DEPARTMENT OF INFORMATICS

TECHNISCHE UNIVERSITÄT MÜNCHEN

Guided research report

Direct object tracking

Nikita Korobov

DEPARTMENT OF INFORMATICS

TECHNISCHE UNIVERSITÄT MÜNCHEN

Guided research report

Direct object tracking

Direkte Objektverfolgung

Author:	Nikita Korobov
Supervisor:	Prof. Dr. Daniel Cremers
Advisor:	Nikolaus Demmel, Dr. Aljosa Osep
Submission Date:	Submission date

I confirm that this guided research report is my own work and I have documented all sources and material used.

Munich, Submission date Nikita Korobov

Abstract

We present a method for object tracking in 3D and 2D domains using direct sparse odometry (DSO). Additionally, the benchmark for 3D multi-object tracking performance evaluation using GIoU and HOTA metrics is introduced. Firstly, the depth and 2D instance segmentation masks are obtained for images. The different instances are initially tracked in 3D using a depth map and direct image alignment. Then, the tracked instances are associated in the 2D domain into tracklets. The poses of tracklets in 3D are then refined using the DSO method. The DSO is applied only to the pixels inside the segmentation mask for the particular tracked object. Then, object 3D pose and object 3D bounding box dimensions are derived from the accumulated sparse point cloud obtained from the DSO. Thus, temporal information in the DSO helps both: object detection and pose estimation. For the proper and accurate testing of multi-object tracking in 3D the HOTA metric is used with extensions of GIoU similarity score. In the experiments on the KITTI dataset, the proposed method achieves the performance of the others SOTA 3D MOT competitors.

1 Introduction

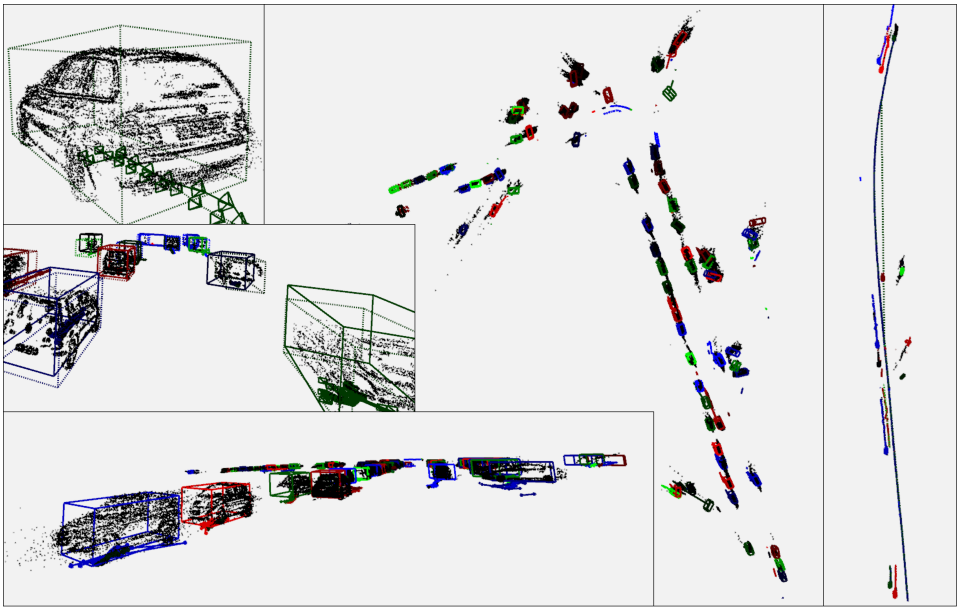


Figure 1.1: Qualitative results of our 3D object tracking system. Different colors mean different objects. Point clouds are accumulated to the last tracked state. Right: Results of tracking of moving cars in the global world frame. Middle: Global scene with the parked cars on the sidewalks in the global world frame. Left: Detailed images of accumulated sparse point clouds.

Multiple object tracking (MOT) and object detection in 3D are two important tasks in autonomous driving, robotics, AR/VR and computer vision. For example, in autonomous driving, a self-driving system has to accurately detect and temporally track surrounded objects on the road to ensure safety. Different sensors such as LiDAR, camera, radar, etc. may be used (and even fused) to achieve the best object detection and tracking performance. However, monocular and stereo cameras are cheap and small, so can be easily mounted on robotics and/or VR/AR platforms. Most of the current approaches do not focus on the accuracy of both detection and tracking of multiple objects in 3D. Thus, in the current work, we focus on the 3D MOT and the fair evaluation of the accuracy of this task.

For the 3D MOT many algorithms use the single-frame detection of the objects, either on images, or point clouds using the learned and/or standard detectors. The following tracking of the detected objects is performed by Kalman filter or SLAM. However, the single view object detection sometimes is not accurate, because the object may be occluded and/or

truncated. Additionally, the recall of the 3D object detectors is usually lower than the recall of the 2D instance segmentators.

Some of the 3D MOT algorithms use the SLAM for the ego-camera pose estimation coupled with the dynamic and static objects tracking [1]. It is shown that this approach improves the accuracy of the ego SLAM, by excluding from the consideration for the optimization of the dynamic objects, that usually introduce large residuals into the standard SLAM system [2]. Some of the algorithms use the 3D tracking task internally to solve the 2D MOT task [3]. However, the accuracy of the 3D MOT is not studied well. Moreover, the complex optimization structure in such systems (e.g. the factor graph) for the only 3D MOT task seems superfluous, because the object trajectories do not depend on the static background and the other dynamic objects. So the tracking of the objects can be decoupled with the ego SLAM.

In this work, we propose a system to detect and track multiple objects in 3D. Given 2D instance segmentation masks, the object hypothesis are initially tracked, associated and the trajectories are refined by the Direct and Sparse odometry. Then, the object detection in a sparse point cloud in 3D is performed for each tracked object. The temporal data is used in two places:

1. to refine sparse point cloud and trajectory in the visual odometry system;
2. to accumulate multiple point clouds of one object into one point cloud to get better object detection.

Finally, the evaluation metric for the 3D MOT task is proposed to evaluate the tracker performance. In summary, the contributions of the current work are as follows:

1. The 3D object tracker using the direct visual odometry method;
2. The metric for the 3D MOT evaluation;
3. The proposed method achieves near state-of-the-art results in the 3D MOT task.

