Large Motion Estimation for Omnidirectional Vision

Jong Weon Lee, Suya You, and Ulrich Neumann
Computer Science Department
Integrated Media Systems Center
University of Southern California
Los Angeles, CA 90089-0781, USA
{jonlee | suyay | uneumann}@graphics.usc.edu

Abstract

We present a method for estimating large camera motions, including combinations of large rotations and displacements. We know that translation estimates are improved as the displacements between two camera positions increase. However, there is little work on motion estimation methods targeted at large motions since it is difficult to find correspondences between two planar perspective images under large motions. We can overcome the correspondence problem with omnidirectional cameras, which obtain a hemisphere or cylinder projection of the environment. We first develop a new motion estimation method for smooth omnidirectional camera motions. Our method incrementally improves estimates from incomplete information provided by a single feature correspondence, using an Implicit Extended Kalman Filter (IEKF). This general motion estimation method produces stable estimates from smooth camera motions, but it is not directly applicable to estimating large motions. We adapt the motion estimation method to large motions by using a novel Recursive Rotation Factorization (RRF) that removes the image motions due to rotation. Simulation results show that the RRF large motion estimates are accurate and robust.

INTRODUCTION

Motion estimation techniques are divided into two categories, based on their use of optical flows or feature correspondences. Both techniques are used to estimate motions from planar projection images that only offer a limited field of view. Due to the lack of global information, estimates from perspective images are very sensitive to noise when the translation direction is outside of the field of view. It is well known that estimation methods have difficulty distinguishing between small pure translations and small pure rotations [3], which often occur between images obtained by smooth camera motions. Figure 1(a, b) shows examples of similar motion flows for pure translation and pure rotation motions obtained by a planar projection. The motions of Figure 1(a) are produced by a small translation motion parallel to the image plane and (b) by a small rotation motion around the axis perpendicular to the optical axis.

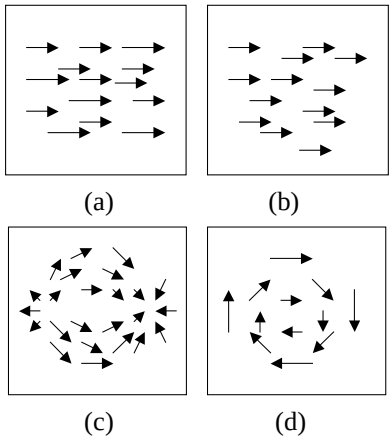


Figure 1. Motion flow for planar perspective and omni-directional images. (a) Translation motion parallel to a scene with a perspective camera (b) Rotation motion around the up-axis with a perspective camera (c) Translation motion parallel to a scene with an omni-directional camera (d) Rotation motion around the up-axis with an omnidirectional camera

Recently developed omnidirectional cameras image 360-degrees horizontally and at least 90-degrees vertically. The omnidirectional images contain global information so motion estimation techniques can overcome the limited fields of view introduced by planar projections. Omnidirectional images easily distinguish between small translations and rotations, as illustrated in Figure 1(c, d).

Researchers have developed methods for estimating motions using the omnidirectional cameras. Gluckman and Nayar showed three algorithms, each developed for planar perspective cameras, applied to projections from omnidirectional images [6]. Svoboda applied the 8-point algorithm [7] [8] to omnidirectional images [10]. Many have observed that omnidirectional vision overcomes

problems associated with traditional perspective cameras, often producing more accurate results.

The Kalman filter is used for several 5DOF (degree of freedom) motion estimation methods using images from perspective cameras. Gennery [5], Dickmanns [4], and Azarbayejani [1] used Extended Kalman filters to recover the shape and motion parameters from image sequences. Soatto developed the motion estimation methods based on essential and subspace constraints using the Implicit Extended Kalman filter [9]. They demonstrated that motion estimates were improved by applying Kalman filters to streams of images.

