Joint Human Pose Estimation and Stereo 3D Localization

Wenlong Deng, Lorenzo Bertoni, Sven Kreiss, Alexandre Alahi

EPFL, Switzerland

{firstname.lastname}@epfl.ch

***Abstract*— We present an end-to-end trainable Neural Network architecture for stereo imaging that jointly locates and estimates human body poses in 3D. Our method defines a 2D pose for each human in a stereo pair of images and uses a correlation layer with a composite field to associate each left-right pair of joints. In absence of a stereo pose dataset, we show that we can train our method with synthetic data only and test it on real-world images (*i.e.*, our training stage is domain invariant). Our method is particularly suitable for autonomous vehicles. We achieve state-of-the-art results for the 3D localization task on the challenging real-world KITTI dataset while running four times faster.**

I. Introduction

The perception stack of autonomous vehicles commonly relies on expensive 3D sensors (*e.g.*, LiDAR) [1], [2]. Stereo vision, a cost-effective alternative, is still below the detection accuracy of LiDAR-based solutions. State-of-the-art stereo-based methods only focus on the *car* category [3], [4], due to the large number of available instances needed for data-hungry deep-learning networks. In this work, we are interested in perceiving *humans* - a fundamental and critical category for any autonomous vehicle operating alongside pedestrians (from social robots [5] to self-driving cars [6]). Note that our definition of *human* generalizes to *pedestrians* and any other category involving humans in the publicly available KITTI dataset [7], such as *person sitting*.

We frame the problem as follows: given a pair of images from stereo imaging, estimate human body poses and locate them in 3D (see Figure 1). Calculating the pixel disparity between humans in a pair of stereo images requires accurate correspondence between pixels or stable keypoints [8]. We propose to use the 17 semantically defined keypoints corresponding to body parts [9]. We address the challenges related to keypoints' stability across image pairs, limited resolution for far-away humans, and occluded body joints in crowds.

Inspired by the recent success of pose estimation [9], [10] and end-to-end object tracking [11], [12] methods, we propose to jointly solve pose estimation and stereo matching with a single feed-forward regression network. To address challenges related to keypoints' stability and limited resolution, we develop our end-to-end method, referred to as Part Spatial Field, which combines composite fields [9] and correlation layers [12]. Furthermore, we address the challenge of occluded joints with an uncertainty-based score and a stereo joint voting procedure. Our method reasons in 3D, creating location proposals in the form of 3D human poses.

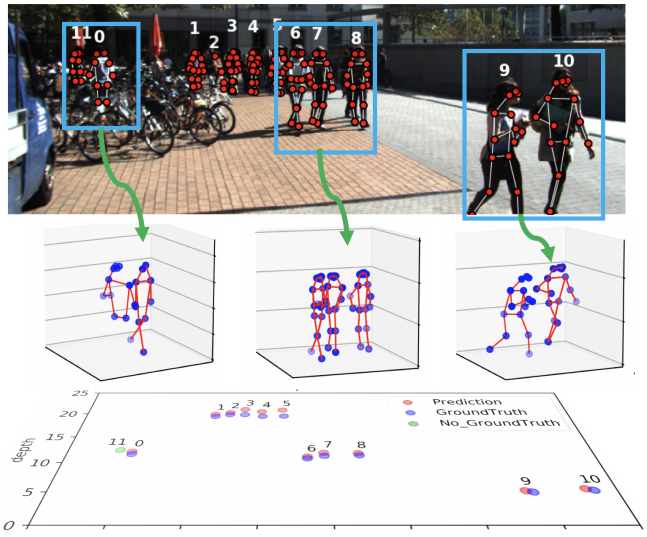

Fig. 1. We present a method for stereo imaging that jointly estimates human body poses (second row) and locate them in 3D (last row). For clarity, only one image is shown (in the first row) from the stereo pair.

Finally, in the absence of a stereo pose dataset, we propose to synthesize one by generating the horizontal shifts of COCO [13] images - a dataset made of randomly collected images annotated with body parts keypoints. Domain invariance between this synthetic stereo and the real stereo is provided by working at the feature level that is used for single-image keypoint estimation.

We summarize our main contributions as follows:

- A new Part Spatial Field that forms an association field from correlation features, which makes stereo matching trivial.
- An uncertainty based decoding procedure that detects occluded joints and matches stereo poses.
- A domain invariant strategy to train our network.

The rest of the paper is structured as follows: after briefly presenting previous works, we present our method in Section III and IV. Then, we run experiments on the KITTI dataset [7] outperforming the accuracy of previous works while running four times faster.

