ECP2.5D - Person Localization in Traffic Scenes

Markus Braun*,1,2, Sebastian Krebs*,1,2 and Dariu M. Gavrila[1]

***Abstract*— 3D localization of persons from a single image is a challenging problem, where advances are largely data-driven. In this paper, we enhance the recently released EuroCity Persons detection dataset, a large and diverse automotive dataset covering pedestrians and riders. Previously, only 2D annotations and image data were provided. We introduce an automatic 3D lifting procedure by using additional LiDAR distance measurements, to augment a large part of the reasonable subset of 2D box annotations with their corresponding 3D point positions (136K persons in 46K frames of day- and night-time).**

The resulting dataset (coined ECP2.5D), now including LiDAR data as well as the generated annotations, is made publicly available for (non-commercial) benchmarking of camera-based and/or LiDAR 3D object detection methods. We provide baseline results for 3D localization from single images by extending the YOLOv3 2D object detector with a distance regression including uncertainty estimation.

I. Introduction

The accurate 3D localization of objects, especially of vulnerable road users like pedestrians and riders, is essential for the safety of self-driving vehicles. Recently, there has been great interest in monocular 3D object detection, as seen by the numerous publications over the last years [1]–[13]. Cameras are still a lot cheaper than LiDAR sensors, usually have a higher resolution and are important because of redundancy. Lately, many computer vision tasks like scene segmentation [14] and 2D detection [15]–[17] have been boosted by deep learning.

However, the performance of monocular 3D detection still lags behind the LiDAR methods mainly because the problem is ill-posed due to missing depth information in a 2D image. Therefore, the commonly used KITTI 3D benchmark [18] is still lead by LiDAR based approaches. The amount of persons in comparison to vehicles is considerably low in the KITTI benchmark and thus many monocular detection methods focus on the detection of the latter.

The EuroCity Persons (ECP) dataset [19] focuses on persons in urban scenarios and is one of the most diverse automotive datasets collected in 31 cities of 12 countries (see. Fig. 1). We introduce a label lifting procedure to generate 2.5D annotations by enriching the already present accurate 2D boxes by their corresponding distance measurements recorded with a HDL-64E Velodyne. After egomotion correction and projection of the point clouds into the image, the corresponding point cluster for every 2D bounding box is found by applying several filters and constraints (see. Fig. 1). We apply this lifting procedure to all annotations in the reasonable subset as defined in [19]. As not all objects are covered by LiDAR measurements, this works for 97% of the annotations. The 3D position of each object is defined by the mean distance of all points contained in the assigned cluster and the 2D bounding box.

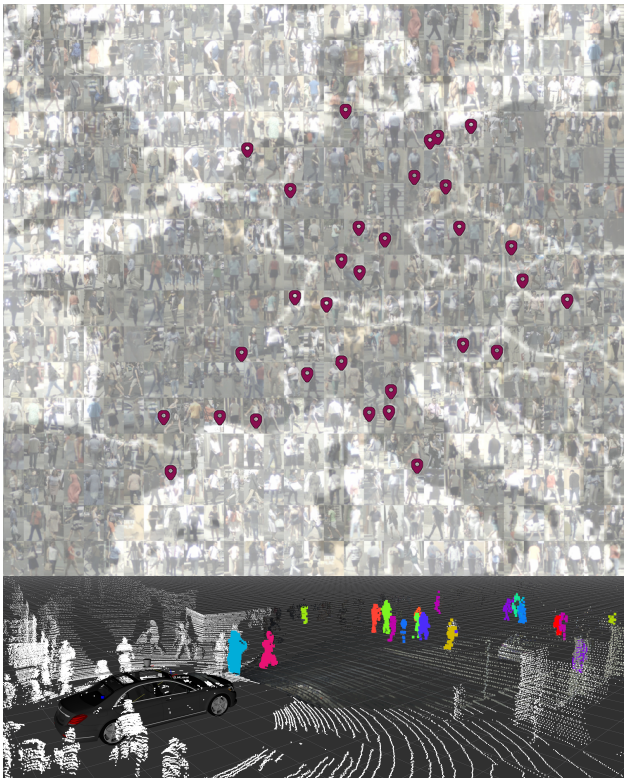

Fig. 1: The ECP dataset [19] has been recorded in 31 cities of 12 European countries (top). We present an automatic label lifting for the provided 2D box annotations, by finding the corresponding point cloud cluster to infer the 3D box center position (bottom).

KITTI and recent 3D datasets like [20] are annotated with 3D bounding boxes. The detection problem is defined by estimating the nine degrees of freedom: three box extent values, three values for the box center and three rotation angles. The KITTI 3D object detection benchmark simplifies the problem by setting two rotation angles to zero. Some monocular methods [21] and [6]–[8] solve this task by estimating the metrical extents of the object and using geometric constraints to infer its 3D box based on its 2D bounding box location. The estimated box is then matched based on a 3D intersection over union (IoU). A high 3D IoU is hard to achieve for monocular object detection and persons in particular, as a small translation offset may result in no overlap at all [12]. In contrast to estimating a full 3D box,

∗) equal contribution
1) Intelligent Vehicles group, TU Delft, The Netherlands
2) Environment Perception group, Mercedes-Benz AG, Germany

TABLE I: Overview of publicly available 3D person detection datasets recorded in an automotive setting, offering LIDAR point cloud and camera data.

Dataset	KITTI [18]	nuScenes [22]	Argoverse [23]	Waymo [20]	ECP2.5D (ours)
# Countries	1	2	1	1	12
# Cities	1	2	2	2	30
# Imgs	15k	34k	350k	800k	46k
# Peds	9.4k	222k	132k	2.8M	123k
# Riders	3.3k	24k	11k	67k	13k
# Seasons	1	-	1	-	4
Weather	dry	dry, rain	dry	dry, rain	dry, rain
Unblurred	✓	✗	✗	✗	✓

[13] only considers the 3D localization problem with three degrees of freedom, namely the x, y and z coordinates of the 3D box. Assuming the projected 3D box center coincides with the 2D box center for pedestrians, only the 2D box itself and the distance has to be estimated to calculate the 3D object center. Thus, in the following we use the term 3D localization for the problem of estimating an object's 3D point position.

