Multiframe Structure from Motion in Perspective

John Oliensis
NEC Research Institute
4 Independence Way
Princeton, N.J. 08540

Abstract

A new approach to multiframe structure from motion for point features is presented. Unlike previous approaches, it gives robust reconstruction in situations commonly encountered in outdoor robot navigation, for general motion and with large perspective effects. Under the appropriate conditions, the algorithm provably gives the correct reconstruction. The typical computation time is seconds. It is argued that the new approach, combined with previous algorithms valid in other domains (e.g., Tomasi's algorithm), gives a general method for structure from motion.

1 Introduction

This paper describes a new approach (previously described in [13, 12]) to the standard problem of multiframe structure from motion (MFSFM) assuming point features with known correspondences. It reconstructs robustly when perspective effects and the depth variations of 3D points are large, unlike Tomasi's approach [20, 21, 14, 2], and is valid for general motion[1]. The reconstructions are designed to approximate maximum likelihood estimates (MLE) and typically are comparable in accuracy to the MLE. Computation times are on the order of one second[2] in MATLAB on an SGI Indigo II (for 15 images with 30 tracked points); this contrasts with Levenberg—Marquardt (LM) which typically takes about two minutes even when started from the ground truth.

The algorithm is well suited to outdoor navigation sequences with moderate size translations, where all the information in all images is needed for good reconstruction. In this domain, under appropriate and reasonable conditions, the algorithm provably gives the correct reconstruction, and a bound on the reconstruction error can be derived. In domains where it should fail, the algorithm experimentally seems not to converge as is also plausible on theoretical grounds. Thus it seems that when it does converge the algorithm reconstructs reliably.

Previously, the most robust algorithm capable of dealing with outdoor sequences was LM; this algorithm however is slow and unreliable due to the problem of local minima[3]. Other algorithms either require small perspective effects as in Tomasi's approach, or else use a small amount of the available information (typically 2 or 3 image frames) [16, 17, 15, 10, 3, 4, 19] and thus are not robust when more is needed.

1.1 Background and Motivation

Our approach is similar in spirit to Tomasi's [20, 21, 14, 2] but is valid in a different domain. Tomasi's approach is usually described as being based on a linear camera approximation; this is not a useful characterization since it does not specify the physical conditions under which the method is valid. No real camera is linear! A more useful description is: Tomasi's algorithm and extensions are valid when the viewed 3D points are closely clumped compared to their depths relative to the cameras.

We have focused on solving MSFSM in the complementary situation where the depths vary significantly. Here Tomasi's approach is invalid and if applied may give no estimate at all. There are two subcases. In the first, **large baseline** subcase, the relative distance between two of the camera positions is large compared to the depths of the 3D points. We have demonstrated experimentally in previous work that this case is easy: robust and accurate reconstruction can be achieved by standard two frame or other methods. The error function essentially has no false local minima and, e.g., even the notoriously unstable "8 point" algorithm reconstructs reliably. The reasons for the robustness are straightforward: the image displacements are large compared to the noise; translation can be easily distinguished from rotation since the depth variation is

[1] As long as the translational motion is not too big, see below.

[2] For one iteration, which often gives results as accurate as the converged reconstruction.

[3] Although good results have been claimed for LM, they often have been achieved in relatively easy cases, where for instance the camera translations were large, or a large number of feature points were tracked, or a reasonable initial guess was available.

large; the triangulation angles are large giving accurate depth results. Our algorithm handles the second **small baseline** subcase, where the camera translations are smaller than the 3D depths. This subcase is difficult since data from many images is necessary for robust estimation due to the small baselines.

Recently, much research in structure from motion has focused on reconstruction from two or three image frames [16, 17, 15, 10, 3, 4], or else from a small number of feature points tracked across many images [19]. These algorithms are well suited for the large baseline subcase but inadequate for the outdoor navigation domain when the baselines are small and more than 3 frames necessary for robust reconstruction[4]. By definition, only an estimate like the MLE is maximally robust in all situations. But computing it is in general a difficult nonlinear estimation problem; usually such problems have no practical solution. Any estimate that is easy to compute will probably differ from the MLE in some situations and be nonrobust. On the other hand, it is possible to identify restricted domains where practical approximations of the MLE can be computed. This is the approach followed in this paper. By combining approaches which approximate the MLE in different domains, effectively a general solution to MFSFM may be obtained. The above discussion suggests that our algorithm, combined with Tomasi's approach and the 2 or 3 frame algorithms, will give such a general approach covering all subcases.

In this paper we assume known camera calibration. We have studied the sensitivity of our algorithm to camera calibration errors both experimentally and theoretically and have shown that it is insensitive. We have also compared the performance of our algorithm to a projective approach that completely factors out the camera calibration by computing projective structure. Our experimental results indicate that the MLE projective estimates are unstable, often giving worse results for the projective structure itself than does a Euclidean estimate (i.e., an estimate based on an assumed calibration) derived for the incorrect calibration parameters[5]. Even when just the affine or projective structure can be computed robustly a Euclidean algorithm will still give the best estimates for this structure unless the calibration is poor (e.g., the results for the PUMA sequence in [13]).

