An Evaluation of Different Methods for 3D-Driver-Body-Pose Estimation

Manuel Martin[1], Michael Voit[1] and Rainer Stiefelhagen[2]

***Abstract*— Driver monitoring systems are increasingly introduced in modern commercial vehicles. Their importance will rise with automated vehicles, requiring the driver to pay attention or to take over in a timely manner. With the success of deep learning methods for human body pose estimation, these systems are also more and more employed in research projects for driver monitoring. However, their accuracy for driver body pose estimation is not yet evaluated thoroughly. We therefore annotate a part of the Drive&Act dataset [1] and evaluate both 2D- and 3D-body-pose performance based on triangulation and depth images. To this end we also introduce a deep learning based post processing step for depth image based 3D-pose-estimation that can be applied without much cost to the result of any 2D-pose detector, lifting the pose prediction to 3D. Our evaluation gives an overview of the performance of current state of the art methods and shows that our depth post processing method can close the gap to triangulation based methods using complex camera setups.**

I. Introduction

Distracted drivers are a major cause for traffic accidents even today. The EU is therefore planning guidelines that require driver monitoring system in all future vehicles [2]. Automated driving functions, that are more capable, will likely make this problem even worse because drivers are less and less occupied by the driving task. It is likely that they will start to occupy themselves with other tasks either self chosen, like using a smartphone, or involuntary, for example by falling asleep [3]. Current commercial systems for automated driving in SAE level 2 usually require the driver to keep the hands on the steering wheel. This is an easy measure to make sure that the driver pays attention and it prevents many activities that require both hands for a prolonged time. However, it greatly diminishes the advantages of automated driving functions and the comfort they could offer.

Because of these reasons there is an ongoing effort to improve driver monitoring systems. Currently, the focus of many of these systems is to detect distractedness or tiredness of the driver. This will likely be necessary to fulfill future legal requirements. A common approach uses cameras to monitor the driver's head. This way it is possible to infer the visual focus of attention [4] or drowsiness [5] of the driver. However, these approaches often disregard the rest of the driver's body. It is not clear if these systems are sufficient to let the driver keep his hands off the steering wheel. There are other approaches that focus less on the face of the driver and try to interpret the situation in the cabin as a whole. They detect which parts of the car the driver interacts with [6] or even detect the activities the driver is occupied with [1]. With the increasing success of human body pose estimation these methods are also more and more used to detect the upper body pose of the driver. Many approaches use just the 2D-body-pose, measured in pixels [7], [8], [9]. However, some use the 3D-body-pose, measured in meters [1], [10], [11]. The 3D-body-pose is harder to acquire but it allows to reason about distances [12]. However, as far as we know there is no public dataset to evaluate the accuracy of 2D- or 3D-body-pose estimation algorithms for driver body pose estimation. Our work tries to fill this gap.

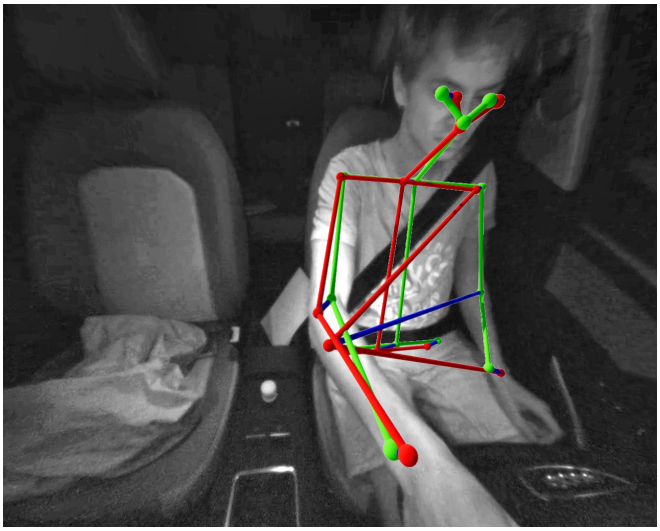

Fig. 1. Example image of our 3D-pose-estimation method using depth images. Input body pose (red) with wrong detection for the left elbow. Correction determined by our algorithm (blue) and final result (green).

We evaluate Openpose [13], a popular method for 2D-body-pose estimation, for driver body pose estimation. In addition, we also investigate different methods to determine the 3D-body-pose based on the 2D-body-pose results of Openpose. To this end, we compare different sensor setups for triangulation in a multi-view system with 3d-pose-estimation using a single depth camera. We can show that occlusions are especially challenging for depth camera based approach and present a small and fast neural network that increases robustness to occlusions significantly (see figure 1). To facilitate the evaluation we manually annotated a small part of the Drive&Act dataset with both the 2D- and 3D-upper body pose.

