CONFIDENCE INTERVALS FOR TRACKING PERFORMANCE SCORES

Ricardo Sanchez-Matilla and Andrea Cavallaro

Centre for Intelligent Sensing, Queen Mary University of London, UK
{ricardo.sanchezmatilla, a.cavallaro}@qmul.ac.uk

ABSTRACT

The objective evaluation of trackers quantifies the discrepancy between tracking results and a manually annotated ground truth. As generating ground truth for a video dataset is tedious and time-consuming, often only keyframes are manually annotated. The annotation between these keyframes is then obtained semi-automatically, for example with linear interpolation. This approximation has two main undesirable consequences: first, interpolated annotations may drift from the actual object, especially with moving cameras; second, trackers that use linear prediction or regularize trajectories with linear interpolation unfairly gain a higher tracking evaluation score. This problem may become even more important when semi-automatically annotated datasets are used to train machine learning modules. To account for these annotation inaccuracies for a given dataset, we identify objects whose annotations are interpolated and propose a simple method that analyzes existing annotations and produces a confidence interval to complement tracking scores. These confidence intervals quantify the uncertainty in the annotation and allow us to appropriately interpret the ranking of trackers with respect to the chosen tracking performance score.

Index Terms— Tracking performance scores, ground truth, annotation quality.

1. INTRODUCTION

The objective evaluation of tracking results helps to identify the strengths and weaknesses of an algorithm and should enable fair comparisons across trackers. The evaluation often relies on measuring the discrepancy between results and a manually set of annotations, known as ground truth (GT). The most common GT annotation consists of bounding rectangles approximating the location and shape of objects. The GT should be produced by multiple annotators and a consolidated GT should be generated after an analysis and correction of their annotations [1]. However, this process is tedious and time-consuming as datasets typically consist of hundreds of thousands annotations (e.g. 300K annotations in the multiple object tracking benchmark 2016, MOTB16 [2]). The annotation task may become impractical for large-scale datasets with millions of object trajectories to be annotated [3].

The GT annotation process can be sped up with semi-automatic or interactive methods. *Semi-automatic* methods allow the annotator to reduce the number of frames to be manually annotated and generate the missing annotations using linear interpolation [4, 5], tracking [1, 6], learning [1, 4, 7] or optimization [1] methods. A commonly used semi-automatic approach is 2D linear interpolation [1, 4, 5]. Keyframes to be manually annotated are selected by the annotator or at fixed intervals [1]. Videos can also be annotated from a sparse set of keyframe annotations by training an appearance-based detector using a convex temporal regularization [7]. More accurate annotations can be achieved by modeling linear interpolation in 3D and by performing image stabilization to compensate for camera motion [8]. *Interactive* methods enable a user to manually correct the annotation automatically generated through semi-automatic methods, after manual annotation of each object in one or multiple frames [4, 6]. For example, tracking and user interaction can be combined by allowing a user to reinitialize the annotation procedure when the tracker fails [6]. Table 1 summarizes the main semi-automatic tools to generate GT annotations.

The process used to generate the GT annotation for publicly available multi-target tracking datasets is shown in Table 2. CAVIAR [9] is the only dataset that provides a GT manually annotated frame-by-frame. PETS2009 [10], ETH [11], TUD [12, 13] and i-LIDS [14] include linearly interpolated annotation [15]. MOTB15 [16] has a combination of existing annotations for five videos (some manually annotated in full, some with linearly interpolated GT and others with an unknown annotation policy) and new annotations for six other videos generated using VATIC [1]. MOTB16 [2] uses a private annotation tool and rules that are not publicly disclosed. Therefore, we aim to identify which frames and objects were interpolated during the annotation process.

In this paper, we empirically estimate the bias in performance scores caused by the use of semi-automatic ground-truth annotation. This type of annotation introduces systematic errors that favor trackers with strategies for prediction or regularization that are similar to those used to generate the annotation. In particular, we analyze the effect of linear interpolation for GT annotation and show that these errors are larger with moving cameras, which are becoming increasingly popular. To address this problem, we first identify interpolated

Table 1: Annotation tools and their interpolation technique(s). Key – ViPER: Video Performance Evaluation Resource; KATRA: Keyframe-based Tracking for Rotoscoping and Animation; VATIC: Video Annotation Tool from Irvine, California; iVAT: interactive Video Annotation Tool; 2DL: 2D Linear interpolation; 3DL: 3D Linear interpolation; Track: Tracking; Learn: Learning; Opt: Optimization; ACM: Accounts for Camera Motion.

