

# Towards Highly Automated Driving in a Parking Garage: General Object Localization and Tracking Using An Environment-Embedded Camera System

André Ibisch\*, Sebastian Houben\*, Marc Schlipping\*,  
Robert Kesten •, Paul Reimche •, Florian Schuller ◦, and Harald Altinger◦

**Abstract**—In this study, we present a new indoor positioning and environment perception system for generic objects based on multiple surveillance cameras. In order to assist highly automated driving, our system detects the vehicle’s position and any object along its current path to avoid collisions. A main advantage of the proposed approach is the usage of cameras that are already installed in the majority of parking garages. We generate precise object hypotheses in 3D world coordinates based on a given extrinsic camera calibration. Starting with a background subtraction algorithm for the segmentation of each camera image, we propose a robust view-ray intersection approach that enables the system to match and triangulate segmented hypotheses from all cameras. Comparing with LIDAR-based ground truth, we were able to evaluate the system’s mean localization accuracy of 0.37 m for a variety of different sequences.

## I. INTRODUCTION

In this study, we introduce an infrastructural embedded approach for localization and tracking of generic objects for indoor environments. We focus on the example of parking garages to establish a positioning system in the context of autonomous driving.

Its main target is to detect and track vehicles and secondary objects along its current path. Due to the lack of GPS information and non-sufficient on-board vehicle sensors an infrastructural system which communicates with the autonomously driving car has to be precise, reliable and real-time capable. If an object crosses the path of the vehicle, the system has to raise a warning to avoid a collision. Because arbitrary objects (e.g., other vehicles, small/tall humans, bicycles, or animals) should be recognized by the system, size and shape constraints are ignored.

To achieve these aims, we use surveillance cameras already installed in the majority of the parking garages and extend their purpose to an external vehicle localization system. Thus, the approach is inexpensive, does not require additional hardware except the infrastructural car-to-environment communication, and is transferable to other indoor scenarios, e.g., tunnels, factories etc.

Firstly, an overview of the related work is described in Sec. II. The developed system is presented in Sec. III

and evaluated by comparison to a LIDAR-based system in Sec. IV. A final conclusion and outlook is discussed in Sec. V.

## II. RELATED WORK

As an example for autonomous indoor-localization, we elaborate on the idea of an autonomous valet parking system mentioned in [1]. The paper describes a precise LIDAR-based localization system which is part of a framework for autonomous driving: A driver hands over his car at a parking garage entrance, steps out, and the vehicle is autonomously driven through the garage towards a free parking spot where it can be claimed later. We also use an extended version of this LIDAR-based setup to evaluate and compare our system’s result in Sec. IV.

Another system for autonomous driving in a parking garage – also using a network of video cameras – is discussed in [2]. Einsiedler et al. used a motion template and the Viola-Jones-Detector to detect a pre-trained vehicle. To determine the vehicle’s position within the parking garage a lane has been divided into segments of 1 m<sup>2</sup>. In contrast, our approach focuses on generic object detection without any knowledge or training of objects that will appear in the parking garage. The authors report an uncertainty in their predicted vehicle hypotheses of 1 m with a coverage of 14 m due to the cameras extrinsic constraints.

Evans et al. presented a multicamera-based system for object detection by using a synergy map based on a defined ground plane and a pre-specified height [3]. In contrast to that, we avoid making any assumptions concerning the objects, the ground plane or the objects’ height.

In [5] the authors present a comparative study for multiple person tracking with overlapping camera views. Because any object can occur in the parking garage, we concentrate on general object detection and tracking. Another very valid approach is presented in [4]. By the use of volumetric 3D reconstruction the visual hulls of the objects were transformed and tracked inside a occupancy volume. The objects’ mass center on the ground plane is generated by an derived occupancy map. A specific problem described by the authors is the handling of objects with similar look within the assignment of image regions to tracked 3D volumes.

\*Institut für Neuroinformatik, Ruhr-Universität Bochum, Germany.  
{andre.ibisch, sebastian.houben,  
marc.schlipping,}@ini.rub.de  
◦ AUDI AG, Ingolstadt, Germany.  
{florian.schuller, harald.altinger}@audi.de  
• GIGATRONIK Ingolstadt GmbH, Ingolstadt, Germany.  
{robert.kesten, paul.reimche,}@gigatronik.com



Fig. 1. The problem at hand: The installed surveillance camera from a parking garage captures several traffic participants. Our aim is to make use of these images to measure their position and motion in order to detect potentially dangerous situations.

