Supplementary Materials for
Superpixel Segmentation with Fully Convolutional Networks

Fengting Yang Qian Sun
The Pennsylvania State University
fuy34@psu.edu, uestcqs@gmail.com

Hailin Jin
Adobe Research
hljin@adobe.com

Zihan Zhou
The Pennsylvania State University
zzhou@ist.psu.edu

In Section 1 and Section 2, we provide the detailed architecture designs for the superpixel segmentation network and the stereo matching network, respectively. In Section 3, we report additional qualitative results for superpixel segmentation on BSDS500 and NYUv2, disparity estimation on Sceneflow, HR-VS, and Middlebury-v3, and superpixel segmentation on HR-VS.

1. Superpixel Segmentation Network

Table 1 shows the specific design of our superpixel segmentation network. We use a standard encoder-decoder design with skip connections to predict the superpixel association map Q. Batch normalization and leaky Relu with negative slope 0.1 are used for all the convolution layers, except for the association prediction layer (assoc) where softmax is applied.

Table 1. Specification of our superpixel segmentation network architecture.

Name	Kernel	Str.	Ch I/O	InpRes	OutRes	Input
cnv0a	3×3	1	3/16	208×208	208×208	image
cnv0b	3×3	1	16/16	208×208	208×208	cnv0a
cnv1a	3×3	2	16/32	208×208	104×104	cnv0b
cnv1b	3×3	1	32/32	104×104	104×104	cnv1a
cnv2a	3×3	2	32/64	104×104	52×52	cnv1b
cnv2b	3×3	1	64/64	52×52	52×52	cnv2a
cnv3a	3×3	2	64/128	52×52	26×26	cnv2b
cnv3b	3×3	1	128/128	26×26	26×26	cnv3a
cnv4a	3×3	2	128/256	26×26	13×13	cnv3b
cnv4b	3×3	1	256/256	13×13	13×13	cnv4a
upcnv3	4×4	2	256/128	13×13	26×26	cnv4b
icnv3	3×3	1	256/128	26×26	26×26	upcnv3+cnv3b
upcnv2	4×4	2	128/64	26×26	52×52	icnv3
icnv2	3×3	1	128/64	52×52	52×52	upcnv2+cnv2b
upcnv1	4×4	2	64/32	52×52	104×104	icnv2
icnv1	3×3	1	64/32	104×104	104×104	upcnv1+cnv1b
upcnv0	4×4	2	32/16	104×104	208×208	icnv1
icnv0	3×3	1	32/16	208×208	208×208	upcnv0+cnv0b
assoc	3×3	1	16/9	208×208	208×208	icnv0

2. Stereo Matching Network

Table 2 shows the architecture design of stereo matching network, in which we modify PSMNet [1] to perform superpixel-based downsampling/upsampling operations. We name it superpixel-based PSMNet (SPPSMNet).

Table 2. Specification of our stereo matching network (SP-PSMNet) architecture.