Ref	Annotation tool	Semi-automatic annotation procedure					ACM
		2DL	3DL	Track	Learn	Opt	
[5]	ViPER	✓					
[6]	KATRA			✓			
[8]	LabelMe		✓				✓
[7]	FlowBoost				✓		
[1]	VATIC	✓		✓	✓	✓	
[4]	iVAT	✓			✓		

Table 2: Datasets for multi-object tracking and their annotation. Key – *, 'ADL-Rundle-' and 'Venice-' annotation generated with linear interpolation with VATIC [1]; **: annotation generated with linear interpolation by [15]; MT: Mechanical Turk; KF: keyframe based; LI: linear interpolation; NA: not available.

Ref	Dataset	Tool	KF	LI
[9]	CAVIAR	CaviarGui		
[12]	TUD**	NA	✓	✓
[11]	ETH**	NA	✓	✓
[10]	PETS2009**	NA	✓	✓
[17]	KITTI	MT	NA	NA
[14]	i-LIDS	ViPER	✓	✓
[16]	MOTB15*	VATIC	NA	✓
[2]	MOTB16	NA	NA	✓

annotations and then quantify their effect in a performance score by encoding the corresponding uncertainty in a confidence interval. This confidence interval is used to complement existing tracking evaluation scores for a given annotated dataset.

We evaluate the impact of the proposed confidence intervals in the ranking for MOTB16 [2], the most commonly used multiple object tracking benchmark.

2. CONFIDENCE INTERVAL

Let $\mathbb{Z} = \{\mathbf{z}_k^\lambda : \lambda = 1 \dots \Lambda; k = 0 \dots K_\lambda - 1\}$ be the GT annotation for a generic dataset with Λ object trajectories, each K_λ frames long. Let an annotated object with identity λ at frame k be defined as

$$\mathbf{z}_k^\lambda = (u, v, w, h), \tag{1}$$

where (u, v) is the top-left corner, and w and h are width and height of the bounding box. $\mathbb{Z}$ may contain manual and interpolated annotations, and can be decomposed as $\mathbb{Z} = \tilde{\mathbb{Z}} \cup \hat{\mathbb{Z}}$, with $\tilde{\mathbb{Z}} \cap \hat{\mathbb{Z}} = \emptyset$, where $\tilde{\mathbb{Z}} = \{\tilde{\mathbf{z}}_k^\lambda\}$ contains manually annotated objects and $\hat{\mathbb{Z}} = \{\hat{\mathbf{z}}_k^\lambda\}$ contains automatically generated annotations, created for example through interpolation. If all the annotations are produced manually, then $\hat{\mathbb{Z}} = \emptyset$ and $\mathbb{Z} = \tilde{\mathbb{Z}}$. We term $\tilde{\mathbb{Z}}$ manual ground truth (MGT) and $\hat{\mathbb{Z}}$ interpolated ground truth (IGT).

Let us assume that linear interpolation was used to generate $\mathbb{Z}$ for a dataset. Our aim is to identify the elements of $\tilde{\mathbb{Z}} = \bigcup_{\lambda=1}^{\Lambda} \tilde{\mathbb{Z}}^\lambda$. These elements are likely to have non-zero acceleration for all the components of the state:

$$\tilde{\mathbb{Z}}^\lambda = \Big\{ \mathbf{z}_k^\lambda : \mathbf{z}_{uu}, \mathbf{z}_{vv}, \mathbf{z}_{ww}, \mathbf{z}_{hh} \neq 0 : k = 0 \dots \tilde{K}_\lambda - 1 \Big\}, \tag{2}$$

where $\mathbf{z}_{ii}$ is the second partial derivative (i.e. acceleration) of component i, and $\tilde{K}_\lambda = |\tilde{\mathbb{Z}}^\lambda| \leq K_\lambda$ is the cardinality of $\tilde{\mathbb{Z}}^\lambda$.

