공학석사학위논문

물체 추적을 응용한 핸드폰 실내 위치 인식 방법에 대한 연구

Indoor Space Localization with a Mobile Phone by Object Tracking

2016년 8월

서울대학교 대학원

기계항공공학부

유 수 곤

Abstract

Recent researches on indoor localization have achieved a rapid progress, thanks to advances in mobile devices and networks. Related into simultaneous localization and mapping (SLAM) problems, several researchers apply different approaches, such as Wi-Fi, IMU sensors, and ultrasonic sensors. However, more intuitive and accessible system for indoor localization is required in order to achieve high-rate recognition of the current pose. In this paper, we propose the system that has the combination of visual data from a camera and inertial data from IMU sensors in indoor localization. Pre-learning of landmark images and setting up the database is the first part of our proposed localization method. Using TLD tracker and sensor data simultaneously, selected image areas are tracked and approximation of the device location can be extracted. EKF-SLAM, which uses extended Kalman filter to estimate device locations with sensor data, leads to real-scale estimation of the device approximately. From the characteristics of the camera and scale estimation from vision data and sensor data, camera poses are estimated and the landmark locations are matched. Even though abrupt changes of camera movement and angles cause errors on trajectories of the mobile device, camera pose estimation is successfully estimated, and the errors has a range from −0.1m to 0.4m, compared to the ground truth of the movement.

Key Words : Monocular camera, object tracking, indoor localization, extended Kalman filter, and simultaneous localization and mapping.

Student Number : 2014−22492

Table of Contents

List of Figures

List of Tables

Chapter 1. Introduction

In recent years, development of mobile devices and networks brings rapid technical advance in navigation system and augmented reality. Mobile platforms and environments provide a high rate of accessibility to information near the area where the users stand. For instance, GPS system is using in outdoor navigation system and it is easily accessible for mobile phone users. As far as indoor navigation is concerned, there are many applications developed with inertial measurement unit sensors [1,2,3], Wi-Fi [4,5], and ultrasonic sensors[6]. Those simultaneous localization and mapping (SLAM) methods have led to a lot of interests to many researchers and companies because the demand of indoor space navigation rises. The challenge of SLAM is high-rate recognition of the current pose and access of local information with a small amount of calculations.

Monocular SLAM research, which is highly related into Structure from Motion (SfM) [7] in computer vision, has a rapid progress due to hardware development. However, although hardware enhancement provides real-time systems to users, error compensation and calculation time problem are still remaining.

Matching the features on frame by frame can also require high computation power if there are so many features to match in images.

Many methods in [8], [9], [10] are recently proposed in monocular case, while the scaling issues causes inaccurate results from image sequences. In order to reduce errors and to solve a scaling problem, sensors that attached in the device can be used. In [11], visual data and sensor data are combined to calculate the scale factor, and, even in SfM, sensor data in mobile phones are simultaneously extracted at the time when visual frames are available. However, physical positions of the device do not link any information on landmarks nearby. Building 3d point clouds with extracted features can provide the realistic maps and scenes, but detecting landmarks and other features in the images might not be effective.

In our proposed method, combination of object tracking method and sensor data with given landmark images and location dataset is provided for indoor localization on mobile phones. SfM in multiple images is applied, and at the same time pixel positions of detecting landmarks in 2D images provide the moving trajectories of the landmarks, which give estimates of camera poses. Also, the characteristics of a camera reflect how far the landmarks are located from the camera, and the device orientation and location from sensor data are fused with the estimates for resolving the scale factor problem. In this case, one of the SLAM algorithm, called EKF SLAM[12], is used for estimating the location from sensor data.

Chapter 2. Related Works

There are many SLAM algorithms proposed in several researches. Especially, monocular SLAM methods, proposed in [8], [9], and [10], have been researched variously in robotics and computer vision. Those researches use different descriptors to extract features in images, such as ORB, SIFT, and SURF. The features are matched on frame by frame, and the distance between the same features in consecutive frames can be measured, so that camera positions in certain frames are estimated. However, coordinate systems used in monocular SLAM methods do not reflect a real scale. For scaling, the distance between the camera and the features should be known, which is not possible while the images are taken. Therefore, various sensors and techniques are applied to scale the objects or moving trajectories.

In [1], for augmented reality system, several sensors are used to check poses of the device at a certain time. The authors built their own devices which consist of IMU sensors and cameras, and the devices are connected with computers to build augmented reality environment. Even though the algorithm proposed in [1] shows how to extract the device poses in real-time cases, there are issues on accessibility and usability on the devices. Also, for indoor localization, several sensors, including Wi-Fi, ultrasonics, and lasers, can be applied in many researches, but the weight and size of the devices are not suitable for general users.

One of the most interesting SLAM methods presented in [12] and [17] is EKF-SLAM. Extended Kalman filter has been used here for nonlinear system models, such as GPS and navigation. In these researches, inertial measurement unit sensors (IMU) measure accelerations and angular velocities that are considered as the state vector. Using the maximum likelihood algorithm for data association, EKF-SLAM successfully calculates the route of the device where IMU sensors are attached. However, if the posterior includes uncertainty more than noises assumed, EKF-SLAM fails to estimate the location.

