A Linear Solution for Multiframe Structure from Motion

John Oliensis*
Department of Computer Science
University of Massachusetts at Amherst
Amherst, Massachusetts 01003

Abstract

We focus on the problem domain of a robot navigating in and reconstructing an unknown environment from a sequence of images. We argue that the correct approach is to find the appropriate approximation that linearizes the problem, yielding fast, non–iterative algorithms that compute structure and motion with no initial guess. A class of algorithms (batch and recursive) is developed that accomplishes this, where the appropriate algorithm depends on the particular image sequence. Experiments are described on the PUMA sequence [13] and the Rocket Field sequence [2].

1 Introduction

The approach to multiframe structure from motion (MFSFM) for point features described here may be seen as a generalization of the earlier work by Tomasi [16] to the case of full perspective—for instance, our approach works well on the Rocket Field sequence, where perspective effects are crucial. It also relates to the work of Heeger and Jepson [6, 4, 7, 5, 15] on recovering translational motion from optical flow, yielding a simple algorithm for recovering translation from sparse as well as dense optical flow.

Our motivation is to develop methods for MFSFM that are approximate but make effective use of the information available and are fast. They are intended to give good, reliable structure/motion estimates quickly, with no initial guess required, using modest amounts of data. In our experiments, the algorithms using nonoptimized MATLAB code generated structure/motion estimates comparable to those of a full maximum likelihood estimate (MLE) in less than 5 seconds on a DecStation 5000. The methods described here are for special cases of motion but they can be generalized to unconstrained motion.

*This work was supported by the National Science Foundation under grants IRI-9113690, CDA-8922572, and by a grant from DARPA, via TACOM, contract number DAAE07-91-C-R035.

2 $T_z \approx 0$ Case

We first describe our technique and present experimental results on a real image sequence for a useful special case, corresponding to an aerial cartography scenario. The assumption is that the translational motion of the camera is largely in the x–y plane, where z is the optical axis at some camera position. For instance, this will be the case for an airplane flying over a landscape with downward–pointed camera. The assumption on the motion need not be exactly satisfied.

Let one image—the first—be selected as a base image and let $\mathbf{b}_i \equiv (x,y)_i$ be the image coordinates of the i-th feature point in this image, with a total of M feature points. Let $\mathbf{p}_i^h \equiv (x,y)_i^h$ denote the the i-th image point in the h-th frame, where $h = \{1, 2, ...N\}$. Assuming a focal length of 1, the displacement in the image position of a feature point between the base image and the h-th image is

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

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.

Throughout, we will assume that the rotational displacement $\mathbf{f}$ can be approximated to first order. This is possible if the rotation between images is small—or if the rotation can be approximately recovered and compensated for in a preprocessing stage. It is important to realize that, for egomotion where a moving camera navigates in a fixed scene, approximate recovery of the rotation often is easily achievable.

Consider 2 images that differ only through a 3D rotation of the camera. This rotation can be simply recovered using standard techniques. For general motion, if the translation baseline between 2 images and the resulting image displacements are not too large (i.e., for the sequences considered here, for translation steps < 10–20 feet), then applying these techniques will recover the rotation to a good approximation. After it is compensated, a first order approximation of the residual rotation should be adequate and our small rotation assumption justified. Experiments support these statements.

Our algorithm also assumes that the cumulative translation over the course of the image sequence is not too large—again, less than 10–20 feet for the sequence considered here. One might as well assume this since otherwise the structure from motion problem becomes easy, at least when there is significant depth variation in the scene (in the standard robot navigation problem, the depth variation is expected to be significant). This too is supported by our experiments: for large translation (here > 10 feet) and depth variation, a standard 2 frame structure from motion algorithm will likely give robust and accurate results.

For large translation, it is anyway unlikely that feature points will be kept in view throughout or that correct tracking will be possible over an extended sequence. For a long sequence with large overall translation, the best strategy may be to break it down into shorter subsequences, long enough to contain enough information for a robust structure estimate yet short enough so that features points appear in all or most images. The algorithm described in this paper is well suited to the task of estimating structure from a moderate length sequence.

