Arbitrary object localization and tracking via multiple-camera surveillance system embedded in a parking garage

André Ibisch, Sebastian Houben, Matthias Michael
Institute for Neural Computation, University of Bochum, Germany
Robert Kesten
GIGATRONIK Ingolstadt GmbH, Ingolstadt, Germany
Florian Schuller
AUDI AG, Ingolstadt, Germany

ABSTRACT

We illustrate a multiple-camera surveillance system installed in a parking garage to detect arbitrary moving objects. Our system is real-time capable and computes precise and reliable object positions. These objects are tracked to warn of collisions, e.g. between vehicles, pedestrians or other vehicles. The proposed system is based on multiple grayscale cameras connected by a local area network.

Each camera shares its field of view with other cameras to handle occlusions and to enable multi-view vision. We aim at using already installed hardware found in many modern public parking garages. The system’s pipeline starts with the synchronized image capturing process separately for each camera. In the next step, moving objects are selected by a foreground segmentation approach. Subsequently, the foreground objects from a single camera are transformed into view rays in a common world coordinate system and are joined to receive plausible object hypotheses. This transformation requires a one-time initial intrinsic and extrinsic calibration beforehand. Afterwards, these view rays are filtered temporally to arrive at continuous object tracks. In our experiments we used a precise LIDAR-based reference system to evaluate and quantify the proposed system’s precision with a mean localization accuracy of 0.24m for different scenarios.

1. INTRODUCTION

In this paper, we illustrate a multiple-camera surveillance system to locate and track arbitrary objects in a parking garage. The purpose of the system is to raise warnings if a collision between pedestrians or other objects (e.g., bicycles, or animals) is immanent. Because of the lack of GPS information inside indoor environments and non-sufficient on-board vehicle sensors an infrastructural system based on other sensors is desirable.

For this reason, the proposed system is based only on cameras, because the majority of parking garages is already equipped with multiple grayscale cameras. All cameras are connected with a local area network and share a collective field of view to handle occlusions and enable multi-view vision. Our system does not require additional hardware apart from the processing unit and is straightforwardly portable to other indoor scenarios, e.g. factory buildings etc.

At first, an overview of the related work is given in Sec. 2. The implemented system is described in Sec. 3. Afterwards, in Sec. 4 we evaluate accuracy of our system with the help of a reference system and present the results. Finally, a conclusion and an outlook is presented in Sec. 5.

E-mail: {andre.ibisch, sebastian.houben, matthias.michael}@ini.rub.de
E-mail: robert.kesten@gigatronik.com
E-mail: florian.schuller@audi.de
2. RELATED WORK

In Ibisch\textsuperscript{1} we presented an indoor positioning system for generic objects. We picked up and solved the named refinements mentioned in the conclusion\textsuperscript{1} : The negative effect of strong illumination and the problem of overlapping objects. We concentrated on improving the image processing pipeline and also implemented a novel tracking method to establish temporally integrated object tracks. The system’s\textsuperscript{1} mean positioning error is approx. 0.37\textit{m}. A LIDAR-based indoor-localization\textsuperscript{2} was used as reference system.

Another system based on a network of surveillance cameras in a parking garage to detect vehicles and pedestrians is discussed in Einsiedler.\textsuperscript{3} The authors utilized a boosted classifier based on Haar-like features trained with front view examples of vehicles to detect them. The detection of pedestrians was performed via HoG features. The position is estimated by backprojecting the detections’ root points. Overall the system provides a 20 m coverage and 95 percentile overall positioning error below 1.5 m. Their test scenarios focus on long lanes to capture objects parallel to the driving direction.

In Evans \textit{et al.}\textsuperscript{4} a multicamera-based system for object detection is described. They utilize a grid map based on a defined ground plane and a pre-specified object height. In contrast to that, we avoid making any assumptions concerning the objects, the objects’ height, or the ground plane.

In contrast, our approach provides multiple views of an object, focuses on arbitrary object detection without any knowledge or training of objects and avoids assumptions concerning the segmented root point of an object. A root point transformation is prone to error when the detection is not accurate for our desired field of application.

3. PROPOSED SYSTEM

The proposed system is based on static surveillance cameras only. It detects and tracks object positions in the parking garage. A typical scenario captured by the surveillance cameras is shown in Fig. 1.

