DEEP SCALE-AWARE IMAGE SMOOTHING

Jiachun Li[⋆]*, Kunkun Qin*[⋆]*, Ruotao Xu*[⋆] *and Hui Ji*[†]

[⋆]School of Computer Science and Engineering, South China University of Technology, China
[†]Department of Mathematics, National University of Singapore, Singapore

ABSTRACT

Image smoothing, a technique for smoothing out insignificant textures while preserving meaningful structures, is an important component in many vision and graphics applications. Scale-awareness plays a fundamental role in image smoothing, as insignificant textures and noise usually are at fine scales while meaningful boundary objects are at coarse scales. This paper proposes a deep-learning-based scale-aware image smoothing method, which is built on a downscaling-upscaling mechanism with attention. The downscaling mechanism is for predicting large-scale salient structures, and the upscaling mechanism is for identifying and inferring insignificant small-scale details from the salient structures. In the experiments, the proposed one provides a noticeable performance improvement over recent methods.

***Index Terms*—** Image smoothing, Deep learning, Texture removal, Deep filtering, Image processing

1. INTRODUCTION

Image smoothing is an important technique widely used in many vision and graphics applications, including image enhancement, segmentation and abstraction, and many others; see *e.g.* [1–5]. Image smoothing is about removing insignificant detailed textures or noise from the input image and keeping important salient structures such as object boundaries. In the last decade, many image smoothing methods (*e.g.* [6–10]) have been developed with a focus on edge-aware filtering, which aim at preserving high-contrast edges and removing low-contrast details from the input.

There are two classes of edge-aware filtering methods: local filtering and global filtering. Local filtering smooths an image by taking the weighted average of local neighboring pixel intensities [6, 7]. Local filtering methods are simple but often produce gradient reversals and halo artifacts on image edges [8]. Global filtering [9, 10] addressed these issues by optimizing the filtering on the whole image. Global filtering is expensive on both time consuming and memory usage [8].

Corresponding author: Ruotao Xu (xrt@scut.edu.cn).

This work is supported by National Natural Science Foundation of China (62106077) and Postdoctoral Foundation of China (2020M682705).

These edge-aware filtering methods characterize insignificant details and meaningful structures based on their low-level image features, which cannot handle important structures with semantic meanings well. In addition, the performance of these methods is very sensitive to the involved hyper-parameters, the tuning-up of which for individual images is quite expensive. In recent years, deep filtering [1, 3, 11–14] has emerged as a promising approach for image smoothing, which trains a deep neural network (DNN) for the task. While DNNs are powerful for capturing semantic information of images, the existing ones [1, 11–13] rely on an edge prediction module, which is either implemented by an edge-map-supervised loss [1, 11, 12], or a recursive filtering scheme [13]. In other words, these deep filtering methods only provide incremental improvements over non-learning edge-preserving filtering, and the main weakness remains.

1.1. Scale-Awareness for Image Smoothing

Edge-aware filtering utilizes the discrepancy between local pixels intensities, in order to distinguish insignificant detailed textures and salient structures. While the discrepancy of local pixel intensities is highly correlated to salient structures, it is insufficient for characterizing detailed textures and salient structures. Scale is another important cue for distinguishing these two as well, and it is often more reliable. Scale cues are based on the observation that detailed textures will vanish at a coarse scale while salient structures remain. Hence, one scale-aware diagram is to first find the coarse-scale structures and then reconstruct the high-resolution smooth image. However, since detailed textures contained in images have varying scales [15], it is a challenging problem on how to automate the determination of the proper scale for exploiting such a cue.

Scale-awareness has been exploited in a few studies. Zhang [16] proposed a scale-aware filtering algorithm based on iterative guided filtering [6], which defines salient structures at the coarse scale as the image smoothed by Gaussian filter. Zhang *et al.*'s method does not address the automation of the determination of scale, *i.e.*, the variance of Gaussian filter. Shen *et al.* [14] proposed a U-net based multi-scale deep filtering method. However, since there is not any parameter-sharing mechanism between the downscaling and upscaling modules, Shen *et al.*'s model usually introduces a great num-

ber of learnable parameters, which may lead to slow convergence and overfitting problems. Motivated by the weakness of existing scale-aware approaches for image smoothing, this paper aims at developing a lightweight deep learning solution that provides better performance than existing solutions yet can run on an environment with limited computing resources.

