MOTION ESTIMATION WITH INCOMPLETE INFORMATION USING OMNI-DIRECTIONAL VISION

Jong Weon Lee and Ulrich Neumann

Computer Science Department
Integrated Media Systems Center
University of Southern California
Los Angeles, CA 90089-0781, USA
{jonlee | uneumann}@graphics.usc.edu

ABSTRACT

We present a new motion estimation framework and apply it to omni-directional imagery. Our method estimates motions incrementally using an Implicit Extended Kalman Filter (IEKF). Each individual feature provides partial information about the camera motion. The motion estimate is incrementally improved as each feature is processed, similar to the SCAAT approach. The SCAAT method was developed for calibrated features and full 6DOF-pose tracking whereas our method estimates 5DOF translation and rotation motions concurrently from uncalibrated features based on the rigidity and the depth independent constraints. The main difference of our method from others is the combination of a recursive estimation framework in an IEKF and the constraints used in motion estimation.

1. INTRODUCTION

We capture panoramic views in a single image using an omni-directional camera. It is well known that motion estimation using a perspective camera presents difficulty distinguishing between motions generated from a small pure translation and a small pure rotation [2]. This confusion no longer exists with the large field of view achieved by omni-directional cameras. Researchers have developed these methods for estimating motions using omni-directional cameras. Gluckman and Naya showed three algorithms, each developed for planar perspective cameras, applied to projections from omni-directional images [3]. Svoboda applied an 8-point algorithm to estimate motions from omni-directional images [8]. Even with the lower resolution of omni-directional images than that of traditional perspective images, the motion estimation methods using omni-directional cameras overcome problems associated with traditional perspective cameras and often result in better accuracy than methods using perspective cameras.

We developed a new motion estimation method for omni-directional images that offer some advantages over previously mentioned methods. The presented method estimates motions incrementally using the IEKF. Each individual feature provides partial information about the camera motion. We incrementally improve motion estimates with each feature as in the SCAAT method [12]. However, the SCAAT method was developed for calibrated features and full 6DOF-pose tracking. Our method estimates 5DOF translation and rotation motions concurrently from uncalibrated features based on the rigidity and the depth independent constraints. The depth independent constraint is originally introduced by Bruss and Horn [1] and modified for the spherical motion field by Gluckman and Nayar [3]. The main difference of our method from previously mentioned approaches is the combination of the recursive estimation framework adapted from the IEKF and the constraints used in motion estimation.

2. MOTION ESTIMATION

An omni-directional camera projects the 360-degree view of the world to a single 2D image. 3D world points can be mapped to points on the unit sphere, creating the spherical projection [11]. The mapping function depends on the mirror and optics used in an omni-directional camera. Few mapping functions can be found in [3].

A point P in the 3D world is represented as a point p on the unit sphere using the spherical projection

$$p = \frac{P}{\|P\|}. \tag{1}$$

The motion of a 3D point P relative to a moving camera can be represented as the instantaneous velocity of P translating along an axis T and rotating along an axis Ω.

$$\dot{P} = -T - \Omega \times P \qquad (2)$$

The motion field equation is derived using the derivative of the spherical projection (Equation 1) with respect to time and substituting it into Equation 2. The derived velocity vector equation at point p is

$$U(p) = \frac{1}{\|P\|}\left((T \bullet p)p - T\right) - \Omega \times p. \qquad (3)$$

Gluckman and Nayar remove a depth from the velocity vector and derive the depth independent constraint [3]

$$T \bullet \left(p \times \left(U + (\Omega \times p)\right)\right) = 0. \qquad (4)$$

We estimate translation and rotation motions concurrently from the set of motion vectors U_i measured at points p_i. The measurements of both U_i and p_i are assumed to contain some error, which we model as Gaussian, white and zero-mean noise.

$$q_i = p_i + n_i \quad n_i \in N\left(0, \textstyle\sum_{n_i}\right), \qquad (5)$$
$$q_i' = U_i + n_i' \qquad (6)$$

A translation velocity T is represented as $\alpha=[\theta,\phi]^T$, a point on the unit sphere: θ is the azimuth angle and ϕ is the elevation. An axis Ω is represented as three parameters, Ω_x, Ω_y, and Ω_z. We derive a nonlinear implicit dynamic system in Equation 7 based on the depth independent constraint and noise measurements.