We extend the YOLOv3 [15] 2D object detector to regress the distance directly for RGB-images as input. This approach is similar to methods like [10]–[12] and serves as a baseline on our dataset. Instead of estimating the Euclidean distance as in [13] we estimate the distance along the z-axis in the optical camera frame as an equivalent surrogate. As in [12], [13] our network estimates uncertainties of the regressed distance values.

II. Related Work

a) 3D datasets: Over the last decade, several datasets have been published to enable the development of detection algorithms in an automotive setting. The pioneering KITTI dataset [24] offers multi-modal sensor data, including camera and LiDAR, with corresponding 2D and 3D object annotations [18]. However, the comparably small number of samples ($\sim$9400 pedestrians) limits the progress in research areas with a high data demand. Lately, several automotive datasets have been published, trying to fill this shortcoming by offering a greater number of 3D annotations [20], [22], [23], along with the raw sensor readings (point cloud, camera). A comparison is given in Tab. I. Most datasets involve human annotations performed directly in the point clouds [20], [22], [23].

In comparison to the above mentioned datasets, the ECP dataset [19] currently only provides 2D annotations. Still, ECP is a large dataset, which offers unblurred images of challenging urban scenarios and has a high geographic diversity being recorded in cities all over Europe. Hence, the extension of ECP to 3D would be beneficial for the research community, e.g.: as shown in [25], where our 2.5D annotations have been utilized to generate approximate 3D bounding boxes used for LiDAR based detection.

b) Monocular 3D detection methods: A lot of research has been put into 3D object detection from LiDAR point clouds or stereo-based RGBD images. Here, we focus on recent work on monocular 3D object detection. [2], [3] use fully convolutional networks to regain pixel-wise depth information for the complete image and apply neural networks on RGBD. [1] argues that convolutions on a d-channel is sub-optimal and converts estimated depth maps from mono or stereo to a LiDAR point cloud to apply LiDAR based detection methods. [4], [5] also transform the feature space before final detection. [4] detects objects in a bird-eye view image generated with inverse perspective mapping, while [5] integrates an orthographic feature transform into its network.

Other works extend RGB based 2D object detectors for 3D detection. [21] presumes that the projected 3D box tightly fits into the 2D detection of an extended MS-CNN network [17] resulting in several geometric constraints. Adding the additional regression values for the extent and rotation angles for each object the 3D box can be inferred by solving an overconstrained equation system. Small errors in the 2D location regression may result in large errors for the 3D IoU. Therefore [6]–[8] optimize performance by leveraging the strict geometric constraint.

There is also a group of methods that directly estimate the 3D position. [9] builds upon the region proposal network introduced by Faster R-CNN [16]. The authors use 2D anchor boxes with 3D properties calculated on the dataset statistics. For each anchor the 3D box is regressed relatively to its 3D properties. [11] regresses 3D bounding boxes in a multistage approach using 2D boxes as input for a feature pooling. They mention that pixel-wise depth estimation often neglects small objects, and therefore prefer depth estimation on an instance level. [10] also directly regresses 2D and 3D bounding boxes. As the combined loss for all regressed values might be hard to train, they introduce a disentangling to optimize the values separately. [12] regresses 26 values as a surrogate of the 3D box. One of these values is the Euclidean distance of the 3D box center. The authors compare results for homoscedastic and heteroscedastic loss formulations whose theoretical foundations are described in [26]. In the latter they get an uncertainty per regression value. They also note that the 3D IoU is very challenging for monocular detection methods. [13] focuses on pedestrians estimating distances based on skeleton points as input. Assuming the projected 3D position coincides with the 2D box center, the distance is sufficient to solve the 3D localization. Furthermore, they provide epistemic and aleatoric uncertainties for the estimated distance as an important information for autonomous driving.

c) Contributions: First, we introduce the ECP2.5D dataset, which will be publicly available for non-commercial scientific use[a]. We publish the 2.5D annotations generated by our proposed uplifting method and egomotion corrected point cloud data for the images of ECP. Futhermore, the corresponding camera parameters as well as the LiDAR to camera transformations are released, to enable training and

[a]https://eurocity-dataset.tudelft.nl/

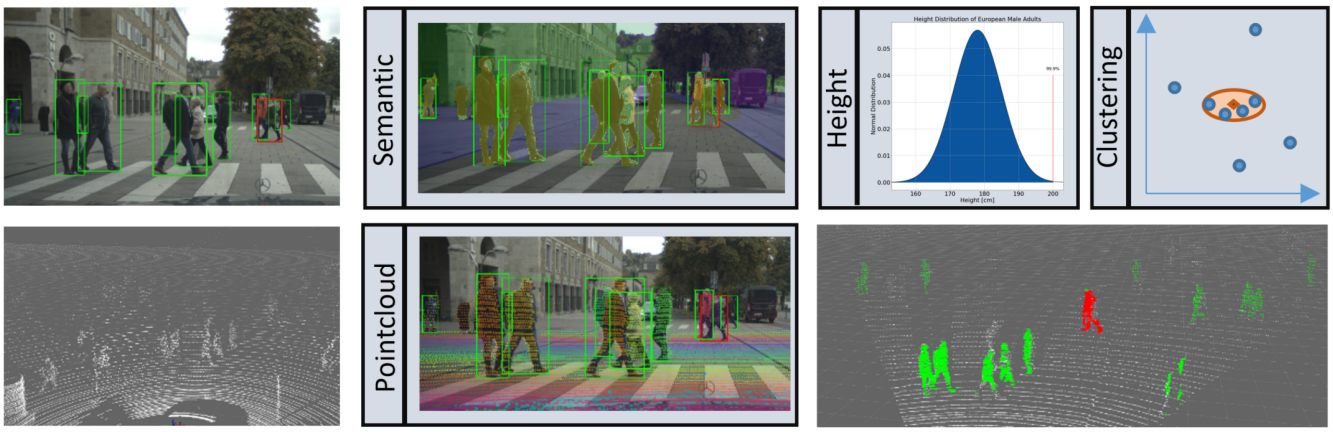

Fig. 2: Overview of our 3D lifting procedure. Inputs are 2D bounding box annotations and 3D point clouds (left column). The LiDAR points are egomotion corrected and projected into the image (middle column bottom). For every bounding box only points within the bounding box supported by the scene segmentation (middle column top) are potential candidates. After applying additional constraints the final point clusters per bounding box are found using a clustering step (right).

