

Estimation of Mobile Robot Ego-Motion and Obstacle Depth Detection by Using Optical Flow

Xuebing Wang, Kenji Ban, Kazuo Ishii
Graduate School of Life Science and Systems Engineering
Kyushu Institute of Technology
Kitakyusyu, Japan
wang-xuebing@edu.brain.kyutech.ac.jp

Abstract—Estimation of ego-motion and obstacle depth detection plays an important role in environment recognition for navigation of mobile robots. In this paper obstacle perception is performed by using pattern matching between the planar flow and the normal optical flow observed by means of a monocular vision system installed on a mobile robot. The obstacle depth and robot velocity then can be estimated through the coordinate transformation between the image coordinate and the robot coordinate. An experiment is also performed under several different environment settings and shows a good performance for estimation of velocity and the obstacle depth for a mobile robot.

Keywords—ego-motion, depth detection, optical flow

I. INTRODUCTION

The autonomous mobile robot that could work under the living environment is requested and is suggested that they could become common in the future. In order to realize this purpose the environment recognition is indispensable for an autonomous robot such as the perception of obstacles, ego-motion and depth detection etc. In this paper we aim to develop an algorithm for estimation of ego-motion of a mobile robot and obstacle depth detection by using optical flow. Obstacle perception used to estimate the ego-motion and depth is performed by using pattern matching between the planar flow and the optical flow observed by means of a monocular vision system installed on a mobile robot.

For the autonomous navigation of mobile robots, usually vision sensors, infrared sensors or laser range sensors are used in environment recognition such as obstacle perception and ego-motion estimation. There are many research topics about the obstacle perception using infrared radiation sensor or ultrasonic sensor [1]. These sensors can provide the environment depth information directly. But using this method only the obstacle information can be obtained, and the other information for robot need to fuse another sensing system [2], also it is difficult to retrieve the large range simultaneously. For vision system, there are many methods of recognizing the environment such as analysis from information on the landmarks [3] or extract from edge detection by using monocular vision [4], which are depend on surrounding environment around a robot, and difficult to apply in general environments. Also the research that uses optical flow is done as one of other obstacle recognition techniques. A simple approach to obstacle perception is to

examine the optical flow in the left and right halves of the visual field separately, and then to turn toward the direction in which the optical flow is smaller [5]. The motion segmentation of objects and background by using optical flow is described in [6]. In this paper we develop the detection of static obstacles from an image sequence observed from a mobile robot. As a previous work an algorithm about estimation of dominant plane is proposed in [7]. We aim to extend this work to estimation of robot's ego motion and obstacle depth by using a mobile robot under several environments. The experimental evaluation using a real-time monocular vision system shows that this method is practicable for navigation of autonomous mobile robot.

In our work optical flow [8] is calculated by the Lucas-Kanade method with pyramids [9] [10]. Next, in optical flow field three random points are selected to calculate affine coefficient in order to obtain planar flow. Then for segmentation of background and objects, pattern matching is performed between optical flow and planar flow. Through the coordinate transformation between the image coordinate and the robot coordinate we can get obstacle depth from the optical flow in obstacle area. And the relation coefficient between planar flow and robot velocity is also estimated by using experimental planar data.

II. ALGORITHM

In this section, we develop an algorithm for the estimation of mobile robot ego-motion and obstacle depth detection by using optical flow which is obtained from a monocular vision system mounted on a mobile robot.

A. Optical flow estimation

Optical flow is motion vectors in the observer's visual image, and also described by means of spatio-temporal distributions, or fields, of motion vectors [11]. There are several methods for estimation of optical flow based on partial derivatives of the image signal, such as Lucas-Kanade method, and Horn-Schunck method etc. In this paper the Lucas-Kanade method with pyramid is used in respect that for a big change of motion patterns in the visual image this method can solve the aperture problem and save processing time.

