

From now on we fix the frame and ignore the subscript j. Every $3D$ point $\mathbf{M}_i = [X_i, Y_i, Z_i, W_i]$ from the 5th on ($i \geq 0$), which is projected to an image point $[a_i, b_i, c_i]$, defines 2 constraints on the projection matrix $\mathtt{P}$. We write these as linear homogeneous equations constraining the projection vector $\mathbf{p}$, $\forall\, i \geq 0$:

$$\begin{bmatrix} c_i\, X_i & 0 & -a_i\, Z_i & c_i\, W_i - a_i\, W_i \\ 0 & c_i\, Y_i & -b_i\, Z_i & c_i\, W_i - b_i\, W_i \end{bmatrix} \cdot \mathbf{p} = 0 \qquad (3)$$

(note that $X_0 = Y_0 = Z_0 = W_0 = 1$). Using these equations, we obtain relations between models and images.

Given $n+4$ points, including the 4 image basis points and n additional points, (3) expands to $2n$ linear equations for $\mathbf{p}$. The matrix representing this over-constrained linear system is $2n \times 4$. Since the linear system is homogenous and has a non-trivial solution $\mathbf{p}$, the rank of this matrix must be smaller than 4. Thus the determinant of every subset of 4 rows of this matrix must be 0. This gives us $\binom{2n}{4}$ constraints on $\mathbf{p}$.

Bilinear equations Given 6 points (or $n = 2$), the constraints matrix is 4×4 and of rank 3; thus we have 1 constraint – the determinant of the matrix must be 0. This gives us the following equation:

$$\begin{aligned}
&(-a_0\,b_1 + a_0\,c_1)\,(W_1\,X_1 - Y_1\,Z_1) \;+\; (a_1\,b_0 - b_0\,c_1)\,(W_1\,Y_1 - Y_1\,Z_1) \;+ \\
&(-a_1\,c_0 + b_1\,c_0)\,(W_1\,Z_1 - Y_1\,Z_1) \;+ \\
&(-a_0\,c_1 + b_0\,c_1)\,(X_1\,Y_1 - Y_1\,Z_1) \;+\; (a_0\,b_1 - b_1\,c_0)\,(X_1\,Z_1 - Y_1\,Z_1) \;=\; 0
\end{aligned} \tag{4}$$

We propose the following shape vector as a shape descriptor, "good" in the sense discussed in the introduction, and fully describing the projective shape of the 6 points. In other words, the projective coordinates of the 6th point can be computed from this vector (if the shape representation is needed for some purpose other then recognition or the generation of new images)[3]:

$$\mathbf{V6} \;=\; [W_1\,X_1,\ W_1\,Y_1,\ W_1\,Z_1,\ X_1\,Y_1,\ X_1\,Z_1] - Y_1\,Z_1\mathbf{1}_5 \tag{5}$$

This representation can be computed from at least 4 pictures (or directly from the $3D$ model).

We can rewrite (4) in a matrix form, in a dual way to the use of the fundamental matrix to describe the epipolar geometry:

$$\mathbf{m}_0^T G_{01}\mathbf{m}_1 \;=\; [a_0, b_0, c_0] \begin{bmatrix} 0 & X_1\,Z_1 - W_1\,X_1 & -X_1\,Y_1 + W_1\,X_1 \\ W_1\,Y_1 - Y_1\,Z_1 & 0 & X_1\,Y_1 - W_1\,Y_1 \\ Y_1\,Z_1 - W_1\,Z_1 & -X_1\,Z_1 + W_1\,Z_1 & 0 \end{bmatrix} \begin{bmatrix} a_1 \\ b_1 \\ c_1 \end{bmatrix} \;=\; 0 \tag{6}$$

Whereas the fundamental matrix depends on the camera calibration, the fundamental shape matrix G_{01} here depends on the $3D$ shape of the object. It has only 4 degrees of freedom in it (up to scale), since its elements sum to 0.

