Parallel Execution of SVOs for Estimation of Moving Objects

Kazuhiro TOMINAGA and Takeshi NISHIDA

***Abstract*—We propose a relative position estimation method of multiple moving objects using a monocular camera. In this method, moving objects are extracted and tracked on captured images, and the relative position and transitions are estimated by a probabilistic visual estimation method SVO (fast semi-direct monocular visual odometry). SVO executes precise and robust visual odometry and has been proposed for use in a micro aerial vehicle (MAV) that has a single downward-looking camera. In our proposed method, we execute multiple SVOs in parallel, not only for executing visual odometry, but also for the relative distance estimation between multiple moving objects. The estimation results of each SVO are integrated, and the relative positions of targets are robustly estimated in real time. The effectiveness of the proposed method is shown through several experimental simulations.**

I. Introduction

Recently, many research studies on mobile robots and drones have been performed. In many of these studies, size and weight limitations make the use of multiple sensors and a large computer system unfeasible. To deal with these restrictions, various methods on high-speed three-dimensional map generating and localization by a small monocular camera have been proposed.

In a traditional localization method, the features of a scene are extracted and compared with known map data in each captured image; then, the estimated camera position and map data are both updated. However, the computational cost of updating of the complete map data is too high. Therefore, the parallel tracking and mapping (PTAM) [1] method updates the map only when the key frame is obtained. The map updating and the tracking of feature points are processed in different threads, thus increasing the speed of the position estimation calculation. Moreover, the fast semi-direct monocular visual odometry (SVO) [2] executes the tracking of feature points, map updating, and feature point extraction when the key frame is obtained, achieving further high speed processing. In this method, the relative pose estimation of the camera and feature points is achieved together with precision by Bayesian estimation. However, SVO is designed for processing of the image obtained from a monocular camera mounted beneath the drone and has a restriction that no object is moving relative to the drone in the image. Thus, when the image includes some relative moving objects, the self-localization by SVO cannot be executed properly. Therefore, in this research, we propose a method for overcoming this limitation of SVO. In the proposed method, multiple SVOs matching the number of moving objects are executed in parallel. Each SVO compiles a relative map against the moving objects independently. By integrating these together, it is then possible to obtain the relative movement of the objects and the structure of the whole map. The flow chart of the proposed method is shown in Fig. 1.

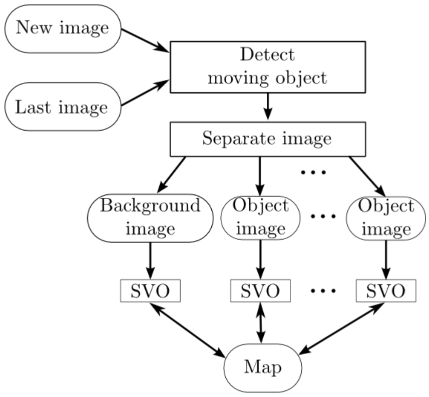

Fig. 1. Flowchart of the proposed method.

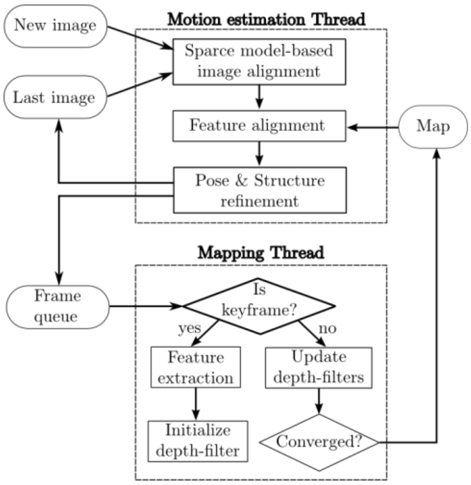

Fig. 2. Flowchart of SVO.

II. Svo

The SVO executes the motion estimation thread and mapping thread parallel in real time. The flow chart is shown in Fig. 2.

A. *Motion Estimation Thread*

In the motion estimation thread, the correspondence between the feature points in time-series images are extracted, and the pose change of camera is estimated from them. In the SVO, because the feature extraction is executed only at a key frame, the resulting feature points in the key frame are tracked in other frames. For robustness in the tracking and searching of correspondence, minimization of the photometric error using the patch and re-projection error are executed.

K. Tominaga and T. Nshida are with Kyushu Institute of Technology, Kitakyushu, Fukuoka, Japan (e-mail: nishida@cntl.kyutech.ac.jp).

B. *Mapping Thread*

In the mapping thread, the 3D position of features in the image is estimated by a probabilistic depth filter. The filter estimates the depths of features using a probabilistic distribution function (PDF). Depths of the feature points are calculated using sequentially obtained coordinate transformation, and the PDFs are updated. After the PDF updates are converged, the maximum a posteriori of the PDF is adopted as a depth of the feature point and integrated into the environmental map.

