Pictures and Trails: a New Framework for the Computation of Shape and Motion From Perspective Image Sequences

Carlo Tomasi*

TR 93-1400
November 1993

Department of Computer Science
Cornell University
Ithaca, NY 14853-7501

* This research supported by the National Science Foundation under contract IRI-9201751.

Pictures and Trails: a New Framework for the Computation of Shape and Motion from Perspective Image Sequences

Carlo Tomasi[1]

December 1993

[1]This research was supported by the National Science Foundation under contract IRI-9201751.

Abstract

This report presents a new framework for the computation of shape and motion from a sequence of images taken under perspective projection. The framework is based on two abstractions, the *picture* and *trail* loci, that represent respectively the set of all pictures of the same scene and the set of all trails that a point in the world can leave on the image for a given camera trajectory. These abstractions lead to a remarkably clean relation between perspective and orthography. Furthermore, image motion is described in terms of *angles* between projection rays, thereby eliminating the need to model camera rotation and leading to more stable results. A numerically sound, global minimization method is developed, based on this framework, for the case of a two-dimensional world, but all concepts also hold in three dimensions. Experiments show that the method is rather immune to noise but critically dependent on camera calibration.

Chapter 1

Introduction

An important goal of computer vision is to build reliable systems for the computation of structure and motion from the images produced by a moving camera. If the world is stationary and if feature points can be tracked from image to image, the computation of structure and motion becomes a purely geometric problem. It is, however, a nonlinear and potentially poorly conditioned one. Conditioning must be addressed by formulating the problem in terms of well-observable parameters only, using generously redundant data, and paying close attention to the numerical aspects of the computation. Nonlinearity, on the other hand, must be addressed by a global solution method, that is, one that does not get caught in local minima.

This report presents a formulation of the problem of computing shape and motion from a sequence of images of a rigid scene under perspective projection. This formulation addresses all the issues mentioned above, as highlighted in the following.

No rotation in the model The proposed model of the imaging situation is independent of the camera rotation around its optical center. This is achieved by describing image changes through the *angles* between the projection rays of point features, similarly to what is done in [TS93]. When the camera rotates, these angles do not change. In contrast, in the traditional framework, rotation and translation can be mistaken for each other, thereby leading to poor observability of the translation-rotation pair and the known sensitivity problems of the standard approaches.

Multiframe and multipoint The new formulation can handle any sufficiently large number of feature points and camera positions. In fact, the first proposed step is to use the available images to build the locus of *all* possible perspective images of the same scene. This locus, called the *picture locus*, turns out to be a three-dimensional variety in a space with roughly as many dimensions as there are features in the set. The images in a specific sequence are then points on the picture locus.

Global minimization The new approach splits the computation into a linear stage in the space of all the data and nonlinear stage in a space with a fixed and small number of dimensions, representing all possible affine deformations of the scene. In this small space, the global minimum can be at least approximately identified by dense sampling.

Perspective vs orthography This two-stage partition of the computation was made possible by a fundamental insight about the picture locus: the subspace tangent to the locus at the origin is the set of all *orthographic* images of the same scene. This insight, in a sense, reduces the problem of shape and motion under perspective to that of shape and motion under orthography, a link that is interesting *per se* even besides the computational methods that it suggests.

Appropriate numerical techniques The two-stage approach allows using the most appropriate techniques for every part of the computation: linear data fitting in the large space of the input data, and efficient variable projection methods in the small space of affine deformations.

Incidentally, the first stage yields shape and motion up to two separate affine transformations. In many applications [WT93] this is sufficient, and the second, more expensive stage that enforces Euclidean metric can be omitted.

In this report, a flat, two-dimensional world is considered, and this for two reasons. First, although all the concepts hold also in three dimensions, the extension is technically less than straightforward, and has not addressed in detail yet. Second, all the concepts introduced are more easily visualized in the two dimensional case, where the picture locus becomes a picture *surface.*

The next two sections present the main abstractions of the framework: the picture surface, and the *trail surface*, a dual concept that will be introduced later on. Section 4 then outlines the reconstruction method. Experiments are discussed in section 5. First, a series of simulations shows that the method works well even with substantial uncertainty in the image measurements. Then, an experiment with a real image sequence gives mixed results, supporting the conjecture that camera calibration is critical for good results.

Chapter 2

The Picture Surface

The plane at the bottom of figure 2.1 represents the two-dimensional world where both camera and scene are supposed to live. The camera looks at a set of point features and only records the tangents of the angles formed by pairs of projection rays. In this two-dimensional case, all the pairs of features have one reference feature in common, so with $P+1$ feature points there are P tangents per frame (in the figure, $P=3$ for visualization purposes). Thus, one feature serves as a landmark and the image positions of the other features are specified by the angles between their projection rays and that of the landmark feature. The tangent t of each angle is given by (see Appendix A)

$$t = \frac{uz - wx}{1 - ux - wz} \tag{2.1}$$

where (x, z) is the position of the feature in the world and

$$K = (u, w) = C/|C|^2 \tag{2.2}$$

is the vector obtained by reflecting the camera coordinates C across the unit circle.

