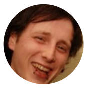
Park marking-based vehicle self-localization with a fisheye topview system

Article *in* Journal of Real-Time Image Processing · January 2016

DOI: 10.1007/s11554-015-0529-z

Sebastian Houben
University of Bonn
30 PUBLICATIONS **243** CITATIONS
SEE PROFILE



Marcel Neuhausen
Ruhr-Universität Bochum
9 PUBLICATIONS **45** CITATIONS
SEE PROFILE

Matthias Michael
Ruhr-Universität Bochum
5 PUBLICATIONS **30** CITATIONS
SEE PROFILE

Florian Mickler
Audi Electronics Venture GmbH, Gaimersheim
1 PUBLICATION **1** CITATION
SEE PROFILE

Journal of Real-Time Image Processing manuscript No.
(will be inserted by the editor)

Sebastian Houben · Marcel Neuhausen · Matthias Michael · Robert Kesten · Florian Mickler · Florian Schuller

Park Marking Based Vehicle Self-Localization with a Fisheye Topview System

Received: date / Revised: date

Abstract Accurately self-localizing a vehicle is of high importance as it allows to robustify nearly all modern driver assistance functionality, *e.g.* lane keeping and coordinated autonomous driving maneuvers.

We examine vehicle self-localization relying only on video sensors, in particular, a system of four fisheye cameras providing a view surrounding the car, a setup currently growing popular in upper-class cars. The presented work aims at an autonomous parking scenario. The method is based on park markings as orientation marks since they can be found in nearly every parking deck and require only little additional preparation.

Our contribution is twofold: 1) We present a new real-time capable image processing pipeline for topview systems extracting park markings and show how to obtain a reliable and accurate ego pose and ego motion estimation given a coarse pose as starting point. 2) The aptitude of this often neglected sensor array for vehicle self-localization is demonstrated. Experimental evaluation yields a precision of 0.15 m±0.18 m and 2.01 °±1.91 °.

The final publication is available at
`http://link.springer.com/article/10.1007/s11554-015-0529-z`

S. Houben, M. Neuhausen, M. Michael
Institut für Neuroinformatik, Ruhr-Universität Bochum
Universitätsstraße 150, Bochum
Germany
tel: +49(0)234 - 32 - 25566
fax: +49(0)234 - 32 - 14210
E-mail: sebastian.houben(at)ini.rub.de

R. Kesten
GIGATRONIK Ingolstadt
Am Augraben 19, Gaimersheim
Germany

F. Mickler, F. Schuller
Audi Electronics Venture GmbH
Sachsstraße 20, Gaimersheim
Germany

1 INTRODUCTION

Today's vehicles come with a large number of driver assistance systems. These include lane keeping, adaptive cruise control, and recognition of obstacles, traffic participants, and further relevant objects like traffic signs and traffic lights. The systems are capable of partly deriving the current traffic regulations and pointing them out to the driver if necessary. Further expanding their areas of application with the long-term objective of fully autonomous driving sets a great challenge. One promising approach to achieve this is to use static map data and information from infrastructure facilities, *e.g.*, surveillance cameras, and other vehicles via car-to-X communication.

To fuse and later utilize this data, the knowledge of the vehicle's accurate position and orientation is essential. We regard this problem in potentially roofed environments, where GNNSs like GPS are not available, and make use of common landmarks that we detect with a camera array of four fisheye cameras providing a view surrounding the whole vehicle. Our work covers a typical example of an all-over infrastructure controlled driving scenario, namely fully autonomous parking in a prototype parking deck.

We start with a detailed description of the scenario we want to investigate (Sec. 2) and juxtapose it to the techniques which are currently followed in the literature (Sec. 3). We present an image processing algorithm (Sec. 4) for the topview system that is independently executed for each of the four camera's input images. We name and address the problems arising from the nonperfect detection results and construct a Kalman filter that estimates both the vehicle's pose and its motion (Sec. 5). Implementation remarks and real-time considerations are provided (Sec. 6). The entire system is analyzed thoroughly with the help of a highly accurate LIDAR-based positioning system as reference (Sec. 7). Likewise we examine its robustness to noisy initial poses and evaluate the landmark detector itself with respect to precision, recall, and range-dependency. We conclude

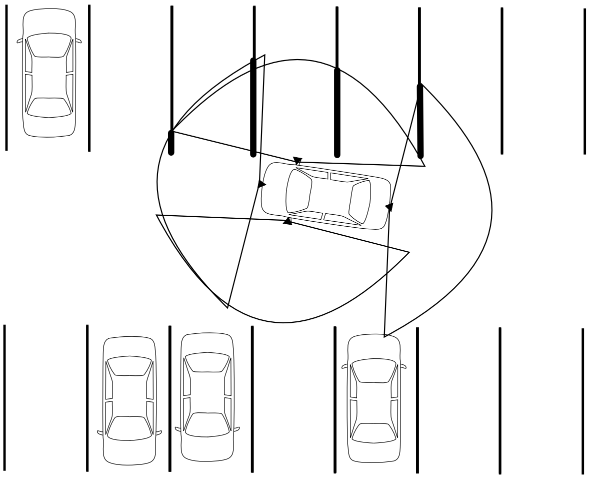


Fig. 1 The problem at hand: The ego-vehicle is equipped with an array of four fisheye cameras detecting park markings in its near vicinity. Comparing those to a map of the parking deck allows for a position estimation.

this paper with a discussion (Sec. 9) on the advantages and drawbacks of the proposed method and map out ways for further improvement (Sec. 8).

2 SETUP

Figure 1 shows an overview of the setup that we elaborate on in this paper. A vehicle is deployed with fisheye cameras that are installed at the two side mirrors, the trunk lid, and the front bumper. We refer to this sensor array as topview system. The choice of fisheye cameras as sensor is plausible as it provides a wide-angle view and, thus, localization of detections near and all around the vehicle. Although we will focus in particular on a four-camera fisheye setup, the methods and results can be easily transferred to other systems permitting a view surrounding the entire vehicle. Since these often rely on wide-angle or fisheye cameras as well, we specifically address the aptitude of those objectives for self-localization.

The landmarks that satisfy our requirements of being already present, easily detectable, and widely used are park markings, *i.e.* the straight bright lines separating the parking lots. They can be described formally by their two end points and, thus, have a very simple representation within the map which the parking deck's control system will have to transmit to the vehicle. On the other hand, the markings are not unique. A self-localization will, thus, only be possible if a coarse vehicle pose is given beforehand. We will, however, demonstrate that the use of park markings as landmarks can be sufficient to perform a frame-by-frame egomotion estimation and position tracking from a single initial vehicle pose. Figure 2 shows the processing pipeline that we will follow throughout the paper.

3 RELATED WORK

The project that we want to embark on touches several fields that we address in the following.

A large body of literature covers the detection of road markings with mono cameras. These markings resemble park markings in visual appearance, but may be curved. Yenikaya et al (2013) provide a broad survey of the more recent approaches. Veit et al (2008) compare local features that are mostly based on relative brightness in order to extract road markings on a per-pixel basis. McCall and Trivedi (2006) present a mature video-based lane departure warning system incorporating a marking detection, lane curvature estimation, and tracking.

Regarding the probabilistic approach for positioning and motion estimation we cite Thrun (2002) as introductory reading and Haykin (2001) and Grewal and Andrews (1993) for the mathematical basics on temporal filtering. Schubert et al (2008) discusses the feasibility of different motion models for passenger cars. Particularly tailored to self-localization and mapping are the survey by Thrun et al (2002) and the more recent papers by Grisetti et al (2010), Lim et al (2011), and Engel et al (2014) that cover different views, approximations, and improvements of the initial formulation.

Vehicle self-localization with video sensors is in the first place considered for refinement of GNSS (*e.g.* GPS) positioning. Lategahn et al (2013) and Laneurit et al (2005) rely on local image features and an IMU to localize a vehicle in urban canyons where GPS is often unreliable. Chausse et al (2005) combine GPS, odometry, and lane markings that are detected with a monocular camera. Levinson et al (2007) propose to detect these markings by the reflection of infra-red light, building maps with the help of a specially prepared vehicle beforehand. Likewise, the works by Pink (Pink 2008; Pink et al 2009) provide a detailed description of a system leveraging road markings for vehicle self-localization. In a later paper (Pink and Stiller 2010), a system for deriving a precise map of possible markings with the help of aerial images is presented. Similarly, Schreiber et al (2013) propose to perform self-localization via detection of lane markings and curbs that they detect with a stereo camera pair. Wu and Ranganathan (2013) achieve a very accurate localization in the vicinity of special road markings like direction arrows and other symbols on the lane. Finally, we cite Schindler (2013) who allows for a self-localization approach via fitting to a detailed map which holds all road marking as arc splines. The use of high-level map data is also promoted in Oberlander et al (2014) in order to achieve sensor-independence.

The literature covering localization for passenger cars in indoor or outdoor-indoor scenarios, on the other hand, builds on very diverse methods. Several approaches rely on radio or network signals (Fernandez-Madrigal et al 2007; Gonzalez et al 2007). Kiwan et al (2012) discusses the optimal placement of base stations for this purpose.