We develop a new motion estimation technique for omnidirectional images. The new technique estimates motions incrementally using an Implicit Extended Kalman Filter (IEKF). Each individual feature provides partial information about the camera motion. We incrementally improve motion estimates with each feature as in the SCAAT method [15], however the SCATT method was developed for calibrated features and full 6DOF-pose tracking. Our technique estimates 5DOF translation and rotation motions concurrently from uncalibrated features, based on the rigidity and the depth independent constraints originally introduced by Bruss and Horn [2] and derived for the spherical motion field by Gluckman and Nayar [6]. The main differences of our new techniques from previously mentioned approaches are a combination of the recursive estimation framework adapted from the Implicit Extended Kalman Filter and the constraints used in motion estimations.

There are several advantages of our approach over previously mentioned techniques. Since the filter incrementally improves the estimation with each individual feature, it generates estimates more frequently and with lower latency than other approaches. It also does not require any special case processing when there are not enough features to fully constrain the motion estimate. Any number of features tracked between two consecutive images will refine the motion estimate to the degree possible. If feature motions are corrupted by independent noise, then incorporating them independently can offer improved filtering over techniques that use all tracked features concurrently. We can also reuse the same set of features to improve estimates. This property is desirable since computing correspondences between two consecutive images is often the major time-consuming part of a motion estimation process.

A major problem with motion estimation techniques is that translation direction estimates are often unreliable. It is well known that translation estimates can be improved using large displacements. However, most researchers have not been considered mixtures of large displacements and rotations. This class of motions is referred to as *large motion* in this paper for convenience. Note that large motion estimation is rarely discussed in the context of planar projection cameras since correspondences are easily lost under large motions. Also, we note that optical flow methods are difficult to apply because of the extensive image displacements and parallax effects that occur with large motions.

Experience and reason show that omnidirectional imagery allows 2D-feature correspondence tracking over large motions due to the global view of the environment. Svoboda also mentions in his thesis that large translation motions are estimated better using omnidirectional rather than perspective cameras [11].

While our estimation method works well for small motions, problems arose when it was directly applied to omnidirectional images taken from a camera undergoing large motions. While most rotation estimates were accurate, the translation estimates were unreliable. Poor translation estimates were obtained when large rotations were present. Image-feature motions caused by translations are relatively small compared to the image motions caused by rotations. We observed significant translation direction errors in the presence of large rotations. To solve this problem we employ a Recursive Rotation Factorization (RRF) with our smooth motion algorithm. The resulting large motion algorithm can generate accurate and robust estimates of camera motion between any two panoramic images for which correspondences are known.

5DOF MOTION ESTIMATION

Algorithm

An omnidirectional camera projects a 360-degree view of the world to a single 2D image. 3D world points can be mapped to points on the unit sphere, creating a spherical projection. The mapping function depends on the mirror and optics used in the omnidirectional camera. Examples can be found in [6]. The point P in the 3D world is represented as the point p on the unit sphere using the spherical projection

$$p = \frac{P}{\|P\|}. \qquad (1)$$

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 a rotation about axis Ω.

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

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

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

Gluckman and Nayar remove a depth from the velocity vector and derive the depth independent constraint [6]

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

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

$$q_i = p_i + n_i \quad n_i \in N(0, \Sigma_{n_i}), \qquad (5)$$

$$q_i^{'} = U_i + n_i^{'} \qquad (6)$$

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. 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 the measurement noise n. The last constraint is called "a-priori" dynamics. T and Ω are not change arbitrarily. They have to follow the model described by the application. We use very 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 Extended Kalman Filter (EKF). The only difference is the measurement equation, which is implicit in our approach. The equations used for estimating general motions using the first order random walk model follow.

Time update ("prediction") step

$$\begin{cases} X(t+1|t) = X(t|t) \quad X(0|0) = X_0 \\ P(t+1|t) = P(t|t) + \Sigma_{\alpha\Omega} \quad P(0|0) = P_0 \end{cases}$$

where the state vector $X=[\alpha,\Omega]$.

