Supplementary Material: De-rendering the World's Revolutionary Artefacts

Shangzhe Wu[1,4*] Ameesh Makadia[4] Jiajun Wu[2]
Noah Snavely[4] Richard Tucker[4] Angjoo Kanazawa[3,4]

[1]University of Oxford [2]Stanford University
[3]University of California, Berkeley [4]Google Research

1. Training Details

All hyper-parameter settings are specified in Table 1 and the network architectures in Tables 2 to 5. Abbreviations of the components are defined as follows:

- Conv$(c_{in}, c_{out}, k, s, p)$: 2D convolution with c_{in} input channels, c_{out} output channels, kernel size k, stride s and padding p.
- Deconv$(c_{in}, c_{out}, k, s, p)$: 2D deconvolution with c_{in} input channels, c_{out} output channels, kernel size k, stride s and padding p.
- Upsample(s): 2D nearest-neighbor upsampling with a scale factor s.
- Linear(c_{in}, c_{out}): linear layer with c_{in} input channels and c_{out} output channels.
- GN(n): group normalization [8].
- IN: instance normalization [6] with n groups.
- LReLU(p): leaky ReLU [5] with a slope p.
- Conv1D and Upsample1D are similarly defined.

2. Synthetic Vases

We generate a synthetic vase dataset in order to conduct quantitative assessment of our de-rendering results. Examples of the synthetic vases are shown in Fig. 2. The detailed procedure to generate this dataset is described in the following.

SoR shapes. We simulate vase-like SoR curves $\mathbf{r} \in \mathbb{R}^L$ using a combination of two sine curves, where L is set to be 32, and each entry r_i is given by:

$$\begin{aligned} r_i &= t + f_1(i) + f_2(i) \\ f_1(i) &= a_1 \cdot (1 + \sin(\frac{L-i}{L} \cdot p_1 + \frac{i}{L} \cdot q_1)) \\ f_2(i) &= a_2 \cdot (1 + \sin(p_2 + \frac{i}{L} \cdot q_2)), \end{aligned} \tag{1}$$

where the random variables are $t \sim \mathcal{U}(0.1, 0.3)$, $a_1 \sim \mathcal{U}(0, 0.3)$, $p_1 \sim \mathcal{U}(-\pi, 0)$, $q_1 \sim \mathcal{U}(\frac{\pi}{2}, 2\pi)$, $a_2 \sim \mathcal{U}(0, 0.1)$, $p_2 \sim \mathcal{U}(0, 2\pi)$ and $q_2 \sim \mathcal{U}(\frac{\pi}{2}, 2\pi)$.

We then render the vases with random elevation angles between 0° and 20°, using a projective camera with a field of view of 10°.

Material. We generate random diffuse albedo maps using texture images from a public material dataset (CC0 Textures [1]), with random augmentations in brightness, contrast and hue. Shininess constant α is randomly sampled between 1 and 196 and specular albedo constant ρ is sampled between 0.1 and 1.

Lighting. We synthesize environment illumination using 3 random spherical Gaussian lobes [7, 4]:

$$L(\eta) = \sum_{k=1}^{3} \sqrt{\lambda_k} F_k G(\eta; \xi_k, \lambda_k), \;\; G(\eta; \xi, \lambda) = e^{-\lambda(1-\eta \cdot \xi)}, \tag{2}$$

where ξ_k controls the direction of each lobe and is a unit vector randomly sampled from the upper-front quarter of the sphere, $\lambda_k \sim \mathcal{U}(10, 30)$ controls the bandwidth, and $F_k \sim \mathcal{U}(0.1, 0.3)$ controls the intensity.

3. Additional Results

Fig. 1 shows a visual comparison of the results obtained from the ablation experiments as well as a supervised baseline, corresponding to the numerical results reported in Table 2 of the paper.

*The work was primarily done during an internship at Google Research.