Trilinear equations Given 7 points (or $n = 3$), the constraints matrix is 6×4, giving us $\binom{6}{4}$= 15 constraints. Using algebraic tools, we show that there are only 7 independent equations: 3 bilinear equations involving subsets of 6 points, and 4 new trilinear equations. The 4 new constraints give us the following set of equations:

$$\begin{bmatrix}
0 & 0 & -a_0\,c_1\,c_2 + b_0\,c_1\,c_2 & 0 \\
-a_0\,a_2\,c_1 + a_2\,c_0\,c_1 & 0 & a_0\,b_2\,c_1 - b_2\,c_0\,c_1 & 0 \\
a_0\,a_2\,c_1 - a_0\,c_1\,c_2 & 0 & -a_0\,b_2\,c_1 + a_0\,c_1\,c_2 & 0 \\
0 & -a_0\,c_1\,c_2 + b_0\,c_1\,c_2 & 0 & 0 \\
0 & -a_2\,b_0\,c_1 + a_2\,c_0\,c_1 & 0 & -b_0\,b_2\,c_1 + b_2\,c_0\,c_1 \\
0 & a_2\,b_0\,c_1 - b_0\,c_1\,c_2 & 0 & b_0\,b_2\,c_1 - b_0\,c_1\,c_2 \\
a_0\,a_1\,c_2 - a_1\,c_0\,c_2 & a_0\,b_1\,c_2 - b_1\,c_0\,c_2 & 0 & 0 \\
0 & 0 & -a_1\,b_0\,c_2 + a_1\,c_0\,c_2 & b_0\,b_1\,c_2 - b_1\,c_0\,c_2 \\
-a_1\,a_2\,c_0 + a_1\,c_0\,c_2 & -a_2\,b_1\,c_0 + b_1\,c_0\,c_2 & a_1\,b_2\,c_0 - a_1\,c_0\,c_2 & -b_1\,b_2\,c_0 + b_1\,c_0\,c_2 \\
-a_0\,a_1\,c_2 + a_0\,c_1\,c_2 & -a_0\,b_1\,c_2 + a_0\,c_1\,c_2 & 0 & 0 \\
0 & 0 & a_1\,b_0\,c_2 - b_0\,c_1\,c_2 & -b_0\,b_1\,c_2 + b_0\,c_1\,c_2
\end{bmatrix}^T
\begin{pmatrix}
X_1\,Y_2 - W_1\,Z_2 \\
X_1\,Z_2 - W_1\,Z_2 \\
X_1\,W_2 - W_1\,Z_2 \\
Y_1\,X_2 - W_1\,Z_2 \\
Y_1\,Z_2 - W_1\,Z_2 \\
Y_1\,W_2 - W_1\,Z_2 \\
Z_1\,X_2 - W_1\,Z_2 \\
Z_1\,Y_2 - W_1\,Z_2 \\
Z_1\,W_2 - W_1\,Z_2 \\
W_1\,X_2 - W_1\,Z_2 \\
W_1\,Y_2 - W_1\,Z_2
\end{pmatrix} = 0$$

[3] $\mathbf{1}_n$ below deontes the vector of length n, whose elements are all 1.

The "good" shape descriptor of the 7 points is the following shape vector:

$$\begin{aligned}\mathbf{V7} \;=\; &[X_1\,Y_2,\; X_1\,Z_2,\; X_1\,W_2,\; Y_1\,X_2,\; Y_1\,Z_2,\; Y_1\,W_2,\; Z_1\,X_2,\; Z_1\,Y_2,\\ &\qquad Z_1\,W_2,\; W_1\,X_2,\; W_1\,Y_2] - W_1\,Z_2\,\mathbf{1}_{11}\end{aligned} \tag{7}$$

As in multi-camera geometry, we can write each new trilinear constraint in a tensor form, using a $3 \times 3 \times 3$ fundamental shape tensor T. More specifically, we have:

$$\sum_{i,j,k} \mathsf{T}_{ijk}(\mathbf{m}_0)_i(\mathbf{m}_1)_j(\mathbf{m}_2)_k \;=\; 0 \tag{8}$$

