Motion and Structure from Time-Varying Optical Flow

J.L. Barron
Dept. of Computer Science, MC-383
University of Western Ontario
London, Ontario, N6A 5B7
barron@csd.uwo.ca

R. Eagleson
Electrical and Computer Engineering
Ryerson Polytechnic University
Toronto, Ontario, M5B 2K3
eagleson@ee.ryerson.ca

Abstract We present a computational framework for recovering both 1^{st}-order motion parameters (observer direction of translation and observer rotation), 2^{nd}-order motion parameters (observer rotational acceleration) and relative depth maps from time-varying optical flow. We cannot recover **absolute** observer translational speed or translational acceleration although because only **relative** depth, which is the ratio of current translational speed and 3d depth, is affected by these parameters. Our assumption is that the observer rotational motion is no more than "second order"; in other words, observer motion is either constant or has at most constant acceleration. We examine the effect of noise – which is ubiquitous in optical flow data – on the solution of the motion and structure parameters. This ensemble of unknowns comprises a solution to the classical 'structure-and-motion from optical flow' problem. Our complete framework utilizes a simple method for interpreting the bilinear image velocity equation by solving simple systems of linear equations. Since our noise analysis yields uncertainty measures for each parameter, a Kalman filter is employed to incrementally intergate new measurements computed as each additional frame in the sequence is processed. We conclude by analyzing this reduction of uncertainty over time as the system converges to a stable solution.

1 Introduction

This paper addresses the problem of robust estimation of motion and structure parameters, which describe an observer's translation and rotation and the environmental structure, i.e. the depth of visible 3d points, from noisy time-varying optical flow. We do not assume a local surface model, but rather, decouple the depth scaled speed and the direction of translation from each other in the image velocity equation and then solve for each separately. Since time-varying image velocity is used, an observer acceleration model is required. We assume the direction of translation in the observer's frame of reference is constant while the entire frame of reference can be rotating with constant rotational acceleration. There can also be acceleration in translational speed but this only causes changes in the relative depth values. With each added frame in the motion sequence, we solve a linear system of equations relating these motion and structure parameters to the time-varying image velocity in the least squares sense. A Kalman filter [8] is then used to integrate these parameters over time.

1.1 Geometry of Projected Motions from 3d to 2d

The standard image velocity equation [14] relates an image velocity measured at image location $\vec{Y} = (y_1, y_2, 1) = \vec{P}/X_3$, i.e. the perspective projection of a 3d point $\vec{P} = (X_1, X_2, X_3)$, to the 3d observer translation $\vec{U}$ and 3d observer rotation $\vec{\omega}$. Assuming a focal length of 1 we can write the image velocity $\vec{v} = (v_1, v_2)$ as

$$\vec{v}(\vec{Y},t) = A_1(\vec{Y})\vec{u}(\vec{Y},t)||\vec{Y}||_2 + A_2(\vec{Y})\vec{\omega}(t), \quad (1)$$

where

$$A_1 = \begin{pmatrix} -1 & 0 & y_1 \\ 0 & -1 & y_2 \end{pmatrix} \quad \text{and} \quad (2)$$

$$A_2 = \begin{pmatrix} y_1 y_2 & -(1+y_1^2) & y_2 \\ (1+y_2^2) & -y_1 y_2 & -y_1 \end{pmatrix}. \quad (3)$$

Since we are dealing with a monocular observer we cannot recover the observer's absolute translation, $\vec{U}$, or the actual 3d coordinates, $\vec{P}$, of environmental points but rather the ratio of the two. We define the depth scaled observer translation as

$$\vec{u}(\vec{Y},t) = \frac{\vec{U}(t)}{||\vec{P}(t)||_2} = \hat{u}\mu(\vec{Y},t), \quad (4)$$

where $\hat{u} = \hat{U} = (u_1, u_2, u_3)$ is the normalized direction of translation and $\mu(\vec{Y}, t) = \frac{||\vec{U}||_2}{||\vec{P}||_2} = \frac{||\vec{U}||_2}{X_3||\vec{Y}||_2}$ is the depth scaled observer speed at $\vec{Y}$ at time t. We refer to μ as relative depth in this paper.

1.2 Literature Survey

In recent years there has been considerable interest in using long image sequences to recover motion and structure [1, 9, 13, 19, 21, 23, 15, 20, 24, 6, 5, 18, 7]. Most of the algorithms use monocular image sequences but a few use binocular image sequences [16, 25]. A few of the algorithms are batch (processing all the data at once) [21, 6, 4, 20] but most use some type of recursive estimation method (for example, an extended Kalman filter) [4, 5, 10] or (often nonlinear) minimization or factorization methods [6, 20]. Batch methods are conceptually less complicated but recursive estimation methods yield a "best" answer at any given time and may be computationally more feasible when using real-time acquired data. Most methods are based on point correspondences [13, 10, 23, 15, 18], feature correspondences (for example corner points or straight line segments) [5, 25, 20, 6] or simple pixel correspondences [16]. Only a few of the methods involve Kalman filtering of motion parameters that are derived from optical flow. In particular, the work proposed by De Micheli, Torre and Uras [17] uses the optical flow method of [22] to compute flow in long image sequences and then computes time-to-collision and angular velocity from the flow. We also use a Kalman filter in this paper to integrate the computed parameters, i.e. direction of observer translation and observer rotation and rotational acceleration, over time. However, our parameters are more general, incorporating both 1^{st} and 2^{nd} order motion parameters (8 degrees of freedom) and relative depth maps (1 degree of freedom).