evaluation of LiDAR and camera+LiDAR based approaches. Second, we provide baseline results for monocular vision-based 3D person localization on this new dataset. As baseline we use a YOLOv3 detector extended by distance regression and uncertainty weighting.

III. ECP2.5D Dataset

In the following the ECP2.5D dataset is introduced. After describing the recording setup, the proposed label lifting process is presented, which is used to extract the 3D position.

A. Sensor Setup and Data Basis

The vehicle sensor setup used to record the ECP dataset included a Velodyne HDL-64E LiDAR scanner, a GPS/INS system, and the front-facing camera. The camera data has already been published in [19]. To enable fusion of the different sensor data, a GPS-based time synchronization approach is employed. In addition, all sensors are calibrated intrinsically and extrinsically (translation and rotation to common axis). Given the continuous measurement process of the LiDAR, the points within one 360° scan can be distorted due to the egomotion of the vehicle. Furthermore, there is a time offset between each 360° LiDAR scan and camera measurements, as the sensors are not simultaneously triggered. To address both issues, the temporally closest point cloud for each image is compensated by the kinematic information between the measurement time of each point and the image trigger time. This results in one point cloud for each image, which apart from other dynamic objects is consistent in a world frame (i.e. no distortion due to the vehicle motion) and is projected onto the image capture time.

To combine the 3D LiDAR information with the annotation information present in the unrectified image, each LiDAR point needs to be projected into the image. Each 3D-LiDAR point given by $p^l = [x^l, y^l, z^l]$, is transformed to the 3D camera coordinate system $p^c = [x^c, y^c, z^c]$ with the homogeneous transformation matrix $T_{l\to c}$, obtained by the extrinsic calibration. By applying the basic pinhole model each p^c is projected to rectified image coordinates $\tilde{p} = [\tilde{u}, \tilde{v}]$. The final position $p = [u, v]$ in the unrectified image is achieved by unrectifying $\tilde{p}$ using the intrinsic camera parameters. The complete projection chain is thus given by:

$$p^l \xrightarrow{T_{l\to c}} p^c \xrightarrow{\text{Pinhole}} \tilde{p} \xrightarrow{\text{unrectify}} p. \tag{1}$$

Note, due to the parallax caused by the different sensor mounting positions, the projected LiDAR point p was not necessarily caused by the same physical object as depicted by the image at pixel $[u, v]$.

B. Label Lifting

The goal is to get the 3D position of a 2D bounding box annotation in the camera coordinate system. To this end we use the egomotion compensated and projected LiDAR points to identify point clusters representing the annotated person. The exact details are as follows.

For each person annotation defined by its bounding box $(u_1^*, v_1^*, u_2^*, v_2^*)$: first, an initial point set is created containing all LiDAR points p_i^l with a corresponding projection p_i inside the bounding box. Where $i < n$ and n is the number of all LiDAR points within a 360° scan. Second, we use the height distribution of persons as prior knowledge. For each point p^c we verify if it could have been reflected/caused by a person. Using a pinhole camera model we can relate an object's height in pixels $\Delta\tilde{v}$ to its real height Δy^c by

$$\frac{\Delta\tilde{v}}{\tilde{f}} = \frac{\Delta y^c}{z^c}, \tag{2}$$

with $\tilde{f}$ being the focal length and z^c the z coordinate. Hereby we assume that the object is standing upright and the 3D points defining upper and lower boundary of the 2D box share the same z^c coordinate. Given the distance of the point z^c and the pixel height $\Delta\tilde{v}$ of the rectified bounding box, we calculate the potential object height Δy^c in metres according to Eq. (2). This height is expected to be in the range of $[\Delta y^c_{min}, \Delta y^c_{max}]$, with $\Delta y^c_{min} = 0.7\ m$ and $\Delta y^c_{max} = 2.0\ m$. The lower bound is given by the 0.1% quantile of the height distribution of 18 months old girls [27],

while the upper bound is given by the 99.9% quantile of the height distribution of adult men [28]. All points resulting in object heights outside of this range are expected to be caused by noise (e.g.: background or occlusions), and are thus removed from the person's points set.

Third, using a scene segmentation method [14] trained on CityScapes [29], we assign the predicted class at the projected image position p_i to each LiDAR point. All LiDAR points which do not support the bounding box class (pedestrian or rider) are excluded from the set of points associated to the current bounding box.

Finally, point clusters are extracted from the filtered point sets. Objects with overlapping bounding boxes are grouped together. For each group the filtered point sets are united and processed jointly, to prevent assigning one point to multiple objects. An extended version of the constrained k-means clustering algorithm [30] is employed to find k point cluster for the point set of every object group, with k being the size of the group. In [30] the original k-means clustering is extended by a set of *must link* and *cannot link* point tuples. For our approach we utilize solely the cannot link set to prevent that points which cannot belong to the same object as they lie within different bounding boxes end up in the same cluster. Furthermore, points with a distance greater than $1.5m$ cannot be caused by the same person and are thus also added to the cannot link set. In the original version the algorithm fails as soon as a point cannot be assigned to any cluster. To avoid that, we extend the algorithm by introducing an outlier class for all points without a fitting cluster. We set the number of initial clusters to k, but allow a cluster generation during the first iterations of the algorithm (to account for additional depth modes e.g. caused by occlusions, those points have not been filtered by the previous steps). For the final cluster to object assignment, the objects are sorted by their occlusion level and base point v_2^* in image coordinates. Thus, non-occluded, closer objects - assuming a flat world - are processed first. For each object the cluster with the highest number of points within its bounding box is chosen. Clusters are only assigned to objects once. The final 3D position is defined by the intersection point of the ray of sight through the 2D bounding box center with the x-y-plane at the mean distance $\bar{z}^c$ of all points within the assigned cluster.

