Seventh International Symposium on
Robotics Research, Springer Verlag, 1995.

Is Structure-From-Motion Worth Pursuing?

Carlo Tomasi and John Zhang
Computer Science Department, Stanford University
Stanford, CA 94305
tomasi@cs.stanford.edu, zhang@sccm.stanford.edu

1 Introduction

Structure-from-motion is the problem of recovering the motion of a camera and the shape of objects in front of it from the images that the camera produces. If vision is to compute a representation of the world from images, structure-from-motion is undeniably a central problem. Yet decades of research have shown that this computation is very sensitive to errors in image measurements, both random and systematic. The time has come to assess this sensitivity quantitatively. Is the problem so bad that only extremely controlled experiments with carefully calibrated, high-quality equipment can yield acceptable reconstructions of the world's geometry? Or can we hope to devise methods that extract useful three-dimensional information from the images produced by an off-the-shelf camera in a wide array of imaging scenarios? It may seem surprising that no clear and complete answer to this question is known yet. After all, if structure-from-motion is a hopeless endeavor, hundred of man-years are being wasted. Conversely, if solutions are practically feasible, only a good understanding of what can and cannot be computed reliably is likely to lead to success.

Presumably, there are two main reasons why a systematic sensitivity analysis of structure-from-motion has not been done yet. One is that the function to be analyzed, which takes image measurements as inputs and produces structure and camera motion as outputs, is not explicitly available. Reconstruction methods exist, but they differ in important ways, so selecting a particular method severely limits the generality of any conclusion that is drawn from its analysis. The other reason is that if sensitivity is expressed as a Jacobian of the ouput versus the input of the reconstruction process, this very large matrix of numbers is by itself uninspiring and needs interpretation, a less than straightforward task.

The main contribution of this paper is a framework for the sensitivity analysis of structure-from-motion that addresses both these difficulties. Although not available directly, the reconstruction function is completely specified by the formulation of structure-from-motion as a least squares problem. Implicit differentiation then leads to the desired Jacobian. The interpretation of the latter is made possible by the Singular-Value Decomposition (SVD) which exposes the structure of the matrix in a way that lends itself directly to geometric interpretation.

Of course, a sensitivity analysis is not the whole answer. Algorithms can fail because of outliers in the data or because of convergence to local minima, whether inherent in the function to be minimized or produced by a particular reconstruction method. Consequently, only an actual working reconstruction system will eventually prove feasibility. While sensitivity analysis gives, in a sense, an expected lower bound on the structure and motion errors that one can hope to attain for a given image quality, actual experiments with specific algorithms provide upper bounds. However, experiments with image sequences can be expensive in terms of setup, computation, and analysis. In fact, processing a whole sequence of images provides but one data point in the large space of experimental parameters. To address this difficulty, we have developed a method that reconstructs structure and motion in Flatland, that is, in a two-dimensional slice of the world, using single image scanlines as its input. The input is small enough that the method works in real time on a workstation. Yet the sensitivity issues are not trivialized relative to the three-dimensional case. In fact, a simple equation counting argument shows that the data constrain the solution better in three than in two dimensions. If we understand the sensitivity issues in Flatland, we

[0]This research was supported under NSF Grants IRI-9201751 and IRI-9496205.

have a good handle on their three-dimensional counterpart as well.

The remainder of this paper describes this two-prong approach to understanding the sensitivity of structure-from-motion. In section 2, we describe our real-time system and what we have learned from our experiments. Then, in section 3, we show how structure-from-motion can be analyzed without explicit reference to a specific reconstruction method. Section 4 discusses implications and future work.

2 A Real-Time System for Experimentation

The theory and full implementation details of our real-time strcture-from-motion system in Flatland are explained in [9]. Here we first sketch the main steps of the computation. We then show how we have been using our system, and we summarize some of the lessons we have learned from our experiments in terms of sensitivity.

2.1 Image Measurements

Suppose that the camera and the world points all live in a two-dimensional world. Point 0 serves as the origin of the global reference system. For every frame $f = 1, \ldots, F$ the camera records 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. The tangent t_{fp} can be found by simple geometry to be (see also [7])

$$t_{fp} = \frac{u_f z_p - w_f x_p}{1 - u_f x_p - w_f z_p} \tag{1}$$

where $\mathbf{s}_p = (x_p, z_p)^T$ is the position of feature number p in the world and

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

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.

The FP measurements $t_{11}, \ldots, t_{FP}$ can be collected into an $F \times P$ matrix T. Each row 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, equation (1) can be rewritten in matrix form for the entire sequence as follows:

$$T = \pi(K, S) \tag{3}$$

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). The reconstruction method described in [9] solves the matrix equation (3) for shape S and reflected motion K in a series of steps, each of which either solves a linear system or takes a ratio of scalars. Initial estimates of shape are refined and the new camera coordinates are computed every time a new image becomes available. Here are, in summary the steps of the computation:

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. Points $0, 1, 2$ establish a common affine reference system for all the quadruples.

