2008 IEEE International Conference on
Robotics and Automation
Pasadena, CA, USA, May 19-23, 2008

Robust and Real-time Egomotion Estimation using a Compound Omnidirectional Sensor

Trung Ngo Thanh, Hajime Nagahara, Ryusuke Sagawa,
Yasuhiro Mukaigawa, Masahiko Yachida, Yasushi Yagi

Abstract—**We propose a new egomotion estimation algorithm for a compound omnidirectional camera. Image features are detected by a conventional feature detector and then quickly classified into near and far features by checking infinity on the omnidirectional image of the compound omnidirectional sensor. Egomotion estimation is performed in two steps: first, rotation is recovered using far features; then translation is estimated from near features using the estimated rotation. RANSAC is used for estimations of both rotation and translation. Experiments in various environments show that our approach is robust and provides good accuracy in real-time for large motions.**

I. INTRODUCTION

Egomotion, which consists of rotation and translation, has been an attractive research topic in computer vision. The egomotion of a camera is recovered by watching the motion on images in a recorded video. Recently, a number of egomotion algorithms have focused on omnidirectional vision due to its large field of view that results in the ability to capture a large range of camera motion. In this research we propose an egomotion estimation algorithm to work with a multi-baseline stereo omnidirectional sensor, that is, a compound omnidirectional vision sensor.

In this research, we use a stereo omnidirectional vision sensor using parabolic mirrors that is similar to the design in Sagawa et al [1] with spherical mirrors. The advantage of this type of sensor is its simplicity; stereo information is provided by a single captured image.

In the computer vision literature, most egomotion algorithms are proposed to work with conventional vision sensors. However, Joshua Gluckman and Shree K.Nayar [2] show that existing egomotion algorithms that rely on the computation of optical flows can work with omnidirectional vision by using a Jacobian to transform motion from a plane to a sphere. Further methods are also proposed to work directly with omnidirectional vision [3,4].

The majority of egomotion estimation algorithms assume correspondence of features between consecutive images [5,6], motion is then recovered after the essential matrix or fundamental matrix is computed. Some other research works rely on the computation of optical flows, or dense correspondences between image frames [7-9]. However, the performance of these types of egomotion estimation depends on the performance of the correspondence computation, which is a non-trivial problem. Moreover, feature tracking or optical flow computation restricts the motion estimation ability of algorithms which means the motion of the camera is limited and fast motion can not be estimated. Another class of egomotion algorithms tries to combine correspondence estimation and motion parameters, so that correspondence and egomotion are simultaneously estimated [10-15]. These methods minimize least-square brightness residuals with respect to motion parameters. One problem for these solutions is that the entire image is used regardless of occlusion, moving objects and so on, which results in errors in estimation and requires a great deal of computation.

In this research, we propose an egomotion estimation solution capable of working with a large range of camera motion, where tracking of feature points is not helpful. Motion and correspondence are estimated simultaneously using a robust estimator: RANSAC. Without correspondence, the computational cost is very high, however our algorithm classifies image features into near and far for different targets, which reduces the computational complexity significantly. Rotation is estimated using distant features, then, translation is estimated using only near features. This separation of motion estimation can be made because: the motion of distant objects is mostly rotation while camera translation is clearly observed by the motion of near objects. Image frames are captured and features are quickly classified using a compound stereo omnidirectional sensor.

The rest of this paper is organized as follows. Section II provides an overview of the compound sensor and feature classification. Section III is an overview of our proposed algorithm. Section IV describes rotation and translation estimations with RANSAC, and their optimization. Finally, an evaluation of our experiments is given in Section V.

II. COMPOUND OMNIDIRECTIONAL VISION SENSOR AND FEATURE CLASSIFICATION

Fig.1 depicts the compound omnidirectional sensor. The sensor has seven conventional parabolic mirrors, six small ones placed around a larger one, and an orthographic camera. For a mirror i, the projection $P_i'(u_{P'}, v_{P'})$ of a space point $P(x_P, y_P, z_P)$ onto the image plane is:

$$\begin{pmatrix} u_{P_i'} \\ v_{P_i'} \end{pmatrix} = \begin{pmatrix} c_x^i + \dfrac{2f_i x_P}{-z_P + \sqrt{x_P^2 + y_P^2 + z_P^2}} \\ c_y^i + \dfrac{2f_i y_P}{-z_P + \sqrt{x_P^2 + y_P^2 + z_P^2}} \end{pmatrix}, \qquad (1)$$

where f_i is the focal length of the parabolic mirror i, (in pixel units); (c_x^i, c_y^i) are the pixel coordinates of the center of the

mirror i on the image plane and the camera coordinate system originates at the optical center of mirror 0 with Oz moving toward the image plane.

