Proceeding of the 2015 IEEE
International Conference on Information and Automation
Lijing, China, August 2015

A Novel Approach to Improve the Precision of Monocular Visual Odometry

Chen Xiao[1,2], Xiaorui Zhu[2], Wei Feng[1,3], Yongsheng Ou[1,3,*]

***Abstract*— Monocular visual odometry is an active research topic for mobile robot navigation due to its availability and simpleness. However, it inherently suffers from scale ambiguity inherently, so that the precesion of odometry becomes poor. In this paper, we propose a new method to resolve scale ambiguity for monocular visual odometry based on ground area extraction and a modified adaptive kalman-filter which is based on Support Vector Machine (SVM). Firstly, instead of using Homography directly, we combine the Watershed Algorithm with the proposed Edge Expansion Method (EEM) to realize the ground extraction. Secondly, for the purpose of reducing the possibility of divergency when using the kalman filtering algorithm in real scenes, this paper applies a modified SVM-based Adaptive Kalman Filtering algorithm (SVMAKF) to visual navigation area. We conduct some experiments in outdoor scenes to validate that these approaches can improve the accuracy of monocular visual odometry.**

***Index Terms*— Visual Odometry, Monocular, Scale Ambiguity, SVM, Kalman**

I. Introduction

The research on the robot navigation which is based on visual odometry has gotten the extensive concern in recent years. Compared with traditional navigation techniques, visual odometry can work well without GPS and does not suffer the problem of wheel slip [1]. The trajectory will not drift if the robot does not move. Visual odometry can be divided into two main types: monocular visual odometry and stereo visual odometry. Stereo visual odometry is widely applied to robot auto-navigation area, due to its higher accuracy of distance measurement. For instance, Nister et. al. [2] introduced a real-time visual odometry which uses Harris operator to extract feature points and then match these features by normalization correlation. They conducted a long distance experiment to validate their odometry. On the other hand, monocular visual odometry only needs one camera, which is more cost-effective and easier to configure on existing equipment. So monocular visual odometry has higher applicative prospect.

Monocular visual odometry is easier to implement than stereo visual odometry, but the estimation of motion based on monocular vision suffers the scale ambiguity inherently. It is hard to get the exact translation distance directly from the consecutive images. The scale ambiguity may be the toughest obstacle for monocular visual applications. In this case, many researchers presented their solutions to this problem. One of the effective solutions is adding some distance sensors(e.g. IMU and speedometer). Thus, combining these sensors with camera to form a navigation solution of multi-sensor information fusion. However, solving this problem without extra sensors seems to be more attractive. As for another solution, MonoSLAM [3] selects a landmark whose 3D position is already known at the beginning of SLAM, then MonoSLAM uses this known initial scale as an invariant scale during the SLAM process. However, this solution will increase the drift error, because we can hardly ensure that the scale remains the same during the whole process of robot movement. In addition to the two solutions above, some researchers are focusing on the way of using the known fixed height between the camera and the ground plane to be the reference scale. Assuming that the robot moves on a flat ground and the height between the camera and the ground is constant, it is consistent with most real scenes. So that the height can be a good reference for scale estimation. For example, Choi et. al. [4] used this height in planar homography. First of all, they used "Surface Context" proposed by Hoiem to extract the ground area from images and then estimated scale by homography of ground. Another work of Choi [5] shared the same idea with LIBVISO2 [6] [7]. They used the Kernel Density Estimation (KDE) method to find the feature points on the ground from the reconstructed 3D feature points, then they got the scale by comparing the reconstructed height of camera with the real height. This paper will also keep a watchful eye on the method based on ground.

In the field of mobile robot navigation, the kalman filtering algorithm is one of the most common techniques [8] [9] [10] which are usually used for robot pose prediction, pose smoothing, and improving the accuracy of navigation.

This work is partially supported by National High-Tech Research and Development Program of China (863 Program) (Grant No. 2015AA042303), the National Natural Science Foundation of China (Grant No. 61273335), Hundred Talents Program of the Chinese Academy of Sciences (Grant No. Y14406), Guangdong Innovative Research Team Program (201001D0104648280), Shenzhen Fundamental Research Programs (JCYJ20120831180626842, JCYJ20140718102705295).

[1]C. Xiao, W.Feng and Y. Ou are with the Center for Intelligent and Biomimetic Systems, Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Guangdong, P. R. China.

