SelectionConv: Convolutional Neural Networks for Non-rectilinear Image Data Supplemental Material

David Hart, Michael Whitney, and Bryan Morse

Brigham Young University, Provo, Utah, USA
{davidmhart,mikeswhitney,morse}@byu.edu

1 Code

We provide our code and the network weights used for our various experiments. These are found at `http://github.com/davidmhart/SelectionConv` along with instructions and more details.

2 Masked Region Stylization

We provide an expanded version of Figure 9 from the paper here for easier viewing (Fig. 1).

We provide additional examples in Figs. 2–7.

Additionally, in Figs. 8–10, we provide examples of combining multiple styles by using multiple masks. In each of these examples, notice how stylizing the entire original image or pre-masking the background results in the style statistics being applied across the entire image even though the stylization is only intended to be applied to a portion of it. With our masked stylization, each region more completely reflects the characteristics of the respective source style.

3 Spherical Segmentation

We provide an enlarged version of Figure 7 from the paper here for easier viewing (Fig. 11). We also provide an additional example of using standard FCN pretrained segmentation weights from Pytorch on a spherical image in Fig. 12. Note the discontinuous nature of the segmentation along the seam in the naive result compared to our method.

4 Superpixel Depth Prediction

We provide an expanded version of Figure 8 from the paper here for easier viewing (Fig. 13).

5 Panoramic Stylization

Though we showed an example of removing seams when stylizing a spherical image in the paper, an even simpler problem is to attempt stylization on a 360° panoramic image. To construct the graph for such a panoramic, edges simply need to be added from the left side of the image to the right side of the image, giving one continuous loop of nodes.

The results of stylization on panoramic images is demonstrated in Fig. 14 through Fig. 17. Just like with spherical stylization, naive stylization of panoramic images results in a noticeable seam where the image wraps around horizontally. Our graph-based approach, transferring from a pre-trained image-based network, avoids such issues.

6 Spherical and Texture Map Stylization

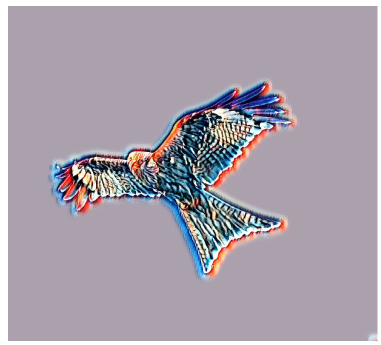
For spherical image and texture map stylization, we provide an expanded version of Figure 6 from the paper here for easier viewing (Fig. 18), and an additional spherical result in Fig. 19. We also provide an enlarged version of Figure 10 from the paper for easier viewing (Fig. 20) and additional texture map results in Fig. 21 through Fig. 23.

We also provide a supplementary video (which can be found on our project website) that includes visualizations of all the examples presented here. This provides the best visualization of these results and their advantages over the naive approach.

a) Original with Mask and Style

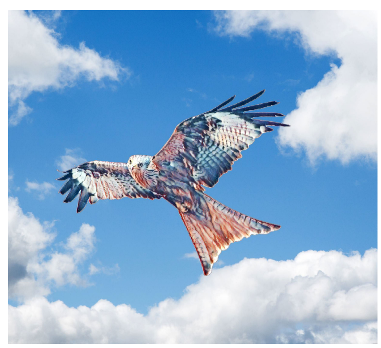
b) Entire stylized image

c) Composition of (a) and (b)



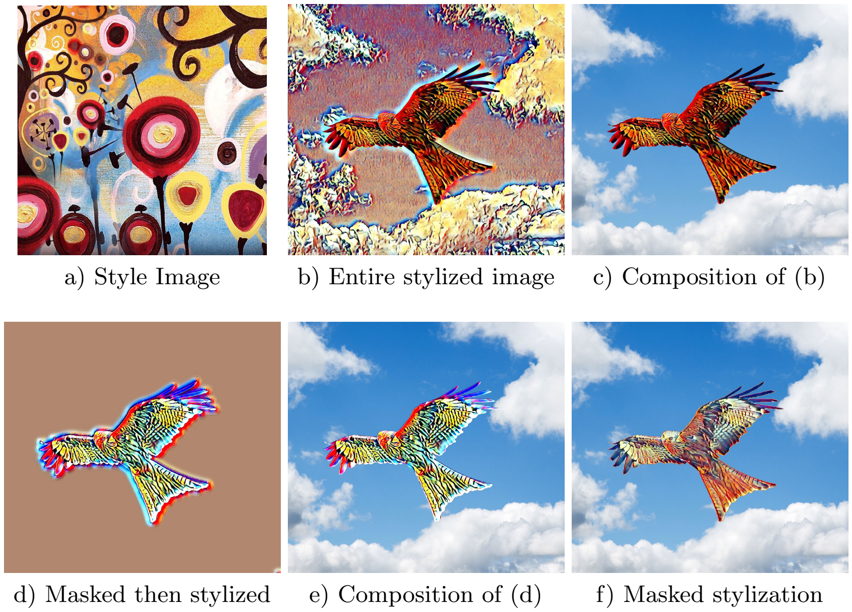
d) Masked then stylized

e) Composition of (a) and (d)



