Vision based Semantic Mapping and Localization for Autonomous Indoor Parking

Michael Shell, *Member, IEEE,* John Doe, *Fellow, OSA,* and Jane Doe, *Life Fellow, IEEE*

***Abstract*—Autonomous indoor parking without human intervening is one of the most demanded and challenging tasks of autonomous driving system. The key point to this task is real-time precise indoor localization. However, most indoor parking lots are composed of monotonous texture-less scenes and thus, are hostile to traditional visual feature-based SLAM methods. In this paper, we proposed a novel and practical solution of real-time indoor localization for autonomous driving in parking lots. High level landmarks, the parking slots, are extracted and enriched with labels to avoid the instability of low-level visual features. We then proposed a robust method for detecting incorrect data associations between parking slots and further extended traditional optimization framework by dynamically eliminating suboptimal data associations. Visual fiducial Tags are also introduced to improve the overall precision. Their number and distribution are also analyzed and compared. As a result, a semantic map of parking lot can be established fully automatically and robustly. We experimented the performance of real-time localization based on the map using our autonomous driving platform TiEV and the average accuracy of 0.3m tracking tracing can be achieve at a speed of 10kph.**

***Index Terms*—Indoor, ParkingLot, Semantic landmark, Robust SLAM**

I. Introduction

AUTONOMOUS driving has been witnessed great progress in recent years, breakthrough has been made in several harsh fields, including obstacle detection, real-time motion planning and high precision localization (mostly based on differential GNSS). Recently, testing self-driving car can already drive safely in urban and suburban areas[**?**]. However, parking in a large indoor parking lot without human interfere is still an unsolved problem. One critical reason is the lack of robust high precision localization mean in these GNSS forbidden area. Traditional indoor localization methods require pre-equipment sensors, such as WiFi, Bluetooth or UWB. Wireless signal suffers from occlusion and decays while users distance to signal sources increases, so a significant number of stations are needed for stability, let alone their relative low precision[**?**]. Laser-based SLAM system is eligible for localization an unmanned vehicle in environments such as a factory or a warehouse[**?**]. However, these range based representation is of high data volume and is vulnerable to dynamic scenes. As a result, visual SLAM (VSLAM) built on low-cost cameras became one of the most favorable localization method.

M. Shell was with the Department of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA, 30332 USA e-mail: (see http://www.michaelshell.org/contact.html).

J. Doe and J. Doe are with Anonymous University.

Manuscript received April 19, 2005; revised August 26, 2015.

VSLAM is known to be effective in texture-rich environment[**?**]. Nevertheless, they can easily fail in monotonously textured scene such an indoor parking lot. **?**] adopted sparse feature point based SLAM method with panorama images to localize a car in parking lots. But the extracted sparse feature can be unstable when the ground floor is stained with tire markings or water spots. The distortion presented in the stitched panorama images can also disturb the feature extraction. Recently, **?**] employ DSO method with forward looking camera for mapping and localization in indoor parking lot. The direct methods estimate camera poses directly based on photometric error derived from the whole image, thus are more robust than sparse methods in less-textured area [1] [2]. However, they often require high frame rate and are susceptible to global illumination change, which restrict their usage in unevenly illuminated indoor parking lot[3].

Moreover, the performances of direct SLAM systems also depend on a good map initialization.[3][**?**] (John: the link between these two line are not clear; Btw, how can a dense method realized re-localization? is it trivial or is it a shortcoming for these methods?)

(Huang:In LSD SLAMit is said that most dense method is simply a VO, while LSD has loop closure procedure. However, LSD itself do not have a map read and write module, but the LSD paper and the survey(Younes2016A) didn't mention this. Also since the direct method is not robust to illumination changes, re-localization using a pre-built map is of course almost unrealistic.)

Therefore, more stable and legible visual landmarks which are immune to various illuminate condition are demanded.

As a typical kind of semantic landmarks in parking lots, parking-space is now a favour for researchers [4] [5] [6]. Recently, deep learning-based method show its capability of accurate and robust detection of such kind of meaningful objects [**?**]. Inspired by these methods, we present an robust VSLAM system based on the recognition of high level landmarks for parking, i.e. parking-spaces and their IDs. Limited visual fiducial markers are introduced for improving overall accuracy and robustness. Facing the visual aliasing problem of parking slots, we proposed a robust outliers detection and elimination strategy in the optimization stage. Finally, a two dimensional map of parking slots can be robustly established which is distinguished from the traditional feature-based or point-cloud map for its stability, re-usability, light weight and human readable. Our system is implemented on an autonomous driving vehicle and tested in real parking lots.

