Supplementary Material

A Data Association Algorithm

Algorithm 1 Data Association

Data: Known Tracks $\mathcal{H} = \{\mathcal{H}_0, \mathcal{H}_1, \cdots, \mathcal{H}_N\}$,
incoming detection $\mathcal{D} = \{\mathcal{D}_0, \mathcal{D}_1, \cdots, \mathcal{D}_M\}$
Result: Update for Known Tracks $\mathcal{H}$

$\mathcal{M} \leftarrow \emptyset$ ▷ Initiliaze set of matched detections
$\mathcal{U} \leftarrow \mathcal{D}$ ▷ Initiliaze set of incoming detections
Compute cost matrix $\mathbf{C} = [c_{i,j}]$ using Eq. 4
$\mathcal{P} \leftarrow \{\text{Kalman\_Filter}(\mathcal{H}_i) \mid \mathcal{H}_i \in \mathcal{H}\}$
$d\,(i,j) \leftarrow \text{Mahalanobis\_distances}(\mathcal{H}_i, \mathcal{P}_j)$
Compute gate matrix $\mathbf{B} = [b_{i,j}]$ using Eq. 5
$\mathcal{M} \leftarrow \mathcal{M} \cup \{(i,j) \mid b_{i,j} \cdot c_{i,j} > 0\}$ ▷ Apply Hungarian algorithm
$\mathcal{U} \leftarrow \mathcal{U} \setminus \{j \mid \sum_i b_{i,j} \cdot c_{i,j} > 0\}$
$\mathcal{H} \leftarrow \{\mathcal{H}_k \mid (i,j) \in \mathcal{M}, k \neq i\} \cup \{\mathcal{H}_i \cup x_i \mid (i,j) \in \mathcal{M}\}^{D_{max}}$
$\mathcal{H} \leftarrow \mathcal{H} \cup \{\mathcal{H}_{N+j} \mid j \in \mathcal{M}\}$

The ablation study of the data assotiation algorithm is summarized in Tab. A.1. The settings for the Kalman filter and Mahalanobis gating correspond to those of DeepSORT[40].

Table A.1: Ablation analysis of data association components. MA, IoU, KF, MG stand for Metric association, Intersection-over-Union, the Kalman filter, Mahalanobis Gating components, respectively.

MA	IoU	KF	MG	**FP** ↓	**FN** ↓	**IDs** ↓	**FP + FN + IDs** ↓
✓				9'849	**340'193**	3'861	353'903
✓	✓			9'849	**340'193**	3'813	353'855
✓	✓	✓	✓	**9'909**	340'282	**2'800**	**352'991**

B Processing Time of ODESA-based DBT Solutions

The average frame processing time for our ODESA-based DBT solutions are summarized in Table B.1. The last column contains the average processing time per frame. The distribution of the total processing time among the components is provided as well. The number of detects affects considerably the processing time of the data association stage. In this respect, our solution is similar to any DBT method. The average number of detects in the case of CVPR'19 Tracking Challenge was at least an order of magnitude higher compared to KITTI.

Table B.1: Average frame processing time for ODESA-based DBT Solutions. The percentage of the total processing time is provided for each component.

Benchmark	Detector	Predictor	Embedding vector extraction	Association stage	Total time, s
CVPR19	-	64.0 %	5.8 %	30.2 %	0.606
KITTI pedestrian	96.7 %	-	0.7 %	2.6 %	0.454
KITTI car	96.9 %	-	0.7 %	2.5 %	0.480

Table C.1: Properties of CNN, which serve as embedding functions. MAC denotes the number of multiply-accumulate operations. The inference latency shown in the last column corresponds to the batch size of one processed by NVidia GTX 1050 Ti.

