Image Description and 3D Interpretation From Image Trajectories Under Rotational Motion

Harpreet Sawhney
John Oliensis

COINS TR 89-90

September 1989

Image Description and 3D Interpretation from Image Trajectories under Rotational Motion

Harpreet S. Sawhney and John Oliensis

Computer and Information Science Department
University of Massachusetts
Amherst, MA 01003
Phone : (413)545-1519
NetAd : sawhney@umass

September 7, 1989

Abstract

We propose that constructing global spatial organizations from individual token trajectories is a powerful technique for motion and structure interpretation. Recovering 3D motion and structure from grouped trajectories has many analogies with 3D structure–from–contours in static imagery, which have not been made explicit in the 3D scene interpretation and motion vision literature. Our proposal is also similar to Stevens' [25] idea of strongly suggested geometric groupings in Glass patterns. Additionally, for our motion model, the grouped trajectories exploit the symmetry inherent in the motion. We demonstrate our approach through a two-stage process of image description of the motion of tracked points and their 3D interpretation from the resulting image trajectories. We solve for the scene motion and structure parameters for a set of 3D points rotating rigidly around an arbitrary axis. Instead of using individual point correspondences directly, we fit conic section curves to grouped sets of point correspondences and compute, in closed form, the 3D motion and structure parameters using the quadratic form matrix representing each conic. We also present an algorithm to obtain stable and robust conic fits to spatially proximal grouped point sets constrained locally by similarity of some parameters. In contrast, the descriptions from individual fits are highly unstable and practically useless for any 3D interpretation.

1 Introduction

Image motion in dynamic images is an important cue for 3D scene and motion understanding. However, temporally local descriptions of image motion in terms of point displacements have proven to be inadequate for the varied tasks of dynamic image understanding. We propose here instead that coherent motion and scene structure percepts can be derived from temporally–extended global geometric structures in the image motion. Our approach makes the image geometry of motion explicit in terms of token trajectories as a prelude to scene structure and motion interpretation. Specifically, we study reliable descriptions of extended-time image trajectories of points and recovery of the 3D motion and structure for a single rigid rotation using these image trajectories under *perspective projection.* Henceforth, we call the set of image correspondences over time for a single point its *point track* and a smooth curve fitted to this set its *point trajectory.* We will demonstrate that point tracks considered independently provide very weak constraints on the 2D trajectory and hence on the corresponding 3D trajectories. We therefore argue for the necessity of grouping individual image trajectories into constrained curve fits exploiting proximity and similarity in their 2D descriptions and the symmetry of the 3D motion. From these curves the 3D motion and structure can be accurately represented and recovered.

First, a method for reconstructing in closed form the 3D trajectory of a point rotating rigidly around a fixed axis given its image trajectory is described. For this method to succeed, a reliable estimate of a substantial portion of the full 360° image path is needed, since the accuracy of the trajectory fit degrades drastically for small segments. The 3D structure and motion of a rotating object can be completely recovered in this way, up to a single unavoidable scale ambiguity, if many points on the object are tracked. Note that the rotational motion studied here is not limited to one for which the axis of rotation has to pass through the origin of the camera-centered coordinate system. Hence, it is possible to recover the object structure when it is moving under off-centered rotation. In the terminology of two-frame motion descriptions, an off-centered rotation is equivalent to a rotation around an axis through the origin and a translation and thus contains the structure information.

We also present a grouping-based trajectory description algorithm which combines fits to multiple image point tracks based on locally common constraints and leads to good curve descriptions. Ample demonstrations of the contrast between poor trajectories for individual point tracks and good ones obtained from the combined fits are presented. Furthermore, we suggest, without a demonstration yet, that our constraint-based trajectory fitting along with clustering for the 3D parameters can lead to 3D interpretation of multiply moving objects. Using these combined fits and exploiting the symmetry of rotational motion, we can in turn make the approximate symmetries inherent in the scene explicit.

For small rotations, motion and structure recovery on the basis of short individual point tracks fails. We emphasize that this failure is intrinsic, not an artifact of a particular fitting routine, and that it provides a compelling argument for the necessity of grouping in motion interpretation. There is also an interesting analogy with the psychophysical literature on deriving global percepts from geometrical organizations. Our proposal is similar to Stevens' [25] idea of strongly suggested geometric groupings in Glass patterns. In the psychophysical motion literature, our work attempts to give a computational framework to Todd's [26] interesting study of perception of global regularities in rigid and non-rigid motion. Motion interpretation from point displacements grouped into motion contours also has obvious analogies with 3D structure–from–contours in static imagery – for instance computing the FOE is comparable to determining the vanishing point in static images.

Although it is particularly easy to envisage the appropriate global organizations for simple motions like pure translation and rotation, the power of the idea that coherent 3D percepts arise from global structure seems general and compelling.

2 Comparison To Related Work

The standard two frame approach to the computation of 3D motion and structure requires minimal assumptions, but does not permit natural descriptions of motions. For instance, the motion of a purely rotating soccer ball will typically be described in terms of a rotation around an axis passing through the camera center rather than through that of the ball. It can be argued that the second, natural description can be constructed from a series of two–frame rotation and translation computations, but the robustness of this is questionable when each computed pair is itself obtained through a non–linear constrained optimization. In our case, it can be argued that the particular natural description of 3D motion emerges out of the particular organization discovered in the image motion. If groups of image trajectories are coherently describable as conic curves, then this provides a strong non–accidental constraint that the underlying 3D motion is a rigid pure rotation and not a more complex composite motion. Hence the trajectory description process makes the type of the underlying 3D motion explicit in its more natural form vis-a-vis the two–frame motion interpretations. For a comprehensive survey of many of the two frame approaches see Barron [6].

Another problem with two frame approaches is that structure computation is based only on one displacement measurement for each feature, so that depth recovery may be erroneous even for correctly determined motion parameters. This problem is ameliorated for known motion, or motion with a stereo camera arrangement, since then the two frame depth computation can be incrementally refined [14].

One can also hope to overcome these problems by explicitly posing the 3D interpretation problem as a multi–frame problem, but this typically requires constraints on the motion, for instance that motion parameters remain constant over a period of time. Weng et al. [29], Chellapa et al. [32], Shariat [23] and Pavlin [18] use various forms of motion constancy over time and solve for the underlying 3D motion and/or structure parameters using a variety of non–linear parameter estimation techniques. Sethi et al. [22] find multiple globally smooth image trajectories but do not impose a model of motion and do not solve for the 3D parameters. Webb and Aggarwal [28] solve for the parameters of rotational motion using ellipses, but they are limited to *orthographic projection* only. Our solution is for perspective projection. Jaenicke [12] applies Webb and Aggarwal's method to radar doppler images. He does constrained fits on trajectories by averaging the common parameters over individual fits. But this method works only if the individual trajectories themselves are good which is the case in his examples. In contrast, our common fits do not rely on the goodness of individual fit parameters at all. Our approach is similar to those above but potentially applicable to less–constrained motions, since it makes use of spatial information to supplement temporal correspondences, and thus requires fewer frames and a shorter period of motion constancy. Moreover, this approach may be an important first step not only for 3D parameter estimation but also for detecting occlusions/deocclusions and object segmentation. A related approach to detecting occlusions is that of Baker et al. [5]. For a survey of various multi-frame approaches see Aggarwal et al. [2].

At the outset, we caution that many of the cited works treat more general models of motion than the one considered here. However, we hope to generalize it beyond the demonstration model studied in this paper. It has implications not only for a robust approach to 3D motion and structure computation, but also for a unified view of 2D grouping and 3D interpretation.

An innovation of our approach is that we demonstrate the necessity of grouping distinct point trajectories even for obtaining accurate individual descriptions when the latter alone are absolutely unreliable. Furthermore, the closed-form solution for perspective projection is new. We show how linking local image trajectories into global spatial organizations can lead to good qualitative and quantitative motion percepts. For instance, Fig. 1 depicts imaged points on two spheres rotating around different axes. The point tracks immediately suggest geometric organizations (elliptic arcs) similar to Glass patterns (see Stevens [25]). For rotations, each track of a point is a conic section curve. Our approach envisages grouping individual point tracks on the basis of smoothly varying relations among the 2D parameters describing these tracks, i.e. based purely on image plane characteristics. The result can be exploited to provide powerful constraints on a globally coherent motion and the underlying 3D structure, as demonstrated in this work. The process of combined trajectory description across point tracks makes the non-accidental relation of a coherent underlying 3D motion explicit amongst various trajectories. More complex motions and grouping relations are under study.

In the next three sections - 3,4 and 5 - we formulate and present a closed-form solution to the 3D estimation problem under rotational motion. Section 6 discusses multiple solutions and their disambiguation. Our grouping algorithm and its demonstrations on a real image sequence form the core of section 7. In section 8, we present the 3D estimation results. Section 9 wraps up this presentation with a summary and discussion of ongoing and future work.

3 The Problem: A Rotating Rigid Body in Space

We consider the case of a sequence of images of a rotating rigid body in space or the camera rotating around an arbitrary axis with respect to a fixed environment. Image point correspondences are assumed given over a period of time, and the camera parameters are known. The problem to be solved is that of determining the orientation and location of the rotation axis, and, for each point the location of its 3D trajectory. We assume perspective projection, and, for simplicity, a square image.

A set of parameters sufficient to define the problem geometry is depicted in Fig. 2 :

$\hat{b}$: *Unit vector in the direction of the rotation axis.*

$\vec{c}$: *Location vector of the rotation axis, given by the point where the axis intersects a plane normal to it that passes through the origin.*

$\vec{r}$: (x, y, z), *Location vector of a 3D point on the body.*

d : *Location of the center of the circular 3D trajectory of a body point, given by its signed distance from the point* $\vec{c}$ *and measured positive in the positive* $z-$*direction.*

k : *Radius of a circular 3D trajectory,* $k > 0$.

f : *Focal length of the camera.*

$\vec{R}$: (X, Y, f), *image vector in homogeneous pixel coordinates.*

This parameterization allows an easy separation of motion from structure parameters. We represent a vector as $\vec{v}$, a unit vector as $\hat{v}$ and $\boldsymbol{v}$ as the corresponding column vector. Quantities enclosed in square brackets , e.g. $[M]$, represent matrices.

The rotation axis can be specified using a minimum of four parameters [20]. Two additional parameters specify each circular trajectory of a body point relative to the rotation axis. These unknowns can be determined from the images only up to an arbitrary scale factor, the knowledge of which fixes the values of all parameters. Thus, for n 3D trajectories, there are $2n + 3$ determinable unknowns. Each image point in each frame gives one constraint equation assuming that the motion does correspond to the model under consideration, namely rigid rotational motion. Therefore one 3D point imaged in five frames, two 3D points imaged in four frames, or more than two 3D points all imaged in more than two frames, provide adequate information for a solution. In order to provide robustness in the face of noise in image measurements and small model deviations, however, more information (i.e. more frames) is necessary.

4 Formulation

An outline of our approach to the problem described above is as follows :

- An expression giving explicitly the perspective projection onto the image plane of a circular trajectory in space is derived. This projection is a general conic section.
- Conditions determining the type of the conic section of the projection are specified.
- Using a general conic fit algorithm, the sequence of image points identified with the motion of a single body point is fit to a conic section. This requires at least five

frames for each body point. The conic section is represented as a quadratic form in the homogeneous image coordinates.

- The 3D trajectory of the point is then solved for in closed form in terms of the conic section fit to the image data, up to an apparent eight-fold ambiguity.
- Six of the eight possible solutions can be rejected by invoking the scene-in-front-of-image criterion, or because they duplicate physically the other solutions.
- The remaining two-fold ambiguity can be resolved by requiring that different 3D trajectories share the same axis of rotation.
- Finally, we obtain a best-fit for the axis of rotation by combining information from all 3D points.

We first derive an expression for the image plane projection of a circular trajectory in space parameterized as depicted in Fig. 2. It is evident that our parameterization obeys the following conditions :

$$((\vec{r}_i - \vec{c}) - d_i\hat{b}) \cdot ((\vec{r}_i - \vec{c}) - d_i\hat{b}) = k_i^2 \tag{1}$$

$$\vec{r}_i \cdot \hat{b} = d_i \tag{2}$$

$$\vec{c}.\hat{b} = 0 \qquad\qquad \hat{b}.\hat{b} = 1 \tag{3}$$

The index i specifies the particular 3D point on the rigid body; we will drop it in the following treatment where there is no ambiguity in doing so.

