Monocular End-to-End Vehicle Pose Estimation for Car Manufacturing

Ahmad Kamal, Jaime Valls Miro
University of Technology, Sydney, Australia

Abstract

An efficient vehicle pose estimation method from a monocular camera to assist paint defect identification in a car manufacturing setting is presented in this paper. Inspired by promising results reported by the self-driving car community with end-to-end schemes, a cascaded deep neural network is proposed for rapid estimation of both translation and rotation of a moving vehicle along a production line, achieving pose estimate average errors below $1.0cm$ in translation and 0.009° in rotation on a ground-truth synthetic database. Notably compelling for the purpose of potential deployment in real factory settings is the ability to infer poses within 1 second. Comprehensive experiments are presented to determine the most accurate camera configuration, and comparisons to traditional two-stage iterative image processing and pose optimisation methods are also provided to demonstrate the network's superior performance in provided accurate vehicle pose estimates in real-time.

1 Introduction and Motivation

Quality control is a crucial component in automobile manufacturing production. Assessing the physical properties at the various painting stages is fundamental to the quality of the final product. Traditionally, the inspection procedure has been performed manually, resulting in reports of up to 80% of defects being overlooked [Armesto *et al.*, 2011], thus compromising the vehicle's physical appearance. Worst-case scenario occurs when extensive defects are detected late in the production process, which leads to the unit failing final quality control metrics and the whole vehicle needing to be stripped and its surface treated in full.

Extensive research in recent years has established automated computer vision surface inspection systems as a maturing alternative to provide more efficient and accurate processes for defect detection [Solanes *et al.*, 2018]. Skilled workers however are still required to manually mend these imperfections once identified, meaning that the vehicle is required to stop for this task to take place. An example of a light tunnel system for automated defect identification, and a surface treatment team in action at an automotive plant production line are depicted in Fig. 2.

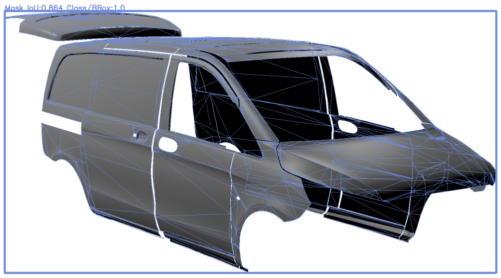


Figure 1: Given a single RGB image of a simulated vehicle, an end-to-end trained CNN estimates the vehicle's 6-DoF pose in real-time (bound within 1 sec). The prediction is illustrated by the reprojected 2D mask, obtained by overlaying the corresponding CAD model mesh onto the image and the outer bounding box (both in blue), as transformed by the attained pose.

Developing a system capable of repairing defects on a vehicle's surface autonomously, eliminating the need to halt the automobile for surface treatment is most desirable from a manufacturing standpoint, thus being able to increase efficiency and throughput. There is however a requirement for a perceptual system to be able to compute an accurate estimate of the vehicle's pose, and do so swiftly to accommodate for the manufacturing plant strict cycle times.

This work proposes using a convolutional neural network (CNN) for rapid vehicle pose estimation targeting a vehicle manufacturing plant setting. The network is trained end-to-end, taking as input an RGB image to

(a) Inspection light tunnel system in a Mercedes automotive plant [Munoz *et al.*, 2019].

(b) Fixing defects identified by a visual detection system in a Ford automotive plant [Armesto *et al.*, 2011].

Figure 2: Illustration of defect rectification processes in modern car manufacturing plants.

provide the estimated vehicle's pose in the camera frame (Fig. 1). The proposed scheme determines the camera viewpoint configuration that yields the most accurate result, both in terms of pose accuracy and mean average precision (mAP). The former demonstrates the system's ability to correctly estimate the pose of a vehicle in an image, while the latter evaluates this task over multiple instances; making both metrics essential in the mass production process of vehicle manufacturing. The proposed approach not only compares favourably with traditional visual optimisation methods for pose estimation on these metrics, but it also exhibits significant improvements in prediction processing times.

Extensive analysis of the implemented solution with synthetic ground truth data derived from real vehicle CAD models is undertaken to prove the system's feasibility for industrial applications. To the best of our knowledge, the use of a deep neural network to estimate the pose of a vehicle in an automotive factory setting has not been previously reported.

2 Related Work

In this section the work is placed in context in a taxonomy that compares more traditional 2D-3D computer vision schemes, two-stage neural network approaches and end-to-end CNNs.

2.1 Non-Deep Learning Methods

Various alternatives were explored such as 3D reconstruction (feature-based [Torr and Zisserman, 1999] and direct [Irani and Anandan, 1999] image alignment) and edge matching methods [Dharampal, 2015]. However, a drawback of the former is the method's reliance on the overlap of adjacent cameras for point correspondence, making edge matching the more plausible approach.

The edge matching application is comprised of two main stages: edge detection and pose optimisation. Essentially, edges are extracted using images captured from real cameras and their simulated equivalents, given a vehicle's CAD model. Based on the data from the simulated images, a point cloud in the world frame is obtained; the latter is then transformed by a pose estimate and then projected onto the respective camera's image plane. Furthermore, Iterative Closest Point (ICP) [Besl and McKay, 1992; Censi, 2008] is used for 2D-3D point correspondence, establishing a relationship between the closest two points in the real and simulated edge images. This process consists of iteratively minimising the distance between the sets of real and simulated edge points until the optimal pose estimate is achieved. Moreover, the optimisation uses a combination of M-estimators (Huber and Tukey (bisquare)) to apply weights to residuals at every iteration, making it robust to outliers.

