Diverging viewing-lines in binocular vision: A method for estimating ego motion by mounted active cameras

Akihiro Sugimoto[†] and Tomohiko Ikeda[‡]

[†]National Institute of Informatics, Tokyo 101-8430, Japan
[‡]School of Science and Technology, Chiba University, Japan
sugimoto@nii.ac.jp

Abstract. We exploit the framework of diverging viewing-lines where cameras do not share the common field of view, and propose a method for incrementally estimating ego motion using two mounted active cameras. Our method independently controls the two cameras so that each camera automatically fixates its optical axis to its own fixation point. This camera control allows diverging viewing-lines of the two cameras and leads to accurate ego-motion estimation independent of the baseline distance between the two cameras. We show that the correspondence of the fixation point over two frames together with the displacement field obtained from optical flow nearby the fixation point gives us sufficient constraints to determine ego motion.

1 Introduction

Multi-camera approaches to vision problems have been studied for decades [10, 11]. In particular, stereo vision is most famous and regarded as one of the most important subjects in the computer vision literatures. In using multi-cameras, we have taken it for granted that cameras are set up so that they share the common field of view; the viewing lines of cameras are convergent. In other words, we have paid little attention to the viewing lines of cameras. Almost all the existing methods using multi-cameras are based on the camera setup where the viewing lines of cameras are convergent.

We have, however, another framework in setting up cameras. It is the camera setup where cameras do not share the common field of view; the viewing lines of cameras are divergent [20]. This paper aims at investigating efficiency of employing the framework of diverging viewing-lines and at promoting approaches to vision problems viewed from the diverging viewing-line framework.

Omnidirectional cameras are recently actively investigated [2, 14, 15]. From the viewpoint of viewing lines, omnidirectional cameras can be located in the diverging viewing-line framework. The omnidirectional vision literatures, however, stress only the point of enlarging the fields of view. Again they have paid little attention to the viewing lines of cameras.

To show the effectiveness of the framework of diverging viewing-lines, we exploit the problem of estimating ego motion using mounted cameras. This is because computing three-dimensional camera motion from image measurements is one of

the fundamental problems in computer vision and robot vision. In robot vision, for example, mobile robot navigation and docking require the robot localization, the process of determining and tracking the position (location) of mobile robots relative to their environments [4, 9]. In the wearable computer, on the other hand, understanding where a person is/was and where the person is/was going is a key issue [6, 18] for just-in-time teaching, namely, for providing useful information at the teachable moment.

Most successful approaches[1] to estimating the position and motion of a moving robot use landmarks such as ceiling lights, gateways or doors [16, 21] and are usually based on the framework of stereo vision [7, 8, 13]. When we employ the stereo vision algorithm, however, we have to make two cameras share the common field of view and, moreover, establish feature correspondences across the images captured by two cameras. This kind of processing has difficulty in its stability. In addition, we have another problem in using the stereo vision framework. Namely, though accuracy of the estimation is well known to highly depend on the baseline distance between two cameras, keeping the baseline distance wide is hard when we mount cameras on a robot or wear cameras. Therefore, accuracy of motion estimation is limited if we employ the stereo vision algorithm.

In this paper, we employ the framework of diverging viewing-lines and propose a method for incrementally estimating ego motion using two mounted active cameras. In our method, the fixation control, the camera control in which a camera automatically fixates its optical axis to a selected point (called the fixation point) in 3D, plays a key role. Our method applies this fixation control independently to each active camera. This camera control is called the binocular independent fixation control[17] (Fig. 1). The correspondence of the fixation point over two frames together with optical flow nearby the fixation point gives us sufficient constraints to determine ego motion in 3D.

In the binocular independent fixation control, each camera fixates its optical axis to its own fixation point in 3D and two fixation points are not necessarily the same. This indicates that the two cameras need not share the common field of view. The viewing lines of the two cameras are divergent in this case in contrast to stereo vision where convergence is always imposed on two viewing lines. Moreover, in the binocular independent fixation control, estimation accuracy becomes independent of the baseline distance between two cameras and is expected to become higher than the case where we use the stereo vision algorithm. This can be understood as follows. If we assume that we set a camera at each fixation point and that the optical axis of each camera is toward a robot or a person, then the binocular independent fixation control can be regarded as the situation where we apply the stereo vision

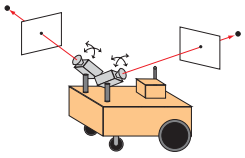


Fig. 1. Binocular independent fixation control.