f) Masked stylization

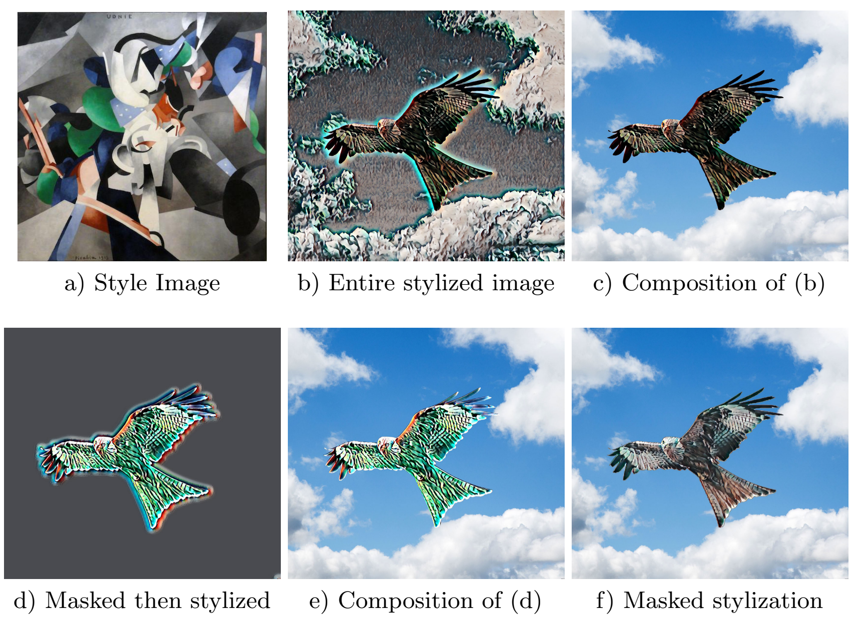
Fig. 1. A content image, a masked region of interest, and a given style image (a). To stylize the masked region with a traditional CNN, the entire image can be stylized (b) or the image can be masked before stylization (d) and then the masked result can be applied back to the original(c,e). In both cases, outside statistics influence the stylization inside the region of interest (making (c) darker than expected and (e) brighter than expected). In comparison, our method (f) can generate a graph just for the masked region, which more closely matches the style image statistics in the region of interest.



a) Style Image b) Entire stylized image c) Composition of (b)

d) Masked then stylized e) Composition of (d) f) Masked stylization

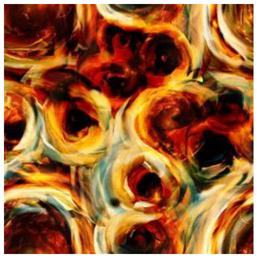
Fig. 2. Another example, similar to Fig. 1, using a different style image.



a) Style Image b) Entire stylized image c) Composition of (b)

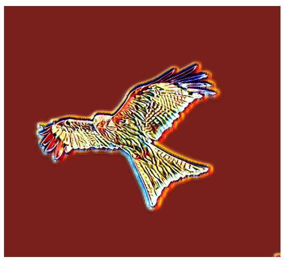
d) Masked then stylized e) Composition of (d) f) Masked stylization

Fig. 3. Another example, similar to Fig. 1, using a different style image.


a) Style Image


b) Entire stylized image


c) Composition of (b)


d) Masked then stylized


e) Composition of (d)

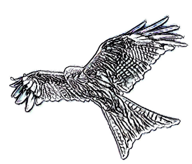

f) Masked stylization

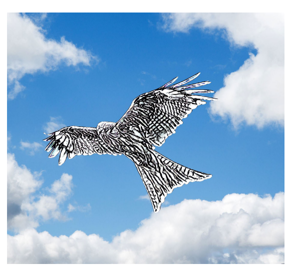
Fig. 4. Another example, similar to Fig. 1, using a different style image.


a) Style Image


b) Entire stylized image

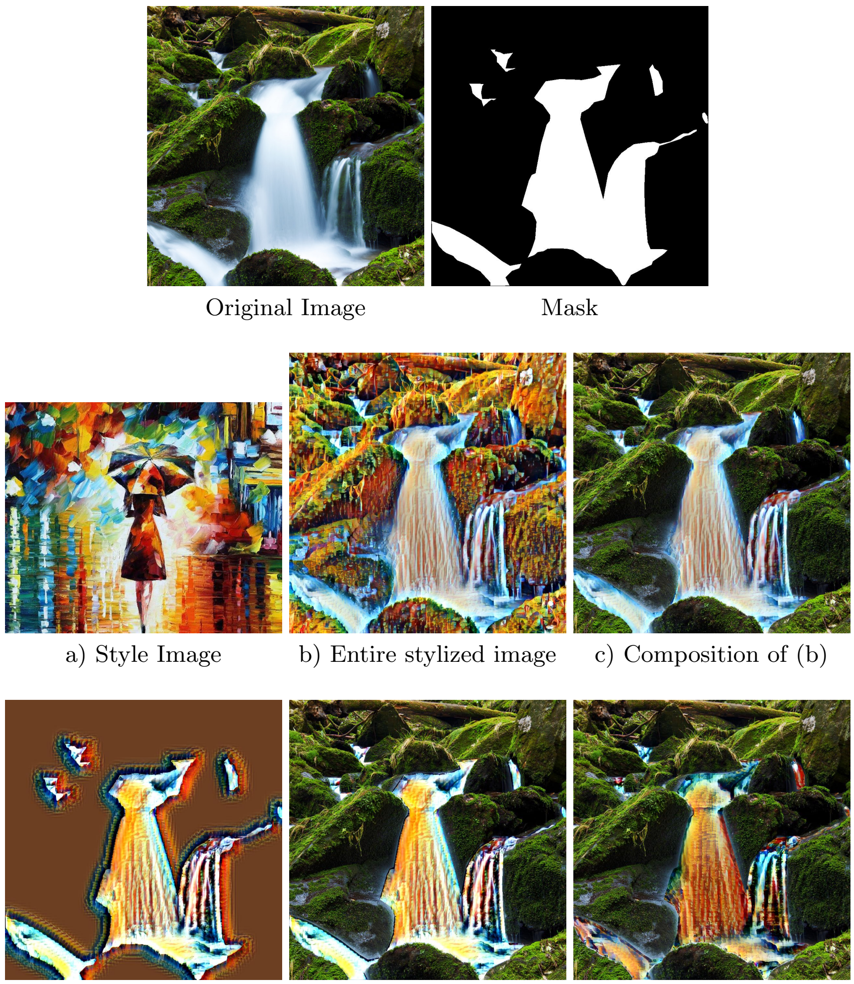

c) Composition of (b)


d) Masked then stylized


e) Composition of (d)


f) Masked stylization

Fig. 5. Another example, similar to Fig. 1, using a different style image.



