Refining 3D Reconstructions: A Theoretical and Experimental Study of the Effect of Cross–Correlations*

J. Inigo Thomas
GRASP Laboratory
Department of Computer and Information Science
University of Pennsylvania
313C 3401 Walnut, Philadelphia, PA 19104-6228

Allen Hanson John Oliensis
Computer Science Department
University of Massachusetts at Amherst
Amherst, MA 01003

email: inigo@grip.cis.upenn.edu
phone: (215) 898 9247 fax: (215) 573 2048

All correspondence should be sent to the first author at the above listed address.

*This work was supported by DARPA (via TACOM) under contract number DAAE07-91-C-R035 and by NSF under grant number CDA-8922572. The first author was at the University of Massachusetts when this work was done.

Abstract

In robot navigation a model of the environment needs to be reconstructed for various applications, including path planning, obstacle avoidance and determining where the robot is located. Traditionally, the model was acquired using two images (two–frame Structure from Motion) but the acquired models were unreliable and inaccurate. Recently, research has shifted to using several frames (multi–frame Structure from Motion) instead of just two frames. However, almost none of the reported multi–frame algorithms has produced accurate and stable reconstructions for general robot motion. The main reason seems to be that the primary source of error in the reconstruction – the error in the underlying motion – has been mostly ignored. Intuitively, if a reconstruction of the scene is made up of points, this motion error affects each reconstructed point in a systematic way. For example, if the translation of the robot is erroneous in a certain direction, all the reconstructed points would be shifted along the same direction. The contributions of this paper include mathematically isolating the effect of the motion error (as correlations in the structure error) and showing theoretically that these correlations can improve existing multi–frame Structure from Motion techniques. Finally it is shown that new experimental results and previously reported work confirm the theoretical predictions.

Key words: Structure from Motion (SFM), Multi–frame SFM, Modeling Motion Error, Kalman Filtering, 3D Scene Reconstruction, Automatic Model Acquisition.

1 Introduction

It is crucial in robot navigation that the robot has an internal model of its environment. The model of the environment can be used in many ways, including path planning, obstacle avoidance, and determining where the robot is located. In order to reconstruct the environment using a camera, at least two different views are required. One method of obtaining the two views is to use a stereo camera pair. However, an inherent problem with stereo is the limited distance between the cameras mounted on a robot; the smaller the distance between the cameras, the less accurate the reconstruction of the environment. The problem of the fixed, short distance between the cameras can be solved if instead of a fixed arrangement, one camera is moved from one point to another. This is the case of *extended stereo*, or Structure From Motion (SFM). [1] Apart from the fact that almost any pair of views of the scene can be obtained – in effect by manipulating the distance between the cameras – SFM and stereo are logically equivalent. Due to its advantage, that the distance between two viewpoints is not constrained, we have chosen the SFM paradigm to reconstruct the environment and argue that the best way to implement such a paradigm crucially involves accounting for the motion error in a recursive multi–frame algorithm.

Once Gibson [13] had noted that observer motion and distance to objects correlates with retinal image changes (also called *image flow* or *optical flow*), several seminal efforts attempted to discover whether given the change in the retinal image it is possible to recover the observer motion and location of objects. Initially, most of this research concentrated around recovering motion and object location based on the image flow obtained from just two retinal images (*two–frame SFM*) – the equivalent of stereo in motion (Tsai and Huang [40], Longuet–Higgins and Prazdny [19], Bruss and Horn [5], Prazdny [23], Adiv [1] among others).

Two–frame SFM has failed to produce accurate reconstructions because of various problems.

[1] *Structure* stands for scene reconstruction which consists of a representation of the scene with e.g. 3D points.

The first problem is due to *inherent ambiguity* in determining the motion, even in the absence of image noise. For example, Faugeras and Maybank [10] have shown that when motion is computed from 5 points, there are as many as 10 possible solutions for the motion. Horn [17] points out that ambiguities are especially prevalent when image points lie on a hyperboloid of one sheet or its degeneracies. Adiv [2] claims that there is a large number of incorrect motion solutions that induce flow fields similar to the correct one. As explained by Spetsakis and Aloimonos [29], another inherent ambiguity seems to be the preference for movement straight along the viewing direction, at least for small fields of view.

The problem of ambiguity is compounded by the presence of *noise* in the image, low image digitization, point mismatching, feature mismatching, and/or erroneous camera calibration. Adiv [2] adds to this list other factors that increase the error, such as a small field of vision, faraway objects, a small absolute translation step, and sparse flow field.

Furthermore, it appears that the limits of the two–frame approach have been reached. Weng, Huang and Ahuja [44] claim (based on simulations) that the performance of their two–frame motion algorithm has reached the theoretically possible Cramer-Rao [7] [24] lower bounds of optimal estimation of the motion parameters. On the other hand, Dutta and Snyder [9] argue that even small *rotation* errors (which all two–frame motion algorithms suffer from, including the near optimal performance algorithms) cause a large error in structure. Assuming a realistic situation [2] they show that most points in the image can only be reconstructed to within an error of 10 to 20%. This means that even algorithms that are shown to be optimal in estimating motion cannot produce a useful model of the environment, as pointed out by Spetsakis and Aloimonos [31].

Due to the problems in two–frame SFM, the obvious solution has been to use more than two frames to reconstruct the environment. Although it is theoretically conceivable that using

[2]The camera is assumed to move directly ahead. The movement is 1/10th the distance to the point being reconstructed. The pixel error in the image coordinates is 1 pixel in an image of size 256×256.

enough different views should make it possible to achieve any required accuracy, stable and reliable 3D reconstructions have been difficult to achieve, particularly in recursive multiframe SFM (MFSFM) work [14] [15] [28] [3] [32] [8].

Based on an algorithm presented by Thomas and Oliensis [36], we show that in order to obtain a stable and reliable reconstruction of the environment (for general motion), the effect of the interframe motion error has to be taken into account; we argue that ignoring this component has resulted in the failure of previous MFSFM algorithms. The theoretical and experimental evidence for the crucial role of motion error (in MFSFM) is the main contribution of this paper.

2 Problems in MFSFM

Although it is reasonable to move from a two–frame to a multi–frame paradigm, using multiple images introduces a new set of problems. The problems vary depending on whether the algorithms are batch or recursive. Batch methods attempt to reconstruct the 3D scene assuming that *all* images from every camera position are available before processing is begun ([26] [4] [18] [11] [30] [33] [39]). Recursive methods, on the other hand, assume that all past images are *not* available; rather, they assume that only the most recent reconstruction of the scene is available along with an estimate of the error in this reconstruction [14] [15] [28] [3] [32] [8].

Since batch methods use all the available information in one shot they should potentially produce the best results. However, the main problem with batch methods is that there is a large number of variables (especially for all the interframe camera motions) when a robot moves in a general environment. If the scene is reconstructed based on $m+1$ pictures (from m arbitrary robot moves) and represented using n features (say, n 3D points), then the most general batch method involves $6m-1+3n$ variables [3]. The number of points, n, determines the granularity of the scene reconstruction. If we assume for the sake of comparison across algorithms that the

[3] Each robot motion involves 6 variables (3 for translation and 3 for rotation) and each 3D point involves 3 variables (x,y,z). The scale ambiguity [40] decreases the total number of unknown variables by 1.

granularity is fixed (i.e., n is fixed), then the batch methods have to deal with $6m-1$ variables in the most general case. This is a very large number of variables (59 for 10 movements) with a complicated non–linear error function.

In an attempt to make the problem manageable, some batch methods have restricted general camera motion to simplified motion ([26] [4] [18] [11] [33]), under the assumption that any deviation from the assumed motion could be dealt with as system *noise*. However, the problem with such a restriction is that it becomes useless in a realistic situation and deviations from the assumed simplistic motion models cannot be usually dealt with as noise. Although the batch multi–frame algorithm due to Spetsakis and Aloimonos [30] is for general motion, the performance of the algorithm using real image sequences is not reported. The most complete experimental result for an algorithm reported for unrestricted motion is that of Tomasi [39]. However, in order to restrict the complexity of the search space of motion variables, Tomasi had to make the assumption that the camera model is an orthographic projection, i.e. the light rays striking the camera image are parallel. This assumption restricts the usability of the method to situations involving objects reasonably far away from the camera, such as objects viewed from an airplane; its applicability to obstacle avoidance – or any situation where the orthographic model of the camera is invalid – is yet to be studied. The accuracy of Tomasi's reconstruction of shape (2.4%) is comparable to the accuracy of the reconstructions reported in this paper (cf. Section 4).