Measurement update ("correction") step

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

where

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

and R is the covariance matrix of the measurement error.

We estimate motions using IEKF described above. Most methods using Kalman Filters collect all features found from images at time t and t+1 before estimating the motion. Our method, on the other hand, incrementally estimates the motion from each individual feature, as described in the SCATT method [15]. The SCATT method was developed to use with calibrated features measured at varying times to compute a full 6DOF camera-pose. Our technique estimates 5DOF translation and rotation motions concurrently from uncalibrated features that are observed in snapshot of time. In both cases individual features provide incomplete information about the camera motion.

Our approach has some advantages over filter methods that use all detected features at the same time. We generate estimates more frequently and with lower latency than methods requiring all feature correspondences before updating the estimate. We need no special cases for varying numbers of corresponded features. Motions are estimated for any number, including one, of tracked features between two images. Our method never fails for lack of constraints; its accuracy degrades gracefully. If feature motions are corrupted by independent noise, then incorporating them independently can offer improved filtering over other techniques that use all tracked features concurrently.

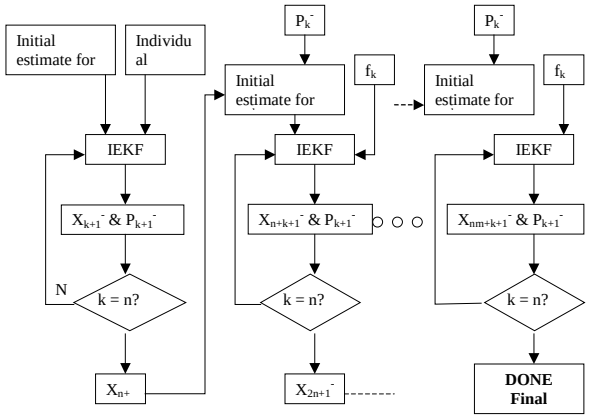
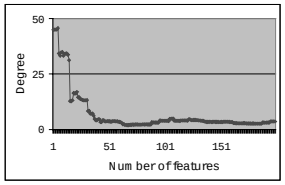
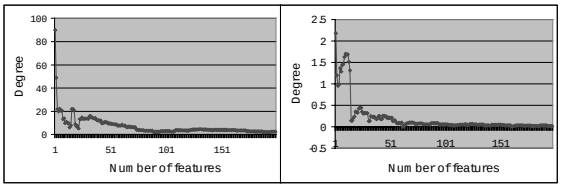


Figure 2. 5DOF motion estimation. X and P indicate the state estimate and the error covariance respectively. f is one feature from the set with n features. m is the number of iterations used for the same set of features

Although, each point can be processed independently, our algorithm can concurrently apply any number of features. For example, improved estimates are obtained under some conditions by processing five features concurrently. Motion estimates are usually similar whether we use *N* motion updates with one feature at a time or one update with *N* features concurrently if all features contain similar noise. The difference is the weighting created by the covariance matrix *P*. The values of *P* usually decrease as the number of updates increase. If only a few features contain relatively large noise (*i.e.*, 2D tracking outliers), we can improve estimates by using *N* features at the same time. One drawback of using multiple features is increased execution time. *N* updates using one feature at a time requires *N* x (671 multiplications + 549 additions). However, one update using N features requires ($15N^2$ + 156*N* + 500) multiplications + (14 N^2 + 135*N* + 400) additions.



(a)



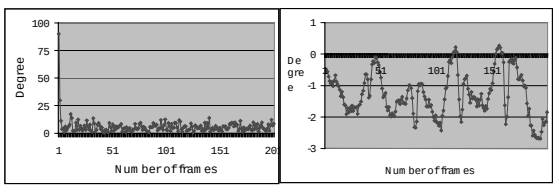
(b) (c)

Figure 3. The motion measured for each feature improves the camera motion estimation. (a) angular error in the translation direction (b) angular error in rotation axes (c) angle difference of rotation angles

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, as illustrated in Figure 2. 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 separately 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. 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 smooth-motion images. This iteration capability is very important since 2D-feature tracking is one of the more computationally expensive processes in motion estimation systems. We can also 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.