The perspective equation in homogeneous coordinates is :

$$\vec{R} = f\frac{\vec{r}}{\vec{r}.\hat{z}} \tag{4}$$

From Equation (2) , using similar triangles, one obtains :

$$\vec{r} = \frac{d}{\vec{R} \cdot \hat{b}} \vec{R} \tag{5}$$

(The degenerate case where $d = \vec{R} \cdot \hat{b} = 0$ causes no difficulty and is discussed later.) Substituting Equation (5) into Equation (1) and rearranging terms, we get :

$$\frac{d^2}{(\vec{R} \cdot \hat{b})^2} \vec{R} \cdot \vec{R} - 2d \frac{\vec{R} \cdot \vec{c}}{\vec{R} \cdot \hat{b}} + \vec{c} \cdot \vec{c} - d^2 - k^2 = 0 \tag{6}$$

After multiplication through by $(\vec{R} \cdot \hat{b})^2$, this can be rewritten in terms of a symmetric quadratic form matrix. In boldface notation :

$$\boldsymbol{R}^T[\ d^2[I] - d[\ \boldsymbol{c}\boldsymbol{b}^T + \boldsymbol{b}\boldsymbol{c}^T\ ] + (\boldsymbol{c}^T\boldsymbol{c} - d^2 - k^2)[\boldsymbol{b}\boldsymbol{b}^T]\ ]\boldsymbol{R} = 0 \tag{7}$$

This equation represents a general conic in X and Y, the image plane coordinates. The linear and constant terms in the standard form of a conic [17] are included in this expression since $\vec{R}$ is in homogeneous form and includes a constant term. Thus, given a circle as the 3D trajectory of a point, the expected image projection is determined by the matrix :

$$[M_{exp}] = [\ d^2[I] - d[\boldsymbol{c}\boldsymbol{b}^T + \boldsymbol{b}\boldsymbol{c}^T] + (\boldsymbol{c}^T\boldsymbol{c} - d^2 - k^2)[\boldsymbol{b}\boldsymbol{b}^T]\ ] \tag{8}$$

The image will be an ellipse if the full circular 3D trajectory lies on the positive z side of the xy plane, i.e. if all points on this circle have positive z component. The image will be a hyperbolic arc when the 3D circle intersects the xy–plane in exactly two points. The four possible directions of approach to these intersections determine the four asymptotes in the image plane. Finally, when the 3D circle is tangent to the xy–plane, the imaged arc is a section of a parabola. Again, the two possible directions of approach towards the tangent point generate the two paths which become unbounded in the image. In the latter two cases, the image trace – i.e. the image projection of a 3D trajectory – is not closed. In the first case of an ellipse, the image trace may be either a closed curve – a complete ellipse – or an open partial ellipse.

Equation (7) specifies the imaged conic section corresponding to a 3D circle. Below, we show how starting from a conic section in the image plane, one can recover the 3D circle of which it is the projection, up to a scale ambiguity. Thus, given our model of rigid rotational motion, if it is possible to describe reliably a point's image track over time as a conic section trajectory, its 3D circular trajectory can also be determined. In a realistic scenario where only a small part of a single point's track may be available, the conic section fits tend to be locally accurate but globally erroneous. However, we will show that by doing spatio–temporal groupings across multiple point tracks, good globally correct image conics can be described and hence fairly correct 3D parameters extracted.

5 Solution

An arbitrary conic section in the image plane can be specified by a quadratic form as in Equation (7), with a symmetric 3–by–3 matrix $[M_{com}]$. The matrix $[M_{com}]$ is derived from the image data by a best fit. The corresponding 3D circle is determined by computing the values of the 3D parameters d, k, $\vec{c}$, $\hat{b}$ that yield a matrix $[M_{exp}]$ specifying the same conic section as $[M_{com}]$. Note that any scalar multiple of the matrix $[M_{exp}]$ specifies the same image curve, so that the 3D parameters can only be recovered up to a multiplicative ambiguity. This is the scale ambiguity mentioned earlier: only the ratios d_n, k_n, c_n and $\hat{b}$

are recoverable, where

$$d_n = \frac{d}{|\vec{c}\,|} \qquad k_n = \frac{k}{|\vec{c}\,|} \qquad \vec{c_n} = \frac{\vec{c}}{|\vec{c}\,|} \qquad c_n^T c_n = 1 \tag{9}$$

(We are assuming here that the rotation axis does not pass through the origin, i.e. that $\vec{c}$ is not the zero vector. This case is easily distinguished from the image data, and will be treated separately.)

Since the absolute magnitudes of the 3D parameters can not be determined, we rescale $[M_{exp}]$ by the magnitude squared of the location vector $\vec{c}$. The rescaled matrix can be written in terms of the recoverable ratios :

$$[M_{exp}] = [\ d_n^2[I] - d_n[c_n b^T + bc_n^T] + (1 - d_n^2 - k_n^2)[bb^T]\ ] \tag{10}$$

The ratios are computed from the image data by requiring that $[M_{exp}]$ is equal to $[M_{com}]$ up to a scale factor, which can also be found. We use our grouping algorithm to determine $[M_{com}]$. For each point, the resulting best fit conic section to the path of its image projections is specified and represented by a matrix $[M_{com}]$.

5.1 Axis Through Origin

We first consider the simpler case where the rotation axis passes through the origin. Then the expected conic form matrix is :

$$[M_{exp}] = [\ d^2[I] - (d^2 + k^2)[bb^T]\ ] \tag{11}$$

As above, we normalize this by d, assuming it is non-zero, and write the matrix in terms of the ratio k/d which is recoverable. (The solution for the special case when d is zero follows trivially from the non-zero case.) The normalized matrix is :

$$[M_{exp}] = \alpha[\ [I] - (1 + \frac{k^2}{d^2})[\boldsymbol{b}\boldsymbol{b}^T]\ ] \tag{12}$$

We have included a scale factor α to represent explicitly the multiplicative ambiguity discussed above.

The eigenvalue–eigenvector pairs for this matrix are :

$$\lambda_1 = -\alpha(\frac{k}{d})^2 \quad \lambda_2 = \alpha \quad \lambda_3 = \alpha \qquad \boldsymbol{n}_1 = \boldsymbol{b} \quad \boldsymbol{n}_2 = \boldsymbol{n}_2 \quad \boldsymbol{n}_3 = \boldsymbol{n}_3 \tag{13}$$

where $\boldsymbol{n}_2$ and $\boldsymbol{n}_3$ are any two independent vectors in a plane normal to $\boldsymbol{b}$. Note that two eigenvalues are identical and the third one is of a different sign and magnitude. In the more general case where the rotation axis does not intersect the origin this redundancy of the eigenvalues will not occur.

In order to recover the 3D parameters of the trajectory from the image data, we compute the eigenvalues of the matrix $[M_{com}]$ derived from the conic fit algorithm. If two of the eigenvalues are same and third one is different in magnitude and of opposite sign, this implies that the sequence derives from a point rotating around an axis passing through the origin. Let the eigenvalues of $[M_{com}]$ be λ_1, and $\lambda_2 = \lambda_3$. Since $[M_{com}] = [M_{exp}]$ (after adjustment of the scale factor α), these eigenvalues can be identified with the eigenvalues of $[M_{exp}]$ calculated above. Then,

$$\alpha = \lambda_2 \ (\ or\ \lambda_3) \qquad \frac{k}{d} = \pm\sqrt{-\frac{\lambda_1}{\alpha}} \tag{14}$$

and $\boldsymbol{b}$ is the unit eigenvector for λ_1. Without loss of generality, $\boldsymbol{b}$ can be forced to lie in the hemisphere of positive z directions. Then a unique sign for $\frac{k}{d}$ can be determined by invoking the fact that the imaged 3D trajectory must lie in front of the camera. For the more general case, the resolution of this type of sign ambiguity will be discussed at length.

Hence, for this case there is a unique, closed-form solution for the circular trajectory in space given $[M_{com}]$, the conic section fit to the image point sequence.

5.2 Generic Case: Rotation Axis Not Through Origin

As in the special case above, we determine the 3D trajectory of a body point by identifying the eigenvalues of $[M_{exp}]$ computed in terms of the 3D parameters with those of the image-derived matrix $[M_{com}]$. For convenience, we rewrite $[M_{exp}]$ with an explicit factor α as in the previous section and drop the subscript indicating ratios :

$$[M_{exp}] = \alpha[\ d^2[I] - d[\boldsymbol{c}\boldsymbol{b}^T + \boldsymbol{b}\boldsymbol{c}^T] + (1 - d^2 - k^2)[\boldsymbol{b}\boldsymbol{b}^T]\ ] \tag{15}$$

We first derive the eigenvalues of this matrix. As $\boldsymbol{b}$ and $\boldsymbol{c}$ are orthogonal, a rotated coordinate system can be chosen in which :

$$\boldsymbol{b} = [\,0\ 0\ 1\,] \qquad \boldsymbol{c} = [\,0\ 1\ 0\,] \tag{16}$$

It is a standard result of linear algebra that the eigenvalues of a matrix remain the same and can be computed in an arbitrary rotated coordinate system. After substituting Equation (16) into Equation (15), $[M_{exp}]$ has the simple form in this rotated system :

$$M'_{exp} = \alpha \begin{bmatrix} d^2 & 0 & 0 \\ 0 & d^2 & -d \\ 0 & -d & 1-k^2 \end{bmatrix}$$

The three eigenvalues of this matrix are :

$$\lambda_1 = \alpha\, d^2 \tag{17}$$

$$\lambda_2 \;=\; \frac{\alpha}{2}\,(\,(1+d^2-k^2)+\sqrt{(1+d^2-k^2)^2+4d^2k^2}\,) \tag{18}$$

$$\lambda_3 \;=\; \frac{\alpha}{2}\,(\,(1+d^2-k^2)-\sqrt{(1+d^2-k^2)^2+4d^2k^2}\,) \tag{19}$$

For α positive, λ_1 and λ_2 are positive and λ_3 is negative, except in the degenerate case when $d = 0$, corresponding to an image trajectory that is a degenerate conic, i.e. a straight line segment. We always choose to have α positive, and thus, for consistency, normalize the computed matrix $[M_{com}]$ similarly so that only one of its eigenvalues is negative and two are positive. Hence the negative eigenvalue of $[M_{com}]$ can be uniquely identified with λ_3. Also, the larger of the two positive eigenvalues of $[M_{com}]$ can be uniquely identified with λ_2, which one can show is always larger than λ_1 except possibly in the degenerate case.

The three eigenvalues of $[M_{com}]$ can therefore be assigned unambiguously to λ_1, λ_2 and λ_3, corresponding to Equations (17), (18) and (19), respectively, and the 3D parameters d, k and α can be solved for in terms of these eigenvalues. Let $\gamma_1 \equiv \lambda_2/\lambda_1$ and $\gamma_2 \equiv \lambda_3/\lambda_1$. Then :

$$d^2 = \frac{1}{\gamma_1+\gamma_2-\gamma_1\gamma_2-1} \qquad k^2 = -\gamma_1\gamma_2 d^2 \qquad \alpha = \frac{\lambda_1}{d^2} \tag{20}$$

Thus d and k are specified up to a sign ambiguity in d. We discuss the sign ambiguities of our solution in the next section.