Unlike batch methods, recursive MFSFM algorithms need not impose such restraints on the interframe camera motion or the camera model (although early research on recursive MFSFM typically was constrained; cf. discussion in Section 4). Recursive methods need to determine only five unknown variables (of interframe motion) at any point, as opposed to the $6m-1$ unknown variables in batch methods.

MFSFM algorithms are also more practical for robot navigation applications since neither time nor storage is lost waiting until enough frames have been acquired. However, in order to recursively refine the 3D structure, a reliable estimate of the *error* in the 3D structure is required. If the estimate of the error is unreliable, this results in random behavior or possibly systematically erroneous behavior. One of the biggest problems in MFSFM is that it is difficult to represent the error in the structure reliably.

One representation of the reconstruction error is a complete covariance matrix.[4] That is, if the scene is reconstructed by n 3D points, then the reconstruction error is represented by a covariance matrix of size $9n^2$. This covariance matrix is difficult to compute, expensive to store, and computationally complex to manipulate. Presumably for these reasons, almost all of the work in recursive MFSFM has only used a portion of the covariance matrix, with poor results. Thomas [38] claims that every entry of the covariance matrix is meaningful; neglecting any entry amounts to a bad approximation of the actual reconstruction error (for general camera motion). A simplistic explanation is as follows. The main source of error in all structure from motion algorithms is the error in the estimated camera motion. The motion error affects all the 3D coordinates of the reconstruction in a systematic way; i.e., the errors in all the 3D coordinates are correlated. For example, if the translation component of the camera motion is erroneous, each 3D coordinate would be displaced along the same direction. Since every element of the covariance matrix represents the correlation of the error between pairs of 3D points, neglecting non–zero elements in the covariance matrix may have dire consequences. The following section is a theoretical analysis of the meaning and the role of cross-correlations in recursive MFSFM algorithms.

[4]If the underlying error is Gaussian in nature, with zero mean, then the covariance is enough to capture the entire nature of the error. Even when the error in the reconstruction is a non–linear function of a Gaussian noise, it has been noticed experimentally that a Gaussian approximation is a reasonable assumption [34].

3 Theoretical Motivation for Using Cross–correlations

If $\mathbf{P}$ is the entire reconstruction, made up of n 3D points ($\mathbf{P}_i,\ i = 1 \ldots n$) then $\mathbf{P}$ can be written as a $3n \times 1$ vector,

$$\mathbf{P} = \begin{pmatrix} \mathbf{P}_1 \\ \mathbf{P}_2 \\ \cdots \\ \mathbf{P}_n \end{pmatrix} \tag{1}$$

Since each $\mathbf{P}_i$ is obtained from a two–frame algorithm effectively by triangulation, it has two sources of error. The first source is the error in the interframe motion, or the relative orientation of the cameras. The second source of error is the noise in the image coordinates. A reasonable approximation of the total error in $\mathbf{P}_i$ is to express it (using first order terms) as the sum of the error due to the interframe motion and the error due to the image coordinates; the error in $\mathbf{P}_i$ is then

$$d\mathbf{P}_i = \frac{\delta \mathbf{P}_i}{\delta \mathbf{W}} d\mathbf{W} + \frac{\delta \mathbf{P}_i}{\delta \mathbf{V}_i} d\mathbf{V}_i \tag{2}$$

where $\mathbf{W}$ represents the interframe motion and $\mathbf{V}_i$ represents the image coordinates of the point; $d\mathbf{W}$ and $d\mathbf{V}_i$ represent the respective errors.

When the error in $\mathbf{P}$ is represented as a covariance matrix the elements of this matrix are given by the following equation :

$$COV(d\mathbf{P}\ d\mathbf{P}^T) = \begin{pmatrix} E(d\mathbf{P}_1\ d\mathbf{P}_1) & E(d\mathbf{P}_1\ d\mathbf{P}_2) & \cdots & E(d\mathbf{P}_1\ d\mathbf{P}_n) \\ E(d\mathbf{P}_2\ d\mathbf{P}_1) & E(d\mathbf{P}_2\ d\mathbf{P}_2) & \cdots & E(d\mathbf{P}_2\ d\mathbf{P}_n) \\ & & \cdots & \\ E(d\mathbf{P}_n\ d\mathbf{P}_1) & E(d\mathbf{P}_n\ d\mathbf{P}_2) & \cdots & E(d\mathbf{P}_n\ d\mathbf{P}_n) \end{pmatrix} \tag{3}$$

where $E(x)$ denotes the expected value of x.

In this theoretical analysis, in order to bring out the meaning and role of the cross–correlation terms clearly, we will assume that we have a reconstruction consisting of just two points.

3.1 The Meaning of Cross–correlations: The Two Point Case

In this case, the covariance matrix is reduced to

$$COV(d\mathbf{P}\ d\mathbf{P}^T) = \begin{pmatrix} E(d\mathbf{P}_1\ d\mathbf{P}_1) & E(d\mathbf{P}_1\ d\mathbf{P}_2) \\ E(d\mathbf{P}_2\ d\mathbf{P}_1) & E(d\mathbf{P}_2\ d\mathbf{P}_2) \end{pmatrix} \tag{4}$$

This covariance matrix has four correlation terms, two of which are equivalent ($E(d\mathbf{P}_1\ d\mathbf{P}_2)$ and $E(d\mathbf{P}_2\ d\mathbf{P}_1)$). [5] The other two ($E(d\mathbf{P}_1\ d\mathbf{P}_1)$ and $E(d\mathbf{P}_2\ d\mathbf{P}_2)$) are the covariance of the error in $\mathbf{P}_1$ and $\mathbf{P}_2$; these are typically assumed to represent the complete error. However, here we will concentrate on the *cross–correlation term*, $E(d\mathbf{P}_1\ d\mathbf{P}_2)$.

Using Equation 2 the cross–correlation term can be expanded as in Equation 5:

$$E(d\mathbf{P}_1\ d\mathbf{P}_2) = E[(\frac{\delta\mathbf{P}_1}{\delta\mathbf{W}}d\mathbf{W} + \frac{\delta\mathbf{P}_1}{\delta\mathbf{V}_1}d\mathbf{V}_1)\ (\frac{\delta\mathbf{P}_2}{\delta\mathbf{W}}d\mathbf{W} + \frac{\delta\mathbf{P}_2}{\delta\mathbf{V}_2}d\mathbf{V}_2)^T] \tag{5}$$

Since it is realistic to assume that any two arbitrary image coordinates (of chosen points) are corrupted by independent noise, [6] i.e.

$$E(d\mathbf{V}_1\ d\mathbf{V}_2) = 0 \tag{6}$$

one of the terms in the expansion of Equation 5 will vanish. The resultant expansion is given in Equation 7:

$$E(d\mathbf{P}_1\ d\mathbf{P}_2) = \frac{\delta\mathbf{P}_1}{\delta\mathbf{W}}E(d\mathbf{W}d\mathbf{W}^T)\frac{\delta\mathbf{P}_2}{\delta\mathbf{W}} + \frac{\delta\mathbf{P}_1}{\delta\mathbf{W}}E(d\mathbf{W}d\mathbf{V}_2^T)\frac{\delta\mathbf{P}_2}{\delta\mathbf{V}_2} + \frac{\delta\mathbf{P}_1}{\delta\mathbf{V}_1}E(d\mathbf{V}_1^T d\mathbf{W})\frac{\delta\mathbf{P}_2}{\delta\mathbf{W}} \tag{7}$$

Given Equation 7, the only way the cross–correlation term will end up being zero is when the three terms fortuitously cancel; in all other cases the cross–correlation term has an effect

[5] Both $E(d\mathbf{P}_1\ d\mathbf{P}_2)$ and $E(d\mathbf{P}_2\ d\mathbf{P}_1)$ represent the cross–correlation of the error in $\mathbf{P}_1$ with the error in $\mathbf{P}_2$ and hence are identical.

[6] If points are tracked separately, tracking algorithms will generally not introduce correlated errors between any two image points.

on the performance of the recursive MFSFM algorithm. Furthermore, the situations in which the three terms cancel each other out are most likely rare.