The rest of the report is organized as follows: The related works are reviewed in section 1.1. In the chapter 2 the proposed method is described. Then, the problems of the current 3D MOT evaluation metrics are analyzed and the new 3D MOT metric is proposed in chapter 3. The experiment results are provided in chapter 4. At the end the conclusions and discussions are given in the chapter 5.

1.1 Related works

1.1.1 Single frame 3D object detection

3D object detection is a much more challenging task than the object detection in 2D, because of the higher dimensionality of the object. Degrees of freedom of the bounding box in 2D is 5 (x, y, width, height, rotation), in 3D is 9 (x, y, z, width, height, length, yaw, pitch, roll) or 7 in the autonomous driving scenario (only one angle - yaw). The learned-based detectors are

becoming state-of-the-art in the recent years. The most accurate detectors are those learned from LiDAR point clouds [4, 5, 6], sometimes the images are used to add the additional information into the detector [5, 6]. These detectors may be reused for the object detection in depth map from stereo images (or from monocular image). The depth map may be converted to the LiDAR data format and the proven SOTA detectors on LiDAR point clouds are used for the detection task [7]. Some other works use the CAD models as the prior shape of the objects to detect the objects on stereo cameras [8], the others use segmentation masks to help the object detector to generate proposals, so achieve the higher recall [9].

1.1.2 3D object tracking

The algorithms of object tracking in 3D can be divided into two main groups: appearance-based and model-based. The appearance-based trackers use the appearance of the tracked objects (either 2D image appearance or 3D point clouds) to perform association of the detections between frames. The most similar to our work takes the 3D detections combined with 2D image appearance, the objects are tracked with Kalman filter in 3D-2D space [10]. A similar work [3] tracks the segmented instances on images with optical flow and performs the 3D dense object reconstruction at the same time. The association is done with the help of the motion model in the 3D world frame. So many of the id switches are resolved by this strategy. Some methods use deep learning to perform the association of the objects based on their visual appearance and their bounding boxes [11]. However, neither of these methods focuses on the 3D MOT task and reports the results of the evaluation of the 3D MOT.

Model-based trackers usually take the detections as the input and only track them based on some motion models. In the recent work, [12] the Kalman filter and motion model was used to track provided 3D detections. The performance of these methods is limited to the quality of the detections and needs to be significantly modified to consider the other sources of the information.

1.1.3 3D MOT evaluation metric

One of the most popular MOT metrics is CLEARMOT [13]. This metric together with the 3D similarity scores (e.g. 3D IoU) can be used for the evaluation of the 3D MOT task [14]. The extension of the CLEARMOT metric was proposed by Weng, Xinshuo, et al. [12]. In their proposed metric authors take into consideration the confidence scores of the detections. The new integral metrics are introduced to summarize the performance of the MOT system over the different confidence thresholds. Some of the works use the CLEARMOT for 2D and the object 3D localization error (as pure translational error or 3D IoU) to show the tracking performance [10, 1].

2 Proposed method

2.0.1 Notation

Throughout the paper, $\mathbf{I}_N$ represents the sequence of N temporally ordered images $\{I_1, .., I_N\}$. Italic capital letter $\mathcal{M}$ denotes the binary mask of the image and we will use the notation of $\mathcal{M}(I)$ to denote the unmasked pixels of image I.

$\mathbf{T}_{A_i} \in SE(3)$ denotes the pose of object A at time t_i. $\mathbf{T}^C_{A_iB_j} \in SE(3)$ represents the transformation from frame B at time t_j to frame A at time t_i relative to fixed frame C.

The tracked object is defined as $O^j_i = \{j, i, \mathbf{T}^O_{C_{ref}C_j}, w, h, l\}$, where j is number of the current frame, i is the object id, $\mathbf{T}^O_{C_{ref}C_j} \in SE(3)$ is the transformation of the camera pose $\mathbf{T}_{C_j}$ at t_j to the camera pose $\mathbf{T}_{C_{ref}}$ at some reference time t_{ref} with respect to moving (w.r.t to background) object frame O. Object id is unique over all object ids for one particular frame. The set of the objects with the same id $\{O^j_k\}$ for all frames $j \in 1, ..N$ is called tracklet $\mathbb{T}_k$.

We denote the reprojected point $\mathbf{p}\prime$ of point $\mathbf{p}$ for cameras with relative transform $\mathbf{T} \in SE(3)$ as

$$\mathbf{p}\prime(\mathbf{p}, d_p, \mathbf{T}) = \pi(\mathbf{T} \times (\pi^{-1}(\mathbf{p})^T d_p, 1)^T), \tag{2.1}$$

where $\pi : \mathbb{R}^3 \to \Omega$ is projection function and $\pi^{-1} : \Omega \to \mathbb{R}^3$ is back projection function. Both functions depend on intrinsic camera parameters (that are considered to be known), d_p is depth of the point $\mathbf{p}$.

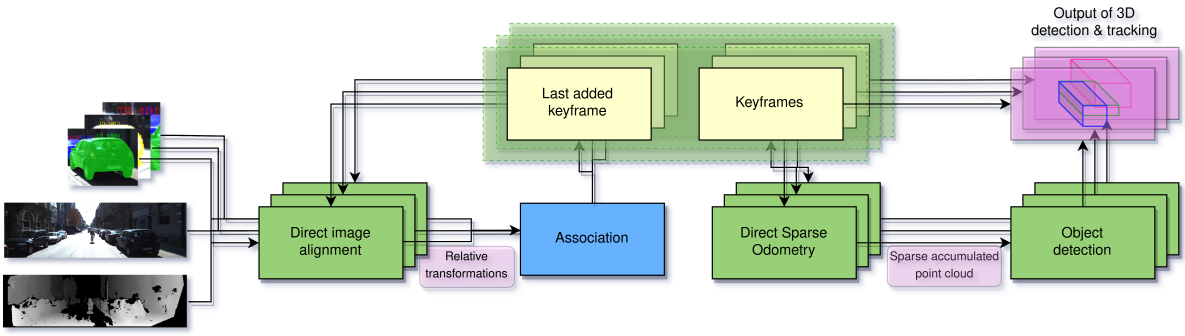


Figure 2.1: Overview of the proposed system. The green color of the rectangles means that the algorithm runs for each instance on the frame/each tracked object. Blue color means that the step is executed for all the objects.

