Vehicle pose estimation via regression of semantic points of interest

1st Javier García López
FICOSA ADAS S.L.U
08232 Barcelona, Spain
jgarcia@iri.upc.edu

2nd Antonio Agudo
Institut de Robòtica i Informàtica Industrial
CSIC-UPC
08028, Barcelona, Spain
aagudo@iri.upc.edu

3rd Francesc Moreno-Noguer
Institut de Robòtica i Informàtica Industrial
CSIC-UPC
08028, Barcelona, Spain
fmoreno@iri.upc.edu

***Abstract*—In this paper we address the problem of extracting vehicle 3D pose from 2D RGB images. An accurate methodology is presented that is capable of locating 3D coordinates of 20 pre-defined semantic vehicle points of interest or *keypoints* from 2D information. The presented two-step pipeline provides a straightforward way of extracting three-dimensional information from planar images and avoiding also the usage of other sensor that would lead to a more expensive and hard to manage system. The main contribution of this work is the presented dedicated network architectures that are able to locate simultaneously occluded and visible semantic points of interest to convert these 2D points into 3D space in a simple but efficient way. The presented method uses a robust network based on Stack-Hourglass architecture for precise prediction of semantic 2D *keypoints* from vehicles even if they are occluded. Furthermore, in the second step another dedicated network converts the 2D points into 3D world coordinates and therefore, the 3D pose of the vehicle can be automatically extracted, outperforming state-of-the-art techniques in terms of accuracy.**

***Index Terms*—image processing, deep learning, 3D pose estimation**

I. Introduction

Obtaining accurate and trustworthy surrounding information is one of the biggest challenges of the autonomous driving. Being able to extract what surrounds a vehicle, its position, shape and relative velocity is very important to avoid obstacles or to follow a driving lane properly, e.g. For that reason, recent techniques like sensor fusion have provided accurate information extracted from several sensors mounted on the ego-vehicle, that collect as much data as possible. This techniques are able to provide accurate environmental data but they also have a limitation which is the required computational time to record, synchronize and process this data.

To avoid such limitations, over the years several image based techniques have been presented for extracting 3D pose by only using cameras. Furthermore, the appearance of deep learning techniques has meant a significant step forward in image processing, by being able to extract and learn simultaneously several features from the same image set, so that by designing and training properly a network, no need of other resources or processing steps are needed to extract the 3D pose.

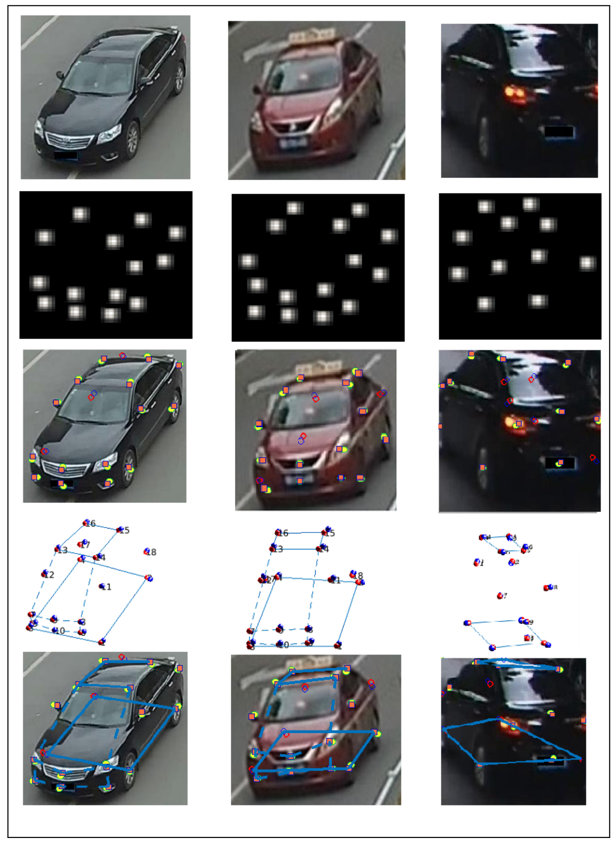

Fig. 1. Predicted 3D semantic *keypoints* from calculated 2D *keypoints* . For each column, the original vehicle image is shown first, followed by keypoint heatmaps, detected vehicle semantic *keypoints* on the original image (being the filled red circles de 2D ground truth, filled yellow circle the predicted keypoint (non-occluded), the red circumferences the ground truth for occluded keypoint and lastly the blue circumferences is the predicted occluded keypoint. Underneath this image is the predicted 3D model of the semantic keypoint and lastly is this model adjusted to the original image to compare the results.