II. Related Work

A. Stereo 3D Object Detection

Researchers in deep learning have paid less attention to stereo-based 3D object detection compared to LiDAR-

based methods [14], [15] on the popular KITTI dataset [7]. 3DOP [16] focuses on generating 3D proposals by encoding an object size prior, a ground-plane prior, and depth information (*e.g.,* free space, point cloud density) into an energy function. 3D proposals are then used to regress the object pose and 2D boxes using the R-CNN approach. Li *et al.* [17] leverage geometrical constraints for localization by extending the Structure from Motion (SFM) approach to the dynamic object case and used the ego-camera pose to fuse both spatial and temporal information. The very recent Stereo R-CNN [3] takes advantage of dense object constraints in raw stereo images and detects matching objects on stereo images at the same time. Wang *et al.* [18] exploit a dense disparity map to create point clouds, while TLNet [4] leverages 3D anchors to explicitly construct object-level correspondences between the regions of interest in stereo images. Yet, all these methods primarily focus on detecting vehicles.

B. Bottom-up 2D Pose Detection

Methods to detect human 2D poses are categorized as either top-down or bottom-up. The latter detects each joint of a person first and then connect them to construct the full pose. The pioneering work by Pishchulin with Deepcut [19] and Insafutdinov with Deepercut [20] solves the part association with an integer linear program but requires high computation complexity. Follow-up methods successfully reduce inference time by using greedy decoders in combination with additional tools as in Part Affinity Fields [21], PersonLab [10], and PifPaf [9]. Other intermediate representations are built on top of 2D pose estimates in the image plane, including 3D pose estimates [22], human pose estimation in videos [23], and dense pose estimation [24]. All of them would profit from improved 2D pose estimates. Our network architecture is built on top of PifPaf [9], which is particularly suitable for low-resolution images.

C. Monocular-based 3D Object Detection

Chen *et al.* [25] use monocular images to extract ground-plane assumption, shape prior, contextual feature, instance segmentation, and predict 3D objects. However, they do not explicitly evaluate their methods for the "human" category. They also assume a fixed ground plane orthogonal to the camera and the proposals. To regress 3D pose parameters from 2D detection, Deep 3D Box [26] and Xu *et al.* [27] propose an end-to-end multilevel fusion approach to detect 3D objects by concatenating the RGB image and the monocular-generated depth map. the MonoPSR [28] method predicts a point cloud to learn shape information. Finally, the recent MonoLoco [29] method learns from the data the relationship between human body poses and their distance to the camera (depth). Since the method outperforms the stereo 3DOP [16] in some conditions, we also compare our method to the state-of-the-art monocular method in Section V.

III. Method

The goal of our method is to jointly estimate and match human poses in a pair of stereo images. Then, we use the pixel disparity to estimate the depth and localize humans in 3D. We address challenges related to faraway and partially occluded humans. We propose a bottom-up method that increases the association resolution from person scale to joint scale and jointly exploits textural and pose similarity. Figure 2 presents our overall model. It includes a shared ResNet [30] base network and PifPaf [9] head networks to predict the 2D poses and a new third head network for predicting association between stereo joints. We name our method Part Spatial Field (PSF) and can train the whole network end-to-end.

A. Correlation Calculation

Our goal is to detect and match across pair of stereo images multiple people at the same time. We compute correlation values for all positions in a feature map and make our model operate on the whole feature maps for matching regression. However, calculating all possible circular shifts will lead to a huge output dimensionality. Yet, the stereo images do not require a large disparity. Hence, we restrict the correlation calculation to small translations. Our correlation module is inspired by Flownet [31], where a correlation layer is aimed to help a convolutional network in matching feature points between stereo images. The correlation layer operates pixel-level feature comparison of two feature maps x_l, x_r:

$$x^{l,r}_{corr}(i,j,p,q) = \langle x_l(i,j), x_r(i+p, j+q) \rangle, \qquad (1)$$

where $-K \leq p \leq K$ and $K \leq q \leq K$ are offsets to compare features in the square neighborhood around the locations i, j in the feature map, defined by the maximum displacement K. The correlation layer output becomes $x_{corr} \in R^{H_l \times W_l \times (2K+1) \times (2K+1)}$. In other words, Equation 1 can be seen as a correlation between two feature maps within a local square window defined by K. We compute this local correlation for left regression and right regression.

B. Part Spatial Field

Our Part Spatial Field (PSF) module will output one intensity map s_c to model the confidence of association and two regression maps $\langle s_{xl(r)}, s_{yl(r)} \rangle$ to convert the similarity into a pixel-level distance. As shown in Figure 3, at every output location, two vectors will point to its left and right stereo joints, respectively.

