A Comparative Study of SURF and SIFT Features' Performances in Simultaneous Localization and Mapping

A project submitted in partial fulfilment of the

requirements for the award of the degree

Master of Science

From

University of Arkansas at Little Rock

By

Rubing Guo

Department of Applied Science

March, 2016

Project Instructor: Dr. Cang Ye

Abstract

In this project, a comparison of two popular visual feature descriptors in terms of their performances in robot pose estimation is presented. The first feature is the Scale-Invariant Feature Transform (SIFT) and the second is the Speeded up robust features (SURF). These two feature descriptors are compared in the context of camera egomotion (pose change) estimation by using a number of datasets containing intensity and range images of a 3D time-of-flight camera, SwissRanger SR4000. In this work, the camera egomotion is estimated by using the Visual Range Odometry (VRO) method and the pose is estimated by optimizing the pose graph created by using egomotion estimates.

This project is an important step towards the development of a robot pose estimation system. The performance metrics of the feature descriptors are the egomotion estimation accuracy, the pose estimation accuracy and the computational time.

The SURF-based VRO and the SIFT-based VRO are compared with various datasets collected from the real world navigational scenarios of a robotic navigation aid, called Smart Cane. The experimental result shows that SIFT-based VRO perform much better than the SURF-based VRO in terms of the accuracy and computational time. As a result, the pose estimated by the SIFT-based pose graph is much more accurate than that of the SURF-based pose graph.

Acknowledgements

I would like to thank my supervisor Dr. Cang Ye for continued support and much helpful advice and idea. I feel extremely grateful for the opportunity to spend one and a half year in a state-of-the-art field under his direction. Also, thanks his IASL lab for providing the device and all the essential tools I have used in this project.

I would also like to thank Soonhac Hong for his key advice concerning my project. He took time from his busy working schedule to respond my enquiry in a very helpful and detailed fashion. This advice proved essential to whatever small success this project may have achieved.

Thanks He Zhang for helping me collect data around ETAS and EIT building, who also helped ignite my interest in programming.

Statement of Originality

I, Rubing Guo, declare that this project, submitted as part of the requirements for the award of Master of Science, at the department of applied science, University of Arkansas at Little Rock, is wholly my own work unless otherwise referenced or acknowledged. The document has not been submitted for qualifications or assessment at any other academic institution.

Signature:

Print Name: Rubing Guo

Student ID Number: T00589317

Date:

Table of Contents

1. Introduction

Robot pose estimation is becoming more and more important in various areas, from sport science to medicine to entertainment. In our lab, we are developing a relatively robust robot motion estimation system for a Portable Navigation Device (PND) for the visually impaired. The PND is used by a blind person for wayfinding and obstacle avoidance.

In this project, we compares the two feature extractors, Scale-invariant feature transform (SIFT) [1] and Speeded up robust features (SURF) [2], in terms of their performances, accuracies and computational complexities in Simultaneous Localization and Mapping (SLAM). SIFT was introduced by Lowe(2004) for extracting distinctive invariant features from images that are invariant to scale, rotation and is widely used in image mosaic, object/ face recognition, etc. Since the feature extraction process of SIFT is very slow, Bay and Tuytelaars (2006) proposed SURF, which turns out to be faster than SIFT and works well.

There are also other feature detection methods, such as edge detection, corner detection, etc. Each of these method has its own cons and pros. However, the most important factor is that feature descriptors must be salient and robust to scale change and rotation so that the Visual-Range Odometry (VRO) [3] method can work well with a large frame-to-frame movement. In [4], robot's pose change estimation by using SIFT and SURF are compared in terms of accuracy and computational time, and the results demonstrates that SIFT outperforms SURF. However, it is not clear that how this performance difference impacts the accuracy of pose (in the world coordinate system) estimated by the Graph-SLAM method [5].

In this project, we first compares the SIFT's and SURF's performance in egomotion estimation in terms of accuracy and computational time, we then apply the two feature extractors in Graph-SLAM and compare their accuracies.

2. Literature Review

The development of feature tracking method used in SLAM can be dated back to the work of Moravec (1981) on stereo matching using a corner detector. Some of the earlier works are limited to corners and short range motion tracking. But the approach was later extended to more difficult problems. It is possible to match Harris corners over a large range motion by using a correlation window around each corner to select the likely matches. However, the Harris corner detector is very sensitive to change in image scale. Recently, a lot of work has been done in motion tracking by using affine invariant features, such as SIFT and SURF.