2. Compute Euclidean shape S by determining a 2×2 matrix A such that
$$S = A\hat{S} \; . \tag{4}$$

3. Compute the matrix K of reflected camera positions from equation (3).

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),
$$\mathbf{m}_f = \mathbf{k}_f / |\mathbf{k}_f|^2 \; . \tag{5}$$

It turns out that the critical step for the understanding of the sensitivity of the computation is the first one. In this step, the scalar projection equation (1) is repeated three times for points $1, 2, p$, so the reflected camera coordinates u_f, w_f can be eliminated to yield the following homogeneous linear equation (see [7] for details)

$$\begin{aligned} & a_1^{(p)} t_{f1}(t_{f2} - t_{fp}) + a_2^{(p)} t_{fp}(t_{f2} - t_{f1}) \\ + \; & a_3^{(p)} t_{f1}(1 + t_{f2} t_{fp}) + a_4^{(p)} t_{f2}(1 + t_{f1} t_{fp}) \\ + \; & a_5^{(p)} t_{fp}(1 + t_{f1} t_{f2}) = 0 \; . \end{aligned}$$

where

$$\begin{aligned} a_1^{(p)} &= -x_p(x_1 - x_2) - z_p(z_1 - z_2) \\ a_2^{(p)} &= -x_1(x_2 - x_p) - z_1(z_2 - z_p) \\ a_3^{(p)} &= -x_2 z_p + z_2 x_p \\ a_4^{(p)} &= x_1 z_p - z_1 x_p \\ a_5^{(p)} &= -x_1 z_2 + z_1 x_2 \; . \end{aligned} \tag{6}$$

Writing this equation once for every frame $f = 1, \ldots, F$ yields an $F \times 5$ homogeneous linear system in $a_1^{(p)}, \ldots, a_5^{(p)}$:

$$H\mathbf{a}^{(p)} = 0 \; . \tag{7}$$

An $F \times 5$ homogeneous system of the form (7) is solved for every point $p = 3, \ldots, P$. In section 2.3 below we show that matrix H is the key to understanding the sensitivity of this particular reconstruction method.

2.2 Experimental Setup

In our experiments, we use a Pulnix CCD camera with a 6.6×8.8 mm sensor and a high quality Schneider Cinegon f1.8/4.8mm lens. Because of the wide field of view (105 degrees along the diagonal), distortion is unavoidable. We calibrate it away by the procedure described in [1]. Frames are acquired by a Digital J300 frame grabber that interfaces directly with the TurboChannel bus of a Digital Alpha 600 workstation. Features are tracked by a one-dimensional version of the system described in [5] at a rate of about one feature per frame per millisecond. Feature selection at this point requires user interaction and the camera is required to remain still until the selection is completed. In our experiments, either the camera or the object is moved by sliding it on a table.

Although the tracker updates image 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 (equation (3)) 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. Consequently, the more expensive part of the computation is performed only rather occasionally for a slowly moving camera. In summary, all the frames produced by the camera are tracked, but only a few of them, called *significant frames*, are used for reconstruction. These significant frames are displayed on the computer screen as the object or the camera are moved. The shape and motion results are also redisplayed whenever they are updated.

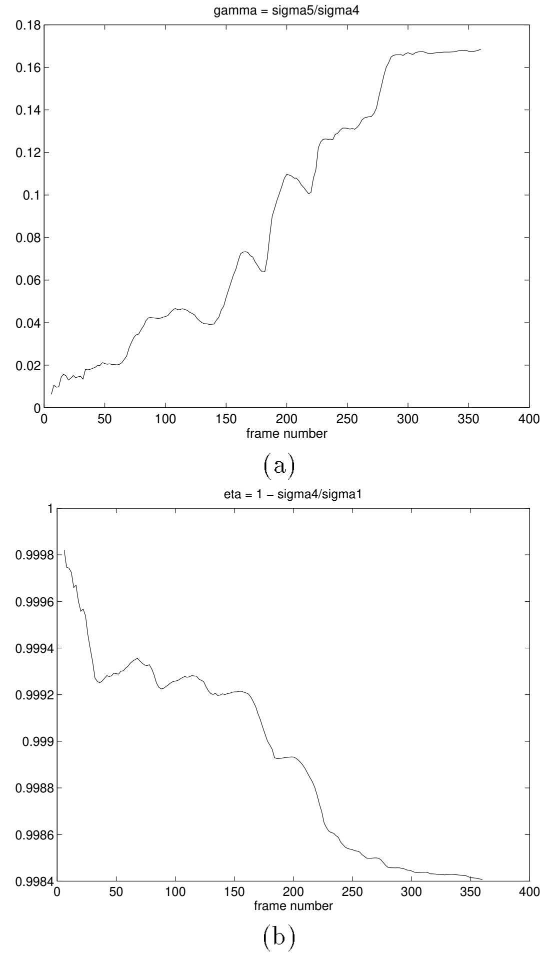


Figure 1: (a) Noise factor γ and (b) sensitivity η as a function of "significant frame" number for a forward moving camera.