Method	**Color Space**	**Patch Size**	**Embedding Size**	**Parameters**, M	**MACs**[4], M	**Latency**, ms
HardNet++[27]	GRAY	32x32	128	1.3346	39.32	1.15 ± 0.15
SOSNet[36]	GRAY	32x32	128	1.3346	39.32	1.40 ± 0.27
AGW[30]	RGB	256x128	2048	25.0768	4'083.39	11.55 ± 0.17
OSNet[32]	RGB	256x128	512	2.5548	1'003.39	13.97 ± 0.25
MLFN[57]	RGB	256x128	1024	33.2428	2'794.52	24.38 ± 0.45
HACNN[33]	RGB	160x64	1024	3.6996	551.95	18.87 ± 0.28
TriNet[64]	RGB	256x128	128	25.7384	2'275.51	13.48 ± 0.35
ODESA (Our)	GRAY	32x32	128	1.3357	39.09	1.89 ± 0.10
	HSV	32x32	128	1.3363	39.68	1.96 ± 0.16
	HSV	64x64	128	1.3641	98.96	2.19 ± 0.20
	HSV	64x64	512	4.5106	102.11	3.50 ± 0.31

C ODESA Characterization

In Table C.1 we summarize the embedding function properties, which are relevant to their practical application, for a number of models compared with ODESA within this study. It is necessary to mention that our goal was to achieve the maximal MOT performance using the minimal number of modifications introduced into L2-Net topology[28]. For this reason, our models cannot be regarded as optimal from a practical point of view. Nevertheless, it is evident from this table that regarding the number of operations ODESA models still stay close to their LLFD origin. Regarding the CNN parameter count, a few person Re-ID models turn out to be comparable with our heaviest option, which is HSV64/512. With respect to the inference time, all ODESA models fit into the range between SOSNet[36] and AGW[30], which is the fastest considered person Re-ID model, while staying closer to its shorter limit.

Table C.2 extends the evaluation results introduced in Table 2 with additional entries. Here the bottom rows, which contain the results for a number of selected local keypoint and person Re-ID embedding functions, were brought to serve as a reference. It is worth noticing that the maximal TPR@FAR0.1 value of 87.95 achieved by HSV64/512 model falls in the middle of the range from 82.1 to 94.8, which is reported by Wang et al. [25] for JDE models trained with the cross-entropy loss. And it is twice as high as the values obtained by JDE models

[4] MAC was regarded here as a combination of two operations, i.e. multiplication followed by accumulation.

trained with triplet loss. At the same time, HSV64--/128 model achieves the highest TPR@FAR10e-6 value of 7.88%. This value exceeds by a factor of two any result exhibited by the considered person Re-ID model.

Table C.2: The validation results obtained according to the retrieval routine described in Ref. [25]. In the case of ODESA models, the first column refers to the color space and the size of the square input patch in pixels.

