Vehicle Localization and Classification Using Off-Board Vision and 3D Models

Stefan Hoermann and Paulo Vinicius Koerich Borges, Member, IEEE

Abstract—Vehicle localization is one of the most common tasks in robotics. Using vision as a sensor, most methods perform localization from cameras mounted on board the vehicle. In contrast, we propose a method based on an off-board camera. The system uses a similarity measure between a camera image and a synthetic image generated using a 3D vehicle model, ideally converging to the true pose of the vehicle. The similarity measure is based on a model of shading appearance which depends on the surface curvature of the 3D model. Considering that the area observed by the camera is fairly planar, a rough initial estimation for the position of the object can be obtained using 2D blob tracking. The 3D model of the vehicle can be rendered near the vehicle in the real image and, using the proposed similarity measure, it converges to the correct pose. A classification function to discriminate between different vehicles is also proposed, making it possible for the system to identify and track multiple vehicles of interest. A number of experiments are performed with different vehicles outdoors, in a real industrial environment, considering different illumination conditions. The low average error rate compared to a laser-based ground truth illustrates the applicability of the method.

Index Terms—Vehicle localization, 3D model, off-board surveillance camera.

I. INTRODUCTION

Reliable pose estimation is an essential component of autonomous navigation systems. To tackle this problem, many methods have been proposed using lasers, radar, radio-frequency identification (RFID), or GPS. In the last decade, significant attention has been given to camera sensors, from which pose estimation is obtained through the use of computer vision techniques. While research towards high-precision localization systems using vision-only is still ongoing, cameras are increasingly being considered to effectively replace, or be combined with, other sensors. Some of the advantages of cameras are that they are passive devices and can work in areas with high magnetic fields or indoors, situations in which RFID or GPS fail. These situations are common in industrial environments, which is the scenario where the proposed work is likely to be most relevant. Motivated by the applicability of laser-based localization systems [1], [2] that have been successfully implemented in industry, the goal of this paper is to provide a vision-based solution. In industrial areas, vehicle localization can be useful for general situation awareness, safety, fleet management, and possibly vehicle control. Because laser-based localization has proven useful for localizing known vehicles in industrial environments, it has led to additional research on solutions using independent sensors; in our case, cameras. In contrast to most vision-based localization methods in which the camera is on board, we use an off-board camera to identify and localize the vehicle. A vision-based off-board localization method can be a cheaper (since cameras are less expensive than lasers and very often already mounted on infrastructure), passive and alternative (or possibly redundant) solution.

We consider the problem of ground vehicle navigation, assuming that the traversed area is observed by an off-board monocular camera. For precise pose estimation, we fit a projected 3D model of the object to the real camera image. Ideally, the fitting converges to the true vehicle position and orientation in the world. Figure 1 illustrates the concept. As mentioned, the proposed system is particularly applicable (and has been tested) in industrial areas, aiming at safety, assisted—and possibly autonomous—navigation, and situation awareness. In contrast to public areas where vehicle tracking and identification should be general (for all of the many possible types of vehicles), in an industrial environment the use of a predefined model can be seen as a practical solution. In addition, 3D models for cars, trucks, and other vehicles can be easily generated in CAD applications or even retrieved from the Internet\(^1\). Using the method proposed in this work, we estimate the pose of a known object without information about its initial pose or the need for a statistical training stage, common in machine learning frameworks. The method also performs classification, assigning the corresponding 3D model to the recognized object and only tracking the objects (vehicles) of interest. An illumination model is not required, and the method is fairly independent from shadows. Unlike the scenarios assumed in many state-of-the-art 3D object tracking methods, important aspects of the application considered include the following: (i) a high resolution view of the object is often not available, (ii) illumination conditions can change severely as the systems must also operate outdoors, and (iii) automated initialization is a requirement. These issues make the use of traditional feature-based and convex optimization methods


\(3\)The link: http://sketchup.google.com/3dwarehouse/ contains thousands of 3D CAD models of popular urban, military and industrial vehicles.
challenging and not always effective, as most require fairly good initialization not to fall in local minima. For these reasons, using the shading appearance extracted from a 3D model, as proposed in this paper, is an effective solution for finding correct poses and providing reliable localization of vehicles in outdoor, real-world scenarios with challenging illumination conditions. A preliminary version of the system [3] also used an off-board camera and a gradient difference to perform localization. In this paper, however, a new cost function is proposed, improving the accuracy of the algorithm. The new search strategy also significantly reduces the computational complexity, and a new classification function is proposed, extending the system to identifying multiple objects.

More specifically, for pose estimation we fit a projected 3D model of the object to the camera image, combining edge geometry and shading appearance-based fitting. The shading appearance follows a model based on curvature by assigning different gray-scale values to different surfaces of the model. Therefore, the gray-scale value assigned to each surface depends on the direction of a vector normal to that surface. To compare the generated synthetic image with the input image from the camera, the direction of the gradient in the 2D images is used. The new cost function proposed is applicable for a different class of object shape, where no shading appearance is visible for some viewpoints due to large plane surfaces. We also use this cost function for classification to identify the target via best fit.

In order to estimate the initial pose of the object, motion segmentation is performed from a calibrated camera and combined with the comparison between synthetic and real images. Once the initial position is estimated, a Kalman filter is employed for tracking, and its output can also be used to reduce the search space, lowering the average error and speeding up the computation process.

We perform experiments with several different vehicles. In particular, we test the system with a 20-tonne autonomous forklift truck—a hot metal carrier (HMC)—in an outdoor industrial environment under different lighting conditions. This forklift has a high-precision laser localizer [1], which is used as ground-truth, enabling us to quantify the evaluation, unlike most works on video tracking presented in the literature. The results show that the method can be used as a reliable method for vehicle localization, providing a low average error. Other vehicles used in the tests include a small electric car, the Gator, and a skid steer loader (Bobcat), shown in Figure 2.

Fig. 1. Concept of the pose estimation method. The projected 3D model is aligned with the object in the camera image. The best fit returns the object position and orientation.

Fig. 2. Sample images of the objects used in our experiments: HMC, Gator and Bobcat (left to right). The first row shows the camera images. The synthetic images of the corresponding 3D models are shown underneath. The edges of the 3D models are painted in white on the camera image to illustrate the fitting result.

We argue that the system is most relevant in industrial areas with known vehicles, environments in which vehicle tracking is becoming increasingly popular.

This paper is organized as follows. In Section II we present an overview of related work, contrasting with the contribution presented in this paper. In Section III we give an outline of the localization framework. In Section IV we present a thorough description of the comparison between a synthetic image of a projected 3D model and a real camera image, aiming at pose estimation. We describe the system initialization and discuss how the method can be used for classification in Section V. In Section VI we show a number of experimental results, followed by conclusions and direction for future work in Section VII.

II. RELATED WORK

Several vision-based approaches for pose estimation have been proposed in the literature. Most of them are based on shape, contour, or appearance, fitting models either in 2D (patterns of learned models) or 3D (projected 3D models) onto an image. The pose fitting techniques use corresponding feature data (natural or fiducial), image segmentation, or comparison of learned or modeled image data.