2.1 $T_z \approx 0$ Case: Formulation

Assuming a small rotation and moderate translation, the rotational displacement has the familiar optical flow form. Also, since, $\mathbf{T}_z^h \approx 0$, (1) becomes

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

Rewrite the matrix of displacements as an $(N-1) \times 2M$ matrix D by putting all the x and then the y coordinates for a given frame on a single row. Let $\bar{\mathbf{T}}_x$ be a $(N-1) \times 1$ vector with elements $\mathbf{T}_x^h$. Similarly, 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

$$\mathrm{D} = [-\bar{\mathbf{T}}_x \bar{\xi}^t, -\bar{\mathbf{T}}_y \bar{\xi}^t] + \bar{\omega}_x \bar{\mathrm{V}}_x^t + \bar{\omega}_y \bar{\mathrm{V}}_y^t + \bar{\omega}_z \bar{\mathrm{V}}_z^t, \tag{3}$$

where

$$\bar{\mathrm{V}}_x^t \equiv [-\{xy\}^t, -\{1+y^2\}^t] \tag{4}$$
$$\bar{\mathrm{V}}_y^t \equiv [\{1+x^2\}^t, \{xy\}^t] \tag{5}$$
$$\bar{\mathrm{V}}_z^t \equiv [-\{y\}^t, \{x\}^t]. \tag{6}$$

Here e.g. $\{xy\}$ is a $M \times 1$ vector with elements $x_i y_i$. Note that D is rank 5: its domain space is spanned by $\bar{\mathrm{V}}_x, \bar{\mathrm{V}}_y, \bar{\mathrm{V}}_z$ and

$$\bar{\mathrm{S}}_1 \equiv \begin{pmatrix} \xi \\ \{0\} \end{pmatrix}, \quad \bar{\mathrm{S}}_2 \equiv \begin{pmatrix} \{0\} \\ \xi \end{pmatrix}, \tag{7}$$

where $\{0\}$ is again a $M \times 1$ vector of zeros.

Since $\bar{\mathrm{V}}_x, \bar{\mathrm{V}}_y, \bar{\mathrm{V}}_z$ are known, we design a measurement matrix from D which is rank 2 with right singular vectors given by just (7). This is done by postmultiplying D by a rank $2M-3$ matrix annihilating $\bar{\mathrm{V}}_x, \bar{\mathrm{V}}_y, \bar{\mathrm{V}}_z$. Heeger and Jepson used this basic trick [6, 4, 7] in the context of recovering motion from optical flow, but their technique did not extend to annihilating general vectors or to sparse flows. Our extension is based on Householder matrices. The Householder matrix [3] H_{ab} is an orthogonal matrix that takes $\mathbf{a}$ to $\mathbf{b}$ by a reflection. With $\hat{\mathbf{n}} \equiv (0, 0 \ldots 1)^t$, the matrix defined by the first $n-1$ rows of $H_{a\hat{n}}$ is a rank $n-1$ matrix annihilating $\hat{\mathbf{n}}$. Matrices annihilating multiple vectors can be computed by taking appropriate products of matrices derived in this way.

Algorithm

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 $\mathrm{D}_H \equiv \mathrm{D}\ \mathrm{H}_V^t$ is rank 2 with domain space spanned by $\bar{\mathrm{S}}_{H1} \equiv \mathrm{H}_V\ \bar{\mathrm{S}}_1$ and $\bar{\mathrm{S}}_{H2} \equiv \mathrm{H}_V\ \bar{\mathrm{S}}_2$, where these are $(2M-3) \times 1$ vectors. We identify the leading 2 dimensional right subspace of D_H using the SVD; the inverse depths can then be computed as follows. $\bar{\mathrm{S}}_{Hi}$ must be contained in this subspace. Let $\bar{\mathrm{A}}_i,\ i = 1, 2$ represent the 2 leading right singular vectors of D_H. Then

$$H_V\ (\ \bar{\mathrm{S}}_1,\quad \bar{\mathrm{S}}_2\ ) = (\ \bar{\mathrm{A}}_1,\quad \bar{\mathrm{A}}_2\ )\, C_2 \qquad (8)$$

for some 2×2 matrix C_2. This is a system of $4M - 6$ linear equations which can be solved by least squares for the $M+4$ unknowns ξ_i and the coefficients of C_2.

