Preprints of the Fourth International
Symposium on Experimental Robotics, ISER'95
Stanford, California, June 30–July 2, 1995

Experiments With a Real-Time Structure-From-Motion System

Carlo Tomasi John Zhang David Redkey

Computer Science Department
Stanford University, Stanford, CA 94305
tomasi,zhang,dredkey@flamingo.stanford.edu

Abstract

We present a real-time system for the reconstruction of shape and motion from an arbitrarily long sequence of single-scanline images. With this system, experiments on reconstruction can be run effortlessly and make it possible to explore the delicate sensitivity aspects of the problem in an empirical fashion. We identify three singular values of a certain matrix of image measurements as the key elements for a sensitivity analysis. Our experimental results suggest that reconstruction is indeed possible in practice with sufficient accuracy at least for navigation and as a guidance to manipulation.

1. Introduction

A vision system that can provide dynamic scene shape and camera motion information reliably and in real time would be of great usefulness to robotics, for instance, in the control of manipulators and in robot navigation. While the literature on the subject is vast, no system is known that works reliably, under perspective projection, and in real time. In [Tom94], we proposed a new formulation of this problem that faces directly the poor conditioning of the shape reconstruction problem. In this formulation, the computation is expressed in terms of well-observable parameters only, using generously redundant data, and paying close attention to the numerical aspects of the computation.

A setup that allows running experiments with minimal effort and overhead is crucial for understanding a problem like shape and motion reconstruction from image sequences. In fact, the problem itself, and not just this or that implementation, is inherently sensitive to noise. Even a good algorithm will fail if the input data are not very good. Even subpixel amounts of positional uncertainty in the measured image feature coordinates can defeat the most sophisticated algorithm. Camera and lens miscalibration can have even worse effects, since they introduce systematic deviations from the correct measurements. Thus, attention to the implementation details is part of a successful system at least as much as attention to the mathematical and numerical aspects of the computation.

In summary, the following three elements are essential for a good shape and motion reconstruction system. First, the problem must be carefully formulated so that only well observable quantities are made part of the required solution. Second, the image measurements must be good enough, with respect to both random and systematic errors, to bring the input of the reconstruction algorithm to within the "basin of attraction" of a solution. Third, close attention must be paid to the numerical aspects of the computation, so that the particular implementation of the algorithm does not add failure modes of its own. Because of the close interaction of these three aspects, only controlled experiments in the laboratory will tell if each of the pieces of a solution is good enough.

Our current reconstruction system is for flatland, where the camera moves in a plane and images are single scanlines. Although all the concepts hold also in three dimensions, the extension is technically less than straightforward, and has not been addressed yet. A two-dimensional version of shape reconstruction, besides being interesting conceptually as an intermediate step towards a fully three-dimensional system, is useful in its own right. For instance, indoor robots often travel on a smooth and level surface, so the camera scanline that shares a horizontal plane with the camera's optical center satisfies the flatland assumption. One can then have one or more separate vision systems, each of which reconstructs one horizontal slice of the environment.

In this paper, we describe a real-time implementation of this system. This implementation required rein-

This research was supported by the National Science Foundation under contract IRI-9496205.

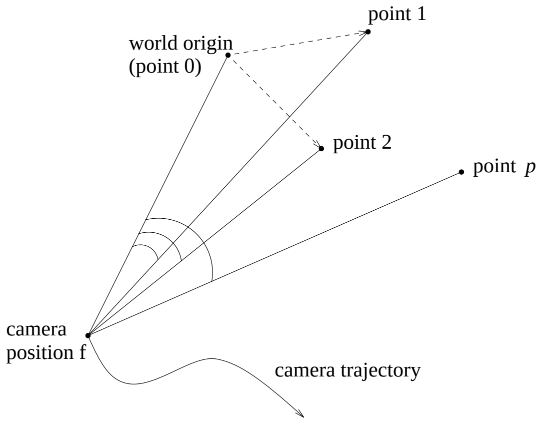


Figure 1. Camera and world points in flatland.

venting the computation completely. Rather than a linear stage followed by a nonlinear one, as done in [Tom94], we now have two linear stages, each implemented as the incremental update of the solution to a linear system. Linearity of the second stage is achieved thanks to the redundancy of the measurements and at the cost of a theoretically less-than-optimal solution. The solution to these two linear stages is then fed to an iterative refinement procedure that proceeds again by repeated solutions of linear systems.