From the identified $\tilde{\mathbb{Z}}^\lambda$ we generate interpolated versions, $\mathbb{Z}_\beta^\lambda$, with different decimation factors, $\beta \geq 2$, through a decimation-interpolation procedure:

$$\mathbb{Z}_\beta^\lambda = \Big\{ \mathbf{z}_{k,\beta}^\lambda = \tilde{\mathbf{z}}_{i\beta}^\lambda + j\Delta_i^\lambda : k = 0 \dots \tilde{K}_\lambda - 1; i = 0 \dots \left\lfloor \frac{\tilde{K}_\lambda}{\beta} - 1 \right\rfloor; j = 0 \dots \beta - 1 \Big\}, \tag{3}$$

where $\Delta_i^\lambda = \frac{\tilde{\mathbf{z}}_{(i+1)\beta}^\lambda - \tilde{\mathbf{z}}_{i\beta}^\lambda}{\beta}$ and $\lambda \in \{1 \dots \Lambda\}$. Note that $\mathbb{Z}_\beta^\lambda$ is composed of manual annotations (when $j = 0$) and linearly interpolated annotations (when $j \neq 0$); and its cardinality is equal to that of $\tilde{\mathbb{Z}}^\lambda$, i.e. $|\mathbb{Z}_\beta^\lambda| = |\tilde{\mathbb{Z}}^\lambda| = \tilde{K}_\lambda$.

To empirically estimate the uncertainty introduced by $\mathbb{Z}_\beta^\lambda$ when evaluating tracking results, we consider a generic tracking performance measure, $s(\cdot, \cdot)$, which allows us to compare $\mathbb{Z}_\beta^\lambda$ against its corresponding $\tilde{\mathbb{Z}}^\lambda$ as:

$$\alpha_{s,\beta} = \frac{1}{\Lambda} \sum_{\lambda=1}^{\Lambda} s(\tilde{\mathbb{Z}}^\lambda, \mathbb{Z}_\beta^\lambda). \tag{4}$$

The value of $\alpha_{s,\beta}$ allows us to define the confidence interval for $s(\cdot, \cdot)$: a tracker with the same tracking results as a $\mathbb{Z}_\beta^\lambda$ should subtract (up to) $\alpha_{s,\beta}$ to the final score; whereas a tracker with the same tracking results as a given MGT should add (up to) $\alpha_{s,\beta}$ to the final score, if it is assessed using $\mathbb{Z}_\beta^\lambda$.

To apply the confidence interval on the performance scores for a specific dataset, we *map* $\alpha_{s,\beta}$ to the amount of interpolation detected in the dataset.

As an example, let us quantify the amount of linearly interpolated annotations in MOTB15 [16] and MOTB16 [2]. Using the definition in Eq. 2, the MOTB16 training dataset results in having 39.7% of linearly interpolation annotations (Table 3). This approximatively corresponds to a decimation factor of $\beta = 3$. We assume that the test dataset and the training dataset have a similar amount of linearly interpolated annotations, as the same annotation policy was used

Table 3: Percentage of linearly interpolated annotations detected in the MOTB15 and MOTB16 datasets using Eq. 2.

Camera motion	MOTB15	MOTB16
Static	6.2	52.7
Moving	17.0	12.7
Overall	11.0	39.7

Fig. 1: Comparison of manually annotated GT (red rectangle), interpolated generated GT with $\beta = 15$ (blue), and MOTB GT (green) for object 32 of MOT16-02 (static camera) at frames 331, 338 and 352 (a-c) and for object 4 of MOT16-10 (moving camera) at frames 1, 8 and 12 (d-f).

for both datasets [2]. MOTB16 uses three times more interpolation than MOTB15. In MOTB16, static-camera videos have a higher percentage of interpolated annotations than moving-camera videos. Surprisingly, in MOTB15 moving-camera videos have a higher percentage of interpolated GT annotation than static-camera videos.

To visualize the GT drifts caused by interpolation, we annotated frame-by-frame object 32 in MOT16-02 (static camera) and object 4 in MOT16-10 (moving camera), totaling 685 annotations. We refer to these annotations as *ideal* GT. Fig. 1 shows object 32 of MOT16-02 (top) and object 4 of MOT16-10 (bottom), and compares the MGT (red) against its decimated version, $\mathbb{Z}_\beta^\lambda$, with decimation factor $\beta = 15$ (blue) and the GT provided with the dataset (green), MOTB GT. The more noticeable drift occurs when the camera moves (Fig. 1(e-f)).