2 Overview

We first present the solution for one frame and then the solution for three frames. The solution using three frames is the actual solution adopted, it uses ideas from the one frame solution. We show how to compute the direction of translation from a linear equation based on sums of velocities in non-overlapping image regions. The improved accuracy in the direction of translation positively impacts on the calculation of rotation and relative depth. The velocity summing reduces the noise sensitivity inherent in computing velocity differences for noisy data. In order to perform the estimation robustly in the presence of noise, we propose that this summed velocity system be solved as a least-squares fit to multiple data points (velocity regions) from the entire optical flow field. Given the computed motion parameters at each time (computed from tuples of flow fields at each of 3 adjacent times), we can integrate these parameters over time by using 3 nested Kalman filters. The result is increased parameter accuracy and stability over time. Lastly, we demonstrate the robustness of our solution technique on simulated test data.

3 The Least Squares Solution

We now re-examine the temporal relationships, over adjacent frames, for the image velocity equation that describe the observer motion (direction of translation and rotation) and object structure (relative depth) for sampled optical flow fields.

3.1 Using One Frame

First, we assume that either $\hat{u}$ or $\vec{\omega}$ are known by some unspecified means. Given $\hat{u}$ we follow the solution technique by Heeger and Jepson [12] to compute $\vec{\omega}$. We first compute the normalized direction of $\vec{v}_T$ as

$$\hat{d} = \frac{(A_1(\vec{Y})\hat{u})}{||A_1(\vec{Y})\hat{u}||_2}. \quad (5)$$

Given $\hat{d} = (d_1, d_2)$ we can compute $\hat{d}^{\perp}$ as $(d_2, -d_1)$. Hence we obtain one linear equation in the 3 components of $\vec{\omega}(t)$

$$\hat{d}^{\perp} \cdot \vec{v} = \hat{d}^{\perp} \cdot (A_2(\vec{Y})\vec{\omega}(t)), \quad (6)$$

This equation can be solved by standard least squares and involves solving a simple 2×2 linear system of equations. If $\vec{\omega}$ is zero or $\hat{d}$ and $A_2(\vec{Y})\vec{\omega}$ are parallel, then $\vec{\omega}$ cannot be recovered.

Given $\vec{\omega}$ we can solve for $\hat{u}$. We can then write

$$\vec{r} = \vec{v} - A_2(\vec{Y})\vec{\omega} = A_1(\vec{Y})\hat{u}\mu||\vec{Y}||_2. \quad (7)$$

The ratios of the two components of $\vec{r}$ give us 1 linear equation in (u_1, u_2, u_3):

$$\frac{r_1}{r_2} = \frac{-u_1 + y_1 u_3}{-u_2 + y_2 u_3}. \quad (8)$$

Since only 2 components of $\hat{u}$ are independent, then if u_3 is not zero we can write

$$r_2 \frac{u_1}{u_3} - r_1 \frac{u_2}{u_3} = r_2 y_1 - r_1 y_2. \quad (9)$$

This is 1 linear equation in 2 unknowns. We can construct and solve a system of linear equations based on (9) using least squares to obtain the direction of translation $(u_1/u_3, u_2/u_3, 1)$, which when normalized yields $\hat{u}$. In the event u_3 is

zero we can always solve for $(1, u_2/u_1, u_3/u_1)$ or $(u_1/u_2, 1, u_3/u_2)$ in a similar manner by appropriate manipulation of equation (9). These equations are given in [3].

Note that if there is no observer translation, the 3 systems of equations will all be singular. We can detect this situation by choosing the result with the smallest condition number, κ. We compute $\hat{u}$ all three ways and then choose the $\hat{u}$ from the system of equations having the lowest condition number. If all of the κ's are large (indicating singularity as $\hat{u} \approx (0,0,0)$) then only $\vec{\omega}$ can be recovered by solving a linear system of equations comprised of equations of the form

$$\vec{v}(\vec{Y},t) \;=\; A_2(\vec{Y})\vec{\omega}(t). \qquad (10)$$

Results in [2] indicate this system of equations is robust in the face of noisy data.

Finally, given $\hat{u}$ and $\vec{\omega}(t)$ each image velocity, $\vec{v}(\vec{Y},t)$ yields two equations for $\mu(\vec{Y}_i,t)$

$$\vec{r} = \vec{v} - A_2(\vec{Y}_i)\vec{\omega}(t) = A_1(\vec{Y})\hat{u}\mu(\vec{Y}_i,t)||\vec{Y}_i||_2 = \vec{s}\mu(\vec{Y}_i,t), \qquad (11)$$

which we solve for each image velocity measurement as:

$$\mu(\vec{Y}_i) = \frac{r_1}{s_1} = \frac{r_2}{s_2}. \qquad (12)$$

If $\vec{v}$ is purely horizontal or vertical at some image location then one of s_1 or s_2 will be zero, the other will allow μ to be recovered. If both s_1 and s_2 are non-zero we average the two computed μ values:

$$\mu(\vec{Y}_i) = \frac{1}{2}\left(\frac{r_1 s_2 + r_2 s_1}{s_1 s_2}\right). \qquad (13)$$

3.2 Using Three Frames

Modelling uniform rotational acceleration as

$$\vec{\omega}(t+\delta t) \;=\; \vec{\omega}(t) + \delta\vec{\omega}\;\delta t, \qquad (14)$$

we can write

$$\left(v_2(\vec{Y},t_1) + v_2(\vec{Y},t_{-1}) - 2v_2(\vec{Y},t_0)\right)\left(\frac{u_2}{u_3} - y_2\right) =$$
$$\left(v_1(\vec{Y},t_1) + v_1(\vec{Y},t_{-1}) - 2v_1(\vec{Y},t_0)\right)\left(\frac{u_1}{u_3} - y_1\right) \qquad (15)$$