Original Image Mask

a) Style Image b) Entire stylized image c) Composition of (b)

d) Masked then stylized e) Composition of (d) f) Masked stylization

Fig. 6. Another example, similar to Fig. 1, using both a different content image and a different style image.

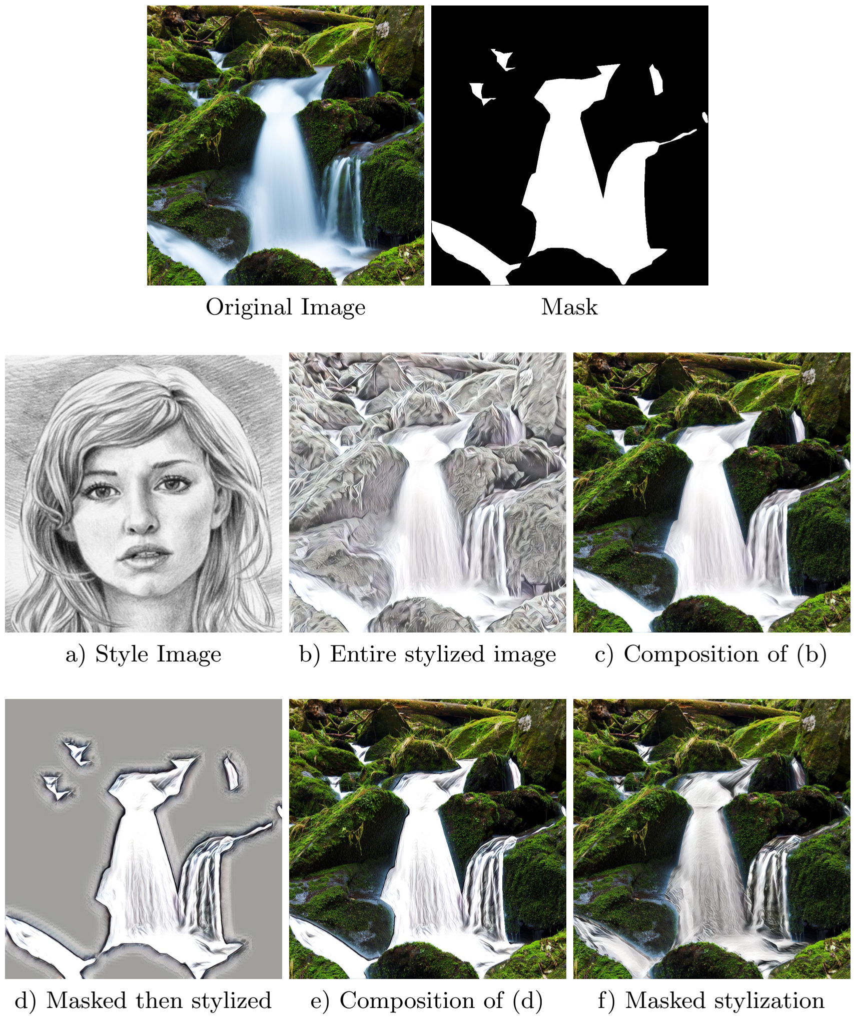


Fig. 7. Another example, similar to Fig. 6, using a different style image.