L. Juan and O. Gwun [6] evaluated SIFT's and SURF's stability of affine transformations using KNN (K-Nearest Neighbor) and Random Sample Consensus (RANSAC). They concluded that both of them have a similar repeatability when viewpoint change is small and degrade when viewpoint change becomes larger. S. Hong, et al. [4] also demonstrate that SIFT and SURF have similar performance in terms of accuracy and repeatability in pose change estimation (PCE) error using VRO when measuring Y movement but in all other cases SURF produces insufficient inliers which and larger PCE errors.

The performance of SLAM predominantly relies on the choice of the feature detector and most researchers prefer feature extractors like SIFT and SURF. To estimate robot pose, a typical method is to integrate measurements from inertial sensors and wheel odometry. But the bias drift of an IMU sensor and wheel slips may accrue pose error over time. Another popular approach is Visual Odometry (VO), which employs a stereovision system to estimate the ego-motion by detecting features in a stereo image pair and track them frame-to-frame. Some of the researchers are still using monocular VO based on 2D images from a mobile phone. Monocular VO estimates the camera motions sequentially according to the perceived movements of pixels in the image sequences. Normally, a VO algorithm consists of four steps, camera calibration, feature tracking, RANSAC outlier rejection, and rigid motion estimation. C. Ye and M. Brunch [3] employs SIFT for feature matching and RANSAC for outlier removal, it is demonstrated that SIFT and RANSAC can produce relative accurate and reliable pose estimation, but both are computationally expensive.

Recent years, more and more researchers use multiple-sensors for robot pose estimation. However the method is unsuitable for a PND which has strict size, weight, power and requirements for its onboard sensor. The ideal solution for small robot pose estimation is to use a single sensor. C. Ye and M. Brunch validated that a new class of 3D imaging sensor called the Flash LADAR – the SwissRanger SR4000 is able to provide relative accurate information for the single-sensor-modality approach.

3. Smart Cane and 3D Camera

Our lab is developing a portable navigation device, called the Smart Cane (SC), as shown in Figure 1. It is a computer-vision enhanced white cane that consists of a 3D camera (SwissRanger SR4000), a portable computer, and a white cane. The computer simultaneously processes the 3D camera's range and intensity data to get the PCs between two camera views. These PCs are then used to obtain the device pose in the world coordinate for wayfinding and 3D map construction. The resulting 3D point cloud can be used to detect obstacles.

Figure 1. The Smart Cane: a computer vision enhanced white cane

The SR4000 is a 3D Time - Of - Flight (TOF) camera. The camera illuminates the scene with modulated infrared light and focuses the reflected light onto the 3D camera's Focal Plane Array (FPA) to produce both range and intensity data simultaneously in its field of view. Each pixel on the FPA measures the TOF to determine the distance based on the phase shift. Figure 2 describes the details about the 3D camera. Table 1 summarizes the key specifications of the 3D camera. The camera is small in size and it output a frame of 3D data points (176 * 144 = 25,344 points) at a frame rate up to 50 Hz.

Figure 2. SwissRanger SR4000

Table 1 Specifications of the SR4000 (h: horizontal, v: vertical)

Pixel Array Size	176(h) × 144(v)
Field of View	43º(h) × 34 º(v)
Non-ambiguity Range	5 m
Angular Resolution	0.24 º
Absolute Accuracy	±10 mm (typical)
Dimension	65 × 65 × 68 mm^3

4. Proposed Approach

4.1 Introduction

VRO [3] is a method to compute the pose change (PC) between two consecutive camera views by using both visual and range data. This method extracts visual features from consecutive intensity images obtained from the SR4000, and use the feature's range data as the 3D data points for pose change estimation (PCE). Two mostly used feature descriptors are employed to estimate the PC using VRO, their performances are presented separately in the following.

SURF is certified to be more efficient in feature detecting and matching, it has been many researchers favor in robot pose change estimation. In [4], SURF [1] and SIFT [2] are compared using SLAM, their performances on camera PCE are analyzed and large experimental results show that SIFT is more accurate and more efficient than SURF when estimating the camera egomotion.