III. Moving Object Detection

For this research, we construct two types of SVO. One executes visual odometry by extracting the ground information, the other estimate the relative position of moving objects by extracting them. The former can be achieved with the use of a normal SVO; however, the execution of the latter requires the extraction of moving objects. In the case of the fixed camera, moving objects can be extracted by background subtraction or the interframe difference. In the case of the moving camera, estimating the object movement and background region at the same time is necessary. In the proposed method, the moving objects are detected using the interframe difference method after executing the alignment of the background between the time sequence images by using a projective transformation.

A. *Alignment of background*

Feature points in the sequential images are tracked by the Lucas-Kanade method [3], and the projection matrix denotes the amount of camera motion by using the obtained correspondence of the feature points. By using this matrix, the backgrounds in the sequential images are aligned.

B. *Inter-frame Difference*

Detected moving objects are tracked by the mean shift method. This is one of the solutions of the mode search problem as it tracks targets by maximizing the Bhattacharyya coefficient. The amount of motion is estimated by this procedure.

IV. Separate Image

The background image is created by removing the region of the moving objects, and is input to the localization SVO. The moving object images are created by removing the background, and they are input to the SVOs, which estimate the moving objects.

V. Results

We conducted several simulations for the evaluation of the proposed method. First, we developed a simulation environment by using Gazebo [4] and ROS (robot operating system) [5]. The overview of the simulation is shown in Fig. 3. In this simulation, there is a drone equipped with a monocular camera on the bottom, and a Kobuki robot moving around the floor. The fluctuation of the equipment operation and the measurement noise are also reproduced. Captured image by the virtual camera on the drone is processed by the proposed SVO system, and the localization, position estimation of the Kobuki robot, and environmental map generation are executed. In this experiment, the drone was hovering at a fixed position in space, and the Kobuki traveled in a circle. The time evolution of the estimation error is shown in Fig. 4. From the results, the maximum tracking error was found to be 0.06 [m] on the z axis, showing the accuracy of the proposed method to be sufficiently high. Furthermore, map generation using the background image was executed and was considered highly accurate.

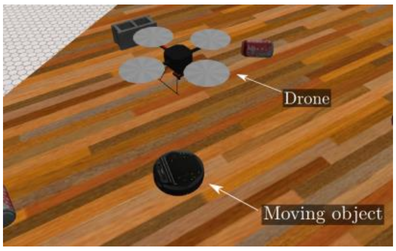

Fig. 3. Physical simulation of a drone equipped SVO in Gazebo.

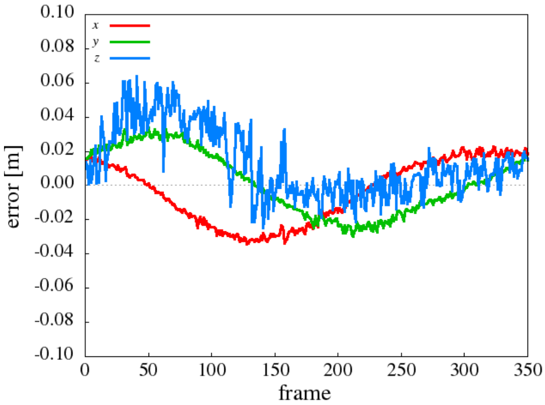

Fig. 4. Time evolution of the tracking error of the Kobuki robot.

VI. Conclusion

In this paper, we proposed an expansion method of SVO for high-speed drone use. By using multiple SVOs in parallel, the positions of multiple moving objects in an image can be measured by a monocular camera. In addition, by the final integration of the SVO outputs, the exact map and changes in the environment can be measured.

References

[1] G. Klein and D. Murray, "Parallel Tracking and Mapping for Small AR Workspaces," Proc. Sixth IEEE and ACM International Symposium on Mixed and Augmented Reality. 2007.

[2] C. Forster, M. Pizzoli, and D. Scaramuzza, "SVO: Fast Semi-Direct Monocular Visual Odometry," IEEE International Conference on Robotics and Automation (ICRA), 2014.

[3] Bruce D. Lucas and Takeo Kanade, "An Iterative Image Registration Technique with an Application to Stereo Vision," Proc. Image Understanding Workshop. pp. 121-130, April 1981.

[4] N. Koenig and A. Howard. "Design and use paradigms for Gazebo, an open-source multi-robot simulator." In Intelligent Robots and Systems, 2004. (IROS 2004). Proceedings. 2004.IEEE/RSJ International Conference on, Vol. 3, pp. 2149–2154 vol.3, 2004.

[5] Morgan Quigley and Ken Conley and Brian P. Gerkey and Josh Faust and Tully Foote and Jeremy Leibs and Rob Wheeler and Andrew Y. Ng, "ROS: an open-source Robot Operating System." ICRA Workshop on Open Source Software, 2009.