Supplementary Material: An Inverse Procedural Modeling Pipeline for Stylized Brush Stroke Rendering

Hao Li*, Zhongyue Guan*, Zeyu Wang

The Hong Kong University of Science and Technology (Guangzhou), China

1. Brush Parameters

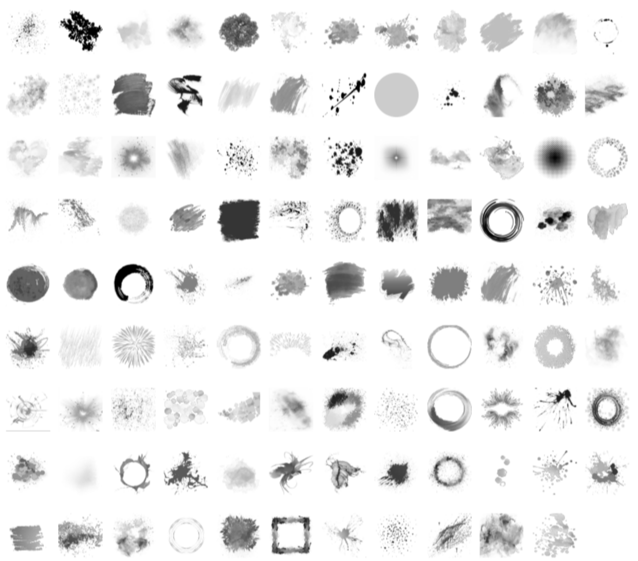

Figure 1: *All stamp images used for brush generation.*

We generated 6,677 sets of brush parameters φ = (stamp image, interval, thickness, rotation randomness, noise factor). For the stamp image, we collected 107 public ones from an online platform [PNG23] as Figure 1 shows.

2. Stroke-based Patch Segmentation

Initially, we used the Canny algorithm [Can86] for recognizing edges in sketch drawings, and then applied dilation and Gaussian blur to smooth the brush edges. However, it is less effective for denser sketches. As the turtle example in Figure 2 shows, the aforementioned edge-based approach recognized the main body as a whole patch, while our approach is capable of identifying individual patch for a single brush.

*Equal contribution.

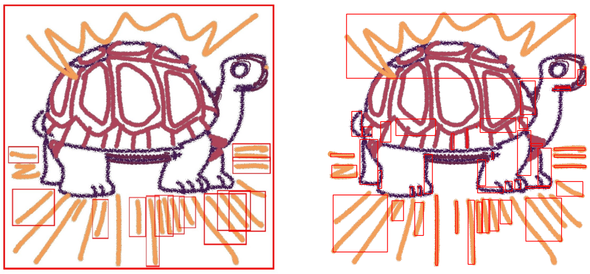

Figure 2: *Comparison of segmentation results. The edge-based approach cannot identify individual brush in strokes that connect together (left), while our approach can separate individual brush as a single patch (right).*

3. Brush Prediction

3.1. Model

We modified ResNet-18 [HZRS15] architecture by changing the fully connected layer to have 111 outputs. Table 1 shows the detailed network architecture.

3.2. Loss

We formulated the prediction of stamp images as a classification problem and the prediction of other parameters as linear regression problems. Firstly, we evaluate the predicted stamp image category $\hat{s}$, as compared to the target category s, using cross-entropy loss:

$$L_s = -\sum_{j=1}^{n} s_j \log \hat{s}_j$$

where n is 107, the number of stamp images we included. We use L_2 loss for interval, comparing a predicted interval value $\hat{i}$ from the corresponding target i:

$$L_i = ||i - \hat{i}||_2$$

The L_2 losses for thickness, rotation randomness, and noise factor are similarly calculated.

Operator	H,W	IC,OC	K,S,P
Conv2d	224, 224	3,64	7, 2, 3
MaxPool2d	112, 112	64,64	3, 2, 1
Conv2d	56, 56	64,64	3, 1, 1
Conv2d	56, 56	64,64	3, 1, 1
Conv2d	56, 56	64,64	3, 1, 1
Conv2d	56, 56	64,64	3, 1, 1
Conv2d	56, 56	64,128	3, 2, 1
Conv2d	28, 28	128,128	3, 1, 1
Conv2d	28, 28	128,128	3, 1, 1
Conv2d	28, 28	128,128	3, 1, 1
Conv2d	28, 28	128,256	3, 2, 1
Conv2d	14, 14	256,256	3, 1, 1
Conv2d	14, 14	256,256	3, 1, 1
Conv2d	14, 14	256,256	3, 1, 1
Conv2d	14, 14	256,512	1, 2, 1
Conv2d	7, 7	512,512	3, 1, 1
Conv2d	7, 7	512,512	3, 1, 1
Conv2d	7, 7	512,512	3, 1, 1
Conv2d	7, 7	512,512	3, 1, 1
AdaptiveAvgPool2d	1, 1	512, 512	
Linear	111		

Table 1: *Network Architecture. H/W denotes height and width, IC input channels, OC output channels, K kernel size, S stride size, and P padding size.*

References

[Can86] CANNY J.: A Computational Approach to Edge Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6 (1986), 679–698. 1

[HZRS15] HE K., ZHANG X., REN S., SUN J.: Deep Residual Learning for Image Recognition. *arXiv preprint arXiv:1512.03385* (2015). 1

[PNG23] PNGEGG: Free Transparent PNG Images. https://www.pngegg.com/, 2023. Accessed on: December 30, 2023. 1