1.2. Main Idea and Contributions

This paper proposed a DNN architecture for image smoothing, named as *Scale-Aware Smoothing Network* (SASNet), which is built on an attentive DNN with parameter-sharing downscaling-upscaling mechanisms. In SASNet, an invertible downscaling module is trained to predict the salient structures at the coarse scale. The main motivation of adopting a module is for the upscaling module being able to quickly infer detailed textures at the fine scale from salient structures at the coarse scale, without introducing additional learnable parameters. For better preserving salient edges and enhancing the discriminability between structures and textures in feature map representations, the channel-spatial attention mechanism is also introduced in the proposed SASNet.

Our method utilizes the disentangling ability of coupling layers [17,18] for image smoothing. The proposed SASNet is designed to disentangle the detailed texture and salient structures, represented by two latent variables, which have their strong orientations in the wavelet transform domain. Then, in the forward pass, two variables are generated. In the reverse pass, the variable representing detailed textures is discarded in the latent space to predict an image with only salient structures. In summary, our major technical contributions are listed as follows.

1. A DNN with parameter-sharing downscaling-upscaling mechanism implemented by attentive invertible modules is proposed for image smoothing, which not only leads to computational efficiency but also introduces the disentangling ability induced from the architecture.
2. Built upon the proposed architecture, a scale-aware image smoothing method is developed for effectively removing detailed textures at coarse scales and keeping salient structures at fine scales. The experiments showed its performance gain over existing solutions.

2. PROPOSED METHOD

2.1. Downscaling-Upscaling Framework

In terms of the scale-awareness for smoothing, meaningful structures refer to the discontinuities with large magnitude at coarse scales. Thus, we first learn a function with the downsampling operation to extract the structures at coarse scales. That is, given an image $\boldsymbol{x} \in \mathbb{R}^{H\times W\times C}$, SASNet learns an invertible structure-extraction module, denoted by $f(\cdot;\Theta)$ with learnable weights Θ, expressed by

$$f(\cdot;\Theta) : \boldsymbol{x} \to [\boldsymbol{y}_{\text{cor}}, \boldsymbol{z}]. \tag{1}$$

where $\boldsymbol{y}_{\text{cor}} \in \mathbb{R}^{\frac{H}{2}\times\frac{W}{2}\times C}$ denotes the predicted salient structures at the coarse scale, and the variable $\boldsymbol{z} \in \mathbb{R}^{\frac{H}{2}\times\frac{W}{2}\times 3C}$ denotes the feature which encodes the information of texture details and noises. Both the input and the output of f are of the same size due to its invertibility. It can also be seen that the invertibility is for disentangling salient structures at the coarse scale and detailed textures or noise at the fine scale.

There are two modules in the f: one for downscaling and the other for upscaling. The downscaling module, denoted by $f_{\text{cor}}(\cdot;\Theta) : \boldsymbol{x} \to \boldsymbol{y}_{\text{cor}}$, is for predicting the salient structures at the coarse scale. The upscaling module, denoted by $f_{\text{fine}}(\cdot;\Theta) : \boldsymbol{y}_{\text{cor}} \to \boldsymbol{y}$, is for predicting the smoothed image at the fine scale directly from the salient structure at the coarse scale. It is noted that such an upscaling module usually requires training an independent sub-network. However, if f is invertible, one can directly predict $\boldsymbol{y}$ by the inversion of f, denoted by f^{-1}, without introducing an additional sub-network. Compared with the often-seen U-net, the parameter-sharing downscaling-upscaling architecture can reduce the number of parameters by half, leading to faster convergence and better generalization ability. Notice that f^{-1} requires the input of $\boldsymbol{z}$. Instead of using $\boldsymbol{z}$ derived from $\boldsymbol{x}$ which will replicate $\boldsymbol{x}$, we replace it by Gaussian white noise $\boldsymbol{n} \sim N(0, \boldsymbol{I})$ as follows.

$$\boldsymbol{y} = f^{-1}([f_{\text{cor}}(\boldsymbol{x};\Theta), \boldsymbol{n}]). \tag{2}$$