$$\begin{cases} T(\alpha) \bullet D(p, U, \Omega) = 0 \\ q_i = p_i + n_i \end{cases} \qquad (7)$$

T and Ω are unknown parameters in the above system, which is subject to three constraints. First, $T(\alpha)$ is the point on the unit sphere. Second, the parameters T and Ω have to evolve in such a way that measurements U satisfy the "a-posteriori" dynamics,

$$T(\alpha) \bullet D(q_i, q_i', \Omega) = \tilde{n} \qquad (8)$$

where $\tilde{n}$ is a residual noise from a measurement noise n. The last constraint is called "a-priori" dynamics. T and Ω are not changing arbitrarily. They have to follow the model described by the application. We use simple statistical model, the first order random walk described in Equation 9 because we lack a mechanical model.

$$\begin{cases} f_\alpha(\alpha) = \alpha \\ f_\Omega(\Omega) = \Omega \end{cases} \qquad (9)$$

We derive a discrete dynamic model for the unknown parameters by applying the constraints in Equation 10.

$$\begin{cases} \alpha(t+1) = f_\alpha(\alpha(t)) + n_\alpha(t) \\ \Omega(t+1) = f_\Omega(\Omega(t)) + n_\Omega(t) \\ G(\alpha, q_i, q_i', \Omega) = T(\alpha) \bullet D(q_i, q_i', \Omega) = \tilde{n} \end{cases} \qquad (10)$$

We use Equation 10 to derive equations for the EKF. The only difference is the measurement equation, which is implicit in our approach.

Measurement update ("correction") step

$$\begin{cases} X(t+1 \mid t+1) = X(t+1 \mid t) + \\ \quad L(t+1)G\left(\alpha(t+1 \mid t), q_i(t+1), q_i'(t+1), \Omega(t+1 \mid t)\right) \\ P(t+1 \mid t+1) = B(t+1)P(t+1 \mid t)B^T(t+1) + \\ \quad L(t+1)D(t+1)R(t+1)D^T(t+1)L^T(t+1) \end{cases}$$

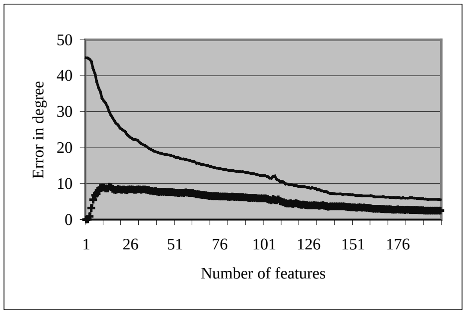
where

$$\begin{cases} A(t+1) = C(t+1)P(t+1 \mid t)C^T(t+1) + \\ \quad D(t+1)R(t+1)D^T(t+1) \\ L(t+1) = P(t+1 \mid t)C^T(t+1)A^{-1}(t+1) \\ B(t+1) = I - L(t+1)C(t+1) \\ C(t+1) = \dfrac{\partial G}{\partial \alpha} \\ D(t+1) = \dfrac{\partial G}{\partial [q, q']} \end{cases}$$

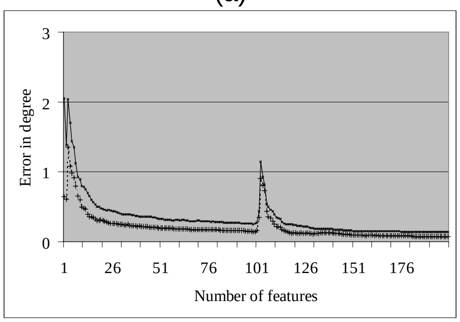
and R is the covariance matrix of the measurement error.

There are several advantages in our approach over previously mentioned methods. Since the filter incrementally improves the estimation with each individual feature, the presented method generates estimates more frequently and with lower latency than other estimation methods. It also does not require special cases when there are not enough features to fully constrain the motion estimate. Any number of features (>1) tracked between two consecutive frames will refine the pose estimate. If the features are corrupted by independent noise, then incorporating them independently can offer improved filtering over the methods that use all tracked feature concurrently.