First, the processing chain starts with the synchronized image capturing for a single camera (\textit{cf.} Sec. 3.1). Moving objects are selected by a foreground segmentation method (\textit{cf.} Sec. 3.2). Afterwards, foreground objects are transformed into view rays in a common world coordinate system (\textit{cf.} Sec. 3.4). These view rays are intersected to receive plausible object hypotheses (\textit{cf.} Sec. 3.5). An initial intrinsic and extrinsic calibration is required once beforehand (\textit{cf.} Sec. 3.3). To establish continuous object tracks these single object hypotheses are filtered temporally (\textit{cf.} Sec. 3.6).
3.1 Camera network
To simulate a garage parking scenario with a surveillance camera system, we use multiple grayscale cameras mounted on high tripods and connected by a local area network (LAN).

We enable multi-view vision of objects by ensuring that each camera shares its field of view with at least one of the other cameras.

Simultaneous image exposure and grabbing is guaranteed by synchronizing all cameras via the IEEE1588 protocol over LAN. Different types of lenses (4.8 mm, 9 mm, and 12.5 mm lenses) are chosen according to the cameras’ position in the parking garage to achieve optimal coverage.

3.2 Image Processing
Because most parts of the surveillance camera images are background we have to distinguish between static environment and moving objects in the foreground. We also have to distinguish between real moving objects in the foreground and falsely segmented image regions due to e.g. illumination changes (vehicle spot lights) and object shadows.

The image processing pipeline is described briefly as follows: At first, we create an adaptive background representation, based on an exponentially time-smoothed mean-image, and subtract it from the original image. To detect the objects edges, we choose a fast learning setup. All overexposed pixels from the original image are segmented, separately dilated and afterwards subtracted from the segmentation result.

Segmentation errors caused by shadows are reduced by applying the normalized cross-correlation method to identify image regions that only change due to varying illumination. Remaining segmented image regions are subdivided into blocks of identical size which are clustered with adjoining blocks to allow for a coarse connectivity of single objects.

The next passages will explain each image processing step mentioned above in detail: First, we generate a representation of the background using the well-established background subtraction method presented by Jacques.\(^5\) In order to establish this representation, we utilize an exponentially smoothed mean-image:

\[
B(x, y) = B(x, y) \ast (1 - \alpha) + I(x, y) \ast \alpha
\]

where \(B\) describes the background image, \(I\) the current camera image with the same size as \(B\) and a weight \(0 \leq \alpha \leq 1\). Overexposed pixels of \(I\) are ignored in the calculation of \(B\) to reduce the negative effects of active light sources.

The binary segmentation image \(S\) represents the difference of the background image \(B(x, y)\) and the current image \(I\) using threshold \(t\):

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

Weak or strong intensity changes are diminished by extending the expression \(|B(x, y) - I(x, y)|\) via clipping it to a minimum and maximum, respectively.

Compared to our foreground segmentation (with \(\alpha = 0.05\)) presented in Ibisch\(^1\) we now aim to learn much faster (\(\alpha = 0.2\)) to detect the edges of moving objects.

Reflections of strong illumination sources cause false segmentation. Discarding overexposed pixels does not eliminate all false positives. To consider these strong illuminated regions we store all overexposed image regions selected by a threshold in a separate image, the so called overexposed image \(O(x, y)\). Several dilations were applied to the overexposed image \(O(x, y)\) to close gaps inside strong illuminated image regions and to desegment weaker illuminations on the margin of strong illuminated regions. Afterwards, the overexposed image \(O\) is subtracted from the segmentation image \(S\) to exclude all overexposed image regions from the segmentation.

The normalized cross-correlation (NCC) was applied to the segmentation image \(S\) to reduce falsely segmented image regions caused by shadows. The NCC method identifies structurally constant image regions. We divide
Figure 2. A brief visual summary of the image processing pipeline based on one exemplary camera image:
Top left: The original camera image. Top Right: The mean image after a few seconds of processing. Bottom left: The segmentation image as a result of the difference of the camera image and the mean image and afterwards the mentioned segmentation enhancement of overexposure and shadow elimination. Bottom Right: The final clustered blocks (green rectangles) based on the segmentation.

The background image $B$ and the current image $I$ into equally sized grid blocks. To reduce the workload, a block is considered only if it contains a minimal number of segmented pixels. The NCC calculates the degree of lighting-independent structural similarity between the corresponding blocks of $B$ and $I$. If the NCC is above a threshold the block of the current image $I$ is similar to its corresponding block in the background image $B$ and the complete block is discarded.