2.1 Object tracking

The proposed method consists of two stages: object tracking and object detection in the resulting per object point cloud. The method gets as the input image sequence $\mathbf{I}_N$, the depth maps and instance segmentation masks for each image in the sequence. The output of the system is the set of all the possible tracklets for the provided sequence. The full system overview is shown in Figure 2.1.

Firstly, the Direct Image Alignment (DIA) is performed for each instance on the current frame to initially track the relative transformation of the object hypothesis between the frames. Then, the found optimized pose is used to wrap the segmentation mask to the previous frame. The object association is performed on the image domain with help of the Hungarian algorithm [15], where the similarities are measured as the IoU of the wrapped instance masks from the current frame and the instance masks from the previous frame. If the DIA fails, then the wrapping of the instance masks from the current frame is performed with help of the averaged optical flow. In this case, all the active keyframes of the DSO for the object are archived and tracking is performed in 2D until the tracking in 3D becomes successful again (example on the Figure 2.2 and Figure 2.3). Thus we track in 3D opportunistically: try to track in 3D every frame, but if it fails, then track in 2D. It coincides with the way of human way of tracking objects: if the object is too far away and too small, it is extremely hard to accurately estimate the relative pose change. However, it is still possible to associate detections.

If the detection is missed, we continue tracking the object during several frames either in 2D or in 3D wrapping the last observed segmentation mask (we will call the part of the track that does not have detections "ghost" track). If the segmentation mask of the "ghost" object has 2D IoU with the new detection higher than some threshold, then the "ghost" track is considered as "real" and the tracking continues. Otherwise, if the "ghost" object does not have the observation of the "real" detection for $N_{wo\_detection}$ frames (or the "ghost" tracking fails), the "ghost" track is discarded and the object is considered to be lost. This strategy helps to connect the part of the tracklets in the case of the missing detections on the segmentation mask level.

After the association, the frame is added as the keyframe candidate to the Direct Sparse Odometry (DSO). For each object instance exists independent from the other instance of the DSO, so each object is tracked independently, without knowledge about the background and the other objects.

After the DSO stage, the sparse point clouds of each keyframe of the DSO are accumulated into one point cloud and the object detection is performed. The output pose of the object needs to be expressed with respect to some fixed world frame. Keyframe to keyframe tracking in DSO gives us the pose relative to the first frame as $\mathbf{T}^{O}_{C_iC_1}$ tracked against fixed frame O. For object tracking, the object frame should be connected to some meaningful point on the object (e.g. center of the bounding box that encloses the object). This pose we obtain from the object detection at the first frame as $\mathbf{T}_{C_1O_1}$. Thus, the object pose w.r.t to some fixed world coordinate can be expressed as:

$$\mathbf{T}_{WO_j} = \mathbf{T}_{WC_j}\mathbf{T}^{O}_{C_1C_j}{}^{-1}\mathbf{T}_{C_1O_1} \tag{2.2}$$

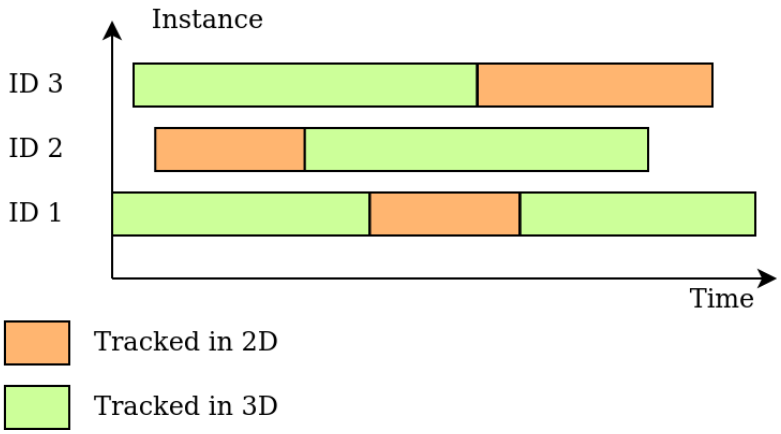


Figure 2.2: The example of the tracking order for different instances. I.e. the tracking is not lost if the depth information is not good enough to track in 3D.

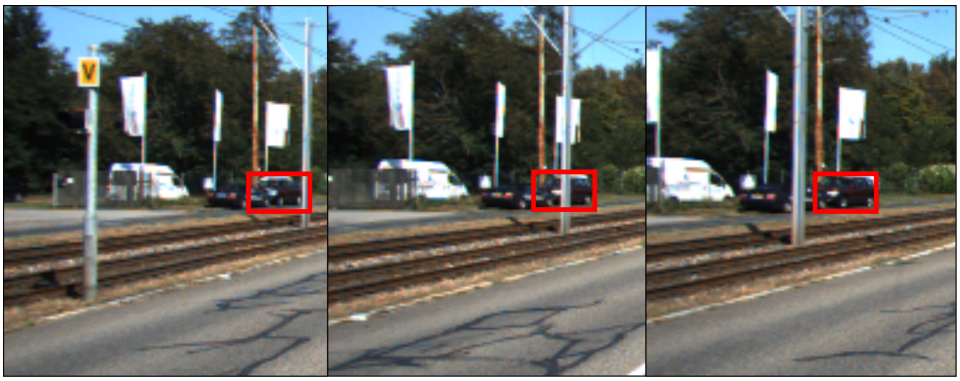


Figure 2.3: The example of 3D to 2D and back to 3D tracking. Left image: the car in the bounding box is tracked in 3D for several frames. Center image: the car in the bounding box is occluded by the pillar and the average residual of the DIA is high, so the DIA fails and tracking is performed in 2D while the car is occluded. Right image: the car in the bounding box is tracked in 3D again. Important that the tracked ID of the car remains the same even though the tracking in 3D fails for the occluded objects.

The accumulation of the point clouds over the several keyframes into the one-point cloud helps the object detection algorithm to detect more accurately the partially visible objects. For example, in the beginning, the object is seen from one side, but then it rotates w.r.t. the camera and it is seen from the other side. The accumulation of the point clouds gives the object detection algorithm an advantage in some cases compared to one-view detectors.

