Generating 3D Person Trajectories from Sparse Image Annotations in an Intelligent Vehicles Setting

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

***Abstract*— This paper presents an approach to generate dense person 3D trajectories from sparse image annotations on-board a moving platform. Our approach leverages the additional information that is typically available in an intelligent vehicle setting, such as LiDAR sensor measurements (to obtain 3D positions from detected 2D image bounding boxes) and inertial sensing (to perform ego-motion compensation). The sparse manual 2D person annotations that are available at regular time intervals (key-frames) are augmented with the output of a state-of-the-art 2D person detector, to obtain frame-wise data. A graph-based batch optimization approach is subsequently performed to find the best 3D trajectories, accounting for erroneous person detector output (false positives, false negatives, imprecise localization) and unknown temporal correspondences. Experiments on the EuroCity Persons dataset show promising results.**

Multi-Object Tracking, Intelligent Vehicles

I. Introduction

Over the last decades the task of tracking objects has received increased attention in the scientific community. Starting from radar based aerospace and military applications in the early 60s, lately the focus has shifted to visual object tracking. Especially the tracking of a variable and varying number of objects, i.e. multi-object tracking, in a continuous video stream is a key component in many application areas. It has a high relevance in autonomous driving for which a robust and reliable perception of the environment and other traffic participants is crucial. By employing tracking methods, the noisy measurements of different sensors can be integrated over time, to account for uncertainties of the sensor, missed or multiple detections.

Tracking methods have profited from the great improvement in object detection over the last few years. This was mainly caused by the shift to deep learning based methods, which was aided by the release of the ImageNet dataset [8] in 2012 for image classification. Additional new datasets allowed the tailoring to further specific tasks (e.g. pedestrian detection in images in an automotive setting) and extra performance gains by adding more training data. Complementary, the associated benchmarks guide the development of corresponding methods by enabling a common evaluation and comparison.

Recently, an increasing number of publications use deep learning based approaches to tackle the tracking task itself. While the Multi-Object Tracking Challenge (MOTChallenge) [9] does offer a centralized dataset and evaluation benchmark for multi-object tracking, it lacks the size needed for the training of deep learning based methods. Other big datasets are publicly available but not directly applicable for the development of tracking systems for intelligent vehicles, which is the focus of this work, since they do not capture the intelligent vehicle setting, or solely offer temporal sparse annotations without track correspondence information. This also holds for the recently published EuroCity Persons (ECP) dataset [4], which focuses on person detection in street scenes. There, only every 80-th image (i.e. every four seconds) was manually annotated with bounding boxes for all persons. The images have been extracted from continuous recordings in urban areas by a vehicle equipped with a stereo camera, LiDAR scanner, as well as an IMU and GPS sensor. Frame-wise image annotations are labor intensive, expensive, and do scale poorly with an increased number of images. Thus, this manual approach would not be feasible to create a dense tracking dataset for the 50 hours of driving data, that were collected.

Therefore, we present an approach to semi-automatically generate dense 3D trajectories of persons (i.e. pedestrians and riders). Hereby, sparse image ground-truth is aggregated with dense detections and additional sensor data of the intelligent vehicle recording setup. By employing a batch-wise optimization the aggregated data is used to extract the dense 3D trajectories of the persons present in the recordings. Our approach depends on reliable 3D and ego-

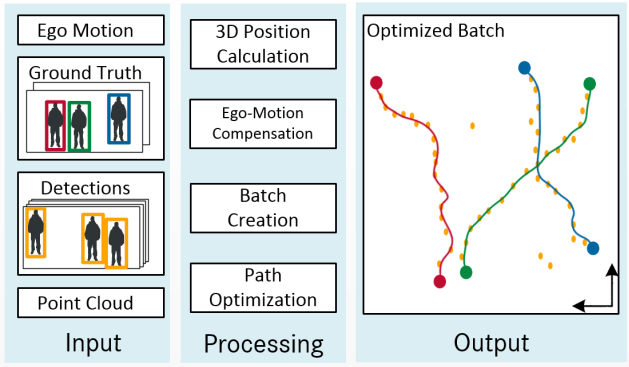

Fig. 1. Schematic depiction of the presented approach. Both the image annotations and detection results are transformed into a 3D position by using the depth information of a LiDAR sensor. For each measurement cycle their positions are compensated by the ego-motion of the vehicle. The ground-truth positions are depicted as bold colored dots, while the intermediate detection results are drawn as smaller orange dots. Using the set of all 3D positions a temporal batch is created and solved by a graph-based optimization method. The final trajectories per object are extracted from the optimization result (solid colored lines).

