IEEE Conference on Computer
Vision and Pattern Recognition
(CVPR94) Seattle, June 1994

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

Carlo Tomasi
Computer Science Department
Stanford University
Stanford, CA 94305

Abstract

This paper 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. A shape and motion reconstruction method is developed 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.*

1 Introduction

An important problem in computer vision is to compute 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, this becomes a purely geometric problem. It is, however, a nonconvex and potentially poorly conditioned one. Conditioning must be addressed by formulating the problem in terms of well-observable parameters only, using redundant data, and paying close attention to the numerical aspects of the computation. Nonconvexity must be addressed by a solution method that does not get caught in local minima.

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

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

No rotation in the model. The proposed imaging model 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 [6]. The sensitivity problems of the standard approaches are thereby avoided.

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 *picture locus*, turns out to be a three-dimensional variety in a space with roughly as many dimensions as there are visible features. Every image is a point 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 world. 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.

Incidentally, the first stage of the computation yields shape and motion up to two separate affine transformations. In many applications [10] this is sufficient, and the second, more expensive stage that enforces Euclidean metric can be omitted.

In this paper, 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 been addressed in detail yet. Second, all the concepts introduced are more easily visualized in 2D, where the picture locus becomes a picture *surface*.

The next two sections present the main abstractions of the framework. Section 4 then outlines the reconstruction method. Experiments are discussed in section 5. Simulations shows that the method works well even with substantial image noise. Then, an experiment with real images gives mixed results, supporting the conjecture that camera calibration is critical for good results.

2 The Picture Locus

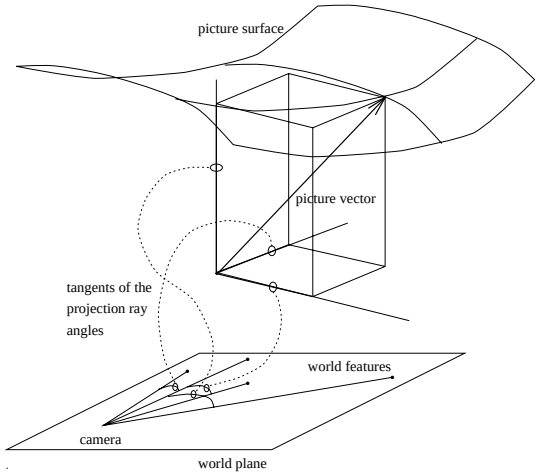

Figure 1: The components of a point on the picture surface (a picture vector) are the tangents of the projection ray angles.

The plane at the bottom of figure 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 between projection rays. In this two-dimensional case, with $P+1$ feature points one feature serves as a landmark and there are P tangents per frame (in the figure, $P = 3$ for visualization purposes). The tangent t of each angle is given by (see [7])

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

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

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

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 $t_1, \ldots, t_P$:

$$t_p = \frac{uz_p - wx_p}{1 - ux_p - wz_p} \tag{3}$$

that can be collected into one vector $\mathbf{t} = (t_1, \ldots, t_P)$, 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 (3). This surface is called the *picture surface*, and 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 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 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 3. This grid is in one-to-one correspondence with the grid on the picture surface of figure 2. Surfaces for more features cannot be visualized directly, but are still two-dimensional objects, because they are traced by two parameters.

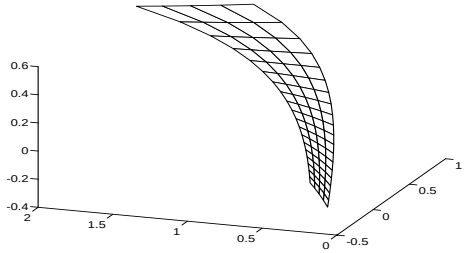

Figure 2: The picture surface for the four features in figure 3. The patch displayed here corresponds to the camera positions shown in figure 3.

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.

Section 4 shows that the picture surface can be determined by linear data fitting from the available image measurements. Unfortunately, the relation between the surface parameters resulting from fitting

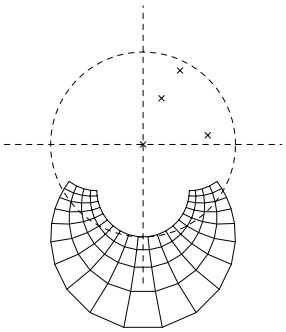

Figure 3: When the reflected camera coordinates K defined in equation (2) vary in the rectangle with vertices $(-1,-1)$ and $(-1,-0.5)$, the camera moves on the grid shown. The cross at the origin is the landmark feature.

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, an important result about the picture surface is now introduced and proven in [7].

