

ScaleLSD: Scalable Deep Line Segment Detection Streamlined

Supplementary Material

Zeran Ke¹ Bin Tan² Xianwei Zheng³ Yujun Shen² Tianfu Wu⁴ Nan Xue^{†1,2}
¹School of Computer Science, Wuhan University ²Ant Group
³LIESMARS, Wuhan University ⁴Department of ECE, NC State University



Figure A1. Some training examples on the generated synthetic dataset and the real SA1B [4] dataset.

A. Additional Implementation Details

Training Data and Pipelines. Our training pipelines are similar with previous studies [2, 7, 14]. In the bootstrapping stage learns the concept of line segments from the synthetic images using 8 simple primitives as shown in Fig. A1a. With the bootstrapping model, we move forward to the small-scale Wireframe [3] dataset to learn the line segments in real-world images, and use this model to achieve the largest-scale training of LSD on the SA-1B [4] dataset, which contains 10 million real-world image samples as shown in Fig. A1b.

Network Architecture. Our network architecture is simple, follows the best practices of vision transformers for dense predictions [10]. In detail, given a batch B of RGB images with shape 512×512 , a ViT-Base model is used to extract 1024 tokens for dense prediction of HAT fields and junction heatmaps. DPT head is applied to first transform the 1024 tokens into high-resolution feature maps with the shape of $[B \times N \times 256 \times 256]$, and then predict the HAT fields and junction heatmaps using 1×1 convolution layers. In our model, there are no neural modules for the verification of line segments, which has greatly simplified the training and inference pipeline compared to HAWPv3 [14].

Loss Functions. We use the \mathcal{L}_1 loss function for the regression of the distance field \mathcal{A}_d , the angle field \mathcal{A}_a and the residual distance $\mathcal{A}_{\Delta d}$, denoted by $(\mathcal{L}_d, \mathcal{L}_a, \mathcal{L}_{\Delta d})$. The loss is computed across the foreground points only based on the mask of foreground pixels. We use binary cross-entropy loss $\text{BCE}(\cdot, \cdot)$ for the regression of the endpoints and use loss \mathcal{L}_1 for the regression of the offset field, record as $(\mathcal{L}_j, \mathcal{L}_o)$. We set the weights of each loss to $(\lambda_d, \lambda_a, \lambda_{\Delta d}, \lambda_j, \lambda_o) = (1.0, 1.0, 1.0, 8.0, 0.25)$, and the total loss of our model is

$$\mathcal{L} = \underbrace{\lambda_d \mathcal{L}_d + \lambda_a \mathcal{L}_a + \lambda_{\Delta d} \mathcal{L}_{\Delta d}}_{\text{HAT Field Learning}} + \underbrace{\lambda_j \mathcal{L}_j + \lambda_o \mathcal{L}_o}_{\text{Junction Learning}}. \quad (\text{A1})$$

The setting of λ_j and λ_o follows HAWPv3 [14] to balance the significant magnitude difference of these two loss terms.

Inference. Our `ScaleLSD` takes any RGB/grayscale image as input, predicts the HAT fields and junction heatmaps using a neural network, and decodes the hat fields and junction heatmaps into sparse line segments. In the decoding stage, the junction heatmaps are first processed by a max-pooling layer with a window size of 3 to suppress the non-maximal predictions, then we extract the top- K pixels as the coarse junction predictions. When the junction score (*i.e.*, heatmap value) of any pixel is less than $\tau_j \in (0, 1)$, it is discarded. The junction score threshold τ_j is set to 0.008 for training and pseudo-label generation and is set to 0.1 for inference and evaluation. For the finally-kept coarse junctions, we apply the learned short-range offset to obtain final junctions with sub-pixel localization accuracy. With the extracted junction, we decode the line segments by matching them to the line segment fields (computed by the HAT fields) according to Eq. (2) of our main paper. The distance threshold τ_{dist} is set to 10 pixels, rejecting low-quality predictions in the HAT fields from the final predictions. By matching the junction to lines, the line segments whose support pixels are larger than τ_l are kept as the final predictions. Here, we set τ_l to 10 for training and pseudo-label generation and is set to 5 for inference and evaluation.

