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 parking lots contain limited features and thus, is hostile to traditional vision-only localization methods. In this paper, we propose an alternative solution for indoor localization in parking spaces. High level landmarks are introduced to avoid the limitations of traditional methods. We choose parking-lot, which act as an aid for the later parking task, as well as visual markers representing pillar walls to achieve the robust performance. To meet both human and machines demands, a 2 dimensional map is established in the offline part other than the traditional point-cloud map. During the online process, our map is updated simultaneously and tends to stabilize. Real time experiment on our test vehicle shows that our method meets the practical demands, and perform better in parking lots than other state-of-art methods.**

***Index Terms*—Indoor, lot detection, AprilTag, Graph Optimize, SLAM.**

I. Introduction

AUTONOMOUS driving has made great progress in recent years, breakthrough has been made in several harsh fields, including motion planning real-time localization and obstacle detection. Self driving car with multi sensors can perform well in urban areas. however, indoor parking without human interfere is still an unsolved question. Traditional indoor navigation methods require many pre-equipment sensors. WiFi and bluetooth signal decays while users distance to signal sources increases, so a significant number of stations are needed for stability. However, visual SLAM only need low-cost cameras and works well in most situations. Therefore, vision sensor is now a favor to car manufacturers .

However, visual SLAM is still immature. Many SLAM methods have their own scopes and may lose control(stop working or work unsteadily) in specific places, parking lots, shopping malls and etc. Traditional visual localization methods[**?**][**?**] develop from structure from motion[8], who aims to recover the 3D scenes from dozens of photos without human interfere. These methods focus on scene reconstruction, and need rich textures for feature detection, so they always fail in low-texture environments such as lot spaces. Recently, direct method has raised general interests. Direct method estimates camera poses directly from images rather than extracts features. Direct methods often require high frame rate[9][10] and are susceptible to global illumination changes[9][11], so they are often limited to room-sized domains[9][12] and are likely to lose in the map[10][11]. As the research on machine learning and pattern recognition develops, visual SLAM with semantic labels[13][14][15] emerges. These solutions use machine learning approaches to collect semantic information from image patches, and label the output maps. However, most semantic labels helps little in localization stage, while the image segmentation and classification work cost much time. The full use of semantic information is still a unsolved problem.

As a typical kind of semantic landmarks in parking spaces, parking-lots is now a favor for researchers.[16][17][18] Inspired by these methods, We present a mapping and localization system based on the robust recognition of high level landmarks for parking, i.e. parking-lots and visual markers representing pillar-wall. In our method, a two dimensional parking map is established other than the traditional point-cloud map for its stability, directness and light weight. The method is tested in real parking lot spaces, and proved to meet the real time usage.

II. Related works

Simultaneous localization and mapping(SLAM) has long been a heated topic in the computer vision field. Going through classical age and algorithmic-analysis age[19][20], now the association of metric and semantic map and the system performance in specific environment are the main focus in SLAM field.

Monocular SLAM methods generally fall into two groups, feature-based methods and direct methods. In feature-based methods, low-level features are detected and treated as landmarks. [21][22] and [23] use Extended Kalman Filter(EKF) to optimize the sensor and landmarks positions constantly. These methods provides promising localization results with high efficiency. However, the computation grows quadratically with the number of landmark[20], so these methods are bounded in room-sized domains. Due to the topological relation among the series of poses and landmarks, other methods(known as Graph SLAM) present them as a factor graph and optimize it with a linear solver locally or globally. The first Graph SLAM is PTAM[6], who is the first to separate tracking and mapping into two tasks, but still works in small areas(a desk or a room corner). After that, graph-based methods blooms.[24] and [25] build city-scale sparse maps using G2O, a graph optimization framework. Nevertheless, the optimization of sparse 3D point cloud is time-consuming, so the offline mapping procedure

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.

is a must. Built on the idea of PTAM[6], ORB-SLAM[7] offers a more stable and efficient graph-based SLAM system. With the keyframe and the loop closure detection, ORB-SLAM performs well in multi-scale environments. Since its a feature-based method, ORB-SLAM is still easy to get lost in less or non-textured environments. Direct methods use all the image pixels to build a dense point cloud. Comparing to feature-based methods, direct methods output a semi-dense point cloud with higher quality at a real time speed. But direct methods require high frame rate since short baselines are needed to estimate the depth accurately[11] and are not robust to motion blur, camera defocus and global illumination changes[9]. SVO[10] combines feature-based method and direct method, and runs quite fast(100Hz). However, lacking of loop closure detection, SVO drifts as time increases and gets lost easily.