In exploring an alternative to the proposed method, the above two-staged approach (edge+ICP) was implemented with similar data and assessed with various camera array setups. Primarily, cameras (x23) positioned to point at a specific part of the vehicle, were first tested for their performance in pose optimisation. Then, a robust sub-set of these cameras (x12) was similarly evaluated; and lastly, a wide-view camera array (x3) was considered. The 3-camera configuration demonstrated the best results and was therefore, used for comparison to the CNN-based pose estimation proposed in this paper (Section 4.4).

2.2 Two-Staged CNN-Based Systems

Single image pose estimation combines the benefits of a deep neural network with those of geometric optimisation to estimate the pose of an object. Specifically, pose estimation is attained through a two-staged system where a CNN is trained to locate points of interest on an image (semantic keypoints), which are then put into an optimisation algorithm to retrieve the pose of the object being considered.

A stacked hourglass network [Newell *et al.*, 2016], comprised of encoder and decoder networks working in tandem, is trained on images annotated with 2D keypoints on an object, equivalent to those found on a predefined 3D model. Class-specific 2D probabilistic maps (heatmaps) are then generated, estimating the keypoint locations on a new image. The features extracted from each image are then matched to those on a 3D model and establish 2D-3D correspondences. Once the semantic keypoints are predicted by the network, their location on the image is input into an optimisation algorithm, initialised by PnP, to retrieve the pose of the desired object.

However, this two-staged approach makes the method prone to outliers, generally caused by inaccuracies in keypoint localisation as a result of partial occlusions or cluttered environments. This problem is solved [Pavlakos *et al.*, 2017] by assigning weights to the different predicted elements during 2D-3D correspondence, therefore taking the network's prediction uncertainty into account. Additionally, due to the instance-specific nature of PnP-based algorithms, pose estimation is extended from objects to classes of objects (e.g. cars, bikes, buses, aeroplanes) [Pavlakos *et al.*, 2017]. Several pre-defined 3D models are used to obtain a deformable shape model for an object class. Therefore, by combining the deformable model with an image's camera parameters, class-based pose regression could be achieved.

Multi-image pose estimation follows an approach similar to [Pavlakos *et al.*, 2017] in that it uses keypoint detection for model alignment to retrieve the pose of a vehicle. A similar method presented in [Ding *et al.*, 2018] differs from the previous paper in two main aspects; they present a four-layer stacked hourglass architecture (rather than two), and their keypoint localisation is based on input from multiple cameras (minimum of two) with overlapping fields of view.

Single image training can often result in deviations for keypoint localisation, caused by self-occlusions, which in turn results in a less accurate pose estimate. This method solves this issue by sending multiple views of the same image as input to the network, therefore, increasing its ability to correctly identify occluded keypoints during inference. Furthermore, inspired by the deformable shape model approach, this method uses a Hierarchical Wire Frame (HWC); a three-layered model constrained by keypoints that allow it to maintain a vehicle shape when establishing 2D-3D correspondences. HWC combined with the benefits of multi-monocular image training (e.g. object depth) constitute an approach more robust than that of single image training, even when faced with keypoint localisation inaccuracies.

2.3 End-to-End Pose Estimation

Previous work [Wu *et al.*, 2019] presents an end-to-end trained network, designed for detecting multiple vehicles in a city environment, while simultaneously estimating the pose of each detection. Originally, the framework, named 6D-VNET, was an extension of Mask-RCNN [He *et al.*, 2017], as it introduces customised branches for vehicle class, rotation and translation regression into the existing system. Additionally, the method implements a non-local block capturing mutual information among detected vehicles in the environment, therefore, allowing the network to collectively regularise pose estimation for detections, rather than analysing cases individually.

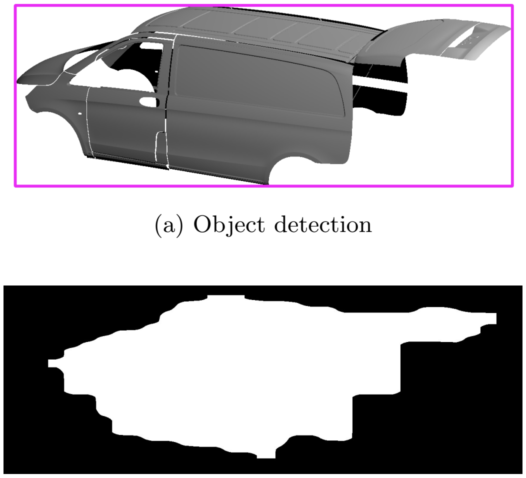
Improvements to 6D-VNET have been reported [Wu *et al.*, 2020], modifying the system's detection framework and establishing a post-processing step for increased network performance. The network was instead built upon a three-stage detector, Hybrid Task Cascade (HTC) [Chen *et al.*, 2019a; 2019b], that adopts feature information flow between each stage, as a mean of achieving better refinement and enhanced accuracy for pose estimation. Furthermore, as in 6D-VNET, three customised branches were added as an extension to HTC, where joint losses function cooperatively to train the network for pose estimation.

This approach demonstrates improvements over state-of-the-art two-step systems, where inaccuracies in the initial stage aggravate the errors during pose optimisation, and therefore, decrease the network's ability to estimate a vehicle's pose accurately.