Parameter	Value/Range
Optimizer	Adam
Learning rate	2×10^{-4}
Number of iterations	40k
Batch size	24
Loss weight λ_{s}	10
Loss weight λ_{dt}	100
Loss weight λ_{im}	1
Loss weight λ_{alb}	1
Loss weight λ_{SAD}	0.01
Loss weight λ_{diff}	1
Input image size	256×256
Whole unwrapped image size	256×768
Frontal unwrapped image size	256×256
Vertex grid size	32×96
Environment map size	16×48
Field of view (FOV)	10°
Radius $\hat{\mathbf{r}}$	$(0.05, 0.9)$
Radius column height $\hat{h}$	$(0.5, 0.95)$
Pitch angles	$(0^\circ, 20^\circ)$
Roll angles	$(-10^\circ, 10^\circ)$
Translation in X, Y axes	$(-0.2, 0.2)$
Albedo $\hat{A}$	$(0, 1)$
Shininess $\hat{\alpha}$	$(1, 196)$
Specular albedo $\hat{\rho}$	$(0, 2)$
Environment map $\hat{E}$	$(0, 1)$

Table 1: Training details and hyper-parameter settings.

Encoder	Output size
Conv(3, 64, 4, 2, 1) + ReLU	128×128
Conv(64, 128, 4, 2, 1) + ReLU	64×64
Conv(128, 256, 4, 2, 1) + ReLU	32×32
Conv(256, 512, 4, 2, 1) + ReLU	16×16
Conv(512, 512, 4, 2, 1) + ReLU	8×8
Conv(512, 512, 4, 2, 1) + ReLU	4×4
Conv(512, 128, 4, 1, 0) + ReLU	1×1
Decoder	**Output size**
Upsample1D(2) + Conv1D(128, 128, 3, 1, 1) + ReLU	2
Upsample1D(2) + Conv1D(128, 128, 3, 1, 1) + ReLU	4
Upsample1D(2) + Conv1D(128, 128, 3, 1, 1) + ReLU	8
Upsample1D(2) + Conv1D(128, 128, 3, 1, 1) + ReLU	16
Upsample1D(2) + Conv1D(128, 128, 3, 1, 1)	32
↳ Sigmoid → output $\hat{\mathbf{r}}$	32
Linear(128, 128) + ReLU	1
Linear(128, 5)	1
Sigmoid → output $\hat{h}, \hat{v}$	1

Table 2: Architecture of the shape network f_S. The network outputs radius column $\hat{\mathbf{r}}$, height $\hat{h}$ and camera pose $\hat{v}$ from two branches.

Additional decomposition and relighting results of real vases are shown in Fig. 3 (from Metropolitan Museum collection [2]) and in Fig. 4 (from Open Images [3]). See the video for more visual results, including animations of rotating vases as well as relighting effects.

Encoder	Output size
Conv(3, 64, 4, 2, 1) + GN(16) + LReLU(0.2)	128×128
Conv(64, 128, 4, 2, 1) + GN(32) + LReLU(0.2)	64×64
Conv(128, 256, 4, 2, 1) + GN(64) + LReLU(0.2)	32×32
Conv(256, 512, 4, 2, 1) + GN(128) + LReLU(0.2)	16×16
Conv(512, 512, 4, 2, 1) + GN(128) + LReLU(0.2)	8×8
Conv(512, 512, 4, 2, 1) + LReLU(0.2)	4×4
Conv(512, 128, 4, 1, 0) + ReLU	1×1
Decoder	**Output size**
Deconv(128, 512, (2,6), 1, 0) + ReLU	2×6
Upsample(2) + Conv(512, 256, 3, 1, 1) + GN(64) + ReLU	4×12
Upsample(2) + Conv(256, 128, 3, 1, 1) + GN(32) + ReLU	8×24
Upsample(2) + Conv(128, 64, 3, 1, 1) + GN(16)	16×48
↳ Sigmoid → output $\hat{E}$	16×48
Linear(128, 128) + ReLU	1
Linear(128, 2)	1
Sigmoid → output $\hat{\alpha}, \hat{\rho}$	1

Table 3: Architecture of the light network f_L. The network outputs environment map $\hat{E}$ and specular albedo $\hat{\rho}$ from two branches.