Early work by Lowe et al. [4] discusses the principles of pose estimation using a priori knowledge about the object, as it occurs in human vision. They present a strategy for projection-based pose fitting using line-to-line correspondence between edges in the camera image and edges of the projected 3D model. Closed form solutions to fit projected points or lines of a 3D model to corresponding points or lines in a 2D image with respect to the pose of a 3D model have been proposed

\[\text{John Deer's all-terrain utility vehicle.}\]
A limitation of these approaches is that the feature correspondences must be known. A challenge in feature correspondence–based methods is the assignment of corresponding points or lines between the projected 3D model and the 2D image. To mitigate this problem, methods like RANSAC [6] are often employed. Yoon et al. [7], for example, follow this strategy to track an object using straight-line features. In their work, however, only tracking is discussed, and for initialization the object pose must be adjusted manually using a graphical user interface (GUI). To alleviate the correspondence–finding problem, assumptions can be made restricting the class of shapes applicable for the correspondence assignment, to contain, for instance, parallel lines and triplet–line junctions. In the work by Ray [8], the pose of a cuboid and a T-block is estimated according to this approach. Segmented edge elements are used as corresponding line features. Assuming that the viewing angle and approximate position of the model is known, the midpoint, orientation, and length of a straight line of the model is used to find the corresponding lines in the camera image. The line-to-line assignment works well with few and exclusively straight lines that describe the used objects completely. It requires, however, a relatively high resolution, with the object covering a large part of the image.

In the field of object recognition, significant work has been proposed in literature using learning–based methods to recognize 3D targets [9], [10]. The recognition of a 3D object requires the consideration of the object pose. Pose estimation with a manifold learning–based approach uses a statistical analysis of sample images to extract features via dimensionality reduction. The work by Arie-Nachimson and Ronen Basri [11] employs “implicit shape models” to identify 3D objects from 2D images. The model consists of a set of learned features, with their 3D locations and the views in which these features are visible. A drawback of learning–based methods is the required training period, which can be a time–consuming process. In the considered application, if the object is observed in conditions dissimilar to the learned scenes, the system can easily fail.

Schmalz et al. [12] identified that pose fitting and image segmentation using 3D models can be effectively combined. Using appearance–based segmentation, they combine a region-based image segmentation with pose estimation. The region-based image segmentation divides the object from the background according to color and texture statistics. The 3D model is projected onto the image, and the pose yielding the best segmentation is assumed as the optimal pose estimate. A cost function is minimized with respect to the pose of the object, to contain a maximum number of pixels fitting into the local Gaussian color model of overlapping pixels between 3D model and camera image. The color model is updated with the new matched pose. The method performs well with targets of a color that stands out from the background. In an outdoor environment, however, the color model must consider a high variance due to the different lighting conditions. A possible approach for matching 3D models to 2D images following a similar strategy is implemented in [13]. Statistics of texture and color are extracted from the real object image either at initialization or in a previously trained model. A particle filter is used to compare projected pixel values of the 3D model to normalized pixel values of the camera image for different poses. The initial pose is assumed to be known in order to map initial pixel values to 3D model points. To reduce the problem of changing illumination in appearance–based methods, appearance–based tracking can be combined with geometry–based tracking and an illumination model [14]. While the appearance–based segmentation approaches assume distinguishable image statistics from object and background, a foreground/background segmentation based on motion is also possible [15]. In this work, the projected silhouette of a 3D wire frame model is compared to the foreground silhouette for classification of moving objects. The wire frame models of vehicles are placed in a predefined grid on a planar ground with a known homography between ground plane and camera. Observing a one-way road, the vehicle’s trajectory is assumed to be known and therefore the orientation of the object is assumed to be constant. For classification, the overlap between the foreground silhouette and the projected model silhouette is maximized with respect to different models and positions. The system is applied in an outdoor environment and shadow removal is required. However, the classification fails if objects are partially occluded or if objects cross the previously defined vehicle trajectory. The foreground/background segmentation often returns a corrupted object silhouette due to changing lighting conditions, making it difficult to determine the pose even for the human eye. As an alternative to wire frame models, Rosenhahn et al. [16] proposed the use of free-form contours. In their approach, a contour is approximated with a series of periodic curves and fitted into the object silhouette. It is assumed that the object silhouette can be segmented from the camera image. A disadvantage of the method is that the image segmentation (possibly based on color, texture, edges, or motion) can be faulty if the object has a similar color to the background or if illumination conditions vary.

The illumination conditions in an outdoor scenario are one of the most challenging problems in the application of outdoor object detection and pose estimation. Image gradient and edges are robust to changing illumination conditions and therefore widely used in outdoor applications. Apart from line-to-line correspondence–based approaches, the similarity measure in hypothesis–driven approaches typically relies on geometry or shading appearance change (the gray level gradient), which can be extracted easily from the shape of the 3D model without training, in contrast to color- or texture–based methods. Following this approach, a geometry–based method using edge fitting is proposed in [17]. The edge-image of an on-board camera is matched to a projected image of a pre-recorded 3D map of distinctive building edges to maximize overlapping edges. Although straight lines which are spread over the entire area were used, the implementation on a GPU with a particle filter is required for the matching process. A drawback is the need for previous recording of the 3D wire model containing the edges of the operation area, which can be expensive and time-consuming. The framework presented in [17] was also used to estimate the pose of submerged structures with respect to the camera [18]. In this work, a direct comparison of
the gradient is possible, as the illumination conditions can be modeled easily due to a known single light source in an otherwise dark underwater environment. Petit et al. [19] recently proposed a method to extract edges of a projected 3D model by considering discontinuities in the depth buffer of the rendered scene. GPU programming is used to achieve frame rates between 12 fps and 17 fps, and equivalently to our approach, a normal map is used where an object surface is colored depending on its normal vector. In contrast, however, they use only edges, while in our method the gradient direction of smooth curved surfaces is also considered.

Similarly to our method, off-board approaches where a stationary surveillance camera observes a traffic scene have been considered [20]–[24]. In [20], the object pose is recovered using wire-frame models. Assuming rectilinear shape of the objects and a known region of interest, a limited number of object orientations can be hypothesized on basis of the image gradient direction. This solution can be refined with the application of a Kalman filter for tracking, as suggested by Lou et al. [21]. For the pose refinement via edge fitting, a variation of the Mahalanobis distance is used in their work, although automated initialization is not addressed. The works by Nagel et al. [22], [23] also considered a traffic scene scenario. The vehicle pose estimation can be performed via line-to-line correspondences [22], and significant improvements can be achieved by comparing the numerical value of the gradient magnitude of edges [23]. For this comparison, the gradient magnitude of edges in the synthetic image is subtracted from the gradient magnitude in the real camera image. The difference is minimized with respect to the pose of the object. Similarly to our approach, the system uses motion analysis to gain a rough initial position guess. However, as this approach assumes equal gradient magnitude in the real and synthetic images, an illumination model and rendered shadows are required in the generation of the synthetic image. Although edges and gradient are robust to changing illumination, these approaches typically require the inclusion of object shadows in the model to perform satisfactorily, similar to other edge-based fitting methods [25], [26] outside of the vehicle–tracking domain. Shadow removal as used in [15] can be considered as an alternative, but this is also challenging with varying lighting conditions. In our approach we achieve high robustness to shadows and changing illumination conditions by considering the object as a Lambertian scatterer and modeling on-vehicle shading gradient only based on the object shape. The Lambertian reflectance model [27] (discussed in more detail in Section IV) is a reasonable approximation for surfaces in many practical problems and has found applicability in other robotics and vision tasks, such as visual servoing [28] and surface reconstruction [29], [30]. The works by Tan et al. [21], [24] present precise pose estimation with results similar to the ones presented in this paper. The ground truth, however, is only visual, making it also hard to quantitatively compare the methods. For additional references the reader is pointed to the survey by Petit and Fua [31], which also covers fundamental concepts used in monocular model-based 3D tracking.