Details on 3D Line Reconstruction. In 3D line reconstruction, we found the threshold of top- K should be increased to 2048 because the buildings usually have more structural information. For the evaluation, we follow the protocol provided by DTU dataset [1] to compute the Chamfer distance between the predicted line segments (sampled in 128 points per line) and the groundtruth surface model. The accuracy (ACC-L) and the completeness (COMP-L) are computed to measure the reconstruction quality. We also add the number of 3D line segments as a reference. We reconstruct the 3D lines using LiMAP [6] by switching the line segment detectors. The line matching module is their built-in GlueStick [9] implementation for all detectors.

B. Visualization of VP Estimation



Figure A2. The illustration of vanishing points estimation. Lines belong to the same one vanishing point are labeled with the same color. Top row shows the results of the Manhattan scenes in the YUD+ [5] dataset and bottom row shows the results of non-Manhattan scenes in the NYU-VP [11] dataset.

We visualize results of vanishing points estimation by drawing the parallel line segments associated with different vanishing points in different colors. Fig. A2 shows that, our method could robustly estimate vanishing points in both Manhattan and Non-Manhattan scenes. To better show the results, the line segments that are not associated with any vanishing points are hidden to display.

C. Visualization of Line Matching

Line segments matching is a challenge task due to common situations of changes of view and illumination, occlusions, background changes, repeatable structures, and textures. Two typical challenging cases for repeatable structures and intensive illumination changes between the input image pairs are shown in Fig. A3. As shown, because our `ScaleLSD` significantly improves detection performance in terms of detection completeness, the applied line segment matcher (*i.e.*, GlueStick [9]) could leverage the global information conveyed in the structural line segment representation for better matching.