[†] sebastian.krebs@daimler.com
[1] Environment Perception, Daimler AG, Stuttgart, Germany
[2] Intelligent Vehicles, Technical University Delft, The Netherlands

motion information. Therefore, to perform our experiments we use a filtered subset of the ECP dataset, which satisfies the requirements of our approach (i.e. limited number of occluded objects in uncrowded scenes). For this we get promising experimental results.

II. Related Work

The task of generating annotation to serve as ground-truth data for MOT tasks, was subject to different publications during the last years.

Early public labeling tools, like the VIPER [11] or Labelme [21] software, offered a common annotation process and label data layout. Both tools featured frame-wise annotation of arbitrary objects with polygons (mostly bounding boxes). Since labeling a dense video stream is labor intensive, while differences in the position of objects are only minor, the labeling of objects in key-frames was established. To extract dense annotations for a whole sequence, a constant object motion was assumed, which allowed the calculation of the positions of the polygons in intermediate frames by linear interpolation.

For the creation of the commonly used Caltech Pedestrian dataset [5] a more advanced interpolation scheme was employed, as described in [6]. Dense bounding box positions were calcualted by cubic interpolation. Additional annotations were made based on the quality of the interpolation results.

Since constant velocities of objects in the world do not project to constant velocities in the image plane, [7] performs a cubic interpolation of the objects position in 3D. The image plane positions are used to solve a 3D reconstruction, which results are interpolated and back-projected into the image plane.

Aside from solely using key-frames to perform an interpolation, additional image modalities can be employed to reduce the label effort. The VATIC system introduced in [19] extracts image features (color histograms and HOG) for each object, based on the bounding box information from multiple annotated key-frames. The features are used to train a discriminative classifier, which is applied to the intermediate frames. Subsequently a dynamic programming approach is employed to calculte the final trajectories. The system is further extended in [18], by an active learning task to chose the key-frames and objects which should be labeled by a human annotator.

In [13] a semi-automatic trajectory generation is performed for one person captured in an indoor multi-camera setup. The results of an ensemble of visual person trackers are fused and compared to the output of a reliable person detector extended by a human verification for uncertain estimations.

To create the MARS dataset [23], the detection results of a Deformable Part Model (DPM) detector were linked over time by a Globally Optimal Generalized Maximum Multi Clique (GMMCP) tracker. The linked bounding box locations were smoothed over time to account for localization errors of the detector.

The PathTrack dataset [10] offers bounding boxes and trajectories for persons in dense videos, while minimizing the human annotation effort. By following one object of interest with the mouse cursor during the playback of a video sequence, human annotators were utilized to generate path supervision, which is paired with a set of detections. Based on the path and video-content potential between the path supervision and the detections, and a global trajectory constraint, the final annotations are extracted by a graph-based optimization.

Other than online-based tracking approaches - like the Kalman filter [20] - non-recursive approaches use a whole batch of observations as input to perform the tracking task. In [22], the data association problem for multi-object tracking is formulated as a flow problem and solved for the optimal set of correspondences by a min-cost flow algorithm. To account for occlusion the authors augment the network to include an explicit occlusion model. Following the graph-based formulation, Pirsiavash et al. [14] introduce a greedy successive shortest-path algorithm with a greatly reduced runtime.

Besides solely solving the data association problem Andriyenko et al. [1] formulate multi-target tracking as a discrete-continuous optimization problem. Thereby the data association is solved by using discrete optimization with label costs, whereas a continuous fitting problem is utilized to approximate the individual object trajectories. The discrete optimization is omitted in [12], where a non-convex continuous energy function is designed and minimized to solve the tracking task. The energy function combines partial image modalities in the form of appearance models and explicit object occlusion reasoning, with physical constraints, such as track persistence, mutual exclusion, and target dynamics. To find a strong local minimum of the non-convex function a combination of discrete jump-moves and continuous gradient descent is utilized.