We next modify the algorithm to better correspond to a MLE. The data available for the estimate are contained in the displacement matrix D_H. With the standard Gaussian noise model with standard deviation σ, the covariance of the measurement matrix D_H is

$$\langle \mathrm{D}_{Hhi} \mathrm{D}_{Hh'j} \rangle = \sigma^2 (\delta_{hh'} + 1_{hh'}) \delta_{ij}, \qquad (9)$$

where $1_{hh'}$ is an $(N-1)\times(N-1)$ matrix of ones, appearing because each displacement entry involves the base image. Let $(\mathrm{C}_H)_{hh'} \equiv \delta_{hh'} + 1_{hh'}$. Define the residual matrix

$$\mathrm{M_R} \equiv \mathrm{D}_H + \bar{\mathbf{T}}_x S^t_{H1} + \bar{\mathbf{T}}_y S^t_{H2} = 0, \qquad (10)$$

then the MLE of the inverse depths can be shown to be approximately that minimizing

$$\mathrm{Tr}(\mathrm{M_R} \mathrm{M_R^t} \mathrm{C}_H^{-1}). \qquad (11)$$

It is important that the matrix C_H^{-1} correctly compensates for the calculation of the displacement with respect to the base image. Our approach makes no important approximation in singling out one image for special treatment, in contrast to the algorithm of [1].

The inverse square root of C_H can be computed exactly: it is

$$[\mathrm{C}_H^{-1/2}]_{hh'} = \delta_{hh'} - \frac{1 + N^{-1/2}}{N-1} 1_{hh'} \qquad (12)$$

Define $\mathrm{D}_{CH} \equiv \mathrm{C}_H^{-1/2}\ \mathrm{D}_H$ and similarly $\bar{\mathbf{T}}_{C(x,y)} \equiv \mathrm{C}_H^{-1/2}\ \bar{\mathbf{T}}_{(x,y)}$. Then the energy (11) for a MLE can be rewritten as Tr $(\mathrm{M_{RC}}\ \mathrm{M^t_{RC}})$ where

$$\mathrm{M_{RC}} \equiv \mathrm{C}_H^{-1/2}\ \mathrm{M_R} = \mathrm{D}_{CH} + \bar{\mathbf{T}}_{Cx}\ \bar{\mathrm{S}}^t_{H1} + \bar{\mathbf{T}}_{Cy}\ \bar{\mathrm{S}}^t_{H2}. \qquad (13)$$

The improved algorithm is summarized as follows: first the measurement matrix $\mathrm{D}_{CH} = \mathrm{C}_H^{-1/2}$ D H_V is constructed. Its 2 leading right singular vectors $\bar{\mathrm{A}}_{1,2}$ are computed and the ξ_i are computed from these by solving (8). The translations $\mathbf{T}^h_x$ and $\mathbf{T}^h_y$ can also be recovered.

2.2 Experimental Results

In experiments with the algorithm described in the previous subsection, some problems were observed. Most important, this algorithm requires a motion such that D_H, D_{CH} are strongly rank 2; otherwise, the lesser singular vector will be contaminated by noise (the perturbing effect of noise on the singular vector is inversely related to the size of the singular value). D_{CH} will be strongly rank 2 if the motion is truly planar; however many motions do cluster around a linear motion. Thus it would be preferable to have an algorithm that only requires extracting the largest singular vector. Such an algorithm was developed and used for the experiments described below. Its description is omitted for lack of space.

We applied our algorithm to part of the PUMA image sequence obtained by R. Kumar and H. Sahnwey, described in [9, 10]. Specifically, we used a sequence of 32 automatically tracked feature points over 16 image frames tabulated and provided to us by J. I. Thomas. The algorithm of [16], when applied to this sequence, fails to produce a reconstruction.