As shown in fig.1, the optical flow can be obtained through two successive images due to camera motion. The estimation equation of optical flow shows as (1),

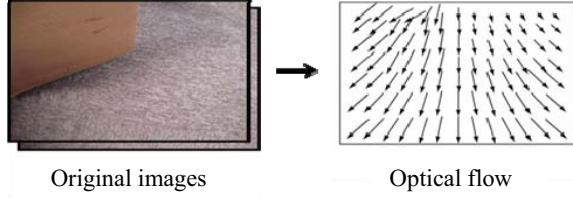


Figure 1. Optical flow can be obtained from successive images due to camera motion .

$$\begin{aligned} \varepsilon^L(d^L) &= \varepsilon^L(d_x^L, d_y^L) \\ &= \sum_{x=u_x^L-w_x}^{u_x^L+w_x} \sum_{y=u_y^L-w_y}^{u_y^L+w_y} (I^L(x, y) - J^L(x + g_x^L + d_x^L, y + g_y^L + d_y^L))^2 \end{aligned} \quad (1)$$

, where I^L is the brightness of previous image and J^L is the brightness of present image. So d^L is the estimation of optical flow when the function of ε^L is near to minimum value. For the details of the derivation of the optical flow using the Lucas-Kanade method with pyramids, see [9].

B. Planar flow estimation

If the camera moves in a plane, planar flow can be obtained when we assume there is no obstacle caught by the camera. That is to say, the planar flow is the background flow. In the real world In order to calculate the planar flow, First, using a pair of successive images from a sequence of images obtained during the camera motion, we compute the optical flow $(u, v)^T$ described in the section of optical flow estimation. The relation of optical flow, start point (u, v) and end point (u', v') of optical flow shows in fig.2, and then we have the relation

$$\begin{pmatrix} u' \\ v' \end{pmatrix} = \begin{pmatrix} u \\ v \end{pmatrix} + \begin{pmatrix} \cdot \\ \cdot \\ \cdot \end{pmatrix} \quad (2)$$

The corresponding points in a pair of successive images which lie on a projection of a plane in a space are connected by an affine transformation as (3).

$$\begin{pmatrix} u' \\ v' \end{pmatrix} = \begin{pmatrix} a & b \\ d & e \end{pmatrix} \begin{pmatrix} u \\ v \end{pmatrix} + \begin{pmatrix} c \\ f \end{pmatrix} \quad (3)$$

Through this equation we can know that a homography between the two images of a planar surface is approximated by an affine transformation if the camera displacement is small,

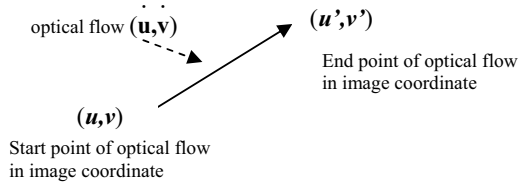


Figure 2. relation of optical flow and start point and end point.

and the affine coefficients can be calculated through 3 pairs of start points and end points. Here we select the points at random in the focal image. The planar flow start points and end points also have the relation as (4).

$$\begin{pmatrix} \hat{u} \\ \hat{v} \end{pmatrix} = \begin{pmatrix} u' \\ v' \end{pmatrix} - \begin{pmatrix} u \\ v \end{pmatrix} \quad (4)$$

If we can obtain the affine coefficients through (4), the planar flow (\hat{u}, \hat{v}) can be calculated by (5).

$$\begin{pmatrix} \hat{u} \\ \hat{v} \end{pmatrix} = \begin{pmatrix} a & b \\ d & e \end{pmatrix} \begin{pmatrix} u \\ v \end{pmatrix} + \begin{pmatrix} c \\ f \end{pmatrix} - \begin{pmatrix} u \\ v \end{pmatrix} \quad (5)$$

C. Segmentation of background and obstacles

Next, we use the estimated planar flow and the computed optical flow to segment the background and obstacles. Setting ε to be the tolerance of the difference between the optical flow vector and the planar flow vector, if

$$\left| \begin{pmatrix} \dot{u} \\ \dot{v} \end{pmatrix} - \begin{pmatrix} \hat{u} \\ \hat{v} \end{pmatrix} \right| < \varepsilon \quad (6)$$

is satisfied, we accept point (u, v) is in the background. In the case that we select at least one point on the obstacle area in the image, the estimated planar flow is no longer the background motion. Therefore, the detected background is very small. In such cases, it becomes evident that the selection of points is incorrect. In those cases, consequently, we select three points randomly again. In our work, we assume that the estimation is success if the detected background area is more than half of image. Otherwise, we need to select the random points again.