2.4 Contributions

This work proposes a vehicle localisation method inspired by end-to-end methods derived from the self-driving car community, tailored to the less cluttered assembly-line setting in a manufacturing plant. The main contributions can be summarised as:

- A deep neural network capable of learning from a single input image is proposed to develop a robust system capable of estimating the pose of a vehicle in a factory setting.
- The use of real vehicle CAD models within a simulation package like Blender to capture a ground-truth dataset of images with realistic calibrated cameras (from a real manufacturing plant). The software is used to collect information on each image, essential for training the network and visualising predictions (e.g. ground-truth pose, 2D mask, bounding box coordinates).
- The vehicle class model can be safely assumed known, allowing tailoring the learning loss functions to more generic cases to the specific needs and circumstances of a manufacturing plant, thus enhancing the efficiency.
- An exhaustive experimental investigation is carried out based on simulated analysis through testing over



(a) Object detection

(b) Object boolean segmentation

Figure 3: Example output of HTC

numerous scenarios of camera setups. The improved performance is also demonstrated against the implementation and reported works of other comparable methods.

- A less time-expensive solution to vehicle pose estimation is presented in comparison to more traditional methods.

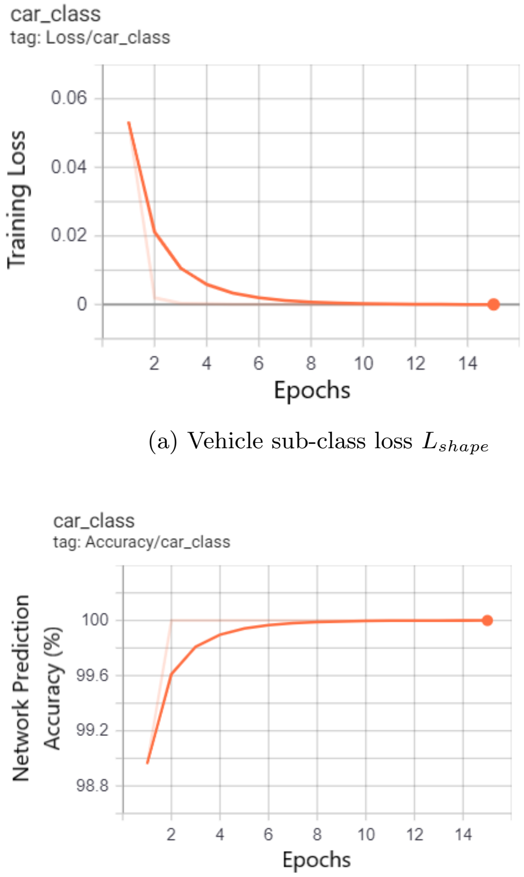
3 Neural Network

The application of vehicle pose estimation within a manufacturing plant requires a high degree of precision, especially for the purposes of quality control. Two-staged systems can often render inaccurate predictions; for instance, minor keypoint localisation errors could aggravate errors in 3D space, hence, resulting in a less precise pose estimate. For this reason, the implementation of a neural network trained end-to-end for pose regression is proposed. This paper discusses the results obtained using the framework presented in [Wu *et al.*, 2020]; note that specific adjustments are introduced to make the network fit for its application in a factory setting (Section 3.2).

3.1 Network Architecture

The network is built upon an existing framework, Hybrid Task Cascade (HTC) [Chen *et al.*, 2019a; 2019b], designed for object detection and segmentation. Furthermore, customised translation and rotation branches are additionally added for pose regression. The network takes as input a monocular RGB image and outputs a prediction consisting of the vehicle's bounding box coordinates, 2D mask and 6-DoF pose.

Hybrid Task Cascade (HTC) An interweaved three-staged network for object detection and segmentation is presented in [Chen *et al.*, 2019a; 2019b] (Fig. 3), establishing progressive refinement from one step to the next, hence, enhancing the output accuracy. Essentially, at each stage, the network performs both bounding box and mask regression, while creating information flow from one step to the next, which allows it to take into account previous outputs for current estimation calculations.



(a) Vehicle sub-class loss L_{shape}

(b) Vehicle sub-class prediction accuracy percentage

Figure 4: Network performance in predicting vehicle sub-class. Data was collected over 15 epochs.

Backbone HTC is implemented with a pre-trained High-Resolution Network (HRNet) [Sun *et al.*, 2019; Wang *et al.*, 2020] as its backbone; specifically, the object detection version, HRNetV2p [Wu *et al.*, 2020]. Primarily, HRNet is designed to provide high-resolution image representation by establishing a four-stage network with convolution streams running in parallel. As a result, image representations with higher spatial precision are achieved in comparison to traditional methods, where such representations are obtained from bottom-up processes connected in series.

Customised heads In establishing a network that outputs a 6-DoF pose of a vehicle, translation and rotation heads are introduced to the existing object detec-