In this paper, hypotheses-driven fitting through image comparison is used to estimate the pose of an object. The gradient direction is employed as a similarity measure. We argue that the focus on the gradient direction can be seen as an advantage, as an illumination model is not required, which makes the system robust to different illumination conditions and shadows. Given that the comparison is not limited to gradients with high magnitude, the analysis is not limited to the use of edges only. In particular, curved surfaces of the 3D models are important features in the comparison algorithm. Since the initial search can be a time-consuming process in a large search space, the system employs methods from foreground segmentation and motion detection and assumes a known homography between the camera plane and the ground plane, reducing the initial search space. After this initial pose search, the system is independent of image segmentation and motion analysis, as discussed in the following sections.

III. LOCALIZATION FRAMEWORK

In this section we summarize the main modules that must be considered in the design of the system proposed. The four essential elements are the 3D models, the pose search, the system initialization, and the tracking filter, as described here.

A. 3D CAD Models

We use common 3D CAD models to predict gradient direction based on the curvature of the 3D models, including their edges. Sample images of the models and the corresponding vehicles are given in Figure 2. Details about the model dimensions and the number of vertices are given in Table I. A detailed description on how the gradient is modeled on the basis of the model curvature is given in Section IV-A.

<table>
<thead>
<tr>
<th>Model</th>
<th>HMC</th>
<th>Bobcat</th>
<th>Gator</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dimensions (m)</td>
<td>6 × 2.6 × 3.6</td>
<td>3.6 × 2 × 1.8</td>
<td>2.7 × 1.6 × 1.5</td>
</tr>
<tr>
<td>Vertices</td>
<td>5296</td>
<td>21956</td>
<td>3392</td>
</tr>
</tbody>
</table>

B. Pose Search

A synthetic image of the 3D model is rendered using the projection matrix of the real camera. The position and orientation of the 3D model is moved over a grid around an estimate (according to the Kalman filter discussed in Section III-D) of the true object pose. The rendered image is then compared to the real camera image for a refined position and orientation estimation. The cost function developed in this paper is ideally at its minimum when the 3D model fits best to the object in the camera image. Although the system can fundamentally estimate a pose with 6 degrees of freedom (DOF), in our implementation the flat-ground assumption is used to reduce the dimension of the search-space to 3 DOF, speeding up the process. Details on the cost function and the metric to compare the synthetic and real images are given in Section IV.
C. Initialization and Identification

For initialization we assume a known homography between the image plane and ground plane, such that recognized 2D motion can be transformed to the 3D domain. A pose search is triggered from this rough 3D position. Using the similarity metric described in this paper, classification can be performed via best match. A more detailed explanation is given in Section V.

D. Tracking Filter

A Kalman filter is used to get higher localization precision and to reduce the search space for the matching algorithm. A kinematic model is stored with the 3D model. Kinematic models fitting different types of objects and tracking filters can be found in [32]. We make use of a model with constant velocity and yaw rate which fits well for several ground vehicles. This model incorporates fundamental vehicle constraints (i.e., the physical characteristics of the vehicle like speed and steering inertia) under typical driving operations. The acceleration is modeled as additive white Gaussian noise. The predicted pose vector of the tracked target is used as the center coordinate for the new search space which is reduced or enlarged according to the filter covariance matrix.

IV. SIMILARITY MEASURE BETWEEN AN OBJECT IN A REAL AND IN A SYNTHETIC IMAGE

To tackle the problem of vehicle localization, we present a vision-based algorithm that estimates the position and orientation of an object by using a 3D model and an off-board camera. The algorithm compares a projected 3D model with the object in the camera image according to the shading appearance. A similarity measure between an object in a synthetic gray-scale image $S_p$ and a real camera image $I$ is returned, so that the projected 3D model can be aligned to the object in the real image with respect to its pose $p = (x, y, z, \psi, \phi, \theta)^T$. Here, $x$, $y$, and $z$ denote the three spatial coordinates and $\psi$, $\phi$, and $\theta$ the yaw, roll, and pitch angles, respectively. The index $p$ indicates dependence on the pose hypothesis during the matching process. Using various 3D models, classification is performed via best match. In this section we provide a detailed explanation of the cost function for pose matching and classification.

A. Shading Appearance Model

Our comparison algorithm relies on a model of shading appearance based on curvature. Although a 3D model could provide color information, due to illumination or dirt the color and texture of the model can differ greatly from the color and texture of the real object. To reduce this problem, the method works with gray-scale image gradients. The gradient of a gray-scale image is considered as a metric for the appearance change of shading. The main motivation for using the proposed method is that, under most lighting conditions, edges and curved surfaces cause a change of image intensity, as exemplified in Figure 3(a). The photograph shows a book with a planar cover and a curved spine. Inside a planar surface of the cover, the gray-scale value is nearly constant. The transition between the individual faces shows a sharp intensity change, while on the curved spine the gray value changes smoothly. This can be explained by Lambert’s cosine law. Many rough surfaces can be assumed to be Lambertian surfaces. If a Lambertian surface is illuminated by an external light source with a constantly emitting luminance $L_0$, the light scattered from the surface has the same radiance in all directions. As a consequence, the emitting radiance is proportional to the incident radiance $L = L_0 \cdot \cos(\alpha)$, with $\alpha$ being the angle between the surface normal and the illuminating source. As the position of a light source is constant in an image, the gray value of a Lambertian surface in a camera image depends only on the surface normal vector. Our 3D models are composed of planar surfaces. Curved elements in the 3D models are composed of a series of planar surfaces. Figure 3(b) explains the assembly of a curved surface with a basic example. The drawing illustrates the top view of the book spine. The real object’s surface seen in the camera image is drawn with a continuous green line. The numbers on the inside of the green line represent the gray-scale values of the surface seen in Figure 3(a). The top view of the object model is drawn with dashed red lines with an offset to the real object’s surface. In the model, the curved book spine is composed of 5 planar surfaces. The green arrows along the object’s surface indicate the gradient in the camera image, or the intensity change. The normal vectors of the model’s plane assembly surfaces are shown as black arrows on the corresponding surface. Since the object surface is a Lambertian scatterer, the intensity of reflected light changes with the orientation of the reflecting surface. In consequence, we use the changing direction of normal vectors of adjacent plane model surfaces to represent the intensity change on the corresponding area of the real object in the camera image. Since our models are composed of a limited number of plane surfaces, we model the intensity change at quantized regions in the synthetic image by assigning different gray-scale values to adjacent surfaces with different normal vectors. In Figure 3(b) the regions where the normal vector in the model changes are marked with either yellow or blue oval areas, where yellow indicates the sharp intensity changes interpreted as edges and blue indicates smooth intensity transitions from curved surfaces. An efficient way to assign different gray-scale values to adjacent surfaces is to assign every assembly surface first to a three–component color value $c = [R, G, B]^T$ which corresponds to the three–component normal vector $n = [n_x, n_y, n_z]^T$. While the components of $n$ can range from $-1$ to 1, they are scaled to $[0, 255]$ before being assigned to the corresponding color components. The synthetic image $S_p$ is then generated by converting the synthetic image with the color assignment $c \equiv n$ to a gray-scale image. An example of a synthetic image with this color assignment and its gray-scale version is given in Figure 4. Using this particular gray value assignment, however, the gradient direction expected in the real image is known only up to a possible $180^{\circ}$ error (i.e. the gray value transition from a surface $s_1$ to a surface $s_2$ can also appear in the opposite direction from $s_2$ to $s_1$). Any given color assignment leads to this sort of uncertainty, unless more assumptions about the object texture or the illumination
5 with an offset to the real object surface. The curved spine is modeled with a
in Section IV-A.