[2]C. Xiao and X. Zhu are with the School of Mechanical Engineering and Automation, Harbin Institute of Technology Shenzhen Graduate School, Guangdong, P. R. China.

[3]W.Feng and Y. Ou are also with the Chinese University of Hong Kong, China.

*The corresponding author (phone: +86-755-86392137, email: ys.ou@siat.ac.cn)

978-1-4673-9104-7/15/$31.00 ©2015 IEEE

The Support Vector Machine (SVM) algorithm [11] [12], proposed by Vapnik firstly, is widely applied in classification, nonlinear function estimation, and convex optimization. Some researchers have applied SVM to robot navigation area. For instance, Sangwoo et. al. [13] used SVM to extract the feature points on clouds from images, then they can estimate the motion of the robot by treating the clouds as invariant landmarks. Others (e.g. [14] [15]) often used collected images of scenes to train SVM in advance, then the robot can recognize the scene by SVM when the robot moves to the same scene trained before, thus the purpose of localization is reached. However, in the actual situation, it is hard to collect images of every key scene and it is unlikely to guarantee the weather is always good if the method is based on cloud. There is a way to associate SVM with Kalman Filtering. Hong-De et. al. [16] proposed a SVM-Based Adaptive Kalman Filtering (SVMAKF) algorithm which uses SVM to tune the measurement noise covariance matrix of kalman filtering model. SVMAKF shows better performance that it can prevent the filter from divergence. This paper will modify the SVMAKF method and apply it to visual navigation after considering some real situations.

This paper proposes a novel method of monocular visual odometry for mobile robots. Compared with present popular visual odometry methods, it has two main innovations: Firstly, in order to resolve the scale ambiguity of monocular visual odometry and estimate accurate translation scale, we propose a novel ground extraction method, called "Edge Expansion Method" (EEM), which can be used with the Watershed Algorithm in combination. We can extract the ground area from video effectively by EEM, then estimate the scale by planar homography. Secondly, after estimating the transformation of camera by combining the eight-point method and RANdom SAmple Consensus (RANSAC) algorithm, for the purpose of improving the accuracy of estimating robot motion, this paper applies SVMAKF algorithm to visual navigation area, and we modify it to adapt the real situation. The rest parts of this paper are organized as follows: the proposed method is shown in detail in Section II; experiment result is shown in Section III; and conclusions and discussions part is in Section IV.

II. Monocular Visual Odometry

A monocular visual odometry usually consist of "image input", "feature tracking", "motion estimation", "motion smoothing". The visual odometry proposed in this paper uses feature-based solution and realizes the estimation of motion by epipolar geometry. The main procedures of our visual odometry can be find in Fig. 1.

The Transformation T gotten from essential matrix is a unit vector without any scale information. we propose two approaches in the process of constructing odometry to improve the precision. The details will be discussed below.