If not mentioned otherwise, the distance is always defined by the distance value along the z-axis, which is orthogonal to the image plane. Our baseline model described in Sec. IV-B is trained using $\bar{z}^c$ as groundtruth. For an overview of the label lifting see Fig. 2.

C. Dataset Statistics, Label Quality, Small Persons Subset

TABLE II: Comparison of the night, day and combined ECP2.5D datasets (numbers are rounded).

ECP2.5D	# Countries	# Cities	# Imgs	# Peds	# Riders
Night	6	7	7k	22k	1k
Day	12	30	39k	101k	11k
Total	12	30	46k	123k	13k

Since challenging occlusion scenarios are out of scope of our lifting procedure, we only apply it to the reasonable subset of the ECP dataset, i.e. pedestrians of at least 40 pixels height and less than 40% occlusion. Using these prerequisites the uplifting procedure is applied to a total of 140729 2D person annotations, resulting in 136096 annotations which can be enriched with 3D distance information. These annotations form the ECP2.5D dataset. We employ the same train, validation, and test splits as in [19], namely $60-10-30$. The labels of the test split will not be published. The numbers for the night and day subset are shown in Table II.

To validate the quality of the generated 3D positions the object center for all uplifted objects on a randomly selected subset is manually annotated. We define an uplifted 2.5D annotation as error if the calculated object center differs more than a pre-defined threshold ϵ_m from the manually labeled one. The manual annotations used for the quality assessment have been performed independently by two experienced labelers on a subset of 90 random frames, containing a total of 337 uplifted objects. For the distance threshold $\epsilon_m = 0.35$ this results in an error of $4,2\%$, with no significant difference between the two labelers. The errors are caused by sensor parallax, imprecise semantic masks, wrong cluster generation, or an erroneous 2D label. The differences between the automatic and manual annotations are depicted in Fig. 3.

We use Eq. (2) to approximate the height for each object based on its pixel height and the annotated distance. The resulting distribution and other statistics are shown in Fig. 3. The mean height of persons in our dataset is 1.68 m. Children and other small persons are rare. Because of the strong dataset bias regarding person heights, it is especially interesting to evaluate methods for smaller persons to verify the network does not overfit on a certain height. This could easily happen especially with methods that depend on geometric constraints and estimate the object dimensions as [21]. As a wrong distance label also results in a wrong height estimate, some small persons according to the height statistics are in fact caused by label errors. Therefore, we manually validate for pedestrians up to a distance of 40 m and an height estimate below 1.3 m, if their distance estimate is correct. This results in a subset of 196 small pedestrians within the day test set, which we refer to as the small subset.

IV. Methodology

Similar to [13], as the projected 3D position $\bar{p}^c$ coincides with the center of the 2D bounding box, only the 2D box and the distance along the z-axis of the camera coordinate system has to be estimated for 3D localization. Doing so, the 3D position is estimated by taking the camera ray through the rectified bounding box center with a length depending on the estimated distance. We adopt this problem formulation. First, we revisit YOLOv3 [15] which is used as our underlying 2D object detector. Similar to [31] we take care that all losses match a probabilistic log-likelihood formulation to make use of task uncertainty weighting as proposed in [26]. In [12] a zero mean Gaussian is used to

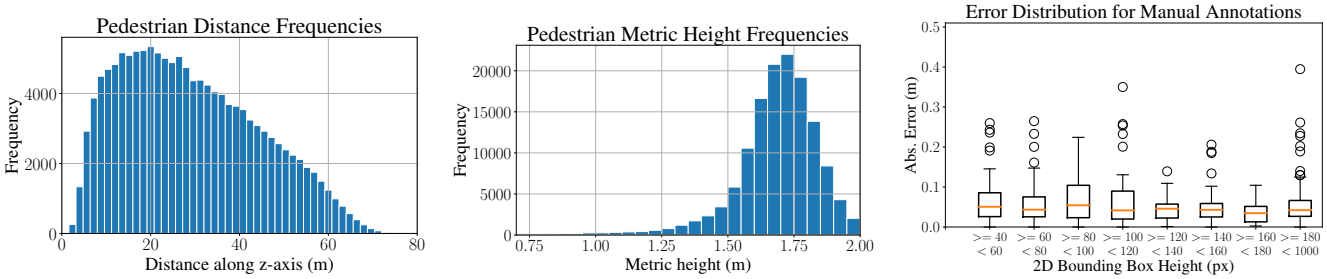

Fig. 3: Frequencies of lifted persons within our ECP2.5D dataset depending on the measured distance (left) and estimated height (middle). On the right the results of our manual quality check is depicted in dependence of the 2D object height.

model the heteroscedastic uncertainty in the regression of the 26 surrogates for the 3D bounding box, of which one is the distance to the 3D box center. We also use a normal distribution based regression loss to estimate the distance and uncertainty along the z-axis. The extended YOLOv3 method serves as baseline on ECP2.5D. Finally, we introduce the metrics applied for evaluation of 2D detection, distance estimation and 3D localization.