[1]Fraunhofer IOSB, Karlsruhe, Germany
manuel.martin@iosb.fraunhofer.de
[2]Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany
rainer.stiefelhagen@kit.de

The main contributions of our work are:

1) We manually annotate a subset of the Drive&Act dataset with 13 upper body keypoints on four views and create 3D-ground-truth in an iterative process.
2) We provide these annotations for future experiments on the website of the Drive&Act dataset[1].
3) We evaluate the popular Openpose approach on the dataset both for 2D-keypoint accuracy and with different camera setups for triangulation of 3D-human-body-keypoints.
4) We present a fast, neural network based, post processing method for 2D-human-body-keypoint estimators to compute 3d-poses using depth images.

II. Related Work

A. Datasets for Body Pose Estimation

The success of human body pose estimation algorithms is both rooted in advancements in deep learning algorithms but also in the public availability of increasingly complex datasets. The two major datasets that drove advancements in deep learning based 2D-body-pose estimation are the MPII dataset [14] followed later by the COCO keypoint dataset [15]. Both datasets use images mined from the Web that were manually annotated with keypoints. They depict real life images of all kinds of situations with multiple persons per image. As far as we know there are no datasets to train or evaluate 2D-pose-estimation algorithms for driver monitoring. The manual annotation of such images is a laborious process, however, it is possible to directly annotate what the algorithms should estimate later. This is not as easy for 3D-body-pose estimation methods because the depth information is lost when using common projective camera systems.

Generating accurate 3D-ground-truth data for body pose estimation is difficult. The best results are achieved with marker based motion-capture-systems. However, this negatively affects the realism of the resulting camera images because the data is often recorded in a static environment and test participants wear motion capture suits [16], [17]. Markerless motion capture systems are less accurate but alleviate the requirements on clothing [18] there are also methods to increase background variability using green screens [19]. To capture realistic camera data a small number of calibrated cameras in an unconstrained setting is suitable. Annotation is done by triangulating the manually annotated 2D-body-pose of all views. The increase in realism comes with the trade-off of decreased accuracy and the datasets are generally smaller because of the manual annotation process [20], [21]. For modern methods these datasets are often too small for training and are only used for evaluation.

There are only few and smaller datasets providing depth images and 3D-body-pose in good quality. They are usually not annotated manually. Instead they use for example markerless motion capture systems [18] or other depth image based approaches with manual postprocessing [22].

[1] www.driveandact.com

To our knowledge there are no publicly available datasets for driver 3D-pose-estimation. We provide a small manually annotated dataset for evaluation purposes that contains data for all three depicted classes of algorithms.

B. Human Body Pose Estimation

The topic of human-body-pose estimation can be divided into groups corresponding to the datasets.

The detection of the 2D-human-body-pose is often the basis for 3D-pose-estimation. Although these methods are often trained on specific datasets to get the best possible results, the detectors trained on the COCO datasets generalize well across many domains and use cases. Methods applied for driver pose estimation are also often trained on this dataset. Most state of the art detectors determine the location of each joint by estimating a confidence map per joint. To handle the detection of multiple persons there are mainly two approaches. Top-down methods first detect the bounding box of each person and then apply body-pose estimation to the cropped area [23], [24]. Bottom-up approaches detect all body parts within the image and solve the association problem by estimating additional tensors that help to group the body parts together [25], [13].

A good baseline for 3D-body-pose estimation with a multi-view system is 2D-body-pose estimation followed by the triangulation of each joint. Using a 3D pictorial structure model (3DPS) for the whole human body instead of triangulating each joint separately can improve results further [26]. However, this only works for a single person. In the case of multiple persons it is necessary to find the right association between detections in multiple views. This can be done for example with a reidentification network [26] or geometric constraints [27] and can also be integrated into 3D pictorial structure models [18]. There are also end-to-end trained methods inferring the 3D-body-pose of multiple people directly from the images [28]. They achieve impressive results. However, they require enough training data in the target domain.

With the introduction of the Microsoft Kinect, depth image based methods were the first 3D-human-pose-estimation systems viable for consumers. The approach relied on synthetic depth data and random decision forests [29]. There are multiple deep learning approaches increasing the quality even further but needing more computing resources [30]. A main challenge of depth image based methods are occlusions [22] because of the single view point of the sensor there is no depth data for occluded joints. Other methods go even beyond body pose estimation. Bashirov et al. [31], for example, fit a parametric body model estimating both body pose and body shape.

Estimating 3D-body-pose [32] and often also body shape [33] from monocular images is currently another popular research area. Often these approaches infer the parameters of a parametric body model. Although their accuracy is improving quickly they have to solve a generally ill posed problem which makes it difficult to determine the absolute

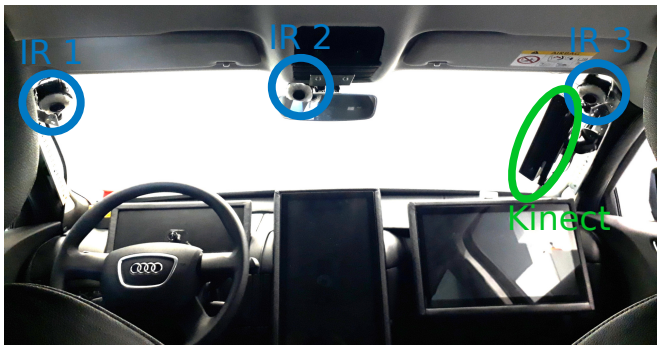

Fig. 2. The cameras used in the evaluation. Blue: NIR Cameras, Green: Kinect. The Kinect is also the origin for all evaluations.

position of body joints in the scene. These methods are therefore not the focus of our work.