To reduce errors in SLAM algorithms, the combination of vision data and other sensor data has been researched in [2] and [3]. The interesting part of those researches is absolute scale estimation.

Compared with IMU sensor data and vision data, real scaling of the moving trajectories can be estimated. Even though the devices used in [2] and [3] are customized by the authors, the chance of the mobile phones using in the real life has shown, because IMU sensors and the camera are built in the mobile phones.

In our proposed system, the disadvantages of monocular SLAM are attempted to resolve. The scaling problems of the SLAM algorithms and the reducing uncertainty from EKF-SLAM are the most interesting factors in our system. Compared to the previous researches, our algorithms consider the physical locations of landmarks and the poses of the mobile devices in a real scale.

Chapter 3. System Overview

In our proposed system, there are three different parts that work for estimating device location and orientation: pre-learning, object tracking, and camera pose estimation, as shown in Figure 1. First, landmarks are selected in pre-learning stage from image sequences with orientations from mobile phones. The landmark locations in world coordinates should be defined, and neighborhoods of one landmark can also be found from landmark information. Orientations from mobile phones indicate camera orientation and possible camera angle, so that we realize which landmark can be detected in the object tracking stage. Camera parameters are also recorded in camera calibration with a chessboard marker.

Second, object detection and tracking is used for tracking pre-learned landmarks and estimating their locations. TLD tracker [13] uses ferns descriptors that can learn an image in a certain selected area easily and draws trajectories of detected landmarks in order to re-detect landmarks after the tracking is lost. The selected landmarks are found in the bounding box form in image sequences. While the landmarks are tracked, camera orientation and acceleration are extracted from mobile devices. They are used for scale compensation and error correction by comparing with camera pose in structure from motion in the next part. The acquired sensor data are used for EKF SLAM[12] which applies extended Kalman filter to estimate device location and a heading angle.

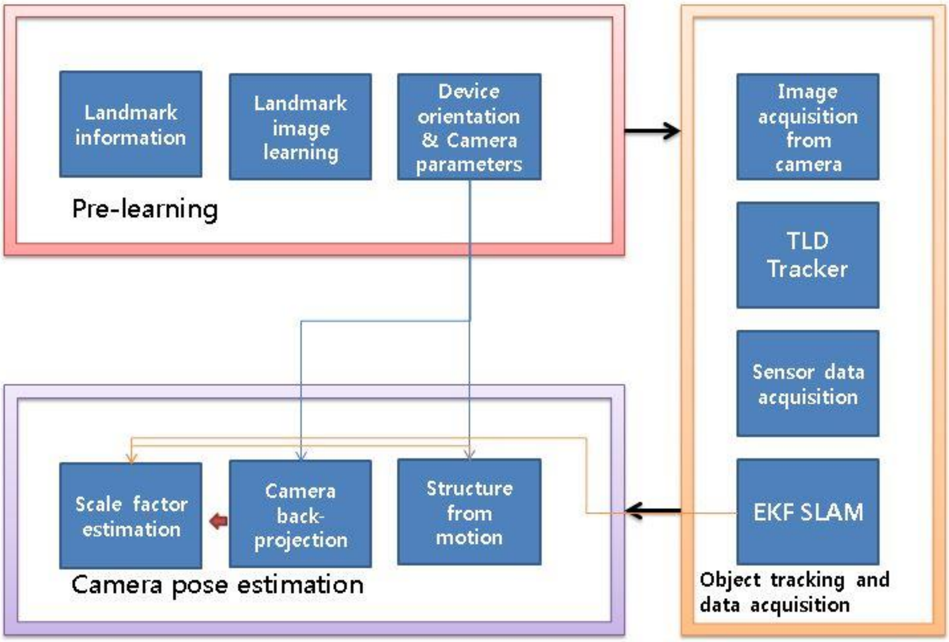


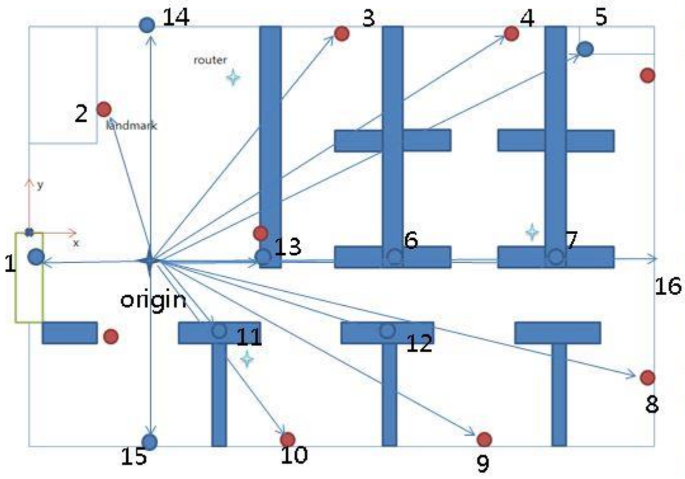
Figure 1. System overview diagram. Pre−learning of landmark images and setting up the database is the first part of our proposed localization method. Using TLD tracker and sensor data simultaneously, approximation of device location can be extracted. From the characteristics of the camera and scale estimation from vision data, camera poses are estimated with the collected data from previous sections.