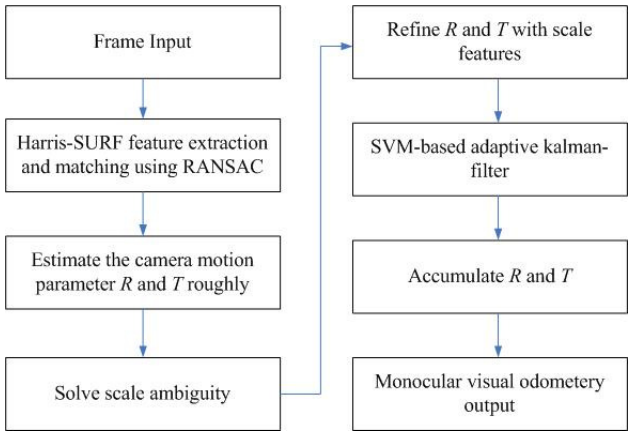


Fig. 1. The framework of the proposed method

A. Scale Estimation

The key procedure of using feature points on ground to estimate the relative motion scale is how to extract the ground area from consecutive images or video. Some works (e.g. [17] [18] [19] [20]) were proposed on how to separate ground from image, which either based on planar homography or based on appearance models. Different from methods above, we propose a novel fast ground extraction method which is based on classical Watershed algorithm and our "Edge Expansion Method" (EEM). The most obvious strength of the proposed method is that it is very easy to realize while keeping the accuracy. Watershed algorithm is a topology-based image segmentation method proposed by L. Vincent. We use the open source software library OpenCV to realize Watershed. To make segmentation, firstly, it is needful to draw some markers on the image manually, Watershed will treat these markers as centers and extend out to shape some areas. If there is only one marker, then the entire image is one area. If there are too many markers, it will cause over-segmentation. As a result, the present Watershed in OpenCV can not be used to extract ground area from consecutive images or video directly. So we design a so-called "Edge Expansion Method" to cover the shortage of Watershed. This new method will search the appropriate markers automatically after an initialization, then use the Watershed to separate the image into ground area and non-ground area. After the initial extraction of ground, the ground area will be tracked in following image sequence. As we can see from Fig. 2 , Our method is working as follows:

1) Initial Segmentation: This step only runs at the first frame of image sequence, the purpose of initialization is to tell the program where the ground is, so that the algorithm can track the ground area in subsequent images. Because the algorithm can not get any priori knowledge about what is ground or what does ground looks like. To make an initialization, it is necessary to draw two markers on the first

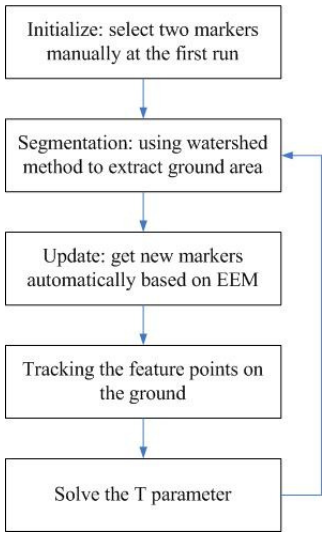


Fig. 2. The procedure for solving the scale ambiguity

frame manually. The markers can be points or curves. Our rules say the first marker should locate in ground area and the second marker should locate in non-ground area. Experiment shows, the second marker should better be a curve around ground area with a short distance.

2) Run Watershed: By treating the two markers as seeds, the Watershed will separate the image into two areas, and the accuracy is related with markers. Now, the area where the first marker locates in is ground.

3) Edge Expansion Method: When ground area is extracted from the current frame, we need to prepare two new markers for the next tracking. For the first marker, random search two points in the current detected ground area which satisfy the following constraints: 1. points in the 9 by 9 window (size can be changed) whose center is the random point are still in the current ground area. This condition ensures that the random point will not be close to the ground edge; 2. The line segment which consists of these two random points is still in the current ground area; 3. The length of line segment which consists of these two random points is within a range (e.g. between 20 pixels and 50 pixels). And then the second marker which is important, as experiment shows, is supports to be the curve which is around the ground area and should not be too close to the ground edge. To select this curve, we firstly extract the detected ground area (like the white area in Fig. 1 (b)), then we expanse this white area to be larger, where "expanse" means the edge of ground should extend out. And the expansion scale is related to the module of camera translation vector:

$$scale = a|t| \tag{1}$$

where t is the accurate translation vector with scale information got in previous frame and it will be introduced in following paragraphs, a is a coefficient which can be tune through experiments. The new edge after expansion represents the extending trend of ground, and let this new edge as the second marker.

4) Ground Tracking: Repeat step 2 and 3 will realize selecting markers automatically and tracking the ground stably. The following Fig. 3 shows our ground extraction method.

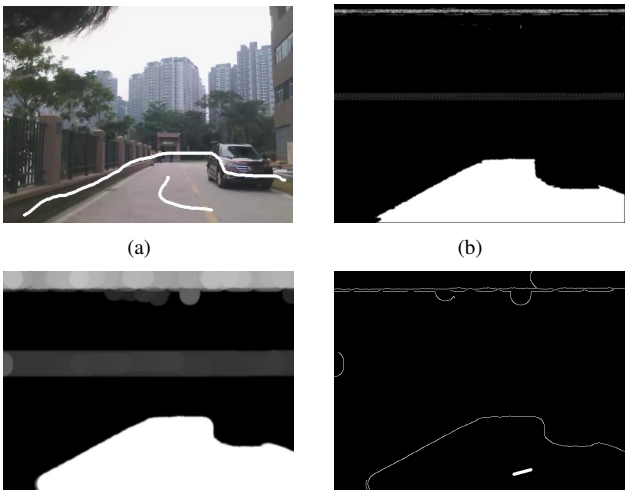


Fig. 3. This figure shows the procedure of selecting markers: (a) select two initial markers manually; (b) get the edge of ground in current image; (c) expand the edge by our EEM; (d) two new markers consist of expanded edge and random line segment.