Although there are many approaches relying on driver body pose estimation mostly using pretrained detectors from other domains [7], [8], [9], [1], [10], [11], there are few works on driver body pose estimation itself and its accuracy. Yuen et al. [34] evaluate 2D-hand and elbow estimation using the popular part affinity field based approach [25]. Martin et al. [6] present a method for depth based 3D-upper-body pose estimation using decision forests.

To our knowledge there are no publications investigating the accuracy of current state of the art approaches for 2D- or 3D-body-pose estimation for driver monitoring. Our work tries to fill this gap.

III. Annotating the Drive&Act Dataset with 3D-human-body-poses

The Drive&Act dataset offers video data of a calibrated multi-view system with five NIR cameras and a Kinect v2. In addition it provides the 3D-body-pose of the driver by triangulating Openpose [13] results. The main purpose of the dataset is fine grained driver activity recognition. Although Openpose combined with triangulation was used to generate the provided 3D-body-poses no evaluation of their accuracy was conducted. We therefore annotate a part of the dataset manually as a basis of our experiments.

The first step of generating the dataset for 3D-body-pose estimation is selecting suitable frames of the Drive&Act dataset. Picking frames at random would lead to many images depicting the same pose because in many sections of the video the test participants are not moving much. On the other hand for tasks like "picking up an object" there is rapid movement for a short time. We therefore use the already existing 3d-body-pose data as a starting point to select frames. We analyze each sequence of the dataset and collected all instances where the body pose differed by at least $10cm$ in at least one keypoint compared to all the data we collected before. This way all frames extracted from a sequence depict a unique pose.

After selecting unique 3d-poses we extracted the corresponding video frames of three NIR-cameras and the Kinect (see Figure 2).

Annotating this data with 3D-human-body-poses is challenging, as previously explained. Because our data consists of only few views with in part challenging occlusions, annotating the data manually was the only option.

Similar to [20], [21] we manually annotate the 2D-landmarks of the human body in each camera view and then use triangulation to generate the 3D-ground-truth. To further improve the quality of the annotation we reproject the 3D-ground-truth back to the image and manually fix any keypoints with a large reprojection error. Overall we annotate 13 upper body joints that are visible in all cameras most of the time (see figure 3).

We did not annotate all extracted unique poses. Because of our limited resources for annotation we selected 2000 unique 3D-body-poses of the test participants 11, 12, 13 and 14 of the Drive&Act dataset. This results in 6000 manually annotated 2D-poses, 1500 for each of the four views and 1500 triangulated and checked 3D-body poses. In addition, the dataset includes the 1500 corresponding depth images of the Kinect v2. The size of this dataset is comparable to the evaluation sets of similar manually annotated datasets. We specifically chose unique poses that cover many secondary activities. In addition, the movement options within cars are more limited compared to general purpose datasets. Our dataset should therefore be a good benchmark for driver-upper-body pose estimation both in 2D and 3D.

IV. 3D-Pose Estimation Methods

As presented in the related work section there are different ways to determine the 3D-body-pose. We focus on methods that can measure 3D-data based on their sensor system. Suitable methods therefore either rely on stereo- or multi-view camera systems that allow triangulation or rely on depth cameras, using for example the time of flight principle. The use of the Drive&Act dataset further restricts the evaluation to multi-view and depth camera based methods. All following 3D-body-pose methods rely on the 2D-body-pose estimation results of Openpose. We use the default network without further training but we also evaluate its accuracy for 2D-driver-pose estimation on our new dataset.

A. Multi-View Triangulation

The triangulation method functions mostly as baseline for the following depth image based methods. In addition, we evaluate different sensor setups both varying the number of used cameras, from 2 to 4, as well as varying the position of the used cameras (see Figure 2). We therefore chose a simple approach. Our method uses the 2D-pose-estimation results and triangulates each joint separately. To filter outliers we determine the average reprojection error of the whole body pose e_p and per joint e_j and remove any joint with an absolute error of $e_j > 20px$ or a relative error of $e_j > 5 * e_p$. Thresholds were intentionally set to large values to only remove results with large errors in 2D-body-pose estimation.

