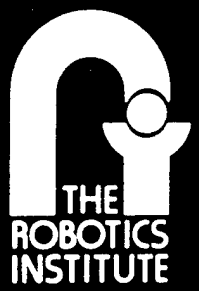
Dense Structure From A Dense Optical Flow Sequence

Yalin Xiong Steven A. Shafer

CMU-RI-TR-95-10

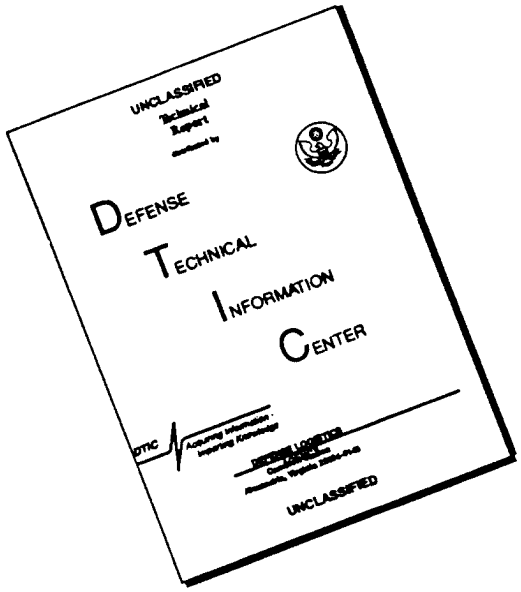


Carnegie Mellon University

The Robotics Institute

Technical Report

DISCLAIMER NOTICE



THIS DOCUMENT IS BEST QUALITY AVAILABLE. THE COPY FURNISHED TO DTIC CONTAINED A SIGNIFICANT NUMBER OF PAGES WHICH DO NOT REPRODUCE LEGIBLY.

Dense Structure From A Dense Optical Flow Sequence

Yalin Xiong Steven A. Shafer

CMU-RI-TR-95-10

The Robotics Institute
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213

April, 1995

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	April 1995	technical

4. TITLE AND SUBTITLE
Dense Structure from a Dense Optical Flow Sequence

5. FUNDING NUMBERS

6. AUTHOR(S)
Yalin Xiong and Steven A. Shafer

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
The Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15213

8. PERFORMING ORGANIZATION REPORT NUMBER
CMU-RI-TR-95-10

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSORING / MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release;
Distribution unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

This paper presents a structure-from-motion system which delivers *dense* structure information from a sequence of dense optical flows. Most traditional feature-based approaches cannot be extended to compute dense structure due to impractical computational complexity. We demonstrate that by decomposing uncertainty information into independent and correlated parts we can decrease these complexities from O (N2) to O(N), where is the number of pixels in the images. We also show that this dense structure-from-motion system requires only local optical flows, i.e. image matchings between two adjacent frames, instead of the tracking of features over a long sequence of frames.

14. SUBJECT TERMS	15. NUMBER OF PAGES
	33 pp
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
unlimited	unlimited	unlimited	unlimited

Contents

List of Figures

Abstract

This paper presents a structure-from-motion system which delivers *dense* structure information from a sequence of dense optical flows. Most traditional feature-based approaches cannot be extended to compute dense structure due to impractical computational complexity. We demonstrate that by decomposing uncertainty information into independent and correlated parts we can decrease these complexities from $O(N^2)$ to $O(N)$, where N is the number of pixels in the images. We also show that this dense structure-from-motion system requires only local optical flows, i.e. image matchings between two adjacent frames, instead of the tracking of features over a long sequence of frames.

1 Introduction

Structure from motion has been one of the most active areas in computer vision during the past decade. The idea is to recover structure or shape information from a sequence of images taken under unknown relative motions between the camera and the scene. Most approaches proposed in the literature can be classified according to whether they are based upon *features* or *optical flows*.

Feature-based methods compute the relative structure information among features by analyzing their 2D motion in images. Examples of such systems are reported by Tsai & Huang [19], Tomasi & Kanade [18], Broida et al [5] and Azarbayejani & Pentland [3] . Because the whole analysis is limited to features which usually number not more than hundreds, the results from those systems yield very sparse shape information. While stripping a full-resolution image to a handful of features may greatly simplify the algorithm and the computation, most of information contained in the image is lost. In many applications such as model acquisition, inspection and navigation, dense structural information is more desirable.

Traditional flow-based methods, such as reported by Bruss & Horn [6], Weng et al [20], Heeger & Jepson [11], Adiv [1], have concentrated on either solving the problem of recovering motion and structure from a single optical flow field or using very low resolution optical flows. As far as we know, little has been done to achieve a dense structure-from-motion system except Heel's work in [13]. Unfortunately, as pointed out in [18] that whether the proposed iterative algorithm in [18] converges is still an open question. Overall, the difficulties of such a system arise from two main factors:

- *Computation.* While a feature-based method can easily afford an $O(N^2)$ or $O(N^3)$ algorithm, where N is the number of features, a flow-based method cannot even afford an $O(N^2)$ algorithm, where N is the number of pixels.

- *Accumulation.* While a feature-based method can accumulate structural information for features because they are tracked across many frames, optical flows usually cannot be used to track pixels because their measurements are uncertain. In other words, while a feature-based approach quantify the image information as either totally unreliable or very reliable, a flow-based approach has to use a spectrum of reliability. Therefore, it is impossible to accumulate structural information by tracking all pixels across many frames in a flow-based method.

This paper shows our attempt at overcoming these difficulties. We demonstrate a system which *incrementally* accumulates *dense* structural information from a sequence of optical flows. The system has the following features:

- The system is based on EKF (extended Kalman filtering) as proposed in [5]. We will show in our experiment that the nonlinearity problem is actually not very serious even when the initial data are very crude.

- The formulation of the structure from motion uses separate independent and correlated structure uncertainty estimations. By employing the separation and

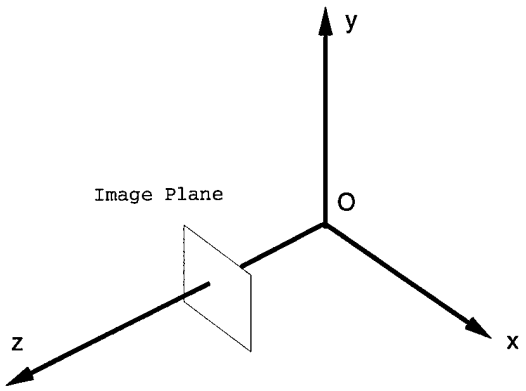


Figure 1: The Camera Coordinate

other mathematical techniques such as Sherman-Morrison-Woodbury inversion and principal component analysis, we can achieve an $O(N)$ numerical algorithm to compute Kalman filtering.

- The underlying motion of the camera can be discontinuous. Unlike many EKF-based approaches, ours computes the initial motion of every frame independently.
- We propose the concept of "Dynamic Motion Parameterization", which means that a different parameterization of the six motion parameters is used at every frame. Such a dynamic parameterization enables that the optical flow is equally sensitive to each of them, and therefore, stabilizes numerical computations.

2 System Overview

2.1 Coordinates and Motion

The system is based on the camera coordinate system $OXYZ$ shown in Figure 1, in which the origin O is the center of projection, the Z axis coincides with the optical axis, and the image plane is located at $Z = 1$.

If the relative motion of the camera with respect to the scene is composed of a translation velocity (U, V, W) and a rotation velocity (A, B, C), we have the following relation between the flow velocity (v_x, v_y) and the depth Z of pixel location (x, y) from [14]:

$$v_x = \frac{-U + xW}{Z} + Axy - B(x^2 + 1) + Cy, \quad (1)$$

$$v_y = \frac{-V + yW}{Z} - Bxy + A(y^2 + 1) - Cx. \quad (2)$$

If we designate the camera motion parameterization as $\vec{M}_0 = (U, V, W, A, B, C)^T$ and the flow velocity as $\vec{v} = (v_x, v_y)^T$, the above equation can be expressed as

$$\vec{v} = \vec{v}(x, y, Z, \vec{M}_0), \quad (3)$$

or its inverse

$$Z = Z(\vec{v}, x, y, \vec{M}_0). \tag{4}$$

2.2 Block Diagram of the System

Functionally, the system is decomposed into three major blocks as in Figure 2. For the sake of simplicity, we will refer to an optical flow and its uncertainty together as optical flow information, the structure and its uncertainty together as structural information, and the motion and its uncertainty together as motion information. We also designate the camera coordinate system before current motion as *a priori* coordinate system and the camera coordinate system after current motion as *posteriori* coordinate system. The computations within each block are as follows.

- *Initial Motion Estimate*: This block uses the current optical flow information and predicted structure information to compute an initial estimate of motion information for the current frame. Since the motion can be discontinuous, the current motion is independent of the previous motions. Once the motion information is estimated, we can re-parameterize the motion parameters such that they are equally sensitive to flow variations.

- *EKF-based Update*: This block uses the current flow information, predicted structure information and initial motion information to compute *posteriori* structural and motion information. The structure is represented with respect to the *a priori* coordinate system.

- *Interpolation and Transformation*: This block converts the structural information from the *a priori* coordinate system into the *posteriori* coordinate system by interpolation, spatial rotations and translations.

3 EKF-based Uncertainty Update

The Extended Kalman Filtering approach has been applied successfully in many fields to combine uncertainty information. In this section, we will briefly go through the general EKF framework as in [8] (Chapter 6), and then apply this framework to our nonlinear problem.

The measurements $\vec{z}$ and the state vector $\vec{x}$, which is what we need to estimate, are related according to:

$$\vec{z} = h(\vec{x}) + \vec{n}, \tag{5}$$

where $\vec{n}$ is a zero-mean measurement error whose covariance is $\mathbf{R}$. If the *a priori* estimates of the state vector and its covariance are $\vec{x}_-$ and $\mathbf{P}_-$ respectively, the

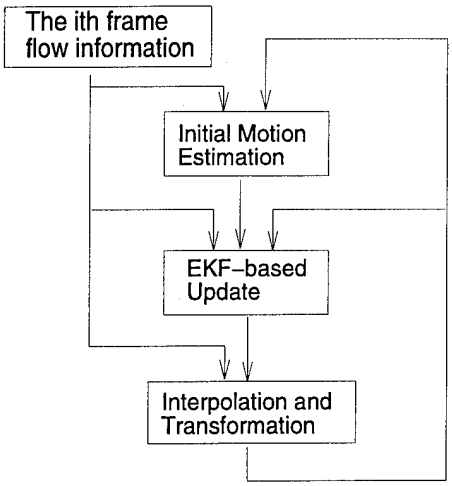


Figure 2: Block Diagram of The System

posteriori estimates of the state vector and its covariance after combining the new measurements are $\vec{x}_+$ and $\mathbf{P}_+$:

$$\vec{x}_+ = \vec{x}_- + \mathbf{K}(\vec{z} - h(\vec{x}_-)), \tag{6}$$

$$\mathbf{P}_+ = (\mathbf{I} - \mathbf{KH})\mathbf{P}_-, \tag{7}$$

where $\mathbf{K}$ is the Kalman gain

$$\mathbf{K} = \mathbf{P}_-\mathbf{H}^T(\mathbf{H}\mathbf{P}_-\mathbf{H}^T + \mathbf{R})^{-1}, \tag{8}$$

and $\mathbf{H}$ is the Jacobian matrix of the measurement equations, i.e.

$$\mathbf{H} = \frac{\partial h(\vec{x})}{\partial \vec{x}}|_{\vec{x}=\vec{x}_-}. \tag{9}$$

In the problem of dense structure from optical flows, the state vector is an $(N + 6)$ vector composed of six motion parameters $\vec{M}$ and depth at every pixel $Z_i, i = 1, 2, ..., N$, where N is the number of pixels. We express the state vector as

$$\vec{x} = (\vec{M}^T, Z_1, Z_2, ..., Z_N)^T. \tag{10}$$

Note that the motion parameters $\vec{M}$ have a linear relation with the original motion parameters $\vec{M}_0$ in Eq. 3 as we will show later, i.e.

$$\vec{M}_0 = \mathbf{T}\vec{M}, \tag{11}$$

where $\mathbf{T}$ is a 6×6 non-singular matrix.

The measurements are stored in a $2N$ vector which represents flow velocities at every pixel, i.e.

$$\vec{z} = (\vec{z}_1^T, \vec{z}_2^T, ..., \vec{z}_N^T)^T, \tag{12}$$

where $\vec{z}_i, i = 1, 2, ..., N$ is the measured flow velocity pari at each pixel. And the error covariance of the measurement vector $\vec{z}$ is a $2N \times 2N$ diagonal block matrix

$$\mathbf{R} = \begin{pmatrix} \mathbf{r}_1 & & & \\ & \mathbf{r}_2 & & \\ & & \ddots & \\ & & & \mathbf{r}_N \end{pmatrix}, \tag{13}$$

in which $\mathbf{r}_i, i = 1, 2, ..., N$ is the 2×2 error covariance matrix of the flow velocity at each pixel.