Stereo Pose Matching algorithms need to consider the diversity of scales that a human pose can have in an image. While a localization error for the joint of a close proximity person can be minor, that same absolute error might be a major mistake for faraway smaller persons. At the same time, measuring the uncertainty of the spatial precision of an association could be helpful when computing a score for each connection. As a result, we use a Laplace loss [32] to train the regressive model:

$$L = |x - u|_1 / b + \log(2b), \qquad (2)$$

where u is the ground truth location of the joints, and b the predicted spread which attenuates the radius of the L_1 loss.

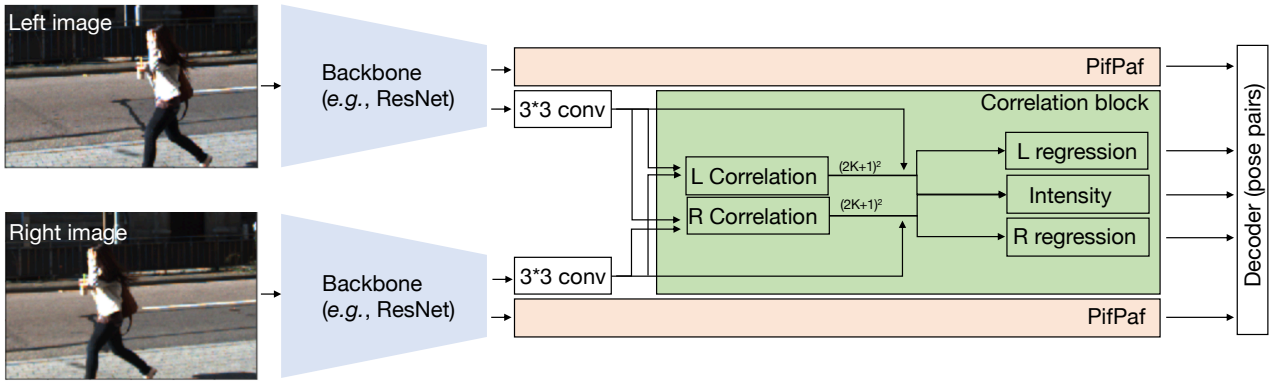

Fig. 2. The architecture of our proposed stereo association method. The Backbone processes images into features. The correlation block and PifPaf [9] head networks receives down-sampled features and outputs Part Spatial Fields (PSF) and 2D regression fields, respectively. The Decoder converts the fields into final stereo pose pairs through greedy decoding.

It represents how hard (or uncertain) the task is. Finally, PSF can be represented as:

$$\overrightarrow{s} = \{s_c^{i,j}, s_{xl}^{i,j}, s_{yl}^{i,j}, s_{bl}^{i,j}, s_{xr}^{i,j}, s_{yr}^{i,j}, s_{br}^{i,j}\}, \tag{3}$$

where (s_{xl}, s_{yl}) and (s_{xr}, s_{yr}) are absolute locations (sum of pixel location and regression distance) of an association vector's two endpoints. (s_c, s_{bl}, s_{br}) represent the confidences for the association and left and right spatial precision, respectively. The output is then decoded to associate each joint in the left image with the one in the right image.

C. Uncertainty-based Pose Matching

Given the 2D pose outputs, the task is to pair the same persons stereo poses. In addition to the confidence map produced by PSF, the spreads b will give the uncertainty of an association. If two vectors' edge locate faraway from the joint, the association score should be low. At the same time, based on the uncertainty of the spatial precision b, a short connection may still be suppressed. As a result, the connection score is designed by the following formula:

$$\begin{aligned} c(\overrightarrow{s}, (x_l, y_l), (x_r, y_r)) = s_c \times \\ \exp\left(-\frac{\sqrt{(x_l - s_{xl})^2 + w(y_l - s_{yl})^2}}{b_l}\right) \times \\ \exp\left(-\frac{\sqrt{(x_r - s_{xr})^2 + w(y_r - s_{yr})^2}}{b_r}\right), \end{aligned} \tag{4}$$

where the connection vectors $\overrightarrow{s}$ associate two joints together, (x_l, y_l) and (x_r, y_r) correspond to joints location on left and right stereo images. Again, (s_{xl}, s_{yl}) and (s_{xr}, s_{yr}) are absolute locations (sum of pixel location and regression distance) of an association vector's two endpoints. The weight w emphasizes that the paired joints should have a small difference in y coordinate. b_l and b_r are the two predicted spread, which indicates the uncertainty of the connection. With the help of the connection score, a stereo matching algorithm is designed to associate stereo poses with high location similarity. The initialization of connection vectors is done by filtering the confidence maps with a manual threshold. To further reduce the computation complexity, we only take keypoints located to the right of the left keypoint. Finally, Non Maximum Suppression (NMS) is used to remove duplicates. The algorithm encourages keypoint pairs to have high location similarity on all joints, and the pose similarity is thus taken into consideration. The algorithm is summarized as follows.