$\boldsymbol{b}$ and $\boldsymbol{c}$ can also be obtained as follows. It is evident from Equation (15) that one of the eigenvectors of $[M_{exp}]$ is a vector $\boldsymbol{n}_1$ normal to the plane formed by $\boldsymbol{b}$ and $\boldsymbol{c}$. Since $\boldsymbol{n}_1$ satisfies :

$$[M_{exp}]\boldsymbol{n}_1 \;=\; \alpha\, d^2\boldsymbol{n}_1 \;=\; \lambda_1\boldsymbol{n}_1 \tag{21}$$

it is associated with the eigenvector λ_1 . The other two eigenvectors $\boldsymbol{n}_2$ and $\boldsymbol{n}_3$ with associated eigenvalues λ_2 and λ_3, respectively, must span the plane formed by $\boldsymbol{b}$ and $\mathbf{c}$, since all the eigenvectors are mutually orthogonal. Therefore :

$$\hat{c} = \cos\theta\,\hat{n}_2 + \sin\theta\,\hat{n}_3 \qquad \hat{b} = \sin\theta\,\hat{n}_2 - \cos\theta\,\hat{n}_3 \tag{22}$$

Further,

$$[M_{com}]\mathbf{c} = [M_{exp}]\mathbf{c} = \alpha\, d^2\mathbf{c} - \alpha\, d\, \boldsymbol{b} = \lambda_2 \cos\theta\, \boldsymbol{n}_2 + \lambda_3 \sin\theta\, \boldsymbol{n}_3 \tag{23}$$

$$[M_{com}]\boldsymbol{b} = [M_{exp}]\boldsymbol{b} = -\alpha\, d\, \mathbf{c} + \alpha(1-k^2)\, \boldsymbol{b} = \lambda_2 \sin\theta\, \boldsymbol{n}_2 - \lambda_3 \cos\theta\, \boldsymbol{n}_3 \tag{24}$$

Substituting Equation (22) into (23) and (24), one obtains :

$$\tan\theta = \frac{\alpha\, d^2 - \lambda_2}{\alpha\, d} = \frac{\alpha\, d}{\lambda_3 - \alpha\, d^2} = \frac{\alpha\, d}{\alpha(1-k^2) - \lambda_2} = \frac{\lambda_3 - \alpha(1-k^2)}{\alpha\, d} \tag{25}$$

Thus, $\tan\theta$ can be computed in closed form in terms of the image parameters up to the sign ambiguity in d. It follows that $\boldsymbol{b}$ and $\boldsymbol{c}$ can also be obtained in closed form up to sign ambiguities, by solving for the eigenvectors $\boldsymbol{n}_2$ and $\boldsymbol{n}_3$ of the image conic form matrix, which are identified unambiguously by their respective eigenvalues.

Hence, apart from the sign ambiguities, all the 3D parameters of the trajectory can be uniquely computed in terms of the eigenvalues and eigenvectors of $[M_{com}]$, the computed image conic form matrix.

6 Multiple Solutions

There are two solutions for d in Equation (20). For each of these, there are four solutions for $\boldsymbol{b}$ and $\boldsymbol{c}$ from the four sets of signed values of $\boldsymbol{n}_2$ and $\boldsymbol{n}_3$ in Equation (22). These *eight* solutions can be written as two sets of four, each corresponding to the same k :

$$S_1 = \{\ \{\boldsymbol{b}_1, \boldsymbol{c}_1, d_1\},\ \ \{-\boldsymbol{b}_1, -\boldsymbol{c}_1, d_1\},\ \ \{\boldsymbol{b}_1, -\boldsymbol{c}_1, -d_1\},\ \ \{-\boldsymbol{b}_1, \boldsymbol{c}_1, -d_1\}\ \}$$

$$S_2 = \{\ \{\boldsymbol{b}_2, \boldsymbol{c}_2, d_1\},\ \ \{-\boldsymbol{b}_2, -\boldsymbol{c}_2, d_1\},\ \ \{\boldsymbol{b}_2, -\boldsymbol{c}_2, -d_1\},\ \ \{-\boldsymbol{b}_2, \boldsymbol{c}_2, -d_1\}\ \}$$

The ambiguity within each set of four is evident, since the different signed values for the parameters all lead to the same computed conic form matrix in Equation (15). To see the relation between the two sets of solutions S_1 and S_2, refer to Fig. 3. Reflecting each of $\{\boldsymbol{b}_i, \boldsymbol{c}_i\}$ in S_1 across any one of the eigenvectors, $\boldsymbol{n}_2$ or $\boldsymbol{n}_3$, leads to a corresponding solution in the other set. Fig. 3 depicts one of these reflections across $\boldsymbol{n}_3$. The result of this reflection again yields a conic form matrix with the same sets of eigenvalues and eigenvectors. Hence, given the measured image conic form matrix, these eight indeed are solutions. Within the constraints of Equation (3), these are the only possible solutions because they exhaust all options in the representation of the given conic form matrix in terms of its eigencomponents, which in turn is a complete representation.

The apparent eight–fold ambiguity found above is not real. Since $\{\boldsymbol{b}, \boldsymbol{c}, \boldsymbol{d}\}$ and $\{-\boldsymbol{b}, \boldsymbol{c}, -\boldsymbol{d}\}$ represent the same point along the rotation axis, four of the above solutions simply duplicate physically the other four. This ambiguity can be eliminated by our convention that the vector $\boldsymbol{b}$ always lies in the positive z–component hemisphere. This leaves two remaining solutions in each set. We next impose the constraint that a 3D point must lie in front of the image plane in order to be imaged. [11]. From Equation 5, this implies that $\beta = \frac{\boldsymbol{d}}{\bar{\boldsymbol{R}}.\bar{\boldsymbol{b}}}$ must be positive. If the parameter pair $\{\boldsymbol{b}, \boldsymbol{d}\}$ satisfies this constraint, then the alternate set $\{-\boldsymbol{b}, \boldsymbol{d}\}$ cannot. Thus, two more solutions are eliminated, one from each set. The remaining two solutions, one from each set, cannot be disambiguated from the image trace of one point alone. This is similar to the well known ambiguity of two–frame motion computation for a planar set of points (see [16]).

However, multiple image trajectories of 3D points rotating rigidly around a common axis can be used to resolve this ambiguity. The true solution for the axis will be common to *all* the points, while the second, incorrect solutions will be unique for each 3D point. The true solution is therefore easily picked out. The reason for the mismatch among the incorrect solutions is that, as illustrated in Fig. 4, the eigenvectors $\boldsymbol{n}_2$ and $\boldsymbol{n}_3$ of the matrix $[M_{com}]$ are different for each 3D point except when the axis $\boldsymbol{b}$ passes through the origin, which anyway can be handled differently as shown earlier. These vectors span the plane formed by the true $\boldsymbol{b}$ and $\boldsymbol{c}$. Hence, the $\boldsymbol{b}$ and $\boldsymbol{c}$ solutions common to all 3D points make different angles with their respective spanning eigenvectors. As the second incorrect solutions are obtained by reflecting the true $\boldsymbol{b}$ and $\boldsymbol{c}$ across an eigenvector, clearly these solutions will be different for each 3D point. This is shown for two points in Fig. 4. Hence, the correct solution can be identified by combining information obtained from several image trajectories assuming they have already been segmented as a single rigid motion. In the presence of noise, the algorithm's ability to discriminate the true solution among the solutions for multiple trajectories will be limited by the similarity in their structure parameters.

7 Grouping Algorithm and Experiments

The goal of our grouping algorithm is twofold – to obtain reliable conic fits to individual image point tracks and further to make the similarities or dissimilarities across point trajectories explicit. Robust fits to point tracks leads to good estimation of 3D parameters. Explicit description of similarities across point trajectories gives us the potential to group various trajectories into a single object motion, to detect outliers and possibly detect multiple object motions.

Our algorithm is based on doing constrained conic fits to a set of point tracks instead of individual fits. We observe that locally in space the projection can be well approximated by a weak perspective. We also assume that point tracks proximal in the image plane are proximal in the scene too. Any errors due to this assumption can be weeded out by the combined fit algorithm. Under these general assumptions, conic fits to proximal image trajectories are constrained to have three of their five parameters the same – they should possess the same orientation and eccentricity and their centers should be collinear. Thus for a given set of proximal point tracks, we solve for the three common parameters across the set and two individual parameters for each trajectory.

There are two steps to the grouping. The first step, the linking stage, chooses a set of point tracks which are locally proximal and similar. Similarity is defined locally by a common direction of motion and approximately the same curvatures. For the demonstrations in this paper, we do this step manually. We are experimenting with automating the linking stage currently.

The second step is the incremental combined fit step. The linking step simply limits the problem size presented to the second step and hence reduces the time complexity of the fitting step. The goal of the constrained combined step is to find the maximal set within the input set of point tracks which can be described by good conic fits. A new point track is added each time and a combined fit tried. We also require that the combined fit not result in an unacceptably high fit error to any single trajectory. The error variance obtained through fitting conic sections to individual point tracks is used as a datum to compare the fit errors resulting from combined fit for each trajectory. The rationale behind this is that although the parameters of individual fits are bad, the variance of fit error indicates how well the underlying data can be described by the chosen model of motion. The instability in the fit parameters is due to extrapolating a point track beyond its tracked length. However, the fit errors resulting out of the independent fit algorithm are small. So the fit error for each point track obtained from the combined fit parameters should be within a small factor (we choose 3 for the experiments here) of this error. We accept a combined fit only if *all* the participating trajectories satisfy this criterion. This process of assimilating new point tracks is repeated until no more can be added. Note that at every new fit step, all the previously accepted point tracks participate afresh with the new one. Linking thus becomes important in reducing the problem set to be tried in the fitting stage.

For the purposes of combined fit, we parameterize conic sections to explicitly represent the common and different parameters. The collinearity of centers and common orientation is represented by a line parameterized by ρ and θ. Eccentricity is the third common parameter. The two distinct parameters are l_i, the location of the conic center along the common line and c_i, the location of a focus from the center along a perpendicular direction. Fig. 5 depicts this parameterization. The error measure we choose to minimize based on this parameterization uses the constraint that a conic section is the locus of points the ratio of whose distances from a fixed point, the focus, and a line, the directrix, is a constant equal to its eccentricity.

$$f_i = PF_i^2 - e^2\ PD_i^2 = 0 \tag{26}$$

where PF_i is the distance of a point from a focus and PD_i its distance from the corresponding directrix. In terms of our parameterization,

$$PF_i^2 = (x_i - c_i \cos\theta - \rho\cos\theta + l_i \sin\theta)^2 + (y_i - c_i \sin\theta - \rho\sin\theta - l_i\cos\theta)^2 \quad (27)$$

$$PD_i^2 = ((\rho + c_i/e^2) - (x_i\cos\theta + y_i\sin\theta))^2 \quad (28)$$

where (x_i, y_i) are the image point coordinates. We minimize,

$$\sum_i f_i^2(\rho, \theta, e^2, l_i, c_i) \quad (29)$$

over all the point tracks in the current set under consideration. We use an implementation of the BFGS–DFP quasi–Newton [30] optimization method for this fit problem. The initial guess for the very first combined fit comes from the individual fits and once a combined fit is found to the first subset, its parameters serve as initial guess.

An outline of our algorithm follows. Given a linked set of point tracks,

Step 1: Sort in decreasing order the arc lengths obtained by polygonal approximations of the point tracks. (Longer arcs constrain the fits better).

Step 2: Fit conics to each point track independently and record the error variance. (Done in two stages – Closed form fit first using Bookstein's algorithm [8] and then a fit minimizing square of a first order distance approximation using the first as initial guess.) [See Appendix].

Step 3: Set NTRY = 4 and CYCLE = 1 for the first NTRY point sets. (Try first four initially).

Step 4: Try a combined fit on the first NTRY point sets.

Step 5: If the fit method did not converge OR the fit errors for each point track are above the allowed range compared with the individually recorded errors, GO TO Step 7.

Step 6: If fit successful,

Step 6a: If no more points left, EXIT.

Step 6b: Else, CYCLE++, NTRY++. Set CYCLE flag for NTRY points. GO TO Step 4.

Step 7: If this was the last point, EXIT with maximal set one less than the full one.

Step 8: Get the next untried point track which should have CYCLE flag less than the current one. Swap it with the last point and GO TO Step 4.

Step 9: If all points have been tried for this CYCLE. EXIT with NTRY-1 points combined.

We first describe our experiments with grouping. Fig. 6 and 7 show 256–by–256 image frames 1 and 10, respectively, of a chequered box which was rotated around an arbitrarily chosen fixed body axis using a cartesian robot arm. Rotation between each frame was approximately 4°. The object was imaged, using a 16mm. focal length lens on a CCD camera with FOV approximately 24°, at a distance of approximately 60cm. Corner points defined as intersections of pairs of lines were tracked over the sequence using the robust line–tracking system of Williams and Hanson [31]. This system incorporates the straight–line detection algorithm of Boldt and Weiss [7] and dense displacement fields computed between two successive frames using an algorithm by Anandan [4]. Detected straight lines are projected into successive frames using the displacement field, and a directed–acyclic–graph (DAG) of line correspondences over time is constructed. Line intersections are then tracked through this DAG representation. Figs. 8 and 9 show the sampled displacement field between frame 1 and 2 and a sample of tracked line segments, respectively.

