2007 IEEE International Conference on
Robotics and Automation
Roma, Italy, 10-14 April 2007

Robust and Real-time Rotation Estimation of Compound Omnidirectional Sensor

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

Abstract— **Camera ego-motion consists of translation and rotation, in which rotation can be described simply by distant features. We present a robust rotation estimation using distant features given by our compound omni-directional sensor. Features are detected by a conventional feature detector, and then distant features are identified by checking the infinity on the omni-directional image of the compound sensor. The rotation matrix is estimated between consecutive video frames using RANSAC with only distant features. Experiments with various environments show that our approach is robust and also gives reasonable accuracy in real-time.**

I. Introduction

Ego-motion is an attractive research topic in computer vision. Ego-motion consists of rotation and translation. Most research studies on ego-motion estimate both rotation and translation at once. However, in some applications, such as video stabilization, only the rotation is important. In addition, in practical applications such as security systems, robustness and real-time processing are required.

There are a large number of approaches, such as [1]-[8], for estimating ego-motion in computer vision, and most of them rely on the advantage of features. Due to the motion of features which are detected and tracked on the video, the motion of the camera can be recovered. All features are treated similarly regardless of their distance to the camera. However, it is fair to say that when estimating the translation of a camera, distant features are ineffective. They do not appear to move on the video when the camera translates, although they similarly move when the camera makes a slight rotation with or without translation.

Our research distinguishes the distant features and near features for separated targets. Distant features are used for estimating the rotation of a camera, while near features are used for translation. This classification of features helps the estimation of ego-motion become more robust and simple. In this paper, we present the first part of our research, rotation estimation using distant features.

Obviously, rotation can be estimated by existing general methods for ego-motion, but such solutions are indirect and time-consuming for rotation estimation issues since they also estimate the translation of the camera. In response to the rotation estimation only, direct methods [9]-[12] have also been proposed. Some approaches use unclassified features. Features are usually tracked between consecutive frames. The robustness and accuracy of estimations also depend on the correspondence and dynamicity of the environment. Moreover, some significant computation cost is needed for finding the correspondence of all the features found available on the images. Among these approaches, Stan et al. [9] has presented a method using annealing M-estimator (AM-estimator) which can explicitly work with the translation of cameras. In another approach, a featureless solution, Makadia et al. [10] proposed a method using the transformation of images. In their method, the whole image is transformed into the frequency domain by spherical Fourier Transform. Then a decoupling of the shift theorem with respect to the Euler angles is exploited in an iterative scheme to refine the initial rotation estimates. Since the whole image is considered, this method needs much computation cost. Moreover, their approach gives poor results with translation of camera and is not very robust with a dynamic environment because of significant distortion of the near scenery by the translation of the camera.

Our approach uses features, but only distant features. In the algorithm, all features are detected by a feature detector and then filtered to eliminate the near features by using our compound omni-directional sensor. The tracking of features is not necessary in our algorithm. Then, features are represented on a unit sphere. RANSAC matching between consecutive frames is performed to simultaneously find correspondence of inliers and the rotation matrix. We assume the environment is much larger than the translation of the camera. Experiments showed that it is a robust approach and can work in real-time.

The following section, Section II, provides an overview of the compound sensor and distant feature detection. Section III describes the motion of distant features. Sections II and III therefore support (as fundamentals) Section IV in showing the rotation estimation using RANSAC. Finally, the evaluation of the experiments is given in Section V.

II. Compound Omni-directional Sensor and Distant Feature Detection

Fig. 1 describes the compound sensor, which is a multi-baseline stereo omni-directional vision sensor using seven conventional parabolic mirrors, six small ones at the sides and a big one in the center, and an orthographic camera. The corresponding omni-directional image of each mirror can be used independently as conventional omni-directional images.