[4]Unlike in the large baseline case, combining estimates from different image triples in the small baseline case will not help since the large errors in these estimates prevent reliable fusing.

[5]These results were derived using a straightforward LM algorithm as implemented in MATLAB and starting from the ground truth. It is possible that this implementation is nonideal and that other implementations would show less instability (R. Hartley, personal communication)

2 Algorithm Description: Initial Step

Let the first image be selected as a base image. We assume that the translational displacements of the camera from this frame are smaller than the depths measured in this frame. By the standards of computer vision the displacements can be quite large, up to 50% or more of the least 3D depth. Let $\mathbf{b}_i \equiv (x,y)_i$ be the image coordinates of the i-th feature point in the base image, with a total of M feature points. Let $\mathbf{p}_i^h \equiv (x,y)_i^h$ denote the i-th image point in the h-th frame, where $h = \{1, 2, ...N\}$. Assuming focal length 1, the image displacement for a point between the base and h-th image is

$$\mathbf{d}_i^h \equiv \mathbf{p}_i^h - \mathbf{b}_i = \frac{\xi_i (T_z^h \mathbf{b}_i - \mathbf{T}_2^h)}{1 - \xi_i T_z^h} + \mathbf{f}(\mathrm{R}^h, \mathbf{p}_i^h), \quad (1)$$

where $\mathbf{T}_2^h \equiv (T_x^h, T_y^h)^t$, $\xi_i \equiv 1/Z_i^1$ is the inverse depth in the base coordinate system and $\mathbf{f}(\mathrm{R}^h, \mathbf{p}_i^h)$ is the rotational displacement. (1) is exact with no optical flow approximation.

Our algorithm needs no assumption about the rotation sizes since the rotations can be recovered accurately when the translations are moderate. The first step in our algorithm is to approximately recover and compensate for the rotations. The rotation is recovered separately between the base frame and each subsequent frame by a standard SVD technique assuming zero translation.

We have derived an explicit bound on the resulting error in determining the rotation [11]. The bound is exact, not just valid to first order for infinitesimal image displacements. It implies that the rotation error is scaled by the magnitude of the translational image displacements. i.e. by $\lambda \sim |\mathbf{T}||\xi|$. Since λ is small, the rotation error will be as well. We assume from now on that the rotation has been approximately compensated for and is small; typically it is less than 5°–10° in our experiments after compensation. Subsequently, we use a technique similar to that of [5, 18] to cancel the first order effect of the rotation so that the its effect is only second order in λ and relatively insignificant.

Up to first order in small quantities, the image displacements are

$$\mathbf{d}_i^h \approx \mathbf{d}_i^{(1)h} \equiv \xi_i (T_z^h \mathbf{b}_i - \mathbf{T}_2^h)$$
$$+ (\omega_y^h, -\omega_x^h)^t + (-y_i, x_i)^t \omega_z^h - \mathbf{b}_i(\omega_x^h y_i - \omega_y^h x_i), \quad (2)$$

where $\omega^h \equiv (\omega_x^h, \omega_y^h, \omega_z^h)^t$ represents the residual rotation between the first and h–th frame. We have omitted terms of order $\omega\lambda$, ω^2, and λ^2 in (2).

Rewrite the first order matrix of displacements as an $(N-1)\times 2M$ matrix $\mathrm{D}^{(1)}$ by putting all the x and then the y coordinates for a given image on a single row. Let $\bar{\mathbf{T}}_x$ be a $(N-1)\times 1$ vector with elements $\mathbf{T}_x^h$, and similarly define $\bar{\mathbf{T}}_{y,z}$. Also, let $\bar{\omega}_x$, $\bar{\omega}_y$, $\bar{\omega}_z$ be $(N-1)\times 1$ vectors, and let $\bar{\xi}$ be an $M\times 1$ vector. Then (2) is

$$\mathrm{D}^{(1)} = \bar{\mathbf{T}}_x\bar{\mathrm{S}}_x^t + \bar{\mathbf{T}}_y\bar{\mathrm{S}}_y^t + \bar{\mathbf{T}}_z\bar{\mathrm{S}}_z^t$$

$$+\bar{\omega}_x\bar{\mathrm{V}}_x^t + \bar{\omega}_y\bar{\mathrm{V}}_y^t + \bar{\omega}_z\bar{\mathrm{V}}_z^t, \tag{3}$$

$$\begin{aligned}
\bar{\mathrm{S}}_x^t &\equiv -[\bar{\xi}^t, \{0\}^t] && (4)\\
\bar{\mathrm{S}}_y^t &\equiv -[\{0\}^t, \bar{\xi}^t] && (5)\\
\bar{\mathrm{S}}_z^t &\equiv [\{x\xi\}^t, \{y\xi\}^t], && (6)
\end{aligned}$$

$$\begin{aligned}
\bar{\mathrm{V}}_x^t &\equiv [-\{xy\}^t, -\{1+y^2\}^t] && (7)\\
\bar{\mathrm{V}}_y^t &\equiv [\{1+x^2\}^t, \{xy\}^t] && (8)\\
\bar{\mathrm{V}}_z^t &\equiv [-\{y\}^t, \{x\}^t]. && (9)
\end{aligned}$$

Here e.g. $\{xy\}$ is a $M\times 1$ vector with elements $x_i^1 y_i^1$.