Each of the 4 constraints has a different shape tensor. For example, the tensor of the first constraint is:

$$T = \left[\begin{bmatrix} 0\;0 & Z_1\,X_2 - W_1\,X_2 \\ 0\;0 & 0 \\ 0\;0 & -X_1\,W_2 + W_1\,X_2 \end{bmatrix} \begin{bmatrix} 0\;0\;0 \\ 0\;0\;0 \\ 0\;0\;0 \end{bmatrix} \begin{bmatrix} 0\;0 & -Z_1\,X_2 + Z_1\,W_2 \\ 0\;0 & 0 \\ 0\;0 & 0 \end{bmatrix}\right]$$

However, there are only 11 unknowns in all 4 tensors, and writing the constraints as above allows us to compute the shape vectors from 3 images or more.

Quadrilinear equations With 8 points we have $\binom{8}{4} = 70$ constraints. Using algebraic tools, we show that there are only 15 independent equations: 3 bilinear equations involving subsets of 6 points, and 12 trilinear equations involving 7 points. However, there are new quadrilinear equations that define 22 independent constraints on a new 41-dimensional shape descriptor.

$$\begin{aligned}\mathbf{V8} \;=\; &[W_1\,W_2\,Z_3,\; W_1\,X_2\,Z_3,\; W_1\,X_2\,W_3,\; W_1\,Y_2\,X_3,\; Z_1\,W_2\,Y_3,\; Z_1\,W_2\,Z_3,\; Z_1\,W_2\,W_3,\; W_1\,X_2\,Y_3,\; Z_1\,X_2\,W_3,\\ &Z_1\,Y_2\,X_3,\; W_1\,Z_2\,W_3,\; W_1\,W_2\,X_3,\; W_1\,W_2\,Y_3,\; W_1\,Y_2\,Z_3,\; W_1\,Y_2\,W_3,\; W_1\,Z_2\,X_3,\; W_1\,Z_2\,Y_3,\; W_1\,Z_2\,Z_3,\\ &Y_1\,Z_2\,X_3,\; Y_1\,Z_2\,Z_3,\; Y_1\,Z_2\,W_3,\; Y_1\,W_2\,X_3,\; Y_1\,W_2\,Z_3,\; Y_1\,W_2\,W_3,\; Z_1\,X_2\,Y_3,\; Z_1\,X_2\,Z_3,\; Y_1\,X_2\,Z_3,\\ &Y_1\,X_2\,W_3,\; Z_1\,Z_2\,Y_3,\; Z_1\,Z_2\,W_3,\; Z_1\,W_2\,X_3,\; Z_1\,Y_2\,Z_3,\; Z_1\,Y_2\,W_3,\; Z_1\,Z_2\,X_3,\; X_1\,Y_2\,W_3,\; X_1\,Z_2\,Y_3,\\ &X_1\,Z_2\,Z_3,\; X_1\,Z_2\,W_3,\; X_1\,W_2\,Y_3,\; X_1\,W_2\,Z_3,\; X_1\,W_2\,W_3] \;-\; X_1\,Y_2\,Z_3\,\mathbf{1}_{41}\end{aligned} \tag{9}$$

The derivation of the equations constraining this shape descriptor are omitted. Using more than 8 points does not lead to any new equation.

3 Experiments

Using a sequence of real images, where features had been automatically detected and tracked, we compute the projective shape of the tracked points using the following procedure:

Initially, we choose an arbitrary basis of 5 points. For each additional point:

1. The corresponding shape vector **V6** is computed in 2 ways:

Linear computation: all the available frames are used to solve an overdetermined linear system of equations, where each frame provides the single constraint given in (4).

