BU CS TR99-016, Dec., 1999 (revised Mar., 2000).
To appear in **Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR 2000)**.

Recursive Estimation of Motion and Planar Structure

Jonathan Alon and Stan Sclaroff
Image and Video Computing Group - Computer Science Dept.
Boston University - Boston, MA 02215

Abstract

A specialized formulation of Azarbayejani and Pentland's framework for recursive recovery of motion, structure and focal length from feature correspondences tracked through an image sequence is presented. The specialized formulation addresses the case where all tracked points lie on a plane. This planarity constraint reduces the dimension of the original state vector, and consequently the number of feature points needed to estimate the state. Experiments with synthetic data and real imagery illustrate the system performance. The experiments confirm that the specialized formulation provides improved accuracy, stability to observation noise, and rate of convergence in estimation for the case where the tracked points lie on a plane.

1 Introduction

Inferring 3D structure and motion from 2D image sequences has been a central problem in computer vision for many years. Many early studies focused on methods of relating pixel coordinates to 3D coordinates via camera calibration [22], that is computing the projection matrix which relates image coordinates to a world coordinate frame. In recent years, the focus has shifted to non-metric reconstruction from uncalibrated cameras [9], by computing the fundamental matrix (two views) [12], and the trilinear tensor (three views) [16]. Also, different camera models were assumed; *i.e.*, orthographic [20, 23], perspective projection [11, 25], or a unified model [1, 15].

Structure and motion algorithms typically assume given correspondences between features in successive frames. Finding such correspondences in a reliable way is a problem that still occupies researchers in the field. The most common approaches used to solve this problem are methods based on optical flow [8, 13] and methods based on feature matching [21, 26]. In flow-based methods, a velocity vector is computed for each pixel in the region of interest using variational techniques. In feature-based methods, image features such as points and lines are extracted in the first frame and are then matched to corresponding features in successive frames using correlation and relaxation techniques. In both types of methods it is typically assumed that the intensity of an image point, given a small motion between successive frames, will remain constant.

When dealing with a sequence of images, as with any sequence of observations, there are two possible frameworks to consider for parameter estimation. In a batch framework [14], observations from all frames are used simultaneously to estimate the state. In a recursive framework [3, 1, 17, 2] the current estimate, based on previous frames, together with new observations from the current frame yield a new state estimate. A batch framework is more suitable when all the information is available ahead of time, while a recursive framework is more suitable for real-time systems.

Algorithms for recovering structure and motion have many practical applications, such as reverse engineering, virtual reality, movie special effects, computer aided design, image compression, etc. Most of these algorithms are general in the sense that they assume no prior knowledge about the scene. However, in practice, the scene typically contains structures with strong geometric regularities. In particular, lines and planes occur frequently in real imagery and in very particular orientations [19]. Planar surfaces are quite common in both indoor and outdoor environments. Planar man-made structures such as table tops, floors, walls as well as buildings, roads, pavements, and playgrounds occur frequently in real image sequences.

1.1 New Approach

The goal in this paper is to reformulate the general recursive framework for pointwise structure recovery, in such a way that it takes into account a planarity constraint. This reformulation results in a smaller state vector, and consequently in a more accurate and stable system. The formulation is based upon the extended Kalman filter (EKF) approach originally proposed in [1]. Experiments with synthetic data and real imagery will be used to illustrate the system performance. The experiments confirm that the specialized formulation provides improved accuracy, stability to observation noise, and rate of convergence in estimation for the case where the tracked points lie on a plane.

Although two views analyses of planar structure were carried out in [10, 25, 6], these algorithms are known to be sensitive to measurement noise. More recently, Szeliski and Torr [19] showed how the quality of reconstruction can be improved via use of planarity constraints. In addition, Dellaert, et al. [4] demonstrated planar structure recovery through the inclusion of texture maps in an EKF measurement model; however, this approach assumed that the plane in the initial frame was front-facing. The approach proposed in this paper models planar structure explicitly and does not make any assumption about the initial orientation of the plane with respect to the camera.

2 Background

In this section we give a brief overview of a recursive estimation framework (EKF) to recover motion, structure and focal length from a sequence of images, given correspondences of feature points between frames. The formulation is due to Azarbayejani and Pentland [1], and will serve as the basis our new formulation for planar structure recovery.

In this formulation, state vector of the EKF consists of $7+N$ parameters, three for translation, three for incremental rotation, one for camera focal length, and N for depth of the feature points. The state vector is written as follows:

$$\mathbf{x} = (t_X, t_Y, t_Z\beta, \omega_X, \omega_Y, \omega_Z, \beta, \alpha_1, \cdots, \alpha_N) \quad (1)$$

where β is the inverse focal length, $(t_X, t_Y, t_Z\beta)$ is the relative translational motion, and $(\omega_X, \omega_Y, \omega_Z)$ describes the incremental rotational motion. Finally, point-wise structure is given by the subvector $(\alpha_1, \cdots, \alpha_N)$, where α_i is the depth associated with the i^{th} feature point. Depth is expressed relative to the coordinate system of the camera in the first frame. Note that depth can be recovered up to a scale factor only; therefore, it is customary to fix one of the α_i for purposes of gaining a solution [1].