II. Related Work

Pose estimation is a well studied problem and many approaches have been presented over the years as a solution to it. In this section, we motivate this work explaining the network architecture this work is based on and defining the challenge of the 3D pose calculation.

A. Stacked Hourglass Architecture

The **Stacked-Hourglass** method proposed by Newell *et al.* in [1] for human pose estimation implied a big step forward into robust and accurate 3D human pose extraction. This network is based on the successive steps of pooling and up-sampling that are done to produce a final set of predictions, as presented in Figure 1.

In the case of human pose estimation, extracting human joints and learning typical spacial relationships between them has proven to provide good results to the problem of human pose extraction. Research works like [2] cluster detections and predict the probable location of a neighboring joint [1]

The hourglass is a simple, minimal design that has the capacity to capture multiple features and bring them together to output pixel-wise predictions [1]. The aim of using this network architecture is to obtain full context information important for pose extraction, meaning that, not only the exact prediction of the position of the *keypoints* is important, but also the pose estimation requires a full understanding of the vehicle.

B. Pose estimation

In the last years, works like DeepPose [3] introduced a new approach to the typical ones for solving the problem of human pose extraction. This work presented a network to calculate the (x,y) coordinates of human joints. Further research like [4] proposed a similar idea as the one presented in this paper based on heatmap calculation around detected human joints.

In the field of vehicle pose extraction, research works like [5] or [6] cluster detections and predict the probable location of a neighboring joint. Then a comparison of multiple projected 3D models and the 2D contours calculated from the detected 2D points is performed, until a 3D model that matches the calculated 2D projection is found. This work uses the 3D projection compared with image contours to refine the pose estimated by discriminative part based model detector using the Pascal3D+ dataset [7].

Deep Manta bases the first step of its pipeline in FAST R-CNN [8] for a precise vehicle location. This is followed by a refinement step using Non-Maximum Suppression [9] algorithm and ending with a 2D-3D matching phase for comparing the extracted 2D vehicles and their 2D information (visible parts, part coordinates, ...) with multiple 3D models the extract the best 3D model that would fit into the information extracted from the first step of the pipeline.

Methods like the mentioned [5] have proofed to be very accurate and trustworthy although one of the biggest constraints of such approaches is the need of a big amount of training data (2D and 3D models) together with a multi-step pipeline that requires long training time. In this work, a fast, straightforward two-step pipeline that overcomes such limitations from other methodologies and also is able to perform a precise keypoint location from occluded points, which has been typically another important challenge in 3D pose extraction problems, is presented.

III. Proposed method

In this paper, we follow a similar approach than [10] by representing the 3D pose from a vehicle with N=20 keypoints and parameterized by a 3N vector P = $[p_1,..,p_N]$, where p_i is the 3D location of the i-th keypoint. Similarly, 2D poses are represented by 2N vectors U = $[u_1,..,u_N]$, where u_i are pixel coordinates. Our goal is then estimate the 3D pose vector y. This will be achieved by using a simple but effective network architecture, adding residual connections and using batch normalization, trained on vKITTI dataset [11] with labelled vehicle keypoint and taking into consideration the camera frame as global coordinate frame following the idea of [12] since this makes the 2D to 3D problem similar across different cameras.

In the presented work we use monocular input images with a resolution of 256x256 pixels that correspond to images in which only one vehicle is present. The vehicle in the image can be in multiple positions and with a hard, moderate or light occlusion. The training and validations image-sets correspond to both real and virtual images (virtual images extracted from the vKITTI dataset [11]). This method also predicts the probability of a keypoint of being occluded based on the notation of the training dataset. This probability is displayed together with the calculated heatmaps with Gaussians around predicted keypoints.

This network needs full input resolution of 256x256 and the highest resolution of the hourglass is 64x64. The full network starts with a 7x7 convolutional layer with stride of 2, followed by a residual module and a round of max pooling to bring the resolution down from 256 to 64 [1].