### III. PROPOSED SYSTEM

Our proposed system is based on video only and recognizes objects and their position inside the coordinate system of a parking garage. A typical scenario captured by a surveillance camera is shown in Fig. 1.

The following sections are structured analogously to the system’s pipeline: At the beginning, each camera captures images from its environment which are analyzed separately (c. f. Sec. III-A): Moving objects are segmented by a background subtraction approach (c. f. Sec. III-B). Afterwards, all foreground objects are transformed into view rays inside a common world coordinate system, wherefore an initial calibration is required (c. f. Sec. III-C). Subsequently, these rays are combined in an appropriate manner to receive plausible object hypotheses (c. f. Sec. III-D). Finally, frame-wise object hypotheses are tracked over time to guarantee continuous object localization.

#### A. Camera network

To demonstrate a garage parking scenario with a surveillance camera system, we use multiple grayscale cameras connected by a local area network (LAN).

The cameras are mounted on high tripods to simulate a typical surveillance system, where cameras are usually installed below the ceiling. We ensure that each camera shares its field of view with at least one of the other cameras to handle occlusions and to enable multiple-view vision. The best case should be a complete coverage of a large area of the parking garage with only two cameras.

The cameras are synchronized by the IEEE1588 protocol over the LAN to guarantee simultaneous exposure and processing of all images. Depending on its position in the parking garage, we use different types of lenses to capture a wide area with a minimum number of cameras (e.g., wide-angle lenses in corners). In the evaluated setup, 4.8 mm, 9 mm, and 12.5 mm lenses were used.

#### B. Foreground Segmentation

Since a surveillance camera is embedded in a static environment, most parts of the camera image are constant background, e.g. traffic signs, walls, etc. These regions of an

image are called image background, and only minor image regions contain objects of interest, defined as the image foreground. In order to separate regions of interests (ROIs) of a single camera image, we decided to model the static background appropriately.

We propose a pipelined approach for the separation of those ROIs from the background (see Fig. 2): First, we generate a representation of the background using the well-established background subtraction method [6]. Then, the difference between the current camera image and the background representation is calculated. The difference contains moving objects, but also noise, e.g., caused by shadows or the vehicle’s spotlights). Morphological operations (*Opening*) were used to reduce this kind of noise.

To minimize false segmentation caused by shadows or active light sources, we apply the normalized cross-correlation (NCC) method by identifying structurally constant image regions in combination with background subtraction presented in [6]. Afterwards, the segmented regions are clustered to obtain connected regions, which are represented by single ROIs. A further preprocessing step is to merge overlapping and adjacent ROIs. In order to establish a background representation, we use an exponentially smoothed mean-image:

$$B(x, y) = (B(x, y) * (1 - \alpha)) + (I(x, y) * \alpha) \quad (1)$$

where  $B(x, y)$  represents the background image,  $I(x, y)$  the current image with the same size as  $B(x, y)$  and a weight  $0 \leq \alpha \leq 1$ . To reduce the negative effects of active light sources the background subtraction method ignores overexposed pixels.

An example of a typical parking garage scenario is shown in Fig. 2(b). The difference of the background image  $B(x, y)$  and the current image  $I(x, y)$  is stored in a binary segmentation image  $S(x, y)$  using threshold  $t$ :

$$S(x, y) = \begin{cases} 1 & , \quad |B(x, y) - I(x, y)| < t \\ 0 & , \quad \text{else} \end{cases} \quad (2)$$

To diminish weak or strong intensity changes, we extended the expression  $|B(x, y) - I(x, y)|$  by clipping it to a minimum and maximum, respectively. In Fig. 2(c) an example of a raw (before any preprocessing step) binary segmentation image is shown.

The background learning process takes place during an initialization phase. To speed up the initial learning, we apply a parameterized decreasing of  $\alpha$  starting with value 1. For an immediate initialization, the system is alternatively able to load a precalculated background image, e.g., if there is no time for a learning phase.

