A. Supplemental

A.1. Reprojection Error

We calculate reprojection error as the L_1 distance between source frame and reprojected target frame. This allows to quantify the 3D consistency. While a texture representation is 3D consistent by design, unoptimized texels might still create visible noise artifacts when rendering a trajectory. We calculate reprojection error on the ScanNet [2] dataset by using their captured camera trajectories. For each source frame, we select a target frame that is (a) two frames after the source frame (short-range consistency) or (b) 20 frames after the source frame (long-range consistency). Using the estimated poses and camera intrinsics, we warp the pixels of the target frame to the source view. We calculate L_1 distance in the normalized image range $[0, 1]$ between reprojected target frame and source frame for all pixels that are visible in both views. Please see Fig. 1 for a visualization of the procedure.

Because the estimated poses are not perfectly accurate, the reprojection error may never sink below a certain threshold that captures this inaccuracy. Still, it allows to quantify the 3D consistency by measuring the additional inconsistencies caused by unoptimized textures, i.e., the error is higher for unoptimized textures that are less consistent.

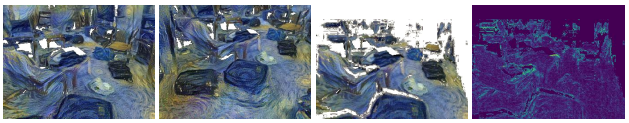


Figure 1. Sample images used for calculating the reprojection error. We select a source frame and a target frame and warp the target pixels from the target view to the source view. Reprojection error is calculated as the L_1 distance of all pixels between source frame and reprojected target frame, that are visible in both views.

A.2. User Study Setup

We conduct a user study on the effectiveness of our proposed depth- and angle-awareness. Users compared our method against each baseline separately by preferring one of two images. They judged in which image stylization patterns (a) have less visible stretch and (b) are smaller in the background. In total, 20 users each answered 70 questions, comparing against NMR [4], DIP [5] and ours without angle- and depth-awareness (Only 2D). We show two sample questions, one for each type, in Fig. 2. As can be seen, users have the possibility to decide for one of two images or to answer that none of the two is better/worse. The order of questions and of the "A" and "B" images is random and different for each user. Users have the possibility to zoom-in on the images for better judgement. Additionally, we add rectangles on image regions that might be especially interesting for evaluation of the questions. Note that users still had to consider the whole image in their answer; the rectangles merely act as additional input.

A.3. Variation of Depth Levels

The number of depth levels θ_l controls the depth variation that can be achieved within rendered poses of a scene. Setting θ_l=4 is sufficient for our datasets, as larger scene extent is rarely captured by many pixels. We could precompute *uv* maps at larger resolutions to enable depth scaling at even larger depth values. Small scenes may not require the last layers, in which case they are simply not utilized during optimization. Adding more layers in-between maps less pixels to one layer, which can yield insufficient Gram matrices and is computationally more expensive. Decreasing θ_l reduces size variation at different depths (see Fig. 3).

