Learning Similarity Metrics for Dynamic Scene Segmentation

Supplementary material

Damien Teney[1] Matthew Brown[2] Dimitry Kit[2] Peter Hall[2]
[1]Carnegie Mellon University [2]University of Bath
dteney@andrew.cmu.edu {m.brown,d.m.kit,maspmh}@bath.ac.uk

This document provides additional results and details on the protocols used in our experimental evaluation.

1. Dynamic texture segmentation (*SynthDB and Dyntex datasets)*

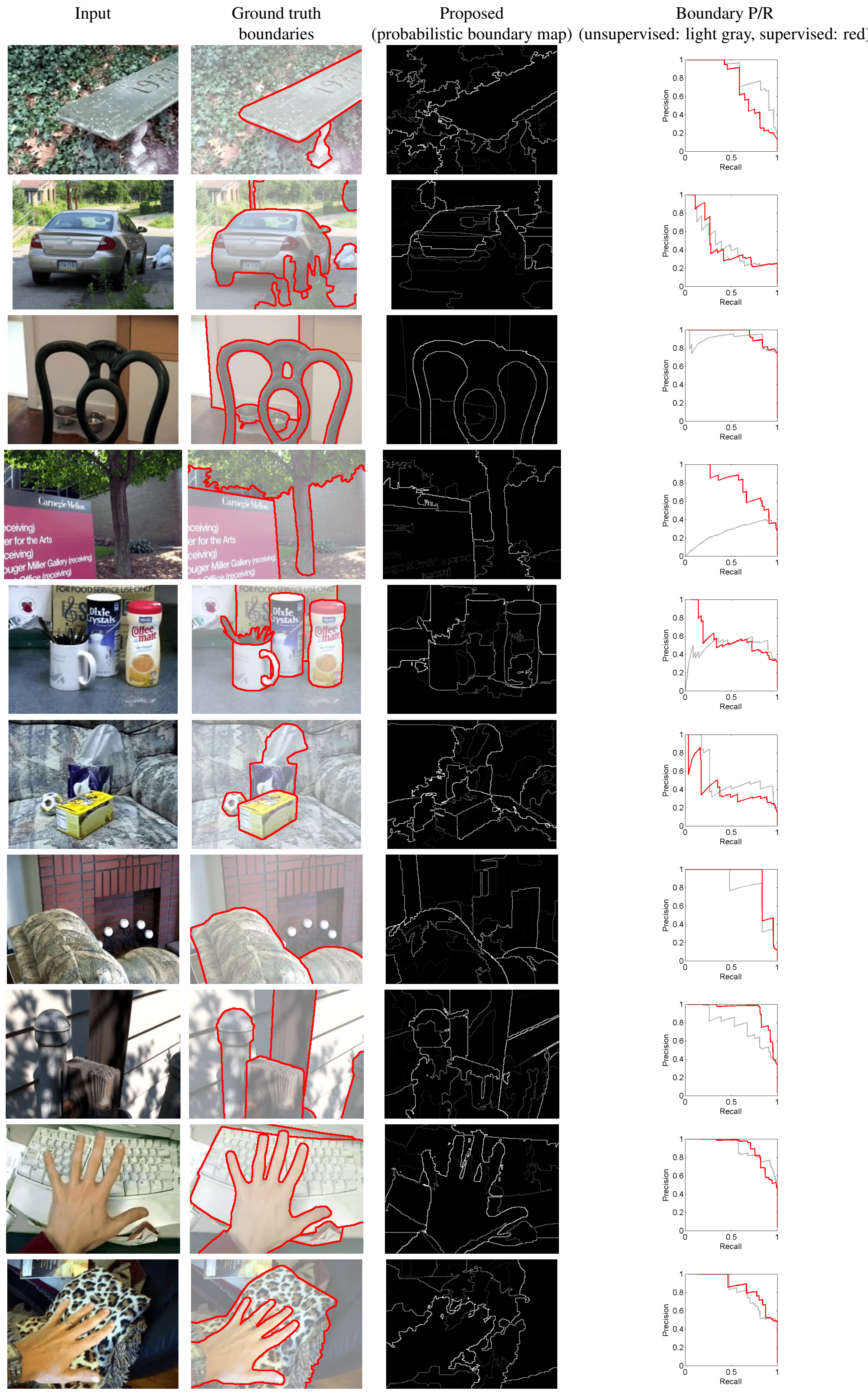
The data from the SynthDB dataset used for training consists of the 99 sequences featuring 2 textures [1]. Each texture is labeled as one of these 12 classes: grass, jellyfish, pond, boiling, escalator, fire, river-far, river, steam, plant-a, plant-i, and sea-far.