Simulation Results

Simulation results show the method is robust and adaptable to varying input data. Since we use a Kalman filter, we need an initial estimate. An initial guess may be far from the correct motion. The initial errors of translation and rotations are 90 or 45 degrees depending on simulations. Rotation estimates are modeled by a rotation axis and a rotation angle about that axis. Translation estimate errors are shown as the angle between correct and estimated translation directions. The rotation errors are shown as the angle between correct and estimated rotation axes and the difference between correct and estimated rotation angles. When we estimate motions for streams of images, we compare between the correct and estimated absolute camera orientation for each image. The absolute orientation estimates are computed by integrating the rotation estimates between all prior images as well as the current image pair. The simulations using streams of



(a) (b)

Figure 4. Motion estimates for general camera motion. (a) angular error in translation directions (b) angle difference of rotation angles.

images show the accumulation errors.

Our simulations are based on features distributed in a 4m square room. For the first simulation, shown in Figure 3, we show how each successive feature contributes to the overall estimate. This simulation also shows the method reusing the same features more than once to improve estimates. 100 features are generated between two consecutive images and used twice to estimate motions. The applied motion is a 2 cm translation and a 3-degree rotation. Gaussian white noise with 0.3-degree standard deviation is added to the projected feature positions.

For the second simulation, we tracked 40 features between consecutive frames. The motion of the simulated camera is a 2cm/frame translation and a 3-degrees maximum sinusoidal rotation about the up-axis. We start the estimation from a zero initial estimate, zero rotation angles, and about 90 degrees of error for each of the translation direction and rotation axis. Gaussian white noise with 0.3-degree standard deviation is added to simulate tracking errors. The results are presented in Figure 4 where we observe that only a few frames are needed to converge to an accurate motion estimate.

The third simulation shows that we can still estimate motions even with only one tracked feature between consecutive frames, under the assumption that the motion is constant over the entire sequence. A single tracked feature is randomly selected for each image pair. The applied motion is a pure translation of 2cm/frame. We added Gaussian white noise with 0.15-degree standard deviation. The initial condition is almost 45-degrees apart from the correct translation direction. Figure 5 shows that the estimate slowly converges to the correct value. Clearly varying motions cannot be estimated with only one tracked feature since it is severely underconstrained. However, even a single tracked feature does provide useful improvement in our approach.

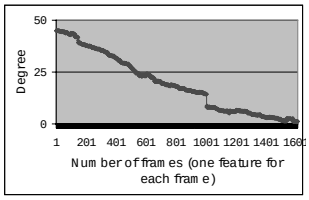
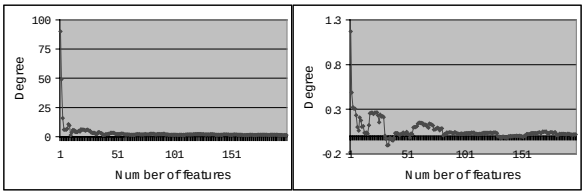


Figure 5. With only one tracked feature, useful motion estimates are still obtained.



(a) (b)

Figure 6. Motion estimation of near pure rotation. (a) angular error in rotation axes and (b) angle difference of rotation angles

The last simulation shows that our algorithm estimates rotations correctly even with very small translations. Some methods estimate translations first and then rotations. These often produce errors for near pure rotations since the translation estimates are error prone. The simulation in Figure 6 shows that this case does not cause problems in our method even though we use the depth independent constraint, which is strongly dependent on the translation direction. The motion for this simulation is a 3-degree rotation about the up-axis of a camera with a very small translation. All of the presented results demonstrate the robust, stable, and flexible behaviors of our motion estimation algorithm.