III. Semi-Automatic Labeling Approach

A. Overview

In this section a general overview of the developed system is presented, which is used to generate dense person trajectories. The overall structure of the approach is depicted in Figure 1. The sparse ground-truth annotations of the ECP dataset [4] are used as the starting point. To greatly increase the number of sample points of the trajectories the detection results of a state of the art pedestrian detector are utilized. Both, the ground-truth annotation as well as the used detection method and results are further explained in Section III-B.

The image annotations received either by the person object detector or the ground truth annotation process are reported in the image plane. Those annotations are projected into 3D world coordinates by fusing the visual information with LiDAR pointclouds. The extraction of object 3D positions is further explained in Section III-D.

Thus, at each time-step the 3D position of an object is calculated in relation to the ego vehicle. Since the vehicle

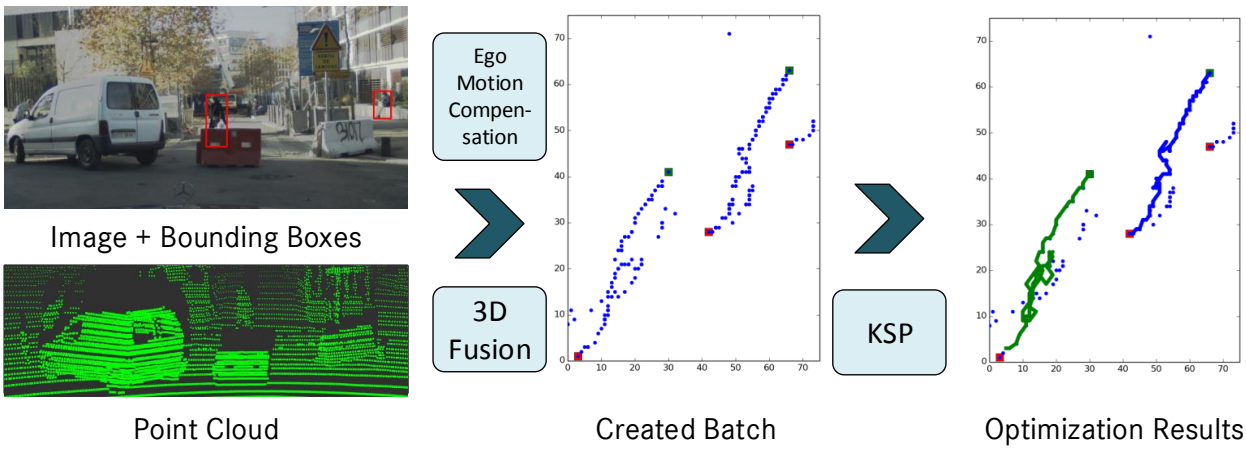

Fig. 2. Visualization of the whole trajectory generation processing chain. Starting with the images and corresponding image annotations, the 3D positions are calculated by incorporating the point cloud information. After the ego-motion compensation, all measurements are located in a common world frame (birds-view). In the middle frame the accumulated batch is depicted. Blue dots represent the fused 3D bounding box position of all intermediate detections, while the ground-truth boxes of the first and last frame are indicated by green and red rectangles, respectively. Lastly, the k-shortest path (KSP) optimization is performed for the created batch to calculate the final tracks (depicted in the right image, different colors indicate different tracks).

itself moves in-between measurement and tracking cycles, the measure object positions would greatly change over time, based on the intrinsic velocity of the object and the velocity of the recording vehicle. To this extent the ego-motion of the vehicle needs to be compensated, such that object positions are reported in a global frame. The ego-motion compensation process is further described in Section III-C.

At this point, for each time-step t there is a set n_t object positions in a fixed world frame (due to the ego-motion compensation in-between cycles). To establish the linking between those positions over the time a constrained flow optimization method is employed, which is further detailed in Section III-E

B. Sparse Ground-Truth Annotations and Detection Results

The ECP dataset [4] introduces a diverse set of person annotations in urban street scenes. These annotations are used as the main source to extract the object trajectories, thus a short overview of the annotations process and the labeled properties is given.