Semantic SLAM associates the semantic understanding with the geometric entities in the surroundings[19]. [26] adds semantic labels to a LSD-SLAM framework, but semantic labels helps little in the localization stage. SLAM++[14] and Semantic Fusion[15] do semantic SLAM based on RGBD SLAM framework, and the semantic label aids the loop closure. However, both methods are only tested in room-sized domains. Semantic labels under specific environments also aroused attention. In [27], shop names and shop facades are recognized as labels in large indoor shopping spaces. [16] detects parking lots from the top view image, and use detected lots to estimate camera poses. [17] and [18] reconstruct the metric map and the semantic map of parking lot, which helps the route planning and parking task.

III. APPROACH

Our vision-only localization and mapping system includes four fisheye cameras and a monocular camera. Four fisheye cameras are fixed at two reflectors, the front engine hood and the trunk lid, which consist a surround-view system. A top-view image is thus fused from the surround-view inputs. In the top-view image, who indicates ground textures, parking lots are detected. The monocular camera is installed at the left of the real-view mirror to capture front scenes. The change of steering wheel angle, as well as the vehicle speed and direction collected by IMU are also used for stability.

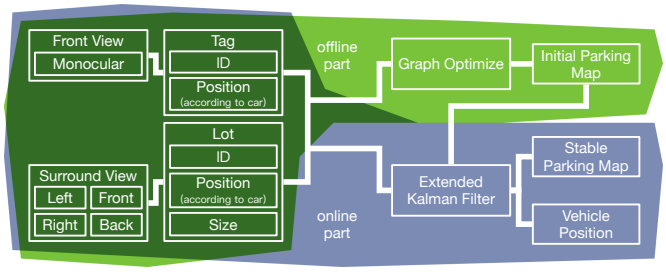

Fig. 1. Pipeline of the method

Two kinds of significant landmarks in the parking lots are used pillars(represented by visual markers) and parking lots. Visual markers are introduced as an aid for the constancy of localization since few parking lot is detected near the entrances and exits. We select AprilTags as visual markers for its robustness and high efficiency(a frame rate of 10 Hz). Parking lots are detected from the top-view image with the state of art CNN based parking lot recognition method[]. Once the four corner points are extracted, the Parking ID, which acts as an aid for the further parking task, can be determined quickly using priori knowledge.

At every time step, the relative car-landmark position, together with the speed and angular changes of vehicle should be added to the parking map incrementally. In the offline 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.

A. CNN based Parkinglot Recognition

Most traditional parking lot recognition methods use low-level features, such as corners and edges, which result in unstable and imprecise detections. CNN based method uses high-level features and offers stable and robust result. In our system, a learning based lot detection method [28]is applied.

First, marking-point patterns are detected by a binary classifier. Marking-point pattern refers to a image patch at the intersection of two parking-lines.[28] The popular AdaBoost frame work is used. A boosted classifier is made up of several weak classifiers, which, in our case, are shallow decision tree. Three types of feature the normalized intensity, the gradient magnitude and the oriented gradient histograms are used. Also, the constant soft-cascade strategy[28] is introduced for acceleration. Since there are five kinds of marking point patterns, there are five classifiers rather than one. Once a marking-point pattern is detected, it is rotated to fit the standard pattern.

Parking lots are then recognized based on the marking-point patterns. In this method, the entrance-line is especially focused. We use the entrance-line rule to make preliminary detections. A further decision is made by conforming the preliminary detection to the parking lot model.

The final step is to remove the miss-detected entrance-line, like $\overrightarrow{P_1P_3}$in Fig 5. 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. Lot IDs are also recognized basing on the priori knowledge of parking lots since IDs are always located in the center of the entrance-line. We fine tuning the PVANet[31] to detect IDs. 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[32] edges with multiple hypothesis including null hypothesis.