We quantify the overlap that the ideal GT produces against its interpolated versions and MOTB GT. We calculate the overlap between a manual annotation, $\tilde{\mathbf{z}}_k^\lambda$, and $\mathbf{z}_{k,\beta}^\lambda$, as: $\omega_k^\lambda = \frac{\tilde{\mathbf{z}}_k^\lambda \cap \mathbf{z}_{k,\beta}^\lambda}{\tilde{\mathbf{z}}_k^\lambda \cup \mathbf{z}_{k,\beta}^\lambda}$.

The effect of different interpolated GT versions on tracking evaluation scores is shown in Fig. 2. In the static-camera example (Fig. 2(a-b)), the overlap decreases moderately up to 0.5 (i.e. 50%). In the moving-camera example (Fig. 2(c-d)),

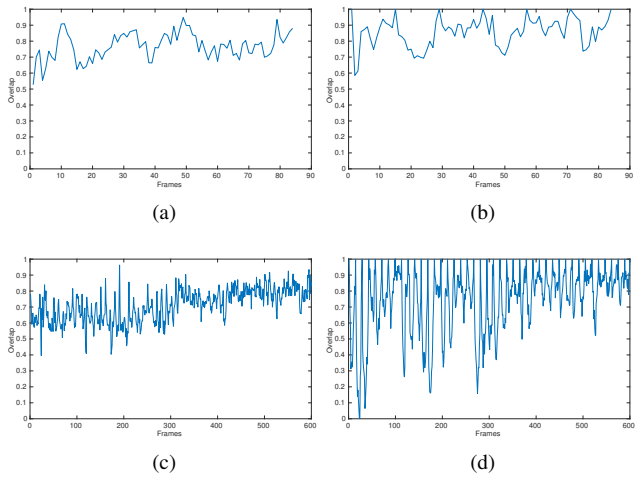

Fig. 2: Comparison of overlaps between GTs for MOT16-02 static-camera video (top row) and for MOT16-10 moving-camera video (bottom row). Left column: overlap between ideal GT and MOTB GT. Right column: overlap between MGT and its interpolated generated version ($\beta = 15$).

the overlap decreases up to having frames with *no overlap* between interpolated GT versions and the ideal GT.

3. IMPACT ON RANKING

In this section, we analyze the impact of the interpolated GT on the ranking of trackers in a specific benchmark, MOTB16 [2]. In order to account for the uncertainty introduced by the interpolated annotation for a given dataset, we define confidence intervals to complement performance scores.

We first identify frames and objects for which no interpolation is used in the publicly available GT, $\mathbb{Z}$, thus generating $\tilde{\mathbb{Z}}$. Then, we generate multiple interpolated versions, $\hat{\mathbb{Z}}_\beta = \bigcup_{\lambda=1}^{\Lambda} \hat{\mathbb{Z}}_\beta^\lambda$, with different decimation factors, $\beta \in \{3, 6, 9, 12\}$. Finally, we compare each $\mathbb{Z}_\beta^\lambda$ against the MGT, $\tilde{\mathbb{Z}}$, using the specific performance scores to define the confidence intervals to be applied to each score (Eq. 4).

We select Multiple Object Tracking Accuracy (MOTA) and Multiple Object Tracking Precision (MOTP) as performance scores [18]. MOTA for object λ is defined as

$$MOTA^\lambda = 1 - \frac{1}{\tilde{K}_\lambda} \sum_{k=0}^{\tilde{K}_\lambda - 1} (FN_k^\lambda + FP_k^\lambda + IDSW_k^\lambda), \quad (5)$$

where FN_k^λ, FP_k^λ and $IDSW_k^\lambda$ are the number of false negatives, false positives and identity switches for the object λ at frame k. MOTP for object λ is defined as

$$MOTP^\lambda = \frac{1}{\tilde{K}'_\lambda} \sum_{k=0}^{\tilde{K}'_\lambda - 1} \frac{\tilde{\mathbf{z}}_k^\lambda \cap \mathbf{z}_{k,\beta}^\lambda}{\tilde{\mathbf{z}}_k^\lambda \cup \mathbf{z}_{k,\beta}^\lambda}, \quad (6)$$

Table 4: Evaluation tracking score with a GT annotated frame-by-frame (*ideal* GT) against its interpolated generated GT, $\mathbb{Z}_{\beta}^{\lambda}$, with decimation factor ($\beta = 15$), and the MOTB GT annotation.