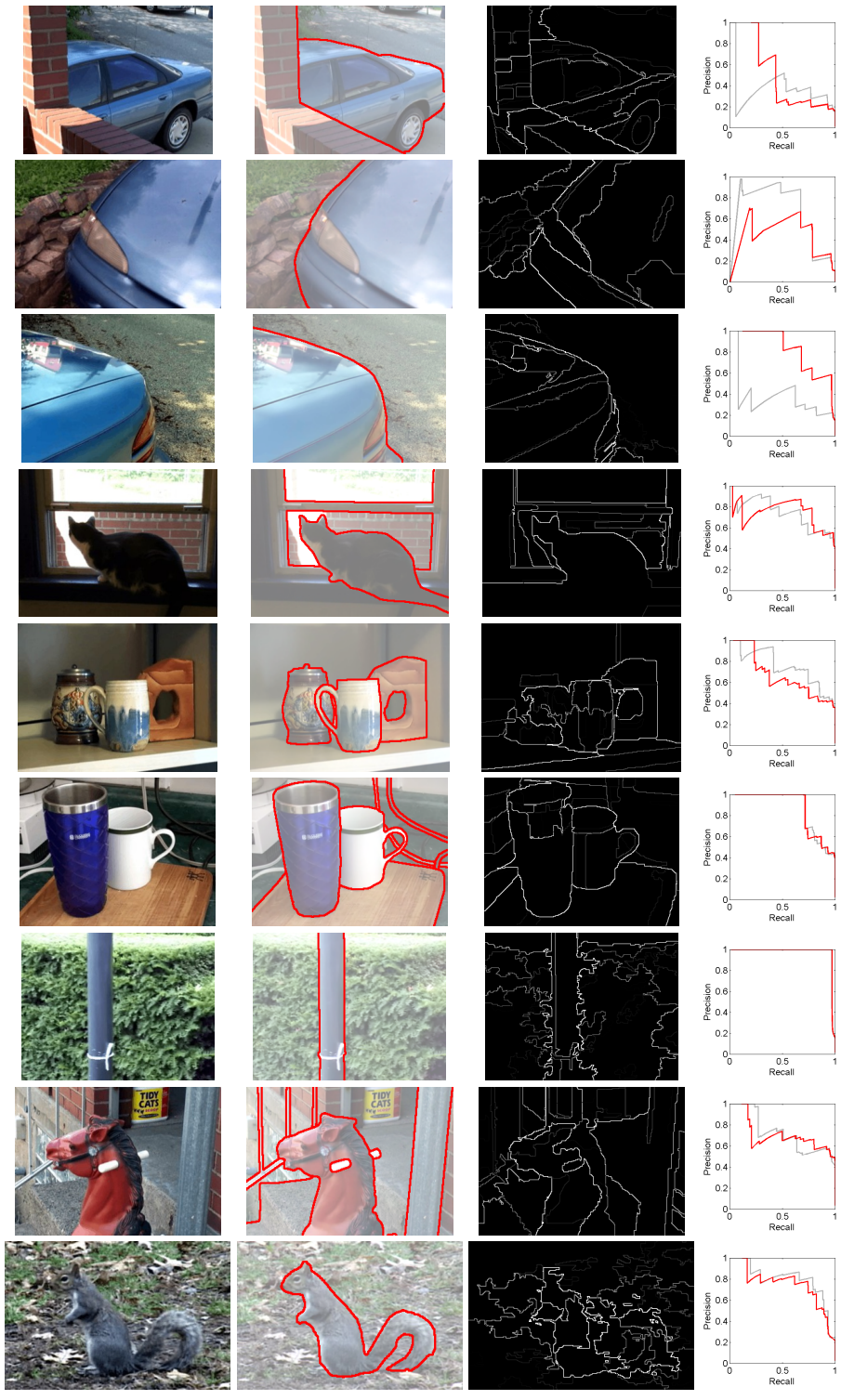
All results reported on the SynthDB and Dyntex datasets used a single scale, *i.e.* S=1. In our experiments, the use of multiple scales did not have a significant impact on the results for this task (unlike with object and motion segmentation). We believe it is a consequence of the limited diversity of training data. Using multiple scales is more likely to be beneficial if the model was trained on scenes that include dynamic textures of varying spatial extent.