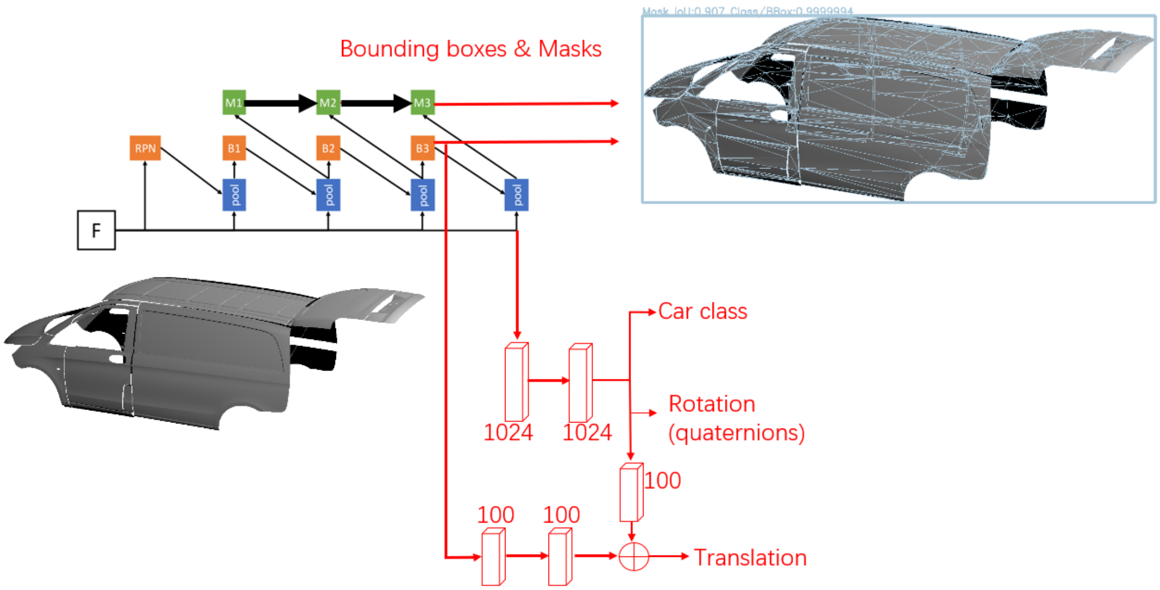


Figure 5: Overview of the implemented network architecture. The network takes a monocular image of a vehicle (left) as input, and outputs the car's 6-DoF pose estimate, bounding box and 2D mask (right). $\bigoplus$ corresponds to a concatenation operation. The black branch represents the multi-staged network HTC [Chen *et al.*, 2019b] and the red branch illustrates the customised heads introduced for pose estimation [Wu *et al.*, 2020].

tion and segmentation framework of HTC [Wu *et al.*, 2020]. Firstly, the rotation head analyses the features within the vehicle's bounding box to estimate a quaternion rotation. Similarly, translation regression uses these features with the additional input of bounding box coordinates, to compute a 3D translation vector.

3.2 Implementation

Joint losses Inspired by [Wu *et al.*, 2020], the network is trained end-to-end by minimising the training loss $\mathcal{L}$, comprised of the joint losses $\mathcal{L}_{det}$ and $\mathcal{L}_{inst}$ (Eq. 1). Firstly, $\mathcal{L}_{det}$ consists of the classification loss $\mathcal{L}_{cls}$, 2D bounding box loss $\mathcal{L}_{box}$ and 2D mask loss L_{mask}, which match those defined in the state-of-the-art detection network, Mask R-CNN [He *et al.*, 2017]. Secondly, $\mathcal{L}_{inst}$ refers to the instance loss for 6-DoF estimation (Eq. 2). It is defined by the joint losses of sub-class classification ($\mathcal{L}_{shape}$), rotation regression ($\mathcal{L}_{rot}$), and translation regression ($\mathcal{L}_{trans}$) as well as the loss weighting factor σ. Note that the weighting factor and translation and rotation losses are defined as per [Wu *et al.*, 2020]. However, this paper only considers the case of a single vehicle subclass (Mercedes Vito van), making the $\mathcal{L}_{shape}$ loss redundant. Fig. 4 illustrates this hypothesis, by demonstrating the network's robustness in identifying a vehicle's car class when only one model is considered. As a result, the car sub-class loss is eliminated in the pose estimation loss formula, thus, defining it as:

$$\mathcal{L} = \mathcal{L}_{det} + \mathcal{L}_{inst} \tag{1}$$

where

$$\begin{aligned}\mathcal{L}_{inst} = \mathcal{L}_{rot} \cdot \sigma_{rot}^{-2} + \log \sigma_{rot}^{2} \\ + \mathcal{L}_{trans} \cdot \sigma_{trans}^{-2} + \log \sigma_{trans}^{2}\end{aligned} \tag{2}$$

Environment modification A network intended for use in autonomous driving scenarios, is designed to identify various traffic participants in a city environment given a monocular RGB image, while simultaneously estimating their 6-DoF pose [Wu *et al.*, 2020]. This paper tailors the network for a less complex application, in that the system needs to instead detect vehicles one at a time. Furthermore, pose estimation data is obtained from a fixed monocular RGB camera in a less cluttered environment. As a result, rather than train the network on a pre-existing dataset, Blender is used as an alternative for creating a unique dataset, also allowing to better simulate the environment of a manufacturing plant. Moreover, this gave us the advantage of testing and analysing several configurations of camera positions in order to determine the optimal method to implement the system for its desired application. Section 4.1 provides further details on data collection.

3D Mesh Decimation For efficiency purposes, the 3D mesh of the vehicle is decimated to facilitate visualising predictions (see Section 4.1 for further details).

Training The network is implemented with PyTorch using two NVIDIA Quadro P4000 8GB GPUs; however, due to memory limitations, the batch size is set to one

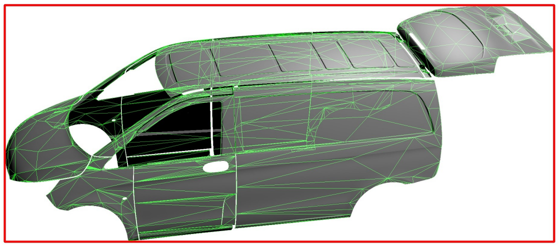


Figure 6: An example of the synthetic rendered vehicle images used in this work, illustrating the auxiliary outer-bounding box (in red) and 2D mesh mask (in green), given the annotation data obtained from Blender.

image per GPU. Moreover, an Adam optimiser is used with an initial learning rate of $3e^{-4}$. The network is trained for 15 epochs (or until loss convergence), and a learning decay of 0.1 at epochs 12 and 14.