[1] Several approaches to ego-motion estimation are carefully compared in [19].

framework to estimating the position of the robot or the person from the two fixation points. The baseline distance in this case is identical with the distance between the two fixation points. (We can embed the baseline into the scene.) This means that estimation accuracy is independent of the baseline distance between two mounted cameras and that selecting fixation points as far as possible from each other allows estimation accuracy to become high. Our intensive experiments demonstrate this effectiveness.

2 Geometric constraints on ego motion

We here derive geometric constraints on ego motion based on information obtained during the binocular independent fixation control. Between two mounted cameras, i.e., a right camera and a left camera, we set the right camera is the base. Moreover, for simplicity, we assume that the orientation of the camera coordinates does not change even though we change pan and tilt of the camera for the fixation control. This means that only the ego motion causes changes in orientation and translation of the camera coordinates. We also assume that the ego motion is identical with the motion of the base-camera coordinates. We thus develop a method to estimate the motion of the right-camera coordinates.

We assume that the extrinsic parameters between the two cameras as well as the intrinsic parameters of each camera are calibrated in advance. Namely, we let the translation vector and the rotation matrix to make the left-camera coordinates identical with the right-camera coordinates be $\boldsymbol{T}_{\rm in}$ in the left-camera coordinates and $R_{\rm in}$ in the right-camera coordinates, respectively. $\boldsymbol{T}_{\rm in}$ and $R_{\rm in}$ are both assumed to be known.

2.1 Constraints from fixation correspondence

The fixation control gives us the correspondence of the viewing lines of a camera toward the fixation point over time-series frames. We call this correspondence a fixation correspondence. The fixation correspondence enables us to derive a constraint on the ego motion [17].

Let the projection centers of the left camera and the right camera be C_ℓ^t and $C_{\rm r}^t$ in 3D at time t. We assume that the both cameras have their own fixation points P_ℓ and $P_{\rm r}$, and that P_ℓ is different from $P_{\rm r}$. We denote by $\boldsymbol{v}_{\rm r}^t$ the unit vector from $C_{\rm r}^t$ to $P_{\rm r}$ in the right-camera coordinates at time t. We see that $\boldsymbol{v}_{\rm r}^t$ represents the viewing line of the right camera toward the fixation point at time t. We also denote by $\boldsymbol{v}_\ell^t$ the unit vector from C_ℓ^t to P_ℓ in the left-camera coordinates at time t (Fig. 2).

We first focus on the right camera. We assume that the projection center of the right camera moves from $C_{\rm r}^t$ to $C_{\rm r}^{t+1}$ in 3D due to the ego motion from time t to $t+1$ (Fig. 3). We also assume that the rotation and the translation of the right-camera coordinates incurred by the ego motion are expressed as rotation matrix R in the right-camera coordinates at time t and translation vector $\boldsymbol{T}$ in the world coordinates. We remark that the orientation of the world coordinates is assumed to be obtained by applying rotation matrix R_0^{-1} to the orientation of the right-camera coordinates at time t. Our aim here is to derive constraints on R and $\boldsymbol{T}$ using $\boldsymbol{v}_{\rm r}^t$ and $\boldsymbol{v}_{\rm r}^{t+1}$, both of which are obtained from the captured images at time t and $t+1$.

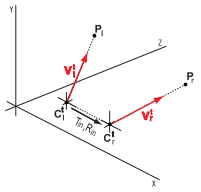


Fig. 2. Relationship between the projection centers and the fixation points at time t.

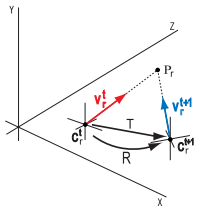


Fig. 3. Geometry based on the fixation correspondence of the right camera.

It follows from the fixation correspondence of the right camera that

$$\det \left[\; R_0 \boldsymbol{v}_{\mathrm{r}}^t \;\middle|\; R_0 R \boldsymbol{v}_{\mathrm{r}}^{t+1} \;\middle|\; \boldsymbol{T} \; \right] = 0, \tag{2.1}$$

which gives the constraint on the ego motion, R and $\boldsymbol{T}$, derived from the fixation correspondence of the right camera.