Based on the gradient direction, using the model assumptions
model fits best. For this purpose, we propose an algorithm
synthetic and real images to determine which image of the
B. Similarity Measure

on the similarity measure, as long as the resulting gradient
and magnitude of the real and synthetic images, the partial
processing, making the method robust to different lighting
conditions can be made, which is not always possible. We
show in the next section that this uncertainty in the calculation
of the similarity measure can be eliminated with additional
processing, making the method robust to different lighting
conditions, including sharp shadows. Finally, it is important
to note that the similarity measure does not rely on the
gradient magnitude; it relies only on the gradient direction
with respect to the opposite direction. Therefore, weighting
the components of \( n \) with different importance, has no impact
on the similarity measure, as long as the resulting gradient
magnitude is different from zero.

B. Similarity Measure

In this section we quantify the similarity between the
synthetic and real images to determine which image of the
3D model fits best. For this purpose, we propose an algorithm
based on the gradient direction, using the model assumptions
in Section IV-A.

1) Gradient Direction: To obtain the gradient direction
and magnitude of the real and synthetic images, the partial
derivatives of the camera image \( I \) and the synthetic image \( S_p \)
are calculated by convolving \( I \) and \( S_p \) with the horizontal
and vertical Sobel matrices. The result of the convolution is
the Sobel gradient images \( G_{h,1}, G_{v,1}, G_{h,S_p}, G_{v,S_p}, \) where
the subscripts I and S denote the real and synthetic images,
respectively, and \( h \) and \( v \) denote the horizontal and vertical
Sobel gradients. By transforming the Sobel images at every
pixel \( (i,j) \) into polar coordinates, we obtain the gradient
direction

\[
\Phi_1(i,j) = \arctan2(G_{v,1}(i,j), G_{h,1}(i,j))
\]

and the gradient magnitude

\[
R_1(i,j) = \sqrt{G_{h,1}(i,j)^2 + G_{v,1}(i,j)^2}
\]

2) The 180° Rotation: As described in Section IV-A, the
synthetic image provides the expected gradient direction in the
real image only with a possible 180° rotation. An example is
given in Figure 5. The figure shows the real image of a cube
and the corresponding synthetic image. The arrows delineate
the gradient direction at a corresponding pixel \( (i,j) \) in the real
and synthetic images, colored in green and red, respectively.
To consider the 180° uncertainty, the gradient direction of the real
image is compared with the gradient direction of the synthetic
image and its 180° shifted variation. Also, the cyclic repetition
of the gradient direction beyond \( 2\pi \) needs to be considered in
the calculation of the similarity measure, as discussed next.

3) Gradient Direction Distance: The gradient direction
distance is the smallest angle between \( \Phi_1(i,j) \) and \( \Phi_{S_p}(i,j) \)
or its \( \pi \) shifted version. To calculate the gradient direction
distance, all \( \Phi(i,j) \) are first rotated to the upper half-plane
with

\[
\tilde{\Phi}(i,j) = \begin{cases} 
\Phi(i,j) & \text{if } \Phi(i,j) < \pi \\
\Phi(i,j) - \pi & \text{if } \Phi(i,j) \geq \pi 
\end{cases}
\]

After this rotation, the transitions \( \pi^- \to \pi^+ \) and \((2\pi)^- \to 0^+ \)
in \( \Phi(i,j) \) are transformed into a transition \( \pi^- \to 0^+ \) in
\( \tilde{\Phi}(i,j) \). To get the shortest distance,

\[
\begin{align*}
D_{1,p}(i,j) &= \mid \tilde{\Phi}_{S_p}(i,j) - \tilde{\Phi}_1(i,j) \mid \\
D_{2,p}(i,j) &= \mid \tilde{\Phi}_{S_p}(i,j) - \tilde{\Phi}_1(i,j) - \pi \mid \\
D_{3,p}(i,j) &= \mid \tilde{\Phi}_{S_p}(i,j) - \tilde{\Phi}_1(i,j) + \pi \mid
\end{align*}
\]

must be compared. The gradient direction distance is then
given by

\[
D_p(i,j) = \min\{D_{1,p}(i,j), D_{2,p}(i,j), D_{3,p}(i,j)\}
\]

We consider \( D_p \) as the similarity image, with sample points
(pixels) \( D_p(i,j) \). An example for \( D_p \) is given in Figure 6. In
Figure 6(b), the 3D model of the Bobcat is drawn in white
to illustrate the fitting result. The corresponding similarity
image \( D_p \) is shown in Figure 6(a). It illustrates \( D_p(i,j) \) with
different colors. A blue pixel indicates a low \( D_p(i,j) \), which
can be interpreted as a sample point of the synthetic image,
fitting “well” to the pixel in the real image. A red pixel on the
other hand indicates a high \( D_p(i,j) \), which can be interpreted
as a sample point of the synthetic image, fitting “badly” to
the pixel in the real image. \( D_p(i,j) \) only exists for pixels

![Fig. 3. Example for an object and the composed model. Figure (a) shows a gray-scale image of a book with planar book cover and a curved spine. Figure (b) shows a drawing of the book. The continuous green line outlines the top view of the book including its curved spine. The numbers represent the gray-scale value in the camera image, and the green arrows along the surface indicate the gradient. The 3D model surface is drawn with dashed red lines with an offset to the real object surface. The curved spine is modeled with a series of 5 planar surfaces.](image)

![Fig. 4. The color assignment in Figure (a) depends on the normal vector of the object surface. As a consequence, gradient appears on surfaces with curvature in the corresponding gray-scale image in Figure (b).](image)
Fig. 5. The three images show (a) the original 3D model colored according to the surface orientation, (b) the gray-scale version of (a), and (c) the real image. The arrows illustrate the gradient directions at one pixel in the synthetic image (middle) and in the corresponding real image (right).

Fig. 6. Similarity image $D_p$ for the Bobcat. The gradient direction distance $D_p(i,j)$ is illustrated with different colors in Figure (a). A low $D_p(i,j)$ indicates a good fitting pixel. Figure (b) illustrates the matching result and (c) shows nonzero gradients in white.

(i,j) with a nonzero gradient $R_{sp}(i,j)$. Figure 6(c) shows the binary gradient mask. In this image, a pixel is white if $R_{sp}(i,j) \neq 0$ and black if $R_{sp}(i,j) = 0$.