Using Eq. 3 and Eq. 11, the measurement equations can be expressed as

$$\vec{z} = \begin{pmatrix} \vec{v}(x_1, y_1, Z_1, \mathbf{T}\vec{M}) \\ \vec{v}(x_2, y_2, Z_2, \mathbf{T}\vec{M}) \\ \vdots \\ \vec{v}(x_N, y_N, Z_N, \mathbf{T}\vec{M}) \end{pmatrix} = \begin{pmatrix} \vec{v}_1 \\ \vec{v}_2 \\ \vdots \\ \vec{v}_N \end{pmatrix}, \tag{14}$$

in which $x_i, y_i, i = 1, 2, ..., N$ and $\mathbf{T}$ are known. The Jacobian matrix of the measurement equations is

$$\mathbf{H} = \begin{pmatrix} \frac{\partial \vec{v}_1}{\partial \vec{M}} & \frac{\partial \vec{v}_1}{\partial Z_1} & & \\ \frac{\partial \vec{v}_2}{\partial \vec{M}} & & \frac{\partial \vec{v}_2}{\partial Z_2} & \\ \vdots & & & \ddots \\ \frac{\partial \vec{v}_N}{\partial \vec{M}} & & & \frac{\partial \vec{v}_N}{\partial Z_N} \end{pmatrix} = \begin{pmatrix} \mathbf{A} & \mathbf{S} \end{pmatrix}, \tag{15}$$

in which $\mathbf{A}$ is a $2N \times 6$ matrix and $\mathbf{S}$ is an $N \times N$ diagonal block matrix with each block a 2×1 matrix.

Now that we have formulated the problem of recovering dense structure from optical flows in the EKF framework, it seems like all we need to do is to plug those formulas into Eq. 6 and Eq. 7 so that the dense structure information can be recursively estimated. And that is exactly what people did in feature-based methods such as [5] and [3]. Unfortunately, if we apply this scheme directly to the dense structure recovery problem, the computation and memory requirements are insurmountable. As pointed out in [17], the uncertainties of the depth values $Z_i, i = 1, 2, ..., N$ are correlated due to uncertain motion. Thus the covariance matrix $\mathbf{P}$ is a full $N \times N$ matrix. And the computation of the Kalman gain in Eq. 8 which contains an inverse of a full $2N \times 2N$ matrix requires at least $O(N^2)$ computation and memory. Considering an ordinary 256×256 image, even a symmetric $2N \times 2N$ (here $N = 256 \times 256 = 65,536$) matrix in single precision will require more than 30 Gigabytes of memory! Even if we could represent such a matrix, it is impractical to consider inverting it on ordinary workstations.

3.1 Decomposition of Independent and Correlated Uncertainty

Fortunately, we can take advantage of this specific problem to overcome these difficulties. As we mentioned before, the uncertainties of the depth values are correlated due to uncertain motion. Because there are only six motion parameters, the correlated uncertainty of the depth values caused by a single uncertain motion is an $N \times N$ matrix with rank of only six!

Because the rank of the correlated uncertainty is much smaller than N, the covariance matrix $\mathbf{P}$ can be decomposed into the following format:

$$\mathbf{P} = \begin{pmatrix} \mathbf{C}_m & \mathbf{C}_p^T \\ \mathbf{C}_p & (\mathbf{C}_s + \mathbf{U}\mathbf{V}^T) \end{pmatrix}, \tag{16}$$

where $\mathbf{C}_m$ is a 6×6 matrix representing the covariance of the motion parameters, $\mathbf{C}_p$ is an $N \times 6$ matrix representing the correlation between the motion and the structure, $\mathbf{C}_s$ is an $N \times N$ diagonal matrix representing the *independent* uncertainty of the depth value of each pixel, and $\mathbf{U}$ and $\mathbf{V}$ are both $N \times k$ matrices whose outer product is a rank k matrix representing the *correlated* uncertainty of the depth values. Therefore, storing the matrix $\mathbf{P}$ sparsely will only require $O(N)$ memory if k is a constant.

Now that we can represent the covariance matrix $\mathbf{P}$, we will show that in the EKF framework, once $\mathbf{P}_-$ can be represented in the format of Eq. 16, $\mathbf{P}_+$ can also be represented in the same format. In fact, because $\mathbf{R}$ and $\mathbf{H}$ are special matrices, the covariance matrix $\mathbf{P}$ can always be represented sparsely as in Eq. 16 throughout the whole optical flow sequence. We never need to explicitly represent $\mathbf{P}$ as an $(N+6) \times (N+6)$ matrix!

In our system, we assume that the motion is discontinuous, i.e. the current motion is uncorrelated to previous motions. Under this assumption, *a priori* correlation between the structure and the current motion $\mathbf{C}_p$ in Eq. 16 is zero. For simplicity, we will assume $\mathbf{C}_p$ is zero in the following sections, though in situations where this assumption is not true we also have similar results. If $\mathbf{P}_-$ is represented as in Eq. 16, after some manipulation, we have

$$\begin{aligned} \mathbf{H}\mathbf{P}_-\mathbf{H}^T + \mathbf{R} &= (\mathbf{S}\mathbf{C}_s\mathbf{S}^T + \mathbf{R}) + \begin{pmatrix} \mathbf{A}\mathbf{C}_m & \mathbf{S}\mathbf{U} \end{pmatrix} \begin{pmatrix} \mathbf{A}^T \\ \mathbf{V}^T\mathbf{S}^T \end{pmatrix} \\ &= \mathbf{C}_1 + \mathbf{U}_1\mathbf{V}_1^T, \end{aligned} \tag{17}$$

where $\mathbf{C}_1 = (\mathbf{S}\mathbf{C}_s\mathbf{S}^T + \mathbf{R})$ is an $N \times N$ diagonal block matrix with each block a 2×2 matrix, $\mathbf{U}_1$ and $\mathbf{V}_1$ are $2N \times (k+6)$ matrices as

$$\mathbf{U}_1 = \begin{pmatrix} \mathbf{A}\mathbf{C}_m & \mathbf{S}\mathbf{U} \end{pmatrix}, \mathbf{V}_1 = \begin{pmatrix} \mathbf{A} & \mathbf{S}\mathbf{V} \end{pmatrix}. \tag{18}$$

By applying the Sherman-Morrison-Woodbury formula as in [9] and Appendix A, we can invert the above matrix

$$(\mathbf{HP\_H}^T + \mathbf{R})^{-1} = (\mathbf{C}_1 + \mathbf{U}_1\mathbf{V}_1^T)^{-1} = \mathbf{C}_2 + \mathbf{U}_2\mathbf{V}_2^T, \tag{19}$$

where $\mathbf{C}_2$ is also an $N \times N$ diagonal block matrix with each block a 2×2 matrix, $\mathbf{U}_2$ and $\mathbf{V}_1$ are $2N \times (k+6)$ matrices.

Substituting Eq. 19 back into Eq. 8, we obtain the Kalman gain

$$\mathbf{K} = \begin{pmatrix} \mathbf{K}_m \\ \mathbf{C}_3 + \mathbf{U}_3\mathbf{V}_3^T \end{pmatrix}, \tag{20}$$

where $\mathbf{K}_m$ is a $6 \times 2N$ matrix

$$\mathbf{K}_m = \mathbf{C}_m\mathbf{A}^T\mathbf{C}_2 + \mathbf{C}_m\mathbf{A}^T\mathbf{U}_2\mathbf{V}_2^T, \tag{21}$$

$\mathbf{C}_3$ is an $N \times N$ diagonal block matrix with each block a 1×2 matrix

$$\mathbf{C}_3 = \mathbf{C}_s\mathbf{S}^T\mathbf{C}_2, \tag{22}$$

$\mathbf{U}_3$ is a $(N+6) \times (3k+6)$ matrix

$$\mathbf{U}_3 = \begin{pmatrix} \mathbf{U} & \mathbf{C}_s\mathbf{S}^T\mathbf{U}_2 & \mathbf{U} \end{pmatrix}, \tag{23}$$

and $\mathbf{V}_3$ is a $2N \times (3k+6)$ matrix

$$\mathbf{V}_3 = \begin{pmatrix} \mathbf{C}_2\mathbf{SV} & \mathbf{V}_2 & \mathbf{V}_2(\mathbf{U}_2^T\mathbf{SV}) \end{pmatrix}. \tag{24}$$

Finally, the updated covariance $\mathbf{P}_+$ is

$$\mathbf{P}_+ = \begin{pmatrix} \mathbf{C}_{mp} & \mathbf{C}_{pp}^T \\ \mathbf{C}_{pp} & (\mathbf{C}_4 + \mathbf{U}_4\mathbf{V}_4^T) \end{pmatrix}, \tag{25}$$

where $\mathbf{C}_{mp}$ is a 6×6 *posteriori* covariance matrix of the motion parameters

$$\mathbf{C}_{mp} = \mathbf{C}_m - \mathbf{K}_m\mathbf{A}\mathbf{C}_m, \tag{26}$$

$\mathbf{C}_{pp}$ is an $N \times 6$ *posteriori* uncertainty correlation between the structure and current motion[1]

$$\mathbf{C}_{pp} = -(\mathbf{C}_s + \mathbf{U}\mathbf{V}^T)\mathbf{S}^T\mathbf{K}_m^T, \tag{27}$$

[1]Note that we assumed *a priori* correlation between structure and motion $\mathbf{C}_p$ is zero. But the *posteriori* correlation $\mathbf{C}_{pp}$ is not zero.

$\mathbf{C}_4$ is an $N \times N$ diagonal matrix representing the independent uncertainty in the structure estimation

$$\mathbf{C}_4 = \mathbf{C}_s - \mathbf{C}_3\mathbf{S}\mathbf{C}_s, \tag{28}$$

and $\mathbf{U}_4$ and $\mathbf{V}_4$ are $N \times (6k+6)$ matrices, whose outer product represents the correlated uncertainty in the structure information

$$\mathbf{U}_4 = \begin{pmatrix} \mathbf{U} & -\mathbf{U}_3 & -\mathbf{C}_3\mathbf{S}\mathbf{U} & -\mathbf{U}_3\mathbf{V}_3^T\mathbf{S}\mathbf{U} \end{pmatrix}, \tag{29}$$

$$\mathbf{V}_4 = \begin{pmatrix} \mathbf{V} & \mathbf{C}_s\mathbf{S}^T\mathbf{V}_3 & \mathbf{V} & \mathbf{V} \end{pmatrix}. \tag{30}$$

If we are careful about the ordering of matrix multiplications in the above equations, we then have an algorithm which updates the state vector and its covariance using $O(kN)$ computation and memory. Unfortunately, k increases linearly after each frame, which makes the above algorithm $O(MN)$ where M is the number of frames. Though M is usually much smaller than the number of pixels N, it is still impractical for long image sequences. In next section, we introduce weighted principal component analysis to keep k constant, and therefore achieve an $O(N)$ algorithm.

3.2 Weighted Principal Component Analysis

First of all, let us consider the eigenvalues and eigenvectors of the correlated uncertainty matrix $\mathbf{U}_4\mathbf{V}_4^T$. In general case, there are $l = 6k+6$ non-zero eigenvalues and corresponding eigenvectors, which can be computed easily as in Appendix B. Because the outer product represents covariance which must be symmetric, it can be expressed as

$$\mathbf{U}_4\mathbf{V}_4^T = \begin{pmatrix} & & & \\ \vec{e}_1 & \vec{e}_2 & \cdots & \vec{e}_l \\ & & & \end{pmatrix} \begin{pmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_l \end{pmatrix} \begin{pmatrix} \vec{e}_1^T \\ \vec{e}_2^T \\ \vdots \\ \vec{e}_l^T \end{pmatrix}, \tag{31}$$

where $\vec{e}_i, i = 1, 2, \ldots, l$ are $N \times 1$ eigenvectors, and $\lambda_i, i = 1, 2, \ldots, l$ are the corresponding eigenvalues ordered by magnitude such that λ_1 is the largest eigenvalue.

Every eigenvector is an $N \times 1$ vector, which represents an eigen-image. This eigen-image illustrates the *pattern* of the depth uncertainty, and the corresponding eigenvalue represents the *magnitude* of this depth uncertainty. For example, if the eigen-image is an image with same value at every pixel, the depth uncertainty represented by this eigen-image is that depth values of all pixels can change but only by the same amount. In other words, changes of depth values allowed by this eigen-image

have to be in the pattern specified by the eigen-image:

$$\delta \begin{pmatrix} Z_1 \\ Z_2 \\ \vdots \\ Z_N \end{pmatrix} = c\vec{e}, \tag{32}$$

where c is a scalar constant and $\vec{e}$ is the eigen-image. The meaning of the eigenvalue is similar to that of σ in a Gaussian distribution, which represents the magnitude of the uncertainty.