Given an input image, the network joint optimization minimizes the global function:

$$L = L_1 + L_2 \tag{1}$$

being L the global network loss function,$L_1$the loss function for the heatmaps prediction following the least squares method (Equation 2) and $L_2$the loss function for the occlusion prediction.

$$L_1 = \frac{1}{N}\sum_i (p_i - p'_i)^2 \tag{2}$$

$$L_2 = log(\frac{e^{y_i}}{\sum_i e^{y_i}}) \tag{3}$$

In the loss calculation, p_i is the predicted location for keypoint i, p'_i the keypoint location as ground truth for keypoint i and y_i is the i-th position of the output vector of the final FCN layer for the occlusion prediction in the forward pass.

A. Dataset generation

For the purpose of this research we have used a set of 43600 images of different vehicles in different positions in which 20 semantic points have been labeled, following the procedure defined by [13]. We have increased the training and validation dataset formed by real images with synthethic images of vehicles extracted from vKITTI dataset [11] and

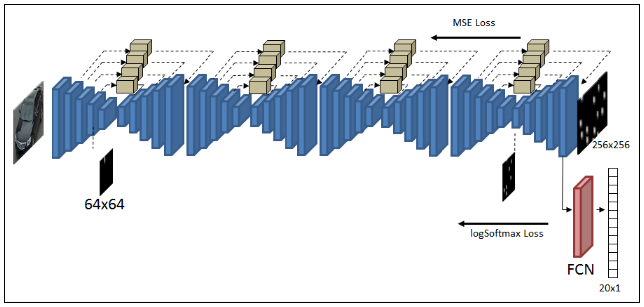

Fig. 2. Presented network architecture in this paper with 8 loops of down-sampling/up-sampling and parallel occlusion prediction. Heatmaps extracted from encoder/decoder architecture are followed MSE (Mean Squared Error) loss calculation and occlusion prediction is followed by a fully connected layer and posterior softmax loss calculation. A parallel residual block runs with the series of convolutional / deconvolutional for a more precise up-sampling of the image from 64x64 to input resolution (256x256).

labeled in a similar way as in 3. Also, the original VeRi Dataset was extended with the notation of the occluded *keypoints* in the images plus one binary term indicating if the keypoint is occluded or not. This labeling is necessary for the occlusion prediction explained in Section III-B.

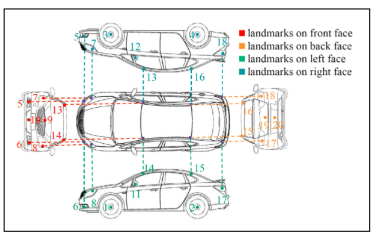

Fig. 3. Labeled semantic *keypoints* on the vehicles present in the training dataset [13].

As explained in section III together with the keypoint detection, our method is capable of predicting the probability of a keypoint to be occluded in the image. This is obtained by the labeling of the training and validation dataset, by defining the position of the *keypoints* followed by a binary term (0 or -1) indicating not occluded or occluded respectively.

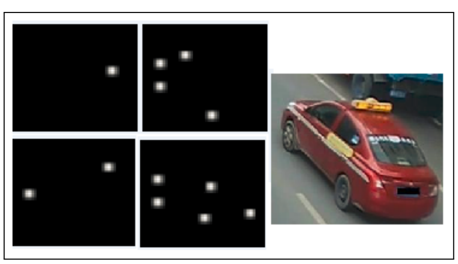

Fig. 4. Predicted heatmaps with Gaussian drawn around detected *keypoints* extracted during training and validation phase.

For the final step of the presented pipeline an image set based on virtual Kitti [11] has been created and self labeled in a similar way than the real dataset used for the keypoint detector. The training data for the 3D calculation network has been pre-processed (rotated and translated) to have all training data in a single global camera frame following the approach presented in [12].

B. *Vehicle keypoint localization*

Once the dataset has been generated, we follow the network architecture presented in [III] to train the model with 35000 images for predicting the semantic *keypoints* and the occlusions of these keypoints.For this purpose, six steps of encoding-decoding (7x7 Convolution followed by Batch-normalization and pooling layers together with their respective residual blocks as specified in [1]) as part of the applied Stacked-Hourglass architecture have been implemented. The training images will come into each processing step and the output of the processing will be input of the next loop of encoding-decoding. This mechanism allows the network to learn not only local but also global context of the extracted features, which is one of the main advantages of this network architecture.