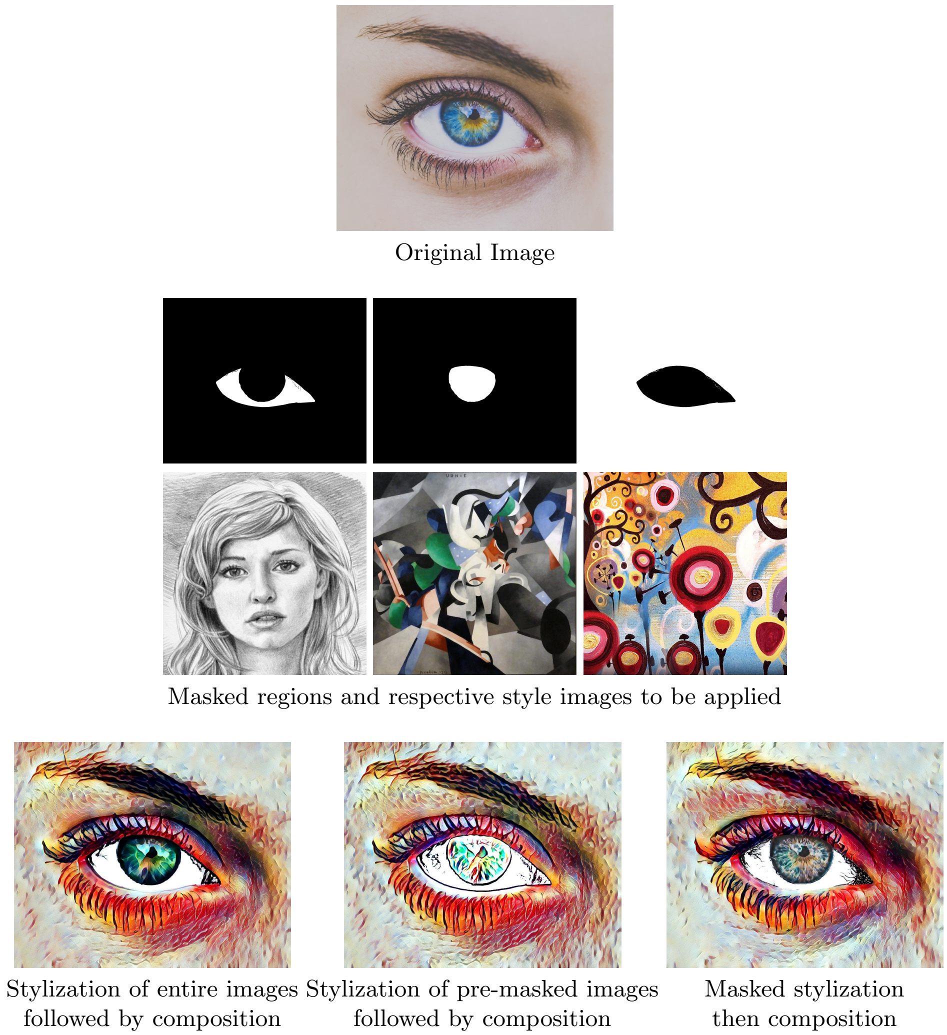


Fig. 8. Examples of using multiple styles and masks. The original image (top) is separated into three masked regions, each with their respective style to be applied (middle). Stylizing the entire image using each of the three styles distributes characteristics of the respective styles images across the entire image before masking, resulting in a composition where each region captures only a portion of its respective style (bottom left). Pre-masking each part of the image and applying the respective styles likewise distributes style characteristics across the entire image since the stylization cannot ignore the pre-masked regions, again resulting in composited regions that capture only a portion of their respective styles (bottom center). Using our masked stylization approach, the stylization is applied to each region without regard to the rest of the image, resulting in a composition where each region reflects more of its respective source style (bottom right).

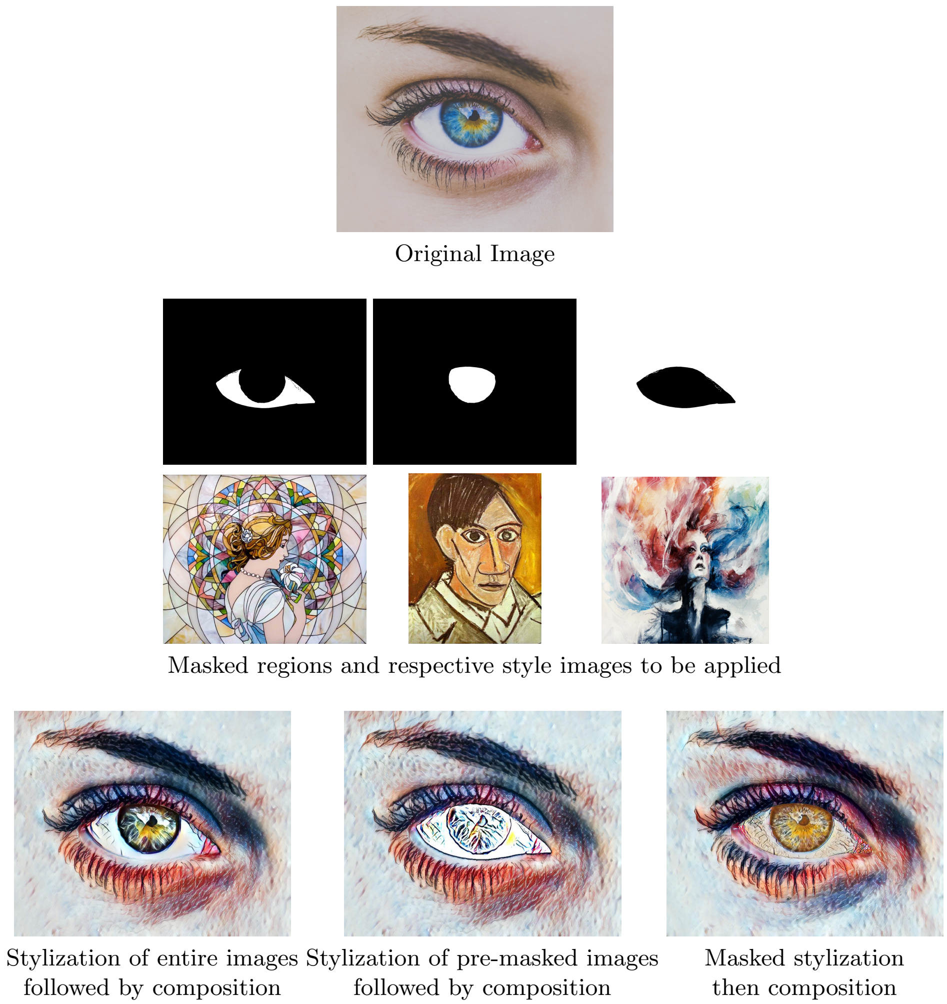
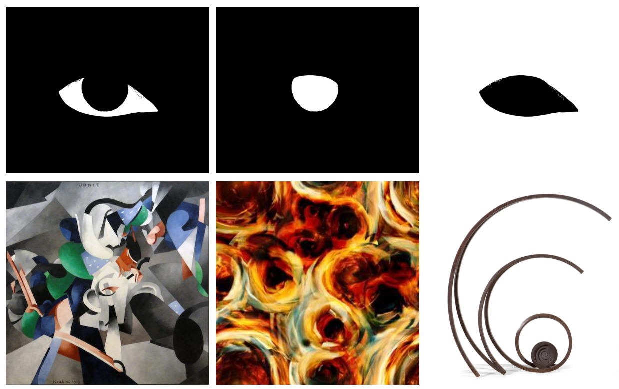


Fig. 9. Another example using multiple styles and masks. The original content image and masked regions are the same as in in Fig. 8, with a different set of styles applied to the regions. Again, masked stylization results in regions that each better represent the full content of the respective source style.