In the following section, we summarize our mathematical formulation of reconstruction problem (section 2) and its solution (section 3). We then discuss the basic numerical issues (section 4) and implementation problems (section 5) that must be addressed for a reliable computation of object shape and camera motion. In section 6, we show our experimental results, and in the conclusion (section 7) we point out the main problems to be overcome for a complete and satisfactory solution to the problem of reconstruction.

2. Formulation

Suppose that the camera and the world points all live in a two-dimensional world (figure 1).

Point 0 serves as the origin of the global reference system. Also, for every frame $f = 1, \ldots, F$ the camera records only the tangents t_{fp} of the angles formed by the projection ray of the feature points $1, \ldots, P$ with that of feature point 0, so with $P+1$ feature points there are P tangents per frame (in the figure, $P = 3$). The tangent t_{fp} can be found by simple geometry to be (see also [Tom94])

$$t_{fp} = \frac{u_f z_p - w_f x_p}{1 - u_f x_p - w_f z_p} \qquad (1)$$

where (x_p, z_p) is the position of feature number p in the world and

$$\mathbf{k}_f = \begin{bmatrix} u_f \\ w_f \end{bmatrix} = \mathbf{m}_f / |\mathbf{m}_f|^2 \qquad (2)$$

is the vector obtained by reflecting the camera coordinates $\mathbf{m}_f$ across the unit circle. This reflection is introduced to make equation (1) bilinear in motion and shape.

With $P+1$ world feature points and F camera positions, the FP measurements $t_{11}, \ldots, t_{FP}$ can be collected into an $F \times P$ matrix T. Each row of T represents one snapshot, and each column is the evolution of one tangent over time. If the reflected camera coordinates u_f, w_f and the shape coordinates x_p, z_p are collected in a $F \times 2$ reflected motion matrix and a $2 \times P$ shape matrix,

$$K = \begin{bmatrix} u_1 & w_1 \\ \vdots & \vdots \\ u_F & w_F \end{bmatrix}, \qquad S = \begin{bmatrix} x_1 & \cdots & x_P \\ z_1 & \cdots & z_P \end{bmatrix}, \qquad (3)$$

then equation (1) can be rewritten in matrix form for the entire sequence as follows:

$$T = \pi(K, S) \qquad (4)$$

where the projection function π operates on the f-th row of K and on the p-th column of S to produce entry t_{fp} of T according to equation (1).

3. Solution Procedure

In [Tom94], this matrix equation was solved for shape S and reflected motion K through a batch procedure, that is, after the entire matrix T of image measurements was acquired. Here instead we reformulate the solution as a series of steps, each of which amounts to solving a linear system or taking a ratio of scalars. This new formulation makes an incremental solution possible, in which initial estimates of shape are refined and the new camera coordinates are computed every time a new image becomes available.

We first list the basic parts of the procedure, and we then show how each of them works.

1. Find shape $\hat{S}$ up to an affine transformation for each quadruple of points with subscripts $(0, 1, 2, p)$ where p ranges from 3 to P. Because points $0, 1, 2$ serve as a reference system for the plane, the coordinates of all the points in space are thereby determined in the same (affine) system of reference.

2. Compute Euclidean shape S by determining a 2×2 matrix A such that

$$S = A\hat{S} \; . \tag{5}$$

3. Compute the matrix K of reflected camera positions (see equations (2) and (3)) by solving the matrix projection equation (4) for K.

4. Determine the matrix M of camera positions by reflecting the rows of K back across the unit circle through the inverse of transformation (2):

$$\mathbf{m}_f = \mathbf{k}_f / |\mathbf{k}_f|^2 \; . \tag{6}$$

We now show that all these steps involve solving a linear system or computing ratios of scalars.