Reflections of strong illumination sources cause false segmentation. Discarding overexposed pixels does not eliminate all false positives. By applying the NCC method to the segmentation image we reduce these influences. We divide background image  $B(x, y)$  and the current image  $I(x, y)$  into equally sized grid cells. To minimize the workload, a grid ROI is examined only if it contains a minimal number of segmented pixels.

The NCC calculates the degree of structural similarity, i.e., lighting-independent, of these corresponding grid ROIs. If the NCC is above a threshold the grid ROI of the current image  $I$  is similar to its corresponding ROI in the background image  $B$  and the complete ROI is discarded in the segmentation image, i.e.,  $S(x, y)$  is set to 0.

An exemplary NCC procedure is shown in Fig. 2(d). Each point of the grid represents the center of an ROI: green relates to a not considered ROI, yellow to a maintained ROI and red marks refused ROIs. The resulting enhanced segmentation image is shown in Fig. 2(e).

Afterwards our system calculates clusters based on the enhanced segmentation image: The cluster algorithm operates by considering an 8-neighborhood of each pixel. An initial cluster result is shown in Fig. 2(f). We substitute overlapping and adjacent clusters for their aggregation and track them in subsequent images with an Alpha-beta-filter (c.f. Fig. 2(g) and Fig. 2(h)). Afterwards we process them to the next module.

### C. System calibration

In order to interpret and combine detections from multiple cameras it is imperative to know their exact relative positions and orientations, a.k.a. extrinsic calibration. Furthermore, an intrinsic calibration, i.e., a mapping between the camera frame and the world coordinate frame, encompassing the lens distortion parameters of each camera, must be determined. For the latter part, we rely on the methods by [7] to obtain vertical and horizontal focal lengths and radial distortion parameters.

For the extrinsics, we measure the three-dimensional coordinates of distinct points in the depicted scene w.r.t. a chosen world origin along with the corresponding image coordinates in the respective camera frames. The goal is now to minimize the squared distances between the backprojected scene points and the marked image coordinates. The backprojection computation includes the inverse distortion function to directly compare distances within the raw images. We follow a steepest-descent optimization technique without known local gradients starting with several initial solutions to avoid local minima. In order to guarantee for numerical stability we initialize the camera center with the measurement of the camera position in the world coordinate system and keep it fixed during the first iterations of the optimization. In later optimization loops we optimize for all parameters, the orientation and the translation. We refer to Sec. IV for a detailed evaluation of the calibration accuracy.

### D. Image-World Transform

In this section, we describe the processing steps that follow foreground segmentation to yield single-frame hypotheses for world objects. In principle, a single camera detection would suffice to generate a world representation of the object via projection onto the ground plane. However, depending on the camera geometry, this procedure can be unstable since small errors in the detected ROI lead to strong misestimations of the points where the object touches the ground plane. The



(a) Original image section.



(b) Mean Image.



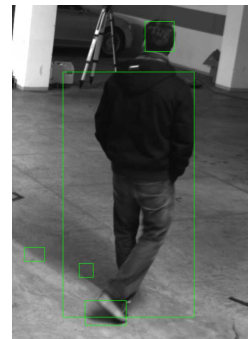
(c) Binary image before preprocessing.



(d) NCC grid, ROI colour: green  $\hat{=}$  ignored, yellow  $\hat{=}$  retained, red  $\hat{=}$  dropped.



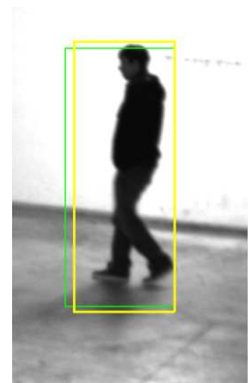
(e) Binary image after NCC.



(f) Initial clusters.



(g) Merged (green colour) and tracked (yellow) clusters.



(h) Same time and processing step from the second camera.

Fig. 2. Foreground segmentation pipeline for an image section of Fig. 1.

main idea to circumvent this problem is to fuse detected regions from several camera images with overlapping field of view. One has to carefully approach this problem because the number of image regions in different cameras can be different. We can regard this situation as a marriage problem with symmetric distances. The image regions from different cameras should be matched together by means of a distance measure that is still to be defined.

