The KITTI Vision Benchmark Suite

A project of Karlsruhe Institute of Technology and Toyota Technological Institute at Chicago

- home
- setup
- stereo
  - Stereo 2012
  - Stereo 2015
- flow
  - Flow 2012
  - Flow 2015
- sceneflow
- depth
  - Depth Completion
  - Depth Prediction
- odometry
- object
  - 2d object
  - 3d object
  - bird's eye view
- tracking
  - multi-object tracking
  - multi-object tracking and segmentation
- road
- semantics
  - pixel-level
  - instance-level
- raw data
- submit results

Andreas Geiger (MPI Tübingen) | Philip Lenz (KIT) | Christoph Stiller (KIT) | Raquel Urtasun (University of Toronto)

Multi-Object Tracking and Segmentation (MOTS) Evaluation

The Multi-Object and Segmentation (MOTS) benchmark consists of 21 training sequences and 29 test sequences. It is based on the KITTI Tracking Evaluation 2012 and extends the annotations to the Multi-Object and Segmentation (MOTS) task. To this end, we added dense pixelwise segmentation labels for every object. We evaluate submitted results using the common metrics CLEAR MOT and MT/PT/ML (adapted for the segmentation case).

- Project page
- Download (trainset images + annotations / testset images)
- Description of annotation format
- Tools for loading data, evaluation, and visualization

Important Policy Update: As more and more non-published work and re-implementations of existing work is submitted to KITTI, we have established a new policy: from now on, only submissions with significant novelty that are leading to a peer-reviewed paper in a conference or journal are allowed. Minor modifications of existing algorithms or student research projects are not allowed. Such work must be evaluated on a split of the training set. To ensure that our policy is adopted, new users must detail their status, describe their work and specify the targeted venue during registration. Furthermore, we will regularly delete all entries that are 6 months old but are still anonymous or do not have a paper associated with them. For conferences, 6 month is enough to determine if a paper has been accepted and to add the bibliography information. For longer review cycles, you need to resubmit your results.

Additional information used by the methods

- Stereo: Method uses left and right (stereo) images
- Laser Points: Method uses point clouds from Velodyne laser scanner
- GPS: Method uses GPS information
- Online: Online method (frame-by-frame processing, no latency)
- Additional training data: Use of additional data sources for training (see details)

CAR

	Method	Setting	Code	sMOTSA	MOTSA	MOTSP	MOTSAL	MODSA	MODSP	MT	ML	IDS	Frag	Runtime	Environment	Compare
1	CCP_ST			**84.50 %**	**94.40 %**	89.70 %	**94.90 %**	**94.90 %**	91.90 %	90.10 %	1.20 %	202	416	0.1 s	GPU @ 2.5 Ghz (Python)	
2	CCP			84.50 %	94.40 %	89.70 %	94.90 %	94.90 %	91.90 %	90.50 %	1.20 %	197	435	0.1 s	GPU @ 2.5 Ghz (Python)	
3	PointTrackV2			83.00 %	93.50 %	89.00 %	94.00 %	94.00 %	91.40 %	90.10 %	0.80 %	210	531	0.1 s	GPU @ 2.5 Ghz (Python)	
4	PointTrack++			82.80 %	92.60 %	89.70 %	93.30 %	93.30 %	92.10 %	89.50 %	1.20 %	270	584	0.095 s	GPU @ 2.5 Ghz (Python)	
5	ViP-DeepLab			81.00 %	90.70 %	**89.90 %**	91.80 %	91.80 %	**92.20 %**	**92.20 %**	0.60 %	392	580	0.1 s	1 core @ 2.5 Ghz (C/C++)	

S. Qiao, Y. Zhu, H. Adam, A. Yuille and L. Chen: ViP-DeepLab: Learning Visual Perception with Depth-aware Video Panoptic Segmentation. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2021.

6	UW_JMV3D			79.60 %	89.60 %	89.00 %	89.90 %	89.90 %	91.40 %	79.10 %	2.90 %	114	532	0.08 s	GPU @ 2.5 Ghz (Python)	
7	PointTrack			78.50 %	90.90 %	87.10 %	91.80 %	91.80 %	89.70 %	90.80 %	0.60 %	346	645	0.045 s	GPU @ 2.5 Ghz (Python)	

Z. Xu, W. Zhang, X. Tan, W. Yang, H. Huang, S. Wen, E. Ding and L. Huang: Segment as Points for Efficient Online Multi-Object Tracking and Segmentation. Proceedings of the European Conference on Computer Vision (ECCV) 2020.