Since the eigenvalues in Eq. 31 are in descending order, and we can truncate the eigenvalues after first k largest ones, i.e.

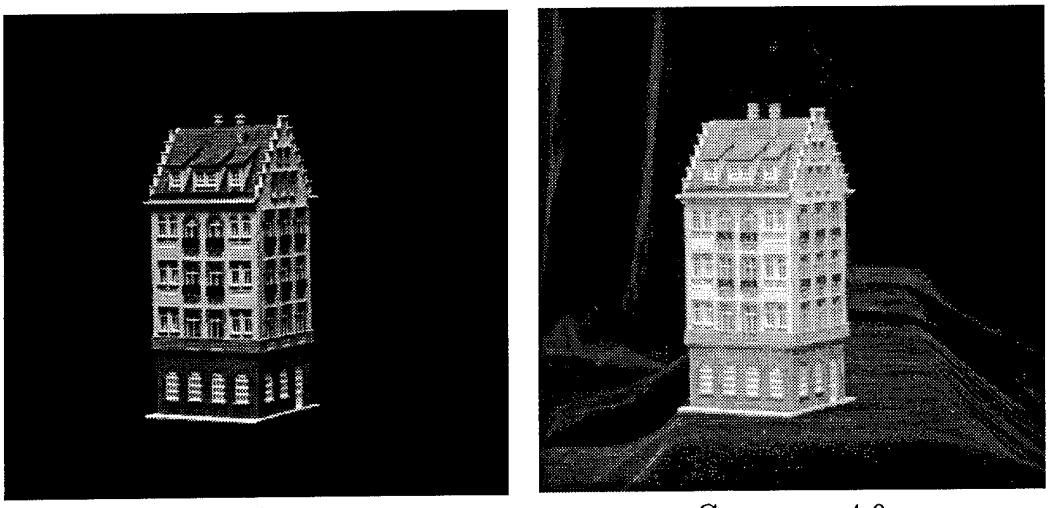
$$\mathbf{U}_4\mathbf{V}_4^T \approx \begin{pmatrix} & & & \\ \vec{e}_1 & \vec{e}_2 & \cdots & \vec{e}_k \\ & & & \end{pmatrix} \begin{pmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_k \end{pmatrix} \begin{pmatrix} & \vec{e}_1^T & \\ & \vec{e}_2^T & \\ & \vdots & \\ & \vec{e}_k^T & \end{pmatrix}. \tag{33}$$

Thus $\mathbf{U}_4$ and $\mathbf{V}_4$ can both be reduced to $N \times k$ matrices. The iterations of EKF updating illustrated in the previous section can be carried out in $O(N)$ for every frame no matter how long the sequence is.

The underlying assumption of truncating small eigenvalues in Eq. 33 is that the uncertainty implied by those eigenvalues/eigen-images is negligible compared to the independent uncertainty $\mathbf{C}_4$. And the reason for keeping large eigenvalues is that we assume the uncertainties implied by these large eigenvalues and their corresponding eigenvectors are at least comparable to the independent uncertainty $\mathbf{C}_4$. But since the independent uncertainties of pixels are not uniform, truncating by the magnitudes of eigenvalues may not make much sense at all because even though a relatively large eigenvalue may imply a large uncertainty in a certain area in the eigen-image, if the independent uncertainty happens to be even larger in the same area, this eigenvalue/eigen-image becomes less significant.

Based on the above speculation, we propose a weighted principal component analysis, i.e. the correlated uncertainty is weighted by independent uncertainty before decomposition as in Eq. 31. Since the independent uncertainty $\mathbf{C}_4$ is a diagonal matrix, and its diagonal elements have to be positive, we can decompose it as

$$\begin{aligned} \mathbf{C}_4 &= \begin{pmatrix} c_1 & & & \\ & c_2 & & \\ & & \ddots & \\ & & & c_N \end{pmatrix} \\ &= \begin{pmatrix} \sqrt{c_1} & & & \\ & \sqrt{c_2} & & \\ & & \ddots & \\ & & & \sqrt{c_N} \end{pmatrix} \begin{pmatrix} \sqrt{c_1} & & & \\ & \sqrt{c_2} & & \\ & & \ddots & \\ & & & \sqrt{c_N} \end{pmatrix} = \mathbf{Q}\mathbf{Q}^T. \end{aligned} \tag{34}$$



Original Image Gamma = 4.0

Figure 3: An Image Sequence of A Toy House

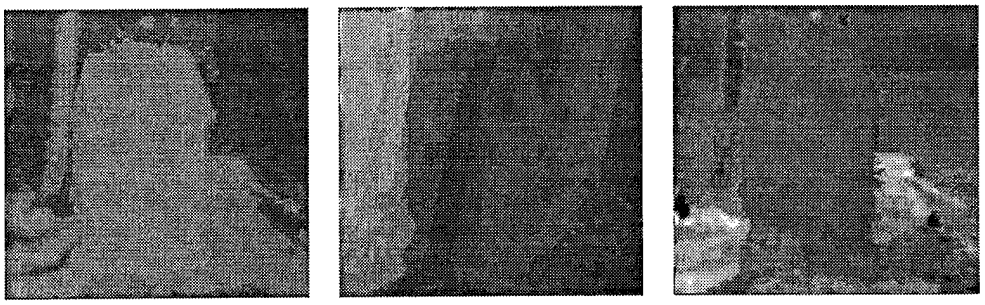
Therefore, the overall uncertainty can be represented as

$$\mathbf{C}_4 + \mathbf{U}_4\mathbf{V}_4^T = \mathbf{Q}\left(\mathbf{I} + \mathbf{Q}^{-1}\mathbf{U}_4(\mathbf{Q}^{-1}\mathbf{V}_4)^T\right)\mathbf{Q}^T. \tag{35}$$

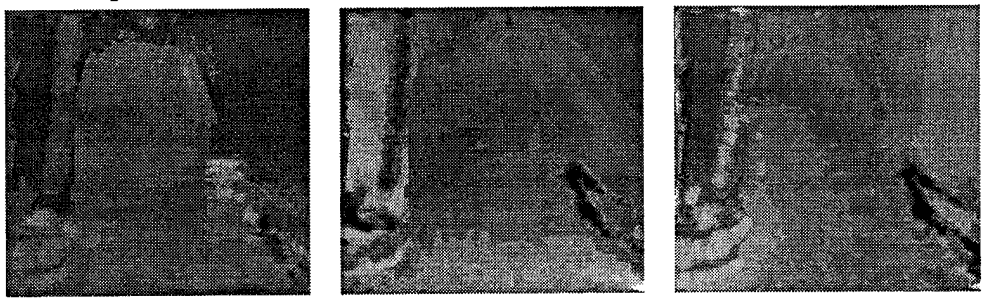
In other words, we have weighted the correlated uncertainty $\mathbf{U}_4$ and $\mathbf{V}_4$ by the independent uncertainty $\mathbf{Q}^{-1}$. We then truncate small eigenvalues of $\mathbf{Q}^{-1}\mathbf{U}_4(\mathbf{Q}^{-1}\mathbf{V}_4)^T$.

Figure 3 (Original Image) shows a toy house in front of the camera. The uncertainties of flow velocity in the dark background are very large comparing to those of the house area. There are small intensity variations in the background, which are visible after Gamma correction as in Figure 3. Figure 4 shows the six most significant eigenvalues/eigen-images of the correlated structure uncertainty using the direct decomposition method from Eq. 33. The eigen-images are shown by linearly quantizing 0 to grey-level 125, -0.025 or smaller values to grey-level 0 and 0.025 or larger values to grey-level 255. As indicated by either high or low grey-levels, the uncertainty information captured in the 2nd, 3rd, 4th, 5th and 6th eigen-images/eigenvalues is mainly in the background area. On the other hand, Figure 5 shows the first six weighted principal components. Obviously, the weighted principal components carry much more *useful* structural uncertainty information.

Since eigen-images represent orthogonal patterns of possible change or deformation of the depth map in Eq. 32, they also lend themselves for intuitive interpretations. For example, across the house in the second eigen-image in Figure 5 there is one skew line of grey-level 125, whose left side is bright and right side is dark. Referring to Eq. 32, we can see that this pattern represents a possible rotation around the skew line. Other eigen-images can be similarly interpreted though the patterns may be more complicated. In the context of structure from motion, we believe that the intrinsic ambiguity [2] of translation versus rotation of camera is represented and carried through recursive estimations by uncertainty patterns like these as we will show in experiments.

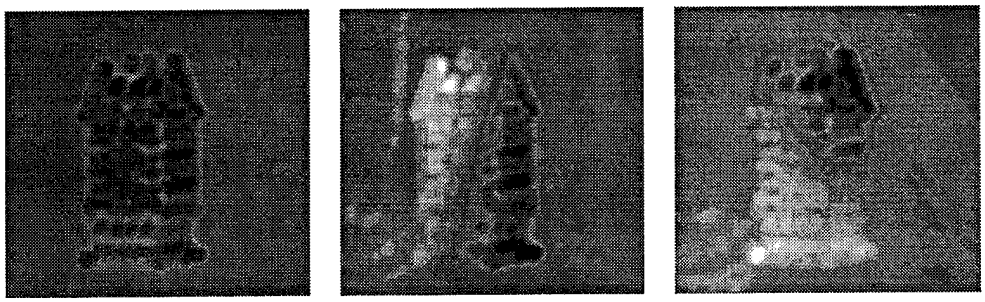


$\lambda_1 = 9143$ $\lambda_2 = 1508$ $\lambda_3 = 121.8$

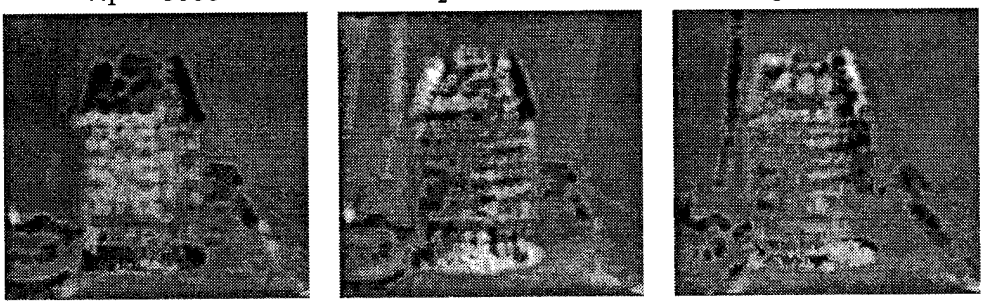


$\lambda_4 = 107.1$ $\lambda_5 = 48.5$ $\lambda_6 = 27.7$

Figure 4: Six Non-Weighted Principal Components



$\lambda_1 = 3056$ $\lambda_2 = 28.46$ $\lambda_3 = 3.373$



$\lambda_4 = 0.4026$ $\lambda_5 = 0.1428$ $\lambda_6 = 0.0527$

Figure 5: Six Weighted Principal Components

4 Initial Motion Estimation

Unlike many other approaches such as [3], we do not assume continuous motion. In other words, we assume that the motion at the current frame is totally unrelated to the motion in the previous frame because we believe that the continuous motion assumption is unrealistic in many cases such as navigating on real roads, hand-held video recording, and so on. Thus, in order to apply the EKF framework to our problem, we need to estimate an initial motion and its *a priori* covariance for each frame. Theoretically we don't need *a priori* covariance because it should be infinitely large. But in practice we need a finite one for numerical stability and re-parameterization.

First of all, given *a priori* structure Z and flow velocity v, we can estimate the initial motion $\vec{M}_0$ from Eq. 3 by a linear least squares fitting, e.g. minimizing

$$\sum_{i=1}^{N}(\vec{v}_i - \vec{v}(x_i, y_i, Z_i, \vec{M}_0))^T \mathbf{r}_i^{-1}(\vec{v}_i - \vec{v}(x_i, y_i, Z_i, \vec{M}_0)), \tag{36}$$

where $\mathbf{r}_i$ is the covariance of the flow velocity $\vec{v}_i$.

But we cannot use the above minimization to estimate the covariance of $\vec{M}_0$ because we didn't consider the uncertainties of structure Z. Designating the *a priori* depth map as $\vec{Z}_a$ and the depth computed from current motion M_0 and $\vec{v}$ by Eq. 4 as $\vec{Z}_c$, we have the following objective function to be minimized

$$\begin{aligned}\sum_{i=1}^{N}(\vec{v}_i - \vec{v}(x_i, y_i, Z_i, \vec{M}_0))^T \mathbf{r}_i^{-1}(\vec{v}_i - \vec{v}(x_i, y_i, Z_i, \vec{M}_0)) \quad + \\ (\vec{Z}_a - \vec{Z}_c)^T(\mathbf{C} + \mathbf{C}_d + \mathbf{U}\mathbf{V}^T)^{-1}(\vec{Z}_a - \vec{Z}_c),\end{aligned} \tag{37}$$

where $\vec{Z}_a$ and $\vec{Z}_c$ are $N \times 1$ vectors, $\mathbf{C} + \mathbf{U}\mathbf{V}^T$ is the structural uncertainty of $\vec{Z}_a$, and $\mathbf{C}_d$ is the depth uncertainty caused by current flow uncertainty given $\vec{M}_0$. Because the flow uncertainties of different pixels are independent, $\mathbf{C}_d$ is an $N \times N$ diagonal matrix.