On the other hand, $\boldsymbol{v}_\ell^t$ in the left-camera coordinates at time t is identical with $R_{\mathrm{in}} \boldsymbol{v}_\ell^t$ in the right-camera coordinates at time t. The rotation R of the right-camera coordinates from time t to $t+1$ causes the translation $-R_0(R-I)R_{\mathrm{in}}\boldsymbol{T}_{\mathrm{in}}$ of the left-camera coordinates in the world coordinates where I is the 3×3 unit matrix. This yields a counterpart of (2.1). The fixation correspondence of the left camera gives the following constraint on the ego motion:

$$\det \left[\; R_0 R_{\mathrm{in}} \boldsymbol{v}_\ell^t \;\middle|\; R_0 R R_{\mathrm{in}} \boldsymbol{v}_\ell^{t+1} \;\middle|\; \boldsymbol{T} - R_0(R-I)R_{\mathrm{in}}\boldsymbol{T}_{\mathrm{in}} \; \right] = 0. \tag{2.2}$$

(2.1) and (2.2) are the constraints on the ego motion in 3D derived from the fixation correspondences obtained by the binocular independent fixation control. When we have estimated the ego motion up to time t, we know R_0. Then, the unknown parameters in (2.1) and (2.2) are R and $\boldsymbol{T}$. We see that (2.1) and (2.2) give quadratic constraints on R and $\boldsymbol{T}$ respectively.

2.2 Constraints from optical flow nearby the fixation point

Ego motion has 6 degrees of freedom: 3 for rotation and 3 for translation. The number of constraints on ego motion derived from two fixation correspondences is, on the other hand, two ((2.1) and (2.2)). We therefore need to derive more constraints to estimate the ego motion.

Sugimoto `et al.`[17] proposed to use line correspondences nearby the fixation point to obtain other constraints on ego motion. This is because line correspondences across images can be easily established due to the spatial extent of the line. Using line correspondences, however, falls in a fatal problem unless lines in 3D are carefully selected. In the case where ego motion is just translation or the case where the projection center moves on the plane defined by an employed line and the projection center, the constraints derived from the line correspondence become the identical equations. Namely, the constraints in such a case do not make sense and no more independent constraint on the ego motion is obtained. This indicates that once a line is selected for obtaining constraints on ego motion, the ego motion

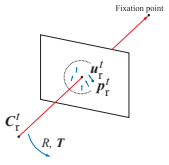


Fig. 4. Optical flow nearby the fixation point of the right camera.

itself is restricted from that time. Estimating ego motion and selecting lines then become a which-came-first-the-chicken-or-the-egg question.

To avoid falling in such a problem, we here employ optical flow[2] nearby the fixation point and use the point displacement obtained from the flow field. Since a camera focuses on its fixation point during the fixation control, computing optical flow only nearby the fixation point is not cost-consuming and not expensive. The computed flow field then enables us to derive constraints on ego motion.

We first focus on the right camera. We assume that we compute optical flow nearby the fixation point over time t and $t+1$ where optical flow is incurred by ego motion only. Let $\boldsymbol{p}_{\rm r}^t$ be the coordinates of a point nearby the fixation point in the right-camera image at time t, and $\boldsymbol{u}_{\rm r}^t$ be the flow vector at the point (Fig. 4). We now have the following relationship between optical flow $\boldsymbol{u}_{\rm r}^t$ and ego motion R and $\boldsymbol{T}$:

$$\det \left[\begin{array}{c|c|c} R_0 R(M\boldsymbol{u}_{\rm r}^t + \widetilde{\boldsymbol{p}}_{\rm r}^t) & R_0 \widetilde{\boldsymbol{p}}_{\rm r}^t & \boldsymbol{T} \end{array} \right] = 0, \tag{2.3}$$

where

$$M := \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}^\top, \qquad \widetilde{\boldsymbol{p}}_{\rm r}^t := \left((\boldsymbol{p}_{\rm r}^t)^\top \ f_{\rm r} \right)^\top,$$

and $f_{\rm r}$ is the focal length of the right camera.