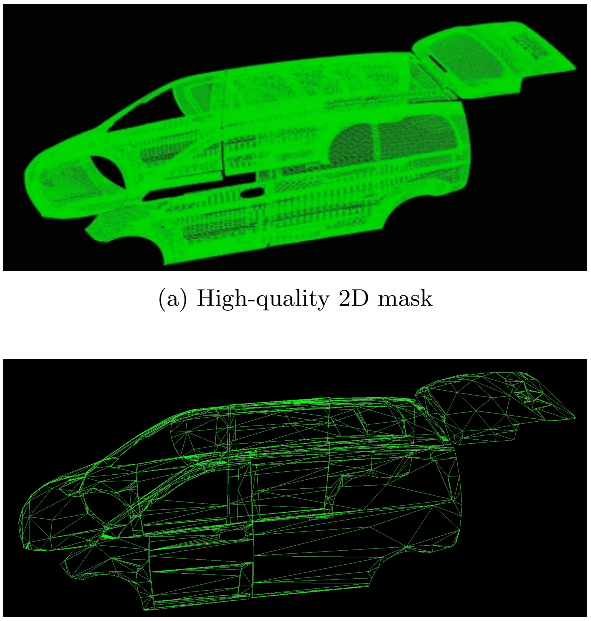
Inference To ensure that the network is evaluated on correctly detected vehicles, a bounding box IoU over 0.8 had to be achieved for the vehicle to be considered for pose estimation. The trained model attained is able to perform vehicle pose estimation in approximately 1 second per image. This makes the proposed approach suitable for applications in a factory setting, namely in the development of an autonomous system capable of conducting tasks with fast real-time constraints. Furthermore, improvements in efficiency are likely given enhanced hardware, specifically, more powerful GPUs.

4 Experimental Investigation

4.1 Data Collection

Given the CAD model of a vehicle (Mercedes Vito van), Blender's Python API is used to record annotations for each image in the dataset (image resolution, ground-truth pose, bounding box coordinates, 2D mask) (Fig. 6). Essentially, given a camera setup for an experiment (Section 4.2), a script is used to generate a dataset composed of images and their respective annotations. Furthermore, the dataset is sorted into training, validation and test datasets (80-10-10 split). For all cases analysed, the full dataset is comprised of 5000 images, each rendered using a monocular RGB camera at a resolution of 1230 x 3384.

2D Mask Given the camera intrinsic parameters, a prediction is visualised by transforming the 3D mesh of the vehicle by the estimated pose and projecting it onto the image, resulting in a 2D mask (Fig. 1). In keeping with simulating a real-world setup, a high-quality CAD model of the vehicle is used to render images. However, since the mask is merely used for visualisation purposes, the 3D mesh and CAD model are comprised of a different number of faces (Fig. 7). Based on the results, the combination of a high-quality image and low-quality mask was found to be the best compromise for accuracy and speed, subject to further findings.



(a) High-quality 2D mask

(b) Low-quality 2D mask

Figure 7: HQ vs. LQ 2D Mask

Ground-truth Pose The camera and vehicle model pose with respect to the world frame, ${}^{W}T_{cam,b}$ and ${}^{W}T_{car}$, respectively, are obtained from Blender and used to calculate the pose of the vehicle model with respect to the camera ${}^{cam,r}T_{car}$. Also, as the direction of the Y and Z axes is inverted in the Blender camera coordinate system, the homogeneous transformation ${}^{cam,b}T_{cam,r}$, is used to convert it to that of a real-world camera:

$$^{cam,r}T_{car} = (^{cam,b}T_{cam,r})^{-1} \cdot (^{W}T_{cam,b})^{-1} \cdot^{W} T_{car} \quad (3)$$

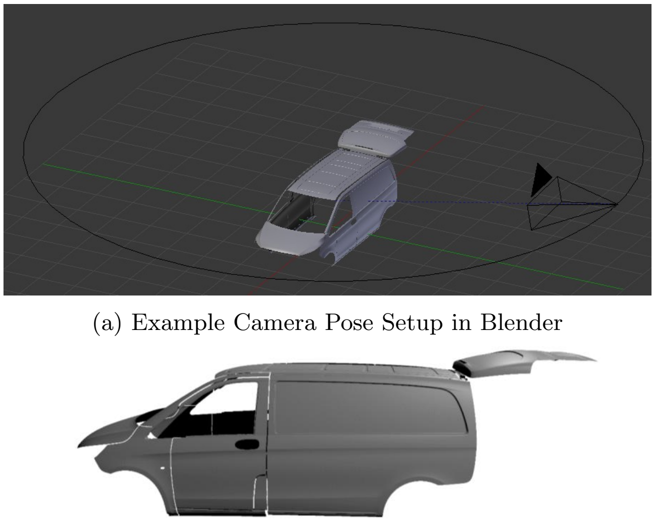
where ${}^{cam,b}T_{cam,r}$ is a 4 x 4 rotation matrix by π radians, about the X-axis.