$\mathrm{D}^{(1)}$ is rank 6. Since $\bar{\mathrm{V}}_x, \bar{\mathrm{V}}_y, \bar{\mathrm{V}}_z$ are known, we can eliminate the first order rotation terms from $\mathrm{D}^{(1)}$ by postmultiplying it by the appropriate matrix whose transpose annihilates $\bar{\mathrm{V}}_x, \bar{\mathrm{V}}_y, \bar{\mathrm{V}}_z$. The surviving first order terms will then involve only the translation and the structure. The appropriate matrix can be computed easily using Householder matrices [11]. The linear elimination of rotation dependence was first proposed by Heeger and Jepson [7, 5, 8] in the quite different context of recovering motion from optical flow. However, their technique did not extend simply to annihilating general vectors or to sparse flows [7] whereas our Householder–based method does.

Denote the $(2M-3)\times 2M$ matrix annihilating $\bar{\mathrm{V}}_x, \bar{\mathrm{V}}_y, \bar{\mathrm{V}}_z$ derived as above by H_V. The $(N-1)\times(2M-3)$ matrix

$$\mathrm{D}_H \equiv \mathrm{D}\ \mathrm{H}_V^t \approx \mathrm{D}^{(1)}\mathrm{H}_V^t \tag{10}$$

is approximately rank 3 with domain space spanned by the $(2M-3)\times 1$ vectors $\bar{\mathrm{S}}_{Ha} \equiv \mathrm{H}_V\ \bar{\mathrm{S}}_a$, where $a \in \{x,y,z\}$. Note that H_V is derived using the measured image coordinates in the first frame and is therefore errorful. However, the resulting errors in D_H are of order $|\omega||\delta b|$ and are small, where $|\delta b|$ characterizes the size of the image noise. They do not significantly affect the accuracy or convergence of our algorithm.

Next, we recover the translation and structure from D_H using an estimate as close as possible to a MLE under the approximation of neglecting second order effects. This requires solving

$$0 = \mathrm{D}_{\mathrm{R}} \equiv \mathrm{D}_H - (\bar{\mathbf{T}}_x\bar{\mathrm{S}}_x^t + \bar{\mathbf{T}}_y\bar{\mathrm{S}}_y^t + \bar{\mathbf{T}}_z\bar{\mathrm{S}}_z^t)H_V^t \tag{11}$$

taking account of the covariance of D_H:

$$\left\langle \mathrm{D}_{Hi}^h \mathrm{D}_{Hj}^{h'} \right\rangle = \sigma^2(\delta_{hh'} + 1_{hh'})\delta_{ij}, \tag{12}$$

where δ is the Kronecker delta function. Let $(\mathrm{C}_H)_{hh'} \equiv \delta_{hh'} + 1_{hh'}$. Within the first order approximation, the MLE is that minimizing

$$\mathrm{Tr}(\mathrm{D}_{\mathrm{R}}\mathrm{D}_{\mathrm{R}}^t\mathrm{C}_H^{-1}). \tag{13}$$

C_H appears because all image displacements are with respect to the base frame and each term of D reflects the errors from this frame. Minimizing (13) rather than $\mathrm{Tr}\mathrm{D}_H\mathrm{D}_H^t$ eliminates (at this stage) all bias in the estimate due to singling out the base frame for special treatment, distinguishing our approach from earlier work such as [1].

The inverse square root of C_H is

$$[\mathrm{C}_H^{-1/2}]_{hh'} = \delta_{hh'} - \frac{1+N^{-1/2}}{N-1}1_{hh'}. \tag{14}$$

Define $\mathrm{D}_{CH} \equiv \mathrm{C}_H^{-1/2}\ \mathrm{D}_H$ and $\bar{\mathbf{T}}_{C(x,y,z)} \equiv \mathrm{C}_H^{-1/2}\ \bar{\mathbf{T}}_{(x,y,z)}$. Minimizing (13) is equivalent to solving in the least square sense

$$0 = \mathrm{D}_{CH} - (\bar{\mathbf{T}}_{Cx}\bar{\mathrm{S}}_x^t + \bar{\mathbf{T}}_{Cy}\bar{\mathrm{S}}_y^t + \bar{\mathbf{T}}_{Cz}\bar{\mathrm{S}}_z^t)H_V^t. \tag{15}$$

Our algorithm, as in [20], does this approximately in two stages. First, the best rank three approximation to D_{CH} is computed, e.g. using the SVD. Let the three $2M-3\times 1$ vectors $\bar{\mathrm{A}}_i,\ i=1,2,3$ represent the three leading right singular vectors D_{CH} and define

$$\mathrm{A} \equiv \begin{pmatrix} \bar{\mathrm{A}}_1^t \\ \bar{\mathrm{A}}_2^t \\ \bar{\mathrm{A}}_3^t \end{pmatrix}. \tag{16}$$