These steps allow the network to obtain several features from the training data We will then obtain a set of vehicle part candidates Δ_J for multiple keypoints, where $\Delta_J = \{d_j : for j \in \{1...N_j\}\}$, with N_j the number of candidate keypoint and $d_j \in \Re^2$ is the location of the j-th detection candidate vehicle keypoint.

Fig. 5. Results of keypoint detection vs. ground truth labeling considering also self-occluded points in the vehicle. The yellow circles are the predictions and the red squares mark the ground-truth with non-occluded points. Blue circles are ground truth are re circles are predictions of keypoint position with self-occluded points (partially or totally occluded).

As detailed in the architecture, after the down-sampling/up-sampling phases of the Stacked-Hourglass we calculate the MSE (mean squared error) as loss function and feed it as input for the back-propagation phase.

In parallel to the keypoint extraction, we predict the occlusion of the detected kepoint by labeling it properly in the ground truth with the position of the vehicle keypoint plus one third number indicating if the point is occluded or not. As shown in Figure 2 we have added one output to the hourglass network for the occlusion prediction including a final

FCN (fully connected layer) plus a logarithmic softmax error calculation that will provide the probability for the point to be occluded. Therefore, this network is not only able to detect the 2D position of visible *keypoints* of the vehicle, but also their position when they are occluded (partially or totally) and their possibility of being occluded.

C. 3D vehicle pose calculation

Typically in human pose extraction there are several approaches that have proven good results like [14], [15], [16], [17], [18], or [19]. One of the main limitations of these proposed methods was the need of large training and validation datasets. For that reason, it seemed reasonable the appearance of new methodologies splitting the pose estimation in a two-step pipeline ([20], [21]) and due to the good results provided by these approaches for human-pose extraction problem, we applied a similar idea to the vehicle pose calculation issue.

In this proposed research, once the 2D *keypoints* have been precisely detected, we would need to convert them from 2D to 3D to obtain the 3D pose of the vehicle in the image. For that, this work proposes a network architecture formed by consecutive linear layer, batch normalization, RELU and pooling (Figure 6). This network is based on [12] which is meant for human pose estimation but adapted to the purpose of this research of obtaining the vehicle keypoint from an image with a single vehicle.

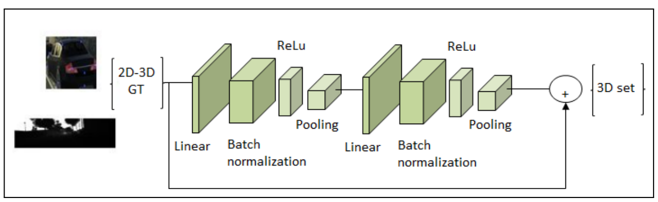

Fig. 6. Implemented network for converting 2D coordinates to 3D coordinates using vKITTI dataset [11] and using a network architecture based on [12].

For training this network we make use of images from the vKITTI dataset [11] due to its full labeling in 2D and 3D, tuned for this purpose. This dataset has depth annotated and the intrinsic and extrinsic parameters of the cameras are known. For the generation of the training dataset the virtual images had to be first cropped and labeled following the semantic *keypoints* explained in III-A using the proper coordinate system. We apply standard normalization to the 2D inputs and 3D outputs by subtracting the mean and dividing by the standard deviation. The results of the 3D model calculation are shown in Table II.

By training the explained network in Figure 6 we are able to convert the detected 2D vehicle *keypoints* to 3D coordinates, so that we can extract 3D information from planar images, which was the main goal of the research.

Following the approach proposed by [12] we have avoided the use of raw images for training the proposed network architecture and use 2D and 3D points labeled in the determined dataset. Although these contain less information as the image, using points we achieve bigger training speed. In the presented work we have trained this second network for 5000 epochs, obtaining a mean error of 43mm (measured on the image plane) between labeled 3D position and predicted one.