First, the images were unrotated automatically to match the first (base) image frame. We used the fact that the rotation was known to be primarily around the optical axis and unrotated around this axis only. The rotation was successfully compensated to within a maximum error of 2.3°, well within the small angle approximation. For comparison, we also unrotated compensating for arbitrary rotation. A maximum error after compensation of 4.3° was found. These results were achieved despite the large overall rotations of up to 60.5°.

The results for the inverse structure in the coordinate system of the first (base) image are summarized in Table 1. The results are not simply a list of the inverse depths for different feature points but have been rotated to a different basis. This was done since two components of the inverse structure are recovered less well than the others due to the bas–relief ambiguity—the changed basis displays these components explicitly as the last two.

Table 1 shows the inverse depth components computed by our algorithm and by an MLE, scaled to compare to the ground truth. The MLE was performed by a brute force Levenberg–Marquardt (LM) minimization of an objective function that summed explicitly the

Table 1: PUMA Sequence: Inverse Depth Results. New, True, and LM respectively label the results for the current algorithm, the ground truth, and the results of a MLE. For compactness, the 32 components of the inverse depth vector are displayed in 8 rows of 4 components each.

New	-0.0101	-0.0061	-0.0114	-0.0040
True	-0.0105	-0.0057	-0.0114	-0.0040
MLE	-0.0100	-0.0063	-0.0116	-0.0040
New	-0.0042	0.0111	-0.0075	-0.0050
True	-0.0038	0.0113	-0.0070	-0.0054
MLE	-0.0041	0.0110	-0.0073	-0.0052
New	0.0022	-0.0016	0.0012	-0.0055
True	0.0016	-0.0018	-0.0004	-0.0056
MLE	0.0021	-0.0017	0.0010	-0.0054
New	0.0039	-0.0062	0.0015	0.0023
True	0.0040	-0.0057	0.0020	0.0019
MLE	0.0041	-0.0060	0.0018	0.0024
New	-0.0012	0.0024	0.0036	-0.0029
True	-0.0013	0.0026	0.0021	-0.0033
MLE	-0.0010	0.0022	0.0036	-0.0029
New	0.0084	-0.0031	0.0034	-0.0043
True	0.0090	-0.0031	0.0027	-0.0042
MLE	0.0082	-0.0031	0.0037	-0.0045
New	-0.0053	-0.0022	0.0081	-0.0066
True	-0.0054	-0.0027	0.0080	-0.0074
MLE	-0.0053	-0.0022	0.0081	-0.0066
New	0.0173	0.0639	-0.3717	-0.1373
True	0.0287	0.0629	-0.2544	-0.105
MLE	0.0187	0.0655	-0.4171	-0.1518

image discrepancies between the measured image points and the projected positions of the 3D feature points. The minimization was carried out with respect to the 3D coordinates of the feature points and the interframe motion parameters, and started with the ground truth values of these parameters (the rotations were started at zero rotation for the unrotated images). The algorithm took over half an hour to converge, despite the head start from the ground truth.

It is clear that most components of the structure are recovered very well; however, some of the larger structure components are recovered imperfectly. Our algorithm recovers all components about as well as or better than the MLE—thus its performance is as good as could be expected on this sequence. The average percentage error in depth after scaling for our algorithm is 11.5%. However, these results are less informative than those of Table 1, since it is clear there that most of the error is due to just a few structure components, in line with the analysis of Jepson and Heeger [8] and Maybank [11, 12] for optical flow. Note that two frame algorithms fail completely on this sequence [14].

We have conducted synthetic experiments to verify the stability of our results against different random noise. We first generated an exact image sequence corresponding to the ground truth for the structure and motion for the PUMA sequence. In each trial, uniform noise of size ± 1 pixel was added independently to each feature point in each image and the structure was recovered using our algorithm. Explicit comparison of the original PUMA images to the exact images generated from the ground truth shows that the noise for the real sequence is somewhat less than this. The result of 250 trials gave comparable results to those obtained on the real image sequence. A single failure of the algorithm occurred which will be investigated in future work.