The three vectors $\bar{\mathrm{S}}_{\{x,y,z\}}$ must lie in the subspace spanned by the $\bar{\mathrm{A}}_i$ (to this order of approximation). In the second stage, we use this fact to solve for the structure. Write the three linear equations

$$\begin{aligned}
\bar{\mathrm{S}}_x^t &\equiv [\{\xi\}^t, \{0\}^t]\mathrm{H}_V^t &&= \mathbf{c}_x^t\mathrm{A}\\
\bar{\mathrm{S}}_y^t &\equiv [\{0\}^t, \{\xi\}^t]\mathrm{H}_V^t &&= \mathbf{c}_y^t\mathrm{A}\\
\bar{\mathrm{S}}_z^t &\equiv [\{\xi x\}^t, \{\xi y\}^t]\mathrm{H}_V^t &&= \mathbf{c}_z^t\mathrm{A} && (17)
\end{aligned}$$

where the $\mathbf{c}_x$, $\mathbf{c}_y$, $\mathbf{c}_z$, are unknown 3×1 vectors. This can be solved by least squares for the inverse depths ξ

and the 9 unknowns contained in the $\mathbf{c}_a$. The translations are calculated similarly [11]. The two stage method, like the one in Tomasi's approach [20], does not correspond to a MLE directly; however, the first stage does give a MLE for the leading three dimensional singular subspaces of D_{CH}. Since the singular subspaces overdetermine the structure and translations, the second stage generally introduces little additional error sensitivity.

The second stage does reintroduce some dependence on the errors in the base images, since the base image coordinates appear explicitly in the equations (17). This dependence is not strong, however, since the base image is involved only in the z equation. Finally, the equations (17) do not correspond as written to a MLE since the noise is in the recovered singular spaces rather than in the $\bar{\mathrm{S}}_a$. To recover a better approximation to the MLE for the second stage, an iterative algorithm can be used. One iteration in general suffices. But in practice a direct least square solution of (17) gives good results.

The initializing algorithm described in this section often gives adequate results for the structure.

2.1 Iterative Step

Two approximations are made in the initial step of our algorithm: second order terms due both to the rotation error and due to the denominator in (1) were neglected. The iterative step of the algorithm corrects for these two approximations in turn.

First, the rotations are recalculated. For each frame h, the base frame is warped towards frame h using the current estimates of the translational image displacements; then the rotation between frame h and the base frame is calculated and compensated for as before. The new rotation error is second order.

The denominator error is corrected by multiplying the matrix of image displacements by the current estimate of the denominator. Let $\mathbf{T}^h_{\{x,y,z\}E}$ and ξ_{Ei} denote the estimated translations and inverse depths. Define

$$\begin{aligned} \mathbf{d}'^h_i &\equiv \mathbf{d}^h_i(1 - \mathbf{T}^h_{Ez}\xi_{Ei}) \\ &= \xi_i(\mathbf{T}^h_z\mathbf{b}_i - \mathbf{T}^h_2)\left(1 + \frac{\delta(\mathbf{T}_z\xi)^h_i}{1 - \mathbf{T}^h_{zE}\xi_{Ei}}\right) \\ &\quad + (1 - \mathbf{T}^h_{zE}\xi_{Ei})o(|\omega|), \end{aligned} \tag{18}$$

where the last term represents all terms involving the rotations, and where $\delta(\hat{T}_z\xi)^h_i \equiv (\mathbf{T}^h_z\xi_i - \mathbf{T}^h_{zE}\xi_{Ei})$. Now the earlier method for estimating translation and structure is reapplied using $\mathbf{d}'^h_i$ instead of $\mathbf{d}^h_i$. $\mathbf{d}'^h_i$ is a bilinear function of the unknowns up to errors which are now second order, implying that the new estimates will be improved compared to the initial ones. The iterative steps just described are repeated until convergence.

The iterative step departs slightly from the ideal MLE: multiplying by the denominator suppresses the information from small depth points although these are actually the most informative. But because the estimate does use redundant information from all points, this deviation from the MLE (of first order in λ) should not cause significant errors. This is borne out in our experiments and in our theoretical analysis. The multiplicative step has another disadvantage: postmultiplying by H^t_V no longer eliminates the first order rotational terms exactly. Again, however, the surviving terms are insignificant since they are scaled by $|\mathbf{T}_z\xi| \sim \lambda$. It is possible to define an iteration which avoids these disadvantages by correcting the matrix of displacements additively rather than multiplicatively, but its domain of convergence is significantly smaller than for the multiplicative iteration.

3 Experiments

Because we approximate D_{CH} as rank 3, our algorithm requires the translational motion to be general. It will not work if the translations are clustered in a plane or along a line as for most real image sequences; in such cases the algorithm presented here must be modified. We have already presented results for real and synthetic sequences using appropriately modified algorithms in [13, 12] and shown experimentally that it is possible to choose the appropriate method directly from the data.