SLAM has long been a classic topic in the robotics field[17] and recently became heated in the autonomous driving since

there are areas where GNSS is not available.[**?**]. **(author?)** [7] **(author?)** [8] and **(author?)** [9] use Extended Kalman Filter(EKF) to simultaneously optimize the sensor and landmarks positions in real-time. These methods modeled the optimization problem as a Markov Chain, thus can provided promising localization results with high efficiency. However, the computation grows quadratically with the number of landmarks[10], so these methods are bounded in room-sized domains when regarding visual features as landmarks. Probability filters have many further extensions such as UKF [], Information filter[], particle filter[] etc. However, they all assume the conditional independence of the current measurement with the historical states, which restricts the SLAM into a predict-update iteration loop. Inspired by the bundle adjustment research [], a factor graph based optimization framework (known as Graph SLAM) was proposed []. The graph-based optimization method is closely related to a Markov random field model, thus can involve the influence of all historical measurements, which together with its flexibility enable the Graph SLAM became the most popular SLAM method[**?**].

Generally VSLAM methods fall into two groups, so called feature-based methods (the indirect methods) and direct methods. In feature-based methods, low-level features are detected and treated as landmarks. To prevent integrating all the features into the factor graph(John:should use correct expression from PTAM), PTAM[11] separates tracking and mapping into two sub-tasks, keyframes are extracted and used in optimization, but because of what?? it still works in small areas(a desk or a room corner). [12] and [13] build city-scale sparse maps using G2O, a graph optimization framework[14]. Nevertheless, the optimization of sparse 3D point cloud is time-consuming, so the off-line mapping procedure is a must. Built on the idea of PTAM[11], ORB-SLAM [15] offers stable and efficient graph-based VSLAM system. With the keyframe detection and the BoW-empowered fast loop closure detection, ORB-SLAM performs well in various indoor and outdoor environments. However, ORB-SLAM is still easy to fail in texture-less environments.

Because feature-based methods are only capable of creating sparse feature-based mapping, they cannot be directly used in applications where full reconstruction is demanded, e.g. AR or structure from motion. Direct methods based on photometric error and utilize all image pixels are proposed [1]. Comparing to sparse feature-based methods, direct methods can output a semi-dense point cloud with higher quality at a real time speed. But in practice, direct methods require high rate of overlapping between consequent frames and high frame rate is also necessity since (to be filled) to estimate the depth accurately. Direct methods are also known to be vulnerable to motion blur, camera defocus and global illumination changes [16]. SVO [2] and DSVO[] combine advantages of feature-based method and direct method, and runs extremely fast (100 Hz). However, lacking of loop closure detection, these odometric methods drifts as time increases and gets lost easily.

Traditional SLAM methods do not incorporate human understandable meanings (semantics) associated with landmarks into the method, which now is recognized to be crucial for construct a human readable map and strengthen the descriptive power of landmarks[]. [17]. [18] added semantic labels to a LSD-SLAM framework to construct a dense map with classes attached to geometric entities, but semantic labels helps little in the optimization or localization stages. SLAM++ [19] and Semantic Fusion [20] employed semantic labels in the RGBD SLAM framework to aid the loop closure. However, both methods...

(!!should be very clear about the defects of these methods, i.e. 3D objects? not precise regarding to the localization of objects? can be confusing when multiple objects are searched?)

In [21], shop names and shop facades are recognized as labels in large indoor shopping spaces. [4] detects parking lots from the (fixed IP camera?) top view image, and use detected lots to estimate camera poses (is this a slam?).

[5] and [6] reconstruct the metric map and the semantic map of parking lot, which helps the route planning and parking task. (!!semantics in VW paper(2015) is not used in SLAM at all, not sure about the second one)

In a short conclusion, existing VSLAM methods generally could not perform robustly in texture-less area like an indoor parking lot. Therefore, more descriptive landmarks, especially landmarks attached with semantics should be used instead of low level feature in such a scenario.

II. APPROACH

Our semantic VSLAM system includes four fisheye cameras and one monocular camera. Four fisheye cameras are fixed at two reflectors, and at the front and rear bumpers, which consist a surround-view system. A top-view image is then fused from the surround-view inputs after intrinsic and extrinsic calibration, as shown in Fig. 2. In the top-view image, who indicates ground textures, parking slots are detected. The monocular camera is installed to the left of the real-view mirror to capture front scenes. The change of steering wheel angle, as well as the vehicle speed and heading direction collected by IMU are also used in our system.