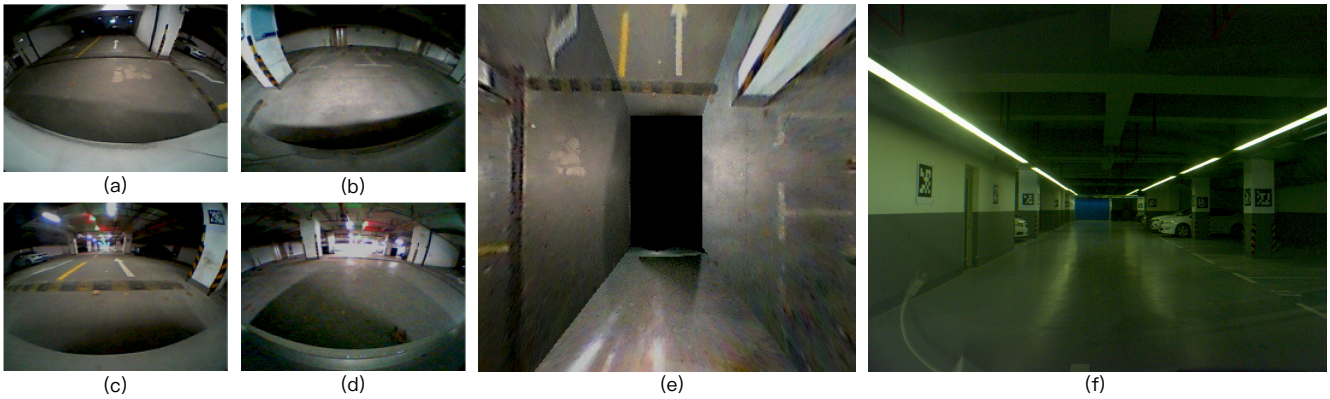

Fig. 2. (a)(b)(c)(d) are images from left, right, front, back fisheye cameras. (e) shows the top-view image fused from (a)(b)(c)(d). The image from the monocular camera is (f).

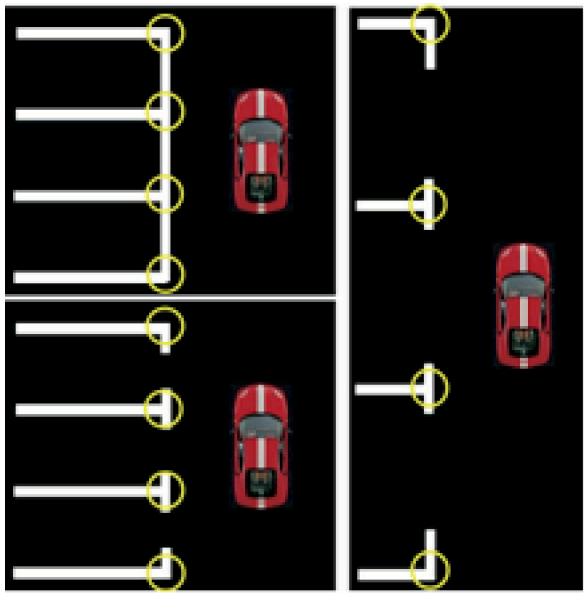

Fig. 3. [28]In this graph, yellow circles show the image patterns to be detected.

B. Visual markers/fiducial detection and Pillar wall inference

The surround view system offers an intuitional ground observation at a high frame rate, however it has limitations. The resolution of the top-view image is far from satisfactory, limiting the lot detection performance. Most part of the surround view images are discarded, only downward parts are available. And the calibration information for image fusion becomes inaccurate as time goes.

Pillar-walls are part and crucial in parking spaces. They can be used for localization, indoor structure inference and obstacle avoidance. However, pillar-wall detections using image segmentation cost much time and still offer unstable results. Though RGBD camera like Kinect can captures walls easily, they do cost a lot. Visual markers, which present pillar-walls are introduced for its robustness and high efficiency.

We use a 2D bar code style visual markerAprilTag. The detection work is completed by AprilTag C source Open Library(https://april.eecs.umich.edu/wiki/AprilTags)[29], and the relative position between visual markers and the vehicle is solved by the PnP model[8]. Every tag texture indicates a unique ID, so tag data association is always correct.

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.

C. Optimization

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[30] framework together with the max-mixture model[32] 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.

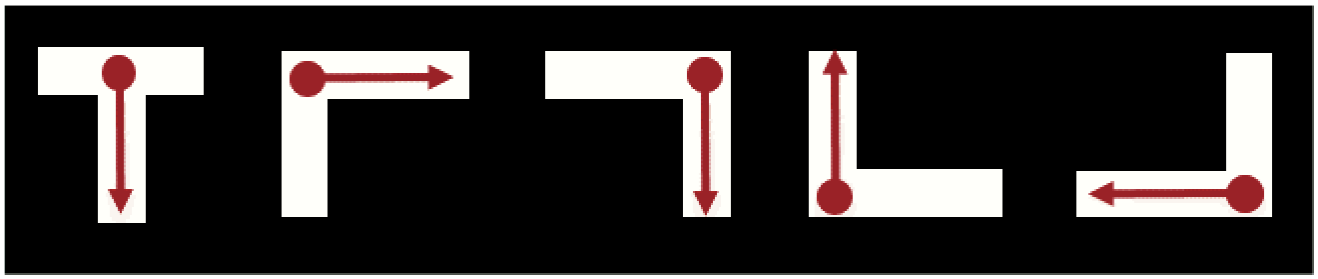

Fig. 4. [28]Examples of marking point patterns, the red arrow indicates the direction of each patterns.