With $P+1$ world feature points, an image from reflected camera position $K = (u, w)$ yields a set of P measurements $t_1, \ldots, t_P$:

$$t_p = \frac{uz_p - wx_p}{1 - ux_p - wz_p} \tag{2.3}$$

that can be collected into one vector $\mathbf{t} = (t_1, \ldots, t_P)$. This vector can be viewed as a point in a P-dimensional space. As the camera moves, the point $\mathbf{t}$ moves within this space. The locus of all possible points $\mathbf{t}$ for a fixed set of world features is a surface, traced by the parameters u, w and whose P components are given in parametric form by equation (2.3). This surface is called the *picture surface*. Notice that the picture surface does not depend on camera position, since it represents the images of the given features from all possible camera positions.

As an example, figure 2.2 shows a region of the picture surface for the four features $S_0 = (0,0)$, $S_1 = (0,4,0.8)$, $S_2 = (0.7,0.1)$, $S_3 = (0.2,0.5)$ of figure 2.3 when the camera moves in the region defined by the rectangle with vertices $K_0 = (-1,-1)$ and $K_1 = (-1,-0.5)$ in the K plane, corresponding to camera positions C on the grid in figure 2.3. This grid is in one-to-one correspondence with the grid on the picture surface of figure 2.2. Surfaces for more features cannot be visualized (except by projecting them to subspaces), but are still two-dimensional objects, because they are traced by two parameters.

The picture surface is univocally related to the positions of the feature points in space: different scenes yield different surfaces, and different points on the same surface represent different pictures of the same scene.

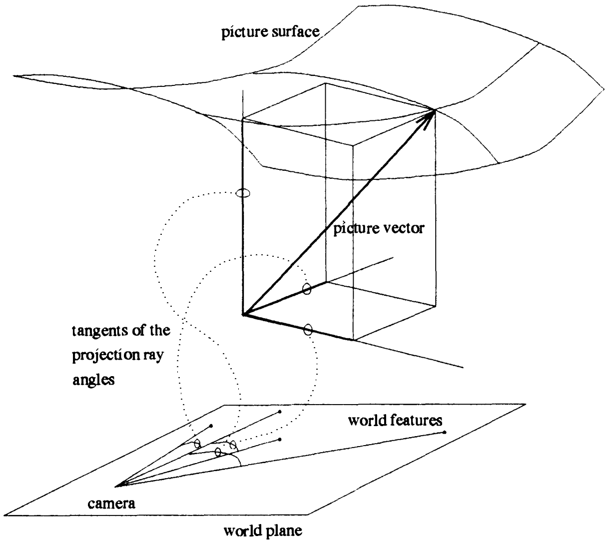

Figure 2.1: The components of a point on the picture surface (a picture vector) are the tangents of the projection ray angles.

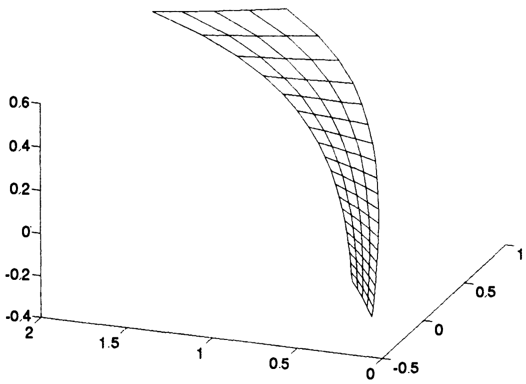

Figure 2.2: The picture surface for the four features in figure 2.3. The patch displayed here corresponds to the camera positions shown in figure 2.3.

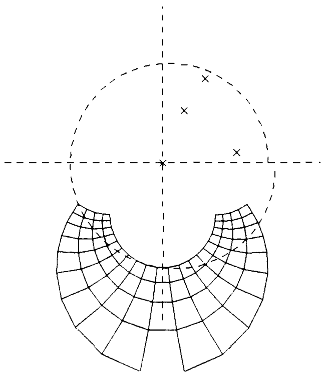

Figure 2.3: When the inverse camera coordinates K defined in equation (2.2) vary in the rectangle with vertices $K_0 = (-1, -1)$ and $K_1 = (-1, -0.5)$, the camera positions C move on the grid in this figure. The four crosses represent four features in the world, with the point at the origin being the reference feature.

Section 4 shows that the picture surface can be determined by a linear data fitting procedure from the available image measurements. Unfortunately, the relation between the surface parameters, resulting from fitting, and the coordinates of the world features that correspond to this surface is complicated. The brute-force approach to establishing this relation leads to a nonlinear constrained minimization problem of difficult solution. To avoid this problem, we now introduce an important result about the picture surface (see Appendix B for a proof).