Camera motion	Sequence	GT	MOTA	MOTP
Static	MOT16-02-id32	$\mathbb{Z}_{\beta=15}^{\lambda=32}$	100	85.70
		MOTB	100	76.91
Moving	MOT16-10-id4	$\mathbb{Z}_{\beta=15}^{\lambda=4}$	72.00	69.23
		MOTB	97.33	70.36

Table 5: Uncertainties in MOTA and MOTP for MOTB16 produced by different decimation factors (β).

β	MOTA confidence	MOTP confidence
3	0.22	3.14
6	0.56	8.68
9	3.74	13.41
12	11.27	17.05

where $\tilde{K}'_{\lambda}$ is the number of frames with overlap over 0.5.

For MOTA we define: $s(\cdot,\cdot) = 100 - MOTA^{\lambda}$. Likewise for MOTP.

Table 4 shows that in the static-camera example (object 32 in MOT16-02) both interpolated versions obtain full MOTA, whereas MOTP considerably decreases due to interpolation, due to its direct relation with the overlap. In the moving-camera example (object 4 of MOT16-10) MOTA and MOTP differ from the ideal result. For example, when object 4 of MOT16-10 is evaluated with the ideal GT obtains a MOTA of 72 (second-last row in Table 4). This means that no tracker with MOTA results closer than 28 ($= 100 - 72$) to another tracker can be confidently said to outperform the other based on MOTA.

Table 5 shows how the confidence on the performance score varies with different decimation factors, β. In order to characterize MOTA and MOTP through the overlap only, we assume for simplicity that $IDSW^{\lambda} = 0$, $\forall\lambda$ for all trackers.

To conclude, let us consider the TOP-15 trackers sorted by MOTA that use public detections on the MOTB16 test dataset. For this dataset, the estimated *MOTA uncertainty* is 0.22 and *MOTP uncertainty* is 3.14 (first row Table 5). These values of $\alpha_{s,\beta}$ for the two scores allow us to analyze the impact of the public GT annotation of MOTB16.

Fig. 3(a-b) shows MOTA and MOTP results obtained for each tracker[1], whereas Fig. 3(c-d) shows the ranking of trackers based on MOTA and MOTP. The estimated MOTA and MOTP confidence intervals are shown as bars. Note that MOTP is very sensitive to GT interpolation as small overlap variations influence the measure (Eq. 6). MOTA is instead less sensitive as it depends on the number of false positives and false negatives, which vary only when the overlap becomes smaller than 50%.

[1] https://motchallenge.net/results/MOT16/ Last accessed on 9^{th} February 2018.

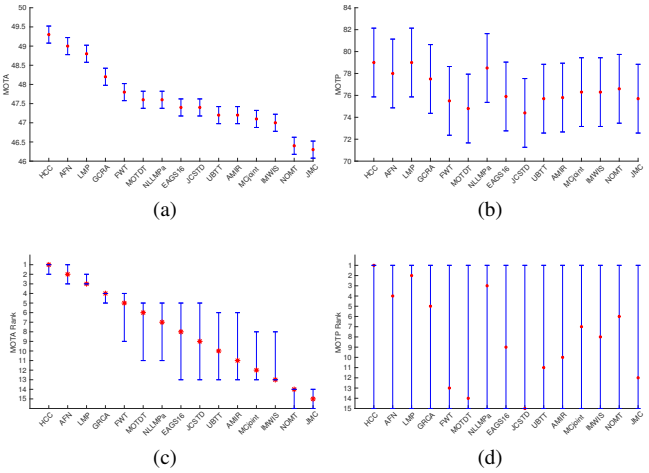

Fig. 3: TOP-15 performing trackers using public detections in MOTB16 according to MOTA measure. Red crosses indicate the MOTB Challenge measure (top row) and rank (bottom row). Blue bars translate the confidence interval (top row) into the ranking uncertainty (bottom row).

In summary, no TOP-15 tracker can be confidently assigned to a specific rank in the MOTB16 benchmark, as neighboring trackers are within their MOTA uncertainty ranges. Moreover, there is no significant difference among any of the TOP-15 trackers in terms of MOTP, i.e. even the first and fifteenth tracker cannot be confidently ranked relative to each other in terms of MOTP.