LARGE MOTION ESTIMATION

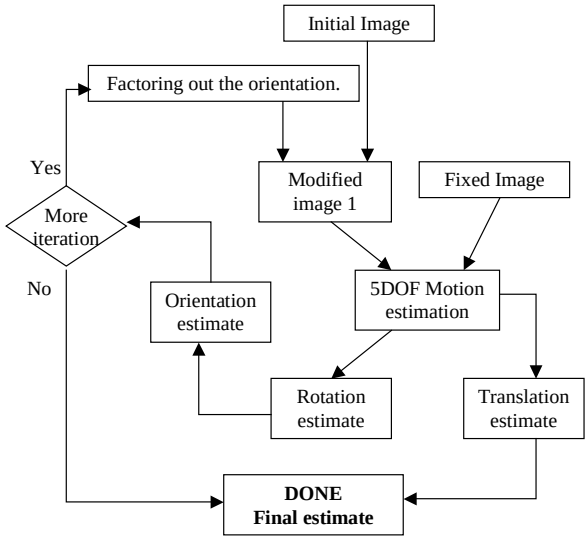


Figure 7. Large motion estimation procedure

Motivation

Motion estimation techniques all suffer from inaccurate translation estimates. As the scene-to-camera distance increases, the translation estimates decrease in accuracy because the motion vectors caused by translations shrink. These translation inaccuracies make motion estimation techniques difficult to use in practical systems. We know that translation estimates are improved as the lengths of motion vectors increase. We can obtain large motion vectors from large translation motions and this is one motivation for investigating effort into large motion estimation.

Without any modifications, we applied our 5DOF motion estimation algorithm to large motions. Relatively pure translation estimates produced accurate results. However, translations mixed with large rotations, e.g. 90

degrees, produced significantly more error. For these more-general large motion cases, the image-feature motions due to translation are short compared to the feature motions due to rotations. Therefore, we developed a new large-motion estimation algorithm based on our IEKF framework and a Recursive Rotation Factorization (RRF).

RRF Large-Motion Algorithm

The RRF algorithm is similar to the IEKF algorithm presented at the previous sections. We add an additional step to the iterations to achieve accurate translation estimates for general mixtures of large translation and rotation motions. As stated previously, large translation motions are estimated accurately in the absence of rotation. General motions containing rotations can produce accurate translation estimates if we remove the feature motions due to rotation. For spherical projections, feature motions due to rotation are easily modeled and subtracted from the feature motion vectors. We model general large motions as TR, and compute the inverse rotation R^{-1} and apply it to the projection prior to estimating the residual (translation) motion $T=TRR^{-1}$. We recursively estimate R^{-1} to remove the rotation motion between two omnidirectional images, leaving only the translation motion to estimate. Figure 7 shows a flowchart of the RRF algorithm for large motion estimation.

The number of iteration required for the RRF procedure depends on the rotation magnitude. Larger angles usually require more iterations. Experiments show that between 5 and 80 iterations may be required to completely remove rotations. (See Figures 9 and 12 in the following section for examples.) We adaptively determine the number of iterations by comparing the convergence rate to a threshold.

As the number of iteration increases the RRF technique requires more execution time. When tested on image streams, the execution time for each estimate can vary greatly, which is not desirable. However, in Figure 12 we show an example that solves this problem by using the previous frame orientation estimate as the initial value of the current frame orientation.