> **Theorem (Orthographic Picture Plane)** *The plane tangent to the picture surface at the origin represents all the images of the same world features under orthography, up to a scale factor.*

This theorem is important because any two distinct orthographic images of a given set of features are the x and z coordinates of the features in the world except only for an affine transformation. In other words, we just need to pick any two points (not colinear with the origin) on the orthographic plane to obtain structure up to an affine transformation.

Thus, we start to see the outline of the shape reconstruction method:

1. find the picture surface by linear data fitting
2. determine the orthographic plane to obtain shape up to an affine transformation
3. replace the results into the original projection equations (2.1) to compute actual shape.

The third step, however, can only be performed once partial motion information has been computed. To this end, we introduce the concept of a *trail surface*, and an important duality result that links shape and motion.

Chapter 3

The Trail Surface and Duality

The picture surface is the set of images obtained by fixing the scene and moving the camera around. Conversely, one can determine a *trail locus* by instead fixing a number of camera positions and collecting the images of a single feature in the world. For each feature in the world, the image measurements from the given camera positions represent the trail that that feature left in the images as the camera covered those positions over time. When the world feature is changed, the trail vector moves on the trail locus.

If the projection equation (2.1) is examined, an important relation of duality can be established between the picture and trail surfaces. In fact, equation (2.1) is symmetric in structure and motion: the equation does not change with the replacements

$$u \leftrightarrow x \tag{3.1}$$

$$w \leftrightarrow -z\,. \tag{3.2}$$

Because of this symmetry in motion and structure, the surface of figure 2.2 can also be interpreted as a trail surface. Mathematically, this corresponds to fixing the camera positions in equation (2.1) rather than the world features, as done in equation (2.3). The image measurements of a point at $S = (x, z)$ as seen from F camera positions $K_1 = (u_1, w_1), \ldots, K_F = (u_F, w_F)$ are given by

$$t_f = \frac{u_f z - w_f x}{1 - u_f x - w_f z}\,. \tag{3.3}$$

with respect to the reference feature. These coordinates can be collected into a vector $\mathbf{t} = (t_1, \ldots, t_F)$, a point in an F-dimensional space. The locus of all possible measurement sets from those F cameras as the point $S = (x, z)$ varies in the world is the trail surface. To obtain the physical situation corresponding to this reinterpretation of the surface in figure 2.2, apply the replacements (3.1) and (3.2). This yields figure 3.1, where now circles represent camera positions and the grid points are the varying position of a feature in the world.

The orthographic-plane theorem holds for the trail surface as well. Because of its importance, we state it here with its proper, dual terminology.

Theorem (Orthographic Trail Plane) *The plane tangent to the trail surface at the origin represents all the trails from the same camera positions under orthography, up to a scale factor.*

Appendix B proves this result as well.

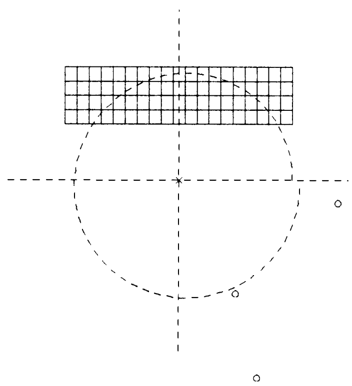

Figure 3.1: The surface of figure 2.2 can also be interpreted as the *trail* surface of the situation in this figure. The reference feature is still at the origin (cross).

Chapter 4

The Reconstruction Method

In this section, we reconstruct shape and motion from images in four steps:

1. determine the parameters of the picture surface by linear fitting;
2. pick two points on the orthographic plane of the picture surface to determine shape up to an affine transformation;
3. determine motion up to an affine transformation with the same technique;
4. replace these results into equation (2.1) to determine the affine transformations that map affine motion and shape into their Euclidean counterparts.

Affine shape and motion are computed with only linear operations, while the nonlinear part is all included in the last step.

4.1 The Parameters of the Picture Surface

We saw in section 2 that the image of a fixed set of points in the world is a point on a surface, the picture surface. Conversely, the image measurements of any one point in space as seen from a fixed set of cameras are a point on the trail surface. The picture and trail surfaces are embedded in highly dimensional spaces: with $P+1$ world points and F cameras, the picture surface lives in a P-dimensional space and the trail surface lives in an F-dimensional space.

Because we know the analytic form of these surfaces (equations (2.3) and (3.3)), determining their parameters from a set of image measurements is a data fitting problem. The problem becomes linear if we eliminate motion from equations (2.3) and shape from equation (3.3). We now show how to do this for the picture surface of equation (2.3), where motion is represented by the camera positions u, w.

