sciendo

Int. J. Appl. Math. Comput. Sci., 2022, Vol. 32, No. 4, 583–599
DOI: 10.34768/amcs-2022-0041

amcs

VISION–BASED POSITIONING OF ELECTRIC BUSES FOR ASSISTED DOCKING TO CHARGING STATIONS

TOMASZ NOWAK [a,*], MICHAŁ R. NOWICKI [a], PIOTR SKRZYPCZYŃSKI [a]

[a]Institute of Robotics and Machine Intelligence
Poznan University of Technology
Piotrowo 3A, 60-965 Poznan, Poland
e-mail: tomasz.nowak@doctorate.put.poznan.pl,
{michal.nowicki,piotr.skrzypczynski}@put.poznan.pl

We present a novel approach to vision-based localization of electric city buses for assisted docking to a charging station. The method assumes that the charging station is a known object, and employs a monocular camera system for positioning upon carefully selected point features detected on the charging station. While the pose is estimated using a geometric method and taking advantage of the known structure of the feature points, the detection of keypoints themselves and the initial recognition of the charging station are accomplished using neural network models. We propose two novel neural network architectures for the estimation of keypoints. Extensive experiments presented in the paper made it possible to select the MRHKN architecture as the one that outperforms state-of-the-art keypoint detectors in the task considered, and offers the best performance with respect to the estimated translation and rotation of the bus with a low-cost hardware setup and minimal passive markers on the charging station.

Keywords: AI transport, localization, monocular vision, deep learning, keypoints, advanced driver assistance system.

1. Introduction

The role of high-capacity electric buses in public transportation increases steadily. Many of these buses use electric charging stations mounted on pylons to re-charge while en route. Approaching precisely an electric charging station while driving a long, articulated vehicle is a difficult task that requires considerable experience. This creates a demand for an advanced driver assistance system (ADAS) (Kukkala *et al*., 2018) that helps less-skilled drivers to dock their buses to the charging stations. This system provides the driver with clear cues on how to operate the steering wheel to perform the desired maneuver with respect to a charging station, while the maneuver itself is executed under full control of the human driver, who takes responsibility for its safety.

The docking task is formulated as the accurate positioning of a selected guidance point of the bus with respect to the charging station's head (Fig. 1). As the geometric relation between the pantograph tip and the guidance point is known, the problem is reduced to compute a feasible trajectory between the initial pose of the bus in the charging station's coordinates, and the desired location of the pantograph tip.

The required positioning accuracy depends on the pantograph mechanical interface, which is responsible for a reliable electrical contact between the tip and the charger's head. In the commonly used pantographs, the tolerance is 0.35 m in lateral positioning error (Schunk Carbon Technology, 2021). The path planning and control procedure for articulated vehicles (Michałek and Kiełczewski, 2015), which guarantees safe and smooth motion of the bus is beyond the scope of this paper that focuses solely on the perception and localization aspects of the problem. A reader interested in the motion planning aspects should consult the work of Gawron *et al*. (2019) for details.

The guidance for docking is challenging to the perception system, as the procedure starts at a distance of 30 to 40 m from the charging station, which has to be detected, recognized, and localized automatically, as for safety reasons the human driver should focus on visually monitoring the road and executing the

*Corresponding author

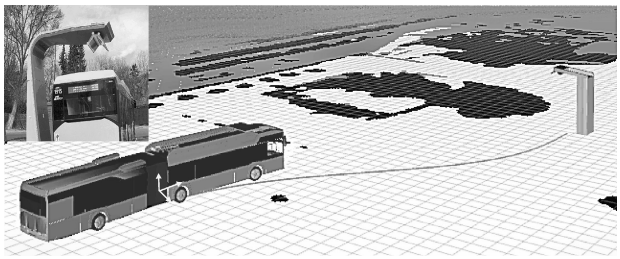

Fig. 1. Illustration of the docking maneuver with an articulated bus. The visualized coordinate system's origin is coincident with the guidance point, while the curved line indicates the planned trajectory. The inset image shows a charging station with example salient features, as seen by the localization system's camera.

motion. Although the bus has a receiver of the Global Positioning System (GPS) as standard equipment, the GPS suffers from outages and signal interferences in urban environments (Youjing and Shuzhi, 2003) and has localization accuracy of several meters, which is unacceptable for the docking task. In ideal conditions, centimeter-level localization accuracy can be obtained with differential GPS (DGPS). The DGPS improves the measurements with additional real-time correction signals from a nearby ground reference station. The lack of line-of-sight visibility of the GPS satellites due to tall buildings, lack of required corrections for a few seconds, or getting delayed corrections result in losing the most accurate mode of operation, and obtaining meter-level localization accuracy, which is insufficient for our application. Also the required permanent network connection without delays (e.g., using LTE), and the need to access a ground reference station in the area of operation, combined with the required clear view of the sky, make the DGPS-based localization procedure rather impractical for deployment by the bus operators. Therefore, although DGPS proved sufficient to provide localization during the ADAS tests (Michałek *et al*., 2021), we are looking for another solution that is easier to use. On the other hand, in controlled testing environments, the DGPS provides accurate, ground truth reference localization to evaluate the proposed solutions, and thus was used as such in our experiments.

The sensory system used for localization of a bus with respect to a charging station has to be affordable for series production, easy to install and maintain. It should also be possible to install it on electric buses already in use. Considering these requirements, we decided to apply passive vision as the sensing modality. Passive cameras are inexpensive, compact, and energy-efficient, while vision makes it possible to detect salient photometric features that can be used for localization (Vivacqua *et al*., 2017). We employ the monocular camera configuration, which is less expensive than a calibrated stereo camera rig and can be integrated more easily into buses. We cannot use wide-baseline stereo (Olson and Abi-Rached, 2010) because of the required flexibility in integration with various bus models, hence we can expect that scene depth estimates from a small-baseline stereo setup will not be significantly better than those from a single camera, particularly for larger distances (Hartley and Zisserman, 2004), while the localization at larger distances from the charging station is crucial for planning and performing maneuvers with a long vehicle (Michałek and Kiełczewski, 2015).

The need for relatively accurate camera pose estimates at distances up to 30 m combined with the very limited acceptance of any additional elements (fiducials/markers) attached to the charging station structure ruled out also the classic passive markers. Such markers, e.g., Apriltags (Wang and Olson, 2016), are often used to localize a camera with respect to a defined object. We experimented with Apriltags of the size 10×10 cm, which were accepted as elements than can be attached to the charger station's supporting structure. Unfortunately, the range of detecting an Apriltag of this size by our high-resolution camera was about 24 m, which is a too short distance for our application. Larger size Apriltags or other similar markers were deemed unacceptable by our industrial partner.

Considering all the specific requirements and limitations of the target applications, we present in this paper a new, integrated approach to object detection and localization using a monocular camera that employs deep learning for both object detection/classification, and extraction of feature keypoints. Our approach uses a geometric model of the charging station, which can be easily obtained from the bus operator, but otherwise is entirely data-driven, thus being able to learn any appearance of the charging station and layout of the keypoints, which makes the proposed solution flexible with respect to real-world applications.

Some aspects of the system supporting docking to electric chargers have been already presented in conference papers: our approach to the detection of charging stations was introduced by Nowak *et al*. (2019), focusing on the explainable object detector, while Nowak *et al*. (2020) and Michałek *et al*. (2021) investigated simple extensions of the neural network from the work of Nowak *et al*. (2019) that allowed our system to estimate the metric distance to the charging station's head. These articles make use only of the object detector and keypoint extractor based directly on the Faster R-CNN, while in this article we present two new neural architectures for keypoint detection and compare them with state-of-the-art neural models, achieving better accuracy of pose estimation compared with the previously published version. The novel contributions of this journal

article are as follows:

- The Regression Keypoint Network, which is an adaptation of the Faster R-CNN neural network architecture to the task of keypoint detection and localization. Experiments with this approach demonstrate that the Faster R-CNN architecture can be easily adopted to new tasks by adding processing heads, but also show that the multiple fixed-size bounding boxes limit the accuracy of the R-CNN architecture in the extraction of keypoints.
- The Max Resolution Heatmap Keypoint Network, which is an entirely new neural network architecture for detection and localization of keypoints. Building on our experience with the R-CNN model, this architecture ensures keeping the maximum resolution when determining the location of keypoints and provides a computation and memory-efficient solution to the problem considered.
- The integrated neural network architecture that combines robust detection of the charging stations with accurate localization of the keypoints within a region of interest.
- Thorough experimental evaluation of different approaches to the extraction of the keypoints on our task-specific dataset.
- Extensive evaluation of the entire localization system with real city buses that determines the minimum requirements for the used camera to improve the cost efficiency. This evaluation results in design recommendations for similar localization systems.

The remainder of this paper is organized as follows: Related work is reviewed in Section 2. Then the system hardware, software structure, and the neural network architectures are described in Section 3. Experiments and results are presented in Section 4, followed by conclusions in Section 5. Finally, Appendix presents an in-depth analysis of the influence of the system parameters on the observed performance and outlines the design recommendations.

2. Related work

The discussed task of localization with respect to a charging station includes detection of the station from a long distance and computation of the vehicle pose with respect to the coordinate system of this station.

The first subtask is an instantiation of the object detection and classification problem, which is widely investigated in many application contexts. Among methods that are relevant to the operation of self-driving vehicles or ADAS, deep neural network (DNN) architectures achieve the best results, making it possible to detect in real-time such objects as road signs (Fan and Zhang, 2015), traffic lights (Kim *et al*., 2018), and advertising billboards (Rahmat *et al*., 2019).