Algorithm 1: Uncertainty based Stereo Pose Matching

Result: Stereo Pose Pairs

Input: connection vectors $\overrightarrow{s}$ and predicted 2D pose sets P_l and P_r ;

while $P_l! = null$ **do**

  $P_l^t \leftarrow P_l.pop()$;

  $< P_l^t, P_r^i > \leftarrow \arg\max_i($
  $\sum_j^{joints} max(c(\overrightarrow{s_{jn}}, (x_{lj}, y_{lj}), (x_{rj}^i, y_{rj}^i)...)),$
  where $n \in N$, N is a circle centered at (x_{lj}, y_{lj})
  with radius k. $i \in P_r$;

end

Non Maximum Suppression(NMS)

D. Human Distance Estimation

Given the depth of each joint, the task is to calculate the distance of the person to the camera (also referred as the depth). Since some joint locations are not accurate due to occlusion or detection error, we include a z-score thresholding procedure to remove outliers and consider the median depth of the remaining joints as our final output. At last, we obtain the distance by calculating the L2 norm of pose center (camera coordinate) and depth.

To determine the z-score threshold z, we evaluate the model on the KITTI training dataset without any z-score threshold. The fraction of distance errors that are less than 2 meters determines our z-score threshold z.

E. Stereo Joint Voting

For the pose estimation phase, the confidence of a joint is given by PifPaf [9]. However, in stereo settings, occluded

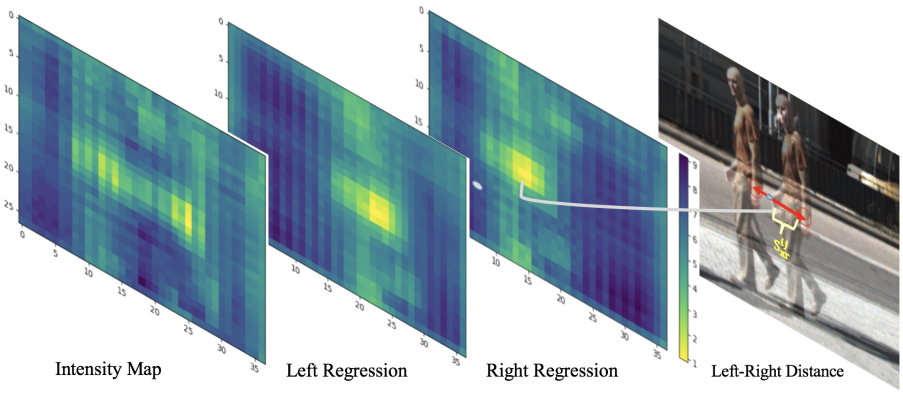

Fig. 3. The Part Spatial Field (PSF) output maps for the "right wrist". The intensity map gives the route of the joint movement. The left and right regression maps measure the down-scaled distance from each pixel location to left and right stereo joints. From the regression map, we generate dense association vectors to associate stereo joints.

Fig. 4. The figure illustrates voting's ability to detect occluded joints. To simulate the occlusion situation, we use a black mask to mask left shoulder, left elbow and left hip. The Red, Green, Blue are the voting result for the right stereo image, which indicates the ability of the model to associate each joint. Left: Right stereo image. Right: Masked left image with PSF voting.

joints are common and not well detected by 2D pose estimation algorithms [23], [33], [34]. Hence, we combine the PifPaf confidence with a confidence specifically designed for occluded joints. We create two high-resolution part confidence maps $f_l(x,y), f_r(x,y)$ with a convolution of a Laplacian kernel N with width $s^{ij}_{bl(r)}$ over the regressed targets from the Part Spatial Field weighted by its confidence s_c:

$$f(x,y) = \sum_{ij} s^{ij}_c \; N\left(x, y | s^{ij}_{xl(r)}, s^{ij}_{yl(r)}, s^{ij}_{bl(r)}\right). \quad (5)$$

The spatial spread b of a joint is learned as part of the field. An example of PSF ability to reason with occluded joints is given in Figure 4, where we simulate an occlusion with a black mask in the image. The PSF confidence can help to detect those partly occluded joints if the joint exists on either side image. To obtain the final confidence, we sum the predicted confidence map of PifPaf with the PSF one.

$$f_{conf}(x,y) = w f_{pif}(x,y) + (1-w) f_{psf}(x,y), \quad (6)$$

where w weights PSF voting results with PifPaf confidence map. We chose w as 0.9. This new confidence map will then be decoded by PifPaf decoder. Similarly, PSF confidence can also be combined with confidence maps from different 2D pose estimation methods.