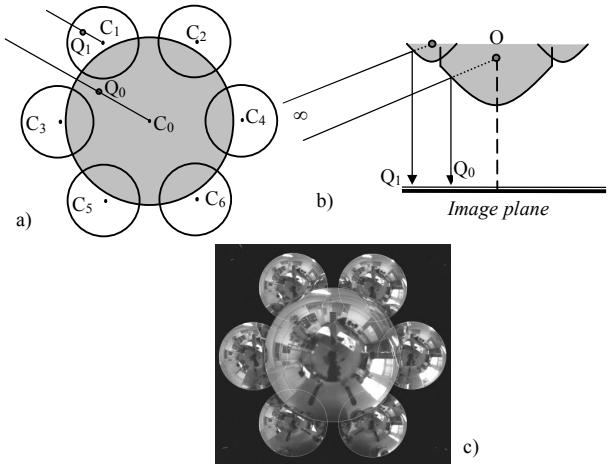

Fig.1. Top view (a), side view (b) of the mirrors and the omni-directional image from the compound sensor (c).

Here we apply a similar algorithm to one used in Sagawa's work [1] to classify near and far features. Each pixel in an omnidirectional image from a mirror (one of seven mirrors) relates to a ray of light from infinity, the projection of rays from the same direction in all other omnidirectional images from the corresponding mirrors produces the corresponding points, Fig.1. If P is from infinity the corresponding points Q_0 and Q_1 of infinite P are related by the equation:

$$c_i \overrightarrow{C_i Q_i} = c_0 \overrightarrow{C_0 Q_0}, \qquad (2)$$

where C_i is the location (in pixel coordinates) of the center of each omnidirectional image for each mirror i, C_0 is the location (in pixel coordinates) of the center omnidirectional image, c_i is the curvature of mirror i, c_0 is the curvature of center mirror and $c_i = 1/2f_i$. Theoretically, if an object is at infinity, all the corresponding points have the same intensity. Therefore, when checking if a pixel is from an object at infinity in an omnidirectional image, we consider the average difference of intensity of all its corresponding points. If the average intensity difference is small, less than a given threshold, this pixel can be determined to come from an object at infinity. Otherwise, the pixel comes from an object closer to the sensor. The criterion to decide the range of each pixel in the center omnidirectional image from the center mirror is:

$$E(Q) = \frac{1}{N_Q} \sum_i | I(Q) - I(Q_i) |, \qquad (3)$$

where N_Q is the number of corresponding points for Q and the maximum for N_Q is 6 here.

Since the above criterion uses intensity difference, the detection is only reliable for pixels that have a large gradient to their neighbors such as at the edge of objects. However, feature detectors such as Harris or Kanade-Lucas-Tomasi can also only detect features like corners in the sensor image.

III. Overview of the Proposed Algorithm

Our algorithm estimates the 5D egomotion of a camera, 3D rotation and 2D translation (direction of translation without magnitude). We use a compound omnidirectional sensor that makes the classification of image features easy and quick. Distant features are processed by the rotation estimator. Estimated rotation is then used to cancel the rotation of near features on both views before they are passed into the translation estimator to obtain the translation. Finally, rotation and translation parameters are optimized using all supporters from RANSAC estimation. A flowchart of the algorithm is presented in Fig.2.

