BINOCULAR ESTIMATION OF MOTION AND STRUCTURE FROM LONG
SEQUENCES USING OPTICAL FLOW WITHOUT CORRESPONDENCE

J. L. Barron
Dept. of Computer Science
The 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 use the left and right monocular motion and structure parameters of two stereo image sequences (direction of translation, relative depth, observer rotation and rotational acceleration) to compute absolute depth, absolute translation and absolute translational acceleration for each pair of left and right images. Individual translation parameters computed at each frame are integrated over time using a Kalman filter to provide more accuracy and a “best” estimate of absolute translation at each time.

1. INTRODUCTION

Recently, we described a monocular motion and structure algorithm that computes the observer’s heading, \( \theta \); rotation, \( \omega \); rotational acceleration, \( \dot{\omega} \); and a relative-depth map (the ratio of translational speed and 3D depth \( \mu \) at each pixel in the image) by solving simple linear systems of equations [1, 2]. These parameters are computed in a camera-centered coordinate system using adjacent 3-tuples of flow fields from a long monocular flow sequence and are then integrated over time using a Kalman filter. The work described here extends that algorithm to use binocular flow sequences to compute, in addition to the monocular motion parameters for both left and right sequences, binocular parameters for absolute translation speed, \( U \), acceleration in translational speed, \( \dot{U} \) and absolute depth, \( X_z \) at each image pixel. The direction of translation is available from the monocular sequences and together with \( U \) and \( \dot{U} \) provides absolute observer translation.

Our algorithm does not require that the correspondence problem be solved between stereo images or stereo flows, i.e., no features or image velocities have to be matched between left and right frames, and it does not require an a priori surface structure model (the scene depth values can be arbitrary). The algorithm does require that the observer be rotating and that the spatial baseline be significant, otherwise only monocular parameters can be recovered. Despite this constraint, we point out that there are at least two applications where the observer rotations will generally be significant; that is, when a camera is mounted on one of the links of a joint-actuated robot arm or when it is mounted on a moving vehicle platform. As in the monocular case [1, 2], we use a nested Kalman filter to integrate the computed binocular motion parameters over time, thus providing a “best” estimate of the parameter values at each time.

2. LITERATURE SURVEY

There is an extensive body of research on recovering monocular motion and structure parameters from long image sequences [6, 8, 23, 10, 29, 16, 25, 31, 5, 22, 7]. Typically these use pixel, point or feature-based correspondence as the input and compute motion and structure parameters for monocular sequences. Due to depth-speed ambiguity, 3D depth and 3D translational speed cannot be recovered, only their ratio. A few algorithms assume stereo image sequences [17, 33] but usually for restricted circumstances (for example, pure translation). Some of the algorithms use some type of recursive estimation method (for example, an extended Kalman filter) [30, 4, 5, 8, 9]. 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 [18] uses the optical flow method of [26] to compute flow in long image sequences and then computes time-to-collision and angular velocity from the flow. A Kalman filter is used to integrate these parameters over time, improving the accuracy and robustness of the computed parameters. In recent work, we also used a Kalman filter to integrate more general motion and structure parameters (direction of translation, observer rotation and rotational acceleration) over time [1, 2]. A more complete set of parameters, which includes translational speed and acceleration and absolute depth, cannot be recovered without resorting to stereo imagery.

There is a smaller body of work on the use of binocular image velocity fields [19, 27, 13, 3, 28, 14, 24, 12, 20, 21, 32, 11]. One approach uses known optical flow and point correspondence to recover 3D motion and depth [19, 27], difference in image velocity in the left and right images for the same image location) and disparity to compute 3D information. Another approach uses relative image velocity differences in the left and right flow fields for the same image locations and disparity information to compute 3D information [28, 24]. Binocular motion and structure parameters can also be recovered from multiple stereo optical flow fields on a single planar patch [13] or directly from monocular mo-
tion and structure solutions computed for the separate left and right stereo image sequences, again for planar patches [3]. Optical flow has also been found helpful in establishing correspondence [12], which in turn can be used to triangulate depth. Motion stereo [20, 21, 32, 11] assumes known camera motion and uses optical flow (or disparity) to recover 3D motion. Under the assumption of known translation, a Kalman filter was used to recursively update depth values [17].

3. GEOMETRY OF PROJECTED MOTIONS FROM 3D TO 2D IN MONOCULAR IMAGES

The standard image velocity equation [15] relates an image velocity measured at image location \( \bar{y} = (y_1, y_2, 1) = \bar{P}/X_3 \), i.e. the perspective projection of a 3D point \( \bar{P} = (X_1, X_2, X_3) \), to the 3D observer translation \( \bar{U} \) and 3D observer rotation \( \bar{S} \). Assuming a focal length of 1 we can write the image velocity \( \bar{v} = (v_1, v_2) \) as

\[
\begin{align*}
\bar{v}(\bar{y}, t) &= \\
&= \begin{pmatrix}
-1 & 0 & y_1 \\
0 & -1 & y_2 \\
\end{pmatrix} \bar{w}(\bar{y}, t) \|ar{y}\|_2 \\
&+ \begin{pmatrix}
y_1 y_2 \\
1 + y_1 \bar{s} \\
\end{pmatrix}
\begin{pmatrix}
-1 & 0 & y_1 \\
0 & -1 & y_2 \\
\end{pmatrix} \bar{s}(t) \\
&\text{where } \bar{y} = \bar{U} = (u_1, u_2, u_3) \text{ is the normalized direction of translation and } \bar{w}(\bar{y}, t) = \mu(\bar{y}, t) \bar{u} \text{ is the depth-scaled observer speed.}
\end{align*}
\]