If we ignore the dependence of $\mathbf{C}_d$ on $\vec{M}_0$, the minimization of Eq. 37 can be achieved using Levenberg-Marquardt method [16]. In fact, this simplification is justifiable because $\mathbf{C}_d$ is usually insensitive to $\vec{M}_0$. Once the objective function is minimized, the curvature at the minimal value can be used to compute the covariance of $\vec{M}_0$.

4.1 Dynamic Motion Parameterization

Motion is traditionally parameterized using three translation parameters and three rotation parameters as in Eq. 1 and Eq. 2. As pointed out in [3], if the camera has a long focal length, the optical flow is much more sensitive to translations in the XY plane to translations in the Z direction. Ideally we want the optical flow to be equally sensitive to all six motion parameters because otherwise the the covariance of motion

$\mathbf{C}_m$ in EQ. 16 could be numerically singular or near singular and therefore ruin the numerical computation of EKF.

We introduce the concept of "dynamic motion parameterization" to equalize sensitivities of motion parameters. There are two sources of sensitivity difference:

1. *Static Sensitivity Difference* is caused by the camera configuration. For example, if the camera has a narrow field of view, the optical flow is usually much more sensitive to rotation than translation.

2. *Dynamic Sensitivity Difference* is caused by current flow or depth estimate instead of the camera. For example, if the optical flow has uncertainty much larger in one direction than others, the optical flow is less sensitive to the motion which caused optical flow in that direction.

If we designate the covariance of $\vec{M}_0$ computed from minimizing the objective function of Eq. 37 as $\mathbf{C}_{mt}$, we can normalize sensitivities by using a new set of motion parameters

$$\vec{M} = \mathbf{T}^{-1}\vec{M}_0, \tag{38}$$

where

$$\mathbf{C}_{mt} = \mathbf{T}\mathbf{T}^T. \tag{39}$$

It can be easily verified that the covariance of the new motion vector $\vec{M}$ is the unit matrix $\mathbf{I}$.

Note that we cannot use the unit matrix as $\mathbf{C}_m$ in Eq. 16. Theoretically the *a priori* motion covariance $\mathbf{C}_m$ should be infinitely large due to uncorrelated motion. In estimating covariance of the motion in this section, we have already used the optical flow information of the current frame. Therefore, it is actually *posteriori* motion covariance! Ideally, we want the *a priori* covariances to be small enough to avoid numerical problems, and yet large enough to not contain any information about the current frame. In practice, we use $1000\mathbf{I}$ as *a priori* motion covariance $\mathbf{C}_m$ because it avoids the numerical problem of an infinitely large covariance and is also large enough (compared to *posteriori* covariance $\mathbf{I}$) to be uninformative.

5 Interpolation and Forward Transformation

We represent the 3D shape by a depth map in the current camera coordinate system as in Figure 1. Therefore, we need to transform the previous depth map into the current camera coordinate system and resample the depth map according to the current sensor grid. There are two new problems which were previously unsolved:

- Though the depth map and its independent uncertainty can be easily interpolated as in [15, 12], the interpolation of correlated uncertainty is a new problem.

- Most existing recursive structure-from-motion systems ignore the fact that motion and structure are actually correlated as $\mathbf{C}_{pp}$ in Eq. 25 when they rotate or

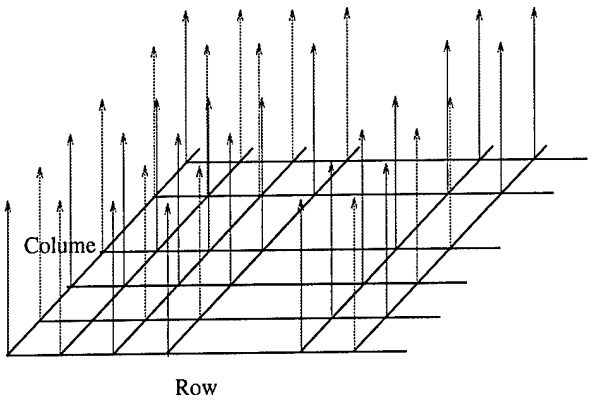


Figure 6: Vector Field as Correlated Uncertainty

translate the structure according to the motion. Though the exact effects of this simplification are still unknown, our system will perform a correlated translation and rotation.

As explained in Appendix B, since a correlated uncertainty is always a positive definite symmetric outer product, it can be represented as

$$\mathbf{U}\mathbf{V}^T = \mathbf{B}\mathbf{B}^T, \tag{40}$$

where $\mathbf{B}$ is an $N \times k$ matrix just like $\mathbf{U}$. In other words, every row of $\mathbf{B}$ is a vector of length k that can be regarded as an attribute of the corresponding pixel. Therefore we can represent the correlated uncertainty as a vector field as in Figure 6. Further more, the correlated uncertainty between any two locations is the dot product of the vectors at the two locations. Interpolating the correlated uncertainty is done by interpolating this vector field.

Since the optical flow establishes the correspondence between two adjacent frames, we can interpolate, resample and transform the depth map represented in the previous camera coordinate such that we have the depth and uncertainty information for grid positions in the current frame. We designate this process as the "forward transform". For every pixel position in the current frame, suppose its correspondence in the previous frame is at location (x_i, y_i) in the image plane, and has depth Z_i in the previous camera coordinate, we can compute the depth Z_i^+ in the current camera coordinate using the motion parameters, e.g.

$$Z_i^+ = f(x_i, y_i, Z_i, \vec{M}_0) = f(x_i, y_i, Z_i, \mathbf{T}\vec{M}), \tag{41}$$

where f represents 3D rotation and translation function. The 3D transformation matrix can be found in [7] (page 52).

To compute the structural uncertainty in the new camera coordinate, we take the derivative of Eq. 41

$$\begin{aligned} dZ_i^+ &= \frac{\partial f}{\partial Z_i} dZ_i + \frac{\partial f}{\partial \vec{M}} \cdot d\vec{M} \\ &= a_i dZ_i + \vec{b}_i^T d\vec{M}, \end{aligned} \tag{42}$$

where $\vec{b}_i$ is a vector of length six. Thus we have the covariance between two arbitrary points as

$$\begin{aligned} E[dZ_i^+ dZ_j^+] &= a_i a_j E[dZ_i dZ_j] + a_i \vec{b}_j^T E[dZ_i d\vec{M}] + \\ & \quad a_j \vec{b}_i^T E[dZ_j d\vec{M}] + \vec{b}_i^T E[d\vec{M} d\vec{M}^T] \vec{b}_j. \end{aligned} \tag{43}$$

From Eq. 25, we know that

$$\mathbf{C}_{mp} = E[d\vec{M} d\vec{M}^T], \tag{44}$$

$$\mathbf{C}_{pp} = \begin{pmatrix} E[dZ_1 d\vec{M}]^T \\ E[dZ_2 d\vec{M}]^T \\ \vdots \\ E[dZ_N d\vec{M}]^T \end{pmatrix}. \tag{45}$$

Therefore, we have the structural uncertainty after the forward transform as

$$E\left[\begin{pmatrix} dZ_1^+ \\ dZ_2^+ \\ \vdots \\ dZ_N^+ \end{pmatrix} \begin{pmatrix} dZ_1^+ \\ dZ_2^+ \\ \vdots \\ dZ_N^+ \end{pmatrix}^T\right] = \mathbf{C}^+ + \mathbf{U}^+ (\mathbf{V}^+)^T, \tag{46}$$

where

$$\mathbf{C}^+ = \begin{pmatrix} a_1^2 & & & \\ & a_2^2 & & \\ & & \cdots & \\ & & & a_N^2 \end{pmatrix} \mathbf{C}_4, \tag{47}$$

$$\mathbf{U}^+ = \begin{pmatrix} a_1 \vec{u}_1^T & a_1 \vec{p}_1^T & \vec{b}_1^T & \mathbf{C}_{mp} \vec{b}_1^T \\ a_2 \vec{u}_2^T & a_2 \vec{p}_2^T & \vec{b}_2^T & \mathbf{C}_{mp} \vec{b}_2^T \\ \vdots & \vdots & \vdots & \vdots \\ a_N \vec{u}_N^T & a_N \vec{p}_N^T & \vec{b}_N^T & \mathbf{C}_{mp} \vec{b}_N^T \end{pmatrix}, \tag{48}$$

$$\mathbf{V}^+ = \begin{pmatrix} a_1 \vec{v}_1^T & \vec{b}_1^T & a_1 \vec{p}_1^T & \vec{b}_1^T \\ a_2 \vec{v}_2^T & \vec{b}_2^T & a_2 \vec{p}_2^T & \vec{b}_2^T \\ \vdots & \vdots & \vdots & \vdots \\ a_N \vec{v}_N^T & \vec{b}_N^T & a_N \vec{p}_N^T & \vec{b}_N^T \end{pmatrix}, \tag{49}$$

where

$$\mathbf{U}_4 = \begin{pmatrix} \vec{u}_1^T \\ \vec{u}_2^T \\ \vdots \\ \vec{u}_N^T \end{pmatrix}, \tag{50}$$

$$\mathbf{V}_4 = \begin{pmatrix} \vec{v}_1^T \\ \vec{v}_2^T \\ \vdots \\ \vec{v}_N^T \end{pmatrix}, \tag{51}$$

$$\mathbf{C}_{pp} = \begin{pmatrix} \vec{p}_1^T \\ \vec{p}_2^T \\ \vdots \\ \vec{p}_N^T \end{pmatrix}, \tag{52}$$

in which $\vec{u}_i$'s and $\vec{v}_i$'s are vectors of length k and $\vec{p}_k$'s are of length six. Since $\mathbf{U}^+$ and $\mathbf{V}^+$ are now $N \times (k+18)$ matrices, they may also be reduced to $N \times k$ by weighted principal component analysis.

6 Implementation Issues and Experiments

6.1 Implementation Issues

We implemented our system using single precision matrices. As always in manipulating large matrices, the numerical stability has to be carefully watched while carrying out those computations. Potentially there are following sources of unstable computations:

1. *Matrix Multiplication*: When computing the inner product of two large matrix $\mathbf{V}^T\mathbf{U}$, where both $\mathbf{U}$ and $\mathbf{V}$ are $N \times k$ ($k << N$), we have to carry those additions in double precision due to large N. For example, if the image is 256×256, we can easily lose four significant digits during multiplications, which could be devastating if they are carried out in only single precision.

2. *Ill-conditioned Matrix*: There is always a danger when one of the matrices in the computation is singular or near singular. In the worst case, we may lose all significant digits. Thus it is extremely helpful to avoid any ill-conditioned matrix if possible. In our system, we pay special attention to the following matrices:

   - Flow Uncertainty: The estimated optical flow uncertainty $\mathbf{r}_i$'s in Eq. 13.
   - Motion Uncertainty: We used dynamic motion parameterization to prevent the motion covariance matrix from being ill-conditioned.
   - Kalman Gain: In computing the Kalman gain as in Eq. 8, it is numerically devastating if $\mathbf{HP\_H}^T + \mathbf{R}$ is ill-conditioned. $\mathbf{HP\_H}^T$ represents the projection of the uncertainty of motion and structure to the uncertainty of optical flow. In order for $\mathbf{HP\_H}^T + \mathbf{R}$ to be well conditioned, we need to make sure that the projected uncertainty of optical flow is not significantly larger ($> 10^5$) than the estimated uncertainty $\mathbf{R}$. That is the reason we choose $1000\mathbf{I}$ as the *a priori* motion uncertainty.

3. *Sherman-Morrison-Woodbury Inversion*: Though Sherman-Morrison-Woodbury inversion significantly reduces the amount of computation and memory required compared to the traditional Gaussian elimination ([16]), it has the disadvantage of being more fragile numerically ([10, 4], Appendix A). From our experience, the eigenvalues of $\mathbf{UV}^T$ have to be at most 10^4 of the eigenvalues of $\mathbf{C}$ to allow a stable numerical inversion of $\mathbf{C} + \mathbf{UV}^T$ using Sherman-Morrison-Woodbury formula.