2.3 Sensitivity Considerations

Sensitivity to noise is the dominant issue in the reconstruction problem. For meaningful results, the field of view of the camera must be wide enough [4], [3], motion must be sufficiently extended around the object [8], and image measurements must be both sufficiently accurate (low bias) and precise (low standard deviation).

From a numerical standpoint, since the homogeneous $F \times 5$ system (7) is expected to have exactly one affine shape solution with perfect data and nontrivial motion, its matrix H should be of rank 4. Of its singular values $\sigma_1 \geq \ldots \geq \sigma_5$, σ_4 should be nonzero and σ_5 should be zero. In reality, noise increases σ_5, and closeness to degenerate shape or motion decreases σ_4. For instance, if the camera does not move, all the rows of H are ideally equal, and H is rank 1: $\sigma_2 = \ldots = \sigma_5 = 0$.

Also, when the camera's field of view approaches zero width (telephoto lens) the rank of H tends to 3: $\sigma_4 = \sigma_5 = 0$ [7]. Yet reconstruction under perspective requires 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*

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

(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 can be optimized by proper setup (η) and good image measurements (γ). Figures 1 (a) and (b) show 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. 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. Furthermore, 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 γ may be counterintuitive at first, but is due to the fact that new frames often add more noise than new shape information.

In summary, our system for reconstructing 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 of the problem. Our experiments 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, and can be understood by looking at the singular values of the matrix H that appears in the system that solves for affine shape. For good results, 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 algorithms must be numerically sound.

The extension of 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.

3 Sensitivity Analysis of Structure-From-Motion

The successful experiments reported in the previous section and in [9] imply that structure-from-motion is not out of reach. However, the quality of the results is still far from optimal. When objects are moved away from the camera, performance degrades quickly because image parallax is reduced and becomes comparable to image noise and the residual lens distortion. When they are moved too close to the camera, performance degrades because of unmodeled near-field lens distortion effects. In brief, the system is still rather sensitive to errors in the image measurements. In this section, we show how this sensitivity can be understood without making reference to a particular reconstruction system.

The most important point is that *sensitivity is a property of the problem.* Sensitivity does not depend on the solution algorithm, although a poor algorithm may add its own problems, and does not depend on noise, although sensitivity *to noise* is what is being measured. In fact, sensitivity expresses a *ratio* of output noise per amount of input noise, assuming that input noise is small. Consequently, sensitivity analysis techniques should exist that depend neither on a particular algorithm nor on a noise generator. Doing away with these two factors has important advantages. First, sensitivity is a local property of a problem, but an actual algorithm operates in the global space of all solutions. If an actual algorithm is used, the suspicion lingers that some of the poorer results may depend on a failure of that algorithm to converge to the correct minimum. Second, using a noise generator makes it necessary to compute averages of performance, leading to potentially long running times, and to the question of how small noise really should be in the simulations, given that sensitivity is a local property. Furthermore, since in these noise-based studies sensitivity is measured by collecting scatter statistics, the results do depend on the statistics of noise which, in principle, they should not do.

In this section, the main issues involved in sensitivity analysis are identified, and a general methodology is developed. In structure-from-motion, the projection relation from unknowns to measurements is simple, but the reverse reconstruction relation from measurements to unknowns is hard to determine and sensitive to noise. Furthermore, reconstruction problems share with many other problems in vision and physics the property that they depend on many parameters, thereby bringing forth important issues of data visualization and summarization. In fact, the latter are perhaps the most important challenges to address if sensitivity analysis is to be of any use.

In the following, the basic structure of a reconstruction problem is identified. Then, function minimization is introduced as the prototypical method for inverting the projection equations. Minimization is, in a sense, the conceptual "solution method" for any reconstruction problem. After that, sensitivity is defined and related to the parameters of the minimization method, and a general sensitivity analysis method is outlined.

3.1 Reconstruction

In all reconstruction problems, the set of image measurements is written as a function of motion and structure parameters. Image measurements can be point feature coordinates, their velocities, a vector field, sometimes raw image intensities as in [3], and can come from one, two, or more images. Motion, on the other hand, can be represented by translation and rotation, or by the corresponding velocities. Structure can be represented as depth or shape, in viewer-centered or object-centered coordinates. Whatever the particular case, however, the structure of the problem is the same, and can be represented by the following generic notation:

$$u = p(m, s)$$

where u represents image measurements, m and s represent motion and shape, respectively, and p is the projection function. In discrete problems, u, m, s are vectors of real numbers. In continuous problems, u and s are functions of two variables and m is a function of one variable (essentially time).