> **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 [8]. 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.

3 The Trail Locus and Duality

The picture locus is a surface in the 2D case and a 3D variety in the 3D case. It 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 positions along the camera's path and collecting the image positions of a single feature in the world, measured with respect to the landmark point. The image measurements from the given camera positions represent the trail that that feature left in the images as the camera moved. When the world feature is displaced, the trail vector moves on the trail locus.

If the projection equation (1) is examined, an important relation of duality can be established between the picture and trail surfaces. In fact, equation (1) does not change with the replacements

$$u \leftrightarrow x \quad \text{and} \quad w \leftrightarrow -z \ .$$

Because of this symmetry, all that is said about the picture locus also hold for the trail locus. In particular, the orthographic-plane theorem holds for the trail surface as well, and the method used to determine affine shape, described in the next section, can also be used for affine motion.

4 The Reconstruction Method

Shape and motion are computed in four steps:

1. determine the picture surface by linear fitting;
2. pick two points on the orthographic plane of the picture surface to determine affine shape;
3. determine affine motion with the same technique;
4. replace affine shape and motion into equation (1) to determine their Euclidean counterparts.

Affine shape and motion are computed with linear operations, while the last step is nonlinear.

4.1 The Picture Surface

Because we know the analytic form of the picture and trail surfaces (equations (3) and a similar one for trails), determining their parameters from a set of image measurements is a data fitting problem. The problem becomes linear if we eliminate motion from the picture surface equation (3) and shape from the trail surface equation. For the picture surface, this yields an equation of the third degree in t_p, t_q, t_r (see [7] for the derivation):

$$\begin{aligned} & a_1 t_p(t_q - t_r) + a_2 t_r(t_q - t_p) + a_3 t_p(1 + t_q t_r) \\ + \quad & a_4 t_q(1 + t_p t_r) + a_5 t_r(1 + t_p t_q) = 0 \ . \end{aligned} \tag{4}$$

where the subscripts p, q, r were dropped for simplicity from the coefficients a_i. These coefficients depend only on shape, since the motion parameters u, w have been eliminated. Determining the a_i from a set of measurements over several frames is an easy linear minimization problem.

4.2 Affine Shape and Motion

The orthographic picture plane defined in section 2 is the tangent plane at the origin for the picture surface of equation (4), that is, the plane

$$a_3 t_p + a_4 t_q + a_5 t_r = 0 \ . \tag{5}$$

Any two points on this plane, not colinear with the origin, represent shape up to an affine transformation [8]. 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 (5). For instance, let[1]

$$\begin{aligned} (\mathbf{t}^{(1)})^T &= (1, 0, -a_3/a_5) \\ (\mathbf{t}^{(2)})^T &= (0, 1, -a_4/a_5) \ . \end{aligned}$$

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 is given by the coordinates of the fourth point, that is, by

$$\hat{x}_r = -a_3/a_5 \quad \text{and} \quad \hat{z}_r = -a_4/a_5 \ .$$

With more than four points, we repeat this procedure 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 the matrix of Euclidean coordinates is

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

Thanks to duality (section 3), the affine camera motion $\hat{K}$ can be found by the same procedure.

[1] It is easy to change this choice if $a_5 = 0$.

4.3 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 reference 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

$$S = A\hat{S} \tag{6}$$

$$K = \hat{K}B^T \ . \tag{7}$$

The origin of the coordinate system is the landmark point $(x_0, z_0) = (0, 0)$. We can fix scale and an overall rotation of the reference system by requiring 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 (6) and (7) into the original measurement equation (1). Ignoring point and camera subscripts, equation (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 [6], we show a method for solving this type of equation, although applied to a different problem.

5 Experiments

Figure 4 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 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.

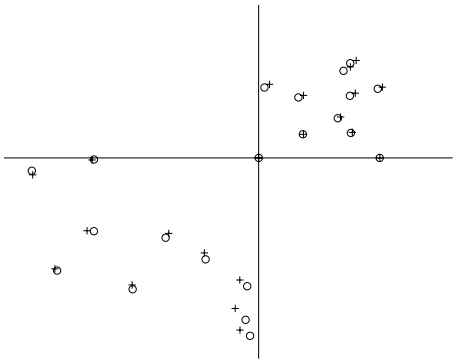

Figure 4: True (circles) and computed (crosses) structure and motion with simulated data. Camera positions are in the lower-left quadrant, feature points in the upper-right one.