Encoder	Output size
Conv(3, 64, 4, 2, 1) + IN + LReLU(0.2)	128×128
Conv(64, 128, 4, 2, 1) + IN + LReLU(0.2)	64×64
Conv(128, 256, 4, 2, 1) + IN + LReLU(0.2)	32×32
Conv(256, 512, 4, 2, 1) + IN + LReLU(0.2)	16×16
Conv(512, 512, 4, 2, 1) + IN + LReLU(0.2)	8×8
Conv(512, 512, 4, 2, 1) + IN + LReLU(0.2)	4×4
Decoder	**Output size**
Upsample(2) + Conv(512, 512, 3, 1, 1) + IN + SC + ReLU	8×8
Upsample(2) + Conv(512, 256, 3, 1, 1) + IN + SC + ReLU	16×16
Upsample(2) + Conv(512, 256, 3, 1, 1) + IN + SC + ReLU	32×32
Upsample(2) + Conv(256, 128, 3, 1, 1) + IN + SC + ReLU	64×64
Upsample(2) + Conv(128, 64, 3, 1, 1) + IN + SC + ReLU	128×128
Upsample(2) + Conv(64, 3, 3, 1, 1)	256×256
Tanh → output $\hat{A}$	256×256

Table 4: Architecture of the albedo network f_A. The network follows a U-Net structure with skip-connections and replaces deconvolution with nearest neighbor upsampling followed by convolution.

Encoder	Output size
Conv(3, 64, 4, 2, 1) + IN + LReLU(0.2)	32×32
Conv(64, 128, 4, 2, 1) + IN + LReLU(0.2)	16×16
Conv(128, 256, 4, 2, 1) + IN + LReLU(0.2)	8×8
Conv(256, 512, 4, 2, 1) + LReLU(0.2)	4×4
Conv(512, 1, 4, 1, 0) → output scalar	1×1

Table 5: Architecture of the discriminator network D. The network outputs a single scalar for each input patch.

References

[1] Cc0 textures. https://cc0textures.com. Accessed: 2020-06. 1

[2] The metropolitan museum of art open access. https://metmuseum.github.io. Accessed: 2020-06. 2, 4