Model	**Keypoint Detector**	**Embedding Size**	**TPR@FAR**, % ↑						
			10e-6	10e-5	10e-4	0.001	0.01	0.05	0.1
GRAY32	HPatches[26]	128	6.35	10.25	16.63	28.20	48.78	68.14	77.40
GRAY32	LF-Net[47]	128	7.72	**12.38**	19.80	32.24	51.79	69.40	77.58
GRAY32	HCKD	128	7.08	11.30	18.48	31.32	52.47	70.38	78.67
GRAY32	D2-Net[48]	128	7.18	11.46	18.80	31.82	52.60	71.10	79.59
GRAY64	D2-Net[48]	128	7.79	12.35	19.74	32.75	53.72	72.24	80.38
RGB64	HCKD	128	4.47	9.32	17.04	31.13	56.60	77.15	85.45
RGB64	LF-Net[47]	128	5.35	10.91	20.14	35.13	60.29	79.25	86.59
LUV64	D2-Net[48]	128	4.41	10.31	20.56	36.35	61.35	79.85	86.92
RGB64	D2-Net[48]	128	4.79	10.65	19.45	35.85	61.62	80.21	87.33
LAB64	D2-Net[48]	128	5.22	11.13	20.69	36.39	62.23	80.61	87.49
HLS64	D2-Net[48]	128	4.16	10.39	19.49	35.45	61.70	80.62	87.66
HSV32	HCKD	128	4.54	9.58	17.84	33.80	58.93	78.95	86.68
HSV32	LF-Net[47]	128	5.23	10.93	20.33	35.18	59.40	79.15	86.92
HSV32	D2-Net[48]	128	5.06	10.93	20.51	36.78	61.78	80.27	87.30
HSV64	LF-Net[47]	128	4.37	8.78	17.16	33.10	59.39	79.58	87.11
HSV64	D2-Net[48]+HCKD	128	3.80	9.55	18.59	35.20	60.85	79.97	87.15
HSV64	HCKD	128	3.53	7.99	16.76	33.87	60.77	79.94	87.41
HSV64	D2-Net[48]	128	4.76	12.01	21.81	37.34	62.37	80.87	87.87
HSV64+-	D2-Net[48]	128	6.29	11.77	20.51	35.24	58.59	76.44	83.98
HSV64-+	D2-Net[48]	128	7.20	10.76	16.59	29.80	52.95	72.61	81.06
HSV64--	D2-Net[48]	128	**7.88**	10.53	16.21	29.27	50.04	67.80	76.00
HSV64	D2-Net[48]	64	4.27	10.12	19.19	35.56	60.66	79.33	86.54
HSV64	D2-Net[48]	256	4.74	11.05	**22.13**	**38.61**	63.11	81.12	87.91
HSV64	D2-Net[48]	512	4.65	10.59	20.70	36.37	**63.67**	**81.35**	**87.95**
HardNet++[27]	HPatches[26]	128	7.80	11.66	17.60	28.04	46.04	63.43	71.84
SOSNet[36]	HPatches[26]	128	7.31	10.79	16.38	27.05	47.88	66.92	75.60
TriNet[64]	-	128	1.23	6.51	14.74	32.82	32.82	56.41	69.88
MLFN[57]	-	1024	3.06	6.14	11.75	22.98	44.39	68.34	80.52
AGW[30]	-	2048	2.79	5.24	9.74	19.00	40.10	68.22	80.65
OSNet[32]	-	512	3.23	5.98	11.35	22.46	44.76	70.01	81.45
HACNN[33]	-	1024	2.96	11.10	20.21	33.46	54.34	74.43	83.93
JDE[25]	-	512	2.88	6.87	14.32	28.25	55.17	81.53	90.10

D ODESA Properties

D.1 ODESA Object Representation

In this section, we would like to trace the evolution of the object representation from LLFD to ODESA models. For this purpose, the features introduced in Section 5.1 will be discussed. To facilitate the comparison, the following models were selected: HardNet++[27], SOSNet[36], GRAY32/128, HSV32/128. All of them accept 32x32 patches as their input and output 128-dimensional embedding vectors projected on the unit hypersphere. Each model is represented by a row in Fig. D.1.

Starting with the top two rows corresponding to HardNet++ and SOSNet models, one could notice that the features depicted in the two left-most columns are quite similar. Apart from certain particularities, the corresponding t-SNE projections also exhibit essential agreement. The manifold extent perceived from both Figs. D.1(b) and (e) agrees well with the data shown in Fig. 4 for HardNet++ model. Unlike the case of HSV64/128, which is shown in Fig. 2(a), the t-SNE projections of HardNet++ and SOSNet indicate that some manifolds get split into a number of segments, e.g. the objects #364 and 132, or some individual elements become detached from the rest, e.g. the object #132. This observation agrees with the presence of sharp maxima in Figs. D.1(c) and (f) for a number of objects. Finally, each solid curve in Figs. D.1(b) and (e) exhibit quite restricted variation along the vertical axis compared to the case of Fig. 2(b).