Let us rewrite equation (2.3) for three distinct points numbered p, q, r:

$$t_p = \frac{uz_p - wx_p}{1 - ux_p - wz_p} \tag{4.1}$$

$$t_q = \frac{uz_q - wx_q}{1 - ux_q - wz_q} \tag{4.2}$$

$$t_r = \frac{uz_r - wx_r}{1 - ux_r - wz_r} . \tag{4.3}$$

We can solve the first two equations for the parameters u, w, and replace the result into the third equation. This yields an equation of the third degree in t_p, t_q, t_r. The coefficients of this equation depend only on shape, since the motion parameters u, w have been eliminated. These coefficients are easy to determine because they appear linearly in the equation.

To solve equations (4.1) and (4.2) above for u, w we multiply these equations by the denominators of their right-hand sides; this yields two linear equations in u, w, which can be written in matrix form as follows:

$$\begin{bmatrix} t_p x_p + z_p & t_p z_p - x_p \\ t_q x_q + z_q & t_q z_q - x_q \end{bmatrix} \begin{bmatrix} u \\ w \end{bmatrix} = \begin{bmatrix} t_p \\ t_q \end{bmatrix} .$$

This system can be solved by Cramer's rule and replaced into equation (4.3), rewritten as follows:

$$(t_r x_r + z_r)u + (t_r z_r - x_r)w = t_r .$$

This substitution yields the desired multilinear equation in t_p, t_q, t_r:

$$a_1 t_p t_q t_r + a_2 t_p t_q + a_3 t_p t_r + a_4 t_q t_r + a_5 t_p + a_6 t_q + a_7 t_r = 0 \tag{4.4}$$

where

$$\begin{aligned} a_1 &= -x_p(z_q - z_r) + x_q(z_p - z_r) - x_r(z_p - z_q) \\ a_2 &= -x_r(x_p - x_q) - z_r(z_p - z_q) \\ a_3 &= x_q(x_p - x_r) + z_q(z_p - z_r) \\ a_4 &= -x_p(x_q - x_r) - z_p(z_q - z_r) \\ a_5 &= -x_q z_r + z_q x_r \\ a_6 &= x_p z_r - z_p x_r \\ a_7 &= -x_p z_q + z_p x_q , \end{aligned} \tag{4.5}$$

and the subscripts p, q, r were dropped for simplicity from the coefficients a_i.

Because there are only six parameters in the right-hand sides of equations (4.5) but there are seven coefficients, the coefficients must satisfy some constraint. It is easy to verify that the following two linear equations hold:

$$a_1 - a_5 - a_6 - a_7 = 0 \tag{4.6}$$
$$a_2 + a_3 + a_4 = 0 \tag{4.7}$$

so that equation (4.4) can be rewritten as follows:

$$\begin{aligned} & a_2 t_p(t_q - t_r) + a_4 t_r(t_q - t_p) + a_5 t_p(1 + t_q t_r) \\ + \quad & a_6 t_q(1 + t_p t_r) + a_7 t_r(1 + t_p t_q) = 0 . \end{aligned} \tag{4.8}$$

Determining the coefficients a_i from a set of measurements over several frames is a linear, overconstrained minimization problem (Appendix C).

4.2 Affine Shape

Although easy to determine, the coefficients of the picture surface are complicated functions of the point coordinates $x_p, z_p, x_q, z_q, x_r, z_r$. Determining these coordinates from the coefficients directly from equations (4.5) is a hard nonlinear constrained minimization problem. However, it is trivial to find the orthographic plane of the picture surface. In fact, the constant term in equation (4.4) is equal to zero, so the picture surface passes through the origin, as expected, and the orthographic plane for the picture surface is given simply by the three linear terms in t_p, t_q, t_r, that is, by a_5, a_6, a_7. In other words, the equation of the orthographic plane is

$$a_5 t_p + a_6 t_q + a_7 t_r = 0 \; . \tag{4.9}$$

As discussed in [TK92], any two points on this plane, not colinear with the origin, represent shape up to an affine transformation. More specifically, let $\mathbf{t}^{(1)} = (t_p^{(1)}, t_q^{(1)}, t_r^{(1)})^T$ and $\mathbf{t}^{(2)} = (t_p^{(2)}, t_q^{(2)}, t_r^{(2)})^T$ be two points satisfying equation (4.9). For instance, let[1]

$$\begin{aligned} (\mathbf{t}^{(1)})^T &= (1, 0, -a_5/a_7) \\ (\mathbf{t}^{(2)})^T &= (0, 1, -a_6/a_7) \; . \end{aligned}$$

Then the four columns of the 2×4 matrix

$$\begin{bmatrix} 0 & (\mathbf{t}^{(1)})^T \\ 0 & (\mathbf{t}^{(2)})^T \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & \hat{x}_r \\ 0 & 0 & 1 & \hat{z}_r \end{bmatrix}$$

represent the coordinates of the origin and the three points numbered p, q, r up to an affine transformation. Because the first three columns are the affine system of reference (origin and two unit points), the only new information in the matrix above is given by the coordinates of the fourth point, that is, by

$$\begin{aligned} \hat{x}_r &= -a_5/a_7 \\ \hat{z}_r &= -a_6/a_7 \; . \end{aligned}$$