Name	Kernel	Str.	Input	OutDim
Input				
Img_1/2				$H \times W \times 3$
Superpixel segmentation and superpixel-based wnwnsampling				
assoc_1/2	see Table 1		Img_1/2	$H \times W \times 9$
sImg_1/2	assoc_1/2	4	Img_1/2	$\frac{1}{4}H \times \frac{1}{4}W \times 3$
PSMNet feature extractor				
cnv0_1	$\mathbf{3 \times 3, 32}$	**1**	**sImg_1/2**	$\frac{1}{4}\mathbf{H} \times \frac{1}{4}\mathbf{W} \times \mathbf{32}$
cnv0_2	$3 \times 3, 32$	1	cnv0_1	$\frac{1}{4}H \times \frac{1}{4}W \times 32$
cnv0_3	$3 \times 3, 32$	1	cnv0_2	$\frac{1}{4}H \times \frac{1}{4}W \times 32$
cnv1_x	$\begin{bmatrix} 3 \times 3, 32 \\ 3 \times 3, 32 \end{bmatrix} \times 3$	1	cnv0_3	$\frac{1}{4}H \times \frac{1}{4}W \times 32$
conv2_x	$\begin{bmatrix} \mathbf{3 \times 3, 64} \\ \mathbf{3 \times 3, 64} \end{bmatrix} \times \mathbf{16}$	**1**	**cnv1_x**	$\frac{1}{4}\mathbf{H} \times \frac{1}{4}\mathbf{W} \times \mathbf{64}$
cnv3_x	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 3$	1	cnv2_x	$\frac{1}{4}H \times \frac{1}{4}W \times 128$
cnv4_x	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 3, dila = 2$	1	cnv3_x	$\frac{1}{4}H \times \frac{1}{4}W \times 128$
PSMNet SPP module, cost volume, and 3D CNN				
output_1 output_2 output_3	Please refer to [1] for details			$\frac{1}{4}H \times \frac{1}{4}W \times \frac{1}{4}D \times 1$ $\frac{1}{4}H \times \frac{1}{4}W \times \frac{1}{4}D \times 1$ $\frac{1}{4}H \times \frac{1}{4}W \times \frac{1}{4}D \times 1$
Superpixel-based upsampling				
disp_prb1	bilinear upsampling assoc_1	N.A. 4	output_1	$\frac{1}{4}H \times \frac{1}{4}W \times D$ $H \times W \times D$
disp_prb2	bilinear upsampling assoc_1	N.A. 4	output_2	$\frac{1}{4}H \times \frac{1}{4}W \times D$ $H \times W \times D$
disp_prb3	bilinear upsampling assoc_1	N.A. 4	output_3	$\frac{1}{4}H \times \frac{1}{4}W \times D$ $H \times W \times D$
PSMNet disparity regression				
disp_1	disparity regression	N.A.	disp_prb1	$H \times W$
disp_2	disparity regression	N.A.	disp_prb2	$H \times W$
disp_3	disparity regression	N.A.	disp_prb3	$H \times W$

The layers which are different from the orignal PSMNet have been highlighted in bold face. In Table 2, we use input image size 256×512 with maximum disparity $D = 192$, which is the same as the original PSMNet, and we set superpixel grid cell size 4×4 to perform $4\times$ downsampling/upsampling.

For stereo matching tasks with high resolution images (*i.e.*, HR-VS and Middilebury-v3), we use input image size 1024×2048 with maximum disparity $D = 768$, and we set superpixel grid cell size 16×16 to perform $16\times$ downsampling/upsampling. To further reduce the GPU memory usage, in the high-res stereo matching tasks, we reduce the channel number of the layers “cnv4a” and “cnv4b” in the superpixel segmentation network from 256 to 128, remove the batch normalization operation in the superpixel segmentation network, and perform superpixel-based spatial upsampling after the disparity regression.

3. Additional Qualitative Results

3.1. Superpixel Segmentation

Figure 1 and Figure 2 show additional qualitative results for superpixel segmentation on BSDS500 and NYUv2. The three learning-based methods, SEAL, SSN, and ours, can recover more detailed boundaries than SLIC, such as the hub of the windmill in the second row of Figure 1 and the pillow on the right bed in the fourth row of Figure 2. Compared to SEAL and SSN, our method usually generate more compact superpixels.

3.2. Application to Stereo Matching

Figure 3, Figure 4, and Figure 6 show the disparity prediction results on SceneFlow, HR-VS and Middlebury-v3, respectively. Compared to PSMNet, our methods are able to better preserve the fine details, such as the headset wire (the seventh row of Figure 3) , street lamp post (the first row of Figure 4) and the leaves (the fifth row of Figure 6). We also observe that our method can better handle textureless areas, such as the car back in the seventh row of Figure 4. It is probably because our method directly downsample the images 16 times before sending them to the modified PSMNet, while the original PSMNet only downsamples the image 4 times, and uses stride-2 convolution to perform another $4\times$ downsampling later. The input receptive filed (w.r.t. the original image) of our method is actually larger than that of original PSMNet, which enables our method to better leverage context information around the textureless area.