Original Image



Masked regions and respective style images to be applied

Stylization of entire images followed by composition | Stylization of pre-masked images followed by composition | Masked stylization then composition

Fig. 10. Another example using multiple styles and masks. The original content image and masked regions are the same as in Fig. 8, with a different set of styles applied to the regions. Again, masked stylization results in regions that each better represent the full content of the respective source style.

a)
b)
c)
d)

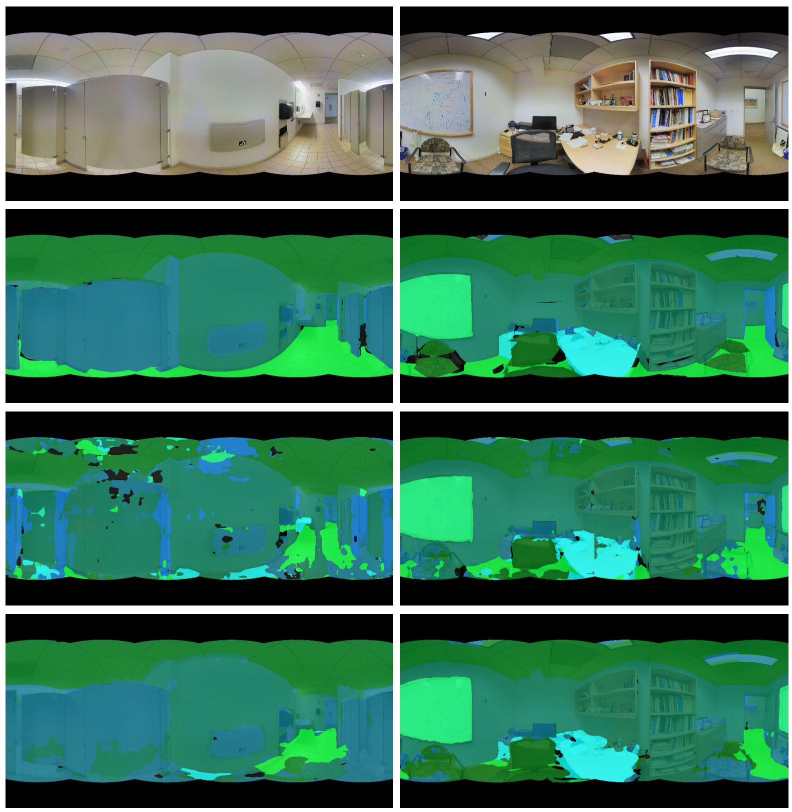
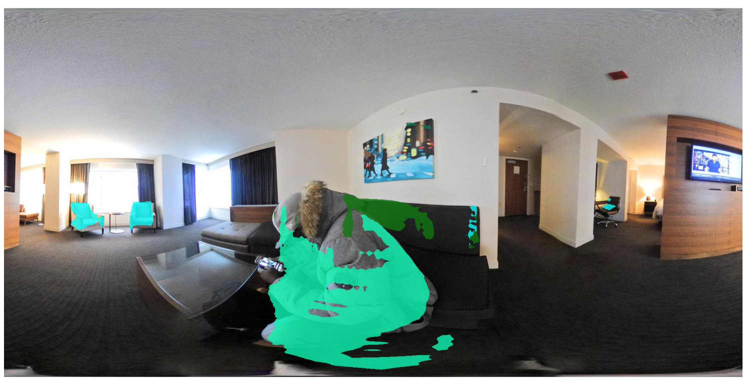


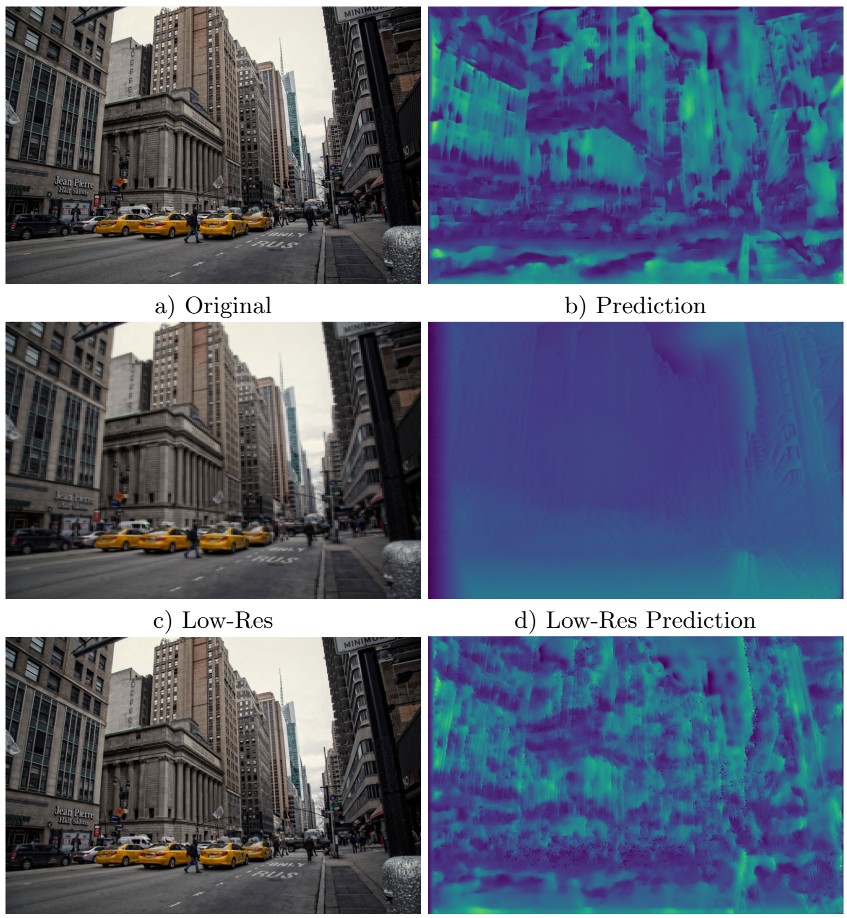
Fig. 11. A visual comparison of semantic segmentation of images from the Stanford 2D-3D-S dataset (a) with ground truth (b) between an FCN with a ResNet-50 backbone using standard convolutions (c) versus our SelectionConv operations (d). Note that the use of SelectionConv gives cleaner segmentation results along the poles of and seam of the image (located in the center of this representation).