However, no specific analysis of SIFT's and SURF's performance on PCE accuracy and computational analysis in terms of VRO result optimized by Graph-SLAM [5]. In this project, the PCs from VRO are substantially processed by PGO to optimize the robot pose and reduce errors, also robot's trajectory and 3D map are shown to justify accuracies of two feature descriptors.

4.2 Visual-Range Odometry

The visual-range odometry method employs feature detector to match features in current image frame to the next frame, whose features are described by feature descriptor. Since our raw data's 3D coordinates are known, the process of feature detecting and matching results in two matched 3D datasets, $\{p_i\}$ and $\{q_i\}$ for $i = 1, \dots, N$.

By minimizing the error residual $e^2 = \sum_{i=1}^{i=N} \|p - Rq - T\|^2$, the rotation and translation between two matched images can be calculated. This least-square data fitting problem can be solved by the Singular Value Decomposition (SVD) method [7]. A RANSAC process is

implemented after feature matching to remove the mismatched points (outliers) because the feature descriptors we are using may cause incorrect correspondences.

The VRO method is summarized to the following steps:

1. Detect features and match them in two consecutive intensity images, find the matched features based on the feature descriptors. This process results in two matched 3D datasets $\{p_i\}$ and $\{q_i\}$. Repeat steps 2&3 for $k = 1, \dots, K$, times till a termination criteria is met.

2. Randomly select four associated point-pairs from the two data sets to form $\{p_m\}$ and $\{q_m\}$ for m = 1…4. Then the rotation and translation matrices (R_k and T_k) can be calculated for the selected point-pairs $\{p_m\}$ and $\{q_m\}$.

3. Project the entire datasets {qi} onto datasets {pi} by the transformation R_k and T_k from last step, and compute the error $e^2 = \sum_{i=1}^{i=N} \|p - Rq - T\|^2, i = 1, \dots N$ for each point-pair. A threshold τ is used to score the support S_k for this transformation, S_k is incremented once for each $e_i^2 < \tau$.

4. Record the transformation with the highest score, and the corresponding datasets $\{p_i\}$ and $\{q_i\}$ for $j = 1 \dots S_k$, (in which each data-pair past the threshold test in step 3) are selected to compute the maximum likelihood estimate of $\hat{R}$ and $\hat{T}$ (PCs) of the camera by the SVD method. The camera's Euler angles are computed from $\hat{R}$ and its translation is from $\hat{T}$.

The RANSAC is terminated at a fixed K number, which might result in an unnecessarily long computation time. In this work, K is automatically determined by the property of data samples. Given the true inlier ratio ε, the minimum number of repetitions required to ensure, and some level of confidence η, the datasets $\{p_i\}$ and $\{q_i\}$ are outlier-free, the following expression is used

$$K_{min} = \frac{\log(1-\eta)}{\log(1-\varepsilon^m)} \tag{4.1}$$

K_{min} is used as the termination criteria to terminate the RANSAC process. In this project, m=4, η = 0.99 are used. In fact, the true inlier ratio ε is a priori unknown. An estimate on this ratio can be found by using the sample which currently has the largest support. This estimation is

then updated as the RANSAC process proceeds. The total computational time of RANSAC process is determined by the total number of datasets and the value of K_{min}.

4.3 Noise Analysis

According to the previous research on VRO [1], the PCE error of the raw data follows Gaussian distribution, which means noises in both the range and intensity data from the camera are Gaussian noises. In [5], a 3*3 low-pass Gaussian filter is applied to intensity and range images to reduce the noise, result shows that Gaussian filter reduces the overall noise levels of the intensity and range data by 57% and 63% separately. In this project, we apply Gaussian filter to reduce the noise in both intensity and range images.

4.4 Feature extraction and classification methods

4.4.1 Introduction

The process of feature extraction is composed of feature detection and feature description. In feature detection, an algorithm determines the appropriate keypoints that represent the most informative part of the image. While in feature description, a local descriptor is computed for each point. The descriptor possesses the neighborhood information of the keypoint in order to identify the same keypoint across various images.

The performance of VRO predominantly relies on the choice of feature descriptors and most commonly used feature descriptors are SIFT and SURF.

