Appeared in: ICCV '98, pp. 696-702, Bombay, India, Jan. 1998

A Unified Factorization Algorithm for Points, Line Segments and Planes with Uncertainty Models

Daniel D. Morris and Takeo Kanade
Robotics Institute, Carnegie Mellon University, Pittsburgh, PA 15213-3890

Abstract

In this paper we present a unified factorization algorithm for recovering structure and motion from image sequences by using point features, line segments and planes. This new formulation is based on directional uncertainty model for features. Points and line segments are both described by the same probabilistic models and so can be recovered in the same way. Prior information on the coplanarity of features is shown to fit naturally into the new factorization formulation and provides additional constraints for the shape recovery. This formulation leads to a weighted least squares motion and shape recovery problem which is solved by an efficient quasi-linear algorithm. The statistical uncertainty model also enables us to recover uncertainty estimates for the reconstructed three dimensional feature locations.

1 Introduction

We address the structure from motion problem: given an image sequence of a scene or rigid object taken by a camera undergoing unknown motion, reconstruct the three dimensional (3D) geometry of the scene. While we also recover camera motion, in this report we focus on 3D structure. Further, we require that the solution be robust to the inevitable imprecision and small errors involved in imaging and registration.

Early structure from motion work focused on the two image problem [5, 17]. Starting from a set of corresponding features in the images, the camera motion and point locations could be recovered. Since then research has shifted to the more difficult multiple-image problem where more data samples can lead to greater accuracy which is our motivation.

Much current work focusses on recovering shape using a full perspective camera model [1, 3, 14, 15, 20]. Solving the perspective equations typically entails non-linear minimization and the usual problems of many local minima, and high computational expense, with convergence and accuracy being sensitive to special parameters and initial guesses. Noting this and that even the perspective model is only a rough approximation when effects like radial distortion are large, we follow [10, 12, 16] and side-step many of these problems by working with an affine camera model. Very accurate results can still be obtained provided the perspective distortion effects are small.

Batch factorization methods have typically weighted features uniformly, or as in [9], allowed confidence weightings. Registration algorithms can often provide directional uncertainty measures for feature location and here we formulate an algorithm to incorporate these directional uncertainties. We note that a significant missing aspect of current methods is their lack of confidence measure in their data and results. Our algorithm provides an uncertainty measure for the resulting shape features.

Recent work with line features [4, 12, 18] illustrates that, while lines provide valuable constraints for shape recovery, the reconstruction process using them is more sensitive to noise and being trapped by local minima than for point features. Further the line factorization algorithm [12] requires at least seven directions for lines in the scene, a significant practical problem. Our uncertainty formulation provides a natural method for modeling line segments, thus avoiding the above difficulties and, in particular, allowing us to jointly recover shape from point and line-segment features. We also extend the algorithm to incorporate multiple planar constraints.

We begin by summarizing the factorization algorithm in Section 2. Then, in Section 3, we describe our uncertainty model and the Bilinear algorithm for points, lines and planes. Next we present results from synthetic and real image sequences in Section 4 and we end with the conclusions.

2 Factorization Algorithm Overview

We provide a brief overview of the Factorization algorithm found in more detail elsewhere [10, 12, 16]. The algorithm is feature-based, where a feature is a distinctive part of an image, such as a corner, edge or mark on a rigid object. A set of P features in F images

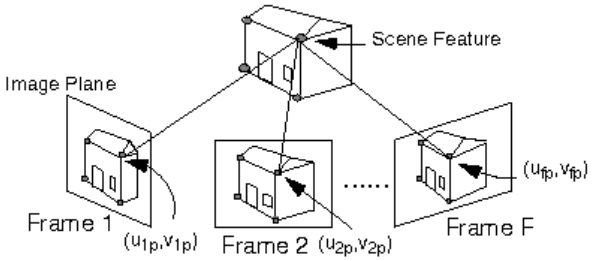

Figure 1: Feature tracked in an image sequence

with coordinates $\{\mathbf{w}'_{fp} = (u'_{fp}, v'_{fp}) | f = 1, \ldots, F, p = 1, \ldots, P\}$ are tracked in the image sequence as illustrated in Figure 1. Feature coordinates are transformed to object coordinates by subtracting their center of mass: $\mathbf{w}'_{fp}$ is replaced by $\mathbf{w}_{fp} = \mathbf{w}'_{fp} - \mathbf{w}_{fc}$ for all f and p, where $\mathbf{w}_{fc} = \sum_p \mathbf{w}_{fp}/P$. A measurement matrix, W, is created containing the relative feature coordinates:

$$W = \begin{bmatrix} u_{11} & \cdots & u_{1P} \\ \cdots & \cdots & \cdots \\ u_{F1} & \cdots & u_{FP} \\ v_{11} & \cdots & v_{1P} \\ \cdots & \cdots & \cdots \\ v_{F1} & \cdots & v_{FP} \end{bmatrix}$$