4. CONCLUSION

We quantified the bias caused by ground-truth generated semi-automatically with linear interpolation, which may undeservedly benefit trackers that use or learn to use linear prediction or regularization models. We showed that this problem is particularly acute with moving cameras. To account for this uncertainty when comparing trackers, we calculate a confidence interval for a given evaluation score and dataset using only information extracted from the GT annotation. We hope that this simple solution, which does not require further annotations, will help to produce more meaningful comparisons when evaluating and ranking trackers.

Future work includes to define confidence intervals for other types of interpolated annotations, such as those that make use of tracking, learning, or optimization; and for other applications such as training deep neural networks on large-scale datasets annotated semi-automatically.

5. REFERENCES

[1] C. Vondrick, D. Patterson, and D. Ramanan, "Efficiently scaling up crowdsourced video annotation," *Interna-*

tional Journal of Computer Vision, vol. 101, no. 1, pp. 184–204, 2013.

[2] A. Milan, L. Leal-Taixé, I. Reid, S. Roth, and K. Schindler, "MOT16: A benchmark for multi-object tracking," *arXiv:1603.00831*, 2016.

[3] A. Alahi, V. Ramanathan, and L. Fei-Fei, "Socially-aware large-scale crowd forecasting," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 2211–2218, 2014.

[4] S. Bianco, G. Ciocca, P. Napoletano, and R. Schettini, "An interactive tool for manual, semi-automatic and automatic video annotation," *Computer Vision and Image Understanding*, vol. 131, pp. 88–99, 2015.

[5] D. Mihalcik and D. Doermann, "The design and implementation of viper," https://www.cs.umd.edu/sites/default/files/scholarly_papers/davidm-viper_1.pdf%20, 2003.

[6] A. Agarwala, A. Hertzmann, D. Salesin, and S. Seitz, "Keyframe-based tracking for rotoscoping and animation," *IEEE Transactions on ACM Graphics*, vol. 23, no. 3, pp. 584–591, 2004.

[7] K. Ali, D. Hasler, and F. Fleuret, "Flowboost - appearance learning from sparsely annotated video," in *Proc. of Computer Vision and Pattern Recognition*, Colorado Springs, CO, USA, 2011, pp. 1433–1440.

[8] J. Yuen, B. Russell, C. Liu, and A. Torralba, "Labelme video: Building a video database with human annotations," in *Proc. of International Conference on Computer Vision*, Kyoto, Japan, 2009, pp. 1451–1458.

[9] "CAVIAR dataset. Last accessed on 9/2/2018," http://homepages.inf.ed.ac.uk/rbf/CAVIARDATA1/, 2004.

[10] J. Ferryman and A. Shahrokni, "PETS2009: Dataset and challenge," in *International Workshop on Performance Evaluation of Tracking and Surveillance*, Miami, FL, USA, 2009.

[11] A. Ess, B. Leibe, K. Schindler, and L. van Gool, "A Mobile Vision System for Robust Multi-Person Tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 10, pp. 1831–1846, 2009.

[12] M. Andriluka, S. Roth, and B. Schiele, "People-tracking-by-detection and people-detection-by-tracking," in *Proc. of Computer Vision and Pattern Recognition*, San Francisco, CA, USA, 2010, pp. 1–8.

[13] M Andriluka and S. RotS. Roth. Schiele, "Monocular 3d pose estimation and tracking by detection," in *Proc. of Computer Vision and Pattern Recognition*, San Francisco, CA, USA, 2010, pp. 623–630.

[14] "Imagery Library for Intelligent Detection Systems (i-LIDS). Last accessed on 9/2/2018," www.ilids.co.uk, 2012.

[15] A. Milan, "Public ground truth by Anton Milan. Last accessed on 9/2/2018," http://www.milanton.de/data.html, 2004.

[16] L. Leal-Taixé, A. Milan, I. Reid, S. Roth, and K. Schindler, "MOTChallenge 2015: Towards a benchmark for multi-target tracking," *arXiv:1504.01942*, 2015.

[17] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *Proc. of Computer Vision and Pattern Recognition*, Providence, RI, USA, 2012, pp. 3354–3361.

[18] K. Bernardin and R. Stiefelhagen, "Evaluating multiple object tracking performance: The clear mot metrics," *EURASIP Journal on Image and Video Processing*, vol. 2008, no. 1, pp. 246–309, 2008.