SmartShadow: Artistic Shadow Drawing Tool for Line Drawings

Anonymous Author(s)

Supplementary Material

We present additional results and comparisons, additional details of the user study and dataset, as well as implementation details in this document.

Contents

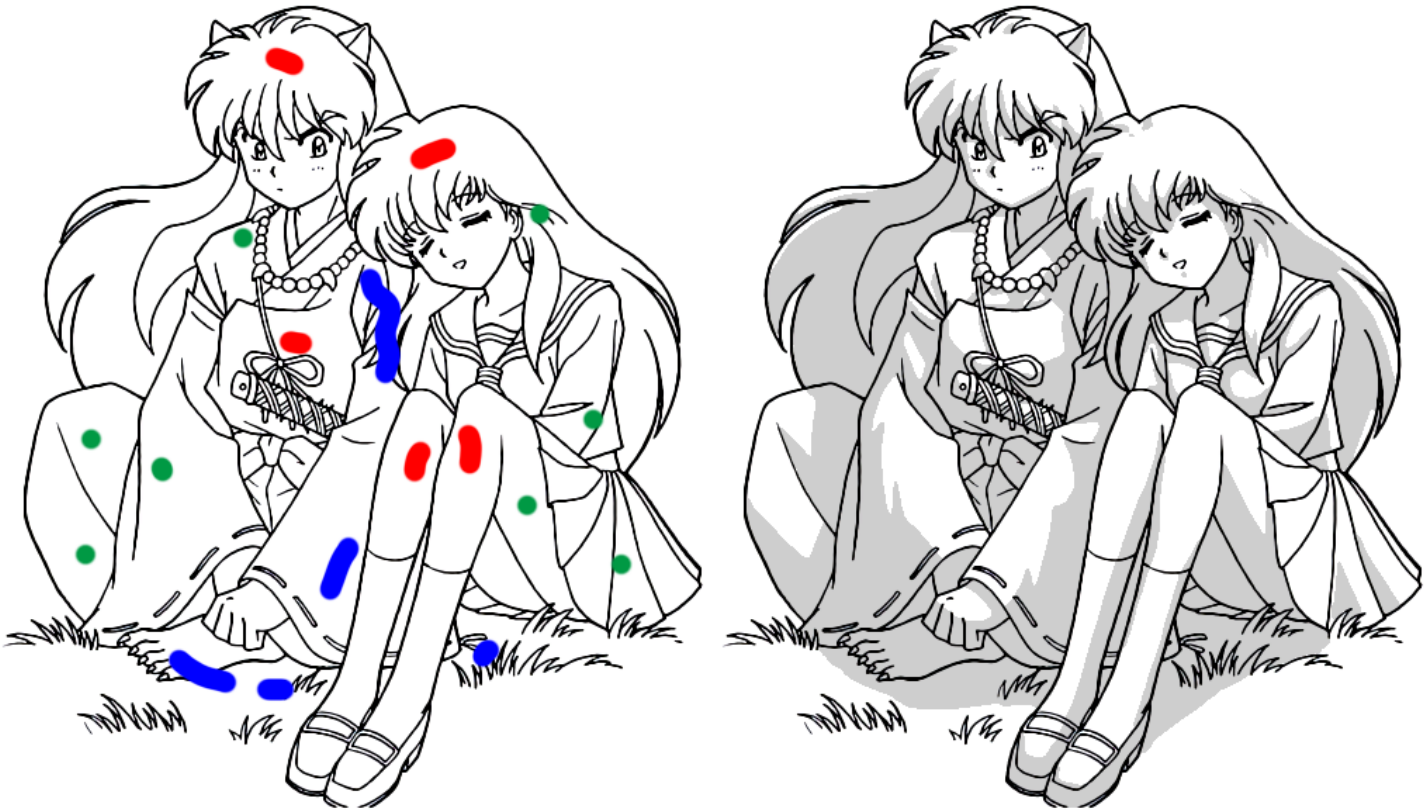


Figure 1: **Additional qualitative result #1.** We present additional qualitative results of our approach. On the left is the line drawing and the user inputs, and the result is on the right. The user scribbles are precisely one-pixel width and we dilated the scribbles for clearer presentation. *Artworks used with artist permissions.*

1 Additional results

We present 30 additional results (20 pure qualitative results and 10 results with time data) and 15 visual comparisons in this document.

1.1 Additional qualitative results

We present 20 qualitative results from Fig. 1 to Fig. 20.

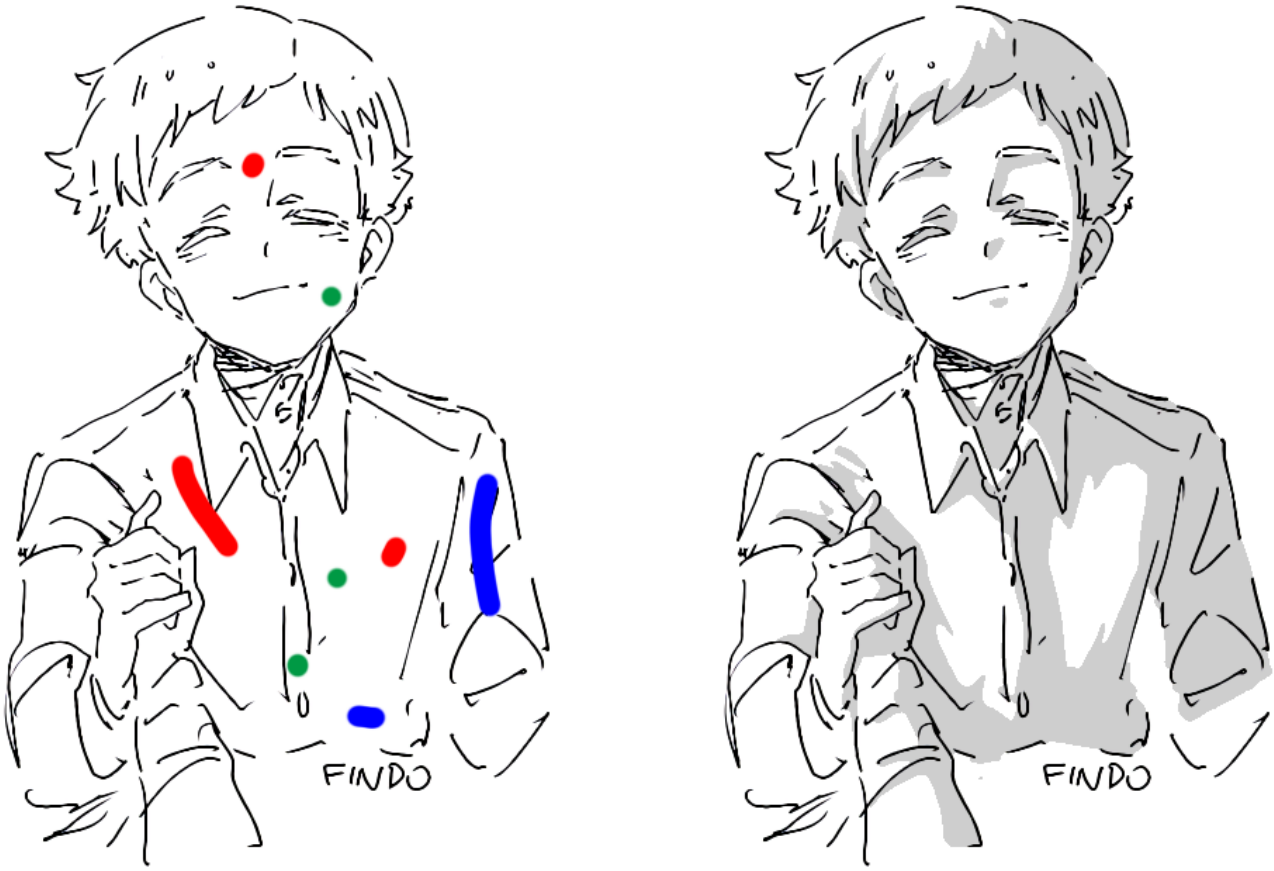


Figure 2: **Additional qualitative result #2.** We present additional qualitative results of our approach. On the left is the line drawing and the user inputs, and the result is on the right. The user scribbles are precisely one-pixel width and we dilated the scribbles for clearer presentation. *Artworks used with artist permissions.*

Figure 3: **Additional qualitative result #3.** We present additional qualitative results of our approach. On the left is the line drawing and the user inputs, and the result is on the right. The user scribbles are precisely one-pixel width and we dilated the scribbles for clearer presentation. *Artworks used with artist permissions.*

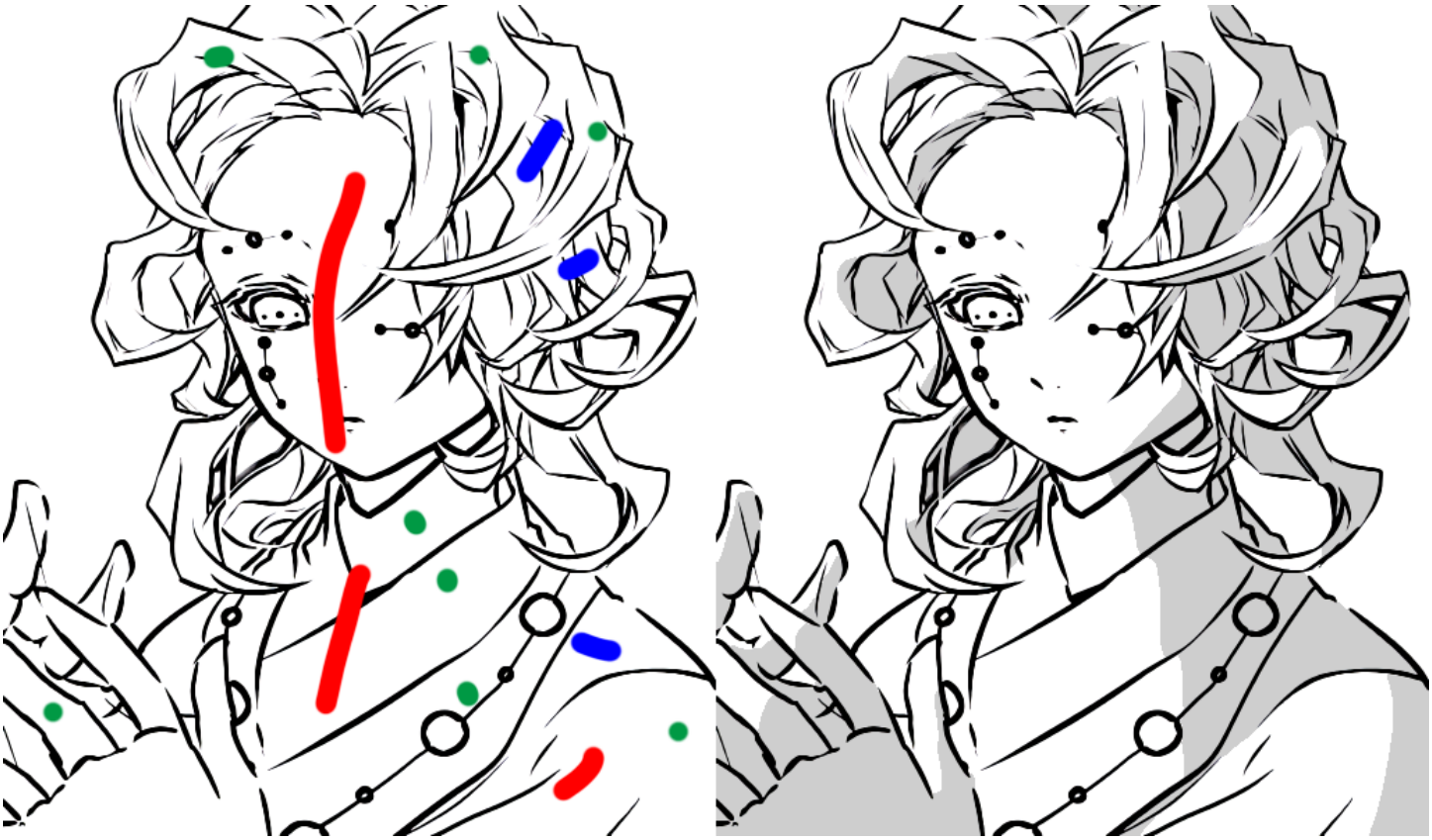


Figure 4: **Additional qualitative result #4.** We present additional qualitative results of our approach. On the left is the line drawing and the user inputs, and the result is on the right. The user scribbles are precisely one-pixel width and we dilated the scribbles for clearer presentation. *Artworks used with artist permissions.*

