Stylized Neural Painting (Supplementary Material)

Zhengxia Zou[1], Tianyang Shi[2], Shuang Qiu[1], Yi Yuan[2], Zhenwei Shi[3]
[1]University of Michigan, Ann Arbor, [2]NetEase Fuxi AI Lab, [3]Beihang University

1 High resolution results

Since our painting results are generated with a vector format. We can render them at any resolutions. In Fig. 1 and Fig. 2, we show two groups of results rendered at a 1024x1024 pixel resolution.

Figure 1: A high resolution image-to-painting translation result produced by our method. The result is rendered at a resolution of 1024x1024 pixels.

Figure 2: A high resolution neural style transfer result of our method. The result is rendered at a resolution of 1024x1024 pixels.

2 Controlled experiments

2.1 Progressive rendering and transportation loss

In our method, both the progressive rendering and the optimal transportation (OT) loss are important technical components and can help generate more detailed results. In Fig. 3, we show a group of ablation results of w/ or w/o using progressive rendering and OT loss. We can easily see their differences (marked by yellow boxes).

2.2 'Coarse-to-fine rendering

We propose two versions of rendering mode - "coarse-to-fine rendering" and "directly rendering from nxn grids". The former one starts from a lower resolution and then gradually added strokes at a finer scale while the latter one renders directly from a fine level. Their final output will not be very different, but their intermediate results will have different behaviors. Fig. 4 shows the rendering process of the two approaches at stroke number = 5, 20, and 400.

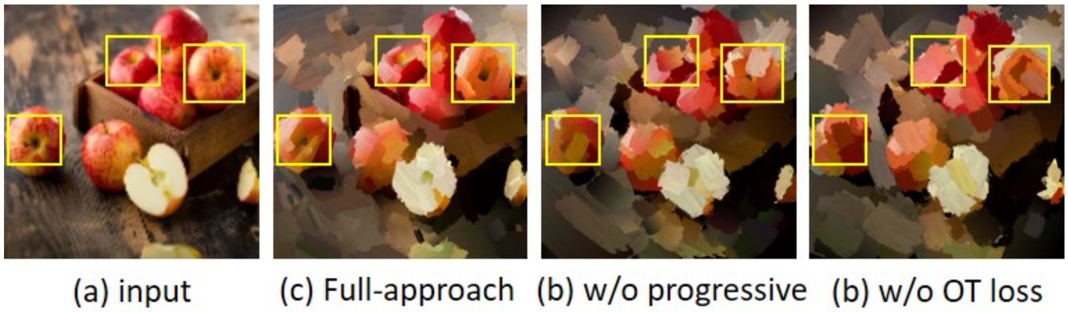

Figure 3: Influence on progressive rendering and OT loss.

Figure 4: Two approaches of coarse-to-fine rendering.

3 Detailed configurations of our neural renderer

In Table 1 and In Table 2, we show a detailed configuration of our rasterization network and our shading network. Specifically, in a $c \times w \times w/s$ layer, c denotes the number of filters, $w \times w$ denotes the filter's size and s denotes the stride size. The output size is formatted as height $\times$ width $\times$ channel.

	Layer	Config	Out size
C1	Deconv + BN + ReLU	512×4×4 / 1	4×4×512
C2	Deconv + BN + ReLU	512×4×4 / 2	8×8×512
C3	Deconv + BN + ReLU	256×4×4 / 2	16×16×256
C4	Deconv + BN + ReLU	128×4×4 / 2	32×32×128
C5	Deconv + BN + ReLU	64×4×4 / 2	64×64×64
C6	Deconv + BN + ReLU	3×4×4 / 2	128×128×3

Table 1: Details of our shading network.

4 Stroke parameterization

We design four types of painting brushes: oil-painting brush, watercolor ink, marker pen, and color tapes. In Table 3, we give a detailed description of each of the stroke parameter, where "Controllers" shows how the parameters of each stroke are configured, and N represents the total number of parameters in each stroke.

For the "marker pen" and the "watercolor ink", we design the main trajectory of the stroke movement as a quadratic Bezier curve. The shape of the Bezier curve is specified by three control points $P_0 = (x_0, y_0)$, $P_1 = (x_1, y_1)$, and $P_2 = (x_2, y_2)$. Formally, the stroke is defined as

$$B(t) = (1-t)^2 P_0 + 2(1-t)tP_1 + t^2 P_2, \tag{1}$$

	Layer	Config	Out size
F1	Full-connected + ReLU	512	512
F2	Full-connected + ReLU	1024	1024
F3	Full-connected + ReLU	2048	2048
F4	Full-connected + ReLU	4096	4096
V1	View	-	16×16×16
C1	Conv + ReLU	32×3×3 / 1	16×16×32
C2	Conv + Shuffle	32×3×3 / 2	32×32×8
C3	Conv + ReLU	16×3×3 / 1	32×32×16
C4	Conv + Shuffle	16×3×3 / 2	64×64×4
C5	Conv + ReLU	8×3×3 / 1	64×64×8
C6	Conv + Shuffle	4×3×3 / 2	128×128×1

Table 2: Details of our rasterization network.

where $0 \leq t \leq 1$. We define another set of parameters to control the stroke thickness and color. For the watercolor ink, we define thickness (r_0, r_2) and colors $(R_0, G_0.B_0, R_2, G_2, B_2)$ separately at P_0 and P_2, while for marker pen, we use constant thickness d and color (R, G, B) in each stroke trajectory.

For the "color-tapes", we define it as a solid color rectangle with a rotation angle $\theta \in [0, 180°]$. The position and size are define by (x_0, y_0, h, w).

For the "oil-painting brush", we define its parameters similar to the color-tapes. The shape parameters include position, size, and orientation. The colors are defined at the head and tail of the stroke separately $(R_0, G_0.B_0, R_2, G_2, B_2)$. Since oil paints are not transparent, we ignore the transparency parameter A and set it to a constant $A = 1$. We also blend a texture map on top of the rendered stroke but we simply treat the texture as a constant map which is not updated during the parameter searching.

		Controllers	N
Oil-painting brush	Shape	x_0, y_0, h, w, θ	5
	Color	$R_0, G_0, B_0, R_1, G_2, B_2, (A = 1)$	6
Markerpen	Shape	$x_0, y_0, x_1, y_1, x_2, y_2, d$	7
	Color	R, G, B, A	3
Watercolor ink	Shape	$x_0, y_0, x_1, y_1, x_2, y_2, r_0, r_2$	8
	Color	$R_0, G_0, B_0, R_1, G_2, B_2, A$	7
Color-tapes	Shape	x_0, y_0, h, w, θ	5
	Color	$R, G, B, (A = 1)$	3

Table 3: Detailed parameterization of oil-painting strokes.