It should be noted that rotation is formulated in terms of the incremental rotation quaternion:

$$\delta\mathbf{q} = (\sqrt{1-\epsilon}, \omega_X/2, \omega_Y/2, \omega_Z/2) \quad (2)$$

$$\epsilon = ({\omega_X}^2 + {\omega_Y}^2 + {\omega_Z}^2)/4. \quad (3)$$

¿From the unit incremental rotation quaternion $\delta\mathbf{q}$, the global rotation matrix R can be computed as described in [1]. This gives the relative rotation between the object reference frame and the current camera reference frame.

The measurement vector of the EKF is given by

$$\mathbf{z} = (u_1, v_1, u_2, v_2, \cdots, u_N, v_N) \quad (4)$$

where (u_i, v_i) is the image location of the i^{th} feature point, and $i = 1..N$ where N is the number of features. Thus, the measurement vector of the EKF consists of $2N$ parameters.

The following equations capture the geometry of the problem [1]. The first equation, relates the 3-D location of a single point (X, Y, Z) to its 2-D image location in the first frame (u^1, v^1):

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \begin{pmatrix} u^1 \\ v^1 \\ 0 \end{pmatrix} + \alpha \begin{pmatrix} u^1\beta \\ v^1\beta \\ 1 \end{pmatrix}, \quad (5)$$

where α is the depth (or structure), and $\beta = 1/f$ is the inverse focal length.

The coordinate frame transformation between the first frame and the current frame is formulated as:

$$\begin{pmatrix} X_C \\ Y_C \\ Z_C\beta \end{pmatrix} = \begin{pmatrix} t_X \\ t_Y \\ t_Z\beta \end{pmatrix} + \begin{pmatrix} 1 & & \\ & 1 & \\ & & \beta \end{pmatrix} R \begin{pmatrix} X \\ Y \\ Z \end{pmatrix}, \quad (6)$$

where R is the rotation matrix and $(t_X, t_Y, t_Z\beta)^T$ is the translation, as described above.

Finally, the camera model for central projection is formulated as follows:

$$\begin{pmatrix} u^k \\ v^k \end{pmatrix} = \begin{pmatrix} X_C \\ Y_C \end{pmatrix} \frac{1}{1 + Z_C\beta}, \quad (7)$$

where the coordinate system origin is fixed at the image plane, rather than at the center of projection.

The measurement equation for the EKF is obtained by combining the Eqs. 5, 6, and 7. For more details see [1].

A known weakness of this formulation is that it assumes that the image coordinates at the first frame are correct, an assumption that in theory biases the results, but has very small effect in practice. A possible remedy to this problem can be found in [2].

The computational complexity of the algorithm is cubic with respect to the number of points. In other methods [2] a seperate filter is run for every 3D point resulting in a linear algorithm. However, the motion and focal length are not modeled explicitly as part of the state, but rather are implicit in the projection matrix which is estimated in a separate stage.

3 Planar Structure Recovery

Given the above formulation, we can now add a constraint for all points to lie on a single plane. The measurement vector of the EKF remains the same, while the new state vector becomes:

$$\mathbf{x} = (t_X, t_Y, t_Z\beta, \omega_X, \omega_Y, \omega_Z, \beta, N_X, N_Y, N_Z). \quad (8)$$

The state now consists of only $7 + 3 = 10$ parameters, the first seven parameters are as in the original formulation, and the other three represent the plane parameters. Of course, a plane is completely determined by three non-collinear points, or equivalently by its unit normal (two DOFs) and its distance from the origin (one DOF).

Note that the dimension of the new state vector is independent of the number of features points. We expect that as the number of feature points grows larger, our estimator will out-perform the previous estimator since the latter's dimension grows with the number of feature points.

If the points lie on a plane then they satisfy the plane equation:

$$\mathbf{N} \cdot \mathbf{X} = 1 \quad (9)$$

where $\mathbf{X} = (X, Y, Z)$ is the 3-D point location and $\mathbf{N} = (N_X, N_Y, N_Z)$ is the plane (non-unit) normal, and the distance of the plane from the origin is given by $d = ||N||^{-1}$.