For the sake of exposition let us assume that the coordinates of the two points have changed considerably between the two images, so that the image displacements are large. Therefore, a small error in measuring the displacements (which corresponds to a small error in V) has little effect on the error in the motion, $d\mathbf{W}$; i.e.

$$E(d\mathbf{W}d\mathbf{V}_i^T) \approx 0 \;\; i = 1,2 \tag{8}$$

For this particular case, the expansion of Equation 7 is :

$$E(d\mathbf{P}_1 \; d\mathbf{P}_2) = \frac{\delta \mathbf{P}_1}{\delta \mathbf{W}} COV(d\mathbf{W} \;\; d\mathbf{W}) \frac{\delta \mathbf{P}_2}{\delta \mathbf{W}} \tag{9}$$

Equation 9 shows that the cross-correlation is directly proportional to the motion error, represented as the covariance of the error in the motion ($d\mathbf{W}$). If Equation 8 does not hold the situation is more complicated: the cross–correlation is influenced not only by the motion error but also (indirectly) by the error in the image coordinates. In either case, the cross–correlation term is closely related to the motion error.

3.2 The Effect of Cross–correlations in Kalman Filtering

In this section the analysis is extended to study the effect of cross–correlations on refining reconstructions using the Kalman filter.

The goal of the Kalman filter is to optimally fuse the reconstructions over time and obtain the best reconstruction (by limiting the reconstruction error). If we assume that the noise in every new reconstruction ($\mathbf{P}(t)$ at time t) is Gaussian (cf. Section 3.1), then the optimal fused reconstruction is the sum of the individual reconstructions weighted by the inverse of their covariances; this makes intuitive sense since if a particular covariance is large – suggesting a large error in the reconstruction – that reconstruction should be given less weight. Given this the optimal fused reconstruction ($\tilde{\mathbf{P}}$) at time t is as follows (i.e. standard Kalman filtering [12]) :[7]

$$\tilde{\mathbf{P}}(t) = N \sum^{t} COV(\mathbf{P}(t))^{-1}\, \mathbf{P}(t) \tag{10}$$

In order to determine the exact contribution of a single reconstruction ($COV(\mathbf{P})^{-1}$ $\mathbf{P}$ or $Weighted\,\mathbf{P}$) at any time (t) the covariance can be expanded using Equation 4 (and assuming Equation 8 is valid) in the following way:

$$COV(P) = \begin{pmatrix} S_1 + M_{11} & M_{12} \\ M_{21} & S_2 + M_{22} \end{pmatrix} \tag{11}$$

where

$$S_i = \frac{\delta \mathbf{P}_i}{\delta \mathbf{V}_i} COV(d\mathbf{V}_i\ d\mathbf{V}_i) \frac{\delta \mathbf{P}_i}{\delta \mathbf{V}_i} \tag{12}$$

and

$$M_{ij} = \frac{\delta \mathbf{P}_i}{\delta \mathbf{W}} COV(d\mathbf{W}\ d\mathbf{W}) \frac{\delta \mathbf{P}_i}{\delta \mathbf{W}} \tag{13}$$

S_i represents the error in the 3D coordinates due to the error in the image coordinates ($d\mathbf{V}$) assuming that the motion is perfectly known; M_{ij} represents the error in the 3D coordinates due to the error in the interframe camera motion ($d\mathbf{W}$) assuming that the image coordinates are perfectly known.

$Weighted$ $\mathbf{P}$ can now be written as

$$Weighted\ \mathbf{P} \quad = \quad \begin{pmatrix} S_1 + M_{11} & M_{12} \\ M_{21} & S_2 + M_{22} \end{pmatrix}^{-1} \begin{pmatrix} \mathbf{P}_1 \\ \mathbf{P}_2 \end{pmatrix} \tag{14}$$

[7] N (in Equation 10) is a normalizing term which is irrelevant for this analysis.

Equation 14 can be expanded (after Bar-Shalom and Fortmann [27]) to obtain:

$$Weighted\ \mathbf{P} = \begin{pmatrix} (S_{11} + M_{11} - M_{12}(S_{22} + M_{22})^{-1} M_{12}^T)^{-1} (\mathbf{P}_1 - M_{12}(S_{22} + M_{22})^{-1} \mathbf{P}_2) \\ (S_{22} + M_{22} - M_{21}(S_{11} + M_{11})^{-1} M_{21}^T)^{-1} (\mathbf{P}_2 - M_{21}(S_{11} + M_{11})^{-1} \mathbf{P}_1) \end{pmatrix} \tag{15}$$

Let us now concentrate on the effect of the cross–correlation on a single optimally fused 3D coordinate ($\tilde{\mathbf{P}}_1$); all of the relevant information is contained in the first row of Equation 15. The second term (in the first row) can be thought of as a $Corrected\, \mathbf{P}_1$:

$$Corrected\, \mathbf{P}_1 = \mathbf{P}_1 - M_{12}(S_{22} + M_{22})^{-1} \mathbf{P}_2 \tag{16}$$

If there is no error in the motion – i.e. M_{12} is zero – the $Corrected\, \mathbf{P}_1$ is identical to $\mathbf{P}_1$. However, since this is generally not true in practice, the value of $\mathbf{P}_2$ has a corrective effect on $\mathbf{P}_1$. The magnitude of the correction depends on the size of the cross-correlation M_{12}. Since we have shown that the cross–correlation captures the motion error (cf. Section 3.1), the magnitude of the correction depends on the (shared) motion error that corrupts both $\mathbf{P}_1$ *and* $\mathbf{P}_2$.

The covariance of $Corrected\, \mathbf{P}_1$ is

$$COV(Corrected\, \mathbf{P}_1) = E([\mathbf{P}_1 - M_{12}(S_{22} + M_{22})^{-1} \mathbf{P}_2][\mathbf{P}_1 - M_{12}(S_{22} + M_{22})^{-1} \mathbf{P}_2]^T) \tag{17}$$

Again, this can be simplified to obtain:

$$COV(Corrected\, \mathbf{P}_1) = S_{11} + M_{11} - M_{12}(S_{22} + M_{22})^{-1} M_{12}^T \tag{18}$$

As stipulated by Kalman filtering, any contribution (towards the fused optimal estimate) has to be weighted by the inverse of its covariance. Thus we expect that $Corrected\, \mathbf{P}_1$ (Equation 16) should be weighted by the inverse of its covariance. Since the right–hand side of Equation 18 turns out to be equal to the first term (in the first row) of Equation 15 above, this is *exactly* the case.

This analysis reveals that the cross–correlation terms are important. If the interframe motion error is large, then the cross–correlation terms become significant and play a crucial role. Since in SFM the motion error is typically large [9] we predict that without cross–correlations the benefits of Kalman filtering are lost, i.e. the fused reconstruction would be neither stable nor accurate. In the next section we present experimental evidence to this effect.

4 Experimental Data

The previously reported MFSFM algorithms conform to the prediction of the last section. Shigang, Tsuji, and Imai [28] use only n terms to approximate their error (i.e., when reconstructing n 3D points, their covariance matrix has n elements rather than the full $9n^2$ elements). They allow the camera to move freely in a plane, and their reconstruction error is 15% even with as many as 40 images. In this context, not much can be said of the recursive MFSFM algorithm by Heel [14] [15] or of the algorithm by Ando [3], both of whom use only n elements; the former reports only qualitative reconstructions and the latter reports only simulations. [8]

The next category of approximations involve using $9n$ elements to approximate the $9n^2$ covariance matrix. Stephens et al. [32] report reconstructions within 1% error *for 1 point* after 50 frames in the case of motion straight ahead. Cui, Weng and Cohen [8] also use $9n$ elements to approximate the full covariance matrix and apply the algorithm for the case of general camera motion. The reported accuracy of the reconstruction (from a real image sequence) fluctuates randomly. Since no comparison with the ground truth is reported it is unclear as to how well this algorithm really does.