Finally, camera poses in several images can be estimated with different sources. EKF SLAM brings estimated poses from sensor data, and camera back-projection [14] and structure from motion [15] provide camera poses estimated from the image sequences. Camera orientations and locations information are compared with these two different methods, but sensor data might be more reliable than non-scaled information from structure from motion algorithm, since depth and scale information is lost. When those two results are compared by solving least-square method, the scale factor can be estimated. The distance between the device and the landmarks is measured after applying the scale factor.

Chapter 4. Methods

4.1. Pre-learning

4.1.1. Landmark location dataset



Land mark	X	Y
1	-3.936	0
2	-1.846	2.925
3	4.607	3.491
4	7.707	4.607
5	10.346	3.383
6	5.596	0
7	8.681	0
8	11.172	-2.338
9	7.748	-3.842
10	3.303	-3.842
11	1.837	-1.228
12	5.456	-1.228
13	2.982	0
14	0	3.491
15	0	-3.842
16	11.528	0

Figure 2. Landmark dataset and selection in the 2D map. Landmark images are also stored and learned via TLD tracker.

Before going through our algorithm, we need to set up landmark information and correlation of selected landmarks. For instance, if a 2D map and image sequences from mobile phones are available, the fixed landmarks can be selected in the map and the images, as shown in Figure 2. There are conditions for landmark selection as follows:

1. Landmarks should be fixed in certain positions and remarkably shown, such as logos and signs. For tracking, landmark images should have comparable features in camera angles. While using TLD tracker, this conditions prevent the tracker from losing tracking of the selected landmarks.
2. Landmark position should be determined in the 2D map. The relations among landmarks are defined with respect to the origin chosen. Hence, the 2D location of the landmarks will be used in pose estimation of the device.
3. Physical locations of the landmarks should be easily measured. Laser distance sensors are using in order to measure physical locations and geometrical calculation may define 2D landmark locations.

A single landmark is selected by the conditions above. Each landmark has more than one neighbor landmark and the distance between the landmark and the neighborhood is measured by laser distance sensors. When the device is moving around the landmarks, the landmark information nearby the camera will provide approximate camera location at time. More considerations in landmark data will be followed in camera pose estimation.

4.1.2. Device orientation

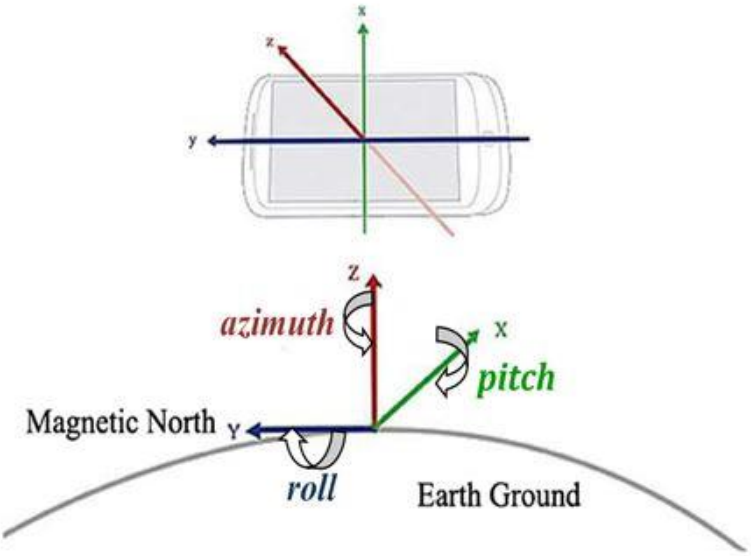


Figure 3. Coordinate systems using in Android mobile phones: device coordinate system(above) and the world coordinate system(below).

In recent mobile phones, several sensors are included, such as an accelerometer, a gyroscope, and a magnetometer. Device orientation can be calculated with those sensors, using filters and combination of extracted data. Android system has own orientation extraction function [19], which is fused with IMU sensors. Depending on the device, different fusion methods can be applied. Android system has functions that extract sensor data and fuse data to calculate device orientation based on the world coordinate system. Using getOrientation() and getRotationMatrix() function, orientation and rotation matrices at a certain time are returned in world coordinate. The coordinate system of calculated orientation is different from device coordinate, as seen in Figure 3. This figure explains the coordinate system of device orientation and the difference between device coordinate and the world coordinate. From the orientation, it is possible to estimate approximate camera angle so that chances of detecting landmarks can be approximated. In samples of the initial image sequence, it is possible to record what landmarks are detected in a certain orientation of the device. Those records are reused in camera pose estimation when the device orientation is close to the recorded orientation. Calculation of camera poses in structure from motion and camera back−projection will be explained in 4.3.1 and 4.3.2.