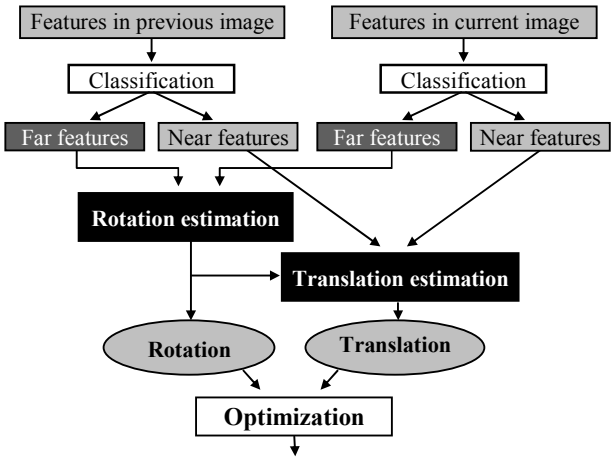

Fig.2. Algorithm flowchart

Since we are dealing with large camera motion, tracking image features is not helpful. We assume a problem without correspondence between consecutive video frames. Rotation and translation are estimated by matching features (distant features for rotation and near features for translation) in a pair of video frames using RANSAC. The RANSAC sampling model for both rotation and translation estimation consists of only two random features on the first frame and another two on the second frame. So that, both rotation and translation estimation are performed uniformly. Rotation estimation alone is referred to in detail in [16].

IV. Estimation of Rotation and Translation with RANSAC

Our method estimates rotation and translation separately. Rotation is estimated first then translation is estimated. A rotation matrix is computed from the motion of two far features; a translation vector is computed from the motion of two near features after their rotation is cancelled. Since we don't assume to know the correspondence of the features, the motion of features between consecutive video frames is assigned randomly using the RANSAC algorithm; the best rotation matrix and translation vector are voted for using the supporters. Rotation parameter estimation is summarized briefly in this section, while most of the section describes

translation estimation using near features.

The camera coordinate system originates at the optical center with Oz as the symmetric axis of the camera system. Camera motion is estimated between two frames with the world coordinate system coinciding with the camera coordinate system at the first frame.

In the following sections, we first describe the motion of image features assuming known correspondence and compute motion parameters (rotation and translation). Then we use these motion parameter computations in RANSAC.

A. Motion computation with known correspondence

The motion of a space point P in the camera coordinate system is described by

$$\mathrm{P}' = R\,\mathrm{P} + \mathrm{T}, \tag{4}$$

where R is the rotation matrix and T is the translation vector of the camera and P and P' are coordinate space point P before and after the motion. The proposed algorithm separates the estimation of rotation and translation using the classified features.

1) Rotation computation

The motion of far features is assumed only by rotation, more details are described in [16]. In the rotation problem, the center of the compound mirror is assumed to remain still. We, thus, need to track the motion of two points, with an additional point known to be the center of the compound mirror, to compute the rotational motion.

Considering a rigid rotation R of two space points P and Q around the optical center O, the cross-product vector $\vec{\mathrm{n}}$ of $\overrightarrow{\mathrm{OP}}, \overrightarrow{\mathrm{OQ}}$ makes the same rotation R. R is computed:

$$R = \left[\mathrm{P}'_{\mathrm{m}}\ \ \mathrm{Q}'_{\mathrm{m}}\ \mathrm{n}'_{\mathrm{m}}\right]\left[\mathrm{P}_{\mathrm{m}}\ \ \mathrm{Q}_{\mathrm{m}}\ \ \mathrm{n}_{\mathrm{m}}\right]^{-1}, \tag{5}$$

where $\mathrm{n}_{\mathrm{m}} = \mathrm{P}_{\mathrm{m}} \times \mathrm{Q}_{\mathrm{m}}, \mathrm{n}'_{\mathrm{m}} = \mathrm{P}'_{\mathrm{m}} \times \mathrm{Q}'_{\mathrm{m}}$ and column vectors $\mathrm{P}_{\mathrm{m}}, \mathrm{Q}_{\mathrm{m}}, \mathrm{P}'_{\mathrm{m}}, \mathrm{Q}'_{\mathrm{m}}$ are projections of P and Q respectively on the unit sphere before and after the motion in the camera coordinate system.

2) Translation computation

As stated above, after rotation is estimated, the rotation of near features is cancelled. Therefore, in this section we assume no rotation occurs between a pair of views. A translation vector can also be estimated from the motion of two near image feature points.