The algorithm developed by Thomas and Oliensis [35] [22] [36] is the only recursive MFSFM algorithm (for general motion) that uses the full covariance matrix with $9n^2$ elements. Apart

[8] Matthies et. al. [21], [20] also use only n elements to represent the reconstruction error; however, they constrain the camera to move exactly parallel to a fixed line, and they assume that *the exact camera motion is known*. Since camera motion is known (an impractical assumption) the cross–correlations should not play a role. Indeed Matthies et. al. obtain a reconstruction of 0.5% error using 11 images.

from using cross–correlations, their algorithm is similar to previous recursive (Kalman filter) MFSFM algorithms. Highly accurate reconstructions (as accurate as the ground truth) have already been reported by Thomas and Oliensis [36] for image sequences with no constraints on the robot camera motion. Here, their algorithm is used to test for the effect of cross–correlations in real image sequences by comparing results from the same algorithm with and without cross–correlations. Such a comparison has not been previously done; it will be presented in the following section for two real image sequences. [9]

4.1 Experiment I: Reconstruction of A Rotating Box

4.1.1 Description of the Input

A sequence of images was taken of a box rotated by a robot arm; the camera was mounted on a stationary tripod. Each rotation of the box was approximately 4 degrees around its vertical axis. The camera used to obtain the images was a Sony black and white CCD camera. The parameters of this camera are given in Table 1. Figure 1 shows the first and the last images

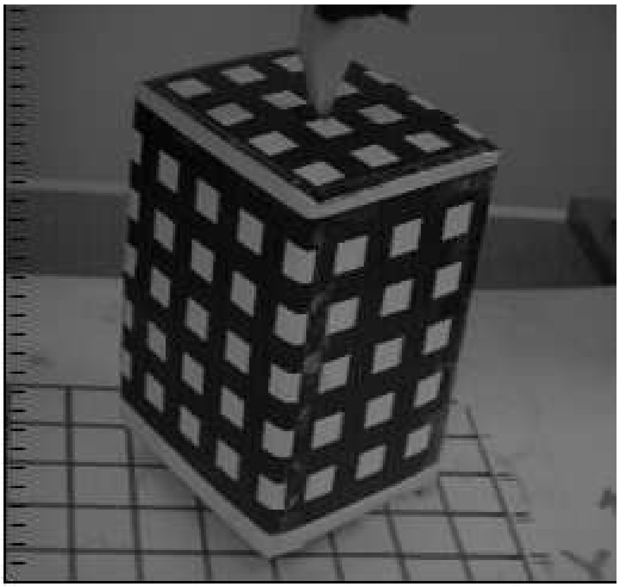
focal length	fov X	fov Y	Size
6 cm	$23.4059°$	$22.387°$	256×242

Table 1: **Camera Parameters.** The camera parameters of the Sony black and white camera used in the rotating box sequence.

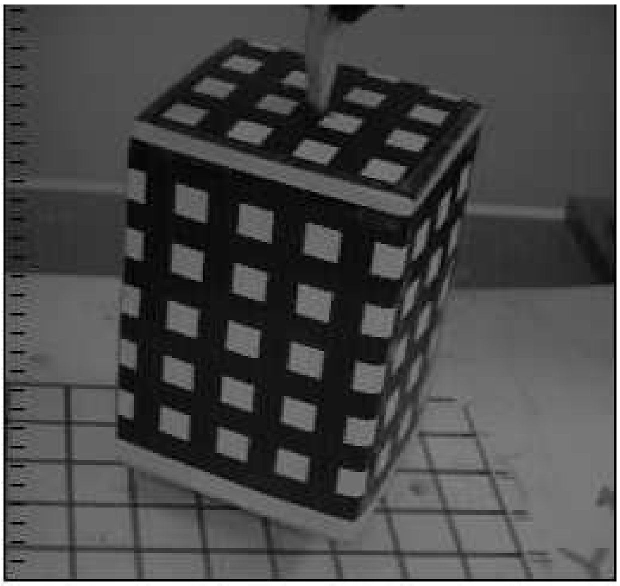
of the nine image sequence. Ground truth measurements of the 3D coordinates of the selected points on the box were done by Sawhney [25]. The measured 3D coordinates of the points at the final position of the box (in the camera coordinate system) are shown in Table 2. The accuracy of these measurements is $\pm 1.5mm$.[10]

Since the algorithm used here is a point–based algorithm (i.e. it recovers 3D coordinates

[9] Thanks to Harpreet Sawhney and Rakesh Kumar for these sequences and the ground truth measurements.
[10] Sawhney (personal communication).



(a) First image



(b) Last (ninth) image

Figure 1: **Rotating Box Image Sequence.**

Point Number	X mm	Y mm	Z mm
1	47.087	48.6205	610.082759
2	37.5461	41.2704	600.732575
3	-33.2854	26.938	592.292156
4	-45.4475	31.0323	600.916377
5	-56.2068	34.7835	609.540598
6	28.3388	34.579	591.382392
7	18.7286	27.4399	582.032208
8	-10.3954	19.573	575.043713
9	-22.43	23.1242	583.667934
10	-2.00129	41.6572	566.390325
11	9.4232	21.1895	572.682025
12	18.4164	106.617	666.338494
13	57.0223	42.159	627.382589
14	72.8858	-0.394479	668.531358
15	36.883	-28.2434	631.130623
16	26.5855	-35.5439	621.780440
17	-25.9591	-35.3044	632.040387
18	-27.3888	-22.5297	624.090741
19	-39.6847	-18.718	632.714962
20	-38.2377	-31.5477	640.664608
21	76.1267	-24.8034	684.430650
22	40.0046	-52.9064	647.029916
23	30.0219	-60.3532	637.679732
24	-23.2579	-60.5819	647.939679
25	-35.3715	-56.511	656.563901
26	-24.695	-47.701	639.990033
27	86.5104	-43.5601	709.680126
28	-4.56151	53.0589	584.364729
29	22.2511	85.194	630.389685
30	-2.49328	-42.8458	614.791944
31	-49.7363	68.2467	618.861614
32	41.3018	87.6421	631.115647
33	-6.50797	63.7761	602.339134
34	-18.8661	78.8061	628.937760
35	69.6582	24.9633	652.632065

Table 2: **Ground truth of tracked points at the final position of the box (in the camera coordinate system).**

of *points*), the algorithm requires a set of 2D image points as input. In this experiment the points that were reconstructed are the same 35 points that have been used in previous work by Sawhney [25]. All 35 points were corners of the small black squares on the box. [11] Tracking of corners was done using the tracking algorithm of Williams and Hanson [45]. [12]

4.1.2 Results – With and Without Cross–Correlations

The performance of the algorithm with and without the cross–correlation terms is presented here. For comparison we include the results from a standard two–frame approach: Horn's relative orientation algorithm [17]. [13]

In order to determine the performance of each algorithm in reconstructing the *shape* of the box, the reconstruction is rotated and translated (rigidly) to align with the ground truth. The mismatch between the aligned reconstruction and the ground truth is the error in the shape. The alignment that minimizes the mismatch error can be determined exactly (in closed form) by Horn's absolute orientation algorithm [16].

The error in the shape after alignment is reported for each of the three motion algorithms. If the 3D coordinate of a point after alignment is $\mathbf{P_i'}$ and the true 3D coordinate of the same point is $\mathbf{T_i}$, then, the mismatch error is simply the Euclidean distance in 3-space:

$$mismatch\ \ error = \mid \mathbf{P_i'} - \mathbf{T} \mid \tag{19}$$

The overall error of the entire reconstruction is reported as an average of the individual mismatch errors over the set of reconstructed points.

[11] This MFSFM algorithm is not limited to corner points; any tracked point can be reconstructed. However, since tracking is robust when the point is a corner, choosing corners prevents confusion of tracking problems with problems that arise due to the reconstruction algorithm.

[12] Due to the well known scale ambiguity in SFM [40], the algorithm needs some information concerning the scale of the reconstruction. The information could be the true distance the camera moves between two frames, the true distance between any two points, or the average distance from the center of the scene to each point. In each case the information given to the algorithm involves a single number introduced at the beginning of the process. For the experiments reported here we use used the third method – introducing the average distance to points.

[13] Horn's algorithm provides the input for Thomas and Oliensis' MFSFM algorithm.

Results from Two-frame Algorithm In the two–frame approach consecutive pairs of images are used to reconstruct the scene (e.g. the reconstruction associated with the 5th frame involves using the 4th and 5th frames). From the graph (Figure 2) it can be observed that the error in the two–frame reconstruction is fairly high (the average error is 8.8 mm; the dimensions of the box are 133 mm x 157 mm x 70 mm and the distance between any two points ranges from 15 mm to 207.19 mm). The random and high fluctuations (e.g. in frame 4 and frame 7) make the two–frame reconstructions unreliable.

Results from MSFSM Without Cross–correlations This algorithm is identical to the one developed in [36] except that cross–correlations of the error covariance matrix are ignored. From the graph (Figure 2) we can see that after an initial drop in error, the error fluctuates around 11 mm, but has a very slow decrease. Notice also that in frames 4 and 7 the error increases, showing that the algorithm is unable to ignore the erroneous individual two–frame reconstructions.

Results from MFSFM With Cross–correlations The MFSFM algorithm with cross–correlations yields the best stability and accuracy of the three approaches compared here. Figure 2 shows that the average reconstruction error falls monotonically and remains as low as the error in the ground truth (1.5 mm) for the last 4 frames. Note that the final reconstruction (after 9 frames) of the MFSFM approach without cross–correlations is 6 times more erroneous than the final reconstruction when cross–correlations are considered. The only other reported result on this sequence is that of Sawhney, Oliensis and Hanson [26] wherein the box is reconstructed with an accuracy of 0.9% which corresponds to an average error of 5.4mm. This algorithm is for a restricted motion model, namely rotation.