4.1.3. Camera parameters

For camera back−projection and bundle adjustment in 4.3, camera parameters from camera calibration are required. Intrinsic parameters, including a focal length and a principal point, are the information required since camera pose estimation from image sequences is based on camera parameters and physical landmark information. In this paper, camera calibration with chessboard markers is used to obtain camera intrinsic parameters. Especially, the principal point is important information because it is considered as an image center which camera Z coordinate axis passes through. Using functions in MATLAB, the calibration has been completed with a chessboard pattern. The pattern is a 7 x 9 chessboard which has 30mm for each square. The parameters obtained from camera calibration are saved in a camera parameter variable and will be used in camera pose estimation.

4.2. Object tracking and data acquisition

4.2.1. TLD tracker

Kalal et al.[13] suggests a novel real−time object tracking method using ferns descriptors and Lukas−Kanede tracker. Tracking−Learning−Detection framework and the new learning method called P−N learning [16] are the contribution on their papers. In order to apply the methods in our work, we rewrite the open source codes to display landmark location and information in image frames. The tracker estimates the detected object trajectories among consecutive image frames. P−N learning method identifies false positives and false negatives effectively, so that the errors are compensated while the moving trajectory of the target image is estimated. When the initial target is selected in a single image, the trajectory of the selected area can be estimated in the next image frame. Also, the position of the selection in 2D images is estimated. When the selection occurs, we add the process to link landmark information with the part of the image. For instance, when the landmark #1 is selected in the first frame, landmark information is displayed and saved in the variable to make a comparison with sensor data position. Figure 4 shows how to compare the 2D image features with 3D landmark location in order to calculate the camera motion. Later, camera back−projection process will lead to the method of camera pose estimation with the result in the TLD tracker.

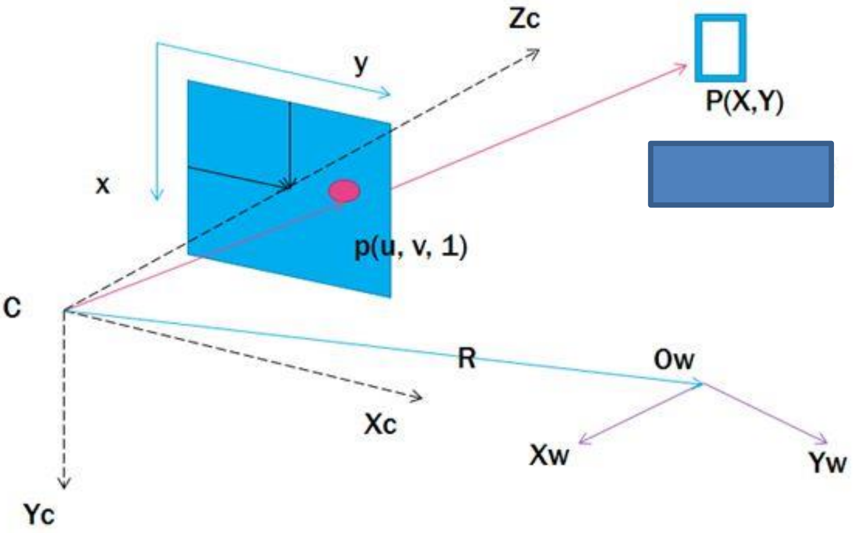


Figure 4. Conversion between 2D features and the 3D points.

4.2.2. Sensor data acquisition

While the device is collecting vision data via a camera, it is possible to extract sensor data as well. However, camera frames and sensor data may not be collected at the same time. To check the motion of the camera at the time when sensor data is extracted, integration of fused sensor data is needed. In order to estimate displacement of camera movement, linear acceleration, which rid of gravity, is extracted by fusing data from a magnetometer and an accelerometer. Because of difference in data acquisition time, sensor data should be rearranged with the same time. However, a frequency of data acquisition is higher in orientation, compared with 5Hz frequency in linear acceleration. Hence, we find orientation data at the same time when linear acceleration is extracted. All sensor data including rotation matrices and orientation of the device can be used after matching extraction time.

After matching extraction time, the system in which only sensor data lead to current pose is applied in our work, called an inertial navigation system (INS). The system performs a double integration of acceleration over time to estimate position. In our test case, the accelerations are measured in the device frame of reference, while the orientation and the rotation matrices are aligned with the world frame of reference. When the device is moving, the dynamics of the sensor system [14] can be defined as follows:

$$x_{t+1} = \varPhi x_t + N(0, Q) \quad (1)$$
$$y_t^i = \mathrm{H} x_t + N(0, R) \quad (2)$$

The state vector x indicates system pose and the derivatives in the world coordinate. The other vector y is the observation vector which contains the outputs of the sensors, such as acceleration and angular velocity. Acceleration and angular velocity are collected from the accelerometer and the gyroscope. $\varPhi$ and H are the matrices for the state model and observation model, respectively. Also, noises from the sensors and measurement are $N(0, Q)$ and $N(0, R)$, where Q and R are the covariance matrices of the noises. While the device is moving, the acceleration is integrated with a decaying velocity model of handheld motion [11] as follows:

$$v_i^{k+1} = v_i^k + \tau \varDelta t \, R(a_B^k - g) \quad (3)$$

v is the velocity and τ stands for inaccuracies of timing and sensor to avoid drift in acceleration data, and R is the rotation matrix with respect to the world coordinate. The last term $(a_B^k - g)$

means linear acceleration, which can be calculated by sensor fusion algorithm. The velocity values and the state vectors from this process will be used in the next step.