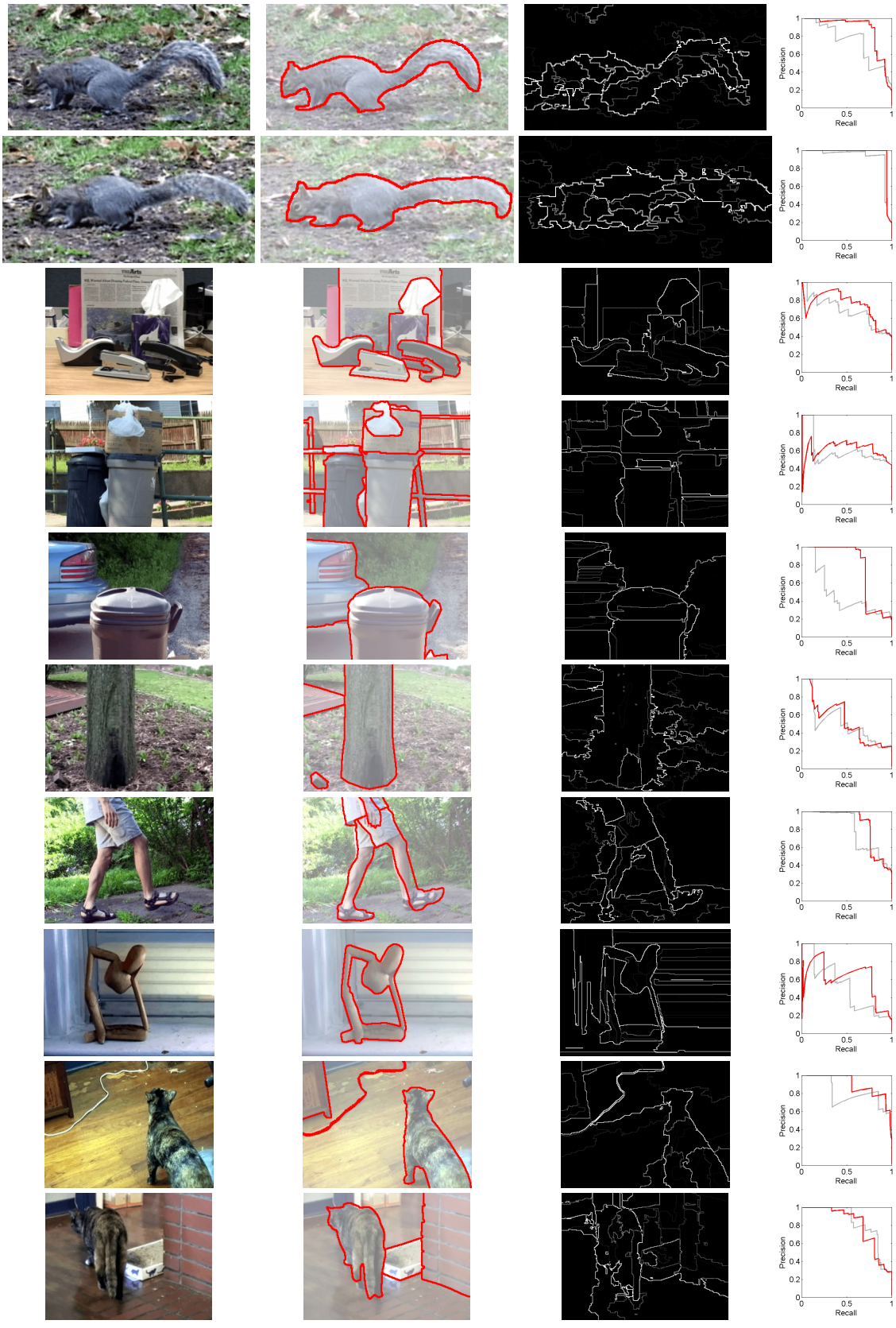
The results on the Dyntex sequences reported in the main paper correspond to the last level of the segmentation, *i.e.* when only 2 segments are remaining. Practically, one segment correspond to the main dynamic texture of the scene, the other to the background (or more static elements, keeping in mind that the camera is moving in some sequences).