When the ground area is successfully extracted from image sequence, grouping these feature points obtained from subsection A into "points on ground" and "points on non-ground". Now, we can use the planar homography of ground to estimate the translation scale. Firstly, the corresponding points on ground satisfy the following homography relationship:

$$m' = Hm \tag{2}$$

$$H = R - \frac{tn^T}{d} \tag{3}$$

where H represents Homography matrix which can be calculated from the matching points on the ground, the n represents normal vector of ground plane which can be set as a constant if robot moves on the flat. d is the real height between the camera and the ground, and R is already obtained from subsection B, so t is the only unknown variable in equation above that we can get t easily.

Using this t to replace the T decomposed from SVD then we get the accurate rotation and translation of camera without scale ambiguity.

B. SVMAKF-Based Odometry

If we use the rotation and translation got from subsections above to structure odometry directly, it will bring big errors to localization. In order to make the localization result more smooth, researchers usually think about using Kalman Filtering (KF) algorithm. The classical five steps of KF are listed below, we won't introduce them in detail anymore.

$$\hat{X}_{k/k-1} = \Phi_{k/k-1}\hat{X}_{k-1} \tag{4}$$

$$P_{k/k-1} = \Phi_{k/k-1}P_{k-1}\Phi^T_{k/k-1} + Q_{k-1} \tag{5}$$

$$\hat{X}_k = \hat{X}_{k-1} + K_k(Z_k - H_k\hat{X}_{k/k-1}) \tag{6}$$

$$K_k = P_{k/k-1}H^T_k(H_kP_{k/k-1}H^T_k + R_k)^{-1} \tag{7}$$

$$P_k = (I - K_kH_k)P_{k/k-1} \tag{8}$$

where Q and R represent system process noise matrix and system measurement noise matrix. These two matrix are usually set to be known and invariant, but in real situation, it is impossible to be invariant.

Support Vector Machine (SVM) is an effictive machine learning algorithm used to classfication and regression, and it is proposed by Vipnik based on statistical learning theory.

In order to improve the performance of kalman filtering, we apply a modified SVM-Based Adoptive Kalman Filtering (SVMAKF) [16], which is modified by this paper to adapt the real scene so that it can be used in visual navigation field.

Firstly, we define the innovation matrix as the difference between actual measurement and estimating measurement, represented as follows:

$$r_k = Z_k - H_kX_{k/k-1} \tag{9}$$

The key problem of adoptive kalman filter is that how to tune Q and R dynamically according to the value of innovation. Covariance-matching technique is known for this purpose. In this paper, we assume that Q is invariant and we just care about R. Defining the theoretical covariance of innovation which is calculated as:

$$P_{rk} = H_k(\Phi P_{k-1}\Phi^T + Q)H^T_k + R_k \tag{10}$$

Then, the actual covariance of innovation can be estimated by sampling:

$$\hat{P_{rk}} = \frac{1}{n}\sum_{j=k-n+1}^{k} r_jr^T_j \tag{11}$$

where k represents the size of sampling window. if the theoretical covariance of innovation is not consistent with the actual covariance of innovation, we need to use the SVM to tune R. So, we define the Degree of Matching (DOM) as :

$$DOM_k = P_{rk} - \hat{P_{rk}} \tag{12}$$

the DOM_k in [16] is different from ours. In their definition, the KF is not valid in multi-input-multi-output system (MIMO). The new measurement noise covariance matrix will be updated according to DOM_k:

$$R_k = S_kR_{k-1} \tag{13}$$

where S_k is called tuning factor which is related to DOM_k directly. We can let DOM_k be the input of SVM, and S_k be the output of SVM, thus it forms a adaptive system. However, before using SVM, we must choose an appropriate training data set in advance, that is to say, we need enough DOM_k and S_k before the KF works. DOM_k is easy to get, the difficulty is that we can not get S_k before SVM was trained. So, this paper introduces a solution that using the S_k got from Sage-Husa Algorithm [21] as the training data of SVM. S_k for training can be calculated as:

$$S_k = R_kR^{-1}_{k-1} \tag{14}$$

where R_{k-1} is the noise matrix before Sage-Husa tuning, and R_k is the noise matrix after Sage-Husa tuning. Because the R of MIMO system is a multidimensional square matrix, so S_k is a multidimensional square matrix too, then we can infer that DOM_k must be a multidimensional square matrix, and this is why we do not use the definition of DOM_k in [16].