4.2.3. EKF SLAM

EKF SLAM [12] is one of the localization and mapping methods that are widely used in robotics. Baileys et al. [17] present this SLAM method which uses extended Kalman filter, and test various environments in common places. With known landmarks and waypoints in world coordinate, the device movement can be estimated, based on the preconditions, such as heading angle and initial velocity. From equation (1) and (2), the state vectors are estimated by extended Kalman filter with a constant velocity model, which considers the device movement is at constant velocity such as walking steps. In our algorithm, a decaying velocity model is applied and the velocity at each time step can be integrated with velocity verlet algorithm. Hence, errors by double integration of acceleration and drift errors are resolved, and the device location errors are reduced.

4.3. Camera pose estimation

4.3.1 Structure from motion in multiple images

Structure from motion [7], called SfM, is one of the techniques that refine the 3D coordinates in the scene geometry, the relative motions, and the camera characteristics when a set of images from different viewpoints are given with many 3D points. In our algorithm, structure from motion in acquired image sequences brings camera poses using only camera vision when scaling and depth estimation are not available. In our test case, landmark features in different images can be extracted, and they are compared to the features in the next consecutive frame. SURF features [18] are extracted and the matching points calculated by point tracker are used for estimating fundamental matrices and epipolar inliers. The two results lead to device orientation and translation motion with respect to the previous device pose. In this part, camera parameters obtained from camera calibration in 4.1.3 are recalled and applied. The calculated camera pose does not apply a scale factor, so compensation with sensor data should be engaged, explained in section 4.3.3.

4.3.2. Camera back−projection

In order to clarify landmark locations, while detecting landmarks by camera vision, camera back−projection estimates camera pose in world coordinate. First, magnetic north is measured by the magnetometer, and rotation matrix between the world coordinate in the 2D map and in the earth map. Landmark locations are reset with respect to the new world coordinate system. From the previous orientation data, rotation matrices are applied to change the coordinates from device to world. Figure 5 also describes how camera back−projection is applied to check landmark location. The pin−hole camera model and prospective projection are assumed in our case. P reflects in the image plane and p(u,v) is shown in the image. In [14], the projection points are related with 3D points by the perspective relations:

$$\mathrm{u} = S_u \lambda \frac{X}{Z} + u_0 \qquad (4)$$

$$\mathrm{v} = S_v \lambda \frac{Y}{Z} + v_0 \qquad (5)$$

, where $(\mathrm{u_0}, \mathrm{v_0})$ is the pixel position of the image center, S_u and λ are indicated as camera focal length, and $\mathrm{S_v}$ is the scale factors associated with the physical dimensions of the pixels. Since we know the given camera parameters from section 4.1.3, X/Z and Y/Z can be estimated. In the landmark database, X and Y positions of landmarks in the world coordinate are already saved, so the combination of landmark database and estimation of X/Z and Y/Z generates approximate Z values. However, the scale factor is the missing information in this case, and this problem will be solved in the next section.

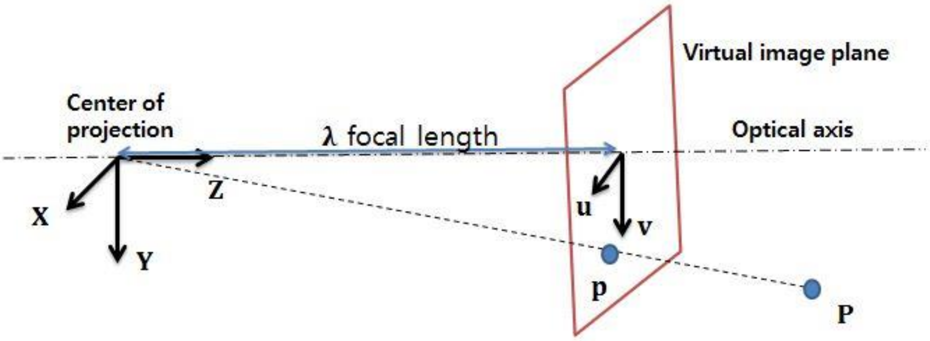


Figure 5. Camera projection and relations between the points in the image plane and the 3D world point.

If the pixel position of the detected landmark is known in the image, the landmark in the next image is tracked using TLD tracker in 4.2.1. The trajectory of the landmark position in the images is presented. For the time when the images are divided, device orientation data are matched.