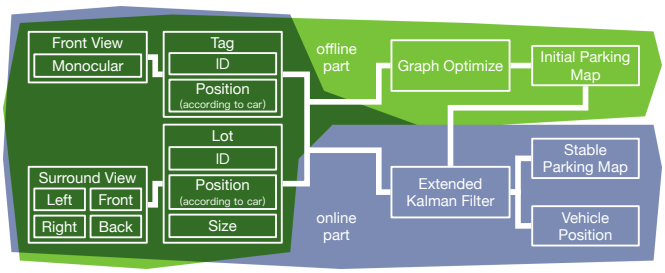

Fig. 1. Pipeline of the method

We choose parking slot as the landmark since parking slots are the most distinguishable objects in a parking lot, and the precise locations of parking slots are informative for localization and navigation during autonomous parking. Our parking slot detector is based on [22], in which corner points of parking slots are detected and classified (Fig. **??**). Afterwards, parking slots are assembled according to the corner points (Fig. **??**). Although the CNN-based method is capable of detection most kinds of corner points fast and robust, the

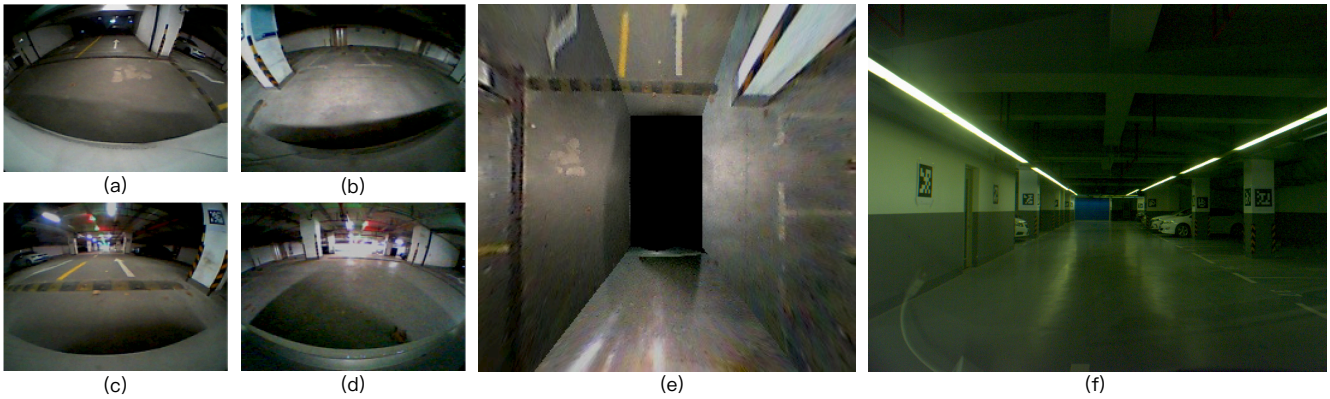

Fig. 2. (a)(b)(c)(d) are images from left, right, front, back fisheye cameras. (e) shows the top-view image fused from (a)(b)(c)(d). The image from the monocular camera is (f).

exact shape of the parking slot cannot be known due to the limited visible range of the surround vision system. As a result, the parking slot can only be guess initially and we have to optimize the shape of the parking slots in the SLAM system. Furthermore, the ID of each parking slot should be detected for facilitating data association between parking slots, which will be elaborated in sec .

Another kind of landmark used in our system is visual fiducial markers. Fiducial markers are introduced as an aid for the constancy of localization since few parking lot is detected near the entrances and exits (John:but we put the markers in the inner path of two rings?). In the practice, we found that fiducial markers may not be fully prohibited but their number can be limited to a small amount. We select AprilTags as visual markers for its robustness and high efficiency(a frame rate of 10 Hz)[23].

Fig. 3. An example of AprilTag

A. CNN based Parking slot Recognition

We adopt the method proposed by [22] to detect parking slots.

(John: Donot repeat existing method, briefly describe the detection method and emphasize on the post processing if there is any and the ID detection)

The raw detection results from the previous method are still not precise. We remove the entrance-line candidate who has more than two marking-point patterns on it to avoid this situation. Once a parking-line is detected, the direction of a lot is determined as well. And the depth of a parking lot is known a prior knowledge.