In Fig. 10 are shown resulting sample sets of tracked points. This figure motivates our idea of Glass–pattern like organizations present in the geometric structure of image motion. The displacement fields between any two frames fail to make the perception of rotational motion clear but the extended-time point trajectories and similarities across them makes this percept compelling. Fig. 11 shows the result of describing a few individual tracks through fitted conics. Although the tracks represent nearly 80 degrees (20 frames) of the 3D circular arc, still the image conics fail to make explicit the common motion geometry across different point sets. The individual fits are very good though. However, the next four Figures 12–15 show the results of our constrained fit algorithm on six groups of point sets. In each group, the number of participating point sets varies from seven to eleven. The combined fits show a dramatic improvement in the individual fits and more importantly, these make explicit the global percept of a common structure among these traces. All the point sets are shown for each group but only a few fitted curves are shown to avoid clutter. In all the groups, only one or two point sets got left out in the combined fit process. The motion and structure of these different points can now be quantitatively estimated with good accuracy. (The results are detailed in the next section.)

We emphasize that the poor results obtained by fitting to individual tracks are not a failure of our fitting routine but inherent in the problem. Fitting a conic to a partial, slightly noisy arc is extremely sensitive to the noise. To human view, widely different ellipses are indistinguishably good fits along the arc itself. Moreover, the difficulties of individual fitting worsen as the viewing time is reduced and the tracked arcs become smaller. Our grouping algorithm exploits the locally common constraint across distributed trajectories which constrains the combined fit well even when the participating trajectories poorly constrain individually.

An effective motion understanding system should successfully interpret even a rotation of only a few degrees of arc. In the extreme case, as Todd [26] suggests, human vision systems may be capable of this interpretation, based on the capture of compelling grouping relations, even from a two-frame span of data. Clearly, for such short viewing spans, it is difficult to interpret the motion from individual point traces. Even for the simplest case of pure rotation studied in this paper, it appears that spatial grouping is essential for making the percept of motion explicit even without 3D reconstruction. Additionally, it does lead to reliable estimation of 3D motion and structure too.

8 3D Estimation Results

Results on both simulated and the box image data are reported here. In our simulation, we generated image point data for four 3D points rotating rigidly around an axis with direction (b) $[1,1,1]$ and location (c) $[7,\quad 10,45]$. We imaged the four points for 50 frames with a 4 degree rotation between successive frames. A focal length of 160 pixels and image size of 256–by–256 were chosen for the camera model. We obtained four conic sections, one for each of the points, by fitting conic arcs to their discrete correspondences. Each of these was represented as a 3–by–3 conic form matrix M_{com}. In one experiment, no noise was added to the image point coordinates while in the second, we added $(-1,1)$ pixel uniform noise before fitting the conics. We report the true and computed 3D parameters in Table 1 for both. In this table and the next, the true axis direction and location are given at the top. For each point, the two computed solutions for the rotation axis are shown in the body of the table at left, the correct one first. The correct solutions stand out as predicted since they are common to all points, in both the noisy and noise–free cases. At right are shown the true and computed normalized structure parameters. All computed parameters agree well with the actual ones.

For the box image, we show the results of estimating the axis location and orientation, and the structure parameters for eight points. In Table 2, detailed results for a sample set of eight out of the forty trajectories obtained are shown. For each point trajectory, the angular error between the **b** and **c** vectors and the percentage errors in the d_i and k_i parameters are shown. We use ellipses generated by our grouping algorithm for each of these sample points within their respective groups. The eight traces chosen are marked in Figs. 12–15. Table 2 shows the results for these eight traces. The ground truth for this image was obtained using a pose estimation algorithm developed by Kumar [13]. These estimated parameters are labelled as *true* in the table. Our results match well with the estimated ground truth. Here again, the wrong solution for each point is evident except for points 1, 4, 5 and 7 which are coplanar. Hence, for these the second solution too is the same.

To get a final estimate of the 3D motion parameters, we do clustering on the unit sphere of all the pairs of **b** and **c** vector estimates obtained from each point set using an algorithm by Collins et al. [9]. Altogether this amounts to 40 pairs of vectors each for **b** and **c**. The clustering gets rid of the spurious solutions across all points. The final estimate is obtained by a least–squares fit to the clustered estimates using the largest eigenvalue–eigenvector pair of the sum matrix of the dyadic product matrices of each of the vectors. These estimates along with the angular errors are also shown below Table 2.

9 Comparisons with Two–frame Algorithms

For the sake of comparison, we ran the two–frame motion algorithms of Horn [11] and Adiv [1] on pairs of frames from the box sequence. Both these algorithms have been shown to work fairly well in a variety of motion situations within the limits of the assumptions of each. Horn's algorithm employs a widely used error measure [10], [27] for the problem within an elegant optimization framework. He uses the constraint that the view–ray for a point in frame one, the corresponding ray in frame 2 and the translation vector should be coplanar as shown in Fig. 16. Formally,

$$f_i = \mathbf{b} \bullet (\mathbf{r}'_{i_1} \times \mathbf{r}_{i_2}) = 0 \tag{30}$$

where,

b : *The translation unit vector in frame 2 coordinates.*

r'_{i_1} : *Frame 1 ray for point i rotated into the frame 2 coordinates.*

r_{i_2} : *Frame 2 ray for point i.*

(31)

Horn minimizes $\sum_i f_i^2$ over all the available point correspondences in two frames with the constraint that b is a unit vector. This is performed iteratively starting with an initial guess for the rotation and translation. At every step of the minimization, each f_i is linearized around the current estimate using Rodriguez' formula for rotations. This is similar to finding the Jacobian of the system [23] relating each f_i with incremental steps in the unknown parameters, three each for rotation and translation. Step changes in these are found by inverting these system of equations along with the constraint equation on b

which itself is linearized. This is carried on till convergence. Horn's experiments and our own, with simulated data have shown that this method works reliably under most motion and imaging situations.

However, as noted by Horn himself and in Spetsakis et al. [24], the above measure can give biased solutions for translation when the field-of-view (FOV) is small. Specifically, for small FOV, because all the viewed rays lie within a small bundle around the optical axis, the z-axis, a translation vector with a high z-component nearly always minimizes the error irrespective of the actual translation. We observe this behavior in pairs of frames for our box sequence as reported later here. Our experiments show that Horn's algorithm finds the biased solution as the global minimum and the correct one as a local minimum, especially when the image displacements are small. When the displacements become relatively large, the error measure for the correct solution becomes lower than that for the biased one. However, this difference still remains small enough to be a problem for any automated algorithm to pick out the correct solution.

Adiv [1] in his algorithm assumes that rotations are small to the extent that the rotation matrix can be approximated by its first order expansion. That is,

$$R = \begin{bmatrix} 1 & -\omega_x & \omega_y \\ \omega_z & 1 & -\omega_x \\ -\omega_y & \omega_x & 1 \end{bmatrix}$$ is the approximate rotation matrix. Further,

$$X' - X = \alpha = -\omega_x \tfrac{XY}{f} + \omega_y (f + \tfrac{X^2}{f}) - \omega_z Y + \tfrac{T_x f - T_z X}{z} \tag{32}$$

$$Y' - Y = \beta = -\omega_x (f + \tfrac{Y^2}{f}) + \omega_y \tfrac{XY}{f} + \omega_z X + \tfrac{T_y f - T_z Y}{z} \tag{33}$$

where,

(X,Y), (X',Y') : *Point coordinates in frames* 1 *and* 2, *respectively.*

α, β : *Displacements in X and Y over two frames.*

(T_x, T_y, T_z) : *The translation vector.*

f : *Focal length.* (34)

Adiv minimizes

$$\sum_{i=1}^{N} W_i * [(\alpha_{i_{meas}} - \alpha_i)^2 + (\beta_{i_{meas}} - \beta_i)^2] \tag{35}$$

where $\alpha_{i_{meas}}$ and $\beta_{i_{meas}}$ are the measured image displacements. He first eliminates the depth, z, out of this error. Then using a sampling on the unit sphere for translations, he computes the rotation vector for each of the sampled translation vectors. The pair of rotation and translation vectors leading to a globally minimum error is found as the solution.

Two observations can be made for Adiv's formulation. Firstly, it cannot handle large rotations. Secondly, his search method is sensitive to the quantization of the unit sphere of translations. In order to get a good performance, we ran his algorithm as three passes of a coarse–to–fine search.

We report results on the box sequence for Horn's, Adiv's and our own algorithms. Results shown are for frames 1 to 7 for Horn and Adiv and for the whole sequence for our own algorithm. Note that in our algorithm, the actual number of frames used for each point varies from 9 to 20. We show comparisons of computations of motion and depth for Horn and Adiv separately on a two–frame basis and between all three on a multi–frame basis. Note again that motion and depth parameters computed from Kumar's pose are used as ground truth for these comparisons too. First we discuss the two–frame computations by Horn and Adiv's algorithms. In Table 3 are shown the 3D motion parameter results for six pairs of frames. The pairs always involve the first frame. This ensures that over time, the length of displacement vectors in the image increases which makes the depth computations less sensitive to noise. As is evident from the comparisons, the errors in computation of rotation axis and translation vector for Adiv increase as the rotation angle increases. However, the rotation angle does not seem to be as sensitive to this. Tables 4, 5 and 6 show a comparison of two-frame depth computations for a selected set of 13 points between Horn and Adiv's algorithms. One observes that for frames 1–2, 1–3 and 1–4, when the image displacements are small, the errors in depth are high – 5–10%. But for frames 1–5, 1–6 and 1–7 these errors remain below 5%. Surprisingly this is true even for Adiv's algorithm which gave increasingly incorrect motion parameters as the rotation angles were increased. We do not have a concrete explanation for this yet. However, we conjecture that a phenomena similar to the one observed by Dutta et al. [19] for the special case of frontal plane might be responsible for this more general case too.

We wish to emphasize here that for frames 1–2 and 1–3, for Horn's algorithm, we chose the solution closer to the ground truth for our comparisons although this was a local minimum for the underlying error measure as shown in Table 9. The solutions returned as the global minimum are totally wrong with respect to the ground truth as is expected. The wrong solutions do show the bias for an alignment with the z–axis as predicted even when the motion in our case had a very small z–component [See table 3 also]. Secondly, we ran the two–frame algorithms between consecutive frames instead of using the first frame always as the datum. But the results, especially the depth results were consistently very erroneous.

In Tables 7 and 8 we show comparisons with our multi–frame algorithm. To be fair to the two-frame algorithms, we found the least-squares fits to their motion and depth computations over the 6 pairs used. We also converted the rotation and translations returned by these to the axis location and axis direction parameterization used by us. For depth computations, this simply implies that we averaged the depths returned in frame 1 for each point over the 6 pairs. Table 7 depicts the comparison of motion parameters. Horn's and our own algorithms have similar higher errors for the axis direction than Adiv's, whereas the latter has a higher error for the axis location compared to the much smaller errors for the other two. We show the comparisons of depth computations in Table 8. The averaged depths from Adiv and Horn and the trajectory based depths from our own algorithm are all within 0–5% errors. Horn's algorithm performs somewhat better than the other two for depth results.

In summary, we wish to state that for the restricted model of motion considered in this work, under imaging with a small FOV camera, our algorithm returns 3D parameters which are atleast as good as two proven two-frame methods, the latter being applied with some favorable conditions. However, our approach stands out in one distinct way in that it directly describes the underlying 3D motion in its natural form, that is a pure rotation. In the two-frame approaches, this description has to be derived by fitting a model to the rotations and translations computed for each interval. This process can be unstable especially when the underlying two-frame computations themselves are unreliable due to some of the inherent ambiguities or biases associated with the algorithms. Furthermore, we conjecture that the insensitivity to small FOV in our case is the result of computations based on an extended-time image measurements and the smoothing resulting from imposing a smooth model of motion within and across image trajectories. Whether this can be said of more general motions and other imaging situations has to wait for more extensive analysis and experimentation. We wish to emphasize that the results discussed above are only for one image sequence. A lot more experimentation should be carried out to evaluate the strengths and weaknesses inherent in various algorithms.

10 Conclusions

We have presented and demonstrated a method for recovering in closed form the 3D structure and motion of a rigid body, based on fits to the image trajectories of selected body points assuming a particular model of motion, namely pure rotational motion. The paths traced by the 3D point projections in the image are first fit to conic sections, which are represented by quadratic form matrices. The eigenvalues and eigenvectors of these matrices are related to the parameters of the 3D trajectories, this relationship is inverted and the 3D trajectories solved for. Hence we obtain a complete recovery of the body structure and motion up to a single scale ambiguity. We gave a discussion and complete analysis of multiple solutions and their disambiguation, discussed the effects of noise, and described the results of experiments with real and synthetic images.