The work of Sagawa [13], using sensor with spherical mirrors, which is similar to the sensor we used, shows that this type of sensor has the advantage of quickly detecting near objects. In this research, we take advantage of the same

1-4244-0602-1/07/$20.00 ©2007 IEEE.

algorithm to detect the distant features by their infinity. The infinity of a feature point is done quickly by checking the corresponding points on the image areas of the seven mirrors. If the difference of the corresponding points is larger than a certain threshold, then those image points are considered to come from a near object. Otherwise, the image points belong to an object at infinity. The infinity depends on the baselines of the sensor and also on the resolution of the CCD sensor. The epipolar constraints can be applied to improve the robustness of the detection. Since we use intensity, the detection is done only at the image points that give a large gradient. However, feature detectors such as Harris or Kanade-Lucas-Tomasi also can only detect features at image points with a large gradient.

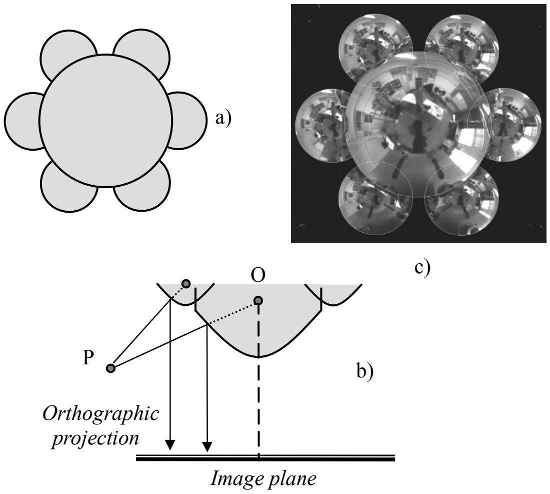

Fig.1. Top view (a), side view (b) of the mirrors and omni-directional image from the compound sensor (c).

III. Motion and motion computation of distant feature points

A. *Motion of distant feature points*

Once having located the coordinate system at the optical center O of the big mirror (Fig.2), the surrounding scenery moves around the sensor. For point P(x_P,y_P,z_P) with rotation R and translation T of the camera in a world coordinate system, P is rotated by R^{-1} and translated by -T:

$$\begin{pmatrix} x'_P \\ y'_P \\ z'_P \end{pmatrix} = R^{-1} \begin{pmatrix} x_P \\ y_P \\ z_P \end{pmatrix} - T \qquad (1)$$

Representing P in a spherical coordinate system originating at the optical center point O of the center mirror (see Fig. 2), (1) is rewritten as

$$\begin{pmatrix} \rho'_P \sin(\theta'_P)\cos(\varphi'_P) \\ \rho'_P \sin(\theta'_P)\sin(\varphi'_P) \\ \rho'_P \cos(\theta'_P) \end{pmatrix} = R^{-1} \begin{pmatrix} \rho_P \sin(\theta_P)\cos(\varphi_P) \\ \rho_P \sin(\theta_P)\sin(\varphi_P) \\ \rho_P \cos(\theta_P) \end{pmatrix} - T \; (2)$$

or

$$\begin{pmatrix} \sin(\theta'_P)\cos(\varphi'_P) \\ \sin(\theta'_P)\sin(\varphi'_P) \\ \cos(\theta'_P) \end{pmatrix} = \frac{\rho_P}{\rho'_P} R^{-1} \begin{pmatrix} \sin(\theta_P)\cos(\varphi_P) \\ \sin(\theta_P)\sin(\varphi_P) \\ \cos(\theta_P) \end{pmatrix} - \frac{T}{\rho'_P} \qquad (3)$$

where $(\theta_P, \varphi_P, \rho_P)$ and $(\theta'_P, \varphi'_P, \rho'_P)$ are spherical coordinates of P before and after the camera motion .