Another common problem of a structure from motion system is the handling of the disappearance and reappearance of parts of the scene due to relative movement between the camera and the scene. Our system had no problem dealing with new parts, which are simply assigned a preset large independent uncertainty and a zero correlated uncertainty. But the depth information of disappearing parts is discarded. In the future, we would like to maintain a global shape module such that the structure information of disappeared parts could be stored and retrieved.

6.2 Experiments

We tested our system on real image sequences taken by a Sony XC-75 video camera. The relative camera movements in all the sequences involve both rotation and translation. We digitized images in two ways. One is to digitize by matrox board while shooting the sequence. In order to digitize while taking images, we mount the camera on a computer-controlled 6 DOF platform in Calibrated Imaging Lab, and stop for every frame. Another way is to record the sequence on Umatic SP video tape, and digitize the tape frame by frame. Unfortunately, the digitizing device we have can only digitize one of two fields in every frame, and the videotape also introduces additional noise in the images. We will demonstrate the performance of our system on image sequences digitized both ways. All images in our experiments are 480×512.

6.2.1 Ambiguities

It is well known that there are intrinsic ambiguities in recovering structure from motion. The first kind of ambiguity, i.e. the scale ambiguity, states that the scale of the object or the absolute depth of the object can never be recovered. Secondly, if the camera has a small field of view, the optical flow caused by a small camera rotation is very similar to that caused by a small camera translation. Therefore, given an optical flow, there is a rotation/translation ambiguity. Thirdly, since the optical flow has its uncertainty, we will always have uncertainty in estimating other motion parameters such as rotation and translation around z axis though they are usually less significant. We also like to point out that there is no fundamental difference in terms of origin between the second and the third kinds of ambiguities other than their magnitudes for an ordinary camera. Historically the second kind of ambiguity was frequently singled out in literature.

In our system, we assign an initial depth and uncertainty to the first frame. Practically we assign a flat depth map and uniform independent uncertainty as *a priori*

depth information. It serves two purposes, which are disambiguation of the scale ambiguity by providing absolute depth, and allowing deformation of the *a priori* depth map to the true depth map by providing large independent uncertainty.

Our system keeps six principal eigenimages to represent the correlated uncertainty as shown in Figure 5. Among these eigenimages, the first one usually represents the first kind of ambiguity, i.e. the scale ambiguity. The second and the third ones usually represent the second kind of ambiguity in two orthogonal directions. And the rest ones represent other minor ambiguities.

Conceptually the independent uncertainty represents a *chaotic* uncertainty pattern, while the correlated uncertainty represents an *organized* uncertainty pattern. For example, if the eigenimage of a correlated uncertainty is uniformly bright, it represents that the corresponding depth map can move back and forth. In other words, the depth values of all pixels have to change uniformly while the shape doesn't change at all. Since we set *a priori* depth uncertainty as totally chaotic, we will expect that as more optical flow information is incorporated, the uncertainty will become less and less chaotic, more and more organized. In our framework, that means that magnitude of the independent uncertainty will decrease while the magnitude of the correlated uncertainty *could* increase.

Optical flow information doesn't provide anything which we could use to eliminate the scale ambiguity. Therefore we expect the eigenimage representing scale ambiguity in correlated uncertainty will have larger and larger eigenvalue. On the other hands, the second and third kinds of ambiguities are strong in some optical flows while weak in other ones. Thus the eigenimages representing these ambiguities can have increasing or decreasing eigenvalues depending on the optical flow sequence.

Figure 7 shows one frame in a fifty-frame sequence. The motions of the camera with respect to the straw hat involve translations in (X, Y, Z) three directions and rotations around (X, Y) two axis from the 1st frame to the 30th frame. From the 30th frame to the 40th frame, the motions are translations in Y direction and small rotations around X axis[2]. From the 40th frame to the 50th frame, the motions are translations in X direction and small rotations around Y axis. Figure 8 shows the first three eigenimages, and Figure 9 shows the evolutions of average independent uncertainty and the eigenvalues corresponding to the three eigenimages. Note that the eigenimages change from frame to frame. In the examples shown here, the eigenimages didn't change dramatically over the whole sequences, which simplifies the analysis of correlated uncertainties. First of all, the fact that the average independent uncertainty decreases monotonically and the first eigenvalue which represents the scale ambiguity increases monotonically indicates a steady improvement from a chaotic pattern to an organized pattern. Secondly, in the interval between the 30th and the 40th frame, there is an accumulating ambiguity of translation in Y direction versus rotation around X axis. This motion ambiguity mapping into structural uncertainty as generally increasing third eigenvalues. And because there is no ambiguity of translation in X direction versus rotation around Y axis, the second eigenvalue de-

[2]X is the column direction, and Y is the row direction

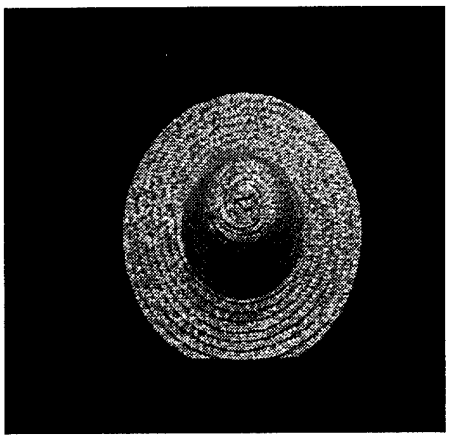
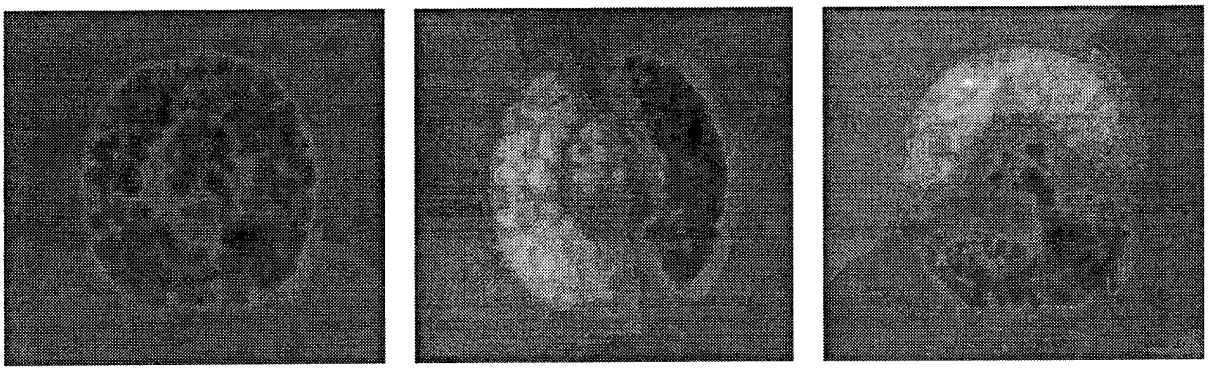


Figure 7: One Frame in A Strawhat Sequence



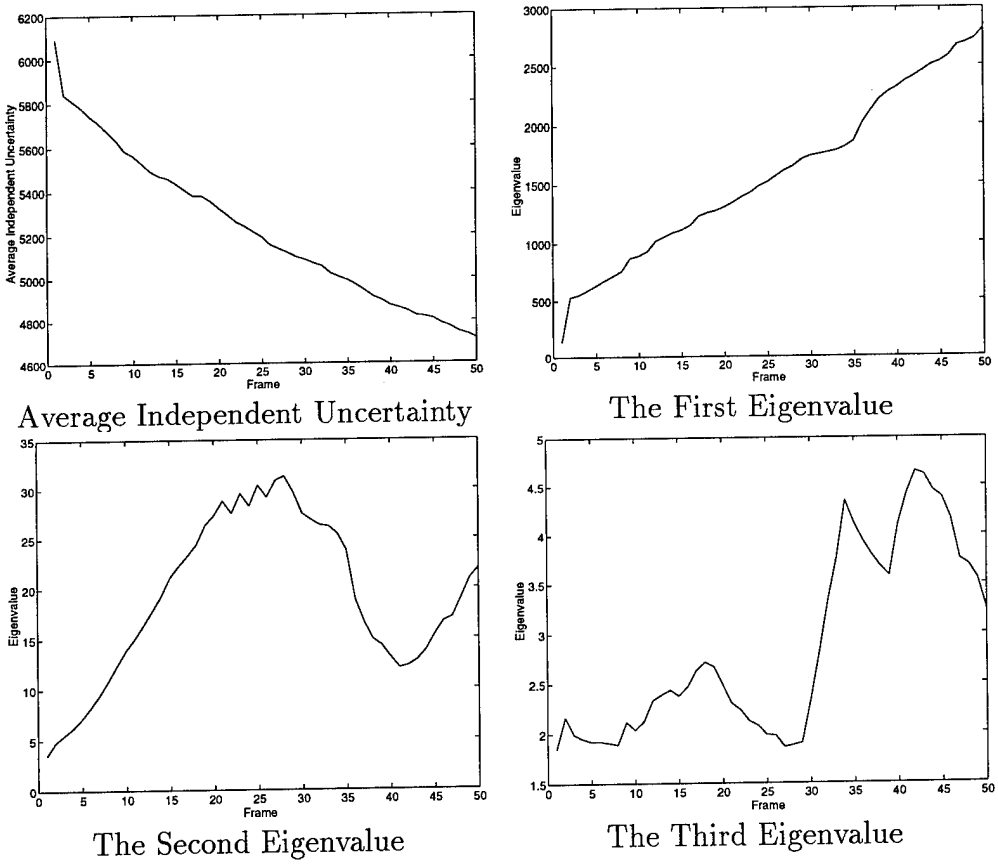
First Second Third

Figure 8: Three Eigenimages For the Strawhat Sequence

creases in the same time. For the similar reason, in the interval between the 40th and the 50th frame, the second eigenvalue increases while the third eigenvalue decreases for the extactly opposite reason.

If the underlying camera motions or the optical flows of the whole sequence tend to be rather homogeneous, the system may never be able to resolve one or more ambiguities intrinsic to this type of optical flows. Under this case the correlated uncertainty eigenimages representing the second kinds of ambiguity may have an ever increasing eigenvalues, which represent lack of information to disambiguate. If we have active control over the camera, the eigenimages could then be used to plan the camera motion in order to resolve the ambiguities.

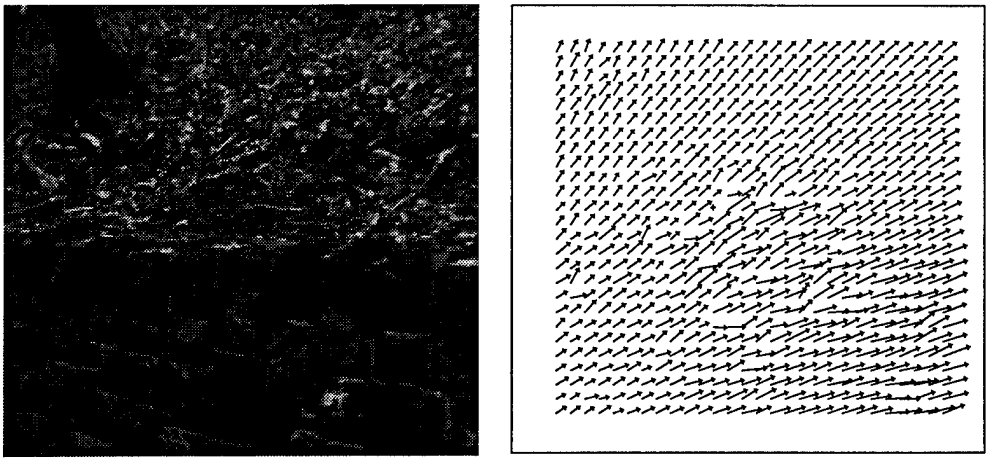
Figure 10 (1) shows one frame of a road sequence which was shoot using a camera fixed on a moving vehicle. The motion of the camera with respect to the scene was very homogeneous. In fact all the optical flows are similar to the one in Figure 10 (2). Figure 11 shows the first three eigenimages representing the correlated uncertainty. Figure 12 shows the average independent uncertainty and the eigenvalues corresponding to the three eigenimages. In this example, we can see that the ambigu-



Average Independent Uncertainty | The First Eigenvalue

The Second Eigenvalue | The Third Eigenvalue

Figure 9: Evolutions of Strawhat Uncertainties



(1) One Image Frame (2) One Optical Flow

Figure 10: A Road Sequence

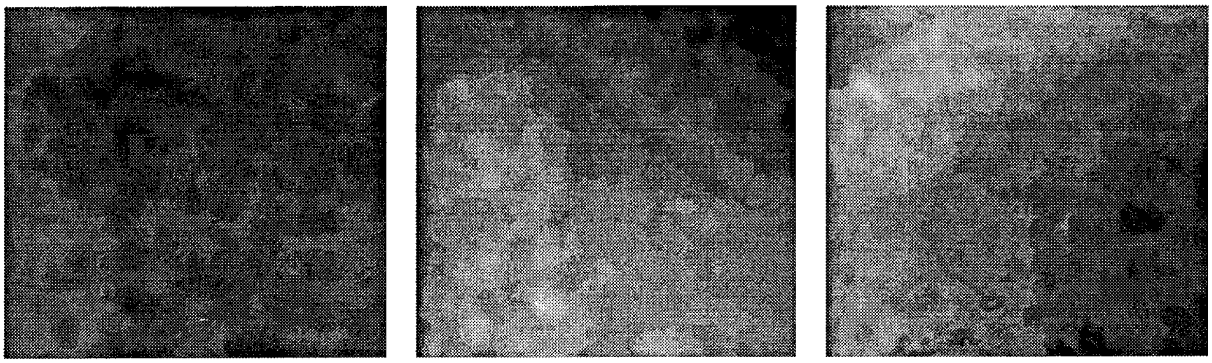


Figure 11: The First Three Eigenimages of the Road Sequence

ity represented by the second eigenimage was never resolved. Comparing the optical flow in Figure 10, it is obvious that these optical flows provided little information to resolve the second kind of ambiguity in the flow direction, i.e. the direction from bottom-left to up-right, while they did provide enough information to resolve the second kind of ambiguity in the direction perpendicular to the flow direction. Secondly, unlike the previous example, the magnitude of the average independent uncertainty and the first eigenvalue were approaching constants. In other words, after about 20 frames, the improvement from the chaotic uncertainty pattern to an organized uncertainty pattern seems stopped. The reason is that in this example, there is continuous appearing of new parts which were assigned chaotic uncertainties and disappearing of old parts whose uncertainties were organized. Therefore after certain number of frames, the improvement from chaotic to organized uncertainty pattern obtained in each new frame was totally cancelled by the introduction of new parts and lose of old parts.