4.2 Experiments

In the search of a robust configuration that would yield an accurate vehicle pose estimate, the following experiments were conducted on data obtained from a simulated environment in Blender:

- Multi-Scale 360° Dataset
- Single-Scale Dataset
- Distinct Scene Evaluation
- Various Distance Evaluation

Experiment 1 aimed to test the network's performance on a multi-scale dataset. The latter was constituted of multiple-scale images of a vehicle, obtained from a set



(a) Example Camera Pose Setup in Blender

(b) Rendered output of image (a)

Figure 8: Multi-Scale Dataset Setup

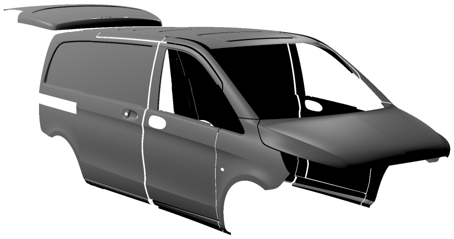
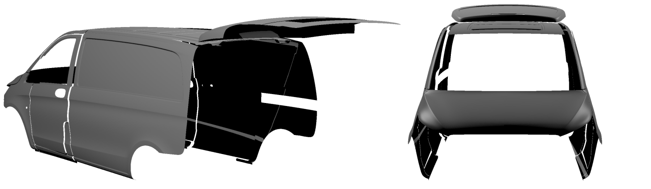


Figure 9: Front left view rendered image.

of random poses of the car with respect to the camera. Given a camera that is always pointing to the centre of a vehicle, images are rendered for one full revolution around the car (360°) before computing a new pose, resulting in 50 annotated images per set (Fig. 8). Furthermore, constraints are allocated to avoid obtaining images where the distance between the camera and the vehicle is too large.

Experiment 2 sought to emulate a scenario that better represents that of a car manufacturing setting, where the camera is fixed. It mainly focused on generating a dataset comprised of single-scale images rendered from a camera in a fixed scene: the front-left corner of the vehicle (Fig. 9). Each image is rendered after computing a unique combination of translation (along the X and Y axes, parallel to the ground plane) and rotation (about the Z-axis, perpendicular to the ground plane) from a given initial pose. To maintain the scene being considered (e.g. front-left corner), translation and rotation are limited to $\pm 5cm$ and $\pm 5°$, respectively.

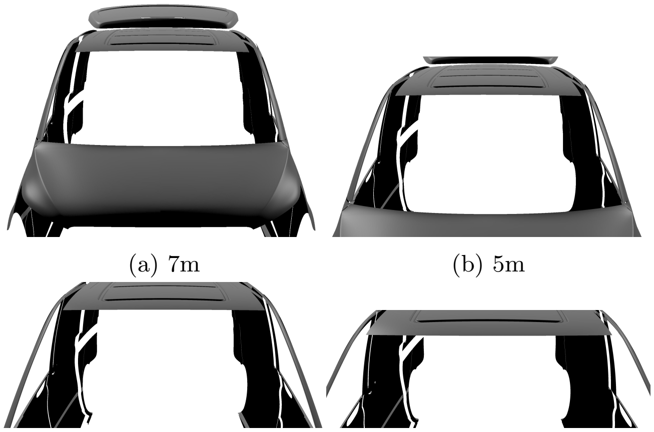
Experiment 3 assessed distinct camera scenes with the aim of determining the setup with the most robust model



(a) Back Left

(b) Front Centre

Figure 10: Two other distinct image scenes considered.



(a) 7m

(b) 5m

(c) 4m

(d) 3m

Figure 11: Rendered images at different camera distances.

(Fig. 10). Following the same procedure as in Experiment 2 for data collection, the network's performance is tested on the front left, back left and front centre camera scenes.

Experiment 4 set out to test the network's performance in a situation that is likely to occur in a factory setting; specifically, when parts of the vehicle are not within the camera's field of view. Several scenarios are evaluated, with the distance between the camera and vehicle decreasing from one setup to the next, i.e., gradually increasing the percentage of the vehicle being occluded (Fig. 11).

4.3 Evaluation Metrics

Wu et al. follow an instance mean AP similar to that of MS-COCO [Lin *et al.*, 2014], where they define a true positive prediction by satisfying the following three thresholds [Wu *et al.*, 2019]:

- Shape similarity
- Translation distance
- Rotation distance

However, as only one car model is considered for experimentation, the vehicle model metric, shape similarity, is eliminated. Instead, the network's performance is evaluated solely by computing the difference between the estimated and ground-truth poses.

Specifically, the error between the prediction and ground-truth is obtained by calculating the Euclidean distance (Eq. 4) for translation, and *arccos* distance (Eq. 5) for rotation.

$$\Delta(T_P, T_{GT}) = \sqrt{\sum(T_P - T_{GT})^2} \tag{4}$$

$$\Delta(R_P, R_{GT}) = \arccos\left(|q(R_P) \cdot q(R_{GT})|\right) \tag{5}$$

where q represents a quaternion rotation.

Accurate pose estimates are crucial for applications in a factory setting. The network's predictions are therefore, evaluated against a single threshold for each task, defined as:

- Translation distance threshold: $< 2cm$
- Rotation distance threshold: $< 5°$

For a prediction to be considered true positive, it had to satisfy both thresholds; otherwise, it would be set as a false positive prediction. Finally, the ratio of true positive estimations identified within the test dataset is calculated as:

$$mAP = \frac{\text{total number of true positive predictions}}{\text{total number of possible predictions}} \tag{6}$$

4.4 Results & Analysis

By analysing suitable configurations for camera positions, each experiment led to the desired outcome of uncovering a robust camera setup for pose estimation. Table 1 demonstrates the results of the experiments that took place with the aim of determining the optimal camera viewpoint for pose estimation. Specifically, experiments were evaluated based on average pose error measured over 500 vehicle pose cases during testing. The second column in the table describes the configuration that is assessed for a given experiment. Additionally, since the results from each test are used to gradually attain the optimal camera setup, the network is initially assessed against more flexible evaluation metrics before eventually zeroing in on the thresholds defined in Section 4.3.