Here we test our algorithm on synthetic sequences. We conducted trials in which the motion, structure, and image noise were varied randomly at each trial. 30 points and 15 image frames were used in all trials. The translations from the base frame were uniformly chosen in a cubic volume $-8 \leq T_{\{x,y,z\}} \leq 8$. The "ground truth" depths of points were varied randomly with a uniform distribution over the range (typically) from 30 to 90 feet, while the other coordinates were varied uniformly in the range (typically) of -15 to +15 feet. Random rotations were added up to a maximum of about 20° (we have also used larger rotations). Finally, image noise from a uniform distribution, typically with maximum size .004 in focal length units (flu), was added independently to each image point. We have experimented with noises up to .016 flu. Typically, the field of view was about 90° implying that one pixel is .004 flu for a 512×512 image.

We ran twenty trials as described above comparing the results of our algorithm to those of a MLE (obtained by a LM minimization initialized from the ground truth). After a scaling of the computed depths

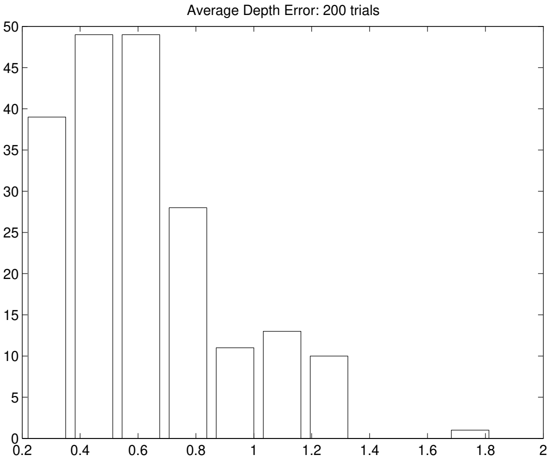

Figure 1: Noise: .004, range 30–90

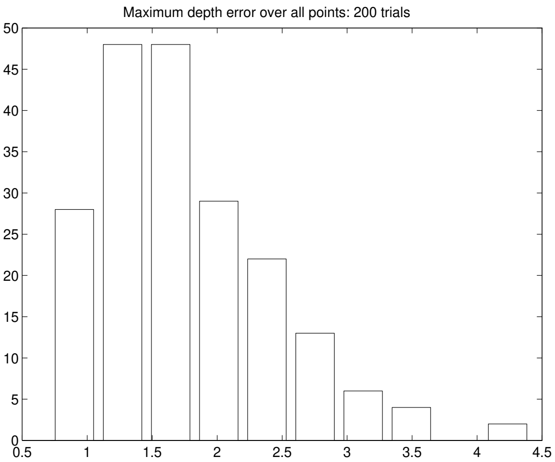

Figure 2: Noise: .004, range 30–90

to the ground truth depths, the average errors in the depths were computed and compared. We computed the ratios between the average errors obtained by our algorithm and those obtained from the MLE. On average over the 20 trials, our algorithm produced an average depth error 1.52 times that of the MLE. The average depth error for MLE, averaged over trials, was .37 feet, and it was .56 for our algorithm. The maximum error made by our algorithm for any point over all trials was 2.8 feet. Results from 200 trials of our algorithm alone appear in Figures 1 and 2.

We compared the directions of the recovered translations to the ground truth directions and recorded the angular errors. For each trial, the average error was typically about 1° for both the MLE and our algorithm. The errors were also about the same for the translation magnitudes. Our algorithm was about

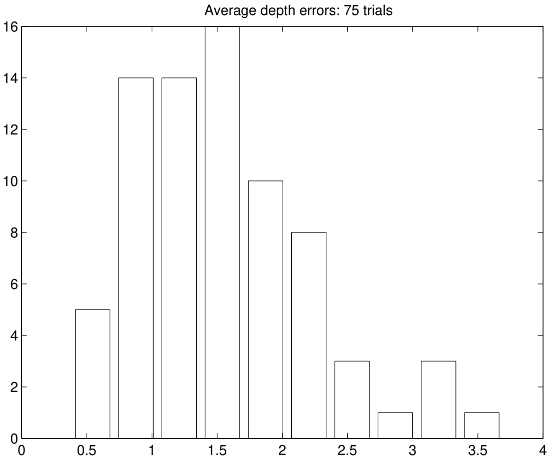

Figure 3: Noise: .008, range: 30–90

20% worse in determining the rotations than was the MLE. The maximum error in recovering the rotation over all frames and all trials was 1.3°. On average, the rotations were recovered to .3° by our algorithm.

We ran 75 comparison trials to MLE under the same conditions as before but with a maximum image noise of .008 flu (Figures 3 and 4). On average over trials, the average depth error for our algorithm was now about 2.1 times that for the MLE algorithm. Unsurprisingly, it shows less graceful degradation under noise than does the MLE. Figure 5 shows the average depth errors obtained after a single iteration; they are nearly as good as the final estimates. The translation directions were recovered about 31% worse by our algorithm than by the MLE, and the magnitudes were recovered 28% worse. Our algorithm recovered the rotations worse by a factor of 1.8 on average but still typically to within less than a degree. The maximum rotation errors over all frames are shown in Figure 6.