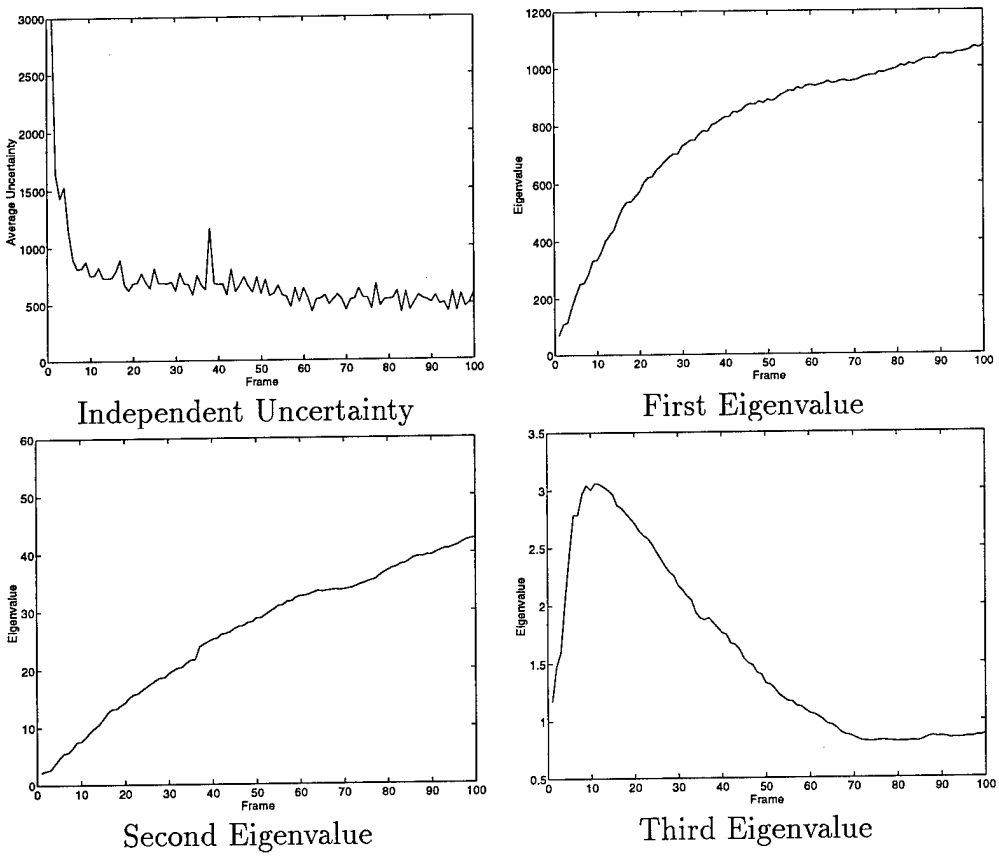


Figure 12: Evolutions of Road Uncertainties

6.2.2 Experiments on Real Sequences

We tested our system on many image sequences with different signal noise ratio and different field of view. No pre-processing was done on image sequences. Once we obtained a depth map sequence as output from our system, we masked out baskground areas since the depth information in these areas is arbitrary. The separation of foreground and background was done by a simple thresholding and hole-filling.

The first sequence includes fifty-one-frame images of a straw hat as we showed in the previous section The images were digitized by a matrox board. The rotations and translations of the camera with respect to the straw hat were discontinuous. The camera had about an 11° field of view. The optical flow and its uncertainty were computed using hyper-geometric filters [21, 22]. Figure 13 shows the intensity images and depth maps computed after the corresponding frames. It clearly shows the converging shape resulting from recursively combining information from multiple frames.

The Chair Sequence: The camera had an 22° field of view. The object was a real chair we used in our lab. The chair was rotating in front of the static camera as in Figure 14. Digitization was done by matrox board. The optical flow computation sometimes returned wrong results at some locations, which we believe were caused by texture aliasing. We can see that even in the tenth frame, the two buttons are very clear in the depth map. Also noticeable is that part of the chair in the left side is moving out and part of the chair in the right side is moving in. The move-in part is at first pretty noisy, and then gradually becomes smoother and smoother.

The Cube Sequence: The camera had about a 22° field of view. The sequence in Figure 15 was taken by a hand-held video camera connected to a Umatic recorder. It was recorded on a Umatic SP videotape and then digitized by a BVU digitizer, which could only capture one field in a frame. The digitized images have significantly higher noise levels than those digitized by matrox board. We can see that the system still performs pretty well on those noisy images.

The Basket Sequence: The camera had a 22° field of view. The target was a basket which moving and rotating in front of the camera. The digitization was done the same way as the cube sequence.

The Sphere and Dog Sequence: The camera had an 11° field of view. The target was a toy dog on top of a ball. The difficulties of this sequence are that (1) the ball had a very low contrast near its boundaries; (2) it had obvious specular reflections which will confuse the optical flow algorithm, and (3) the toy dog had a sparse texture. Despite these difficulties, our system still performs reasonably well as in Figure 17. The shape of the sphere and dog are both visible, and even the depth of the tail tip of the dog is correctly shown.

In all the experiments, the initial structure information was set as a flat surface parallel to the image plane at depth 200 with independent uncertainty $\sigma = 100$ at every pixel. Despite such a crude initial estimation, there was no trouble caused by nonlinearity in our experiments.

From all these experiments, we conclude that our system of recursively recovering

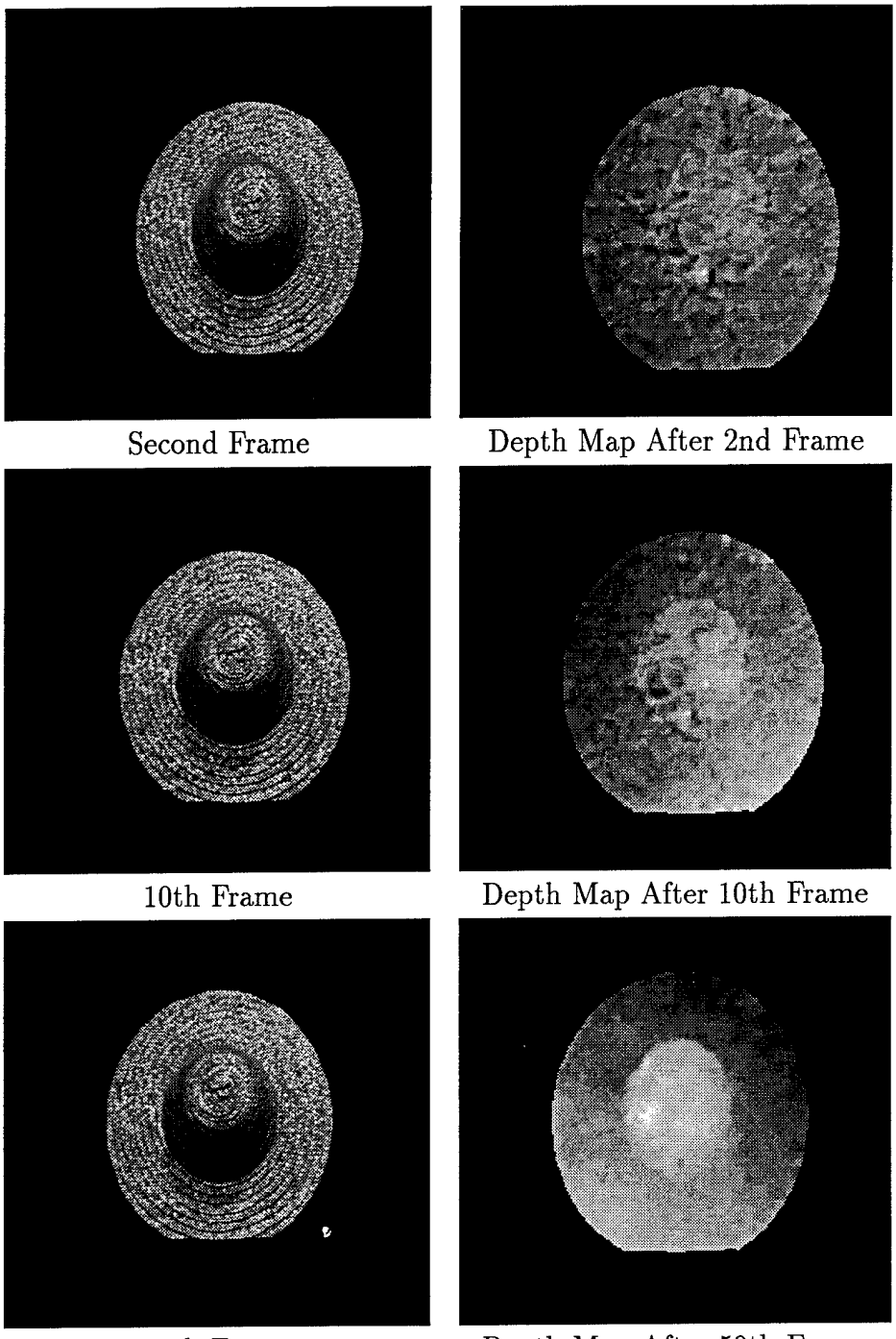
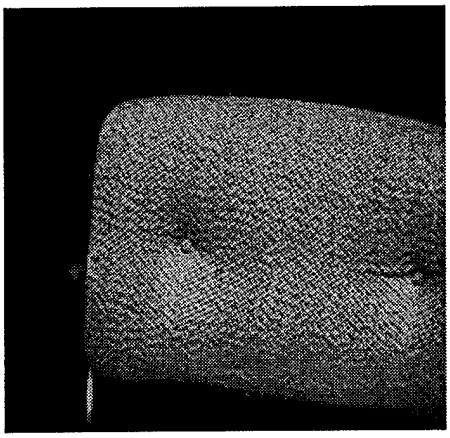
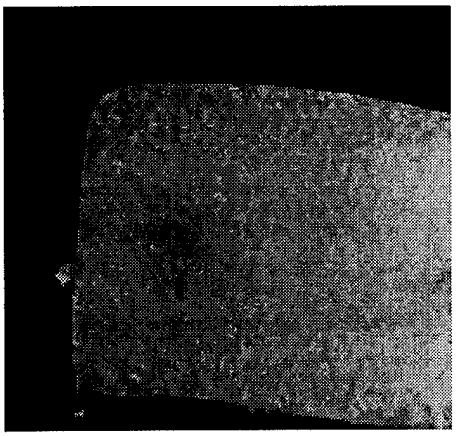