With more than four points, we repeat the procedure just described once for every value of r different from p and q, for a total of $P-2$ independent problems. This yields a $2 \times P$ matrix $\hat{S}$ of all the affine coordinates in the same reference system, because the origin and the two landmark points p and q are always mapped to $(0,0)$, $(1,0)$, $(0,1)$. For instance, with $p = 1$ and $q = 2$, we have

$$\hat{S} = \begin{bmatrix} 1 & 0 & \hat{x}_3 & \cdots & \hat{x}_P \\ 0 & 1 & \hat{z}_3 & \cdots & \hat{z}_P \end{bmatrix} \; .$$

Because affine coordinates differ from Euclidean coordinates only by an affine transformation, there must be a 2×2 matrix A such that

$$S = A\hat{S} \; .$$

We will determine A in section 4.4 below. Before that, however, we need to recover the camera motion up to an affine transformation.

4.3 Affine Motion

Because of the symmetry of equation (2.1) discussed in section 3 and expressed by equations (3.1) and (3.2), motion can be determined up to an affine transformation by the same procedure used

[1] It is easy to change this choice if $a_7 = 0$.

for shape in sections 4.1 and 4.2. Specifically, if the procedure for computing shape is summarized by the function

$$\hat{S} = \phi(T) \; ,$$

where T is the matrix of image measurements, then the $F \times 2$ matrix $\hat{K}$ that has the reflected affine coordinates (see equation (2.2)) as its rows is simply given by

$$\hat{K}^T = \phi(T^T) \; .$$

Duality saved us half of the work. Also, analogously to what happened for shape, if K is an $F \times 2$ matrix that collects the *Euclidean* coordinates of all these camera positions, reflected around the unit circle, there must be a 2×2 matrix B such that

$$K = \hat{K} B^T \; .$$

4.4 Euclidean Shape and Motion

To summarize, we now have affine shape, $\hat{S}$, and affine motion, $\hat{K}$. These two matrices of coordinates are expressed in two different coordinate systems, so we need to find two 2×2 matrices A and B that yield the Euclidean coordinates S and K according to the transformations

$$\begin{aligned} S &= A\hat{S} && (4.10) \\ K &= \hat{K} B^T \; . && (4.11) \end{aligned}$$

Notice that the origin of the coordinate system is fixed at the reference point $(x_0, z_0) = (0, 0)$. Because the image measurements do not constrain scale and an overall rotation of the reference system, we can impose the constraint that

$$(x_1, z_1) = (1, 0) \; .$$

Since $(\hat{x}_1, \hat{z}_1) = (1, 0)$, this constraint yields two of the entries of A:

$$a_{11} = 1 \quad \text{and} \quad a_{21} = 0 \; .$$

To find B and the remaining entries of A, we replace equations (4.10) and (4.11) into the original measurement equation (2.1). Ignoring point and camera subscripts, equation (2.1) becomes

$$t = \frac{(b_{11}\hat{u} + b_{12}\hat{w})a_{22}\hat{z} - (b_{21}\hat{u} + b_{22}\hat{w})(\hat{x} + a_{12}\hat{z})}{1 - (b_{11}\hat{u} + b_{12}\hat{w})(\hat{x} + a_{12}\hat{z}) - (b_{21}\hat{u} + b_{22}\hat{w})a_{22}\hat{z}}$$

which is separately linear in the two vectors $\alpha = (a_{12}, a_{22})$ and $\beta = (b_{11}, b_{12}, b_{21}, b_{22})$. In [TS93], we show a method for solving this type of equation, although applied to a different problem.

In conclusion, in the proposed method, a linear stage for affine structure and motion is followed by a nonlinear stage to determine the Euclidean metric. Because of this, the proposed method can be seen on one hand as a successor of techniques based on essential matrices pioneered by Longuet-Higgins [LH81], independently reinvented by Tsai and Huang [TH84] and surveyed in [May93]; and on the other hand it is a successor of the factorization method described in [TK92]. However, essential matrices work on two frames at a time, thereby either introducing a hard correspondence problem when the two frames are distant or leading to a poorly conditioned reconstruction when they are close. The multiframe factorization method, on the other hand, works only under orthographic projection, which limits its applicability to distant scenes and narrow fields of view. The current method, in contrast, is multiframe, multifeature, and works for perspective images. In addition, in contrast to multiframe and multifeature *local* methods such as [SA89], our method is global, in that it does not require an initial estimate of either structure or motion.

Chapter 5

Experiments

Figure 5.1 shows the result of a simulation with noisy images. Both true and computed structure and motion are shown. Noise on the image feature coordinates is Gaussian with a standard deviation of 0.5 pixels for a 512×512 image. In the simulation, both features and camera positions are scattered randomly, each in one quadrant of the plane. The two points at the origin and along the positive horizontal axis (at $(1,0)$) are the reference points, and their computed values are therefore exact.