We have assumed that $|t_{i+1} - t_i| = |t_i - t_{i-1}|$ where t_i is the central frame, i.e. the flow is sampled at every δt. Note that in order to compute $\hat{u}$ we must be able to measure image velocities at the same image locations in three adjacent frames. In that case equation (15) can be reformulated as a simple linear least squares problem involving solving 2×2 systems of equations. Once $\hat{u}$ has been computed we can solve for $\vec{\omega}$ and μ using the one-frame solution. Lastly, rotational acceleration can be found at frame i as

$$\delta\vec{\omega}(t_i) \;=\; \frac{\vec{\omega}(t_{i+1}) - \vec{\omega}(t_{i-1})}{2}. \qquad (16)$$

This type of observer motion is a generalization of the 2^{nd} type of motion considered in [2].

3.3 Robustness through Summing

Assuming mean zero random error in the velocities, we sum (15) in $2^a \times 2^a$ regions of the image and then solve it using standard least squares. This causes the random error in the image velocities to cancel out on average and results in accurate image velocity differences and hence a more accurate calculation of $\hat{u}$. Full details are in [3].

4 Kalman Filtering

We have outlined how to recover $\hat{u}$, $\vec{\omega}$, $\delta\vec{\omega}$ and μ in batch mode given three images in a sequence. These are computations involving the construction and solution of 2×2 and 3×3 systems of linear equations. We refer to the solutions found from these computations as the motion and structure "measurements" at that time and subscript them with M. We compute the variances in these measurements as the difference between computed and measured data (the residuals), that is:

$$\sigma^2_{M\hat{u}} \;=\; ||WM_2\hat{u}_M - WB_2||^2_2 \qquad (17)$$

and

$$\sigma^2_{M\vec{\omega}} \;=\; ||WM_3\vec{\omega}_M - WB_3||^2_2, \qquad (18)$$

where M_2, M_3, B_2 and B_3 are matrices/vectors involved in the least squares calculations. Since $\hat{u}$ has only two independent components we can express them as angles, θ and ϕ, in a right hand coordinate system with respect to the $(0,0,1)$ axis. We use a Kalman filter to update the θ and ϕ angles rather than updating the $\hat{u}$ vectors directly to avoid adding normalized vectors together in the Kalman equations. Updating θ and ϕ can be viewed simply as moving a point on the surface of a unit sphere.

We use a series of nested Kalman filters to integrate these measurements over time. Below we outline the steps in our filter computation for $n+2$ images, numbered 0 to $n+1$, and show how we continually update the solutions for frames 1 to n. We subscript items by M if they are measured quantities, by C if they are computed quantities and by P if they are predicted quantities.

1. Initialize predicted parameters to zero or ∞ as appropriate (see [3]).

Motion	$\|\|\vec{U}\|\|_2$	$\vec{\omega}$	$\delta\vec{\omega}$	Description
1	40.0	(0,0,0)	(0,0,0)	Pure Translation, Single Plane
2	40.0	(0,0.175,0) [10°]	(0,0,0)	General Motion, Single Plane
3	40.0	(0,0.175,0) [10°]	(0,0,0)	General Motion, Multiple Planes
4	15.707963	(0.0.157,0) [9°]	(0,0,0)	Circular Motion, Constant Rotation
5	7.853986	(0,0.0785,0) [4.5°]	(0,0.00785,0) [0.45°]	Circular Motion, Constant Rotational Acceleration

Table 1: Summary of Experimental Motions.

2. Compute $\hat{u}_{Mi}$ from (15) and σ^2_{ui} from (17). These are considered the measured quantities. Compute θ_{Mi} and ϕ_{Mi} from $\hat{u}_{Mi}$ and θ_{Pi} and ϕ_{Pi} from $\hat{u}_{Pi}$. Then:

$$\begin{aligned} K_{\hat{u}i} &= \frac{\sigma^2_{P\hat{u}i}}{\sigma^2_{P\hat{u}i}+\sigma^2_{M\hat{u}i}} \\ \theta_{Ci} &= \theta_{Pi} + K_{\hat{u}i}(\theta_{Mi} - \theta_{Pi}) \\ \phi_{Ci} &= \phi_{Pi} + K_{\hat{u}i}(\phi_{Mi} - \phi_{Pi}) \\ \sigma^2_{C\hat{u}i} &= K_{\hat{u}i}\sigma^2_{M\hat{u}i} \end{aligned}$$

$K_{\hat{u}i}$ is the Kalman filter gain for $\hat{u}_{Ci}$ at time i. From θ_{Ci} and ϕ_{Ci} we can compute $\hat{u}_{Ci}$.