3 Constant Translation Direction

Often a navigating robot does not change direction quickly. In this section we describe an algorithm that takes advantage of this fact: it recovers structure from an image sequence assuming that the translation direction remains approximately constant over the sequence. Experimental results are presented for a real image sequence—the Martin–Marietta Rocket Field sequence—for which the translation direction does in fact change.

The constant direction assumption is more powerful than it may at first appear. It applies not to the interframe translation (between successive image frames) but to the cumulative translation with respect to a base frame. Assuming that the camera does move with an overall trend in some direction and that the interframe translations are relatively small, then the larger cumulative translations will cluster around the trend direction. Since the larger translations are the most important in determining the structure, the constant direction assumption is likely to be a useful one for this common situation. Note that we constrain just the direction of the translations.

Under the constant direction assumption, the translational part of (1) becomes

$$\mathbf{d}_i^h \approx \frac{\lambda^h \xi_i(\hat{T}_z \mathbf{b}_i - \hat{\mathbf{T}}_2)}{1 - \lambda^h \hat{T}_z \xi_i}, \quad (14)$$

where $\hat{\mathbf{T}} \equiv (\hat{\mathbf{T}}_2, \hat{T}_z)^t \equiv (\hat{T}_x, \hat{T}_y, \hat{T}_z)$ is the fixed translation direction (a unit vector), and λ^h is the scale of the translation in the h-th frame.

We determine the translation direction first, since it can be recovered accurately, and then use this to recover the structure. Multiply (14) for each feature point by $\chi_i \equiv \hat{T}_z(-y_i, x_i)^t - (-\hat{T}_y, \hat{T}_x)^t$. This annihilates the translational term in the displacement, and the remaining rotational terms are of the form

$$\alpha_1 + \alpha_2 x_i + \alpha_3 y_i + \alpha_4 x_i^2 + \alpha_5 x_i y_i + \alpha_6 y_i^2, \quad (15)$$

where the α's are independent of the feature point index i.

Define as before a $((M-6) \times M)$ Household matrix H_V that annihilates the 6 vectors $\{1\}$, $\{x\}$, $\{y\}$, $\{x^2\}$, $\{xy\}$, $\{y^2\}$. Multiplying by H_V^t produces

$$\sum_i [\mathbf{d} \cdot \chi]_i^h [\mathrm{H}_V^t]_j^i \approx 0. \quad (16)$$

Here, as previously, we have assumed the translation scale small compared to the scale of the depths; this restriction is not stringent since it is applied only to terms that are already small by virtue of an factor of ω.

(16) is a linear equation for the translation

$$0 \approx \sum_i [(\mathbf{d}_y, -\mathbf{d}_x)^t \cdot (\hat{T}_z \mathbf{b}_i - \hat{\mathbf{T}}_2)]_i^h [H_V^t]_{ij} \quad (17)$$

which can be solved by least squares. However, prior to doing so we first multiply as before on the left by the matrix $\mathrm{C}_H^{-1/2}$ to better approximate a MLE.

3.1 Maximum Likelihood Estimation

Assuming constant direction of translation, the structure depends critically on accurately determining this direction. Thus we attempt to improve the linear solution for the translation direction obtained above.

Let the residual matrix

$$[\mathrm{M_R}] \equiv [(\mathbf{d}_y, -\mathbf{d}_x)^t \cdot (\hat{T}_z \mathbf{b}_i - \hat{\mathbf{T}}_2)][H_V]^t. \quad (18)$$

To construct a MLE we must compute the covariance of this matrix. To first order, and neglecting terms that are suppressed relatively by factors of order $o(\xi|T|)$ or $o(|\omega|)$,

$$\left\langle [\mathrm{M_R}]_j^h [\mathrm{M_R}]_{j'}^{h'} \right\rangle \approx \sigma^2(\delta_{hh'} + 1_{hh'})[C_I C_I^t]_{jj'} \quad (19)$$

where σ is the standard deviation of the image noise, and C_I is a $(M-6) \times 2M$ matrix

$$[C_I]_{j;ia} \equiv [\mathrm{H}_V]_{ji}(\hat{T}_z \mathbf{b}_{ia} - \hat{\mathbf{T}}_{2a}) \quad (20)$$

where a selects the x or y coordinate.