Assuming orthography, Tomasi and Kanade [16] showed that W is rank 3 or less, and can be factored into the product of a motion matrix, M, and shape matrix, S, i.e.,

$$W = MS \tag{1}$$

where

$$M = \begin{bmatrix} \mathbf{m}_1^T \\ \cdots \\ \mathbf{m}_F^T \\ \mathbf{n}_1^T \\ \cdots \\ \mathbf{n}_F^T \end{bmatrix}_{2F \times 3} \quad \text{and } S = \begin{bmatrix} \mathbf{s}_1 & \cdots & \mathbf{s}_P \end{bmatrix}_{3 \times P}$$

The rows of M encode the rotation for each frame, and the columns of S contain the 3D position of each feature in the reconstructed object.

When there are errors or noise in the measurement matrix, we seek a least squares approximation to it. We define an error function:

$$E_{SVD}(M, S) = \|W - MS\|^2 = \sum_{f,p} \epsilon_{fp}^T \epsilon_{fp}, \tag{2}$$

where $\epsilon_{fp} = \begin{bmatrix} u_{fp} - \mathbf{m}_f^T \mathbf{s}_p \\ v_{fp} - \mathbf{n}_f^T \mathbf{s}_p \end{bmatrix}$, which we seek to minimize. This can be achieved by performing Singular Value Decomposition (SVD) on the measurement matrix, $W = U\Sigma V$, and zeroing all but the three largest singular values in Σ to get $\hat{W} = U\hat{\Sigma}V$. Factoring this

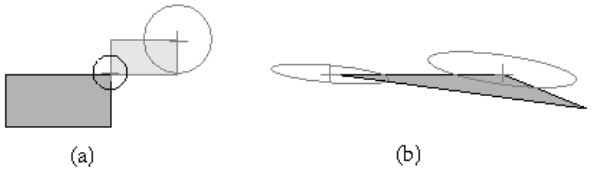

Figure 2: Feature uncertainty indicated by ellipses

provides least squares estimates for motion and shape: $\hat{M} = U\hat{\Sigma}^{1/2}$, and $\hat{S} = \hat{\Sigma}^{1/2}V^T$. We note that $\hat{M}$ and $\hat{S}$ are defined only up to an affine transformation, but we can obtain Euclidean coordinates by applying appropriate constraints, depending on the projection model, to the rows of $\hat{M}$ as in [16]. From here on we drop the "$\hat{}$" symbol and assume this step has been performed.

3 Bilinear Factorization Algorithm

To improve the standard factorization algorithm, we propose using a more general Gaussian feature uncertainty model and derive our Bilinear algorithm for this model extending a similar technique in [9].

3.1 Directional Uncertainty Formulation

Since feature tracking and registration algorithms do not provide exact values for feature positions, we model the actual feature position, $\mathbf{x}_{fp}$, with a 2D Gaussian probability density function:

$$\rho_{fp}(\mathbf{x}_{fp}) = k_{fp} \exp(-\frac{1}{2}(\mathbf{w}_{fp} - \mathbf{x}_{fp})^T G_{fp} (\mathbf{w}_{fp} - \mathbf{x}_{fp})) \tag{3}$$

where G_{fp} is the inverse covariance and $\mathbf{w}_{fp}$ the mean. The covariance determines an ellipse of equal probability density, and the major and minor axes give the directional uncertainty in feature location as indicated in Figure 2. Assuming independence of feature densities, the total probability density function for P features in F images will be the product of all the individual density functions:

$$\rho_T = \prod_{fp} \rho_{fp}(\mathbf{x}_{fp}). \tag{4}$$

The maximum likelihood solution to this is obtained by minimizing the cost function, E_B, given by:

$$E_B = \sum_{fp} \frac{1}{2}(\mathbf{w}_{fp} - \mathbf{x}_{fp})^T G_{fp} (\mathbf{w}_{fp} - \mathbf{x}_{fp}) \tag{5}$$

To recover shape and motion, we assume a 2×3 affine projection model, M_f, for each image acting on the 3D object centered point coordinates $\mathbf{s}_p = \begin{bmatrix} s_{px} & s_{px} & s_{pz} \end{bmatrix}$. Thus, feature locations are constrained by the following equation:

$$\mathbf{x}_{fp} = M_f \mathbf{s}_p = \begin{bmatrix} \mathbf{m}_f^T \mathbf{s}_p \\ \mathbf{n}_f^T \mathbf{s}_p \end{bmatrix} \tag{6}$$

We note that, when this constraint is substituted into (5) and the covariance matrix is identity, the cost E_B reduces to the standard factorization cost, E_{SVD}, showing that SVD factorization is a special case of our algorithm.