Rearranging 9 we get:

$$Z = \frac{1 - N_X X - N_Y Y}{N_Z}. \quad (10)$$

Since $\alpha = Z$, we can rewrite Eq. 5 using Eq. 10 :

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \begin{pmatrix} u^1 \\ v^1 \\ 0 \end{pmatrix} + \frac{1 - N_X X - N_Y Y}{N_Z} \begin{pmatrix} u^1 \beta \\ v^1 \beta \\ 1 \end{pmatrix}. \tag{11}$$

Rearranging 11 to solve for (X, Y, Z) we get:

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \eta \begin{pmatrix} (N_Z + \beta) u^1 \\ (N_Z + \beta) v^1 \\ 1 - N_X u^1 - N_Y v^1 \end{pmatrix}, \tag{12}$$

where $\eta = 1/(N_X u^1 \beta + N_Y v^1 \beta + N_Z)$.

As in the original formulation, the dynamic model in the EKF can be chosen trivially as an identity transform plus noise. The measurement model is given by Eqs. 12, 6, and 7. For derivation of the measurement Jacobian, see Appendix 5. Note that depth can be recovered up to a scale factor only; therefore, it is necessary to keep one of the N_i fixed for purposes of gaining a solution.

3.1 Relation to Two Frames Analysis

In this section, we discuss the degree of determinacy of our system and compare it with the system presented in [1]. In general, the parameters of a system can be recovered when the number of constraints outnumbers the degrees of freedom, or equivalently, the number of parameters in the system. The number of constraints in the original pointwise structure formulation is $(1 + 2N)$: one for scale and $2N$ for number of measurements (u and v coordinates of N feature points). The number of degrees of freedom is $(6 + 1 + N)$ for motion, camera and structure at every frame. So, whenever $(1 + 2N) > (1 + 6 + N)$ or $N \geq 7$, the motion, structure and camera can be recovered.

In our formulation, the number of constraints remains the same, while the number of parameters to recover is reduced to $(6 + 1 + 3) = 10$ for camera motion and planar structure at every frame. Note that this number is fixed and does not depend on the number of feature points N. So, whenever $1 + 2N > 10$ or $N \geq 5$, the motion, camera and planar structure can be recovered. If we work with a normalized camera model, that is, $f = 1 (\beta = 1)$, then the number of constraints increases to $2N + 2$ and we can recover motion and planar structure whenever $N \geq 4$. This result coincides with the "four corner" model [7, 25] or the "eight parameter" model [18] for esimating motion and planar structure from two perspective views.

3.2 Motion of Multiple Planes

It is possible to extend the formulation for estimation of a single plane to the case of multiple planes. Assume that there are m planes. Also assume that each image feature is assigned to its corresponding plane. In the case where the environment is rigid (all planes undergoe same translation and rotation), the state vector of Eq. 8 becomes:

$$\begin{aligned} \mathbf{x} = (&t_X, t_Y, t_Z \beta, \omega_X, \omega_Y, \omega_Z, \beta, \\ &N_X^1, N_Y^1, N_Z^1, \cdots N_X^m, N_Y^m, N_Z^m), \end{aligned} \tag{13}$$

where the superscripts denote the i^{th} plane parameters. The state now consists of $7 + 3m$ parameters, where the first seven parameters are as in the original formulation and the others represent the three parameters for each of the m planes. In this case, the translation, rotation, and focal length are the same for all planes; however, it is also possible to formulate the state vector such that each plane has its own, independent motion and translation.

4 Experiments

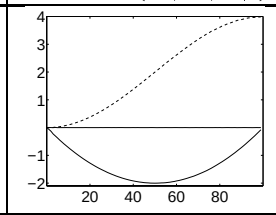
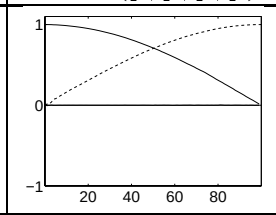
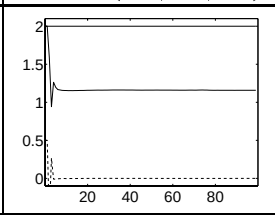
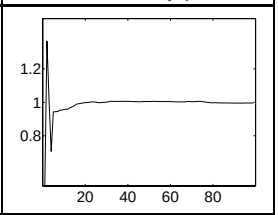
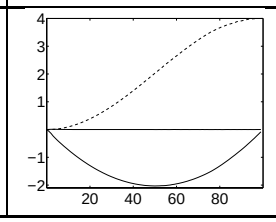
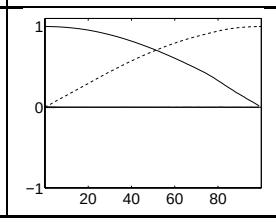
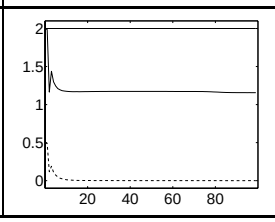
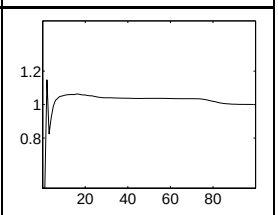
We now present two experiments for comparing our formulation with the original one. Planar structure for the original formulation is obtained by fitting a plane to the recovered 3D points.