This experiment is limited by the number of frames in the image sequence with ground truth of the interframe motion. However, performance characteristics of the approaches in

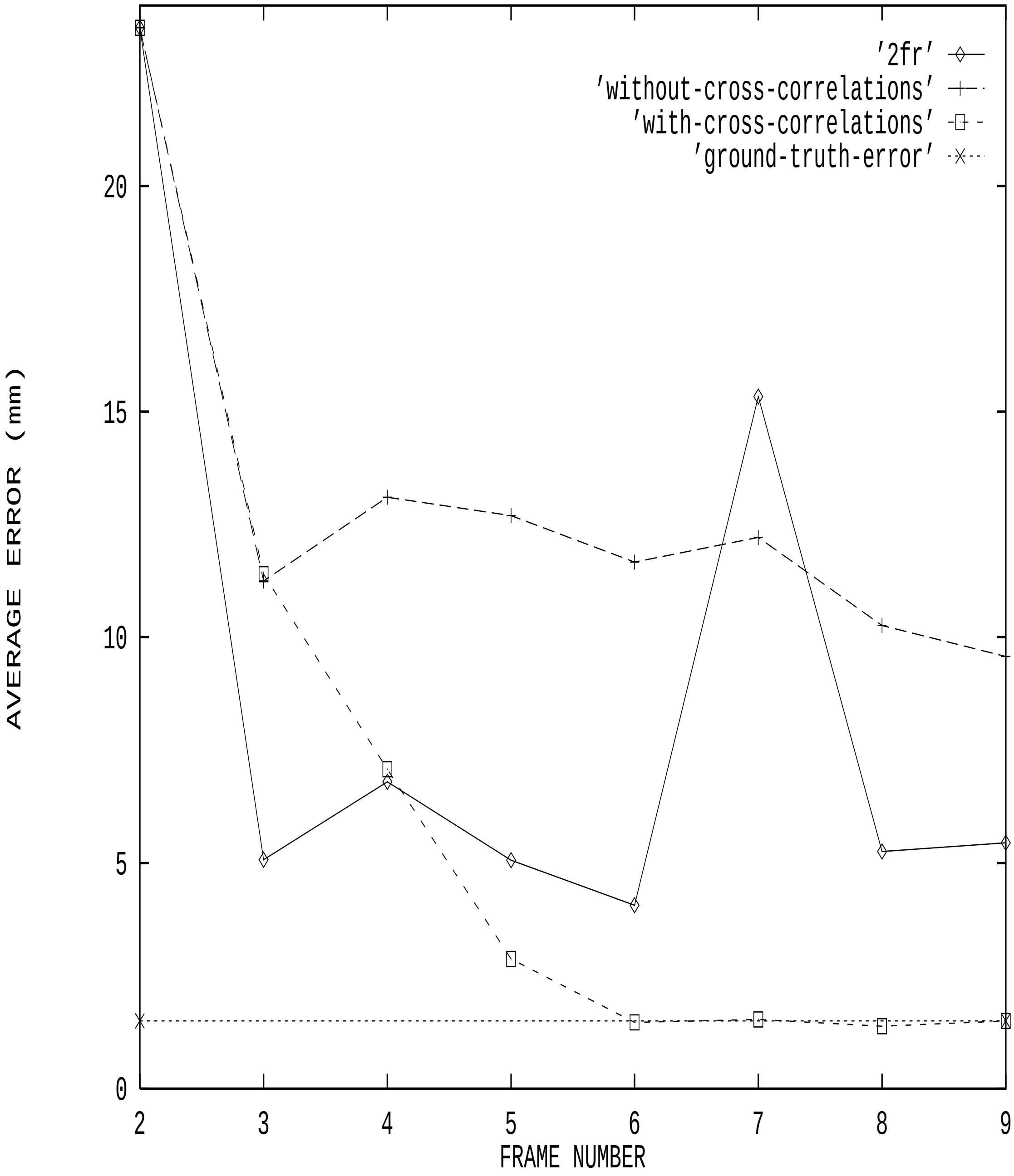


Figure 2: **Box reconstruction error for MFSFM algorithm with and without cross–correlations. The error in the two–frame algorithm is also plotted for comparison.**

longer *simulated* image sequences are reported in [38].

4.2 Experiment II: Reconstruction of the Computer Science Lobby

4.2.1 Description of the Input

A sequence of pictures were taken of the Computer Science (CS) lobby by a camera mounted on a moving Denning mobile robot. Since the CS Lobby is more or less featureless, several posters were placed on the walls and a few obstacles were placed along the path of the robot. The robot was commanded to move straight ahead, but the actual movement contains a rotation and a drift. The camera used to obtain the images was a Sony black and white CCD camera. The parameters of this camera are given in Table 3.

focal length	fov X	fov Y	Size
16 mm	29.27°	22.865°	256×242

Table 3: **Camera Parameters.** The camera parameters of the Sony AVC-D1 camera that is mounted on the mobile robot.

Figure 3 depicts the first and the last images of the sequence. For this experiment 29 points were selected from the first image, making sure that each point was visible in the rest of the images. Of all the corners of posters or doors or other physical objects the 29 points were randomly selected. Corners were selected so that they could be tracked robustly – as in the previous experiment (cf. Section 4.1) – using a version of the tracking algorithm proposed by Williams and Hanson [45].

For 24 points, the ground truth measurements were made by hand using a tape measure with an accuracy of about an inch (at distances of 25–40 feet). The 3D coordinates of the remaining five points were approximated by using the measured distance to the wall, Z, and the coordinates, (X, Y), which were obtained by an algorithm due to Collins [6]. This algorithm

(a) First image

(b) Final (tenth) image

Figure 3: **Lobby Image Sequence.**

is based on perspective invariants and is highly accurate in adding 3D points to a partial model provided they points lie on a single plane, such as the wall. Table 4 shows the ground truth coordinates in the first robot position.

4.2.2 Results – With and Without Cross–correlations

Again, the MFSFM algorithm which uses the cross–correlations produces the best results. Ignoring the cross–correlation introduces 50% more error in the reconstruction than in the case when cross–correlations are taken into account.

The error in the reconstructed 3D coordinates is reported as a percentage of the distance of the true 3D coordinates from the camera. If the 3D coordinate of a point is $\mathbf{P_i}$ and the true 3D coordinate of the same point is $\mathbf{T_i}$, then,

$$percentage\ \ error = \frac{\mid \mathbf{P_i} - \mathbf{T_i} \mid}{\mid \mathbf{T_i} \mid} \times 100\% \tag{20}$$

The overall error of the entire reconstruction is reported as an average of the individual percentage errors over the set of reconstructed points.

Results from Two-frame Algorithm From the graph (Figure 4) it can be observed that the error in the two–frame reconstructions is high and fluctuates randomly. Its behavior is consistent with the error predicted in a similar scenario by Dutta and Snyder [9], with an average error of approximately 8%. The reconstruction errors are large making the reconstruction by itself useless for any realistic application in robot navigation tasks.

Results from MFSFM algorithm without Cross–correlations From the graph (Figure 4) we can see that this approach leads to better accuracies than individual two–frame results. However, the reconstruction error does not decrease monotonically; instead it fluctuates around 3.5 %.