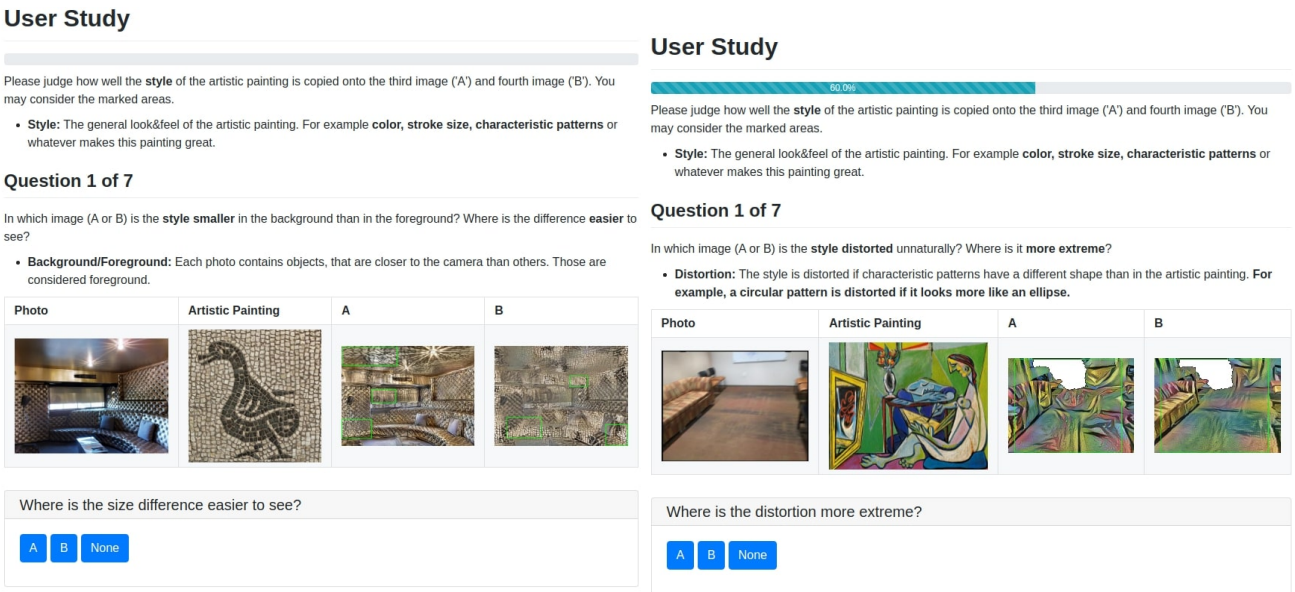
A.4. Circle-Stretch and -Size Metric

We describe in more detail the metrics and principles used in the main paper to quantify the effects of our depth and angle awareness. In order to measure the effects, we stylize a scene with a hand-crafted "circle" image (see main paper) and only use the (multi-resolution, part-based) style loss. After optimization, the red circles are stylized all over the scene and are well-suited to describe the two drawbacks of missing angle- and depth-awareness. For example, circles become ellipsoidal if a small grazing angle is used for stylization and circles change their radius inconsistently without depth awareness. We can now measure the degree we alleviate these issues by measuring the *size* and *stretch* of the circles/ellipses. Naturally, NST creates ellipses of different shapes, but their overall distribution reveals the degree of 3D awareness for the complete scene.

A.4.1 Segmentation of Ellipses

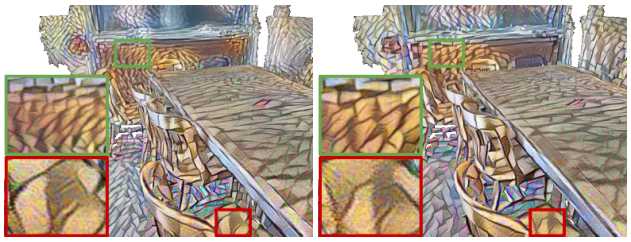
First, we automatically segment red ellipses from each image of the stylized scene (see Fig. 4). We first apply an HSV filter and only keep pixels in the ranges $0.6 \leq S, V \leq 1.0$, $0.0 \leq H \leq 0.08$ and $0.88 \leq H \leq 1.0$. Then we turn the filtered image into a binary mask by thresholding colors above 0.15 intensity and denoise it with OpenCV's "fastNLMeansDenoising" function [1]. Afterwards, we use OpenCV's contour detection to get an edge map. We filter out all contours with $max_d > 2$, where max_d is the maximum deviation from a convex hull, as measured by "convexityDefects" [1]. We now fit ellipses to the remaining contours with "fitEllipse". We extract the pixel-radius as

$$r_p = \frac{h_p + v_p}{2} \tag{1}$$



(a) Size Sample (b) Stretch Sample

Figure 2. Sample images used for the user study. Users judged in which image stylization patterns (a) are smaller in the background and (b) have less visible stretch.