Model GRAY32/128 differs from SOSNet solely by the utilization of our own set of patches during training, see Section 3.2 for details. The loss function and CNN topology are identical. In the case of GRAY32/128, this set of patches was converted to grayscale. This transition results in noticeable changes. In the first place, the manifolds in t-SNE projection do not show any obvious signs of abrupt evolution. The peaks, which are still observed in Fig. D.1(i), become rather suppressed in comparison with HardNet++ and SOSNet models. Next, the manifold extent represented by the solid lines and the distance to the nearest angle-wise neighbors get scaled down. Finally, t-SNE projections corresponding to the objects #162 and 461 start to exhibit some structure. It is also reflected by considerably broader intervals, where corresponding solid curves vary in Fig. D.1(h).

The last option of HSV32/128 differs from GRAY32/128 by the preserved color information and some minor modifications of CNN topology. Perhaps, the latter factor could be disregarded, since the difference is negligible, see Tab. C.1 for details. Among obvious differences between these two models one could list the following. The manifold extent in the former case gets further scaled down. Some peaks in Fig. D.1(l) get significantly suppressed in comparison to their counterparts shown in Fig. D.1(i). As for the comparison between HSV32/128 and HSV64/128, the former appears to already posess all essential features of the latter. This observation is also supported by the data from Table C.1, where these models achieve quite similar values. Single visible difference concerns a