The next and final step in the image processing pipeline prepares the enhanced segmentation image $S$ that it could be processed by the image-world-transform. In contrast to Ibisch $^1$, where we generated and tracked larger connected image regions, in this new approach we decided to subdivide the segmentation image $S$ into small, variable, and identical grid blocks. This decision has several advantages: First, the whole occupied grid block area is almost equal to the segmented regions. Second, the number of view-rays in the image-world-transform increases and are evenly distributed inside a segmented image region than just transforming the corners of a big connected image region.

To guarantee the spatial vicinity of the blocks after subdividing for later processing steps and to prevent objects from decompensation a clustering is executed on the grid. The cluster algorithm operates by considering the 8-neighborhood of each block center point to the other adjacent block center points. Afterwards, every
3.3 System calibration

In order to interpret and combine detections from multiple cameras, it is essential to know their extrinsic calibration consisting of their exact relative positions and orientations. Furthermore, an intrinsic calibration, i.e., a mapping between the camera frame and the same coordinate system, encompassing the lens distortion parameters of each camera must be determined. For the latter part, we rely on the methods described by Zhang\(^6\) 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 iterations loops we optimize for all parameters, the orientation and the translation. We refer to Sec. 4 for a detailed evaluation of the calibration accuracy.

3.4 Image-World Transform

In this section, we describe the generating of view rays emerging from the respective camera center to the center of a single block by using an image-world transformation. We used the partitioning into blocks (cf. Sec. 3.2) to create multiple view rays for a single object rather than regarding only rays through the region of interest’s corners of the entire object. This method helps us to make the following procedure more robust by averaging over the huge number of view rays.

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’s view angle w.r.t. the ground plane, this method leads to strong misestimations of the points where the object’s root point touches the ground plane if the segmentation of the object’s image region is inaccurate. To circumvent this problem, we aim at fusing detected regions from several camera images with the overlapping field of view. One has to carefully approach this problem because the number of image regions in varying cameras can be diverse. 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 shown in Fig. 3.

Considering a pair of blocks from two different cameras. In a first step we compute the view rays emerging from the respective camera center through all block center points and intersect* them pairwise.

The distance in 3D space is a quality criterion on the matching blocks. Since it depends on perspective, we suggest to consider the backprojected coordinates of the intersection point into the corresponding camera frames. The distance between both reprojected block center points is the matching distance for the regarded block pair.

Formally, let \(B_i, B_j\) be two regarded blocks in two camera frames \(m\) and \(n\), \(c_i, c_j\) their respective center points and \(q_k^i\) the backprojected intersection point of \(q_k^j\) of image \(k\) and intersection point \(l\) and \(e(x, y)\) calculates the euclidean distance in image coordinates from point \(x\) to point \(y\)(cf. Fig. 3). The matching distance \(d(B_i, B_j)\) with the intersection point \(p_l\) caused by the view rays emitting through the centers \(c_i\) and \(c_j\) is then defined as

\[
d(B_i, B_j) = \frac{e(c_i, q_m^i) + e(c_j, q_m^j)}{2}.
\]

*For the sake of simplicity we use the notion of intersection also for skew lines where it refers to computing a single point in space that minimizes the distance to both intersecting lines.
Figure 3. The reprojection error: View rays, emanating from the origin of the camera (red dots), are projected through the center of a block $B_1, \ldots, B_4$ of a detected person in the left image plane $I_1$ into the world scene. Together with other view rays corresponding to the blocks of the object of image $I_2$ they are generating 3D intersection points $p_1, \ldots, p_4$ in the scene, i.e. points with minimal distance to the respective view rays, shown in blue. The backprojected points are illustrated in the camera frames by the green dashed line starting from the intersection point to the image planes. The green points $q_1^2, q_1^1, q_2^2$ on the image plane represent the reprojected points of the intersection point. As shown, Block $B_4$ did not match the object (e.g. caused by segmentation errors), but also generates the intersection point $p_2$. This false intersection point $p_2$ can be eliminated by the error: The distance of the reprojected point $q_1^2$ is closer to its block center of $B_1$ in the image $I_2$ than the reprojected point $q_2^2$. 
3.5 Detection

The detection module receives all clustered and transformed blocks from every attached camera of the network. The goal of the detection module is to aggregate all clustered blocks from every camera image and to generate reliable object hypotheses based on this aggregation.