(a) θ_l=4 (b) θ_l=2

Figure 3. Variation of the number of depth levels θ_l. Decreasing θ_l reduces size variation at different depths.

for every fitted ellipse and calculate its pixel-stretch as

$$s_p = max(\frac{h_p}{v_p}, \frac{v_p}{h_p}) \tag{2}$$

where h_p is the pixel-length of the horizontal ellipse radius and v_p the vertical, respectively. We remove the remaining wrongfully detected ellipses with $r_p < 10$, $r_p > 1000$ and $s_p > 10$ to get a result like in Fig. 4. We use these ellipse characteristics to calculate metrics for depth- and angle-awareness.

A.4.2 Calculation of Depth Metrics

We calculate the correlation between per-pixel depth d_{xy} and ellipse radius r_p to quantify the effect of our depth-awareness in the 2D image plane (Corr. 2D). For each detected ellipse we use the depth value of the pixel corresponding to the ellipse center. A high negative correlation (e.g., -0.5) signals, that ellipse size decreases with increasing depth, whereas a low correlation (e.g., -0.05) signals, that ellipse size is independent of changes in depth. A method that is able to stylize a scene depth-aware would create ellipses with smaller size in the background and thus have a high negative correlation in the 2D image plane.

To quantify the correlation in 3D, we backproject h_p and v_p to world-space using the estimated pose and camera intrinsics and calculate the world-space radius as

$$r_w = \frac{h_w + v_w}{2} \tag{3}$$

where h_w and v_w are the backprojected axis lengths. We then calculate the correlation between r_w and the per-pixel depth d_{xy} (Corr. 3D). A high negative correlation (e.g., -0.5) signals, that ellipse size in world-space still decreases with increasing depth, whereas a low correlation (e.g., -0.05) signals, that ellipse size in world-space is independent of changes in depth. A method that is able to stylize a scene depth-aware would create ellipses with uniformly distributed size in world-space (because the ellipse size should only change when rendering a scene from different poses, due to perspective projection).

Note that the stylized ellipses naturally vary in their sizes (e.g., ellipses can be smaller and larger independent of depth). Therefore, the correlations will be precise up to a certain threshold. However, the distribution of all seg-

mented ellipses across the whole scene still allows to quantify the depth-awareness.

A.4.3 Calculation of Angle Metric

We backproject the pixel-stretch s_p back to world-space as

$$s_w = max(\frac{h_w}{v_w}, \frac{v_w}{h_w}) \tag{4}$$

. Then we calculate the arithmetic mean over all s_w values for all detected ellipses. A higher mean value means that overall we have more stretch, whereas a lower value signals a more uniform stylization result. A method that is able to stylize a scene angle-aware would create ellipses with small stretch.

A.5. Additional Qualitative Results

We show additional qualitative results for our method.

Additional comparisons on the ScanNet [2] dataset can be found in Fig. 5 and Fig. 6.

Additional comparisons for our ablation study can be found in Fig. 7.

A.6. Style Image Assets

Throughout the main paper and the supplemental material, we use style images created by artists. In Fig. 8 we list all images and give credit to their respective creators.

References

[1] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000. 1

[2] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5828–5839, 2017. 1, 3, 5

[3] Hsin-Ping Huang, Hung-Yu Tseng, Saurabh Saini, Maneesh Singh, and Ming-Hsuan Yang. Learning to stylize novel views. *arXiv preprint arXiv:2105.13509*, 2021. 5

[4] Hiroharu Kato, Yoshitaka Ushiku, and Tatsuya Harada. Neural 3d mesh renderer. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3907–3916, 2018. 1, 4, 5

[5] Alexander Mordvintsev, Nicola Pezzotti, Ludwig Schubert, and Chris Olah. Differentiable image parameterizations. *Distill*, 3(7):e12, 2018. 1, 4, 5