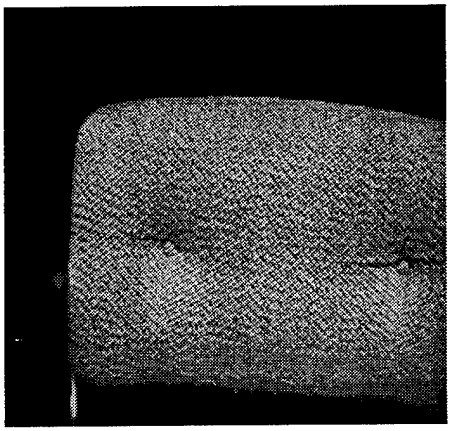
Figure 13: The Straw Hat Sequence



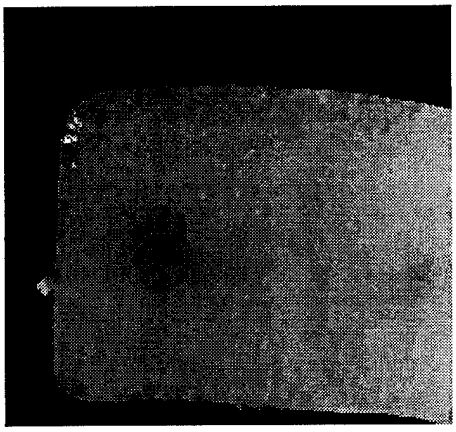
Second Frame



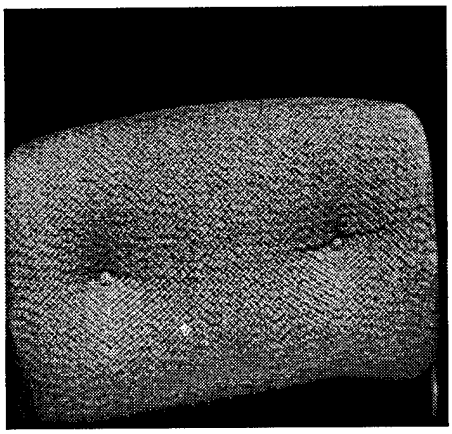
Depth Map After 2nd Frame



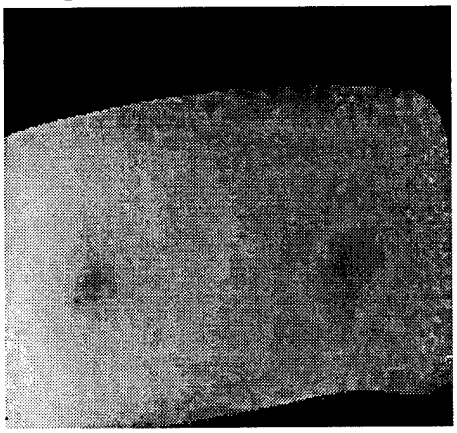
10th Frame



Depth Map After 10th Frame



50th Frame



Depth Map After 50th Frame

Figure 14: The Chair Sequence

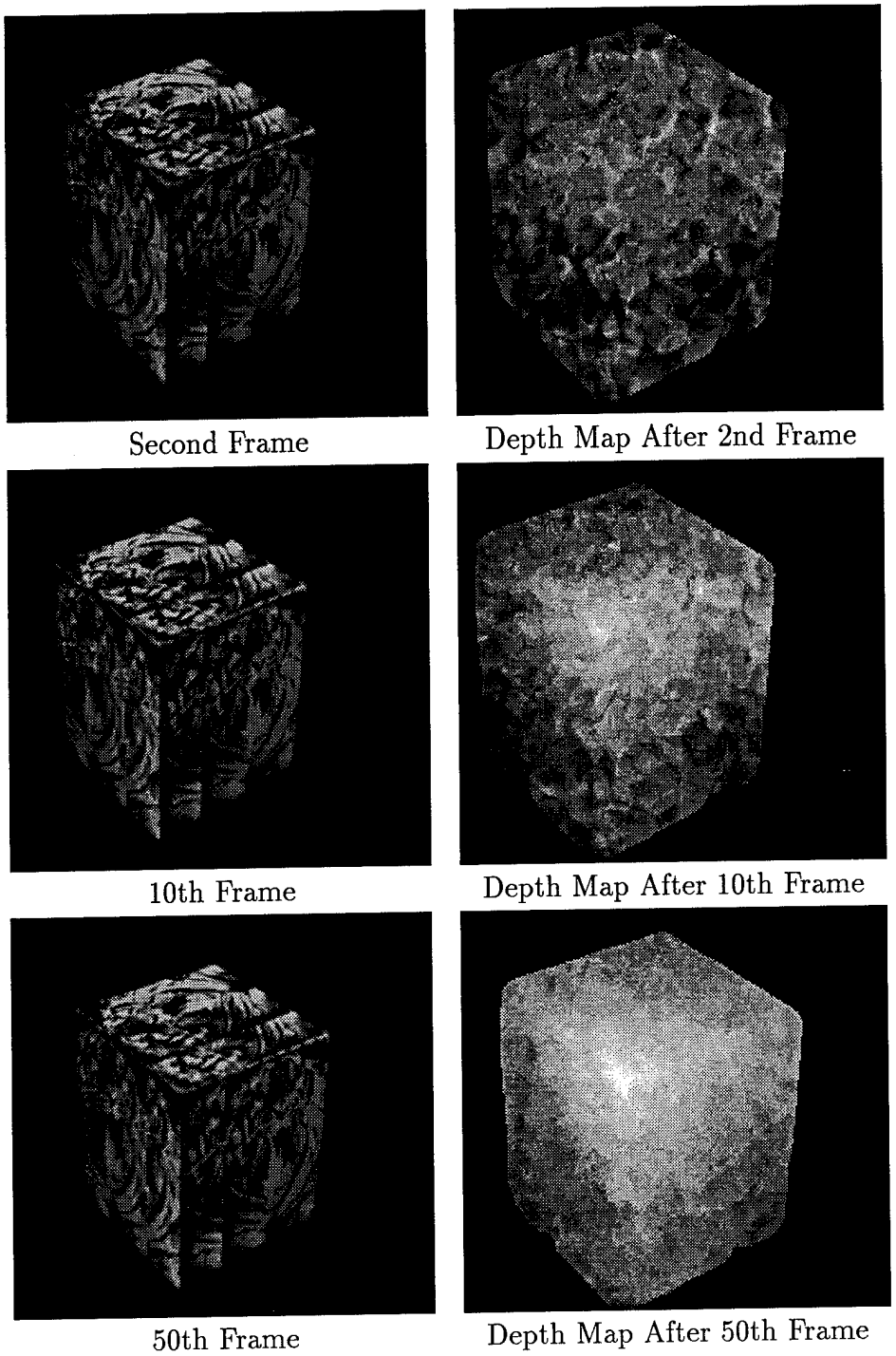
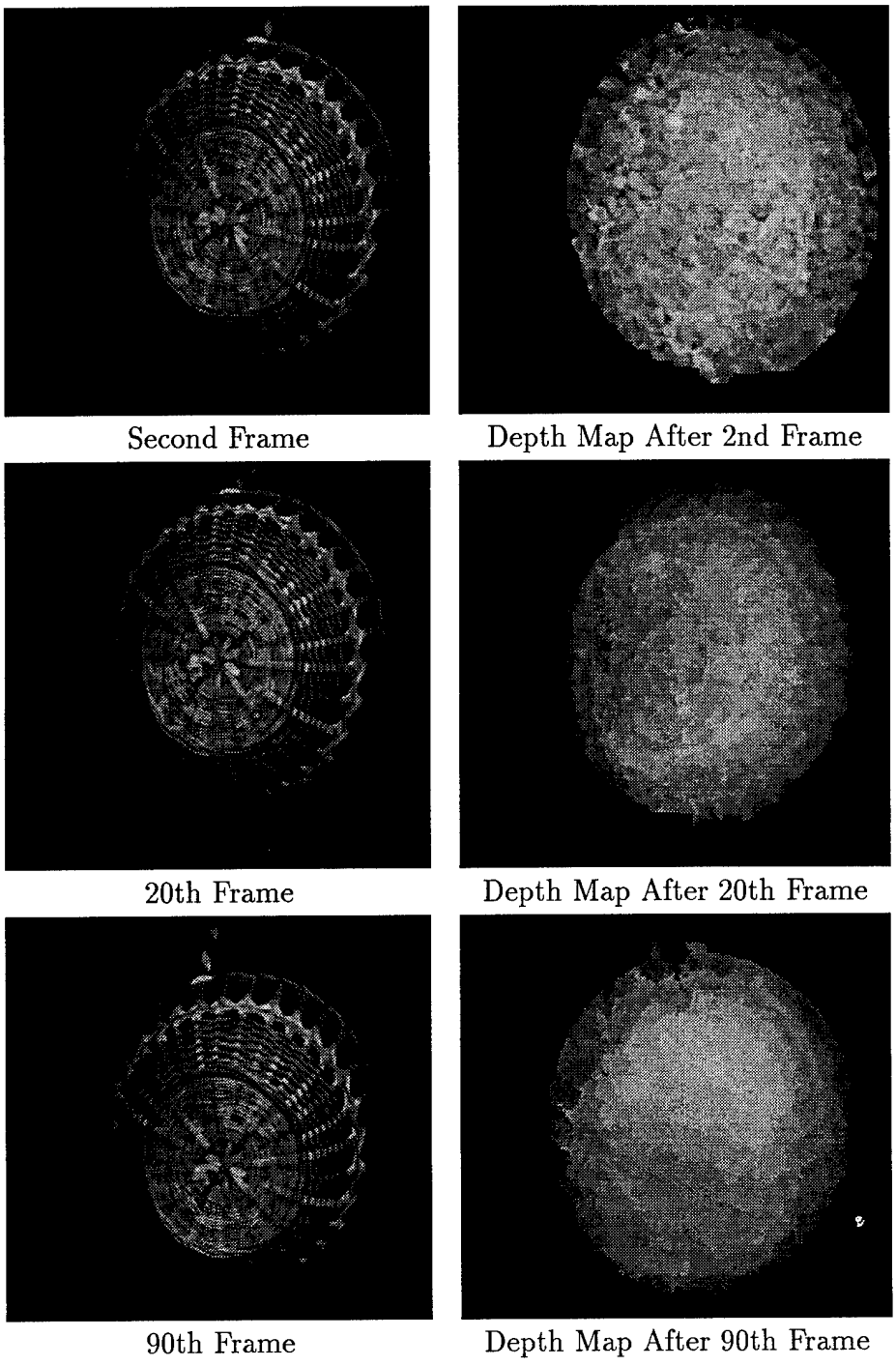