Because reconstruction is sensitive to image measurement errors, it should be posed as a minimization problem. To this end, the residue

$$r(u, m, s) = u - p(m, s) \tag{8}$$

is defined and either the sum or the integral of $||r||^2$ is minimized over all possible choices of m and s, depending on whether a discrete or a continuous formulation of the problem is being used. Here, we restrict our approach to a discrete, least squares formulation, so the double vertical bars represent the standard Euclidean norm. The minimization may be constrained by some requirement on the solution. Because absolute size cannot be inferred from images alone, camera translation is often normalized in some way.

3.2 Sensitivity

For the time being, it is not necessary to draw a distinction between motion and shape variables. We will therefore introduce the *world* vector

$$\mathbf{w} \stackrel{\Delta}{=} [\mathbf{m}, \mathbf{s}] \tag{9}$$

of motion and shape combined. Let q be the number of entries of $\mathbf{w}$. The projection equations are now represented by

$$\mathbf{u} = \mathbf{p}(\mathbf{w}) \tag{10}$$

and we look for the[1]

$$\begin{aligned} \min_{\mathbf{w}} e(\mathbf{u}, \mathbf{w}) &= \frac{1}{n} \sum_{i=1}^{n} ||\mathbf{r}_i(\mathbf{u}, \mathbf{w})||^2 \\ &= \frac{1}{n} \mathbf{r}^T(\mathbf{u}, \mathbf{w}) \mathbf{r}(\mathbf{u}, \mathbf{w}) . \end{aligned}$$

If there are h measurements per image point (typically $h = 1$ or 2), the residue vector $\mathbf{r}_i$ has h components, which are stacked into the hn-component vector $\mathbf{r}$.

A necessary condition for $\mathbf{w}$ to be a minimum is that the partial derivatives of e with respect to $\mathbf{w}$ vanish:

$$\mathbf{g}(\mathbf{u}, \mathbf{w}) \stackrel{\Delta}{=} \frac{\partial e}{\partial \mathbf{w}} = 0 . \tag{11}$$

For sensitivity analysis, the distinction between a generic stationary point and the global minimum is irrelevant: since we know the solution, we need not search for it. The system of equations $\mathbf{g}(\mathbf{u}, \mathbf{w}) = 0$ is called the system of *normal equations* for the reconstruction problem.

[1] Parentheses denote functional dependency.

Sensitivity analysis addresses the following question: How does the solution $\mathbf{w}$ change as a result of small variations in the image measurements $\mathbf{u}$? In other words, we consider the normal equations as implicitly defining a transformation

$$\mathbf{w} = \mathbf{f}(\mathbf{u}) \tag{12}$$

and seek to characterize the derivatives of $\mathbf{w}$ with respect to $\mathbf{u}$, that is, we look for the Jacobian of $\mathbf{f}$. Notice that equation (12) is, conceptually, the inverse of the projection equation (10). For this reason, we call equation (12) the *reconstruction* equation. The Jacobian of the reconstruction function $\mathbf{f}$ can be found from the partial derivatives of $\mathbf{g}$ through the implicit function theorem [6]. The derivatives of $\mathbf{g}$ can then be written in terms of those of the projection function $\mathbf{p}$ thanks to equation (11). These two steps are carried out in the Appendix. The final result is that the Jacobian $J_{\mathbf{f}} = \frac{\partial \mathbf{f}}{\partial \mathbf{u}^T}$ of $\mathbf{f}$ is the solution of the linear system

$$\left(J_{\mathbf{p}}^T J_{\mathbf{p}}\right) J_{\mathbf{f}} = J_{\mathbf{p}}^T \tag{13}$$

where $J_{\mathbf{p}} = \frac{\partial \mathbf{p}}{\partial \mathbf{w}^T}$ is the Jacobian of the projection function.

The term in parentheses turns out to be the Hessian matrix of the total error function $e(\mathbf{u}, \mathbf{w})$ with respect to $\mathbf{w}$ at the solution point of the reconstruction problem. If this Hessian happens to be invertible, that is, if the Jacobian $J_{\mathbf{p}}$ of the projection function $\mathbf{p}$ has full rank, then equation (13) can be multiplied by $\left(J_{\mathbf{p}}^T J_{\mathbf{p}}\right)^{-1}$ from the left and by $J_{\mathbf{p}}$ from the right to obtain the new equation

$$J_{\mathbf{f}} J_{\mathbf{p}} = I_q \tag{14}$$

where I_q is the $q \times q$ identity matrix. Therefore, if the Hessian is invertible, the Jacobian of the reconstruction function $\mathbf{f}$ is the left pseudoinverse of the Jacobian of the projection function $\mathbf{p}$.

3.3 Interpretation of the Jacobian

In this section, we show an example of the sensitivity analysis outlined above. The main point here is to show how the Jacobian can be interpreted. Interpretation is not a trivial matter, as there are important issues related to choosing a reference system in which comparing different variables becomes meaningful. This problem also includes choosing an appropriate scaling of all the variables.

The most obvious difficulty stems from the fact that some variables are incommensurable. Most notably, rotations on one hand and translations and point coordinates on the other cannot be compared directly. From an abstract point of view, two approaches to this issue appear possible. We either find a way to scale variables so they become somehow comparable, or we avoid comparisons altogether.