The two plots in figure 5.2 show the structure and motion errors for increasing levels of noise. Ten features and camera positions are used in all experiments, and each experiment is repeated ten times with different random samples to produce ensemble averages. Structure errors are measured as the ratio between the average error per feature and the size of the bounding box of the true feature positions. A similar measure is used for the camera position errors.

Even with relatively few points and viewing positions, performance is good for subpixel noise levels. When the standard deviation of noise increases beyond one pixel, performance degrades sharply but continuously. We point out that in feature tracking the position of features can usually be determined with an accuracy of 0.1 or so pixels [TK91] for typical 512 by 512 images. From the plots of figure 5.2 we see that the corresponding structure and motion errors are a fraction of one percent.

With real images, the results are less satisfactory. The central part of figure 5.3 shows an epipolar slice (like the ones in [BBM87]) from a sequence of images taken with a Panasonic camera mounted on a micrometric translation and rotation stage. Figure 5.4 shows the setup from above, without the camera, which used to be on the platform visible at the bottom.

Features were obtained by detecting sharp intensity transitions in the first row of the epipolar slice and were tracked by continuity from one row to the next. Features were found on the leftmost block in figure 5.4, on the block closest to the camera at the center, and on the Crayola box on the right. Figure 5.5 shows the actual positions of the features (crosses) and of the camera (circles) as measured in figure 5.4.

No camera calibration was performed, and the nominal focal length of 16 mm was used, converted to pixels based on the manufacturer's specification of the size of the sensor's active area. The lens was a c-mount lens for surveillance applications, with consequently poor optical properties.

Figure 5.6 shows the reconstruction for the ten camera positions (circles) in the sequence and the eighteen features tracked in the epipolar slice. The camera motion is fairly accurately recovered, the overall distance between the camera and the scene is essentially correct, and each of the three feature groups is approximately of the right shape and size. However, the position of the three groups of features is considerably distorted with respect to the ideal positions of figure 5.5. The contrast between these results, with features tracked with about 0.1 pixels accuracy, and the simulations described in figures 5.1 and 5.2, run under greater positional uncertainty values, seems to support the conjecture that the camera calibration is crucial. We are working on camera calibration in order to verify this assertion.

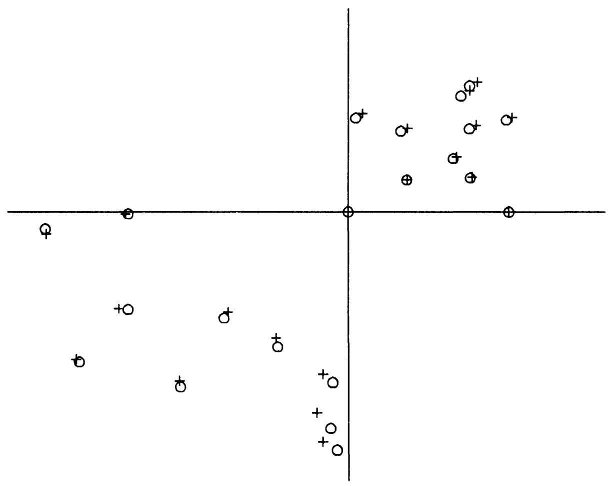

Figure 5.1: True (circles) and computed (crosses) structure and motion with simulated data. Camera positions are in the lower-left quadrant, feature points in the upper-right one.

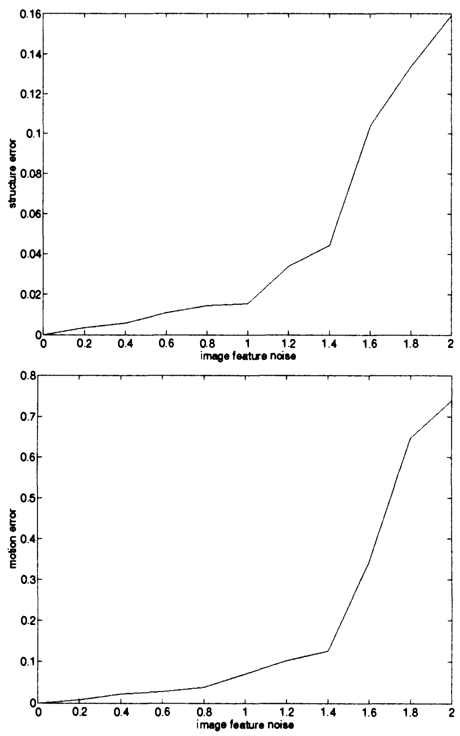

Figure 5.2: Errors in the computed structure (top) and motion (bottom) for increasing levels of image feature noise, measured in pixels for a 512 by 512 image. See text for the units of the vertical axes.