The maximum likelihood estimate corresonds to minimizing

$$\mathrm{Tr}(\mathrm{C}_H^{-1/2} \mathrm{M_R} [C_I C_I^t]^{-1} [\mathrm{M_R}]^t \mathrm{C}_H^{-1/2}), \quad (21)$$

which is quadratic in $\mathbf{T}$ both in numerator and "denominator."

We use a simple iterative procedure to find the minimum of (21). At this stage our approach ceases to be purely linear—but the consequent advantage of significantly improved accuracy of structure recovery outweighs the minor loss in speed. Since the algorithm of the previous section already produces an excellent starting guess for the translation, convergence requires few iterations.

4 Experiments: Rocket Sequence

For this sequence, with the prior knowledge that rotations were small, we did not unrotate the images prior to applying our motion/structure recovery algorithms.

The result of our initial algorithm estimating the translation direction was $\mathbf{T} = (0.0618, -0.2213, 0.9733)^t$, compared with the direction of the average ground truth translation $(-0.0476, -0.2211, 0.9741)^t$. The angular error is $6°$.

The approximate MLE iteration reduced the angular error to $2.2°$ giving a translation direction of $\hat{\mathbf{T}} = (-0.0565 \;\; -0.1842 \;\; 0.9813)^t$; the time required for convergence was less than 2 seconds in MATLAB on a DecStation 5000. For comparison, we implemented a true MLE estimate of the translation direction, implemented using a standard LM algorithm. The error function minimized was the actual image error between projected 3D points and their measured image positions; the difference from the brute force MLE of the previous section was that the

Table 2: Comparison to Ground Truth of Scaled Depths.

MLE	BFCT	True	LIN
23.6989	23.6821	24.6203	25.2456
33.3382	33.3552	33.7936	33.2820
22.7341	22.7046	24.2459	25.1189
16.8783	16.8902	17.7859	20.9122
26.0059	25.9719	27.8879	28.9933
23.2953	23.2948	24.7700	26.4058
48.1055	47.5408	60.5430	87.1700
29.3514	29.2206	32.3651	30.9896
46.4439	46.5271	43.8711	42.6899
40.8853	41.2243	40.1131	33.5594
50.5292	50.5373	47.7790	46.3051

translation was constrained to have a constant direction. This procedure yielded an angular error of 1.71°, comparable to the result of our algorithm. The depth estimates obtained by this brute force approach (BFCT), scaled to the ground truth, are shown in Table 2. Here, and in the results reported below, the scaling to ground truth is done using all but the 7-th and 10-th points; since these points are both distant and near the FOE (as can be determined a posteriori) their depths are expected to be difficult to recover. Omitting them in the scaling gives a better picture of the accuracy with which the depths of other points are recovered. For the remaining 9 points, the average magnitude of the depth error is 1.8 feet, and the average percentage error is 5.7% for the brute force algorithm.

Using our estimate $\hat{\mathbf{T}}$ of the translation direction, the structure was computed by a linear algorithm (LIN) similar to that for the $T_z \approx 0$ case. Table 2 shows the results scaled to compare with the ground truth depths. For the nine points described above, the average magnitude of the depth error is 1.3 feet and the average percent error is 5.1%, comparable to the results of the brute force MLE. However, the superiority of the latter algorithm is shown in the accuracy with which it recovers the depths of the difficult 7-th and 10-th points—for these two points, precise determination of the FOE is crucial.

Finally, we have done a full MLE estimate as for the PUMA sequence, making no assumption on the translation—this algorithm therefore should give the most accurate results. However, its results are comparable to those obtained previously by our algorithm and by the constant direction BFCT algorithm. The scaled depths are shown in Table 2. The average magnitude of the depth error is 1.7 feet and the average percentage error is 5.6%. Though this algorithm was started from the ground truth, about 4 minutes were required for convergence.