This constraint is derived as follows. From the relationship of the orientations among the world coordinates, the right-camera coordinates at time t and the right-camera coordinates at time $t+1$, we have

$$\alpha R_0 \widetilde{\boldsymbol{p}}_{\rm r}^t = \alpha' R_0 R \widetilde{\boldsymbol{p}_{\rm r}^{t+1}} + \boldsymbol{T},$$

where α and α' are unknown positive constants. On the other hand, the definition of $\widetilde{\boldsymbol{p}}_{\rm r}^t$ gives

$$\widetilde{\boldsymbol{p}_{\rm r}^{t+1}} = M^\top \boldsymbol{p}_{\rm r}^{t+1} + \left(0 \ 0 \ f_{\rm r} \right)^\top.$$

Combining the above two equations with the definition of optical flow $\boldsymbol{u}_{\rm r}^t := \boldsymbol{p}_{\rm r}^{t+1} - \boldsymbol{p}_{\rm r}^t$, we obtain

$$\alpha R^\top \widetilde{\boldsymbol{p}}_{\rm r}^t = \alpha' \left[M\boldsymbol{u}_{\rm r}^t + \widetilde{\boldsymbol{p}}_{\rm r}^t \right] + R^\top R_0^\top \boldsymbol{T},$$

[2] A number of methods have been proposed for optical flow computations (see [1, 3] for the current state of the art). Optical flow is now stably and accurately recovered with simple computation [12].

from which (2.3) follows.

In the similar way, optical flow obtained from the left camera gives us the constraint on the ego motion:

$$\det \left[\; R_0 R R_{\rm in}(M\boldsymbol{u}_\ell^t + \widetilde{\boldsymbol{p}_\ell^t}) \;\middle|\; R_0 R_{\rm in}^\top \widetilde{\boldsymbol{p}_\ell^t} \;\middle|\; \boldsymbol{T} + R_0(I-R)R_{\rm in}\boldsymbol{T}_{\rm in} \; \right] = 0, \quad (2.4)$$

where $\boldsymbol{p}_\ell^t$ is a point in the left-camera image at time t and $\boldsymbol{u}_\ell^t$ is the flow vector at the point. Note that $\widetilde{\boldsymbol{p}_\ell^t}$ is defined by $\widetilde{\boldsymbol{p}_\ell^t} := \left((\boldsymbol{p}_\ell^t)^\top \;\; f_\ell\right)^\top$ where f_ℓ is the focal length of the left camera.

Whenever we obtain one flow vector (at a point in an image), we have one constraint on ego motion in the form of (2.3) or (2.4) depending on the camera that captures the image. This constraint is quadratic with respect to unknowns, i.e., R and $\boldsymbol{T}$.

2.3 Estimation of ego-motion

In the binocular fixation control, two fixation points have no relationship. This indicates that two constraints, (2.1) and (2.2), are algebraically independent. An optical-flow vector obtained nearby a fixation point has no relationship with the fixation point except that the direction of the flow vector may be similar to that at the fixation point. The constraint derived from the flow vector is, thus, algebraically independent of the constraints derived from the fixation correspondences. We can therefore estimate ego motion if we have optical-flow vectors at more than four points.

To be more concrete, we form a simultaneous system of nonlinear equations that consists of the constraints derived from the fixation correspondence of each camera, the constraints derived from optical flow and the orthogonality constraints imposed on rotation, i.e., $RR^\top = I$, and then apply a nonlinear optimization algorithm such as the Levenberg-Marquart method to solve the system. Parameters optimizing the system give the estimation of ego motion.

3 Algorithm

Based on the discussion above, we present here the algorithm for estimating ego motion based on the binocular independent fixation control. In the algorithm below, ego motion is assumed to occur just before Step 3.

Step 1: Capture an image by each camera, and set $t = 1$ and $C := \{\mathrm{r}, \ell\}$.
Step 2: While $C \neq \phi$, do the following procedures for each $i \in C$.
- (a) Detect fixation point P_i by camera i and control camera i so that its optical axis is toward P_i, and compute $\boldsymbol{v}_i^t$.
- (b) In the image captured by camera i, set a small region R_i whose center is the projection of P_i.
- (c) $C := C - \{i\}$

Step 3: Capture an image by each camera.
Step 4: For $i = \mathrm{r}, \ell$, do the following procedures.
- (a) Compute optical flows in R_i.
- (b) Control camera i so that its optical axis is toward P_i, and compute $\boldsymbol{v}_i^{t+1}$. If camera i cannot capture P_i, go to Step (c). Otherwise, goto Step 5.

(c) Add i to C and then return to Step 2.

Step 5: Select at least four points whose flow vectors are computed at Step 4.