a) Naive



b) Ours

Fig. 12. Segmentation comparison using a ResNet-50 backbone (a) and our method with transferred weights (b). Both are circularly rotated by 180 degrees to place the original vertical seam location in the center. Note that naive CNN-based segmentation results in a disjoint region for the foreground person while the proposed SelectionConv method allows for more complete selection.



a) Original b) Prediction

c) Low-Res d) Low-Res Prediction

e) Superpixel f) Superpixel Prediction

Fig. 13. A high-resolution image (a) requires ∼25.9 seconds on a CPU to create a predicted depth map (b). A lower-resolution 256 × 256 version can be processed (c) by a network in ∼0.8 seconds on a GPU, but with low-fidelity results when upscaled to the same resolution (d). Generating approximately the same number of superpixels as the low-resolution image (e) then using our graph-based network requires only ∼5.1 seconds on a GPU with higher-fidelity results (f).

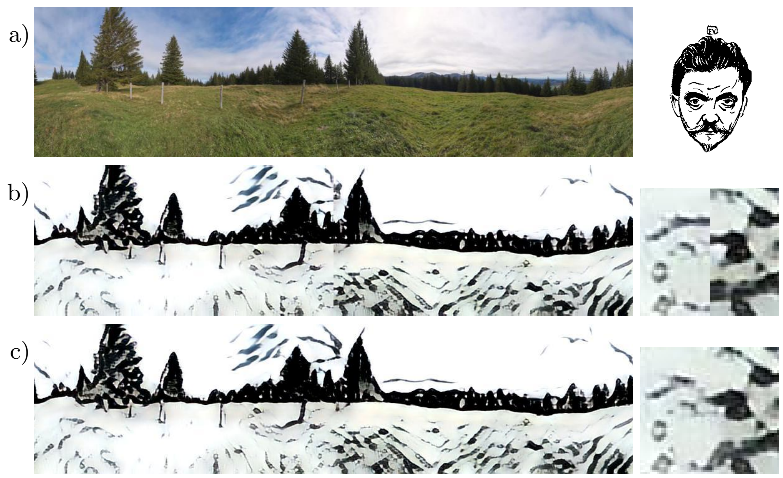


Fig. 14. A panoramic content image and a given style image (a). The stylized panoramic image using a traditional CNN (b) compared to our method (c). The images are rotated by 180 degrees to place the seam location at the center and a magnified portion is shown at the seam location.

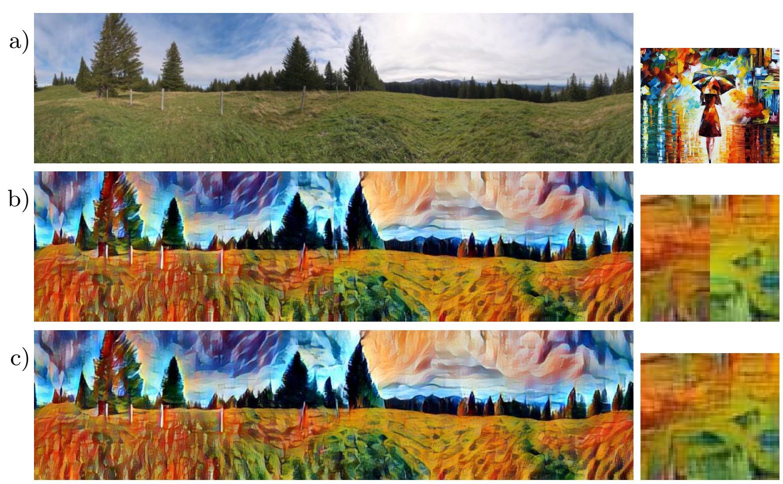


Fig. 15. Another example, similar to Fig. 14, using a different style image.

Fig. 16. Another example, similar to Fig. 14, using different content and style images.