Figure 5 visualizes the superpixel segmentation results of **Ours_fixed** and **Ours_joint** methods on HR-VS dataset. In general, Superpixels generated by **Ours_joint** are more compact and pay more attentions to the disparity boundary. The color boundaries that are not aligned with the disparity boundary, such as the water pit on the road in the second row of Figure 5, are often ignored by **Ours_joint**. This phenomenon reflects the influence of disparity estimation on the superpixels in the joint training.

References

[1] Jia-Ren Chang and Yong-Sheng Chen. Pyramid stereo matching network. In *CVPR*, pages 5410–5418, 2018. 1

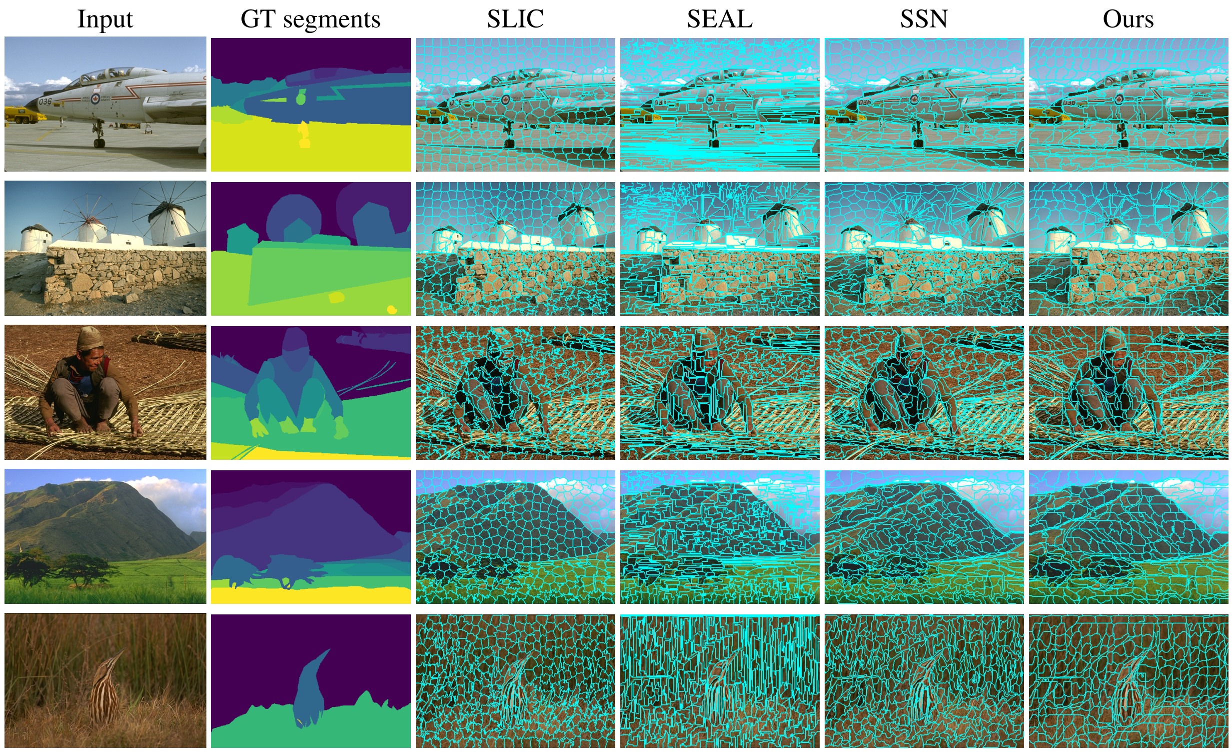

Figure 1. Additional superpixel segmentation results on BSDS500.