From (3) we can see that if the distance ρ'_P is much larger than T then we can ignore the term $\frac{T}{\rho'_P}$, and that then the motion of this distant point is only rotation. Since we consider only the rotation $\frac{\rho_P}{\rho'_P} \approx 1$, (3) then becomes

$$\begin{pmatrix} \sin(\theta'_P)\cos(\varphi'_P) \\ \sin(\theta'_P)\sin(\varphi'_P) \\ \cos(\theta'_P) \end{pmatrix} \approx R^{-1} \begin{pmatrix} \sin(\theta_P)\cos(\varphi_P) \\ \sin(\theta_P)\sin(\varphi_P) \\ \cos(\theta_P) \end{pmatrix}. \qquad (4)$$

We can see that $(\sin(\theta'_P)\cos(\varphi'_P),\ \sin(\theta'_P)\sin(\varphi'_P),\ \cos(\theta'_P))$ are the Cartesian coordinates of P on the unit sphere.

From (4) we can see that the motion of a distant point, which is approximated by only the rotation, can be understood by the motion of its projection on the unit sphere. Equation (4) also prompts us to represent the feature point P on the unit sphere. A map from the image coordinate system to the unit sphere needs to be made for real-time processing.

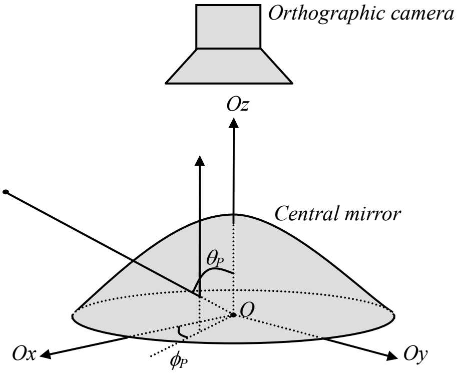

Fig. 2. Camera coordinate system.

B. *Rotation computation from known correspondence*

In general ego-motion, we can compute the rotation and translation of a camera by tracking three feature points. However, in our case the motion of distant features is assumed only by rotation, and therefore the problem is easier to solve. The center of the compound mirror is assumed to remain still. We can thus track the motion of two points, with the additional point known as the center of the compound mirror.

Considering a rigid rotation M of two space points P and Q around O, the cross-product vector $\vec{n}$ of $\overrightarrow{OP},\overrightarrow{OQ}$ makes the same rotation. As shown above, their images P_m, Q_m, n_m on the unit sphere also make the same rotation:

$$\begin{aligned} P'_m &= M.P_m, \\ Q'_m &= M.Q_m, \\ n'_m &= M.n_m, \end{aligned} \tag{5}$$

where $\overrightarrow{n_m} = \overrightarrow{OP_m} \times \overrightarrow{OQ_m}$, $\overrightarrow{n'_m} = \overrightarrow{OP'_m} \times \overrightarrow{OQ'_m}$ and P_m, Q_m, n_m are column vectors. Then we can get the rotation matrix easily from the motion of the feature points on the unit sphere:

$$M = \left[P'_m \;\; Q'_m \;\; n'_m\right]\left[P_m \;\; Q_m \;\; n_m\right]^{-1}. \tag{6}$$

In the estimation algorithm using RANSAC, this computation is used to initialize the rotation model, which needs only four points on two consecutive frames. Two points are on the previous frame and two others are on the current frame.

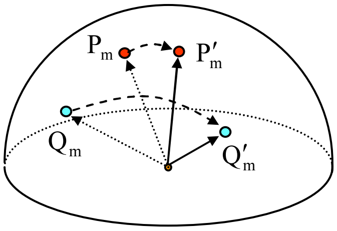

Fig. 3. Estimate rotation from motion of 2 points.