We also demonstrated the importance of global spatial groupings of individual token displacements in motion interpretation. For pure rotation, since full conic section curves are significantly underdetermined by any small sub–arcs, we argued that organizing trajectories into global curves can give improved results over simply combining the very uncertain motion interpretations based on short, individual point displacement trajectories. We presented an algorithm for grouping locally constrained trajectories into combined fits thus making their similarities explicit while obtaining reliable curve descriptions. We also noted the relationship of this idea to psychophysical results on global organization, and proposed grouping as a general and powerful approach to motion interpretation.

We are developing techniques to automate the process of linking described above. We also want to experiment further with our grouping algorithm on multiple moving objects. In particular, the output parameters of combined fits could be used to do clustering in the image plane to delineate multiple motions. We wish to extend this notion of generic model-based extended-time motion geometries to more general models of motion.

Acknowledgments

We have derived constant intellectual and emotional support from Rakesh Kumar, Lance Williams, R. Manmatha, Brian Burns and P.Anandan throughout this work. We are grateful for Prof. Ed Riseman's and Prof. Al Hanson's consistent support and encouragement. Financial support for this research came from NSF/CER grant DCR8500332 and DARPA contracts F30602-87-C-0140.

References

[1] Gilad Adiv. *Interpreting Optical Flow.* PhD thesis, University of Massachusetts at Amherst, 1985. COINS TR 85-35.

[2] J. K. Aggarwal and N. Nandhakumar. On the computation of motion from sequences of images - a review. Technical Report TR-88-2-47, University of Texas at Austin, 1988.

[3] G. J. Agin. Fitting ellipses and general second-order curves. Technical report, The Robotics Institute, Carnegie-Mellon University, 1981.

[4] P. Anandan. *Measuring Visual Motion from Image Sequences.* PhD thesis, University of Massachusetts at Amherst, 1987. COINS TR 87-21.

[5] H. H. Baker and R. C. Bolles. Generalizing epipolar-plane image analysis on the spatiotemporal surface. *Proceedings DARPA Image Understanding Workshop*, pages 1022–1030, 1988.

[6] J. Barron. A survey of approaches for determining optic flow, environmental layout and egomotion. Technical Report RBCV-TR-84-5, University of Toronto, 1984.

[7] M. Boldt and R.Weiss. Token-based extraction of straight lines. Technical Report COINS TR 88-, University of Massachusetts at Amherst, 1988.

[8] F. L. Bookstein. Fitting conic sections to scattered data. *Computer Graphics and Image Processing*, 9:56–71, 1979.

[9] R. T. Collins and R. Weiss. An efficient and accurate method for computing vanishing points. *Topical Meeting of the Optical Society of America on Image Understanding*, 1989.

[10] O. D. Faugeras, F. Lustman, and G. Toscani. Motion and structure from motion from point and line matches. *IEEE First International Conference on Computer Vision*, pages 25–34, 1987.

[11] B. K. P. Horn. Relative orientation. *Proceedings DARPA Image Understanding Workshop*, pages 826–837, 1988.

[12] R. A. Jaenicke. Structure from limited motion of complex objects. *Proceedings IEEE Wkshp. on Visual Motion*, pages 256–263, 1989.

[13] R. Kumar. Determination of camera location and orientation. *Proceedings DARPA Image Understanding Workshop*, 1989.

[14] L. Matthies and T. Kanade. The cycle of uncertainty and constraint in robot perception. *The Fourth International Symposium on Robotics Research*, pages 327–335, 1988.

[15] V. S. Nalwa and E. Pauchon. Edgel–aggregation and edge–description. *The Eighth International Conference on Pattern Recognition*, pages 604–609, 1986.

[16] S. Negahdaripour and B. K. P. Horn. Direct passive navigation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(1):168–176, 1987.

[17] T. Pavlidis. Curve fitting with conic splines. *ACM Transactions on Graphics*, 2(1):1–31, 1983.

[18] I. Pavlin. Motion from a sequence of images. *Proceedings DARPA Image Understanding Workshop*, 1988. Cambridge, Massachusetts.

[19] E. M. Riseman R. Dutta, R. Manmatha and M. A. Snyder. Issues in extracting motion parameters and depth from approximate translational motion. *Proceedings IEEE Wkshp. on Visual Motion*, 1989.

[20] K. S. Roberts. A new representation for a line. *Proceedings Computer Vision and Pattern Recognition*, pages 635–640, 1988.

[21] P. D. Sampson. Fitting conic sections to very scattered data. *Computer Graphics and Image Processing*, pages 97–108, 1982.

[22] S. K. Sethi and R. Jain. Finding trajectories of feature points in a monocular image sequence. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(1), 1987.

[23] H. Shariat. *The Motion Problem: How to Use More than Two Frames.* PhD thesis, University of Southern California, Los Angeles, 1986.

[24] M. E. Spetsakis and J. Aloimonos. Optimal computing of structure from motion using point correspondences in two frames. *Proc. 2nd Intl. Conf. on Computer Vision*, pages 449–453, 1988.

[25] K. A. Stevens. Computation of locally parallel structure. *Biological Cybernetics*, 29:19–28, 1978.

[26] J. Todd. Visual information about rigid and non-rigid motion : A geometric analysis. *Journal of Experimental Psychology : Human Perception and Performance*, 8(2):238–252, 1982.

[27] R. Y. Tsai and T. S. Huang. Uniqueness and estimation of 3-d motion parameters and surface structures of rigid objects. *Image Understanding 1984*, pages 135–171, 1984.

[28] J. A. Webb and J. K. Aggarwal. Structure from motion of rigid and jointed objects. *Artificial Intelligence*, 19:107–130, 1982.

[29] J. Weng, T. S. Huang, and N. Ahuja. 3-d motion estimation, understanding and prediction from noisy image sequences. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(3), 1987.

[30] S.A.Teukolsky W.H.Press, B.P.Flannery and W.T.Vetterling. *Numerical Recipes in C.* Cambridge University Press, 1986.

[31] L. R. Williams and A. R. Hanson. Translating optical flow into token matches. *Proceedings DARPA Image Understanding Workshop*, pages 970–980, April 1988.

[32] G. Young and R. Chellappa. 3-d estimation using a sequence of noisy stereo images. *Proceedings Computer Vision and Pattern Recognition*, pages 710–716, 1988.

A Conic Curve Description

A plane conic curve in its generality can be written in an algebraic implicit form as :

$$f(x,y) = Ax^2 + Bxy + Cy^2 + Dx + Ey + F = 0 \tag{36}$$

or equivalently:

$$f(x,y) = z^T Q z + z^T h + F = 0 \tag{37}$$

where:

$$Q = \begin{bmatrix} A & B/2 \\ B/2 & C \end{bmatrix} \qquad z = \begin{bmatrix} x \\ y \end{bmatrix} \qquad h = \begin{bmatrix} D \\ E \end{bmatrix}$$

This is a homogeneous equation in six variables. Hence, it constrains its defining six parameters only upto a scale factor. An advantage of representing a conic in this form is that the particular type of conic emerges out of the parameters obtained through a best fit. A negative *discriminant*, $B^2 - 4AC$, implies that the conic is an ellipse, a positive one implies that it is a hyperbola and an identically zero discriminant is a parabolic form. Degenerate forms like the linear and circular are special instances of parabolic and elliptical forms, respectively. One requirement for a curve-fitting routine is that the parameters of the fit be covariant with respect to rotational and translational transformations of the coordinate system [3]. That is, if we rotate and translate the image coordinate system through a rotation R and a translation T which transforms each point z into z' as $z = Rz' + T$. Then Q, h and F should transform as :

$$Q' = R^T Q R \qquad h' = 2R^T Q T + Rh \qquad F' = T^T Q T + T^T h + F$$

In other words, if the fitting algorithm computes Q, h and F as the conic parameters in the original coordinate system, then in the transformed coordinate system, the resultant parameters should be Q', h' and F'. One way to achieve this is to use the Euclidean distance between a point and the corresponding curve as an error measure which when minimized over the conic parameters leads to the optimum parameters. But the distance of a point to a conic is expressible exactly as a quartic equation [15] which is not useful for the purposes of formulating an optimization function. So we try to approximate the distance using simpler measures.

A.1 Algebraic Distance Measure

One simple error measure which can be used for conic fitting is the algebraic distance measure. It is assumed that $|f(x_i, y_i)|$ is an approximation to the distance measure. Then, $\sum_i f^2(x_i, y_i)$ is minimized over the whole set of sponsoring points in the six-dimensional parameter space. Of course, a constraint on the six defining parameters is imposed to fix the scale. The algebraic measure $f^2(x_i, y_i)$ is a scalar, invariant to the coordinate transformations. So the variation of the constraining norm determines if the fitting algorithm using this algebraic distance measure is covariant with the coordinate transformations.

A straightforward norm on the six parameters is that these form a unit vector. Then the problem is :

$$\min_{\mathbf{p}} \ \mathbf{p}^T S \mathbf{p} \qquad subject\ to \qquad \mathbf{p}^T \mathbf{p} = 1 \tag{38}$$

where:

$$\begin{aligned}
\mathbf{p} &= [\ A\ B\ C\ D\ E\ F\ ]^T \quad is\ the\ six-dimensional\ parameter\ vector,\\
\mathbf{m}_i &= [\ x_i^2\ \ x_i y_i\ \ y_i^2\ \ x_i\ \ y_i\ \ 1\ ]^T \quad is\ the\ image\ measurement\ vector,\ and\\
S &= \sum_i \mathbf{m}_i \mathbf{m}_i^T \quad is\ the\ scatter\ matrix\ whose\ each\ term\ is\ the\ dyadic\ product\\
&\quad of\ a\ measurement\ vector.
\end{aligned}$$

The solution to this minimization problem [11] is the unit-norm eigenvector corresponding to the smallest eigenvalue of S. Unfortunately, this simple scheme does not satisfy our covariance requirement. The unit-norm condition on the parameter vector is not covariant to coordinate transformations.

Bookstein [8] suggests the use of inherently covariant measures of a conic as constraining norms. It is well known that the forms $A+C$ and B^2-4AC are covariant under the Euclidean group. The only positive definite invariant that can be formed from these quantities is $(A+C)^2+(B^2-4AC)/2 = A^2+B^2/2+C^2$. Bookstein's suggests setting the value of this norm to 2. So the optimization problem becomes:

$$\min_{\boldsymbol{p}} \ \boldsymbol{p}^T S \boldsymbol{p} \qquad \textit{subject to} \qquad \boldsymbol{p}^T N \boldsymbol{p} = 1 \tag{39}$$

where:

S is the earlier scatter matrix and N = diag(1, 1/2, 1, 0, 0, 0).

We can solve this in closed-form by partitioning matrices S and N and vector $\boldsymbol{p}$. Let,

$$S = \begin{bmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} \end{bmatrix} \qquad N = \begin{bmatrix} N1 & 0 \\ 0 & 0 \end{bmatrix} \qquad N1 = diag(\ 1,\ 1/2,\ 1\ ) \qquad \boldsymbol{p} = [\boldsymbol{p_1}^T \ \ \boldsymbol{p_2}^T]^T$$

where the matrices S and N are partitioned as 3-by-3 sub-matrices and the vector $\boldsymbol{p}$ is partitioned into 3-by-1 vectors. Thus, we have the following problem now:

$$\min_{\boldsymbol{p}} \ \boldsymbol{p}^T S \boldsymbol{p} = \boldsymbol{p_1}^T S_{11} \boldsymbol{p_1} + 2\boldsymbol{p_1}^T S_{12} \boldsymbol{p_2} + \boldsymbol{p_2}^T S_{22} \boldsymbol{p_2} \qquad \textit{subject to} \qquad \boldsymbol{p_1}^T N_1 \boldsymbol{p_1} = 2 \tag{40}$$

For any fixed $\boldsymbol{p_1}$, $\boldsymbol{p}^T S \boldsymbol{p}$ is minimal when,

$$\boldsymbol{p_2} = -S_{22}^{-1} S_{12} \boldsymbol{p_1} \tag{41}$$

This implies,

$$\boldsymbol{p}^T S \boldsymbol{p} = \boldsymbol{p}_1^T (S_{11} - S_{12} S_{22}^{-1} S_{21}) \boldsymbol{p}_1 = \boldsymbol{p}_1^T S_{11\_2} \boldsymbol{p}_1 \tag{42}$$

This is to be minimized with $p_1^T N_1 p_1 = 2$. The solution to this is the eigenvector of matrix $N_1^{-1} S_{11\_2}$ corresponding to that eigenvalue which minimizes the fit error.