(a) Given $\hat{u}_{Ci}$ compute $\vec{\omega}_{M_{i,j}}$ from (6) and $\sigma^2_{M\vec{\omega}_{i,j}}$ from (18), for $j = i-1, i, i+1$. Then for each of these j values we compute:

$$\begin{aligned} K_{\vec{\omega}_{i,j}} &= \frac{\sigma^2_{P\vec{\omega}_{i,j}}}{\sigma^2_{M\vec{\omega}_{i,j}}+\sigma^2_{\vec{\omega}_{i,j}}} \\ \vec{\omega}_{C_{i,j}} &= \vec{\omega}_{P_{i,j}} + K_{\vec{\omega}_{i,j}}(\vec{\omega}_{M_{i,j}} - \vec{\omega}_{P_{i,j}}) \\ \sigma^2_{C\vec{\omega}_{i,j}} &= \frac{\sigma^2_{P\vec{\omega}_{i,j}}}{\sigma^2_{P\vec{\omega}_{i,j}}+\sigma^2_{M\vec{\omega}_{i,j}}} \end{aligned}$$

(b) Compute $\delta\vec{\omega}_{Mi}$ from $\vec{\omega}_{Ci-1}$, and $\vec{\omega}_{Ci+1}$ as per (16). Compute $\sigma^2_{M\delta\vec{\omega}_i}$ as $\sigma^2_{C\vec{\omega}_{i-1}} + \sigma^2_{C\vec{\omega}_{i+1}}$. This sum indicates we place considerably less confidence in $\delta\vec{\omega}_i$ than we do in $\vec{\omega}_{i-1}$ and $\vec{\omega}_{i+1}$. Then:

$$\begin{aligned} K_{\delta\vec{\omega}_i} &= \frac{\sigma^2_{P\delta\vec{\omega}_i}}{\sigma^2_{P\delta\vec{\omega}_i}+\sigma^2_{M\delta\vec{\omega}_i}} \\ \delta\vec{\omega}_{C_i} &= \delta\vec{\omega}_{P_i} + K_{\delta\vec{\omega}_i}(\delta\vec{\omega}_{M_i} - \delta\vec{\omega}_{P_i}) \\ \sigma^2_{C\delta\vec{\omega}_i} &= \frac{\sigma^2_{P\delta\vec{\omega}_i}}{\sigma^2_{P\delta\vec{\omega}_i}+\sigma^2_{M\delta\vec{\omega}_i}} \end{aligned}$$

3. Update predicted quantities:

$$\begin{aligned} \hat{u}_{P_{i+1}} &= \hat{u}_{Ci} \\ \sigma^2_{P\hat{u}_{i+1}} &= \sigma^2_{M\hat{u}i}K_{\hat{u}i} \\ \vec{\omega}_{P_{i+1,j}} &= \vec{\omega}_{C_{i,j}} + \delta\vec{\omega}_{C_i}, \quad j = i-1, i, i+1 \\ \sigma^2_{P\vec{\omega}_{i+1,j}} &= \sigma^2_{C\vec{\omega}_{i,j}}K_{\vec{\omega}_{i,j}}, \quad j = i-1, i, i+1 \\ \delta\vec{\omega}_{P_{i+1}} &= \delta\vec{\omega}_{C_i} \\ \sigma^2_{P\delta\vec{\omega}_{i+1}} &= \sigma^2_{\delta\vec{\omega}_i}K_{\delta\vec{\omega}_i} \end{aligned}$$

4. i=i+1

Since μ values are unrelated over time (the μ values at the same image location at different times correspond to different 3d environmental points) they are simply computed in batch mode at each time using (12) or (13) and the current best estimates for $\hat{u}_C$ and $\vec{\omega}_C$ from the Kalman filter.

5 Experimental Technique

Since we use simulated data in our experiments we can measure the actual error in the motion and structure parameters. In the 3 equations below, we subscript variables with k to indicate known (correct) parameters and with m to indicate measured (computed) parameters. For $\hat{u}_m$ we compute angle error as

$$\theta_{\hat{u}} = \cos^{-1}(\hat{u}_m \cdot \hat{u}_k) \tag{19}$$

For $\vec{\omega}_m$ we compute relative or absolute error. If either $||\vec{\omega}_m||_2$ and $||\vec{\omega}_k||_2$ are zero we compute absolute error

$$\theta_{\vec{\omega}} = ||\vec{\omega}_m - \vec{\omega}_k||_2. \tag{20}$$

Otherwise we compute relative error as

$$\theta_{\vec{\omega}} = \frac{||\vec{\omega}_m - \vec{\omega}_k||_2}{||\vec{\omega}_k||_2} \times 100\% \tag{21}$$

We also use absolute or relative error to measure error, $\theta_{\delta\vec{\omega}}$ in $\delta\vec{\omega}$. We compute individual relative error for each μ value and then histogram the results.

6 Experimental Results

We used square regions of size $a = 3$ for velocity summing in the results reported here, although results for $a = 4$ and $a = 5$ were similar. We performed five experiments in total: (1) pure translation relative to a single plane, (2) general observer motion relative to a single plane, (3) general observer motion relative to multiple planes, (4) circular observer motion with no rotational acceleration in an enclosed room and (5) circular observer motion with constant rotational acceleration in the same enclosed room. Table 1 summarizes the motions. All translation directions are $(0, 0, 1)$ but