IV. Implementation Details

A. Data Generation

We simulate stereo images based on the COCO keypoint datasets [13] with the following procedure:

- We translate the image in the x-direction exploiting the depth-disparity relationship of KITTI dataset:

$$Depth = 0.54 * 721 / Disparity \quad (7)$$

  According to statistical analysis of KITTI object dataset [7], the depth of 93% of instances ranges from 4 meters to 40 meters, so we set the maximum depth as 40 meters and minimum depth as 4 meters. According to the equation above, in our training schedule, the translation ranges from 10 to 100 pixels.
- We randomly scale down and up the images in x direction with ratio from 0.99 to 1.01 to simulate the slight scale changing between stereo image pairs.
- We randomly modify contrast and light augmentation to simulate different light conditions of stereo cameras.

Then, the model is trained on the synthetic stereo data and tested on KITTI [7] datasets. The results are shown in Table I. Compared to other models, our method has not been trained on KITTI and can directly generalize to other datasets.

B. Network and Training

Our model structure is shown in Figure 2. We use a ResNet [30] with stride 16 as a base network to extract features. A convolutional network compresses the features and distributes them to the correlation block. Inside the block, the two correlated features are concatenated with input dense features to output fields. At last, the pixel shuffle [35] is used to up-sample the feature maps. To train the correlation layer, we first load the pretrained pifpaf network and freeze the base network. The network is trained using SGD with a momentum of 0.95. The learning rate starts with 10^{-5} and then is divided by 10 every 10 epochs. A large batch of 16 images is used to include diverse-translated image pairs. After 20 epoch training, we unfreeze the ResNet and set a

small mini-batches of 4 image pairs with a small learning rate 10^{-6} to fine-tune the model.

V. Experiment

We use the synthetic stereo data to train our model, details shown in section IV-A. The model is evaluated on KITTI dataset [7] to illustrate the human 3D localization performances. We consider both *pedestrian* and *person sitting* categories of KITTI dataset and we refer to them as pedestrians for simplicity.

A. Evaluation

In the absence of a stereo keypoint dataset, we only consider the 3D localization error for the whole person rather than each joint.

1) Localization Error: We evaluate human 3D localization using the Average Localization Accuracy (ALA). ALA demonstrates the models ability to accurately localize a 3D object. A prediction will be considered as correct if the error between the predicted distance and the ground truth is smaller than a certain threshold. We also analyzed the average localization error (ALE) in different conditions. The average location error demonstrates how accurate our model estimates depth. Following KITTI guidelines, we split the detection into three difficulty regimes based on bounding box height, levels of occlusion and truncation: easy, medium and hard.

2) Evaluation Protocol: For evaluation, we follow the train/val split of Chen *et al.* [20] and evaluate on the 3769 validation images. We train and evaluate our main model on two different datasets to analyze the generalization capabilities of our network.

B. Results

1) Quantitative results: The localization error for pedestrians is shown in Table I. We outperform all the monocular approaches on most metrics. We also obtain better results than the stereo approach 3DOP [16], which has been trained and evaluated on KITTI and makes use of stereo images and point cloud during training. On the other hand, our method has been only trained on the augmented COCO dataset [13], making it less likely to overfit on KITTI dataset. We also calculated ALE for pedestrians commonly detected by all methods to make fair comparison. Stereo methods have a lower detection performance for error less than 2 meters because stereo methods require the person to appear on both stereo images. If a person is heavily occluded by an object from either camera view, miss-detections will have a high possibility to happen.

2) Distance and performance relationship: In Figure 5 we show how the performances drop with the increasing distance. Our method is much more stable than other methods: at 35 meters distance, 95% of our PSF outputs have less than 4 meters error, compared with 6 meters error of Monoloco [29] and 8 meters error of 3DOP [16]. Our PSF's mean distance error is more stable compared to other methods. As a result, PSF can maintain the depth error under a safe region and guarantee more accurate detections.