We make use of the fact that the cameras are all aligned upright w.r.t. the ground plane, thus, all detected regions share the same vertical orientation. We propose the following ROI distance measure shown in Fig. 3. Let us regard a pair of ROIs from two different cameras. In a first step we compute the view rays emerging from the respective camera center of all ROI corner points. The rays corresponding to the upper and lower ROI corners are intersected with those from the respective other image<sup>1</sup>. The distance in 3D space would be a quality criterion on the matching ROIs. Since it depends on perspective, we suggest to consider the backprojected coordinates of the intersection points into the corresponding camera frames. We arrive at four backprojected points for the upper two and four backprojected points for the lower two ROI corners. The average of the distance from each corner to the closer of its two backprojections computed for both ROIs finally donates a matching distance for the regarded ROI pair. Formally, let  $R_1, R_2$  be two ROIs in two camera frames,  $r_1^1, \dots, r_1^4, r_2^1, \dots, r_2^4$  their respective corner points, and  $q_i^{j,(1,\dots,2)}$  the two backprojected intersection points of  $r_i^j$  (c. f. Fig. 3). The matching distance  $d(R_1, R_2)$  is then defined as

$$d(R_1, R_2) = \frac{1}{2} \sum_{i=1}^2 \frac{1}{4} \sum_{j=1}^4 \min \left\{ \|q_i^{j,(1)} - r_i^j\|, \|q_i^{j,(2)} - r_i^j\| \right\}$$

With the help of this measure we set up the marriage problem and compute an optimal matching via the well-known propose-and-reject-algorithm. Since not every ROI pair corresponds to the same object we define a distance threshold that leaves us with only those object detections that can reliably be assigned to one another. Those useful detection pairs yield two polygons in 3D space consisting of the aforementioned intersection points, one for the upper and one for the lower ROI corners. The polygons define the detected world object contours.

Remaining ROIs, that cannot be matched due to a too high distance measure, are projected to the ground plane: The lower ROI corner of the remaining single camera detections are projected to the ground plane and, thus, still define a coarse approximate object contour. These unmatched and ground-plane-projected ROIs are later utilized to confirm stable tracks.

### E. Tracking

The tracking module receives the hypotheses from the image-world-transformation and has to guarantee a complete

<sup>1</sup>For the sake of simplicity we use the notion of intersection also for skew lines where it refers to computing the single point in space that minimizes the distance to both intersecting lines.

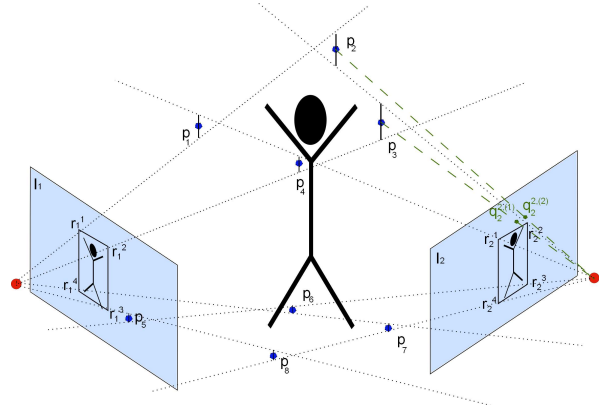


Fig. 3. Our object reconstruction method: Four view rays, emanating from the origin of the camera (red dots), are projected through the four ROI corner points  $r_1^1, \dots, r_1^4$  of a detected object in the left image  $I_1$  into the world scene. Together with four other view rays corresponding to the ROI of image  $I_2$  they are generating eight 3D intersection points  $p_1, \dots, p_8$  in the scene, i. e. points with minimal distance to the respective view rays, shown in blue. The backprojected points are illustrated in the right camera frame by the green dashed line starting from the intersection point to the image plane  $I_2$ . The green points  $q_2^2(1), q_2^2(2)$  on the image plane represent the reprojected points of the intersection point. The shortest distance between a reprojection and the next ROI corner is the reprojection error. This error is used to match multiple detections from different cameras.

temporal integration, either based on an observation, or a plausible prediction and closes gaps where no valid hypotheses are generated.

We use an extended Kalman filter (see [8]) with a physical motion model and reasonable observation noise.