In these and other trials run under the same conditions, our algorithm converged to within one part in 10^4 after about 2-3 iterations of the full iterative step, where in each step about 4 iterations of the denominator correction step were necessary.

We also ran trials where the depth varied uniformly between 20 and 100 feet. For 50 trials with a maximum noise of .006 flu, the average depth error was .86 feet. The maximum depth error for any point over all 50 trials was 5.6 feet. We have run a 100 trial comparison of our algorithm to the MLE under these conditions but with .008 flu maximum noise. The average depth error for our algorithm and for the MLE were computed for each trial (Figures 7). On average, our algorithm does only 39% worse than the MLE de-

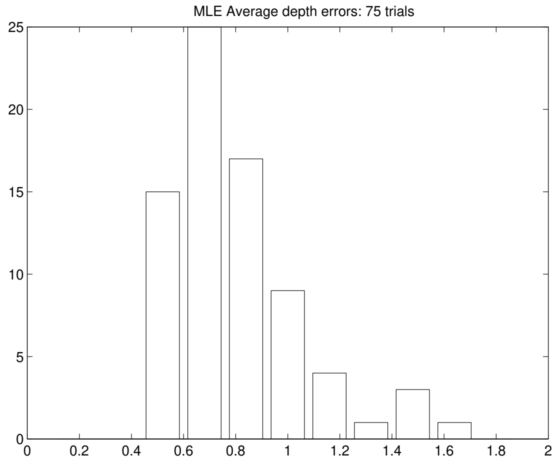

Figure 4: Noise: .008, range: 30–90

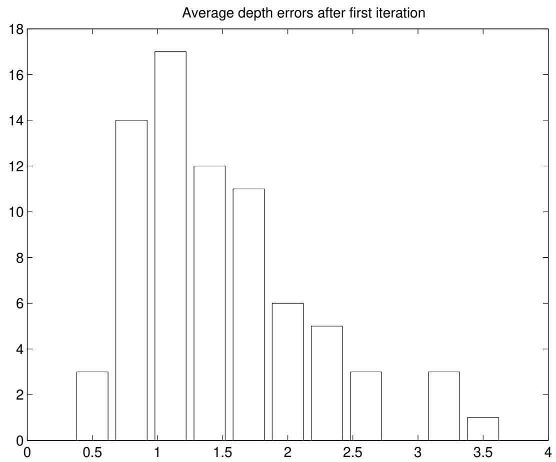

Figure 5: Noise: .008, range: 30–90

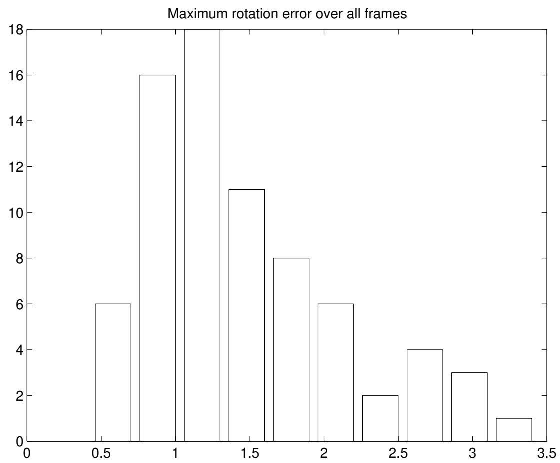

Figure 6: Noise: .008, range: 30–90

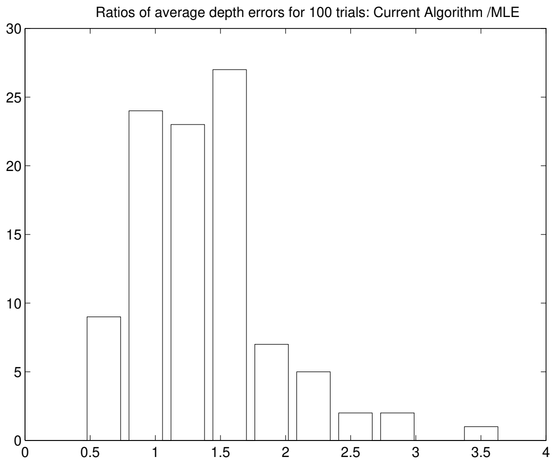

Figure 7: Noise .008, range 20–100

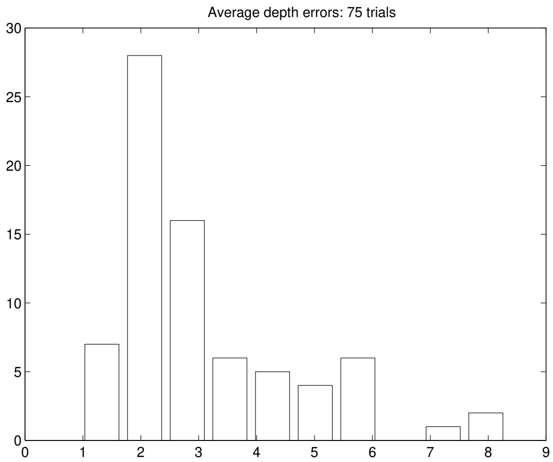

Figure 8: Noise: .016, range: 20–100

spite the large noise. Averaged over trials, the average depth error for our algorithm was 1 foot. The maximum depth error over all frames and all trials was 7.4. Our algorithm does about as well as the MLE in estimating the translations. The rotations are estimated about 20% worse; the average rotation error is .4° and the maximum error over all frames and trials is 1.6°.