In the first experiment we evaluate system performance with different noise levels, in a synthetic data setup similar to that used by [1]. The test sequence consists of 100 frames. The camera motion is predetermined for the entire sequence, and corresponds to a rotation about the y-axis located at $(0, 0, 2)$. The true focal length is set to one, $\beta = 1$ and the true structure consists of some 30 points uniformly scattered on a plane, with a fixed (non-unit) normal direction $(N_X, N_Y, N_Z) = 2(\frac{1}{\sqrt{3}}, 0, 1)$. The initial motion parameter estimates are set to zero, with variances zero. The initial inverse focal length is set to 0.5, with variance 0.1. The components of the plane normal direction are set to (1.5, 0.5, 2) with variances (0.1, 0.1, 0). In each trial, uniform noise with varying standard deviation is added to both x and y image coordinates. The standard deviation corresponds to 2, 6, and 10 pixels, based on an image size of (512,512).

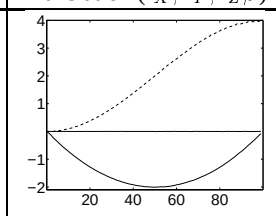
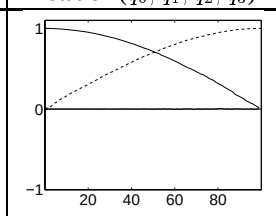
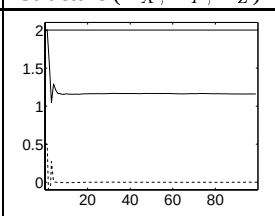
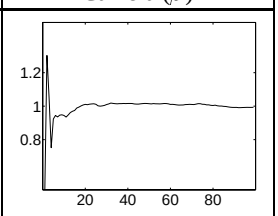
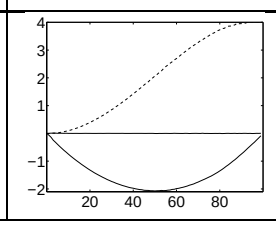
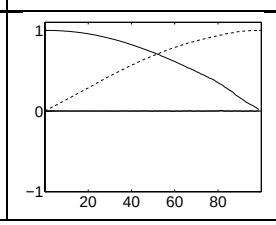
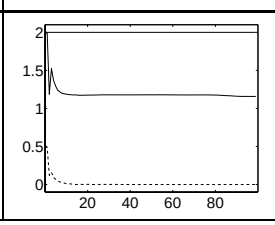
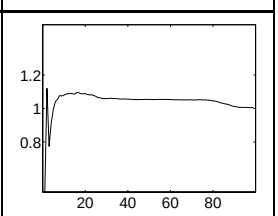
Fig. 1 illustates recovery of planar structure, camera motion, and focal length for the three noise levels. Multiple trials (twenty at each noise level) were conducted, and the average estimates were plotted on the graphs. Parameters t_X, q_0, N_X, and β are represented by solid lines on the graphs; $t_Z \beta$, N_Y q_2 are represented by broken lines. To avoid clutter in the graphs we do not show t_Y, q_1, q_2, and N_Z.

Table 1 shows a summary of statistics for the experiment. As can be seen in both the graphs and the table, the new formulation converges faster to the planar structure and camera parameter estimates. In addition, the new formulation tends to be more stable as noise levels increase; the mean error and variance in estimating the plane normal and the camera parameters are both smaller when the new formulation is employed.

Uniform noise ± 2 pixels.

Method	Translation ($t_X, t_Y, t_Z\beta$)	Rotation (q_0, q_1, q_2, q_3)	Structure (N_X, N_Y, N_Z)	Camera (β)
New				
Old				

Uniform noise ± 6 pixels.

Method	Translation ($t_X, t_Y, t_Z\beta$)	Rotation (q_0, q_1, q_2, q_3)	Structure (N_X, N_Y, N_Z)	Camera (β)
New				
Old				

Uniform noise ± 10 pixels.