3.2 Bilinear Minimization

The maximum likelihood solution for shape and motion is obtained by minimizing E_B with respect to the shape and motion parameters. We note that the partial derivatives of E_B are bilinear in these parameters: $\mathbf{m}_f$, $\mathbf{n}_f$ and $\mathbf{s}_p$:

$$\begin{aligned}
\frac{\partial E_B}{\partial \mathbf{m}_f} &= \sum_p -c_{fp}(u_{fp} - \mathbf{m}_f^T\mathbf{s}_p)\mathbf{s}_p - d_{fp}(v_{fp} - \mathbf{n}_f^T\mathbf{s}_p)\mathbf{s}_p \\
\frac{\partial E_B}{\partial \mathbf{n}_f} &= \sum_p -d_{fp}(u_{fp} - \mathbf{m}_f^T\mathbf{s}_p)\mathbf{s}_p - e_{fp}(v_{fp} - \mathbf{n}_f^T\mathbf{s}_p)\mathbf{s}_p \quad (7) \\
\frac{\partial E_B}{\partial \mathbf{s}_p} &= \sum_f -c_{fp}(u_{fp} - \mathbf{m}_f^T\mathbf{s}_p)\mathbf{m}_f - d_{fp}(u_{fp} - \mathbf{m}_f^T\mathbf{s}_p)\mathbf{n}_f - \\
&\quad d_{fp}(v_{fp} - \mathbf{n}_f^T\mathbf{s}_p)\mathbf{m}_f - e_{fp}(v_{fp} - \mathbf{n}_f^T\mathbf{s}_p)\mathbf{n}_f
\end{aligned}$$

Here $G_{fp} = \begin{bmatrix} c_{fp} & d_{fp} \\ d_{fp} & e_{fp} \end{bmatrix}$ is positive definite and is split into its components. By setting the partials to zero and rearranging the resulting equations we obtain a set of bilinear equations for motion and shape:

$$\begin{bmatrix} \sum_p c_{fp}\mathbf{s}_p\mathbf{s}_p^T & \sum_p d_{fp}\mathbf{s}_p\mathbf{s}_p^T \\ \sum_p d_{fp}\mathbf{s}_p\mathbf{s}_p^T & \sum_p e_{fp}\mathbf{s}_p\mathbf{s}_p^T \end{bmatrix} \begin{bmatrix} \mathbf{m}_f \\ \mathbf{n}_f \end{bmatrix} = \begin{bmatrix} \sum_p (c_{fp}u_{fp} + d_{fp}v_{fp})\mathbf{s}_p \\ \sum_p (d_{fp}u_{fp} + e_{fp}v_{fp})\mathbf{s}_p \end{bmatrix} \quad (8)$$

$$\left[\sum_f c_{fp}\mathbf{m}_f\mathbf{m}_f^T + d_{fp}(\mathbf{m}_f\mathbf{n}_f^T + \mathbf{n}_f\mathbf{m}_f^T) + e_{fp}\mathbf{n}_f\mathbf{n}_f^T\right][\mathbf{s}_p] = \left[\sum_f c_{fp}u_{fp}\mathbf{m}_f + d_{fp}(u_{fp}\mathbf{n}_f + v_{fp}\mathbf{m}_f) + e_{fp}v_{fp}\mathbf{n}_f\right] \quad (9)$$

for $p = 1, \ldots, P$ and $f = 1, \ldots, F$. We start with an initial guess for motion M_g, and use equation (9) to obtain the least squares solution for S, and then this new S and equation (8) to solve for the least squares motion M. This step is repeated iteratively until convergence is achieved. We call this our Bilinear Factorization Algorithm.

The complexity of each iteration is $O(FP)$ operations, compared to SVD which, from [2], has complexity $O(FP^2)$ for $F > P$ or $O(PF^2)$ otherwise. The Bilinear algorithm monotonically reduces the cost function but it is not guaranteed to converge to the global minimum, hence we usually start with reasonable estimates for motion or shape such as obtained from the SVD solution which is a global minimum of E_{SVD}. With this estimate our experiments typically converged within a few iterations.

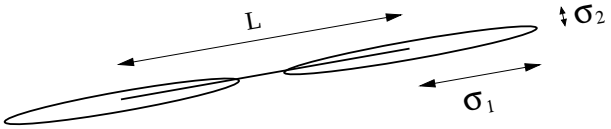

Figure 3: Line segment feature uncertainty

3.3 Line Segments

Recent work has extended the factorization algorithm to infinite lines [12], but this involves more stages and is more sensitive to noise than the point feature algorithm. The primary reason for this sensitivity is that each line provides a single constraint, whereas point features provide two constraints. Practical line registration algorithms actually work with line segments which are fit to edges the images. The positions of the recovered line endpoints may correspond to different parts of the edge in any given image, but they are restricted to fall within the physical limits of the object. Thus, by using line segments which we know correspond to a particular region of the infinite line, we can extract more constraint information than is possible for infinite lines. We define line segments by their end-points and model each end-point as a Gaussian density function with large uncertainty along the length of the line and small uncertainty perpendicular to the line, as illustrated in Figure 3. Line segments defined by their end-point density functions have the same form as point features and so can be used directly by the Bilinear algorithm on their own or in conjunction with point features.