In other words, we have $f_{\text{fine}}(\boldsymbol{y}_{\text{cor}}) = f^{-1}(\boldsymbol{y}_{\text{cor}}, \boldsymbol{n})$. Thus, given the training sample $(\boldsymbol{x}, \hat{\boldsymbol{y}})$, *i.e.* the pair of the image and its ground-truth smoothed version, the learnable function f is supervised by the following loss:

$$\mathcal{L}(\Theta) = \|\boldsymbol{y} - \hat{\boldsymbol{y}}\|_1 = \|f^{-1}([f_{\text{cor}}(\boldsymbol{x};\Theta), \boldsymbol{n}]) - \hat{\boldsymbol{y}}\|_1. \tag{3}$$

Then, the smoothed image $\boldsymbol{y}$ for an input $\boldsymbol{x}$ is obtained by (2).

2.2. Network Architecture

See Fig.1 for the diagram of the proposed SASNet for image smoothing. In the downscaling module, the input image $\boldsymbol{x}$ is firstly decomposed into one low-pass component and three high-pass components by a Haar transform (HT)

$$\mathcal{W} : \boldsymbol{x} \in \mathbb{R}^{H\times W\times C} \to [\boldsymbol{a}_L, \boldsymbol{a}_H] \in \mathbb{R}^{H\times W\times C},$$

where $\boldsymbol{a}_L \in \mathbb{R}^{H\times W\times \frac{C}{4}}$ denotes the low-pass coefficient array, and $\boldsymbol{a}_H \in \mathbb{R}^{H\times W\times \frac{3C}{4}}$ denotes the high-pass coefficient arrays. The array $\boldsymbol{a}_L$ can be viewed as a down-sampled version of $\boldsymbol{x}$ after smoothing by the low-pass filter of HT. Note that HT is invertible with inverse Harr Transform (IHT) $\mathcal{W}^{-1}$, which can be used in the backward pass.

Next, the array $\boldsymbol{a}_L$ is a smoothed version of $\boldsymbol{x}$ at a coarse scale, edges in $\boldsymbol{a}_L$ will be smoothed out too. To keep the

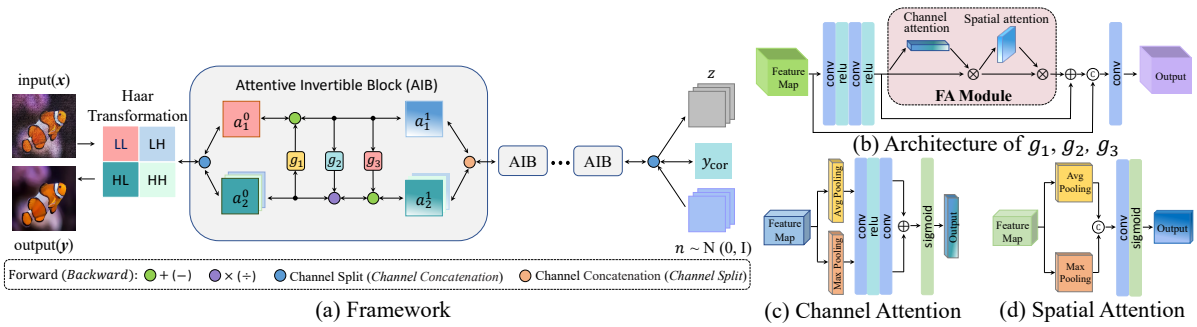

Fig. 1. Architecture of proposed SASNet.

edges sharp at the coarse scale, we introduce additional learnable invertible blocks with additional input of high-frequency information. Following [19], we adopt the invertible coupling layers: Let $\boldsymbol{a}_1^0 = \boldsymbol{a}_L$ and $\boldsymbol{a}_2^0 = \boldsymbol{a}_H$. For $l = 1, 2, \ldots,$

$$\boldsymbol{a}_1^l = \boldsymbol{a}_1^{l-1} + g_1(\boldsymbol{a}_2^{l-1}; \boldsymbol{\Theta}),$$
$$\boldsymbol{a}_2^l = \boldsymbol{a}_2^{l-1} \odot \exp(g_2(\boldsymbol{a}_1^l; \boldsymbol{\Theta})) + g_3(\boldsymbol{a}_1^l; \boldsymbol{\Theta}),$$