A. Single stage detection with task uncertainty weighting

YOLOv3 extends the Darknet53 architecture and predicts bounding boxes based on three feature layers, that are downscaled by a factor of 8, 16 and 32 respectively. Each cell within these feature layers encodes prior boxes of different aspect ratios that are centered within the cell. Given an input image $\boldsymbol{x}$ our convolutional neural network f parameterized with w predicts four coordinate offsets $f^w_{loc}(\boldsymbol{x})$ and c class scores $f^w_{cls}(\boldsymbol{x})$ per prior box p. In contrast to [15] we skip the objectness classification and directly regress the four bounding box edges as in [32]. The class likelihood is calculated by

$$p(y|f^w_{cls}(\boldsymbol{x})) = \text{softmax}(f^w_{cls}(\boldsymbol{x})). \tag{3}$$

Similar to [31] we model the regressed bounding box values to follow a multivariate normal distribution. We use a diagonal covariance matrix with identical entries σ_{loc} and minimize the negative log-likelihood, which results in a L2 loss $\mathcal{L}_{loc}(w)$. Regarding classification we apply a standard cross-entropy loss $\mathcal{L}_{cls}(w)$, which is the log-likelihood of the probability function in Eq. (3). Our detection losses match the regression and classification losses described in [26], so we can use task uncertainty weighting for the total loss

$$\mathcal{L}(w) = \frac{1}{\sigma^2_{cls}}\mathcal{L}_{cls}(w) + \log\sigma_{cls} + \frac{1}{2\sigma^2_{loc}}\mathcal{L}_{loc}(w) + \log\sigma_{loc} \tag{4}$$

with the aleatoric, homoscedastic uncertainty weights σ_{loc} and σ_{cls} optimized during training. During training all person samples those bounding boxes have an IoU > 0.5 with a prior box are associated as positive training targets. Prior boxes with no associated sample only contribute to the classification loss.

B. Distance regression with heteroscedastic uncertainty

Regarding the ill-posed distance estimation of persons we use a heteroscedastic uncertainty (similar to [12], [13]), as it highly depends on the input data. If the context information is low, e.g. there are barely objects of known sizes in the surrounding of the person, the uncertainty is expected to be higher. Hence, apart from regressing the distance μ_z in the optical camera frame, we also predict the uncertainty σ^2_z as direct output of the model. In total, the network estimates two additional values $f^w_z(\boldsymbol{x}) = (\mu_z, \sigma^2_z)$ per prior box p. Similar to the localization in the previous section, the likelihood of the distance y_z is modelled as a normal distribution:

$$p(y_z|f^w_z(\boldsymbol{x})) = \mathcal{N}(\mu_z, \sigma^2_z), \tag{5}$$

and we minimize the negative log likelihood loss for a given groundtruth distance $\bar{z}^c$:

$$\mathcal{L}_z(w) = -\log p(\bar{z}^c|f^w_z(\boldsymbol{x})) \propto \frac{1}{2\sigma^2_z}\|\bar{z}^c - \mu_z\| + \log\sigma_z. \tag{6}$$

For the joint network we add all three losses with an additional manual weight λ_z for the distance loss. For prior boxes without an associated sample or without an associated distance label the distance loss is zero.

C. Metrics

The 2D detection performance on ECP2.5D is evaluated using the standard LAMR metric as in [19]. It is the geometric mean of the miss-rate for ten logarithmically scaled false-positives-per-image (fppi) reference values. We pass on the 3D IoU based matching criterion, that is used by the KITTI benchmark [18], since it is hard to fulfill by monocular 3D detection methods, as mentioned in Sec II. [12] and [13] apply the average localization precision (ALP) metric proposed in [33]. There, predictions and groundtruth objects are matched on an absolute distance threshold of one or two metres. As we expect a linearly increasing distance error in dependence of the groundtruth distance $\bar{z}^c$, we apply a relative error metric for distance evaluation. We evaluate the predicted distance μ_z for all detections at a fppi of 1.0. The relative distance error is defined by

$$e_z(\bar{z}^c, \mu_z) = \frac{|\bar{z}^c - \mu_z|}{\bar{z}^c}, \tag{7}$$

and is calculated for each matched detection ground-truth pair $(\bar{z}^c, \mu_z)$ in the set of all n matches Ω. The mean relative error (MRE) is given by

$$\text{MRE} = \frac{1}{n}\sum_{(\bar{z}^c,\mu_z)\in\Omega} e_z(\bar{z}^c, \mu_z). \tag{8}$$

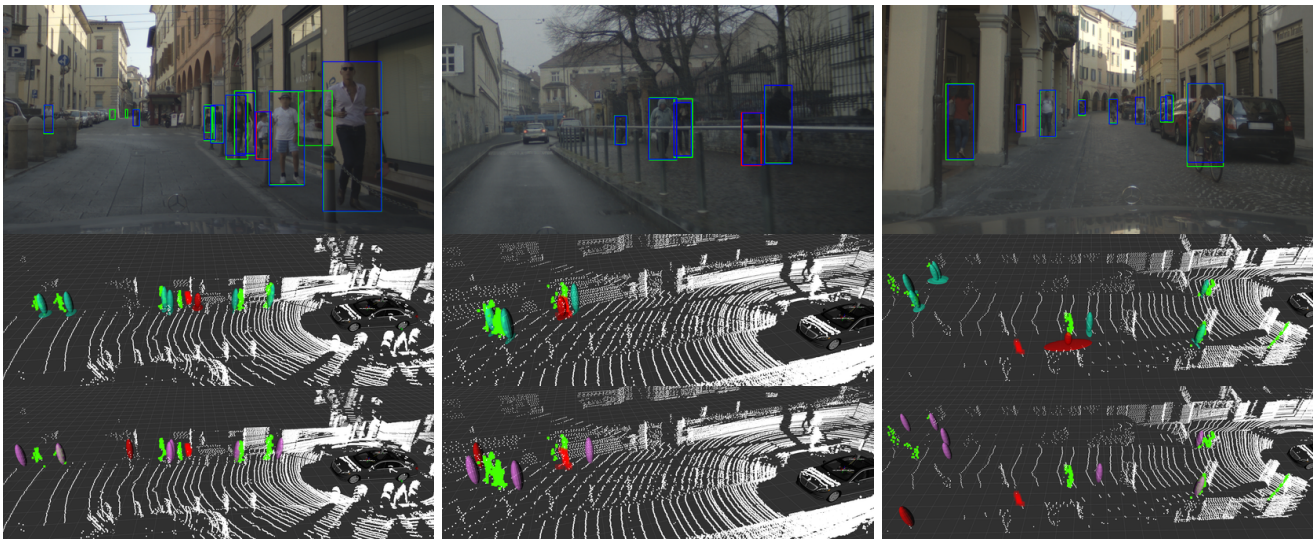

Fig. 4: Qualitative results of our proposed approach. Top row: Input image, detected pedestrians (blue) and ground-truth annotations (green). Middle row: point clusters per person (green), estimated distance (teal upright ellipsoid) and uncertainty for each detection (teal covariance ellipse). Bottom row: Distance estimations of the fixed height baseline, visualized as magenta cylinders. The selected object is shown in red in all rows.