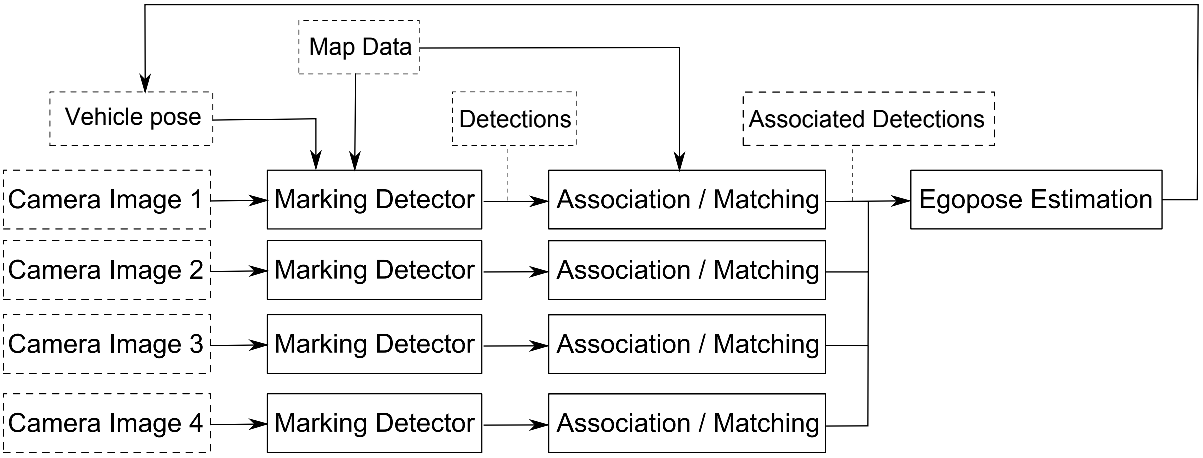


Fig. 2 Data flow diagram of the proposed self-localization setup: Each camera image is searched for park markings (*cf*. Sec. 4) using the currently estimated vehicle pose and the map data from the parking deck for presegmentation. The detections are matched with the map data (*cf*. Sec. 5.2) and used for updating the vehicle pose (*cf*. Sec. 5.3). The detection and association stage can be (and are) performed in parallel.

Due to the restricted routes a vehicle can follow indoors the use of RFID tags is also discussed (Saab and Nakad 2011; Lakafosis et al 2010). Other forms of outside-in systems for vehicle positioning include the use of LiDAR (Ibisch et al 2013) or surveillance cameras (Ibisch et al 2014). Self-localization has been achieved via vehicle-mounted laser range scanners (Kümmerle et al 2009, 2008). Literature that promotes the use of close-to-production vehicle sensors are Bojja et al (2013) and Wagner et al (2010) using odometry and near-range obstacle detection for positioning. The authors report accuracies of 2.5m – 5m which is sufficient for navigation but to low to perform autonomous maneuvers.

Park markings are, in a sense, specialized road markings as they have a given length, road markings may be continuous, and are not curved. While their finite dimensions allow for an absolute positioning as will be pointed out later, the straightness enables us to use specialized features for a more reliable detection. We regard the localization problem with a topview camera system, which is unusual but bears some advantages. Park markings are sparsely distributed and similarly oriented making their use for pose estimation ambiguous and, thus, difficult. At the same time the driving maneuvers will be slower but more complex which justifies the need for a detection of landmarks all around the vehicle.

4 PARK MARKING DETECTION

Video-based detection of road markings with today's real-time capable image processing methods cannot, despite all efforts, be performed reliably (Yenikaya et al 2013) although the markings have outstanding characteristics in order to facilitate their perception: they are a) uniform b) bright c) straight lines with d) constant width e) on the ground plane. Most methods, like pixel-based segmentation or Hough transform, are fixed on a single one of these features, like brightness or straightness, respectively, thus, making it difficult to embrace the others. On the other hand, an algorithm respecting all features would have to traverse a large search space which is unlikely to be performed in real-time. In the presented specialized scenario, it is, however, possible to restrict the search space by taking advantage of the vehicle's coarse pose and backprojecting the park markings from their known map position into the camera images (*cf*. Fig. 3 (a)). A sufficient region around the predicted markings yields a useful presegmentation. Here, we present a partly heuristic image processing pipeline considering all the features a) - e) in order to arrive at a sufficiently reliable detector.

In the following, we assume that the lens distortion function and the extrinsic calibration, *i.e.* the position and orientation w.r.t. the vehicle, of all cameras are known. The former allows us to map an image coordinate pair to a view ray emanating from the camera origin, the latter yields the camera position and orientation with respect to the vehicle coordinate system. We refer to section 5.1 for the formal details.

The presented pipeline consists of five stages (Fig. 3):

1. Transform the input image a) to a ground plane representation b) (Sec. 4.1)
2. Determine symmetric structures c) within a given width interval storing information about brightness, width, orientation, and strength of the symmetry (Sec. 4.2)
3. Join nearby symmetry pixels to short line segments d) passing the stored pixel-wise features along (Sec. 4.3)
4. Cluster these line segments to the final detections e) which use the statistics to compute a confidence value

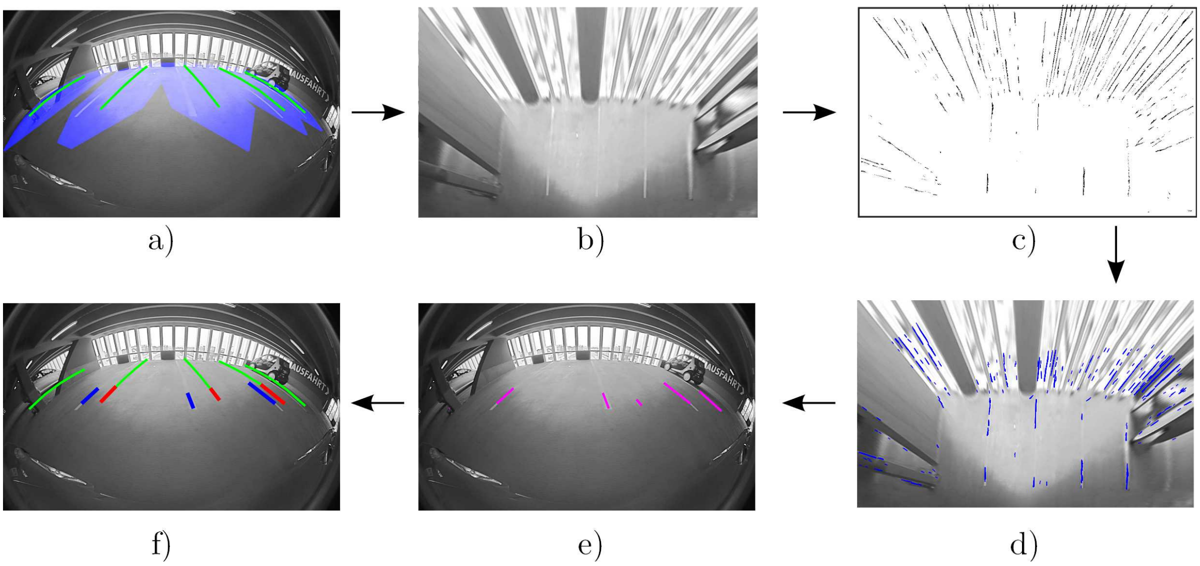


Fig. 3 The proposed image processing pipeline: a) the input image (right side mirror camera) with the backprojected park markings providing a search region (light blue) for further detections, b) the input image transformed to the ground plane, c) visualization of the symmetry strength for every pixel, d) result of initial clustering into smaller line segments, e) result of clustering to final detections, f) associations to backprojected park markings

5. Associate these detections with the predicted position of the park markings within the camera image f) (Sec. 5.2)

4.1 Ground plane representation

In order to deal with the strong image distortion that the wide-angle cameras exert, the camera image is transformed into a ground plane image, also known as birdseye view. It provides an often used representation to facilitate the detection of ground-based objects since park markings that appear curved in the input image become rectilinear in this representation. With special regard of the used wide-angle cameras it is worth mentioning the advantage of applying a direct image transformation instead of undistorting the image in 2D and subsequently applying a homography. While this does not apply for moderately distorting camera lenses, the effects on fisheye cameras can be significant since the undistortion in 2D might interpolate substantially between pixel positions, thus, providing already imprecise input for the following homography.

Since most of the algorithm is based on the ground plane representation, in the remainder, operations are assumed to be performed in this coordinate system unless stated otherwise.

4.2 Symmetry detection

Albeit simple, the search for symmetry in images is still an often adopted method for presegmenting image regions, *e.g.*, for further object detection. In the present case, symmetry is one of the core features. We, hence, set up a symmetry detector that gathers additional characteristics on the observed structures in the process. In the beginning, after contrast-equalizing the birdseye view, its gradient images in x- and y-directions are computed and a non-maxima suppression is performed leaving only the most pronounced edges from the image and, thus, speeding up the subsequent symmetry search. Although the resolution varies due to irregular sampling of the camera image, it suffices to compute the gradient of the birdseye view instead of performing the ground plane transform on the gradients of the camera image itself and applying Leibniz' rule. The following description is formalized in algorithm 1.

For each pixel p with adequate gradient magnitude, a nested search is initiated that follows the direction given by the gradient. A pixel q with a likewise strong gradient forms a symmetry pair with p if their orientation differs by a vicinity of π, *i.e.* if they are nearly converse. The directed search can be performed by the highly efficient Bresenham sampling.