3.4 Shape Uncertainty

We seek an estimate for the uncertainty in the recovered 3D positions of the features by the factorization algorithm. We model second order error terms with 3D Gaussian distributions around the maximum likelihood solution to ρ_T from (4). Our inverse covariance for each feature will be the Hessian of E_B in the shape parameters which we obtain as:

$$H_p = \begin{bmatrix} \frac{\partial^2 E_B}{\partial s_{p1}^2} & \frac{\partial^2 E_B}{\partial s_{p2}\partial s_{p1}} & \frac{\partial^2 E_B}{\partial s_{p3}\partial s_{p1}} \\ \frac{\partial^2 E_B}{\partial s_{p1}\partial s_{p2}} & \frac{\partial^2 E_B}{\partial s_{p2}^2} & \frac{\partial^2 E_B}{\partial s_{p3}\partial s_{p2}} \\ \frac{\partial^2 E_B}{\partial s_{p1}\partial s_{p3}} & \frac{\partial^2 E_B}{\partial s_{p2}\partial s_{p3}} & \frac{\partial^2 E_B}{\partial s_{p3}^2} \end{bmatrix} = \sum_f M_f^T G_{fp} M_f \quad (10)$$

For this to be an exact and full description of the error probability density, our equations for shape would need to be linear in measurement values, and individual features must be independent of each other, i.e. the equations could be expressed in the form: $\mathbf{s}_p = \sum_f A_f\mathbf{w}_{fp} + b_p \ \forall p$ with some constants A_f and b_p. However, our equations, (8) and (9), are bilinear

in shape and motion, resulting in coupling between motion and shape, and hence coupling also between feature positions. To second order this coupling is described by the cross terms of the large covariance matrix of the shape vector, $\mathbf{s}_v$, formed by stacking the columns of S vertically. With no inter-feature correlation this would be a 3×3 block diagonal matrix, but with coupling there will be non-zero cross terms. By performing multiple experiments we can estimate the value of these cross-terms as well as the accuracy of our feature covariance estimate H_p^{-1}. Our experiments indicated that in some instances, depending on the motion and shape of the object, the block diagonal dominate the cross-terms, but in others there were significant cross-correlation effects, see Figure 4(a) and (b). For the purpose of this report we ignore these cross-terms but note that, depending on the application of our uncertainty models, these terms may or may not be of significance. Independent of the cross-terms, however, we found that equation (10) accurately predicted the individual feature covariance terms as shown in Figure 4(c).

3.5 Planes

A plane can be determined using three non-collinear points $\{\mathbf{s}_1, \mathbf{s}_2, \mathbf{s}_3\}$. Points on the plane can all be written as a linear combination of the difference vectors:

$$\begin{aligned} \mathbf{s}_p &= \mathbf{s}_1 + a_p(\mathbf{s}_2 - \mathbf{s}_1) + b_p(\mathbf{s}_3 - \mathbf{s}_1) \\ &= \begin{bmatrix} (\mathbf{s}_2 - \mathbf{s}_1) & (\mathbf{s}_3 - \mathbf{s}_1) & \mathbf{s}_1 \end{bmatrix} \begin{bmatrix} a_p \\ b_p \\ 1 \end{bmatrix} \end{aligned} \quad (11)$$

with some constants a_p and b_p defining the position on the plane.

Multiple points in a plane can thus be factored into two matrices:

$$S = RP = \begin{bmatrix} \mathbf{r}_1 & \mathbf{r}_2 & \mathbf{t} \end{bmatrix} \begin{bmatrix} a_1 & a_2 & \cdots & a_p \\ b_1 & b_2 & \cdots & b_p \\ 1 & 1 & \cdots & 1 \end{bmatrix} \quad (12)$$

The vectors $\mathbf{r}_1$ and $\mathbf{r}_2$ are parallel to the plane and $\mathbf{t}$ is a point on the plane. The equation for a point, $\mathbf{p}$, on the plane is thus:

$$(\mathbf{p} - \mathbf{t}) \cdot (\mathbf{r}_1 \times \mathbf{r}_2) = 0. \quad (13)$$