After SVM was trained, a SVM-based adaptive kalman filter will work, R and t are the inputs of this kalman filter, then we get a SVMAKF-based odometry now.

III. Experiments

In this section, in order to validate the performance of the proposed methods, we conduct a series of experiments. The experimental images are captured in the real scenes by an ordinary webcam whose resolution is 640 by 480, and these images are processed in a quad-core 2.8Ghz laptop, but only one core is used in the experiments.

A. Ground Extraction Result

We tested the proposed EEM method in a real outdoor scene. The method extracts ground from image sequence and tracks it, Fig. 4 shows the results. When the robot moves straight, from Fig. 4(a) - 4(b), one can see, the ground area is marked in green, and it is almost overlap with the real area. When the robot turns, it is hard to predict the extending direction of ground in camera view in advance, but the proposed algorithm shows a potential direction for this situation, results are show in Fig. 4(c) - 4(d). In addition, the running time of EEM is show in Tab. I. We select 8 frames between 100 and 107 to count the time, as one can see, the average running time is about 40.1340ms. and it is enough to real application. It is equally competent for most real-time applications.

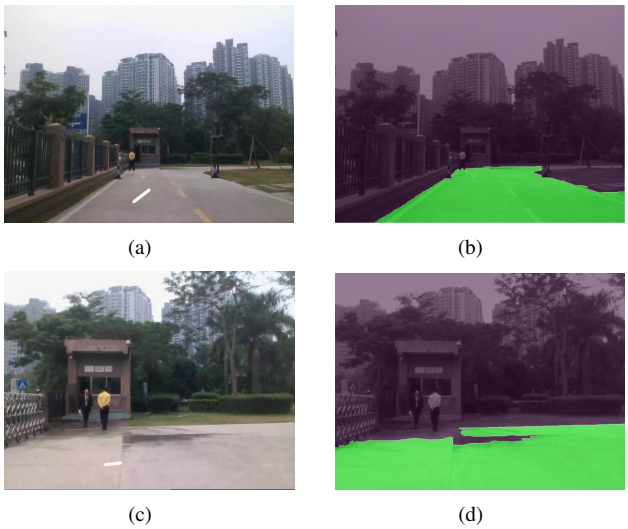


Fig. 4. This figure shows the ground extraction result: (a) the original image when robot moves straight; (b) extract ground area from image (a); (c) the original image when robot turns; (d) extract ground area from image (c).

TABLE I
RUNNING TIME OF EEM

Frame	100	101	102	103
Time(ms)	38.6859	38.5442	46.2989	40.9116
Frame	104	105	106	107
Time(ms)	38.1612	38.2056	40.8135	39.4509

B. Localization Result

We compare the scale estimation results of our proposed method and Kernel Density Estimation method (KDE) proposed by Choi [8], and trajectory figure shows the improvement made by SVMAKF. We fix the camera at a height of 1.15m and its pitch angle is zero degree, the robot moves about 380m. Fig. 6 shows the estimating translation errors of EEM and KDE when robot moves different lengths, and the EEM method seems cause smaller error and improve the accuracy of odometry. Then we treat the localization result of GPS in smartphone as Ground Truth, and compare it with localization trajectory got from SVMAKF. Fig. 5(a) shows the ground truth recorded by GPS, while Fig. 5(c) shows the trajectories estimated by EEM and KDE. From Fig. 5(c), the path length estimated by KDE is about 420m which has a translation error of about 10.5%, while the path length estimated by EEM is about 390m which has a translation error of about 2.6%. Fig.5(b) shows the trajectories estimated by ordinary KF and SVMAKF. It is observed that SVM improves the performance of kalman filtering, especially when robot turns. Taking the requirement of real-time navigation into consideration, we conduct many experiments in the open air then count the running time of our odometry as show in Tab. II. We select 8 frames between 200 and 207 to count the time, as we can see, our odometry takes about 364.160ms every time. It means the proposed odometry can run at the frequency of about 2 Hz to 3 Hz which is qualified for some navigation applications.