IV. RANSAC TO ESTIMATE THE ROTATION

RANSAC is well-known as a robust estimator in computer vision, which is preferable when the image data is highly noisy. For our case, there are a lot of outliers that do not hold the rotation (4), including near features, moving features, and features contaminated by occlusion. Further, with distant features, three issues must be solved: finding the correspondence of the inliers on the current frame and on the previous frame, removing the outliers for the rotational motion, and estimating the rotation. RANSAC is a reasonable selection for doing these tasks. RANSAC can simultaneously find the correspondence and the rotation of correspondences and remove the outliers of the rotation. Using our compound vision sensor, a large number of outliers (near features) are easily eliminated, resulting in lower computation cost and improved accuracy. In this paper we use the standard RANSAC, but in practical use the RANSAC extensions should be applied for better performance.

A. Algorithm

In this section, the algorithm to estimate the rotation of consecutive frames is briefly described. First, the features are detected on both frames, and the near ones are eliminated. The remaining features are then mapped on the unit sphere. RANSAC is performed to match the two sets of spherical points to estimate the motion on the unit sphere. The motion then shows us the rotation. A *quartet* is defined as a group of four points, two inliers on the previous frame and two supporters on the current frame. If the selected four random points make up a quartet then the initialization is successful for a rotation matrix.

Initialization of rotation matrix M is done as shown in Section III, by assuming the correspondence between two random points on previous frame and two random (within the vicinity of two previous points) points on current frame. Since our approach is real-time RANSAC, the criteria for stopping the search is the processing time.

B. Computational cost of RANSAC

If the probability of inliers on the previous frame is p_{in} then the probability of outliers is 1- p_{in}. Then, the probability of selecting one correct pair (P_m,Q_m) of inliers on previous frame is p_{in}^2. The probability of selecting a supporter on current frame of a correct pair is p_{sup}. This is the probability of selecting the correct correspondences of P_m or Q_m on previous frame. Then, the probability of selecting a quartet (two inliers, two supporters) for the rotation motion is

$$p_{qua} = p_{in}^2 p_{sup}^2 . \tag{7}$$

The minimum required number of iterations is as follows

$$k = \frac{\log(z)}{\log(1 - p_{qua})}, \tag{8}$$

where z is the probability of seeing only bad samples:

$$z = (1 - p_{qua})^k . \tag{9}$$

More detailed information about this number k can be found in the book [14]. In practical implementation, in order to specify the minimum number of iterations k, we have to supply a predefined value of z. In our algorithm, because k is not large this algorithm can be applied in real-time. For instance, if the probability of selecting an inlier on previous frame is 0.4 and if there are on average 10 feature points in the vicinity of one point on the previous frame, meaning the probability of finding a supporter on current frame for an inlier on the previous frame p_{sup} is 1/10 = 0.1, predefined value of z=0.001, then k =4313. Moreover, this number k is reduced when a significant number of outliers are eliminated by using the compound sensor.

C. Theoretic error of the estimation

In this algorithm, we accept an approximation of the infinity. This section will show how large error can come from the approximation. If the translation of the camera per frame is d_t and the distance from the camera to the real feature point is d_f then the maximum error of accepting this as a distant point is $\arcsin(\frac{d_t}{d_f})$. In other words, the error depends on the environment size, the translation, and the direction of translation of the camera. For example, if the camera translates 20[cm] per frame in the direction perpendicular to the direction to the feature, and the distance to the feature point is 500[cm], then the error is about

0.04[Rad]. In our algorithm, the searching of the supporters in the algorithm depends on the threshold *th*. This threshold helps us to obtain the accuracy of the estimation, the smaller the more accurate. However, the value of *th* must be close to the theoretic error $\arcsin(\frac{d_t}{d_f})$, in which d_f is the distance to the nearest acceptable distant feature point.

V. Experiments

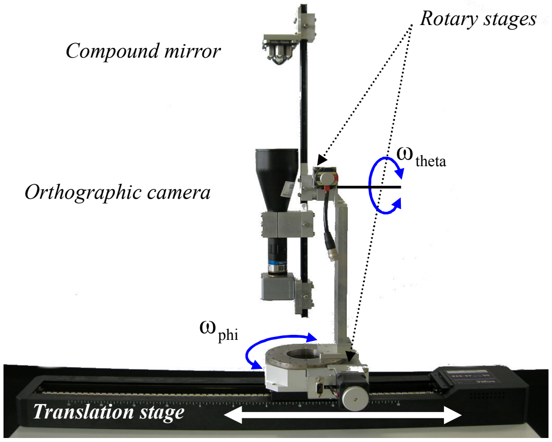

Fig. 4. The evaluation system.
Rotary stages and vision sensor are mounted on the translation stage.