1. If the scalar projection equation (1) is repeated three times for points $1, 2, p$, then the first two equations can be algebraically solved for the reflected camera position coordinates u_f, w_f. These can then be replaced into the third equation to yield the following homogeneous linear equation (see [Tom94] for details)

$$a_1^{(p)} t_1(t_q - t_r) + a_2^{(p)} t_r(t_q - t_1) + a_3^{(p)} t_1(1 + t_q t_r) + a_4^{(p)} t_q(1 + t_1 t_r) + a_5^{(p)} t_r(1 + t_1 t_q) = 0 \; .$$

where

$$\begin{aligned} a_1^{(p)} &= -x_p(x_1 - x_2) - z_p(z_1 - z_2) \\ a_2^{(p)} &= -x_1(x_2 - x_p) - z_1(z_2 - z_p) \\ a_3^{(p)} &= -x_2 z_p + z_2 x_p \\ a_4^{(p)} &= x_1 z_p - z_1 x_p \\ a_5^{(p)} &= -x_1 z_2 + z_1 x_2 \; . \end{aligned} \tag{7}$$

Writing this equation once for every frame $f = 1, \ldots, F$ yields a $F\times 5$ homogeneous linear system

$$T\mathbf{a}^{(p)} = 0 \tag{8}$$

where the 5-dimensional vector $\mathbf{a}^{(p)}$ collects the unknown coefficients in (7). Notice that because the system is homogeneous the solution can only be determined up to a multiplicative constant, that is, only $\lambda\mathbf{a}^{(p)}$ can be determined. An $F\times 5$ system of the form (8) must be solved for every point $p = 3, \ldots, P$.

The coordinates of point p in the reference system formed by points $0, 1, 2$ are then given by (see [Tom94])

$$\hat{x}^p = -a_5^{(p)} / a_7^{(p)} \quad \text{and} \quad \hat{z}^p = -a_6^{(p)} / a_7^{(p)} \; .$$

These coordinates can be collected into a $2\times P$ affine shape matrix

$$\hat{S} = \begin{bmatrix} 1 & 0 & \hat{x}_3 & \cdots & \hat{x}_P \\ 0 & 1 & \hat{z}_3 & \cdots & \hat{z}_P \end{bmatrix} . \tag{9}$$

2. Since the coordinates in $\hat{S}$ are expressed in the same affine reference shape, they can differ from the Euclidean coordinates of the feature points only by a 2×2 transformation (see equation (5)). In order to compute this transformation A, we notice that the final solution is determined up to an overall scale factor and a rotation. Therefore we can assume without loss of generality that

$$A = \begin{bmatrix} 1 & a \\ 0 & b \end{bmatrix} \tag{10}$$

which amounts to keeping the origin at point 0 and the unit point of the x axis at point 1. Then, equations (5), (9), and the last four equations (7) yield

$$\lambda a_2^{(p)} = \hat{x}_p + a(\hat{z}_p - 1) \; , \qquad \lambda a_3^{(p)} = -b\hat{x}_p$$
$$\lambda a_4^{(p)} = b\hat{z}_p \; , \qquad \lambda a_5^{(p)} = -b \; .$$

Using the first of these equations to eliminate λ yields three equations in a and b:

$$\begin{bmatrix} a_3^{(p)}(1 - \hat{z}_p) & a_2^{(p)}\hat{x}_p \\ a_4^{(p)}(1 - \hat{z}_p) & a_2^{(p)}\hat{z}_p \\ a_5^{(p)}(1 - \hat{z}_p) & -a_2^{(p)} \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} a_3^{(p)}\hat{x}_p \\ a_4^{(p)}\hat{x}_p \\ a_5^{(p)}\hat{x}_p \end{bmatrix} .$$

A triple of equations like these can be written for every $p = 3, \ldots, P$, so that the two unknown entries a, b of the transformation matrix A are found as the solution of an overdetermined system of $3(P-2)$ equations,

$$C \begin{bmatrix} a \\ b \end{bmatrix} = \mathbf{c} \; . \tag{11}$$

Equation (5) then yields the Euclidean shape matrix S.

3. Once shape is known, the bilinear matrix projection equation (4) is linear in the reflected camera coordinates K. With P points (ignoring the origin), each of the F rows of K,

$$\mathbf{k}_f = (u_f, w_f)^T \; ,$$

can be found as the solution to the system

$$\begin{bmatrix} z_1 + t_{f1}x_1 & x_1 + t_{f1}z_1 \\ \vdots & \vdots \\ z_P + t_{fP}x_P & x_P + t_{fP}z_P \end{bmatrix} \mathbf{k}_f^T = \begin{bmatrix} t_{f1} \\ \vdots \\ t_{fP} \end{bmatrix} \tag{12}$$

of P equations in 2 unknowns.