Step 6: Estimate ego motion by combining (2.1), (2.2), (2.3), (2.4) and $RR^\top = I$.

Step 7: Set $t = t + 1$, and return to Step 3.

4 Experiments

4.1 Numerical evaluation on estimation accuracy

To verify the potential applicability of the proposed method, we here present two kinds of numerical evaluation on estimation accuracy: one is on estimation accuracy depending on the distance between two fixation points, and the other is on noise sensitivity with a fixed distance between two fixation points.

The parameters in the simulation is as follows. Two cameras are set with the baseline distance of 1.0 where each camera is with 21^o angle of view and with the focal length[3] of 0.025. The two cameras are in the same pose: the orientation of the camera coordinates is the same. The size of images captured by the cameras is 512×512 pixels. For each camera, we generated one fixation point at 10.0 depth forward from the projection center. Note that the Newton-Raphson algorithm was used in the nonlinear optimization.

A. Accuracy depending on the distance between two fixation points

In the first experiment, we changed distances between two fixation points and then evaluated estimation accuracy at each distance.

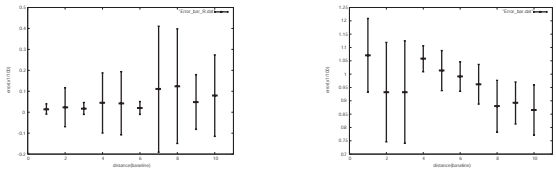
We set a distance between two fixation points and generated two fixation points in 3D for the two cameras satisfying the distance. We also randomly generated 2 points in 3D nearby each fixation point for the optical flow computation. The images of the two points were within the window of 20×20 pixels whose center is the image of the fixation point. We then randomly generated a translation vector with the length of 0.25. We also generated a rotation matrix where the rotation axis is vertical and the rotation angle is within 1.0 degree. The restrictions imposed in generating translations and rotations come from the realization of the fixation control and the optical flow computation in the practical situation. Based on the generated translation and rotation, the cameras were moved.

Before and after the motion, we projected all the points generated in 3D onto the image plane to obtain image points that were observed in terms of pixels. To all the image points, we added Gaussian noise. Namely, we perturbed the pixel-based coordinates in the image by independently adding Gaussian noise with the mean of 0.0 pixels and the standard deviation of 2.0 pixels. Next, we computed optical flow of the points generated nearby the fixation points to obtain the flow vectors. We then applied our algorithm to obtain ego-motion estimation: rotation matrix $\widehat{R}$ and translation vector $\widehat{\boldsymbol{T}}$.

To evaluate the errors in the estimation, we computed the rotation axis $\widehat{\boldsymbol{r}}$ ($\|\widehat{\boldsymbol{r}}\| = 1$) and rotation angle $\widehat{\theta}$ from $\widehat{R}$. We then defined the evaluation of the errors of $\widehat{R}$ and $\widehat{\boldsymbol{T}}$ by

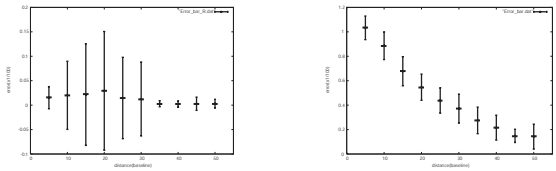
$$\frac{\|\widehat{\theta}\widehat{\boldsymbol{r}} - \theta\boldsymbol{r}\|}{|\theta|} \text{ and } \frac{\|\widehat{\boldsymbol{T}} - \boldsymbol{T}\|}{\|\boldsymbol{T}\|},$$

[3] When the baseline distance is 20cm, then the focal length is 0.5cm, for example.



(a) rotation (b) translation

Fig. 5. Estimation errors depending on small changes in distance between two fixation points (average with standard deviation).



(a) rotation (b) translation

Fig. 6. Estimation errors depending on great changes in distance between two fixation points (average with standard deviation).

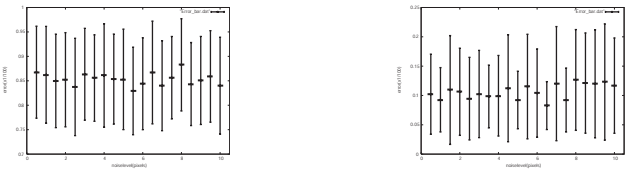
where $\boldsymbol{r}$ ($\|\boldsymbol{r}\| = 1$) is the rotation axis and θ is the rotation angle of the ground truth, and $\boldsymbol{T}$ is the true translation vector ($\|\boldsymbol{T}\| = 0.25$). We iterated the above procedures 200 times and computed the average and the standard deviation of the errors in the estimation at the distance between two fixation points.