The first option, scaling variables into a common frame, could be realized by considering relative sensitivities instead of absolute ones. In this approach, both image and world variables are divided by their typical values, represented by suitable individual or cumulative averages. Comparison are now possible, because all quantities are dimensionless. However, relative sensitivities are not too useful. Often, a rotation error of, say, one degree is equally significant whether it affects a zero-degree rotation or a sixty-degree rotation. Similarly, an uncertainty of one pixel on an image coordinate can be achieved with similar effort (after lens calibration) at the center and at the periphery of the field of view. In brief, absolute sensitivities are really what we are after.

Another method for scaling variables is to multiply them by coefficients that make the columns of the projection Jacobian equal in norm (this is called column scaling, see for instance [2]). This, however, merely makes all absolute sensitivities similar to each other, without altering the nature of the problem. One would still have to determine whether a sensitivity of, say, one rotation unit is more or less than a sensitivity of one translation unit: quantities are still incommensurable.

A third scaling scheme is to compare camera rotations and translations by their effects on the image. A given camera rotation moves image points of a given scene by a certain amount, measured, say, by the average motion of all the image points. The average motion of image points caused by a camera translation can now be compared with that produced by the rotation in homogeneous units. In other words, the existence of a viewer, and the fact that we are interested in the relation between images and the world, induces an inherent metric for the comparison of rotations and translations: rotations and translations are compared in terms of their perceptual importance.

Whether this comparison is useful, however, depends on the application. For a general analysis, the less committal approach of avoiding comparisons when they are not possible is both conceptually simpler and more general. We propose

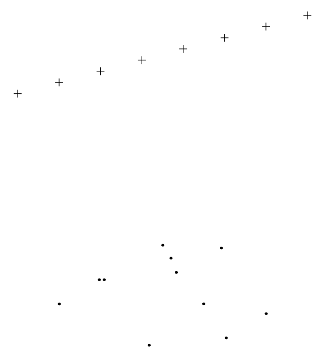


Figure 2: Eight camera positions (crosses) and eleven points in the world (dots). As the camera moves laterally, it fixates one of the points.

to analyze each of shape, translation, and rotation independently, and in terms of their absolute sensitivities. Comparisons between different types of quantities can be made *a posteriori*, once the accuracy requirements of a given application are known. To illustrate, consider a camera moving laterally while observing a cloud of points. Figure 2 shows this scenario. There are eight camera positions and eleven feature points.

The z axis connects the centroid of the world feature points with the centroid of the camera positions. This choice makes z essentially equal to the camera-to-scene distance. The x axis is along the camera motion, and x, y, z are an orthonormal reference system. Projection equations are written for this scenario, and the derivative of each image coordinate is computed analytically with respect to each world variable. Eight images of eleven points yield $11 \times 8 \times 2 = 176$ image coordinates. Of the world parameters, $11 \times 3 = 33$ coordinates specify the positions of the points in space, while $8 \times 6 = 48$ rotation and translation parameters specify the camera positions, for a total of $33 + 48 = 81$ world parameters. This leads to a 176×81 projection Jacobian $J_{\mathbf{p}}$. This Jacobian, however, is singular because the image measurements determine the world parameters only up to an overall scale factor, a rotation, and a translation. To remove this degeneracy, one of the eleven points in space is made to coincide with the origin, and one with point $[1, 0, 0]^T$. A third point is forced to have a y coordinate of zero. The columns of $J_{\mathbf{p}}$ corresponding to these seven fixed components are then removed, leaving a full-rank 176×74 Jacobian. Image coordinates are listed in the 176-dimensional vector

$$\begin{aligned} \mathbf{u} &= [u_{11}, v_{11} \ldots u_{1,11}, v_{1,11} \ldots \\ &\quad \ldots u_{81}, v_{81} \ldots u_{8,11}, v_{8,11}]^T \end{aligned}$$

where the first subscript denotes the frame number and the second the feature number. The world parameters are in the 74-dimensional vector