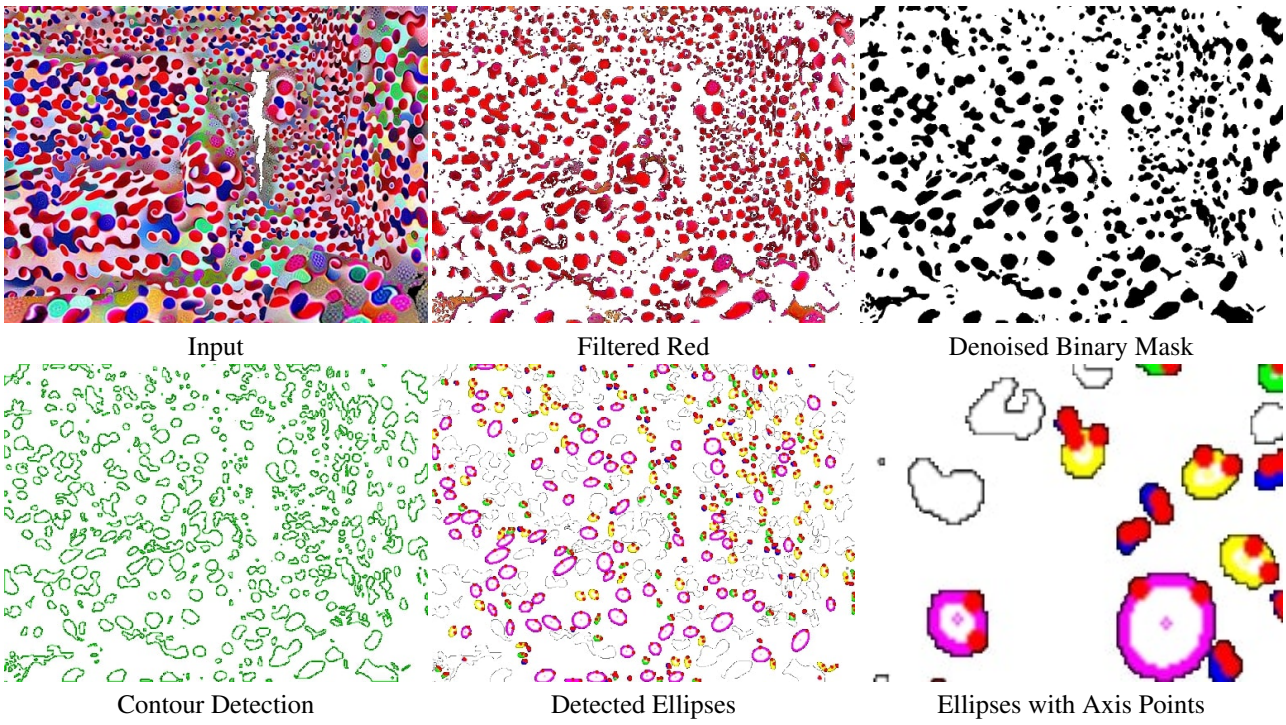


Figure 4. Our pipeline for segmenting ellipses out of a stylized image for quantification of the depth and stretch effects. We filter the image to only contain red colors (Filtered Red) and then transform that to a denoised binary image to remove remaining pixels (Denoised Binary Mask). Afterwards we use contour detection to convert the binary mask into edge maps and finally fit ellipses to all applicable contours (Detected Ellipses). Lastly, we find the horizontal and vertical axis of the ellipses as the distance from the center to corresponding points on the edge of each ellipse.