Once the vehicle *keypoints* have been converted to 3D coordinates, the pose will be extracted by calculating the direction-vector of the vehicle in VCS (Vehicle Coordinate System) as shown in Figure 7 and following the methodology for pose calculation proposed in [22], which origin is on the rear axle on the floor and in the middle point between the rear wheels. The direction-vector will be extracted starting on the origin of the VCS and pointing to the keypoint 9 (front side). Comparing the calculated pose with the labeled vehicle orientation from the vKITTI dataset the results of table II were extracted.

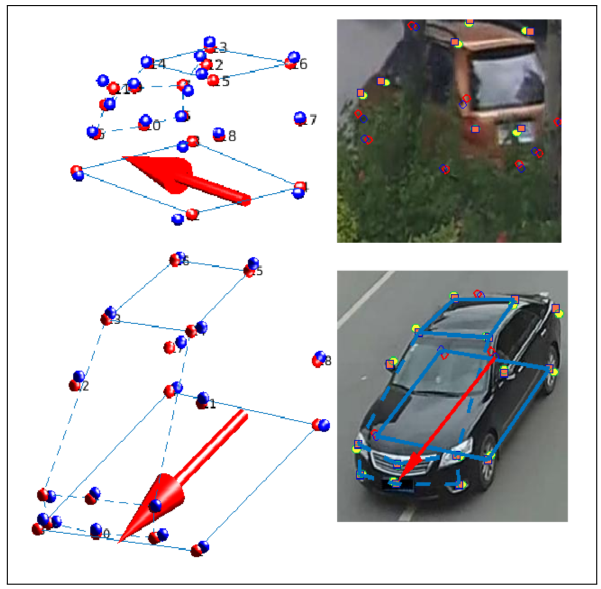

Fig. 7. Pose extraction on Vehicle Coordinate System. Red arrows represents the direction-vector from the orientation vehicle with origin in the center of the rear vehicle axle.

IV. Evaluation and Experimental Results

In this section we will evaluate the presented method and compare it with results obtained from other methods.

The results for the presented pipeline has been compared to other state-of-the-art methodologies for pose estimation. This comparison can be seen in table II. As proposed by the KITTI Benchmark [23] the used indicator for evaluating the calculated vehicle pose is the average orientation similarity (AOS).

The results shown in Table II demonstrate a good performance compared to other methods of the presented simple and straightforward pipeline in terms of orientation extraction for different occlusion possibilities.

The accuracy of the final pose reconstruction compared to other methods is shown in Table III.

Keypoint	PCK1 (%)	PCK2(%)
Front left wheel	92.26	65.12
Rear left wheel	90.11	62.45
Front right wheel	93.14	68.39
Rear right wheel	88.74	59.25
Front right anti-fog	86.89	66.27
Front left anti-fog	84.41	61.88
Front right light	72.74	61.49
Front left light	69.31	52.97
Front brand symbol	94.11	68.67
Front license plate	93.23	64.98
Left mirror	88.14	72.03
Right mirror	87.69	72.34
Front left roof corner	76.12	63.91
Front right roof corner	68.64	58.91
Rear left roof corner	70.45	63.27
Rear right roof corner	61.98	53.41
Rear left light	88.54	71.13
Rear right light	89.67	69.47
Rear brand symbol	89.14	69.91
Rear license plate	88.23	68.24

TABLE I: Table representing the accuracy of the semantic keypoint detection measured in PCK (percentahe of correct keypoint) with a threshold of 15 pixels (second column, PCK1). This has been evaluated over 12077 images of vehicles in different positions. Third column, PCK2, shows the performance of the keypoint detection only for points labeled as occluded (partially or totally).

Method	Hard	Moderate	Easy
Deep Manta [5]	80.55	89.91	96.32
3DVP [24]	65.38	75.77	87.46
SubCNN [25]	76.68	88.62	90.67
3DOP [26]	76.62	86.10	91.44
DPM [27]	46.54	61.84	72.28
OC-DPM [28]	52.40	64.42	73.50
AOG [29]	24.75	30.77	33.79
Mono3D [30]	76.84	86.62	91.01
Voxel [24]	78.29	65.73	54.67
Ours	79.25	86.11	92.47

TABLE II: Results for orientation extraction (AOS) on validation set. AOS is defined as ”average orientation similarity” ($AOS = \frac{1}{N}\sum s(r)$). s(r) is measuring what fraction of detected car orientations are similar to ground truth car orientations in the image. In order to make comparative results with other state-of-the-art methods, the test set of the KITTI Benchmark [31] was used. The labeling of this dataset has the occlusion distinguished between visible point (easy), partially occluded (moderate) and very occluded (hard).