In order to gather the aforementioned characteristics about the symmetry pair (p, q), which are later used to support the clustering, online mean and variance estimators are deployed for selected features along the entire symmetry detection stage. Let I be the gray value image, ∇I its gradient image, $p = p_1, ..., p_m = q$ the Bresenham-sampled pixel positions and $c = round\,(0.5(p+q))$ the symmetry pair's centre. We define the following quantities in order to propagate the substantial features of each symmetry pixel:

- *Line intensity*: The marking is assumed to be bright and distinct from its environment. The intensity $L := I(c)$ at the center of the symmetry pair is, hence, an important feature.
- *Line intensity variance*: Likewise, the region between the two pixels defining the symmetry is expected to be of constant brightness. For each directed search, the gray value variance $V := Var_{k=1,\dots,m}(I(p_k))$ is, thus, estimated as a measure of brightness uniformity.
- *Average gradient strength*: As a measure for the distinctness of the symmetric structure, the mean gradient magnitude $G := Mean(||\nabla I(p)||, ||\nabla I(q)||)$ of the symmetry pair is assigned to their central position. The value is averaged over all symmetry pairs that share the same center pixel c.
- *Average symmetry distance* $S := ||p - q||$: The markings' width is assumed to be constant. In order to use this information in later algorithm stages, the distance of the symmetry pixel pair is assigned to the center pixel and averaged analogously to the average gradient strength.
- *Average symmetry orientation*: Also, with the marking's rectilinearity in mind, the orientation $O := \nabla I(q) + (-\nabla I(p))$ of the symmetry pixel pair is assigned to and likewise averaged in the center pixel.

In conclusion, one obtains a symmetry image with the most distinct symmetric structure where each valid pixel carries the aforementioned statistics.

4.3 Initial Clustering

As a method to efficiently extract the most plausible line segments, we propose an initial clustering method based on pixel-wise neighborhood relations resembling a *region growing method*. Starting from a pixel in the symmetry image, *i.e.* a center of at least one symmetry pair from the previous stage, that has not been clustered so far and carries a sufficiently large average gradient strength and a sufficiently low line intensity variance, the method performs a region growing in direction of the average symmetry orientation likewise by Bresenham's sampling algorithm (*cf*. algorithm 2). The adjacent pixel is added to the current candidate line segment if it is no member of another line segment and carries similar line intensity L, average symmetry distance S, and average symmetry orientation O. Similar to the individual center pixels, the following statistical measures are computed on each cluster and updated for each new pixel that is added to the line segment:

- *Average symmetry distance* $\bar{S} := Mean(S)$
- *Average line intensity* $\bar{L} := Mean(L)$
- *Average line intensity variance* $\bar{V} := Mean(V)$
- *Average symmetry orientation covariance* $\bar{O} := Cov(O)$

Algorithm 1: Symmetry detection

```
Input :
  – Gradient image $\nabla I = (I_x, I_y)$ of ground plane representation
  – Magnitude threshold $\theta$, orientation threshold $\alpha$, and maximum symmetry width $\tau$
Output:
  – $L(c)$, $V(c)$, $G(c)$, $S(c)$, and $O(c)$ for every detected symmetry pixel $c$
for $p = (x, y)$ pixel position in $\nabla I$ do
    if $||\nabla I(p)|| > \theta$ then
        Init Bresenham's algorithm at $p$ in direction $\nabla I(p)$
        for $q$ as next pixel according to Bresenham do
            if $||p - q|| > \tau$ then
                break
            Update online variance $\tilde{V}$ with $I(q)$
            if $||\nabla I(q)|| > \theta$ AND $|\measuredangle(\nabla I(p), \nabla I(q)) - \pi| < \alpha$ then
                Compute center pixel $c := round(0.5(p + q))$
                Compute $\tilde{L}$, $\tilde{G}$, $\tilde{O}$, and $\tilde{S}$ via definitions of sec. 4.2
                Update online averages $L(c)$, $G(c)$, $S(c)$, $O(c)$, and $V(c)$ at $c$ with $\tilde{L}$, $\tilde{G}$, $\tilde{O}$, $\tilde{S}$, $\tilde{V}$
```

In this way, the statistics are propagated to the final clustering stage. The region growing process is able to close smaller gaps by a continuously updated penalty value that is incremented for every improper pixel and decremented for every fitting pixel. The amount depends on the feature the mismatch occurred on: G, L, V, S, or O. If it exceeds a given threshold, the growing process is stopped.

This primary clustering is mostly heuristic. In particular, pixels are not assigned to the line segment that they match best, but the one that considers them first and sufficiently matches the requirements. Likewise, the use of a penalty value provides a rather crude model in order to control the line growing. The overall procedure is, however, fast and efficient enough to be applied in a real-time capable system.

4.4 Line Segment Clustering

The final stage is a single-linkage clustering of the line segments from the previous step that exceed a given length τ_{Len}. Each such cluster is represented by a single line which we obtain via the principal component of all clustered lines' endpoints. Furthermore, the characteristics $\bar{S}, \bar{L}, \bar{V}, \bar{O}$ are propagated from the line segments to the encompassing clusters. Let $C^{(1)}, C^{(2)}$ be two separate clusters considered for linkage, $C_p^{(i)}, C_q^{(i)}, i = 1, 2$

```
Algorithm 2: Initial clustering
Input :
  – $L(c)$, $V(c)$, $G(c)$, $S(c)$, and $O(c)$ from symmetry
    detection for each symmetry pixel $c$
  – Thresholds $\tau_G$, $\tau_V$, $\tau_L$, $\tau_S$, $\tau_O$, $\tau_{pen}$
  – Penalties $\phi_L$, $\phi_S$, $\phi_O$, $\phi_{gap}$, $\phi_{hit}$
Output:
  – Found line segments
  – Start and end points $a(l)$, $b(l)$ and $\bar{L}(l)$, $\bar{V}(l)$, $\bar{S}(l)$, and
    $\bar{O}(l)$ for every line segment $l$
for $c$ set by symmetry detection with $G(c) > \tau_G$ AND
$V(c) < \tau_V$ AND yet unmarked do
    Start new line segment $l$ and include $c$
    for $s = \{-1, 1\}$ do
        Init Bresenham's algorithm at $c$ in direction
        $\measuredangle(O(c)) + s\frac{\pi}{2}$
        Init penalty $P := 0$
        for $q$ as next pixel according to Bresenham do
            if $G(q) > \tau_G$ AND $V(q) < \tau_V$ AND $q$ yet
            unmarked then
                if $|L(c) - L(q)| < \tau_L$ AND
                $|S(c) - S(q)| < \tau_S$ AND
                $|O(c) - O(q)| < \tau_O$ then
                    Include $q$ into $l$
                    Update $\bar{L}(l)$, $\bar{V}(l)$, $\bar{S}(l)$, $\bar{O}(l)$
                    Update end points $a(l)$, $b(l)$
                    $P := P - \phi_{hit}$
                    Mark $q$
                else
                    if $|L(c) - L(q)| \geq \tau_L$ then
                        $P := P + \phi_L$
                    if $|S(c) - S(q)| \geq \tau_S$ then
                        $P := P + \phi_S$
                    if $|O(c) - O(q)| \geq \tau_O$ then
                        $P := P + \phi_O$
            else
                $P := P + \phi_{gap}$
            if $P > \tau_{pen}$ then
                break
```

both of their respective representing lines' endpoints and $C_{\bar{S}}^{(i)}$, $C_{\bar{L}}^{(i)}$, $C_{\bar{V}}^{(i)}$, $C_{\bar{O}}^{(i)}$, $i = 1, 2$ the propagated characteristics. The decision whether or not to continue clustering is based on the magnitude of the following quantities:

- *Distance in image coordinates:*
  Clustered line segments should lie in local distance. The quantity $min(||C_p^{(1)} - C_p^{(2)}||, ||C_p^{(1)} - C_q^{(2)}||, ||C_q^{(1)} - C_p^{(2)}||, ||C_q^{(1)} - C_q^{(2)}||)$ is to be restricted.
- *Orientation deviation:*
  They should be similarly oriented. The orientation $\measuredangle(C_p^{(1)} - C_q^{(1)}, C_p^{(2)} - C_q^{(2)})$ is, thus, thresholded.
- *Difference of their average symmetry distance:*
  This ensures that the line segments belong to symmetric structures with similar width. The measured width deviation $|C_{\bar{S}}^{(1)} - C_{\bar{S}}^{(2)}|$ is supposed to be small.
- *Difference of their average line intensity:*
  The line segments originate from markings with similar brightness. The brightness deviation $|C_{\bar{L}}^{(1)} - C_{\bar{L}}^{(2)}|$ is restricted.
- *Difference of their average line intensity variance:*
  The brightness variation is supposed to be similar. The difference in brightness variance $|C_{\bar{V}}^{(1)} - C_{\bar{V}}^{(2)}|$ should not exceed a given threshold.

In order to arrive at a confidence for each of the final clusters, the propagated measures are processed by a logistic regression and mapped to a value between 0 and 1 as in,

$$\begin{aligned}&(1 + exp\{b + w_S C_{\bar{S}} + w_L C_{\bar{L}} \\ &+ w_V C_{\bar{V}} + w_O C_{\bar{O}} + w_i l_i + w_g l_g\})^{-1}\end{aligned} \qquad (1)$$

for chosen weights w_S, w_L, w_V, w_i, w_g and an offset b. Additionally, the regression encompasses the length of the detected line in image l_i and ground plane l_g coordinates as further important indicators. This provides an easy and reliable measure to decide which detections can be used for the following self-localization stage.