To test our algorithm with a large noise, we ran 75 trials as above but with a maximum noise of .016 flu (Figures 8 and 9). The algorithm converged in all trials. On average, our algorithm was worse than a MLE for the average depth error by a factor of 2. This was also true for the error in the rotation recovery. Our algorithm recovered the translations about 30% worse than the MLE. The average depths recovered after the first iteration are about as accurate as the final converged results (Figure 10).

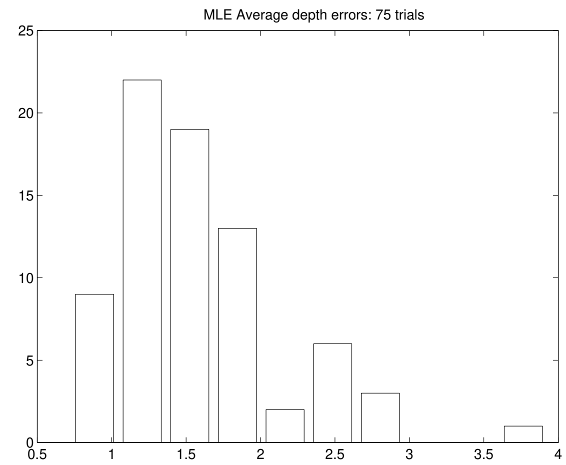

Figure 9: Noise: .016, range: 20–100

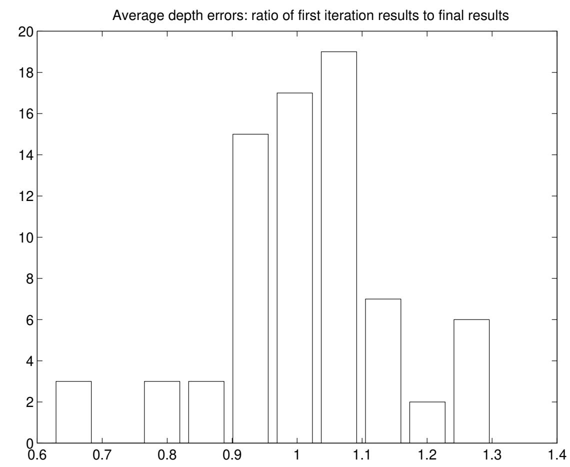

Figure 10: Noise: .016, range: 20–100

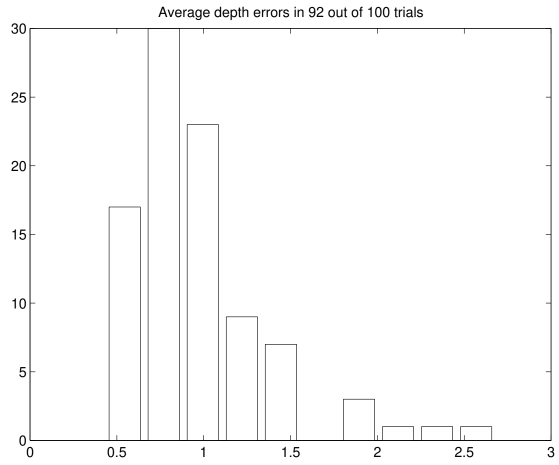

Figure 11: Noise .008, range 15–105

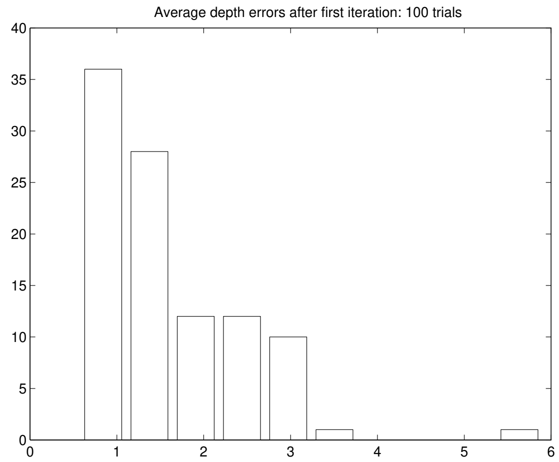

Figure 12: Noise .008, range 15–105

We also ran experiments where the depth range was 15–105 feet. Here the maximum size of the translation can be over 50% of the least 3D depth so that convergence becomes problematic. In 100 trials with .008 noise, the algorithm converged in 92. The average depth errors when convergence was achieved are shown in Figure 11. For these trials, the average rotation error was .38 degrees. The average depth errors obtained after the first iteration for all 100 iterations are shown in Figure 12. Note that the result after the first iteration is always reasonably accurate even when there is no convergence. It is possible to bound theoretically the first iteration error even without convergence.