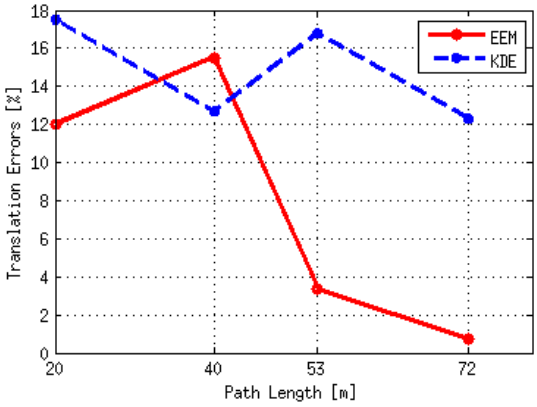


Fig. 6. Generally, the estimating scale errors by EEM (the red solid line) are smaller than KDE (the blue dash line) at different path lengths.

TABLE II
RUNNING TIME OF ODOMETRY

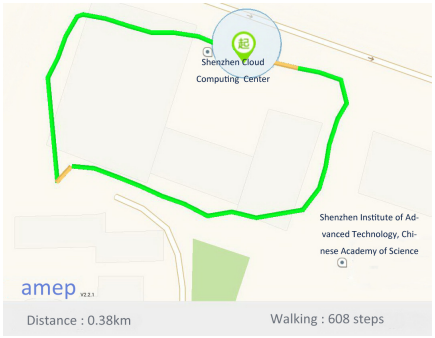
Frame	200	201	202	203
Time(ms)	345.181	366.984	365.588	332.818
Frame	204	205	206	207
Time(ms)	395.512	378.923	336.248	392.027

IV. CONCLUSIONS

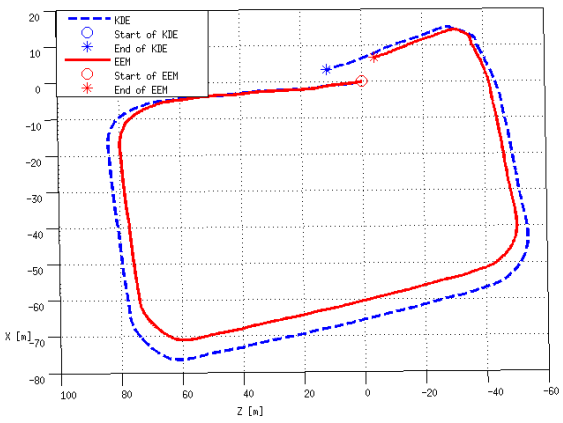
In this paper, we propose a novel ground area extraction method which is fast, accurate and convenient to realize. We use this novel method to resolve the scale ambiguity of monocular vision by making the height between camera and ground as a constraint, it can improve the precision of monocular visual odometry. In addition, because the ordinary kalman filtering algorithm may divergence in real scene if we use the fixed noise matrix, this paper proposes a modified SVM-based adaptive kalman filtering algorithm to tune the noise matrix automatically. The experiments show that the SVMAKF method can improve the effectiveness and accuracy of odometry, and the proposed method allows us to apply kalman filtering to the visual navigation area from another perspective.

REFERENCES

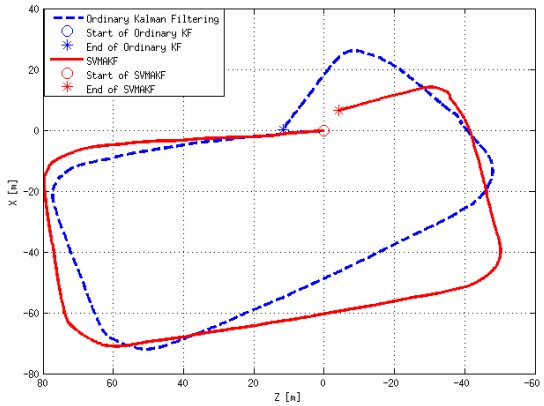
[1] X. J. Song, et al. “Optical flow-based slip and velocity estimation technique for unmanned skid-steered vehicles.” in IEEE/RSJ Int. Conf. Intelligent Robots and Systems., Sept. 2008, pp.101-106.
[2] D. Nister, O. Naroditsky and J. Bergen, “Visual Odometry,” in IEEE Computer Society Conf. Computer Vision and Pattern Recognition., 2004, Vol. 1, pp. 652-659.