On the algorithmic side, the most popular detectors of specified-class objects are based either on the two-step approach using region proposals or on the single-shot approach. The former one was introduced by Girshick *et al*. (2014), and further improved to the Faster R-CNN (Ren *et al*., 2015) variant. This approach is accurate but relatively slow due to the many proposals that are generated. On the other hand, the single-shot algorithms, represented by the YOLO (You Only Look Once) family (Redmon *et al*., 2016), estimate the bounding boxes and class labels in parallel, which results in a considerable speed improvement, but at the cost of decreased accuracy. In this paper, the detection of the charging stations is based on the Faster R-CNN (Ren *et al*., 2015) network architecture, as it yields accurate results and is compatible with our initial approach to the detection of keypoints, which is also based on the R-CNN backbone.

We apply the explanation-guided training procedure for the object detector proposed in our previous work (Nowak *et al*., 2019). Unlike road signs or traffic lights, the charging stations for electric buses are not a common feature in the urban landscape. Hence, we have considered (Nowak *et al*., 2019) learning from a limited number of annotated images with pre-learning on large datasets (e.g., KITTI (Geiger *et al*., 2012)), and we have shown how to guide the inclusion of counterexamples for data augmentation (Dreossi *et al*., 2018) using explanations obtained as attention maps (visual interpretations) produced as a side-output of the R-CNN detection network.

The subtask of pose estimation with respect to a known object can be solved with a variety of methods. Docking to a charging station/device was investigated in mobile robotics using vision, e.g., for security robots (Luo *et al*., 2005). Recent works in this area apply deep learning for object detection and navigation (Taghibakhshi *et al*., 2021). However, there are relatively few works concerning docking larger-scale electric vehicles for charging (Clarembaux *et al*., 2016). Automatized docking systems proposed for self-driving cars usually rely on some active devices that plug into the car's charging port (Petrov *et al*., 2012; Miseikis *et al*., 2017). Perception is accomplished using regular (Miseikis *et al*., 2017) or infra-red (Petrov *et al*., 2012) cameras, while these systems do not need to precisely localize the entire vehicle, focusing on the guidance of the active device. In contrast, the autonomous docking system for recharging of electric vehicles described by Pérez *et al*. (2013) uses a camera localizing active infra-red beacons installed in a recharging booth, which allows the vehicle to dock precisely to the charging station. One should notice that

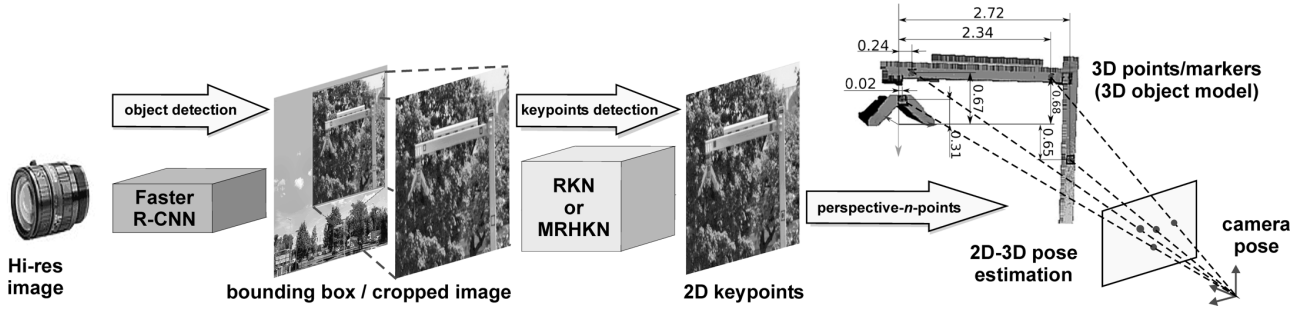

Fig. 2. Block diagram of the image processing and pose estimation pipeline used for positioning of electric buses with respect to a charging station. The CAD model is a visualization of the 3D mesh obtained from SURPHASER 100HSX with annotated dimensions between markers.

in contrast to these systems we do not use any mechanical devices other than those already present in a human-driven city bus (i.e., a pantograph), and our modifications to the charging station itself are minimal (few passive markers), or not necessary at all, if we use the naturally existing salient points. This minimalistic approach is of great practical importance, as due to various reasons many bus operators do not permit the installation of any active elements on the existing charging stations, and they even do not allow to alter the appearance of these structures.

It was demonstrated by Royer *et al*. (2005), that even simple monocular visual odometry can achieve localization accuracy comparable to the pose estimates obtained from DGPS. A more robust solution is simultaneous localization and mapping (SLAM), which localizes a robot/vehicle with respect to a map updated on-line (Skrzypczyński, 2009). State-of-the-art SLAM algorithms can accurately localize a vehicle with monocular vision, as shown, e.g., by ORB-SLAM2 (Mur-Artal and Tardós, 2017) on the KITTI dataset (Geiger *et al*., 2012). However, visual odometry and SLAM require a significant number of salient features acquired over several image frames to work reliably (Lim and Bräunl, 2020) and need to start from a known position in the reference frame, while our reference can be determined only by observing the charging station. The task of docking to a specified object with visual localization is also similar to the visual servoing problem, but typical visual servoing methods (Marchand *et al*., 2005) require known objects that appear big enough in the images. Unfortunately, the charging station detected from a distance of more than 30 m appears very small, and visual servoing methods do not work well in such a scenario.

Therefore, we propose to apply a method that directly computes the camera pose with respect to the target object coordinate frame using a small number of features and a known model of this object. This problem is well-known in computer vision and usually solved using the perspective-n-point pose estimation algorithm (Lepetit *et al*., 2009) or applying the bundle adjustment technique (Triggs *et al*., 2000). In the last decade, end-to-end neural network models were introduced that directly regress the camera pose parameters from an input image (Kendall *et al*., 2015; Xiang *et al*., 2018). However, the accuracy of these methods is still inferior to the geometric solution with a perspective-n-point algorithm, if we have accurate keypoint detections. Therefore, we use the geometric approach to pose computation but prefer a learning-based approach to keypoint detection, which avoids setting parameters for classic feature detectors. A similar idea of detecting point features using a multi-tasking variant of the Faster R-CNN network was presented by Zhang *et al*. (2020) but in the context of enhancing contours detection, rather than detecting specific keypoints.

A number of neural network architectures for keypoint detection have been introduced in the context of human pose estimation and tracking (Toshpulatov *et al*., 2022). Although this is a very different area of application compared with our bus localization problem, some of the models proposed for human pose estimation can be considered applicable, after training on an application-specific sequence of labeled images. The keypoint detection model introduced by Papandreou *et al*. (2017) is based on Faster R-CNN, similarly to the Regression Keypoint Network architectures considered in this paper. Papandreou's network architecture bears also some similarities to our MRHKN network, although the main conceptual difference lies in the method used to determine the final keypoint coordinates. Namely, in Papandreou's architecture, the size of the image and heatmap is relatively small, and high estimation accuracy is achieved by considering offsets, while MRHKN generates a single, bigger heatmap for each point, applies the DBSCAN algorithm to cluster the results, and then ensures accuracy in determining the position of the keypoint as the center of mass of the cluster that has the highest activation. As we demonstrate in the experimental part (Section 4), this approach allows our model to process

images of much higher resolution, promoting higher accuracy.

Currently, a leading solution for keypoint detection in human pose tracking is the HRNet (High Resolution Network (cf. Wang *et al*., 2021)). Similarly to our Max Resolution Heatmap Keypoint Network, this model has been designed to keep the high resolution of the feature maps through the entire processing pipeline; however, it uses a dedicated backbone network, while our architecture relies on the ResNet101 (He *et al*., 2016), which is also used in the Faster R-CNN, and we apply to detect the entire charging station from long distance. A different approach is taken in the recent paper by Liu *et al*. (2020), which demonstrated the use of novel self-calibrated convolutions that expand fields-of-view of each convolutional layer to detect keypoints, also with the application to human pose estimation.

3. Proposed processing system

Docking to a charging station's head requires localizing the bus with respect to the station's coordinate system in a wide range of distances, starting from almost 40 meters. Providing accurate results at the beginning of the maneuver is challenging because the observed objects are very small, whereas at the end of the maneuver, the station might not fit into the image. Moreover, the lateral distance offset between the pantograph's tip and the charger's head, and the angular offset between the longitudinal axis of the charger's head, and the direction of approach shall be small enough at the end of the maneuver to prevent any mechanical damage to either the head or the pantograph. However, we can assume some tolerance in both the translational and rotational components of the estimated pose, because the mechanical design of the charger's head tolerates lateral offsets and allows safe plugging with small angular errors (Schunk Carbon Technology, 2021).

The pose estimated by the vision system is not used directly in path planning and steering of the ADAS, but is integrated with an odometric pose estimate computed upon a mathematical model of the vehicle, and measurements from the proprioceptive sensors of the bus. In the actual application, this approach compensates for the occasional lack of the pose estimate caused by occlusions or image artifacts (e.g., due to direct sunlight), and makes it possible to provide the pose estimates with a higher frequency to the control system. However, the bus odometry and the integration are part of the ADAS control system and thus are beyond the scope of this paper. We focus on the performance of vision-based localization and do not use the odometric data to improve the pose estimates in the presented experiments.

Because of the wide range of observation distances, we use a high-resolution FLIR Blackfly S camera (5472×3648 pixels). Such a high resolution yields the best performance but can be then scaled down to find a trade-off between the performance and the cost, considering real-world scenarios (see Appendix). Apart from the resolution, the camera field of view (FoV) plays an important role. Choosing a small FoV makes the object appear bigger on the image plane, but reduces the set of possible maneuvers that contain the charger within the part of the scene observed by the camera. We also have to consider that the charging stations are sometimes placed within the bus bays that require a rather sharp steering while docking. Hence, we assume that the FoV of 60° performs well in all realistic scenarios while providing the necessary resolution for further processing. Considering these design choices and the technical requirements imposed by the bus manufacturer, we assume that sufficient characteristics of the vision-based localization system performance are as follows: translational error below 0.35 m, and rotational error below 1°.