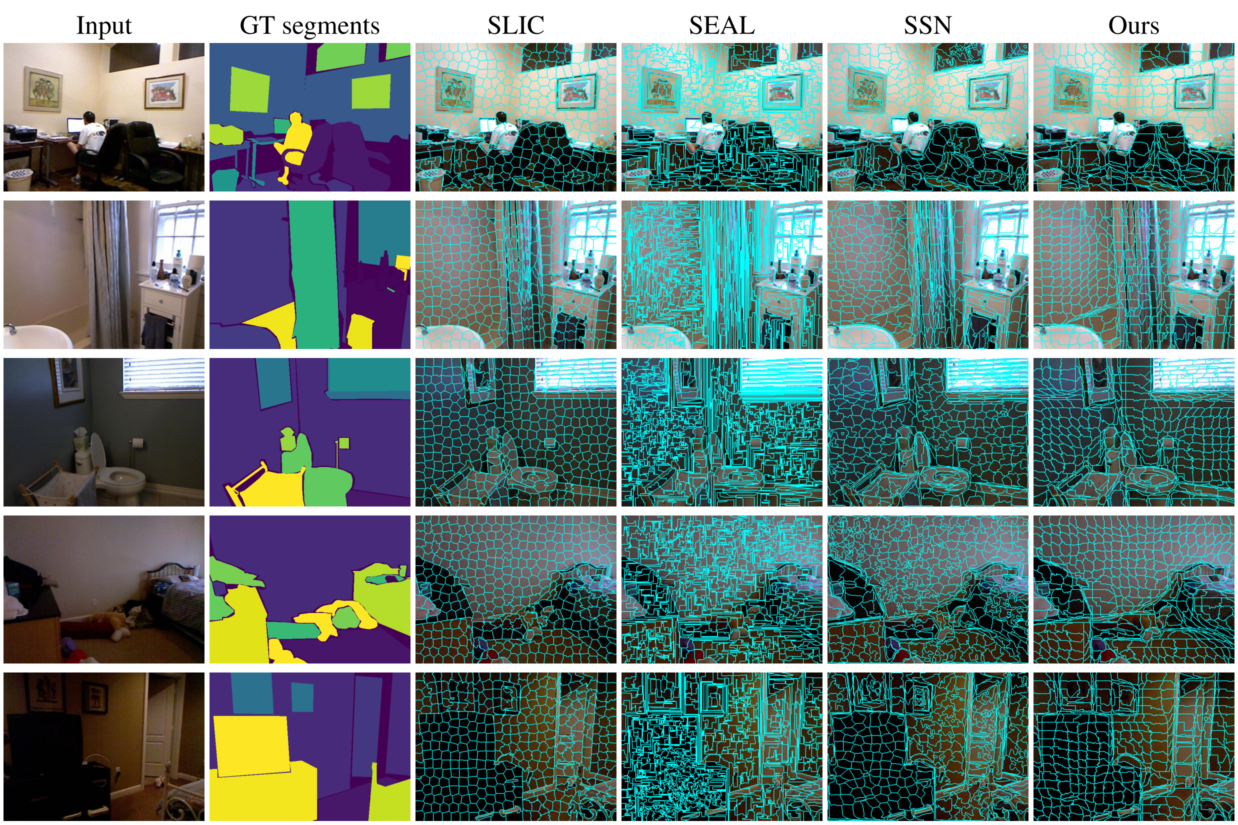

Figure 2. Additional superpixel segmentation results on NYUv2.