Experiment 1 *Cam. Dist.* corresponds to the maximum possible absolute distance between the vehicle and the camera. The network performed better in terms of true positive predictions (mAP) for a maximal distance of $15m$; however, limiting the distance to $10m$ proved to be superior in computing a pose estimation, with regards to both translation and rotation accuracy. As this system primarily aims to find the optimal camera configuration for pose estimation accuracy, a camera-vehicle distance of $10m$ is used as the basis for Experiment 2.

Experiment 2 In comparing the results of a fixed camera single-scale dataset to that of a 360° multi-scale one, the former significantly outperforms the latter in both pose estimation and mAP, making it a suitable configuration with which to move forward.

Experiment 3 Given the network performance achieved in Experiment 2, this test evaluated the network with the metrics defined in Section 4.3. Table 1 shows the network achieving similar results in pose estimation for various camera viewpoints. Still, the Front-Centre configuration proved to achieve a higher percentage of true positive predictions, therefore making it a suitable setup to explore further.

Experiment 4 demonstrated the significant effect that occlusions had on the network's performance. While the prediction accuracy is similar across various distances, the network's performance in achieving true positive predictions is significantly higher at a distance of $5m$. Furthermore, the increase in translation error is the result of bounding box information being lost at closer distances. As mentioned in Section 3.1, the translation uses both feature extraction and bounding box information to estimate the translation of the vehicle; therefore, as the vehicle occlusion increases, the branch is forced to rely on the features extracted, hence, decreasing the network's prediction accuracy.

Based on the experimental results collected in Table 1, the scene setup that would yield the most accurate network outcomes was determined. A single-scale front centre scene at a camera distance of $5m$ was revealed as the optimal configuration for increased accuracy and mAP. In comparison, Wu et al. reported a mAP of 9.4% for their method. It is however important to note that they performed their experiments in substantially challenging conditions; pose estimation was conducted for multiple vehicles simultaneously in a cluttered city environment. Moreover, the task is performed using a camera mounted onto a moving vehicle.

Results from a number of comparable schemes are collected in Table 2, with the aim of validating the best performing CNN-based configuration of the proposed scheme against alternative methods of deriving the vehicle pose from single or multi view geometry schemes reported in the literature. These consist of the implementation of a representative two-staged iterative algorithm, "Ours(edge+ICP)", as described in Section 2.1, and tested on the same data. And also results reported from representative multi-stage CNN schemes described in Section 2.2. These include [Pavlakos *et al.*, 2017], which introduced a solution for 6-DoF pose estimation of multiple-class objects in cluttered backgrounds, whereby given a single RGB image, semantic keypoint predictions by a convolutional network and subsequent pose optimisation were proposed, impartial to an object's textural

Exp.	Test Type	Avg. Tx. Err.	Max/Min Tx. Err.	Avg. Rot. Err.	Max/Min Rot. Err.	mAP
		(cm)	(cm)	(°)	(°)	(% true positives)
1	Cam. Dist.(m)					
	20	12.7	19.9/2.1	0.473	2.899/0.019	49.4%
	15	10.1	20/1.4	0.468	1.852/0.003	77.8%
	10	7.0	19.9/0.2	0.304	1.31/0.004	54.2%
	Scene (at 10m)					
2	Front-Left(10m)	4.8	15.9/0.2	0.025	0.133/0.003	100%
3	Scene (at 10m)					
	Front-Left	1.2	2.0/0.2	0.018	0.041/0.003	15.8%
	Back-Left	1.2	2.0/0.3	0.282	0.414/0.167	12.8%
	Front-Centre	1.2	2.0/0.3	0.016	0.045/0.001	17.2%
4	Cam. Dist.(m)					
	7	1.2	2.0/0.1	0.016	0.087/0.001	46.4%
	5	**1.0**	2.0/0.1	**0.009**	0.059/0.001	**88.8**%
	4	1.2	2.0/0.3	0.008	0.035/0.001	12.4%
	3	1.5	2.0/0.9	0.009	0.021/0.003	1.8%

Table 1: Results of Experiments 1, 2, 3 & 4. Thresholds: Transl: $< 20cm$ (in gray) — Transl: $< 2cm$ (in white) — Rot: $< 5°$ (All exps.). Best combination in bold.

Method	Tx. Err. (cm)	Rot. Err. (°)
Single-Cam Kpt. Loc. [Pavlakos *et al.*, 2017]	27.57	5.57
Multi-Cam Kpt. Loc. [Ding *et al.*, 2018]	4.73	2.87
Ours (edge+ICP)	**0.05**	0.05
Ours (CNN)	1.0	**0.009**

Table 2: Comparison of the proposed approach to previous methods. Best pose estimation errors in bold.

Method	Inference	Num. Cams.
Ours (edge+ICP)	24s	3
Ours (CNN)	$\sim$ **1s**	**1**

Table 3: Required prediction duration: 2-stage iterative vs. CNN. Most efficient method in bold.

properties (textured or textureless). And [Ding *et al.*, 2018] who proposed a similar approach with multiple input images, focused solely on vehicle localisation, providing a solution aimed at modern surveillance camera network applications. The errors reported are extracted from their published works and highlight the shortcomings of keypoints in attaining accurate pose estimates. A comparison to the method in [Wu *et al.*, 2020] is not included in Table 2 as authors did not report pose errors on their dataset. It is shown how the performance of the proposed approach ranks substantially superior when compared to consonant methods represented by keypoint and optimisation-based alternatives, and is comparable to a large extent with the two-stage iterative scheme based on edge detection and ICP.

Finally, Table 3 highlights the notable improvement efficiency of the proposed approach, a stand-out feature of end-to-end schemes over iterative optimisation techniques, which firmly supports its feasibility for real-time applications in automotive manufacturing settings with very little additional infrastructure.