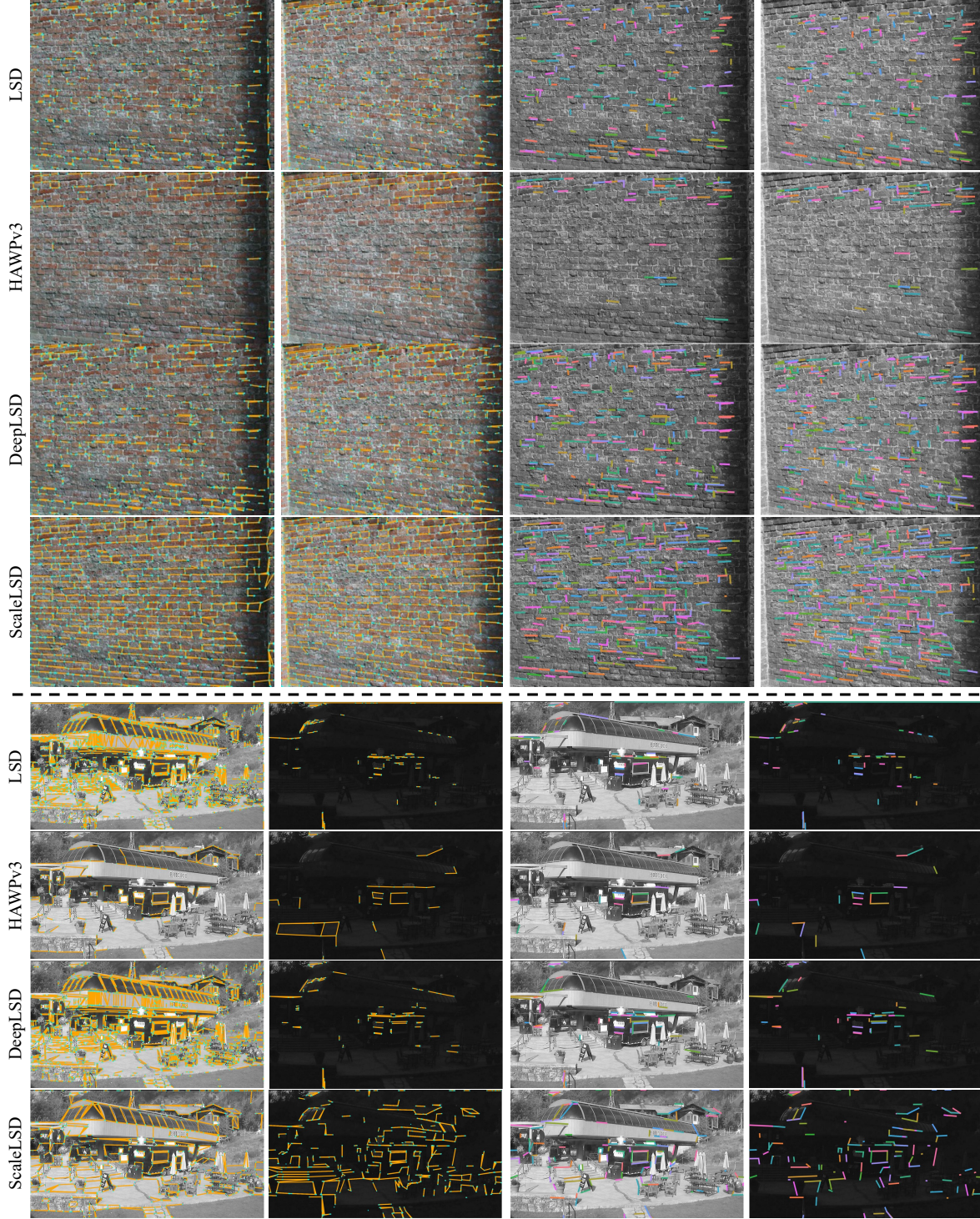


Figure A3. Challenging examples of line segment matching. For each case, from left to right, we first show the detection results for the two-view input images, and then show the matched line segments. Top: we show the challenge pair of images that have similar structure and texture as well as change of viewpoint. Bottom: we show the challenge pair of images that have significant illumination changes. Lines with the same color in the last two images means the matched pairs.

D. Video results of 3D Line Reconstruction

The attached video, [scalelsd.mp4](#), as a supplementary video for Fig. 6 of our main paper, provides a vivid comparison for 3D line reconstruction.

E. Additional Results

Except for the zero-short performance shown in the paper, we also provide the in-domain results on the Wireframe dataset [3] and the SA-1B dataset [4] in Tab. A1. We show some results in Fig. A4, Fig. A5, Fig. A6 and Fig. A7 to provide a more comprehensive and intuitive comparison.

Method	Wireframe					SA1B-1000				
	Rep-5 (S) ↑	Loc-5 (S) ↓	Rep-5 (O) ↑	Loc-5 (O) ↓	#Lines/Image	Rep-5 (S) ↑	Loc-5 (S) ↓	Rep-5 (O) ↑	Loc-5 (O) ↓	#Lines/Image
LSD [13]	0.383	2.198	0.719	1.028	441	0.432	2.179	0.665	1.153	614
SOLD ² [7]	0.566	2.039	0.805	1.135	116	0.480	2.226	0.688	0.954	97
HAWPv3 [14]	0.751	<u>1.487</u>	0.874	<u>0.841</u>	145	0.519	<u>1.680</u>	0.664	0.905	125
DeepLSD [8]	0.512	2.236	0.707	1.085	210	0.396	2.400	0.601	1.265	181
ScaleLSD@Wireframe(Ours)	0.723	1.694	<u>0.822</u>	0.897	413	<u>0.555</u>	1.856	<u>0.692</u>	0.955	419
ScaleLSD@SA1B(Ours)	<u>0.725</u>	1.466	0.820	0.837	764	0.634	1.535	0.728	<u>0.911</u>	<u>580</u>

Table A1. Evaluation of repeatability scores and localization errors on in-domain datasets. The image resolution are fixed to 512×512 in evaluation. Numbers with **bold-font** and underline indicate the best and the second best performance on specific metrics.