2. Object boundaries (*CMU dataset)*

Our evaluation on the CMU dataset for motion boundaries [5] use the same protocol as in [5, 3, 8]. We use half of the sequences (even/odd ones) as training and test sets, and average the results over two runs after exchanging the sets. Most methods [5, 3, 8] do not enforce closed segments but only predict boundaries (*i.e.* curves in the image), and the performance is measured with the precision and recall of the recovered boundaries (in our case, the outline of the segments). These boundaries are compared, not against the hand-drawn ground truth, but against the "candidate" boundaries closest to the ground truth [5]. In our case, these are the boundaries of the small segments after the initial "bootstrapping" stage that perform merges using pixel-wise differences. We turn our hierarchical segmentation into a probabilistic boundary map in the same way as in [9]. The strength of a boundary is proportional to the highest level it appears in. For example, the boundary remaining between the two large segments at the last iteration of the segmentation is assigned the largest strength. Results are given below for each sequence of the dataset.

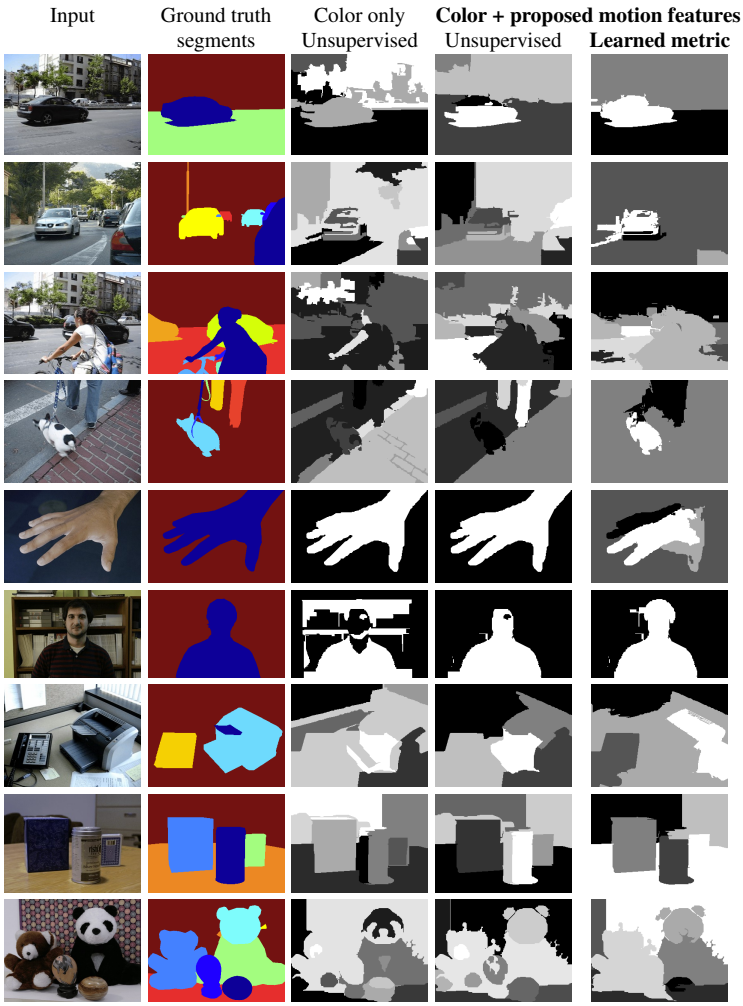

Input
Ground truth boundaries
Proposed (probabilistic boundary map)
Boundary P/R (unsupervised: light gray, supervised: red)
Precision
Recall