In our experiments, the compound sensor was mounted on a system of two rotary stages and a 50[cm] translation stage (Fig. 4). One rotation measured rotation ω_{phi} on the z axis, the other measured rotation ω_{theta} on the y axis, while the translation state measured the one dimensional translation of the system. Since there was no motor to control rotation ω_{psi} on the x axis, this angular velocity was set to 0 during the experiments we carried out to evaluate the estimation of ω_{psi} with a ground-truth of 0[deg/frame]. The vision sensor was a 1600x1200 [pixel] CCD camera (Scorpion: Point Grey Research) with a telecentric lens (0.16x TML: Edmond Optics). In the experiments, the effective infinity detection of our compound sensor was about 4[m]. All these sensors were connected to a PC, a Pentium D 3.2GHz. On this PC, the algorithm was evaluated. OpenCV helped us with image processing and feature detecting.

Experiments were carried out with various environments to evaluate the accuracy with respect to computation cost translation and the rotation of the camera. The RANSAC was evaluated with a conventional omni-directional vision sensor (without near feature point elimination, here called STDRANSAC) and compound vision sensor (with near feature point elimination, here called PROPOSED) and compared to the ground-truth from the rotary states. The computational cost for each frame of STDRANSAC consists of feature detection and RANSAC computation. Meanwhile, PROPOSED requires some additional computation to eliminate near features.

More detailed results of these experiments are described in the following sections, showing the averages of the frame-by-frame error estimation and the 100 trials for each video sequence.

A. Angular error definition

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

$$E = \hat{R}.R_{tr}^{-1}. \qquad (10)$$

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

$$\text{Angle error} = \sqrt{\sum_{i=1,j=1}^{3,3}(E_{ij} - I_{ij})^2} \qquad (11)$$

B. Indoor environment

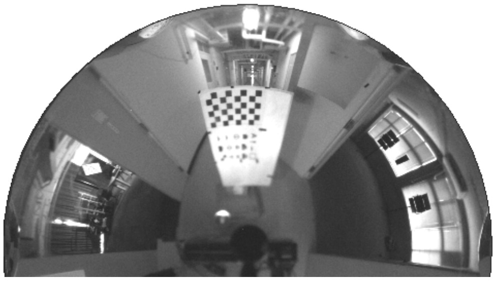

Fig. 6. Indoor scene.

The experiments were carried our along a corridor in our building (Fig. 6). The extracted features ranged from 1[m] to about 8[m]. STDRANSAC and PROPOSED were run with the same conditions in order to compare the results.

1) Experiments with translation of camera

In these experiments, the processing times of STDRANSAC and PROPOSED were similar, at 0.095[sec], and the angular velocities were ω_{theta} =10[deg/frame], ω_{phi} =5[deg/frame] and ω_{psi} =0[deg/frame]. The translation of the camera varied from 5[cm/frame] to 25[cm/frame] in the experiments. The errors of estimation are shown in Fig. 7, in which the pink small boxes depict the errors of STDRANSAC, while the dark blue round dots denote the errors of PROPOSED.
As shown by the results, the translation contaminates the estimation. However, the proposed method is less affected by the translation because the near points, which are changed drastically with the translation, are eliminated. Therefore, the accuracy of the proposed method is better.