8	ReID MOT			78.00 %	90.40 %	87.10 %	91.80 %	91.80 %	89.70 %	90.80 %	0.60 %	533	836	1000 s	1 core @ 2.5 Ghz (C/C++)	
9	Lif_TS			77.50 %	88.10 %	88.30 %	88.60 %	88.60 %	90.90 %	79.60 %	2.70 %	183	569	1 s	1 core @ 3.0 Ghz (Python + C/C++)	
10	MCFPA			77.00 %	87.70 %	88.30 %	89.10 %	89.10 %	90.80 %	82.90 %	0.60 %	503	724	1 s	1 core @ 2.5 Ghz (Python)	
11	GMPHD_MAF			76.50 %	87.10 %	88.40 %	88.40 %	88.40 %	90.90 %	82.10 %	0.60 %	475	842	0.18 s	4 cores @ >3.5 Ghz (C/C++)	
12	UMotsNet			76.50 %	88.50 %	87.20 %	90.40 %	90.40 %	89.90 %	85.40 %	0.90 %	707	1008	0.14 s	1 core @ 2.5 Ghz (Python)	
13	IA-MOT-ST			76.50 %	87.40 %	88.10 %	89.20 %	89.20 %	90.60 %	82.10 %	1.10 %	649	873	TBD s	GPU @ 2.5 Ghz (Python)	
14	COSTA_TS		code	76.30 %	87.50 %	88.10 %	89.90 %	89.90 %	90.50 %	84.70 %	1.10 %	903	1081	2000 s	GPU @ 2.5 Ghz (Python)	
15	ReMOTS			75.90 %	86.70 %	88.20 %	88.70 %	88.70 %	90.70 %	84.50 %	0.60 %	716	905	3 s	1 core @ 2.5 Ghz (Python)	

F. Yang, X. Chang, C. Dang, Z. Zheng, S. Sakti, S. Nakamura and Y. Wu: ReMOTS: Self-Supervised Refining Multi- Object Tracking and Segmentation. 2020.

16	HRNt			75.80 %	86.50 %	88.30 %	88.10 %	88.10 %	90.70 %	78.10 %	2.90 %	599	703	1.0 s	GPU @ 2.5 Ghz (Python)	
17	Struct_MOTS			75.50 %	85.40 %	88.60 %	86.30 %	86.30 %	91.20 %	75.20 %	1.40 %	300	895	8.7 s	1 core @ 2.5 Ghz (Python)	
18	GMPHD_SAF			75.40 %	86.70 %	87.50 %	88.20 %	88.20 %	90.10 %	82.00 %	0.60 %	549	874	0.08 s	4 cores @ >3.5 Ghz (C/C++)	

Y. Song and M. Jeon: Online Multi-Object Tracking and Segmentation with GMPHD Filter and Simple Affinity Fusion. arXiv preprint arXiv:2009.00100 2020.

19	MOTSFusion		code	75.00 %	84.10 %	89.30 %	84.70 %	84.70 %	91.70 %	66.10 %	6.20 %	201	572	0.44 s	GPU @ 2.5 Ghz (Python)	

J. Luiten, T. Fischer and B. Leibe: Track to Reconstruct and Reconstruct to Track. IEEE Robotics and Automation Letters 2020.

20	EagerMOT			74.50 %	83.50 %	89.60 %	84.80 %	84.80 %	92.10 %	67.10 %	3.50 %	457	811	0.011 s	4 cores @ 3.0 Ghz (Python)	
21	USN			74.00 %	84.60 %	88.30 %	87.40 %	87.40 %	90.80 %	78.70 %	1.20 %	1045	1243	0.5 s	1 core @ 2.5 Ghz (Python)	
22	TES			73.50 %	84.00 %	88.30 %	87.40 %	87.40 %	90.80 %	78.20 %	1.10 %	1248	1446	0.3	1 core @ 2.5 Ghz (Python)	
23	LidarMOTS			73.30 %	84.00 %	88.50 %	86.40 %	86.40 %	91.00 %	84.10 %	1.80 %	894	1061	0.1 s	1 core @ 2.5 Ghz (C/C++)	
24	CPMOTS			72.80 %	84.90 %	86.40 %	86.00 %	86.00 %	89.20 %	75.70 %	1.80 %	439	891	0.06 s	GPU @ 3.0 Ghz (Python)	
25	TrackR-CNN_CCP			71.80 %	85.10 %	85.30 %	86.80 %	86.80 %	88.30 %	79.40 %	1.70 %	609	963	0.1 s	GPU @ 2.5 Ghz (Python)	
26	MOTS R-CNN			69.80 %	82.90 %	85.00 %	83.90 %	84.00 %	88.20 %	75.10 %	1.80 %	399	893	0.25 s	1 core @ 2.5 Ghz (Python)	
27	TrackR-CNN		code	67.00 %	79.60 %	85.10 %	81.50 %	81.50 %	88.30 %	74.90 %	2.30 %	692	1058	0.5 s	GPU @ 2.5 Ghz (Python)	

P. Voigtlaender, M. Krause, A. O\usep, J. Luiten, B. Sekar, A. Geiger and B. Leibe: MOTS: Multi-Object Tracking and Segmentation. CVPR 2019.