We define two modes of the system: offline and online. In the online mode, the reference pose $\mathbf{T}_{C_1O_1}$ is updated for each and only new keyframe. In the offline mode, the reference pose is also updated, but this update is propagated to the all previous keyframes. So, in the offline mode the object is detected in the point cloud accumulated from all the frames in the tracklet (so the object is seen from all the possible views) and the reference pose is set to all the previous keyframes.

2.1.1 Direct Image Alignment

The DIA is used to find the relative pose of the object between the two frames I_i and I_j, w.l.o.g. let us assume $t_j > t_i$. Note, that only instance segmentation mask $\mathcal{M}_i$ and depth map D_i at frame i are required for DIA.

DIA is formulated as the optimization task, where the minimization energy is formulated as following:

$$E(\mathbf{T}) = \sum_{\mathbf{p} \in \mathcal{M}_i(I_i)} \left\| \mathcal{M}_i(I_i)(\mathbf{p}) - I_j(\mathbf{p}\prime(\mathbf{p}, D_i(\mathbf{p}), \mathbf{T})) \right\|_{\mathcal{H}}, \tag{2.3}$$

where the $\|\cdot\|_{\mathcal{H}}$ is the Huber norm.

So, the DIA finds the relative transformation between images I_i and I_j that minimizes the intensity difference between the intensities of the pixels $\mathbf{p} \in \mathcal{M}_i(I_i)$ belonging to one object on the masked image $\mathcal{M}_i(I_i)$ and the pixels $\mathbf{p} \in \mathcal{M}_i(I_i)$ reprojected to the image I_j.

Since this optimization task is non-convex and the convergence is not guaranteed we run the DIA several times for one object on the image pyramid: from the coarse resolution to the high resolution using the pose found on the previous level of the image pyramid as the initialization hypothesis for the optimization on the current level.

The initial relative transformation for DIA comes from the RANSAC tries of different transformations. The transformation with the minimal cost 2.3 on the coarsest level of the image pyramid is chosen as the initialization. The space of allowed transformations is dictated by the specifics of the dataset: in the autonomous driving scenario objects are assumed to move parallel to the $x - z$ plane (in the camera frame) with only rotation around y axis.

The DIA is considered to be failed if any of the following conditions are true:

1. Number of object pixels is less than some threshold N_p;
2. Mean cost per pixel is higher than threshold C_p;
3. Length of the translation of the optimized relative pose $\mathbf{T}$ is higher than threshold L. This rule is specific for the dataset and defined by the maximum speed of the objects in the camera frame and the time difference between two consecutive images I_i and I_{i+1}.

2.1.2 Direct Sparse Odometry

The DSO is inspired by the work of Engel et al. [16] with the following differences:

1. Affine brightness function is not optimized. Instead, denoting the arithmetic mean of all pixel intensities in the pattern patch on the host frame $\bar{I}_i$ and target frame $\bar{I}_j$, the multiplication factor $\frac{\bar{I}_j}{\bar{I}_i}$ is introduced into residual to neutralize any of the multiplicative changes in the host and target frames. Thus, the residual is $r = I_j(\mathbf{p}\prime) - \frac{\bar{I}_j}{\bar{I}_i} I_i(\mathbf{p})$;
2. Marginalization of the frames is not used for the simplicity of the algorithm;
3. Initial frame tracking is performed by the DIA step with known a priori depth. Since the objects (cars and pedestrians) in the autonomous driving scenario have poor texture, the initial tracking is a complex task, that, however, need to be accurate;
4. Keyframe deletion strategy is "the oldest keyframe".

2.2 Object detection

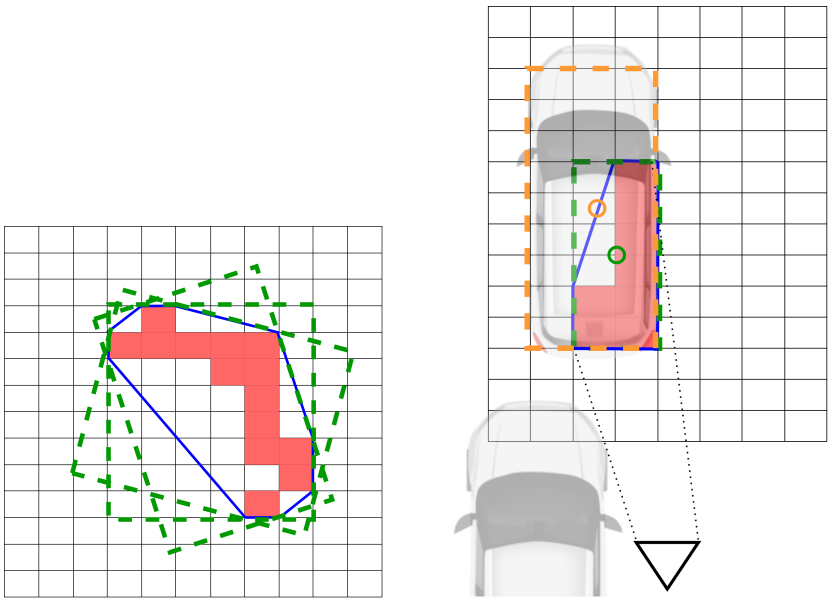
The input of the object detection algorithms is the point cloud (PC). In our task PC is sparse and obtained from the DSO for an individual object, so the existence of the object in the PC is guaranteed. The task is to derive the bounding box around the object out of the PC.

2.2.1 Convex hull bounding box regression

This method is dictated by the nature of the autonomous driving scenario that is in the focus of the current work. The specific of the objects in the car scenario is that all of them are located on the road and the only available rotation is rotation around y axis in the camera frame. This allows to carry out the detection at the bird's eye view of the PC. Then the height of the bounding box can be taken as $h = y_{1-\alpha} - y_{\alpha}$, where y_{α} is the α percentile of all the numbers in the sorted array of y coordinates.