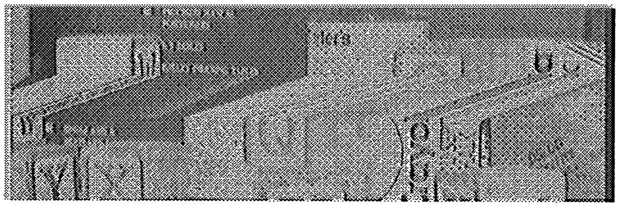

Figure 5.3: An epipolar slice (center) sandwiched between the top of the first and bottom of the last frame of a 50-frame image sequence.

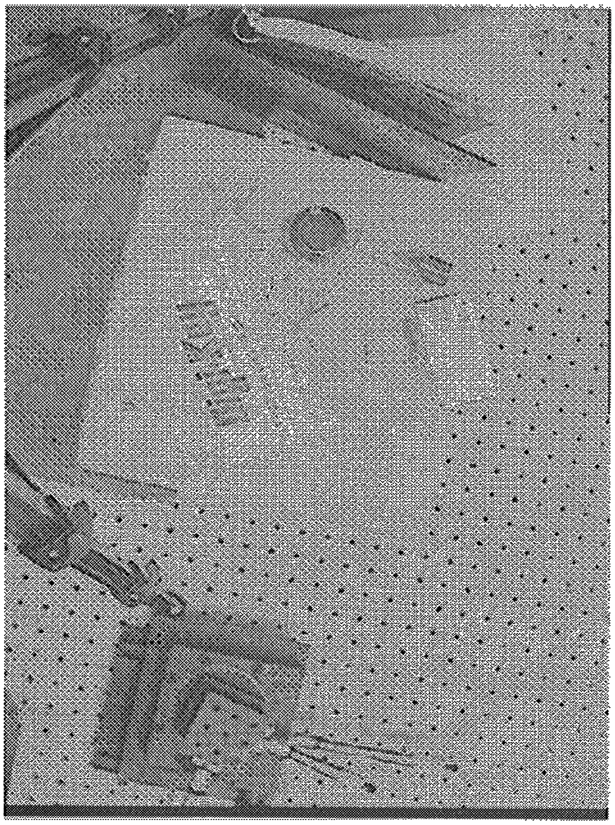

Figure 5.4: The imaging setup viewed from above (the camera has been removed from the motion stage at the bottom).

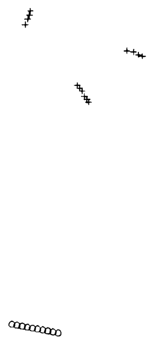

Figure 5.5: The actual positions of the camera (circles) and world features (crosses) as measured in figure 5.4.

Figure 5.6: The positions of the camera (circles) and world features (crosses) as computed by the method described in this report.

Chapter 6

Conclusion

This report presented a radically new conceptual framework, as well as a computational procedure, for the recovery of shape and motion from a sequence of images taken under perspective. While more and better experiments are obviously necessary, a good case can be made for this new way of thinking about an old and important problem.

In fact, the picture and trail loci are useful abstractions *per se*, and the results about their tangent subspaces (or planes in the two-dimensional case) are one of their primary advantages, since they establish an unsuspectedly clean and clear relation between perspective and orthography. Furthermore, the new, rotation-independent model of the imaging situation, which made this relation apparent, removes the slack that was caused by the poor distinguishability of rotation and translation in previous formulations. Finally, the reduction of the nonlinear part of the shape and motion reconstruction to the small space of affine scene deformations gives a handle on the intrinsic nonconvexity of this vision task.

Future work on both camera calibration and the extension of the computation to three dimensions will hopefully imprint the seal of practical usefulness on this new framework.

Bibliography

[BBM87] R. C. Bolles, H. H. Baker, and D. H. Marimont. Epipolar-plane image analysis: An approach to determining structure from motion. *International Journal of Computer Vision*, 1(1):7–55, 1987.

[LH81] H. C. Longuet-Higgins. A computer algorithm for reconstructing a scene from two projections. *Nature*, 293:133–135, September 1981.

[May93] S. Maybank. *Theory of Reconstruction from Image Motion.* Springer-Verlag, Berlin Heidelberg, 1993.

[SA89] M. E. Spetsakis and J. (Yiannis) Aloimonos. Optimal motion estimation. In *Proceedings of the IEEE Workshop on Visual Motion*, pages 229–237, Irvine, California, March 1989.

[TH84] R. Y. Tsai and T. S. Huang. Uniqueness and estimation of three-dimensional motion parameters of rigid objects with curved surfaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-6(1):13–27, January 1984.

[TK91] C. Tomasi and T. Kanade. Shape and motion from image streams: a factorization method - 3. detection and tracking of point features. Technical Report CMU-CS-91-132, Carnegie Mellon University, Pittsburgh, PA, April 1991.

[TK92] C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: a factorization method. *International Journal on Computer Vision*, 9(2):137–154, 1992.

[TS93] C. Tomasi and J. Shi. Direction of heading from image deformations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR93)*, pages 422–427, New York, NY, June 1993.

