A Robust Visual SLAM System using Multiple Monocular Cameras

[1] Kuan-Ting Yu, [2] Feng-Chih Liu, [1] Shih-Huan Tseng, [1,2] Li-Chen Fu, and [3] Yu-Kuen Tsai

[1] Dept. of Computer Science and Information Engineering, National Taiwan University,
[2] Dept. of Electrical Engineering, National Taiwan University
[3] Mechanical and Systems Research Laboratories, Industrial Technology Research Institute
E-mail: r99922070@ntu.edu.tw

***Abstract*—SLAM using monocular cameras is a promising solution because of their low cost and ability to capture accurate bearing information. However, this problem is challenging because we need to infer depth information from bearing-only cues in a sequence of images through triangulation and meanwhile consider uncertainty. In our previous work, the results of the SLAM system using one monocular camera were not stable and accurate enough for real application due to limited field of view (FOV) of a single camera. Therefore, we propose a SLAM system that combines information from multiple monocular cameras to achieve a wider FOV in order to improve the stability of our system. Our SLAM system bases on a probabilistic framework of Extended Kalman filter (EKF), which fuses odometry information and bearing-only features from multiple cameras. Also, an extrinsic calibration process is proposed to accurately estimate the relative transformation between cameras and a robot. Besides, our approach is implemented efficiently by taking advantages of parallel processing techniques using CUDA. We conduct the experiments on the ITRI Ubot platform. Results of our proposed multi-monocular SLAM and localization using the constructed map are evaluated. We achieve mean error of 18 cm on two datasets without artificial landmarks. Finally, we compare the result from one camera with that from multiple cameras and discuss the effects of various parameters on the overall SLAM performance.**

***Keywords:* monocular SLAM, visual SLAM, Bi-cam SLAM, bearing-only SLAM.**

I. INTRODUCTION

Autonomy is the top issue in building service robots. One basic skill of robot is simultaneous localization and mapping (SLAM), on which mobile robots rely for navigation. A mobile robot must find its own location and possible routes to the destinations. Various sensors have been utilized for SLAM on mobile robots, such as laser range finders, sonars, vision sensors, etc. Localization techniques using laser range finders [1] have been extensively studied. To produce robots with more affordable price, researches propose to use low-cost vision sensors like cameras [2-11]. However, visual SLAM remains a challenge to apply in real-world environments in that the depth information cannot be measured directly and sensor noise.

Three categories of vision sensors used in visual SLAM include stereo cameras, omnidirectional cameras, and monocular cameras. Stereo cameras are often used for visual SLAM problem [5, 8]. They estimate depth information by calculating the disparity between pairs of images. On the other hand, omnidirectional camera has the advantage of wide field of view [6] but the distorted image reduce the matching capability of features. In recent studies considering the cost and popularity, the monocular camera is a promising direction for affordable service robots. However, the depth information of landmarks has to be estimated from sequence of images.

In this work, we propose a system that applies multiple monocular cameras with odometry information. The system integrates visual landmarks using Extended Kalman Filter (EKF) framework to achieve Visual SLAM on mobile robots. The benefits of light weight and low cost of monocular cameras are helpful for extensive robot applications.

From the viewpoint of computation speed, CUDA provides an effective framework to utilize GPU to speed up independent processes. Therefore, we apply it to speed up computation-intensive components in order to achieve real-time process.

A. Related Work

In the aspect of probabilistic framework, vSLAM can be classified into two categories, EKF-based and particle-filter-based methods. Here we only focus on EKF-based approaches. Among them MonoSLAM [3] is the most popular method. The authors apply a motion model for camera's movement prediction and propose a solution for monocular camera to initialize a feature without delay. Still, MonoSLAM requires high frame rate and slow movement to track features successfully. Also, a pantoscopic lens is needed to track the features in the view for a longer time. Miro and Dissanayke [9] use stereo camera in their proposed EKF-based vSLAM.

Feature extraction for vSLAM must overcome the issues of illumination variation, scale difference, and viewpoint change. Scale-invariant feature transform (SIFT) [12] meets the above three conditions. SIFT is currently widely used because of its capability in feature detection and description. SIFT selects feature points by using Difference of Gaussians. The selected feature points are local minima in size and space. Histogram of oriented gradient is used to describe these feature points. Speed-up robust feature (SURF) [13] has been applied on vSLAM in recent years. SURF adopts integral image method to approximate Gaussian second order partial derivation, which reduces the time to detect feature

This work was supported by the Industrial Technology Research Institute (ITRI) of the Republic of China under Grant A353C40000B1-3.

points. Since SURF sums up the absolute value of block gradients rather than construct HOGs, it computes faster than SIFT.

