SketchEdit: Mask-Free Local Image Manipulation with Partial Sketches Supplementary Material

Yu Zeng
Johns Hopkins University
yzeng22@jhu.edu

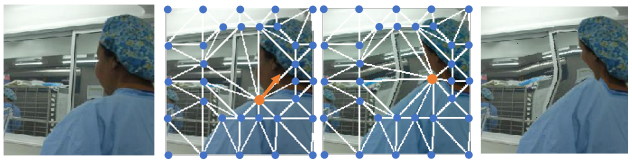
Zhe Lin
Adobe Research
zlin@adobe.com

Vishal M. Patel
Johns Hopkins University
vpatel36@jhu.edu

1. Warping Algorithm

For general images and background regions of face images, we randomly place and move the vertices within the modification area to create diverse distorted versions of the image. For the facial region of each face image, to make the warped result remain visually plausible, we use landmark-guided warping and blendshape-guided warping to simulate the local shape and expression variation of the face.

Random warping. As illustrated in Fig. 1, given an input image of size $h \times w$, we first sample a random modification area and place vertices evenly along the boundary of the image and the modification area. Then we place several control points at random locations inside the modification area (the orange point in Fig. 1 (b)) and construct a triangular mesh using the Delaunay triangulation method. To warp the modification area, we randomly move the control points by a small amount $(\Delta x, \Delta y)$ drawn uniformly from $U(0, \frac{w+h}{2})$[1] and apply the affine transformation on each triangle associated the new location. For simplification, we use rectangular modification areas with random sampled width and height no greater than $\frac{3}{4}$ the image size in our implementation. Still, note that the random warping also works with modification areas of arbitrary convex shapes.



(a) (b) (c) (d)

Figure 1. Illustration of triangular warping. The figure shows only one control point to avoid clutter. In actual implementation, there could be multiple control points.

Landmark-guided warping is similar to the random warping described above. The difference is that the modification areas are randomly sampled facial components (eyes, mouth, part of the contour *etc*.), and control points are the landmarks of the facial components. The landmarks are obtained using a landmark detection method [3]. We use random affine transformations to compute the movement of control points rather than independent random translation on each control point as in random warping.

Blendshape-guided warping. The 2D landmark-guided warping can create diverse shape variations of faces, however, the synthesized samples are not realistic enough and can not simulate complex expression changes. Therefore, we also synthesize samples by warping guided by blendshapes. For each face image, we fit a face model with blendshapes [2] to get the 3D face mesh and the blendshape weights, of which each dimension represents the magnitude of the movement of one face action unit. Then the warping flow for creating the motion of a facial component can be generated by manipulating the corresponding weight. Given a randomly sampled blendshape index, we set the corresponding weight to a random value in $[1, 3]$.

[1]The new location is clipped to avoid control points going out of the modification area.

2. More Analysis

2.1. Mask Estimation Accuracy

Our method predicts the masks of the modification region to save users' effort and achieve "mask-free" editing. The predicted masks capture users' intent in most cases. Fig. 2 shows the predicted masks compared to user-drawn masks. It can be seen that the predicted masks are close to the user-drawn ground-truth masks. To evaluate the quality of predicted masks more accurately, we binarize the predicted masks by thresholding at 0.5 and measure the precision, recall and MIOU with the user-drawn masks as ground-truth. Table 1 reports the quantitative evaluation results.

Table 1. Precision, recall and MIOU of the predicted mask with the user-drawn masks as ground-truth.

Dataset	Precision	Recall	MIOU
SketchFace	0.7515	0.8026	0.6118
SketchImg	0.6721	0.7193	0.5099

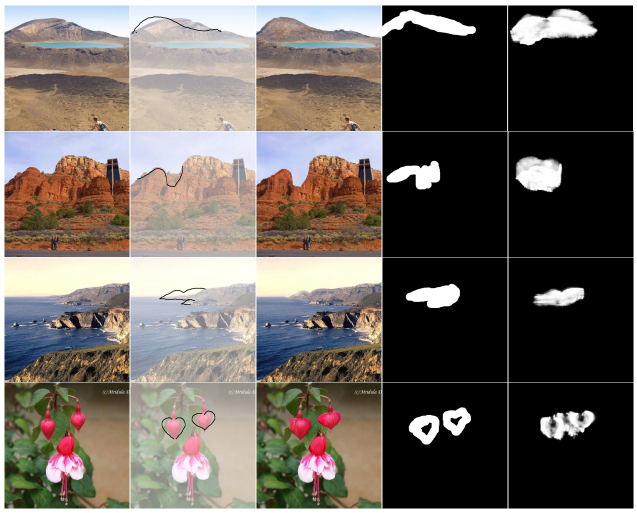


Figure 2. Visual comparison of the ground-truth masks and the user-drawn ground-truth masks.