We assign a previously unobserved hypothesis to a new track, which becomes stable after receiving further similar hypothesis in the following time steps. It is eliminated after a certain period of predictions by the Kalman filter without measurement. The stable tracks are the final output of our system.

## IV. EXPERIMENTS

In this section, we discuss the evaluation and comparison of the presented system with a LIDAR-based reference system both deployed in a parking garage that serves as proving ground for our experiments.

### A. Reference System

We used the LIDAR system presented in [1] as a reference. Briefly, an array of distributed LIDAR sensors is installed a few centimeters above the ground to detect and measure the distance to nearby obstacles. Learning to distinguish between static and dynamic, i. e., active points, the setup is then used to recognize and accurately track the four wheels of a vehicle and create a trajectory thereof.

This reference system was extended to detect general objects: We subtract the amount of active LIDAR measurements with the measurements of a final vehicle hypothesis to obtain only those points that are not related to a vehicle. These filtered active points are clustered locally. If they exceed a given size, clusters are regarded as general object detections (e.g. feet of a pedestrian).

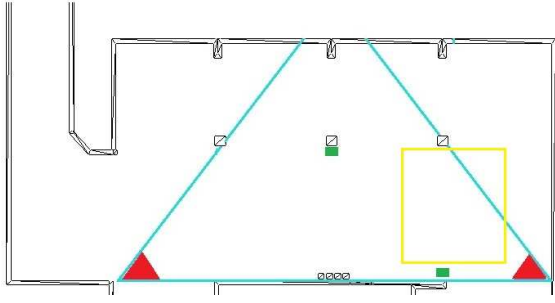


Fig. 4. The experimental setup: The *red* triangles represent the cameras, the *blue* triangle their shared field of view, and the *green* rectangles the LIDAR sensors. The parking garage ( $30m \times 15m$ ) layout is taken from the aforementioned CAD representation. An image captured from the right camera is shown in Fig. 1. The yellow rectangle (Coordinates: Upper corner left (28, 27), lower right corner (34, 30)) represents the observed area for the experiments. Fig. 6 plots this area left rotated through  $90^\circ$ .

In [1] the LIDAR system’s results were compared to human-labeled ground-truth data. Based on this comparison the authors report a mean lateral and longitudinal error of  $0.063m$  and  $0.085m$  for vehicle detection. Additionally, they achieve a mean distance between the system’s result and the reference data of  $0.121m$  with a standard deviation of  $0.051m$ . Thus, being highly accurate, these object detections are used as reference data in the following comparison.

### B. Evaluation of the Reference System

In order to generate a more objective evaluation of the LIDAR-based system than a human labeling approach, we evaluate their system with our own precise reference system. As recommended by the authors in their experiments section, we used a setup with a precise Inertial Measurement Unit (IMU) and a Differential-GPS (DGPS). Both results were transformed into the world coordinate system and compared to the LIDAR-based system result. Within different test runs (parking manoeuvre, circle drive and half-circle-drive), we determine the distances between the trajectories produced by the LIDAR-based system and by the presented positioning system with a mean euclidean distance of  $0.19m$ .

### C. Setup

We installed two GigE-Vision Prosilica AVT GT 1380 monochrome cameras, with an image resolution of  $1360 \times 1024$ , equipped with  $9mm$  lenses mounted on a  $2m$  tripod. Both were positioned vis-á-vis and share an intersecting field of view. We utilize two SICK LMS 500-20000pro as LIDAR sensors in the reference system. The setup is shown in Fig. 4, an example image in Fig. 3.

For a more precise detection we only use the lower intersection points  $p_{5,\dots,8}$  (see Fig. 3) as the view rays of the upper ROI corners  $r_1$  and  $r_2$  tend to intersect in a very narrow angle. Thus, the localization of the reconstructed object corner points  $p_{5,\dots,8}$  is numerically more stable than for  $p_{1,\dots,4}$ .