To see the performance of estimation accuracy depending on the distance between two fixation points, we changed the distance by 1.0 from 1.0 to 10.0. We also changed the distance by 5.0 from 5.0 to 50.0. These results are shown in Figs. 5 and 6. The former can be regarded as the case where our method is applied to the indoor scene while the latter is the case to the outdoor scene. This is because in the indoor scene, the distance between fixation points is at longest 10 times of the baseline distance, for example.

From Figs. 5 and 6, we see that rotations are almost accurately estimated regardless of the distance between two fixation points. As for the translation estimation, we observe the improving tendency in accuracy as the distance between two fixation points become larger. In particular, this tendency is marked when the distances are greatly changed (Fig. 6 (b)). This observation means that our method is more effective when the distance between two fixation points is very large, i.e., when it is applied to the outdoor scene, for example. The standard deviations, on the other hand, are a little bit large in the all cases. Local minimum traps in the nonlinear optimization sometimes cause inaccurate estimation; this may lead to a large standard deviation. Incorporation of a more sophisticated optimization algorithm is required for more stable estimation results.

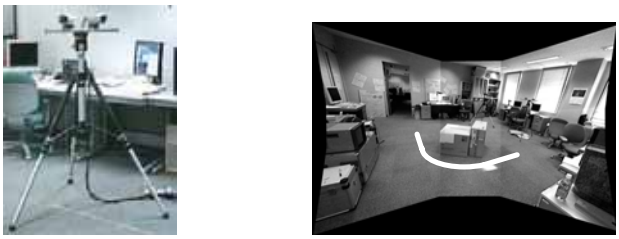
B. Robustness against noise

In the second experiment, we evaluated the robustness against noise in our estimation. We focused here on the evaluation of accuracy of estimated translation vectors. This is because we have found in the first experiment that rotations are fairly accurately estimated by our method independent of the distance between two fixation points, and because position information is more important for localization of a mobile robot or a moving person.



(a) with the distance of 10.0 (b) with the distance of 50.0

Fig. 7. Position errors in the estimation against noise level under the constant distance between two fixation points.



(a) binocular vision sensor (b) ego-motion trajectory

Fig. 8. Experimentation setup.

In this experiment, we first set the distance between two fixation points to be 10.0. We then generated points in 3D, i.e., fixation points and points for the optical flow computation, and rotation and translation in the same way as the first experiment.

We added Gaussian noise to all the image points. In this case, we perturbed the pixel-based coordinates in the image by independently adding Gaussian noise with the mean of 0.0 pixels and several standard deviation levels. The standard deviations were changed by 0.5 pixels from 1.0 to 10.0 pixels. We then estimated the translation vector and evaluated accuracy. The results are shown in Fig. 7 (a). We also conducted the same experiment in the case where the distance between two fixation points is 50.0, the results of which are shown in Fig. 7 (b).

Figure 7 shows that estimation accuracy remains stable up to the noise level of 10 pixels. It is reasonable that when greater noises are added in observation, estimation becomes less accurate. However, our results show that our method is robust even against 10-pixels noise. We also observe large standard deviations in estimation errors as in the case of the first experiment. Enhancing stableness in estimation is left for the future work.

4.2 Ego-motion trajectory estimation using real images

We applied our method to estimating ego-motion trajectory in the real situation. We employed two off-the-shelf cameras (EVI-G20 from Sony) as active cameras and set up a binocular vision sensor where two cameras with the baseline distance of about 20cm were mounted on the stage of a tripod (Fig. 8 (a)). We remark that in our setup the viewing lines of the two cameras are divergent. We then calibrated the intrinsic and extrinsic parameters of the two cameras with the method proposed by Zhang [22]. The size of images captured by each camera was 640×480 pixels.

(a) left-camera image (b) right-camera image (c) computed optical flow

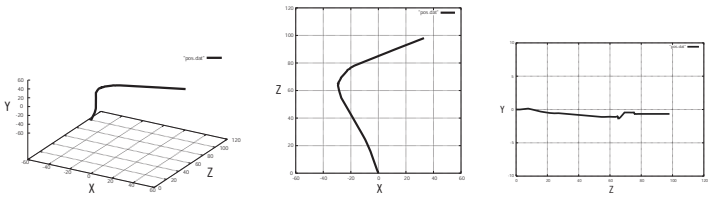
Fig. 9. A pair of captured images and computed optical flow.