SIFT transforms image data into scale-invariant coordinates relative to local features, which can be used to perform reliable feature matching between different views of an object or scene. The extracted features are invariant to image scale and rotation, and have been shown by Lowe to provide robust matching across a substantial range of affine distortions.

Whereas, SURF is proved to be of outstanding performance with respect to repeatability, distinctiveness, and robustness. It is also computationally efficient. SURF detects feature points by using determinant of the Hessian matrix $H(X, \sigma)$ with its definition shown below

$$H(X, \sigma) = \begin{bmatrix} L_{xx}(X, \sigma) & L_{xy}(X, \sigma) \\ L_{yx}(X, \sigma) & L_{yy}(X, \sigma) \end{bmatrix} \tag{4.2}$$

Where L is the convolution of the Gaussian second order derivation of image at point (x, y) in scale σ. The maxima and minima are used in this classification and the descriptor works by constructing the window around the detected feature points.

4.4.2 Accuracy and repeatability

Extensive experiments [5] with individual movements (roll, pitch, yaw, X or Y) were performed to compare the performances of SIFT-VRO and SURF-VRO. Figure 3 shows the experimental results, from which one can see that SIFT attains better result than SURF in terms of repeatability and accuracy. The only exception is that SURF-VRO has a slightly smaller mean error than SIFT-VRO in Y measurement while SIFT-VRO has a smaller standard deviation. It is concluded that SIFT and SURF have similar performance in measuring Y movement, since SURF-VRO produces sufficient inliers for PCE and brings comparable errors to that of SURF-VRO. While in all other cases, SURF feature extractor produces insufficient inliers and thus causes larger error when compared to SIFT-VRO.

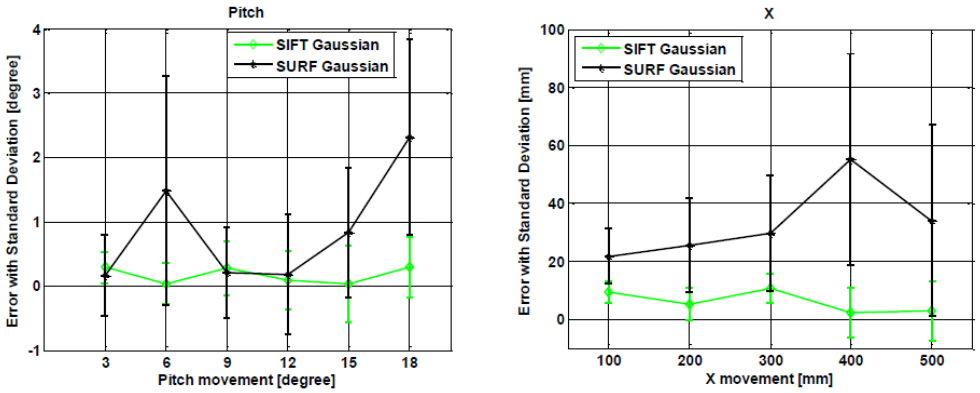

Figure 3. SURF-VRO and SIFT-VRO pose error and STD with slight individual movements

4.4.3 Computational time

The following experiment from [5] depicts the inlier and outlier numbers of matched feature points and the run-time of the two feature extractors when the pitch movement is in the range [3, 18].