C. Matching Cost

The similarity image $D_p$ indicates the matching cost at single pixels $(i,j)$ satisfying $R_{sp}(i,j) \neq 0$. In order to obtain a meaningful similarity measure, the ensemble of all $D_p(i,j)$ are interpreted jointly. A possible solution is to use the arithmetic mean $\delta_p$ as a cost function [3] with

$$\delta_p = \frac{1}{N_p} \sum_{(i,j) \in \{ (i,j) | R_{sp}(i,j) \neq 0 \}} D_p(i,j)$$  \hspace{1cm} (6)

where $N_p$ is the number of pixels $(i,j)$ with $R_{sp}(i,j) \neq 0$. By minimizing $\delta_p$ with respect to $p$, we can reliably estimate the pose of the object. In the present work, however, we propose an alternative cost function approaching a different class of object shapes with a strongly varying number of sample points. This is especially important if a object presents a shape so that in a given pose only a few sample points are visible in a basic structure, like a single line (in an extreme case). Such pose is likely to yield a significantly lower arithmetic mean $\delta_p$ compared to one which generates many sample points with complex structure or that are spread randomly over the object silhouette. In the worst-case scenario, in one pose only a single point is visible satisfying $R_{sp}(i,j) \neq 0$. This point can be matched easily to a pixel with an equal gradient direction in a natural image, which yields a $\delta_p = 0$. To counteract this positive discrimination of poses with a low number of $N_p$, we use a cost function that considers the variation of $N_p$ in relation to the object silhouette area $A_p$. Since the silhouette area is also variable with the pose, we define $C$ to be the largest silhouette area in a limited search space in the surrounding of an initial pose guess. In our pixel-wise comparison we use now $C \geq N_p$ sample points including $N_p$ pixels obtained from the object curvature, and additional $C-N_p$ pixels which can fill the planar parts of the silhouette area. The cost function is then a weighted sum of $C$ sample points, where $N_p$ sums are defined by the similarity image $D_p(i,j)$ and $C-N_p$ are sums with no information about their matching cost. Since $D_p(i,j)$ ranges from 0° to 90°, we define the extra $C-N_p$ sums with a value of $\delta_n = 45°$. We consider $\delta_n$ as the neutral value, since a $D_p(i,j) < \delta_n$ reinforces the pose hypothesis and a $D_p(i,j) > \delta_n$ weakens the pose hypothesis. The new cost function is therefore

$$\overline{\delta}_p = \frac{1}{C} \left( \Sigma_{D_p} + (N_p - C) \cdot \delta_n \right)$$  \hspace{1cm} (7)

with

$$\Sigma_{D_p} = \sum_{(i,j) \in \{ (i,j) | R_{sp}(i,j) \neq 0 \}} D_p(i,j).$$  \hspace{1cm} (8)

From a computational perspective, it is advantageous to avoid the calculation of $C$. By shifting all elements $D_p(i,j)$ by $-45°$, the neutral value becomes 0 and the additional $C-N_p$ sample points do not affect the sum. The new cost function can then be written as

$$\gamma_p = \frac{1}{C} \left( \Sigma_{D_p} - N_p \cdot \delta_n \right).$$  \hspace{1cm} (9)

Since $C$ does not depend on the pose, it can be ignored when minimizing $\gamma_p$ with respect to the pose. However, $C$ can change greatly for different objects and is therefore important for classification. The main difference between $\gamma_p$ and $\delta_p$ is that $\gamma_p$ considers not only pixels $(i,j)$ that satisfy $R_{sp}(i,j) \neq 0$ which is caused by the curvature of the object, but also sample points on planar surfaces. Since for planar surfaces no gradient direction is known in the model, these pixels cannot be used as individual indicators to reinforce or weaken the pose hypothesis. The number of pixels with unknown gradient direction, in contrast, can be interpreted as a confidence factor in the matching cost. For objects with large planar surfaces, the new cost function is particularly efficient.

D. Analysis of the Matching Cost

Ideally, the cost function has its minimum at the true pose. In this section we examine different cases to show that, with the assumptions considered, the true pose is determined by minimizing (9) with respect to the object pose.
Let $u$ index all pixels $(i,j)$ where $R_{Sp}(i,j) \neq 0$, such that $D_{p,u}$ is one of the $N_p$ pixels of $D_p(i,j)$ for which $(i,j) \in \{(i,j)|R_{Sp}(i,j) \neq 0\}$ applies. We can then rewrite (8) as

$$
\Sigma_{D,p} = \sum_{(i,j) \in \{(i,j)|R_{Sp}(i,j) \neq 0\}} N_p D_p(i,j)
$$

(10)

and (9) as

$$
\frac{1}{C} (\Sigma_{D, p} - N_p \cdot \delta_n) = \frac{1}{C} \sum_{u=1}^{N_p} (D_{p,u} - \delta_n)
$$

(11)

where $p$ indicates the pose hypothesis. $(D_{p,u} - \delta_n)$ ranges from $-45$ (when the sample point is in line with the corresponding pixel of the test image) to $+45$ (when the sample point differs the most from the test pixel).

We consider the pose $p_0$ to be the true pose. If the global minimum of $\gamma_p$ is at $p_0$, it follows that

$$
\gamma_{p_0} < \gamma_p
$$

$$
\gamma_{p_0} < \gamma_p
$$

$$
\Sigma_{u=1}^{N_p} (D_{p,u} - \delta_n) < \frac{1}{C} \sum_{u=1}^{N_p} (D_{p,u} - \delta_n)
$$

(12)

For the pose hypothesis $p_0$, which is the true pose, most sample points are similar to the test pixels, and $\gamma_{p_0} \rightarrow -N_p \cdot \delta_n$. Two cases are considered:

1) $N_p < N_{p_0}$:

Even if the sample points of the pose hypothesis $p$ are similar to the test pixels of the synthetic image, it applies $\gamma_p \rightarrow -N_p \cdot \delta_n > -N_{p_0} \cdot \delta_n$. So, the true pose hypothesis yields a lower cost.

2) $N_p \geq N_{p_0}$:

From the assumption of uniformly distributed gradient directions in the camera image, $(D_{p,u} - \delta_n)$ is also uniformly distributed between $-45$ and $+45$, if the model curvature is not aligned to gradient in the overlaid area of the camera image. For $N_p > N_{p_0}$, for “common” vehicles or objects (i.e., not artificial examples) there exists a very high likelihood that the additional sample points of the pose hypothesis $p$ are not aligned with overlaid test pixels in the camera image. As a consequence, the additional sample points will increase (or at least not decrease) $\gamma_p$. To achieve a $\gamma_p = \gamma_{p_0}$, the object must be equivalent in two poses. That implies that the object is either indistinguishable in its orientation according to its shape, or one face is the exact “negative mold” of another symmetric face, which does not correspond to a typical vehicle.

Therefore, the true pose is at the global minimum and is determined by minimizing the cost function proposed.

### E. Dealing with Shadows and Texture

The proposed cost function combines aspects from geometry–based and appearance–based fitting. With methods based on edges or the gradient magnitude, the region of convergence is very small and therefore a shadow model and illumination model are required [23]. In our method the region of convergence enlarges since we do not compare only edges but also gradient with low magnitude and, more importantly, we consider the gradient direction. For this reason our method is considerably robust to changing illumination conditions and shadows without the use of prior knowledge of light sources.

Although color and texture are characteristics that are well suited (if known) to match a model to an object, they are not essential for the pose fitting. In contrast to texture and color statistics, appearance change based on shading is more constant under different or changing illumination conditions. It is also more invariant if equal objects have different color, or if the texture changes with time, such as a dirty surface, which is common in an outdoor industrial environment.

Although we do not consider color and texture information of local surfaces, the proposed system presents good robustness to local texture changes. Figure 7 shows a sample image of the forklift-like vehicle used in our experiments. The vehicle has elements with different colors, stickers on the left and right sides, dirt, on-vehicle shadows, scratches, and plates, which we do not include in the 3D model.