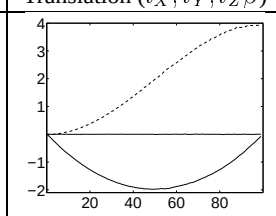
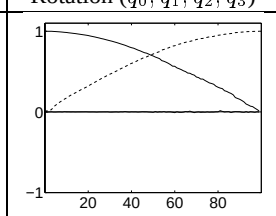
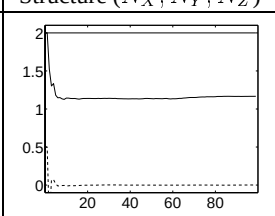
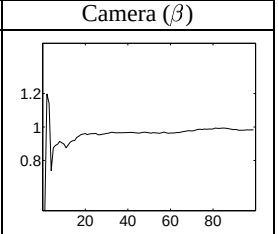
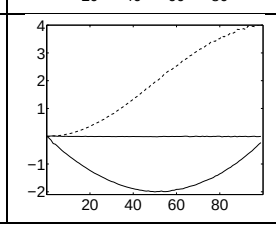
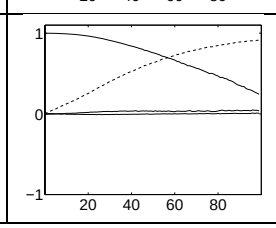
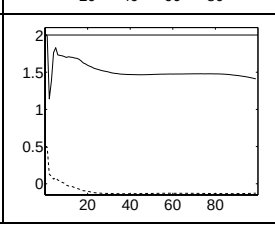
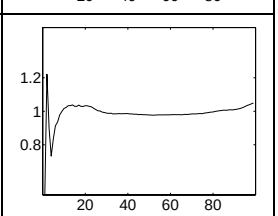
Method	Translation ($t_X, t_Y, t_Z\beta$)	Rotation (q_0, q_1, q_2, q_3)	Structure (N_X, N_Y, N_Z)	Camera (β)
New				
Old				

Figure 1: Experiment using synthetic data with random noise added. Accuracy and convergence of the new *vs.* old formulations was measured against ground truth, as described in the text. Multiple trials (twenty at each noise levels) were conducted, and the average estimates shown on the graphs. Each graph's x-axis is the frame number and the y-axis is the state variable. Parameters t_X, q_0, N_X, and β are represented by solid lines on the graphs; $t_Z\beta$, N_Y q_2 are represented by broken lines. To avoid clutter in the graphs we do not show t_Y, q_1, q_2, and N_Z. For a summary of statistics, see Table 1.

Performance Statistics: New Formulation

Noise pixels	Motion Estimation Error				Convergence	
	m_t	σ_t	m_q	σ_q	r_s	r_c
2	-0.0012	0.0129	-0.0037	0.0298	7	9
6	-0.0018	0.0304	0.0009	0.0584	7	14
10	-0.0033	0.0634	-0.0058	0.3352	7	19

Performance Statistics: Old Formulation

Noise pixels	Motion Estimation Error				Convergence	
	m_t	σ_t	m_q	σ_q	r_s	r_c
2	0.0002	0.0325	-0.0638	0.3195	12	25
6	0.0012	0.0790	-0.0810	0.3274	14	25
10	-0.0238	0.3316	-0.2366	0.4673	>100	19

Table 1: Average performance statistics for synthetic data experiments with increasing noise level. Experiments were conducted in trials with varying uniform noise (standard deviation 2, 6, and 10 pixels). Mean error and root mean squared error are shown for the recovered camera motion parameters (translation, rotation). For the static parameters (structure and camera) the table provides the frame number for which the camera parameters converge to within 5% of the true value, and frame number for which the normal converges to within 0.5^o of its true value.

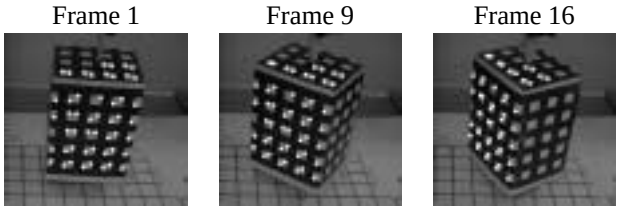


Figure 2: Example frames from the real image sequence. Tracked features are shown in white.

4.1 Experiments on Real Imagery: Box sequence

In this section, we describe experiments with the BOX sequence available from the UMass database [5]. Fig. 2 shows example frames from the sequence. Corner features were tracked using an implementation of the Kanade-Lucas-Tomasi feature tracker available from http://vision.stanford.edu/birch/klt/. The corner features on the front face of the box were tracked and used as measurement input to both the new and old EKF formulations. As in the previous experiments, planar structure for the original formulation was obtained by fitting a plane to the recovered 3D points.

Graphs showing the estimated translation, rotation, structure, and inverse focal length for both formulations are shown in Fig. 3. The ground truth for each translation and rotation parameter for this sequence lies approximately along a line [1]. As can be seen in the graphs in Fig. 3, the estimates of camera motion obtained by the new EKF formulation tend to converge faster. A more pronounced difference in convergence can be seen in the estimated of the planar structure. At the time of this writing, the authors have been unable to obtain the "ground truth" for the Box sequence. Quantitative RMS error comparisons will be reported in the final version of the paper.