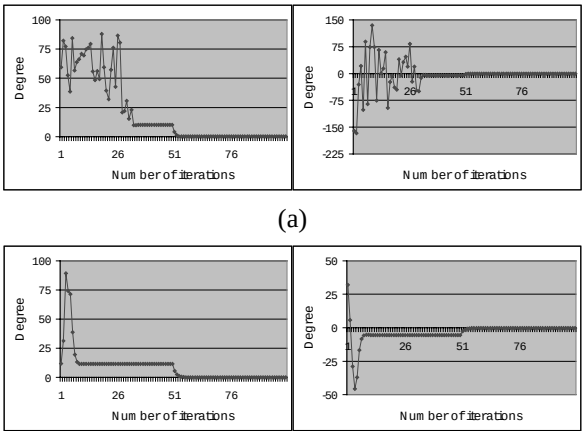


Figure 8. Motion estimation with various translation displacements with 180 degree rotation angle. Charts on the left column describe the angular error in translation directions, and charts on the right column describe the angle difference of rotation angles. (a) 1 m (b) 2 m

Simulation Results

We present five different simulations to show robustness and flexibility of the RRF method over different conditions. Fifty 3D points are generated randomly in the environment, a room of size 4m × 4m × 3m. A subset of features is corresponded in two images for large motion estimation. This number ranges between 15 to 25, depending on the distance between the two camera positions. We add Gaussian white noise with 0.15 to 3.0-degree standard deviations. For most simulations we show results for rotation angles up to 180 degrees to demonstrate the effectiveness of the RRF approach even for extreme conditions.

The first simulation shows two factors presented in earlier sections. We estimate large motions with two different displacements mixed with a fixed rotation of 180-degrees (Figure 8). Gaussian white noise with 0.15-degree standard deviation is added to tracked features. Figure 8 shows that both cases converge to the correct motions. Larger displacements converge faster than smaller displacements. The errors in the initial iterations show how translation estimates are influenced by residual rotations prior to their complete removal.

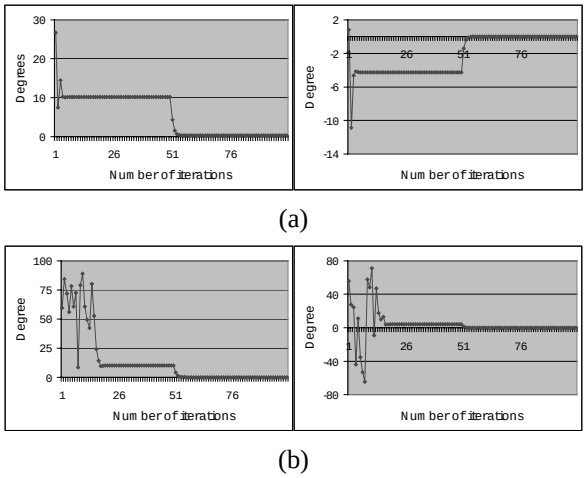
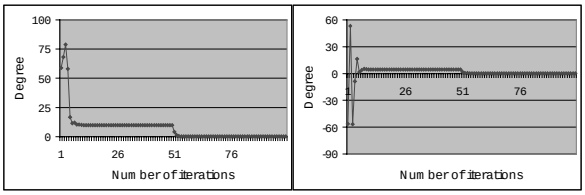


Figure 9. Motion estimates with various rotation angles with the 1 m translation displacement. Charts on the left column describe the angular error in translation directions, and charts on the right column describe the angle difference of rotation angles (a) 45 (b) 90 degrees

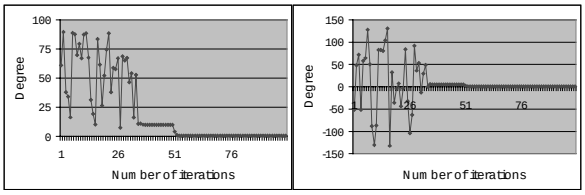
Figure 8 and Figure 9 show how varied rotation angles affect the translation estimates. The largest rotation of 180-degree is shown in Figure 8. Rotation angles in Figure 9 are 45 and 90-degrees. Gaussian white noise with 0.15-degree standard deviation is added to tracked features. The convergence rates are different for each angle, but larger rotation angles do not always converge more slowly than smaller angles.

The third simulation shows estimates with various noise levels. The same motion, a 1-meter displacement with a 180-degree rotation, is applied with two different noise levels set to 0.3-degree and 0.6-degree. Figure 10 shows that accuracy depends on the noise level, but the number of iterations needed to converge is largely unaffected.