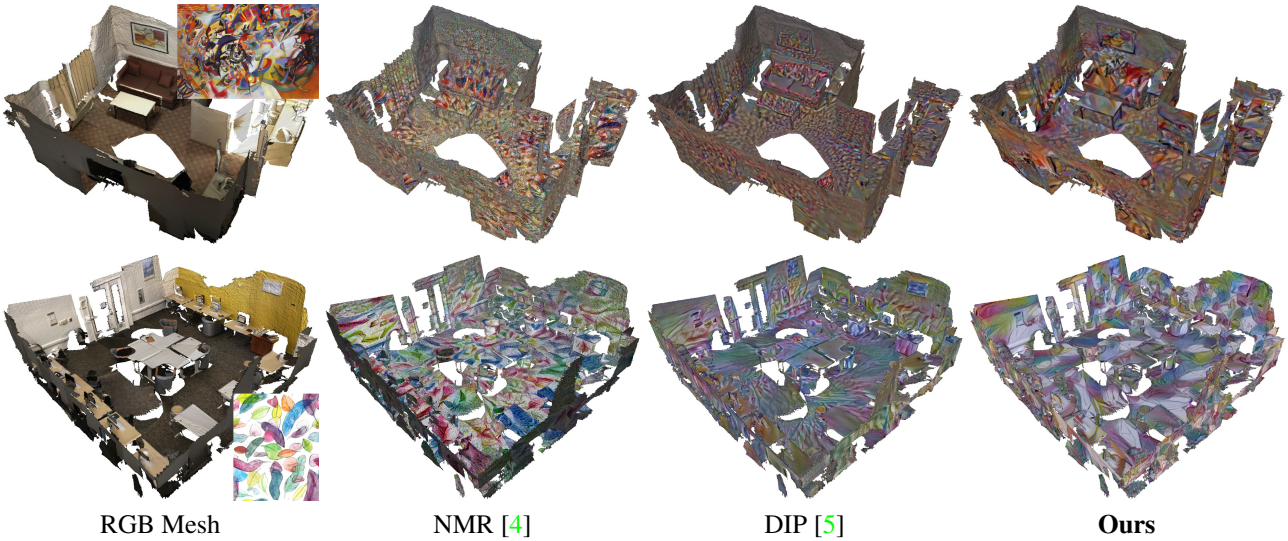


Figure 5. Top-down view on stylized meshes in comparison to previous work.