28	resut			61.60 %	73.00 %	88.00 %	76.30 %	76.30 %	90.50 %	88.90 %	**0.50 %**	1218	1410	1 s	1 core @ 2.5 Ghz (Python)	

Table as LaTeX | Only published Methods

PEDESTRIAN

	Method	Setting	Code	sMOTSA	MOTSA	MOTSP	MOTSAL	MODSA	MODSP	MT	ML	IDS	Frag	Runtime	Environment	Compare
1	CCP_ST			**70.50 %**	**86.30 %**	82.40 %	**87.50 %**	**87.60 %**	94.00 %	72.60 %	3.00 %	254	477	0.1 s	GPU @ 2.5 Ghz (Python)	
2	CCP			70.20 %	85.80 %	82.30 %	87.20 %	87.20 %	94.00 %	72.20 %	3.70 %	275	514	0.1 s	GPU @ 2.5 Ghz (Python)	
3	ViP-DeepLab			68.70 %	84.50 %	82.30 %	85.50 %	85.50 %	93.90 %	73.30 %	2.60 %	209	443	0.1 s	1 core @ 2.5 Ghz (C/C++)	

S. Qiao, Y. Zhu, H. Adam, A. Yuille and L. Chen: ViP-DeepLab: Learning Visual Perception with Depth-aware Video Panoptic Segmentation. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2021.

4	PointTrackV2		68.70 %	84.40 %	81.90 %	85.20 %	85.20 %	94.00 %	66.30 %	4.40 %	166	468	0.1 s	GPU @ 2.5 Ghz (Python)	☐
5	PointTrack++		68.10 %	83.60 %	82.20 %	84.80 %	84.80 %	94.00 %	66.70 %	4.80 %	250	521	0.095 s	GPU @ 2.5 Ghz (Python)	☐
6	MCFPA		67.20 %	83.00 %	81.90 %	84.30 %	84.30 %	93.80 %	67.00 %	3.00 %	265	484	1 s	1 core @ 2.5 Ghz (Python)	☐
7	ReID MOT		66.60 %	81.90 %	82.00 %	82.60 %	82.60 %	93.90 %	61.50 %	5.20 %	150	555	1000 s	1 core @ 2.5 Ghz (C/C++)	☐
8	ReMOTS		66.00 %	81.30 %	82.00 %	83.20 %	83.20 %	94.00 %	62.60 %	5.60 %	391	551	3 s	1 core @ 2.5 Ghz (Python)	☐

F. Yang, X. Chang, C. Dang, Z. Zheng, S. Sakti, S. Nakamura and Y. Wu: ReMOTS: Self-Supervised Refining Multi- Object Tracking and Segmentation. 2020.

9	GMPHD_MAF		64.90 %	79.60 %	82.30 %	81.30 %	81.30 %	94.00 %	58.90 %	4.80 %	348	639	0.18 s	4 cores @ >3.5 Ghz (C/C++)	☐
10	UW_JMV3D		64.90 %	80.90 %	81.00 %	81.90 %	81.90 %	93.60 %	61.50 %	8.90 %	206	577	0.08 s	GPU @ 2.5 Ghz (Python)	☐
11	IA-MOT-ST		63.90 %	80.30 %	81.50 %	83.30 %	83.30 %	93.60 %	73.00 %	**2.20 %**	611	745	TBD s	GPU @ 2.5 Ghz (Python)	☐
12	GMPHD_SAF		62.80 %	78.20 %	81.60 %	80.40 %	80.50 %	93.70 %	59.30 %	4.80 %	474	696	0.08 s	4 cores @ >3.5 Ghz (C/C++)	☐

Y. Song and M. Jeon: Online Multi-Object Tracking and Segmentation with GMPHD Filter and Simple Affinity Fusion. arXiv preprint arXiv:2009.00100 2020.