With the help of the reprojection error (cf. Sec. 3.4) we set up the marriage problem and compute an optimal matching via the well-known propose-and-reject-algorithm. Since not every block pair corresponds to the same object we define a comparison between pairs of blocks by a distance threshold that leaves us with only those object detections that can reliably be assigned to one another: First, we define a reprojection error threshold to determine the reliability of a single intersection point.

Afterwards we consider the clustered blocks from the image processing pipeline (cf. Sec. 3.2) and calculate every possible block-cluster constellation for each camera pair. To decide which block-cluster pair we match, we calculate the average error of a cluster and store the best average error of a cluster pair. At last all stored cluster-block pairs are transferred with their intersection points into the main coordinate system.

In the next and final step of the detection module, we cluster the transferred intersection points to object hypotheses: We generate iteratively object hypotheses based on the density of a cluster spanned by the intersection points. We used several geometric shapes to calculate the density of a cluster: cylinders, circles, or a convex hull. The best results were caused by a circle shape. To decide which cluster will be processed to the next modules we used the parameters of a minimal amount of intersection points inside a cluster and the clusters’ minimal density.

To stabilize the clustering we trace the result of a previous tracking step (cf. Sec. 3.6) back to the detection module to get an initial estimation for the positions of the world-clusters.

An exemplary visualization of the detection module is shown in Fig. 4.

3.6 Tracking

The tracking module has to guarantee a complete temporal integration, either based on an observation, or a plausible prediction and closes gaps where no valid hypotheses are generated. It receives the hypotheses (which consists of all points that were placed in a single cluster) from the image-world-transformation. Each hypothesis is transformed into a two-dimensional normal distribution where \( \mu \) is the center and \( \Sigma \) is the covariance of the cluster. Now \( \mu \) can be used as the primary tracking object while \( \Sigma \) describes the extent of the tracked object. A track consists of such a normal distribution with some additional information like the number of updates the track has received.

A two-dimensional filter, which consists of two one-dimensional Alpha-beta-filters – one for horizontal and vertical movement, was used. The standard configuration of such a filter primarily allows tracking of the position \( \mu \). To incorporate \( \Sigma \), a simple extension has to be made: When the next position of a track is predicted, the \( \Sigma \) of the current track is assigned to it with the assumption that tracked objects have consistent sizes. However, the shape of tracks can vary due to orientation changes or inaccurate hypotheses. Therefore, \( \Sigma \) is adapted by exponential smoothing whenever a track is updated with a new observation.

The first step of the tracking process is to predict the position of all maintained tracks in the current time-step. This leads to a bipartite matching problem between the predictions and the hypotheses which can be solved by applying the propose-and-reject-algorithm. Different distance measures for normal distribution like the Kullback-Leibler Divergence and the Bhattacharyya distance were investigated. However, the following metric achieved the best results for

\[
\begin{align*}
d_{e_1} &= \min((e_{11} - e_{21}), (e_{11} - e_{22})) \\
d_{e_2} &= \min((e_{12} - e_{21}), (e_{12} - e_{22})) \\
d &= w \cdot (\mu_1 - \mu_2) + (1 - w) \cdot (d_{e_1} + d_{e_2}),
\end{align*}
\]

where \( d_{e_1} \) and \( d_{e_2} \) are the minimal distances between the eigenvectors \( e_i, e_j \) of the covariance matrices of two consecutive time steps and \( w \) is an adjustable weight factor.
Afterwards, implausible matchings with distances that exceed a certain threshold are removed. Some hypotheses might be unmatched even though they are close to a predicted track because it has already been assigned to another hypothesis. This can happen if points belonging to a single object are divided into several clusters. If that is the case, the distributions of these hypotheses are retroactively merged into a single one.

Now each track can be updated with its assigned hypothesis using the aforementioned Alpha-beta-filter\textsuperscript{10} to smooth the movement of the track. If a track could not be matched to a hypothesis, it is updated with its own prediction as long as the number of time-steps without a matching hypothesis does not exceed a certain threshold, otherwise the track is deleted. Usually this threshold should depend on the number of updates the track has already received. For each unmatched hypothesis a new track is set up. The new track is regarded as uncertain until it has received a certain number of consecutive updates.

It might happen that a single object produces several tracks that are close to each other. The merging of distributions of updated tracks is executed, if their distance falls below a certain threshold for an extended period of time. It is noteworthy that the merged distributions are not maintained as individual tracks and the filter is still working on the underlying tracks.