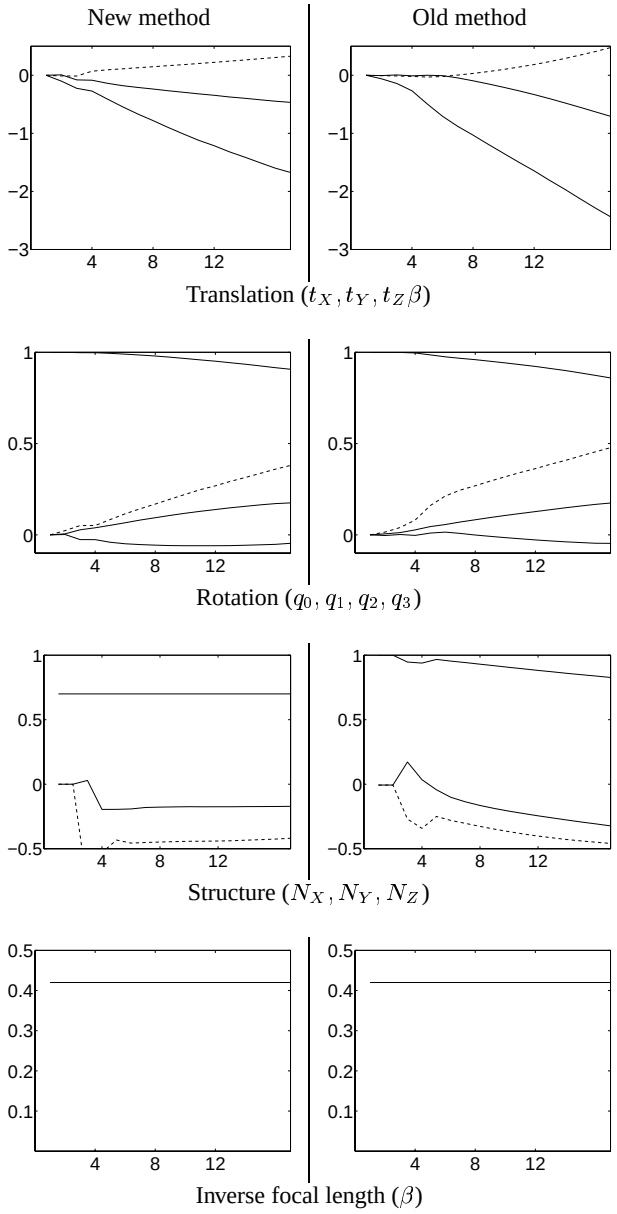


Figure 3: Graphs showing the estimated translation, rotation, structure, and inverse focal length estimated in the UMass box sequence. Features belonging to the front face of the box were used as measurement input to the new and old EKF formulation.

5 Conclusion

We have presented a specialization of the Azarbayejani and Pentland feature-based recursive estimator, to the case of planar structure. We have shown how adding this geometric constraint reduces the dimension of the state vector, and consequently yields a more numerically stable estimator. Since planar surfaces are quite common in man-made environments, this new formulation should prove valuable. It

is likely that this idea of adding a geometric constraint can be carried over to other surfaces which have the analytical form $z = f(x, y)$. In future work we plan to extend this approach to the case of quadric surfaces, which also occur frequently in many real image sequences.

Acknowledgments

This work was supported in part through ONR Young Investigator Award N00014-96-1-0661, and NSF grants IIS-9624168 and EIA-9623865. We thank Paul Beardley and Stefano Soatto for inspiring discussions and for pointing out the strengths and weaknesses of different approaches. We thank Ali Azarbayejani for making his code available.

References

[1] A. Azarbayejani and A.P. Pentland. Recursive estimation of motion, structure, and focal length. *PAMI*, 17(6), 1995.

[2] P. Beardsley, A. Zisserman, and D. Murray. Sequential updating of projective and affine structure from motion. *IJCV*, 23(3):235–259, 1997.

[3] T.J. Broida, S. Chandrashekhar, and R. Chellappa. Recursive 3-d motion estimation from a monocular image sequence. *AeroSys*, 26(4), 1990.

[4] F. Dellaert, S. Thrun, and C. Thorpe. Jacobian images of super-resolved texture maps for model based motion estimation and tracking. *WACV98*, 1998.

[5] R. Dutta, R. Manmatha, L.R. Williams, and E.M. Riseman. A data set for quantitative motion analysis. *CVPR*, 1989.

[6] O. Faugeras. *Three-Dimensional Computer Vision*. MIT Press, 1993.

[7] Paul Heckbert. Fundamentals of texture mapping and image warping. MA Thesis, Berkeley, 1989.

[8] B.K.P. Horn and B.G. Schunck. Determining optical flow. *AI*, 17, 1981.

[9] J.J. Koenderink and A.J. vanDoorn. Affine structure from motion. *JOSA-A*, 8(2), 1991.

[10] C.H. Lee. Structure and motion from two perspective views via planar patch. *ICCV*, 1988.