Since the relative car coordinate overlaps the top-view image coordinate, the lot coordinate in the top-view image is the same as that in the vehicle relative coordinate.

(John: ADD how to parameterize the parking slot, if we planned to optimize the shape of a parking slot?)

ID of a parking slot is important for the association of this semantic landmark. We fine turned PVANet to detect each character in one ID [24]. The entrance-line of the parking slot is firstly utilized to roughly locate the ID, as shown in Fig. **??**??). Then the image patches are extracted and send to the CNN (Fig. **??**??)). Due to the distorted and blurred texture in the surround view image, even the latest detection network could not offer the satisfactory performance. As a result, we devised a semantic-based fuzzy association method to cope with the uncertainty, which will be detailed in **??**.

B. Visual fiducial marker localization

The surround view system offers an intuitive ground observation at a high frame rate, however it has limitations. The visibility range and resolution of the top-view image is far from satisfactory, limiting the slot detection performance. And the calibration for image fusion becomes inaccurate as time goes, which will deteriorate the slot measurement.

Recalling the goal of our robustly localization system for autonomous driving and parking in various parking lots, certain numbers of faithful landmarks such as visual fiducial markers still have to be incorporated.

We adopted AprilTag and employed the detection framework from the AprilTag C source Open Library [1][23]. We further solved the relative position between visual markers and the vehicle by the PnP model in a fast and accurate way [26].

[1]https://april.eecs.umich.edu/wiki/AprilTags

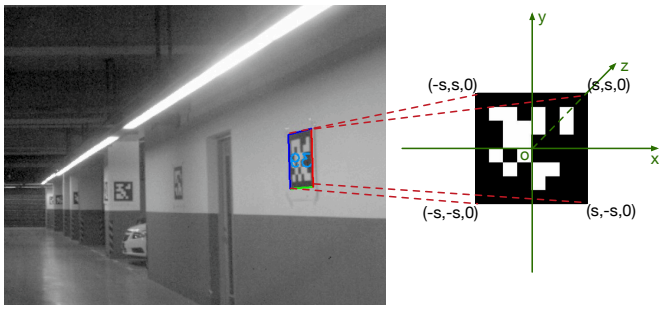

Fig. 4. An example of the relationship between a detected tag in the image plane(left figure) and its corresponding hypothetical 3D tag coordinate(right figure)

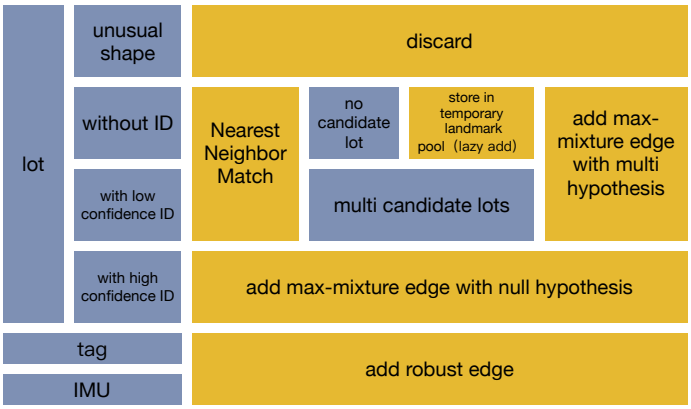

Fig. 5. The overall pipeline of the offline optimization stage

(John: Can we improve the efficient or the accuracy when localization a tag?)

In the PnP model, we assume four corner points of the tag always lie on the same plane, and thus definite the hypothetical 3D tag coordinate(whose origin is a tags geometrical center, x axis and y axis are both parallel to tags edge, pointing rightward and upward respectively, and z axis is perpendicular to the tag plane, pointing inward). So the relationship between the tag corner in the image and that in the hypothetical 3D tag coordinate can be described as, $R_{3\times3}\cdot {K_{3\times3}}^{-1}\cdot x_{3\times1}+t_{3\times1}=X_{3\times1}$

where $R_{3\times3}$ and $t_{3\times1}$ is the relative rotation and translation vector, $K_{3\times3}$ is the intrinsic camera matrix, $X_{3\times1}$ is the coordinate in the hypothetical 3D tag coordinate and $x_{3\times1}$ is the homogeneous coordinate in the image plane. $X_{3\times1}$ and $K_{3\times3}$ are prior knowledge, and $x_{3\times1}$ has already been detected. Only $R_{3\times3}$ and $t_{3\times1}$ are unknown parameters with 6 DOF, more than 3 corresponding points are needed for iteration.