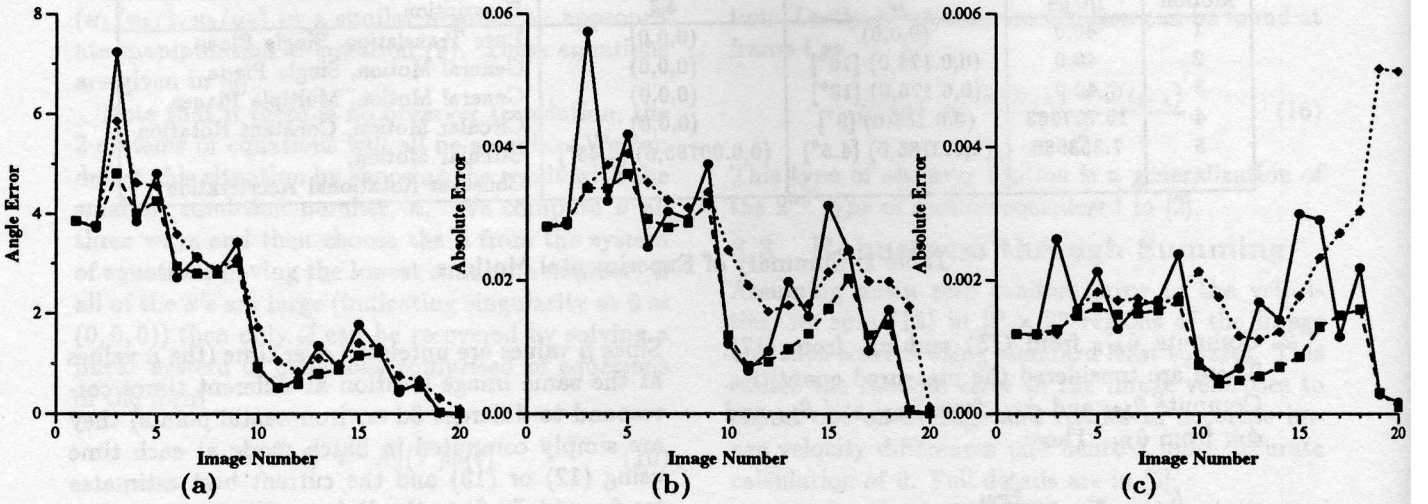

Figure 1: Exp 1: Measured (solid lines) and actual and residual Kalman filtered error (dashed and dotted lines) for (a) $\hat{u}$, (b) $\vec{\omega}$ and (c) $\delta\vec{\omega}$ for 5% random error for the single plane, pure translation sequence.

have various tangential speeds, $||\vec{U}||_2$ in f units per frame. Numbers in square brackets are the vertical rotation or rotational acceleration in degrees. For the first 3 experiments we use $\delta t = 0.1$ while for the last 2 experiments we use $\delta t = 1.0$. This allows 40 sets of parameters to be computed for each sequence. Note that our algorithm is completely general and can handle arbitrary translational directions. The experimental motion data in table 1 is specific to autonomous vehicle motion with a rigidly attached camera that is pointing direct ahead.

6.1 Single Planar Motion

Figure 1 show the results for the pure translation sequence. Each graph has 3 plots, the original measured parameter error as *solid* lines (computed from each consecutive set of 3 adjacent images) and 2 plots of the Kalman filtered parameter error (where the parameters are $\hat{u}$, $\vec{\omega}$ and $\delta\vec{\omega}$). As noted above, error in $\hat{u}$ is in degrees while error in $\vec{\omega}$ and $\delta\vec{\omega}$ are magnitudes (if the either the correct $\vec{\omega}$ or $\delta\vec{\omega}$ is zero) or relative error (if either the correct $\vec{\omega}$ or $\delta\vec{\omega}$ are non-zero). One of the Kalman plots (*dashed* lines) shows parameter error when the variances were computed as the actual error squared (a "perfect" variance measurement) while the other Kalman plot (*dotted* lines) shows parameter error when the residual squared is used as the variance.

Figure 2 show the results for the general motion sequence. Again, we have plotted the corresponding data as for the pure translation case, except this time $\vec{\omega}$ is non-zero and relative error is used to measure its error. For both experiments, there is a downwards trend in measured parameter error. The Kalman filter error for $\hat{u}$ follows the measured error, but suppresses large error increases. For $\vec{\omega}$ and $\delta\vec{\omega}$ Kalman filtering works especially well with error reductions of 50%-100% or more.

6.2 Multiple Planar Motion

In this experiment we use the same general motion parameters as above but use a ray-tracing program to generate multiple planes at depth ranging from 100 to 150 f units. Figure 3 show the results for this motion sequence. The results here are better than those for the general motion single plane sequence because there are 5 planes with significant depth variation. In this case, we can recover $\vec{\omega}$ with only 15%-20% error (compared to about 30% error for the single plane case).

6.3 Circular Motion

In the last 2 experiments we used a ray tracing program to simulate an observer moving on a circular path in a room. The room is simply a $300 \times 300 \times 300$ cube with the observer centered at $(0,0,0)$. This provides significant local depth variation, not present to the same degree in the other sequences. The maximum depth was $150\sqrt{2}$ f units and the minimum depth was 50 f units. The initial observer position is $(100,0,0)$ (a circular radius of 100 f units). The observer cannot see the top and bottom faces of the cube, but only the left, right, front and back faces that comprise the 4 walls of the room.

In the first circular sequence, the observer moved on the circle with a constant tangential and rotational motion. We generated 42 flows fields for this