Point Number	X ft	Y ft	Z ft	Distance $\sqrt{X^2+Y^2+Z^2}$ ft
1	5.64405	3.90701	30.73	31.4873
2	4.94663	3.93331	31.93	32.5494
3	4.92497	1.5739	31.93	32.3459
4	5.64108	1.53307	30.73	31.2811
5	5.40205	3.26483	31.13	31.7635
6	5.1062	3.25969	31.53	32.1067
7	5.12769	2.31601	31.53	32.0281
8	5.40635	2.34123	31.13	31.6826
9	4.52312	3.6612	33.33	33.8342
10	3.92329	3.70723	35.13	35.5423
11	3.92095	1.34919	35.13	35.3739
12	4.50811	1.27218	33.33	33.6575
13	1.11749	5.66567	42.73	43.1185
14	2.57623	-1.56078	25.68	25.856
15	-0.160206	1.26698	35.83	35.8528
16	-0.452535	0.592612	35.83	35.8378
17	-0.444421	-0.137322	35.83	35.833
18	-0.707709	-2.17432	35.83	35.9029
19	-0.710209	-1.24783	35.83	35.8588
20	-1.39912	-2.17932	35.83	35.9235
21	5.85031	-0.862638	29.93	30.5086
22	1.42646	-1.24638	25.68	25.7498
23	1.8694	-1.48509	25.68	25.7907
24	3.50672	5.66567	42.6653	43.1825
25	2.93134	3.93516	43.4213	43.6977
26	1.35078	3.49882	42.8189	42.9828
27	-0.244807	4.34697	42.2107	42.4346
28	-0.255581	3.48176	42.2066	42.3507
29	-1.21691	3.39113	41.8402	41.995

Table 4: **Ground truth of 29 tracked points with respect to the 1st robot position.**

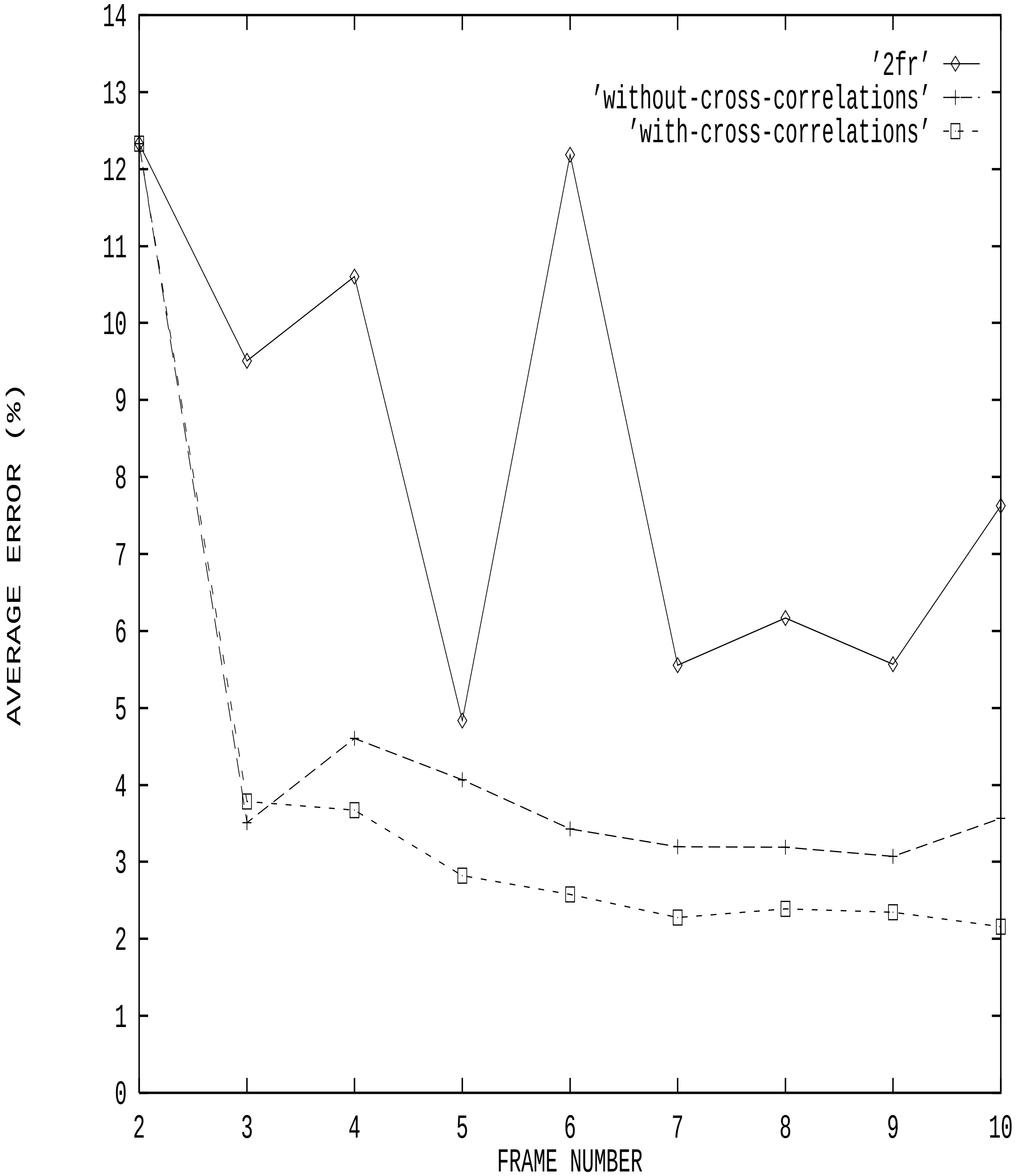


Figure 4: **Lobby reconstruction error for the MFSFM algorithm with and without cross–correlations. The error in the two–frame algorithm is also shown for comparison.**

Results of MFSFM algorithm with Cross–correlations Again, using Cross–correlations yields the best accuracy of the three approaches compared here. Figure 4 shows that the average reconstruction error falls almost monotonically, with a final error of 2.16% after ten frames. The final reconstruction (after ten frames) of the same MFSFM algorithm which ignores cross–correlations is 65% more erroneous than the final reconstruction which uses cross–correlations. The only other reported result using this lobby sequence is that of Sawhney and Hanson [25] wherein shallow planar structures of the lobby where reconstructed. The accuracy of the reconstruction is 2.3%. For this algorithm, approximate shallow planar structures must exist in the environment.

Although the MFSFM algorithm with Cross–correlations does well in the lobby sequence, the advantage of the algorithm in the lobby sequence is not as good as in the reconstruction of the rotating box in Experiment I. This difference in performance may be attributable to the effectively larger baseline in the direction orthogonal to the optical axis for the case of the rotating box, as compared with the lobby sequence where the camera translated in the direction of the optical axis.

Figure 5 shows a complete covariance matrix of the two–frame reconstruction for frame two. The cross–correlations (the off-diagonal terms) in this matrix are clearly significant in this case. The significant cross–correlations terms in this covariance matrix are typical of the terms in the rest of the two–frame covariances across the sequence.

Since at the second frame, the two–frame reconstruction is identical to the multi–frame reconstruction, Figure 5 also denotes the first multi–frame covariance. However, as the multi–frame reconstruction gets refined, the covariance matrix changes to accomodate the refinement. Figure 6 is the multi–frame covariance at frame five and Figure 7 is the multi–frame covariance at the final frame (frame ten). The magnitude of the covariances falls over time; for the purposes of display, the covariances have been scaled up. The covariance of the fifth frame (Figure 6) is

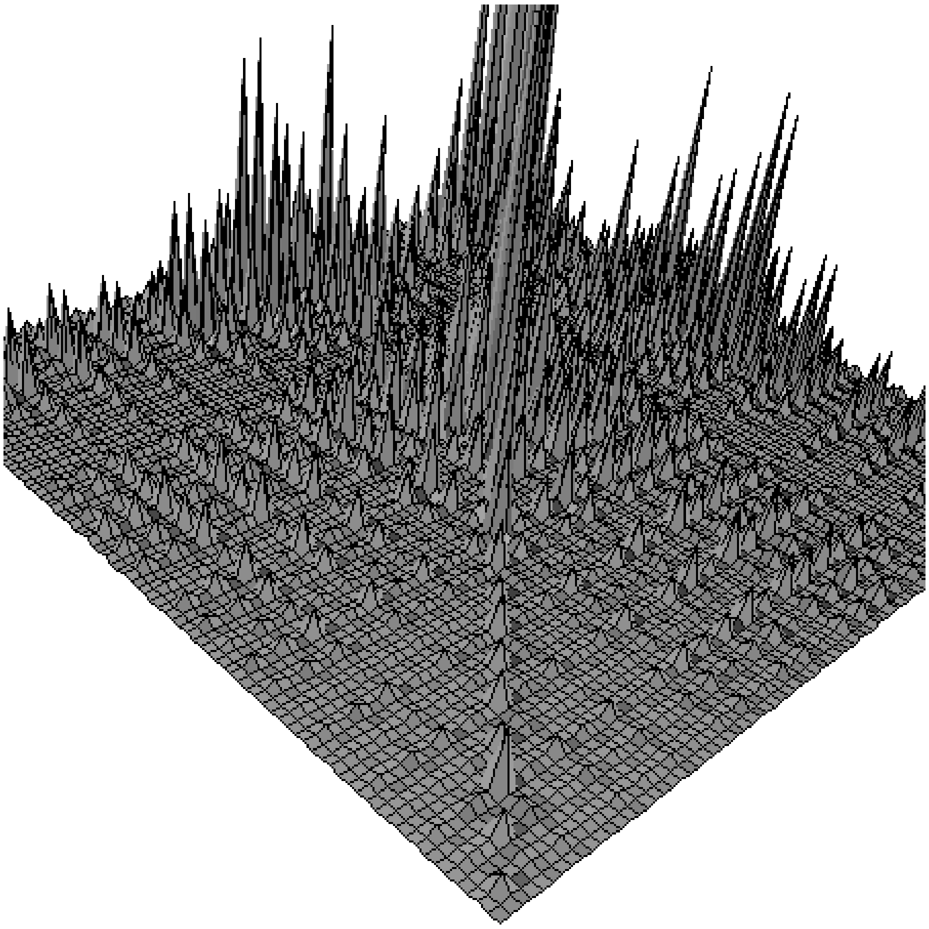


Figure 5: **The Covariance Matrix of the two–frame reconstruction at frame 2 (obtained from using images 1 and 2). The typical two–frame covariance matrix is of size** 87×87 **since it captures the error in 29 3D points. All terms of have been made positive for the clarity of the display. Note that the cross–correlations are significant.**