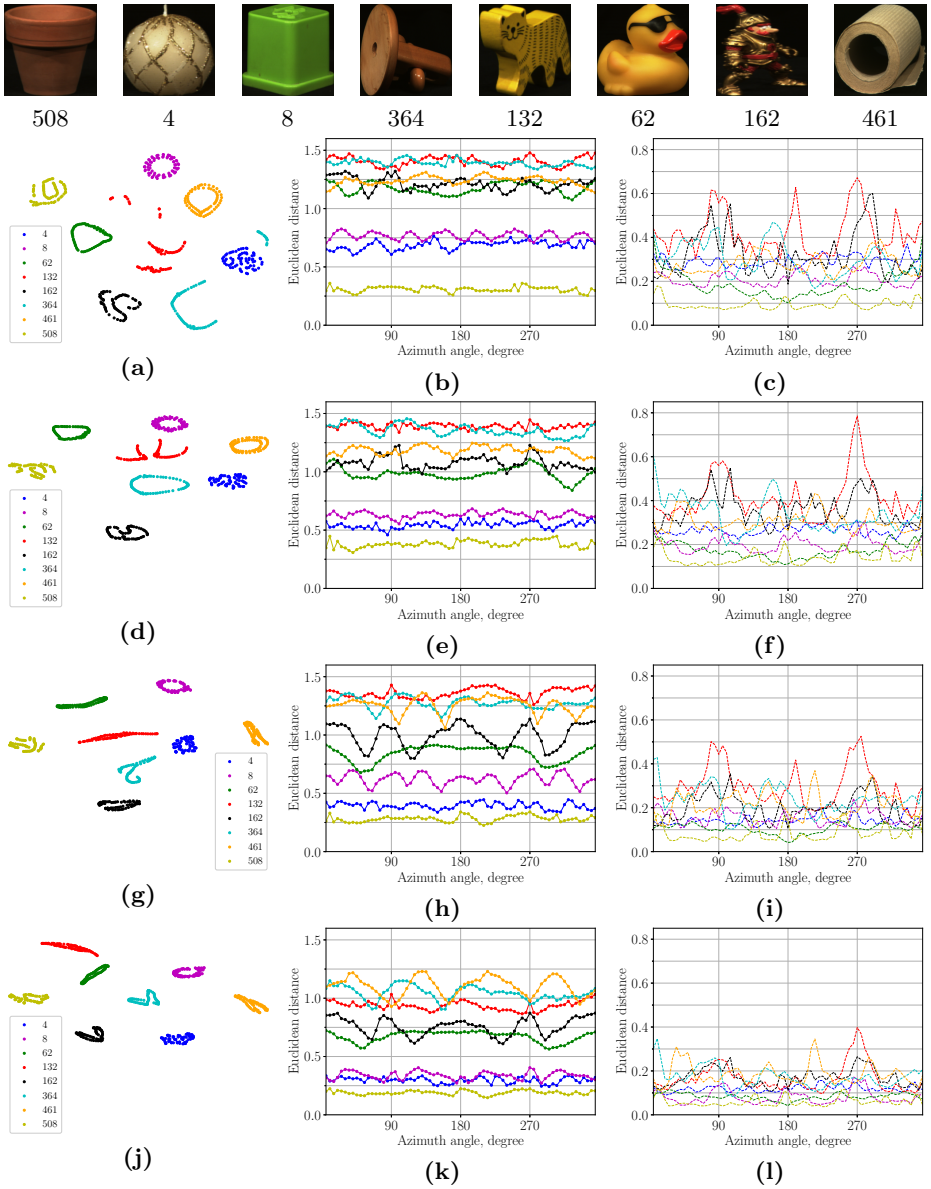


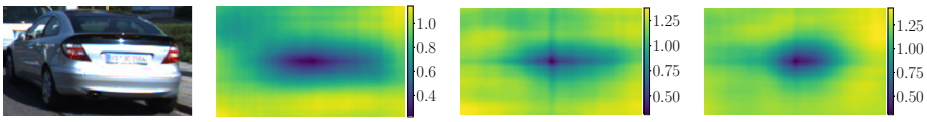
Fig. D.1. Object mapping into the metric space: (a-c) HardNet++[27], (d-f) SOSNet[36], (g-i) GRAY32/128, (j-l) HSV32/128. (a, d, g, j) t-SNE projections for all embedding vectors corresponding to the objects depicted at the top. The distances from a given sample embedding vector to (b, e, h, k) its furthest element of the same manifold; (c, f, i, l) its two closest angle-wise neighbors halved. Best viewed in color.

number of solid curves in Fig. 2(b), which tend to be smoother and, perhaps for this reason, exhibit several narrow minima. The latter may be speculated to serve as a sign of a rather developed manifold structure.

D.2 The Influence of Data Augmentation

It was mentioned in Section 5 that the utilization of random keypoint transformations, as described in HPatches[26], considerably reduces the sensitivity of our models to the patch periphery. Figure D.5 serves as another illustration of this effect. As this augmentation technique tends to affect several aspects of resulting embedding functions, we summarize some of them below. Along with the random transformations of keypoints we also consider the utilization of random patch flips during training as they expected to produce somewhat similar effects.

As another evidence of a strong effect from these two data augmentation options, one could regard the results shown in Table C.2 for the set of models, which could be referred together to as HSV64**/128. Here the first asterisk corresponds to the application of the random keypoint transformations, while the second one represents the random patch flips. The '-' signs indicate disabled option, whereas '+' or the sign absence stand for enabled one. From the comparison between HSV64/128, which is equivalent to HSV64++/128, and HSV64--/128, it is evident that the former model benefits