Non-linear computation: we use 3 frames only: the first, middle, and last, and the following non-linear constraint on the elements of $\mathbf{V6}$: $\mathbf{V6}_1\mathbf{V6}_2\mathbf{V6}_5 - \mathbf{V6}_1\mathbf{V6}_3\mathbf{V6}_4 + \mathbf{V6}_2\mathbf{V6}_3\mathbf{V6}_4 - \mathbf{V6}_2\mathbf{V6}_3\mathbf{V6}_5 - \mathbf{V6}_2\mathbf{V6}_4\mathbf{V6}_5 + \mathbf{V6}_3\mathbf{V6}_4\mathbf{V6}_5 = 0$
(where $\mathbf{V6}_i$ denotes the i-th component of the shape descriptor $\mathbf{V6}$.) A similar non-linear constraint was derived in [8]. Optimally, the non-linear computation should use the solution of the linear system defined by all the frames, and project the result onto the surface defined by the equation above. The non-linear computation normally gives 3 solutions.

2. From the shape vector $\mathbf{V6}$, the homogeneous coordinates of the point are computed as follows:

$$\frac{X_1}{W_1} = \frac{\mathbf{V6}_4 - \mathbf{V6}_5}{\mathbf{V6}_2 - \mathbf{V6}_3}\,, \quad \frac{Y_1}{W_1} = \frac{\mathbf{V6}_4}{\mathbf{V6}_1 - \mathbf{V6}_3}\,, \quad \frac{Z_1}{W_1} = \frac{\mathbf{V6}_5}{\mathbf{V6}_1 - \mathbf{V6}_2}$$

3. Finally, in order to compare the results with the real $3D$ shape of the points, the projective homogeneous coordinates are multiplied by the actual $3D$ coordinates of the projective basis points, to obtain the equivalent Euclidean representation.

We used a sequence obtained from the 1991 motion workshop. It includes 16 images of a robotic laboratory, obtained by rotating a robot arm 120^o (one frame is shown in Fig. 1a). 32 corner-like points were tracked. The depth values of the points in the first frame ranged from 13 to 33 feet; moreover, a wide-lens camera was used, causing distortions at the periphery which were not compensated for. (See a more detailed description in [9] Fig. 4, or [5] Fig. 3.)

We computed the shape of the 32 points as described above, using all the 16 frames in the linear computation, and using the non-linear computation. The real $3D$ coordinates of about half the points, the corresponding linearly reconstructed $3D$ coordinates, and the best reconstructed $3D$ coordinates (among the 4 solutions provided by the linear and non-linear computations), are shown below. We also give the median relative error (where the error at each point is divided by the distance of the point from the origin), computed over all points:

real shape:

$$\begin{bmatrix} -0.3 & -1.7 & -0.3 & 1.8 & 5.3 & 9.9 & 3.2 & -2.3 & 1.5 & -0.6 & 0.5 & 1.5 & -0.5 \\ -4 & -2.6 & 4.4 & 6.3 & 4.2 & -1.6 & -2.8 & -2 & 5 & 3 & 2 & 0.9 & 1 \\ 16.4 & 17.1 & 19.7 & 20 & 25.3 & 29.8 & 31.6 & 15.1 & 21.7 & 21.5 & 21.6 & 21 & 21.6 \end{bmatrix}$$

linear computation:

$$\begin{bmatrix} -0.3 & -1.8 & -0.6 & 0.9 & 3.8 & 8.9 & -0.8 & -2.4 & 0.7 & -0.6 & 0.3 & 1.3 & -0.4 \\ -3.6 & -1.3 & 5 & 6.2 & 4.4 & -1.5 & 0.7 & -1.3 & 4.8 & 2.6 & 1.8 & 0.4 & 0.7 \\ 15.8 & 14.7 & 21.5 & 24.9 & 27.5 & 30.1 & 9.7 & 13.4 & 23.5 & 20.2 & 21.1 & 21.1 & 19.2 \end{bmatrix}$$