Once $R_{3\times3}$ and $t_{3\times1}$ is calculated, the Rodrigues transformation is performed on $R_{3\times3}$ to separate the angle(α) bonded by the array starts from the tag to the north and the array starts from the tag center to the vehicle. The distance(d) is the 2-norm of $t_{3\times1}$. So the tag locates at $(x=sin(a)\cdot d, y=cos(a)\cdot d)$ in the vehicle relative coordinate.

These visual fiducial markers are flexible and easily implemented. They brought another benefit for the autonomous parking purpose, that fiducial markers can easily indicate the existences of pillars and walls which can only be robustly detected by expensive laser scanners. These obstacle information can facilitate the route planning inside of a parking lot.

C. Optimization

At every frame, the relative car-landmark position, together with the speed and angular changes of vehicle should be added to the parking map incrementally. These values are archived from a Kalman-based extrapolater with the steering wheel, car speed and the compass readings from a cheap IMU as the input. During the mapping stage, the popular graph optimizer G2O is used, some experimental designed rules are also introduced for outliner detection. Thus, the offline part outputs a 2D parking map and an optimized vehicle trajectory. During the online stage, the former parking map acts as the initial map, and is updated simultaneously with the extended Kalman filter(EKF). This results in a smoother trajectory, which helps the further control task.

In this section, a 2D map is built by fusing all the observation datum collected in the former sections. The accumulated error of IMU may cause drifting in localization. Due to imprecise detection of tags or lots, there will be ambiguities in the map. To avoid these problems, we use the popular g2o[14] framework together with the max-mixture model [25] to optimize both car and landmarks positions and ensure correct data association in the offline section, the figure above give the overall view of the optimization step.

During the offline stage, we use [14] to build the initial parking map. All the observations are treated as edges in the graph. Observations from vehicle to tag or vehicle are presented by G2O standard edges. However, observations from vehicle to parking lot sometimes contain miss data association causing by lot ID detection error and lead to great residual error in the graph. ID detection error is inevitable, so max mixture models [25] are introduced to eliminate the mistaken associations.

1) G2O Optimization Framework: [14] offers a global framework for nonlinear least square problems. Under the G2O framework, all the elements to be optimized are regarded as vertices, while all the observation constraints are known as edges who connect vertices. The least square problem of graph optimization in SLAM field can be expressed by the following equation:

$$F(x)=\sum_{k\in C} e_k(x_k,z_k)^T\Omega_k e_k(x_k,z_k)\text{[14]}$$

$$x^*=\underset{x}{argmin}F(x)\text{[14]}$$

where k denotes the k Th vertex in the graph(either a vehicle or a landmark position); Cdenotes the total ID group; $x_k$is the parameter block of each vertex(containing the vertexs location and direction), while z_k and Ω_k is the mean and covariance of the k Th edge; $e_k(x_k,z_k)$ is the error function of x_k and z_k, it measures how well x_k matches z_k. These elements consist the global error function $F(x)$, and x^* denotes the global optimal solution. To find the solution x^*, $F(x)$ is rewrote as $F(\tilde{x}+\Delta x)\simeq c+2b^T\Delta x+\Delta x^T H\Delta x$, while $c=\sum_{k\in C} e_k^T\Omega_k e_k$;

$b = \sum_{k \in C} e_k^T \Omega_k J_k$; $H = \sum_{k \in C} J_k^T \Omega_k J_k$; $\breve{x}$ indicates the initial value of x. Δx is the residual between $\breve{x}$ and x^*, and J_k is the Jacobian Matrix of each e_k. Then we can get the global optimal solution x^* by solving the linear function $H\Delta x^* = -b$ and add Δx^* to x^* iteratively.

However, the IDs are so small on the top view image, only limited number of ID are detected, others are initially associated by Nearest Neighbor Method. Finally, all the detected ID and measurement of parking lots are added to the graph as max mixture [25] edges with multiple hypothesis including null hypothesis.

2) Max-Mixture Model: Max-Mixture Model [25] aims to detect loop closure errors in the back-end part of a pose graph. Traditionally, all distributions in a back-end graph system are considered as unimodal Gaussians, thus wrongly associated datum result in great global errors. When sum-mixture model replaces unimodal Gaussian, wrongly associated data can be suppressed by other mixture elements. To simplify the problem, the sum is substituted by a max operator. When an uncertain loop closure occurs, a max-mixture of several elements is thus added to the graph: several possible loop closure alternatives and a null hypothesis indicating that all the former alternatives are wrong. A null hypothesis is quite effective even with a extremely small weight.[25] This method works well in pose graphs[27][28][17], and we will prove its effectiveness in landmark association in the experiment section.