B. Depth Image Based

Similar to the presented triangulation approach the goal is to leverage the progress in 2D-body-pose estimation for

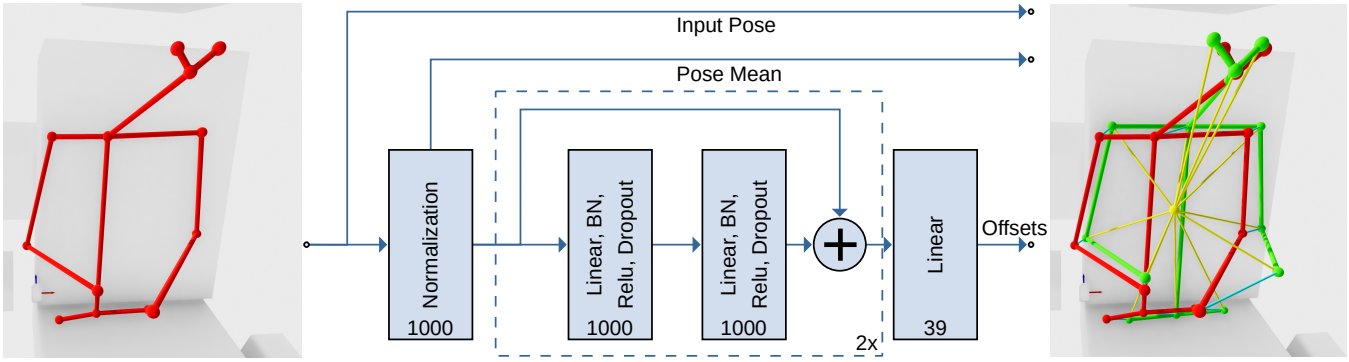

Fig. 3. Depiction of the depth fix method. It uses the 3D-body-pose of the direct method as input(red). The input gets normalized by subtracting the mean of the pose. The method produces correction offsets via a small feed forward network. We test two different offset methods. Offsets starting from the pose center, used for normalization, (yellow) and offsets starting from each joint of the input pose (turquoise).

depth image based 3D-body-pose estimation without retraining the 2D-pose detector. One way to achieve this is to detect the 2D-body-pose of the driver on the luminance image of a time of flight sensor and to use the matching depth image to determine 3D-coordinates for the 2D-detections. A direct approach would just look up the depth value of each 2D-body-joint (u, v) in the depth image d and then use the inverse camera matrix K to compute the 3D-joint J in camera coordinates:

$$J_{direct} = K^{-1} \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} d(u, v) \tag{1}$$

The main flaw of this approach is, that the depth image depicts the surface of the scene. In case of the driver this would be the clothing or the skin. Because the true body joints are within the body there is an offset in Z-direction. The multi-view approach on the other hand does not suffer from this problem because 2D-poses from different views show different sides of the body. Their intersection in world space, determined by triangulation, can describe the true position of the joint. We test the approach of Shotton et al. [29] who address this problem by learning offset vectors o_j for each joint on a training set $t \in T$:

$$J_{\text{offset}} = J_{\text{direct}} + o_j \quad \text{,with } o_j = \frac{1}{N} \sum_{t \in T} J_{gt} - J_{direct} \tag{2}$$

This already improves results as our evaluation shows. However, there are additional challenges. Fixed offsets do not account for different body shapes or sizes. Furthermore, the depth image does not contain valid z-values for occluded joints. This can lead to large errors (see figure 1).

Our approach therefore calculates offsets that adapt based on the input pose. The method is able to correct surface joint positions to the true joint positions within the body. In addition, it tries to fix any errors introduced by occlusions by regressing correcting offsets. This is handled by a small and very fast neural network that can be applied as a post processing step after generating the 3D-pose using the direct method (see equation 1). The idea is inspired by Moon et al. [35] who used a neural network as post processing to fix common errors of 2D-pose estimators like left-right swaps of limbs. It is also inspired by Martinez et al. [32]. They showed that the 3D-body-pose can be regressed from a single 2D-body-pose estimation in a post processing step using a small and fast neural network. Our architecture is very similar to their approach.

Figure 3 shows our network architecture. The basic building block is a linear layer followed by batch normalization, rectified linear units and dropout. The first block increases the feature dimensionality while all remaining blocks keep the dimensionality the same. The main network consists of groups of two basic building blocks and a skip connection. Our network consists of two main groups. The final layer of the network is a linear layer that regresses correction offsets.

The network uses the 3D-body-pose determined by the direct method as input. The offsets the network outputs are location independent. The location of the input pose is therefore normalized by subtracting the center of the input pose c_p computed as the mean of all vaild body joints.

We experiment with two different methods to generate offsets for corrected poses.

The *depth regression* method (see Figure 3 yellow) determines the output pose by adding the offsets to the body center c_p used for normalization. This forces the network to reconstruct the full body pose centered at the origin that is then moved to the correct location by adding the body center point c_p.

The *depth fix* method applies the offsets to the corresponding input body joints (see Figure 3 turquoise). This resembles the method described in equation 2 but with dynamic offsets that depend on the input pose. This approach can make better use of the input pose which is already quite accurate in many cases. Offsets are therefore small except for errors caused by occlusions.

To train the network we use the euclidean loss function. However, because both our input data as well as the labels can be incomplete we add additional masking to the loss function to not penalize the network in training.

V. Evaluation

A. Implementation Details

Openpose was not retrained. We used the default implementation with the default 25 keypoint model for 2D-pose estimation. However, we only use the 13 keypoints depicted in figure 3. Openpose is trained on color data and requires three channels as input. We therefore generate three channel images by replicated the grayscale data. Images of camera IR1,3 and the Kinect had to be rotated by 90° so the driver was upright in the image. Otherwise Openpose would not work. The images are unevenly lit. We tested different methods to adapt the brightness. The best results were achieved using adaptive histogram equalization (CLAHE) of opencv with a limit of 2.