Figure 15: The Cube Sequence

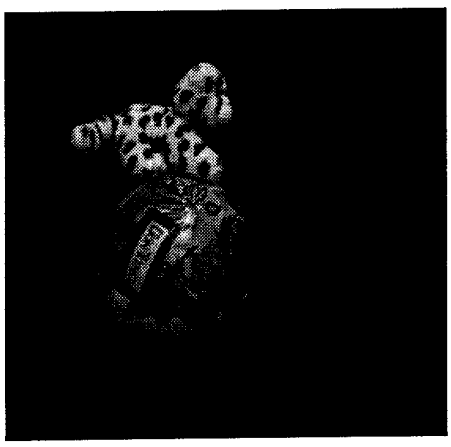


Second Frame | Depth Map After 2nd Frame

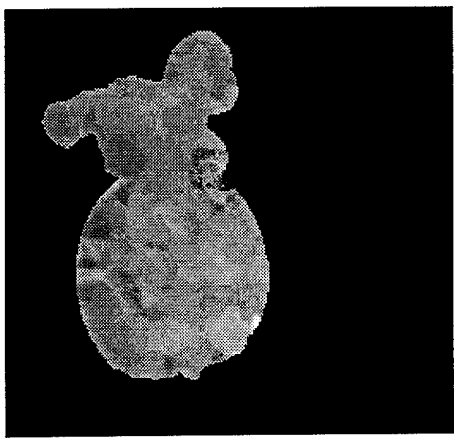
20th Frame | Depth Map After 20th Frame

90th Frame | Depth Map After 90th Frame

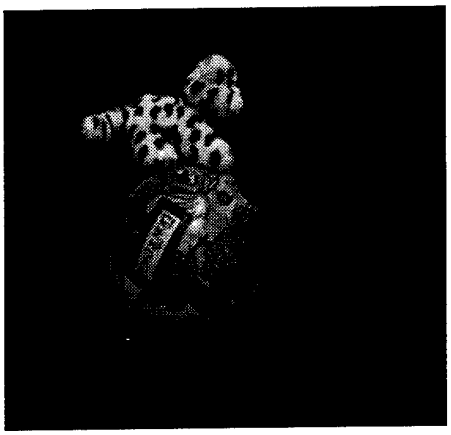
Figure 16: The Basket Sequence



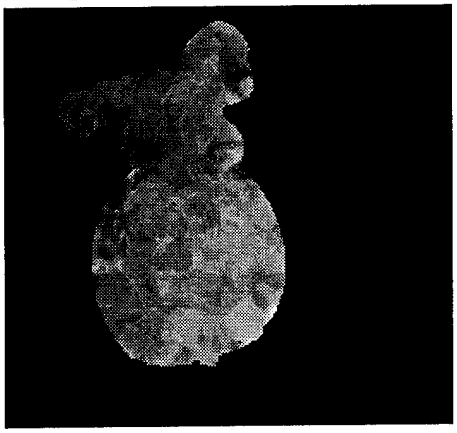
Second Frame



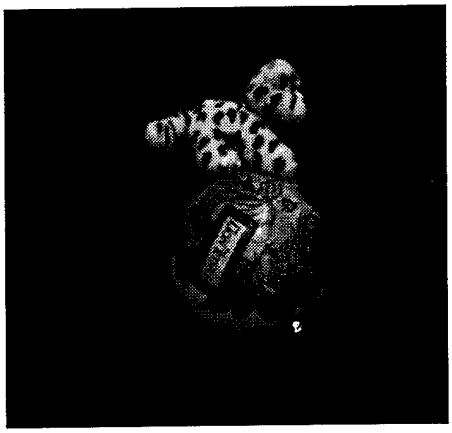
Depth Map After 2nd Frame



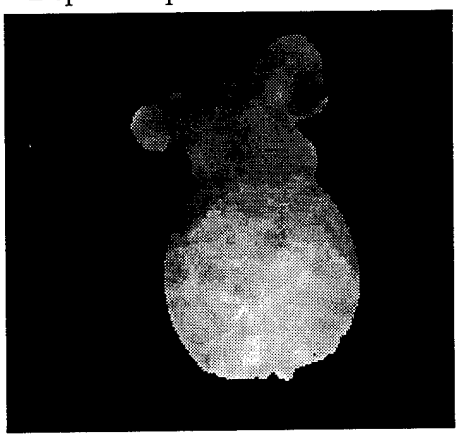
10th Frame



Depth Map After 10th Frame



50th Frame



Depth Map After 50th Frame

Figure 17: The Sphere and Dog Sequence

dense structure from a dense optical flow sequence can converge to the true 3D shape of the scene quickly and accurately. We have demonstrated its performance under adverse conditions such as noisy images, specularity, and texture aliasing. Even under these conditions, the system performed robustly.

7 Summary

In summary, we presented an EKF-based system which recursively combine dense structural information from a sequence of optical flows. At current stage, our system is able to deliver an evolving sequence of depth maps using optical flows. We also showed that the system was very robust when the optical flows were noisy or contain outliers caused by texture aliasing and specularity.

Current representation of 3D dense information by depth maps and their uncertainty is very limiting in that complicated objects can not be represented. In the future, we would like to expand our system to deliver a final 3D model of the scene based on the image sequence. In other words, we would like to maintain an independent module to store, retrieve and update 3D structural information. Therefore we could extract *a priori* depth information from the module for every optical flow frame, and merge *posteriori* depth information into the module. The problem of representing 3D dense structure and its uncertainty (independent and correlated) still remains to be very challenging.

Acknowledgement

Thanks to Keith Gremban for pointing us to Sherman-Morrison-Woodbury formula. This research is sponsored by the Department of Army, Army Research Office under grant number DAAH04-94-G-0006. The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing official policies or endorsements, either expressed or implied, of the Department of the Army or the United States Government.

A Sherman-Morrison-Woodbury Inversion

Given a full-rank $n \times n$ matrix $\mathbf{C}$, and its perturbed by a rank m matrix $\mathbf{U}\mathbf{V}^T$, where $\mathbf{U}$ and $\mathbf{V}$ are both $n \times m$ matrices, the Sherman-Morrison-Woodbury formula [9] (page 225) states that

$$(\mathbf{C} + \mathbf{U}\mathbf{V}^T)^{-1} = \mathbf{C}^{-1} - \mathbf{C}^{-1}\mathbf{U}(\mathbf{I}_m + \mathbf{V}^T\mathbf{C}^{-1}\mathbf{U})^{-1}\mathbf{V}^T\mathbf{C}^{-1}, \tag{53}$$

where $\mathbf{I}_m$ is the $m \times m$ unit matrix. The validity of this inverse can be easily verified by multiplying both sides by $(\mathbf{C} + \mathbf{U}\mathbf{V}^T)$.

In a more concise format, we can write the above equation as

$$(\mathbf{C} + \mathbf{U}\mathbf{V}^T)^{-1} = \mathbf{C}_1 + \mathbf{U}_1\mathbf{V}_1^T, \tag{54}$$

where

$$\begin{aligned} \mathbf{C}_1 &= \mathbf{C}^{-1}, & (55)\\ \mathbf{U}_1 &= -\mathbf{C}^{-1}\mathbf{U}(\mathbf{I}_m + \mathbf{V}^T\mathbf{C}^{-1}\mathbf{U})^{-1}, & (56)\\ \mathbf{V}_1 &= \mathbf{C}^{-T}\mathbf{V}. & (57) \end{aligned}$$

In our application, because $\mathbf{C}$ is a diagonal matrix, we can compute its inverse $\mathbf{C}_1$ accurately. Therefore the only source of numerical error is $\mathbf{A} = (\mathbf{I}_m + \mathbf{V}^T\mathbf{C}^{-1}\mathbf{U})^{-1}$. Suppose the error is

$$\mathbf{E} = \mathbf{I}_m - (\mathbf{I}_m + \mathbf{V}^T\mathbf{C}^{-1}\mathbf{U})\mathbf{A}, \tag{58}$$

the final error is

$$(\mathbf{C} + \mathbf{U}\mathbf{V}^T)(\mathbf{C}_1 + \mathbf{U}_1\mathbf{V}_1^T) - \mathbf{I} = \mathbf{U}\mathbf{E}\mathbf{V}^T\mathbf{C}^{-1}. \tag{59}$$

Eq. 59 shows that there is a potential danger that the error $\mathbf{E}$ could be magnified in the final error. That is the source of fragility of Sherman-Morrison-Woodbury formula. In our system, we reduce that risk by computing $\mathbf{A}$ is in double precision and limiting the magnification factor (roughly the ratio between eigenvalues of $\mathbf{U}\mathbf{V}^T$ and those of $\mathbf{C}$) to be less than 10^4 as we did in Section 6.

B Eigen Analysis of Symmetric Outer Products

In this section, we consider eigen analysis and singular value decomposition of a special kind of matrices, i.e. symmetric outer products of two low-rank matrices. Because those matrices are symmetric, the problem of computing eigenvalues and eigenvectors is identical to the problem of singular value decomposition because

$$\mathbf{U}\mathbf{V}^T = \begin{pmatrix} & & & \\ \vec{e}_1 & \vec{e}_2 & \cdots & \vec{e}_m \\ & & & \end{pmatrix} \begin{pmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_m \end{pmatrix} \begin{pmatrix} \vec{e}_1^T \\ \vec{e}_2^T \\ \vdots \\ \vec{e}_m^T \end{pmatrix}, \tag{60}$$

where $\mathbf{U}$ and $\mathbf{V}$ are both $n \times m$ ($n >> m$) matrices, $\lambda_i, i = 1, 2, \cdots, m$ are eigenvalues, and $\vec{e}_i, i = 1, 2, \cdots, m$ are normalized eigenvectors.

Theorem B.1 *Suppose $\lambda_i^0, i = 1, 2, \cdots, m$ and $\vec{e}_i^0, i = 1, 2, \cdots, m$ are eigenvalues and normalized eigenvectors of the $m \times m$ matrix $\mathbf{V}^T\mathbf{U}$, we have the eigenvalues and eigenvectors of $n \times n$ matrix $\mathbf{U}\mathbf{V}^T$ as*

$$\lambda_i = \lambda_i^0, \tag{61}$$

$$\vec{e}_i = \frac{\mathbf{U}\vec{e}_i^0}{\| \mathbf{U}\vec{e}_i^0 \|}, \tag{62}$$

where $\| \mathbf{U}\vec{e}_i^0 \|$ is the norm.

The proof of the above theorem is straightforward. Since λ_i^0 and $\vec{e}_i^0$ are an eigenvalue and eigenvector of $\mathbf{V}^T\mathbf{U}$, we have

$$\mathbf{V}^T\mathbf{U}\vec{e}_i^0 = \lambda_i^0\vec{e}_i^0. \tag{63}$$

Multiplying both sides by $\mathbf{U}$, we obtain

$$(\mathbf{U}\mathbf{V}^T)\mathbf{U}\vec{e}_i^0 = \lambda_i^0\mathbf{U}\vec{e}_i^0. \tag{64}$$

Thus we have the eigenvalue and eigenvector of $\mathbf{U}\mathbf{V}^T$ as λ_i^0 and $\mathbf{U}\vec{e}_i^0$.

Additionally, if the outer product $\mathbf{U}\mathbf{V}^T$ is also positive semi-definite, which is true if it represents covariance, we can rewrite it as

$$\mathbf{U}\mathbf{V}^T = \mathbf{B}\mathbf{B}^T, \tag{65}$$

where

$$\mathbf{B} = \begin{pmatrix} & & & \\ \vec{e}_1 & \vec{e}_2 & \cdots & \vec{e}_m \\ & & & \end{pmatrix} \begin{pmatrix} \sqrt{\lambda_1} & & & \\ & \sqrt{\lambda_2} & & \\ & & \ddots & \\ & & & \sqrt{\lambda_m} \end{pmatrix}. \tag{66}$$

References

[1] Golad Adiv. Determining three-dimensional motion and structure from optical flow generated by several moving objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 7(4):384–401, July 1985.

[2] Golad Adiv. Inherent ambiguities in recovering 3-D motion and structure from a noisy flow field. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(5):477–489, May 1989.

[3] A. Azarbayejani and Alex Pentland. Recursive estimation of motion, structure, and focal length. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1995. to appear.

[4] Edward J. Baranoski. Triangular factorization of inverse data covariance matrices. In *International Conference on Acoustics, Speech, and Signal Processing*, pages 2245–2248, Toronto, Canada, May 1991.

[5] Ted J. Broida, S. Chandrashekhar, and Rama Chellappa. Recursive estimation of 3D motion from a monocular image sequence. *IEEE Transactions on Aerosp. Electron. Syst.*, 26(4):639–656, July 1990.

[6] Anna R. Bruss and Berthold K. P. Horn. Passive navigation. *Computer Vision, Graphics and Image Processing*, 21:3–20, 1983.

[7] John J. Craig. *Introduction to Robotics: Mechanics and Control.* Addison-Wesley Publishing Company, Inc., second edition, 1989.

[8] Arthur Gelb, editor. *Applied Optimal Estimation.* The MIT Press, 1989.

[9] Gene Golub and James M. Ortega. *Scientific Computing: An Introduction with Parallel Computing.* Academic Press, Boston, 1993.

[10] William W. Hager. Updating the inverse of a matrix. *SIAM Review*, 31(2):221–239, June 1989.

[11] David J. Heeger and Allan D. Jepson. Subspace methods for recovering rigid motion I: Algorithm and implementation. *International Journal of Computer Vision*, 7(2):95–117, 1992.

[12] Joachim Heel. Dynamic systems and motion vision. Technical Report AI Memo 1037, MIT Artificial Intelligence Laboratory, 1988.

[13] Joachim Heel. Temporally integrated surface reconstruction. In *Proceedings of International Conference on Computer Vision*, pages 292–295, 1990.

[14] H. C. Longuet-Higgins and K. Prazdny. The interpretation of a moving retinal image. *Proc. R. Soc. Lond. B*, 208:385–397, 1980.

[15] Larry Matthies, Richard Szeliski, and Takeo Kanade. Kalman filter-based algorithms for estimating depth from image sequences. *International Journal of Computer Vision*, 3:209–236, 1989.

[16] William H. Press, Brian P. Flannery, Saul A. Teukolsky, and William T. Vetterling. *Numerical Recipes in C.* Cambridge University Press, 1988.

[17] J. Inigo Thomas and J. Oliensis. Incorporating motion error in multi-frame structure from motion. Technical Report COINS TR91-36, Computer and Information Science, University of Massachusetts at Amherst, 1991.

[18] Carlo Tomasi. Shape and motion from image streams: A factorization method. Technical Report CMU-CS-91-172, The School of Computer Science, Carnegie Mellon University, 1991.

[19] Roger Y. Tsai and Thomas S. Huang. Uniqueness and estimation of three-dimensional motion parameters of rigid objects with curved surfaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(1), January 1984.

[20] Juyang Weng, Narendra Ahuja, and Thomas Huang. Closed-form solution + maximum likelihood: A robust approach to motion and structure estimation. In *Proceedings of Computer Vision and Pattern Recognition*, pages 381–386, 1988.

[21] Yalin Xiong and Steven A. Shafer. Moment and hypergeometric filters for high precision computation of focus, stereo, and optical flow. Technical Report CMU-RI-TR-94-28, The Robotics Institute, Carnegie Mellon University, 1994.

[22] Yalin Xiong and Steven A. Shafer. Hypergeometric filters for optical flow and affine matching. In *International Conference on Computer Vision*, 1995.