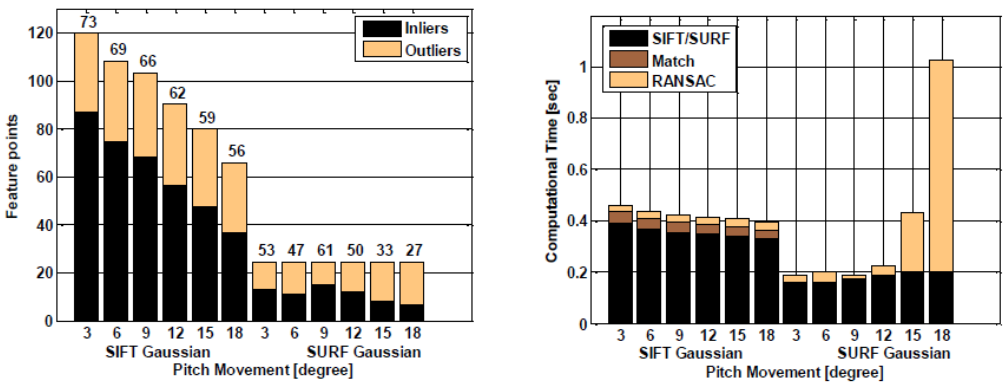

Figure 4. Outliers and Inliers of SURF-VRO and SIFT-VRO

From figure 4, it is obvious that SIFT-VRO has much more inliers than SURF-VRO in each pitch movement. Also, the computational time of SURF is almost 200 ms less than that of SIFT. It is worth mention that when the outlier is increasing, the computational time of SURF is drastically increasing, which can be seen when pitch movement at 18.

4.5 PGO

4.5.1 Introduction

Pose change (PC) is defined between two camera views by VO, and there are a number of features over the two images. Once the features of two images are extracted and tracked, PCs are computed through data association, also we can get camera poses in the world coordinate by dead reckon with PCs.

However, the dead reckon process can incur large pose error during integration process, to reduce this accumulative pose error, the representative method is state estimation and filtering. The state estimation and filtering approach attempts to limit the pose estimation error by a state filter such as an Extended Kalman Filter (EKF). However, our recent work demonstrates that an EKF-based SLAM method degrades its performance and sometimes fails if the camera is in feature sparse environment.

Thus, we use pose graph based method to mitigate the pose drift problem and finally get relatively more accurate result. For simplicity, we call this method VRO-PGO. It constructs a

pose graph with PCs computed by VRO, PCs are optimized through the process of pose graph construction and optimization. The main idea of factor graph structure is to construct a graph, whose node represents a camera pose and an edge between two nodes represents the PC between the two camera views that constraints the connected poses. After the graph is constructed, the crucial problem is to find a configuration of the nodes that is maximally consistent with the measurements. Our final goal is to apply pose graph based SLAM method on PCs we get from VRO so that we can compare the optimized pose estimation result of the two methods, SURF and SIFT.

4.5.2 Pose graph construction

Let $x = (x_1, \dots x_N)^T$ be a vector consisting of nodes $x_1, \dots x_N$, where x_i for $i = 1, \dots, N$ is the camera pose at i. The mean and information matrix of a virtual measurement between nodes i and j are represented by Z_{ij} and Ω_{ij}. The virtual measurement is a transformation that associates the two 3D point sets acquired at i and j (camera pose x_i and x_j). $\hat{z}_{ij}$ is the estimated virtual measurement given x_i and x_j. The measurement error can thus be represented by $e_{ij}(x_i, x_j) = z_{ij} - \hat{z}_{ij}(x_i, x_j)$ and the information matrix Ω_{ij} are used to describe the edge connecting nodes i and j. Figure 5 shows an example of pose graph with 5 nodes.

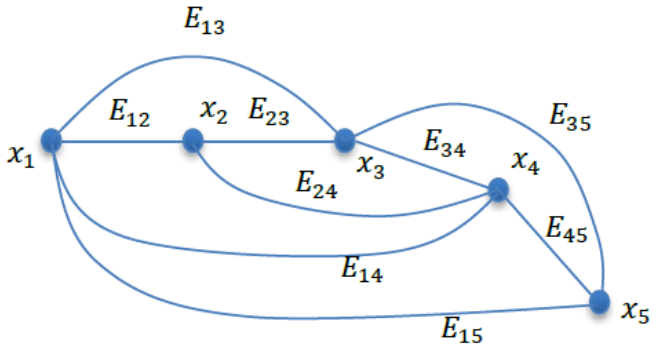

Figure 5. Example of pose graph with five nodes, $E_{ij} = < e_{ij}, \Omega_{ij} >$ represent the edge between nodes i and j.

According to our earlier work [1], the uncertainty of the PCE computed by VRO increases as the movement between the two camera views increases. A large uncertainty value may deteriorate the graph quality and thus the PGO result. In this work, we create an edge between two non-consecutive nodes only if the PC uncertainties are below some thresholds.