4.5 Parameter choice

The presented image processing pipeline depends on 40 parameters. Only some of them are addressed explicitly in the preceding explanations: thresholds for pairing symmetry pixels, penalty terms for clustering of line segments, weights for the distance computation of the final clustering, and weights for the logistic regression providing the confidence. Few of them relate to physical characteristics, *e.g.*, the interval of the symmetry pair distance corresponds to the width of a park marking. For the choice of most parameters no reasonable heuristic can be found. Therefore, we set up a Covariance Matrix Adaptation (CMA) (Hansen 2006) optimization routine. The park markings were labeled manually in a small number of representative camera images. In order to obtain a fitness function, the detections' confidence was thresholded with different values to create a precision-recall curve (*cf*. Fig. 6). The area under this curve, a value between 0 and 1, provides a useful measure for the aptitude of the current candidate parameters.

There is no side condition on the detector's execution time. In principle, the optimization may suggest parameters with a lot of thresholded symmetry pixels entering the computationally demanding clustering stages (*cf*. 6) which would then forbid real-time applicability. As it turns out, passing only few symmetry pixels to the clustering results in more reliable detections which in turn increases the precision. Thus, this option is preferred by the optimization routine. Also, the detection length is an important criterion but not explicitly enforced within the optimization. Again, clustering detections to longer results tends to yield better precision.

5 EGO POSE ESTIMATION

Despite all effort the proposed image processing system can not be assumed to yield perfect results. Park markings may be detected only partially or missed entirely. In the design of the following ego pose estimation routine we do assume, however, that a detection either lies completely on the respective marking in the image or is a false detection. To be precise, we, hence, assume that a correct detection will not exceed the respective marking's end points. This poses an important clue for positioning as is shown in the following. The presented approach can be separated in two parts: the association part assigning the plausible detections to park markings from map data using the currently assumed vehicle pose (or discarding them if there is no plausible matching) and the estimation part computing this pose and the vehicle motion.

5.1 Mathematical formalism

The vehicle's pose at a point of time t is given by 5 parameters

$$x_t = (p_x^{(t)}, p_y^{(t)}, \phi^{(t)}, v^{(t)}, \omega^{(t)})^T \in \mathbb{R}^5 \tag{2}$$

where (p_x, p_y) is the current position in ground plane coordinates, ϕ its orientation w.r.t. the ground plane normal, v its velocity in direction of its nose, and ω its turn rate around the vehicle coordinate system's origin. Whenever the point of time is unambiguous, we drop the subscript t. The ground plane is assumed to be flat in order to constrain the number of degrees of freedom to three (2D position and orientation). Section 7.2 covers the applicability of this model. The Kalman filter (*cf.* Sec. 5.3) models the state uncertainty as a multidimensional Normal distribution which is given by the covariance matrix $P \in \mathbb{R}^{5\times5}$. Furthermore, let $R_{cam} \in \mathbb{R}^{3\times3}$ and $t_{cam} \in \mathbb{R}^3$ be the orientation and position of the regarded camera within the vehicle coordinate system where

$$p \mapsto R_{cam}(p - t_{cam}), p \in \mathbb{R}^3 \tag{3}$$

defines a rigid transformation from vehicle into camera coordinates. With $d : \mathbb{R}^3 \to \mathbb{R}^2$ as lens distortion function mapping view rays from the camera centre t_{cam} to their respective image coordinates we obtain

$$p \mapsto d\left(R_{cam}\left(R_{veh}\left(p - t_{veh}\right) - t_{cam}\right)\right), p \in \mathbb{R}^3 \tag{4}$$

to predict a point p from the map data to the respective camera. For a common choice of d we refer to Scaramuzza et al (2006). The vehicle pose t_{veh}, R_{veh} can be derived from the first three elements of x via

$$t_{veh} = \begin{pmatrix} p_x \\ p_y \\ 0 \end{pmatrix} \text{ and } R_{veh} = \begin{pmatrix} \cos(\phi) & \sin(\phi) & 0 \\ -\sin(\phi) & \cos(\phi) & 0 \\ 0 & 0 & 1 \end{pmatrix}. \tag{5}$$

In order to vice versa map a given image point $p' \in \mathbb{R}^2$ to its view ray within vehicle coordinates, consider

$$p' \mapsto R_{cam}^T d^{-1}\left(p'\right) + t_{cam}. \tag{6}$$

5.2 Association

First, a 1-on-1 association of detections with park markings cannot be deployed since a marking can encompass several partial detections. The association routine, thus, independently selects an adequate park marking for each detection or discards it if no such marking is found. Wrong associations are particularly critical as they can later introduce estimation errors which quickly accumulate over time and lead to divergence of the pose estimation. In order to find those adequate markings a distance measure is introduced incorporating the distance of the detection to the marking's expected position in image coordinates and a deviation in orientation. Hence, the current vehicle pose x is needed to predict a marking's position in the camera image and its uncertainty P in order to usefully threshold this distance measure. Therefore, if the vehicle position is uncertain, the matching will be more eager to also match to more distant line detections. Two heuristics avoid wrong associations: First, only park markings in front of the regarded camera within a given vicinity are considered. Second, the detection's distance to the second closest marking compared to its distance to the closest marking has to be large.

We present the algorithm to compute the distance from a line detection to its park marking: The use of wide-angle optics does not allow to simply project the line's end points to the park marking whose image is a curve in the camera image. It is, however, important for the later estimation stage to measure the deviation of detection and expectation in sensor, *i.e.* image coordinates. In a first step, we project the detection's end points to the marking's curve in the image.

Refer to Fig. 4 for a scheme on the geometric operations. Basically, an orthogonal projection of the detection on the regarded marking is performed in ground plane coordinates. The result is backprojected into camera coordinates.

First, the currently assumed pose x allows us to predict the camera pose and, hence, the relative position of both end points CD of a regarded park marking. In order to compute the distance from the detection with end points a and b to CD, we regard A and B, the intersection of a's and b's view rays (*cf.* eq. (6)) with the ground plane. A and B are projected onto CD obtaining A' and B'.

One has to pay attention on the limited dimensions of CD. Thus, if A' or B' do not lie between C and D, *i.e.*

$$(A' - D)^T(A' - C) > 0 \text{ or } (B' - D)^T(B' - C) > 0, \tag{7}$$

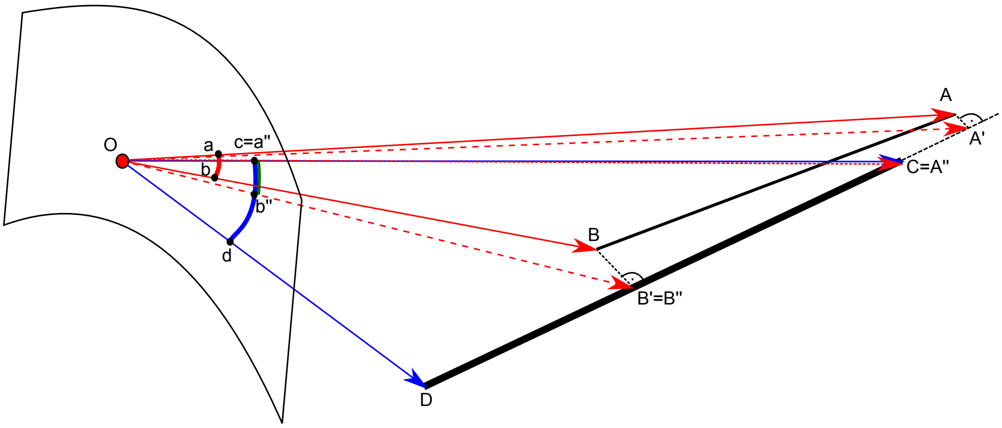


Fig. 4 Geometric operations in order to project the partial detection ab in the distorted camera image to the park marking CD in world coordinates. An example of this procedure is also sketched in Fig. 3 (f). The camera pose is obtained from the current estimated pose x.

they are assigned the position of the closer point of C and D, respectively, yielding A'' and B'' which are then backprojected via eq. (4) into the image to retrieve the final line segment $a''b''$. It is possible that $a'' = b''$. In this case the according detection-marking association is discarded.

To summarize, for a given pose x it is laid out how to project a detection to map data and, thus, how to define a distance measure between both of them. In order to also account for the present pose uncertainty P, the entire procedure is repeated for several candidate poses. As stated in line 8 from algorithm 3, we consider the sigma points via P and compute $\chi^{(k)}$ with $k = 0, ..., 10$ denoting 11 candidate poses that can be used to retrieve a''_k and b''_k for $k = 0, ..., 10$ via the procedure presented in the previous paragraph. Let further $\varphi_k := \measuredangle(b''_k - a''_k), \varphi'' := \measuredangle(a'', b'')$, and $\varphi := \measuredangle(a, b)$ be the orientation of $b''_k - a''_k$, $b'' - a''$, and $b - a$ w.r.t. the horizontal axis, respectively. We capture the distribution of the 11 backprojected line points by the covariance matrix

$$C := Cov(\{a''_0, ..., a''_{10}, b''_0, ..., b''_{10}\}) \tag{8}$$

and the orientation variance by

$$V := Var_k(\varphi_k). \tag{9}$$

The distance of the actual detection to the park marking with assumed pose x is then taken in terms of the Mahalanobis distance w.r.t. C and V:

$$(a - a'')^T C^{-1}(a - a'') + (b - b'')^T C^{-1}(b - b'') + \frac{(\varphi - \varphi'')^2}{V}. \tag{10}$$

Thus, with the presented heuristic we are able to transform the ego pose uncertainty P into C and V which provides a usable form to measure matching distances in distorted camera coordinates.