where $\boldsymbol{a}^{l-1}$, $\boldsymbol{a}^l$ denote the input and output of the l-th invertible block in the forward pass respectively, g_1,g_2,g_3 are some learnable functions implemented with neural networks, and $\odot$ denotes element-wise product. The procedure above admits the existence of its inversion:

$$\boldsymbol{a}_2^{l-1} = (\boldsymbol{a}_2^l - g_3(\boldsymbol{a}_1^l; \boldsymbol{\Theta})) \odot \exp(-g_2(\boldsymbol{a}_1^l; \boldsymbol{\Theta})),$$
$$\boldsymbol{a}_1^{l-1} = \boldsymbol{a}_1^l - g_1(\boldsymbol{a}_2^{l-1}; \boldsymbol{\Theta}),$$

which is used in the backward pass of the coupling layers.

The coupling layers are implemented as invertible blocks for their simple but unconstrained form, and each of them takes two inputs and two outputs, for fitting our scheme. Note that the design of learnable functions g_1,g_2,g_3 also has a noticeable impact to the performance. In our DNN, the residual block shown in Fig. 1(b) is used for these learnable functions. In addition, the spatial and channel attention mechanism [20] is introduced for better preserving structures, which aggregates the same feature from all the positions with different aggregation strategies, transformations, and strengthening functions. As a result, it benefits the sharpness of the edges in the result. The proposed invertible block, termed as Attentive Invertible Block (AIB) is illustrated in Fig. 1. We stack L AIBs to predict the coarse-scale structures, *i.e.* $\boldsymbol{y}_{\text{cor}} = \boldsymbol{a}_1^L$ and $\boldsymbol{z} = \boldsymbol{a}_2^L$. We set L to 16 in implementation.

For better performance, we further adopt a data augmentation based self-ensemble strategy in testing, which flips and rotates the input image at different angles and averages their outputs as the final smoothed results.

3. EXPERIMENTS

The SPS dataset [12] is employed for training. Our model is trained using Adam with momentum of $\beta_1 = 0.9$, $\beta_2 = 0.999$ and batch size of 16. The initial learning rate is set to $2e{-}4$ and decayed by half every 10k iterations. We augment the data with flipping, and random rotations of 90°, 180°, 270°. Image patches are cropped into 128×128 for training.

In evaluation, our SASNet is compared with several state-of-the-art image smoothing methods, including two local filtering methods: GF [6] and RGF [16], six global filtering methods: L0 [21], RTV [22], SDF [4], PNLS [23], ISO [24] and ILS [25], and three deep filtering methods: DEAF [11], DRF [13], ResNet [8] and JESSNet [12]. Note that the ResNet is the baseline used in [8]. For DEAF and DRF, the models are retrained on the same training dataset as ours. For quantitative comparison, we test our trained SASNet model and the compared methods on two standard datasets SPS [12] and NKS [23]. Both datasets contain paired input images and ground-truth smoothed versions for quantitative comparison. In addition, we also test the methods on natural images without ground truths, where the evaluation is done by subjective comparison according to the human eye's judgment. For fair comparison, the parameters of each local or global filtering method have been tuned to the universal ones on the datasets.

3.1. Performance Evaluation

See Table 1 for quantitative results. It can be seen that SASNet outperformed others by a large margin on both the SPS and NKS datasets. It is noted that the characteristics of two datasets: SPS and NKS, are quite different. SPS uses texture-less natural images as ground-truths, while NKS uses hand-drawn-like cartoon images. Nevertheless, by only calling training samples in SPS, the proposed method still achieves remarkable performance on NKS, which indicates the good generalization ability of SASNet. In Table 2, we compare the number of FLOPs and the GPU memory cost of different deep models. It can be seen that SASNet requires much less computational resource compared to the two recent methods: ResNet and JESSNet, while more than DRF which uses an extremely lightweight design with recursive filters but performed much worse than SASNet in Table 1. See also Fig. 2 for a visual inspection on several synthetic and real images. Clearly, SASNet can remove the fine-scale textures on

Fig. 2. Visual comparison on synthesized (1st row) and natural (2nd∼4th rows) images.