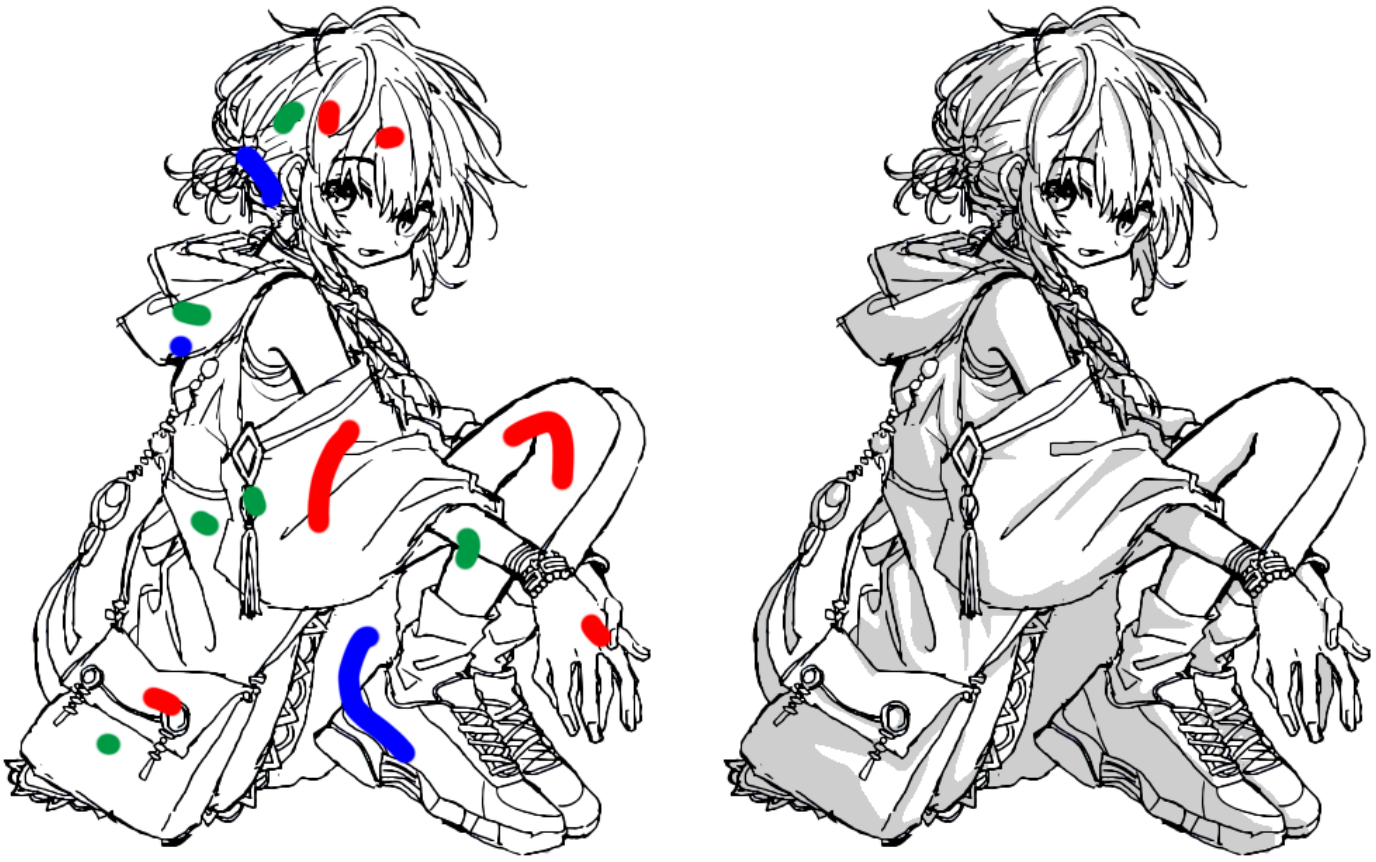


Figure 5: **Additional qualitative result #5.** We present additional qualitative results of our approach. On the left is the line drawing and the user inputs, and the result is on the right. The user scribbles are precisely one-pixel width and we dilated the scribbles for clearer presentation. *Artworks used with artist permissions.*

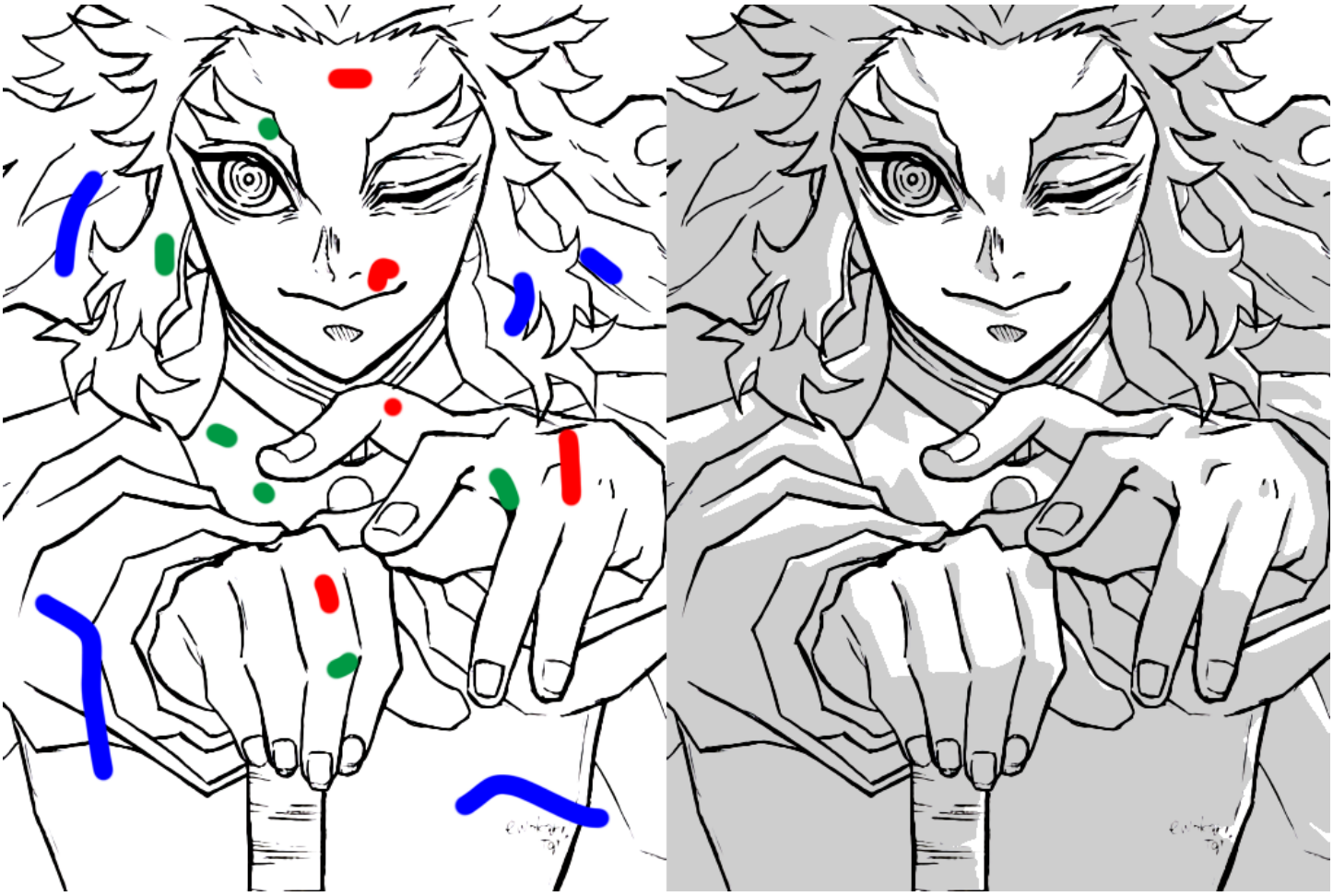


Figure 6: **Additional qualitative result #6.** We present additional qualitative results of our approach. On the left is the line drawing and the user inputs, and the result is on the right. The user scribbles are precisely one-pixel width and we dilated the scribbles for clearer presentation. *Artworks used with artist permissions.*

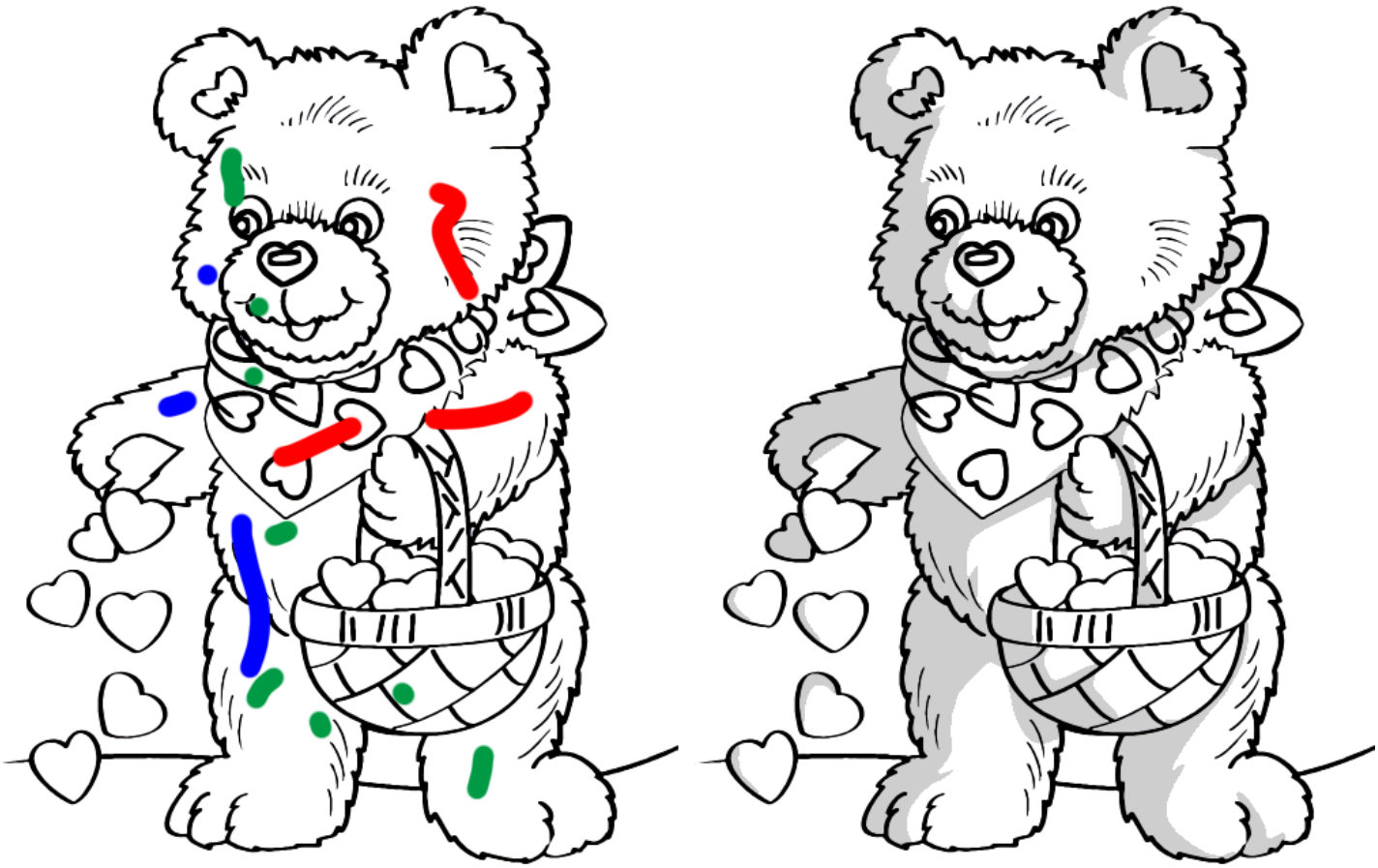


Figure 7: **Additional qualitative result #7.** We present additional qualitative results of our approach. On the left is the line drawing and the user inputs, and the result is on the right. The user scribbles are precisely one-pixel width and we dilated the scribbles for clearer presentation. *Artworks used with artist permissions.*