4.5.3 Pose optimization

The pose optimization process is to minimize the following cost function of the graph:

$$F(x) = \sum_{ij} e_{ij}{}^T \Omega_{ij} e_{ij} \tag{4.3}$$

Here, i and j represents the index-pair, between which an observation (measurement) z_{ij} exists. As the cost function is a nonlinear one, a nonlinear optimization algorithm should be utilized. If a good initial guess of the pose is known, a numerical solution of the cost function can be obtained by using the Levenberg-Marquardt (LM) algorithm [42]. The LM uses approximation of the error function by its first order Taylor expansion around the current initial guess:

$$e_{ij}\left(\tilde{x}_i + \Delta x_i, \tilde{x}_j + \Delta x_j\right) = e_{ij}(\tilde{x} + \Delta x) \approx e_{ij}(\tilde{x}) + J_{ij}\Delta x \tag{4.4}$$

Here, J_{ij} is the Jacobian of $e_{ij}(x)$ computed at $\tilde{x}$. Therefore,

$$F_{ij}(\tilde{x} + \Delta x) = e_{ij}(\tilde{x} + \Delta x)^T \Omega_{ij}\, e_{ij}(\tilde{x} + \Delta x) \tag{4.5}$$

$$\approx \left(e_{ij} + J_{ij}\Delta x\right)^T \Omega_{ij} \left(e_{ij} + J_{ij}\Delta x\right)$$

$$= a_{ij} + 2b_{ij}\Delta x + \Delta x^T H_{ij}\Delta x$$

Where, $a_{ij} = e_{ij}{}^T \Omega_k e_{ij}$, $a_{ij} = J_{ij}{}^T \Omega_k J_{ij}$. Then Eq. (4.1) may be expressed as,

$$F(\tilde{x} + \Delta x) = \sum_{ij} F_{ij}(\tilde{x} + \Delta x) \tag{4.6}$$

$$\approx \sum_{ij} \left(a_{ij} + 2b_{ij}\Delta x + \Delta x^T H_{ij}\Delta x\right)$$

$$= a + 2b\Delta x + \Delta x^T H \Delta x \ ,$$

where $a_{ij} = \sum_{ij} a_{ij}$, $b_{ij} = \sum_{ij} b_{ij}$, and $H_{ij} = \sum_{ij} H_{ij}$. It can be minimized in terms of Δx by solving the linear system

$$\left(H + \lambda diag(H)\right)\Delta x^* = -b \tag{4.7}$$

Where λ is a damping factor whose value is adjusted at each iteration by the LM algorithm. The linearized solution is then obtained by

$$x^* = \hat{x} + \Delta x^* \tag{4.8}$$

The PGO progress iterates the linearization in Eq. (4.6), solution in Eq. (4.7) and then update step in Eq. (4.8) until an optimal x is found. More details can be reached at [5]. In this work, the GTSAM C++ class library [8] is used for pose graph optimization.

5. Experimental Validation

5.1 Datasets

We acquired 5 datasets around the lobbies and hallways in ETAS building and EIT building at University of Arkansas at Little Rock.

The performance of SURF-SLAM and SIFT-SLAM is compared and evaluated through trajectory of the two proposed methods with 3D datasets derived from SC prototype in the real world environment. We mark the loop starting point on the floor on each data collection.

The SC user holds the device steadily and walks slowly (with an average speed of 0.21 m/s). m. Due to the lack of ground truth in our experiments, we use the position error norm of the endpoint (in percentage of the path-length) to evaluate the performances of the pose estimation methods. In each experiment, the SC holder walks along the looped trajectories and return the SC to the start point position. The path-length is manually measured each loop.

5.2 Comparison of SURF-SLAM and SIFT-SLAM

The accuracy performance of the two SLAM methods are investigated on the five data sets by computing the mean and standard deviation of the error reduction for each methods. The result shows in Table 2. For simplicity, we call SIFT-VRO based graph SLAM as SIFT-SLAM, and call SURF-VRO based graph SLAM as SURF-SLAM.

As we can see from Table 2, the mean error of SURF-SLAM is almost seven times greater than that of SIFT-SLAM, and the mean error of SURF-SLAM is almost seven times greater than that of SIFT-SLAM, which demonstrates that SIFT has much better accuracy consistency on VRO based graph SLAM and thus is a relative better choice of pose estimation for the VRO based graph SLAM.