For evaluation of the 3D localization for a groundtruth position $\bar{p}^c$ and its estimate $\hat{p}^c$ we use the relative 3D error

$$e_{3D} = \frac{\|\bar{p}^c - \hat{p}^c\|}{\|\bar{p}^c\|}. \tag{9}$$

In addition, we calculate a joint metric called LAMR_{3D}. There, predictions and test samples only match if their relative error e_{3D} is below a threshold of 0.1 or 0.2. Everything else, including the 2D IoU matching criterion for an IoU threshold of 0.5, is identical to the definition of the LAMR.

V. Experiments

We build upon the YOLOv3 tensorflow implementation provided by [31] for our experiments. As in [19] nine prior box sizes are calculated with the dimension clustering proposed in [15] on the training split of the ECP dataset and distributed on the three output layers. Hence, we get the same prior box recall as in [19], which is about 100% for an IoU of 0.5. The networks are trained to discriminate pedestrians and riders. Still, we focus on the evaluation of the former as pedestrians are a lot more frequent. Flipping and a crop and scale augmentation have been used in all trainings. Predictions are filtered in 2D with a greedy non-maximum suppression parametrized with an IoU threshold of 0.5. We used an enhanced debayering for the images of ECP, that improved the visual appearance but did not show an influence on detection performance for our YOLOv3 implementation. Experiments are run and evaluated on day-time data only.

A. Base 2D detection model

First, we train a base 2D detection model (named Base) on the ECP training dataset to be used for initialization of our joint model on ECP2.5D. The Darknet53 part of our adapted YOLOv3 network - using the detection losses and task-uncertainty weighting described in Sec. IV-A - is initialized with weights optimized for classification on ImageNet [34]. The network is trained for 800,000 iterations with an initial learning rate of 1e-5, which is decreased by a factor of 0.1 after 300,000 and 600,000 iterations. A focal loss (see [35]) weighting with $\gamma = 2.0$ instead of the standard cross entropy loss is used, as it improves the detection performance. The best performing model with the lowest LAMR on the reasonable scenario on the validation subset is selected and evaluated on the ECP test dataset. It reaches a LAMR of 7.0 in contrast to 8.5 as shown in [19], where the Darknet implementation of [15] was used. In Tab. III the LAMR of 6.2 is lower on ECP2.5D. This difference occurs as ECP2.5D only contains 97% of the objects of the ECP reasonable scenario. Instances of the ECP dataset that are not lifted and thereby not part of the ECP2.5D annotations still serve as ignore instances during evaluation.

B. Joint detection and distance estimation

For the training of our distance model (named L2) on ECP2.5D we add randomly initialized convolutional layers for the two distance outputs described in Sec. IV-B. All other layers are initialized with the base detection model. Due to numerical stability, we predict $\log \sigma_z^2$ instead of σ_z^2 as in [31]. At the beginning of the training the L2 loss for metric distances in metres is higher than for the other losses. Therefore, we adapt the bias values of the uncertainty estimation in the last convolutional layer. The manual loss weight λ_z is set to 0.5. All layers of the joint model including those of the Darknet-53 base network are optimized for another 550,000 iterations on the training split of ECP2.5D. Not uplifted instances of ECP still contribute to the detection losses. The initial learning rate is set to 1e-5 and reduced by a factor of 0.1 after 300,000 and 500,000 iterations. The results for the joint model generating 2D predictions including the distance estimations are shown in Tab. III. Similar to [13] we also compare results for a fixed height baseline (named

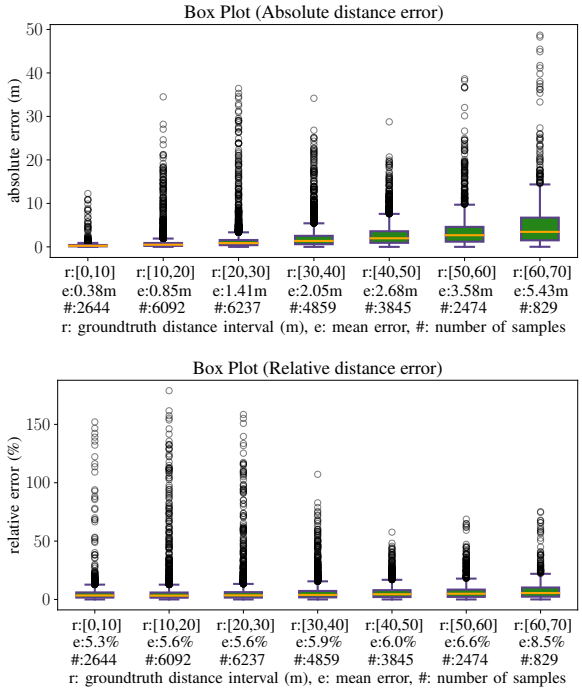

Fig. 5: Boxplots for absolute (top) and relative (bottom) distance error in dependence of groundtruth distance.