Till to segmentation of background and obstacles, the algorithm can be summarized as follows.

1. Compute optical flow (\dot{u}, \dot{v}) from two successive images.
2. Compute affine coefficients in (3) by random selection of three points.
3. Estimate planar flow (\hat{u}, \hat{v}) from affine coefficients.
4. Matching the computed optical flow and planar flow by using (6).
5. Segment the background and the obstacles. If the background area occupies less than half of the image, then go to step 2.

We test these steps by using real successive images and the result shows as Fig.3.

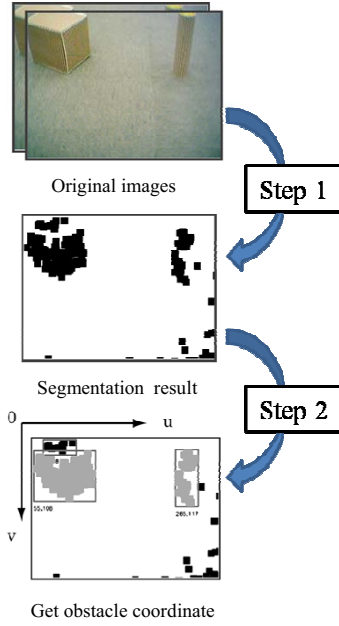


Figure 6. Obstacle depth perception procedure.

$$X_R(u, v) = \frac{H(u - u_0)}{f \cos \theta + (u - u_0) \sin \theta} \quad (10)$$

III. EXPERIMENT

An experiment is performed by using a monocular vision system mounted on a 6-wheeled mobile robot named “Zaurus” in our lab. The aim of this experiment is to make a verification of algorithm in real-time. In order to obtain a robust ego-motion and obstacle depth estimation system the experiment is also performed in several environments by changing the background pattern, the height of camera and the velocity of the mobile robot.

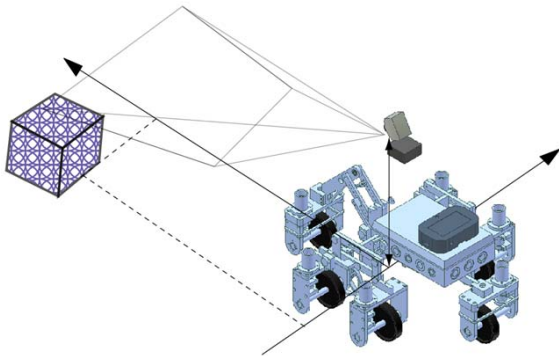


Figure 7. Experiment diagrammatic illustration

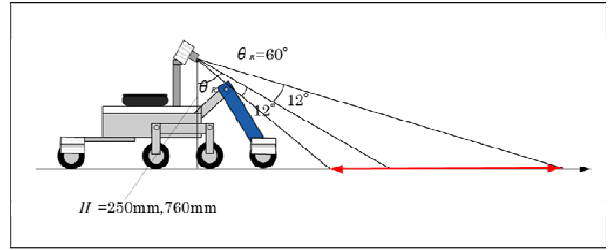


Figure 8. Camera set-up illustration

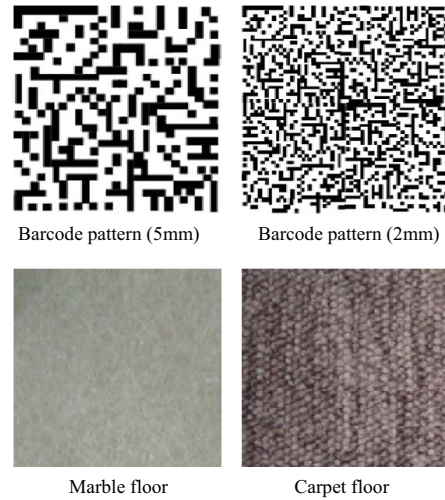


Figure 9. The floor patterns in the experiment.