The PC is projected top-down on the bird's eye view and divided into the 2D grid. The state of the grid cell is set to "occupied" if the number of points falling into the cell is above some threshold T_p, "free" otherwise. Then the grid is filtered with the connected component filter by the area of the components to get rid of the outliers. The cells with "occupied" are extracted from the resulted grid and used to construct the convex hull. The number of all the possible circumscribed around convex hull 2D rotated bounding boxes is finite (and equal to the number of the edges of the convex hull). All the circumscribed bounding boxes are used as the hypothesis and scored by the average 2D distance of the points in the original grid to the bounding box hypothesis. The hypothesis with the lowest score is chosen as the coarse regressed bounding box. The example of the intermediate step of the detection algorithm is shown in Figure 2.4a.

If the resulted bounding box dimensions are too small compared to the hard-coded, dataset dependent sane dimensions of the bounding box, the size of the bounding box is increased up



(a) Bounding box regression example

(b) Example of the bounding box size correction

Figure 2.4: Red - occupied cells, white - unoccupied cells. Blue line - convex hull. Left: example of the bounding box hypothesis generation using convex hull algorithm. Green dashed rectangles - several bounding box hypothesis, that are scored according to the average distance of the points in PC to bounding box hypothesis. Right: example of the bounding box size correction. black triangle - camera position. The green dashed rectangle - the best-scored bounding box hypothesis. The orange dashed rectangle - bounding box after the size correction. Circles represent the centers of the bounding boxes with respectful color. Notably, the center of the corrected bounding box is moved from the camera origin, such that the closest to the camera bounding box corner remains in the same place.

to the minimal dimensions of the bounding box. The center of the bounding box during this increasing is moved away from the camera such that the closest to the camera corner of the bounding box stays on the same place (Figure 2.4b) (the other case when the bounding box edges are parallel to the axis if the camera frame, the center is moved such that the closest to the camera edge stays on the same place). The reason for that is that the unseen parts of the object are not located closer to the camera. So if the real object is larger than the detected bounding box, the unobserved parts are hidden further from the camera. If the resulted bounding box is too big, then the size of it is decreased down to some maximal predefined size.

3 Generalized 3D HOTA metric

One of the most popular metrics for the MOT evaluation is MOTA [13]. However, it has been shown by Jonathon, Luiten, et al. [17] that MOTA has many drawbacks. Some of them:

1. Detection performance significantly outweighs association performance;
2. Association errors only take into account the short term associations;
3. Fixed threshold of the similarity score to count detection as the TP.

Instead of MOTA authors proposed a HOTA metric [17]. The HOTA metric solves many of the problems with MOTA that are discussed above. In particular, it measures the association accuracy $AssA$ as well as the detection accuracy $DetA$. The resulted HOTA metric is the geometric mean of the $DetA$ and $AssA$ integrated over the whole range of the similarity score thresholds.

$$HOTA = \int_0^1 \sqrt{AssA_\alpha * DetA_\alpha} d\alpha \tag{3.1}$$

The integration over the range of the similarity scores is especially important in the task of 3D MOT, as the localization error increases with the distance from the sensor (this error is significantly notable for the stereo cameras). So, the object may be detected, but the detection may be biased from the ground truth and the MOTA metric with a fixed threshold will count this object and ground truth as false positive and false negative respectively.

The problem described above applies even to HOTA metric if the bias of the detection from the ground truth such large that the intersection between detection and ground truth is equal to 0 and as the consequences, the Intersection over Union (IoU) metric (that is usually used to compute the similarity score) is equal to 0. The problem with the IoU metric is that it does not consider the distance between bounding boxes. This problem is addressed by Rezatofighi, Hamid, et al. [18]. The authors proposed Generalized IoU (GIoU) as the extension of IoU that takes into account the geometric distance between boxes. Notably, GIoU is not zero even if the intersection of the bounding boxes is the empty set. GIoU is formulated as following:

$$GIoU = \frac{A \bigcap B}{A \bigcup B} - \frac{C \backslash (A \bigcap B)}{C}, \tag{3.2}$$

where C is the smallest convex object enclosing A and B. Obviously, $GIoU \in (-1, 1]$ and $GIoU <= IoU$. For the convenience we scale GIoU to the range $[0, 1]$.

We propose to use the HOTA metric with the 3D GIoU for the fair evaluation of the 3D MOT task. The code can be found on my github page [1].

[1] `https://github.com/nekorobov/HOTA-metrics`

4 Experiments

4.1 Implementation details

The implementation of the tracking algorithm is done in the C++ programming language. The Ceres library [19] is used for the optimization in DIA and DSO. The OpenCV library [20] is used for the operations with images and the sparse optical flow estimation. The thresholds that are used through the experiments are the following: the size of the window in DSO is 5, the number of active points in DSO is 500, mean max cost per pixel is C_p is 40, min number of pixels N_p is 50, the maximal length of the translation L is 5m, the number of levels in the DIA image pyramid is 3, $N_{wo\_detection}$ is 5. By default (if the opposite is not stated), we use the online mode of the system. If the tracking in 3D is not available (only 2D tracking), the 3D object is not outputted until the 3D tracking is reinitialized.

The instance segmentation masks are provided by the method proposed by Luiten, Jonathon et al. [21]. The depth is obtained with the DispNet [22].

4.2 3D Multi-object tracking evaluation

4.2.1 Dataset

The KITTI object tracking [23] data and ground truth are used for the experiments. Since our method requires the rigidity of the object, we evaluate the result only for the Car class. The split on the validation and training set for the 3D MOT task as proposed in [12] is used. The split proposed in [24] is used for the 2D MOT task.

4.2.2 Metrics

For the evaluation, we used the metric HOTA with 3D GIoU proposed in chapter 3, but to show the difference the HOTA with 3D IoU is shown as well. Finally, we evaluate the performance of our tracker with the metric CLEARMOT [13] that is used for the publicly available 2D MOT tracking benchmark by projecting 3D bounding boxes to the image plane.

4.2.3 Baseline

For the baseline of the system we have chosen the following setup: standard tracking and association of the objects as in the full system, but the bounding boxes are detected in the depth maps, rather than in the sparse accumulated point clouds. The bounding boxes are outputted per frame, without the consideration of the optimized point clouds and trajectories.