Solà *et al.* [11] claim that using multiple cameras in vSLAM could improve the efficiency of EKF SLAM. The method is called Bi-camera SLAM. It combines the advantages of both monocular and stereo vision. The landmark is initialized by (1) using the triangulation of stereo vision when the feature is close to the robot and (2) using angular information of the monocular camera when the feature is far from the robot.

B. System Overview

The main considerations in this work are: vSLAM stability of using monocular camera, computation time on feature extraction, matching, and EKF iterations. In this work, we apply multiple monocular cameras to achieve SLAM on a mobile robot. SIFT features and EKF-based approaches are adopted to fuse feature information from multiple cameras.

The structure of this paper is as follows. In section II we first introduce the procedure of using multiple cameras in vSLAM. Then, we describe feature extraction and landmark initialization. In section III, we illustrate how to apply CUDA to accelerate vSLAM procedure. In section IV, the experiments are presented to verify the proposed method.

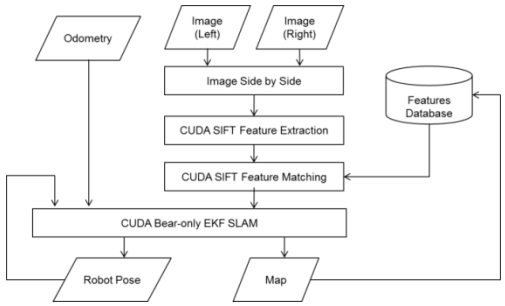


Fig. 1. Architecture of vSLAM system with multiple cameras.

II. USING MULTIPLE CAMERAS IN SLAM

In most of the SLAM process, the camera on the mobile robot is facing towards walking direction. However, this configuration does not good for landmark initialization. Landmark initialization depends on the variation of the viewing angle. According to experiments, landmarks located near the center of an image are not initialized when the robot walks straight. The reason is its parallax is limited between consecutive frames. The landmarks located near the boundary of an image are more possible to be initialized, but they would be soon out of sight as the robot moves. The inherent problem of a perspective camera is the limited field of view. Figure 2 and 3 explain the limited usage of the facing-forward camera configuration in vSLAM.

In this work, multiple cameras are placed on the mobile robot to enlarge the field of view in order to prolong the tracking time of features.

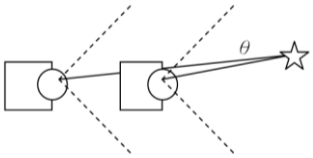


Fig. 2. A landmark appears near the center of the image.

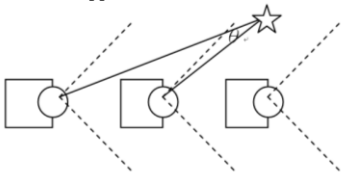


Fig. 3. A landmark appears at the boundary of an image.

A. EKF vSLAM

The key issues in EKF-based vSLAM are highlighted as follow. First the algorithm of visual feature extraction must acquire robust features that can serve as candidate targets. Second, uncertainty occurs during the transformation between the coordinates of a planar image, camera, and global frame, which leads to the accumulation of estimation errors. Lastly for the step of target initialization, the depth of a target is represented by a Gaussian model.

Figure 4 shows four stages of using a monocular camera in EKF-based vSLAM. Suppose the distance and orientation of a feature are represented by the mean and variance of the Gaussian model in the initialization stage, the subsequent prediction is similar to that using a laser range finder. Next, the robot observes the same target in another frame by data association. From the information of robot odometry and observation, the robot location is repeatedly updated.

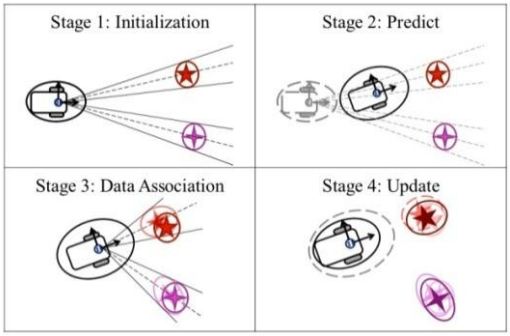


Fig. 4. Four steps of vSLAM using multiple cameras.