We moved the binocular vision sensor in the scene. The trajectory of the right-camera motion is shown in Fig. 8 (b). The length of the trajectory was about 3m. We marked 85 points on the trajectory and regarded them as samples during the ego motion. (In other words, 85 points were sampled during the ego motion of about 3m.) We then applied our method only to the samples, i.e., the marked points, to estimate the ego motion.

In each image captured by each camera at the starting point of the motion, we manually selected a point to serve as the fixation point. During the estimation, we updated fixation points 4 times for each camera. This updating was also conducted by hand. The distance between two fixation points for the binocular vision sensor was about 3m. We remark that we selected fixation points so that the viewing lines of the two cameras always become divergent. We computed optical flow within the window of 100×100 pixels whose center is the image of the fixation point. We used two optical-flow vectors for each camera (we thus used four optical-flow vectors in total). The vectors were randomly selected within the windows of 30×30 pixels whose center is the image of the fixation point. In computing optical flow, we used the Kanade-Lucas-Tomasi algorithm [5, 12]. (a) and (b) in Fig. 9 show an example of an image pair captured by the right and left cameras at a marked point. We see that no field of view of the two cameras is common. (c) in Fig. 9, on the other hand, shows the computed optical flow for the right camera at the marked point. We remark that the fixation point (the black circle) and the window of 30×30 pixels are overlaid onto the images in Fig. 9.

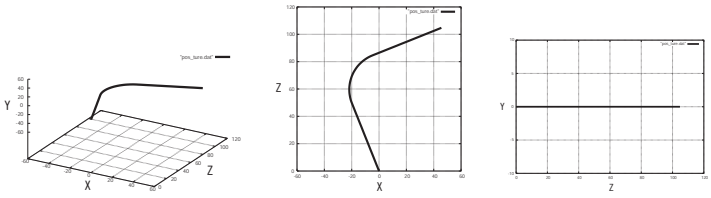
Under the above conditions, we estimated the right-camera motion at each marked point. Fig. 10 shows the trajectory of the right-camera motion that was obtained by concatenating the estimated motions at the marked points. For the comparison, the ground truth of the trajectory is shown in Fig. 11.

From the comparison of Figs. 10 and 11, we see that the shape of the trajectory is almost correctly estimated. The shape of the estimated trajectory almost coincides with that of the actual trajectory when viewed from some viewpoints. In fact, the average of errors in position estimation over the marked points was 0.271cm and the standard deviation was 0.349cm. (The distance between two adjacent marked points was 4.20cm on the average.) The position error at the terminal marked point was about 25cm. At some marked points, we observe great errors in estimation and aberration from the actual trajectory. We have two reasons that may cause this aberration. One is the incorrect estimation of the motion at the marked points and the other is the accumulation of errors due to incremental estimation. The estimation error can be caused by errors in the fixation correspondences or errors in the optical-flow computation.



(a) 3D representation (b) top view (c) side view

Fig. 10. Estimated trajectory of the ego motion.



(a) 3D representation (b) top view (c) side view

Fig. 11. Trajectory of the ego motion (the ground truth).

As we see above, we may conclude that though we have some errors in estimation and need some improvements on our method, our experiments demonstrate the applicability of our method to the real situation.

5 Concluding Remarks

We proposed a method for incrementally estimating ego motion by two mounted active cameras. Our method independently controls the two active cameras so that each camera automatically fixates its optical axis to its own fixation point. The correspondence of the fixation point over two frames together with the displacement field obtained from optical flow nearby the fixation point gives us sufficient constraints to determine ego motion in 3D.

In our method, two cameras need not share the common field of view because each camera fixates its optical axis to its own fixation point in 3D and because two fixation points are not necessarily the same. In using binocular cameras, the framework of stereo vision has a long history in its usage and only that framework has been studied so far where the viewing lines of two cameras are convergent. In contrast, our method stands in the other framework where the viewing lines of two cameras are divergent. We believe that our method puts the binocular vision in a light and promotes the framework of diverging viewing-lines in using multiple cameras.