When the shape matrix contains points from more than one plane, points from each plane can be gathered into blocks and factored separately. For Q planes the global factorization equation becomes:

$$W = MS = M \begin{bmatrix} R_1P_1 & R_2P_2 & \cdots & R_QP_Q \end{bmatrix} \quad (14)$$

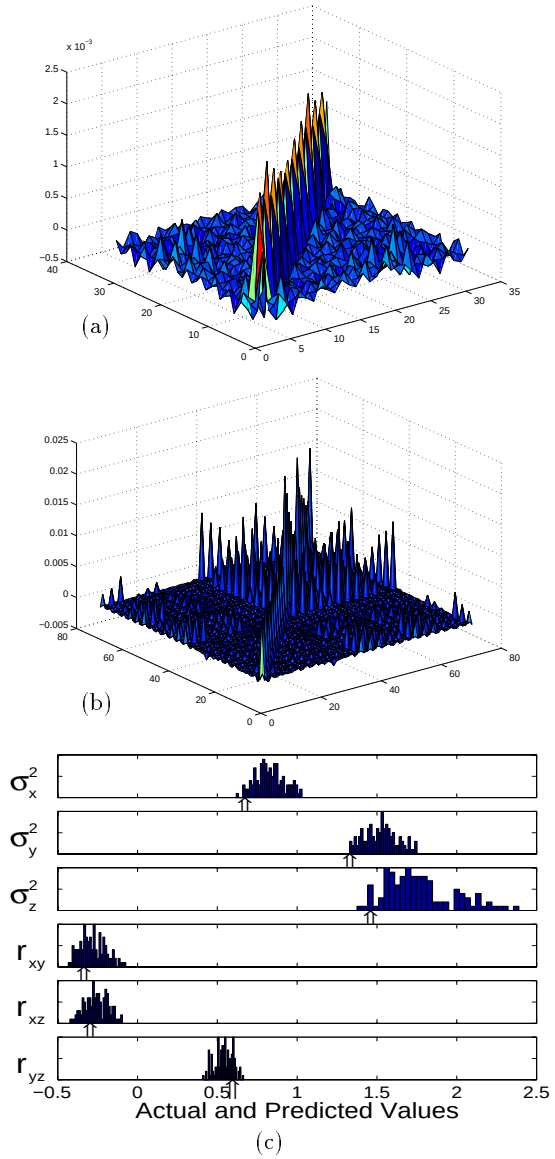

Figure 4: Error distribution of reconstructed shape data of the synthetic cube from 1000 runs with equal, uncorrelated Gaussian noise added to all image features. (a) Part of the covariance matrix for $\mathbf{s}_v$ illustrating the 3×3 block diagonal terms dominating cross-correlation elements. Here there is 60 degrees pitch and yaw rotation. (b) Part of the covariance matrix for the same object undergoing different motion (25 degree rotation around each axis) illustrating large cross terms. (c) A histogram of the actual values of the elements in the 3×3 covariance matrices for each feature, with the values predicted by equation (10) marked with a "⇑" symbol, showing close agreement. (The variances, σ_{ij}^2, are scaled by 10^3, and the r_{ij} terms are the standard correlation coefficients)

Equation (14) is trilinear in motion, plane and position parameters. While it could be solved with a trilinear set of equations analogous to equations (8) and (9), we choose to break it into two bilinear steps. In the first step we directly apply the Bilinear algorithm to solve for motion and shape. In the next step we define an error function, E_{Pi}, for each plane, P_i:

$$E_{Pi} = \sum_{p \in P} \frac{1}{2}(\mathbf{s}_p - R_i q_p)^T H_p (\mathbf{s}_p - R_i q_p) \qquad (15)$$

where q_p are the columns of P_i, and H_p is the inverse covariance from (10). This error is minimized with a similar set of bilinear equations to (8) and (9). The parameter $\mathbf{t}_i$ can be either solved for in these equations, or calculated directly as a weighted center of mass of the points in a plane. This algorithm differs from simply fitting planes to the resulting shape in that the camera motion as well as uncertainty estimates are used to guide the fitting using the inverse covariance matrices H_p. The two bilinear steps can be iterated until convergence is achieved.