Method	Rotation (Âº)	Translation (cm)
Viewpoints [32]	9.10	N/A
3DVP [24]	11.18	N/A
ObjProp3D [33]	17.37	N/A
Reconstruct [34]	12.57	N/A
Monocular [35]	2.87 / 4.4134	4.73 / 6.21
Ours	3.40	6.10

TABLE III: Results for pose extraction on test set of KITTI dataset evaluated as orientation and translation errors. These errors of the estimated pose with respect to the ground truths are expressed as geodesic distance for the rotation and distance between the centroids of two point sets for the translation error [35] respectively.

V. Conclusions

In this work a two-step pipeline for vehicle pose estimation has been presented. It has been proofed that good results can be achieved by using a dedicated network architecture and proper labeling. It is one of the motivations of this work to present a methodology that avoids usage of more sensors than cameras, which would lead to an expensive and hard to manage system. This shown methodology outperforms most of the state-of-the-art methods (see Table II and III) in terms of accuracy and performance.

Acknowledgment

This work was supported by the Catalan Government inside the program ”Doctorats Industrials” and by the company FICOSA ADAS S.L.U. J. García López is supported by the industrial doctorate of the AGAUR.

References

[1] A.Newell, K. ang and J. Deng, “Stacked hourglass networks for human pose estimation,” *European Computer Vision Conference (ECCV)*, 2016.

[2] Chen, X., Yuille, A., “Articulated pose estimation by a graphical model with image dependent pairwise relations.” *Advances in Neural Information Processing Systems (NIPS)*, 2014.

[3] Alexander Toshev, Christian Szegedy, “Deeppose:human pose estimation via deep neural networks,” *IEEE International Conference on Computer Vision*, 2014.

[4] Tompson, J., Goroshin, R., Jain, A., LeCun, Y., Bregler, C., “Efficient object localization using convolutional networks,” *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[5] Florian Chabot , Mohamed Chaouch , Jaonary Rabarisoa , Celine Teuliere, Thierry Chateau, “Deep manta: A coarse-to-fine many-task network for joint 2d and 3d vehicle,” *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[6] Zhu, Derpanis, Yang, Brahmbhatt, Zhang, Phillips, Lecce, and Daniilidis, “Single image 3d object detection

and pose estimation for grasping," *IEEE International Conference on Robotics and Automation (ICRA)*, 2014.

[7] R. M. Y. Xiang and S. Savarese, "Beyond pascal: A benchmark for 3d object detection in the wild," *IEEE Winter Conference on Applications of Computer Vision*, 2014.

[8] Kaiming He, Georgia Gkioxari, Piotr Dollár, Ross Girshick, "Mask r-cnn," 2017.

[9] Jan Hosang, Rodrigo Benenson, Bernt Schiele, "Learning non-maximum suppression," 2017.

[10] F. Moreno-Noguer, "3d human pose estimation from a single image via distance matrix regression," *Computer Vision and Pattern Recognition (CVPR)*, 2017.

[11] Adrien Gaidon, Qiao Wang, Yohann Cabon, Eleonora Vig, "Virtual world as proxy for multi-object tracking analysis," *Computer Vision and Pattern Recognition (CVPR)*, 2016.

[12] J. Martinez, R. Hossain, J. Romero, and J. J. Little, "A simple yet effective baseline for 3d human pose estimation," in *ICCV*, 2017.

[13] Liu X., Liu W., Mei T., Ma H., "A deep learning-based approach to progressive vehicle re-identification for urban surveillance," *European Conference for Computer Vision*, 2016.

[14] F. Zhang, X. Zhu, and M. Ye, "Fast human pose estimation," *CoRR*, vol. abs/1811.05419, 2018. [Online]. Available: http://arxiv.org/abs/1811.05419

[15] Wei, S.E., Ramakrishna, V., Kanade, T., Sheikh, Y, "Convolutional pose machines." *Computer Vision and Pattern Recognition (CVPR)*, 2016.