significantly from the data augmentation whenever FAR$\geqslant 10^{-5}$. The intermediate options of HSV64-+/128 and HSV64+-/128 indicate that the effect from the random transformations of keypoint is stronger. At the same time, the opposite tendency takes place for the FAR value of 10^{-6}. Moreover, HSV64--/128 achieves the highest TPR@FAR10e-6 value of 7.88% among all entries from Table C.2. Perhaps, these results provide the best inside into the cause of such behavior when considered along with the evolution of $D_m(\delta x, \delta y)$ shown in Fig. D.2. The data depicted in Fig. D.2(b), (c), and (d) were calculated according to Eq. 7 while employing HSV64++/128, HSV64-+/128, and HSV64--/128 as embedding functions $f(\cdot)$, respectively. The last two options make the minimum of $D_m(\delta x, \delta y)$, which corresponds to low $(\delta x, \delta y)$ values, much narrower compared to the case of our default augmentation routine, i. e. HSV64++/128. Such behavior is rather typical as evident from Figs. D.3 and D.4. We assume the random transformations of keypoint scale and orientation to produce the effects similar to those shown in Fig. D.2(b), (c), and (d), on condition that the distance between embedding vectors is estimated as a function of scale factor or rotation angle. If it is, indeed, the case, one could expect that narrower minima imply better discrimination capability for corresponding embedding functions. At the same time the models exhibiting broader minima shall show better generalization, i. e. become less susceptible to object appearance variation. In our opinion, the results for HSV64**/128 models presented in Table C.2 fit rather well into the picture described above. On condition that this assumption is correct, the utilization of random keypoint transformations could serve as an additional tuning parameter for ODESA-like embeddings.