[11] H.C. Longuet-Higgins. A computer algorithm for reconstructing a scene from two projections. *Nature*, 293, 1981.

[12] Q.T. Luong, R. Deriche, O.D. Faugeras, and T. Papadopoulo. On determining the fundamental matrix: Analysis of different methods and experimental results. INRIA, 1993.

[13] H.H. Nagel. On the estimation of optical flow: Relations between different approaches and some new results. *AI*, 33(3), 1987.

[14] J. Oliensis. A multi-frame structure-from-motion algorithm under perspective projection. *IJCV*, 34(2/3), 1999.

[15] A. Shashua and N. Navab. Relative affine structure: Canonical model for 3D from 2D geometry and applications. *PAMI*, 18(9), 1996.

[16] A. Shashua and M. Werman. Trilinearity of three perspective views and its associated tensor. *ICCV*, 1995.

[17] S. Soatto and R. Brockett. Optimal structure from motion: Local ambiguities and global estimates. *CVPR*, 1998.

[18] R. Szeliski. Video mosaics for virtual environments. *IEEE CG&A*, 16(2), 1996.

[19] R. Szeliski and P. H. S. Torr. Geometrically constrained structure from motion: Points on planes, MSR-TR-98-64. Microsoft Research, 1998.

[20] C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: A factorization method. *IJCV*, 9(2), 1992.

[21] C. Tomasi and J. Shi. Good features to track. *CVPR*, 1994.

[22] R. Y. Tsai. An efficient and accurate camera calibration technique for 3D machine vision. *CVPR*, Miami Beach, FL, 1986.

[23] S. Ullman. *The Interpretation of Visual Motion*. MIT Press, 1979.

[24] G. Welch and G. Bishop. An introduction to the kalman filter. CS TR 95-041, UNC, 1995.

[25] J. Weng, T.S. Huang, and N. Ahuja. *Motion and Structure from Image Sequences*. Springer-Verlag, 1993.

[26] Z.Y. Zhang, R. Deriche, O.D. Faugeras, and Q.T. Luong. A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry. *AI*, 78(1-2), October 1995.

Appendix: Measurement Jacobian

Using the notation of [24], the measurement equation of the EKF is given by

$$\mathbf{z}^k = h(\mathbf{x}^k, \nu^k)$$

where $\mathbf{z}$ is the measurement vector, $\mathbf{x}$ is the state vector, ν is the measurement noise and superscript k is the frame

number. The measurement function h is given by combining Eqs. 6,7, and 12. The measurement Jacobian is given by

$$H_{i,j} = \frac{\partial h_i}{\partial x_k}(\tilde{\mathbf{x}}^k, 0).$$

The partials with respect to translation are as follows:

$$\begin{aligned}
\frac{\partial h_i}{\partial t_X} &= \frac{1}{1+Z_C\beta}\begin{pmatrix}1\\0\end{pmatrix},\\
\frac{\partial h_i}{\partial t_Y} &= \frac{1}{1+Z_C\beta}\begin{pmatrix}0\\1\end{pmatrix},\\
\frac{\partial h_i}{\partial t_Z\beta} &= \frac{-1}{(1+Z_C\beta)^2}\begin{pmatrix}X_C\\Y_C\end{pmatrix}.
\end{aligned}$$

The partials with respect to incremental rotation are as follows:

$$\frac{\partial h_i}{\partial \omega_j} = \frac{1}{1+Z_C\beta}\begin{pmatrix}\frac{\partial X_C}{\partial \omega_j}\\ \frac{\partial Y_C}{\partial \omega_j}\end{pmatrix} + \frac{\partial Z_C\beta}{\partial \omega_j}\cdot\frac{\partial h_i}{\partial t_Z\beta},$$

where $j = (X, Y, Z)$, and

$$\frac{\partial}{\partial \omega_j}\begin{pmatrix}X_C\\Y_C\\Z_C\beta\end{pmatrix} = \begin{pmatrix}1&&\\&1&\\&&\beta\end{pmatrix}\frac{\partial R^k_{\delta q}}{\partial \omega_j}R_q^{k-1}\begin{pmatrix}X\\Y\\Z\end{pmatrix}.$$