Results of using this closed-form method to describe conic fits to a sample set of individual point tracks of the box sequence are shown in Fig. 17. As is clear from the picture, although the fitted conic curve passes close to the sponsoring points, yet the fit parameters are extremely poor. Specifically, they fail to make explicit the common axis of rotation as it emerges in the image plane. There seems nothing common amongst the set of curve fits. Moreover, most of the fits we tried using the closed-form algorithm on various point trajectories not even resulted in elliptic fits but hyperbolic ones. Only the few elliptic fits obtained are shown in the figure.

A.2 First Order Distance Measure

One problem, in general, with the algebraic distance measure examined above is that it underestimates the distance to a conic in high curvature regions of the conic [21]. The level curves of the algebraic distance measure, $f(x,y)$, are closely packed in low-curvature regions and sparsely packed in high-curvature ones as illustrated in Figs. 18a and 18b taken from [21]. So, when there is a considerable noise in the data, Bookstein's algorithm, based on $f(x,y)$, may result in a bad fit as it tries to pass closely the points in relatively "flat" regions tolerating bigger deviations in the more "curved" regions.

An improvement to the algebraic distance measure is a first order distance measure. Assuming that the points in the input data lie close to a level curve of $f(x,y)$ with level 0, it can be assumed that $\nabla f(x_i, y_i)$ is a good approximation to the gradient at (x_i, y_i) even when (x_i, y_i) does not exactly lie on $f(x,y) = 0$. Given this observation, a first order approximation to the distance of a point from the conic is $|f(x_i, y_i)/\,|\nabla f(x_i, y_i)||$. We can use the sum of squares of these distances over all the points i as an error measure to be minimized by the fitting algorithm. Note that no explicit constraint on the fit parameters need be imposed in this method as the scale factor gets factored out between the ratio of the two terms. Now the unconstrained minimization problem is:

$$\min_{(A,B,C,D,E,F)} \sum_i (f(x_i, y_i)/\,|\nabla f(x_i, y_i)|)^2 \tag{43}$$

where $f(x, y)$ is as in Equation (36) and

$$\begin{aligned} \nabla f(x,y) &= \frac{\partial f(x,y)}{\partial x} i + \frac{\partial f(x,y)}{\partial y} j \\ &= (2Ax + By + D)i + (2Cy + Bx + E)j \end{aligned} \tag{44}$$

We carry out this minimization using the BFGS–DFP quasi–Newton unconstrained optimization routine as implemented in *Numerical Recipes in C* [30]. The solution computed by Bookstein's algorithm is used as an initial guess to this algorithm. Convergence is very fast. Results on a few point tracks are shown in Fig. 11. Most of the earlier hyperbolic fits do become elliptic now. Therefore is an improvement in the fit parameters. But still as far as making explicit the common rigid body motion is concerned, the fits are absolutely unreliable. Reliable fitting algorithm which exploits the common constraints across various point trajectories has already been presented in the main body of this paper.

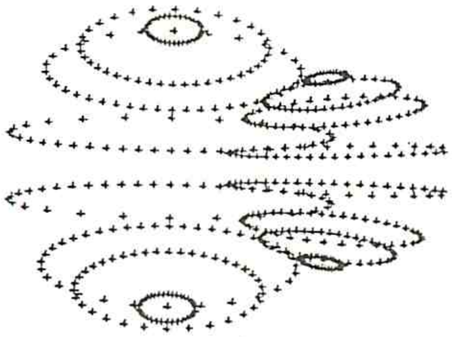

Fig. 1: Image point trajectories of two spheres rotating independently at different depths.

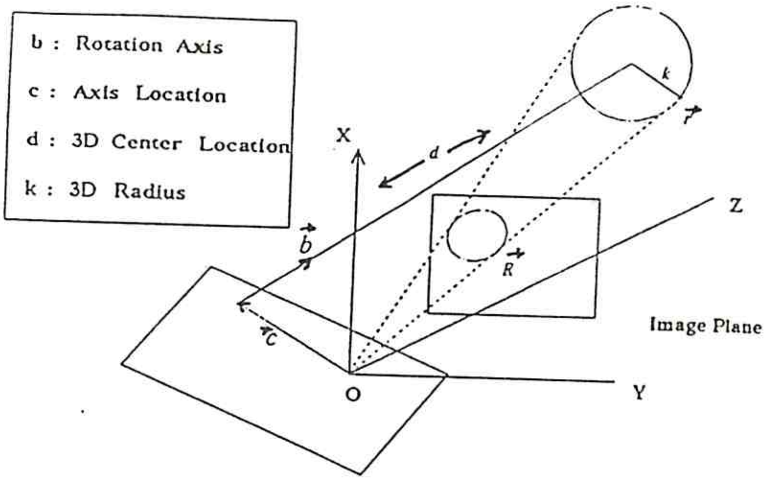

Fig. 2 : Geometry of a point's rotation

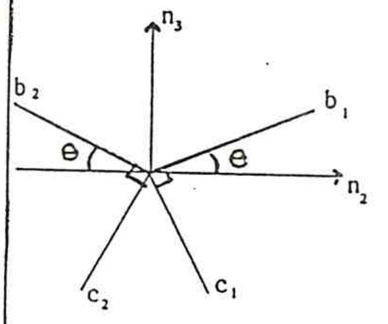

Fig. 3: Two final solutions for one point.

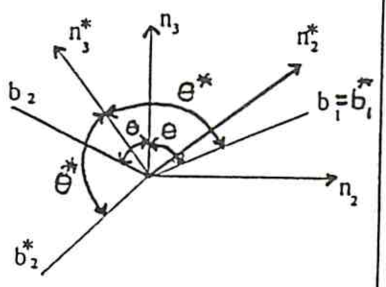

Fig. 4: Unique solution from two points.

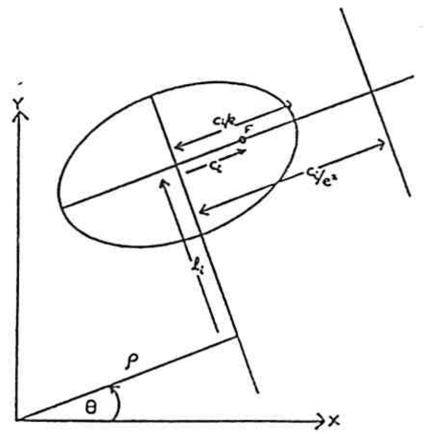

Fig. 5: Conic section parameterization illustrated for an ellipse

Table 1: Results for Four imaged 3D points rotating rigidly around an axis

Data Type		Axis dir.			Axis locn.			True Structure		Computed Structure	
		b_x	b_y	b_z	c_x	c_y	c_z				
True		0.577	0.577	0.577	-0.603	-0.176	0.778	d	k	d	k
without noise in imaged data											
Pt 1	1	0.577	0.577	0.577	-0.603	-0.176	0.778	0.986	0.497	0.986	0.497
	2	-0.535	-0.111	0.837	0.640	0.593	0.488				
Pt 2	1	0.577	0.577	0.577	-0.603	-0.176	0.778	0.381	0.363	0.381	0.363
	2	-0.835	-0.525	0.168	0.004	0.298	0.955				
Pt 3	1	0.577	0.577	0.577	-0.603	-0.176	0.778	0.768	0.168	0.768	0.168
	2	-0.724	-0.310	0.616	0.415	0.518	0.748				
Pt 4	1	0.577	0.577	0.577	-0.603	-0.176	0.778	1.682	0.322	1.682	0.322
	2	-0.235	0.135	0.962	0.801	0.588	0.113				
with (-1,1) pixel uniform noise in imaged data											
Pt 1	1	0.570	0.577	0.585	-0.607	-0.183	0.773	0.986	0.497	1.003	0.502
	2	-0.529	-0.108	0.842	0.643	0.595	0.481				
Pt 2	1	0.579	0.579	0.573	-0.599	-0.174	0.781	0.381	0.363	0.378	0.363
	2	-0.834	-0.526	0.166	0.003	0.297	0.955				
Pt 3	1	0.582	0.550	0.560	-0.591	-0.166	0.790	0.768	0.168	0.736	0.174
	2	-0.730	-0.325	0.602	0.393	0.520	0.758				
Pt 4	1	0.583	0.577	0.572	-0.600	-0.169	0.782	1.682	0.322	1.651	0.317
	2	-0.243	0.132	0.961	0.800	0.587	0.121				