Since we are dealing with a monocular observer we cannot recover the observer’s absolute translation, \( \bar{U} \), or the actual 3D coordinates, \( \bar{P} \), of environmental points but rather the ratio of the two. We define relative depth as the ratio of 3D translation and 3D depth as

\[
\mu(\bar{y}, t) = \frac{\|ar{U}(t)\|_2}{\|ar{P}(t)\|_2} = \frac{\|ar{U}\|_2}{X_3 \|ar{y}\|_2}
\]

where \( \bar{u} = \bar{U} = (u_1, u_2, u_3) \) is the normalized direction of translation and \( \mu(\bar{y}, t) \) is the depth-scaled observer speed.

4. GEOMETRY OF PROJECTED MOTIONS FROM 3D TO 2D IN BINOCULAR IMAGES

![Figure 1: The binocular setup. Subscripts L and R are used to indicate variables in the left and right image sequences. \( \bar{s} \) is the spatial baseline (the two camera are rigidly connected) and is assumed known.](image-url)

In our binocular setup a second camera is rigidly attached to the first camera with a known baseline \( \bar{s} \) as shown in Figure 1. We subscript variables in the left image with \( L \) and variables in the right image with \( R \). Our solution depends on using left and right monocular solutions [1, 2] computed from separate left and right long image sequences. That is, both left and right cameras have common translational direction, \( \bar{a} \), the same rotation and rotation acceleration, \( \bar{\omega} \), and \( \bar{\delta} \), but different relative depth \( \mu_L \) and \( \mu_R \), different 3D translational speeds, \( U_L \) and \( U_R \), and different translational accelerations, \( \bar{U}_L \) and \( \bar{U}_R \). Of course, \( U_L \) and \( U_R \) can be computed from \( U_L \) and \( U_R \) and \( \bar{a} \). We use the relationships between \( U_L \) and \( \bar{U}_L \) and \( U_R \) and \( \bar{U}_R \) that use the values of \( \bar{\omega} \) and \( \bar{\delta} \) to avoid the correspondence problem. Given 3D translational information, \( \bar{U}_L \) and \( \bar{U}_R \), we can compute 3D depths, \( X_{3L} \) and \( X_{3R} \) or equivalently the 3D coordinates, \( \bar{P}_L = X_{3L} \bar{y} \) and \( \bar{P}_R = X_{3R} \bar{y} \). We give the main relationships below.

The physical stereo setup allows us to write

\[
\bar{U}_R = \bar{U}_L + \bar{\omega} \times \bar{s}
\]

Thus \( \bar{a}_L \) and \( \bar{a}_R \) can be written as

\[
\begin{align*}
\bar{a}_L &= \frac{\bar{U}_L}{X_{3L} \|ar{y}\|_2} \\
\bar{a}_R &= \frac{\bar{U}_R}{X_{3R} \|ar{y}\|_2}
\end{align*}
\]

Using \( \bar{a}_L \) and \( \bar{a}_R \) in the left and right image velocity equations, we can subtract the image velocity at \( \bar{y} \) to obtain

\[
\begin{align*}
\bar{v}_R(\bar{y}) - \bar{v}_L(\bar{y}) &= \\
&= \begin{pmatrix}
-1 & 0 & y_1 \\
0 & -1 & y_2 \\
\end{pmatrix} \bar{a}_L \|ar{y}\|_2 \bar{a}(\bar{y}) \\
&+ \begin{pmatrix}
-1 & 0 & y_1 \\
0 & -1 & y_2 \\
\end{pmatrix} \bar{a}_R \|ar{y}\|_2 \bar{a}(\bar{y})
\end{align*}
\]

These are 2 linear equations in 2 unknowns, \( \bar{a}(\bar{y}) \) and \( \bar{b}(\bar{y}) \), where \( b = \frac{\bar{a}}{X_{3R}} \) and \( a = X_{3L} b \). Note that if there is no observer rotation, i.e. \( \bar{\omega} = 0 \), \( b \) cannot be recovered (only the parameter \( a \) can be computed), and as a result absolute depth cannot be computed. A yields the ratio of depth in the left and right images at each image point. Given the monocular parameters \( \bar{a}, \bar{\omega} \) and \( \mu_L \) and \( \mu_R \) (computed separately from the left and right image sequences) we can compute \( a \) and \( b \) for each image point (\( \bar{s} \) is known). We compute absolute depth values \( X_{3L} \) and \( X_{3R} \) and then compute absolute translations \( \bar{U}_L \) and \( \bar{U}_R \) as

\[
\begin{align*}
\bar{U}_L &= X_{3L} \|ar{y}\|_2 \bar{a}_L \\
\bar{U}_R &= X_{3R} \|ar{y}\|_2 \bar{a}_R
\end{align*}
\]