A. Experimental set-up

The experiment diagrammatic illustration including camera, obstacle, mobile robot and robot coordinate system arrangement shows in Fig.7. The 6-wheeled mobile robot “Zaurus” can give us a lasting approximate constant velocity and less vibration during the experiment so that the error from optical flow estimation can be decreased. A commercially available web-camera with a resolution of 320*240 pixels is used and the length of focus is 187.5mm. The arrangement of camera’s height and elevation shows in Fig.8. And the floor patterns used in the experiment show in Fig.9. Barcode patterns are used to detect the background flow easily. Marble floor and carpet floor are common floor planes in our teaching building.

B. Robot ego-motion experimental result

As described in algorithm section, estimation of robot ego-motion needs average planar flow. A procedure of getting planar flow in experiment shows in Fig.10. First, get the 20*30 planar flows using optical flow. Then average 100 frames’ planar flow by using 3*3 flows in the center of the image. Finally α , the proportionality coefficient, can be calculated using this average value and the robot velocity already known. The result shows in Tab.1. In coordinate between planar flow and robot velocity, the proportionality coefficient α can be linearly-approximated by using linear least-squares method and the result shows in Fig.11. In order to evaluate the ego-motion estimation system, the experiments

are also performed by using calculated proportionality coefficient α under several different robot velocities and background patterns. The result shows in Tab.2 and error ratio in different environment is shown in Tab.3. Robot velocity estimation according to the different patterns on the floor gives us a scattering result. Especially on the marble floor there is a big error by three major reasons. First, there has some reflection of light on the ground, so the optical flow estimation is misrecognition in the successive images sometimes. Second reason is the fewer features on the marble ground, which has led to be unrecognizable for optical flow estimation because optical flow is the distribution of apparent of movement of brightness patterns in an image. The last one is also the most important reason that the image processing speed is a little bit slow for on-line experiments as only 1 or 2 frames per second, which means the features in one frame move more than 1 pixel between this frame and next frame. So the aperture problem happened resulting in the big error of optical flow estimation. To solve this problem the image processing can be embedded in the system or use the high speed optical flow detection image sensor such as Alan stocker 2-D optical flow sensor.

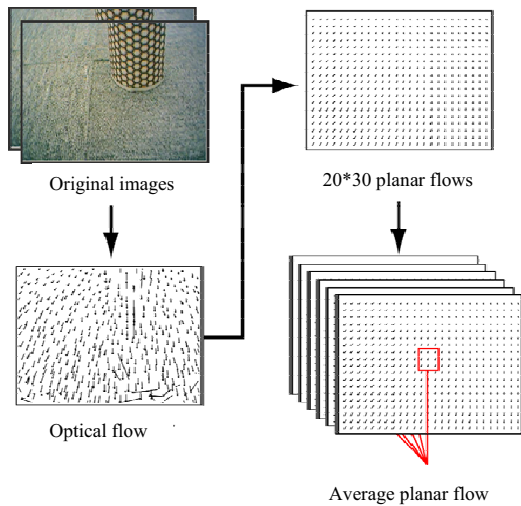


Figure 10. procedure of Average planar flow calculation.

TABLE I. COEFFICIENTE ESTIMATION RESULTS

Robot velocity	av_x (pixel)	av_y (pixel)	$\sqrt{x^2+y^2}$ (pixel)	Coefficient α
0.06m/sec	0.739056	10.2825	10.30903	0.00582
0.08m/sec	1.279331	13.35462	13.41576	0.005963
0.10m/sec	1.517734	15.62204	15.69559	0.006371
0.12m/sec	1.754797	17.77629	17.8627	0.006718
0.14m/sec	2.214845	21.56238	21.67583	0.006459
0.16m/sec	2.143345	23.4556	23.55332	0.006793
0.18m/sec	1.931728	25.98321	26.05492	0.006908
0.20m/sec	2.360653	28.99686	29.09279	0.006875

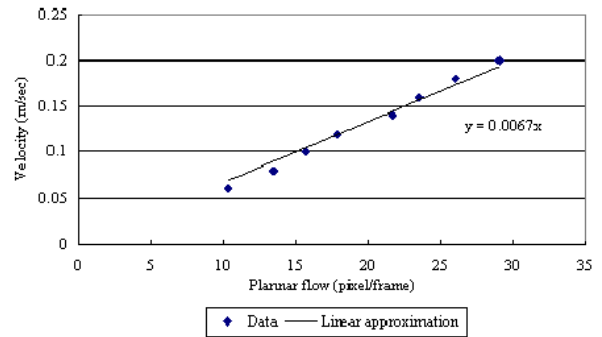


Figure 11. linear approximation of α .