4.3.3. Scale factor estimation

The collected data from section 4.2.3 and 4.3.1 are compensated for camera pose estimation. The collected data from 4.2.3 and 4.3.1 are compensated for camera pose estimation. We apply the least-square method to compensate for the data from EKF SLAM and structure from motion. In [11], if camera positions are available from both sensors and the camera, the scale factor can be calculated by the least-square method as follows:

$$\arg min_{\lambda} = \sum_{i \in I} \|\vec{x}_i - \lambda \vec{y}_i\|^2 \qquad (6)$$

, where λ is the scale factor, $\vec{\mathrm{x}}_{\mathrm{i}}$ is the displacement hypothesis by the accelerometer, and $\vec{\mathrm{y}}_{\mathrm{i}}$ is the displacement measured by vision. The distance between the consecutive camera locations estimated in SfM is compared to the measurement by sensors. When λ is calculated, the value can be applied to the equation (4) and (5) in 4.3.2 for calculating Z values.

Chapter 5. Results

In our test case, we use an Android smartphone Samsung Galaxy S III (SHV-E210K) with OS 4.4.4 version. Sensor list and camera specification are available in Table 1. Video frame rate with 22 fps is applied and 640 x 360 pixel size video has been taken. In pre-learning process, learning landmark images takes time to check bounding box size and center point. Landmark dataset is based on the measurement by a laser distance module, and x and y positions are arranged in the 2D floor plan. The test place is at 302-209, Seoul National University, Seoul, and landmarks are defined in Figure 2.

Sensor	Resolution	Range	MinDelay	Unit
Accelerometer	0.010	19.613	10000 μs	m/s^2
Gyroscope	0.000	8.727	5000 μs	Rad/s
Magnetometer	0.060	2000.00	10000 μs	μT
RotationVector	0.000	8.727	10000 μs	quat
LinearAcceleration	0.010	8.727	10000 μs	m/s^2

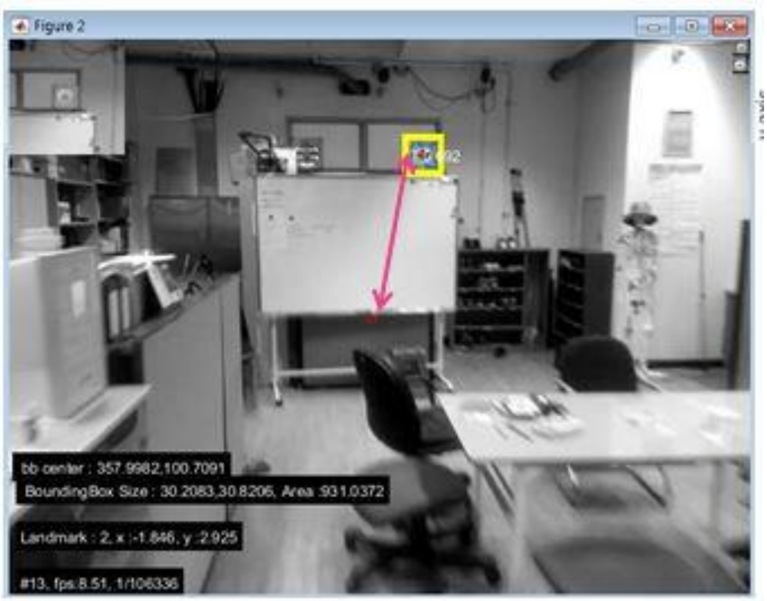
Table 1 Sensor list from the test device (Android mobile phone, GSIII, OS 4.4.4.)

Parameters	Values
Focal length	[545.1967 , 546.1751]
Principal points	[316.3669 , 238.6514]
Radial distortion	[0.2011 , −0.2549]
Tangential distortion	[0, 0]
Skew	0
MeanReprojectionError	0.5593

Table 2. Camera parameters from camera calibration.

In Table 1, the list of sensors indicates the type of the accuracy, data extraction delay, units of the results, and the range. Camera parameters from camera calibration are also determined by the image sequences that include the chessboard in the image. The values are presented in Table 2.

The selected landmark is tracked while sensor data is collected in the device. An image set in image sequences is provided in structure from motion. Camera poses are estimated by structure from motion in 198 images and the trajectory of the motion is extracted in a ply file. Camera poses from SfM are defined in Figure 7. In our test case, about 200 seconds calculation time is spent for sparse reconstruction by SfM. If we build server−client system to reduce extraction time, the time spent for SfM will decrease. The data in Figure 7 and Figure 8 are used in the least−square methods in order to calculate the scale factor. The difference of the shape of the trajectories in Figure 7 and Figure 8 indicates the abrupt change in the heading angle of the device and rotation matrices with respect to magnetic north. EKF SLAM does not reflect loop closure algorithm; that is, when the device is moving in a loop, the trajectory does not connect with the point where the device already passed through.



```
bb center : 357.9982,100.7091
BoundingBox Size : 30.2083,30.8206, Area 931.0372
Landmark : 2, x :-1.846, y :2.925
#13, fps:8.51, 1/106336
```

Figure 6. Landmark Selection from camera using TLD tracker. The linked information of landmarks is shown and the distance between the bounding box and the principal point is represented.