For each image a human annotator indicates the presence and position of all objects of interest. The labeled objects include pedestrians and riders, while for the latter the ride-vehicle type (bicycle, wheelchair, scooter, buggy, motorbike, tricycle) is also annotated. Each object is captured by a tight fitting bounding box in the image, in the presence of occlusion the full extent of the object is estimated. Furthermore, for each object the occlusion and truncation level is indicated as a discretized value and the body orientation of the person in provided. Additionally, all objects which either fall below a minimum size, or are not precisely identifiable (e.g.: in big groups) are captured by an ignore region.

While the original annotations offer a precise localization, they are provided in a temporal sparsely manner (every 4 seconds) and lack any correspondence information.

To overcome the sparsity of the provided annotations and increase the number of sampling points for the trajectory extraction, the results of a person detection algorithm are utilized. As reported in [4], the best detection results on the ECP dataset were achieved with the optimized Faster R-CNN [15], using the VGG-16 [16] as base network. The Faster R-CNN network implements an internal region proposal network and combines it with a subsequent detection and classification network. Thus, the two-stage network is learnable in an end-to-end fashion. The applied optimization to the original network include a modifications of the anchor-boxes used during the proposal generation stage, as well as the integration of the ignore regions during training.

The final detection performance of the Faster R-CNN in terms of missed vs. false positive detections depends on the chosen detection score threshold. In this work we apply a score threshold that corresponds with 0.7 false positives per image and a miss rate of 0.027 on the ECP test dataset.

C. Ego-Motion Compensation

To perform the ego-motion compensation two different sensors are utilized to measure the needed kinematic information of the car. On the one hand, a GPS/INS system, which internally pairs a highly accurate internal measurement unit with a GPS receiver, is used to measure linear accelerations, angular velocities, absolute angles, and the GPS position of the car. The GPS/INS system combines these information using an integrated Extended Kalman Filter and outputs the fused estimated quantities. Since the GPS reception can be impaired in urban areas - especially in the presence of street canyons - this can lead to an imprecise estimation of the linear velocities. To account for these aspects the internal car sensors are utilized as well. By using the information from the wheel speed sensors along with an Ackermann steering model, the current velocity in the heading direction as well as

the yaw rate are calculated. These values are used to detected and compensate failure cases of the GPS/INS system.

Between two time frames the ego-motion of the vehicle is calculated in a XY driving plane, which is assumed to be fixed in between tracking cycles. To calculate the change in position (X and Y) of the car, the velocity v in driving direction, the yaw rate $\dot{}$ (rotation around Z axis), and the time delta $t = t \quad t_{\ 1}$ need to be known. By assuming a fixed velocity and yaw rate, the change in angle and position can be calculated by:

$$' = \dot{} \quad t \tag{1}$$

$$x = \frac{v}{\dot{}} \ \sin(\quad ') \tag{2}$$

$$y = \frac{v}{\dot{}} \ (1 \quad \cos(\quad ')): \tag{3}$$

The change of the vehicle position and angle can be expressed as the affine transformation

$$T_{ego} = \begin{bmatrix} \cos(\quad ') & \sin(\quad ') & x \\ \sin(\quad ') & \cos(\quad ') & y \\ 0 & 0 & 1 \end{bmatrix} : \tag{4}$$

Finally the compensated X and Y position of each object can be calculated by:

$$\begin{bmatrix} x_{comp} \\ y_{comp} \\ 1 \end{bmatrix} = T_{ego}^{\ 1} \begin{bmatrix} x_{old} \\ y_{old} \\ 1 \end{bmatrix} : \tag{5}$$

The ego-motion compensation is depicted in Figure 3.

D. 3D Position Extraction

Based on the detections and ground-truth annotations in the image the corresponding position of each object in world coordinates is calculated. Each object in the image is specified by a tight fitting bounding box $bbox = (u; v; w; h)$ which estimates the full extent of the object in case of occlusion. Given the location in the image plane, the distance to the object needs to be known to calculate the position in the world coordinate frame. Therefore, the point cloud information of a Velodyne HDL-64 LiDAR sensor is used, which was recorded along with the camera images. The sensor set of the vehicle used to perform the data recording was calibrated. First, the camera was calibrated to obtain its intrinsic camera parameters, like the optical center, distortion coefficients, and the focal length. Furthermore each sensor (camera, LiDAR, GPS/INS) was calibrated extrinsically. Thus, the position (translation and rotation) of all sensors is known in relation to the origin of the recording vehicle.