$$\mathbf{w} = [\mathbf{s}, \mathbf{t}, \mathbf{r}]^T$$

where shape is represented by the $33 - 7 = 26$ components of

$$\mathbf{s} = [x_3, z_3, x_4, y_4, z_4 \ldots x_{11}, y_{11}, z_{11}]^T \ ,$$

translation is the 24-dimensional vector

$$\mathbf{t} = [t_{x1}, t_{y1}, t_{z1} \ldots t_{x8}, t_{y8}, t_{z8}] \ ,$$

and rotation is represented by the 24 entries of

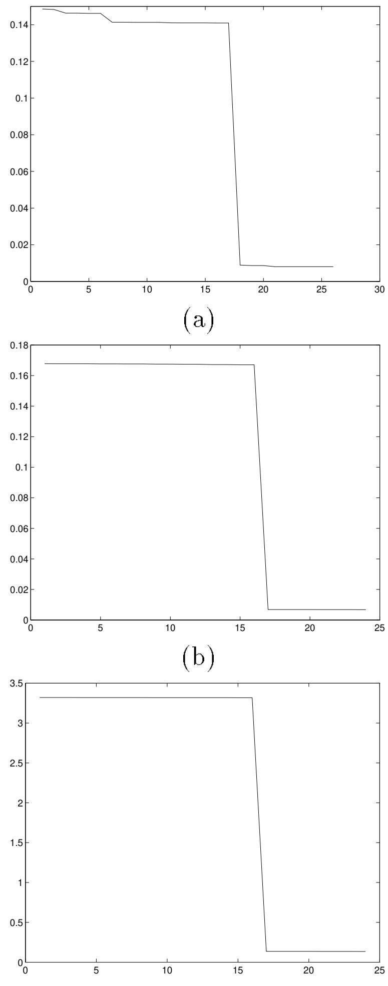
$$\mathbf{r} = [r_{\alpha 1}, r_{\beta 1}, r_{\gamma 1} \ldots r_{\alpha 8}, r_{\beta 8}, r_{\gamma 8}] \ ,$$

which represent the eight camera rotations with quantities similar to Euler angles. Correspondingly, the Jacobian $J_{\mathbf{p}}$ of the projection function can be partitioned into three Jacobians

$$J_{\mathbf{p}} = [\ J_s \ \ | \ \ J_t \ \ | \ \ J_r \ ] \ .$$

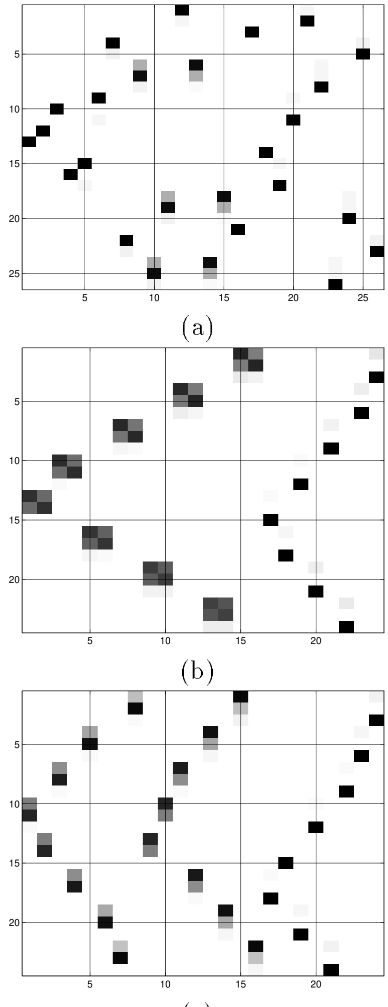
Inverting these Jacobians would not lead to reliable results. In fact, these Jacobians are very poorly conditioned, a direct consequence of the high sensitivity of the reconstruction problem to image perturbations. The part of each Jacobian that can be inverted reliably is revealed by taking its Singular Value Decomposition (SVD). Figure 3 shows the singular values of J_s, J_t, and J_r.

In each plot, there are several large singular values, a clear gap, and several singular values that are much smaller. The magnitudes of the entries of the corresponding right singular vector matrices, shown in figure 4, show which components correspond to the large and to the small singular values, respectively. Each column of a right singular vector matrix corresponds to a singular value, and each row corresponds to an entry of $\mathbf{s}$, $\mathbf{t}$, or $\mathbf{r}$. For instance, figure 4 (a) shows that the 18 larger singular values correspond to the x and y components of shape, while the last eight columns of the right singular vector matrix have nonzero entries essentially in correspondence with the z components. Since in the inverse Jacobians the sizes of the singular values are inverted, this means that the z component of shape (approximately along



(a) (b) (c)

Figure 3: The singular values of (a) the shape Jacobian J_s, (b) the translation Jacobian J_t, and (c) the rotation Jacobian J_r of the projection function.



(a) (b) (c)

Figure 4: The right singular vectors of (a) the shape Jacobian J_s, (b) the translation Jacobian J_t, and (c) the rotation Jacobian J_r of the projection function. Singular vectors are columns, and large entries are dark.

the optical axes) can be reconstructed with an accuracy that is about 17 times worse than for the x and y components. Similar conclusions can be drawn for figures 4 (b) and 4 (c).

These results stand to reason. In fact, they show that in this scenario depth and translation along the optical axis can be recovered only with much less accuracy than the other parameters, because of the relatively large distance between scene and cameras. Also, and for the same reason, a rotation around the optical axis can be determined with accuracy about 24 times worse than a rotation around axes parallel to the image plane.