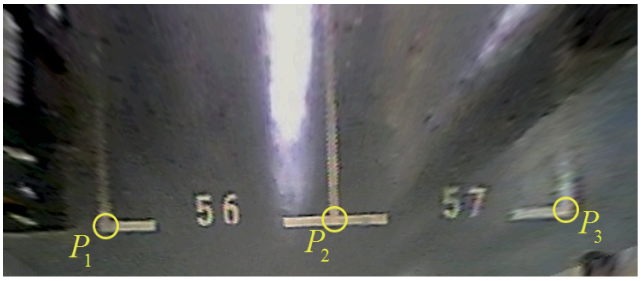

Fig. 5.

Fig. 6. An example of AprilTag

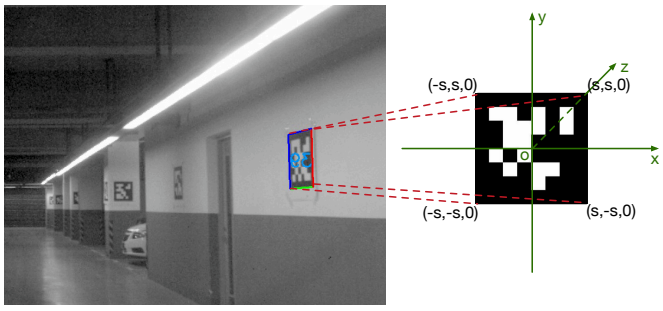

Fig. 7. An example of the relationship between a detected tag in the image plane(left figure) and its corresponding hypothetical 3D tag coordinate(right figure)

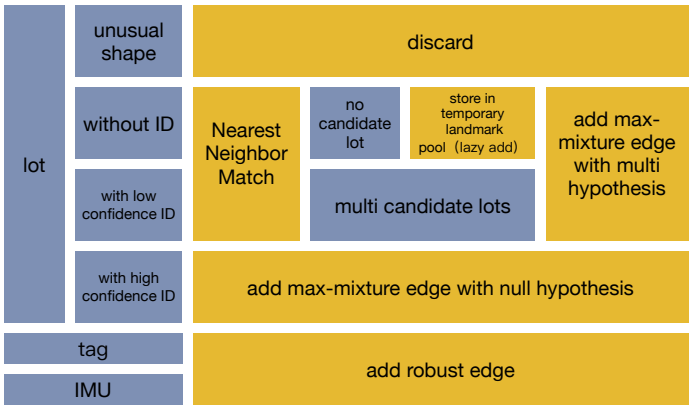

Fig. 8. The overall pipeline of the offline optimization stage

During the offline stage, we use G2O[30] 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[32] are introduced to eliminate the mistaken associations.

1) G2O Optimization Framework: G2O[30] 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:

$$F(x) = \sum_{k \in C} e_k(x_k, z_k)^T \Omega_k e_k(x_k, z_k)[30]$$

$$x^* = \underset{x}{argmin} F(x)[30]$$

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 $F(\breve{x} + \Delta x) \simeq c + 2b^T \Delta x + \Delta x^T H \Delta x$,

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.

2) Max-Mixture Model: Max-Mixture Model[32] 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.[32] This method works well in pose graphs[32][33][34], 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 reexamined 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 neighbor 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)[20] is introduced to filter the inconsistency in localization results from various observation.

3) Extended Kalman Filter: Extended Kalman Filter[20] is a classical back-end choice in SLAM field. Since

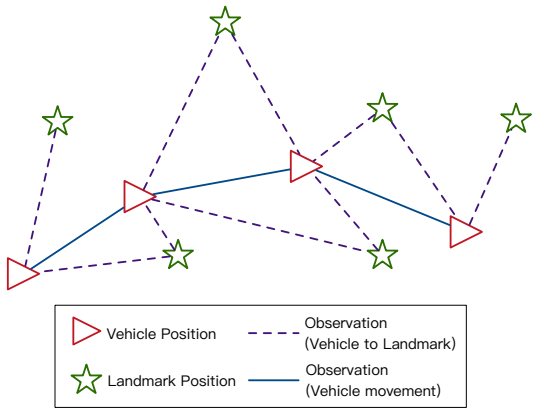

Fig. 9. The connection between the vehicle at each time stamp and the landmarks.

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[20], 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.

IV. Experiments

In this section, we test our method both online and offline, and discuss several questions about the usage of visual markers. In the online part, our system is tested on Roewe E50 in a parking lot in Tongji University. Experiment draft

describe the experiment environment where we test how much visual markers we have used how big our visual markers are the choose of visibility distance describe the online mapping and localization result(with tag) the system speed the localization result shown as a trace the localization result using a control system describe the offline mapping result(with tag) the G2O parameters the impact of loop closure describe the offline mapping result with multi level of tags show result with multi level of tags 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

V. 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...