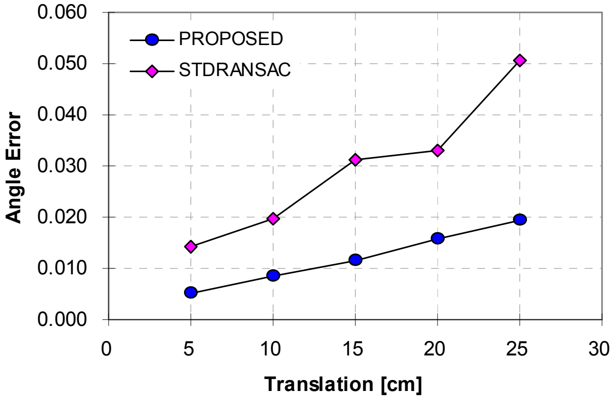

Fig. 7. Indoor environment: estimation errors with different camera translation. Processing time is 0.95 [sec/frame] and angular velocities are ω_{theta} = 10 [deg/frame], ω_{phi} = 5 [deg/frame] and ω_{psi} = 0.

2) *Experiments with angular velocities*

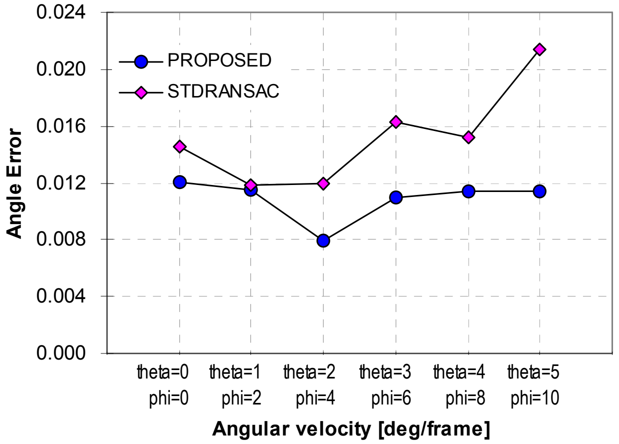

Fig. 8. Indoor environment: estimation errors with different angular velocities. Processing time is 0.1[sec/frame] and translation is 15[cm].

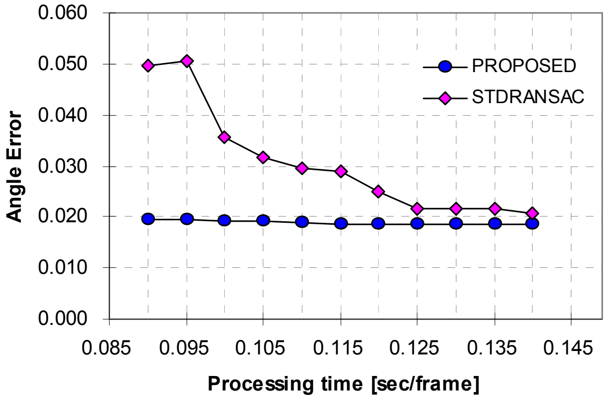

Fig. 9. Indoor environment: estimation errors with different processing time. Translation is 25[cm] and angular velocities are ω_{theta} = 10 [deg/frame], ω_{phi} = 5 [deg/frame] and ω_{psi} = 0.

Experiments were carried out with same processing time (0.1[sec]) and same translation (15[cm]) for both methods. The evaluated errors are shown in Fig. 8. The legends are the same as those in the previous section.

It seems that both methods do not depend much on the angular velocities. However, the proposed method produced more accurate and robust results.

3) *Experiments with processing time*

Experiments were carried to evaluate the accuracy using a different processing time. The camera translation and angular velocities were the same for both methods (25[cm/frame] translation and ω_{theta} = 10 [deg/frame], ω_{phi} = 5 [deg/frame] and ω_{psi} = 0 [deg/frame]). The errors are described in Fig. 9, with the same legends as in the previous sections.

These experiments showed that our method produced good results. Due to the number of outliers removed, however, the processing time was reduced significantly. Nonetheless, with the input images having a resolution of 1600x1200 [pixel], the processing time of 0.1[sec] is reasonable for use in real applications with acceptable accuracy.