B. Camera Model

Comparing to using a laser range finder in SLAM, using monocular cameras needs to conduct triangulation to estimate the distance information of a target feature. Each image captured from cameras is attached with the estimated position information provided by odometer.

$$s\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \underbrace{\begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}}_{\mathbf{A}} \underbrace{\begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix}}_{[\mathbf{R}\,|\,t]} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (1)$$

In equation 1, [*X Y Z*] is the coordinate of the feature in 3D space. [*u*,*v*] is the coordinate of the feature after

being projected on the image. **A** is called the matrix of intrinsic parameters. [C_x, C_y] is the principal point. [F_x, F_y] is the focal length of the camera. [**R** | t] contains translational parameters, which is called the matrix of extrinsic parameters.

C. Calibration of Multiple Cameras

The camera calibration involves obtaining intrinsic and extrinsic parameters. Intrinsic parameters contain focal length, image center, and distortion parameter. They can be estimated by tools such as [14, 15]. Twenty photos of calibration boards in total were captured, including all possible positions and rotations of the calibration board and one picture of the calibration board covering the whole image (see Fig. 5).

On the other hand, the extrinsic parameters of the camera, which represent the relationship between the camera and the robot, are acquired by our proposed method. The steps are described as follows.

1) Align the center of the robot to a crossing point on the ground and orient the robot straight toward the wall (see Fig. 6).
2) Input the captured photos and calibration pictures to the calibration tool to compute the relationship $\mathbf{T}_{bc}$ between the calibration board and camera coordinate. $\mathbf{T}_{bc}$ includes the information of rotation and translation.
3) Obtain the relationship $\mathbf{T}_{br}$ between the calibration board and the center of the robot.
4) According to the equation $\mathbf{T}_{br}=\mathbf{T}_{bc}*\mathbf{T}_{cr}$, the relationship $\mathbf{T}_{cr}$ between the camera and the robot can be obtained.

The experimental result shows the accuracy of the camera calibration parameters affects the result of vSLAM considerably. Therefore, a larger calibration board is made for the experiment. The size of the calibration board is 6×9 squares and the size of each square is 108×108 mm.

D. Feature Extraction

In our system, we extract SIFT feature to all captured images. Each feature extracted is indexed and kept in a temporary database. The feature is compared to the feature database by Euclidean distance. If the observed feature is matched to any features from previous frames, the association information is recorded for the following SLAM procedure. Otherwise, the feature is added to the database.

The accuracy of feature matching is especially important for EKF vSLAM since the architecture of EKF does not deal with matching error. SURF feature was chosen for its speed in extraction, which is 3 times the speed of SIFT. However, SURF feature matching is inaccurate compared to SIFT. Also, FLANN [16] was adopted for feature matching. The method applies kd-tree approximation in searching but the optimal solution of the searching result is not guaranteed.

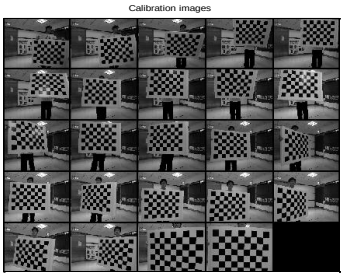


Fig. 5. The checkerboard images used for camera calibration.

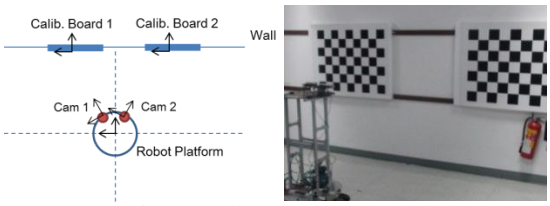


Fig. 6. The configuration for acquiring the extrinsic parameters of a camera. *Left*: the vertical view. *Right:* snapshot of the setup.

To improve the accuracy of our feature extraction system and to maintain appropriate computation speed, we apply CUDA-based SIFT implementation and complete search method for feature matching.

E. Landmark Initialization

After a feature has been observed for several times and forms a well-conditioned pair [17], the collected information is enough for the robot to estimate the depth of the feature. Then, the feature is initialized and becomes a landmark. This process occupies much of the time calculating well-conditioned pairs by sampling. The algorithm is suitable for being speeded up by parallelization. Therefore, GPU can be applied to accelerate the computation at this stage.