5 Conclusions

A method for vehicle pose estimation in a factory setting has been presented in this paper. An end-to-end cascade network approach tailored for an automotive manufacturing plant setting is proposed. A simplified loss function and an extensive evaluation of camera configurations was conducted to produce a robust method of estimating a vehicle's 3D translation and rotation from monocular RGB image inputs and reference CAD models. The system is shown to outperform comparable traditional visual 2D-3D optimisation methods for 6D localisation with a combined mean error of $1.0cm$ and $0.009°$ in translation and rotation respectively for over the 500 vehicle tests cases evaluated. Most notably, the computational advantages of the proposed scheme is proven an approach best suited for hard-bound real-time application in the automotive manufacturing sector when compared to more traditional localisation methods.

Data from a vehicle manufacturing plant is currently being collected to test the approach in a real setting, both in terms of direct learning from new real visual data collected at the plant, but also in relation to sim-to-real transfer learning.

Acknowledgments

The authors would like to extend their thanks to Brendan Emery from ESR Labs AG, Munich (Germany), and Ernesto Solanes and Josep Tornero from the Institute of Design and Manufacturing, Universidad Politecnica Valencia (Spain), for their past and present contributions to the project.

References

[Armesto *et al.*, 2011] Leopoldo Armesto, Josep Tornero, Alvaro Herraez, and Jose Asensio. Inspection system based on artificial vision for paint defects detection on cars bodies. In *2011 IEEE International Conference on Robotics and Automation*, pages 1–4. IEEE, 2011.

[Besl and McKay, 1992] Paul J Besl and Neil D McKay. Method for registration of 3-d shapes. In *Sensor fusion IV: control paradigms and data structures*, volume 1611, pages 586–606. International Society for Optics and Photonics, 1992.

[Censi, 2008] Andrea Censi. An icp variant using a point-to-line metric. In *2008 IEEE International Conference on Robotics and Automation*, pages 19–25. IEEE, 2008.

[Chen *et al.*, 2019a] Kai Chen, Jiangmiao Pang, Jiaqi Wang, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jianping Shi, Wanli Ouyang, et al. Hybrid task cascade for instance segmentation. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 4974–4983, 2019.

[Chen *et al.*, 2019b] Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jiarui Xu, et al. Mmdetection: Open mmlab detection toolbox and benchmark. *arXiv preprint arXiv:1906.07155*, 2019.

[Dharampal, 2015] Vikram Mutneja Dharampal. Methods of image edge detection: A review. *J Electr Electron Syst*, 4(2), 2015.

[Ding *et al.*, 2018] Wenhao Ding, Shuaijun Li, Guilin Zhang, Xiangyu Lei, and Huihuan Qian. Vehicle pose and shape estimation through multiple monocular vision. In *2018 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pages 709–715. IEEE, 2018.

[He *et al.*, 2017] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2961–2969, 2017.

[Irani and Anandan, 1999] Michal Irani and P Anandan. About direct methods. In *International Workshop on Vision Algorithms*, pages 267–277. Springer, 1999.

[Lin *et al.*, 2014] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European Conference on Computer Vision*, pages 740–755. Springer, 2014.

[Munoz *et al.*, 2019] Adolfo Munoz, Xavier Mahiques, J Ernesto Solanes, Ana Marti, Luis Gracia, and Josep Tornero. Mixed reality-based user interface for quality control inspection of car body surfaces. *Journal of Manufacturing Systems*, 53:75–92, 2019.

[Newell *et al.*, 2016] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *European Conference on Computer Vision*, pages 483–499. Springer, 2016.

[Pavlakos *et al.*, 2017] Georgios Pavlakos, Xiaowei Zhou, Aaron Chan, Konstantinos G Derpanis, and Kostas Daniilidis. 6-dof object pose from semantic keypoints. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2011–2018. IEEE, 2017.

[Solanes *et al.*, 2018] J Ernesto Solanes, Luis Gracia, Pau Muñoz-Benavent, Alicia Esparza, Jaime Valls Miro, and Josep Tornero. Adaptive robust control and admittance control for contact-driven robotic surface conditioning. *Robotics and Computer-Integrated Manufacturing*, 54:115–132, 2018.

[Sun *et al.*, 2019] Ke Sun, Yang Zhao, Borui Jiang, Tianheng Cheng, Bin Xiao, Dong Liu, Yadong Mu, Xinggang Wang, Wenyu Liu, and Jingdong Wang. High-resolution representations for labeling pixels and regions. *arXiv preprint arXiv:1904.04514*, 2019.

[Torr and Zisserman, 1999] Philip HS Torr and Andrew Zisserman. Feature based methods for structure and motion estimation. In *International workshop on vision algorithms*, pages 278–294. Springer, 1999.

[Wang *et al.*, 2020] Jingdong Wang, Ke Sun, Tianheng Cheng, Borui Jiang, Chaorui Deng, Yang Zhao, Dong Liu, Yadong Mu, Mingkui Tan, Xinggang Wang, et al. Deep high-resolution representation learning for visual recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[Wu *et al.*, 2019] Di Wu, Zhaoyong Zhuang, Canqun Xiang, Wenbin Zou, and Xia Li. 6d-vnet: End-to-end 6-dof vehicle pose estimation from monocular rgb images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019.

[Wu *et al.*, 2020] Di Wu, Yihao Chen, Xianbiao Qi, Yuyong Jian, Weixuan Chen, and Rong Xiao. Neural mesh refiner for 6-dof pose estimation. *arXiv preprint arXiv:2003.07561*, 2020.