References

- [1] Henrik Aanæs, Rasmus Ramsbøl Jensen, George Vogiatzis, Engin Tola, and Anders Bjorholm Dahl. Large-scale data for multiple-view stereopsis. *IJCV*, 120(2):153–168, 2016. [2](#)
- [2] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superpoint: Self-supervised interest point detection and description. In *CVPRW*, pages 224–236, 2018. [1](#)
- [3] Kun Huang, Yifan Wang, Zihan Zhou, Tianjiao Ding, Shenghua Gao, and Yi Ma. Learning to parse wireframes in images of man-made environments. In *CVPR*, pages 626–635, 2018. [1](#), [4](#)
- [4] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloé Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross B. Girshick. Segment anything. In *ICCV*, pages 3992–4003. IEEE, 2023. [1](#), [4](#)
- [5] Florian Kluger, Eric Brachmann, Hanno Ackermann, Carsten Rother, Michael Ying Yang, and Bodo Rosenhahn. Consac: Robust multi-model fitting by conditional sample consensus. In *CVPR*, pages 4634–4643, 2020. [2](#)
- [6] Shaohui Liu, Yifan Yu, Rémi Pautrat, Marc Pollefeys, and Viktor Larsson. 3d line mapping revisited. In *CVPR*, pages 21445–21455. IEEE, 2023. [2](#)
- [7] Rémi Pautrat, Juan-Ting Lin, Viktor Larsson, Martin R. Oswald, and Marc Pollefeys. SOLD2: self-supervised occlusion-aware line description and detection. In *CVPR*, pages 11368–11378, 2021. [1](#), [4](#)
- [8] Rémi Pautrat, Daniel Barath, Viktor Larsson, Martin R. Oswald, and Marc Pollefeys. Deeplsd: Line segment detection and refinement with deep image gradients. In *CVPR*, pages 17327–17336. IEEE, 2023. [4](#), [5](#), [6](#), [7](#), [8](#)
- [9] Rémi* Pautrat, Iago* Suárez, Yifan Yu, Marc Pollefeys, and Viktor Larsson. GlueStick: Robust image matching by sticking points and lines together. In *ICCV*, pages 9706–9716, 2023. [2](#)
- [10] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *ICCV*, pages 12159–12168. IEEE, 2021. [1](#)
- [11] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, pages 746–760, 2012. [2](#)
- [12] Rafael Grompone von Gioi, Jérémie Jakubowicz, Jean-Michel Morel, and Gregory Randall. LSD: A fast line segment detector with a false detection control. *IEEE TPAMI*, 32(4):722–732, 2010. [5](#), [6](#), [7](#), [8](#)
- [13] R G von Gioi, J Jakubowicz, J M Morel, and G Randall. LSD: A Fast Line Segment Detector with a False Detection Control. *IEEE TPAMI*, 32(4):722–732, 2010. [4](#)
- [14] Nan Xue, Tianfu Wu, Song Bai, Fu-Dong Wang, Gui-Song Xia, Liangpei Zhang, and Philip H. S. Torr. Holistically-attracted wireframe parsing: From supervised to self-supervised learning. *IEEE TPAMI*, 45(12):14727–14744, 2023. [1](#), [4](#), [5](#), [6](#), [7](#), [8](#)