C. *Outdoor environment*

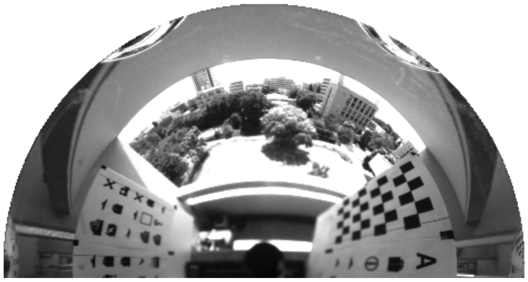

Fig. 10. Outdoor scene.

Experiments were also carried out in an outdoor environment (Fig. 10) to validate our method in large environments. The comparison between STDRANSAC and our method PROPOSED were made using the same criteria as those of the above indoor experiments.

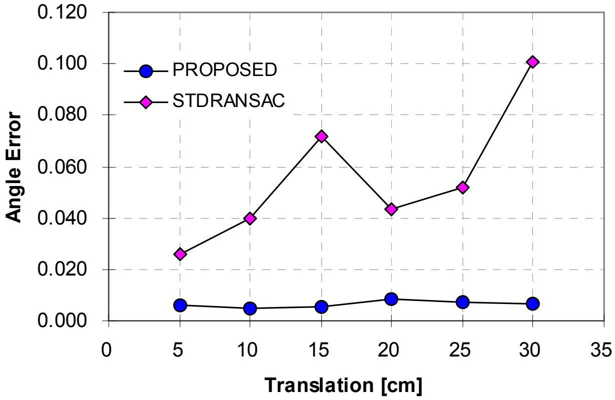

Fig.11. Outdoor environment: estimation errors with different camera translation. Processing time is 0.1 [sec/frame] and angular velocities ω_{theta} = 10 [deg/frame], ω_{phi} = 5 [deg/frame] and ω_{psi} = 0.

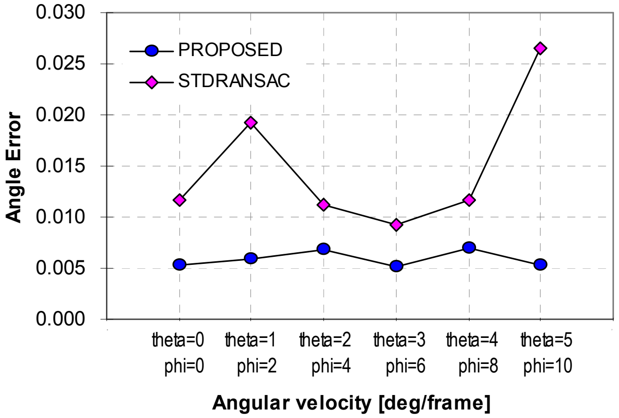

Fig.12. Outdoor environment: estimation errors with different angular velocities. Processing time is 0.1[sec/frame] and translation is 15[cm].

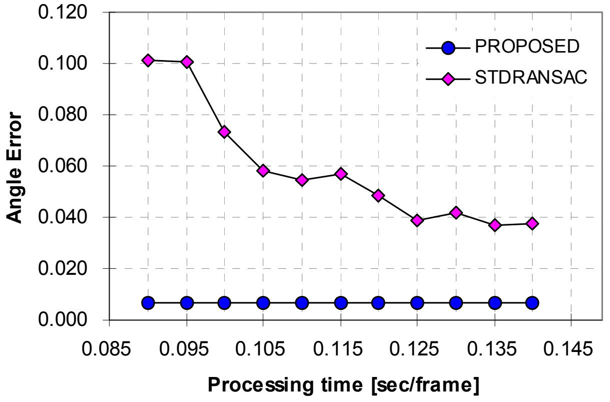

Fig.13. Outdoor environment: estimation errors with different processing time. Translation is 30[cm] and angular velocities are ω_{theta} = 10 [deg/frame], ω_{phi} = 5 [deg/frame] and ω_{psi} = 0.