Two motion estimation methods are compared in the fourth simulation to demonstrate that the new method has advantages over existing omnidirectional motion estimation methods. Optical flow methods are difficult to apply because of extensive image displacements produced by large motions. Svoboda's 8-point method is suitable for large motion estimations but he did not consider them. Therefore, we implemented an 8-point method based on Svoboda's paper. Both our method and the 8-point method estimate pure translation (1m) and pure rotation (10 degrees along the up-axis) motions accurately. Rotations of large motions (1m displacement and 10 ~ 90 degree rotation along the up-axis) are estimated accurately by both methods. However, our method estimates translation directions more accurately in these cases. The accuracy improvement gets even larger as noise increases as shown in Figure 11. The standard deviations of the 8-point method translation estimates are bigger than our average translation errors.



(a)



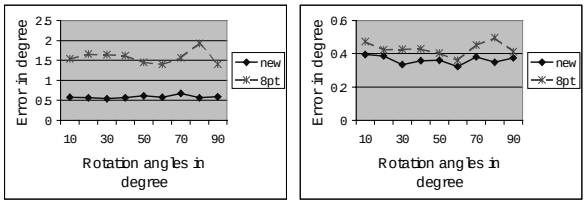
(b)

Figure 10. Motion estimate with various noise levels (1 meter displacement and 180 degree rotation angle). Charts on the left column describe the angular error in translation directions, and charts on the right column represent angle difference of rotation angles. (a) 0.3 degree (b) 0.6 degree noise

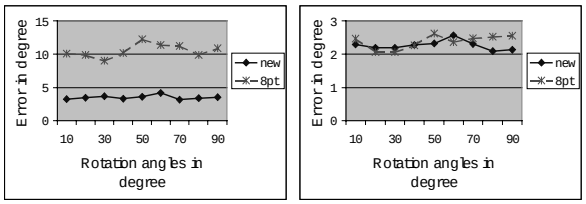
The last example is different from previous cases. This simulation estimates motions over a sequence of images. The camera motion is shown in Figure 12 (d). *A* indicates the start position of the moving camera, and *B* indicates a fixed image from which the large motion is estimated. The camera moves along the path shown in Figure 12 (d), and various rotations are applied along the path, including a sinusoid rotation of 5-degree maximum about the up-axis. In this case, the prior rotation estimate is used to initialize the current estimation. This significantly reduces the number of iterations required for convergence. We also ran this simulation without using previous rotation estimates, and similar estimates were obtained. The main difference is the number of iterations required. By using prior rotations, only five or fewer iterations are needed. Without using prior rotations, 10 to 80 iterations are needed (Figure 12c).

CONCLUSION

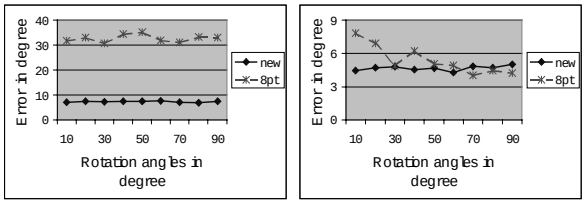
Research in motion estimation mainly targets small (smooth) motions because of the practical limitations of planar-projection imagery. Recent developments in omnidirectional imaging systems provide practical means



(a)

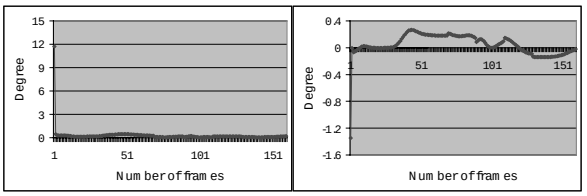


(b)