(a) **(b)** **(c)** **(d)**

Fig. D.2. The influence of the training routine on the distribution of misalingment distances $D_m(\delta x, \delta y)$ calculated according to Eq. 7. The following factors were considered: the random transformations of keypoint with respect to their position, scale and orientation; and the utilization of random flips during training. (a) image patch, which corresponds to the ground truth bounding box. $D_m(\delta x, \delta y)$ corresponding to (b) the default training routine, i.e. HSV64/128 model; (c) HSV64-+/128 model, where the random transformations were excluded; (d) HSV64--/128, where both the random transformations and flips were disabled. Best viewed in color.

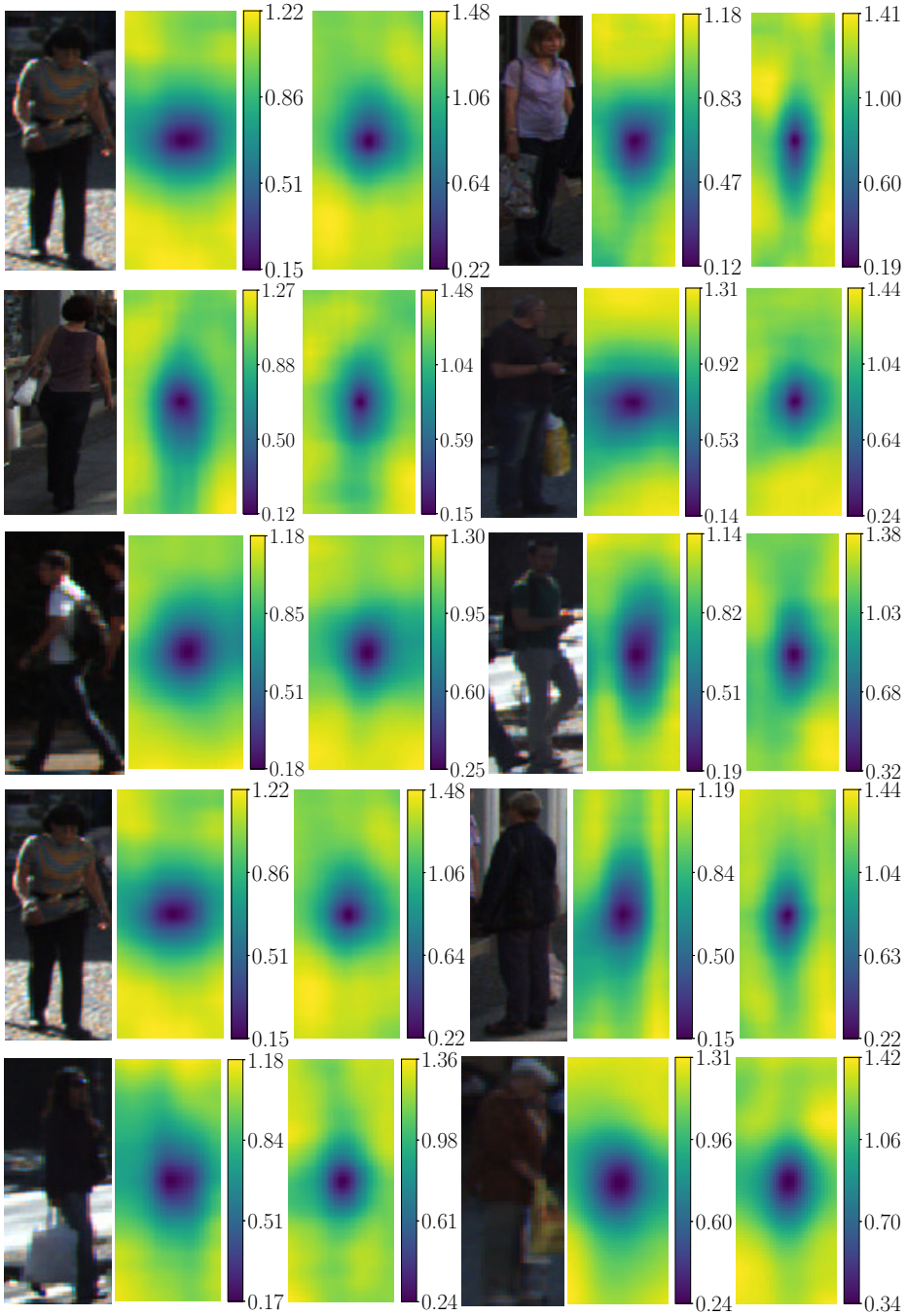


Fig. D.3. Examples of $D_m(\delta x, \delta y)$ for the samples corresponding to *pedestrian* category: Each set of examples contains from left to right: the content of the ground truth bounding box; $D_m(\delta x, \delta y)$ obtained by means of HSV64/128; $D_m(\delta x, \delta y)$ produced by HSV64--/128. Best viewed in color.