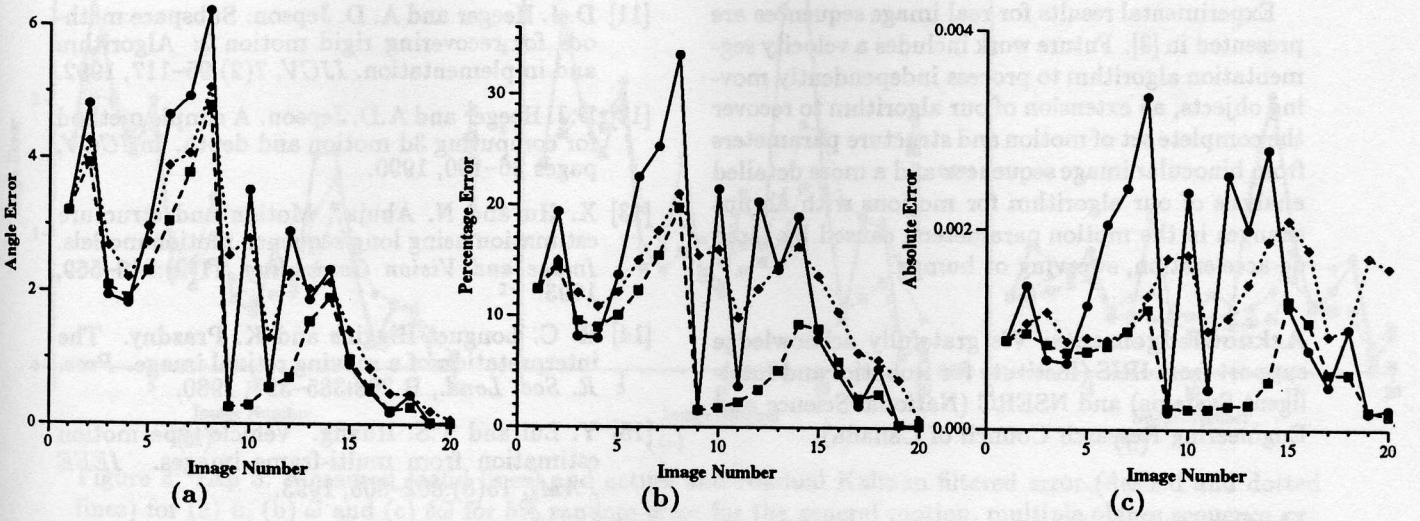

Figure 2: Exp 2: Measured (solid lines) and actual and residual Kalman filtered error (dashed and dotted lines) for (a) $\hat{u}$, (b) $\vec{\omega}$ and (c) $\delta\vec{\omega}$ for 5% random error for the general motion, single plane sequence.

sequence to allow 40 measurements (corresponding to one full revolution) to be obtained. Figure 4 show the results. From these graphs we can observe four distinctive "spikes". Motion and structure calculations are more accurate when there is large local depth variations (at the corners of the room) and very unstable for single planes (in between corners), as we saw above. The error "spikes" correspond to these latter calculations when 3 adjacent images have the same single surface in them. Again, the Kalman filter errors suppress large error increases in most cases.

The second circular sequence included constant observer rotation acceleration and results are in Figure 5. Because the rotational acceleration is 10% of the rotation it only takes a little while for both the observer's tangential speed and angular rotation to increase significantly. Thus only in the 1^{st} quarter of the circular motion are there 3 adjacent images with only a single wall visible (with no corners) and this results in the only error "spike" in the graphs. In the subsequent quarters of the circle a corner is always present in at least every third image, yielding more robust parameter calculations. The Kalman filter works especially well for error in $\delta\vec{\omega}$ with error improving by a factor of 5.

7 Conclusions and Discussion

1. The Kalman filtered parameters are more accurate that the measured parameters. For $\hat{u}$ we see that outlier measurements are suppressed and that the error in the filtered parameters is a bit less than the error in the corresponding individual measurements. This behaviour is especially evident when the $\vec{\omega}$ and $\delta\vec{\omega}$ parameters integrated over time using the Kalman filter.

2. The residual variance approximation is a good approximation to the error variance in most cases (which, in general, we would not know), with the exception of a few measurements in Figures 1c, 2c and 4c. In these cases, the residual variances incorrectly indicate good $\delta\vec{\omega}$ values while the actual measurements are not that good.

3. The algorithm is quite robust for random noise data. We have shown results for 5% random error but the algorithm can handle larger errors as well. For example, for 10% random error and single planar motion, we typically obtain errors of 2°-5° for $\hat{u}$ measurements but many individually computed rotation vectors have 50% or more error (no information). Even for 5% noise, some of the rotational error are quite large, but Kalman filtering can reduces them to the 15%-20% range. Rotational error of this magnitude was also observed in [11]. For the circular data 10% random error can easily be handled.

4. Surprisingly, the effect of more accurate motion parameters on the computation of relative depth parameters is minimal. The error is dependent on the specific motion and the error in the image velocities but independent of the motion parameter error and tends to remain constant over time for each sequences.

Experimental results for real image sequences are presented in [3]. Future work includes a velocity segmentation algorithm to process independently moving objects, an extension of our algorithm to recover the complete set of motion and structure parameters from binocular image sequences and a more detailed analysis of our algorithm for motions with abrupt changes in the motion parameters, caused by rapid de-acceleration, swerving or bumps.

Acknowledgements: We gratefully acknowledge support from IRIS (Institute for Robotics and Intelligent Systems) and NSERC (National Science and Engineering Research Council of Canada).

References

[1] C. Debrunner abd N. Ahuja. Motion and structure factorizing and segmentation of long multiple motion image sequences. In *ECCV*, pages 217–221, 1992.

[2] J. L. Barron, A. D. Jepson, and J. K. Tsotsos. The feasibility of motion and structure from noisy time-varying image velocity information. *IJCV*, 5(3):239–269, 1990.

[3] J.L. Barron and R. Eagleson. Recursive estimation of time-varying motion and structure parameters. *accepted by Pattern Recognition*, 1995.

[4] T.J. Broida and R. Chellappa. Estimation of object motion parameters from noisy images. *IEEE PAMI*, 8(1):90–99, 1986.

[5] S. Chandrashekhar and R. Chellappa. Passive navigation in a partially known environment. In *IEEE Workshop on Visual Motion*, pages 2–7, 1991.

[6] N. Cui, J.J. Weng, and P. Cohen. Recursive-batch estimation of motion and structure from monocular image sequences. *CVGIP: Image Understanding*, 59(2):154–170, 1994.