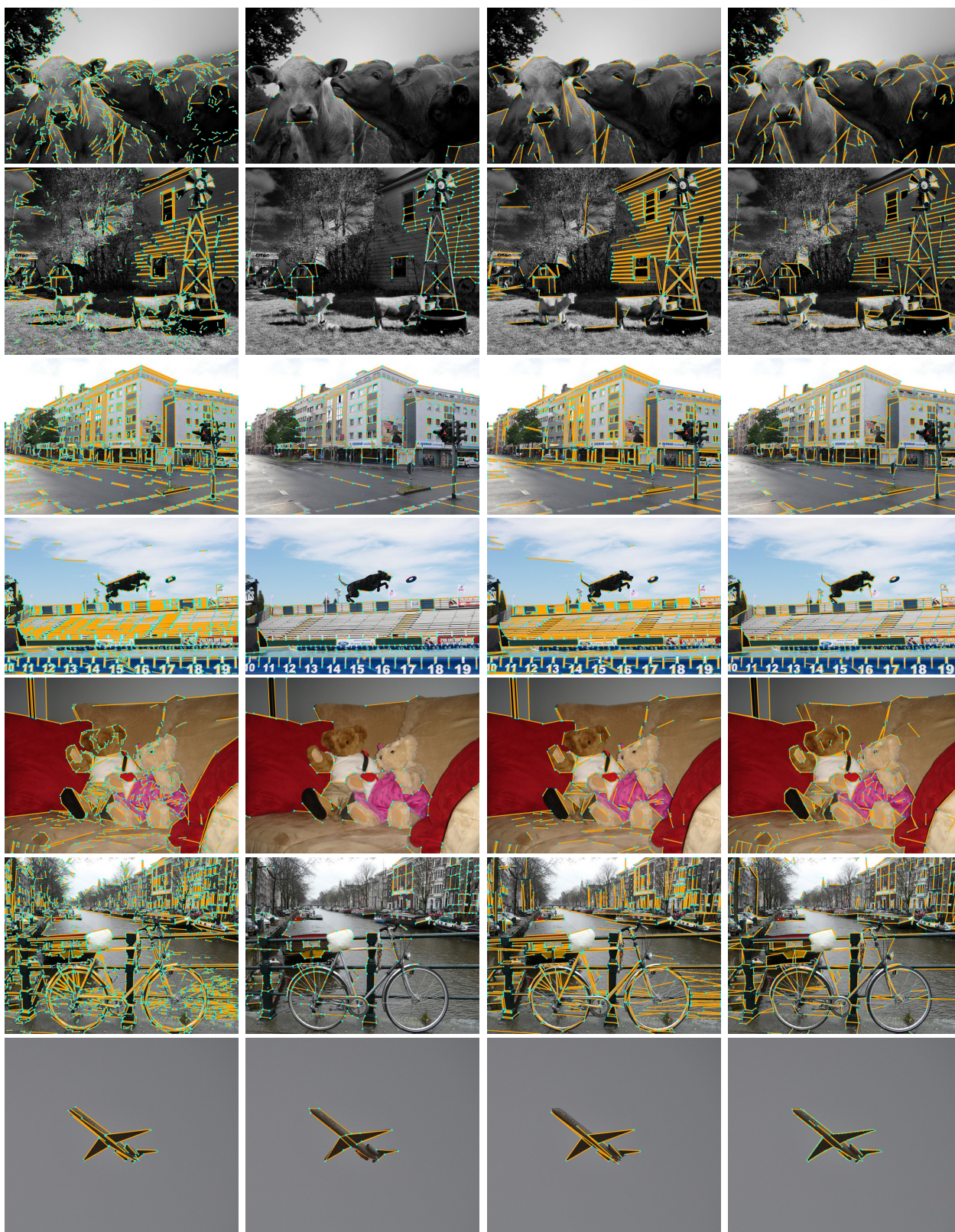
LSD [12]

HAWPv3 [14]

LSD [8]

ScaleLSD

Figure A4. Qualitative results of line segments detection.