Most motion estimation methods use all tracked features from two consecutive images once. We can use the same set of features more than once to improve estimates. Each new estimate depends on the prior state and the measurement quality of an individual feature. We improve the motion estimate from the initial state by using all tracked features independently for the first iteration. The result of the first iteration is used as the initial state for the second iteration where we again use the same set of features. In processing each feature, only the initial state differs from the previous iteration. Iterating with the same set of features produces improved final motion estimates because of the better initial estimate obtained from the previous iteration. Iterations continue until the estimate converges. The number of iterations needed depends on the number of available tracked features. We typically iterate five times with 20-40 tracked features from two consecutive images. This iteration capability is very

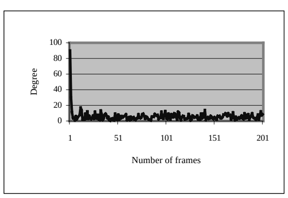


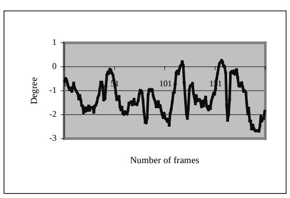
(a)



(b)

Figure 1. Estimating a motion using an individual feature. The solid line and the dashed line with "+" indicate the average and the standard deviation respectively. (a) The angular deviation of translation motions (b) The difference between the correct and estimated rotation angles.





(a) (b)

Figure 2. Motion estimates for general camera motion. (a) The angular deviation of translation motions (b) The angular difference of rotation angles.

important since 2D feature tracking is one of the computationally expensive processes in motion estimation systems. We can generate valid estimates with fewer tracked features using the iteration procedure. For example, the estimate generated by two iterations of 50 features is very similar to the estimate generated with one iteration of 100 features.

3. SIMULATION

Simulation results show the presented method is robust and flexible. Since we use a Kalman filter, we need an

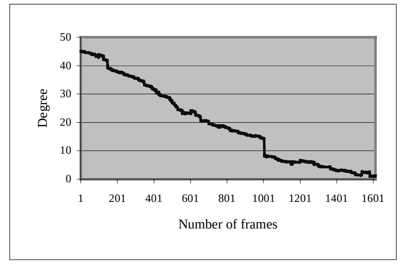
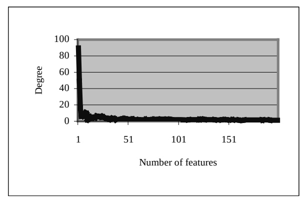
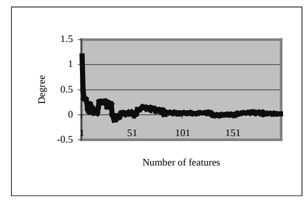


Figure 3. With only one tracked feature, useful motion estimates are still obtained.





(a) (b)

Figure 4. Motion estimation of near pure rotation. (a) The angular deviation of rotation axes (b) The angle difference of rotation angles.

initial estimate. For the following simulations, the initial estimate errors are 90 and 45-degrees. The rotation is modeled as a rotation axis and a rotation angle about that axis. Translation errors are represented as an angular deviation between the correct and estimated translation directions. Rotation errors are represented as an angular deviation between the correct and estimated rotation axes and an angle difference between the correct and estimated rotation angles. When we estimate motions for the streams of images, the correct and estimated absolute camera orientations are compared for each image. The absolute orientations are computed by integrating the rotation estimates between all prior images as well as the current image pair. The simulations using the streams of images show accumulation errors.

The simulation is based on features distributed in a 2m square room with the omni-directional camera located at the center. In the first simulation, we demonstrate how each successive feature contributes to the overall estimate. A hundred features are generated between two consecutive images and used twice to estimate motions. The applied motion is a 2cm translation and a 3-degree rotation about the up-axis. Gaussian white noise with 1-pixel standard deviation is added to the projected feature positions. The simulation is executed a hundred times. The average and the standard deviation of translation errors are approximately 5 and 2.5-degrees respectively. For rotation axes, the average error is 2.4-degrees and the standard deviation is 1.2-degrees. The average and the standard deviation of rotation angle errors are 0.15 and 0.07-degrees respectively.