The details of the comparison are described in Fig. 11, Fig. 12 and Fig. 13. Since the environment was large, the inliers were very far from the camera, and as a result the overall results of the three experimental factors for our proposed method were much better than our experiments in indoor environments. Meanwhile the STDRANSAC shows results similar to those of the indoor experiments, which were worse than those of the proposed method. The translation of the camera within a range of 30[cm] did not cause any significant contamination to the proposed method, while it did to the STDRANSAC.

VI. Conclusions and Future Work

In this paper, we have proposed an approach to rotation estimation using RANSAC and an omni-directional compound sensor. Using the compound sensor, a number of outliers are removed, which significantly reduces the computational cost and improves the robustness of the estimation.

Experiments were carried out to validate our approach. Our proposed method was found to produce reasonable accuracy and robustness in real-time with various environments, and especially with large environments such as outdoor environments. The accuracy of the proposed method does not depend much on the rotation velocity but rather on the translation of the camera. However, when the translation is much smaller compared to the size of the environment, the accuracy is invariant.

The current research on rotation estimation has shown that our division of near and far distant features could effectively improve the performance of rotation estimation. Additional work with translation using near features is currently being conducted, and we hope to report those results in the near future.

Acknowledgement

We would like to give great thanks to Yuichiro Kojima for supporting us with the code of near object detection.

References

[1] A. Ansar and K. Daniilidis, “Linear pose estimation from points or lines”, in ECCV, edited by A. Heyden et al., Copenhagen, Denmark, May 2002, Springer, New York, vol. 4, pp. 282-296.

[2] O. Urfalioglu, “Robust Estimation of Camera Rotation, Translation and Focal Length at High Outlier Rates”, CRV 2004, pp.464-471.

[3] P. Sturm, "Algorithms for plane based pose estimation", Proc. CVPR'00, pp. 1010-1017, 2000

[4] K. Daniilidis, “Fixation Simplifies 3D Motion Estimation”, CVIU, 68(2):158-169, Nov. 1997.

[5] R. F. Vassallo, J. Santos-Victor, and H. J. Schneebeli, “A general approach for egomotion estimation with omnidirectional images”, IEEE Workshop on Omnidirectional Vision, pp. 97-103, June 2002.

[6] K. Daniilidis and H. H. Nagel, "The coupling of rotation and translation in motion estimation of planar surfaces," in Proc. Conf. CVPR, 1993, pp. 188-193.

[7] O. Shakernia, R. Vidal, and S. Sastry, “Omnidirectional egomotion estimation from back-projection flow”, IEEE Workshop on Omnidirectional Vision (in conjunction with CVPR), June 2003.

[8] M. Irani, B. Rousso, and S. Peleg, “Recovery of Ego-Motion Using Region Alignment”, IEEE Trans. on PAMI, 19(3), pp.268-272, Mar. 1997.

[9] Stan Z. Li, Han Wang, and William Y. C. Soh, **“**Robust Estimation of Rotation Angles from Image Sequences Usingthe Annealing M-Estimator”, Journal of Mathematical Imaging and Vision, Vol.8 , Issue 2, March 1998**,** pp.181-192.

[10] A. Makadia, K. Daniilidis, "Rotation Recovery from Spherical Images without Correspondences", IEEE Trans. on PAMI, vol. 28, no. 7, pp. 1170-1175, Jul., 2006.

[11] 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, Vol.14 , issue.2, June 2003, pp.112-120.

[12] R.C. Nelson and J. Aloimonos, “Finding motion parameters from spherical motion fields (or the advantages of having eyes in the back of your head)”, Biological Cybernetics, Vol.58, pp.261-273, 1988.

[13] 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.

[14] D.A. Forsyth and J. Ponce, “Computer Vision: A Modern Approach”, Prentice Hall, 2002.