### D. Calibration

For a precise determination of world calibration points, we used a Leica Builder 306 tachymeter. We define a distinct

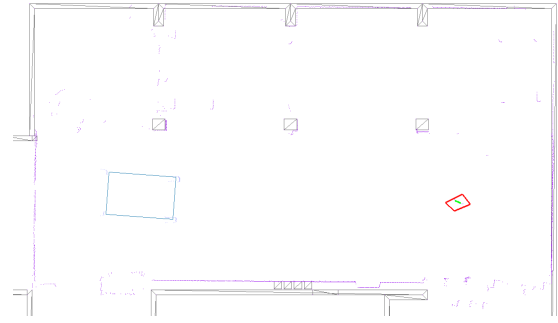


Fig. 5. Both systems’ results within the experimental setup (described in Fig. 4). It is the same scene as shown in Fig. 1 and in Fig. 2. The *violet* points represent active LIDAR measurements, the *blue* rectangle the vehicle hypothesis, the *green* polygon the LIDAR-based system result of the above mentioned cluster, and the *red* polygon our system’s final hypothesis.

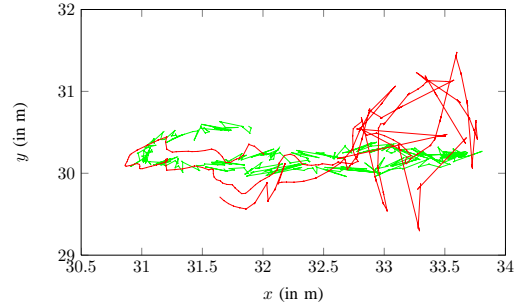


Fig. 6. The plots illustrates the world-trajectories of the pedestrian-centers inside the yellow rectangle in Fig. 4: Reference data by the LIDAR-based system in *green*, camera hypothesis in *red*. At position 32.5, 30 the vehicle switches on its lighting and causes the aberration in our system’s result.

world origin, measure it, and transfer the origin and all measured calibration points into a CAD representation of the test environment. We used the same CAD representation within the presented and the LIDAR reference system for a valid comparison.

### E. Results

For a demonstration of both systems’ trajectories we choose a representative sequence of a pedestrian. The sequence is recorded with 15 fps. A single frame result is shown in Fig. 5, a camera image of this result in Fig. 1. The origin of the coordinate system is in the left bottom of the image. Both trajectories are presented in Fig. 6. This figure is an excerpt from the CAD representation located in the right part of the image from Fig. 4. The pedestrian starts at position of 31, 30, walks to 34, 30 and back to 31, 30.

The sequence exemplifies four difficult situations:

- Time frame 0–250: Moving pedestrian in front of a static background.
- Time frame 251–500: Pedestrian overlaps with the vehicle in one camera and creates a merged ROI.
- Time frame 501–750: The vehicle switches on its lighting and the ROI with the pedestrian in overexposed.
- Time frame 750 – 850: The vehicle switches off the lighting, similar scenario as in time step 0–250.

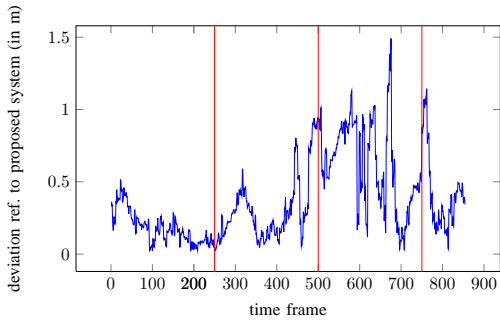


Fig. 7. The deviation of the LIDAR and the camera-based hypothesis. The red vertical lines separate the above-named four difficult situations.

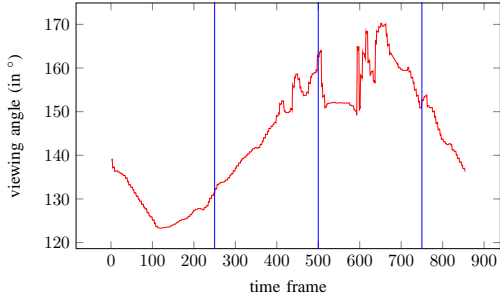


Fig. 8. The viewing angle over the examined sequence of the walking pedestrian. The viewing angle represents the angle in the triangle  $\gamma$  between both camera world-centers and the systems world-center.

This composition of different scenarios within the sequence is documented in Fig. 7 where the distances correspond to these scenarios. The mean deviation between the LIDAR cluster center and the camera polygon center over the entire sequence is  $0.37m$  with a standard deviation of  $0.28m$ .