[16] Agarwal, A., Triggs, B., "3d human pose from silhouettes by relevance vector regression." *Computer Vision and Pattern Recognition (CVPR)*, 2004.

[17] G. Rogez, J. Rihan, S. Ramalingam, C. Orrite and P. H. S. Torr, "Randomized trees for human pose detection." *Computer Vision and Pattern Recognition (CVPR)*, 2008.

[18] C. Sminchisescu and A. D. Jepson, "Generative modeling for continuous non-linearly embedded visual inference," *International Conference on Machine Learning (ICML)*, 2004.

[19] P. . D. T. Shakhnarovich, Gregory Viola, "Fast pose estimation with parameter-sensitive hashing," *Computer Vision and Pattern Recognition (CVPR)*, 2004.

[20] E. Simo-Serra, A. Quattoni, C. Torras, and F. Moreno-Noguer, "A joint model for 2d and 3d pose estimation from a single image," *Computer Vision and Pattern Recognition (CVPR)*, 2013.

[21] E. Simo-Serra, A. Ramisa, G. Alenyà, C. Torras, and F. Moreno-Noguer, "Single image 3d human pose estimation from noisy observations," *Computer Vision and Pattern Recognition (CVPR)*, 2012.

[22] J. G. López, A. Agudo, and F. Moreno-Noguer, "Vehicle pose estimation using g-net: Multi-class localization and depth estimation," in *Artificial Intelligence Research and Development - Current Challenges, New Trends and Applications, CCIA 2018, 21st International Conference of the Catalan Association for Artificial Intelligence, Alt Empordà, Catalonia, Spain, 8-10th October 2018*. [Online]. Available: https://doi.org/10.3233/978-1-61499-918-8-355

[23] A. Geiger, P. Lenz, and R. Urtasun. A, "Are we ready for autonomous driving? the kitti vision benchmark suite." *Computer Vision and Pattern Recognition (CVPR)*, 2012.

[24] Y. Xiang, W. Choi, Y. Lin, and S. Savarese, "Data-driven 3d voxel patterns for object category recognition," *Computer Vision and Pattern Recognition (CVPR)*, 2015.

[25] ——, "Subcategory-aware convolutional neural networks for object proposals and detection," *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2017.

[26] X. Chen, K. Kundu, Y. Zhu, A. Berneshawi, H. Ma, S. Fidler and R. Urtasun, "3d object proposals for accurate object class detection," *Advances in Neural Information Processing Systems (NIPS)*, 2015.

[27] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2010.

[28] B. Pepik, M. Stark, P. Gehler, and B. Schiele, "Occlusion patterns for object class detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2013.

[29] B. Li, T. Wu, and S.-C. Zhu, "Integrating context and occlusion for car detection by hierarchical and-or model," *European Conference for Computer Vision (ECCV)*, 2014.

[30] X. Chen, K. Kundu, Z. Zhang, H. Ma, S. Fidler, and R. Urtasun, "Monocular 3d object detection for autonomous driving," *Computer Vision and Pattern Recognition (CVPR)*, 2016.

[31] Andreas Geiger, Philip Lenz, Christoph Stiller and Raquel Urtasun, "Vision meets robotics: The kitti dataset," *The International Journal of Robotics Research (IJRR)*, 2013.

[32] S. Tulsiani and J. Malik, "Viewpoints and keypoints," *IEEE Conf. on Comput. Vision and Pattern Recognition (CVPR)*, 2015.

[33] X. Chen, K. Kundu, Y. Zhu, A. G. Berneshawi, H. Ma, S. Fidler and R. Urtasun, "3d object proposals for accurate object class detection," *Advances in Neural Inform. Process. Syst. (NIPS)*, 2015.

[34] J. K. Murthy, G V. S. Krishna, F. Chhaya and K. M. Krishna, "Reconstructing vehicles from a single image: Shape priors for road scene understanding," *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2017.

[35] Wenhao Ding, Shuaijun Li, Guilin Zhang, Xiangyu Lei, Huihuan Qian, Yangsheng Xu, "Vehicle pose and shape estimation through multiple monocular vision," *International Conference on Intelligent Robots and Systems (IROS)*, 2018.