To cope with these textures, we consider the gradient at the inside of a surface only at discretized lines: the transition of planar surfaces approximating the object curvature. If no shading appearance is modeled in a pixel of the synthetic image, this pixel is not compared to the real image. Even if the gradient direction of a texture fits locally to the gradient direction of shading appearance, this cluster is only a small part in the calculation of the arithmetic mean.

### V. Initialization and Identification

The target object and its position and orientation are not known initially. A specific search for the objects with their 3D models would be very time-consuming, if performed over the whole field of view. Instead, we use motion detection and perform the initial pose search in a reduced search space around the detected motion coordinate to accelerate the initialization and to reduce the probability of a wrong initialization. To
As a similarity measure for the pose search we can use $\delta_p$ (6) or $\gamma_p$ (9). At the initialization in particular, where the uncertainty about the object and the pose is high, a combination of both metrics seems reasonable. One way to combine the two measures is to consider only measurements in which both measures yield equal pose estimation. This simple approach seems reasonable in a defined search grid, since it can be considered as a two-of-two filter for valid pose estimations. This combined pose filter returns a very reliable pose but can take more frames. However, it can be more advantageous to consider more frames aiming at accurate pose initialization, than to initialize the tracking with a less accurate prior for the next frame. Experiments to show the initialization performance using $\delta_p$, $\gamma_p$, and a combination of both are shown in Section VI.

### A. Rough Initial Position Guess

As a first step in the initialization process, we use the motion template algorithm described in [33] and [34] to recognize motion in the 2D image. If an object enters the camera view and the detected motion cannot be assigned to a target which is already tracked by the system, we assume the presence of a new object. We use the flat world assumption and consider that the homography describing the transformation between the camera and the ground plane is known. Assuming that the lowermost point of the motion silhouette is either in contact with or close to the ground plane, the transformation from the camera to a 3D point returns a rough guess of the initial position. The pose of the object is searched for in a wide-range search space around this rough initial position guess. Figure 8 shows a sample image of a moving object. The green cross in the center of the vehicle indicates the motion center coordinate. The yellow diamond below the vehicle marks the rough initial position guess by shifting the green cross to the center of the bottom of the motion silhouette. Depending on the orientation of the object, the yellow point is outside the real object center. This offset is counteracted in the initial pose estimation, when the orientation of the pose hypothesis can be used to estimate the offset and reduce the error.

### B. Initial Pose Estimation and Filter Initialization

The initial search for the object orientation and a more accurate object position is performed with a simple grid search strategy around the rough initial position guess, acquired as described in Section V-A. By repeating this procedure with different 3D-models, the object is identified via best match as explained in Section V-C. The search grid in the first frame is coarse in order to reduce the computation time. It becomes more refined in the next frames when the search space can be reduced, due to a priori knowledge from the previous frame. In particular, in the orientation dimension the search space can be reduced severely, since in the first frame $360^\circ$ has to be covered and in the second frame the orientation range is mostly below $90^\circ$, depending on the object agility. After the initial pose estimation, a Kalman filter can be initialized and the filter covariance can be used to reduce or enlarge the search space.
without a priori information about the vehicle position and orientation. The pose search is triggered with a rough initial position guess as explained in Section V-A.

The metrics $\delta_p$ and $\gamma_p$ (defined in Section IV) are used for the object pose estimation. The search is triggered only if the object is completely visible. However, once the model is tracked (i.e., after the initial pose estimation), the algorithm can cope with objects that are partially outside of the field of view or partially occluded. To quantify the system performance, a high precision laser–based localization system [1] is used on the HMC and serves as a ground truth to evaluate the results. In addition, a manual (human-interpreted) pose estimation is also used in the evaluation. The manual estimation was performed for every frame, by placing the virtual object manually in the 3D scene to align it with the object in the camera image.

The test videos contain the three vehicles at different positions and orientations traversing the outdoor industrial environment. The objects are located at various distances from the camera, yielding significantly different resolutions. During some of the tests, weather conditions changed greatly between overcast and sunny, such that the video contains sharp shadows as well as soft or no shadows.

The overall results for experiments using $\delta_p$ and $\gamma_p$ as individual cost functions and in combination are given in Table II. The table shows the standard deviation for the two spatial coordinates $x$ and $y$ and for the orientation $\psi$. The number of frames for each test series is also listed in this table. The search grid for all initialization experiments was set to $\Delta x = \Delta y = 0.1\,\text{m}$ and $\Delta \psi = 5^\circ$.

Using $\delta_p$ and $\gamma_p$ stand-alone, the pose was estimated between frames 752 and 1427, depending on the object. In these frames, motion was detected by the motion history algorithm triggering the initial pose search, as described in Section V-A. The results compared against the ground truth are listed in Table II under “$\delta_p$” and “$\gamma_p$” for the three objects. Trajectory performance is also presented in Figures 9, 10, and 11. The plots in Figures 9(a), 10(a), and 11(a) show the estimated position compared to the ground truth. In those figures, the coordinates $(x, z)$ are rotated to $(\tilde{x}, \tilde{z})$ so that $\tilde{x}$ points north, which was originally parallel to the camera plane. Figures 9(b), 10(b), and 11(b) show the estimated orientation, also compared to the ground truth.

By combining the different similarity measures $\delta_p$ and $\gamma_p$ as explained in Section V-B, the pose estimation for the initial pose search can be improved although it can require more frames in the computation. As a consequence, the number of pose estimates is smaller than the number of frames where the object is entirely visible. Using both cost functions combined, the pose for each object was estimated in between 266 and 606 samples. The calculation time for both measures is negligibly longer than the calculation time for only one measure, as the computationally intensive calculations for (6) and (9) is shared for both measures. The combined initial pose search returns a reliable pose with an orientation error $< 7.5^\circ$ for each object. Using this accurate initial pose as a prior, the object can be tracked using only one similarity measure combined with a tracking filter. The average errors for each object are listed in Table II under “$\delta_p \& \gamma_p$”.

---

**Table II** Root mean square error ($\sigma$) for position and orientation estimation of the three objects using the two similarity measures $\delta_p$ and $\gamma_p$ and a combination of them.

<table>
<thead>
<tr>
<th>Object</th>
<th>Metric</th>
<th>$\sigma_x$ [m]</th>
<th>$\sigma_y$ [m]</th>
<th>$\sigma_\psi$ [deg]</th>
<th>Frames</th>
</tr>
</thead>
<tbody>
<tr>
<td>HMC</td>
<td>$\delta_p$</td>
<td>0.1025</td>
<td>0.2414</td>
<td>14.85</td>
<td>1147</td>
</tr>
<tr>
<td></td>
<td>$\gamma_p$</td>
<td>0.1132</td>
<td>0.2560</td>
<td>19.57</td>
<td>1147</td>
</tr>
<tr>
<td></td>
<td>$\delta_p &amp; \gamma_p$</td>
<td>0.0134</td>
<td>0.1158</td>
<td>6.66</td>
<td>266</td>
</tr>
<tr>
<td>Bobcat</td>
<td>$\delta_p$</td>
<td>0.2763</td>
<td>0.3871</td>
<td>39.92</td>
<td>1427</td>
</tr>
<tr>
<td></td>
<td>$\gamma_p$</td>
<td>0.1782</td>
<td>0.3282</td>
<td>9.10</td>
<td>1427</td>
</tr>
<tr>
<td></td>
<td>$\delta_p &amp; \gamma_p$</td>
<td>0.1197</td>
<td>0.1804</td>
<td>5.43</td>
<td>513</td>
</tr>
<tr>
<td>Gator</td>
<td>$\delta_p$</td>
<td>0.4352</td>
<td>0.1920</td>
<td>16.81</td>
<td>752</td>
</tr>
<tr>
<td></td>
<td>$\gamma_p$</td>
<td>0.1326</td>
<td>0.1606</td>
<td>19.38</td>
<td>752</td>
</tr>
<tr>
<td></td>
<td>$\delta_p &amp; \gamma_p$</td>
<td>0.0650</td>
<td>0.0889</td>
<td>7.40</td>
<td>606</td>
</tr>
</tbody>
</table>