3.6 Object Centered Coordinates

Past work generally used the center of mass as the origin for the object coordinate frame which, for equal density features, is the minimum variance point. Here we have covariance estimates for each feature in each frame so we seek an origin, $\mathbf{w}_{fc} = \sum_p a_p \mathbf{w}_{fp}$, which is a weighted sum of individual features in an image, that has minimum total variance. Since the relative orientation of features between images is unknown, we use the non-directional radial variance, σ^2_{fpr}, for each feature which is equal to the sum of the principal components of its covariance matrix, G^{-1}_{fp}, i.e.: $\sigma^2_{fpr} = \sigma^2_{fp1} + \sigma^2_{fp2}$. Solving for the minimum variance point we obtain the following coefficient values (see [8]):

$$a_p = \left(\sum_{j=1}^{P} \frac{\sum_{i=1}^{F} (\sigma^2_{ip1} + \sigma^2_{ip2})}{\sum_{i=1}^{F} (\sigma^2_{ij1} + \sigma^2_{ij2})} \right)^{-1} . \qquad (16)$$

Essentially this amounts to weighting each feature by the inverse of its total uncertainty in all frames. We note that this uncertainty involves the sum of the directional uncertainties: σ^2_{fp1} and σ^2_{fp2}, and hence features with a large dominant uncertainty, such as a line segment, will contribute far less than point features having small variances in both directions. This indicates that line segments are generally inferior to point features for determining object centered coordinates and so provides added motivation for combining line-segment and point-feature shape recovery.

4 Experiments

We report results for a synthetic shape, and then for a real image sequence of a cube. A synthetic cube was used to create image sequences to which Gaussian noise was added with standard deviation chosen randomly from the range: $0 \leq \sigma_1, \sigma_2 \leq 0.1$ which is up to 5 percent of the size of the object. The sequence consisted of 20 frames and rotations of about 50 degrees around each axis. The SVD and the Bilinear algorithms were compared based on their respective reconstructions from this data. Figure 5 shows sample reconstructions by both algorithms and compares planes fit to the SVD reconstruction, with planar factorization. Parts (e) and (f) plot the total error between the recovered and true shape for point and planar reconstruction over a sequence of runs. The Bilinear algorithm, which uses knowledge of the covariance in its reconstruction, gives better results for both of these cases than the SVD algorithm.

Next, we performed our shape recovery algorithm on a sequence of 11 images of a cube, the first and last being shown in Figure 6. Features are represented as square templates that are tracked in the image sequence with a warpable Sum of Squared Difference technique [6, 8, 13]. Here we permitted skewing of the templates in the minimization rather than full affine warping. We took the positional Hessian of each template error function to be our estimate for the inverse covariance of each feature obtaining the ellipses shown in the figure. The large ellipses indicate poorly located features. Figure 7 shows the Bilinear shape reconstruction with ellipsoids representing the relative uncertainty of features. By fitting planes to the SVD and Bilinear cube sides, we estimate that the SVD results have 15 percent larger error than the Bilinear.

Figure 8 shows the end-points of the automatically tracked line segments with ellipses representing positional uncertainty for each end-point. Here our uncertainty estimates were crude; we simply choose the variance along the line to be proportional to the line length and the variance perpendicular to the line to be inversely proportional to the length, since we expect to localize longer lines more accurately. We plan to do better in the future by characterizing the line tracking errors, but, despite this rough estimate, the results were good. The recovered shape is shown in Figure 9 with each end-point being represented by an ellipsoid corresponding to its 3D covariance, and corresponding features are joined to form the recovered line segments. To test the robustness of the algorithm to changes in line segment lengths, we added noise along the line direction to the end-points and compared the

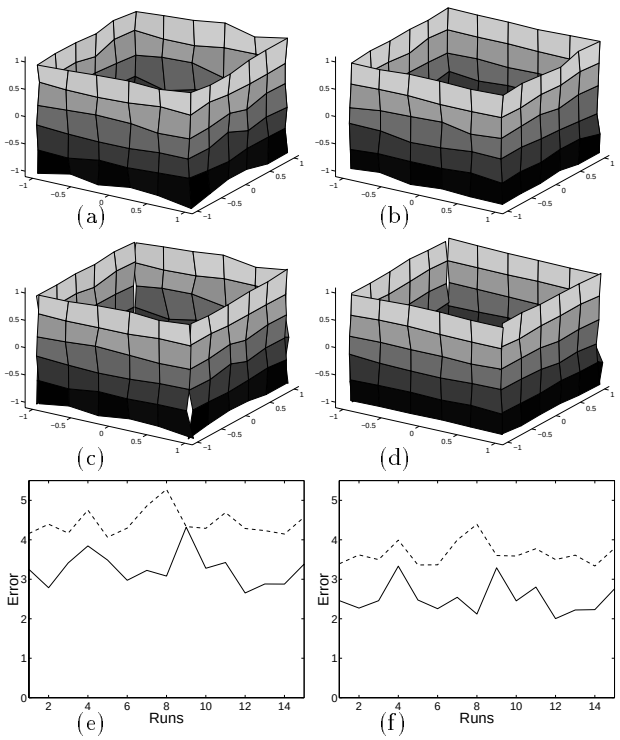

Figure 5: Recovered shapes with (a) SVD, (b) Bilinear, (c) fitting planes to SVD, and (d) planar factorization. (e) Shows Factorization, and (f) Planar factorization 3D errors for a sequence of runs comparing SVD (dashed) and Bilinear (solid) reconstruction.