III. PARALLEL PROCESSING USING CUDA

Nvidia provides an easy way for GPU program development and its CUDA library has been widely applied on parallel computing in many fields. The major differences between GPU and CPU lie in the efficiency of graphical computation and the memory bandwidth of GPU is much bigger than that of CPU. In this respect, GPU serves as a processor for parallel data. With this framework, programmers shift from using low-level development environments such as OpenGL or DirectX to using high-level environments such as CUDA or OpenCL.

CUDA program is executed in multiple stages. The program run on CPU is called host; the program run on GPU is called device. The stage of parallel data is implemented on device and is skipped when the program run on host. Therefore, a complete CUDA program consists of the programs for host and device.

Parallelization of vSLAM Algorithm

Two segments of the vSLAM Algorithm are the most time-consuming: feature extraction and landmark initialization. These two segments include procedures that

will be executed several times sequentially. Since each execution is independent of each other, the program can be speeded up by applying CUDA's parallel processing.

SIFTGPU [18] is utilized for feature extraction, where hundreds of cores contribute to feature extraction and matching. As for landmark initialization, there is a process of calculating KL distance by sampling 1000 times.

Overall, the following sub-programs are parallelized:

1) *Coordinate transformation of sampling results*
2) *Matrix multiplication of* $\mathbf{HCH}^T$
3) Inverse-matrix computation
4) n-dimension Gaussian sampling
5) Gaussian distribution computation

In general, speed-up of ten times for each sub-program can be achieved if it is properly parallelized. However, exchanging data between GPU memory and host memory should be treated carefully so that frequent and massive data exchange does not hinder the benefit of parallelization.

IV. Experimental Results

A. UBot and Experimental Environment

We conduct experiments of our multi-cam system on both Pioneer 3DX and ITRI UBot. Two Logitech V-UBH44 webcams mounted on an adjustable aluminum rack are used to capture images. The laser range finder on our platform is used to acquire the ground truth. There are two experimental setups: Environment 1 is a 7×5 m^2 field and Environment 2 is a 8×4 m^2 field (see Fig. 7). In Environment 1, we do not decorate the environment with any artificial landmarks, but in environment 2 we do post some posters to increase the feature number. The computational platform has Intel Core-i7 2600K CPU, 16GB memory and Nvidia GTX 560.

Fig. 7. *Top:* Environment 1 and Environment 2. *Bottom:* Pioneer 3DX and ITRI UBot platform.

B. CUDA-Accelerated Feature Extraction and Matching

From preliminary experimental results, the speed of SURF feature extraction is 3 times the speed of SIFT. However, the accuracy of feature matching of SURF is not as good as the accuracy of SIFT. To improve the accuracy and speed, we adopt the SIFT program in CUDA, which accelerates the procedure by parallelization on GPU. On our hardware platform, the speed of SIFT extraction is 1.5 times faster than SURF on CPU. On the other hand, complete searching requires more time than using FLANN but reduces matching errors greatly (see Table I, II and Fig. 9).

Since SIFT or SURF is not robust to the change of the viewpoint (only 5° ~ 15° tolerance), incorrect feature matching occurs. Thus, extracting plenty of feature points is necessary to maintain the quality of visual SLAM. However, too many features also jeopardize the system efficiency. In our experiment, the number of feature extraction is set to 400. The following shows the experimental result with 1280*480 images (see Fig. 8).

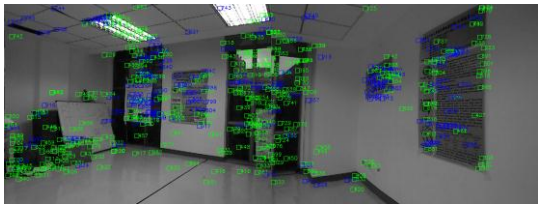


Fig. 8. Rectangles represent the extracted SIFT features. The left part of the picture is captured by the camera facing left and the right part is by the camera facing right. Blue: first detected features. Green: features detected in multiple frames.

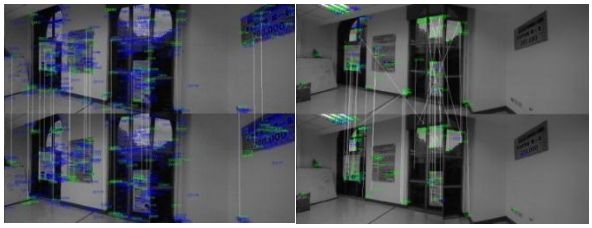


Fig. 9. The result of feature matching between two consecutive frames. *Left:* complete search with SIFT. *Right:* FLANN with SURF.