Tracker	Detector	HOTA	DetA	AssA	DetRe	DetPr	AssRe	AssPr	LocA
GIoU									
AB3DMOT [12]	PointRCNN (LiDAR) [4]	73.85	70.92	77.38	81.37	79.27	80.93	90.27	88.74
AB3DMOT [12]	DSGN (12 Gb) [25]	48.46	47.08	53.59	50.58	76.23	55.75	91.41	81.70
AB3DMOT [12]	DSGN (full) [25]	55.78	52.94	62.41	57.56	76.17	64.84	91.66	82.11
AB3DMOT [12]	DispRCNN (vob) [9]	67.21	66.18	69.59	69.79	84.43	72.00	91.09	85.92
AB3DMOT [12]	DispRCNN (pob) [9]	67.35	67.49	68.28	71.71	83.16	70.65	89.07	85.27
Our	Bbox	49.16	46.50	54.06	54.32	56.00	59.51	68.11	70.77
IoU									
AB3DMOT [12]	PointRCNN (LiDAR) [4]	65.59	61.65	70.67	72.38	70.51	75.14	82.98	81.80
AB3DMOT [12]	DSGN (12 Gb) [25]	40.60	35.32	51.93	40.08	60.40	55.19	84.29	75.10
AB3DMOT [12]	DSGN (full) [25]	46.51	39.82	59.71	46.00	60.87	63.46	85.44	75.89
AB3DMOT [12]	DispRCNN (vob) [9]	58.37	55.57	63.40	61.59	71.31	67.06	82.45	78.59
AB3DMOT [12]	DispRCNN (pob) [9]	57.62	55.519	61.24	61.346	71.145	64.71	80.342	78.243
Our	Bbox	31.60	26.90	39.05	35.48	36.58	45.93	51.27	65.07

Table 4.1: HOTA with 3D GIoU and 3D IoU to show the difference between the proposed metric (HOTA + GIoU) with the standard (HOTA + IoU). HOTA with GIoU for our method is almost 60% higher than HOTA with IoU, because we have many of the biased detections, so IoU for them is 0.

4.2.4 Runtime

The real-time performance of the proposed algorithm is not the main concern of this work. So, some of the parts are unoptimized and can be implemented more efficiently, the performance of the algorithm needs to be precisely evaluated. The nature of the proposed direct tracker is that the all the tracklets are independent and can be tracked concurrently. The only synchronization needed is the object association.

From the experience of the author the DIA is the slowest step in the proposed system setup. Since the DIA is algorithm performed for the image pyramid and for the each pixel of the object. This step becomes time-expensive for the large (in the image domain) objects. It is proposed to use either feature based initial tracking for the large objects or to use smaller resolution for the large object and full resolution for the small objects.

4.2.5 Tracking performance

We compare our method against the other methods with different 3D object detectors and trackers. More specifically, for the 3D tracker baseline we use AB3DMOT [12], for the detectors we use SOTA 3D object detectors on stereo images [9, 25] and LiDAR [4]. We compare against the two versions of the DispRCNN detector [9] (trained with the use of the predefined shape (pob) and lidar (vob)). We use the two versions of the DSGN detector [25]. The full version is the top-performing model provided by the authors of the work, the 12Gb version is the light version of the network that takes less memory for training and inference, than the top-performing one. It is worth mentioning that the pre-trained models provided by the authors of the mentioned methods are taken. The training of these models carried out by the authors on the KITTI object detection dataset and some of the frames in the training set may

Tracker	Detector	MOTA	MOTP	IDs	FP	FN
AB3DMOT [12]	PointRCNN (LiDAR) [4]	75.62	86.97	28	701	1110
AB3DMOT [12]	DSGN (12 Gb) [25]	57.24	87.03	204	224	2797
AB3DMOT [12]	DSGN (full) [25]	64.64	88.93	130	314	2223
AB3DMOT [12]	DispRCNN (vob) [9]	84.05	89.97	64	91	1048
AB3DMOT [12]	DispRCNN (pob) [9]	87.76	89.99	65	93	1067
MOTSFusion [3]	RRC [26] + BB2SegNet [21]	94.0	-	9	45	400
CIWT [10]	Regionlets [27]	74.38	82.85	26	-	-
Our	BB2SegNet [21] for 2D + Convex hull for 3D	89.13	82.18	78	192	550

Table 4.2: MOTA for 2D bounding box MOT task. Since our method relies on the instance segmentation masks, our result for 2D tracking only is better than some other "detection in point cloud" methods.

come from the validation set of the KITTI MOT validation split. So, for the fair comparison, one should re-train models on the frames only from the KITTI MOT training set split. The detections from the [9, 25, 4] are given with the confidence scores. The threshold for the confidence filtering is chosen based on the HOTA score of the tracker on the test split of the dataset.

Results of the evaluation of methods with HOTA + GIoU and HOTA + IoU are provided in Table 4.1. The results of the performance of our method on HOTA + GIoU are 60% larger than HOTA + IoU (49.16 vs 31.60). This can be explained as follows: our method relies on the 2D instance segmentation that has high recall and precision. The inaccuracies of our method come from the inaccuracies of the object detection in the accumulated sparse point cloud. So, with the GIoU metric, which takes into account the spatial distance between bounding boxes, the inaccurate detections are considered as the true positives for some threshold α. For the other methods that do the detections in the 3D directly, the described above problem has a weaker effect, as the FPs and FNs are usually "True FPs" and "True FNs" in the sense that the object does not exist in reality (or the real object has no detection) rather than the detection is not accurately localized.

The detection accuracy of the proposed method is much worse than the other learning-based detectors. This means that the improvement of the detector has a lot of potential for the further development of the method.

The results of the 2D bounding box MOT for the validation dataset split proposed in [24] are provided in the Table 4.2. For the evaluation, we backproject the 3D bounding boxes on the image plane.