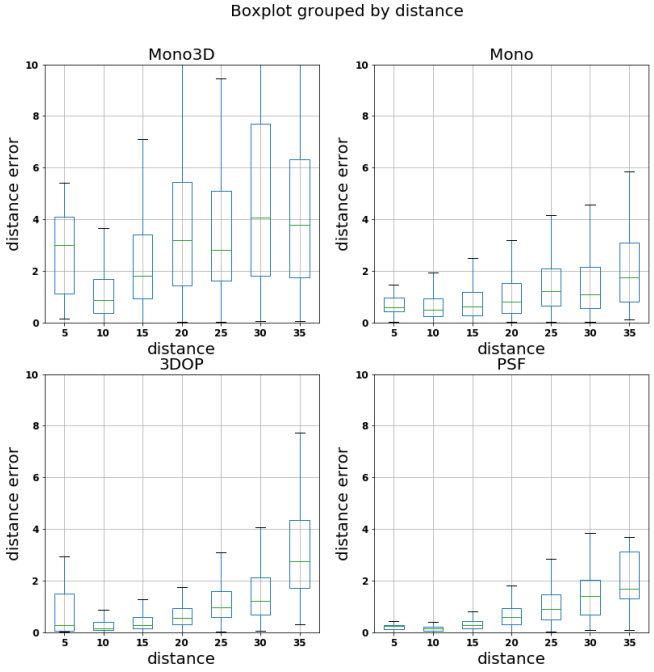

Fig. 5. Relationship between distance and distance error (as box plot). We compare our PSF method against Mono3D [25], MonoLoco [29], and 3DOP [16]. PSF has the most stable and accurate performance.

3) Qualitative results for 3D pose estimation: Another advantage of our method is the 3D pose reconstruction of a person using stereo keypoint pairs. In the absence of a stereo 3D pose dataset, we qualitatively evaluate the results using KITTI dataset. The results are shown in Figure 6, which illustrates how a stereo 2D pose pair can reconstruct the 3D pose of a pedestrian. KITTI dataset is very challenging as distances of pedestrians are normally above 10 meters. Therefore, we expect a better performance when testing on stereo 3D pose data.

4) Benefits of the correlation: To illustrate the improvement of the correlation layer, we perform an ablation study using the ALA and ALE metrics. Detailed results can be found in Table II. The correlation layer slightly helps the detection ability but significantly improves the detection quality. The ALE error is reduced by $\sim 5\%$ in all categories, and the variance in the *hard* category is reduced by half, allowing for more stable results.

5) Benefits of Stereo Voting: Following the correlation ablation study, we perform stereo voting test with the same metrics. Detailed results can be found in Table II. The voting procedure slightly helps the detection ability and quality. It is reasonable because the voting procedure helps to detect some occluded joints, which do not significantly affect the estimation of the overall 3D localization task.

6) Run time: We present the computational cost of our method in Table III. Most of the computational complexity of our method comes from the pose detector t_{pose}. For Mono3D [25] and 3DOP [16] we report published statistics on a Titan X GPU which are 1.8 s and 2.0 s, respectively. Our method takes 0.57 s on average, being 4 times faster than

TABLE I

PEDESTRIAN AL AND ALE COMPARISON

KITTI 3D Object	Type	ALE(m) Easy	Moderate	hard	ALA(%) <0.5m	<1.0m	<2.0m
Mono3D [25]	Mono	2.11(2.42)	2.93(3.27)	3.59(4.31)	12.8	22.5	37.8
MonoDepth [36]	Mono	1.40(1.69)	2.19(2.98)	2.31(3.77)	19.1	33.0	47.5
MonoLoco [29]	Mono	0.85(0.88)	0.97(1.23)	1.14(1.49)	27.6	47.8	**66.2**
3DOP [16]	Stereo	0.54(0.72)	0.85(1.13)	1.56(1.65)	41.5	54.5	63.0
Our PSF	Stereo	**0.50(0.59)**	**0.59(0.72)**	**0.73(0.65)**	**47.6**	**56.9**	63.2

We calculated ALE for pedestrians commonly detected by all methods to make a fair comparison. Values in parentheses are ALE for all ground truth. Our method outperforms all state-of-the-art methods in most situations. Especially for the very hard part, our method can well address the occlusion problem. The ALA $<$ 2m result is limited by Stereo 2D pose detection. Stereo requires detection on both stereo images, a heavy occlusion on either side or edge object missing will limit the performance.

TABLE II

ABLATION STUDY

	ALE(m) Easy	Moderate	Hard	Number	ALA(%) <0.5m	<1.0m	<2.0m
Our method without correlation	0.79[2.93]	0.77[3.5]	0.89[3.7]	2446(gt)	45.8	55.2	61.7
				1579(matched)	70.9	85.5	94.8
Our method without voting	0.62[2.10]	0.73[2.4]	0.68[1.9]	2446(gt)	47.1	56.3	62.3
				1579(matched)	71.5	85.5	94.6
Our method with corr voting	**0.59[1.90]**	**0.72[2.30]**	**0.65[0.90]**	2446(gt)	**47.6**	**56.9**	**63.2**
				1579(matched)	**71.7**	**85.7**	**95.1**

Distances larger than 45m are filtered by setting maximum depth as 45 meters. Values in brackets represent the error variance. In the notation, "corr" indicates the correlation layer, and "voting" stands for stereo joint voting.