This allows to transform a measured point from the coordinate system of one sensor into the measurement space of another sensor. To extract the depth information for the pedestrians in the image, the entire point cloud is projected into the image coordinate frame. The transformed 3D points are projected onto the two dimensional image plane, by using the pin hole camera model. After the transformation and projection the measured point clouds align with the image. To account for the movement of the ego vehicle during the recording of the point cloud, an ego-motion compensation is performed. A full scan (360°) of the LiDAR comprises a fixed number of slices measured at equidistant angle resolutions. The movement of the car between each slice is calculated analog to the ego-motion compensation performed for the object positions. Finally, to calculate the distance of an object, all transformed and corrected LiDAR points within the bounding box are utilized. The median of the corresponding distances of the selected points is used as the distance.

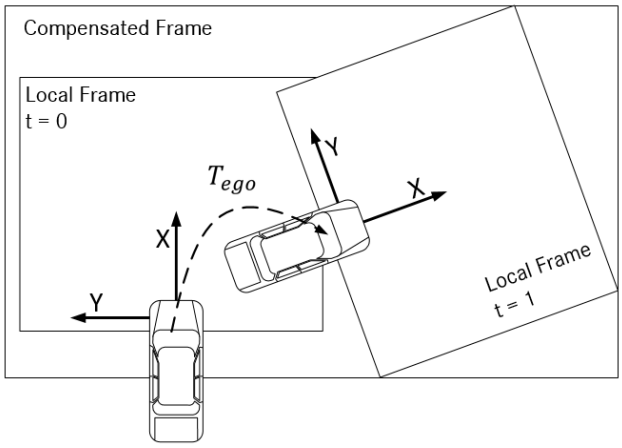

Fig. 3. Schematic depiction of the ego-motion compensation. At each time, the measurements are received in a local coordinate system. By applying the inverse of the ego movement, the measurements can be reported in the previous coordinate frame.

E. Path Optimization

By transforming all ground-truth annotations and detections into a global world frame and correcting them by the ego-motion of the vehicle - as described above - for each time step there is a set of object positions. To find the correct correspondences between these positions, which are needed to extract the final object trajectories, the k-shortest path (KSP) approach presented by Berclaz et al. [3] is adopted. Finding the correct objects paths over time is formulated as an Integer Programming (IP) problem.

The algorithm operates by discretization both in the temporal and spatial domain. As input for each time-step a probabilistic occupancy grid map (POM) is used. The POM is a two dimensional representation of the XY world plane, with a given number of equally spaced cells in X and Y. Each cell holds the probability of the presence of an object at this location. The final representation of a sequence is a three dimensional graph (two spatial dimensions, and the time), where the nodes indicate the position and presence of an object, and the edges indicate object movement. The optimization is performed in a batch wise manner. Thereby a batch is constructed between two ground-truth frames, and the detection responses in between those two frames are used as input for the path optimization.

Based on the constructed graph, the path optimization works as follows: Let the number of time instants in the

batch be denoted by T and the number of possible object locations for each map by K. Further the set of all possible locations an object can reach at time $t+1$ given its location k at t is denoted by $N(k) \quad 1;:::;K$. Each edge $e^t_{i;j}$ indicates a possible object movement from i at time t to j at time $t+1$, implying that $j \ 2 \ N(i)$. Lastly, the actual number of objects moving from i to j is represented by the discrete variable $f^t_{i;j}$.

Based on these definitions the following constraints are formulated. For each t, the sum of outgoing flows $f^t_{j;k}$ of a location is equal to the sum of incoming flows $f^t_{i;j\ 1}$ to this location at the previous time. Furthermore, a node can not be occupied by more than one object at a time. Lastly, there can be no negative flows.

The appearance and disappearance of objects in the batch is enabled by a source and sink node. All flows need to emerge from the source node, and ultimately end up in the sink node. Given these definitions , and since object movement can only occur from $t \ _1$ to t (objects only move forward in time), the graph is directed and acyclic.

Thus, to find the set of correct correspondences between object locations over time for one object translates into finding the best path in the graph. Finding this solution can be expressed as an Integer Program - maximizing the flows in the network - which could be solved by a generic Linear Programming solver. By leveraging the directed acyclic graph representation, the Integer Program can be reformulated as a k shortest node-disjoint paths problem which drastically reduces the complexity. By using the iterative disjoint path algorithm [17] the k-shortest paths in between the source and sink node are extracted, which is the optimal solution to solve the assignment problem.