4.2.6 Ablation study

In Table 4.3 the results of the ablation study are shown. For the results we used HOTA and GIoU. Notably that the system quality highly depends on the quality of the input depth map (system with the depth map from LibELAS and SGBM [28] from OpenCV gives HOTA 44.35 and 38.87 respectively). The depth maps from libELAS have noisy depth measurements for

the far-away objects, so the tracking them in 3D becomes unfeasible (even if the tracking is successful, the object detection in 3D is hardly biased). This effect is even stronger for the SGBM from OpenCV. For the setup "w/o DSO optimization" we used sparse accumulated (but un-optimized with DSO) point clouds. In setup "w/o DIA failure filtering" we allowed all the optimization results from the DIA to be accepted for the association and further optimization. In setup "w/o RANSAC pose initialization" we initialized the transformation for the DIA with identity transformation matrix. In the setup "W/o 2D tracking" we do not use the 2D tracking using optical flow, so if the object is failed to be tracked by DIA, the track is considered to be lost. The system setup "with GT masks" uses ground truth masks and he association in 2D is considered to be known. However, the association during the computation of HOTA with 3D GIoU is performed in 3D, so the AssA is only 58.90. It means the the object detection is the narrow place of the system.

The baseline of the proposed system outperforms the full algorithm according to HOTA metric. However, Figure 4.1 shows that the proposed method outperforms the baseline for the objects up to 20 meters from the camera according to MOTP3D error:

$$MOTP3D = \frac{\sum_{i,t} \left\| \mathbf{p}^i_{gt}(t) - \mathbf{p}^i_{det}(t) \right\|}{\sum_t n_{tp}(t)}. \tag{4.1}$$

The further from the camera, the bigger the depth estimation error and the bigger the tracking error as the consequence, and even the DSO step is not able to refine the point cloud for such a far distance. Another reason for this might be the fact that the further object from the camera, the fewer points are available for the DSO, so for the faraway objects, it might be not enough pixels selected to robustly do tracking and the object detection in the resulted point cloud.

Setup	HOTA	DetA	AssA	DetRe	DetPr	AssRe	AssPr	LocA
LibELAS depth [29]	44.35	40.30	51.53	48.65	50.43	57.57	65.78	69.19
SGBM Depth [28], OpenCV	38.87	36.08	43.62	44.77	46.37	50.89	58.85	68.06
W/o RANSAC pose initialization	48.00	45.11	52.81	52.17	56.78	58.16	68.08	71.04
W/o DIA failure filtering	48.04	45.88	52.47	53.43	55.69	57.61	68.09	70.56
W/o DSO optimization	48.67	46.05	53.29	53.43	56.37	58.67	68.00	70.82
W/o 2D tracking	46.69	45.97	49.26	53.24	56.41	53.79	68.82	70.73
With GT masks	51.17	45.98	58.90	54.18	54.75	66.02	66.40	70.348
Baseline	50.88	48.67	54.66	56.40	58.21	60.09	68.57	71.65
Full system	49.16	46.50	54.06	54.32	56.00	59.51	68.11	70.77

Table 4.3: Ablation study. HOTA with 3D GIoU. The quality of the input depth maps has strong influence on the results. 2D tracking is important part of the system. The overall performance of the baseline is higher than the proposed method (see Figure 4.1).

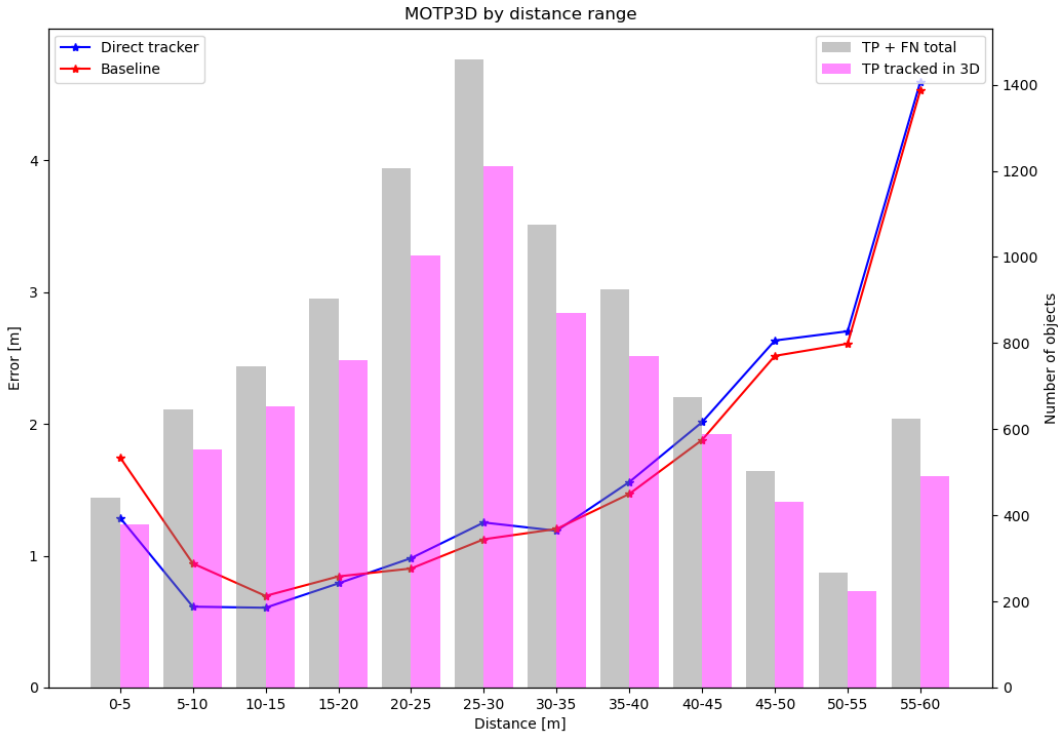


Figure 4.1: Comparison of the baseline with the full system MOTP3D (see 4.1) for the distance range. The association of the detections with the ground truth for this experiment is carried out by the IoU of the segmentation masks. Gray and pink bars show the total number / tracked in 3d objects for each distance range. Even though the summary performance of the whole method is worse than the performance of the baseline, the proposed method has lower MOTP3D error for close to the camera objects (up to 20 meters).

5 Conclusion and Discussions

We proposed the multi-object tracking system based on the 2D image segmentation masks and the direct sparse odometry. From the experiments, we have shown that the proposed system shows promising results compared to the other vision methods of 3D MOT. As the performance of the proposed method on the 2D MOT task shown in Table 4.2 is better than the other methods, precise object detection in the accumulated point cloud is required for the improvement of the system. It is shown that the proposed method outperforms the baseline for the close to the camera objects, but performs worse for the far-away objects. So more sophisticated strategy that considers the distance of the object from the camera is needed for the object representation. E.g. the neural network based object detectors in the point cloud can be used [4, 7].