[WT93] D. Weinshall and C. Tomasi. Linear and incremental acquisition of invariant shape models from image sequences. In *Proceedings of the Fourth International Conference on Computer Vision (ICCV93)*, pages 675–682, Berlin, Germany, May 1993.

Appendix A

The Projection Equation

Figure A.1 shows the landmark point S_0, used as the origin of the coordinate reference system, and a second point S_1 with coordinates $(x_1, z_1) = (0, 1)$. This second point establishes both the orientation and the metric of the coordinate system. Two more points appear in the figure: a camera center C, which stands for any of the F camera positions $C_1, \ldots, C_F$, and an object point S, which stands for any of the P object points $S_1, \ldots, S_P$ other than S_0. The angle α between the projection rays of S_0 and S can be determined from image measurements and is independent of the camera rotation.

Let D be the vector difference between the point position S and the camera position C. Then, the tangent t of the image measurement α is given by minus the ratio of the projections of D along the direction of C and along the direction orthogonal to C. If $C = (c_x, c_z)$, the vector counterclockwise orthogonal to C and with the same magnitude as C is $C^\perp = (-c_z, c_x)$, so that

$$t = \tan\alpha = -\frac{(C^\perp)^T D}{C^T D} = \frac{(C^\perp)^T S}{|C|^2 - C^T S} .$$

If we let

$$K = (u, w) = C/|C|^2$$

be the vector obtained by reflecting C across the unit circle, and $K^\perp = (-w, u)$ be its orthogonal vector, we can also write

$$t = \frac{(K^\perp)^T S}{1 - K^T S} .$$

In scalar form,

$$t = \frac{uz - wx}{1 - ux - wz}$$

(equation (2.1) in the main text).

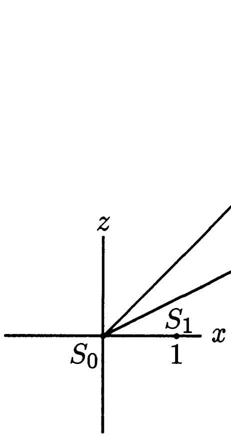

Figure A.1: The symbols used in the projection equation.

Appendix B

Proof of the Orthographic Plane Theorems

We first prove the orthographic plane theorem for the picture surface. The dual theorem, for the trail surface, could be obtained simply by duality (see section 3). However, because its meaning is less intuitive than the result for the picture surface, a few remarks are added below.

The equation of the plane tangent to the picture surface at the origin is the numerator of equation (2.3):

$$t_p = uz_p - wx_p \ .$$

But this is also the limit of equation (2.3) when the norm of $K = (u, w)$ tends to zero. From equation (2.2), when K tends to zero, the norm of the camera position vector C tends to infinity. Thus, the images taken from very distant cameras are very close to the origin of the picture surface. We can now think of an infinitesimally small circle on the picture surface and around the origin: when the radius of this circle shrinks to zero, the projection rays become parallel and orthographic projection is approached, except that the image coordinates become smaller and smaller. Every point on the tangent plane differs from some point on that circle only by a scale factor. This completes the proof for the picture surface. The reasoning for the trail surface is similar: when the norm of the world point position vector $S = (x, y)$ tends to zero in equation (3.3), we again obtain the orthographic projection equation

$$t_f = u_f z - w_f x \ ,$$

and the same reasoning applies. This is a little less intuitive than it is for the picture surface, since we usually think of orthography as the case when the camera positions go to infinity. However, it is equivalent to instead keep the camera positions fixed and shrink the world points towards the origin: the two situations differ only by a scaling factor.

Appendix C

Determining the Picture Surface Parameters

If we reintroduce the frame subscript f into equation (4.8) and suppose that $F \geq 5$ frames have been collected, then $3F$ measurements t_{fp}, t_{fq}, t_{fr} are available for points p, q, r. From these, the following $F \times 5$ matrix M can be formed whose entries m_{fj} are defined by

$$
\begin{aligned}
m_{f1} &= t_{fp}(t_{fq} - t_{fr}) \\
m_{f2} &= t_{fr}(t_{fq} - t_{fp}) \\
m_{f3} &= t_{fp}(1 + t_{fq}t_{fr}) \\
m_{f4} &= t_{fq}(1 + t_{fp}t_{fr}) \\
m_{f5} &= t_{fr}(1 + t_{fp}t_{fq})
\end{aligned}
$$

and the vector $\mathbf{a} = (a_2, a_4, a_5, a_6, a_7)^T$ is then the best nonzero solution to the linear homogeneous system

$$M\mathbf{a} = 0 \; , \tag{C.1}$$

while a_1 and a_3 are determined by equations (4.6) and (4.7). The solution to system (C.1) can be found by computing the singular value decomposition of M,

$$M = U\Sigma V^T$$

and letting $\mathbf{a}$ be the fifth column of V, that is, the eigenvector corresponding to the smallest right singular value.