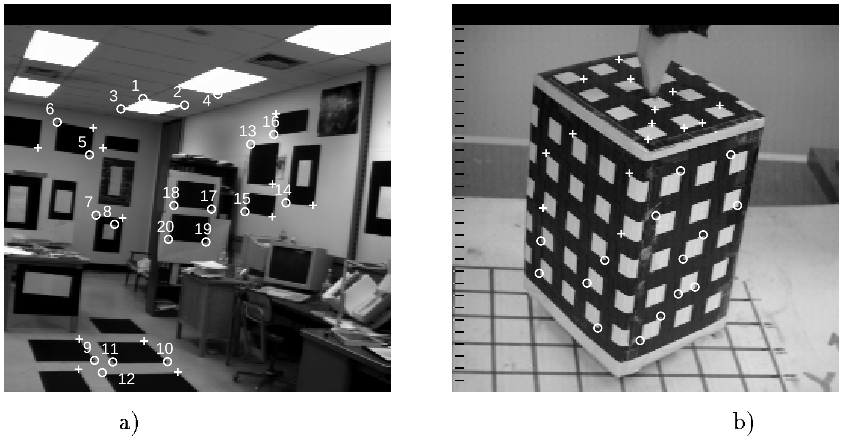

Fig. 1. a) One frame from the lab sequence. b) One frame from the box sequence.

median relative error: 12%

best of linear and non-linear computations:

$$\begin{bmatrix} -0.3 & -0.8 & -0.4 & 1.3 & 3.8 & 8.9 & 3.2 & -2.4 & 1.4 & -0.6 & 0.2 & 1.3 & -0.4 \\ -3.6 & -2.4 & 4.6 & 7.9 & 4.4 & -1.5 & -2.2 & -1.3 & 5.3 & 3.3 & 1.9 & 0.4 & 0.7 \\ 15.8 & 18.6 & 19.4 & 18 & 27.5 & 30.1 & 31.9 & 13.6 & 21.3 & 20.9 & 21.3 & 21.1 & 19.2 \end{bmatrix}$$

median relative error: 5.5%

4 Shape from many points and many frames

W.l.o.g., consider the trilinear shape descriptor $\mathbf{V7}$ defined in (7).

Lemma 1 rank 4. *Given $6+n$ points, the $11 \times n$ matrix whose i-th column is the shape vector $\mathbf{V7}$ of the points $< 1, 2, \ldots, 6, 6+i >$, $i = 1, \ldots, n$, is of rank 4.*

In other words, we first select 6 fixed points and recover the shape vectors of the sets $< 1, \ldots, 6, 7 >, < 1, \ldots, 6, 8 >, \ldots, < 1, \ldots, 6, 6+n >$. We then concatenate the shape vectors into a $11 \times n$ matrix denoted $\mathtt{W}$. Our claim is that the resulting matrix $\mathtt{W}$ is of rank 4 (instead of 11).

More specifically, let $\mathbf{V7}^i$ denote the shape vector of the set of points $< 1, \ldots, 6, 6+i >$. Let $\mathtt{W} = [\,\mathbf{V7}^1 \quad \ldots \quad \mathbf{V7}^n\,]$. It follows from (7) that:

$$\mathrm{W} = \begin{bmatrix} 0 & 0 & 0 & Y_1 & 0 & 0 & Z_1 & 0 & 0 & W_1 & 0 \\ X_1 & 0 & 0 & 0 & 0 & 0 & 0 & Z_1 & 0 & 0 & W_1 \\ -W_1 & X_1 - W_1 & -W_1 & -W_1 & Y_1 - W_1 & -W_1 & -W_1 & -W_1 & -W_1 & -W_1 & -W_1 \\ 0 & 0 & X_1 & 0 & 0 & Y_1 & 0 & 0 & Z_1 & 0 & 0 \end{bmatrix}^T \begin{bmatrix} X_2 & \ldots & X_n \\ Y_2 & \ldots & Y_n \\ Z_2 & \ldots & Z_n \\ W_2 & \ldots & W_n \end{bmatrix}$$

Thus $\mathtt{W}$ is of rank 4.