Each image point potentially yields a different \( \bar{U} \) value. We average the individually computed \( \bar{U}_L \) and \( \bar{U}_R \) at each time in the stereo images and compute their standard deviations. We can then compute constant acceleration in translational speed in the left and right images using averaged \( \bar{U}_L \) and \( \bar{U}_R \) values as

\[
\begin{align*}
\delta\bar{U}_L &= \|ar{U}_L(t + \delta t) - \bar{U}_L(t)\|_2 \\
\delta\bar{U}_R &= \|ar{U}_R(t + \delta t) - \bar{U}_R(t)\|_2
\end{align*}
\]

Then \( \delta\bar{U}_L = \delta\bar{U}_L \bar{a} \) and \( \delta\bar{U}_R = \delta\bar{U}_R \bar{a} \).
5. KALMAN FILTERING

The 3D depth and translation can, in principle, be determined for every \( \hat{Y} \) location. Since speeds \( U_L \) and \( U_R \) should be the same for every location we compute the average and variance of these values at every image location, but noise can seriously corrupt individual speeds. Some were obviously wrong, i.e. \( \|U_L\|_2 > 1000.0 \) or \( \|U_R\|_2 > 1000.0 \), and so our algorithm classified them as outliers and removed from further consideration. We use the least squares variance \( \sigma_{U_L}^2 \) as the variance in our Kalman filter calculations. We subscript variables with \( M \) for measured quantities (computed from individual stereo flows), with \( C \) for quantities computed using a Kalman update equation and with \( P \) for predicted quantities. The steps of the Kalman filter are:

1. Initialize predicted parameters:
   \[
   U_{iP} = 0, \quad \sigma_{U_{iP}}^2 = \infty, \quad i = 1.
   \]

2. Compute \( X_3 \) values and \( U_{LM} \) from (4) and (5), average \( U_L \) and compute its standard deviation. Then:
   \[
   U_{iL} = \frac{\sigma_{U_{iL}}^2}{\sigma_{U_{iMF}}^2 + \sigma_{U_{iLM}}^2}, \quad U_{iC} = U_{iP} + K_{iL}(U_{LM} - U_{iP}) \quad \text{and} \quad \sigma_{iC}^2 = K_{iL}\sigma_{iLM}^2.
   \]

3. Update predicted quantities:
   \[
   U_{iP} = U_{iC}, \quad \sigma_{U_{iP}}^2 = K_{iL}\sigma_{iLM}^2, \quad i = i + 1.
   \]

These steps are performed for each pair of speeds recovered from each stereo flow pair. We also ran the Kalman filter with the actual error squared as the variance for the purpose of comparison.

6. EXPERIMENTAL RESULTS

A Kalman filter was applied to the \( U_L \) and \( U_R \) estimates at each time to recursively refine them and decrease their uncertainty. We present experimental results for one stereo sequence that consists of 22 stereo flows. The left sequence was generated for \( \vec{U}_L = (0, 0, 40) \) with an angular rotation of 0.174 radians. The initial observer position was \( (0, 0, 0) \) and the flow was sampled at every \( 1/30 \) time unit. The right sequences were generated using a spatial baseline of \( \vec{s} = (10, 0, 0) \) units, yielding \( \vec{U}_R = (-1.74, 0, 41.74) \). We could not use \( \vec{s} = (10, 0, 0) \) because then both \( U_L \) and \( U_R \) were in the same direction and equation (4) is singular. As well, \( X_{3L} \) and \( X_{3R} \) have to be different at the same image location or equation (4) cannot be solved there. The images consisted of a number of ray-traced planes viewed in stereo at various depths. The 22 stereo flows allow 20 left and right monocular motion and structure calculations to be performed [1, 2]. Figures 2 and 3 show the measured error in speeds \( U_L \) and \( U_R \) for 1% random error in the left and right image velocity fields (we can use up to 3% random error before speed error approaches 50%). The average measured error ranges from 2% to 15%. The Kalman filtering suppresses bad speed measurements. The performance for the actual variance and least squares variance was nearly identical.

7. DISCUSSION

The recovery of binocular parameters from stereo flow fields is very sensitive to noise in the flow vectors. This is because the differences in similar stereo image velocities can have large error even if the individual image velocity error
is quite small. We are currently investigating the direct application of Kalman filtering on both relative and absolute depth calculations and the application of velocity filtering and segmentation on the optical flow fields to produce more accurate depth values. Other pending work includes using a local depth model to make depth calculations more robust and testing our algorithms using image sequences acquired from a camera mounted on a robot arm.

Acknowledgments Each author gratefully acknowledges support from NSERC (Natural Sciences and Engineering Research Council of Canada).

8. REFERENCES