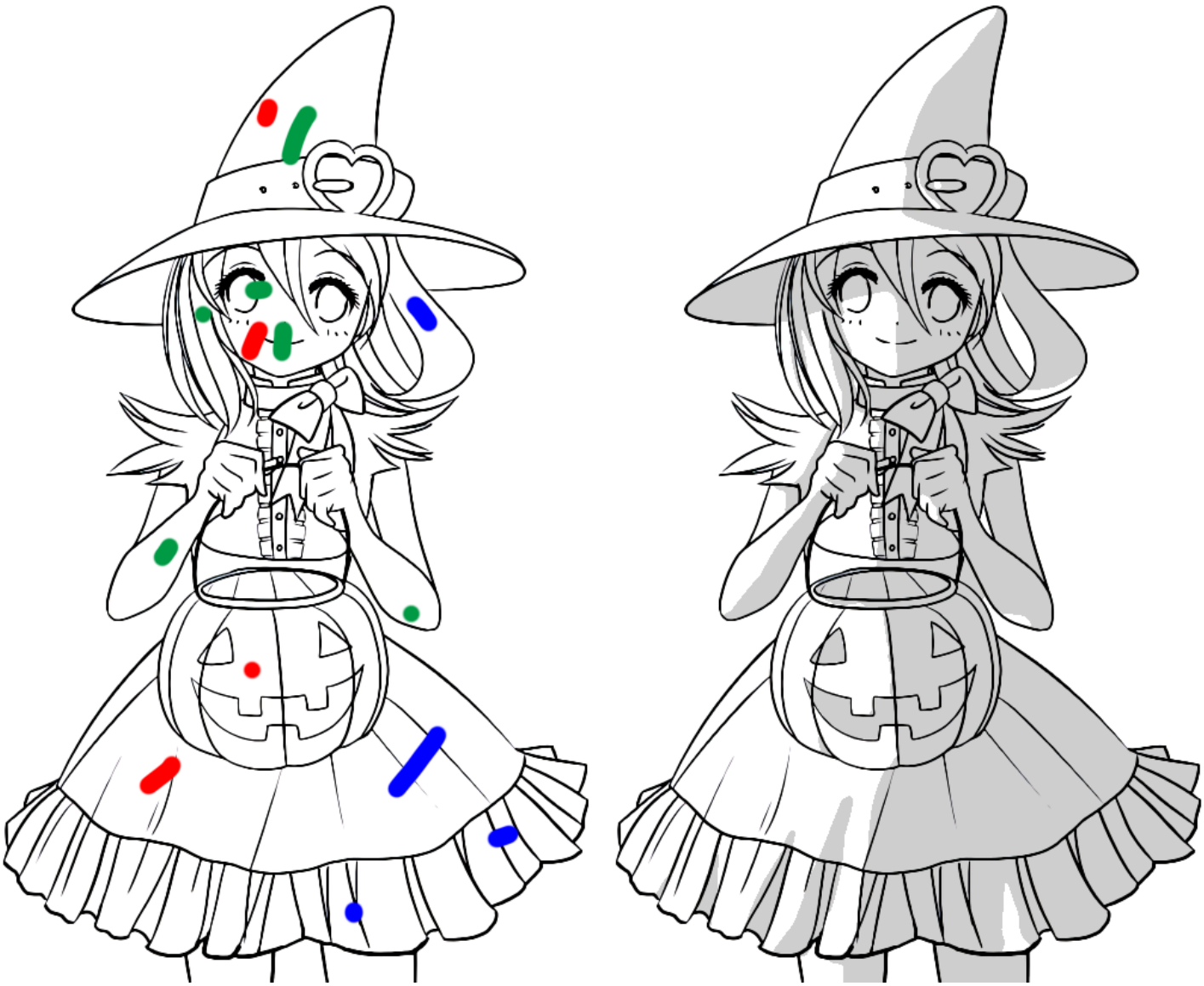


Figure 8: **Additional qualitative result #8.** We present additional qualitative results of our approach. On the left is the line drawing and the user inputs, and the result is on the right. The user scribbles are precisely one-pixel width and we dilated the scribbles for clearer presentation. *Artworks used with artist permissions.*

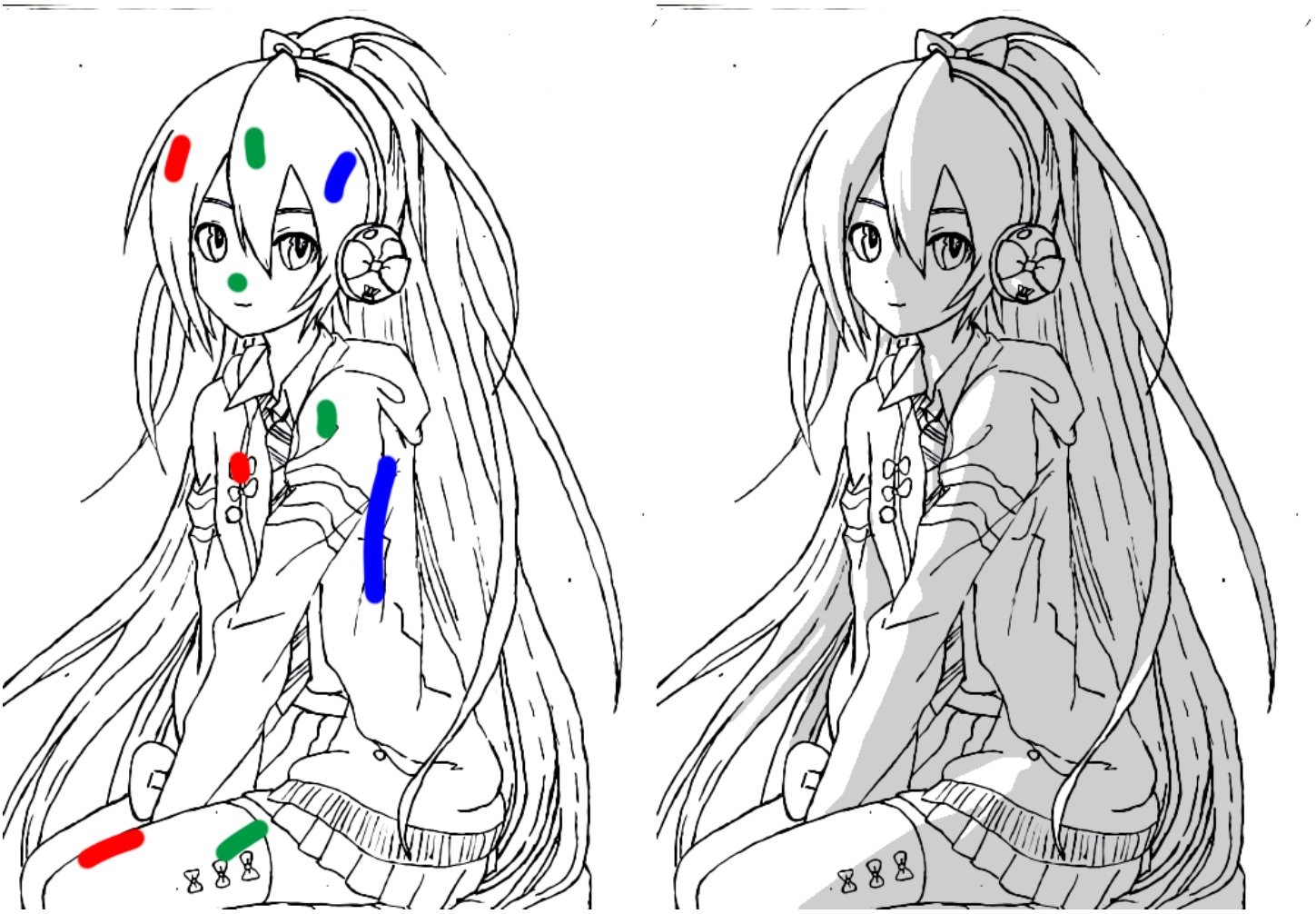


Figure 9: **Additional qualitative result #9.** We present additional qualitative results of our approach. On the left is the line drawing and the user inputs, and the result is on the right. The user scribbles are precisely one-pixel width and we dilated the scribbles for clearer presentation. *Artworks used with artist permissions.*

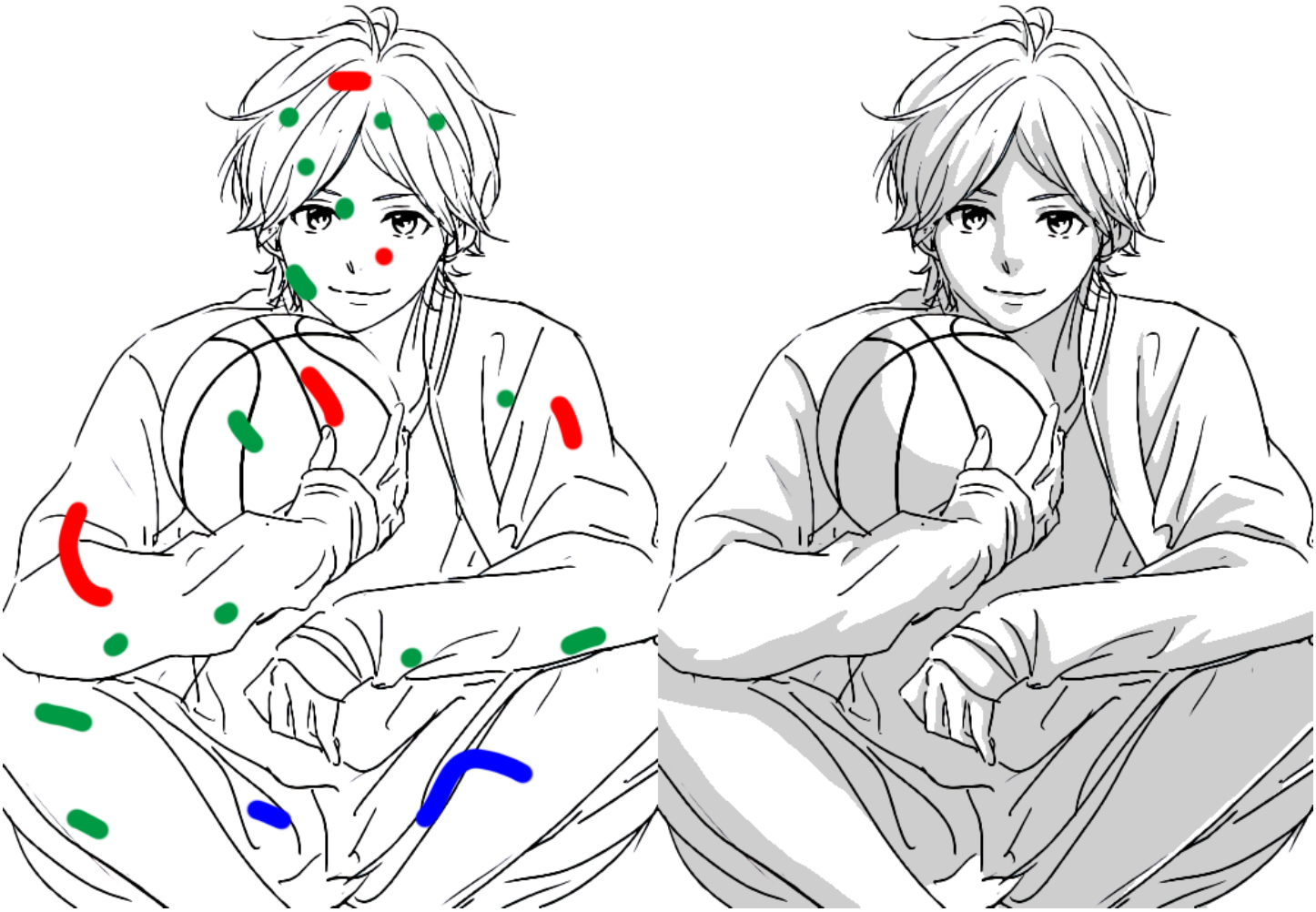


Figure 10: **Additional qualitative result #10.** We present additional qualitative results of our approach. On the left is the line drawing and the user inputs, and the result is on the right. The user scribbles are precisely one-pixel width and we dilated the scribbles for clearer presentation. *Artworks used with artist permissions.*