This result gives us the following algorithm for the reconstruction of shape using many views and many points, for an object with $n+6$ points:

1. Choose a subset of 6 "good" points (an algorithm on how to choose good basis points in described in [14]).
2. For every additional point M_i, $i \geq 7$:
   Using all available frames (but at least 3), compute the shape vector $\mathbf{V7}^i$ of the set of 7 points $< 1,2,3,4,5,6,6+i >$
3. Define the $11 \times n$ matrix $\mathtt{W}$, whose i-th column is the shape vector $\mathbf{V7}^i$. From Result 1, the rank of $\mathtt{W}$ is 4. Let $\tilde{\mathtt{W}}$ denote the matrix of rank 4 which is the closest (in least squares) to $\mathtt{W}$; $\tilde{\mathtt{W}}$ is computed using SVD factorization of $\mathtt{W}$ (see Section 4), with the decomposition: $\mathtt{W} \approx \tilde{\mathtt{W}} = \mathtt{U} \cdot \mathtt{V}$, where $\mathtt{U}$ is a 11×4 matrix, and $\mathtt{V}$ is a $4 \times n$ matrix.
4. Notice that $\tilde{\mathtt{W}} = \mathtt{UTT}^{-1}\mathtt{V}$ for every non-singular 4×4 matrix $\mathtt{T}$. Compute $\mathtt{T}$ such that

$$\mathrm{UT} = \begin{bmatrix} 0 & 0 & 0 & Y_1 & 0 & 0 & Z_1 & 0 & 0 & W_1 & 0 \\ X_1 & 0 & 0 & 0 & 0 & 0 & 0 & Z_1 & 0 & 0 & W_1 \\ -W_1 & X_1 - W_1 & -W_1 & -W_1 & Y_1 - W_1 & -W_1 & -W_1 & -W_1 & -W_1 & -W_1 & -W_1 \\ 0 & 0 & X_1 & 0 & 0 & Y_1 & 0 & 0 & Z_1 & 0 & 0 \end{bmatrix}^T \tag{10}$$

   for some constants $X_1,\ Y_1,\ Z_1,\ W_1$. This defines a homogeneous linear system of equations, which can be solved using, e.g., SVD decomposition. Note that this system can only be solved in a least-squares sense, as there are 40 equations with only 16 unknowns (the elements of $\mathtt{T}$).
5. (a) $\mathtt{T}^{-1}\mathtt{V}$ is the shape matrix of points $7, \ldots, 6+n$.
   (b) the coordinates of point 6 are obtained from $\mathtt{UT}$ and (10).

When using this algorithm with real images, our first results have been very sensitive to noise. Clearly the use of robust statistics (or outlier removal), in the solution of the linear system that defines the trilinear shape vector, is necessary.

5 Simulation of new images:

Rather than compute projective shape, the shape vectors described above can be used directly to simulate new (feasible) images of the object. It follows from Section 2.3 that 5 $3D$ points can be projected to any location in the image. Moreover, there is only 1 constraint on the location of the 6th point. Thus we start by choosing a random location for the first 5 points, and the first coordinate of the 6th point. The location of the remaining points can now be computed using the shape vectors:

The 6th point: we compute the shape vector $\mathbf{V6}$ of the first 6 points, from which we obtain the fundamental shape matrix G_{01}. We plug into (6) G_{01}, the coordinates of the 5th point in the new frame, and 1 constraint on the coordinates of the 6th point. This gives us a linear equation with 1 unknown, and we solve for the unknown coordinate of the 6th point.

The remaining points: for each point P_i, $i = 7..n$, we compute the shape vectors $\mathbf{V7}^l$, $l = 1..4$, which describe the shape of the first 6 points and P_i. From each shape vector we compute the trilinear shape tensor T_l. We plug each T_l, the coordinates of the 5th point in the new frame, and the coordinates of the 6th point computed in the previous stage, into (8). Thus the 4 trilinear constraints give us 4 homogeneous linear equations with 3 unknowns, the coordinates of the i-th point. For each point P_i we solve this system using SVD, thus obtaining the coordinates of all the points.

To test this algorithm we used a box sequence, which includes 8 images of a rectangular chequered box rotating around a fixed axis (one frame is shown in Fig. 1b). 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, and were given. (See a more detailed description of the sequence in [9] Fig. 5, or [5] Fig. 2.)