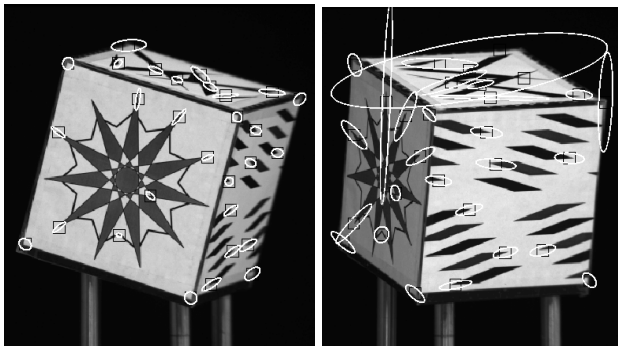

Figure 6: First and last images in sequence with feature uncertainty estimates

reconstruction results using SVD and the Bilinear algorithm as shown in Figure 10. Part (c) shows the improvement gained in this case when point features are used to help determine the object centered coordinates. Point and line segment reconstruction are easily combined and the result is illustrated in Figure 11.

Figure 7: Reconstructed point features with exaggerated uncertainty ellipsoids. (a) (left) is a view of the left face, (b) (right) is a top view.

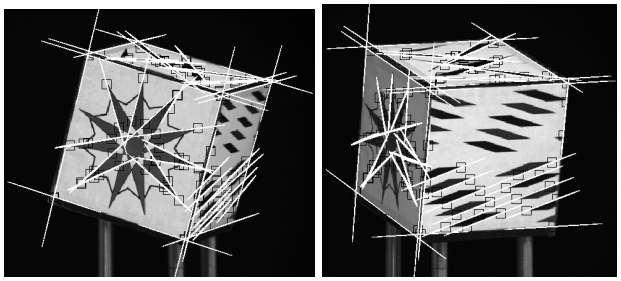

Figure 8: Ellipses modeling line segments

5 Conclusion

Our new Bilinear factorization algorithm uses a 2D Gaussian model for points and line segments, and, along with planar constraints, unifies reconstruction from points, line segments and planes. The computation for each iteration is very fast, $O(FP)$, and convergence rapid.

Shape recovery accuracy is improved by incorporating directional uncertainty measures that weight each feature's constraints by our confidence in them. Thus the algorithm can utilize a wider spectrum of features than previous algorithms to raise accuracy. The 3D uncertainty estimates that are generated can be used in subsequent surface-modeling steps or in active vision applications.

Further work we would like to do includes investigating the determining factors for correlation between features and the extension of uncertainty modeling from the affine to perspective case. We also hope to investigate better methods to estimate feature uncertainty in the tracking process.

Acknowledgements

The automatic line tracker provided by Naoki Chiba is greatly appreciated.

References

[1] A. Azarbayejani B. Horowitz, and A. Pentland, "Recursive estimation of structure from motion using relative orientation constraints," *Proc. IEEE Conf. Computer Vision Pattern Recognition (CVPR'93),* pp. 294-299, June 1993.

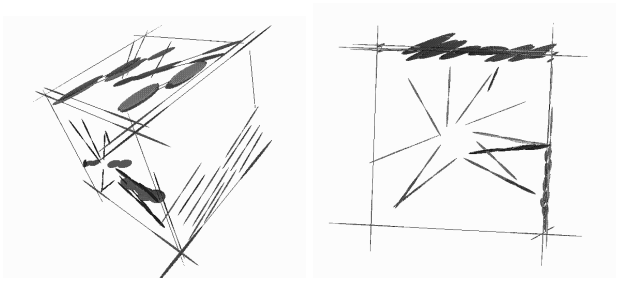

Figure 9: Reconstructed line segments with ellipsoids giving uncertainty of endpoints

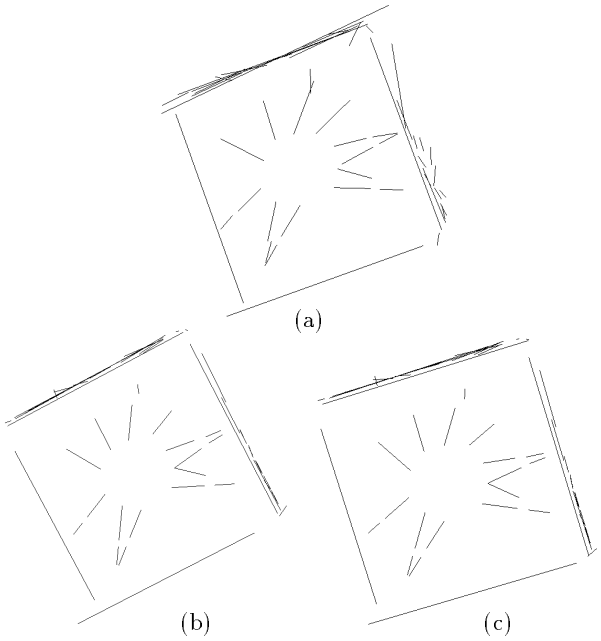

Figure 10: Cube reconstruction using line segments with ±20 percent line length variation at each endpoint. (a) SVD result, (b) Bilinear line segment result, and (c) Bilinear line segment result but with point features aiding in determining object coordinates