TABLE II. COEFFICIENTE ESTIMATION RESULTS

Robot velocity	Floor pattern	av_x (pixel)	av_y (pixel)	$\sqrt{x^2+y^2}$ (pixel)	Velocity (m/s)
0.05(m/s)	Barcode pattern (2mm)	0.640706	9.006215	9.028976	0.060494
	Barcode pattern (5mm)	0.299386	9.490064	9.494785	0.063615
	Marble floor	0.325644	5.198402	5.208592	0.034898
	Carpet floor	0.106521	7.759395	7.760126	0.051993
0.08(m/s)	Barcode pattern (2mm)	1.279331	13.35462	13.41576	0.089886
	Barcode pattern (5mm)	0.670209	14.46787	14.48338	0.097039
	Marble floor	0.259735	7.315211	7.319821	0.049043
	Carpet floor	0.565834	10.31882	10.33432	0.069240

TABLE III. RELATIVE ERROR [%]

Velocity	Barcode(2mm)	Barcode(5mm)	Marble	Carpet
0.05 m/s	21.0	27.2	30.2	4.0
0.08m/s	12.3	21.3	38.7	13.4

C. Obstacle depth estimation result

As described in algorithm of obstacle depth estimation, an experiment is performed by calculating the obstacle area's nearest point to the observer. The segmentation of obstacle from the background, labeling of obstacle areas and the estimation of obstacle nearest points' coordinate between successive images in on-line experiment are shown in Fig.12. The distance between two objects used in experiment is near to 50cm and the distance from robot start point to obstacles is near to 2m. We set the height of camera to 0.76m and background to the carpet floor. Experimental results in robot velocity at 0.05 m/s and 0.08m/s are shown in Fig.13 and Fig.14 separately. Condition 1 means the corresponding points' number (feature points) in the image is set to small, and oppositely, the number is high in condition 2. This means that we adjust the balance between precision and processing cost.

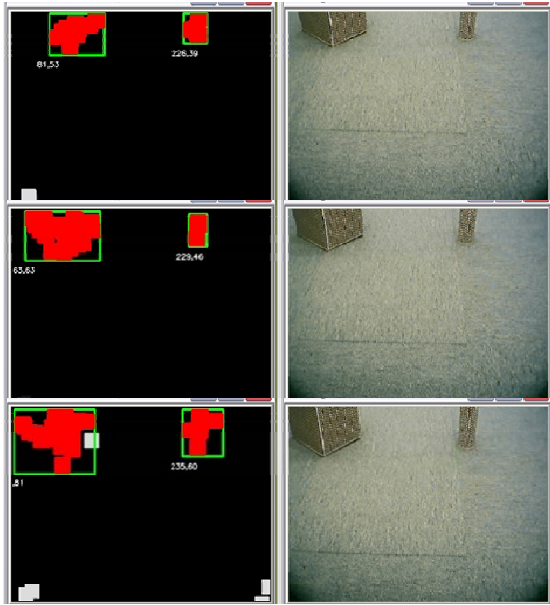


Figure 12. Experimental result on obstacle recognition and obstacle nearest points' coordinate estimation is shown in 2nd, 4th and 6th frame in order.

And from the result we can know the estimation points is back-and-forth and also disconnection happened sometimes because the false recognition of optical flow. To solve this one corresponding point's correspondence become important [13].

IV. CONCLUSIONS

In this paper the planar flow is estimated from optical flow observed by means of a monocular vision system installed on a mobile robot. Through the relation of planar flow and robot velocity, the coefficient is estimated by using experimental data and tested in real time experiment. For obstacle depth perception, the segmentation of background and obstacles is done. The position coordinate is extracted so that we can use the coordinate transformation between image coordinate and world coordinate to calculate the depth of obstacle.