Included in the future works are (1) enhancing stableness in estimation, (2) incorporating a mechanism to eliminate accumulation errors in estimation and (3) developing a fully automatic system that realizes the proposed method.

Acknowledgements This work is in part supported by Grant-in-Aid for Scientific Research of the Ministry of Education, Culture, Sports, Science and Technology of Japan under the contract of 13224051 and 14380161.

References

1. L. Alvarez, J. Weickert and J. Sanchez: Reliable Estimation of Dense Optical Flow Fields with Large Displacements, Int. J. of Computer Vision, 39 (2000), 1, 41–56.
2. P. Baker, R. Pless, C. Fermüller and Y. Aloimonos: New Eyes for Shape and Motion Estimation, Proc. of IEEE Workshop on Biologically Motivated Computer Vision, 118–128, 2000.
3. S.S. Beauchemin and J. L. Barron: The Computation of Optical Flow, ACM Computer Surveys, 26 (1995), 433–467.
4. J. Borenstein, B. Everentt and L. Feng: Navigating Mobile Robots: Systems and Techniques, A. K. Peters, Ltd., Wellesley, MA, U.S.A., 1996.
5. J.-Y. Bouguet: Pyramidal Implementation of the Lucas Kanade Feature Tracker Description of the algorithm, Technical Report of Intel. Research Lab. 1999.
6. A. J. Davison, W. W. Mayol and D. W. Murray: Real-Time Localisation and Mapping with Wearable Active Vision, Proc. IEEE/ACM Int. Symposium on Mixed and Augmented Reality, 18–27, 2003.
7. A. J. Davison and D. W. Murray: Mobile Robot Localisation Using Active Vision, Proc. of ECCV, Vol. 2, 809–825, 1998.
8. A. J. Davison and D. W. Murray: Simultaneous Localization and Map-Building Using Active Vision, IEEE Trans. on PAMI, 24, 7, 865–880 (2002).
9. G. N. DeSouza and A. C. Kak: Vision for Mobile Robot Navigation: A Survey, IEEE Transactions on PAMI, 24, 2, 237–267 (2002).
10. O. Faugeras and Q.-T. Luong: The Geometry of Multiple Images– The Laws that Govern the Formation of Multiple Images of a Scene and Some of Their Applications– , MIT press, 2001.
11. R. Hartley and A. Zisserman: Multiple View Geometry in Computer Vision, Cambridge Univ. Press, 2000.
12. B. D. Lucas and T. Kanade: An Iterative Image Registration Technique with an Application to Stereo Vision, Proc. of Int. Joint Conf. on Artificial Intelligence, 674–679, 1981.
13. N. Molton and M. Brady: Practical Structure and Motion from Stereo When Motion is Unconstrained, Int. J. of Computer Vision, 39, 1, 5–23 (2000).
14. T. Pajdla: Stereo with Oblique Cameras, Int. J. of Computer Vision, 47, 161–171 (2002).
15. R. Pless: Using Many Cameras as One, Proc. of CVPR, 2003.
16. R. Sim and G. Dudek: Mobile Robot Localization from Learned Landmarks, Proc. of IEEE/RSJ Conf. on Intelligent Robots and Systems, 1998.
17. A. Sugimoto, W. Nagatomo and T. Matsuyama: Estimating Ego Motion by Fixation Control of Mounted Active Cameras, Proc. of ACCV, 67–72, 2004.
18. A. Sugimoto, A. Nakayama and T. Matsuyama: Detecting a Gazing Region by Visual Direction and Stereo Cameras, Proc. of the 16th International Conference on Pattern Recognition, Vol. III, 278–282, 2002.
19. T. Y. Tian, C. Tomasi and D. J. Heeger: Comparison of Approaches to Egomotion Computation, Proc. of CVPR, 315–320, 1996.
20. J. S.-Victor, G. Sandini, F. Curotto, S. Garibaldi: Divergent Stereo in Autonomous Navigation: From Bees to Robots, Int. J. of Computer Vision, 14, 159–177 (1995).
21. M. Werman, S. Banerjee, S. D. Roy and M. Qiu: Robot Localization Using Uncalibrated Camera Invariants, Proc. of CVPR, Vol. 2, 353–359, 1999.
22. Z. Zhang: A Flexible New Technique for Camera Calibration, IEEE Transactions on PAMI, Vol. 22, No. 11, 1330–1334 (2000).