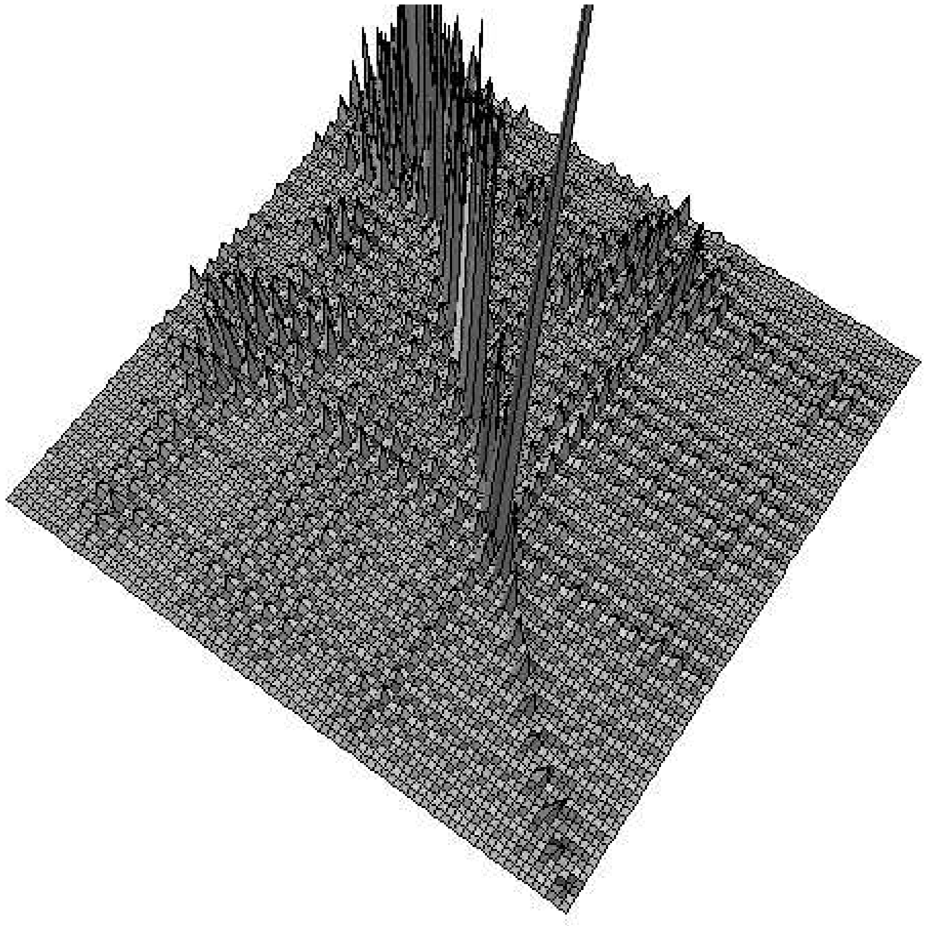


Figure 6: **The Covariance Matrix of the multi–frame reconstruction at frame 5 The covariance matrix is of size** 87×87 **since it captures the error in 29 3D points. All terms have been made positive and scaled up by two orders of magnitude for the clarity of the display.**

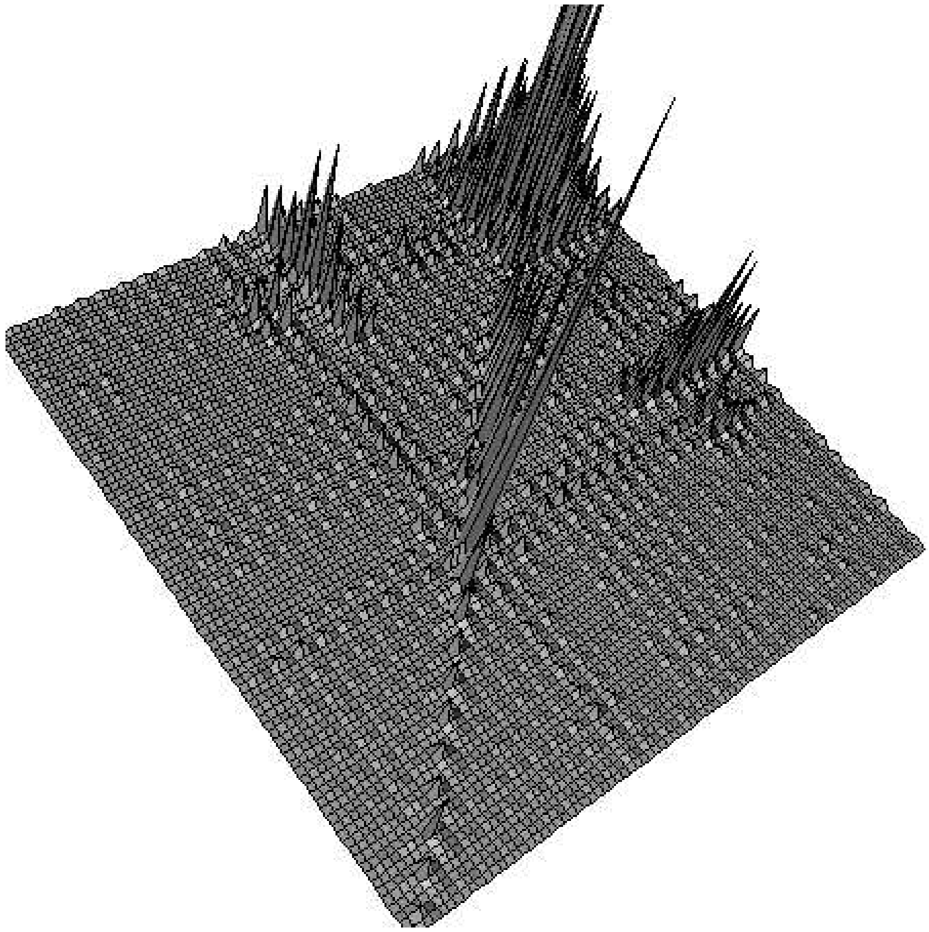


Figure 7: **The Covariance Matrix of the multi–frame reconstruction at frame 10. The covariance matrix is of size** 87×87 **since it captures the error in 29 3D points. All terms have been made positive and scaled up by three orders of magnitude for the clarity of the display.**

scaled up by two orders of magnitude and the covariance at frame ten (Figure 7) is scaled up by three orders of magnitude.

The covariances predict which points in the reconstruction are more erroneous than others. For example, in the multi–frame reconstruction after ten frames, the points that have the most error turn out to be those points that lie on the distant wall (cf. Figure 3b). In fact, the erroneous 3D points are identical to the ones that are stipulated to be erroneous by the peaks in the covariance matrix in Figure 7.

5 Conclusion

We have argued that the cross–correlation terms capture the interframe motion error and account for it. Ignoring the cross–correlations has direct consequences on the accuracy and usefulness of the reconstructed models of the environment.

Although the cross–correlations have presumably been ignored because of their computational complexity, we have shown that they are crucial enough to warrant an attempt to make using them computationally feasible. Since the bottleneck of including cross–correlations is the time required to invert large matrices, one solution is a straightforward parallel implementation of the algorithm on a SIMD parallel machine such as the Image Understanding Architecture [41] [42] [43]. Such a parallel algorithm should bring down the running time markedly (e.g. from $O(n^3)$ down to $O(n)$ for a reconstruction of n points). Future research will also be directed towards discovering other ways of reducing computational time such as using smaller (intersecting) subsets of points which are yet large enough to capture the underlying motion error. However, sparse models of the environment can be built in almost real–time (cf. [38]) and even such sparse models have been shown to be useful for the task of tracking the position of a navigating robot [37].