[7] W. O. Franzen. Structure and motion from uniform 3d acceleration. In *IEEE Workshop on Visual Motion*, pages 14–20, 1991.

[8] A. Gelb. *Applied optimal estimation.* MIT Press, 1974.

[9] A. Giachetti, M. Campani, and V. Torre. The use of optical flow for the autonomous navigation. In *ECCV*, pages 146–151, 1994.

[10] C. G. Harris and J. M. Pike. 3d positional integration from image sequences. *Image and Vision Computing*, 6(2):87–90, 1988.

[11] D. J. Heeger and A. D. Jepson. Subspace methods for recovering rigid motion 2: Algorithm and implementation. *IJCV*, 7(2):95–117, 1992.

[12] D.J. Heeger and A.D. Jepson. A simple method for computing 3d motion and depth. In *ICCV*, pages 96–100, 1990.

[13] X. Hu and N. Ahuja. Motion and structure estimation using long sequence motion models. *Image and Vision Computing*, 11(9):549–569, 1993.

[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] Y. Lui and T.S. Huang. Vehicle-type motion estimation from multi-frame images. *IEEE PAMI*, 15(8):802–808, 1993.

[16] L. Matthies, R. Szeliski, and T. Kanade. Kalman filter-based algorithms for estimating depth from image sequences. *IJCV*, 3(3):209–238, 1989.

[17] E. De Micheli, V. Torre, and S. Uras. The accuracy of the computation of optical flow and of the recovery of motion parameters. *IEEE PAMI*, 15(5):434–447, 1993.

[18] J. Oliensis and J.I. Thomas. Incorporating motion error in multi-frame structure from motion. In *IEEE Workshop on Visual Motion*, pages 8–13, 1991.

[19] C.J. Poelman and T. Kanade. A paraperspective factorization method for shape and motion recovery. In *ECCV*, pages 97–108, 1994.

[20] M. E. Spetsakis and J. Aloimonos. A multi-frame approach to visual motion perception. *IJCV*, 6(3):245–255, 1991.

[21] T.N. Tan, K.D. Baker, and G.D. Sullivan. 3d structure and motion estimation from 2d image sequences. *Image and Vision Computing*, 11(4):203–210, 1993.

[22] S. Uras, F. Girosi, A. Verri, and V. Torre. A computational approach to motion perception. *Biological Cybernetics*, 60:79–87, 1988.

[23] J. Weng, T.S. Huang, and N. Ahuja. 3d motion estimation, understanding and prediction from noisy image sequences. *IEEE PAMI*, 9(3):370–389, 1987.

[24] J.J. Wu, R.E. Rink, T.M. Caelli, and V.G. Gourishankar. Recovery of 3d location and motion of a rigid object through camera motion (an extended kalman filter approach). *IJCV*, 2(4):373–394, 1989.

[25] Z. Zhang and O.D. Frageras. Three-dimensional motion computation and object segmentation in a long sequence of stereo frames. *IJCV*, 7(3):211–242, 1992.

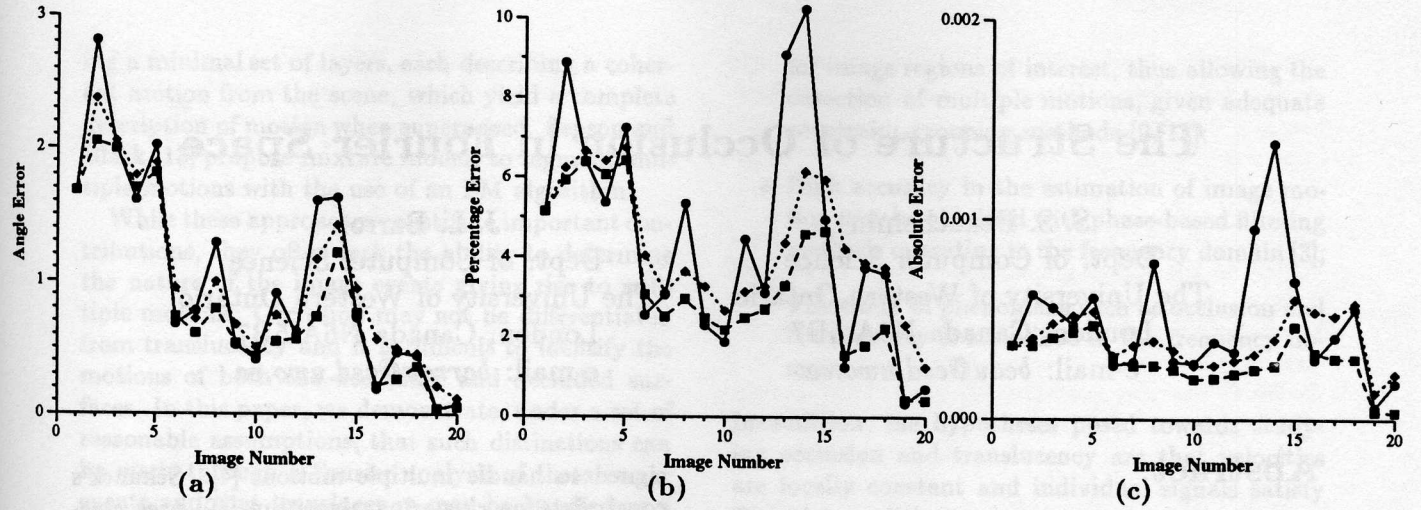

Figure 3: Exp 3: Measured (solid lines) and actual and residual Kalman filtered error (dashed and dotted lines) for (a) $\hat{u}$, (b) $\vec{\omega}$ and (c) $\delta\vec{\omega}$ for 5% random error for the general motion, multiple planes sequence.