Fig. 3. Visualization of a two-step processing pipeline. Using the full input frame at a reduced size, the object detector predicts the charger station position on the image (small rectangle). Then from the image at full resolution, the ROI containing the charging station is cropped for further processing (big frame).

3.1. Processing pipeline and object detection. We use high-resolution images which cannot be processed in real-time using standard hardware and neural network architectures. As from long distances the charging station's mast occupies only a small fraction of the whole image, a two-stage processing pipeline was implemented that firstly detects the object of interest, and then determines the keypoints belonging to that object (Fig. 2).

In the first step, the processed frame is resized to 960×960 pixels and passed to the Faster R-CNN network to detect the charging station. Images at such resolution are sufficient to properly detect charging stations during scenarios considered. Having coordinates of a bounding box from the object detector network, the region of

interest (ROI) is cropped from the original image (Fig. 3). After that, the ROI is resized to a 960×960 pixels image, which is processed by another neural network to detect the positions of the keypoints on the object. This procedure allows us to use a common object detector architecture on images from our high-resolution camera and exploit the ROI in the maximum possible resolution to assert the best keypoint estimation accuracy.

Having the coordinates of keypoints along with the 3D positions of those points in the real scene and camera intrinsic parameters, the perspective-n-point algorithm (Lepetit *et al.*, 2009) can be used to determine the position of the camera with respect to the charging station.

We investigated two architectures to estimate the position of keypoints. The first one is based on the Faster R-CNN network, and the second one directly predicts the probability of keypoint location in the form of a heatmap. Both variants of the keypoint estimation neural network are trained on our datasets with images acquired from different viewpoints and manually labeled keypoints (cf. Section 4.1). During the training procedure, in much the some way as during inference, the cropped image fragments with the charging station are resized to 960×960 pixels to fit the network architecture regardless of the observation distance.

3.2. Keypoints detection: Regression keypoint network. The Faster R-CNN network is a well-known object detector, that was an inspiration to create the regression keypoint network (RKN). The input image is processed by a backbone network, which extracts feature maps from this image. In our case (Fig. 4), the ResNet101 is used as the backbone, which creates 1024 feature maps, that are then passed to the region proposal network (RPN). The RPN produces a set of regions, which most likely contain an object of the sought class. Then, from the feature maps generated by the backbone network, the appropriate regions are cropped and unified by the ROIPooling layer to be used parallelly by the predictor's heads. The parameters of the RPN network have been modified to preserve the highest possible resolution of processed crops to avoid information loss about exact keypoint locations. Finally, the regions returned by the RPN have a unified size of 32×32 pixels. This is the maximum size that could be applied to fit into the used GPU memory. We assume that this is the bottleneck that limits the keypoint localization accuracy of this approach.

The first standard predictor of RKN is a regressor, which improves the bounding box position. The second one predicts the class association and confidence of each proposal. The third predictor is a new component in our architecture, which is responsible for the estimation of the keypoint positions on images and does not exist in the original Faster R-CNN architecture. The keypoint prediction head consists of a stack of 8 convolution layers

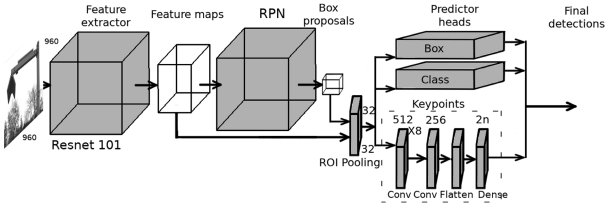

Fig. 4. Block diagram of the regression keypoint network (RKN) architecture for keypoint detection.

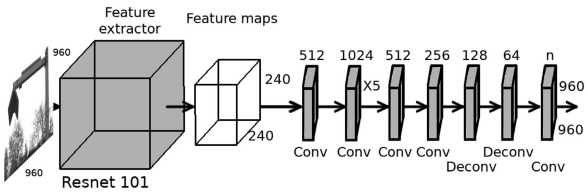

Fig. 5. Block diagram of the max resolution heatmap keypoint network (MRHKN) architecture for keypoint detection.

with 512 filters. After that, there is a convolution layer with 256 filters, with each filter size of (3, 3). The output of the last convolutional layer is flattened to form a vector and then passed to a fully connected layer with the number of outputs equal to twice the number of the defined keypoints. These outputs correspond to the directly estimated x and y coordinates of each keypoint.

3.3. Keypoint detection: Max resolution heatmap keypoint network. The idea of the second approach to the localization of keypoints arose from the observation that the standard heads of the RKN network are not necessary, as they only confirm the class and location of the object that is already detected by the Faster R-CNN part of the pipeline. On the other hand, the multiple bounding boxes of relatively small sizes in the RKN prevent the additional head from detecting the keypoints with a better resolution. Therefore, we propose a new architecture called the max resolution heatmap keypoint network (MRHKN) that is designed to keep as much as possible of the input image resolution (Fig. 5). Compared with the RKN, this architecture does not have the RPN network and any other heads except the keypoint head. The new design allows us to increase the depth and width of the keypoint head.

As is the case in the RKN approach, the ResNet101 network is used as the backbone. It produces feature maps downsampled four times, which for a 960×960 pixels input image results in a 240×240 pixels resolution of the features maps. Then, those feature maps are processed by a stack of 8 convolution layers without any further resolution loss. To make the output heatmap size the same as the size of an input image, two deconvolution layers are applied. The last layer is a convolutional one, with a filter

size of (1, 1), to generate a heatmap for each of the n keypoints.

However, the heatmap indicates only the probability of the keypoints positions, and post-processing is required to get actual keypoint coordinates. Figure 6 shows the postprocessing steps for an example keypoint marked by the circle (Fig. 6(a)). The output from the network (Fig. 6(b)) can contain some false activations, as pointed by the arrows, which should be filtered out. A closeup of the true positive activation is shown in Fig. 6(c), while the Figs. 6(d) and (e) depict the false positives. To remove false activations, thresholding is applied to the heatmap, to get a binary image (Fig. 6(f)). Then, the keypoint proposals are determined on that image by applying the DBSCAN clustering algorithm (Schubert *et al*., 2017). To find the proper cluster, a confidence score is calculated for each proposal. The confidence score S_i of the i-th cluster K_i is defined as the sum of intensities $I(\mathbf{x})$ of all pixel locations $\mathbf{x} = [u, v]$ on the raw heatmap belonging to the cluster K_i:

$$S_i = \sum_{\mathbf{x} \in K_i} I(\mathbf{x}). \tag{1}$$

The final keypoint location $\mathbf{c}_i$ is computed as the center of mass of the cluster with the highest confidence score (Fig. 6(g)):

$$\mathbf{c}_i = \frac{1}{S_i} \sum_{\mathbf{x} \in K_i} \mathbf{x} \cdot I(\mathbf{x}). \tag{2}$$

3.4. State-of-the-art approaches to keypoint detection. The neural network architectures proposed in this paper are dedicated to the bus charger localization problem, which to the best of our knowledge was not considered before using passive vision. Similar setups for keypoint detection are explored neither by computer vision nor the robotics community. However, the estimation of 2D keypoints from images is leveraged for localization in the human body pose estimation applications (Toshpulatov *et al*., 2022). As this area of applications has large commercial potential, a number of learning-based keypoint detectors have been proposed. Some of these neural network models have been gathered within the MMPose framework (MMPose, 2020), which allows to train and inference in a uniform setup of neural models in different variants, changing the metaparameters and even using different backbones. We use this framework for comparison, selecting the HRNet (Wang *et al*., 2021) and SCNet (Liu *et al*., 2020) models, which are recent and widely used architectures for keypoint detection achieving top scores in the COCO 2017 Keypoint Detection Task.

Moreover, we implemented for comparison a ResNet101 based keypoint detector using the MMPose framework. This architecture consists of the ResNet101 backbone and a 4-layer detection head, the same as in the HRNet detector. This detector allows us to test how much the results depend on the head architecture, and how much on the backbone, as our neural network architectures for keypoint detection also use ResNet101 as a backbone. To further demonstrate the advantages of the MRHKN architecture, we used also the model of Papandreou *et al*. (2017) for comparison.

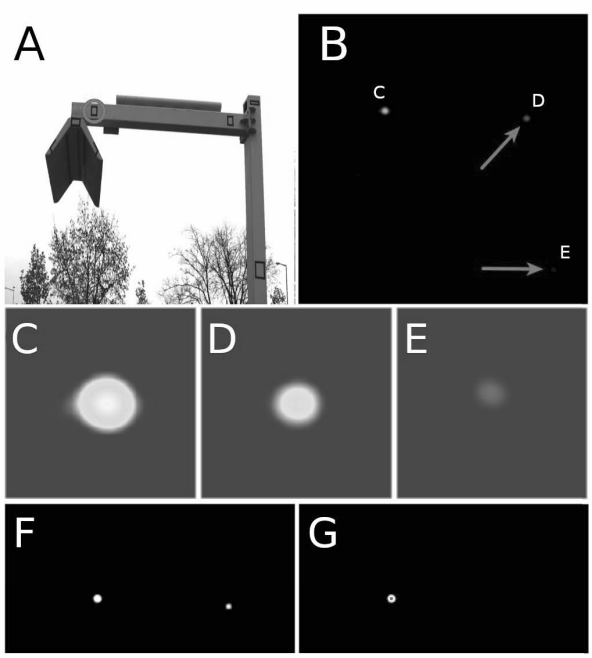

Fig. 6. MRHKN postprocessing: input image (a), network output (b), closeup of the network output near a marker to be found (c), false positive markers (d) and (e), thresholding (f), DBSCAN clustering and center of mass as actual keypoint coordinates (g).

In order to ensure a fair comparison, we fed to all the investigated detectors network fragments of the original input images containing only the bounding box with the charger, and the size of the input image was set to 960×960 pixels, the same as in our proposed networks, with an exception to the model from (Papandreou *et al*., 2017), which turned out to be unable to process images of that size on the GPUs used in our experiments.