The depth image based methods require training. Because of our limited annotated data we used the best results of the triangulation methods as ground truth and used all sequences not part of the annotated testset for training. This likely reduces performance because of the additional label noise. This approach results in a cross person evaluation of these methods. The neural networks are implemented in pytorch 1.4.0. We use the Adam optimizer with default parameters and train for 80 epochs using a batch size of 128. The learning rate is multiplied by 0.1 after 30 and 60 epochs. Participants 2 and 3 were used for validation, manually annotated test participants 11 to 14 for testing and the remaining participants for training.

B. Metrics

Object Keypoint Similarity (OKS) This is the main metric introduced by the COCO benchmark [15] for 2D-human-body pose evaluation. Its basis is the distance between predicted and Ground Truth joints. There are additional weights that model both the annotation error and a acceptable quality for each joint. The main metrics are the mean average precision (AP) over 10 OKS thresholds and the average recall (AR).

Mean per joint position error (MPJPE) This metric is widely used especially for 3D-body-pose evaluation. It measures the mean euclidean distance between measured and ground truth joints of the body pose. For multiple frames it is defined as the mean of the result for each pose. This metric is susceptible to missing joints. We therefore only take into account joints that are valid in measured and ground truth poses.

Percentage of Correct Keypoints (PCK) This metric considers joints correctly classified if the distance to the ground truth is within a threshold. We follow Mehta et al. [19] and determine the area under the curve (AUC) with a maximum threshold of $150mm$ (PCK_{1-15}). The advantage of this metric is that it penalizes missing joints only slightly and covers all performance aspects of the detectors.

C. 2D-Pose-Estimation Results

Table I shows the results for 2D-pose-estimation with Openpose. Note again, that the method was not retrained on any data of the interior of the car. Overall, the scores are comparable to the results on the MS Coco dataset. This is not expected because of the domain shift to NIR images and the differing environment. The reason could be that the driver fills most of the image so the detection itself is not as challenging. Nevertheless, there are differences in quality between the camera views. The results of the camera at the A-pillar of the driver side are worst. The reason might be that this camera is the closest and exhibits the most extreme view of the driver. There are also lots of self occlusions by the arms of the driver. The cameras on the Co-Driver side work best likely because their view is usually least obstructed. However, a Co-Driver would likely cause occlusions quite often. The results of the Kinect are by far the best. The cause for this is likely, that the field of view is not as big so some challenging poses with wide spread arms are not in frame. The Kinect also has the most even illumination and the best contrast of all the cameras in the dataset likely contributing to the resulting performance.

TABLE I
Results for 2D keypoint detection.

Configuration	$\mathbf{AP_{50-95}}$	AP_{50}	AP_{75}	$\mathbf{AR_{50-95}}$	AR_{50}	AR_{75}
IR 1	63.3	90	70.2	75	94.1	83.5
IR 2	65.5	88.9	74.5	77.9	94.3	86.3
IR 3	71.3	96.5	86	79	98.1	92.1
Kinect	**79.3**	**98.5**	**93.5**	**85.3**	**99.1**	**96.8**
Overall	68.9	92.6	79.9	79.3	96.4	89.7

D. 3D-Pose-Estimation Results

Table II shows the results of the 3D-body-pose estimation approaches. The results of the left side of the body are generally worse than the results of the right side of the body. The reason for this is worse visibility because of sensor placement issues and self occlusion. Except for camera IR1 all other sensors view the driver from the front or from the right side. Overall most multi-view setups work better than the depth image based methods.

In addition, using more cameras for triangulation improves results further. The driver fills a large part of the image in most cases. As already shown this makes the task of 2D-body-pose estimation less challenging compared to other datasets. However, the large size of the person also means that the estimated joints can move by a few pixels without necessarily being wrong because it is hard to locate the true joint position exactly. This can affect triangulation method negatively. The effect is bigger with smaller baselines as shown by the two-camera-setup *IR3 + Kinect*. Camera system *IR13* on the other hand has the widest baseline of the setups with two cameras. Its performance is comparable to camera setups with more than two cameras. Overall, triangulation with all four cameras results in the best performance according to the PCK metric.

The results of the depth image based methods follow the reasoning presented at the introduction of each method. The direct approach works worst having consistently larger errors than the method using fixed offsets. Applying the two deep

TABLE II

RESULTS FOR 3D KEYPOINT DETECTION.