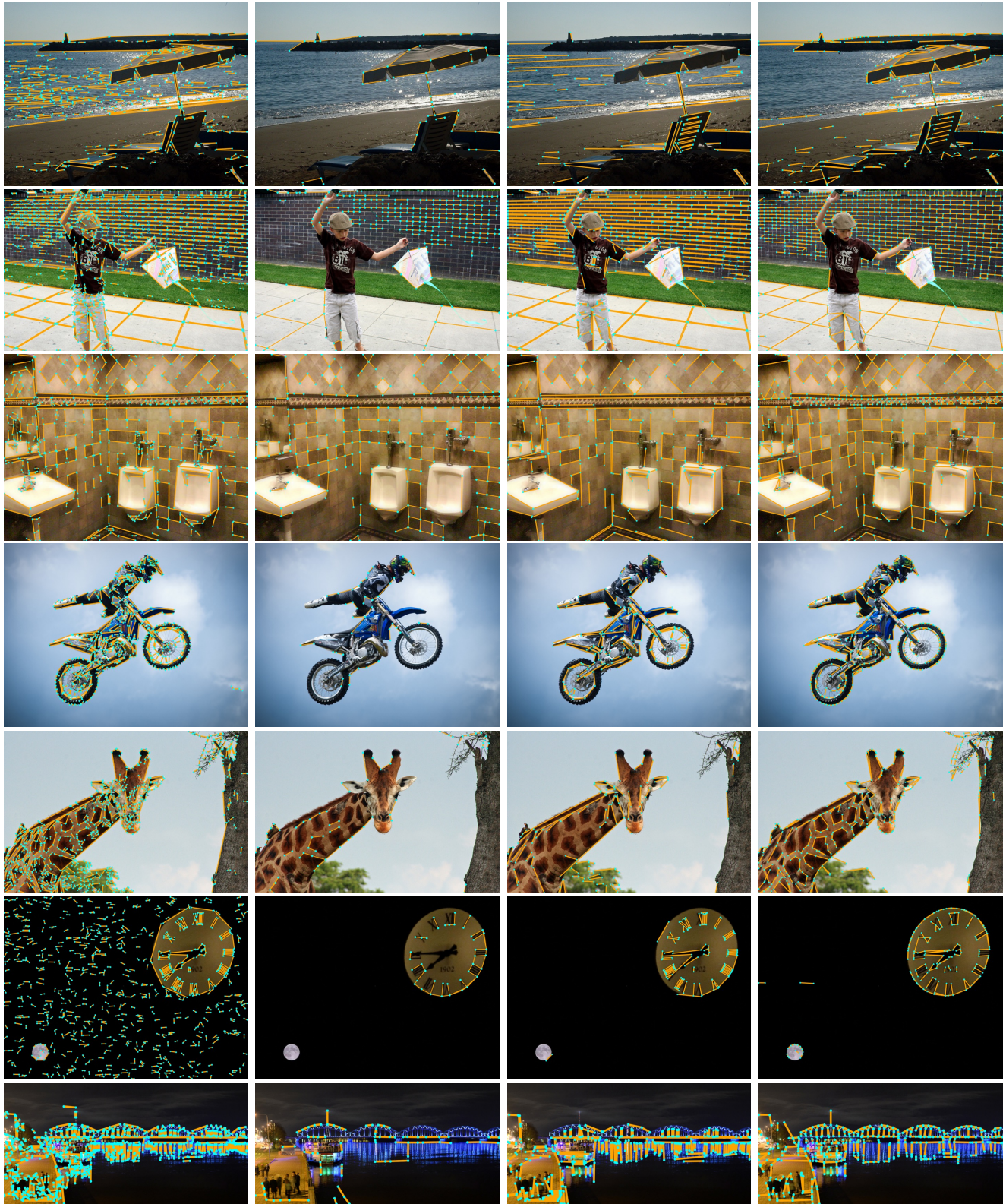
LSD [12]

HAWPv3 [14]

DeepLSD [8]

ScaleLSD

Figure A5. Qualitative results of line segments detection.