4 Conclusion

Successful experiments are always the definitive proof that structure-from-motion is feasible. Our preliminary experiments in Flatland show that reconstruction is indeed possible. In addition, they show how explicit attention to sensitivity from the very beginning of the design pays off in the end. The matrix formulation of the input to our reconstruction method provides an interesting handle for understanding sensitivity issues, as we have demonstrated by our arguments regarding singular values of the matrix H in equation (7). However, the performance of our system depends critically on geometric factors like the shape to be recovered and its distance from the camera; on camera calibration; and on the quality of image measurements. Consequently, we do not propose this system as a tool to use directly in the applications, but as a vehicle for investigation of a particular approach to structure-from-motion.

In order to solve structure-from-motion well we must first understand the sensitivity of the problem itself. In this paper, we have presented a framework for this task which we claim is superior to the traditional approach of simulation of noisy images. In fact, our approch is not tied to any particular solution algorithm nor to a particular noise model. Computing derivatives of the reconstruction function, which is only defined implicitly by the optimization criterion, was possible thanks to the implicit function theorem, a mathematical result of deep importance in the analysis of the sensitivity of inverse problems. Again, a matrix proved to be the central element of our analysis. Also, the SVD turned out once more to be the ideal mathematical tool for understanding the structure of the matrix. Our framework needs to be fleshed out, and scaling issues must be addressed in more detail. Most importantly, the theoretical results from our analysis must be compared with results from experiments. All the tools, however, are now in place. We hope that our investigation will soon produce a satisfactory and useful analysis of the sensitivity of structure from motion, and that this analysis will in turn generate useful and reliable algorithms.

A The Jacobian of the Reconstruction Function

In this Appendix, we derive the fundamental equation (13) for the Jacobian of the reconstruction function f. To write the computations compactly, we introduce the following matrix notation for derivatives. Only the following types of derivatives are allowed: derivatives of a scalar with respect to a vector or a matrix, or vice versa; and derivatives of a row vector with respect to a column vector, or vice versa. The meaning of these derivatives is that of either a vector or a matrix, with the following rules. Let $A(r,s)$ be an $r \times s$ matrix (possibly with $r = 1$ or $s = 1$, or both). Then,

$$\begin{aligned}
\frac{\partial A(r,s)}{\partial A(1,1)} &= \left[\frac{\partial A_{ij}}{\partial A_{11}} \right] \\
\frac{\partial A(1,1)}{\partial A(r,s)} &= \left[\frac{\partial A_{11}}{\partial A_{ij}} \right] \\
\frac{\partial A(1,s)}{\partial B(r,1)} &= \left[\frac{\partial A_j}{\partial B_i} \right] \\
\frac{\partial A(r,1)}{\partial B(1,s)} &= \left[\frac{\partial A_i}{\partial B_j} \right]
\end{aligned}$$

where i is the row index and j is the column index.

By replacing the reconstruction function (12) into the normal equations (11), we obtain the new equation

$$\gamma(\mathbf{u}) = \mathbf{g}(\mathbf{u}, \mathbf{f}(\mathbf{u})) = 0 \qquad (15)$$

where only image measurements appear as variables. From the implicit function theorem (see for instance [6]) we obtain

$$\frac{\partial \mathbf{g}}{\partial \mathbf{u}^T} + \frac{\partial \mathbf{g}}{\partial \mathbf{w}^T} \frac{\partial \mathbf{f}}{\partial \mathbf{u}^T} = 0 \; . \qquad (16)$$

Because all quantities involved are vectors, this is a matrix equation. The function $\mathbf{g}$ has q components, because it gathers the partial derivatives of the error e with respect to the q components of the world vector $\mathbf{w}$ (see equation (11)). Since there are hn scalar image measurements in the measurement vector $\mathbf{u}$, the matrix $\frac{\partial \mathbf{g}}{\partial \mathbf{u}^T}$ is $q \times hn$. For similar reasons, $\frac{\partial \mathbf{g}}{\partial \mathbf{w}^T}$ is $q \times q$. Finally, the reconstruction function $\mathbf{f}$ has one component for each of the q entries in the world vector (see equation (12)), so that $\frac{\partial \mathbf{f}}{\partial \mathbf{u}^T}$ is a $q \times hn$ matrix.

Equation (16) expresses the Jacobian $\frac{\partial \mathbf{f}}{\partial \mathbf{u}^T}$ of the implicit function $\mathbf{f}$ as the solution to a linear equation whose coefficients are derivatives of $\mathbf{g}$. We now compute the sensitivity matrix, that is, the Jacobian of the reconstruction function $\mathbf{f}$, by writing the derivatives of the normal function $\mathbf{g}$ in terms of those of the projection function $\mathbf{p}$. This manipulation pays off, since it leads to simplifications. In fact, although the first derivatives of $\mathbf{g}$ are combinations of first and second order derivatives of $\mathbf{p}$, the terms that contain second order derivatives vanish at the solution of the reconstruction problem, leaving only first order derivatives to compute.