Configuration	lEye	rEye	nose	lShoulder	rShoulder	neck	lElbow	rElbow	lWrist	rWrist	lHip	rHip	midHip	mpjpe [mm]	PCK_{1-15} [%]	availability [%]
							mpjpe [mm]									
IR123 + Kinect	9.4	12.2	15.6	30.0	25.5	51.7	49.6	25.8	45.2	28.7	88.2	**77.4**	72.1	43.4	**66.8**	88.9
IR123	**8.7**	11.9	15.7	**27.5**	**23.6**	48.5	**42.1**	**22.0**	38.6	**26.8**	86.3	78.2	**70.2**	**39.4**	62.6	82.7
IR23	16.5	15.7	21.2	53.4	35.5	63.4	84.9	27.4	72.6	32.5	128.5	92.3	100.6	58.5	58.8	88.6
IR12	9.5	**11.6**	**15.7**	28.2	44.3	**43.8**	**42.1**	33.2	**32.4**	34.3	105.3	121.9	98.4	50.8	58.0	81.1
IR13	12.8	12.8	17.8	33.2	29.2	45.7	42.2	35.8	34.5	30.9	**86.8**	84.5	75.5	43.9	61.5	84.5
IR3 + Kinect	31.3	27.8	32.0	142.0	74.7	86.7	200.7	67.7	129.6	61.5	178.5	159.9	144.3	105.5	47.9	**94.1**
Depth Direct	32.8	29.5	25.2	97.0	52.6	108.4	158.4	49.7	109.3	**36.1**	235.4	128.7	197.1	99.5	50.2	**97.0**
Depth Fixed Offset	22.5	**15.4**	**21.0**	78.2	35.7	78.2	133.5	39.5	109.8	40.2	117.5	94.8	88.6	68.9	61.8	**97.0**
Depth Regress Direct	20.8	20.9	22.6	45.8	36.0	61.2	69.4	48.1	82.1	54.5	83.2	80.1	70.6	55.9	63.8	**97.0**
Depth Fix	**18.2**	15.8	22.1	**44.2**	**31.7**	**58.9**	**64.6**	**40.9**	**75.7**	41.5	**81.5**	**80.0**	**69.0**	**51.9**	**66.5**	**97.0**

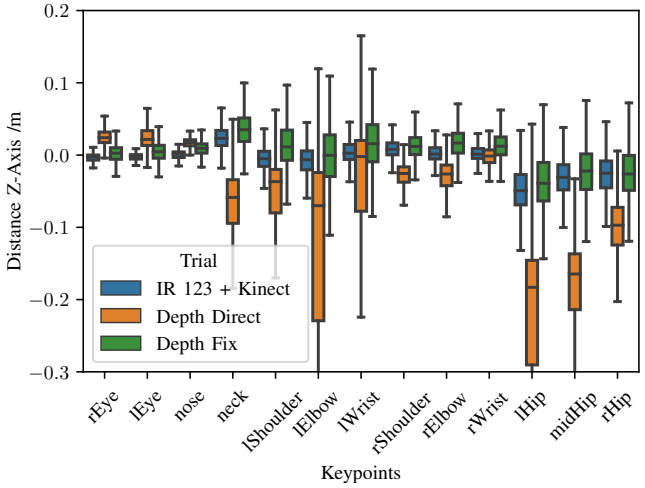

Fig. 4. Spread of errors along the Z-Axis highlighting the challenges of using depth images with occlusions.

learning based methods improves results further. As expected regressing the full pose based on the body center performs worse than regressing shorter offsets for each individual input joint with the depth fix method. However, compared to the *depth offset* method using fixed offsets the deep learning based methods improve results mostly on joints that are often occluded like the hips and the left side of the body. This highlights the effectiveness of these methods to deal with occlusions. The overall performance of the best depth method surpasses the triangulation approaches using just two cameras and reaches the quality of the best triangulation method according to the PCK metric. However, there are tradeoffs while the best triangulation method produces joints that are more accurate (mpjpe) the best depth method has higher availability and therefore detects overall more joints but with less precision.

Figure 4 further investigates the accuracy of the best triangulation method compared to the simplest and best depth image based method. The graph depicts the spread of errors along the Z-Axis. The evaluation is conducted in the camera coordinate system of the Kinect. The Z-Axis in this coordinate system is the axis that relies on the values in the depth image. It is affected the most by occlusions. Compared to the previous evaluation the triangulation method does not exhibit overly large errors on this axis. This indicates a more uniform spread of the error on all three axes. The depth methods on the other hand exhibit large errors especially on the left side of the body which is occluded more often. The deep learning based depth fix method is able to substantially reduce the errors to a similar size compared to the triangulation method.

E. Runtime Discussion

The following runtime analysis was conducted on a system with an AMD Ryzen Threadripper 1920X CPU and a nVidia 2080Ti graphics card using cuda 10.1 and cudnn 7.6. Openpose was used with default setting using the python wrapper. It is the basis of all presented methods. The runtime for a single frame was 28.3 ms. All other components of the system have negligible runtime. The triangulation takes about 1 ms and even the deep learning based methods for depth offset estimation take just 1.5ms.

The overall framerate of all methods is therefore depending on the speed of Openpose and the number of necessary cameras. The best depth image based method achieves 33fps and can utilize the full speed of the Kinect. A two camera triangulation based system achieves 17fps and the most accurate four camera system 8.7fps. All methods are therefore usable for many use cases needing soft real time.