5.3 Estimation

The estimation stage computes the position and orientation of the ego vehicle in two dimensions for a fixed detection-marking association that has been determined in the previous stage (*cf*. Sec. 5.2). A Constant-Turn-Rate-Velocity-Model (CTRV) is applied to model the vehicle dynamics. Schubert et al (2008) report sufficiently precise results for CTRV compared to several more sophisticated motion models. Pose and motion parameters are combined in a single status vector which is updated by a Kalman Filter (Haykin 2001) that combines two paradigms, an Extended Kalman Filter (EKF) for transferring the status vector (prediction) and an Unscented Kalman Filter (UKF) for processing the landmark observations (update). Refer to algorithm 3 for a detailed overview.

The EKF uses the transfer function's gradient in the Gaussian propagation of uncertainty to transform the status distribution from one time step to the next with the help of a local linear approximation. The transfer function f, *i.e.* the motion model, can be expressed by an analytic representation:

$$f(x) = \begin{pmatrix} p_x + \frac{v}{\omega}(-\cos(\phi + \omega\Delta t) + \cos\phi) \\ p_y + \frac{v}{\omega}(\sin(\phi + \omega\Delta t) - \sin\phi) \\ \phi + \omega\Delta t \\ v \\ \omega \end{pmatrix} \tag{11}$$

$$Jf(x) = \begin{pmatrix} 1 & 0 & J_{13} & J_{14} & J_{15} \\ 0 & 1 & J_{23} & J_{24} & J_{25} \\ 0 & 0 & 1 & 0 & \Delta t \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \tag{12}$$

```
Algorithm 3: Ego pose estimation filter
Input:
  – Transfer function $f$ with Jacobian $Jf$ and noise covariance $Q$
  – Observation function $h$ with noise covariance $R$
  – Observations $z_t$ at time $t$
  – Initial state $x_0$ and state covariance $P_0$
1  $t := 0$
2  $x := x_0$
3  for t do
4      $t := t + 1$
5      // Prediction (EKF)
       Predict state: $\hat{x} := f(x)$
6      Transform state uncertainty:
       $\hat{P} := Jf(x)PJf(x)^T + Q$
7      Compute eigenvectors $p^{(i)}$ from $\hat{P}$
8      // Update (UKF)
       Compute sigma points:
       $\chi^{(0)} := x$
       $\chi^{(i)} := x + \sqrt{\frac{10}{1-\alpha}} p^{(i)}$ for $i = 1, ..., 5$
       $\chi^{(i)} := x - \sqrt{\frac{10}{1-\alpha}} p^{(i-5)}$ for $i = 6, ..., 10$
9      Compute weights:
       $w^{(0)} := \alpha$
       $w^{(i)} := \frac{1-\alpha}{10}$ for $i = 1, ..., 10$
10     Compute predicted observations $\hat{z}$ via eq. (21):
11     Compute observation uncertainty:
       $S := \sum_{i=0}^{10} w^{(i)} \left(h\left(\chi^{(i)}\right) - \hat{z}\right) \left(h\left(\chi^{(i)}\right) - \hat{z}\right)^T + R$
12     Compute innovation:
       $y := z - \hat{z}$
13     Compute Kalman gain:
       $V := \sum_{i=0}^{10} w^{(i)} \left(\chi^{(i)} - x\right) \left(h\left(\chi^{(i)}\right) - \hat{z}\right)^T$
       $K := VS^{-1}$
14     if $\sqrt{(Ky)^T \hat{P}^{-1} (Ky)} < 3$ then
15         Compute new state:
           $x := \hat{x} + Ky$
16         Compute new state uncertainty:
           $P := \hat{P} - KSK^T$
17     else
18         Use predicted state only:
           $x := \hat{x}$
19         Use predicted state uncertainty only:
           $P := \hat{P}$
```

$$J_{13} = \frac{v}{\omega}(\sin(\phi + \omega \Delta t) - \sin \phi) \tag{13}$$

$$J_{14} = \frac{1}{\omega}(-\cos(\phi + \omega \Delta t) + \cos \phi) \tag{14}$$

$$J_{15} = \Delta t \frac{v}{\omega} \sin(\phi + \omega \Delta t) \tag{15}$$

$$-\frac{v}{\omega^2}(-\cos(\phi + \omega \Delta t) + \cos \phi) \tag{16}$$

$$J_{23} = \frac{v}{\omega}(\cos(\phi + \omega \Delta t) - \cos \phi) \tag{17}$$

$$J_{24} = \frac{1}{\omega}(\sin(\phi + \omega \Delta t) - \sin \phi) \tag{18}$$

$$J_{25} = \Delta t \frac{v}{\omega} \cos(\phi + \omega \Delta t) \tag{19}$$

$$-\frac{v}{\omega^2}(\sin(\phi + \omega \Delta t) - \sin \phi) \tag{20}$$

Similar to a particle filter, a UKF generates candidates, called sigma points, from the current state's covariance but in an unrandomized fashion (line 8 in algorithm 3). For each of these poses the projected line segments of the detections to the backprojected park markings are computed as described in Sec. 5.2. The end points a_0'' and b_0'' of the projections of the most likely pose are used as the predicted observations which are compared to those of the actual detections a and b in order to compute the update to the status vector refining the vehicle pose. Let accordingly $a^{(j)}$ and $b^{(j)}$ be the end points of all associated marking detections and $a_0''^{(j)}$ and $b_0''^{(j)}$ those of the backprojections of the associated map marking under the currently assumed vehicle pose x_{t+1} for $j = 1, ..., m$. The observation z, expected observation $\hat{z}$ and observation prediction $h(.)$ are chosen as:

$$z := \left(a^{(1)}, b^{(1)}, ..., a^{(m)}, b^{(m)}\right)^T \tag{21}$$

$$\hat{z} := \left(a_0''^{(1)}, b_0''^{(1)}, ..., a_0''^{(m)}, b_0''^{(m)}\right)^T \tag{22}$$

$$h\left(\chi^{(i)}\right) := \left(a_i''^{(1)}, b_i''^{(1)}, ..., a_i''^{(m)}, b_i''^{(m)}\right)^T \tag{23}$$

The use of the rather uncommon UKF allows to apply this procedure, which is cumbersome to express in analytic terms. It avoids explicitly computing the gradient of this transformation, which would be necessary in an EKF, and coping with possible degeneration of a particle filter. Furthermore, it allows to handle some subtleties we have not described in detail, *e.g.* as not all relevant park markings may be visible from every candidate vehicle pose, the observation vector's size has to be adapted dynamically.

In order to handle wrong detection associations, another simple criterion for outlier detection is deployed. The Mahalanobis distance of the Kalman update, *i.e.* the difference of the predicted state $\hat{x}_t$ and the corrected state x_{t+1}, is taken w.r.t. the current state covariance matrix (*cf.* line 14 in algorithm 3). If this distance exceeds a fixed threshold, the new state is highly unlikely which is then accounted to a wrong detection. Thus, in this case all detections of that frame are discarded.

6 IMPLEMENTATION

The whole processing pipeline consisting of image processing, detection association, and pose estimation was implemented in C++ in fast but not particularly optimized code. Being executed on a single core, the average runtime per camera and frame amounted to 40.0 ms on a test system with a Intel i7-4702MQ CPU at 2.20GHz with 8 GB RAM. Table 1 gives a detailed analysis of the runtimes of all algorithm sub-steps. The runtime was taken by processing all test sequences (*cf.* Sec. 7). Thus, different weather, lighting, and parking occupation conditions were covered. The total average runtime over all

Algorithm sub-step	**Execution time [ms] mean ± std**
Image segmentation via vehicle pose	7.3± 2.9
Line detection (total)	32.1± 3.4
Brightness equalization	5.8± 0.3
Groundplane transform	5.0± 0.6
Gradient computation	8.8± 1.2
Threshold largest gradients	3.7± 0.4
Symmetry detection	4.9± 0.9
Pixel clustering	1.9± 0.6
Line clustering	1.8± 0.6
Landmark association	0.4± 0.3
Pose estimation	0.3± 0.2
Total runtime	40.0± 4.5

Table 1 Execution times of all algorithm sub-steps

scenarios was between 43.0 ms and 38.9 ms. Thus, despite different conditions the runtime proves itself stable.

The biggest percentage of the computational resources can be attributed to the image processing stage (39.4 ms), rendering the association stage (0.4 ms) and the estimation stage (0.3 ms) negligible. Since four cameras have to be processed, this results in a frame rate of 5 Hz, which we assume to be sufficient for ego pose estimation. However, for the experiments we had a multi-core CPU at our disposal allowing to process all cameras in parallel. Regarding the details of the image processing pipeline, we learn that the preprocessing steps dominate the computation costs: contrast equalization , ground plane transform (Fig. 3 b)), and gradient computation and thresholding are much more computationally demanding than the more sophisticated parts, namely symmetry detection (Fig. 3 c)), line growing (Fig. 3 d)), and final clustering (Fig. 3 f)).

The most critical parts of the algorithm are those with superlinear complexity: symmetry detection, pixel clustering, and line clustering. In order to guarantee for a restricted execution time, the problem size of these sub-steps is restricted via the parameters θ (Sec. 4.2), τ_G (Sec. 4.3), and τ_{Len} (Sec. 4.4).

The presegmentation using the known vehicle ego pose (Fig. 3 a)), which aims at robustifying the detection results, also decreases the runtime for the marking detection stage. While the detection on a full image (900×550 pixels) takes 53.8 ms, the presegmentation is at 7.3 ms, reducing the remaining image processing cost to 32.1 ms per image. In conclusion, even assuming the runtime on a full camera image as worst case, the presented processing pipeline is real-time capable.