To see this, consider the total error

$$e(\mathbf{u}, \mathbf{w}) = \frac{1}{n} \mathbf{r}^T(\mathbf{u}, \mathbf{w}) \mathbf{r}(\mathbf{u}, \mathbf{w})$$

where the residue vector $\mathbf{r}$ is (see equation (8))

$$\mathbf{r}(\mathbf{u}, \mathbf{w}) = \mathbf{u} - \mathbf{p}(\mathbf{w}) \; .$$

The derivatives of $e(\mathbf{u}, \mathbf{w})$ with respect to the world vector $\mathbf{w}$ are

$$\frac{\partial e}{\partial \mathbf{w}^T} = -\frac{2}{n} \mathbf{r}^T \frac{\partial \mathbf{p}}{\partial \mathbf{w}^T}$$

and the second derivatives can be collected in the $q \times q$ *Hessian matrix* of the residue e:

$$\frac{\partial^2 e}{\partial \mathbf{w} \partial \mathbf{w}^T} = \frac{2}{n} \left(\frac{\partial \mathbf{p}^T}{\partial \mathbf{w}} \frac{\partial \mathbf{p}}{\partial \mathbf{w}^T} - \mathbf{r}^T \frac{\partial^2 \mathbf{p}}{\partial \mathbf{w} \partial \mathbf{w}^T} \right) . \tag{17}$$

Although the general expression of the Hessian of e is given by equation (17), we are only interested in the Hessian *at the solution points*, that is, where $\mathbf{w} = \mathbf{f}(\mathbf{u})$. At these points, the residue $\mathbf{r}$ vanishes, and

$$\frac{\partial^2 e}{\partial \mathbf{w} \partial \mathbf{w}^T} = \frac{2}{n} \frac{\partial \mathbf{p}^T}{\partial \mathbf{w}} \frac{\partial \mathbf{p}}{\partial \mathbf{w}^T} , \tag{18}$$

which simplifies the computation of derivatives considerably. A similar argument holds for the derivatives $\frac{\partial \mathbf{g}}{\partial \mathbf{u}^T}$ that appear in equation (16), for which we obtain

$$\begin{aligned} \frac{\partial \mathbf{g}}{\partial \mathbf{u}^T} &= \frac{\partial^2 e}{\partial \mathbf{u}^T \partial \mathbf{w}} = \frac{\partial}{\partial \mathbf{u}^T} \left[-\frac{2}{n} \frac{\partial \mathbf{p}^T}{\partial \mathbf{w}} \mathbf{r} \right] \\ &= -\frac{2}{n} \left[\frac{\partial^2 \mathbf{p}^T}{\partial \mathbf{u}^T \partial \mathbf{w}} \mathbf{r} - \frac{\partial \mathbf{p}^T}{\partial \mathbf{w}} \right] \end{aligned}$$

since

$$\frac{\partial \mathbf{r}}{\partial \mathbf{u}^T} = \frac{\partial \mathbf{u}}{\partial \mathbf{u}^T}$$

is the identity matrix. At solution points, this general expression simplifies to the following, since $\mathbf{r}$ vanishes there:

$$\frac{\partial \mathbf{g}}{\partial \mathbf{u}^T} = \frac{\partial^2 e}{\partial \mathbf{u}^T \partial \mathbf{w}} = -\frac{2}{n} \frac{\partial \mathbf{p}^T}{\partial \mathbf{w}} . \tag{19}$$

By replacing the expressions (18) and (19) into equation (16) with $J_\mathbf{f} = \frac{\partial \mathbf{f}}{\partial \mathbf{u}^T}$ and $J_\mathbf{p} = \frac{\partial \mathbf{p}}{\partial \mathbf{w}^T}$ we obtain the equation (13) for the Jacobian of the reconstruction function $\mathbf{f}$.

References

[1] Fleck M. Shape and the wide-angle image. Technical Report 04, University of Iowa, 1994.

[2] Gill PE, Murray W, Wright MH. *Practical Optimization*. Academic Press, London San Diego New York, 1981.

[3] Horn BKP, Weldon Jr. EJ. Direct methods for recovering motion. *Int'l J Computer Vision*, 1988; 2:51–76.

[4] Koenderink JJ, van Doorn AJ. Facts on optic flow. *Biological Cybernetics*, 1987; 56:247–255.

[5] Shi J, Tomasi C. Good features to track. In: *Proc IEEE Conf Computer Vision and Pattern Recognition*, 1994; 593–600.

[6] Taylor AE, Mann WR. *Advanced Calculus*. John Wiley and Sons, New York, NY, 1983.

[7] Tomasi C. Pictures and trails: a new framework for the computation of shape and motion from perspective image sequences. In: *Proc IEEE Conf Computer Vision and Pattern Recognition*, 1994; 913–918.

[8] Tomasi C, Kanade T. Shape and motion from image streams – a factorization method. In: *Proc Nat'l Acad Sci USA*, 1993; 90(21):9795–9802.

[9] Tomasi C, Zhang J, Redkey D. Experiments with a real-time structure-from-motion system. In: *Proc IV Int'l Symp Experimental Robotics*, 1995; in press.