(a)



(b)



(c)

Fig. 5. (a) The green trajectory shows the moving path of robot which is recorded by smartphone with GPS; (b) localization trajectories of EEM (which is red) and KDE (which is blue); (c) localization trajectories of ordinary Kalman Filtering (which is blue) and SVMAKF (which is red).

[3] J. Andrew, et al. "MonoSLAM: Real-time single camera SLAM." in IEEE Trans. on Pattern Analysis and Machine Intelligence., 2007, Vol. 29, No. 6, pp.1052-1067.

[4] S. Choi, et al. "What does ground tell us? monocular Visual Odometry under planar motion constraint." in IEEE International Conference on Control, Automation and Systems, 2011, pp.1480-1485.

[5] S. Choi, P. Jaehyun, and Y. Wonpil. "Resolving scale ambiguity for monocular Visual Odometry." in IEEE International Conference on Ubiquitous Robots and Ambient Intelligence, 2013, pp.604-608.

[6] B. Kitt, G. Andreas, and L. Henning. "Visual Odometry based on stereo image sequences with ransac-based outlier rejection scheme." Intelligent Vehicles Symposium (IV), 2010, pp.486-492.

[7] A. Geiger, Z. Julius, and S. Christoph. "Stereoscan: Dense 3d reconstruction in real-time." Intelligent Vehicles Symposium (IV), 2011, pp 963-968.

[8] H. R. Song, W. S. Choi, and H. D. Kim. "Depth-aided robust localization approach for relative navigation using RGB-depth camera and LiDAR sensor." in IEEE International Conference on Control, Automation and Information Sciences (ICCAIS), 2014, pp.105-110.

[9] J. Simanek, K. Vladimir, and R. Michal. "Improving multi-modal data fusion by anomaly detection." Autonomous Robots, 2015, pp.1-16.

[10] T. Genevois , and T. Zieli?ska. "A simple and efficient implementation of EKF-based SLAM relying on laser scanner in complex indoor environment." Journal of Automation Mobile Robotics and Intelligent Systems, Apr. 2014, Vol 8, Issue 2, pp.58-67.

[11] C. Cortes, and V. Vladimir. "Support-vector networks." Machine Learning, 1995, Vol. 20, No.3, pp.273-297.

[12] J. Suykens, and J. Vandewalle. "Least squares support vector machine classifiers." Neural Processing Letters, 1999, Vol.9, No.3, pp.293-300.

[13] C. Sangwoo, E. Dunn, and J. Frahm. "Rotation estimation from cloud tracking." in IEEE Winter Conference on Applications of Computer Vision (WACV), 2014, pp.917-924.

[14] G. John , et al. "Simultaneous localization and mapping with learned object recognition and semantic data association." in IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2011, pp.1264-1270.

[15] C. Jack, and R. Alejandro. "Environment classification for indoor/outdoor robotic mapping." in IEEE Canadian Conference on Computer and Robot Vision, 2009, pp.276-283.

[16] H. D. DAI, et al. "Study of support vector machine based adaptive Kalman filtering." Control and Decision, 2008, Issue 8, pp.949-952.

[17] D. Hoiem, A. Efros, and M. Hebert. "Recovering surface layout from an image." International Journal of Computer Vision, 2007, Vol.75, No.1, pp.151-172.

[18] J. Arrspide, et al. "Homography-based ground plane detection using a single on-board camera." Intelligent Transport Systems, 2010, Vol. 4, pp.149-160.

[19] L. Narayanan, and V. Prashanth. "Improved ground plane detection in real time systems using homography." in IEEE International Conference on Consumer Electronics (ICCE), 2014, pp.199-200.

[20] J. Zhou, and B. X. Li. "Homography-based ground detection for a mobile robot platform using a single camera." in IEEE International Conference on Robotics and Automation, 2006, pp.4100-4105.

[21] P. Andrew, and W. Husa. "Algorithms for sequential adaptive estimation of prior statistics." in IEEE Symposium on Adaptive Processes Decision and Control, 1969, pp.61-61.