Since there is no wrongly associated data in IMU and tag observations, they are considered as robust edges in the graph, while the parking lot observations are always with high uncertainty, and several steps are added to guarantee the robustness. Once parking lots are detected, they are re-examined using some prior knowledge. Unusual lots(too small, too large or with abnormal length-width ratio) will thus be discarded. The rest lots are classified by the confidence level of lot IDs. Lot observations with high confidence IDs are added to the graph as Max-Mixture edges with null hypothesis. Due to the existence of false positive IDs with high confidence, a null hypothesis indicating the wrong edge association is essential. The location of lot observations with low confidence IDs and without ID are matched with all the excited lot landmarks by nearest neighbour method. As several candidate lots are detected, these candidate lot association together with the original association obtained by lot detection are add to the graph as a Max-Mixture edge with multi hypothesis. If no candidate detected, the lot observation will be pushed into the temporary landmark pool. In case the certain number(M) of same landmark is collected in the pool, the M observations are added to the graph as a max-mixture edge with multi hypothesis. The best car lot association will be selected as the initial landmark observation. Information matrix of landmark edges are shifted according to their detection confidence level.

The vertices collection of the graph includes vehicle position at every time stamp and all landmark(tags and lots) positions. The graph is optimized by Gauss-Newton Method once a new landmark edge is added. All the optimized landmarks consists of the initial 2D map.

In the online part, the well-known Extended Kalman Filter(EKF)[10] is introduced to filter the inconsistency in localization results from various observation.

3) Extended Kalman Filter: Extended Kalman Filter[10]is a classical back-end choice in SLAM field. Since there are errors in all the observations, EKF is used to determine to which extent the observations can be believed. The following equations express the whole optimization procedure:

$$\hat{x}_{k|k-1} = f(\hat{x}_{k-1|k-1}, u_{k-1})$$
$$P_{k|k-1} = F_{k-1} P_{k-1|k-1} F_{k-1}^T + Q_{k-1}$$

$$\widetilde{y}_k = z_k - h(\hat{x}_{k|k-1})$$
$$S_k = H_k P_{k|k-1} H_k^T + R_k$$
$$K_k = P_{k|k-1} H_k^T S_k^{-1}$$
$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k \widetilde{y}_k$$
$$P_{k|k} = (I - K_k H_k) P_{k|k-1}$$

$x$is the state matrix(containing real-time car pose and positions of all landmarks); $\hat{x}_{k|k-1}$denotes the calculated from the former state, $\hat{x}_{k-1|k-1}$ while denotes the former state matrix. z_k denotes the observations at k time stamp, and u_{k-1} is the control matrix(indicating IMU datum and real-time control of the vehicle control system). P is the corresponding covariance matrix of x, while F_{k-1} and H_k are the Jacobian matrix of $f(\hat{x}_{k-1|k-1}, u_{k-1})$ and $h(\hat{x}_{k|k-1})$. Both Q and R are pre-given noise matrix. Once updated, the state and covariance matrix are switched to fit current time stamps observations.

We discard the graph optimization method to attain a smoother vehicle localization trace, which is meaningful for the later control part. Except for lot and tag observations, the car direction and speed from both IMU and vehicle control system (the steering wheel) are also added to the filter. This combination enables a more robust localization performance in various harsh situations(etc. long term localization under limited tags or lots with large direction drifting in IMU, which is discussed in the Experiments part). Since the covariance matrix increases quadratically with the number of landmark[10], local map is introduced to achieve real-time performance. Those landmarks who has not been observed for a period of time is removed from the local map. The online localization can start from either an empty map or the initial parking map built in the offline stage. However, due to the local map strategy, the online lot map starting from an empty map is to improved.

III. EXPERIMENTS

In this section, we test our method both online and offline, and discuss several questions about the usage of visual markers. All the parking lot datasets are collected by Roewe E50 in Jiading Campus of Tongji University.

A. Semantic landmark detection and localization

Fig. **??** show a subset of the training samples used.