Consider the case where the camera moves on the world coordinate system while watching two space points P and Q. At the moment of the current video frame, the camera is located at O' and the projections of P and Q are $\mathrm{P}'_{\mathrm{m}}, \mathrm{Q}'_{\mathrm{m}}$. At the moment of the previous video frame, the camera is located at O and observes P and Q through the projection points $\mathrm{P}_{\mathrm{m}}, \mathrm{Q}_{\mathrm{m}}$. A group of three points $\mathrm{O}, \mathrm{P}, \mathrm{O}'$ makes an epipolar plane $\Pi(\mathrm{O}, \mathrm{P}, \mathrm{O}')$, as the group $\mathrm{O}, \mathrm{Q}, \mathrm{O}'$ does with the epipolar plane $\Pi(\mathrm{O}, \mathrm{Q}, \mathrm{O}')$. $\Pi(\mathrm{O}, \mathrm{P}, \mathrm{O}')$ can be represented by $\Pi(\mathrm{O}, \mathrm{P}_{\mathrm{m}}, \mathrm{P}'_{\mathrm{m}}, \mathrm{O}')$ and $\Pi(\mathrm{O}, \mathrm{Q}, \mathrm{O}')$ can be represented by $\Pi(\mathrm{O}, \mathrm{Q}_{\mathrm{m}}, \mathrm{Q}'_{\mathrm{m}}, \mathrm{O}')$. Since O, O' are common points for all epipolar planes, we can calculate the direction of $\overrightarrow{\mathrm{OO}'}$ from the intersection of the two epipolar planes $\Pi(\mathrm{O}, \mathrm{P}_{\mathrm{m}}, \mathrm{P}'_{\mathrm{m}}, \mathrm{O}')$ and $\Pi(\mathrm{O}, \mathrm{Q}_{\mathrm{m}}, \mathrm{Q}'_{\mathrm{m}}, \mathrm{O}')$, Fig.3. Let n_1, n_2 be the normal vectors of $\Pi(\mathrm{O}, \mathrm{P}_{\mathrm{m}}, \mathrm{P}'_{\mathrm{m}}, \mathrm{O}')$ and $\Pi(\mathrm{O}, \mathrm{Q}_{\mathrm{m}}, \mathrm{Q}'_{\mathrm{m}}, \mathrm{O}')$ respectively. Then the orientation of the intersection line OO' is given as:

$$\mathrm{T} = \mathrm{n}_1 \times \mathrm{n}_2 \quad \text{where } \mathrm{n}_1 = \mathrm{P}_{\mathrm{m}} \times \mathrm{P}'_{\mathrm{m}},\ \mathrm{n}_2 = \mathrm{Q}_{\mathrm{m}} \times \mathrm{Q}'_{\mathrm{m}} \tag{6}$$

or

$$\mathrm{T} = (\mathrm{P}_{\mathrm{m}} \times \mathrm{P}'_{\mathrm{m}}) \times (\mathrm{Q}_{\mathrm{m}} \times \mathrm{Q}'_{\mathrm{m}}) \tag{7}$$

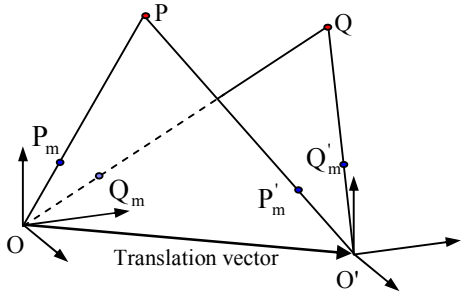

Fig.3. O, O' are located on the intersection of two epipolar planes

Since the motion of the space point on the image sensor is created by the translation of the camera, the projection of the translation vector and the motion vector of the feature on the image plane must be opposite. We use this criterion to adjust the absolute direction of the translation vector.

B. Using RANSAC to estimate rotation and translation

The RANSAC algorithms implemented for both rotation and translation estimation are quite similar. For both algorithms, a random sample is made of two features on the first frame and two more on the following frame to calculate motion. RANSAC simultaneously finds the motion parameters and correspondence of image features. The different is that for rotation estimation, near features which do not display the pure rotation are filtered out by our stereo compound sensor. Far features, however, cannot feasibly estimate translation and should be excluded from the translation estimation. We use near feature points only for this task.

The RANSAC estimation of both rotation and translation is summarized as follows:

a) Randomly select two image features (far for rotation and near for translation estimation) from the previous video frame.

b) Randomly select two image features (far for rotation and near for translation estimation) from the current video frame to assign two pairs of correspondences. These two features are within the vicinity of the two

previously selected features from the previous frame.

c) Calculate the motion parameters (rotation matrix R_t, or translation T_t),

d) Count the supporting pairs of correspondence that match the above motion parameters,

e) Record the current best solution with the maximum number of supporting pairs,

f) If not stopped then return to a).