References

[1] G.Adiv,"Generating three-dimensional motion and structure from optic flow generated by several moving objects," *IEEE Trans. PAMI, 7(4)*, 1985, pp. 384-401.

[2] G.Adiv,"Inherent ambiguities in recovering 3d information from a noisy flow field," *IEEE Trans. PAMI, 11(5)*, 1989.

[3] H.Ando, "Dynamic Reconstruction of 3D Surfaces and 3D Motion," *Proc. IEEE workshop on visual motion*, Princeton, NJ, pp. 101-110, 1991.

[4] T. J. Broida and R. Chellappa, "Estimating the Kinematics and Structure of a Rigid Object from a Sequence of Monocular Images", *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 13, no. 6, pp. 497–513, 1991.

[5] A.R.Bruss and B.K.P.Horn, "Passive Navigation," *Computer Vision, Graphics and Image Processing,* Vol. 21, No.1, 1983, pp. 3-20.

[6] R.T.Collins, "Single Plane Model Extension using Projective Transformations," Proc. Darpa I.U. Workshop, San Diego, CA., Jan 1992, pp. 917–923.

[7] H.Cramer, Mathematical Methods of Statistics. Princeton Univ. Princeton, New-Jersy, 1946.

[8] N. Cui, J. Weng and P. Cohen, "Extended Structure and Motion Analysis from Monocular Image Sequences," *Proceedings 3rd IEEE International Conference on Computer Vision,* Osaka, Japan, 1990, pp. 222-229.

[9] R. Dutta and M. Snyder, "Robustness of Correspondence-Based Structure from Motion," *Proceedings 3rd IEEE International Conference on Computer Vision,* Osaka, Japan, Dec. 1990.

[10] O.D.Faugeras and S.Maybank, "Motion from point matches: multiplicity of solutions," *Proc. IEEE Workshop on Motion,* Irvine, CA, March, 1989. pp. 248-255.

[11] W.O. Franzen, "Structure and Motion from Uniform 3-D Acceleration," *Proc. IEEE workshop on visual motion*, Princeton, NJ, pp. 14-20, 1991.

[12] Technical Staff, The Analytical Sciences Corp., and A. Gelb, ed., *Applied Optimal Estimation*, MIT Press, 1986.

[13] J.J.Gibson, *The perception of the visual world*, Cambridge, Mass, Riverside, 1950.

[14] J. Heel, 'Dynamic Motion Vision," *Image Understanding Workshop*, Palo Alto, CA, pp. 702-713, 1989.

[15] J.Heel, "Temporal Surface Reconstruction," *CVPR*, Hawaii, June, 1991, pp. 607-612.

[16] B. K. P. Horn, "Closed Form Solution of Absolute Orientation Using Unit Quaternions," *J. Opt. Soc. Am.* A, vol. 4, pp. 629-642, 1987.

[17] B. K. P. Horn, "Relative Orientation," *International Journal of Computer Vision,* Vol. 4, pp. 59-78, 1990.

[18] R. V. R. Kumar, A. Tirmalai and R.C. Jain, "A Nonlinear Optimization Algorithm for the Estimation of Structure and Motion Parameters," *CVPR*, San Diego, CA, pp. 136-143, 1989.

[19] H.C.Longuet–Higgins and K.Prazdny, "The interpretation of a moving retinal image," pp. 179-193.

[20] L. Matthies, T. Kanade, and R. Szeliski, "Kalman Filter–Based Algorithms for Estimating Depth from Image Sequences," *International Journal of Computer Vision*, vol 3, pp. 209-236, 1989.

[21] L. Matthies, R. Szeliski, and T. Kanade, "Incremental Estimation of Dense Depth Maps from Image Sequences," *CVPR*, Ann Arbor, Michigan, pp. 366-374, 1988.

[22] J. Oliensis and J. I. Thomas, "Incorporating Motion Error in Multi–frame Structure from Motion," *Proceedings IEEE Workshop on Visual Motion,* Princeton, pp 8-13, 1991.

[23] K.Prazdny, "On the information in Optical Flow," *Computer Vision, Graphics and Image Processing,* Vol. 22, 1983, pp.239-259.

[24] C.R.Rao, Linear Statistical Inference and Its Applications, 2nd Ed., Wiley, New York, 1973.

[25] H.S.Sawhney, Doctoral dissertation, Dept. of Computer Science, Univ. of Massachusetts, Amherst, 1992.

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

[27] Y.Bar–Shalom and T.E.Fortmann, Tracking and Data Association, Academic Press, Orlando, Fl, 1991, pp. 277.

[28] L.Shigang, S.Tsuji and M. Imai, "Determining of Camera Rotation from Vanishing Points of Lines on Horizontal Planes," *Proceedings 3rd IEEE International Conference on Computer Vision,* Osaka, Japan, 1990, pp. 499-502.

[29] M.E.Spetsakis and J.Aloimonos, "Optimal Computing of Structure from Motion Using Point Correspondences in Two Frames," *Proceedings 2nd IEEE International Conference on Computer Vision,* Tampa, Florida, Dec. 1988, pp. 449-538.

[30] M. Spetsakis and J.Aloimonos, "A Multi–Frame Approach to Visual Motion Perception," *Intl. J. Computer Vision,* Vol 6, No. 3, 245–255, 1991.

[31] M. Spetsakis and J.Aloimonos, "Optimal Visual Motion Estimation: A Note," *IEEE PAMI*, Vol 14, No. 9, 959–964, 1992.

[32] M.J.Stephens, R.J.Blissett, D.Charnley, E.P.Sparks and J.M.Pike, "Outdoor Vehicle Navigation Using Passive 3D Vision," *CVPR*, San Diego, CA, pp. 556-562, 1989.

[33] C.J.Taylor, D.J.Kreigman, and P.Anandan, "Structure and Motion in Two Dimensions from Multiple Images: A Least Squares Approach," *Proc. IEEE workshop on visual motion*, Princeton, NJ, pp. 242-248, 1991.

[34] J. I. Thomas and J. Oliensis, "Fusing Structure by Kalman Filtering," TR 90-93, COINS, UMASS, May 1990.

[35] J. I. Thomas and J. Oliensis, "Incorporating Motion Error in Multiframe Structure from Motion," *7th Scandinavian Conference on Image Analysis*, Denmark, pp. 950-957, 1991.

[36] J. Inigo Thomas and J. Oliensis, "Recursive Structure from Multi–frame Motion,", *Proc. Darpa Image Understanding workshop*, San Diego, CA, 1992.

[37] J. Inigo Thomas and J. Oliensis, "Automatic Position Estimation of a Mobile Robot," *Proc. Ninth IEEE Conference on Artificial Intelligence for Applications*, Orlando, FL, March 1993.

[38] J. Inigo Thomas, Reducing Noise in 3D Models Recovered from a Sequence of 2D Images, Ph.D. Dissertation, University of Massachusetts, Amherst, September 1993.

[39] C. Tomasi and T. Kanade, "Factoring Image Sequences into Shape and Motion," *Proc. IEEE workshop on visual motion*, Princeton, NJ, pp. 21-28, 1991.

[40] R.Y.Tsai and T.S.Huang, "Uniqueness and estimation of 3–D motion parameters and surface structures of rigid objects," pp. 135–171.

[41] C.C.Weems, J.H.Burrill. "The Image Understanding Architecture and its programming environment," *Parallel Architectures and Algorithms for Image Understanding,* V.K.Prassana-Kumar (ed.), Academic Press, Orlando, FL, 1990.

[42] C.C.Weems, D.Rana, D.B.Shu and J.G.Nash, "A progress report on the development of the Image Understanding Architecture," *Proc. IEEE Intl. Conf. on Pattern Recognition,* Atlantic City, N.J., June, 1990.

[43] C.C.Weems, S.P.Levitan, A.R.Hanson, E.R. Riseman, D.B.Shu and J.G.Nash, "The Image Understanding Architecture," *IJCV,* 2, pp. 251-282.

[44] J.Weng, T.Huang and N.Ahuja, "Motion from Images: Image Matching, Parameter Estimation and Intrinsic Stability," *Proc. IEEE Workshop on Motion,* Irvine, CA, March, 1989, pp. 359-366.

[45] L. R. Williams and A. R. Hanson, "Translating Optical Flow into Token Matches and Depth from Looming," *Proc. 2nd Intl. Conf. on Computer Vision*, pp. 441-448, 1988.