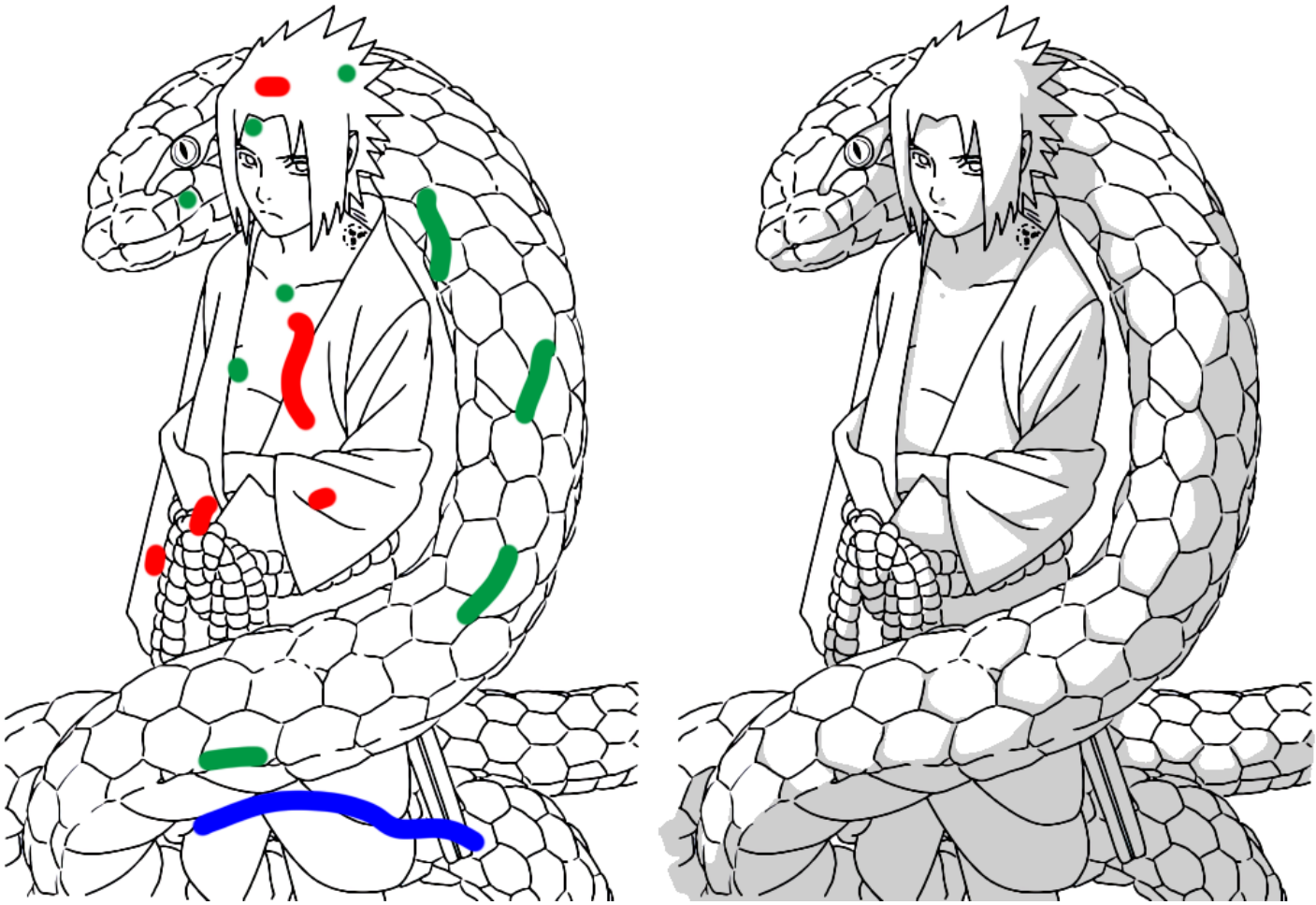


Figure 11: **Additional qualitative result #11.** We present additional qualitative results of our approach. On the left is the line drawing and the user inputs, and the result is on the right. The user scribbles are precisely one-pixel width and we dilated the scribbles for clearer presentation. *Artworks used with artist permissions.*

Figure 12: **Additional qualitative result #12.** We present additional qualitative results of our approach. On the left is the line drawing and the user inputs, and the result is on the right. The user scribbles are precisely one-pixel width and we dilated the scribbles for clearer presentation. *Artworks used with artist permissions.*

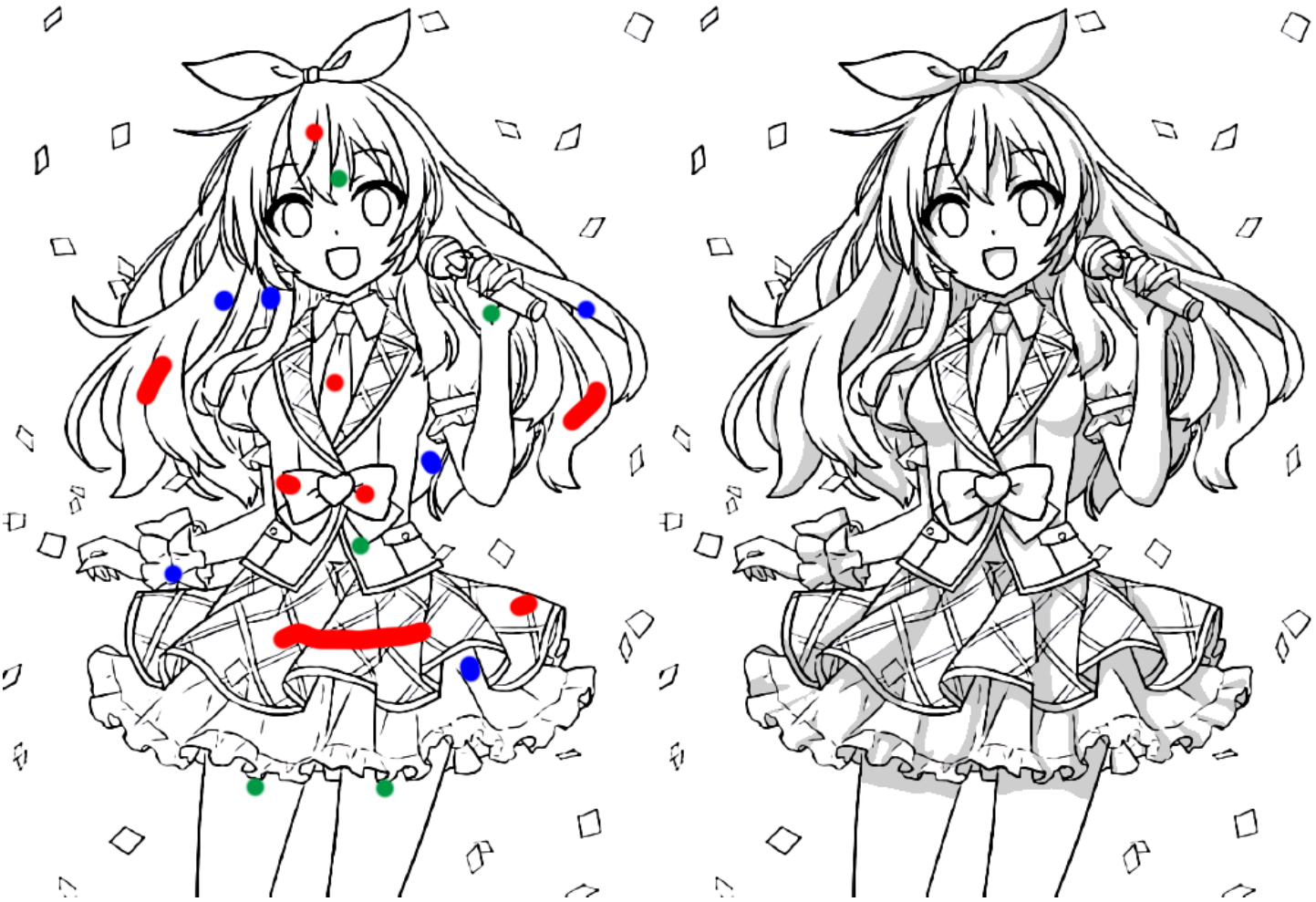


Figure 13: **Additional qualitative result #13.** We present additional qualitative results of our approach. On the left is the line drawing and the user inputs, and the result is on the right. The user scribbles are precisely one-pixel width and we dilated the scribbles for clearer presentation. *Artworks used with artist permissions.*

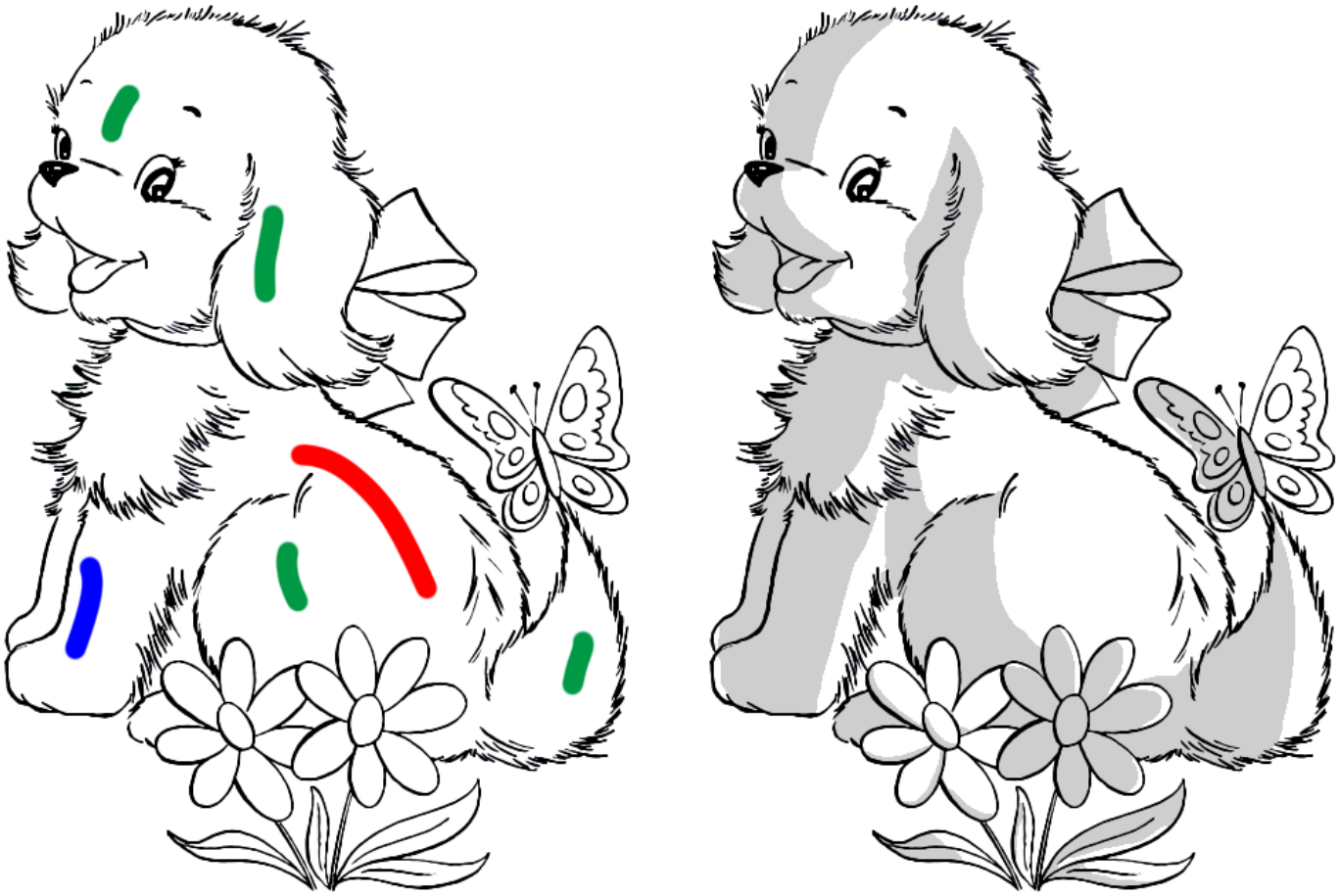


Figure 14: **Additional qualitative result #14.** We present additional qualitative results of our approach. On the left is the line drawing and the user inputs, and the result is on the right. The user scribbles are precisely one-pixel width and we dilated the scribbles for clearer presentation. *Artworks used with artist permissions.*

Figure 15: **Additional qualitative result #15.** We present additional qualitative results of our approach. On the left is the line drawing and the user inputs, and the result is on the right. The user scribbles are precisely one-pixel width and we dilated the scribbles for clearer presentation. *Artworks used with artist permissions.*