For translation estimation, the translation vector T_t is computed as shown in Section IV.A.2. To evaluate the translation direction of a trial sample, the number of supporting pairs is counted. Once the direction T_t of OO' is calculated, the epipolar plane $\Pi_1^t(O, P_m, O')$ for each feature P_m on the unit sphere from the first view is given. If one feature P_m' on the second frame is close enough to $C_2^t(P_m)$ we obtain a corresponding pair (P_m, P_m') as a supporting pair for the translation direction. We count all these pairs to evaluate the trial translation. In practice, we apply more constraints to reduce ambiguity; for example, we limit the maximum translation of the camera for consecutive views.

We can assign the size of the vicinity by setting the size of the rotation and translation for any pair of video frames which we want to cope with. The stop criterion for RANSAC sampling is processing time.

C. Optimization

After estimation by RANSAC, the rotation and translation parameters are roughly given. The rotation matrix given by RANSAC may not meet some conditions of a rotation matrix like the orthogonality condition and its determinant being +1.

Optimization is performed as follows:

a) Extract rough rotation parameters (θ_0,ϕ_0,ψ_0), and combine them with the translation parameters (t_{x0},t_{y0},t_{z0}),

b) Make a list of feature correspondence by combining supporting pairs of rotation and translation estimation by RANSAC,

c) Optimize the motion parameters by Levenberg Marquardt optimization to tune the motion parameters with the initial values in step a):

$$(\theta_{opt}, \varphi_{opt}, \psi_{opt}, t_{xopt}, t_{yopt}, t_{zopt}) = \underset{\theta, \varphi, \psi, tx, ty, tz}{\operatorname{argmin}} \sum \left[P_m' E(\theta, \varphi, \psi, t_x, t_y, t_z) P_m\right]^2$$

where $E(\theta, \varphi, \psi, t_x, t_y, t_z)$ is an essential matrix built from the motion parameters and (P_m, P_m') are a correspondence pair.

V. Experiments

In our experiments, the compound sensor was mounted on a system of two rotary stages and a 50cm translation stage (Fig.4). One rotation stage measured the rotation ω_{phi} on the z-axis, the other measured rotation ω_{theta} on the y-axis, while the translation stage measured the one dimensional translation of the system. The vision sensor was a 1600x1200 pixel CCD camera (Scorpion: Point Grey Research) with a telecentric lens. In the experiments, the maximum distance of near features was about 3m for our sensor. The algorithm was processed offline on a PC with a Pentium D 3.2GHz processor. OpenCV was used with image processing and Harris feature detection.

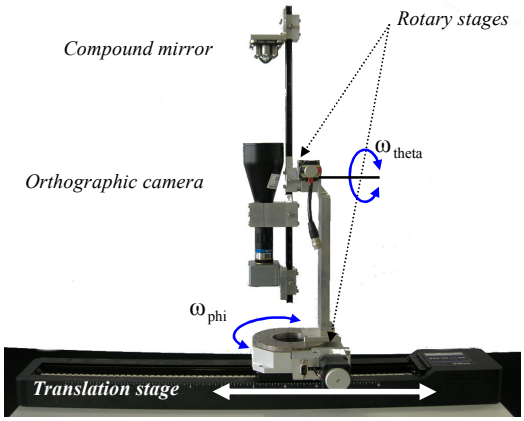

Fig.4. The evaluation system.
Rotary stages and the vision sensor are mounted on the translation stage.

Experiments were carried out in various environments to evaluate accuracy with respect to processing time and the motion of the camera. Our experimental results were compared with the results from the essential matrix-based solution. We implemented the seven-point algorithm based on work by Torr [17] to estimate the essential matrix using RANSAC and the Kanade-Lucas- Tomasi hierarchical feature tracker (OpenCV implementation); we denote this as 7ALGRANSAC. While it is possible to implement the seven-point algorithm using RANSAC without knowing the correspondence, it is very time-consuming to sample a set of 14 features on both frames (7 features on each). Detailed results of these experiments are described below and show the averages of frame-by-frame estimation error and the 100 trials for each video sequence.