We ran a similar set of trials where the depths for 20 points varied uniformly in the range 15 to 45, while 10 points had depths in the range 100–300. Since the

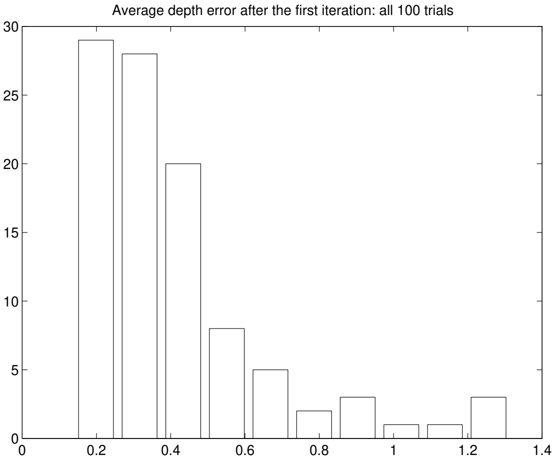

Figure 13: Noise .008, range 15–45, 100–300

latter have depths too large to be reconstructed accurately, we report results only for the former. In this experiment, we found convergence for 91 out of the 100 trials. In the converged trials, the average depth error for the 20 near points was .18. The maximum depth error over all frames and converged trials was 1.4 feet. The average depth error after the first iteration averaged over all 100 trials was .41 feet (Figure 13). Again, the first iteration seems to provide good and reliable accuracy. An additional 10 trials were run with depths in the range of 12–88 feet. The algorithm converged in all these trials.

References

[1] A. Azarbayejani, B. Horowitz, and A. Pentland, "Recursive estimation of structure and motion using relative orientation constraints," *CVPR*, 294-299, 1993.

[2] Stephane Christy and Radu Horaud, "Euclidian Shape and Motion from Multiple Perspective Views by Affine Iterations," INRIA TR 1994.

[3] R. Hartley, "Lines and points in three views—a unified approach," *IUW*, 1009-1016, 1994.

[4] R. Hartley, "Estimation of Relative Camera Positions for Uncalibrated Cameras," *ECCV*, 579-587, 1992.

[5] D.J. Heeger and A.D. Jepson, "Subspace methods for recovering rigid motion I: Algorithm and implementation," IJCV 7, 95-117, 1992.

[6] R. Hummel and V. Sundareswaran, "Motion parameter estimation from global flow field data," PAMI 15, 459-476, 1993.

[7] A.D. Jepson and D.J. Heeger, "Linear subspace methods for recovering translational direction," University of Toronto RBCV-TR-92-40, 1992, page 19.

[8] A.D. Jepson and D.J. Heeger, "A fast subspace algorithm for recovering rigid motion," *Motion Workshop*, 124-131, 1991.

[9] A.D. Jepson and D.J. Heeger, "Subspace methods for recovering rigid motion II: Theory," University of Toronto RBCV-TR-90-36, 1990.

[10] R. Kumar, P. Anandan, K. Hanna, "Shape recovery from multiple views: a parallax based approach," *IUW*, 947-955, 1994.

[11] J. Oliensis, "Multiframe Structure from Motion in Perspective," NEC TR, April 1995, http://www.neci.nj.nec.com/homepages/oliensis .html.

[12] J. Oliensis, "A Linear Solution to Multiframe Structure from Motion: Constant Translation Case," NEC Technical Report, December 1994.

[13] J. Oliensis, "A Linear Solution to Multiframe Structure from Motion," *IUW*, 1225-1231, 1994.

[14] C. Poelman, T. Kanade, "A Paraperspective Factorization Method for Shape and Motion Recovery," *ECCV*, 97-108, 1994.

[15] H. Sawhney, "3D Geometry from Planar Parallax," *CVPR*, 929-934, 1994.

[16] A. Shashua, "Trilinearity in visual recognition by alignement," in *ECCV* Vol. 1 479-484, 1994.

[17] A. Shashua, R. Navab, "Relative Affine Structure: Theory and Applications to 3D Reconstruction From Perspective Views," *CVPR*, 483-489, 1994.

[18] V. Sundareswaran, "Egomotion from global flow field data," *Motion Workshop*, 140-145, 1991.

[19] C. Tomasi, "Pictures and Trails: a New Framework for the Computation of Shape and Motion from Perspective Image Sequences," *CVPR*, 913–918, 1994.

[20] C. Tomasi and T. Kanade, "Shape and motion from image streams under orthography: A factorization method," IJCV 9, 137-154, 1992.

[21] D. Weinshall and C. Tomasi, "Linear and incremental acquisition of invariant shape models from image sequences," *ICCV*, 675-682, 1993.