13	USN		62.80 %	77.00 %	82.40 %	79.60 %	79.60 %	94.20 %	52.60 %	7.40 %	547	734	0.5 s	1 core @ 2.5 Ghz (Python)	☐
14	TES		62.20 %	76.60 %	82.40 %	80.10 %	80.10 %	94.10 %	53.70 %	5.90 %	741	974	0.3	1 core @ 2.5 Ghz (Python)	☐
15	PointTrack		61.50 %	76.50 %	81.00 %	77.40 %	77.40 %	93.80 %	48.90 %	9.30 %	176	632	0.045 s	GPU @ 2.5 Ghz (Python)	☐

Z. Xu, W. Zhang, X. Tan, W. Yang, H. Huang, S. Wen, E. Ding and L. Huang: Segment as Points for Efficient Online Multi-Object Tracking and Segmentation. Proceedings of the European Conference on Computer Vision (ECCV) 2020.

16	Struct_MOTS		60.30 %	72.60 %	**83.30 %**	73.30 %	73.30 %	**94.50 %**	44.40 %	14.10 %	142	635	8.7 s	1 core @ 2.5 Ghz (Python)	☐
17	MOTSFusion	code	58.70 %	72.90 %	81.50 %	74.20 %	74.20 %	94.10 %	47.40 %	15.60 %	279	534	0.44 s	1 core @ 2.5 Ghz (C/C++)	☐

J. Luiten, T. Fischer and B. Leibe: Track to Reconstruct and Reconstruct to Track. IEEE Robotics and Automation Letters 2020.

18	HRNt		58.30 %	72.30 %	82.40 %	76.70 %	76.70 %	94.00 %	53.30 %	6.70 %	922	978	1.0 s	GPU @ 2.5 Ghz (Python)	☐
19	EagerMOT		58.10 %	72.00 %	81.50 %	73.30 %	73.30 %	94.10 %	43.30 %	13.70 %	270	633	0.011 s	4 cores @ 3.0 Ghz (Python)	☐
20	CPMOTS		58.10 %	73.70 %	79.70 %	74.70 %	74.70 %	93.40 %	46.30 %	12.20 %	209	631	0.06 s	GPU @ 3.0 Ghz (Python)	☐
21	LidarMOTS		57.60 %	72.10 %	80.90 %	73.40 %	73.40 %	94.00 %	45.60 %	15.90 %	275	545	0.1 s	1 core @ 2.5 Ghz (C/C++)	☐

22	resut		57.30 %	74.20 %	81.40 %	76.00 %	76.00 %	93.60 %	**79.60 %**	2.20 %	374	627	1 s	1 core @ 2.5 Ghz (Python)
23	UMotsNet		57.10 %	75.70 %	77.00 %	77.90 %	77.90 %	92.50 %	53.00 %	10.40 %	454	715	0.14 s	1 core @ 2.5 Ghz (Python)
24	Lif_TS		55.80 %	67.70 %	82.90 %	68.00 %	68.00 %	94.40 %	44.10 %	13.70 %	66	536	1 s	1 core @ 3.0 Ghz (Python + C/C++)
25	COSTA_TS	code	48.10 %	64.00 %	77.60 %	67.50 %	67.50 %	92.90 %	40.00 %	14.80 %	716	809	2000 s	GPU @ 2.5 Ghz (Python)
26	TrackR-CNN_CCP		48.10 %	66.40 %	75.20 %	68.20 %	68.20 %	91.90 %	44.40 %	11.90 %	381	781	0.1 s	GPU @ 2.5 Ghz (Python)
27	TrackR-CNN	code	47.30 %	66.10 %	74.60 %	68.40 %	68.40 %	91.80 %	45.60 %	13.30 %	481	861	0.5 s	GPU @ 2.5 Ghz (Python)

P. Voigtlaender, M. Krause, A. O\usep, J. Luiten, B. Sekar, A. Geiger and B. Leibe: MOTS: Multi-Object Tracking and Segmentation. CVPR 2019.

28	MOTS R-CNN		46.70 %	65.10 %	74.10 %	66.50 %	66.60 %	92.00 %	39.30 %	15.90 %	293	746	0.25 s	1 core @ 2.5 Ghz (Python)

Table as LaTeX | Only published Methods

Citation

When using this dataset in your research, we will be happy if you cite us:

```
@INPROCEEDINGS{Voigtlaender2019CVPR,
  author = {Paul Voigtlaender and Michael Krause and Aljosa Osep and Jonathon Luiten and Berin Balachandar Gnana Sekar and Andreas Geiger and Bastian Leibe},
  title = {MOTS: Multi-Object Tracking and Segmentation},
  booktitle = {Conference on Computer Vision and Pattern Recognition (CVPR)},
  year = {2019}
}
```