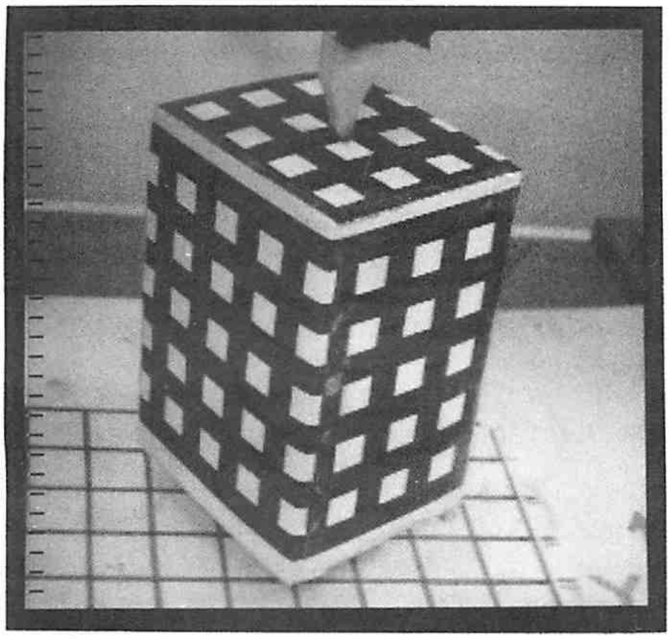

Fig. 6: Frame 1 of the box sequence

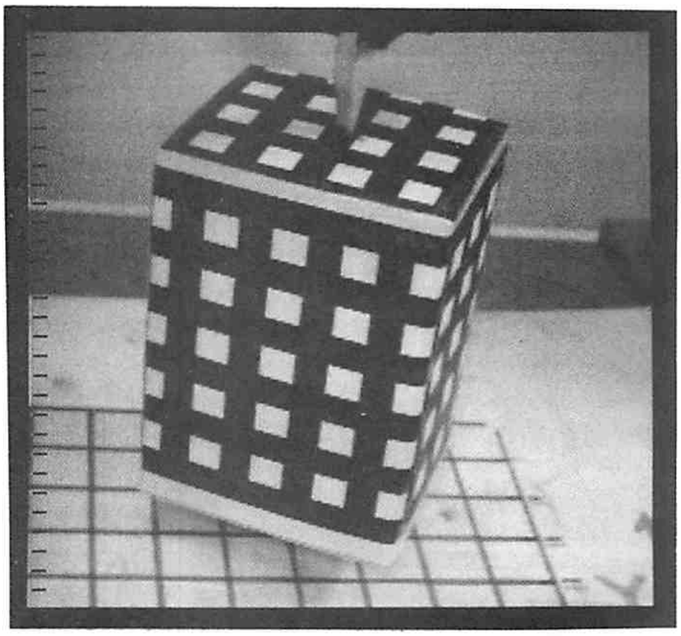

Fig. 7: Frame 10 of the box sequence

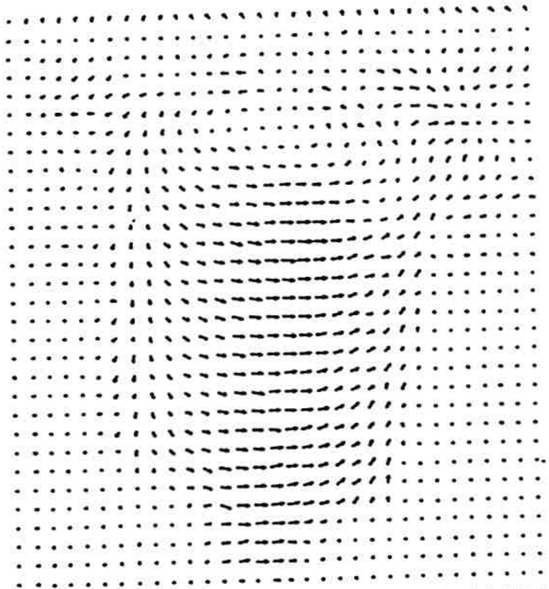

Fig. 8: Sub-sampled displacement field between frames 1 and 2

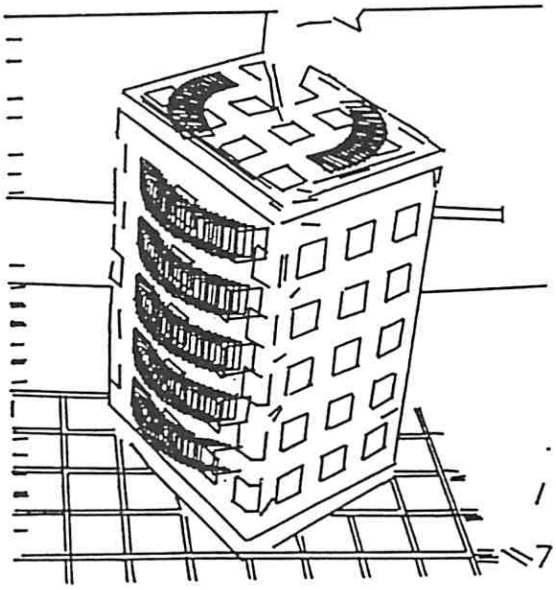

Fig. 9: Sample set of tracked lines

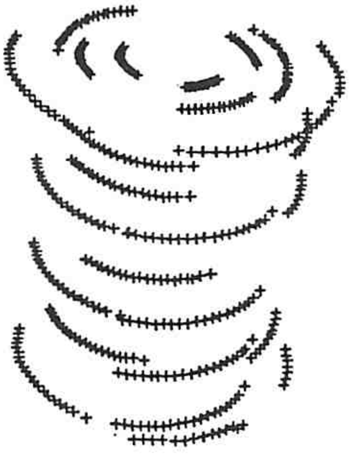

Fig. 10: Sample set of tracked line intersections

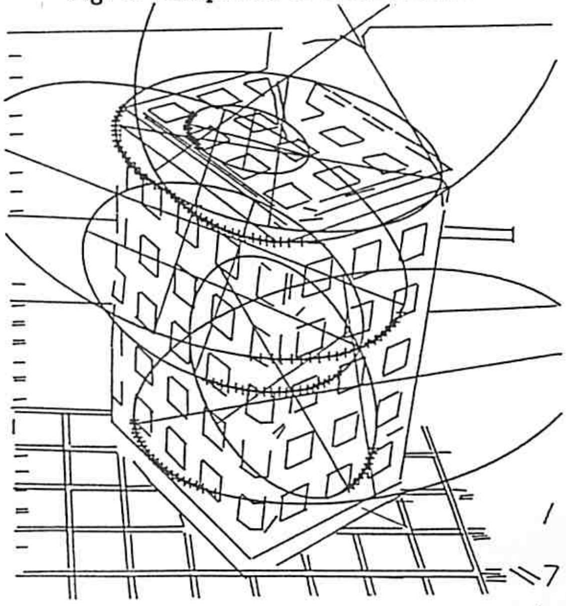

Fig. 11: Conic fits to individual 20-frame point tracks

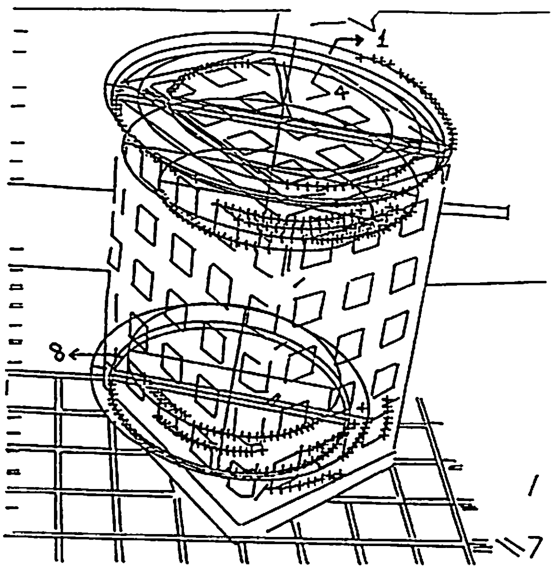

Fig. 12: Conic fits to groups 1 and 2 of 20-frame point tracks

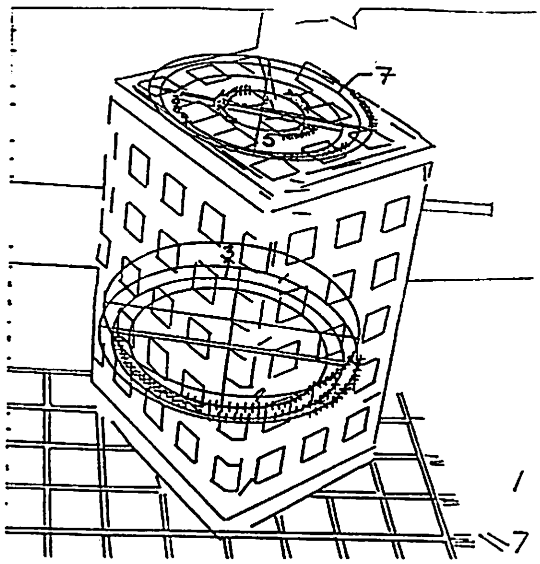

Fig. 13: Conic fits to groups 3 and 4 of 20-frame point tracks

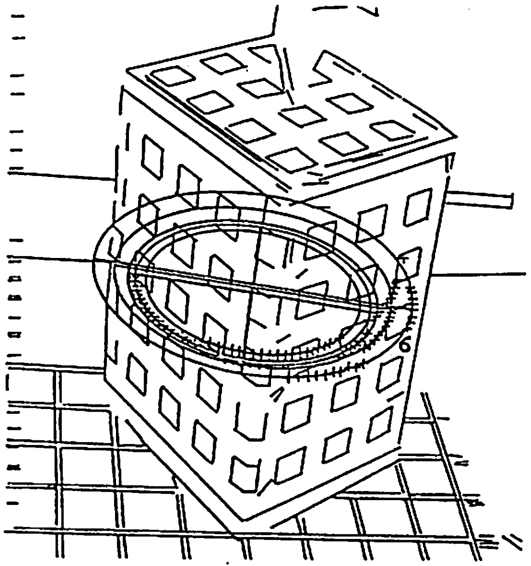

Fig. 14: Conic fits to group 5 of 20-frame point tracks

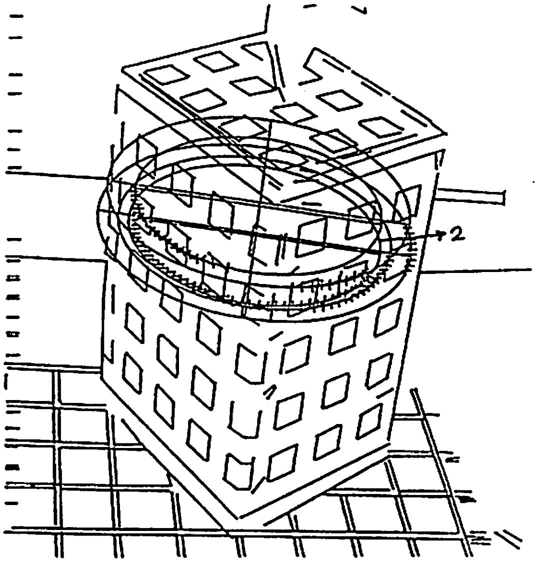

Fig. 15: Conic fits to group 6 of 20-frame point tracks

Table 2: Results for Eight sample points for the Box sequence

Data Type		Axis dir.			Axis locn.			Error(deg.)		True Structure		Computed Structure		Error(%)	
		b_x	b_y	b_z	c_x	c_y	c_z	b	c					d	k
True		0.102	-0.845	0.525	-0.051	0.522	0.851			d	k	d	k		
Pt 1	1	0.148	-0.836	0.528	-0.068	0.524	0.849	2.7	1.0	0.458	0.142	0.466	0.138	1.8	3.1
	2	-0.147	0.935	0.322	0.072	-0.314	0.947								
Pt 2	1	0.127	-0.838	0.530	-0.057	0.527	0.848	1.5	0.5	0.575	0.116	0.585	0.110	1.8	5.0
	2	-0.112	0.866	0.487	0.064	-0.480	0.873								
Pt 3	1	0.125	-0.849	0.513	-0.051	0.511	0.858	1.5	0.7	0.681	0.113	0.664	0.106	2.5	5.9
	2	-0.095	0.796	0.597	0.096	-0.590	0.802								
Pt 4	1	0.148	-0.835	0.531	-0.068	0.527	0.847	2.7	1.1	0.459	0.094	0.468	0.092	1.9	1.8
	2	-0.147	0.938	0.315	0.071	-0.308	0.949								
Pt 5	1	0.163	-0.841	0.515	-0.067	0.512	0.857	3.6	1.1	0.458	0.039	0.448	0.036	2.3	6.8
	2	-0.159	0.942	0.296	0.077	-0.287	0.955								
Pt 6	1	0.135	-0.838	0.528	-0.061	0.525	0.849	1.9	0.6	0.628	0.128	0.625	0.139	0.5	8.5
	2	-0.113	0.833	0.542	0.096	-0.533	0.840								
Pt 7	1	0.163	-0.840	0.517	-0.068	0.513	0.856	3.6	1.1	0.458	0.083	0.446	0.084	2.5	1.2
	2	-0.159	0.942	0.296	0.077	-0.287	0.955								
Pt 8	1	0.164	-0.849	0.503	-0.071	0.499	0.864	3.8	1.9	0.733	0.129	0.701	0.107	4.3	17.25
	2	-0.121	0.753	0.646	0.131	-0.633	0.763								

Estimates of true b and c after clustering data from 40 trajectories

b : 0.144 -0.842 0.520 err : 2.5° c : -0.063 0.516 0.854 err : 0.8°

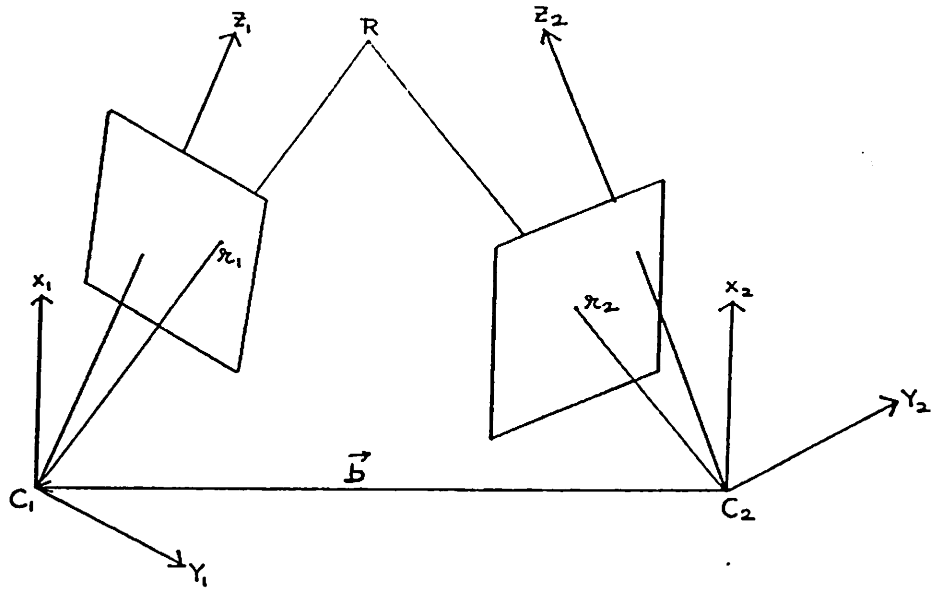

Fig. 16: Coplanarity Constraint for a Scene Point
Imaged from Two Different Camera Positions