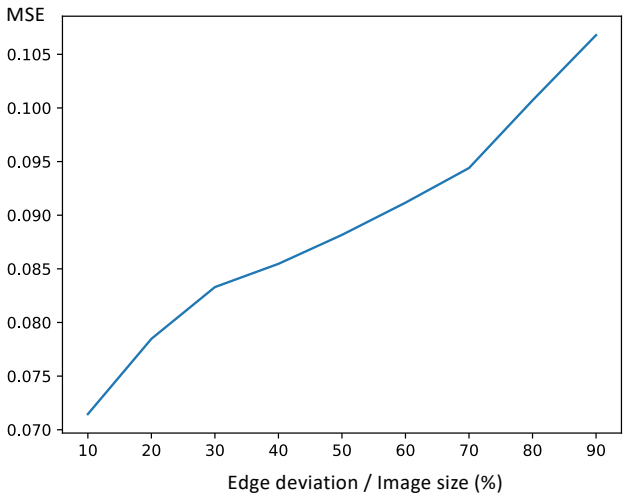


Figure 3. Mean square error of the edge maps with different deviation rate.

Figure 4. Failure cases. Top: images and sketches. Bottom: results.

2.2. Working boundaries and Failure Cases

The working boundaries depend on the degree of warping used in training. As described in Sec. 1, the largest range of modification and deviation used in training is 75% of image size. For local manipulation, this is adequately large in most cases. To find the extent to which the edits can deviate from the original edges, we synthesize a sequence of datasets $\{\mathcal{D}_n\}_{n=1}^{1}0$ with different deviation rates. Each dataset $\mathcal{D}_n$ contains 1,000 256×256 images and surrogate sketches created by moving the original edges by a distance of $\frac{n}{10} \times 256$, *i.e.* the deviation rate of the dataset $\mathcal{D}_n$ is $\frac{n}{10}$. Then we obtain the edge maps of the edited images and measure the mean squared error (MSE) to the surrogate sketch maps as a reflection of editing quality. Fig. 3 shows the MSE on the datasets of different deviation rates. It can be seen that the editing quality decreases drastically after the deviation rate grows over 70%, which is roughly the amount of edge deviation in training data.

Fig. 4 shows example failure cases. As demonstrated by the first two examples, when the sketch deviates too much from the original edge, the model fails to localize the manipulation regions and ends up adding new structures. Another limitation is that the model sometimes fails to generate structures that are necessary but not indicated in sketch maps. For instance, in the third example of Fig. 4, a more reasonable solution is to move the jaw accordingly when opening the mouth, however, the model does not change the jaw region as it is not indicated by the sketch. In the fourth example, the model erases the eyes and eyebrows as the sketch map only contains the eyeglasses.

3. More Results

Fig. 5 and Fig. 6 shows more results of the proposed methods.

4. Network Architecture

Table 2, Table 3 and Table 4 report the details of the network architecture of the mask estimator, style encoder and the generator. Data, more results, videos, an interactive demo and code can be found at https://zengxianyu.github.io/sketchedit

5. License Information for Images

All face images used in this paper are from the public FFHQ dataset [1]. The license information of the face images are as follow,

Main paper's:

Figure 1

Photos in the second row are from Places2 dataset

Figure 5. Face manipulation results. Images: input images, Sketches: sketches drawn by users, Results: manipulation results produced by the proposed method.

Table 2. Architecture of the mask estimator.

Channel number	Kernel size	Stride	Dilation rate	Activation
48	5	1	1	ELU
96	3	2	1	ELU
96	3	1	1	ELU
192	3	2	1	ELU
192	3	1	1	ELU
192	3	1	1	ELU
192	3	1	2	ELU
192	3	1	4	ELU
192	3	1	8	ELU
192	3	1	16	ELU
192	3	1	1	ELU
192	3	1	1	ELU
2× nearest neighbor upsample				
96	1	1	1	
96	3	1	1	ELU
2× nearest neighbor upsample				
48	1	1	1	
24	3	1	1	ELU
1	3	1	1	Tanh

Table 3. Architecture of the style encoder.