It is clear from these experiments that the assumption of constant translation direction is an appropriate one for the Rocket sequence, and that our essentially linear algorithm gives comparable results to full MLE estimates, while requiring only a few seconds of computation time. To check these results, we ran a series of synthetic experiments. In one set, we generated an image sequence corresponding exactly to the grouth truth for the rocket sequence. We then ran 200 trials of our algorithm, in which at each trial approximately 1 pixel random noise (with a uniform distribution) was added to each image point. For each trial, the average magnitude of the depth error for the selected 9 points was measured. The average of this quantity over the 200 trials was 1.7 feet. The average of the average percentage error was 6%. The average angular error in the translation direction was 1.5°. These results are in line with those for the real sequence.

In a second set of sythetic experiments, both the motion and structure were varied randomly at each trial. 22 points and 9 image frames were used in all trials. The "ground truth" depths of points were varied randomly with a uniform distribution over the range from 20 to 60 feet, while the other coordinates were varied uniformly in the range of -15 to +15 feet. The motion was predominantly forward, with random small rotations. Finally, uniform image noise was added independently to each image feature corresponding to about 1 pixel noise for the rocket sequence. Over 200 trials, the average of the average magnitude of the depth error was 1.6 feet with one failure of the algorithm at 31 feet. The average angular error in the translation direction was 3.8°, again with one outlier corresponding to that for the depth error.

5 Recursive Implementation

We sketch a recursive implementation of our approach. Consider the algorithm for $T_z \sim 0$ and for simplicity assume that there is zero rotation. Then the measurement matrix in (3) is $\mathrm{D} = [-\bar{\mathbf{T}}_x \bar{\xi}^t, -\bar{\mathbf{T}}_y \bar{\xi}^t]$. Again for simplicity, consider $\mathrm{D}_x \equiv [-\bar{\mathbf{T}}_x \bar{\xi}^t]$, the x-components of the measured image displacements. The task of determining the inverse depths from D_x is simply that of determining the leading eigenvector of

$$M_x^h \equiv [D_x]^t D_x = \sum_h [D_x^h]^t D_x^h, \qquad (22)$$

where D_x^h is the h-th row of D_x. A recursive algorithm then consists of: 1) as each new image is acquired, let $M_x^{h+1} = M_x^h + [D_x^{h+1}]^t D_x^{h+1}$, where D_x^{h+1} represents the new image measurements; 2) compute the leading eigenvector of the updated matrix M_x^h. This idea clearly extends to the case where there is rotation.

Our algorithm for the constant translation direction case first calculates the translation direction, using all available information, and then uses this to compute the structure. Thus as it stands it cannot be implemented as a recursive algorithm for structure. However, it is easy to implement as a recursive algorithm for estimating the direction of translation. It is also possible to derive a variant of our approach that can be implemented as a recursive structure estimation algorithm.

References

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

[2] R. Dutta, R. Manmatha, L.R. Williams, and E.M. Riseman, "A data set for quantitative motion analysis," *CVPR*, 159-164, 1989.

[3] G. Golub and C. F. Van Loan, *Matrix Computations*, John Hopkins Press, Baltimore, Maryland,1983.

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

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

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

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

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

[9] R. Kumar and A.R. Hanson, "Sensitivity of the Pose Refinement Problem to Accurate Estimation of Camera Parameters," *ICCV*, 365-369, 1990.

[10] R. Kumar and A.R. Hanson, "Pose Refinement: Application to Model Extension and Sensitivity to Camera Parameters," *IUW*, 660-669, 1990.

[11] S. Maybank, *Theory of Reconstruction from Image Motion*, Springer, Berlin, 1992.

[12] S. Maybank, "A Theoretical Study of Optical Flow," Doctoral Dissertation, University of London, 1987.

[13] H. S. Sawhney, J. Oliensis, and A. R. Hanson, "Description and Reconstruction from Image Trajectories of Rotational Motion", *ICCV*, 494-498, 1990.

[14] H.S. Sawhney and A.R. Hanson, "Comparative results of some motion algorithms on real image sequences," *IUW*, 307-313, 1990.

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

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

[17] C. Tomasi and T. Kanade, "Factoring Image Sequences into Shape and Motion," *Motion Workshop*, Princeton, 21-28, 1991.