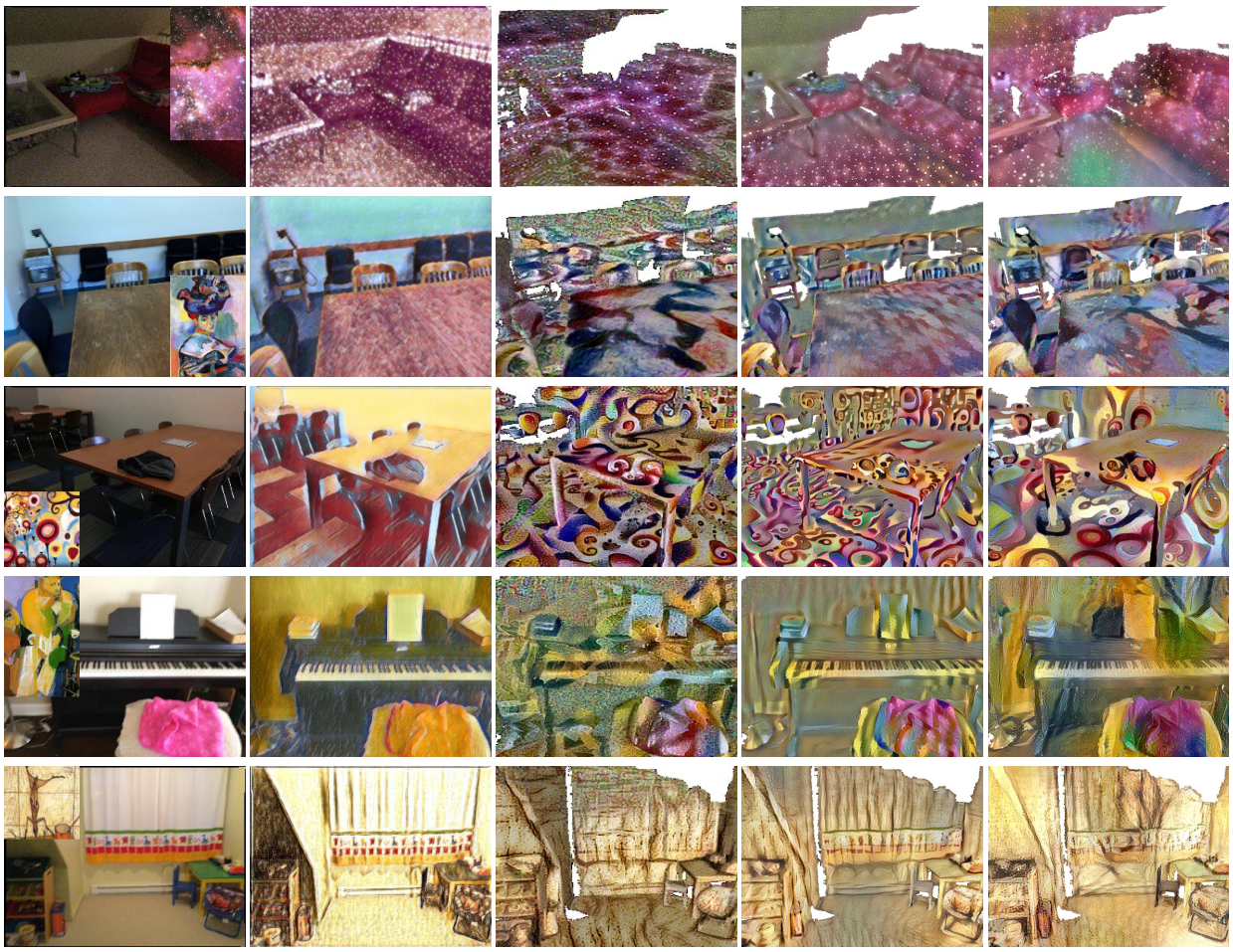
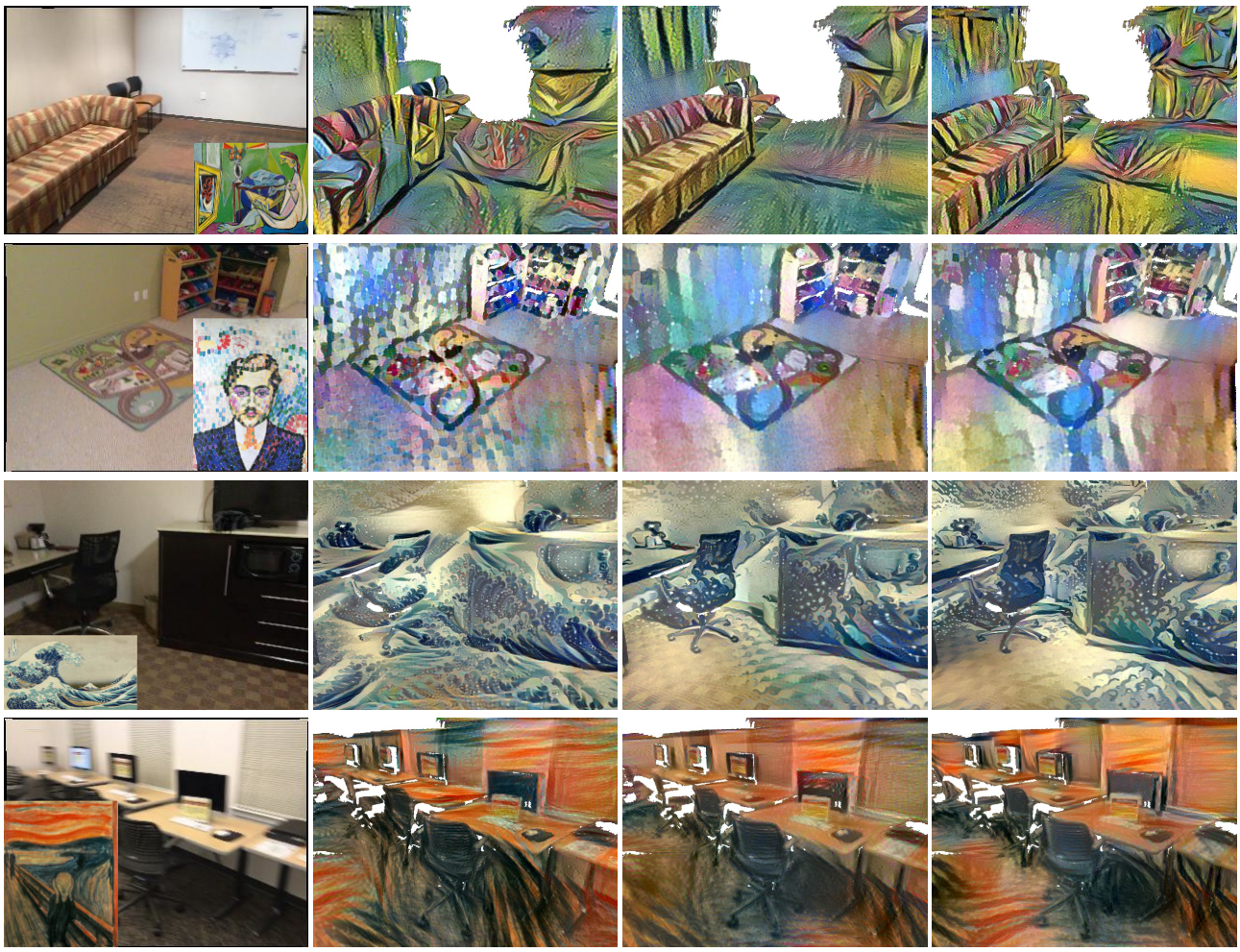