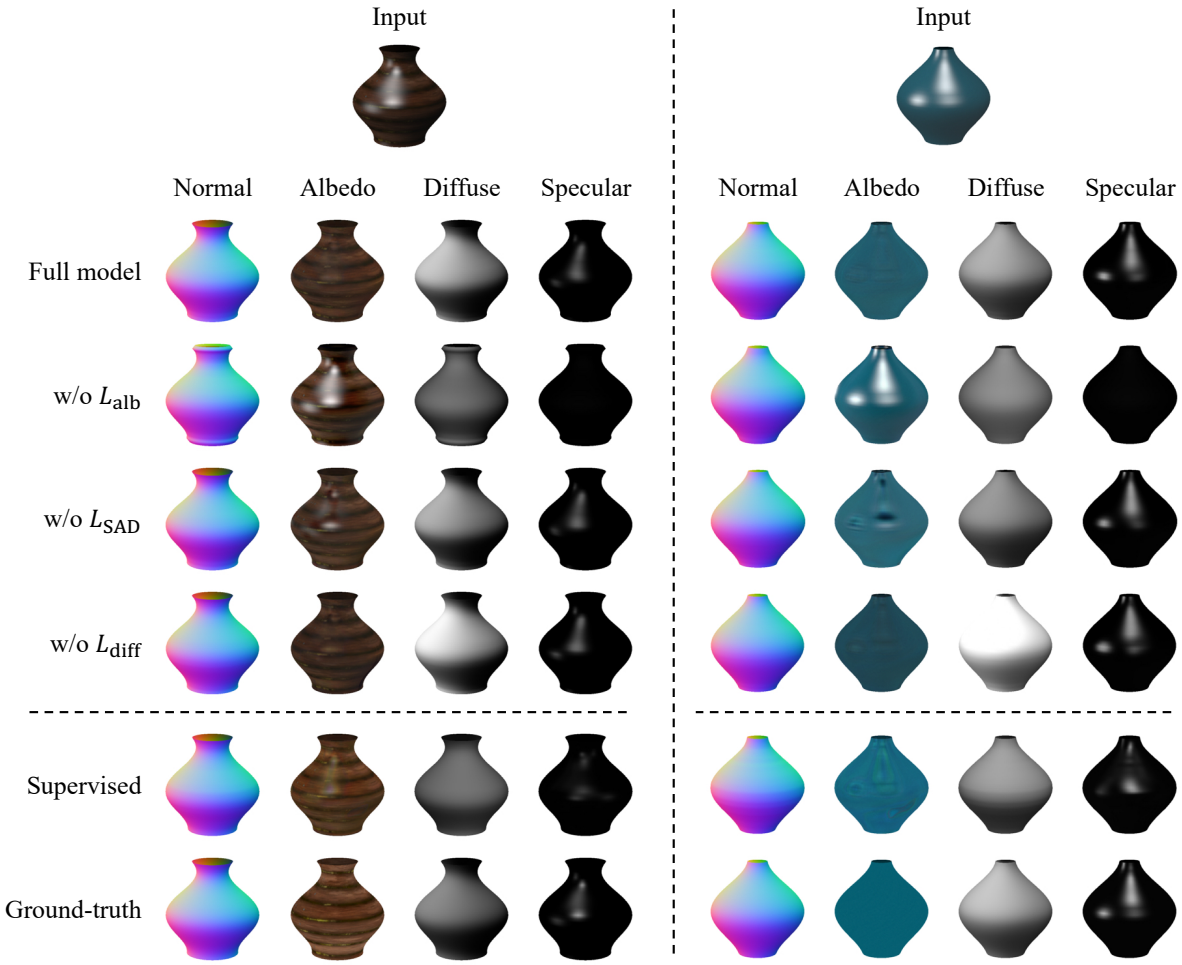

Figure 1: Qualitative comparison of the ablation experiments and the supervised baseline.

Figure 2: Examples of the synthetic vases.

[3] Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Alexander Kolesnikov, Tom Duerig, and Vittorio Ferrari. The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. *IJCV*, 2020. 2, 4

[4] Zhengqin Li, Mohammad Shafiei, Ravi Ramamoorthi, Kalyan Sunkavalli, and Manmohan Chandraker. Inverse rendering for complex indoor scenes: Shape, spatially-varying lighting and svbrdf from a single image. In *CVPR*, 2020. 1

[5] Andrew L. Maas, Awni Y. Hannun, and Andrew Y. Ng. Rectifier nonlinearities improve neural network acoustic models. In *ICML*, 2013. 1

[6] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Instance normalization: The missing ingredient for fast stylization. *arXiv preprint arXiv:1607.08022*, 2017. 1

[7] Jiaping Wang, Peiran Ren, Minmin Gong, John Snyder, and Baining Guo. All-frequency rendering of dynamic, spatially-varying reflectance. *ACM TOG*, 2009. 1