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. In feature tracking, the position of features can usually be determined with an accuracy of 0.1 or so pixels [8] for typical 512 by 512 images. From the plots of figure 5 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 6 shows an epipolar slice (like the ones in [1]) from a sequence of images taken with a Panasonic camera mounted on a micrometric translation and rotation stage. The full first frame of this sequence appears as figure 10 in [4] in these proceedings.

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. No camera calibration was performed, and the computation used the nominal focal length of 16 mm, 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 7 compares the actual positions of the features (crosses) and of the camera (circles), measured from a top-view picture of the setup, with the coordinates computed by the algorithm. The camera motion is fairly accurately recovered, the overall distance between the camera and the scene is essentially correct, and each of

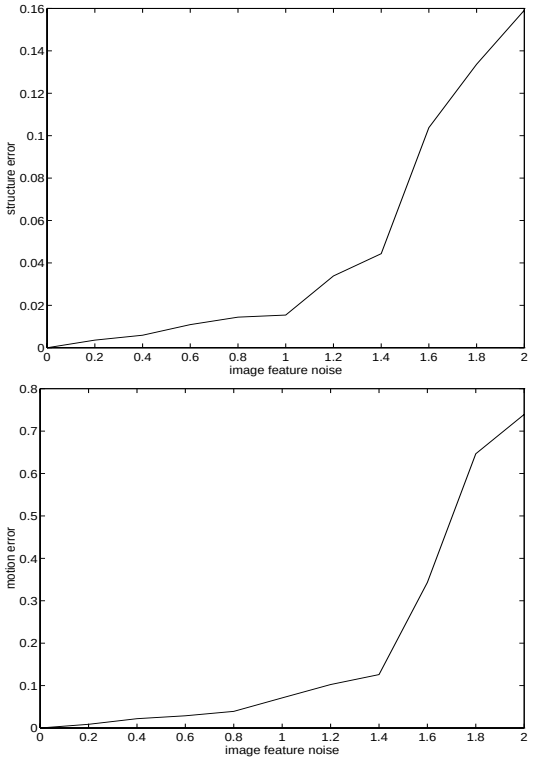

Figure 5: Errors in the computed structure (top) and motion (bottom) for increasing levels of image feature noise, measured in pixels for a 512 by 512 image. See text for the units of the vertical axes.

the three feature groups is approximately of the right shape and size. However, the computed positions of the three groups of features are considerably distorted with respect to the ideal positions. The contrast between these results, with features tracked with about 0.1 pixels accuracy, and the simulations described in figures 4 and 5, 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 6: An epipolar slice (center) sandwiched between the top of the first and bottom of the last frame of a 50-frame image sequence.

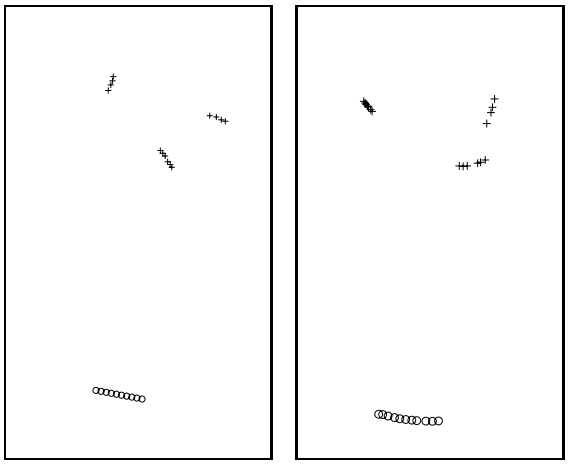

Figure 7: Actual (left) and computed (right) positions of the camera (circles) and world features (crosses).

6 Conclusion

This paper 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. 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 [2], independently reinvented by Tsai and Huang [9] and surveyed in [3]; and on the other hand it is a successor of the factorization method described in [8]. 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 *local* multiframe and multifeature methods such as [5], our method is global, in that it does not require an initial estimate of either structure or motion.

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 nonconvex 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.

References

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

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

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

[4] J. Shi and C. Tomasi. Good features to track. *CVPR*, 1994.

[5] M. E. Spetsakis and J. Aloimonos. Optimal motion estimation. *IEEE Workshop on Visual Motion*, 229–237, 1989.

[6] C. Tomasi and J. Shi. Direction of heading from image deformations. *CVPR*, 422–427, 1993.

[7] C. Tomasi. *Pictures and trails: a new framework for the computation of shape and motion from perspective image sequences.* TR 93-1400, Cornell U., 1993.

[8] C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: a factorization method. *IJCV*, 9(2):137–154, 1992.

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

[10] D. Weinshall and C. Tomasi. Linear and incremental acquisition of invariant shape models from image sequences. In *ICCV93*, 675–682, 1993.