The partials of the incremental rotation matrix with respect to ω_j are as follows:

$$\frac{\partial R^k_{\delta q}}{\partial \omega_X} = \begin{pmatrix}0 & \frac{2\omega_Y\gamma+\omega_X\omega_Z}{4\gamma} & \frac{2\omega_Z\gamma-\omega_X\omega_Y}{4\gamma}\\ \frac{2\omega_Y\gamma-\omega_X\omega_Z}{4\gamma} & -\omega_X & \frac{{\omega_X}^2}{4\gamma}-\gamma\\ \frac{2\omega_Z\gamma+\omega_X\omega_Y}{4\gamma} & -\frac{{\omega_X}^2}{4\gamma}+\gamma & -\omega_X\end{pmatrix},$$

$$\frac{\partial R^k_{\delta q}}{\partial \omega_Y} = \begin{pmatrix}-\omega_Y & \frac{2\omega_X\gamma+\omega_Y\omega_Z}{4\gamma} & -\frac{{\omega_Y}^2}{4\gamma}+\gamma\\ \frac{2\omega_X\gamma-\omega_Y\omega_Z}{4\gamma} & 0 & \frac{2\omega_Z\gamma+\omega_X\omega_Y}{4\gamma}\\ \frac{{\omega_Y}^2}{4\gamma}-\gamma & \frac{2\omega_Z\gamma-\omega_X\omega_Y}{4\gamma} & -\omega_Y\end{pmatrix},$$

$$\frac{\partial R^k_{\delta q}}{\partial \omega_Z} = \begin{pmatrix}-\omega_Z & \frac{{\omega_Z}^2}{4\gamma}-\gamma & \frac{2\omega_X\gamma-\omega_Y\omega_Z}{4\gamma}\\ -\frac{{\omega_Z}^2}{4\gamma}+\gamma & -\omega_Z & \frac{2\omega_Y\gamma+\omega_X\omega_Z}{4\gamma}\\ \frac{2\omega_X\gamma+\omega_Y\omega_Z}{4\gamma} & \frac{2\omega_Y\gamma-\omega_X\omega_Z}{4\gamma} & 0\end{pmatrix}.$$

Here, $\gamma = \sqrt{1-\epsilon}$ and ϵ is given in Eq. 3. $R^k_{\delta q}$ is the interframe rotation matrix and R^k_q is the global rotation matrix, that is the rotation of the camera from the first frame to the predecessor of the current frame. R^k_q is updated every frame by multiplying the interframe and global rotation matrices. $R^{k+1}_q = R^k_{\delta q}R^k_q$.

The partials with respect to β are:

$$\frac{\partial h_i}{\partial \beta} = \frac{1}{1+Z_C\beta}\begin{pmatrix}\frac{\partial X_C}{\partial \beta}\\ \frac{\partial Y_C}{\partial \beta}\end{pmatrix} + \frac{\partial Z_C\beta}{\partial \beta}\cdot\frac{\partial h_i}{\partial t_Z\beta},$$

where

$$\begin{aligned}
\frac{\partial}{\partial \beta}\begin{pmatrix}X_C\\Y_C\\Z_C\beta\end{pmatrix} &= \begin{pmatrix}0&0&0\\0&0&0\\0&0&1\end{pmatrix}R_q\begin{pmatrix}X\\Y\\Z\end{pmatrix} +\\
&\eta^2(1-N_Xu-N_Yv)\times\\
&\begin{pmatrix}1&&\\&1&\\&&\beta\end{pmatrix}R_q\begin{pmatrix}N_Zu\\N_Zv\\-(N_Xu+N_Yv)\end{pmatrix}
\end{aligned}$$

and where $\eta = 1/(N_Xu\beta + N_Yv\beta + N_Z)$.

The partials with respect to the plane parameters are:

$$\frac{\partial h_i}{\partial N_j} = \frac{1}{1+Z_C\beta}\begin{pmatrix}\frac{\partial X_C}{\partial N_j}\\ \frac{\partial Y_C}{\partial N_j}\end{pmatrix} + \frac{\partial Z_C\beta}{\partial N_j}\cdot\frac{\partial h_i}{\partial t_Z\beta},$$

where $j = (X, Y, Z)$, and

$$\frac{\partial}{\partial N_j}\begin{pmatrix}X_C\\Y_C\\Z_C\beta\end{pmatrix} = \begin{pmatrix}1&&\\&1&\\&&\beta\end{pmatrix}R_q\frac{\partial}{\partial N_j}\begin{pmatrix}X\\Y\\Z\end{pmatrix}.$$

The partials of 3-D position with respect to the plane parameters are:

$$\frac{\partial}{\partial N_X}\begin{pmatrix}X\\Y\\Z\end{pmatrix} = -\eta^2u(N_Z+\beta)\begin{pmatrix}u\beta\\v\beta\\1\end{pmatrix},$$

$$\frac{\partial}{\partial N_Y}\begin{pmatrix}X\\Y\\Z\end{pmatrix} = -\eta^2v(N_Z+\beta)\begin{pmatrix}u\beta\\v\beta\\1\end{pmatrix},$$

$$\frac{\partial}{\partial N_Z}\begin{pmatrix}X\\Y\\Z\end{pmatrix} = -\eta^2(1-N_Xu-N_Yv)\begin{pmatrix}u\beta\\v\beta\\1\end{pmatrix}.$$