---

Fig. 9. Pose estimation of the HMC using $\delta_p$ and $\gamma_p$ compared to the manual estimated pose data used as ground truth information. The orientation grid in the search space is $\Delta \psi = 5^\circ$. In the regions where no pose was estimated, the object is either not wide enough in the field of view, or the motion history algorithm could not return an initial position guess.
grid in the search space is \( \Delta \).

The orientation Fig. 10. Pose estimation of the Bobcat using \( \delta \) with the viewpoint.

where the number of sample points \( N \) where the number of sample points \( N \) cannot be estimated effectively, which leads to the poor fitting performance. Using \( \gamma_p \) instead of \( \delta_p \), the initial pose search improved from \( \sigma_\psi = 39.9^\circ \) to \( \sigma_\psi = 9.1^\circ \). The experiment showed that with the new metric \( \gamma_p \), the pose of the Bobcat can be successfully estimated at large distances (about 45 m), despite its small size.

2) Results for the Bobcat: For the Bobcat, a sample of the fitting results is shown in the third column of Figure 2. An example of partial occlusion is given in Figure 13(e), with successful localization. Also with the Bobcat, we experienced robustness to changing illumination conditions. The object pose could be estimated in the shadow (Figure 13(e)), and in the sun with sharp object shadows (Figure 12(b)). However, the pose estimation fails if a large part of the object silhouette is occluded or if the true pose is not included in the search space due to a poor initial position guess.

An example of a wrong pose initialization is given in Figure 14(c). Using \( \delta_p \), as the similarity measure, the front view is returned as the best fit in several cases, due to a very low \( N_p \). In particular, the side view of the object cannot be estimated effectively, which leads to the poor fitting performance. Using \( \gamma_p \) instead of \( \delta_p \), the initial pose search improved from \( \sigma_\psi = 39.9^\circ \) to \( \sigma_\psi = 9.1^\circ \). The experiment showed that with the new metric \( \gamma_p \), the pose of the Bobcat can be successfully estimated at large distances (about 45 m), despite its small size.

3) Results for the Gator: The second column of Figure 2 gives an example of the fitting results for the Gator. An example of a wrong initialization is given in Figure 14(b). Using \( \gamma_p \), the view at the back of the object is recognized as the
Fig. 12. Example for challenging illumination conditions. The algorithm can initialize the pose of the object despite sharp shadows on the ground and on the vehicle.

Fig. 13. Fitting results with partial occlusions (second row). The first row shows the scenes without the model drawn in white to make the occluding parts visible.

front view in many sample images. Due to the object symmetry both views contain similar elements, but the back view has a lower $N_p$. Therefore, the front view can be recognized reliably with the two similarity measures.

Fig. 14. Failure cases of the fitting results. In these images, for the Bobcat the fitting failed using $\delta_p$ (second row) but can be considered successful using $\gamma_p$ (first row). For the Gator the initialization failed using $\gamma_p$ while $\delta_p$ returned the right pose.

<table>
<thead>
<tr>
<th>True object</th>
<th>Metric</th>
<th>Identification rate (percent)</th>
<th>Samples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bobcat</td>
<td>$\delta_p$</td>
<td>88.25 0.08 11.66</td>
<td>1472</td>
</tr>
<tr>
<td></td>
<td>$\gamma_p$</td>
<td>94.00 1.97 4.03</td>
<td>1472</td>
</tr>
<tr>
<td>HMC</td>
<td>$\delta_p$</td>
<td>20.58 74.28 5.14</td>
<td>1147</td>
</tr>
<tr>
<td></td>
<td>$\gamma_p$</td>
<td>6.54 90.41 3.05</td>
<td>1147</td>
</tr>
<tr>
<td>Gator</td>
<td>$\delta_p$</td>
<td>11.57 0.27 88.16</td>
<td>752</td>
</tr>
<tr>
<td></td>
<td>$\gamma_p$</td>
<td>4.92 2.13 92.95</td>
<td>752</td>
</tr>
</tbody>
</table>

B. Classification

In this section, the classification performance of the two similarity measures $\delta_p$ and $\gamma_p$ is evaluated. The identification of different objects is efficient as long as the object images are scaled to similar sizes. The classification is performed via best match, which is done in two steps:

1) We perform the initialization for each model to obtain the best-fitting pose for the individual object hypotheses.

2) For the pose hypotheses of the different objects to be numerically comparable, we scale the camera image for each model to achieve equivalent silhouette areas. On the scaled images, we repeat the calculation of the similarity metric for each object in its estimated pose in step 1.

In the first step, the similarity metric for the pose fitting of the individual object was chosen based on the results in Section VI-A, as this experiment focuses on the classification and not on the pose estimation. Therefore, $\delta_p$ was used for the HMC and the Gator, whereas $\gamma_p$ was the metric employed for the Bobcat.

In this experiment we evaluate the two similarity measures with respect to the second step. $\delta_p$ and $\gamma_p$ were calculated for each object in its estimated pose in step 1. The object yielding the lowest value for the similarity metric is considered the best-fitting object. The experiments were carried out on the same video footage used in Section VI-A. The results are given in Table III. The results for the correct identification with the new metric $\gamma_p$ are marked in dark gray. It is observed that, with the new metric $\gamma_p$, all objects could be identified with a rate > 90%.

C. Field Test

In contrast to the results above, where the main goal is to explore the computer vision aspects of the method, this experiment investigates the performance of the system from a robotics perspective. We use a very high precision laser-based localization system [1] as ground truth to evaluate the technique. A Kalman filter is used to get higher localization precision and to reduce the search space for the matching algorithm. The initialization in this experiment is triggered even if the object is close to the image border and a pose hypothesis in the wide-range initial search space might be partially out of the image border, as illustrated in Figure 15.
Fig. 15. Vehicle partially out of the field of view. The algorithm is able to track the position and orientation, although a significant part of the object is not visible.

The results are shown in Figure 16 and Table IV. Figure 16 plots a top view of the test run, where the red line represents the laser–based ground truth. The green line indicates the estimated position from the matching algorithm. The camera is mounted at $\tilde{x} = 0$, $\tilde{z} = 0$ in the plot. In the region around $\tilde{x} = 30 \text{ m}$, $\tilde{z} = 35 \text{ m}$ the tracking becomes unreliable, as the vehicle drives out of the field of view. However, once motion is again detected a new initialization is performed in a larger search space and the object is tracked (even though it drives partially out of the image border in this example). Figure 16(b) illustrates the orientation matching. The average errors are listed in Table IV.