[8] Yuxin Wu and Kaiming He. Group normalization. In *ECCV*, 2018. 1

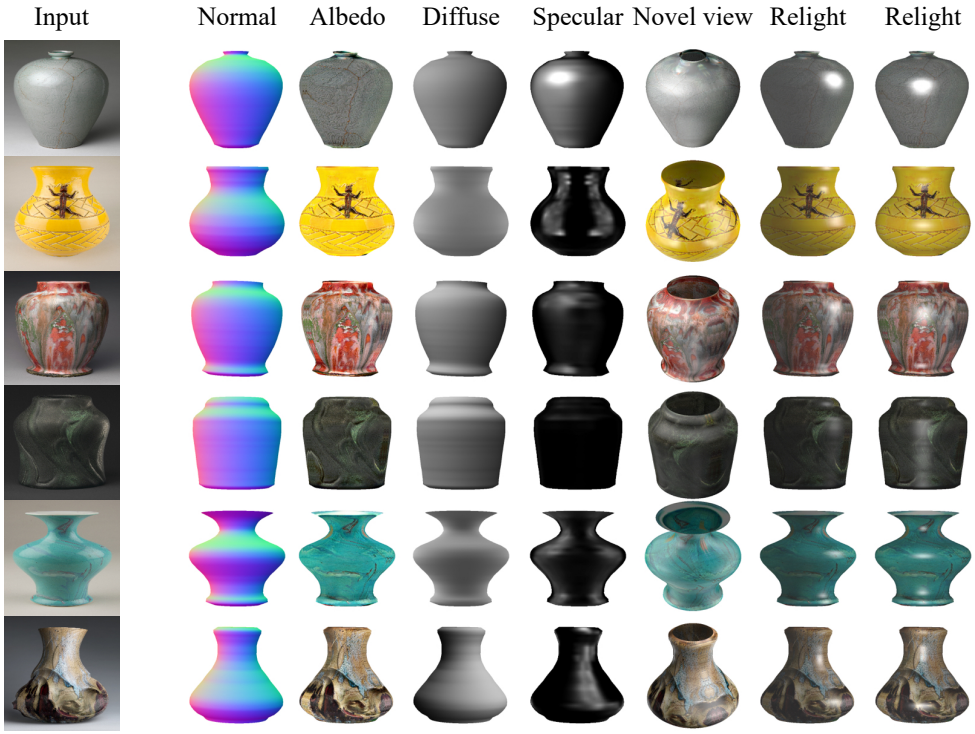

Figure 3: Additional results on Metropolitan Museum collection [2].

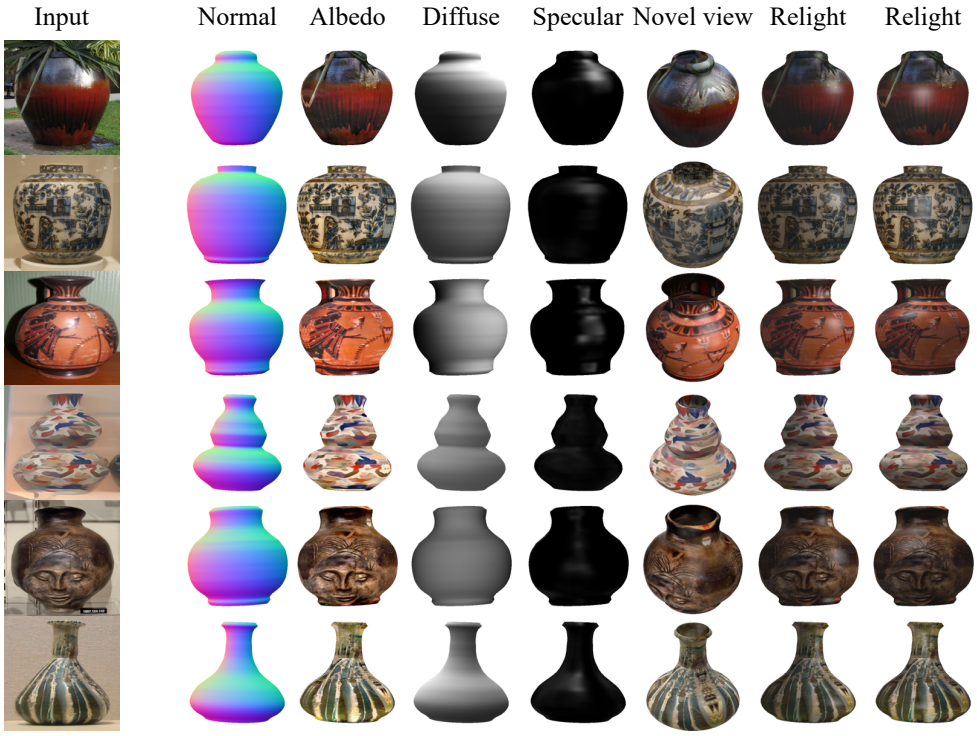

Figure 4: Additional results on Open Images vases [3].