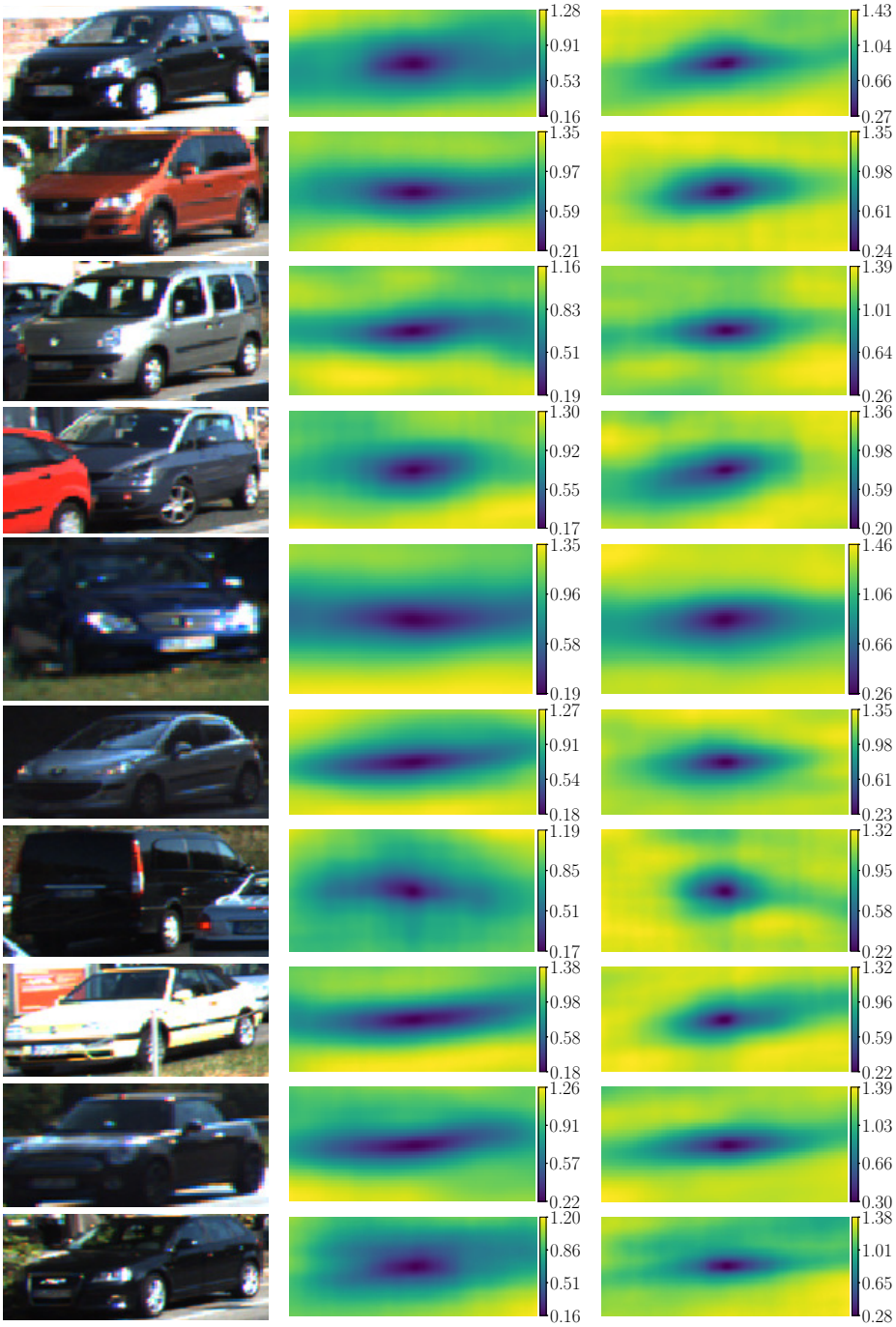


Fig. D.4. Additional examples of $D_m(\delta x, \delta y)$. Each row contains from left to right: the content of the ground truth bounding box; $D_m(\delta x, \delta y)$ obtained by means of HSV64/128; $D_m(\delta x, \delta y)$ produced by HSV64--/128. Best viewed in color.

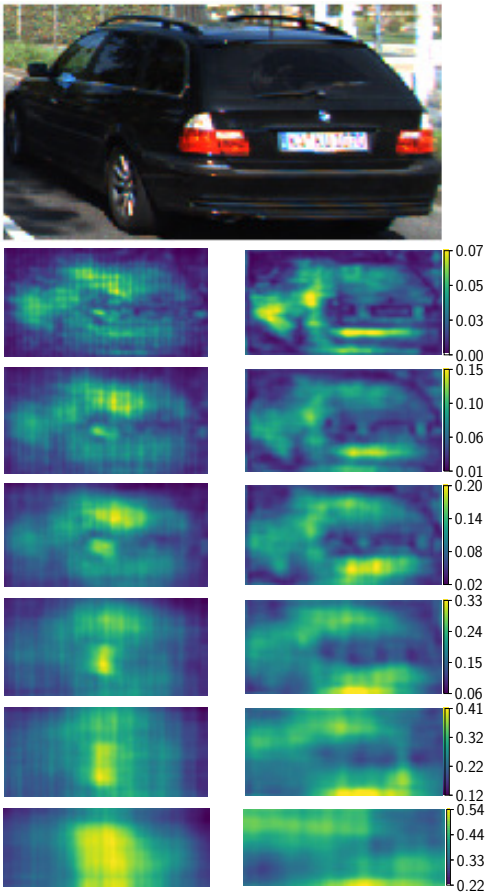


Fig. D.5. The influence of the occluder size s on $D_o(\delta x, \delta y)$ calculated using Eq. 6. The top image represents the content of the ground truth bounding box. The rows representing $D_o(\delta x, \delta y)$ correspond to the occluder size amounting 5%, 10%, 15%, 25%, and 35% of the bounding box height h. They are ordered from top to bottom. The left column was produced by means of HSV64/128 model, which corresponds to the default training routine. The right one was obtained using HSV64--/128, where the random transformations of keypoints and flips were disabled during training. Best viewed in color.