Figure 6. Comparison of stylization results for our method and related work on the ScanNet [2] dataset. We texture the mesh with each method (point cloud for Huang et al. [3] respectively) and render a single pose that is also captured in the RGB images.



(a) RGB and Style (b) Only 2D (c) With Angle (d) With Angle and Depth

Figure 7. Qualitative ablation study of our method. We compare ours (d) against only using angle (c) and not using angle and depth (b). Using angle better distinguishes surfaces and using depth creates smaller/detailed stylization in the background.

June Tree,
Natasha Wescoat

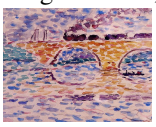
The Starry Night,
Vincent van Gogh, 1889

Femme au chapeau,
Henri Matisse, 1905

Dinamismo di un' automobile,
Luigi Russolo, 1913

The Muse,
Pablo Picasso, 1935

Il cavaliere rosso,
Carlo Carra, 1913



The Viaduct,
Henri Edmond Cross

Kanagawa oki nami ura,
Katsushika Hokusai, 1830-1832

Skrik,
Edvard Munch, 1893

Mosaic in Opus tessellatum

Mosaic (unknown),
WikiArt.org

Self-Portrait,
Pablo Picasso, 1907

Feathers Leaves and Petals,
Kathryn Corlett

Small Magellanic Cloud,
NASA, ESA and A. Nota

Edgar Poe, Charles Baudelaire, Um Orangotango e o Corvo,
Julio Pomar, 1985

Lapin et casserole rouge,
Bernard Buffet, 1948

L'homme à la tulipe,
Jean Metzinger, 1906

The Shipwreck of the Minotaur,
J.M.W. Turner, 1805

Sketch 2 for composition VII,
Wassily Kandinsky, 1913

Figure 8. List of all artistic paintings used throughout the main paper and supplemental material. We list the name of the painting and its author if known.