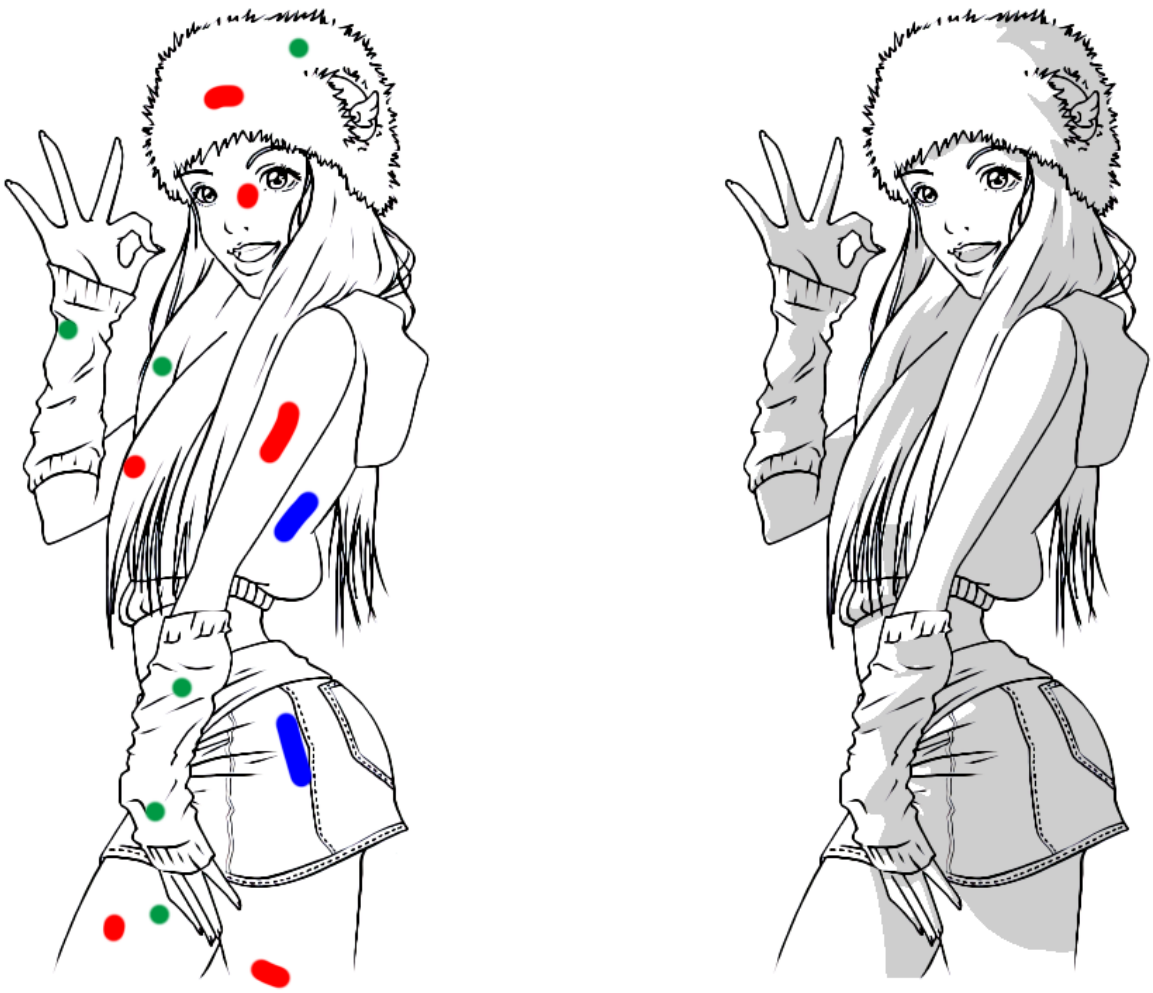


Figure 16: **Additional qualitative result #16.** We present additional qualitative results of our approach. On the left is the line drawing and the user inputs, and the result is on the right. The user scribbles are precisely one-pixel width and we dilated the scribbles for clearer presentation. *Artworks used with artist permissions.*

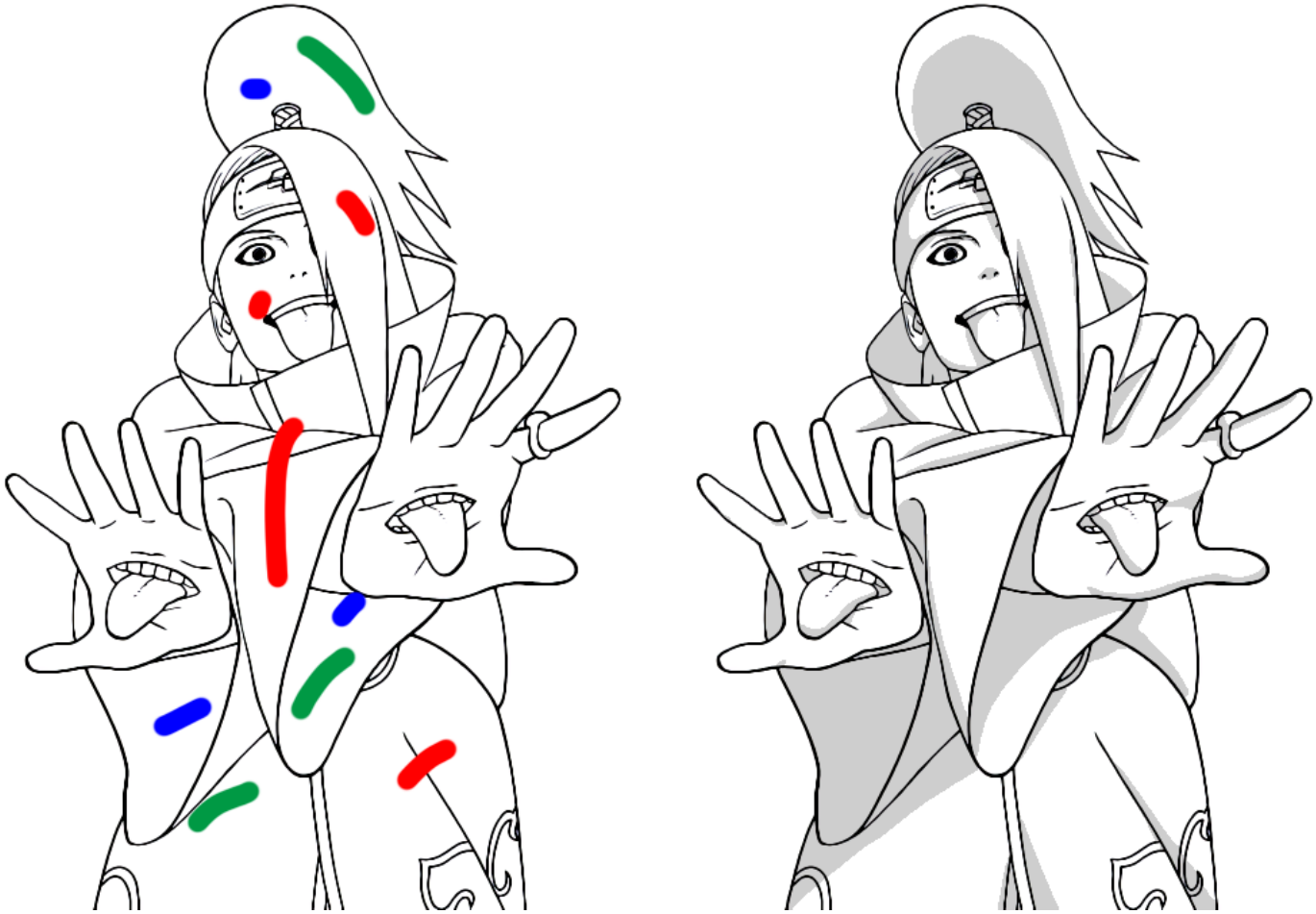


Figure 17: **Additional qualitative result #17.** We present additional qualitative results of our approach. On the left is the line drawing and the user inputs, and the result is on the right. The user scribbles are precisely one-pixel width and we dilated the scribbles for clearer presentation. *Artworks used with artist permissions.*

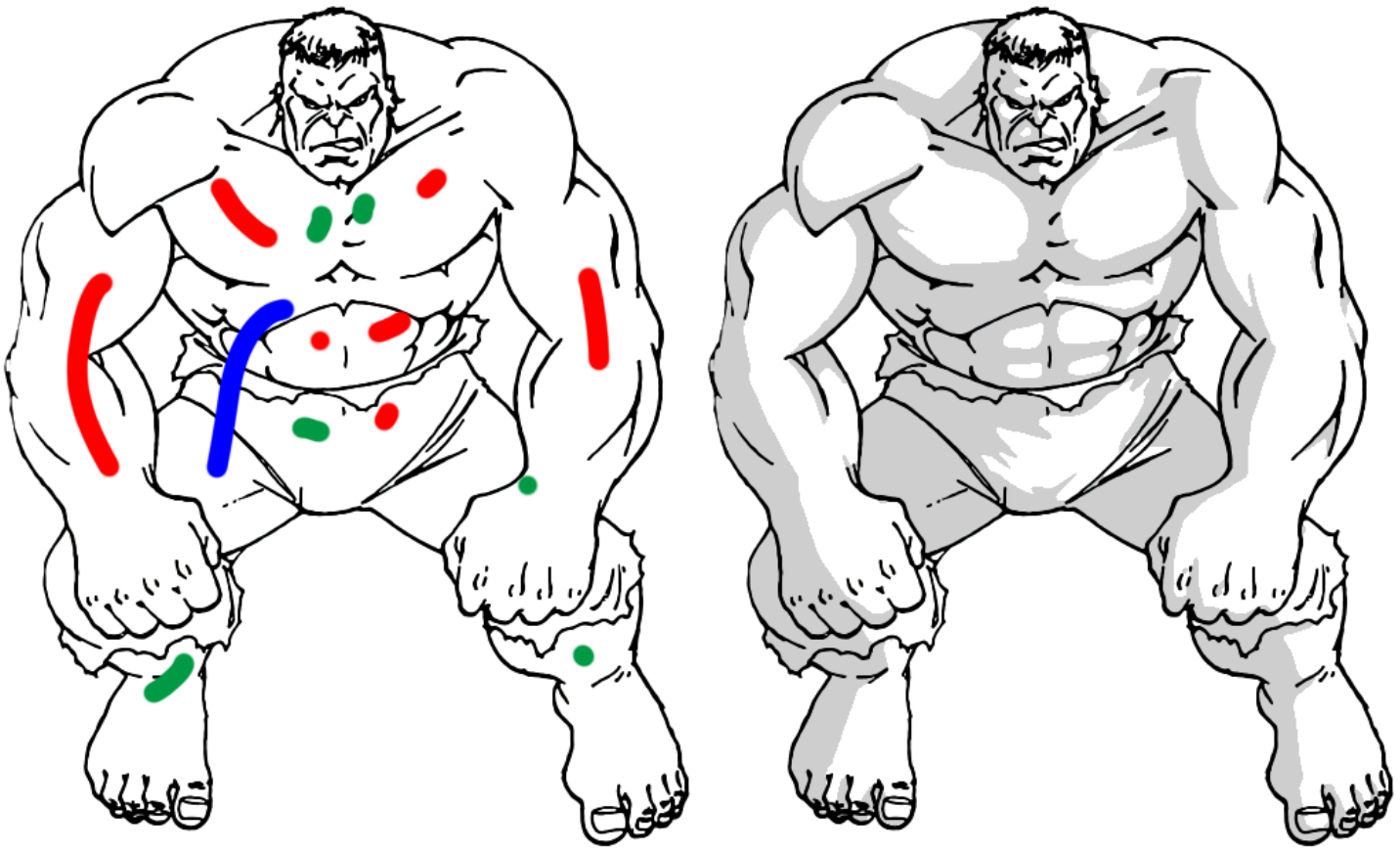


Figure 18: **Additional qualitative result #18.** We present additional qualitative results of our approach. On the left is the line drawing and the user inputs, and the result is on the right. The user scribbles are precisely one-pixel width and we dilated the scribbles for clearer presentation. *Artworks used with artist permissions.*

Figure 19: **Additional qualitative result #19.** We present additional qualitative results of our approach. On the left is the line drawing and the user inputs, and the result is on the right. The user scribbles are precisely one-pixel width and we dilated the scribbles for clearer presentation. *Artworks used with artist permissions.*