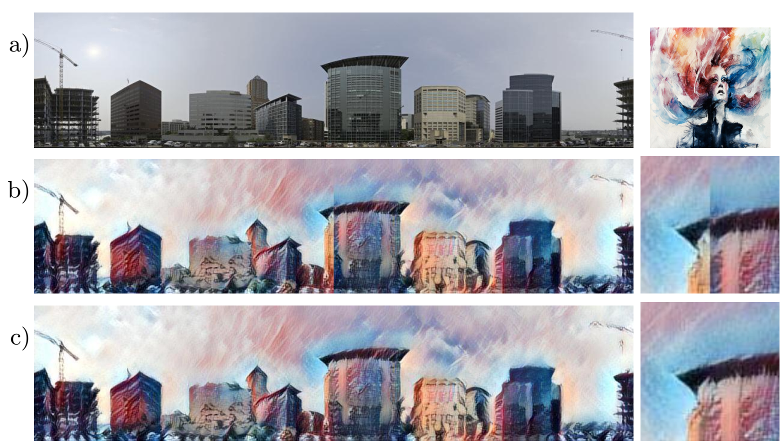
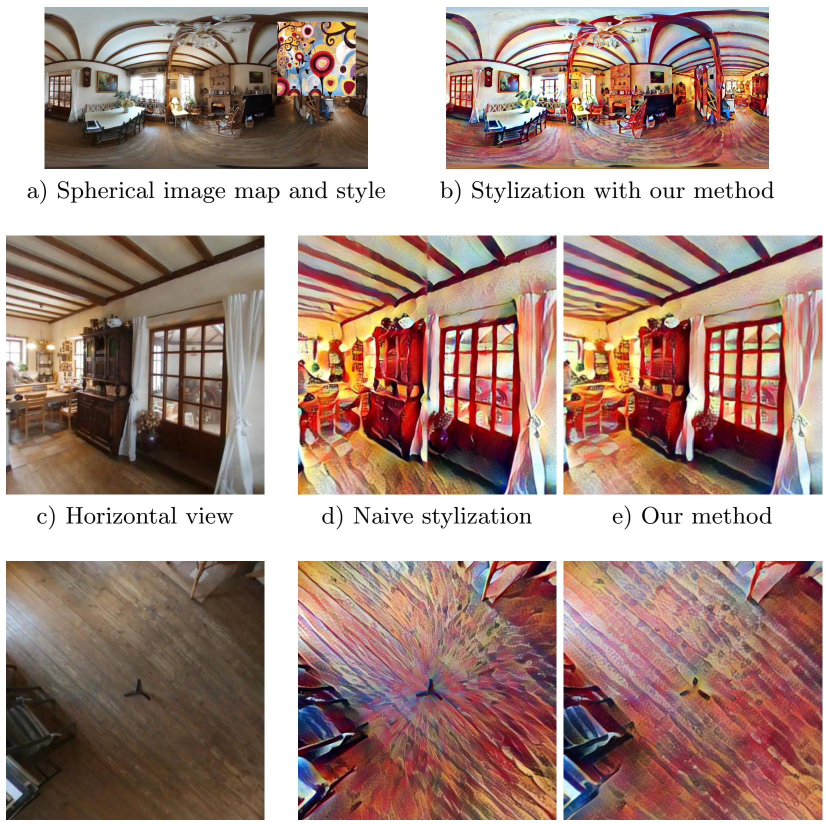


Fig. 17. Another example, similar to Fig. 16, using a different style image.



a) Spherical image map and style b) Stylization with our method

c) Horizontal view d) Naive stylization e) Our method

f) Polar view (looking down) g) Naive stylization h) Our method

Fig. 18. A 360° image (a) and its stylization using our feed-forward method (b). Example views (c,f) have seams and distortion due to discontinuities when the original image map is stylized naively (d,g), but those seams and distortion are minimized with our method (e,h). A sequence panning around the sphere is shown in our supplementary video.

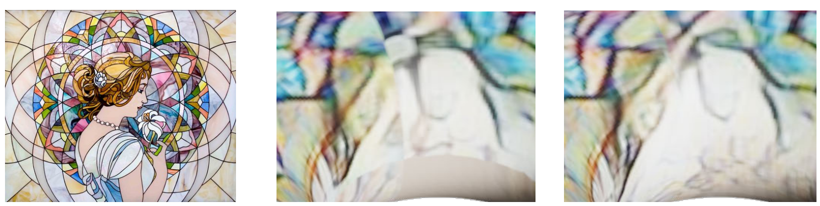
a) Original rendering

b) Stylized environment map (with reflective sphere)

Fig. 19. A rendering of a mirrored sphere with a high resolution environment map (a) and a rendering with a stylized environment map using our method (b). Note that there is no direct stylization of the reflective sphere. A visualization of this scene is shown in our supplementary video.

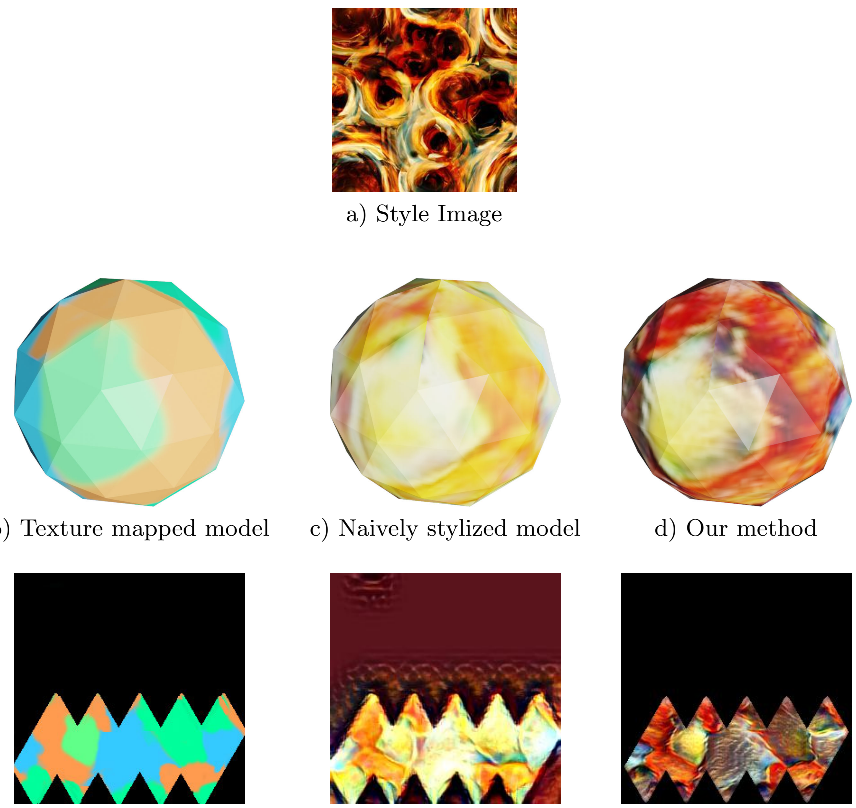
Fig. 20. 3D mesh (a) and a style image (d), the result of naively stylizing the texture map (b) and a magnification (e), and the result of using our method (c) and a magnification (f). Note the visible seams shown in the magnifications of the naive method (e), whereas our method in (f) minimizes the visibility of those seams.