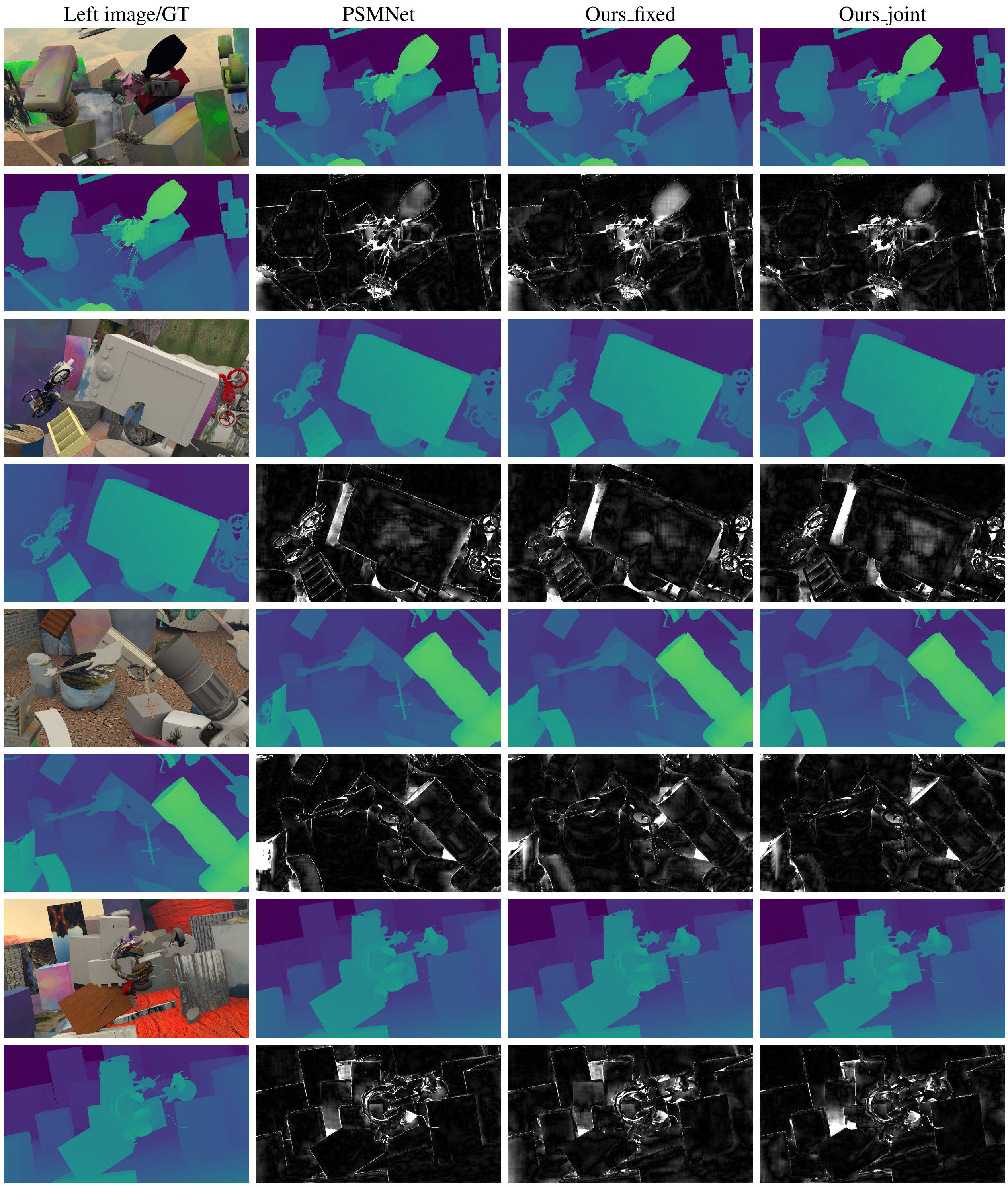

Figure 3. Disparity prediction results on SceneFlow. For each method, we show both the predicted disparity map (top) and the error map (bottom). For the error map, the darker the color, the lower the end point error (EPE).