7 EXPERIMENTS

The aim of the experimental evaluation presented in this section is twofold: we provide a thorough analysis of the detectors capabilities independently from the self-localization setup and look into the entire ego pose estimation pipeline. The tests were performed in a roofed prototype parking deck that was open to public traffic, thus, allowing for real-life scenarios and parking lot occupation. In order to arrive at a highly accurate representation of the park markings, their position was measured by a tachymeter with an accuracy below 1 cm. The two used vehicles were equipped with a topview camera system as described in Sec. 2. The cameras had a view angle of 177 ° and were mounted at a height of 0.6 m (front bumper), 1.0 m (side mirrors), and 0.8 m (trunk lid). The vehicle dimensions amounted to 2.1 m × 4.8 m with the side mirror cameras at a distance of 1.8 m from the front end. The cameras captured images at a resolution of 1280×800 pixels, the ground plane representation, however, was chosen at a resolution of 900×550 pixels (4 cm^2 per pixel). The image processing pipeline was initiated with the same set of optimized parameters (*cf*. Sec. 4.5) for all 8 cameras. In some experiments the rear camera was switched off (*cf*. Fig. 10).

The evaluation dataset comprised twelve sequences covering drives along the parking deck as well as parking maneuvers recorded on different times of day with different lighting scenarios and parking deck occupation. As reference localization method a LIDAR-based system (Ibisch et al 2013) was used which is reported to have a positioning error of less than 8.5 cm. The initial pose was taken from this system as well, all other poses are computed only based on the pose (and its uncertainty) from the previous frame estimation. Since the LIDAR system did not cover regions for more complex maneuvers, the two end points of the park markings were labeled manually in some sequences and an ego pose was framewisely estimated by gradient descent. Noise in the range of the mean back projection error of the pose was added to the labeled markings and yielded a deviation of 0.04 m±0.03 m and 2.01 °±1.91 °. This can at the same time be seen as the maximum possible accuracy with the proposed setup.

7.1 Detector Performance

In order to inspect the performance of the video-based park marking detection proposed in Sec. 4, the algorithm was executed on all sequences. Since the backprojected park markings are accurate enough, a correct association was deemed a correct detection, hence, the backprojected map data is used as ground truth. Visual inspection revealed no falsely associated detections. Figures 5 and 6 show the detection stage's performance in terms of precision and recall. The results are presented w.r.t. the distance of the marking to the camera which we regard as primary factor for the detection rate. We learn that a reliable detection of around 95% precision is possible when detecting park markings at a distance between 4 and 8 m. The detection rate for markings in even farther distance is very low. Furthermore, far-distant markings

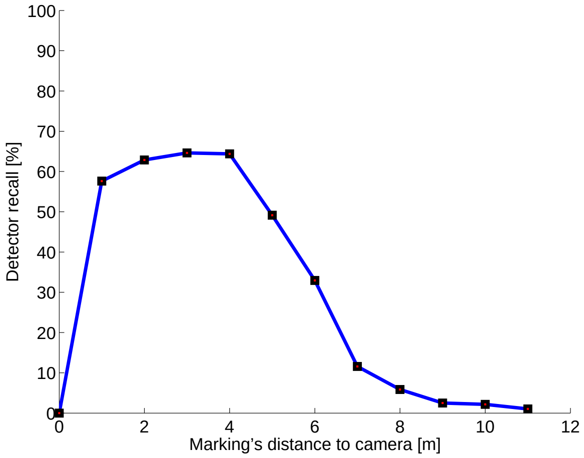


Fig. 5 Detection rate w.r.t. distance of the park marking

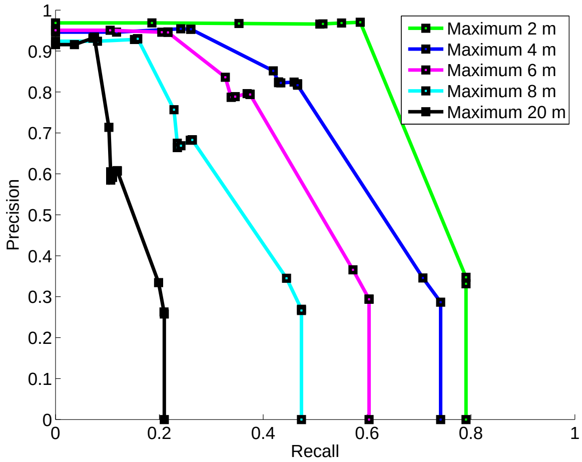


Fig. 6 Detector performance for different detection ranges. The input image was presegmented for the range-restricted detector runs which narrows its search space.

lie in close proximity when projected to the camera image. A possible confusion is therefore likely which is then prevented by the association stage (*cf*. Sec. 5.2).

7.2 Estimation error due to flat world assumption

As pointed out in Sec. 5, the Kalman filter manages a two dimensional vehicle position and orientation. However, the ground plane was slightly sloped to the middle of the deck. The slope amounted to 0.01 m at a distance of 1.0 m. Indeed, in a real-life scenario it is more likely to have two dimensional rather than three dimensional map data at one's disposal. Since this model inconsistency is rather common in self-localization, we examine the effect of the flat world assumption on the localization accuracy with the help of simulations:

A position was randomly drawn on the slanted plane and checked to have 6 markings closer than 8.0 m which we considered sufficient for self-localization. The relative directions of the markings w.r.t. the position were computed and rotated into the unslanted ground plane. The discrepancy between the rotated relative directions and the markings from the flat map data were minimized by adapting the vehicle's position and orientation in 2D. Thus, effects of imprecise detection or ego pose estimation are excluded. The pose before and after the minimization yields the estimation error due to the flat world assumption. Based on 73 trials, we arrived at an average positioning error of 0.09 m and an orientation error of 0.2 °. Figure 7 shows a distribution of the drawn positions and their flat world estimations.

7.3 Dependency on Inital Position

The ego pose estimation pipeline has to be initialized with a coarse vehicle pose as the used markings are indistinguishable and can, thus, provide only a relative pose estimation. The necessary accuracy of this initial pose was examined by considering several positions in the available sequences. To arrive at a useful accurate position, which can be regarded as current ground truth, the system was initialized with the coarse LIDAR pose and presented the same camera image several times, thus, mimicking a stationary vehicle and refining the pose gradually. Afterwards, offsets in position and orientation were added to the ground truth pose and the refinement was started all over again. Figure 8 shows the deviation of the repeatedly refined poses from the possibly noisy ground truth. The added offsets in position are separated in the directions parallel and orthogonal to the park markings since parallel positioning along the markings is in principle ambiguous.

7.4 Ego Pose Estimation Accuracy

Figure 10 shows the course of the LIDAR-based positioning system or alternatively the poses obtained by manual labelling (*cf*. Sec. 7) in comparison to the proposed self-localization pipeline from some of the test sequences. Please note that the LIDAR system did not provide continuous tracks. Again the positioning error is split into the directions parallel and orthogonal to the park markings. Figure 9 shows histograms of the respective pose estimation errors. They amounted to 0.23 m±0.24 m (parallel position), 0.15 m±0.18 m (orthogonal position), and 2.01 °±1.91 ° (orientation).

8 FUTURE WORK

In future work we will examine ways to provide the necessary initial positioning and stabilize the motion estimation by sensor fusion. Further stabilization can be achieved by encompassing additional distinguishable absolute landmarks into the detection stage. These landmarks will, however, most likely be artificial. Furthermore, we want to evaluate the use of IMU readings to stabilize the motion estimation during difficult maneuvers when too few markings are close-by. Alternatively, as an external approach relying on already installed hardware we want to use surveillance cameras to detect and approximately localize the vehicle (Ibisch et al 2014).

9 CONCLUSION

The analysis shows that the image processing pipeline yields very reliable detections if the confidence threshold is chosen adequately high. We arrive at a detection rate of about 30% of all markings closer than 8 m which turns out to be sufficient for self-localization. This shows that a topview wide-angle camera system can usefully be deployed for near-range vehicle-surrounding landmarker detection even if only few of them are close-by and barely distinguishable.

Although the system turned out to be very reliable, ambiguous markings are used which enables a purely relative motion estimation only. For safety-critical applications an update of the absolute position from time to time is therefor inevitable. This update can, however, be coarse (12 ° orientation and 0.9 m position discrepancy) as has been shown in the analysis.

The accuracy of the entire self-localization pipeline lies within the reference system's order of magnitude. A slightly worse accuracy can be found for positioning along the park marking axis. Although this is certainly due to the special configuration of the parking garage that was used in the presented experiments and only posesses parallelly aligned lines, a similar setup is usual in many other parking decks.

In conclusion, the presented setup yields reliable and precise information that we consider a highly important building block for operating a fully-autonomous parking deck.