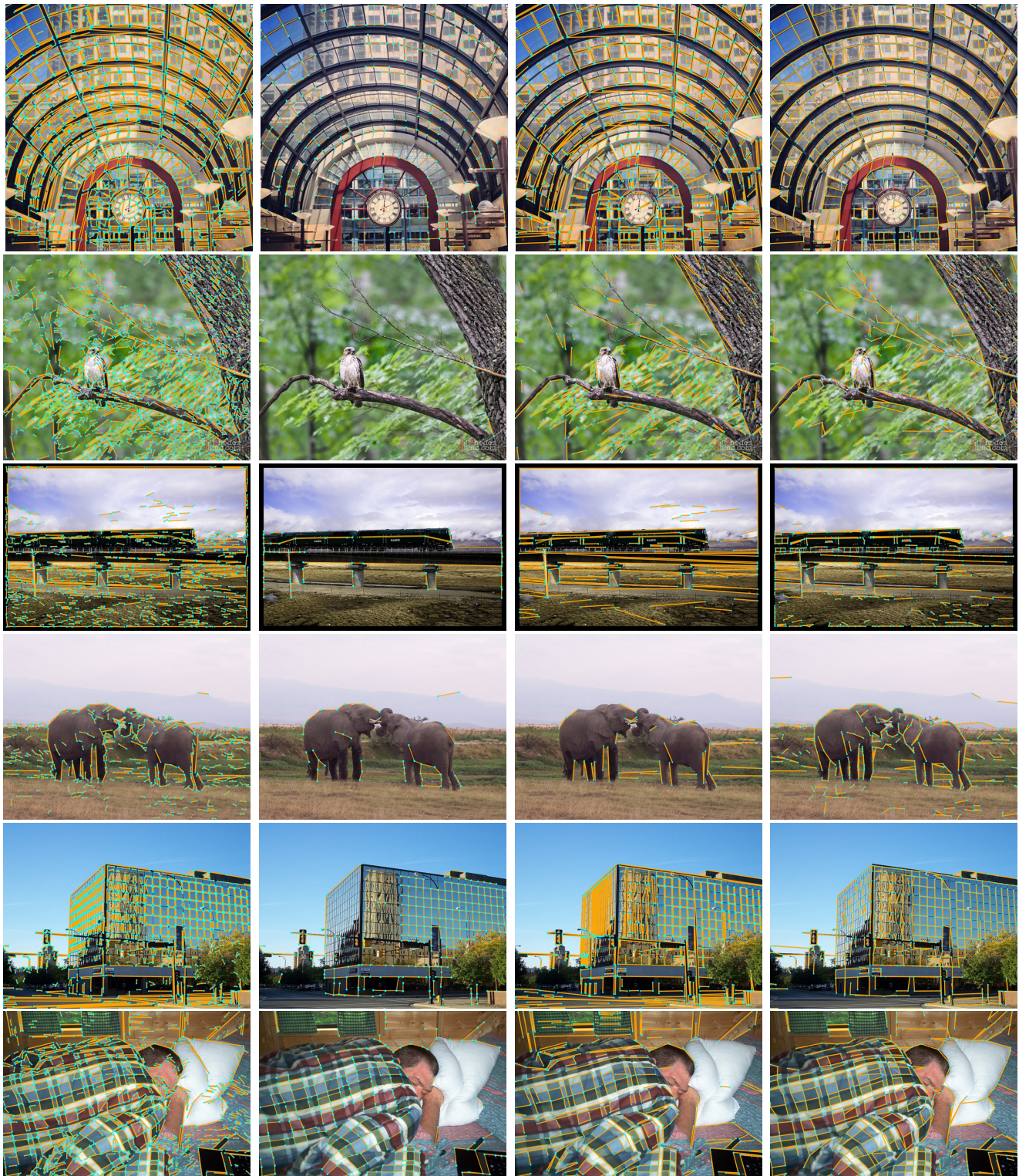
LSD [12]

HAWPv3 [14]

DeepLSD [8]

ScaleLSD (Ours)

Figure A6. Qualitative results of line segments detection.



LSD [12]

HAWPv3 [14]

DeepLSD [8]

ScaleLSD (Ours)

Figure A7. Qualitative results of line segments detection.