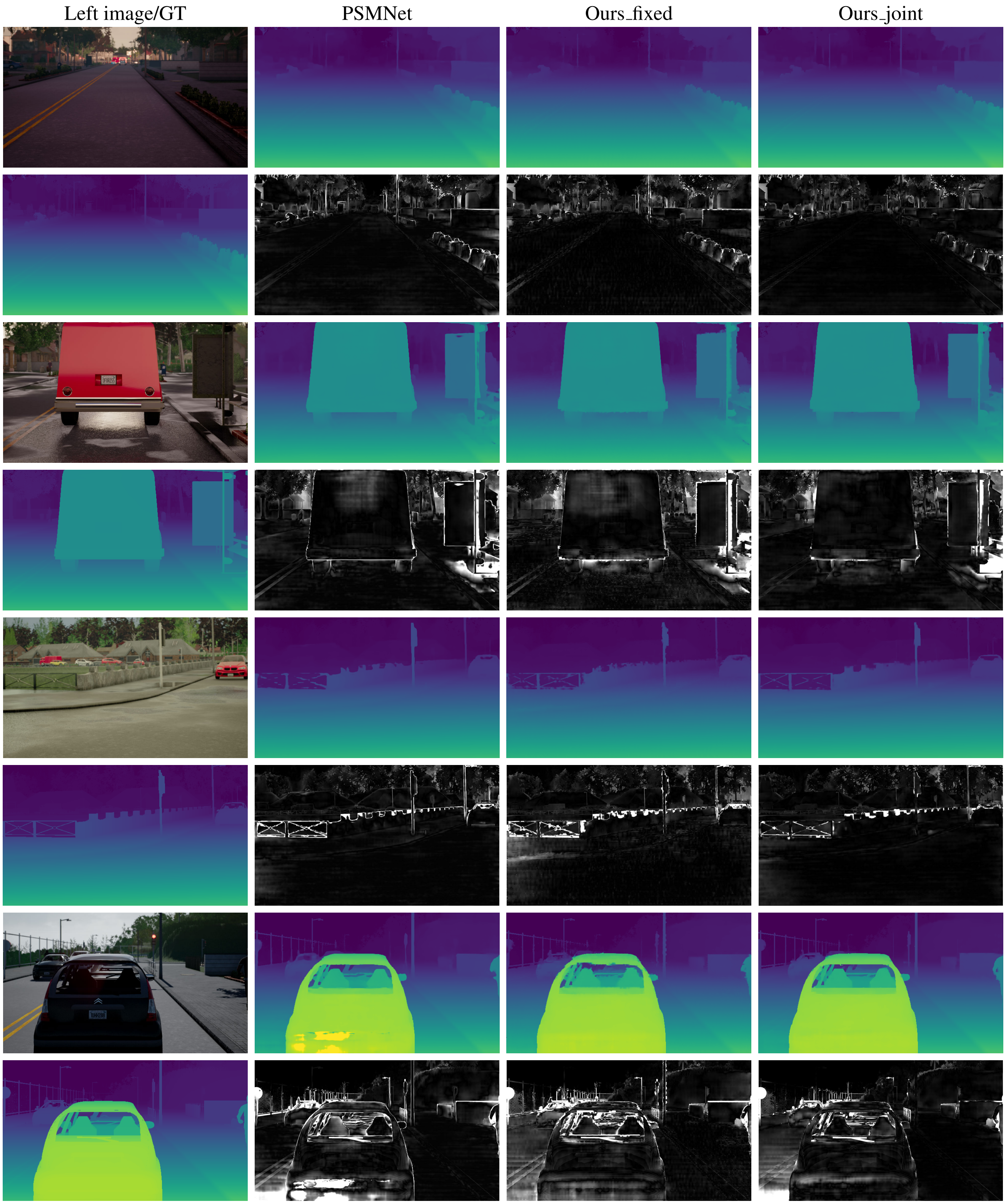

Figure 4. Disparity prediction results on HR-VS. For each method, we show both the predicted disparity map (top) and the error map (bottom). For the error map, the darker the color, the lower the end point error (EPE).