Precision
Recall
0
0.2
0.4
0.6
0.8
1
0.5
TIDY CATS


Precision
Recall
0
0.2
0.4
0.6
0.8
1
0.5

3. Rigid motion segmentation (*MIT dataset)*

Our evaluation on the MIT human-labeled dataset [4] uses the same protocol as in [9, 6, 7]. We compare the produced segmentation with the ground truth using the Rand index. We measure the Rand index on one frame in the middle of the sequence (the same frames as in [6]). The sequences contain varying numbers of segments. Although the Rand index has a limited sensibility to the number of segments being compared, we do need to select a segmentation level to report. It is chosen as the best-matching one (*i.e.* of highest Rand) in the range of levels with N +/- 3 segments, where N is the number of segments in the ground truth. Results are given below for each sequence of the dataset.



		Rand index (%)									
Method		Avg.	Car1	Car2	Car3	Dog	Hand	Person	Phone	Table	Toy
Features	*Metric*										
Color histograms only	Unsupervised	68.5	77.4	58.9	61.3	49.5	97.9	64.1	48.5	85.5	73.7
Color + filter-based motion	Unsupervised	78.8	86.3	53.5	83.0	60.4	97.9	91.7	63.4	88.7	83.9
Color + filter-based motion	Learned, logistic regression	76.3	59.5	71.5	82.8	59.0	97.1	97.0	55.0	90.6	74.6
Color + filter-based motion	Learned, ITML [2]	70.3	61.3	48.9	84.1	69.5	80.1	58.8	64.0	86.7	79.5
Color + motion	Learned as proposed, $S = 1$	**83.0**	97.6	71.0	86.2	89.4	79.5	86.5	58.9	88.8	89.1
Color + motion	Learned as proposed, $S = 2$	81.1	96.5	64.7	87.2	86.4	83.6	85.9	55.9	92.9	76.9
Layers++ [6]		77.5	61.2	51.2	77.8	96.4	81.4	98.6	56.7	90.9	83.2
nLayers [7]		82.3	83.6	58.9	76.6	97.4	88.1	94.4	57.8	97.9	85.8
Teney and Brown [9]		**83.2**	90.0	64.5	79.6	95.9	94.5	83.4	56.1	93.7	90.8

References

[1] A. Chan and N. Vasconcelos. Modeling, clustering, and segmenting video with mixtures of dynamic textures. *IEEE Trans. Pattern Anal. Mach. Intell.*, 30(5):909–926, 2008. 1

[2] J. V. Davis, B. Kulis, P. Jain, S. Sra, and I. S. Dhillon. Information-theoretic metric learning. In *ICML*, pages 209–216. ACM, 2007. 6

[3] X. He and A. Yuille. Occlusion boundary detection using pseudo-depth. In *ECCV*, pages 539–552. 2010. 1

[4] C. Liu, W. T. Freeman, E. H. Adelson, and Y. Weiss. Human-assisted motion annotation. In *CVPR*, 2008. 5

[5] A. N. Stein and M. Hebert. Occlusion boundaries from motion: Low-level detection and mid-level reasoning. *IJCV*, pages 325–357, 2009. 1

[6] D. Sun, E. Sudderth, and M. J. Black. Layered image motion with explicit occlusions, temporal consistency, and depth ordering. *Advances in Neural Information Processing Systems*, 23, 2010. 5, 6

[7] D. Sun, E. B. Sudderth, and M. J. Black. Layered segmentation and optical flow estimation over time. In *CVPR*, pages 1768–1775, 2012. 5, 6

[8] P. Sundberg, T. Brox, M. Maire, P. Arbelaez, and J. Malik. Occlusion boundary detection and figure/ground assignment from optical flow. In *CVPR*, pages 2233–2240, 2011. 1

[9] D. Teney and M. Brown. Segmentation of Dynamic Scenes with Distributions of Spatiotemporally Oriented Energies. In *BMVC*, 9 2014. 1, 5, 6