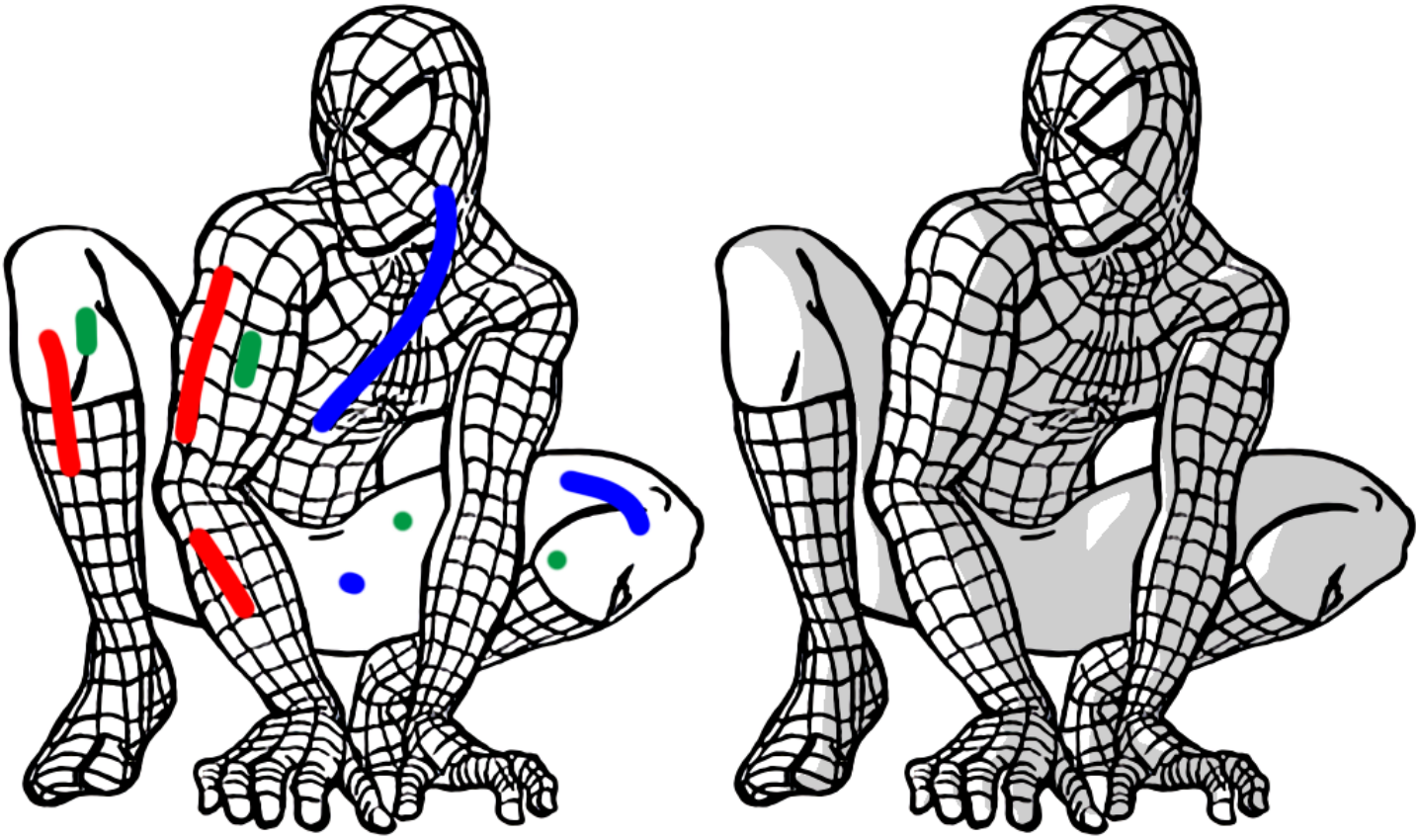


Figure 20: **Additional qualitative result #20.** We present additional qualitative results of our approach. On the left is the line drawing and the user inputs, and the result is on the right. The user scribbles are precisely one-pixel width and we dilated the scribbles for clearer presentation. *Artworks used with artist permissions.*

1.2 Additional time data and comparisons with commercial tool (Adobe PhotoShop)

We present 10 results with time data from Fig. 21 to Fig. 30, compared with the commercial tool Adobe PhotoShop.

User edit (ours) | Ours | Adobe PhotoShop (manual drawing)

2 minutes 13 seconds | **26 minutes 17 seconds**

11.86× speed up

Figure 21: **Comparison #1 with the Commercial Tool (Adobe PhotoShop).** We also present the time data below images. On the left is the line drawing and the user inputs, and the result is on the right. The user scribbles are precisely one-pixel width and we dilated the scribbles for clearer presentation. *Artworks used with artist permissions.*

User edit (ours) | Ours | Adobe PhotoShop (manual drawing)

51 seconds | **15 minutes 22 seconds**

18.07× speed up

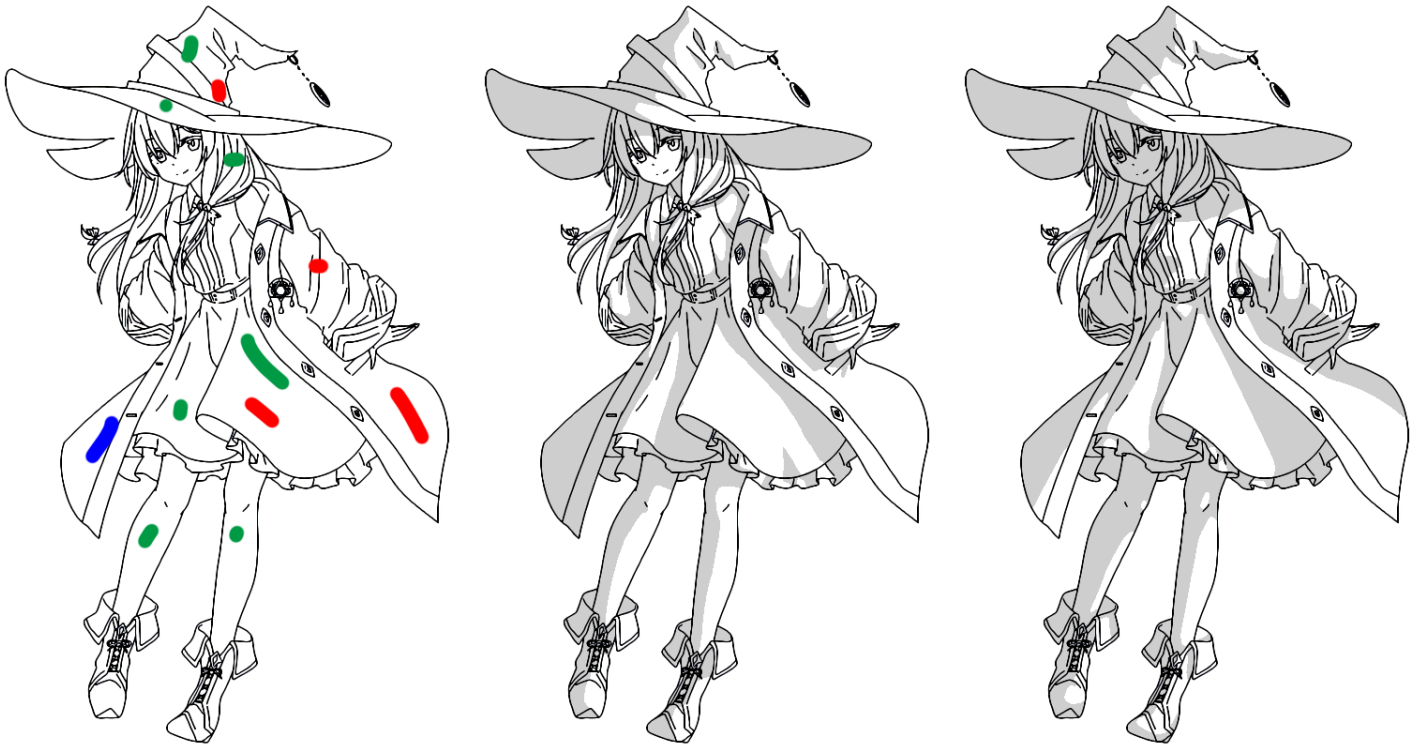
Figure 22: **Comparison #2 with the Commercial Tool (Adobe PhotoShop).** We also present the time data below images. On the left is the line drawing and the user inputs, and the result is on the right. The user scribbles are precisely one-pixel width and we dilated the scribbles for clearer presentation. *Artworks used with artist permissions.*

User edit (ours) | Ours | Adobe PhotoShop (manual drawing)

3 minutes 11 seconds | **27 minutes 34 seconds**

8.65× speed up

Figure 23: **Comparison #3 with the Commercial Tool (Adobe PhotoShop).** We also present the time data below images. On the left is the line drawing and the user inputs, and the result is on the right. The user scribbles are precisely one-pixel width and we dilated the scribbles for clearer presentation. *Artworks used with artist permissions.*



User edit (ours)

Ours

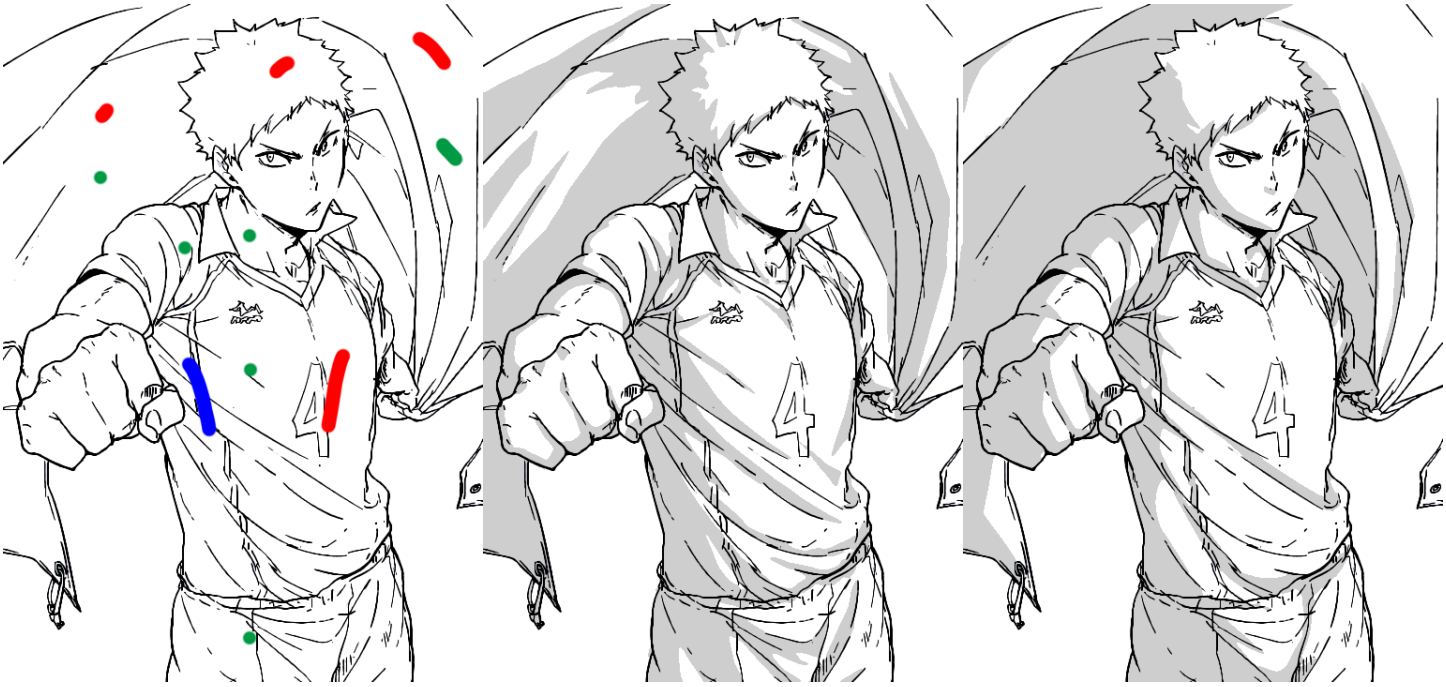
1 minutes 59 seconds

Adobe PhotoShop (manual drawing)

23 minutes 13 seconds

11.70× speed up

Figure 24: **Comparison #4 with the Commercial Tool (Adobe PhotoShop).** We also present the time data below images. On the left is the line drawing and the user inputs, and the result is on the right. The user scribbles are precisely one-pixel width and we dilated the scribbles for clearer presentation. *Artworks used with artist permissions.*