3.5. Pose estimation. Camera pose estimation in our solution is based on keypoints detected on the image using a deep learning framework and the knowledge of the 3D model of the charger. As a result, the pose of the camera can be estimated using an algorithm that solves the perspective-n-point problem. Our initial tests revealed that best results were obtained with an iterative algorithm (Lu, 2018), which minimizes the reprojection error defined as the sum of squared distances between the point localization on the image, and the object model

points projected on this image:

$$\mathbf{T}^* = \arg\min_{\mathbf{T}} \sum_{i=1}^{n} \left(\mathbf{c}_i, \pi\left(\mathbf{T}\mathbf{w}_i\right)\right)^T \left(\mathbf{c}_i, \pi\left(\mathbf{T}\mathbf{w}_i\right)\right), \quad (3)$$

where n is the number of points, $\mathbf{c}_i$ stands for the image coordinates of the i-th point of the charger detected on the image by the deep learning system, $\pi(\cdot)$ is a camera projection function, $\mathbf{T}$ is a rigid transformation matrix (rotation and translation), and $\mathbf{w}_i$ denotes the coordinates of the i-th 3D object's point based on the 3D model of the charger.

A drawback of this optimization-based approach is that a reasonable initial guess of the camera pose is needed. This is not a problem when our system is used to localize the bus along the planned path while approaching the station, as long as we can use the previous pose estimates and the odometry to fill in the gaps between the vision-based measurements. However, we lack the first initial guess, and therefore a separate initialization procedure was proposed. To initialize, we run the perspective-n-point algorithm from several different initial guesses within the working area of the maneuver and accept the pose estimate that has the lowest reprojection error between the detected points and the points projected from the 3D model. As a result, our complete pose estimation system works well without the knowledge of the initial guess overcoming the typical limitation of the iterative perspective-n-point solution.

The presented approach requires also detailed 3D locations of the keypoints on the charging station. Although a CAD model can be used for that purpose, in the experiments involving a mockup of the charging station that was assembled partially using non-standard elements, we produced a detailed 3D model of the station using a SURPHASER 100HSX 3D laser scanner that captured a mesh-based model from a single point of view with an accuracy of 1 mm. With this approach, we were able to modify the location of points as needed when the whole mast was already mounted in place. For the system's deployment, we would see these markings to be already mounted on the components of the mast and then mounted in the desired location. Therefore, the production-ready system would work based on the 3D CAD model without the need for an accurate 3D laser scanner.

3.6. Pose estimation error. The result of our entire processing pipeline is a pose estimation of the charging station (E) with respect to the camera coordinate system (C) noted as ${}^{C}\mathbf{T}_{E}$ (Fig. 7). In practice, our system has to provide the pose of the bus front axis (F) with respect to the charger coordinate system (E), as this information is required to plan and control the motion of the bus. To gather the ground truth data for evaluation, we prepared two masts mounted to the roof of the bus.

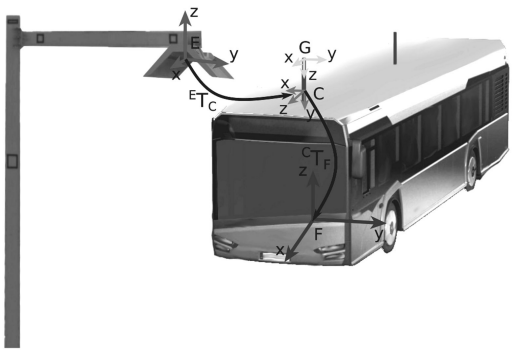

Fig. 7. Overview of the coordinate systems: camera (C), DGPS (G), charger's head (E), and bus front axis (F). We are interested in the localization of the bus front axis with respect to the charger's head.

In the first setup, mounted at the front of the bus, an FLIR camera, and a GPS antenna were attached. The second mast supports the second GPS antenna and was placed about 5 meters behind the front one to achieve an accurate orientation estimate from the DGPS. The final estimate is computed as

$${}^{E}\mathbf{T}_{F} = \left({}^{C}\mathbf{T}_{E}^{-1}\right) {}^{C}\mathbf{T}_{F}, \quad (4)$$

where ${}^{E}\mathbf{T}_{F}$ is the pose of the bus's front axis with respect to the charger, ${}^{C}\mathbf{T}_{E}$ is the original measurement of the electric charger in the camera coordinate system, and ${}^{C}\mathbf{T}_{F}$ is the pose of the camera coordinate system with respect to the front axis of the bus.

The camera location on the bus (i.e., the ${}^{C}\mathbf{T}_{F}$ transformation) was determined from CAD files, and verified by manual distance measurements and the camera setup attitude obtained from an Inertial Measurements Unit (IMU) XSens MTi attached to the camera's mast.

4. Experiments

We have proposed two alternative deep neural network architectures for extraction of the keypoints used for camera pose estimation. Hence, we examine the performance of the positioning system using either the RKN or the MRHKN networks. Both architectures are tested on three different spatial arrangements of keypoints. The first arrangement, called the *head*, consists of four points located in the corners of the charger's head (Fig. 8(a)). The second configuration, called *corners*, uses two points located on the corners of the head, and two other points, located on the supporting mast, where the head is attached (Fig. 8(b)). The third configuration, called *markers*, employs points located inside simple artificial landmarks—small rectangles made from black

tape and located on both the head and mast (Fig. 8(c)). The evaluation of both solutions allows us to determine which architecture should be used in the final system.

Tests with the different spatial arrangements of the keypoints assess the influence of the spatial layout and type of physical features (i.e., natural corners or markers) on the recall of the point detector and the accuracy of the computed pose. These results help us to determine the best configuration of the keypoints on the charging station. All experiments were conducted using Nvidia 1080-Ti GPU except evaluation of Papandreou's network in Section 4.5 which, due to the memory requirements of this model, involved a more recent and powerful Nvidia A100.

4.1. Experimental setup and image sequences. The dataset used for training the neural networks was gathered in May and June, mostly during sunny weather. For data gathering, we used two electric buses, a single-body, 12 meters long, and another one, 18 meters long and articulated. The bus driver performed a variety of paths toward the charging station to form a diverse dataset. The training dataset consists of 1000 manually labeled images, and each image was augmented by applying random brightness and contrast changes, random resizing, and cropping. Augmentation increased the training dataset to the size of 10 000 different samples.

The proposed methods were evaluated on a dataset of images gathered using an 18 m articulated bus over five days in late autumn during cloudy, rainy, and sunny weather (Fig. 9). This dataset consists of 81 sequences when the bus followed various trajectories towards the charging station. The diversity of the data was achieved by assuming different starting points and starting orientations, curved or slalom-like paths while enforcing various bus speeds along the paths (Fig. 10). During these maneuvers the vision-based positioning system was active, but the bus driver did not use the ADAS-generated suggestions for driving.

We purposefully let the driver take maneuvers far from the usual way the bus approaches the charging station in order to have more diversified trajectories, including oscillations and sharp turns. The maneuvers resulted in many trajectories that did not end with successful docking (plotted with dotted lines in Fig. 10), but we wanted to know if the vision system can position the bus also in such scenarios. In this dataset, the total time when the entire charging station's mast is visible on the camera is 1630 seconds (∼27 min), which equals 12366 frames. Due to its smaller dimensions, the charging station's head was visible longer on these recordings – 1783 seconds (∼30 min), which equals 13530 frames.

Unfortunately, to the best of our knowledge, there exists no publicly available dataset that could be used to evaluate our approach on third-party data. In particular,

Fig. 8. Location of keypoints for charger pose estimation: four points located on the head and called *head* (a), four points located on the natural corners of the head and the mast, called *corners* (b), and four points located inside the artificial markers, called *markers* (c).

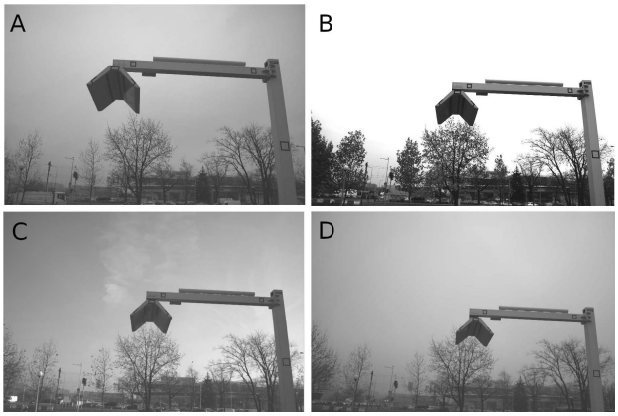

Fig. 9. Example images from the test dataset with different weather and lighting conditions: cloudy morning (a), sunny midday (b), sunny morning (c), foggy morning (d).

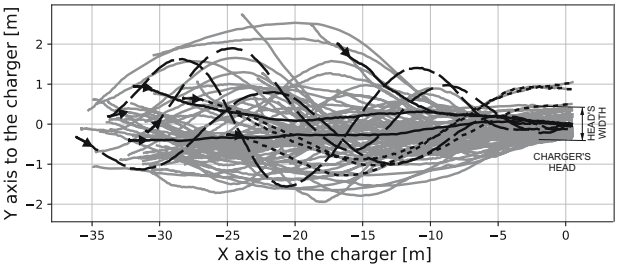

Fig. 10. Bus trajectories used to gather the evaluation sequences. Plotted lines highlight a few representative trajectories that are straight and end in proper docking (solid black line), are unrealistic in real-world scenarios, but still end in proper docking (dashed line), or miss the charger's head by a large margin imitating a driver not following the suggestions of ADAS (dotted line). Short arrows demonstrate start orientations for example trajectories. Notice that the vertical axis is scaled differently than the horizontal one in order to make the plot with a large number of trajectories more readable.

we cannot use a human pose estimation dataset, such as MPII Human Pose (Andriluka *et al*., 2014) to perform a fair comparison of keypoint detection, as our system is tailored specifically to the task of positioning with respect to a rigid object.