4. From K, the motion matrix M with all the camera coordinates can be found by the reflection (6).

5. Shape and motion computed from the previous steps are suboptimal because shape is computed one quadruple at a time, rather than from all the points at once, and because system (11) uses only four out of the five equations (7) to avoid nonlinear equations. It is then advisable to use the computed shape and motion as starting values for an iterative procedure that refines shape and motion by repeatedly solving the original projection equation (4). Since this equation is bilinear in motion and shape, the iterative procedure can work by interleaving solutions to linear systems.

4. Numerical Aspects

To summarize, we need to solve the following linear systems:

1. $P-2$ versions of the $F \times 5$ system (8), one version for each quadruple of points with subscripts $0, 1, 2, p$ for $p = 3, \ldots, P$. This is a homogeneous system. The number of its rows grows with time, and the solution improves in quality. This system must be solved in an incremental fashion to avoid unbounded storage and computation time.

2. The $3(P-2) \times 2$ system (11) for the computation of the two unknown entries a, b of the matrix A (equation (10)). The size of this system is fixed, but its solution must be recomputed afresh with every frame because its entries depend on the improving affine shape estimates $\hat{x}_p, \hat{z}_p$.

3. The $P \times 2$ system (12) that computes reflected camera coordinates u_f, w_f at every frame. Also this system is fixed size and needs to be recomputed for every frame.

4. P systems of size $F \times 2$ for the refinement of shape and F systems of size $P \times 2$ for the refinement of motion through equation (4). Both the size of the shape systems and the number of the motion systems grow with the number of frames. To keep storage and computation time fixed, we only remember a fixed-size set of past frames for this computation. The choice of these frames is discussed in section 5, and the idea is to store an approximately uniform sample of the past sequence of frames. With this device, the two systems used in the refinement stage are fixed in size.

Notice that all linear systems can be easily updated if features disappear because of occlusions or other reasons, with the only exception of the three reference features. Similarly, update is straightforward if features are added at different points in time. In the current version of the system, disappearance of any of the reference features causes the reconstruction system to stop. We are working on overcoming this important limitation.

Because of the large number of linear systems to solve, and because of the sensitivity of the solution to image noise, it is crucial to employ efficient and numerically stable solution methods. We have essentially two different types of linear systems:

- A growing, homogeneous, $F \times 5$ system for the computation of affine shape for every quadruple of points.
- Several fixed systems of size $m \times 2$ where m depends on the particular system but is usually large.

Both systems are first converted to square (5×5 or 2×2), upper-triangular systems by incorporating one row at a time into the R matrix of a QR decomposition. These square, triangular systems are then solved by backsubstitution. For the homogeneous system, one of the diagonal elements of R is zero in the absence of noise. With noise, we assume that the element that must be zeroed is the smallest diagonal element of R. The corresponding entry of the solution vector is set to 1, and backsubstitution can proceed as usual.

5. Implementation

In our experiments, we used a standard Pulnix CCD camera with a 2/3" sensor (corresponding to an actual sensor size of 6.6×8.8 mm) and a high quality Schneider Cinegon 1.8/4.8mm lens corrected for chromatic aberration both in the visible and the infrared parts of the spectrum. Because of the wide field of view (105 degrees along the diagonal), distortion in unavoidable even in a lens as good as this, and calibration is crucial. To this end, we adapted the procedure described in [Fle94], and we fitted a fifth-degree polynomial to the coordinates measured on an image of a calibration target made of concentric circles.

Frames are acquired by a Digital J300 frame grabber that interfaces directly with the TurboChannel bus of a Digital Alpha 600 workstation, where both the tracker and the reconstruction algorithm are implemented in C++. To select and track features we have implemented a one-dimensional version of the system described in [ST94]. With a feature window width of 7 pixels and a filter kernel width of 5 pixels for image smoothing the system can currently track one feature per frame in about one millisecond. Feature selection at this point requires user interaction and does not run in real time. The camera is required to remain still until feature selection is completed. The user selects features by pointing and clicking in a feature selection window (upper-left window in figure 2). The tracker is then activated and follows features at subpixel resolution as described in [ST94].