Channel number	Kernel size	Stride	Dilation rate	Activation
48	5	1	1	ELU
96	3	2	1	ELU
96	3	1	1	ELU
192	3	2	1	ELU
192	3	1	1	ELU
192	3	1	1	ELU
192	3	1	2	ELU
192	3	1	4	ELU
192	3	1	8	ELU
192	3	1	16	ELU
Pooling				

Table 4. Architecture of the generator.

Channel number	Kernel size	Stride	Dilation rate	Activation
48	5	1	1	ELU
96	3	2	1	ELU
96	3	1	1	ELU
192	3	2	1	ELU
192	3	1	1	ELU
192	3	1	1	ELU
192	3	1	2	ELU
192	3	1	4	ELU
192	3	1	8	ELU
192	3	1	16	ELU
192	3	1	1	ELU
192	3	1	1	ELU
2× nearest neighbor upsample				
96	1	1	1	
96	3	1	1	ELU
2× nearest neighbor upsample				
48	1	1	1	
24	3	1	1	ELU
3	3	1	1	Tanh

Branch 1					Branch 2				
C. n.	K. s.	S.	D. r.	A.	C. n.	K. s.	S.	D. r.	A.
48	5	1	1	ELU	48	5	1	1	ELU
48	3	2	1	ELU	48	3	2	1	ELU
96	3	1	1	ELU	96	3	1	1	ELU
96	3	2	1	ELU	192	3	2	1	ELU
192	3	1	1	ELU	192	3	1	1	ELU
192	3	1	1	ELU	192	3	1	1	ReLU
192	3	1	2	ELU	Contextual Attention				
192	3	1	4	ELU					
192	3	1	8	ELU	192	3	1	1	ELU
192	3	1	16	ELU	192	3	1	1	ELU
Concatenate									

Channel number	Kernel size	Stride	Dilation rate	Activation
192	3	1	1	ELU
192	3	1	1	ELU
2× nearest neighbor upsample				
96	1	1	1	
96	3	1	1	ELU
2× nearest neighbor upsample				
48	1	1	1	
24	3	1	1	ELU
3	3	1	1	Tanh

16066: Face photo by Ed Hunsinger / CC BY-NC 2.0
20748: Face photo by Paul O' Russa / CC BY-NC 2.0
Figure 2, Figure 3, Figure 4 and Figure 8
Photos are from from Places2 dataset.
Figure 5
The first two photos are from Places2 dataset
00141: Face photo by stephen davis / CC BY 2.0
42690: Face photo by Andrew Choy / CC BY 2.0
Figure 6
The first photo is from Places2 dataset
The second photo is by Luke Stackpoole on Unsplash

Figure 7
06550: Face photo by Ford DSFL / CC BY 2.0
01102: Face photo by ajot / CC BY-NC 2.0
00367: Face photo by Ministerio da Cultura / CC BY 2.0
26948: Face photo by Rutgers Nursing / CC BY-NC 2.0
Figure 9

09638: Face photo by fossilmike / CC BY-NC 2.0
18100: Face photo by Danish Model United Nations / CC BY 2.0
Figure 10
02744: Face photo by Rebecca Rowlands / CC BY 2.0
22804: Face photo by hangmin1 / CC-Mark 1.0
Supplementary Material's:
Figure 1, Figure 2, and Figure 6
Photos are from from Places2 dataset.
Figure 4

06550: Face photo by Ford DSFL / CC BY 2.0

Figure 5
09624: Face photo by Ordiziako Jakintza Ikastola / CC BY 2.0
18563: Face photo by Ben Sutherland / CC BY 2.0
23714: Face photo by Rose Lamela / CC BY 2.0
23283: Face photo by dave.neeley56 / CC BY-NC 2.0
17654: Face photo by RussellReno / CC BY-NC 2.0
10077: Face photo by Minh-Kiet Callies / CC BY-NC 2.0
00560: Face photo by Owen Lucas / CC-Mark 1.0
10378: Face photo by Montclair Film / CC BY 2.0
01535: Face photo by Steve Cook / CC BY-NC 2.0
09439: Face photo by Betsy Weber / CC BY 2.0
05085: Face photo by Raj Taneja / CC BY-NC 2.0
25425: Face photo by Benjamin Harrison / CC BY-NC 2.0
24710: Face photo by enderMC / CC BY-NC 2.0
20379: Face photo by Luciana Santos / CC BY-NC 2.0

References

[1] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[2] Tianye Li, Timo Bolkart, Michael J Black, Hao Li, and Javier Romero. Learning a model of facial shape and expression from 4d scans. *ACM Transactions on Graphics*, 36(6):194–1, 2017.