4.2. Ground truth and the evaluation procedure. To evaluate the performance of the localization method, we used a DGPS system (Ublox C099-F9P boards with ZED-F9P modules) with two receivers mounted on the bus and one serving as an external reference station placed nearby the experimental site. The DGPS system was working in the moving base scenario with external corrections from the reference station providing the location of the bus with an accuracy of approximately 1 cm and 1° when working in the RTK (Real-Time Kinematic) mode (u-blox, 2020).

The accuracy of the proposed camera-based system was compared with the DGPS position estimates from the perspective of the requirements of the motion planning and control system. In practice, some detections from the neural networks might be wrong and have to be rejected. We manage to filter most of these wrong pose measurements by checking how well the detected points fit the 3D model points projected onto the image plane. We invalidate the measurements if the RMSE of all charger points exceeds a threshold of 10 pixels. The remaining detections, considered valid, are evaluated using the 2D pose of the camera (location in the ground plane and orientation as a single yaw angle) with respect to the DGPS measurements. Errors of this 2D pose reflect errors in those components of the bus pose that influence motion planning and control procedures (Fig. 11).

Despite our efforts, a small percentage of the computed camera poses might have large translational errors that are easy to reject while executing the path, considering the time relations between consecutive detections. However, we decided to treat each detection independently including these inaccurate detections in the evaluation and statistics, as we only evaluate the approach to vision-based localization, without the bus odometry model, which has to be identified for each vehicle type (Michałek *et al*., 2020).

For each evaluated configuration of the positioning system, we present plots of the cumulative distribution functions (CDFs) of errors, which makes it possible to visually distinguish between the number of valid detections, and the whole distribution of errors reported on the testing sequences. Note that in the presented experiments we used multiple initial guesses for each detection to ensure that each detection is independent of the previous ones, we did not use the odometry, and the DGPS was used only to obtain ground truth.

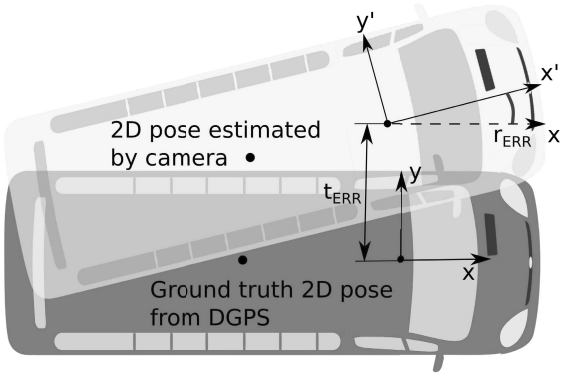

Fig. 11. Vision-based localization system is evaluated with respect to the translation error on the ground plane (2D position) and the orientation error (yaw) understood as the difference between the estimated and the ground truth headings of the bus.

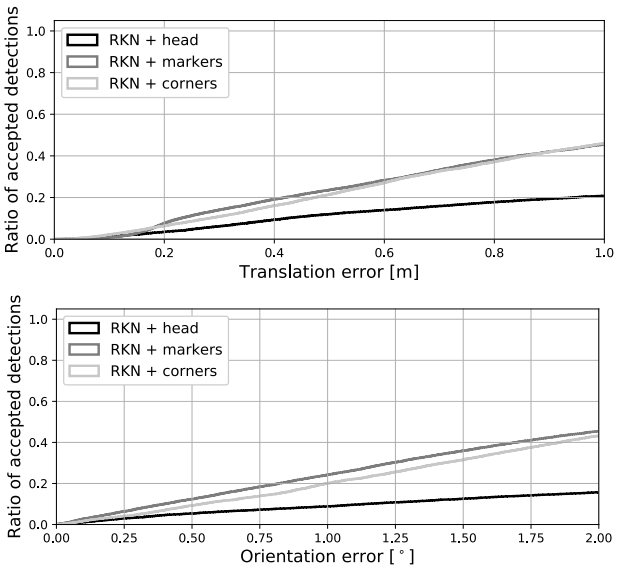

Fig. 12. Cumulative distribution functions of 2D translation error (up) and orientation error (down) for the scenarios considered using RKN with *head*, *markers*, and *corners* points for pose estimation.

4.3. Evaluating the RKN approach. The first aspect of detection efficiency that should be compared is the ratio of accepted detections to all frames with a visible charger. The worst version, i.e., *head*, managed to properly detect keypoints on about 77% of frames. Version *markers* and *corners* achieved better performance with a coverage of 84% and 89% percent, respectively.

The results of the pose error evaluation of this approach are shown in Fig. 12. The first observation to be made is that the distribution of the translation errors is almost linear. The slope of the *markers* and *corners* versions is similar and steeper than the *head*

version. The distribution of the rotation error for *markers* and *corners* versions is more convex in comparison with the *head* version which means that those methods significantly better cope with estimation of the rotation angles. Quantitative results show that the median 2D translation error of the *head* version is nearly 4 meters and the median error of yaw angle is 10°. The fact that the points are relatively densely packed in the image contributes to these errors.

As mentioned in the RKN approach description, this method has a bottleneck of size 32×32 pixels, which probably drops the information about the location of such densely packed points during image processing by the network. The densely packed keypoints also make the perspective-n-point algorithm often return unreliable camera pose estimates, because small changes in the location of keypoints in the image result in relatively large changes in the estimated 3D pose. Moreover, some of the keypoints could not be accurately labeled for training because of the rounded corners of the charging station's head. Also, large changes in the range of object observation that occurs during the entire maneuver make it very hard to keep a proper camera focus for all frames, which results in some blurry images (Fig. 13(a)).

The *corners* and *markers* versions achieved median 2D errors of 0.97 m, 1.94°, and 0.91 m, 1.96°, respectively. These errors are significantly smaller than the errors in the *head* configuration, and are similar; therefore, it is hard to tell which version is better. However, despite the improved performance, neither *corners* nor *markers* version meets the requirements as to the accuracy of localization in the ADAS system. A closer visual inspection of the results revealed that in both configurations the estimated locations of keypoints were inaccurate, e.g., the keypoints significantly missed the centers of landmarks, as shown in Fig. 13(b). This suggests that there is a necessity to modify the image processing system to achieve more accurate locations of the keypoints.

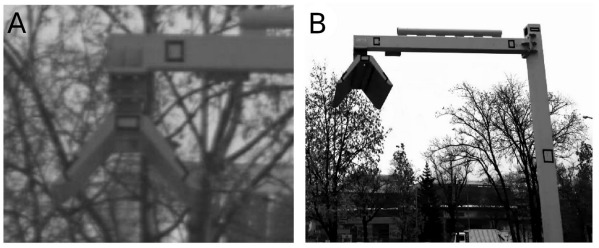

Fig. 13. Example of a blurry image of the head with round corners (a) and inaccurately detected keypoints from RKN (b).

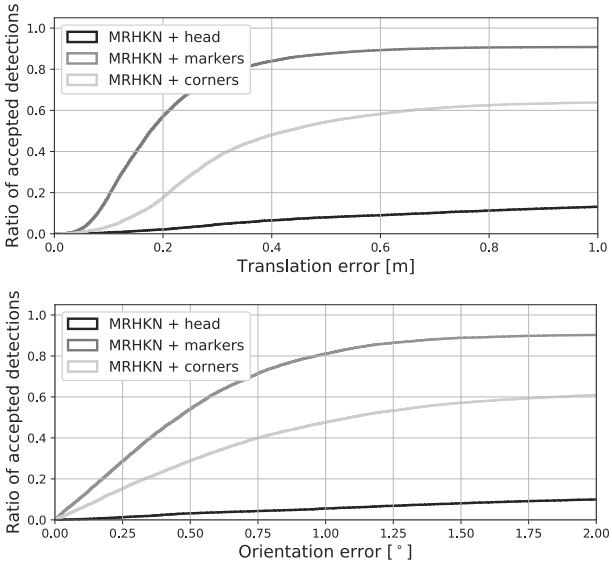

Fig. 14. Cumulative distribution functions of 2D translation error (top panel) and orientation error (bottom panel) for the MRHKN approach with *head*, *markers*, and *corners* points for pose estimation. The version *head* performs poorly due to the small size of the object. Natural *corners* points work worse than the artificial *markers* approach.

4.4. Evaluating the MRHKN approach. The MRHKN approach with three distinct point configurations was evaluated on the same dataset and the obtained localization errors are presented as cumulative error distributions in Fig. 14.

Evaluation of the *MRHKN + head* version resulted in 87% of the accepted detections with a median error in 2D translation of 6.46 m and a yaw angle median error of 18°. These results clearly show that this version should not be considered for localization in ADAS.

Although the *MRHKN + corners* version achieved only 66.6% of the accepted detections, the shape of the error curves in Fig. 14 shows that there is an improvement in the accuracy of pose estimation with respect to the RKN model. The median of 2D translation error equals 0.28 m and the yaw error median of about 0.6° highlights the importance of the keypoints location used for pose estimation. The majority of invalid detections are for distances larger than 25 meters. One of the possible causes is the fact that this method uses two points located on the charger's head, which are not clearly visible from long distances, as shown in the previous section.

The best result (90.9% accepted detections) was achieved by the *MRHKN + markers* version. Considering the requirements defined in Section 3 we assume that this configuration should be sufficient for localization. The quantitative evaluation resulted in a 2D translation median error of 0.17 m and a yaw median error of 0.41°. This version better detects the keypoints from large distances, and the median error for camera positions more distant than 25 m from the charging station is significantly

smaller compared with the *MRHKN + corners* version. Those results confirm that even very simple and cheap artificial markers improve the robustness of the keypoint detection procedure.