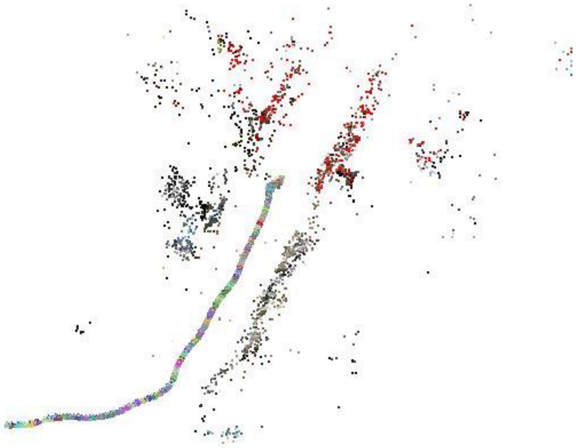


Figure 7. Camera pose estimation from structure from motion (A line segment)

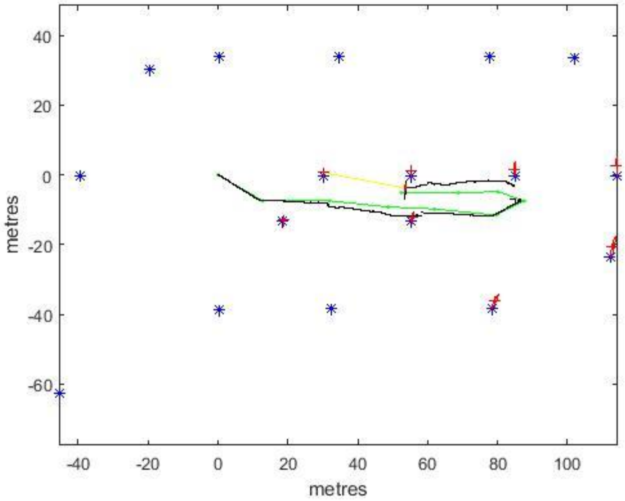


Figure 8. Camera pose estimation from EKF SLAM. When the heading angle changes more than 90 degrees, discontinuity of the estimation (black line) occurs. A green line stands for ground truth.

Figure 9 shows the difference between the estimated poses to which the scale factor is applied and the ground truth of the device movement. In the linear movement the error is not notably increasing before the heading angle is suddenly changed. Sudden changes in heading angles may cause that the difference bounces up and down because the sudden change of the device cause noises in all directions and absence of loop closure algorithm might affect the calculation in EKF SLAM.

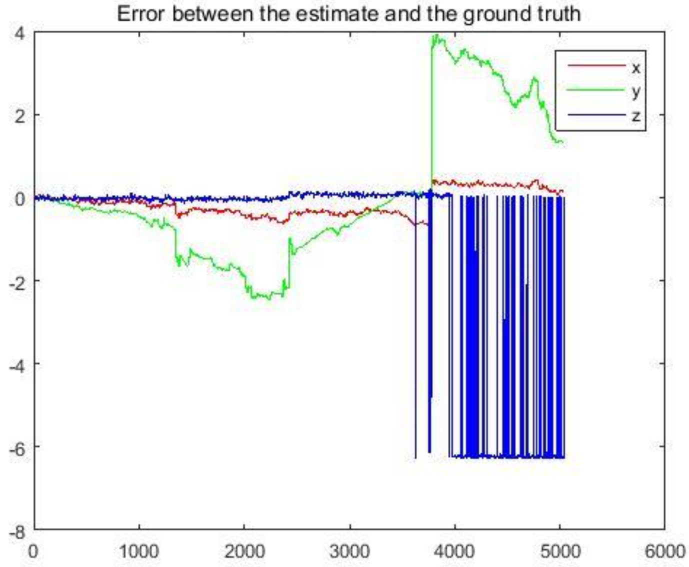


Figure 9. Error calculation between the estimates of camera poses and the ground truth after the scale factor is applied. When the heading angle changes to 180 degrees, error range is increasing.

Chapter 6. Conclusion

In our proposed system, the scaled camera poses can be extracted by using visual and inertial data. Sparse 3D reconstruction from SfM algorithm brings non-scaled camera poses, which has the trajectory of the camera movement. More consideration on calculation time will be needed because SfM needs high-level computation power, which the mobile phone may not have. During the tests, we have noticed that storing landmark data and camera parameters are very critical in our algorithm. We will continue to test our proposed system with various environments, such as school buildings and shopping malls. Unless the sudden change in the heading angle occurs, the errors between the estimates and the ground truth are in the range of −0.1 m to 0.4 m. However, after abrupt angle change, the error comparison does not work due to bounce the values. It can be assumed that loop closure algorithm may work critically if there is an overlap on the moving trajectory. In our future work, applying outlier rejections and loop closure method will provide better results. Also, a large portion of calculation time is spent when SfM reconstruction occurs. In the future work, server-client network for our system will be considered in order to reduce calculation time.

Bibliography

[1] Hol, Jeroen D., et al. "Sensor fusion for augmented reality." *Information Fusion, 2006 9th International Conference on*. IEEE, 2006.

[2] Martinelli, Agostino. "Vision and IMU data fusion: Closed-form solutions for attitude, speed, absolute scale, and bias determination." *Robotics, IEEE Transactions on* 28.1 (2012): 44-60.