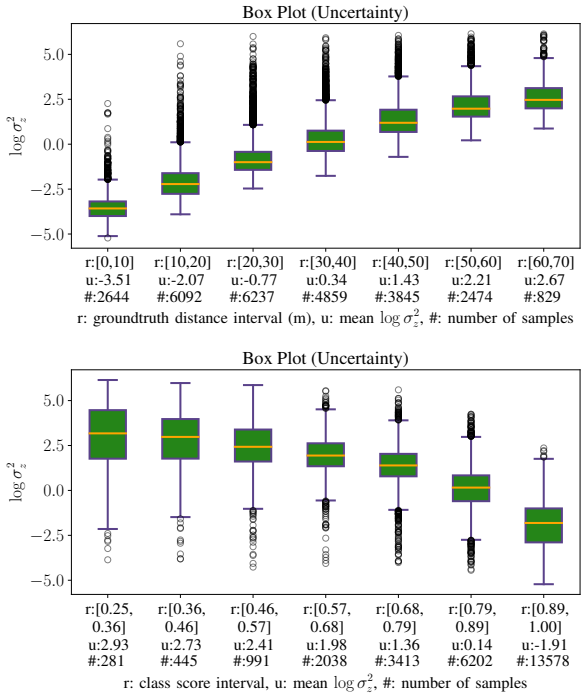

Fig. 6: Boxplots for estimated uncertainty value sigma depending on distance (top) and the predicted class score (bottom).

$L2_{fixed}$). Hereby, for each bounding box prediction of L2 we calculate the distance based on the pixel height and the mean pedestrian height of 1.68 m measured in our dataset (see Fig. 3) using Eq. (2). The distance metric MRE is calculated for the true positives of the complete test dataset (MRE_c) and the small subset (MRE_s) at a fppi of 1.0.

TABLE III: Detection and distance estimation results for pedestrians of the ECP2.5D day test subset. All values are given in percentage points.

Model	LAMR	MRE_c	MRE_s
Base	6.2	-	-
L2	6.8	5.9	15.7
$L2_{fixed}$	-	7.1	47.1

The 2D detection performance represented by LAMR of L2 is reduced by 0.6 in comparison to the Base model. In contrast to [26] our multitask training with weighting of the three tasks based on their uncertainty does not improve results of the single tasks. The L2 MRE_c score of 5.9% is 1.2% better in comparison to the fixed size baseline. Apparently, the network not only relies on a fixed size assumption to estimate distances. This is further backed by the presented MRE_s results. The L2 distance performance for small persons is worse than the performance on the full ECP2.5 dataset, but still significantly better than the fixed size baseline. As small persons are rare cases also in our training dataset, this is a very challenging subset. In Fig. 4 we show qualitative results including instances of the small persons subset. The first sample (left column) includes several persons at varying distances. Our method estimates the distance (and hereby the 3D location) of most of the persons with good accuracy. While the fixed size baseline by design is accurate for average sized pedestrians, it leads to large errors in particular for children. Our network correctly estimates the distance of both children in the first two samples with a low uncertainty. This could be facilitated as they walk right next to other persons. In the third sample the child is partly occluded by the pillar and not in the proximity of another person or objects of known sizes. While the estimated distance is too small, the network gives a higher uncertainty for this challenging scenario. Similar, the uncertainties for the two occluded pedestrians in the last sample are higher than those of the non occluded ones at the same distance. The analysis of the absolute distance error in dependence of the groundtruth distance in Fig. 5 shows a nearly linear correlation. This confirms our expectation for monocular distance estimation and the hypothesis that an absolute matching threshold as in ALP might be inappropriate, due to the high dependency on the data distribution. We analyse the uncertainty output $\log \sigma_z^2$ of the network in Fig. 6. The higher the groundtruth distance, the greater the estimated uncertainty (first boxplot), which reflects the greater absolute distance error for higher distances. Note that in our loss formulation in Eq. (6) we model the distribution for an absolute distance estimation. A relative/normalized

TABLE IV: 3D localization results for pedestrians of the ECP2.5D day test subset. All values are given in percentage.

Model	$LAMR_{3D/0.1}$	$LAMR_{3D/0.2}$	$MRE_{3D/c}$	$MRE_{3D/s}$
L2	37.3	13.1	5.9	15.7
$L2_{fixed}$	51.2	15.9	7.1	47.1

distance estimation formulation would most likely result in constant uncertainty values. The second boxplot shows a lower distance uncertainty for higher classification scores. This might be due to the fact that challenging instances regarding discrimination (e.g. occluded pedestrians) are also more challenging for the distance estimation.

C. Evaluation of 3D localization

Using the camera ray through the bounding box center of each detection according to the estimated distance, we get an estimate $\hat{p}^c$ for the 3D position of the object. We evaluate the MRE for the e_{3D} instead of e_z (see Sec. IV-C). Results are shown in Tab. IV. The resulting numbers for the 3D localization are identical with the distance estimation in Tab. III, as for an accurate 2D bounding box regression the accuracy of the 3D localization only depends on the distance estimate. The LAMR_{3D} values are significantly greater than for the 2D LAMR in Tab. III, because of a large number of 3D estimates that do not fulfil the matching criterion in particular for a low matching threshold of 0.1.

VI. Conclusion

We have presented our new ECP2.5D dataset, which provides 360° LiDAR point clouds as well as 2.5D annotations for 97% of all objects in the reasonable subset of the ECP detection dataset. By publishing this dataset we hope to facilitate advances in the field of monocular, LiDAR, and camera+LiDAR based 3D person localization. We have extended the YOLOv3 2D object detector by a distance regression including uncertainty estimation to serve as a first baseline. The second baseline using a fixed size assumption is outperformed by the network. It is left for future work to apply attention analysis techniques on ECP2.5D to better understand the influence of a person's surrounding for monocular distance estimation, as well as investigate the generalization capabilities of the presented baseline on other datasets. Finally, using the extracted point cloud clusters, the 2D extent and the orientation information already present in the ECP dataset, enclosing 3D bounding boxes could be generated automatically. This would facilitate the comparison with other 3D object detection approaches.