Method	Metric	Original	GF	RGF	L0	RTV	SDF	ILS	PNLS	ISO	DEAF	DRF	ResNet	JESSNet	SASNet
SPS	PSNR	20.28	25.33	25.86	28.48	26.89	27.06	25.46	25.43	27.10	27.36	27.01	29.84	31.73	**33.18**
	SSIM	0.26	0.65	0.64	0.78	0.82	0.80	0.62	0.66	0.81	0.82	0.79	0.88	0.92	**0.94**
NKS	PSNR	28.04	28.15	32.56	28.32	30.69	33.17	31.5	33.2	33.25	30.45	30.02	33.24	34.24	**34.75**
	SSIM	0.54	0.83	0.91	0.90	0.90	0.89	0.81	0.92	0.95	0.90	0.88	0.92	0.94	**0.95**

Table 1. PSNR(dB) and SSIM comparison on two datasets.

each image while well preserving the coarse-scale structures, leading to the best visual quality among all the competitors.

Metric	DRF	ResNet	JESSNet	SASNet
#FLOPs(Giga)	14.42	719.04	672.04	209.12
Memory(MB)	214	3638	1053	748

Table 2. FOLPs, GPU memory cost on a 512×384 image.

Network	Scale	Attention	SPS		NKS	
			PSNR	SSIM	PSNR	SSIM
AutoEncoder	-	-	22.02	0.81	24.70	0.84
U-net	-	-	32.15	0.90	33.41	0.94
SASNet	$\times 2$	-	32.59	0.92	34.20	0.95
SASNet	$\times 2$	$\surd$	**33.18**	**0.94**	**34.75**	**0.95**
SASNet	$\times 4$	$\surd$	32.20	0.91	33.88	0.94
SASNet	$\times 8$	$\surd$	31.43	0.92	32.50	0.93

Table 3. PSNR(dB) and SSIM values in ablation study.

3.2. Ablation Study

To investigate the improvement of the proposed downscaling-upscaling architecture, we compared the proposed model with two baselines: a naïve-autoencoder-based network and a UNet-based network. The results are shown in Table 3, where the proposed method achieves much higher results compared with the baselines. We further investigate how the performance of the SASNet will be impacted by using different levels of downscaling. See also Table 3, where the SASNet achieved the highest PSNR and SSIM values when a single downscaling operation is used, which may be caused by the preference of tiny-pattern textures in the synthesizing procedures of the two standard datasets. In addition, we also study the performance impact of the attention mechanisms. The results are also listed in Table 3, which show that the attention mechanisms can bring noticeable performance gain.

4. CONCLUSION

Built on a parameter-sharing downscaling-upscaling network with attention, this paper proposed a scale-aware method for image smoothing. Owing to the great disentangling ability and computationally efficient implementation induced by the coupling layers and invertible modules, our method can have an effective scale-ware scheme that preserves meaningful structures at a coarse scale very well. Extensive experiments on two datasets demonstrate the advantages of our method over existing ones both quantitatively and visually.

5. REFERENCES

[1] Qingnan Fan, Jiaolong Yang, Gang Hua, Baoquan Chen, and David Wipf, "A generic deep architecture for single image reflection removal and image smoothing," in *Proc. ICCV*, 2017, pp. 3238–3247.

[2] Zihan Zhou, Jing Li, Yong Xu, and Yuhui Quan, "Full-reference image quality metric for blurry images and compressed images using hybrid dictionary learning," *Neural Computing and Applications*, vol. 32, no. 16, pp. 12403–12415, 2020.

[3] Qifeng Chen, Jia Xu, and Vladlen Koltun, "Fast image processing with fully-convolutional networks," in *Proc. ICCV*, 2017, pp. 2497–2506.

[4] Bumsub Ham, Minsu Cho, and Jean Ponce, "Robust guided image filtering using nonconvex potentials," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 1, pp. 192–207, 2017.

[5] Yuhui Quan, Huan Teng, Tao Liu, and Yan Huang, "Weakly-supervised sparse coding with geometric prior for interactive texture segmentation," *IEEE Signal Proc. Lett.*, vol. 27, pp. 116–120, 2020.

[6] Kaiming He, Jian Sun, and Xiaoou Tang, "Guided image filtering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 6, pp. 1397–1409, 2012.

[7] Zhuo Su, Xiaonan Luo, Zhengjie Deng, Yun Liang, and Zhen Ji, "Edge-preserving texture suppression filter based on joint filtering schemes," *IEEE Trans. Multimedia*, vol. 15, no. 3, pp. 535–548, 2012.