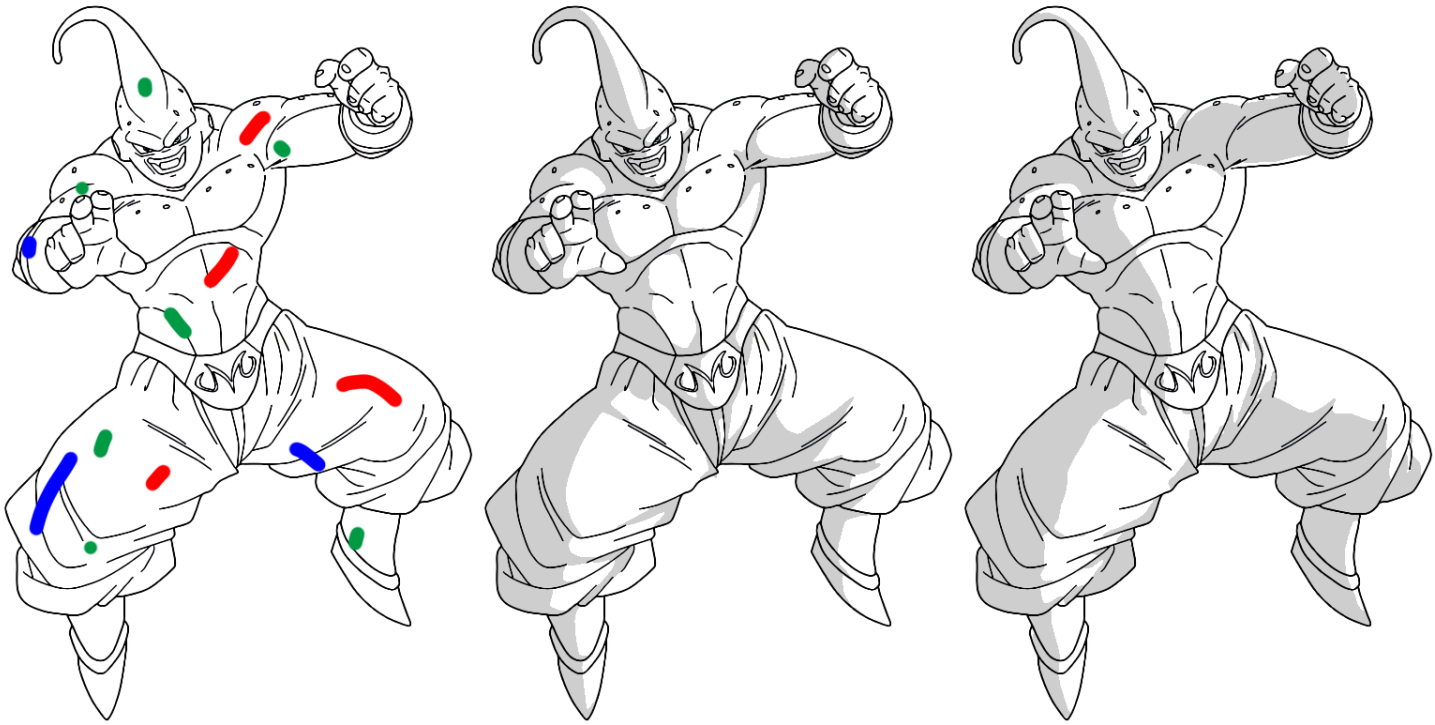


User edit (ours) Ours Adobe PhotoShop (manual drawing)

5 minutes 17 seconds **17 minutes 25 seconds**

3.29× speed up

Figure 25: **Comparison #5 with the Commercial Tool (Adobe PhotoShop).** We also present the time data below images. On the left is the line drawing and the user inputs, and the result is on the right. The user scribbles are precisely one-pixel width and we dilated the scribbles for clearer presentation. *Artworks used with artist permissions.*

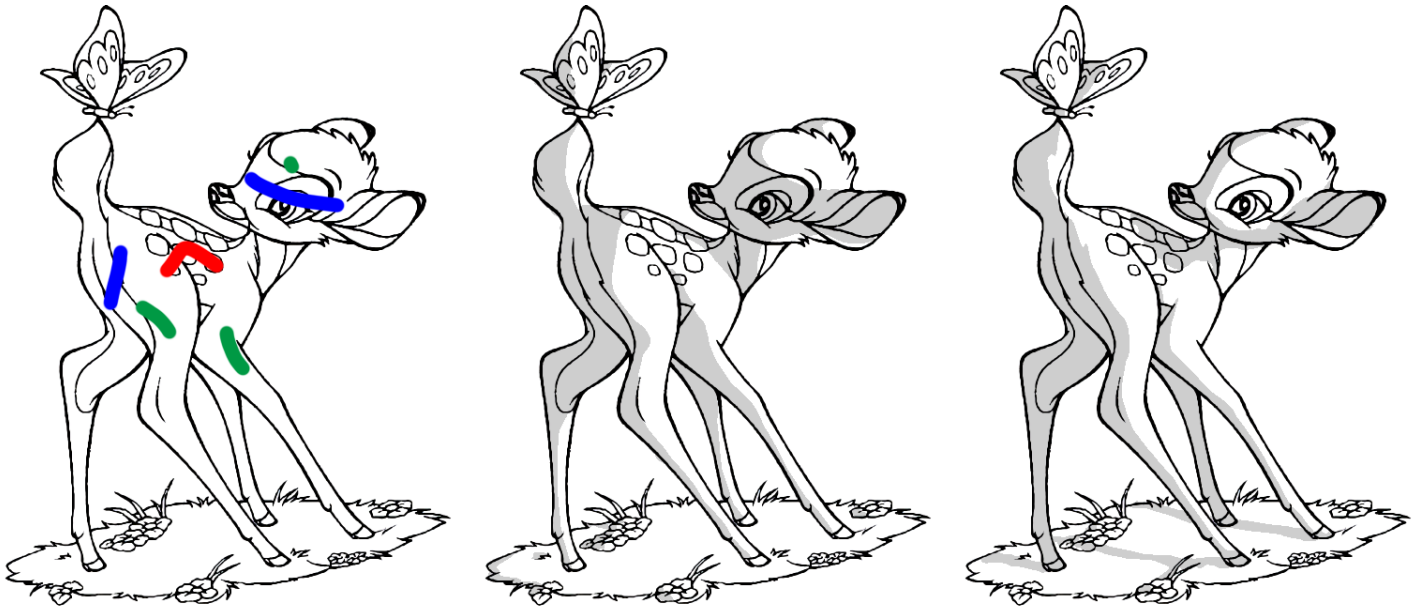


User edit (ours) | Ours | Adobe PhotoShop (manual drawing)

1 minutes 25 seconds | **11 minutes 14 seconds**

7.92× speed up

Figure 26: **Comparison #6 with the Commercial Tool (Adobe PhotoShop).** We also present the time data below images. On the left is the line drawing and the user inputs, and the result is on the right. The user scribbles are precisely one-pixel width and we dilated the scribbles for clearer presentation. *Artworks used with artist permissions.*



User edit (ours) | Ours | Adobe PhotoShop (manual drawing)

1 minutes 55 seconds | **17 minutes 13 seconds**

8.98× speed up

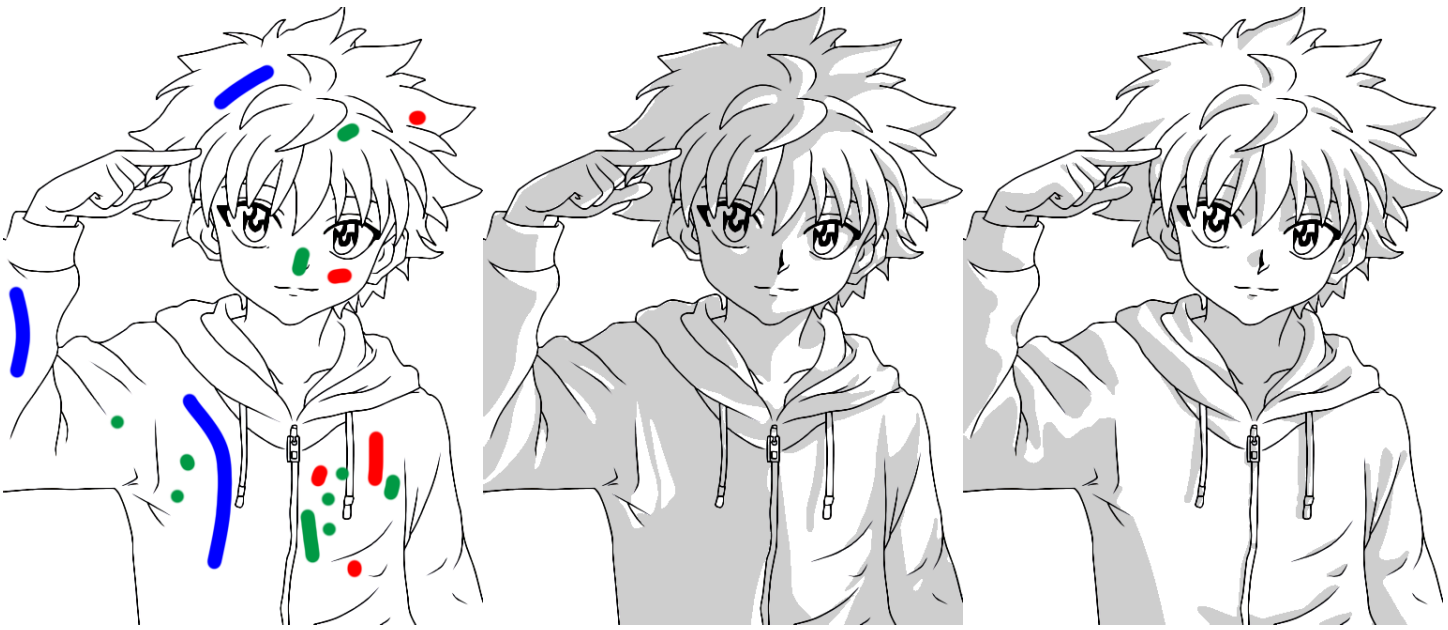
Figure 27: **Comparison #7 with the Commercial Tool (Adobe PhotoShop).** We also present the time data below images. On the left is the line drawing and the user inputs, and the result is on the right. The user scribbles are precisely one-pixel width and we dilated the scribbles for clearer presentation. *Artworks used with artist permissions.*

User edit (ours) | Ours | Adobe PhotoShop (manual drawing)

1 minutes 31 seconds | **8 minutes 6 seconds**

5.34× speed up

Figure 28: **Comparison #8 with the Commercial Tool (Adobe PhotoShop).** We also present the time data below images. On the left is the line drawing and the user inputs, and the result is on the right. The user scribbles are precisely one-pixel width and we dilated the scribbles for clearer presentation. *Artworks used with artist permissions.*

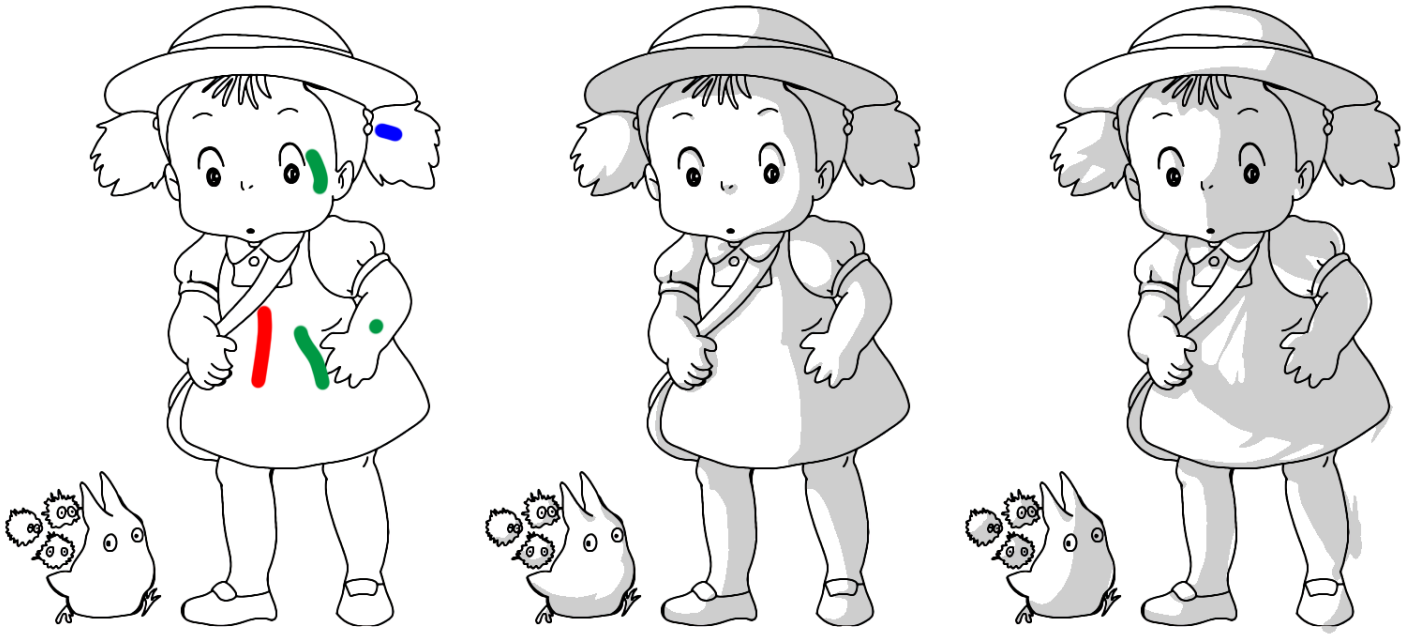


User edit (ours) | Ours | Adobe PhotoShop (manual drawing)

2 minutes 15 seconds | **23 minutes 32 seconds**

10.45× speed up

Figure 29: **Comparison #9 with the Commercial Tool (Adobe PhotoShop).** We also present the time data below images. On the left is the line drawing and the user inputs, and the result is on the right. The user scribbles are precisely one-pixel width and we dilated the scribbles for clearer presentation. *Artworks used with artist permissions.*



User edit (ours) | Ours | Adobe PhotoShop (manual drawing)

1 minutes 1 seconds | **9 minutes 17 seconds**

9.13× speed up

Figure 30: **Comparison #10 with the Commercial Tool (Adobe PhotoShop).** We also present the time data below images. On the left is the line drawing and the user inputs, and the result is on the right. The user scribbles are precisely one-pixel width and we dilated the scribbles for clearer presentation. *Artworks used with artist permissions.*

1.3 Additional comparisons with other possible alternative methods

We present 15 visual comparisons from Fig. 31 to Fig. 45.