a) Texture mapped model b) Naively stylized c) Our method



d) Style image e) Magnification (Naive) f) Magnification (Ours)

Fig. 21. Example of texture map stylization. When a texture mapped model (a) is stylized (d) by naively stylizing the entire texture map (b), visible seams result because the stylization ignores discontinuities inherent in texture mapping. Stylizing the texture map using a graph-based approach that respects these discontinuities removes these seams (c). Magnifications of (b) and (c) can be seen in (e) and (f) respectively. Note the clear seams in (e) for the seam running around the inner ring of the torus and around the torus from the inside to the outside. While typical artifacts of texture mapping can still be seen in (f), there are not the same discontinuities in the applied style.



a) Style Image

b) Texture mapped model c) Naively stylized model d) Our method

e) Original texture map f) Naively stylized texture map g) Our method

Fig. 22. A given style image (a), an original 3D mesh (b) and its texture map (e), the result of naively stylizing the texture map (c,f), and the result of using our method (d,g). Note that the large amount of unused space on the texture map has a significant effect on the naive approach (e), mapping the darker reds of the style image to the black (unused) part of the texture map and the lighter portions to the colored (used) part of the map, resulting in a stylized model that draws primarily from the lighter colors of the style. Our approach (f), by stylizing only the used portion of the texture (similar to Fig. 1), more accurately captures the style within the region of interest, resulting in more complete use of the source style.

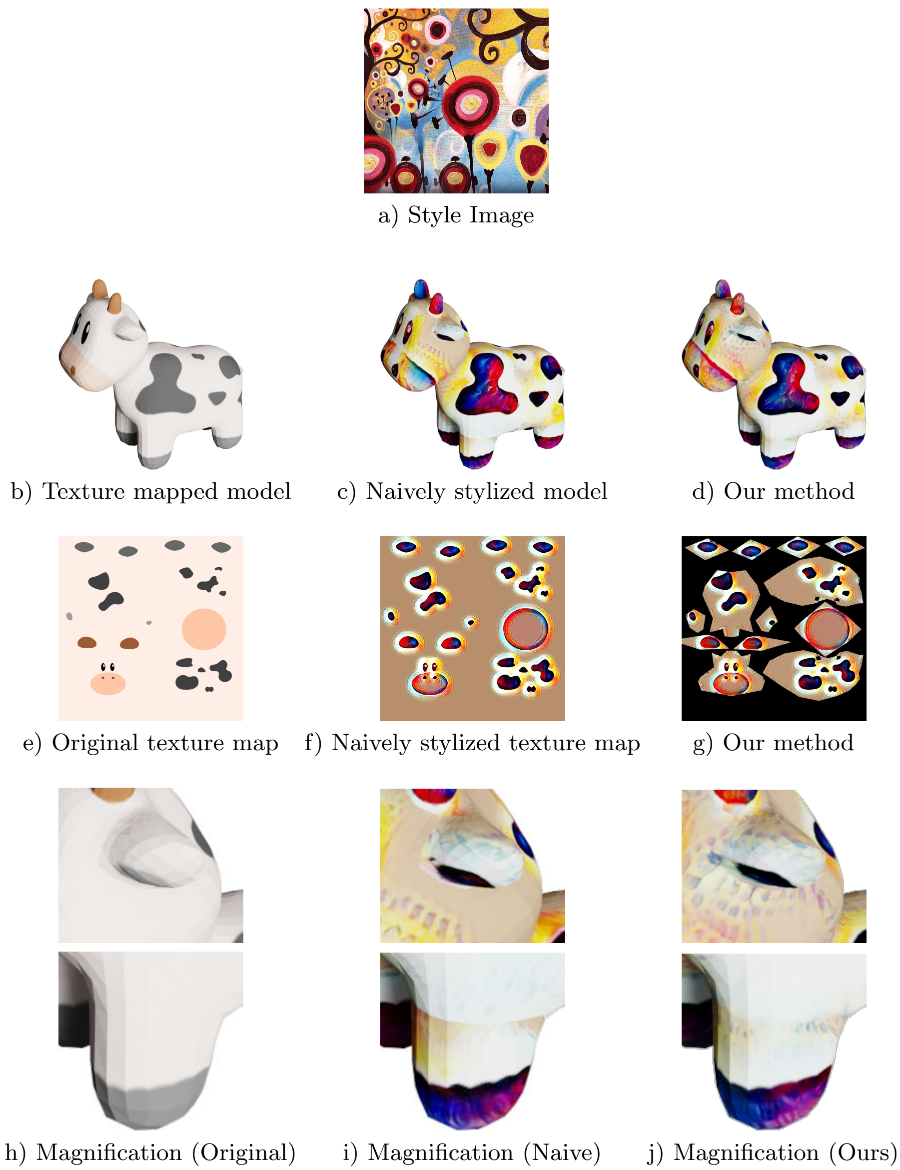


Fig. 23. A given style image (a), the original 3D mesh (b) and its texture map (e), the result of naively stylizing the texture map (c,f), and the result of using our method (d,g). We provide magnifications of the results in two locations (h,i,j). Note the visible seams shown in the magnifications of the naive approach (i), whereas our method in (j) minimizes the visibility of those seams since the visible regions of the texture map can be interconnected in the graph-based approach.