A. Frame-by-frame estimation error definition

To evaluate the rotation error we first compute the residual rotation after canceling the estimated motion $\hat{R}$ with the true motion R_{tr} from the rotary stage control:

$$E = \hat{R}.R_{tr}^{-1}. \quad (8)$$

This is the error of the estimated rotation and is represented by a matrix. If the estimation is perfect, matrix E is the identity rotation matrix. The difference between E and the identity rotation matrix I is assumed to be the error of estimation. The Frobenius norm of the matrix (E-I) is one way to evaluate the difference:

$$\text{Angle error} = \sqrt{\sum_{i=1,j=1}^{3,3}(E_{ij} - I_{ij})^2} \quad (9)$$

The translation error is the Euclidean difference between the normalized estimated translation vector and the normalized ground-truth translation vector.

B. *Experiment results*

The experiments were carried out along a balcony in our building. We extracted 140 features for each frame using a Harris feature detector (OpenCV implementation). The experiment showed that for each frame the Harris feature detector needed 0.066 sec to extract 140 features. However, the Kanade-Lucas-Tomasi feature tracker needed only 0.0078 sec to track 140 features so our algorithm had less time to carry out the RANSAC estimation than 7ALGRANSAC.

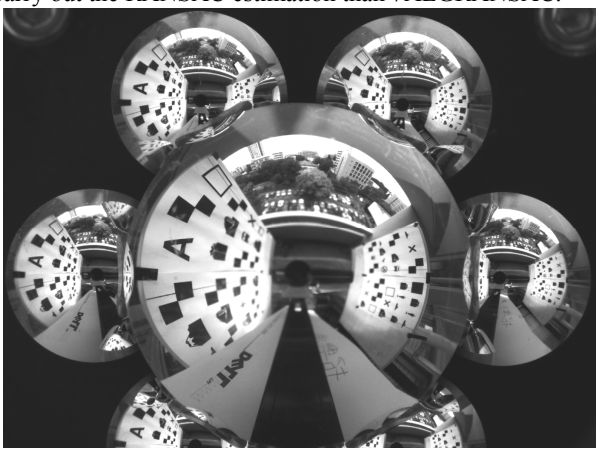

Fig.5. Omnidirectional image of the experiment environment.

We also set up the parameters so that our algorithm could cope with a maximum 23 Deg/frame of rotation velocity and 0.085 m/frame of translation velocity. For our algorithm, the processing time includes feature extraction, feature classification, RANSAC motion estimation and Levenberg Marquardt optimization for motion parameters. While the processing time for 7ALGRANSAC includes initial feature detection, frame by frame feature tracking, RANSAC estimation of the essential matrix, motion parameter extraction and optimization using Levenberg Marquardt.

In the shown experimental environment, shown in Fig.5, the near feature was actually about 0.5m away from the camera.

1) *Experiments with processing time*

Experiments were carried out for both algorithms with various rotation velocities. Fig.6 shows the errors with respect to processing time for both algorithms. We can see that, the convergence of 7ALGRANSAC starts earlier than our algorithm since the preprocessing time (feature tracking) of 7ALGRANSAC is less than the preprocessing time (feature detection) for our proposed algorithm (0.0078 sec compared to 0.066 sec). Both algorithms converged to a reasonable accuracy after running for 0.1 sec, when RANSAC has run for 0.034 sec in our proposed method and 0.0922 sec in 7ALGRANSAC. Further, when the rotation velocity was increased, the feature tracker became less accurate; 7ALGRANSAC then slowly converged and was less accurate. Meanwhile, our algorithm gave a stable convergence for all rotation velocities within a predefined maximum rotation.