IV. Experiments

The semi-automatic labeling approach is applied to the ECP dataset [4], which was recorded from a moving vehicle. The recording setup of the vehicle includes an automotive-grade two mega pixel camera (1920 1024 pixel), with 16 bit color-depth, a state-of-the-art Velodyne HDL-64 LiDAR sensor, as well as an ADMA GPS/INS sensor. To allow fusing different sensor modalities, an advanced time synchronization approach was developed and employed. Additionally, an accurate intrinsic and extrinsic sensor calibration was performed.

To capture natural, complex, and diverse motion patterns of persons, the data was recorded in urban areas in 31 different cities in 12 European countries.

As described in Section III the extraction of trajectories is performed in a batch wise manner. Each extracted batch has a length of four seconds, since it starts and ends with a key-frame, for which the ground truth annotations of the ECP dataset are available. All 79 images in between the key-frames, are used to generate detection results.

Given the prerequisites of our system, we employ a pre-filtering of the dataset, based on the ground-truth information in the key-frames of each batch. First, to prevent for erroneous trajectories caused by incorrect 3D estimations for heavily occluded persons (i.e. more than 40% occlusion), batches are omitted which contain several occluded persons in one of the key-frames. Secondly, batches are filtered out if they contain anomalies in the GPS position, to avoid incorrect ego-motion compensations. Such anomalies are detected by sudden changes in the GPS position and velocities reported by the GPS/INS sensor, which do not align with the measured velocity of internal car sensors.

In total 95 recordings were used, from which 17644 batches could be extracted. After filtering based on the ego-motion quality and the number of occluded objects, 4537 batches were used to perform the trajectory generation. Overall, 21193 person trajectories were extracted. The average trajectory length is 39.6 frames. The whole processing pipeline is depicted for one example batch in Figure 2, while example results of different recordings are shown in Table I.

As a baseline a linear interpolation approach is employed, which is still widely used for the creation of tracking dataset annotations. The results are depicted as white boxes in the intermediate frames in Table I. The presented approach clearly outperforms the interpolation results in terms of location and size accuracy.

V. Conclusion

In this paper a novel approach was presented, which leverages the intelligent vehicle setup, sparse 2D image annotations and detected bounding boxes to calculate dense 3D person trajectories. This can be used to significantly reduce the label effort needed for the creation of a large scale dataset for mulit-object tracking.

First experiments based on the ECP dataset, show promising results. Compared to the linear interpolation of image annotations, the presented results better capture the position of the objects in intermediate frames.

Future work will include incorporating object association confidences based on image features in addition to position information in the graph-based optimization, similar to [2], to better deal with partial occlusions, and a more quantitative evaluation.

References

[1] A. Andriyenko, K. Schindler, and S. Roth. Discrete-continuous optimization for multi-target tracking. In *CVPR*, pages 1926–1933, 2012.

[2] H. Ben Shitrit, J. Berclaz, F. Fleuret, and P. Fua. Tracking multiple people under global appearance constraints. In *ICCV*, pages 137–144, 2011.

[3] J. Berclaz, F. Fleuret, E. Turetken, and P. Fua. Multiple object tracking using k-shortest paths optimization. *TPAMI*, 33(9):1806–1819, 2011.

[4] M. Braun, S. Krebs, F. Flohr, and D. Gavrila. EuroCity Persons: A novel benchmark for person detection in traffic scenes. *TPAMI*, 2019.

[5] P. Dollár, C. Wojek, B. Schiele, and P. Perona. Pedestrian detection: A benchmark. In *CVPR*, 2009.

[6] P. Dollár, C. Wojek, B. Schiele, and P. Perona. Pedestrian detection: An evaluation of the state of the art. *TPAMI*, 34, 2012.

[7] P. Gil-Jiménez, H. Gómez-Moreno, R. López-Sastre, and S. Maldonado-Bascón. Geometric bounding box interpolation: an alternative for efficient video annotation. *EURASIP Journal on Image and Video Processing*, 2016(1):8, 2016.

[8] A. Krizhevsky, I. Sutskever, and G. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, pages 1097–1105, 2012.