Still, the localization does not only deteriorate due to strong illumination and overlapping image regions. Another impact on the system’s precision is the viewing angle: We construct a triangle between both involved world-camera-centers (A and B) and the resulting world-center of our system (C) and measure the triangles angle  $\gamma$ , from now on called viewing angle. This effect is shown in Fig. 8: When the person is located away from the virtual line connecting both cameras, the positioning error is significantly lower. We substantiate this assumption with a comparison of the localization error to the viewing angle in Fig. 9.

## V. CONCLUSION AND OUTLOOK

This study presents an indoor positioning system for generic objects by means of a camera network. Objects are segmented using a background representation. To generate precise and plausible world hypotheses we intersect view rays of these objects and track them in a world representation. We focus on the detection of generic objects of arbitrary size which can be performed without prior training.

The system’s mean positioning error in a sequence containing several difficult situations is  $0.37m$ . Compared to the LIDAR-based reference system with an error of  $0.19m$  this

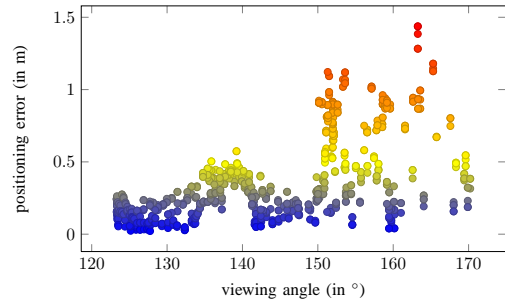


Fig. 9. The correlation between the positioning error of Fig. 7 and the viewing angle of Fig. 8.

value is fairly high. Similar camera-based systems proposed in the literature report higher deviations, e.g. [2], with a positioning error of  $1m$ .

However, our system is indeed precise enough to locate an object for applications like collision warning. We also want to point out that the proposed system is based on surveillance cameras, a majority of modern parking decks are equipped with. Therefore, it does not require additional hardware expense.

In the future, we want to investigate refinements of the image processing pipeline to handle remaining drawbacks we have identified. The effect of strong light sources needs to be reduced by further segmentation methods. The problem of overlapping objects in a single ROI – which occurred in only one camera image – has to be analyzed more deeply and can be solved by extending the proposed triangulation method.

## REFERENCES

- [1] A. Ibisch, S. Stümper, H. Altinger, M. Neuhausen, M. Tschentscher, M. Schlipfing, J. Salmen, and A. Knoll, “Autonomous driving in a parking garage: Vehicle-localization and tracking using environment-embedded lidar sensors,” in *Proceedings of the IEEE Intelligent Vehicles Symposium*, 2013, pp. 829 – 834.
- [2] J. Einsiedler, O. Sawade, B. Schaufele, M. Witzke, and I. Radusch, “Indoor micro navigation utilizing local infrastructure-base positioning,” in *Proceedings of the IEEE Intelligent Vehicles Symposium*, 2012, pp. 993–998.
- [3] M. Evans, C. Osborne, and M. Ferryman, “Multicamera object detection and tracking with object size estimation,” in *AVSS 2013*. IEEE, 2013, pp. 177–182.
- [4] H. Possegger, S. Sternig, T. Mauthner, P. Roth, and H. Bischof, “Robust real-time tracking of multiple objects by volumetric mass densities.” IEEE Computer Society, 2013, pp. 2395–2402.
- [5] M. C. Liem and D. M. Gavrila, “A comparative study on multi-person tracking using overlapping cameras,” in *Proceedings of the 9th International Conference on Computer Vision Systems*, ser. ICVS’13. Springer-Verlag, 2013, pp. 203–212.
- [6] J. Jacques, C. Jung, and S. Musse, “Background subtraction and shadow detection in grayscale video sequences.” in *SIBGRAPI*. IEEE Computer Society, 2005, pp. 189–196.
- [7] Z. Zhang, “A flexible new technique for camera calibration.” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 11, pp. 1330–1334, 2000.
- [8] M. S. Grewal and A. P. Andrews, *Kalman Filtering: Theory and Practice*. Prentice-Hall, 1993.