In the second simulation, we tracked 40 features between two consecutive frames. The applied motion is a 2cm/frame translation and a sinusoidal rotation of 3-degree maximum about the up-axis. We start the simulation from an initial estimation that is about 90 degrees apart from the correct translation direction and rotation axis. Gaussian white noise with 1-pixel standard deviation is added to simulate tracking error. The results are presented in Figure 2 showing that only a few frames are needed to converge to a nearly correct motion.

The third simulation demonstrates that the new method can still estimate motions even with only one tracked feature between two consecutive frames, under the assumption that the motion is constant over the long sequence. This single tracked feature is randomly selected for each image pair. The applied motion is a pure translation of 2cm/frame. The initial estimate is almost 45-degrees from the correct translation vector. The result in Figure 3 describes that the estimate slowly converges to the correct value. Clearly, varying motions cannot be estimated with only one tracked feature because it is severely underconstrained. However, even a single tracked feature can yield a useful improvement in our approach.

In the last simulation, we demonstrate that the new method can estimate rotations correctly even with very small translation motions. Some motion estimation methods estimate translations first and then rotations. These methods often produce errors for motions close to a pure rotation since the translation estimates are error prone. The simulation in Figure 4 shows that small translations do not cause problems in our method even if we use the depth independent constraint that is strongly dependent on the translation direction. The motion for Figure 4 is a 3-degree rotation about the up-axis with a small translation motion.

4. CONCLUSION

We combine several ideas in a novel fashion to achieve stable 5DOF camera motion estimates for omni-directional cameras. Our estimation framework uses an IEKF that processes each input feature sequentially. A novel characteristic of this approach is its ability to reuse the feature motions obtained for a single video frame to improve the motion estimate. To the best of our knowledge, the whole idea of incremental 5DOF motion estimation is a new approach to this problem. Its use with omni-directional imaging appears particularly appropriate given the stability and robustness of the results obtained, even in the case of relatively pure rotations that create difficulties for some methods.

5. REFERENCES

[1] A. Bruss and B. K. P. Horn, "Passive navigation," *Computer Vision, Graphics, and Image Processing*, vol. 21, pp. 3-20, 1983.

[2] K. Daniilidis and H.-H Nagel, "The coupling of rotation and translation in motion estimation of planar surfaces," *IEEE Conference on Computer Vision and Pattern Recognition*, New York, NY, pp. 188-193, June 1993.

[3] Joshua M. Gluckman and Shree K. Nayar, "Egomotion with omnicameras," *IEEE International Conference on Computer Vision*, Bombay, India, 1998.

[4] Richard I. Hartley, "In defense of the 8-point algorithm," *5th International Conference on Computer Vision*, IEEE Computer Society Press, Cambridge Massachusetts, pp. 1064-1070, 1995.

[5] Thomas S. Huang and Arun N. Netravali, "Motion and structure from feature correspondences: a review," *Proceedings of the IEEE*, vol. 82, No. 2, pp. 251-268, February 1994.

[6] Michal Irani, Benny Rousso and Shmuel Peleg, "Recovery of ego-motion using image stabilization," *CVPR-94*, Seattle, pp. 454-460, June 1994.

[7] Stefano Soatto and Pietro Perona, "Recursive 3-D motion estimation using subspace constraints," *International Journal of Computer Vision*, 1997.

[8] Tomas Svoboda, Tomas Pajdla and Vaclav Hlavac, "Motion Estimation Using Central Panoramic Cameras," *IEEE Conference on Intelligent Vehicles*, Stuttgart Germany, October 1998.

[9] Roger Y. Tsai and Thomas S. Huang, "Uniqueness and estimation of three-dimensional motion parameters of rigid objects with curved surfaces," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-6, No. 1, pp. 13-27, January 1984.

[10] Juyang Weng, Thomas S. Huang and Narendra Ahuja, "Motion and structure from two perspective views: algorithms, error analysis, and error estimation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, No. 5, pp. 451-476, May 1989.

[11] B. L. Yen and T. S. Huang, "Determining 3-D motion and structure of a rigid body using the spherical projection," *Computer Vision, Graphics, and Image Processing*, vol. 21, pp. 21-32, 1983.

[12] Greg Welch and Gary Bishop, "SCAAT: Incremental Tracking with Incomplete Information," *SIGGRAPH97*, August 1997.