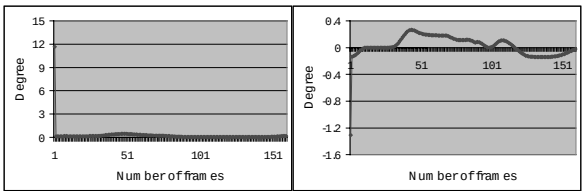


(c)

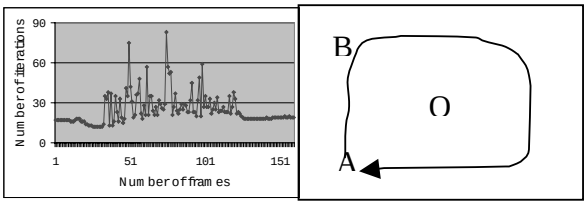
Figure 11. Motion estimation with various noise levels (1 m displacement and 10 ~ 90 degree rotation along the up-axis). Left and right side charts describe translation and rotation error respectively. (a) 0.3 degree (b) 1.5 degree (c) 3.0 degree noise



(a)



(b)



(c) (d)

Figure 12. Motion estimation over the streams of images. Left column of (a) and (b) describe the angular error in translation directions and right column describe the angle difference of rotation angles. (a) Use prior estimation (b) Adaptive motion estimation (c) The number of iteration used for an adaptive motion estimation for each frame (d) The motion path. A and B are two starting positions and A moves along the path. A and B are located on (-1m, -1m, 0) and (-1m, 1m, 0) respectively. O is the center of the space, (0, 0, 0).

of viewing an entire environment and therefore tracking features between widely spaced cameras. This drives the need for estimating large camera motions. Svoboda considered motions with large displacement and no rotation in his thesis. In this paper, we consider general large motions composed of large displacements and rotations. The simulation results show that our large motion estimation method generates accurate translation and rotation estimates for a wide range of conditions.

REFERENCE

[1] A. Azarbayejani, B. Horowitz, and A. Pentland, "Recursive Estimation of Structure and Motion using Relative Orientation Constraints." IEEE Conf. on Computer Vision and Pattern Recognition, Los Alamitos, CA, June 1993, pp. 294-299.

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

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

[4] E. D. Dickmanns and V. Graefe, "Dynamic monocular machine vision," Machine Vision and Applications, 1, 1988, pp. 223-240.

[5] D. B. Gennery, "Tracking known 3-dimensional object," Proc. of AAAI 2nd National Conference Artificial Intelligence, Pittsburg, 1982, pp. 13-17.

[6] J. M. Gluckman and S. K. Nayar, "Egomotion with omnicameras," in Proc. of IEEE International Conference on Computer Vision, ICCV 98, Bombay, India.

[7] R. I. Hartley, “In defense of the 8-point algorithm,” in 5th Int. Conf. on Computer Vision, MIT Cambridge Massachusetts. 1995, pp. 1064-1070.

[8] T. S. Huang and A. N. Netravali, “Motion and structure from feature correspondences: a review,” Proc. of the IEEE, vol. 82, No. 2, Feb. 1994, pp. 251-268.

[9] S. Soatto and P. Perona, "Recursive 3-D motion estimation using subspace constraints," Int. journal of Computer Vision, 1997.

[10] T. Svoboda, T. Pajdla and Vaclav Hlavac, "Motion Estimation Using Central Panoramic Cameras," in IEEE Conf. on Intelligent Vehicles, Stuttgart, Germany, October 1998.

[11] T. Svoboda, "Central Panoramic Cameras Design, Geometry, Egomotion," Ph.D. Thesis, Czech Technical University, ftp://cmp.felk.cvut.cz/pub/cmp/articles/svoboda/phdthesis.ps.gz.

[12] R. Y. Tsai and T. 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, Jan. 1984, pp. 13-27.

[13] J. Weng, T. S. Huang and N. 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, May 1989, pp. 451-476.

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

[15] Greg Welch and Gary Bishop, "SCAAT: Incremental Tracking with Incomplete Information," SIGGRAPH 97, August 1997.