4.5. Comparing our models with existing solutions. The neural network architectures considered for comparison were HRNet and SCNet from the MMPose framework (MMPose, 2020), the Faster R-CNN-based architecture from the work of Papandreou *et al*. (2017) and the ResNet101 backbone with our keypoint extraction head. They were evaluated in a uniform way, and the numerical results of this evaluation are gathered in Table 1. In general, it can be seen that the results produced using natural *corners* are worse than those obtained with artificial *markers*. As previous tests revealed that the *head* points arrangement performs poorly for our network architectures, we decided to omit this arrangement in the comparison of methods. When *markers* are applied, the MRHKN approach reports the greatest percentage of accepted detections with the lowest median translation and the lowest median rotation error. The RKN approach reports the largest median translation and rotation errors among the evaluated solutions, but the lowest percentage of accepted detections can be observed for HRNet.

Unfortunately, the model of Papandreou *et al*. (2017) failed to run with 960×960 size images due to the insufficient RAM of the GPU card. The reason is inherent to this network architecture, as for each point, Papandreou's network generates heatmaps and two offset maps (for the x and y axes), and calculates the final location of the point based on these data. To calculate the offsets, it is required to create four-dimensional tensors, with each dimension matching the length of one of the heatmap sides. This takes a lot of GPU memory, and as a result, the largest input image size we were able to run on a recent Nvidia A100 card with 40 GB of RAM was 500×500 pixels. As for as the processing time is concerned, the MRHKN network produces better results than Papandreou's model, using a nearly double image input size. Running on the Nvidia A100 GPU, Papandreou's network achieved 4 FPS, requiring 32 GB of RAM, while MRHKN ran at 5.5 FPS using only 8 GB of the GPU RAM.

Figure 15 presents the performance of RKN and MRHKN approaches compared with that of the model from (Papandreou *et al*., 2017), and state-of-the-art solutions available in the MMPose library, while trained and evaluated on the images with *markers* ground truth points. The *corners* arrangement is no longer considered taking into account the significantly worse performance compared with the *markers* arrangement. The shape of the curves shows that the MRHKN method significantly better estimates both the position and orientation of the camera with respect to the remaining methods. For our RKN and Papandreou's method, the error distribution is roughly linear and worse than for the evaluated ResNet101, HRNet, SCNet, and our MRHKN models. We assume that the worse results of the state-of-the-art networks compared with our MRHKN solution are caused by the different architecture of the keypoint head which contains a smaller number of convolution layers. The HRNet result can be also worse because of the relatively small number of feature maps returned by the backbone network. All methods implemented in MMPose (HRNet, SCNet, and ResNet101) perform quite similarly with the best translation error obtained with ResNet101 but smaller orientation errors can be observed for the HRNet and SCNet.

Table 1. Comparison between the number of accepted detections, median 2D translation errors, and median 2D rotation errors of the RKNand MRHKN models and state-of-the-art HRNet, SCNet, ResNet101 (with head), and Papandreou's approaches (best results are bolded). The model marked with * was evaluated with image sizes reduced to 500×500 due to memory limits.

Method	Version	Fraction of accepted detections	Median t_{2D} [m]	Median r_{2D} [deg]
ResNet101	corners	39.4%	0.54	1.14
HRNet	corners	44.5%	0.43	1.52
SCNet	corners	18.0%	0.64	0.93
Papandreou*	corners	40.9%	0.53	1.15
RKN	corners	**84.5**%	0.92	1.96
MRHKN	corners	66.6%	**0.28**	**0.60**
ResNet101	markers	87.7%	0.30	0.75
HRNet	markers	82.0%	0.34	0.59
SCNet	markers	88.7%	0.31	0.59
Papandreou*	markers	63.6%	0.47	1.10
RKN	markers	88.7%	0.97	1.95
MRHKN	markers	**90.9**%	**0.17**	**0.41**

Based on the presented evaluation, it is clear that the proposed MRHKN outperforms state-of-the-art solutions and is most appropriate to our needs. Therefore, only MRHKN is considered for the docking scenario and is evaluated in the ablation study (see Appendix).

5. Conclusions

This article introduced a system for accurate positioning of electric buses with pantographs with respect to their charging stations. The localization task is accomplished using a low-cost, monocular vision system and does not require mounting any active landmarks or additional equipment on the charging station. The core of the proposed approach is a trained neural network model that extracts predefined keypoints from high-resolution images of the charging

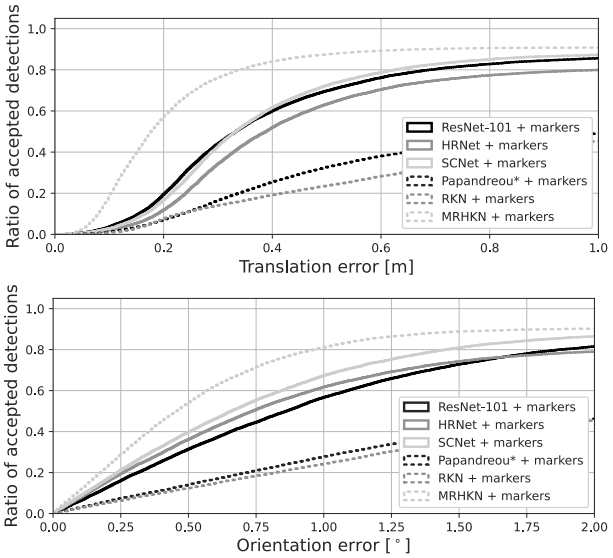

Fig. 15. Cumulative distribution functions of 2D translation error (top) and orientation error (bottom) for the RKN and MRHKN approaches compared with the approaches implemented in the MMPose framework and Papandreou's network reveal the superiority of the MRHKN approach. The model marked with * was evaluated with image sizes reduced to 500×500 due to memory limits.

station. This model is trained using a limited number of labeled images (1000 in our experiments), which makes the learning-based approach practical for deployment. Although a conventional approach based on a hand-crafted feature detector/descriptor can estimate the camera pose accurately, it struggles whenever the features are difficult to detect. Conversely, the approach proposed in this paper follows the conventional pipeline only with respect to the pose computation algorithm, while using a dedicated neural network as a feature extractor. Hence, we obtain a well-defined pattern of a few keypoints that are already associated with the 3-D model points, and we do not need to use RANSAC for outlier rejection.

Two alternative architectures for the extraction of keypoints are presented, one inspired by the well-known Faster R-CNN, but equipped with a new head for the detection of keypoints (RKN), and another one, that predicts the probability of keypoint location in the form of a heatmap (MRHKN). Experiments with these neural architectures revealed that keeping the highest possible resolution of the intermediate layers in the bottleneck of the network result in high accuracy of the keypoints location. On the other hand, evaluation of different layouts of the predefined keypoints resulted in the best pose estimation accuracy when projections of the points defined on the charger's station model span maximally the image space. Artificial markers additionally improve the reliability of detections, particularly from long distances. The proposed MRHKN model clearly outperforms the approach from (Papandreou *et al*., 2017) using the same backbone. It also outperforms selected state-of-the-art networks available in the MMPose library, reaching the localization accuracy that was required for the assisted docking task of ADAS. The achieved processing speed (1.25 FPS on Nvidia 1080-Ti and 5.5 FPS on Nvidia A100) is sufficient to successfully dock to the charging station.

Moreover, we provide in Appendix an ablation study of the best configuration, involving the MRHKN network and markers as keypoints. This study demonstrates the influence of such factors as the observation distance, observation angle, and acquired image size on the accuracy of pose estimation. These results allowed us to develop design recommendations for similar positioning systems. We also have determined the minimal camera parameters that keep an acceptable accuracy of positioning, which is critical for the cost-effective deployment of the proposed system on a production scale.

Acknowledgment

The research is part of the project *Advanced Driver Assistance System (ADAS) for Precision Maneuvers with Single-Body and Articulated Urban Buses,* co-financed by the European Union through the European Regional Development Fund within the Smart Growth Operational Programme 2014–2020 (contract no. POIR.04.01.02-00-0081/17-01). M.R. Nowicki is supported by the Foundation for Polish Science (FNP).

References

Andriluka, M., Pishchulin, L., Gehler, P. and Schiele, B. (2014). 2D human pose estimation: New benchmark and state of the art analysis, *IEEE Conference on Computer Vision and Pattern Recognition, Columbus, USA*, pp. 3686–3693.

Clarembaux, L.G., Pérez, J., Gonzalez, D. and Nashashibi, F. (2016). Perception and control strategies for autonomous docking for electric freight vehicles, *Transportation Research Procedia* **14**: 1516–1522.

Dreossi, T., Ghosh, S., Yue, X., Keutzer, K., Sangiovanni-Vincentelli, A. and Seshia, S.A. (2018). Counterexample-guided data augmentation, *Proceedings of the 27th International Joint Conference on Artificial Intelligence, Stockholm, Sweden*, pp. 2071–2078.

Fan, Y. and Zhang, W. (2015). Traffic sign detection and classification for advanced driver assistant systems, *International Conference on Fuzzy Systems and Knowledge Discovery (FSKD), Zhangjiajie, China*, pp. 1335–1339.

Gawron, T., Mydlarz, M. and Michalek, M.M. (2019). Algorithmization of constrained monotonic maneuvers for an advanced driver assistant system in the intelligent urban buses, *IEEE Intelligent Vehicles Symposium, Paris, France*, pp. 232–238.

Geiger, A., Lenz, P. and Urtasun, R. (2012). Are we ready for autonomous driving? The KITTI vision benchmark suite, *Conference on Computer Vision and Pattern Recognition, Rhode Island, USA*, pp. 3354–3361.

Girshick, R., Donahue, J., Darrell, T. and Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation, *IEEE Conference on Computer Vision and Pattern Recognition, Columbus, USA*, pp. 580–587.