We rotated the box by up to $\pm 60^o$, translated it in $\mathcal{R}^3$ by up to ± 100 mms, and then projected it with uncalibrated perspective projection, to obtain new images of the box. The new images differed markedly from the original 8 images used for the computation of the shape vectors. We selected a "good" basis of 5 points, using the procedure described in [14]. In each image, we transformed 4 of the basis points to the non-standard basis of $\mathcal{P}^2$: $[1,0,1]$, $[0,1,1]$, $[0,0,1]$, $[1,1,1]$.

We used the image coordinates of the 5th point, and the x coordinate of the 6th point, to compute the shape of the remaining points as described above. In a typical image, in which the size of the projected box was 83×68 pixels, the median prediction error was 0.32 pixels; the mean prediction error was 0.46. The mean error could get larger in some simulated images, when large errors occurred in outlier points. The error at each point was computed in the image, by the Euclidean distance between the real point and its predicted location, using the original (metric) coordinate system of the image.

6 Discussion

When looking at data streams containing sequences of points, we wish to use all the available data: all frames, all points, or all frames and all points. Under weak perspective projection: [12] showed how to use all the points and 1.5 frames to linearly compute affine structure, [13] showed how to use all the frames of 4 points to linearly compute Euclidean structure, [11] used all the data to linearly compute affine shape and camera orientation, and [14] used all the data to linearly compute Euclidean shape.

The situation under perspective projection is more complex: [6, 7, 10, 4] (among others) showed how to linearly compute the camera calibration from 2-4 frames using all the points. Here (as well as in [1]) we showed how to linearly compute the projective shape of 6-8 points from all the frames. We also showed a 2-step algorithm to linearly compute the projective shape of all the points from all the frames. This does not yet accomplish the simplicity and robustness demonstrated by the algorithms which work under the weak perspective approximation.

References

1. S. Carlsson. Duality of reconstruction and positioning from projective views. In *IEEE Workshop on Representations of Visual Scenes*, Cambridge, Mass, 1995.
2. 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.
3. O. Faugeras and B. Mourrain. On the geometry and algebra of the point and line correspondences between N images. In *Proceeding of the Europe-China Workshop on geometrical modeling and invariant for computer vision*. Xidian University Press, 1995.
4. R. Hartley. Lines and points in three views - an integrated approach. In *Proceedings Image Understanding Workshop*, pages 1009–10016, San Mateo, CA, 1994. Morgan Kaufmann Publishers, Inc.
5. 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.
6. H. C. Longuet-Higgins. A computer algorithm for reconstructing a scene from two projections. *Nature*, 293:133–135, 1981.
7. Q.-T. Luong and O. Faugeras. The fundamental matrix: theory, algorithms, and stability analysis. *International Journal of Computer Vision*, 1995. in press.
8. L. Quan. Invariants of 6 points from 3 uncalibrated images. In *Proceedings of the 3rd European Conference on Computer Vision*, pages 459–470, Stockholdm, Sweden, 1994. Springer-Verlag.
9. 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.
10. A. Shashua and M. Werman. Trilinearity of three perspective views and its associated tensor. In *Proceedings of the 5th International Conference on Computer Vision*, Cambridge, MA, 1995. IEEE, Washington, DC.
11. 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.
12. S. Ullman and R. Basri. Recognition by linear combinations of models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(10):992–1006, 1991.
13. D. Weinshall. Model-based invariants for 3D vision. *International Journal of Computer Vision*, 10(1):27–42, 1993.
14. D. Weinshall and C. Tomasi. Linear and incremental acquisition of invariant shape models from image sequences. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(5):512–517, 1995.
15. D. Weinshall, M. Werman, and A. Shashua. Shape tensors for efficient and learnable indexing. In *Proceedings of the IEEE Workshop on Representations of Visual Scenes*, Cambridge, Mass, 1995.

This article was processed using the LaTeX macro package with ECCV'96 style