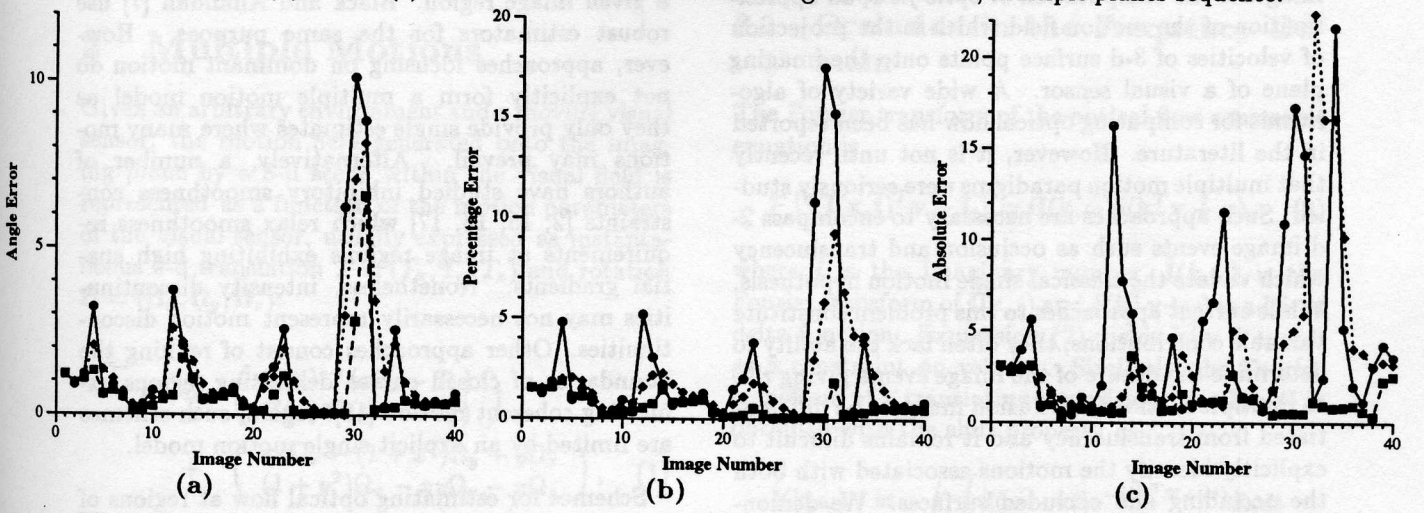

Figure 4: Exp 4: Measured (solid lines) and actual and residual Kalman filtered error (dashed and dotted lines) for (a) $\hat{u}$, (b) $\vec{\omega}$ and (c) $\delta\vec{\omega}$ for 5% random error for the circular motion, constant rotation sequence.

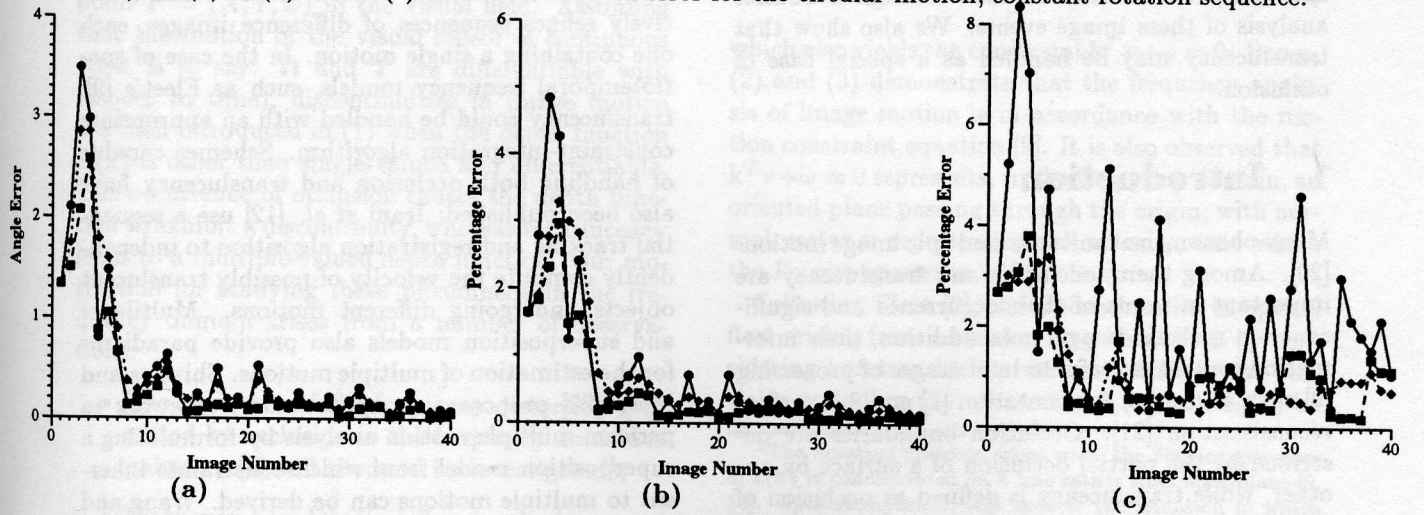

Figure 5: Exp 5: Measured (solid lines) and actual and residual Kalman filtered error (dashed and dotted lines) for (a) $\hat{u}$, (b) $\vec{\omega}$ and (c) $\delta\vec{\omega}$ for 5% random error for the circular motion, constant rotational acceleration sequence. Observe the good performance of both Kalman filtered plots for the noisy measurements.