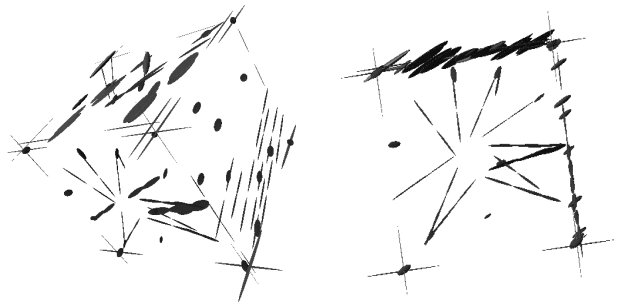

Figure 11: Joint point and line segment reconstruction

[2] P. Comon and G. H. Golub, "Tracking a Few Extreme Singular Values and Vectors in Signal Processing," *Proc. of IEEE,* vol. 78, No. 8, pp. 1327-1343 Aug. 1990.

[3] N. Cui, J. Weng and P. Cohen, "Extended Structure and Motion Analysis from Monocular Image Sequences," *IEEE Proc. Third Int. Conf. Computer Vision (ICCV'90),* pp. 222-29, Osaka, Japan, Dec. 1990.

[4] O. Faugeras, F. Lustman and G. Toscani, "Motion and Structure from Motion from Point and Line Matches," *ICCV '87,* pp. 25-34, London, UK, June 1987.

[5] H. Longuet-Higgins, "A computer algorithm for reconstructing a scene from two projections," *Nature,* 293:133-135, 1981

[6] B. Lucas and T. Kanade, "An Iterative Image Registration Technique with an Application to Stereo Vision," *Proc. DARPA Image Understanding Workshop,* pp. 121-130, April 1981.

[7] L. Matthies and S. Shafer, "Error Modelling in Stereo Navigation," *CMU-CS-86-140,* 1986.

[8] D. D. Morris and T. Kanade, "A Unified Factorization Algorithm for Points, Line Segments and Planes with Uncertainty Models," *CMU-RI-97-32,* 1997.

[9] C. Poelman, "The Paraperspective and Projective Factorization Methods for Recovering Shape and Motion," *CMU-CS-95-173,* July 1995.

[10] C. Poelman and T. Kanade, "A Paraperspective Factorization Method for Shape and Motion Recovery," *IEEE Trans. Pattern Analysis Machine Intelligence (PAMI-19),* vol. 19, no.3, pp. 206-18, March 1997

[11] W. Press, S. Teukolsky, W. Vettering and B. Flannery, *Numerical Recipes in C,* Cambridge University Press, 1994.

[12] L. Quan and T. Kanade, "A Factorization Method for Affine Structure from Line Correspondences," *CVPR'96,* pp. 803-8, San Fran., CA, USA, June 1996.

[13] J. Shi and C. Tomasi, "Good Features to Track," *CVPR'94,* pp. 593-600, Seattle, June 1994.

[14] M. Spetsakis and J. Aloimonos, "A Multi-frame Approach to Visual Motion Perception," *Int. J. Computer Vision,* Vol. 6, No. 3, Aug. 1991, pp. 245-55.

[15] R. Szeliski and S. B. Kang, "Recovering 3D shape and motion from image streams using non-linear least squares." *TR CRL 93/3,* DEC, Cambridge Research Lab, Mar. 1993.

[16] C. Tomasi and T. Kanade, "Shape and motion from image streams under orthography: a factorization method," *Int. J. Computer Vision,* vol. 9, no. 2, pp. 137-154, Nov. 1992

[17] R. Y. Tsai and T. S. Huang, "Uniqueness and Estimation of Three-Dimensional Motion Parameters of Rigid Objects with Curved Surfaces," *PAMI-6,* pp. 13-27, Jan. 1984.

[18] C. J. Taylor, "Structure and Motion from line segments in Multiple Images," *PAMI-17,* no. 11, pp. 1021-32, Nov. 1995.

[19] C. Therrien, *Discrete Random Signals and Statistical Signal Processing,* Prentice-Hall Inc., 1992

[20] B. Triggs, "Factorization methods for projective structure and motion," *CVPR'95,* pp. 845-851, 1995.