TABLE I
COMPARISON ON FEATURE EXTRACTION METHODS

Approach	Extraction Time	Average localization error (use FLANN for feature matching)
SURF w/ CPU	365 ms	23.98 cm
SIFT w/ GPU	259 ms	22.17 cm

TABLE II
COMPARISON ON FEATURE MATCHING METHODS

Approach	Matching Time	Average localization error (use SIFT for feature extraction)
FLANN	1432 ms	22.17 cm
Complete search by GPU	2097 ms	22.57 cm

C. Comparison of Using Multiple Cameras and One Monocular Camera

Figure 10 compares localization errors of using multiple cameras and using the monocular camera. The result shows that using multiple cameras can improve the accuracy in robot localization. The localization error of using the monocular camera is 43.6cm in average; the localization error of using multiple cameras is 27.9cm in average. We found that adequate viewing angle is an important factor of the robustness of the SLAM system because it facilitates feature tracking and landmark initialization. (see Fig. 10, 11).

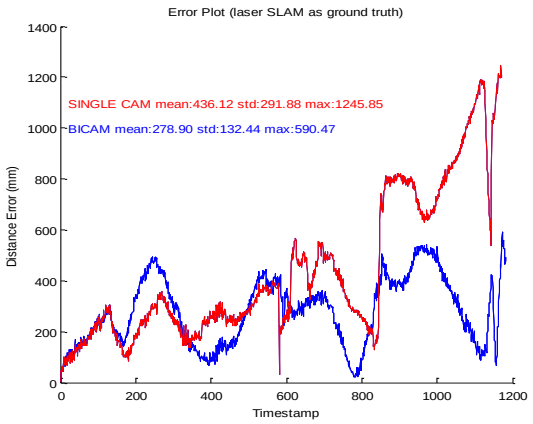


Fig. 10. Comparison on the error of localization. *Blue:* Two cameras, *Red:* single camera.

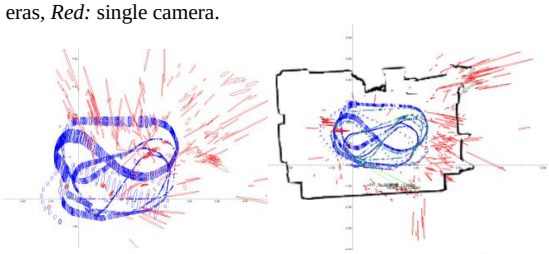


Fig. 11. The resulting maps built by single camera (left) and two cameras (right). *Black:* the map built by laser SLAM. Using two cameras obtains better result than using one camera.

D. Refining Extrinsic Parameters of the Camera

In our previous work [19], the camera was set on top of the robot without significant displacement from the robot center. As for the bi-cam system, the displacements of the cameras are non-negligible. The following experiment makes comparisons between the system considering the displacement of the cameras and the system ignoring this issue. For the cameras without extrinsic calibration, the initialized visual landmarks are mostly not aligned with the ground truth. As Fig. 12 shows, after the camera calibration of extrinsic parameters, the localization error reduces from 27.9cm to 23.1cm in average after refinement. The result shows that by adjusting the extrinsic parameters more accurately, the localization error will reduce significantly and the map built will also improve. The quality of the constructed vSLAM map is comparable to the quality of laser-built map (See Fig. 12, 13, 14).

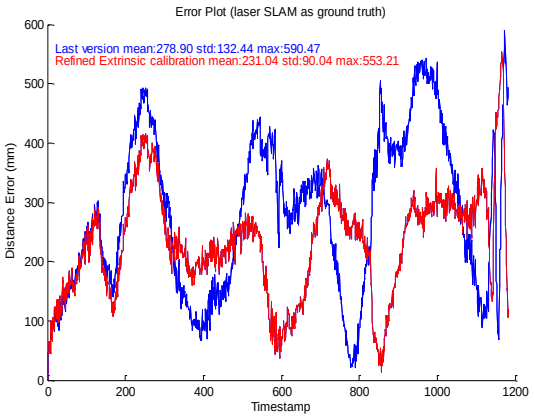


Fig. 12. Comparison on the error of localization before (in red) and after (in blue) the refinement of camera extrinsic parameter.