Table 2. Mean error and STD of SIFT-SLAM and SURF-SLAM

Datasets	Final Position Error[%]	
	SIFT- SLAM	SURF- SLAM
1(16.5m)	3.7	15.6
2(16.5m)	2.7	20
3(22m)	1.3	13.6
4(13.5m)	2.7	16
5(11.8m)	2.2	8.8
Average	2.52	14.8
Standard deviation	0.87293	4.0792

The efficiency performance of the two methods are shown in the Figure 6 and Figure 7. Figure 6 depicts the total computational time of the SIFT-SLAM on a 900 frames dataset on EIT 2nd floor, and Figure 7 illustrates the computational time of the SURF-SLAM on that same dataset.

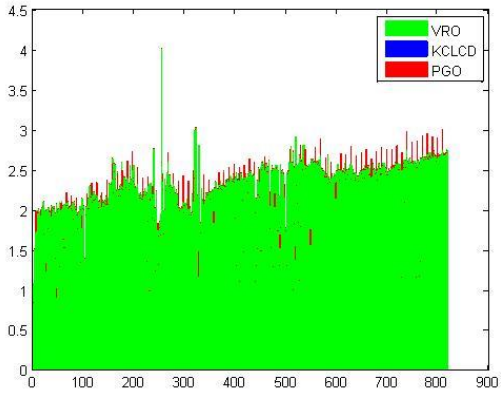

Figure 6. eit_2nd_floor_02_fast_sift

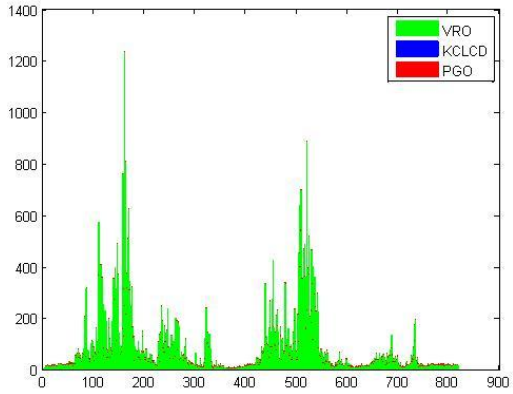

Figure 7. eit_2nd_floor_02_fast_surf

As can be seen, the SURF-SLAM method performs thousands times longer than that of SIFT-SLAM method and SURF-SLAM has an unstable performance in terms of computational time over different images.

A 3D point-cloud is built using the optimized camera poses for each dataset. Figure 8-11 shows two datasets of the 3D maps with camera's trajectories estimated by SIFT-VRO based graph SLAM and SURF-VRO based graph SLAM. Figure 8 and Figure 10 shows the result of

the method based on SIFT, the surroundings of the smart can be clearly seen, however compared with the corresponding same scene shown in Figure 9 and Figure 11, which are calculated by the method based on SURF, the maps contains much more blur and the floor are distorted. The visual results agree with the performance data in Table 2.