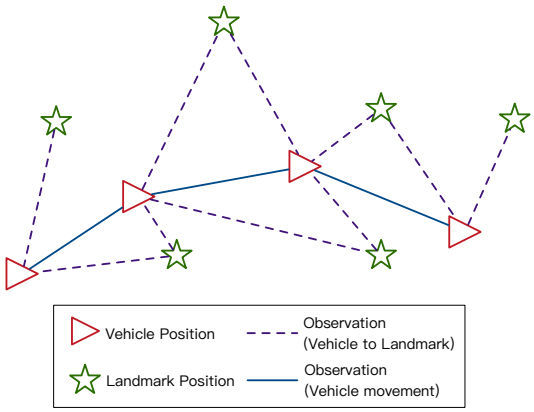

Fig. 6. The connection between the vehicle at each time stamp and the landmarks.

B. Online real-time mapping and localization

During the online part, the vehicle is first operated by human to initialize the parking map. Once the map stabilizes, a car trace is recorded. Then the vehicle drives automatically at the speed of 3 km/h following this pre-recorded trace and fixes its direction constantly according to real-time localization record. The automatic driving procedure is repeated 10 times, automatic driving traces are also recorded and compared to the pre-recorded one. Fig 11 shows the traces of both manual and automatic driving trace. Quite a number of visual markers(60 tags) are used to cover the lot-free parts near the entrance and to guarantee a fully stable and credible localization result. Each tag is printed on a A2-size paper with 48.8 cm side length. While observing, those tags who are 20 meters or farther than the vehicle are discarded since the accuracy decreases as tags become smaller in the image. However, observations with dip-angle are reliable constraints as long as they are within 20 meters.

Fig. 7. The dip-angle doesn't affect the accuracy while the distance makes great influence.

Fig. 8. A comparison between the human-driving and automatic driving trace

C. Offline mapping test

As to the offline part, several datasets with different starting points are collected. Each dataset is processed independently. The covariance of each edge shifts from 0.1 to 0.25 according to its confidence level. Fig 12 shows the map result for two of these datasets. Map blocks with more tags have higher precision as tags are ideal landmarks who will never be wrongly associated. The low detection rate of parking ID also affects the mapping result, as shown in fig 13.

In one dataset, there is a no-tag map block (marked in fig 12), where wrongly associated data always occurs. However, once a loop is detected, all landmark-positions are corrected. Then, duplicate landmarks are merged.

D. How many tag is needed?

As described in the previous section, tags are ideal landmarks promising a robust localization and mapping result. In this part, we discuss which tags are indispensable while others are not a must.

We think several factors may affect the importance of each tag: the visibility rate, the distribution and the distance to the nearest rate. Considering the factors above, we make several tag combinations and test their performance. The final result is shown in fig 14, both visibility and distribution play an important role. The crucial tags should be separated in the lot, and have a good visibility rate. In our parking lot case, at least 3 tags are needed, while 10 evenly distributed tags offer the best result.

a statistic result of the rightly located lots in each graph give the tag position solution based on the visibility of cameras based on visibility of the front camera(put tags at where there are multi visibility) be separated in the lot compare the solution to the statistic result very similar

(john: should add Tags can be replaced by instruction arrows or parkinglot ids on the pillar, here restricted by the goal and the environment tags are required)

IV. Conclusion

The conclusion goes here.

Appendix A
Proof of the First Zonklar Equation

Appendix one text goes here.

Appendix B

Appendix two text goes here.

Acknowledgment

The authors would like to thank...

References

[1] J. Engel, T. Schps, and D. Cremers, "Lsd-slam: Large-scale direct monocular slam," vol. 8690, pp. 834–849, 2014.

[2] C. Forster, M. Pizzoli, and D. Scaramuzza, "Svo: Fast semi-direct monocular visual odometry," pp. 15–22, 2013.

[3] G. Younes, D. Asmar, and E. Shammas, "A survey on non-filter-based monocular visual slam systems," 2016.

[4] S. Houben, M. Neuhausen, M. Michael, R. Kesten, F. Mickler, and F. Schuller, "Park marking-based vehicle self-localization with a fisheye topview system," *Journal of Real-Time Image Processing*, pp. 1–16, Sep. 2015.

[5] H. Grimmett, M. Buerki, L. Paz, and P. Pinies, "Integrating metric and semantic maps for vision-only automated parking," in *IEEE International Conference on Robotics and Automation*, 2015, pp. 2159–2166.

Fig. 9. toDo

Fig. 10. toDo

Fig. 11. toDo

[6] M. Himstedt and E. Maehle, "Online semantic mapping of logistic environments using rgb-d cameras," vol. 14, no. 4, p. 172988141772078, 2017.