F. vSLAM on UBot with our Proposed Calibration

We run our whole system on the UBot mobile platform developed by ITRI. Also, the calibration accuracy of the camera is improved. The system is accelerated with the help of CUDA. The result is a vSLAM system with higher localization accuracy and computation speed. In experimental environment 1 (see Fig. 13), the average localization error is 15.2cm, and average computation time is 3.4 sec/step (1 step is approximately 10 cm). In experimental environment 2 (see Fig. 14), the average localization error is 21.5cm, and average computation time is 6.43 sec/step. A more challenging dataset is presented in Fig. 16.

G. Local Localization

Although our simultaneous localization and mapping utilizes the acceleration of GPU, there is a gap to real-time processing. The following experiment inspects the issue of real-time localization after the map is generated and the functionality of mapping is turned off. The robot circled the environment 2 four times. Simultaneous localization and mapping was run for the first two loops, and only localization was run for the last two loops. The experimental result shows the computation time remains stably around 1.67/step after the mapping functionality was turned off (see Fig. 15(a) red line). Also, the localization error stays at the same level (see Fig. 15(b)).

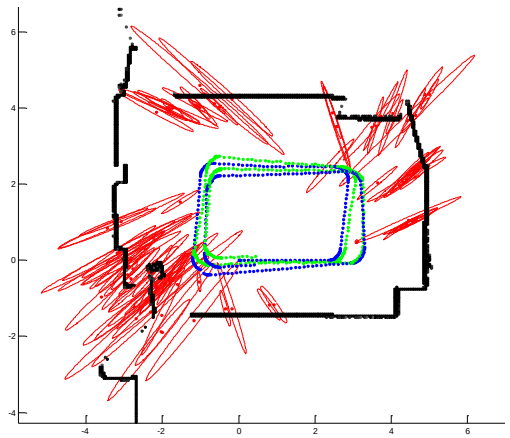


Fig. 13. Comparison on the resulting maps of vSLAM (blue: vSLAM trajectory, red: vSLAM map) and SLAM using laser (green: laser trajectory, black: laser map).

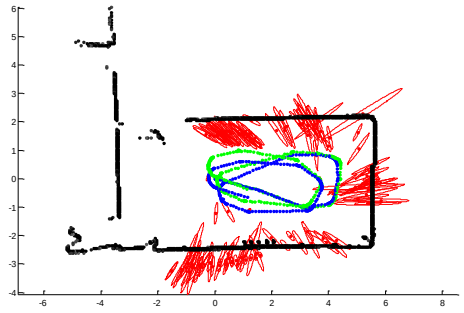
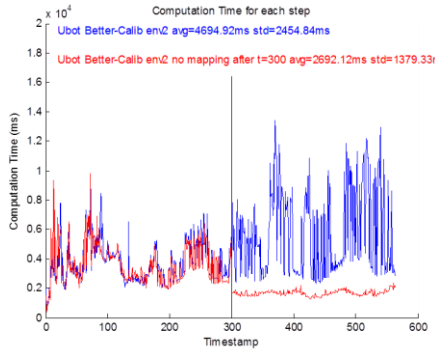
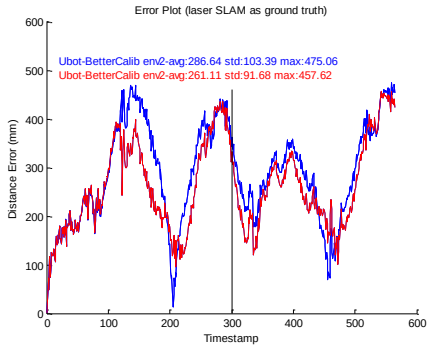


Fig. 14. Comparison on the resulting maps of vSLAM (blue: vSLAM trajectory, red: vSLAM map) and SLAM using laser (green: laser trajectory, black: laser map).



(a)



(b)

Fig. 15. Comparison on the (a) computation time and (b) localization error between SLAM processes with map building on and off (blue: map building was on, red: map building was off after two loops (T=300)). The computation is accelerated by turning the mapping off, while the localization error stays the same.