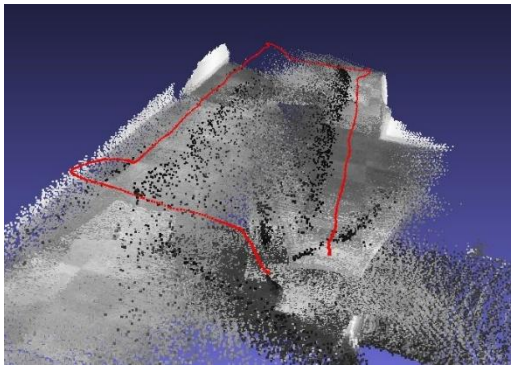

Figure 8. eit_2nd_floor_02_fast_sift

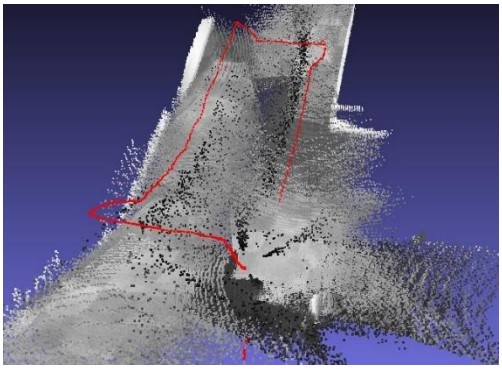

Figure 9. eit_2nd_floor_02_fast_surf

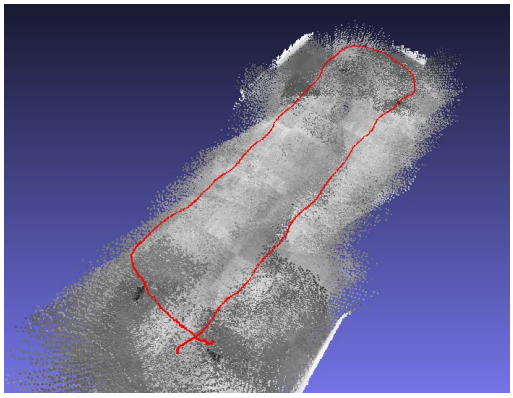

Figure 10. eit_2nd_elevator_normal_sift

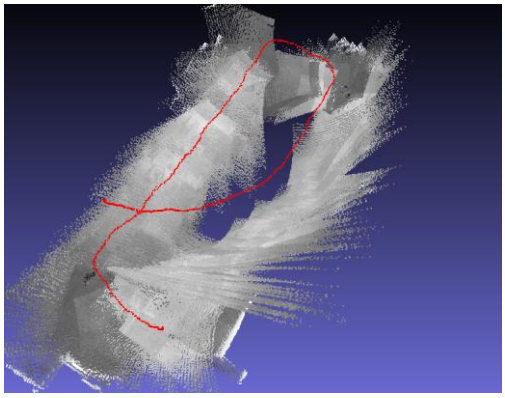

Figure 11. eit_2nd_elevator_normal_surf

5.3 Summary

This section presents performance comparison of pose graph based visual SLAM using both SIFT-VRO and SURF-VRO for 6-DOF pose estimation. We compared the optimized final pose of the camera and the 3D map of the environment. The result demonstrates that: (1) SURF-VRO based graph SLAM consumes much more computational time compared with SIFT-VRO

based graph SLAM. (2) SIFT-VRO based graph SLAM has much better performance consistency than SURF-VRO based graph SLAM in a feature-sparse environment in terms of their mean error and STD.

6. Conclusion

In this project, we have evaluated the performance of SURF-VRO based graph SLAM and the SIFT-VRO based SLAM. Through various experiments, it is concluded that feature extractor plays an important role in PCE, and SIFT can perform much better results than SURF in terms of their pose graph based visual SLAM results. SURF's feature extraction is fast, but the RANSAC process degrades due to the lower inlier ratio, which leads to SURF's much longer computational time compared with SIFT. Extensive experiments also shows that the method based on SURF performs large pose error, which is much worse than SIFT when estimates the camera pose. Thus, SIFT-VRO based graph SLAM has more accuracy and efficiency in pose estimation performance in feature sparse environments compared with SURF-VRO based graph SLAM.

In the future work, SIFT and SURF may be tested in feature rich environment, one more small-sized sensor could be implemented to mitigate the feature descriptors' robustness to scale and rotation.

7. References

[1] H. Bay, T. Tuytelaars, and L. Van Gool, "SURF: Speeded up robust features," *European Conference on Computer Vision*, 2006, pp. 404-417.

[2] D. G. Lowe, "Object recognition from local scale-invariant features," *International Conference on Computer Vision, 1999*, pp. 1150.

[3] C. Ye and M. Bruch, "A Visual Odometry Method Based on the SwissRanger SR4000," In *proc. the Unmanned Systems Technology XII Conference at the SPIE Defense, Security, and Sensing Syposium*, 2010.

[4] S. Hong, C. Ye, M. Bruch and R. Halterman, "Performance Evaluation of a Pose Estimation Method based on the SwissRanger SR4000," *IEEE International Conference on Mechatronics and Automation*, 2012, pp. 499-504.

[5] G. Grisetti, R. Kummerle, C. Stachniss, and W. Burgard, "A Tutorial on Graph-Based SLAM".

[6] L. Juan and O. Gwun, "A Comparison of SIFT, PCA-SIFT and SURF".

[7] K.S. Arun, et al, "Least square fitting of two 3-d point sets," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 9, no. 5, pp. 698-700, 1987.

[8] GTSAM 2.3.0 at http://collab.cc.gatech.edu/borg/gtsam/