References

Bojja J, Kirkko-Jaakkola M, Collin J, Takala J (2013) Indoor 3d navigation and positioning of vehicles in multi-storey parking garages. In: Proceedings of the International IEEE Conference on Acoustics, Speech and Signal Processing, pp 2548–2552

Chausse F, Laneurit J, Chapuis R (2005) Vehicle localization on a digital map using particles filtering. In: Proceedings of the IEEE Intelligent Vehicles Symposium, pp 243–248

Engel J, Schöps T, Cremers D (2014) Lsd-slam: Large-scale direct monocular SLAM. In: Proceedings of the European Conference on Computer Vision

Fernandez-Madrigal JA, Cruz-Martin E, Gonzalez J, Galindo C, Blanco JL (2007) Application of uwb and gps technologies for vehicle localization in combined indoor-outdoor environments. In: Proceedings of the 9th International IEEE Symposium on Signal Processing and Its Applications, pp 1–4

Gonzalez J, Blanco J, Galindo C, Ortiz-de Galisteo A, Fernández-Madrigal J, Moreno F, Martinez JL (2007) Combination of uwb and gps for indoor-outdoor vehicle localization. In: Proceedings of the International IEEE Symposium on Intelligent Signal Processing, pp 1–6

Grewal MS, Andrews AP (1993) Kalman Filtering: Theory and Practice. Prentice-Hall

Grisetti G, Kümmerle R, Stachniss C, Burgard W (2010) A tutorial on graph-based slam. Intelligent Transportation Systems Magazine 2(4):31–43

Hansen N (2006) The cma evolution strategy: A comparing review. In: Towards a New Evolutionary Computation, Studies in Fuzziness and Soft Computing, vol 192, Springer Berlin Heidelberg, pp 75–102

Haykin S (2001) Kalman Filtering And Neural Networks. Wiley Online Library

Ibisch A, Stümper S, Altinger H, Neuhausen M, Tschentscher M, Schlipsing M, Salmen J, Knolls A (2013) Autonomous driving in a parking garage: Vehicle-localization and tracking using environment-embedded lidar sensors. In: Proceedings of the IEEE Intelligent Vehicles Symposium, pp 829 – 834

Ibisch A, Houben S, Schlipsing M, Kesten R, Reimche P, Schuller F, Altinger H (2014) Towards highly automated driving in a parking garage: General object localization and tracking using an environment-embedded camera system. In: Proceedings of the IEEE Intelligent Vehicles Symposium, pp 426 – 431

Kiwan H, Bais A, Morgan Y (2012) A new base stations placement approach for enhanced vehicle position estimation in parking lot. In: Proceedings of the 15th International IEEE Conference on Intelligent Transportation Systems, pp 1288–1293

Kümmerle R, Triebel R, Pfaff P, Burgard W (2008) Monte carlo localization in outdoor terrains using multilevel surface maps. Journal of Field Robotics 25(6-7):346–359

Kümmerle R, Hähnel D, Dolgov D, Thrun S, Burgard W (2009) Autonomous driving in a multi-level parking structure. In: Proceedings of the International IEEE Conference on Robotics and Automation, pp 3395–3400

Lakafosis V, Vyas R, Tentzeris MM (2010) A localization and position tracking solution utilizing solar-powered rfid tags. In: Proceedings of the Fourth IEEE European Conference on Antennas and Propagation, pp 1–4

Laneurit J, Chapuis R, Chausse F (2005) Accurate vehicle positioning on a numerical map. International Journal of Control, Automation, and Systems 3(1):15–31

Lategahn H, Schreiber M, Ziegler J, Stiller C (2013) Urban localization with camera and inertial measurement unit. In: Proceedings of the IEEE Intelligent Vehicles Symposium, pp 719–724

Levinson J, Montemerlo M, Thrun S (2007) Map-based precision vehicle localization in urban environments. In: Robotics: Science and Systems, Citeseer, vol 4, p 1

Lim J, Frahm JM, Pollefeys M (2011) Online environment mapping. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 3489–3496

McCall JC, Trivedi MM (2006) Video-based lane estimation and tracking for driver assistance: survey, system, and evaluation. Transactions on Intelligent Transportation Systems 7(1):20–37

Oberlander J, Klemm S, Essinger M, Roennau A, Schamm T, Zollner JM, Dillmann R (2014) A semantic approach to sensor-independent vehicle localization. In: Proceedings of the IEEE Intelligent Vehicles Symposium, pp 1351–1357

Pink O (2008) Visual map matching and localization using a global feature map. In: Computer Vision and Pattern

Recognition Workshops, pp 1–7

Pink O, Stiller C (2010) Automated map generation from aerial images for precise vehicle localization. In: Proceedings of the 13th International IEEE Conference on Intelligent Transportation Systems, pp 1517–1522

Pink O, Moosmann F, Bachmann A (2009) Visual features for vehicle localization and ego-motion estimation. In: Proceedings of the IEEE Intelligent Vehicles Symposium, pp 254–260

Saab SS, Nakad ZS (2011) A standalone rfid indoor positioning system using passive tags. IEEE Transactions on Industrial Electronics 58(5):1961–1970

Scaramuzza D, Martinelli A, Siegwart R (2006) A toolbox for easy calibrating omnidirectional cameras. In: Proceedings of International IEEE Conference on Intelligent Robots and Systems, pp 7–15

Schindler A (2013) Vehicle self-localization with high-precision digital maps. In: Proceedings of the IEEE Intelligent Vehicles Symposium, pp 141–146

Schreiber M, Knöppel C, Franke U (2013) Laneloc: Lane marking based localization using highly accurate maps. In: Proceedings of the IEEE Intelligent Vehicles Symposium, pp 449–454

Schubert R, Richter E, Wanielik G (2008) Comparison and evaluation of advanced motion models for vehicle tracking. In: Proceedings of the International Conference on Information Fusion, pp 1–6

Thrun S (2002) Probabilistic robotics. Communications of the ACM 45(3):52–57

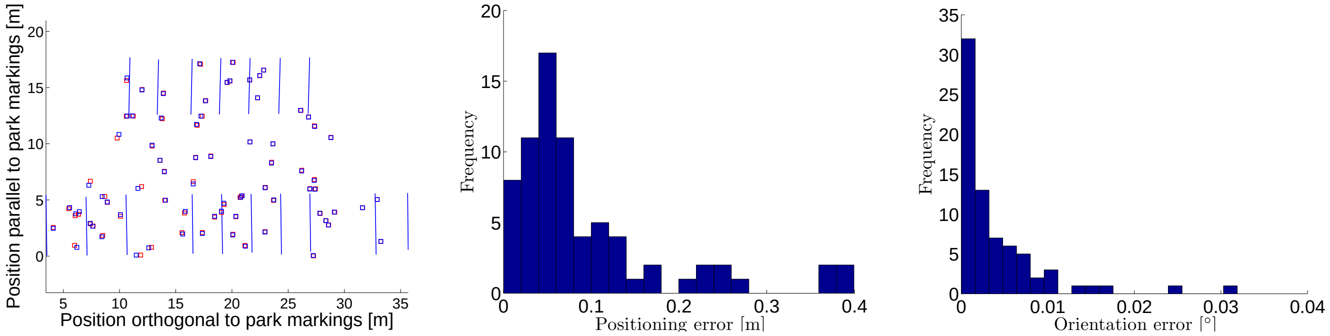
Thrun S, et al (2002) Robotic mapping: A survey. Exploring artificial intelligence in the new millennium pp 1–35

Veit T, Tarel JP, Nicolle P, Charbonnier P (2008) Evaluation of road marking feature extraction. In: Proceedings of IEEE Conference on Intelligent Transportation Systems, pp 174–181

Wagner J, Isert C, Purschwitz A, Kistner A (2010) Improved vehicle positioning for indoor navigation in parking garages through commercially available maps. In: Proceedings of the International IEEE Conference on Indoor Positioning and Indoor Navigation, pp 1–8

Wu T, Ranganathan A (2013) Vehicle localization using road markings. In: Proceedings of the IEEE Intelligent Vehicles Symposium, pp 1185–1190

Yenikaya S, Yenikaya G, Düven E (2013) Keeping the vehicle on the road: A survey on on-road lane detection systems. Computing Surveys 46(1):2:1–2:43

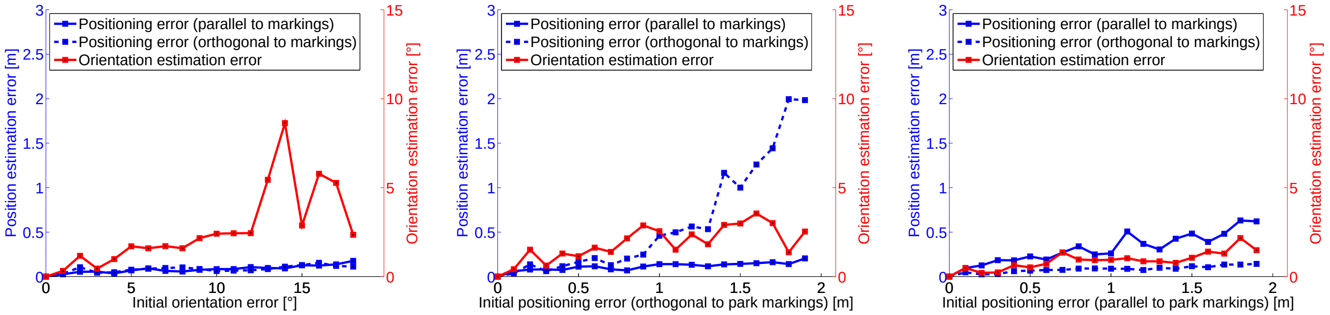


(a) Distribution of drawn vehicles poses and poses estimated under flat world assumption

(b) Distribution of position estimation error

(c) Distribution of orientation estimation error

Fig. 7



(a) Pose estimation error w.r.t. noisy initial orientation

(b) Pose estimation error w.r.t. noisy initial position orthogonal to park markings

(c) Pose estimation error w.r.t. noisy initial position parallel to park markings