Hartley, R.I. and Zisserman, A. (2004). *Multiple View Geometry in Computer Vision*, Cambridge University Press, Cambridge.

He, K., Zhang, X., Ren, S. and Sun, J. (2016). Deep residual learning for image recognition, *IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, USA*, pp. 770–778.

Kendall, A., Grimes, M. and Cipolla, R. (2015). Posenet: A convolutional network for real-time 6-DOF camera relocalization, *IEEE International Conference on Computer Vision (ICCV), Santiago, Chile*, pp. 2938–2946.

Kim, J., Cho, H., Hwangbo, M., Choi, J., Canny, J. and Kwon, Y.P. (2018). Deep traffic light detection for self-driving cars from a large-scale dataset, *International Conference on Intelligent Transportation Systems (ITSC), Maui, USA*, pp. 280–285.

Kukkala, V.K., Tunnell, J., Pasricha, S. and Bradley, T. (2018). Advanced driver-assistance systems: A path toward autonomous vehicles, *IEEE Consumer Electronics Magazine* **7**(5): 18–25.

Lepetit, V., Moreno-Noguer, F. and Fua, P. (2009). EPnP: An accurate $o(n)$ solution to the PNP problem, *International Journal of Computer Vision* **81**(2): 155–166.

Lim, K.L. and Bräunl, T. (2020). A review of visual odometry methods and its applications for autonomous driving, *arXiv* abs/2009.09193.

Liu, J.-J., Hou, Q., Cheng, M.-M., Wang, C. and Feng, J. (2020). Improving convolutional networks with self-calibrated convolutions, *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10093–10102, (online).

Lu, X. X. (2018). A review of solutions for perspective-n-point problem in camera pose estimation, *Journal of Physics: Conference Series* **1087**(5): 052009.

Luo, R.C., Liao, C.T., Su, K.L. and Lin, K.C. (2005). Automatic docking and recharging system for autonomous security robot, *IEEE/RSJ International Conference on Intelligent Robots and Systems, Edmonton, Canada*, pp. 2953–2958.

Marchand, E., Spindler, F. and Chaumette, F. (2005). ViSP for visual servoing: A generic software platform with a wide class of robot control skills, *IEEE Robotics and Automation Magazine* **12**(4): 40–52.

Michałek, M. and Kiełczewski, M. (2015). The concept of passive control assistance for docking maneuvers with n-trailer vehicles, *IEEE/ASME Transactions on Mechatronics* **20**(5): 2075–2084.

Michałek, M.M., Gawron, T., Nowicki, M. and Skrzypczyński, P. (2021). Precise docking at charging stations for large-capacity vehicles: An advanced driver-assistance system for drivers of electric urban buses, *IEEE Vehicular Technology Magazine* **16**(3): 57–65.

Michałek, M.M., Patkowski, B. and Gawron, T. (2020). Modular kinematic modelling of articulated buses, *IEEE Transactions on Vehicular Technology* **69**(8): 8381–8394.

Miseikis, J., Rüther, M., Walzel, B., Hirz, M. and Brunner, H. (2017). 3D vision guided robotic charging station for electric and plug-in hybrid vehicles, *arXiv* abs/1703.05381.

MMPose (2020). OpenMMLab pose estimation toolbox and benchmark, `https://github.com/open-mmlab/mmpose`.

Mur-Artal, R. and Tardós, J.D. (2017). ORB-SLAM2: An open-source SLAM system for monocular, stereo and RGB-D cameras, *IEEE Transactions on Robotics* **33**(5): 1255–1262.

Nowak, T., Nowicki, M., Ćwian, K. and Skrzypczyński, P. (2019). How to improve object detection in a driver assistance system applying explainable deep learning, *IEEE Intelligent Vehicles Symposium, Paris, France*, pp. 226–231.

Nowak, T., Nowicki, M., Ćwian, K. and Skrzypczyński, P. (2020). Leveraging object recognition in reliable vehicle localization from monocular images, *in* C. Zieliński *et al.* (Eds), *Automation 2020: Towards Industry of the Future*, Springer, Cham, pp. 195–205.

Olson, C. and Abi-Rached, H. (2010). Wide-baseline stereo vision for terrain mapping, *Machine Vision and Applications* **21**(5): 713–725.

Papandreou, G., Zhu, T., Kanazawa, N., Toshev, A., Tompson, J., Bregler, C. and Murphy, K. (2017). Towards accurate multi-person pose estimation in the wild, *IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, USA,* pp. 3711–3719.

Pérez, J., Nashashibi, F., Lefaudeux, B., Resende, P. and Pollard, E. (2013). Autonomous docking based on infrared system for electric vehicle charging in urban areas, *Sensors* **13**(2): 2645–2663.

Petrov, P., Boussard, C., Ammoun, S. and Nashashibi, F. (2012). A hybrid control for automatic docking of electric vehicles for recharging, *IEEE International Conference on Robotics and Automation, St. Paul, USA*, pp. 2966–2971.

Rahmat, R., Dennis, D., Sitompul, O., Sarah, P. and Budiarto, R. (2019). Advertisement billboard detection and geotagging system with inductive transfer learning in deep convolutional neural network, *TELKOMNIKA (Telecommunication Computing Electronics and Control)* **17**(5): 2659.

Redmon, J., Divvala, S., Girshick, R. and Farhadi, A. (2016). You only look once: Unified, real-time object detection, *IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, USA*, pp. 779–788.

Ren, S., He, K., Girshick, R. and Sun, J. (2015). Faster R-CNN: Towards real-time object detection with region proposal networks, *Advances in Neural Information Processing Systems, Montreal, Canada*, pp. 91–99.

Royer, E., Lhuillier, M., Dhome, M. and Chateau, T. (2005). Localization in urban environments: Monocular vision compared to a differential GPS sensor, *IEEE Conference on Computer Vision and Pattern Recognition, San Diego, USA*, Vol. 2, pp. 114–121.

Schubert, E., Sander, J., Ester, M., Kriegel, H.P. and Xu, X. (2017). DBSCAN revisited: Why and how you should (still) use DBSCAN, *ACM Transactions on Database Systems* **42**(3): 1–21.

Schunk Carbon Technology (2021). Schunk smart charging, `https://www.schunk-carbontechnology.com/en/smart-charging`.

Skrzypczyński, P. (2009). Simultaneous localization and mapping: A feature-based probabilistic approach, *International Journal of Applied Mathematics and Computer Science* **19**(4): 575–588, DOI: 10.2478/v10006-009-0045-z.

Taghibakhshi, A., Ogden, N. and West, M. (2021). Local navigation and docking of an autonomous robot mower using reinforcement learning and computer vision, *2021 13th International Conference on Computer and Automation Engineering (ICCAE), Bruxelles, Belgium*, pp. 10–14.

Toshpulatov, M., Lee, W., Lee, S. and Haghighian Roudsari, A. (2022). Human pose, hand and mesh estimation using deep learning: A survey, *The Journal of Supercomputing* **78**(6): 7616–7654.

Triggs, B., McLauchlan, P.F., Hartley, R.I. and Fitzgibbon, A.W. (2000). Bundle adjustment—A modern synthesis, *in* B. Triggs *et al.* (Eds), *Vision Algorithms: Theory and Practice*, Springer, Berlin, pp. 298–372.

u-blox (2020). ZED-F9P: u-blox F9 high precision GNSS module, `https://content.u-blox.com/sites/default/files/ZED-F9P-04B_DataSheet_UBX-21044850.pdf`.

Vivacqua, R., Vassallo, R. and Martins, F. (2017). A low cost sensors approach for accurate vehicle localization and autonomous driving application, *Sensors* **17**(10), Article no. 2359.

Wang, J. and Olson, E. (2016). AprilTag 2: Efficient and robust fiducial detection, *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Daejeon, Korea*, pp. 4193–4198.

Wang, J., Sun, K., Cheng, T., Jiang, B., Deng, C., Zhao, Y., Liu, D., Mu, Y., Tan, M., Wang, X., Liu, W. and Xiao, B. (2021). Deep high-resolution representation learning for visual recognition, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **43**(10): 3349–3364.

Xiang, Y., Schmidt, T., Narayanan, V. and Fox, D. (2018). PoseCNN: A convolutional neural network for 6D object pose estimation in cluttered scenes, *Proceedings of Robotics: Science and Systems, Pittsburgh, USA*.

Youjing, C. and Shuzhi, S.G. (2003). Autonomous vehicle positioning with GPS in urban canyon environments, *IEEE Transactions on Robotics and Automation* **19**(1): 15–25.

Zhang, W., Fu, C. and Zhu, M. (2020). Joint object contour points and semantics for instance segmentation, *arXiv* abs/2008.00460.

Tomasz Nowak received his BSc degree in automatic control and robotics at the Poznan University of Technology (PUT). In 2018 he received his MSc in computer science from the same university. Since 2020 he has been a PhD student at the PUT, working on data-efficient and explainable machine learning in visual perception for autonomous vehicles.

Michał R. Nowicki is a graduate of the Poznan University of Technology (PUT), having received his BSc and MSc in automatic control and robotics in 2013 and 2014, respectively, and his BSc in computer science in 2014. He obtained (with honors) a PhD in robotics in 2018, and currently is a research assistant professor in the Institute of Robotics and Machine Intelligence at the PUT. His research interests include robot perception and localization, particularly SLAM, and applications of optimization-based techniques in robotics.

Piotr Skrzypczyński received his PhD and DSc degrees in robotics from the Poznan University of Technology (PUT) in 1997 and 2007, respectively. He is a professor in the Institute of Robotics and Machine Intelligence (IRIM) at the PUT, and the head of the IRIM Robotics Division. Professor Skrzypczyński also leads the Mobile Robotics Laboratory at the IRIM. His current research interests include AI-based robotics, robot navigation, localization and SLAM, autonomous vehicles, computer vision, and machine learning.