[3] Nützi, Gabriel, et al. "Fusion of IMU and vision for absolute scale estimation in monocular SLAM." *Journal of intelligent & robotic systems* 61.1-4 (2011): 287-299.

[4] Zou, Han, et al. "A fast and precise indoor localization algorithm based on an online sequential extreme learning machine." *Sensors* 15.1 (2015): 1804-1824.

[5] Beder, Christian, and Martin Klepal. "Fingerprinting based localisation revisited: A rigorous approach for comparing RSSI measurements coping with missed access points and differing antenna attenuations." *Indoor Positioning and Indoor Navigation (IPIN), 2012 International Conference on*. IEEE, 2012.

[6] Lazik, Patrick, and Anthony Rowe. "Indoor pseudo−ranging of mobile devices using ultrasonic chirps." *Proceedings of the 10th ACM Conference on Embedded Network Sensor Systems*. ACM, 2012.

[7] Hartley, Richard, and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[8] Mur−Artal, Raul, J. M. M. Montiel, and Juan D. Tardos. "ORB−SLAM: a versatile and accurate monocular SLAM system." *Robotics, IEEE Transactions on* 31.5 (2015): 1147−1163.

[9] Davison, Andrew J., et al. "MonoSLAM: Real−time single camera SLAM."*Pattern Analysis and Machine Intelligence, IEEE Transactions on* 29.6 (2007): 1052−1067.

[10] Engel, Jakob, Thomas Schöps, and Daniel Cremers. "LSD−SLAM: Large−scale direct monocular SLAM." *Computer Vision–ECCV 2014*. Springer International Publishing, 2014. 834−849.

[11] Tanskanen, Petri, et al. "Live metric 3d reconstruction on mobile phones."*Proceedings of the IEEE International Conference on Computer Vision*. 2013.

[12] Castellanos, José A., et al. "Robocentric map joining: Improving

the consistency of EKF-SLAM." *Robotics and autonomous systems* 55.1 (2007): 21-29.

[13] Kalal, Zdenek, Krystian Mikolajczyk, and Jiri Matas. "Tracking-learning-detection." *Pattern Analysis and Machine Intelligence, IEEE Transactions on*34.7 (2012): 1409-1422.

[14] Corke, Peter, Jorge Lobo, and Jorge Dias. "An introduction to inertial and visual sensing." *The International Journal of Robotics Research* 26.6 (2007): 519-535.

[15] Koenderink, Jan J., and Andrea J. Van Doorn. "Affine structure from motion."*JOSA A* 8.2 (1991): 377-385.

[16] Kalal, Zdenek, Jiri Matas, and Krystian Mikolajczyk. "Pn learning: Bootstrapping binary classifiers by structural constraints." *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE, 2010.

[17] Bailey, Tim, et al. "Consistency of the EKF-SLAM algorithm." *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*. IEEE, 2006.

[18] Zhang, Zhanyu, et al. "Monocular vision simultaneous localization and mapping using SURF." *Intelligent Control and*

Automation, 2008. WCICA 2008. 7th World Congress on. IEEE, 2008.

[19] Milette, Greg, and Adam Stroud. *Professional Android sensor programming*. John Wiley & Sons, 2012.

초 록

최근 모바일 기기와 네트워크의 발전을 통해서 실내 위치 기반 서비스 관련 연구가 빠르게 이루어지고 있다. SLAM 이라고 불리는 위치 및 맵핑 문제와 관련된 여러 연구들은 Wi-Fi, IMU 센서, 초음파 센서와 같은 다양한 기기들을 통해서 해당 문제에 접근하고 있다. 하지만 보다 직관적이고 접근성이 편한 실내 위치 기반 서비스에 대한 필요성이 대두되고 이를 통해 현재 위치와 기기의 자세에 대한 높은 인식률을 요구하고 있다. 본 연구에서는 핸드폰의 카메라를 통한 비전 데이터와 IMU 센서를 통한 관성 데이터를 결합한 위치 정보 확인 시스템을 제시하고자 한다. 지정된 랜드마크의 이미지 세트와 위치 정보 데이터 베이스를 생성을 포함하여 미리 학습하는 과정이 우선 이루어진다. 이후 TLD 추적기라고 불리는 이미지 추적 알고리즘을 센서데이터와 함께 동시에 사용하여 사용자의 기기의 위치를 예측할 수 있다. 기존에 얻은 카메라의 파라미터들과 비전 데이터로부터 얻은 스케일 팩터를 통해서 실제 크기에 가까운 카메라 및 기기의 위치 및 자세를 추정할 수 있다. 비록 기기의 급작스러운 움직임이 기기 이동을 예측하는데 심한 오차를 내는 경우도 있지만, 미리 측정한 기존 이동 경로와 비교하여 -0.1m 에서 0.4m 내의 오차로 기기의 위치 추정이 가능하다.

키워드 : SLAM, 싱글 카메라, 확장 칼만 필터, 실내 위치 추정,
물체 추적

학 번 : 2014-22492