The experimental stage should be performed more accurately. E.g. the single frame object detection neural networks need to be retrained on the KITTI tracking training test split, or the evaluation should be performed for the test set. The experiment of the evaluation of the object trajectories based on the odometry evaluation metric ATE can show the superiority of the DSO over the other methods.

Additionally, we have proposed a new benchmark for the evaluation of the 3D MOT task. The proposed evaluation metric is the HOTA metric with 3D GIoU. It is shown that the proposed metric makes sense to the 3D MOT task and overcomes the drawbacks of MOTA and IoU.

Bibliography

[1] S. Yang and S. Scherer. "Cubeslam: Monocular 3-d object slam". In: *IEEE Transactions on Robotics* 35.4 (2019), pp. 925–938.

[2] L. Xiao, J. Wang, X. Qiu, Z. Rong, and X. Zou. "Dynamic-SLAM: Semantic monocular visual localization and mapping based on deep learning in dynamic environment". In: *Robotics and Autonomous Systems* 117 (2019), pp. 1–16.

[3] J. Luiten, T. Fischer, and B. Leibe. "Track to reconstruct and reconstruct to track". In: *IEEE Robotics and Automation Letters* 5.2 (2020), pp. 1803–1810.

[4] S. Shi, X. Wang, and H. Li. "Pointrcnn: 3d object proposal generation and detection from point cloud". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 770–779.

[5] J. Ku, M. Mozifian, J. Lee, A. Harakeh, and S. L. Waslander. "Joint 3d proposal generation and object detection from view aggregation". In: *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2018, pp. 1–8.

[6] C. R. Qi, W. Liu, C. Wu, H. Su, and L. J. Guibas. "Frustum pointnets for 3d object detection from rgb-d data". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 918–927.

[7] 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: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 8445–8453.

[8] R. Wang, N. Yang, J. Stückler, and D. Cremers. "DirectShape: Direct Photometric Alignment of Shape Priors for Visual Vehicle Pose and Shape Estimation". In: *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2020, pp. 11067–11073.

[9] J. Sun, L. Chen, Y. Xie, S. Zhang, Q. Jiang, X. Zhou, and H. Bao. "Disp r-cnn: Stereo 3d object detection via shape prior guided instance disparity estimation". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 10548–10557.

[10] A. Osep, W. Mehner, M. Mathias, and B. Leibe. "Combined image-and world-space tracking in traffic scenes". In: *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2017, pp. 1988–1995.

[11] E. Baser, V. Balasubramanian, P. Bhattacharyya, and K. Czarnecki. "Fantrack: 3d multi-object tracking with feature association network". In: *2019 IEEE Intelligent Vehicles Symposium (IV)*. IEEE. 2019, pp. 1426–1433.

[12] X. Weng and K. Kitani. "A baseline for 3d multi-object tracking". In: *arXiv preprint arXiv:1907.03961* (2019).

[13] K. Bernardin and R. Stiefelhagen. "Evaluating multiple object tracking performance: the clear mot metrics". In: *EURASIP Journal on Image and Video Processing* 2008 (2008), pp. 1–10.

[14] S. Scheidegger, J. Benjaminsson, E. Rosenberg, A. Krishnan, and K. Granström. "Mono-camera 3d multi-object tracking using deep learning detections and pmbm filtering". In: *2018 IEEE Intelligent Vehicles Symposium (IV)*. IEEE. 2018, pp. 433–440.

[15] H. W. Kuhn. "The Hungarian method for the assignment problem". In: *Naval research logistics quarterly* 2.1-2 (1955), pp. 83–97.

[16] J. Engel, V. Koltun, and D. Cremers. "Direct sparse odometry". In: *IEEE transactions on pattern analysis and machine intelligence* 40.3 (2017), pp. 611–625.

[17] L. Jonathon, O. Aljosa, P. Dendorfer, P. Torr, A. Geiger, L. Leal-Taixé, and L. Bastian. "HOTA: A Higher Order Metric for Evaluating Multi-object Tracking". In: *International Journal of Computer Vision* 129.2 (2021), pp. 548–578.

[18] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, and S. Savarese. "Generalized intersection over union: A metric and a loss for bounding box regression". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 658–666.

[19] S. Agarwal, K. Mierle, et al. *Ceres Solver*. `http://ceres-solver.org`.

[20] G. Bradski. "The OpenCV Library". In: *Dr. Dobb's Journal of Software Tools* (2000).

[21] J. Luiten, P. Voigtlaender, and B. Leibe. "Premvos: Proposal-generation, refinement and merging for the davis challenge on video object segmentation 2018". In: *The 2018 DAVIS Challenge on Video Object Segmentation-CVPR Workshops*. Vol. 1. 2. 2018, p. 6.

[22] E. Ilg, T. Saikia, M. Keuper, and T. Brox. "Occlusions, motion and depth boundaries with a generic network for disparity, optical flow or scene flow estimation". In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, pp. 614–630.

[23] 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*. IEEE. 2012, pp. 3354–3361.

[24] P. Voigtlaender, M. Krause, A. Osep, J. Luiten, B. B. G. Sekar, A. Geiger, and B. Leibe. "Mots: Multi-object tracking and segmentation". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 7942–7951.

[25] Y. Chen, S. Liu, X. Shen, and J. Jia. "Dsgn: Deep stereo geometry network for 3d object detection". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 12536–12545.

[26] J. Ren, X. Chen, J. Liu, W. Sun, J. Pang, Q. Yan, Y.-W. Tai, and L. Xu. "Accurate single stage detector using recurrent rolling convolution". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 5420–5428.

[27] X. Wang, M. Yang, S. Zhu, and Y. Lin. "Regionlets for generic object detection". In: *Proceedings of the IEEE international conference on computer vision*. 2013, pp. 17–24.

[28] H. Hirschmuller. "Stereo processing by semiglobal matching and mutual information". In: *IEEE Transactions on pattern analysis and machine intelligence* 30.2 (2007), pp. 328–341.

[29] A. Geiger, M. Roser, and R. Urtasun. "Efficient large-scale stereo matching". In: *Asian conference on computer vision*. Springer. 2010, pp. 25–38.