Although the tracker updates feature coordinates at every frame, shape computation waits until the changes in these coordinates are large enough to warrant incorporating a new set of input data. To check for this event, we first monitor the RMS displacement of all the features in the scanline. Once this measure has exceeded one pixel, a new row of the matrix T of angle tangents is computed, and its RMS variation with respect to the previous row used for reconstruction is checked against another threshold (0.005 radians in our implementation). Only when this threshold is exceeded is the most recent frame passed to the reconstruction algorithm and are the linear system solutions updated. Consequently, the more expensive part of the computation is performed only rather occasionally for a slowly moving camera. Even more importantly, data are used only when they add sufficiently new information to the computation. In summary, all the frames produced by the camera are tracked, but only a few of them are used for reconstruction. These select frames are called *significant frames*.

For the iterative refinement stage described in section 3 a set of 2^k *key frames* is stored, where k is a fixed number (3 in our implementation). These key frames should be spread as uniformly as possible over the past tracking history. We achieve this with a caching algorithm that after tracking about 2^K significant frames remembers one significant frame every 2^{K-k}.

6. Experiments

In our experiments, the camera is mounted on a translation and rotation stage that can be moved by hand. This setup ensures motion in a plane. Sliding objects by hand on a table is an alternative mode of operation. Figure 2 shows the program interface. The upper left window shows the scene with the selected features superimposed. All features are taken from the central scanline. The lower-left window stacks the scanlines used for reconstruction (significant frames) on top of each other, and the lower-right window displays shape (squares) and camera position estimates (dots). The

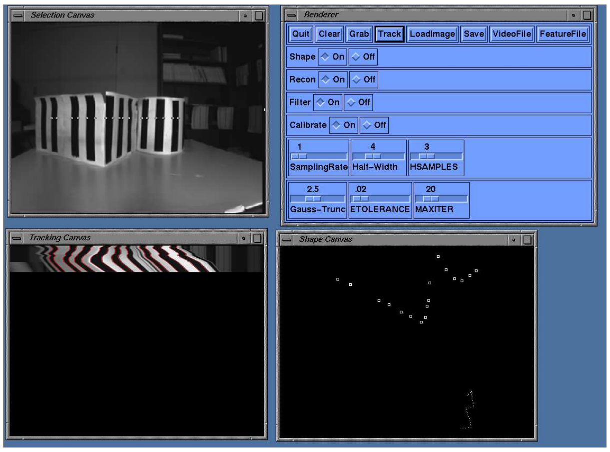


Figure 2. The user interface.

menu in the upper right window allows setting various system parameters.

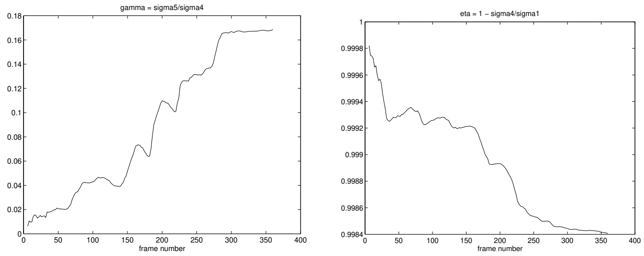
Sensitivity to noise is the dominant issue in the reconstruction problem. The problem itself is inherently sensitive, and no algorithm or numerical implementation will make this go away. As a consequence, reconstruction can only be applied when the field of view of the camera is sufficiently wide [KvD87], [HW88], motion is sufficiently extended around the object [TK93], and image measurements are both sufficiently accurate (low bias) and precise (low standard deviation).

From a numerical standpoint, since the homogeneous $F \times 5$ system (8) is expected to have exactly one affine shape solution with perfect data and nontrivial motion, its matrix T should be of rank 4. Consequently, the R term in the QR decomposition of T should have exactly four nonzero diagonal terms. A better look at the structure of T can be gleaned from its singular values $\sigma_1 \geq \ldots \geq \sigma_5$: specifically, σ_4 should be nonzero and σ_5 should be zero. In reality, noise increases σ_5, and closeness to either degenerate shape, motion, or imaging situation decreases σ_4. For instance, if the camera does not move, all the rows of T are ideally equal, and T is rank 1: $\sigma_2 = \ldots = \sigma_5 = 0$. Also, when the camera's field of view approaches zero width (orthographic projection) the rank of T tends to 3: $\sigma_4 = \sigma_5 = 0$ [Tom94]. Yet reconstruction under perspective assumes that there is a stable, substantial gap between the last two singular values, leading to a low *noise factor*

$$\gamma = \frac{\sigma_5}{\sigma_4} \ll 1$$

and a good conditioning of the rank-4 part of the system, leading to a low *sensitivity factor*

$$\eta = 1 - \frac{\sigma_4}{\sigma_1} \ll 1$$



(a) (b)