After the update process a list containing all stable – and possibly merged – tracks is returned by the tracking module. This list can be used as an additional feedback input to the cluster module to get an initial estimation for the positions of possible clusters. An exemplary visualization of the tracking module is displayed in Fig. 5.

4. EXPERIMENTS

This section presents the evaluation and comparison of the camera system with a LIDAR-based reference system. Both systems are deployed in a parking garage that serves as proving ground.
Figure 5. The result of the clustering and tracking algorithm is displayed in this figure: The yellow intersection points are the same as in Fig. 4. The yellow ellipse represents the track based on these intersection points.

4.1 Reference System

Due to the lack of GPS information inside indoor environments, we used the LIDAR system, presented in Ibisch,\textsuperscript{2} to gather reference data. The system is based on an array of distributed LIDAR sensors, aligned parallelly a few centimeters above the ground, to detect and measure the distance to nearby obstacles. The systems learns to distinguish between static and dynamic measurements. The setup is then used to recognize and accurately track vehicles and general objects.

Ibisch et al.\textsuperscript{1} determine the distances between the trajectories produced by the LIDAR-based system and by a precise Differential-GPS (DGPS) with a mean euclidean distance of 0.19\textit{m}. Thus, being highly accurate, these object detections are used as reference data in the following evaluation.

4.2 Setup

Two GigE-Vision Prosilica AVT GT 1380 monochrome cameras were installed to map a realistic parking garage scenario, with an image resolution of $1360 \times 1024$, equipped with 9\textit{mm} lenses mounted on a 2\textit{m} tripod. Both were positioned vis–\(\acute{a}\)–vis and share an intersecting field of view.

The reference system uses two SICK LMS 500-20000 pro LIDAR sensors.

4.3 Calibration

For a precise determination of calibration points, a Leica Builder 306 tachymeter was used. A distinct world origin was defined, measured, and the origin and all measured calibration points were transferred into a CAD representation of the parking garage environment. The same CAD representation for the described system and for the LIDAR reference system was utilized to guarantee a valid comparison.
4.4 Results

For a demonstration of both systems’ trajectories a representative sequence of a pedestrian was chosen. The sequence was recorded with 15 fps. A single frame result is shown in Fig. 7, a camera image of this result is shown in Fig. 6. The origin of the coordinate system is in the left bottom of the image. Both trajectories are presented in Fig. 8. The pedestrian starts at position of (31, 30), walks to (34, 30) and back to (31, 30).

The sequence exemplifies four situations:

- Time frame 0–150: Moving pedestrian in front of a static background.
- Time frame 151–200: Pedestrian overlaps with the vehicle in one camera frame.
- Time frame 201–260: The vehicle switches on its light and the image becomes overexposed.
- Time frame 261–380: The vehicle turns off its light and the pedestrian walks around.

This composition of different scenarios within the sequence is documented in Fig. 9 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 0.24m with a standard deviation of 0.19m.

A discussion of the results is presented in Sec. 5.

5. CONCLUSION AND OUTLOOK

This paper presents a multiple-camera surveillance system installed in a parking garage to detect arbitrary moving objects. Objects in every camera image are segmented using a background representation and enhancements are applied to this representation to reduce noise. To generate precise and plausible hypotheses in world coordinates the system calculates intersection of view rays of these objects and tracks them in a world representation. The detection of generic objects of arbitrary size can be performed without prior training.
Figure 7. Both systems’ results within the experimental setup. It is the same scene as shown in Fig. 6. 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 the presented system’s final hypothesis.

Figure 8. The plots illustrates the trajectories of the pedestrian-centers in world coordinates: Reference data by the LIDAR-based system in green, camera hypothesis in red. At position (32.5, 30) the vehicle switches on its light and causes a stronger positioning error.

Figure 9. The deviation of the LIDAR and the camera-based hypothesis. The red vertical lines separate four situations mentioned above.
The system’s mean positioning error in a sequence containing several difficult situations is 0.24 m. Similar camera-based systems, proposed in the literature report higher deviations, e.g. Einsiedler\textsuperscript{3}, with a positioning error of 1.5 m. This is, however, very dependent on the scene geometry and the chosen camera setup.

The system is indeed precise enough to locate an object for applications like collision warning. 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 conclusion, the system provides good results if objects are spatially separated in the camera images. In the presence of occlusion it is in many cases not possible to disjoint them with the presented method. In the future, a classification of segmented image regions with a detector or a texture analysis of the blocks can provide more information on matching view ray pairs, thus, accelerating and stabilizing their intersection.

REFERENCES