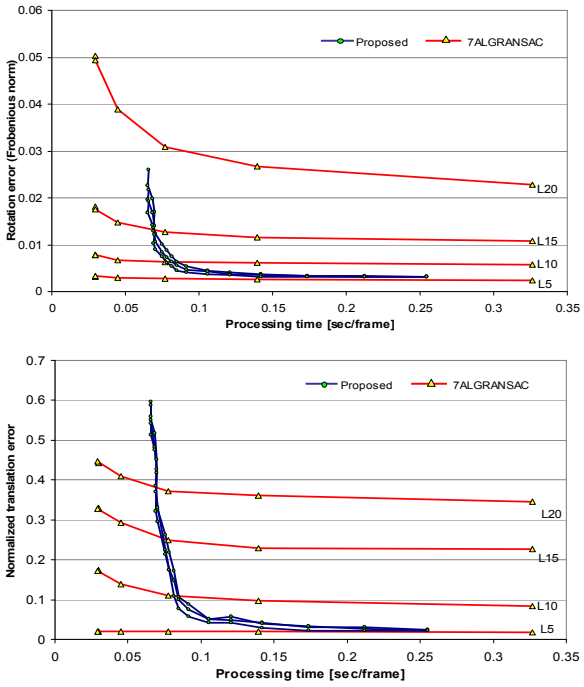

Fig.6. Errors of our proposed algorithm and 7ALGRANSAC with various rotation velocities (5(L5), 10(L10), 15(L15), 20(L20) Deg/frame) and the same translation velocity 10 cm/frame.

2) *Experiments with rotation and translation velocities*

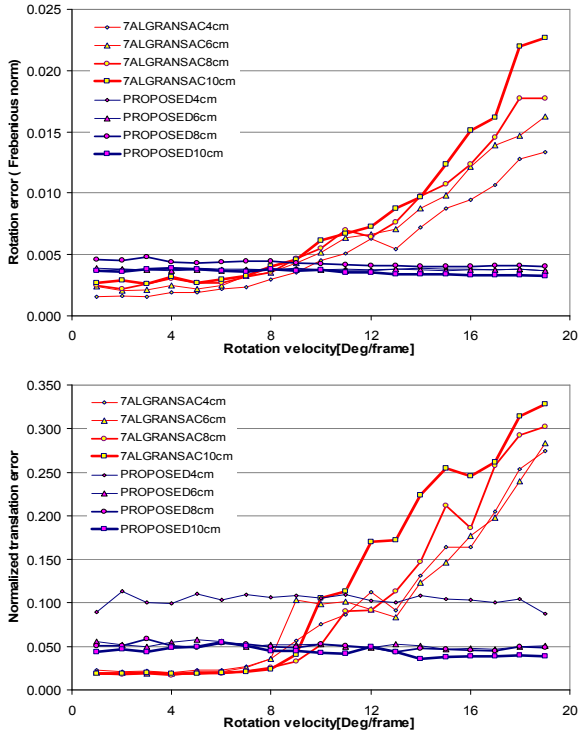

Fig.7. Errors for our proposed algorithm and 7ALGRANSAC under various rotations and four translations 4,6,8,10[cm/frame].

Experiments were carried out to show the robustness of our algorithm with motion as shown in Fig.7 (which has the same legends as the previous graphs. The graphs show the estimation error (rotation and translation) with various rotation velocities from 1 to 20 Deg/frame and translation velocities: 4,6,8,10 cm/frame. In these experiments, the processing time for both algorithms is 0.1 sec/frame. Experiments showed that for the proposed method the accuracy does not depend on rotation velocities even though they are large. Accuracy depends rather on translation velocities; the larger translation the more accurate. For 7ALGRANSAC, the results become less accurate when the rotation velocities increased and the feature tracker becomes less accurate. For small motion (small rotation and translation velocity) 7ALGRANSAC performed better than our proposed method. However, for large motion, our proposed algorithm worked better.

VI. Discussion and Conclusion

In this paper, we propose an approach to egomotion estimation using RANSAC and an omnidirectional compound sensor. Using the compound sensor, image features are separated into near and far features. Far features are used for rotation estimation while near features are used for translation estimation. Pure rotation is not expressed with the near features and far features are ineffective to describe translation.

Rotation is estimated using only far features which keeps the estimation simple and works without the computation of the correspondences. Consequently, larger camera motion can be estimated free from the small motion assumption of correspondence computation. Translation is then also simple after rotation is estimated.

Rotation and translation estimation are, however, possible without feature classification, and with the same algorithms, though classification obviously helps the rotation estimation ignore a large number of outliers of pure rotation and helps translation estimation ignore a large number of far features that give weak support for translation estimation. Therefore, both rotation and translation estimation work more efficiently with feature classification. Moreover, the classification of near and far features requires insignificant computational time with our compound omnidirectional sensor.

Our algorithm assumes the range of far features is much larger than the actual translation of the camera, and therefore requires a sufficiently large environment. Our algorithm does not perform quite as well as small egomotion estimation algorithms such as the seven-point algorithm using feature tracking because we accept an approximation of far features as features at infinity. However, the accuracy of our algorithm is acceptable and stable for a large range of motion velocity, while the seven-point algorithm performs worse or can not work at all in these situations due to unreliable correspondence computations.