There are other detectors running at higher frame rates but often with lower accuracy. All presented methods work with any generic 2D-pose detector and can make use of any advancements in the area. In addition our published dataset allows anyone to evaluate different setups to find the best compromise between speed and accuracy.

VI. CONCLUSION

We investigated the performance of state of the art detectors for 2D- and 3D-human-body-pose estimation methods for driver pose estimation. Because there was no suitable dataset we annotated part of the Drive&Act dataset as a test set and made it publicly available. We evaluated the popular Openpose method without retraining on this data and can show good performance despite the domain shift to NIR images. Our methods therefore build on top of Openpose or any other 2D-body-pose detector making full use of large datasets in other domains. We presented a method to triangulate the 3D-driver-body-pose based on two or more camera views and compare different setups and their advantages in our evaluation. We also investigated depth

image based 3D-driver-body-pose estimation. Our method lifts 2D-body-pose detections to 3D using the depth image. Our evaluation shows that this method can handle the common artifacts of this approach caused by occlusions closing the gap to triangulation based methods that require more complex camera setups. In addition, although our depth based method uses deep learning it uses negligible computing resources and runs at over 600 fps. It can therefore easily be combined with any 2D-human-body-pose detector without negatively affecting the frame rate of the whole system.

REFERENCES

[1] M. Martin, A. Roitberg, M. Haurilet, M. Horne, S. Reiss, M. Voit, and R. Stiefelhagen, "Drive&Act: A Multi-Modal Dataset for Fine-Grained Driver Behavior Recognition in Autonomous Vehicles," in *The IEEE International Conference on Computer Vision (ICCV)*, 2019.

[2] "Briefing: EU Mobility Package III including new vehicle safety standards — ETSC," https://etsc.eu/briefing-eu-mobility-package-iii-including-new-vehicle-safety-standards/.

[3] V. A. Banks, A. Eriksson, J. O'Donoghue, and N. A. Stanton, "Is partially automated driving a bad idea? Observations from an on-road study," *Applied Ergonomics*, vol. 68, 2018.

[4] S. Vora, A. Rangesh, and M. M. Trivedi, "Driver Gaze Zone Estimation Using Convolutional Neural Networks: A General Framework and Ablative Analysis," *IEEE Transactions on Intelligent Vehicles*, vol. 3, no. 3, pp. 254–265, 2018.

[5] C. Zhang, X. Wu, X. Zheng, and S. Yu, "Driver Drowsiness Detection Using Multi-Channel Second Order Blind Identifications," *IEEE Access*, vol. 7, pp. 11 829–11 843, 2019.

[6] M. Martin, S. Stuehmer, M. Voit, and R. Stiefelhagen, "Real time driver body pose estimation for novel assistance systems," in *International Conference on Intelligent Transportation Systems (ITSC)*, 2017.

[7] A. Behera, A. Keidel, and B. Debnath, "Context-driven Multi-stream LSTM (M-LSTM) for Recognizing Fine-Grained Activity of Drivers," in *Pattern Recognition*, ser. Lecture Notes in Computer Science, T. Brox, A. Bruhn, and M. Fritz, Eds. Cham: Springer International Publishing, 2019, pp. 298–314.

[8] A. Behera, Z. Wharton, A. Keidel, and B. Debnath, "Deep CNN, Body Pose and Body-Object Interaction Features for Drivers' Activity Monitoring," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–8, 2020.

[9] A. Behera and A. H. Keidel, "Latent Body-Pose guided DenseNet for Recognizing Driver's Fine-grained Secondary Activities," in *2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, 2018, pp. 1–6.

[10] P. N. Murthy, O. Kovalenko, A. Elhayek, C. Couto Gava, and D. Stricker, "3D human pose tracking inside car using single RGB spherical camera," in *ACM Chapters Computer Science in Cars Symposium CSCS 2017 —. ACM Chapters Computer Science in Cars Symposium (CSCS-17), July 6, Munich, Germany*. ACM, 2017.

[11] J. D. Ortega, N. Kose, P. Cañas, M.-A. Chao, A. Unnervik, M. Nieto, O. Otaegui, and L. Salgado, "DMD: A Large-Scale Multi-modal Driver Monitoring Dataset for Attention and Alertness Analysis," in *Computer Vision – ECCV 2020 Workshops*, ser. Lecture Notes in Computer Science, A. Bartoli and A. Fusiello, Eds. Cham: Springer International Publishing, 2020, pp. 387–405.

[12] M. Martin, M. Voit, and R. Stiefelhagen, "Dynamic Interaction Graphs for Driver Activity Recognition," in *International Conference on Intelligent Transportation Systems (ITSC)*, 2020.

[13] Z. Cao, G. Hidalgo, T. Simon, S.-E. Wei, and Y. Sheikh, "OpenPose: Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields," *arXiv:1812.08008 [cs]*, 2018.

[14] M. Andriluka, L. Pishchulin, P. Gehler, and B. Schiele, "2D Human Pose Estimation: New Benchmark and State of the Art Analysis," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 3686–3693.