Left image | Ours_fixed | Ours_joint

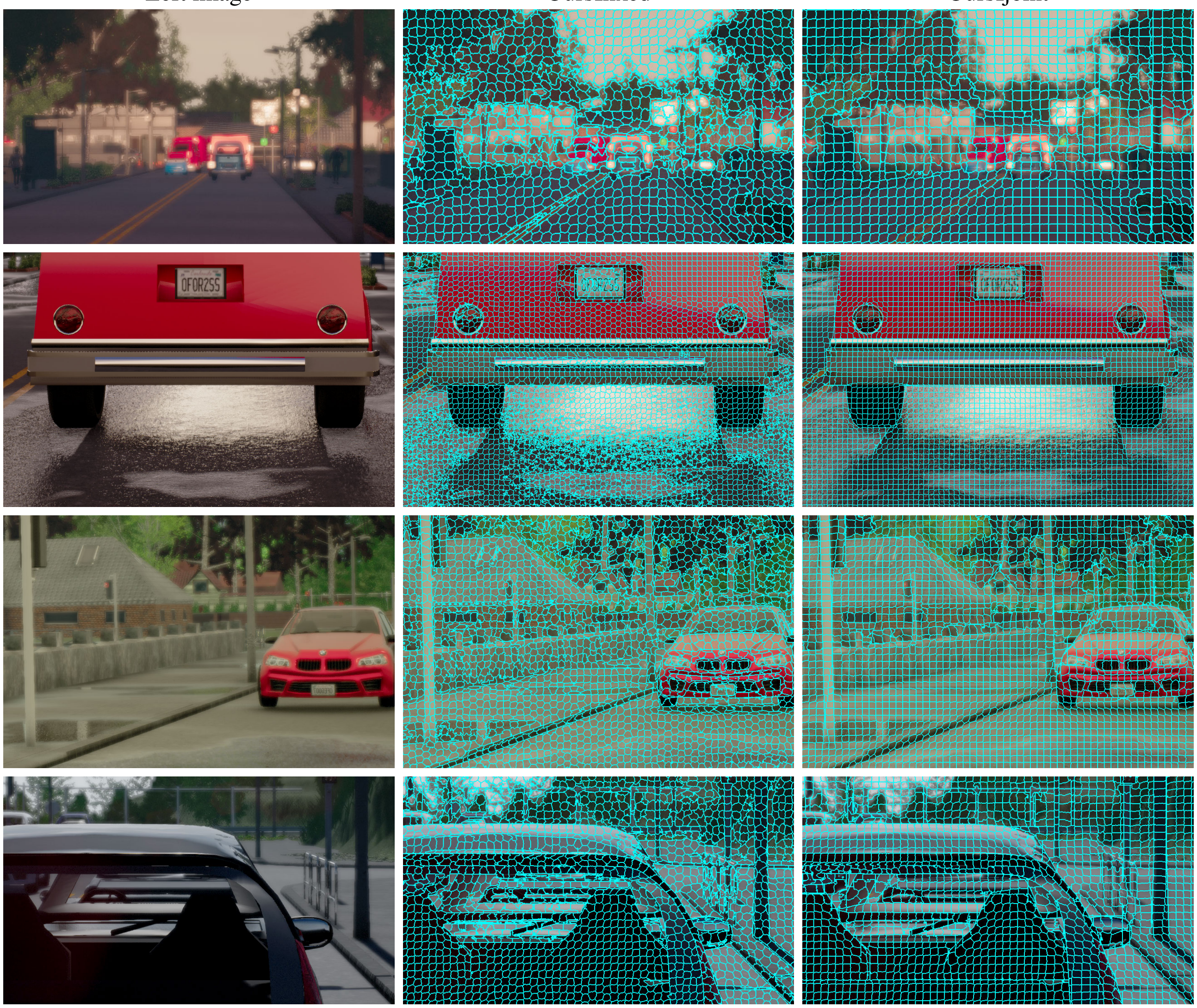

Figure 5. Comparsion of superpixel segmentation results on HR-VS. Note we do not enforce the superpixel connectivity here.