Fig. 8

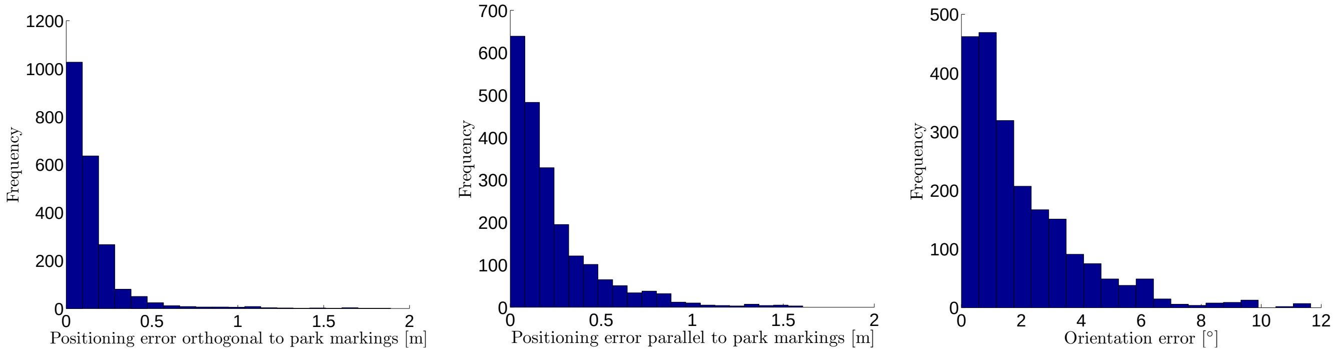
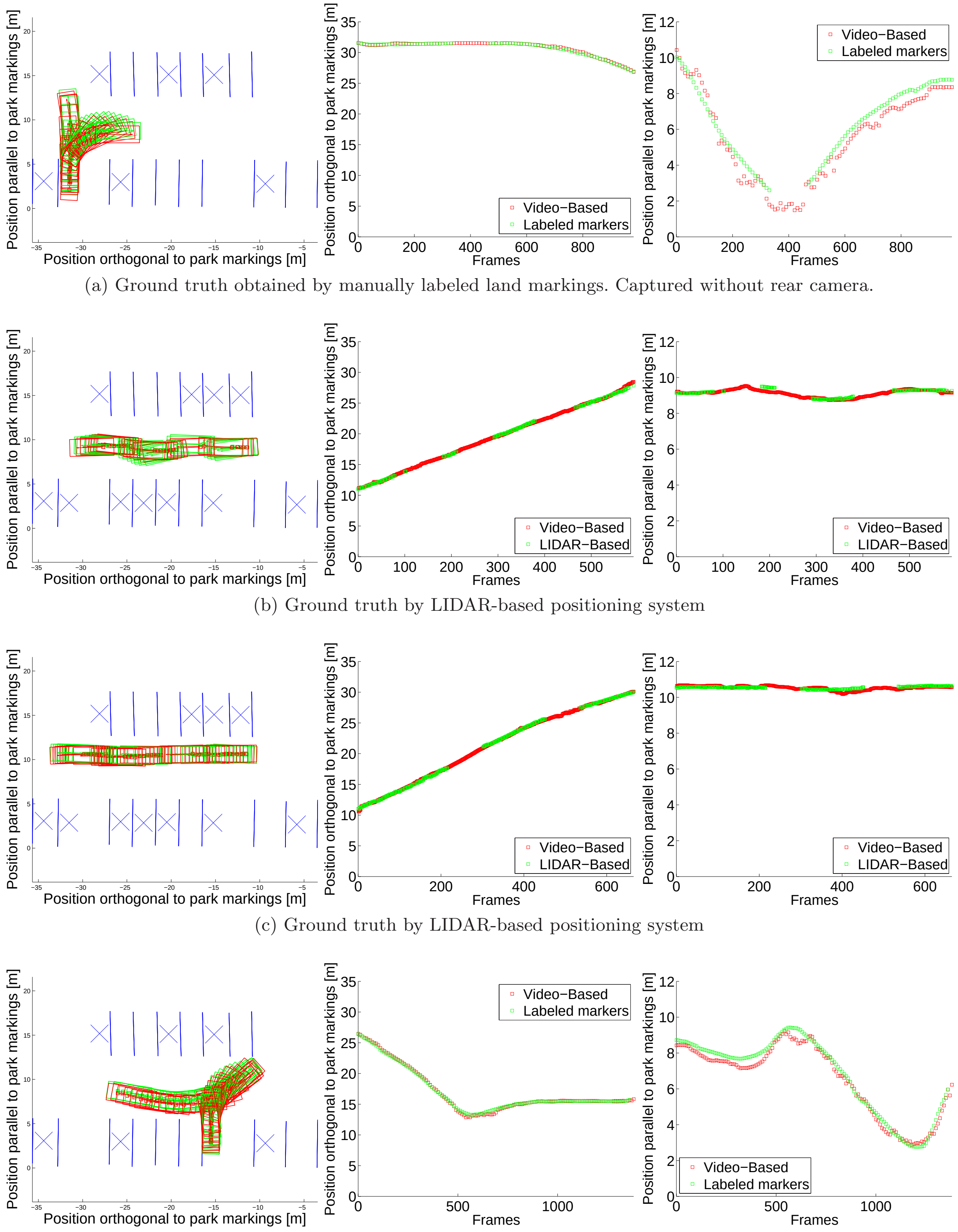


Fig. 9 Distribution of localization errors



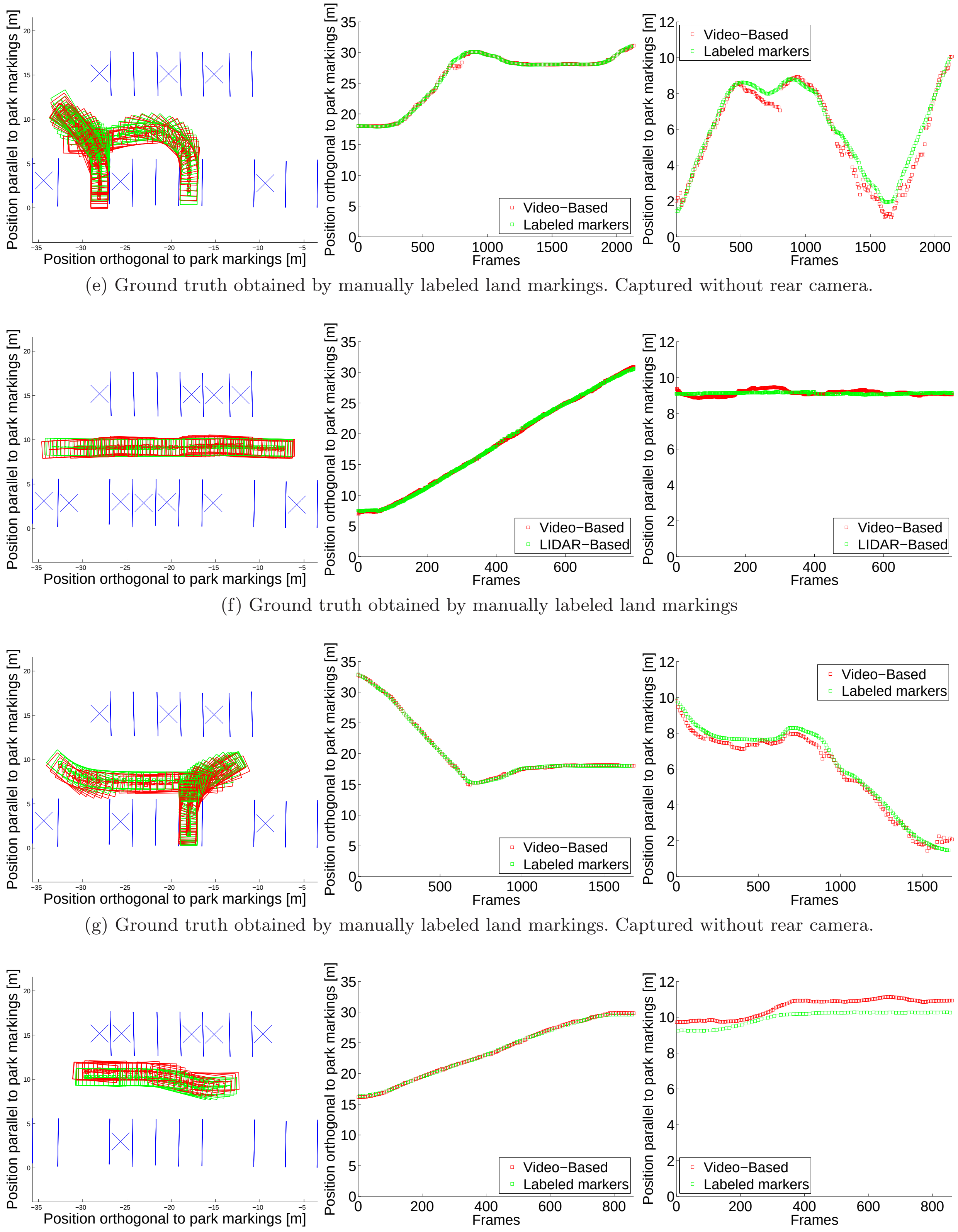
(a) Ground truth obtained by manually labeled land markings. Captured without rear camera.

(b) Ground truth by LIDAR-based positioning system

(c) Ground truth by LIDAR-based positioning system

(d) Ground truth obtained by manually labeled land markings. Captured without rear camera.

Fig. 10 Estimation and ground truth of vehicle pose. The parking lot occupancy during experiments is given by the crosses in the map overview.



(e) Ground truth obtained by manually labeled land markings. Captured without rear camera.

(f) Ground truth obtained by manually labeled land markings

(g) Ground truth obtained by manually labeled land markings. Captured without rear camera.

(h) Ground truth obtained by manually labeled land markings

Fig. 10 Estimation and ground truth of vehicle pose. The parking lot occupancy during experiments is given by the crosses in the map overview.