Figure 3. (a) Noise factor γ, and (b) sensitivity η as a function of "significant frame" number for a forward moving camera.

(the condition number is usually defined as σ_1/σ_4; we prefer η because it remains between 0 and 1; $\eta = 0$ is ideal). These two conditions mean little noise and no degeneracy. They can be optimized by proper setup (η) and good image measurements (γ). However, the numerical computation should be prepared to deal with relatively large values of γ and η if the reconstruction system is to be applicable to a wide range of interesting situations. For instance, figure 3 shows the two parameters γ and η for a typical point quadruple seen by a forward-moving camera. The object is initially 25 cm away, and the four points span about 15 degrees of the field of view. The focal length of the lens is 4.8 mm.

These plots show the difficulty of the problem. While noise is relatively under control ($\gamma < 0.2$), the sensitivity factor η is dangerously close to 1 throughout. Even more disturbingly, the sensitivity η declines very slowly when more frames are added (the camera moved forward by about 8 cm during the 360 frames), and the noise factor γ increases. The increase of the noise factor γ may be counterintuitive at first, but is due to the fact that new frames often add more noise than novel shape information.

7. Future Work

Having a system that reconstructs shape and motion from image sequences in real time makes it possible for us to run many experiments with little effort, and at the same time forces us to consider the real difficulties that a satisfactory solution to the reconstruction problem must overcome.

Our experimental results suggest that reconstruction is indeed possible with sufficient accuracy at least for navigation and as a guidance to manipulation. Sensitivity to image noise is by far the dominant problem in reconstruction. In this paper, we have started exploring the issues related to sensitivity in a neighborhood of the solution. These issues are best understood by looking at the singular value of the matrix T that collects the tangents of the angles between pairs of projection rays. For good reconstruction, the field of view must be wide enough and the camera must move by a sufficiently large amount; image measurements must be accurate (good calibration) and precise (low noise); the formulation of the problem must be in terms of a minimal number of parameters (camera and feature positions, but no camera rotation); and the algorithm must be numerically sound.

The two limitations of our reconstruction method that require most immediate attention are its dependence on three reference features throughout the sequence and the fact that affine shape is computed one quadruple of points at a time. The final iterative refinement stage addresses the latter problem, because it combines all the measurements into one minimization procedure, but more elegant and efficient solutions may be possible. Also, the extension to reconstruction to three dimensions is mathematically far from straightforward, and the computation requires more time or resources than in two dimensions. However, the sensitivity of the problem should, if anything, improve, because the ratio of unknowns to measurements is reduced by a factor of 3/4 from approximately 2(P+F)/PF to 3(P+F)/2PF, where P is the number of points and F is the number of frames. In conclusion, the results are auspicious and encourage us to continue our investigation of solutions of the reconstruction problem.

References

[Fle94] M. Fleck. Shape and wide-angle image. Tech. Rep. 04, U. of Iowa, 1994.

[HW88] B. K. P. Horn and E. J. Weldon Jr. Direct methods for recovering motion. *IJCV*, 2:51–76, 1988.

[KvD87] J. J. Koenderink and A, J. van Doorn. Facts on optic flow. *Biol. Cyb.*, 56:247–255, 1987.

[ST94] J. Shi and C. Tomasi. Good features to track. *CVPR*, 593–600, 1994.

[TK93] C. Tomasi and T. Kanade. Shape and motion from image streams – a factorization method. *Proc. of the Nat'l Acad. of Sci. of the USA*, 90:21, 9795–9802, 1993. Also appears in *IJCV*, 9(2), 137–154, 1992.

[Tom94] C. Tomasi. Pictures and trails: a new framework for the computation of shape and motion from perspective image sequences. *CVPR*, 913–918, 1994.