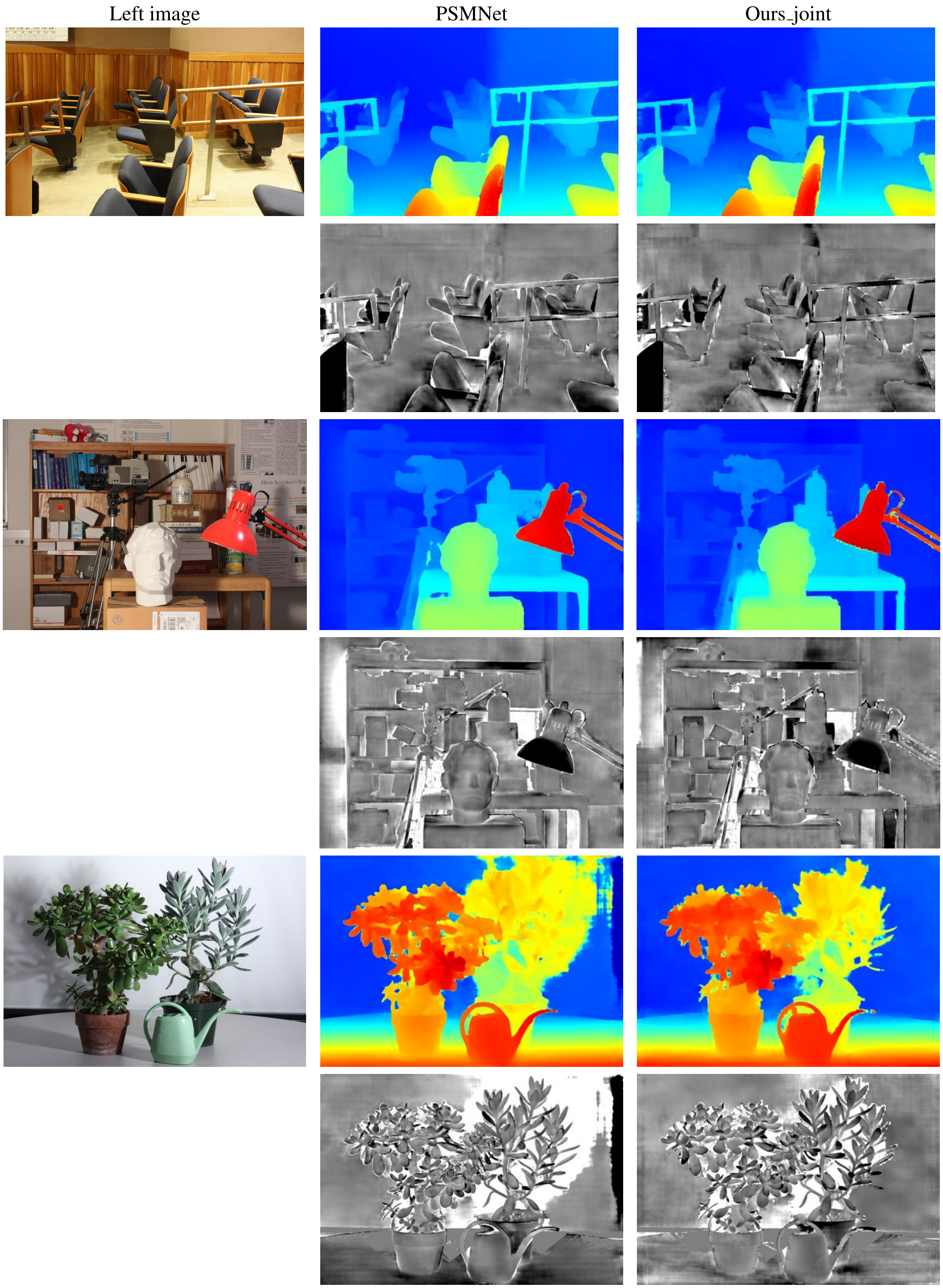

Figure 6. Disparity estimation results on Middlebury-v3. For each method, we show both the predicted disparity map (top) and the error map (bottom). For the error map, the darker the color, the lower the error. All the images are from Middlebury-v3 leaderboard.