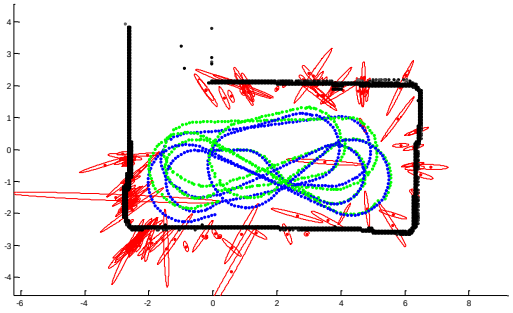


Fig. 16. The result of a more challenging dataset in environment 2. Comparison on the resulting maps of our overall vSLAM system (blue: vSLAM trajectory, red: vSLAM map) and SLAM using laser. (green: laser trajectory, black: laser map).

V. Conclusion

This work addresses vSLAM by adopting multiple cameras with EKF-based approach. We tested the accuracy of the proposed system under different experimental conditions. The parallel processing technique of CUDA is successfully applied to speed up feature extraction and landmark initialization. An accurate camera calibration procedure is also presented. The experimental result shows the vSLAM stableness and accuracy can be effectively improved by providing adequate field of view from multiple cameras.

VI. References

[1] W.-J. Kuo, S.-H. Tseng, J.-Y. Yu, and L.-C. Fu, "A hybrid approach to RBPF based SLAM with grid mapping enhanced by line matching," in *IROS 2009*.

[2] T. D. Barfoot, "Online visual motion estimation using FastSLAM with SIFT features," in *IROS 2005*.

[3] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse, "MonoSLAM: Real-Time Single Camera SLAM," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, , vol. 29, pp. 1052-1067, 2007.

[4] E. Eade and T. Drummond, "Scalable Monocular SLAM," in *CVPR 2006*.

[5] P. Elinas, R. Sim, and J. J. Little, "/spl sigma/SLAM: stereo vision SLAM using the Rao-Blackwellised particle filter and a novel mixture proposal distribution," in *ICRA 2006*.

[6] S. Hochdorfer and C. Schlegel, "Bearing-Only SLAM with an Omnicam Robust Selection of SIFT Features for Service Robots," K. Berns and T. Luksch, Eds., ed: Springer Berlin Heidelberg.

[7] N. Karlsson, E. di Bernardo, J. Ostrowski, L. Goncalves, P. Pirjanian, and M. E. Munich, "The vSLAM Algorithm for Robust Localization and Mapping," in *ICRA 2005*.

[8] T. Lemaire, C. Berger, I.-K. Jung, and S. Lacroix, "Vision-Based SLAM: Stereo and Monocular Approaches," *International Journal of Computer Vision*, vol. 74, pp. 343-364, 2007.

[9] J. V. Miro, G. Dissanayake, and Z. Weizhen, "Vision-based SLAM using natural features in indoor environments," in *Proceedings of the 2005 International Conference on Intelligent Sensors, Sensor Networks and Information Processing Conference, 2005.* .

[10] R. Sim, P. Elinas, M. Griffin, A. Shyr, and J. J. Little, "Design and analysis of a framework for real-time vision-based SLAM using Rao-Blackwellised particle filters," in *The 3rd Canadian Conference on Computer and Robot Vision, 2006*.

[11] J. Sola, A. Monin, and M. Devy, "BiCamSLAM: Two times mono is more than stereo," in *2007 IEEE International Conference on Robotics and Automation*.

[12] D. G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *International Journal of Computer Vision*, vol. 60, pp. 91-110, 2004.

[13] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-Up Robust Features (SURF)," *Computer Vision and Image Understanding*, vol. 110, pp. 346-359, 2008.

[14] *Camera Calibration Toolbox for Matlab*. Available: http://www.vision.caltech.edu/bouguetj/calib_doc/

[15] *Mobile Robot Programming Toolkit (MRPT)*. Available: http://www.mrpt.org/

[16] M. Muja and D. G. Lowe, "Fast approximate nearest neighbors with automatic algorithm configuration,"

[17] T. Bailey, "Constrained initialisation for bearing-only SLAM," in *ICRA 2003*.

[18] C. Wu. (2007). *SiftGPU: A GPU implementation of scale invariant feature transform (SIFT)*. Available: http://www.cs.unc.edu/~ccwu/siftgpu/

[19] Kuan-Ting Yu, Feng-Chi Liu, Jia-Yuan Yu, and L.-C. Fu, "An Integrated Feature Selection Strategy For Monocular SLAM," in *CVGIP 2011*.