The results and average errors presented above include practical challenges, which are inherent to the set–up considered. As an example, our camera is mounted on a building wall, and it is often exposed to winds due to its mounting height, which can cause the camera to shake. In addition, the hysteresis of the actuators of the pan-tilt camera decreases the precision of the extrinsic camera calibration. Despite the fact that no vision-based image stabilization is used, the results show that the system can estimate the pose of the object with a low average error in a measurement range with a maximum distance of about 50 m.

Another important aspect of the system is the accuracy of the 3D model compared to the real object. Generally, small variations that do not offset the whole model do not significantly affect the performance. In Figure 17(a), for example, the object contains an additional passenger, which is not a part of the model. In Figure 17(b), the truck mast is adjusted differently in the model compared to the real object. A visual evaluation indicates that, for the vehicles and scenario tested, the algorithm can cope with these small amounts of distortion with negligible effect on the localization precision.

![Fig. 15. Vehicle partially out of the field of view. The algorithm is able to track the position and orientation, although a significant part of the object is not visible.](image1)

![Fig. 16. Pose estimation of the HMC using $\delta_\omega$ with a Kalman filter compared to the laser–based ground truth information. The orientation grid in the search space is $\Delta \psi = 5^\circ$.](image2)

![Fig. 17. Experiments with model inaccuracy: Although the passenger of the Gator in (a) is not included in the model and the mast of the HMC in (b) is adjusted differently, the errors are generally negligible.](image3)
We compare our method with the works by Nagel et al. [22], [23], performing experiments on the same video footage. In contrast to our method, however, their algorithm relies on a shadow model. In this section we present a case where, because of the illumination condition assumptions, they experienced wrong pose estimation, whereas our method presented satisfactory results. Figure 18 shows one of the results presented in [22]. In Figure 18(a), a wire-frame model of a sedan employed in [22] is matched to a “taxi”. The wire-frame model of the object represents the most salient object edges, which are used for the fitting process. This example illustrates the poor matching due to edges of the object’s shadows, which did not agree with their shadow model. Figure 18(b) shows our fitting results, indicating that our algorithm can cope with the sharp shadows, although no shadow model is used. For this experiment, we used a CAD model of a Mercedes 190E, similar to the vehicle in the footage. From the CAD model, our method extracts not only the most salient edges, but also possible shading appearance. Edge-based methods only consider gradient or shading with a high gradient magnitude. Especially in an outdoor scenario where sharp shadows can occur, model edges can be aligned to the sharp shadow edges. In contrast, because the proposed method considers shading independently from its magnitude, it is more robust to shadows on the ground, whether soft or sharp. In addition, the method relies not only on edges but also on shading caused by curved surfaces. In the example of the taxi footage, the side doors and the trunk are often in the shadow side of the vehicle, which leads to visible shading edges that are not always aligned with salient object edges. Since our method considers salient object edges with the same importance as possible shading, the fitting with our method leads to better performance. As the authors do not have quantitative ground-truth, it is not possible to compare the errors measures.

E. Preliminary Experiments with Night Vision

In this experiment, a thermal camera (Thermoteknix Miricle KS-307K) with resolution $640 \times 480$ pixels is mounted at approximately $4.7 \text{ m}$ above the ground. With a tilt angle of $16.3^\circ$, the field of view of the thermal camera ranges from approximately $8 \text{ m}$ to $45 \text{ m}$ on the ground. The camera is fairly sensitive and can detect emitted and reflected infrared radiation. The gray level of a (comparatively) cold surface in the thermal image is mainly caused by thermal reflection. The reflection of the vehicle heat appears as a white thermal “shadow” on the ground below the HMC. Also, the cabin window is a strong reflecting surface, and changes the gray level for some incident angles. Apart from these effects, the reflection of the surrounding infrared radiation is favorable for the algorithm, as it appears as shading gradient on the vehicle. Figure 19 also shows a heat gradient on planar surfaces such as the engine hood, which consists of brighter patches where the engine is hotter.

Ideally, the method should deal with heat gradient on planar surfaces since it is similar to local texture changes (as discussed in Section IV-E). However, homogeneously hot surfaces, which are predominantly self–emitting infrared sources, can reduce the localization precision. This type of hot surface can be approximated as a Lambertian emitter, emitting light in all directions equally, independent of the viewpoint of the observer. With these assumptions, a curved hot surface has no shading appearance. Notice that when a surface is reflecting, the incident light and thus the reflecting intensity varies due to Lamberts’ cosine law. If the surface is emitting, the intensity seen by the thermal camera is constant even if the surface rotates. The same principle applies in the case of curvature. Therefore, the difference (apart from possible saturations) between the thermal and visible cases is that, in the latter the surface is only reflecting, while in the thermal case the surface is also emitting. In our tests, these concepts are also illustrated by the view of the exhaust tube and of the tread of the tire (Figure 20). Such surfaces contain no gradient in the thermal image, although they are curved.

The experiments have shown that the pose estimation can occasionally fail if large parts of the object have homogeneous temperature without visible shading. The camera employed is not sensitive enough to allow for visible curvature and edges in the thermal image, and this can lead to local saturation. Nonetheless, the results are generally satisfactorily, as illustrated in Figure 20, although the use of a more sensitive camera could potentially increase the localization performance.

VII. Conclusions

We have presented a method to estimate the pose of ground vehicles, using an off-board camera and a 3D model of the vehicle. The gradient direction serves as a similarity measure to align the 3D model with the object in the camera image. The method can determine the object pose even at large distances without the need for a previous training stage or an illumination model. It is also significantly robust to changing illumination conditions. A novel method was derived that allows the fitting of objects that have a different number of sample points for different points of view. A combination of this new matching cost with the matching cost developed in
an earlier work [3] yields a very reliable first pose estimation, which is suitable to initialize the tracking. In addition, a method for vehicle identification was proposed, achieving a detection and classification rate of > 90% for the different vehicles tested. The high accuracy of the proposed method allows the recovery of an appearance model online at the initial pose estimation. The results suggest that color and texture are not essential for the determination of the object pose in the 3D space. Moreover, the object pose can be reliably estimated from 2D vision if a priori knowledge about the object shape is used.

With the obtained performance, the method finds applicability in industrial environments; for example, where vehicle localization is becoming increasingly popular. In our particular scenario, the system can work as a redundant or alternative solution to on-board localization systems [1], [2] that have been recently implemented in industry. Although possible with the accuracy obtained, vehicle control is not implemented in this version of the system as higher precision would be desirable, which is potentially achievable with the planned future work.

Directions for further research have been identified, including the combination of $\delta_p$ and $\gamma_p$ in a Bayesian framework and the extension to multiple cameras. A hybrid tracking using an appearance model of the object could potentially improve the localization performance when initialized with a reliable prior, determined by the system proposed here. Such a system would combine the fast convergence of appearance-based methods with the robustness to changing illumination of the proposed method.

In summary, the system has been implemented and tested in a real industrial environment, where vehicle localization is considered a key component of situation awareness, with the aim of safety and efficient operations. The proposed solution is relatively affordable and reliable, providing satisfactory results for the scenario considered.

ACKNOWLEDGMENT

This work was supported by the CSIRO Computational Informatics and the University of Ulm. The authors would like to thank the Institut fuer Algorithmen und Kognitive Systeme at University of Karlsruhe for providing the image sequences used in their experiments.

REFERENCES