[8] Feida Zhu, Zhetong Liang, Xixi Jia, Lei Zhang, and Yizhou Yu, "A benchmark for edge-preserving image smoothing," *IEEE Trans. Image Process.*, vol. 28, no. 7, pp. 3556–3570, 2019.

[9] Feihu Zhang, Longquan Dai, Shiming Xiang, and Xiaopeng Zhang, "Segment graph based image filtering: fast structure-preserving smoothing," in *Proc. ICCV*, 2015, pp. 361–369.

[10] Zhiqiang Zhou, Bo Wang, and Jinlei Ma, "Scale-aware edge-preserving image filtering via iterative global optimization," *IEEE Trans. Multimedia*, vol. 20, no. 6, pp. 1392–1405, 2017.

[11] Li Xu, Jimmy Ren, Qiong Yan, Renjie Liao, and Jiaya Jia, "Deep edge-aware filters," in *Proc. ICML*. PMLR, 2015, pp. 1669–1678.

[12] Yidan Feng, Sen Deng, Xuefeng Yan, Xin Yang, Mingqiang Wei, and Ligang Liu, "Easy2hard: Learning to solve the intractables from a synthetic dataset for structure-preserving image smoothing," *IEEE Trans. Neural Networks Learn. Syst.*, 2021.

[13] Sifei Liu, Jinshan Pan, and Ming-Hsuan Yang, "Learning recursive filters for low-level vision via a hybrid neural network," in *Proc. ECCV*. Springer, 2016, pp. 560–576.

[14] Xiaoyong Shen, Ying-Cong Chen, Xin Tao, and Jiaya Jia, "Convolutional neural pyramid for image processing," *arXiv preprint arXiv:1704.02071*, 2017.

[15] Yuhui Quan, Yong Xu, Yuping Sun, and Yu Luo, "Lacunarity analysis on image patterns for texture classification," in *Proc. CVPR*, 2014, pp. 160–167.

[16] Qi Zhang, Xiaoyong Shen, Li Xu, and Jiaya Jia, "Rolling guidance filter," in *Proc. ECCV*. Springer, 2014, pp. 815–830.

[17] Mingqing Xiao, Shuxin Zheng, Chang Liu, Yaolong Wang, Di He, Guolin Ke, Jiang Bian, Zhouchen Lin, and Tie-Yan Liu, "Invertible image rescaling," in *Proc. ECCV*. Springer, 2020, pp. 126–144.

[18] Yang Liu, Zhenyue Qin, Saeed Anwar, Pan Ji, Dongwoo Kim, Sabrina Caldwell, and Tom Gedeon, "Invertible Denoising Network: A Light Solution for Real Noise Removal," in *Proc. CVPR*, 2021, pp. 13365–13374.

[19] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio, "Density estimation using real nvp," *arXiv preprint arXiv:1605.08803*, 2016.

[20] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon, "Cbam: Convolutional block attention module," in *Proc. ECCV*, 2018, pp. 3–19.

[21] Li Xu, Cewu Lu, Yi Xu, and Jiaya Jia, "Image smoothing via l0 gradient minimization," in *SIGGRAPH Asia Conference*, 2011, pp. 1–12.

[22] Li Xu, Qiong Yan, Yang Xia, and Jiaya Jia, "Structure extraction from texture via relative total variation," *ACM Trans. Graphics*, vol. 31, no. 6, pp. 1–10, 2012.

[23] Jun Xu, Zhi-Ang Liu, Yingkun Hou, Xiantong Zhen, Ling Shao, and Ming-Ming Cheng, "Pixel-level non-local image smoothing with objective evaluation," *IEEE Trans. Multimedia*, 2020.

[24] Ruotao Xu, Yong Xu, and Yuhui Quan, "Structure-texture image decomposition using discriminative patch recurrence," *IEEE Trans. Image Process.*, vol. 30, pp. 1542–1555, 2021.

[25] Wei Liu, Pingping Zhang, Xiaolin Huang, Jie Yang, Chunhua Shen, and Ian Reid, "Real-time image smoothing via iterative least squares," *ACM Trans. Graphics*, vol. 39, no. 3, pp. 1–24, 2020.