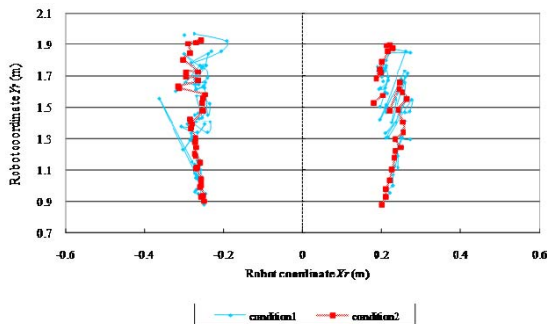


Figure 13. obstacle depth estimation result at 0.05m/s

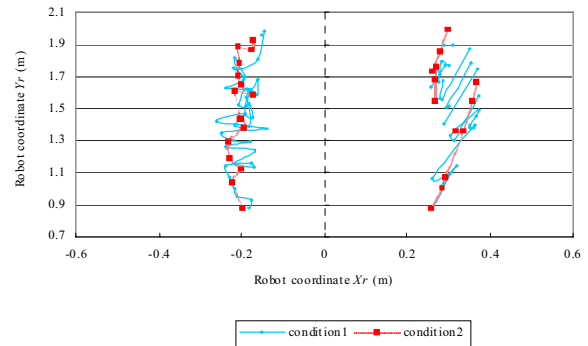


Figure 14. obstacle depth estimation result at 0.08m/s

In the future work, the system can be improved on the side of processing cost decreasing and improvement of optical flow estimation precision. For example, use the optical flow sensor.

- [1] D. Hahnel, R. Triebel, W. Burgard and S. Thrun, "Map building with mobile robots in dynamic environments," Proceedings of the IEEE ICRA 03, 2003
- [2] Hee-Chang Moon, Jae-Hwan Kim, Jung-Ha Kim, "Obstacle detecting system for unmanned ground vehicle using laser scanner and vision," International Conference on Volume, Issue, 17-20, pp.1758 – 1761 October 2007.
- [3] N.D. Guilherme, C.K. Avinash, "Vision for mobile robot navigation: A survey," IEEE Trans. PAMI 24, pp. 237–267, 2002
- [4] S.B. Kang, R. Szeliski, "3D environment modeling from multiple cylindrical panoramic images," Panoramic Vision: Sensors, Theory, Applications, Springer-Verlag, pp. 329–358, 2001.
- [5] J. Santos-Victor, G. Sandini, "Uncalibrated obstacle detection using normal flow," Machine Vision and Applications, Volume 9, Issue 3, Springer-Verlag, 1996.
- [6] J. Barron, D. Fleet and S. Beauchemin, "Performance of optical flow techniques," Int. J. Computer Vision 12 1, pp. 42–77, 1994.
- [7] Naoya Ohnishi and Atsushi Imiya, "Dominant Plane Detection from Optical Flow for Robot Navigation," Pattern Recognition Letters, Vol.27, Issue 9, pp.1009-1021, 2006.
- [8] B.K.P. Horn and B.G. Shunck, "Determining optical flow," Artificial Intelligence, Volume.17, pp.185-203, 1981.
- [9] S. Baker, I. Matthews, "Lucas-Kanade 20 years on: A unifying framework," Internat. J. Computer Vision 56, 221–255, 2004.
- [10] J.Y. Bouguet, "Pyramidal implementation of the Lucas-Kanade feature tracker description of the algorithm," Intel Corporation, Microprocessor Research Labs, OpenCV Documents, 1999.
- [11] H.A. Mallot, "Computational Vision: information processing in perception and visual behavior," Bradford books published, November 2000.
- [12] WH Warren, KJ Kurtz, "The role of central and peripheral vision in perceiving the direction of self-motion," Perception & Psychophysics, 51(5), pp. 443-454, 1992
- [13] T. Carlo and Takeo Kanade, "Detection and Tracking of Point Features," Carnegie Mellon University Technical Report CMU-CS-91-132, April 1991