References

[1] Y. Wang *et al.*, "Pseudo-lidar from visual depth estimation: Bridging the gap in 3d object detection for autonomous driving," in *Proc. IEEE CVPR*, 2019, pp. 8445–8453.

[2] F. Manhardt, W. Kehl, and A. Gaidon, "ROI-10D: Monocular lifting of 2d detection to 6d pose and metric shape," in *Proc. IEEE CVPR*, 2019, pp. 2069–2078.

[3] B. Xu and Z. Chen, "Multi-level fusion based 3d object detection from monocular images," in *Proc. IEEE CVPR*, 2018, pp. 2345–2353.

[4] Y. Kim and D. Kum, "Deep learning based vehicle position and orientation estimation via inverse perspective mapping image," in *IEEE Intell. Veh*, 2019, pp. 317–323.

[5] T. Roddick, A. Kendall, and R. Cipolla, "Orthographic feature transform for monocular 3d object detection," in *Proc. BMVC*, 2019.

[6] A. Naiden, V. Paunescu, G. Kim, B. Jeon, and M. Leordeanu, "Shift R-CNN: Deep monocular 3d object detection with closed-form geometric constraints," in *Proc. ICIP*, 2019, pp. 61–65.

[7] L. Liu, J. Lu, C. Xu, Q. Tian, and J. Zhou, "Deep fitting degree scoring network for monocular 3d object detection," in *Proc. IEEE CVPR*, 2019, pp. 1057–1066.

[8] B. Li, W. Ouyang, L. Sheng, X. Zeng, and X. Wang, "GS3D: An efficient 3d object detection framework for autonomous driving," in *Proc. IEEE CVPR*, 2019, pp. 1019–1028.

[9] G. Brazil and X. Liu, "M3D-RPN: Monocular 3d region proposal network for object detection," in *Proc. IEEE ICCV*, 2019, pp. 9287–9296.

[10] A. Simonelli, S. R. Bulo, L. Porzi, M. López-Antequera, and P. Kontschieder, "Disentangling monocular 3d object detection," in *Proc. IEEE ICCV*, 2019, pp. 1991–1999.

[11] Z. Qin, J. Wang, and Y. Lu, "MonoGRNet: A geometric reasoning network for monocular 3d object localization," in *Proc. AAAI*, 2019, pp. 8851–8858.

[12] E. Jörgensen, C. Zach, and F. Kahl, "Monocular 3d object detection and box fitting trained end-to-end using intersection-over-union loss," *arXiv preprint arXiv:1906.08070*, 2019.

[13] L. Bertoni, S. Kreiss, and A. Alahi, "MonoLoco: Monocular 3d pedestrian localization and uncertainty estimation," *Proc. IEEE ICCV*, 2019.

[14] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in *Proc. of the ECCV*, 2018.

[15] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018.

[16] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Adv. in NIPS*, 2015, pp. 91–99.

[17] Z. Cai, Q. Fan, R. S. Feris, and N. Vasconcelos, "A unified multi-scale deep convolutional neural network for fast object detection," in *Proc. of the ECCV*, 2016, pp. 354–370.

[18] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the KITTI vision benchmark suite," in *Proc. IEEE CVPR*, 2012, pp. 3354–3361.

[19] M. Braun, S. Krebs, F. B. Flohr, and D. M. Gavrila, "EuroCity Persons: A novel benchmark for person detection in traffic scenes," *IEEE TPAMI*, pp. 1844–1861, 2019.

[20] P. Sun *et al.*, "Scalability in perception for autonomous driving: An open dataset benchmark," *arXiv preprint arXiv:1912.04838*, 2019.

[21] A. Mousavian, D. Anguelov, J. Flynn, and J. Kosecka, "3d bounding box estimation using deep learning and geometry," in *Proc. IEEE CVPR*, 2017, pp. 7074–7082.

[22] H. Caesar *et al.*, "nuScenes: A multimodal dataset for autonomous driving," *arXiv preprint arXiv:1903.11027*, 2019.

[23] M.-F. Chang *et al.*, "Argoverse: 3d tracking and forecasting with rich maps," *arXiv preprint arXiv:1911.02620*, 2019.

[24] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The KITTI dataset," *IJRR*, 2013.

[25] J. R. van der Sluis, E. A. I. Pool, and D. M. Gavrila, "An experimental study on 3d person localization in traffic scenes," in *IEEE Intell. Veh*, 2020.

[26] A. Kendall, Y. Gal, and R. Cipolla, "Multi-task learning using uncertainty to weigh losses for scene geometry and semantics," in *Proc. IEEE CVPR*, 2018, pp. 7482–7491.

[27] M. Onis, "WHO child growth standards based on length/height, weight and age: WHO child growth standards," *Acta Paediatrica - ACTA PAEDIAT*, 2007.

[28] P. M. Visscher, "Sizing up human height variation," *Nature Genetics*, 2008.

[29] M. Cordts *et al.*, "The Cityscapes dataset for semantic urban scene understanding," in *Proc. IEEE CVPR*, 2016.

[30] K. Wagstaff, C. Cardie, S. Rogers, S. Schrödl, *et al.*, "Constrained k-means clustering with background knowledge," in *Proc. ICML*, vol. 1, 2001, pp. 577–584.

[31] F. Kraus and K. Dietmayer, "Uncertainty estimation in one-stage object detection," in *Proc. of the IEEE ITSC*, 2019, pp. 53–60.

[32] Y. He, C. Zhu, J. Wang, M. Savvides, and X. Zhang, "Bounding box regression with uncertainty for accurate object detection," in *Proc. IEEE CVPR*, 2019, pp. 2888–2897.

[33] Y. Xiang, W. Choi, Y. Lin, and S. Savarese, "Data-driven 3d voxel patterns for object category recognition," in *Proc. IEEE CVPR*, 2015, pp. 1903–1911.

[34] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE CVPR*, 2009, pp. 248–255.

[35] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proc. IEEE ICCV*, 2017, pp. 2980–2988.