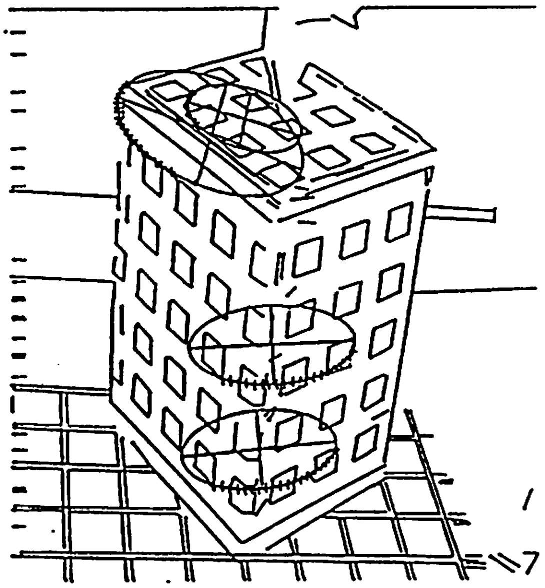

Fig. 17: Closed-form fits to Individual Point Tracks

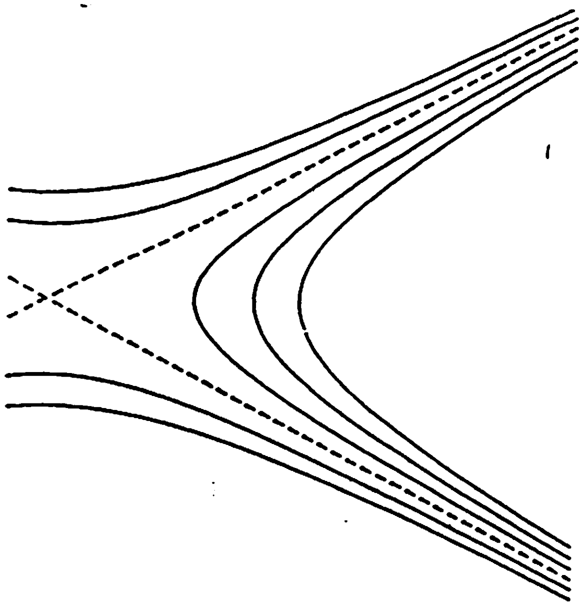

Fig. 18a: Hyperbolic Level Curves

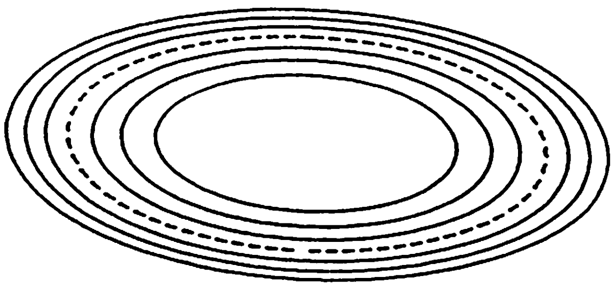

Fig. 18b: Elliptic Level Curves

Table 3: **Two-frame 3D Motion Parameter Estimation Comparisons**

Algorithm	Rotation Axis			Error (deg.)	Angle (deg.)	Error (%)	Trans. Vector			Error (deg.)
	a_x	a_y	a_z				b_x	b_y	b_z	
Frames 1-2										
Pose	-0.147	0.833	-0.533		3.63		-0.989	-0.143	0.047	
Horn	-0.133	0.850	-0.509	1.83	3.82	5.32	-0.991	-0.126	0.047	1.03
Adiv	-0.120	0.855	-0.505	2.52	3.85	6.22	-0.992	-0.127	0.028	1.48
Frames 1-3										
Pose	-0.145	0.827	-0.544		7.16		-0.989	-0.124	0.080	
Horn	-0.141	0.851	-0.506	2.55	7.69	7.36	-0.990	-0.117	0.082	0.42
Adiv	-0.114	0.856	-0.504	3.34	7.73	7.92	-0.992	-0.120	0.044	2.10
Frames 1-4										
Pose	-0.153	0.828	-0.540		10.99		-0.988	-0.115	0.102	
Horn	-0.142	0.816	-0.561	1.53	10.46	4.89	-0.990	-0.105	0.099	0.58
Adiv	-0.103	0.826	-0.554	2.95	10.60	3.56	-0.993	-0.109	0.055	2.75
Frames 1-5										
Pose	-0.155	0.825	-0.543		14.66		-0.987	-0.100	0.129	
Horn	-0.148	0.825	-0.545	0.44	14.38	1.89	-0.988	-0.093	0.125	0.48
Adiv	-0.095	0.842	-0.531	3.60	14.79	0.86	-0.993	-0.098	0.060	3.98
Frames 1-6										
Pose	-0.149	0.825	-0.545		17.87		-0.986	-0.077	0.151	
Horn	-0.146	0.816	-0.559	0.94	17.38	2.72	-0.986	-0.076	0.148	0.17
Adiv	-0.081	0.832	-0.550	3.93	17.67	1.09	-0.994	-0.082	0.075	4.35
Frames 1-7										
Pose	-0.147	0.829	-0.540		21.35		-0.982	-0.059	0.179	
Horn	-0.150	0.817	-0.557	1.21	20.73	2.92	-0.984	-0.063	0.166	0.80
Adiv	-0.072	0.837	-0.543	4.31	21.33	0.10	-0.994	-0.071	0.077	5.91

Table 4: Two-frame Depth Estimation Comparisons

Point Num.	Pose Depth	Frames 1-2				Frames 1-3			
		Horn	Error(%)	Adiv	Error(%)	Horn	Error(%)	Adiv	Error(%)
1	631.40	593.90	5.94	585.69	7.24	577.70	8.51	570.05	9.72
2	664.64	617.06	7.16	609.60	8.28	604.76	9.01	599.09	9.86
3	682.86	630.64	7.65	621.62	8.97	618.30	9.45	609.98	10.67
4	635.86	598.02	5.95	591.16	7.03	581.98	8.47	577.17	9.23
5	708.14	666.16	5.93	656.02	7.36	641.78	9.37	632.64	10.66
6	582.51	552.39	5.17	546.97	6.10	537.23	7.77	533.97	8.33
7	600.49	591.09	1.57	584.11	2.73	575.40	4.18	570.02	5.07
8	655.57	580.57	11.44	575.09	12.28	561.76	14.31	559.24	14.69
9	611.95	581.78	4.93	577.27	5.67	560.73	8.37	560.16	8.46
10	616.81	592.19	3.99	586.77	4.87	566.03	8.23	564.04	8.56
11	629.51	594.78	5.52	587.28	6.71	578.12	8.16	571.87	9.16
12	627.04	568.51	9.33	562.74	10.25	554.14	11.63	550.56	12.20
13	651.08	600.07	7.83	591.77	9.11	591.16	9.20	583.45	10.39

Table 5: Two-frame Depth Estimation Comparisons

Point Num.	Pose Depth	Frames 1-4				Frames 1-5			
		Horn	Error(%)	Adiv	Error(%)	Horn	Error(%)	Adiv	Error(%)
1	631.40	675.73	7.02	656.72	4.01	645.34	2.21	613.43	2.85
2	664.64	711.26	7.01	694.37	4.47	681.31	2.51	651.32	2.00
3	682.86	732.56	7.28	711.04	4.13	698.28	2.26	662.52	2.98
4	635.86	679.22	6.82	664.74	4.54	649.93	2.21	624.24	1.83
5	708.14	764.97	8.02	741.25	4.67	728.91	2.93	689.72	2.60
6	582.51	620.71	6.56	609.67	4.66	596.73	2.44	576.12	1.10
7	600.49	668.82	11.38	653.49	8.83	642.24	6.95	615.09	2.43
8	655.57	656.35	0.12	645.78	1.49	629.32	4.00	608.95	7.11
9	611.95	654.03	6.88	645.92	5.55	629.07	2.80	611.32	0.10
10	616.81	661.34	7.22	651.03	5.55	635.52	3.03	614.94	0.30
11	629.51	673.04	6.92	656.30	4.26	644.77	2.42	615.78	2.18
12	627.04	641.07	2.24	629.09	0.33	614.94	1.93	592.79	5.46
13	651.08	694.29	6.64	674.74	3.63	664.41	2.05	631.60	2.99

Table 6: Two-frame Depth Estimation Comparisons

Point Num.	Pose Depth	Frames 1-6				Frames 1-7			
		Horn	Error(%)	Adiv	Error(%)	Horn	Error(%)	Adiv	Error(%)
1	631.40	654.77	3.70	628.51	0.46	657.50	4.13	618.29	2.08
2	664.64	689.83	3.79	667.20	0.39	694.42	4.48	658.52	0.92
3	682.86	710.56	4.06	681.40	0.21	713.95	4.55	670.41	1.82
4	635.86	663.28	4.31	644.60	1.38	665.04	4.59	635.15	0.11
5	708.14	739.87	4.48	707.80	0.05	744.84	5.18	697.09	1.56
6	582.51	605.76	3.99	591.53	1.55	607.83	4.35	584.03	0.26
7	600.49	649.91	8.23	629.33	4.80	654.61	9.01	622.06	3.59
8	655.57	638.86	2.55	626.24	4.47	644.11	1.75	621.41	5.21
9	611.95	639.21	4.45	629.60	2.88	642.29	4.96	622.80	1.77
10	616.81	644.30	4.46	631.49	2.38	649.23	5.27	625.81	1.46
11	629.51	654.09	3.91	631.30	0.28	657.09	4.38	622.00	1.19
12	627.04	623.28	0.60	607.82	3.07	626.85	0.03	601.03	4.15
13	651.08	675.32	3.72	648.55	0.39	679.01	4.29	638.90	1.87

Table 7: 3D Motion Parameter Comparisons between Our and 2-Frame Estimates averaged over 6 Frame Pairs

Algorithm	Rotation Axis			Error	Axis Location			Error
	a_x	a_y	a_z	(deg.)	b_x	b_y	b_z	(deg.)
Pose	0.104	-0.840	0.533		-0.054	0.530	0.846	
Horn	0.143	-0.830	0.540	2.36	-0.066	0.536	0.842	0.85
Adiv	0.098	-0.842	0.531	0.39	-0.144	0.517	0.844	5.3
Own	0.144	-0.842	0.520	2.43	-0.063	0.516	0.854	1.06

Table 8: Depth Comparisons between Our and 2-Frame Estimates Averaged over 6 Frame Pairs

Point	True Depth	Horn	Error(%)	Adiv	Error(%)	Own	Error(%)
1	631.40	634.16	0.44	612.11	3.05	640.66	1.47
2	664.64	666.44	0.27	646.68	2.70	684.72	3.00
3	682.86	684.05	0.17	659.50	3.42	668.09	2.16
4	635.86	639.58	0.59	622.84	2.05	623.74	1.91
5	708.14	714.42	0.89	687.42	2.93	678.03	4.25
6	582.51	586.77	0.73	573.72	1.51	574.96	1.30
7	600.49	630.35	4.97	612.35	1.98	620.08	3.26
8	655.57	618.49	5.65	606.12	7.54	615.62	6.10
9	611.95	617.85	0.96	607.85	0.67	640.43	4.65
10	616.81	624.77	1.29	612.35	0.72	611.56	0.85
11	629.51	633.65	0.66	614.09	2.45	617.61	1.90
12	627.04	604.80	3.55	590.67	5.80	593.53	5.30
13	651.08	650.71	0.06	628.17	3.52	662.84	1.80

Table 9: Two Solutions from Horn's Algorithm with Error Measures

Frames	Solutions	Rotation Axis			Angle	Trans. Vector			Error
		a_x	a_y	a_z	(deg.)	b_x	b_y	b_z	Measure
1–2	Correct	-0.133	0.850	-0.509	3.82	-0.991	-0.126	0.047	2.502e-6
	Wrong	-0.015	-0.097	-0.995	1.70	-0.160	0.021	0.987	7.061e-7
1–3	Correct	-0.141	0.851	-0.506	7.69	-0.990	-0.117	0.082	2.606e-6
	Wrong	-0.015	-0.090	-0.996	3.66	-0.271	0.001	0.963	2.385e-6
1–4	Correct	-0.142	0.816	-0.561	10.46	-0.990	-0.105	0.099	4.416e-6
	Wrong	-0.016	-0.084	-0.996	5.57	-0.286	0.002	0.958	5.853e-6
1–5	Correct	-0.148	0.825	-0.545	14.38	-0.988	-0.093	0.125	7.375e-6
	Wrong	-0.017	-0.086	-0.996	7.42	-0.249	0.010	0.968	1.082e-5
1–6	Correct	-0.146	0.816	-0.559	17.38	-0.986	-0.076	0.148	8.768e-6
	Wrong	0.006	-0.102	-0.995	7.84	-0.106	0.041	0.994	1.582e-5
1–7	Correct	-0.150	0.817	-0.557	20.73	-0.984	-0.063	0.166	1.585e-5
	Wrong	0.012	-0.104	-0.994	7.77	-0.097	0.044	0.994	2.272e-5