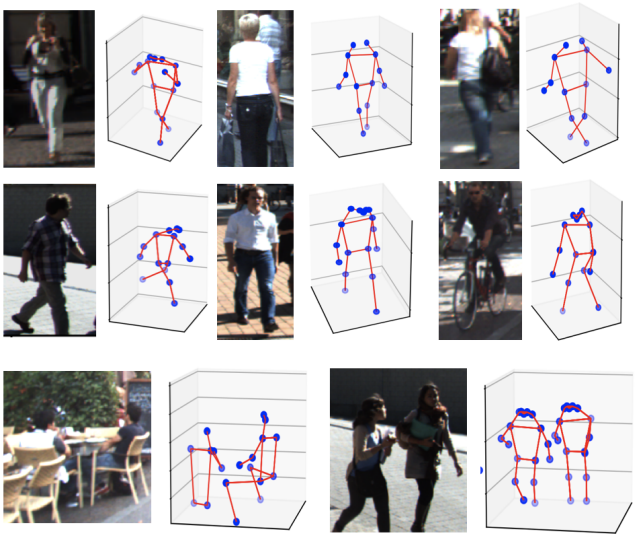

Fig. 6. Qualitative results for 3D pose estimation by stereo image pairs. The method can infer 3D poses of humans in the wild even in the presence of occluded joints and challenging lighting conditions.

Mono3D and 3DOP and slightly slower than the monocular MonoLoco [29].

VI. CONCLUSIONS AND FUTURE WORK

We present a feed-forward Neural Network architecture for stereo imaging that jointly locates and estimates human body poses in 3D. By only training on a synthetically modified COCO dataset [13], our model successfully learned composite fields from correlation features. Our uncertainty-based decoding method refines the poses and achieves accurate stereo matching. Without 3D supervision, we outperform all existing image-based methods in the human 3D localization task. As future work, our method could be used for human pose tracking in videos by relaxing the limitation on the decoding procedure.

TABLE III

RUN TIME

	Type	t_{pose}	t_{psf}	t_{model}	t_{total}
MonoLoco [29]	Mono	162	-	10	172
Mono3D [25]	Mono	-	-	1800	1800
3DOP [16]	Stereo	-	-	2000	2000
Our method	Stereo	470	85	15	570

t_{pose} represents 2D pose decoding time, t_{psf} represents PSF decoding time, t_{model} represents network inference time.

Acknowledgments. We acknowledge the support of Samsung for this work.

REFERENCES

[1] Z. Yang, Y. Sun, S. Liu, X. Shen, and J. Jia, "STD: sparse-to-dense 3d object detector for point cloud," *ICCV*, vol. abs/1907.10471, 2019.

[2] M. Liang, B. Yang, Y. Chen, R. Hu, and R. Urtasun, "Multi-task multi-sensor fusion for 3d object detection," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[3] P. Li, X. Chen, and S. Shen, "Stereo R-CNN based 3d object detection for autonomous driving," *CVPR*, vol. abs/1902.09738, 2019.

[4] Z. Qin, J. Wang, and Y. Lu, "Triangulation learning network: from monocular to stereo 3d object detection," *CVPR*, vol. abs/1906.01193, 2019.

[5] C. Chen, Y. Liu, S. Kreiss, and A. Alahi, "Crowd-robot interaction: Crowd-aware robot navigation with attention-based deep reinforcement learning," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 6015–6022.

[6] A. Alahi, M. Bierlaire, and M. Kunt, "Object detection and matching with mobile cameras collaborating with fixed cameras," in *The 10th European Conference on Computer Vision*, no. CONF, 2008.

[7] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *2012 IEEE Conference*

on Computer Vision and Pattern Recognition, June 2012, pp. 3354–3361.

[8] Y. Verdie, K. Yi, P. Fua, and V. Lepetit, "Tilde: a temporally invariant learned detector," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 5279–5288.

[9] S. Kreiss, L. Bertoni, and A. Alahi, "Pifpaf: Composite fields for human pose estimation," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[10] G. Papandreou, T. Zhu, L.-C. Chen, S. Gidaris, J. Tompson, and K. Murphy, "Personlab: Person pose estimation and instance segmentation with a bottom-up, part-based, geometric embedding model," in *ECCV*, 2018.

[11] C. Feichtenhofer, A. Pinz, and A. Zisserman, "Detect to track and track to detect," in *2017 IEEE International Conference on Computer Vision (ICCV)*, Oct 2017, pp. 3057–3065.

[12] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox, "Flownet 2.0: Evolution of optical flow estimation with deep networks," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017, pp. 1647–1655.

[13] 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*, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds. Cham: Springer International Publishing, 2014, pp. 740–755.

[14] Y. Zhou and O. Tuzel, "Voxelnet: End-to-end learning for point cloud based 3d object detection," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 4490–4499.