Appendix

Ablation study of the *MRHKN + markers* configuration

The experiments presented in Section 4 showed that the new *MRHKN* detection method yields more accurate keypoint locations than the RKN architecture adopted from Faster R-CNN. As expected, the best results are achieved with the artificial markers attached to the charging station's structure. As the minimal markers are an acceptable and easy to deploy modification to the existing infrastructure, we recommend this variant, named *MRHKN + markers*, for practical applications. In this section, we conduct an ablation study of this configuration to explore in-depth its characteristics. These experiments evaluate how the recommended variant of the localization system performs depending on the distance, observation angle between the camera and the charging station, and the speed of the bus motion. These parameters determine

the range of docking maneuvers, where vision-based localization can be safely used in ADAS. Moreover, the ablation study allows us to select the minimum image resolution in our method, thus supporting the design choices for the hardware configuration of the localization system, in order to achieve the required accuracy at a minimal cost.

A1. Performance dependence on the distance to the charging station

The distance to the observed object affects the accuracy of pose estimation. All accepted detections from the *MRHKN + markers* model were divided into 10 bins ranging from 7 to 37 meters, and the translation and rotation median errors were calculated for each of them (Fig. A1). As could be expected, the translation estimation error increases with the observation distance. A significant drop in the accuracy appears at distances greater than 30 meters. However, while the translation error increases with the distance, the rotation error is approximately constant. These error characteristics are acceptable for the motion planning and execution procedures (Michałek and Kiełczewski, 2015), as the rotation error remains small even for distances exceeding 30 meters, which is crucial for trajectory planning, while the translation error decreases as the bus approaches the charging station. Within the last few meters, the translation error drops to about 10 cm, which makes it possible to use the bus odometry if the roof-mounted camera no longer sees all the markers. Once accurately positioned with respect to the charger station head and being very close to the station, the bus moves along a straight line and can safely plug in the pantograph using its mechanical adaptation system.

A2. Performance dependence on the observation angle of the charging station

Another factor that may influence the accuracy of the localization system is the observation angle. In our test dataset, all observations were acquired at angles smaller than 15°, as we assume realistic maneuver scenarios when at the start of the maneuver the bus is moving along a lane roughly parallel to the x axis of the charger station's coordinate system, while the observation angle is mostly depending on the lateral offset between the bus path and the charging station located at the roadside (cf. Fig. 1). In much the same way as in the previous analysis, all observations were divided into 10 bins, and the median error is presented in Fig. A2. The chart shows that the translation error is rather independent of the observation angle, which is the desired property, as the system can handle even more unusual scenarios with

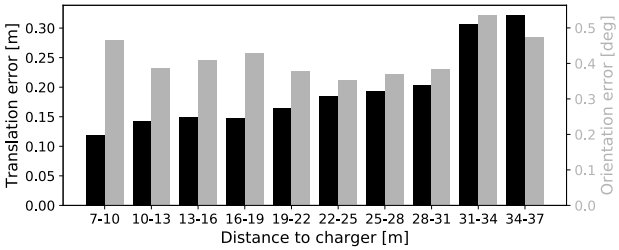

Fig. A1. Distribution of the translation error (black bars) and orientation error (gray bars) as a function of the distance to the charging station for *MRHKN + markers*.

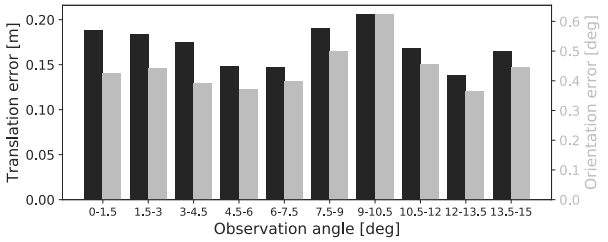

Fig. A2. Distribution of translation error (black bars) and orientation error (gray bars) as a function of the orientation to the charger for *MRHKN + markers*. Errors are similar regardless of the observation angle.

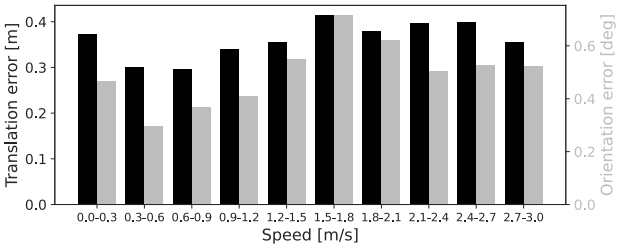

Fig. A3. Distribution of translation error (black bars) and orientation error (gray bars) as a function of the bus speed for *MRHKN + markers*. We observe similar errors regardless of the bus speed.

greater observation angles that might be underrepresented in the charger training dataset. Similarly, the accuracy of the yaw angle estimation does not depend on the observation angle. Comparing the values of the error with Fig. A1, we can conclude that the observation angle within the ranges considered is a less important factor in determining the accuracy of localization than the distance.

A3. Performance dependence on the bus speed

The task of docking with the bus to a charging station is accomplished at low speed, as it requires precise steering. The whole maneuver is less than 40 meters long and at the end, the bus must stop. Bus operators typically require the drivers not to exceed the speed of

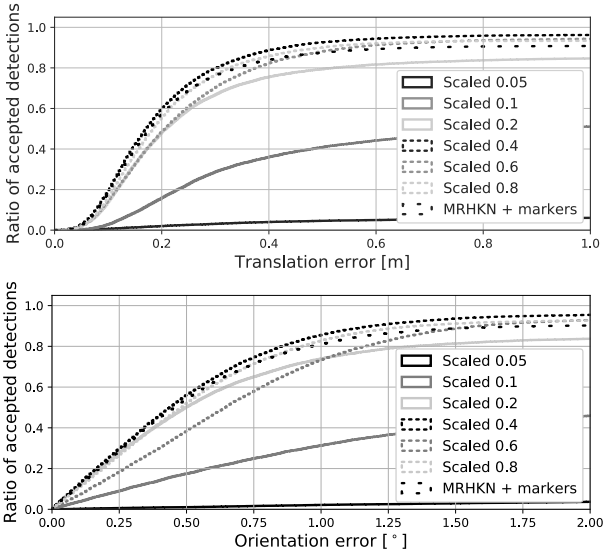

Fig. A4. Cumulative distribution of translation and orientation errors for experiments with reduced image sizes for *MRHKN + markers*. We notice no significant drop in performance even if the image is reduced up to 2188×1459 pixels.

Table A1. Performance comparison between versions of the *MRHKN + markers* method configured with different image sizes during training and testing.

Scaling factor	Image size	Fraction of accepted detections	Median t_{2D} [m]	Median r_{2D} [deg]
0.05	273×182	7.1%	0.35	1.95
0.1	547×364	56.4%	0.30	0.87
0.2	1094×729	85.5%	0.18	0.42
0.4	2188×1459	96.3%	0.17	0.42
0.6	3283×2188	94.8%	0.20	0.61
0.8	4377×2918	93.4%	0.18	0.45
1.0	5472×3648	90.9%	0.17	0.41

20 km/h while approaching the charging station, but in practice, the drivers use a much smaller speed for docking. Nevertheless, we evaluate the vision-based localization for the full range of speed values we have observed in our docking experiments. As shown in Fig. A3, neither the translation, nor the rotation estimation error depends on the speed. Despite using the camera with a rolling shutter, the performance was not affected by the vehicle motion, as the MRHKN approach ensures a robust detection of keypoints.

A4. Reduced image size performance

The camera finally used in the buses fielded by an operator can be a different one than the high-resolution camera used to acquire the training data for the previously presented results. Therefore, during the evaluation, we examined the influence of image size on the pose estimation accuracy (Fig. A4) in order to determine the minimal camera resolution resulting in localization accuracy within set requirements.

To achieve comparable results between different resolutions, we simulate the camera with a lower resolution by resizing the training and testing data to a fraction of the original size while appropriately rounding the ground truth keypoint locations used for training. Using this method, we trained six different models with scaled image resolutions to represent performance with lower resolution images. We resized the width and height of the image by the scaling factors of 0.05 (*Scaled 0.05*), 0.1 (*Scaled 0.1*), 0.2 (*Scaled 0.2*), 0.4 (*Scaled 0.4*), 0.6 (*Scaled 0.6*), and 0.6 (*Scaled 0.8*). In each case, a new network was trained and tested on appropriately scaled images with ground truth labels rounded to discrete pixel values, fully representing the training process on resized images.

Performance is similar to the original *MRHKN + markers* approach both on accepted detection coverage and pose estimation error for the versions *Scaled 0.8*, *Scaled 0.6*, and *Scaled 0.4*. For those versions, we observe the percent of accepted detections exceeding 90% while the reported median pose errors are below 0.17 m and 0.61°. A further reduction of the image size caused a noticeable performance loss. Version *Scaled 0.2* preserved the acceptable median errors of 0.18 m and 0.42° values but the percentage of accepted detections drops to 85.5%. As it could be foreseen, the image reduction affects mostly detection coverage and pose accuracy on larger distances. A further reduction leads to worse performance for version *Scaled 0.1* while completely breaking down for the *Scaled 0.05* version, which detects less than one-tenth of charger keypoints detected by the original network working on full-size images. Numerical results for all versions considered are summarized in Table A1.

Satisfying the performance of the localization system using images of reduced resolution makes it possible to use a lower resolution camera on the bus. This should broaden the choice of industrial-grade cameras that can be employed in the production version of the system (considering the interface, degree of protection provided by the enclosure, etc.), but also decrease the costs. From the results of our experiments, we conclude that a camera with a resolution of 2188×1459 (4 MP class) would be a great fit for the presented positioning system, providing accurate results for a fraction of the cost of the 20 MP camera used in the experiments.

Received: 9 January 2022
Revised: 17 May 2022
Re-revised: 25 July 2022
Accepted: 27 July 2022