[7] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. J. Leonard, "Past, Present, and Future of Simultaneous Localization And Mapping: Towards the Robust-Perception Age," *arXiv.org*, Jun. 2016.

[8] A. Bansal, H. Badino, and D. Huber, "Analysis of the c-mu localization algorithm under varied conditions," 2015.

[9] A. J. Davison, "Real-time simultaneous localisation and mapping with a single camera," in *IEEE International Conference on Computer Vision*, 2003, p. 1403.

[10] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse, "Monoslam: Real-time single camera slam," *IEEE Transactions on Pattern Analysis Machine Intelligence*, vol. 29, no. 6, p. 1052, 2007.

[11] J. Civera, O. G. Grasa, A. J. Davison, and J. M. M. Montiel, "1-point ransac for extended kalman filtering: Application to real-time structure from motion and visual odometry," *Journal of Field Robotics*, vol. 27, no. 5, p. 609631, 2010.

[12] T. Bailey and H. Durrantwhyte, "Simultaneous localization and mapping (part i)," 2006.

[13] G. Klein and D. Murray, "Parallel tracking and mapping for small ar workspaces," in *IEEE and ACM International Symposium on Mixed and Augmented Reality*, 2007, pp. 1–10.

[14] H. Lategahn and C. Stiller, "City gps using stereo vision," in *IEEE International Conference on Vehicular Electronics and Safety*, 2012, pp. 1–6.

[15] ——, "Vision-only localization," *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 3, pp. 1246–1257, 2014.

[16] R. Kmmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, "G2o: A general framework for graph optimization," in *IEEE International Conference on Robotics and Automation*, 2011, pp. 3607–3613.

[17] R. Mur-Artal, J. M. M. Montiel, and J. D. Tards, "Orb-slam: A versatile and accurate monocular slam system," *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1147–1163, 2017.

[18] R. A. Newcombe, S. J. Lovegrove, and A. J. Davison, "Dtam: Dense tracking and mapping in real-time," in *IEEE International Conference on Computer Vision*, 2011, pp. 2320–2327.

[19] X. Li and R. Belaroussi, "Semi-dense 3d semantic mapping from monocular slam," 2016.

[20] R. F. Salas-Moreno, R. A. Newcombe, H. Strasdat, P. H. J. Kelly, and A. J. Davison, "Slam++: Simultaneous localisation and mapping at the level of objects," in *Computer Vision and Pattern Recognition*, 2013, pp. 1352–1359.

[21] J. Mccormac, A. Handa, A. Davison, S. Leutenegger, J. Mccormac, A. Handa, A. Davison, and S. Leutenegger, "Semanticfusion: Dense 3d semantic mapping with convolutional neural networks," in *IEEE International Conference on Robotics and Automation*, 2017, pp. 4628–4635.

[22] S. Wang, S. Fidler, and R. Urtasun, "Lost shopping! monocular localization in large indoor spaces," in *IEEE International Conference on Computer Vision*, 2015, pp. 2695–2703.

[23] L. Li, L. Zhang, X. Li, X. Liu, Y. Shen, and L. Xiong, "Vision-based parking-slot detection: A benchmark and a learning-based approach," in *IEEE International Conference on Multimedia and Expo*, 2017, pp. 649–654.

[24] E. Olson, "Apriltag: A robust and flexible visual fiducial system," in *IEEE International Conference on Robotics and Automation*, 2011, pp. 3400–3407.

[25] S. Hong, B. Roh, K. H. Kim, Y. Cheon, and M. Park, "Pvanet: Lightweight deep neural networks for real-time object detection," 2016.

[26] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2003.

[27] M. Pfingsthorn and A. Birk, "Representing and solving local and global ambiguities as multimodal and hyperedge constraints in a generalized graph slam framework," in *IEEE International Conference on Robotics and Automation*, 2014, pp. 4276–4283.

[28] Y. Latif, C. Cadena, and J. Neira, "Robust graph slam back-ends: A comparative analysis," in *Ieee/rsj International Conference on Intelligent Robots and Systems*, 2014, pp. 2683–2690.

[29] N. Sunderhauf and P. Protzel, "Switchable constraints vs. max-mixture models vs. rrr - a comparison of three approaches to robust pose graph slam," in *IEEE International Conference on Robotics and Automation*, 2013, pp. 5198–5203.