[15] C. R. Qi, W. Liu, C. Wu, H. Su, and L. J. Guibas, "Frustum pointnets for 3d object detection from rgb-d data," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 918–927.

[16] X. Chen, K. Kundu, Y. Zhu, A. G. Berneshawi, H. Ma, S. Fidler, and R. Urtasun, "3d object proposals for accurate object class detection," in *Advances in Neural Information Processing Systems 28*, C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, Eds. Curran Associates, Inc., 2015, pp. 424–432. [Online]. Available: http://papers.nips.cc/paper/5644-3d-object-proposals-for-accurate-object-class-detection.pdf

[17] P. Li, T. Qin, and S. Shen, "Stereo vision-based semantic 3d object and ego-motion tracking for autonomous driving," in *Computer Vision – ECCV 2018*, V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, Eds. Cham: Springer International Publishing, 2018, pp. 664–679.

[18] Y. Wang, W.-L. Chao, D. Garg, B. Hariharan, M. Campbell, and K. Q. Weinberger, "Pseudo-lidar from visual depth estimation: Bridging the gap in 3d object detection for autonomous driving," in *The IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 8445–8453.

[19] L. Pishchulin, E. Insafutdinov, S. Tang, B. Andres, M. Andriluka, P. Gehler, and B. Schiele, "Deepcut: Joint subset partition and labeling for multi person pose estimation," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016, pp. 4929–4937.

[20] E. Insafutdinov, L. Pishchulin, B. Andres, M. Andriluka, and B. Schiele, "Deepercut: A deeper, stronger, and faster multi-person pose estimation model," in *Computer Vision – ECCV 2016*, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds. Cham: Springer International Publishing, 2016, pp. 34–50.

[21] Z. Cao, G. Hidalgo Martinez, T. Simon, S. Wei, and Y. A. Sheikh, "Openpose: Realtime multi-person 2d pose estimation using part affinity fields," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2019.

[22] J. Martinez, R. Hossain, J. Romero, and J. J. Little, "A simple yet effective baseline for 3d human pose estimation," in *2017 IEEE International Conference on Computer Vision (ICCV)*, Oct 2017, pp. 2659–2668.

[23] T. Pfister, J. Charles, and A. Zisserman, "Flowing convnets for human pose estimation in videos," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1913–1921.

[24] R. A. Gler, N. Neverova, and I. Kokkinos, "Densepose: Dense human pose estimation in the wild," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, June 2018, pp. 7297–7306.

[25] X. Chen, K. Kundu, Z. Zhang, H. Ma, S. Fidler, and R. Urtasun, "Monocular 3d object detection for autonomous driving," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016, pp. 2147–2156.

[26] M. Z. Zia, M. Stark, and K. Schindler, "Are cars just 3d boxes? jointly estimating the 3d shape of multiple objects," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, June 2014, pp. 3678–3685.

[27] B. Xu and Z. Chen, "Multi-level fusion based 3d object detection from monocular images," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, June 2018, pp. 2345–2353.

[28] J. Ku, A. D. Pon, and S. L. Waslander, "Monocular 3d object detection leveraging accurate proposals and shape reconstruction," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 11 867–11 876.

[29] L. Bertoni, S. Kreiss, and A. Alahi, "Monoloco: Monocular 3d pedestrian localization and uncertainty estimation," *IEEE International Conference on Computer Vision (ICCV)*, vol. abs/1906.06059, 2019.

[30] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016, pp. 770–778.

[31] A. Dosovitskiy, P. Fischer, E. Ilg, P. Husser, C. Hazirbas, V. Golkov, P. v. d. Smagt, D. Cremers, and T. Brox, "Flownet: Learning optical flow with convolutional networks," in *2015 IEEE International Conference on Computer Vision (ICCV)*, Dec 2015, pp. 2758–2766.

[32] A. Kendall and Y. Gal, "What uncertainties do we need in bayesian deep learning for computer vision?" in *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds. Curran Associates, Inc., 2017, pp. 5574–5584. [Online]. Available: http://papers.nips.cc/paper/7141-what-uncertainties-do-we-need-in-bayesian-deep-learning-for-computer-vision.pdf

[33] M. Kocabas, S. Karagoz, and E. Akbas, "Multiposenet: Fast multi-person pose estimation using pose residual network," *ECCV*, vol. abs/1807.04067, 2018.

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

[35] W. Shi, J. Caballero, F. Huszr, J. Totz, A. P. Aitken, R. Bishop, D. Rueckert, and Z. Wang, "Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016, pp. 1874–1883.

[36] C. Godard, O. M. Aodha, and G. J. Brostow, "Unsupervised monocular depth estimation with left-right consistency," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017, pp. 6602–6611.