[15] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common Objects in Context," in *Computer Vision – ECCV 2014*, ser. Lecture Notes in Computer Science, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds. Cham: Springer International Publishing, 2014, pp. 740–755.

[16] "HumanEva : Synchronized Video and Motion Capture Dataset and Baseline Algorithm for Evaluation of Articulated Human Motion — SpringerLink," https://link.springer.com/article/10.1007/s11263-009-0273-6.

[17] C. Ionescu, D. Papava, V. Olaru, and C. Sminchisescu, "Human3.6M: Large Scale Datasets and Predictive Methods for 3D Human Sensing in Natural Environments," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 7, pp. 1325–1339, 2014.

[18] H. Joo, T. Simon, X. Li, H. Liu, L. Tan, L. Gui, S. Banerjee, T. Godisart, B. Nabbe, I. Matthews, T. Kanade, S. Nobuhara, and Y. Sheikh, "Panoptic Studio: A Massively Multiview System for Social Interaction Capture," *arXiv:1612.03153 [cs]*, 2016.

[19] D. Mehta, H. Rhodin, D. Casas, P. Fua, O. Sotnychenko, W. Xu, and C. Theobalt, "Monocular 3D Human Pose Estimation in the Wild Using Improved CNN Supervision," in *2017 International Conference on 3D Vision (3DV)*, 2017, pp. 506–516.

[20] V. Belagiannis, S. Amin, M. Andriluka, B. Schiele, N. Navab, and S. Ilic, "3D Pictorial Structures for Multiple Human Pose Estimation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1669–1676.

[21] A. Elhayek, E. de Aguiar, A. Jain, J. Tompson, L. Pishchulin, M. Andriluka, C. Bregler, B. Schiele, and C. Theobalt, "Efficient ConvNet-Based Marker-Less Motion Capture in General Scenes With a Low Number of Cameras," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3810–3818.

[22] A. Haque, B. Peng, Z. Luo, A. Alahi, S. Yeung, and L. Fei-Fei, "Towards Viewpoint Invariant 3D Human Pose Estimation," *arXiv:1603.07076 [cs]*, 2016.

[23] K. He, G. Gkioxari, P. Dollar, and R. Girshick, "Mask R-CNN," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2961–2969.

[24] A. Newell, K. Yang, and J. Deng, "Stacked Hourglass Networks for Human Pose Estimation," in *Computer Vision – ECCV 2016*, ser. Lecture Notes in Computer Science, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds. Cham: Springer International Publishing, 2016, pp. 483–499.

[25] Z. Cao, T. Simon, S. Wei, and Y. Sheikh, "Realtime Multi-person 2D Pose Estimation Using Part Affinity Fields," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 1302–1310.

[26] J. Dong, W. Jiang, Q. Huang, H. Bao, and X. Zhou, "Fast and Robust Multi-Person 3D Pose Estimation From Multiple Views," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 7792–7801.

[27] A. Kadkhodamohammadi and N. Padoy, "A generalizable approach for multi-view 3D human pose regression," *Machine Vision and Applications*, vol. 32, no. 1, p. 6, 2020.

[28] H. Tu, C. Wang, and W. Zeng, "VoxelPose: Towards Multi-Camera 3D Human Pose Estimation in Wild Environment," *arXiv:2004.06239 [cs]*, 2020.

[29] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake, "Real-time human pose recognition in parts from single depth images," in *CVPR 2011*, 2011, pp. 1297–1304.

[30] G. Moon, J. Y. Chang, and K. M. Lee, "V2V-PoseNet: Voxel-to-Voxel Prediction Network for Accurate 3D Hand and Human Pose Estimation From a Single Depth Map," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 5079–5088.

[31] R. Bashirov, A. Ianina, K. Iskakov, Y. Kononenko, V. Strizhkova, V. Lempitsky, and A. Vakhitov, "Real-Time RGBD-Based Extended Body Pose Estimation," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2021, pp. 2807–2816.

[32] J. Martinez, R. Hossain, J. Romero, and J. J. Little, "A Simple yet Effective Baseline for 3D Human Pose Estimation," in *International Conference on Computer Vision (ICCV)*, 2017, pp. 2640–2649.

[33] D. Mehta, O. Sotnychenko, F. Mueller, W. Xu, M. Elgharib, P. Fua, H.-P. Seidel, H. Rhodin, G. Pons-Moll, and C. Theobalt, "XNect: Real-time Multi-Person 3D Motion Capture with a Single RGB Camera," in *ACM Transactions on Graphics (SIGGRAPH)*, 2020.

[34] K. Yuen and M. M. Trivedi, "Looking at Hands in Autonomous Vehicles: A ConvNet Approach using Part Affinity Fields," *IEEE Transactions on Intelligent Vehicles*, pp. 1–1, 2019.

[35] G. Moon, J. Y. Chang, and K. M. Lee, "PoseFix: Model-agnostic General Human Pose Refinement Network," *arXiv:1812.03595 [cs]*, 2019.