[3] Xinyao Wang, Liefeng Bo, and Li Fuxin. Adaptive wing loss for robust face alignment via heatmap regression. In *International Conference on Computer Vision*, 2019.

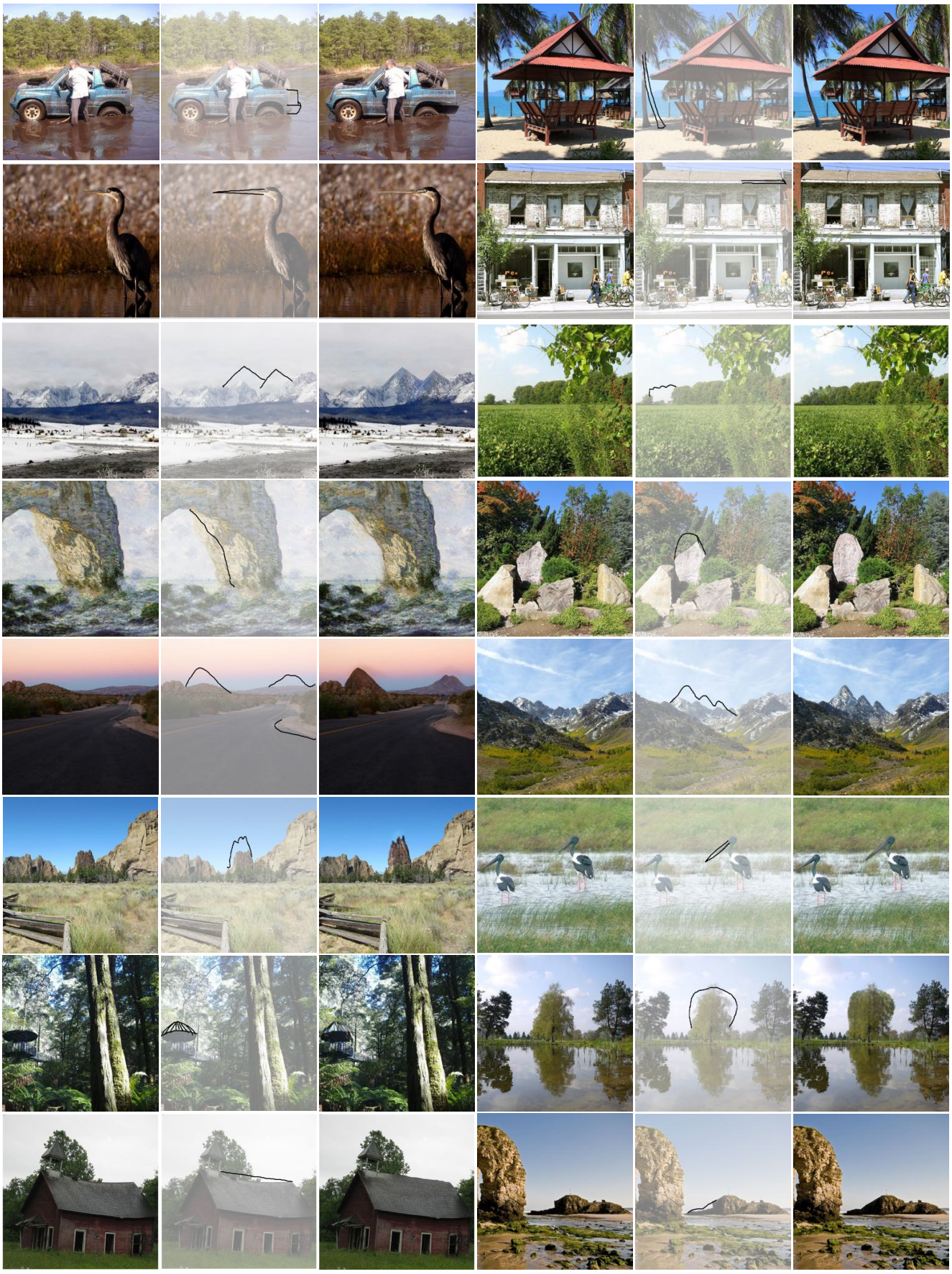


Figure 6. General image manipulation results. Images: input images, Sketches: sketches drawn by users, Results: manipulation results produced by the proposed method.