Since our solution is to estimate large camera motion, it can be applied in practice to the motion of a wearable camera system; the subject of our current research. The motion of wearable cameras can be very large due to the extreme movements of the human body, especially rotation. We believe that egomotion observed by a low frame-rate camera system can be as readily estimated by our algorithm as large motion and progress with demonstrating this to be true.

Acknowledgement

We would like to give great thanks to Yuichiro Kojima for supporting us with the code for near object detection using a compound omnidirectional sensor.

References

[1] R. Sagawa, N. Kurita, T. Echigo, Y. Yagi, "Compound Catadioptric Stereo Sensor for Omnidirectional Object Detection", Proc. IEEE/RSJ IROS, vol.2, pp.2612-2617, Sendai, Japan, Sep., 2004.

[2] J. Gluckman, S.K. Nayar, “Egomotion and omnidirectional cameras”, In: Proceedings of International Conference on Computer Vision, Bombay, pp. 999–1005, 1998.

[3] Y. Yagi, W. Nishi, K. Yamazawa, M. Yachida, “Rolling motion estimation for mobile robot by using omnidirectionalimage sensor HyperOmniVision”, ICPR 1996, p.p 946-950 vol.1

[4] Y. Yagi,W. Nishi,N. Benson, M. Yachida, “Rolling and swaying motion estimation for a mobile robot by using omnidirectional optical flows”, Journal of Machine Vision and Applications, p.p 112-120,Volume 14, No. 2, June, 2003.

[5] H.C.Longuet-Higgins, “A computer algorithm for reconstructing a scene from two projections”, Nature 293, pp. 133-135,1981.

[6] O. D. Faugeras, "What can be seen in three dimensions with an uncalibrated stereo rig?", Proc. European Conference on Computer Vision, LNCS 588, pp.563–578, Springer-Verlag, 1992.

[7] D.J. Heeger and A.D. Jepson, “Subspace methods for recovering rigid motion I: Algorithm and Implementation”, Intl. Journal of Computer Vision Vol.7(2), pp.95-117, 1992.

[8] A.R. Bruss, B.K. Horn, "Passive navigation", Computer Graphics and Image Processing, Vol. 21, pp. 3-20, 1983.

[9] G. Adiv. "Determining three-dimensional motion and structure from optical flow generated by several moving objects". IEEE Trans. PAMI. Vol.7, pp.384-401, 1985.

[10] B. Horn and E. Weldon, “Direct methods for recovering motion,” Int’l J. Computer Vision, pp. 51–76, 1988.

[11] S. Negahdaripour, N. Kolagani, and B. Hayashi, “Direct motion stereo for passive navigation,” in Proc. Conf. Computer Vision and Pattern Recognition, June 1992, pp. 425–431.

[12] R. Kumar, P. Anandan, and K. Hanna, “Direct recovery of shape from multiple views: a parallax based approach,” in Proc. 12th IAPR Int’ Conf. Pattern Recognition, vol. 1, Jerusalem, Israel, 1994, pp. 685–688.

[13] M. Irani, P. Anandan, and M. Cohen, “Direct recovery of planar parallax from multiple frames,” IEEE Trans. Pattern Anal. Machine Intell., vol. 24, no. 11, pp. 1528–1534, 2002.

[14] G. Stein and A. Shashua, “Model based brightness constraints: On direct estimation of structure and motion,” IEEE Trans. Pattern Anal. Machine Intell., vol. 22, pp. 992–1015, Sept. 2000.

[15] A. Agrawal and R. Chellappa, "Robust Egomotion Estimation and 3D Model Refinement using Surface Parallax", IEEE Transactions on Image Processing, Vol. 15, No. 5, May 2006.

[16] T.T. Ngo, H. Nagahara, R. Sagawa, Y. Mukaigawa, M.Yachida, Y. Yagi, “Robust and Real-time Rotation Estimation of Compound Omnidirectional Sensor”, ICRA 2007, Italy, April 2007.

[17] P.H.S. Torr, “Outlier Detection and Motion Segmentation”, PhD dissertation, Dept. of Eng. Science, Univ. of Oxford, 1995.