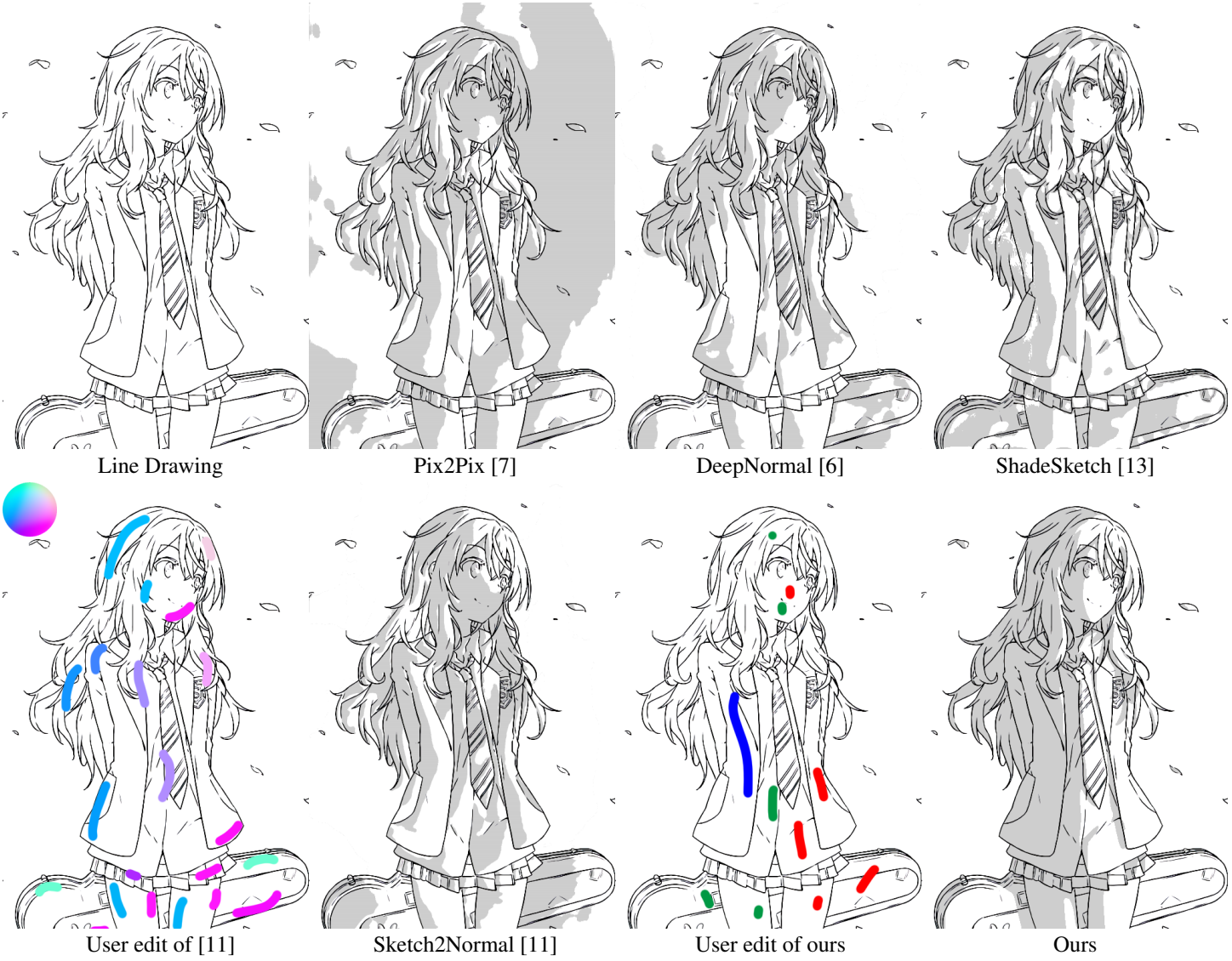


Figure 31: **Additional visual comparison #1.** We compare our approach to other possible alternatives. In the first row are automatic methods, and interactive methods are in the second row. The user scribbles are precisely one-pixel width and we dilated the scribbles for clearer presentation. *Artworks used with artist permissions.*

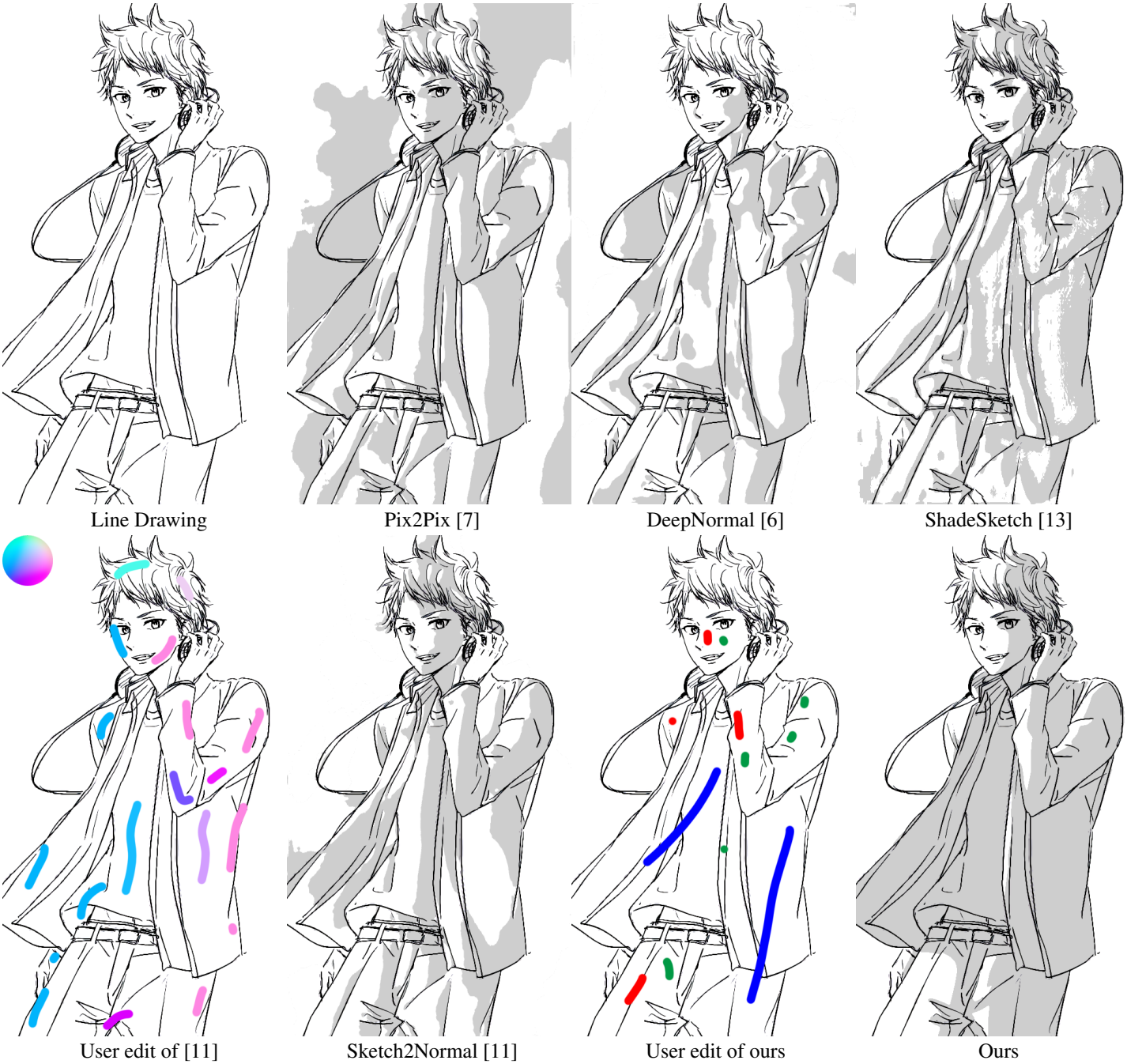


Figure 32: **Additional visual comparison #2.** We compare our approach to other possible alternatives. In the first row are automatic methods, and interactive methods are in the second row. The user scribbles are precisely one-pixel width and we dilated the scribbles for clearer presentation. *Artworks used with artist permissions.*

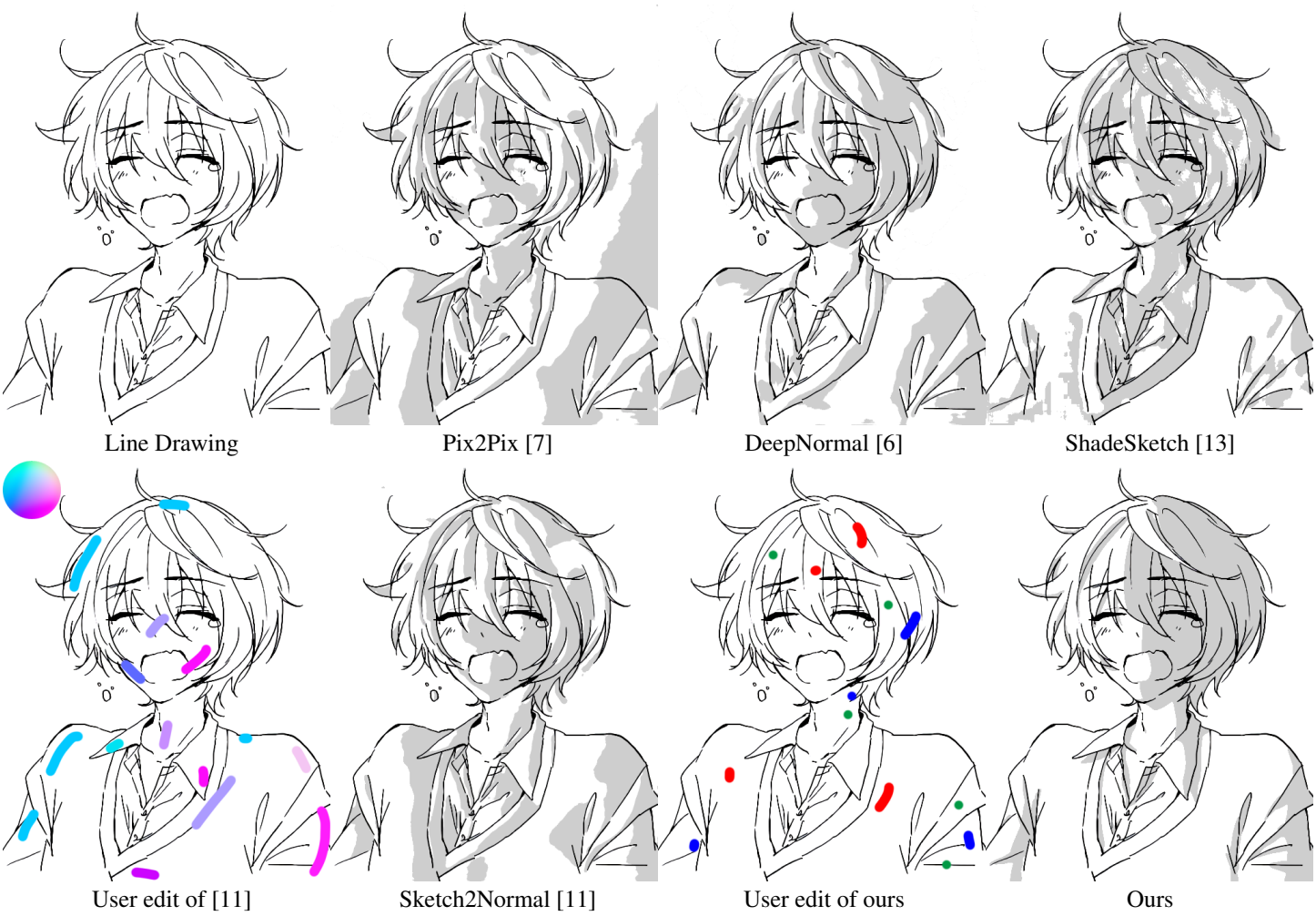


Figure 33: **Additional visual comparison #3.** We compare our approach to other possible alternatives. In the first row are automatic methods, and interactive methods are in the second row. The user scribbles are precisely one-pixel width and we dilated the scribbles for clearer presentation. *Artworks used with artist permissions.*

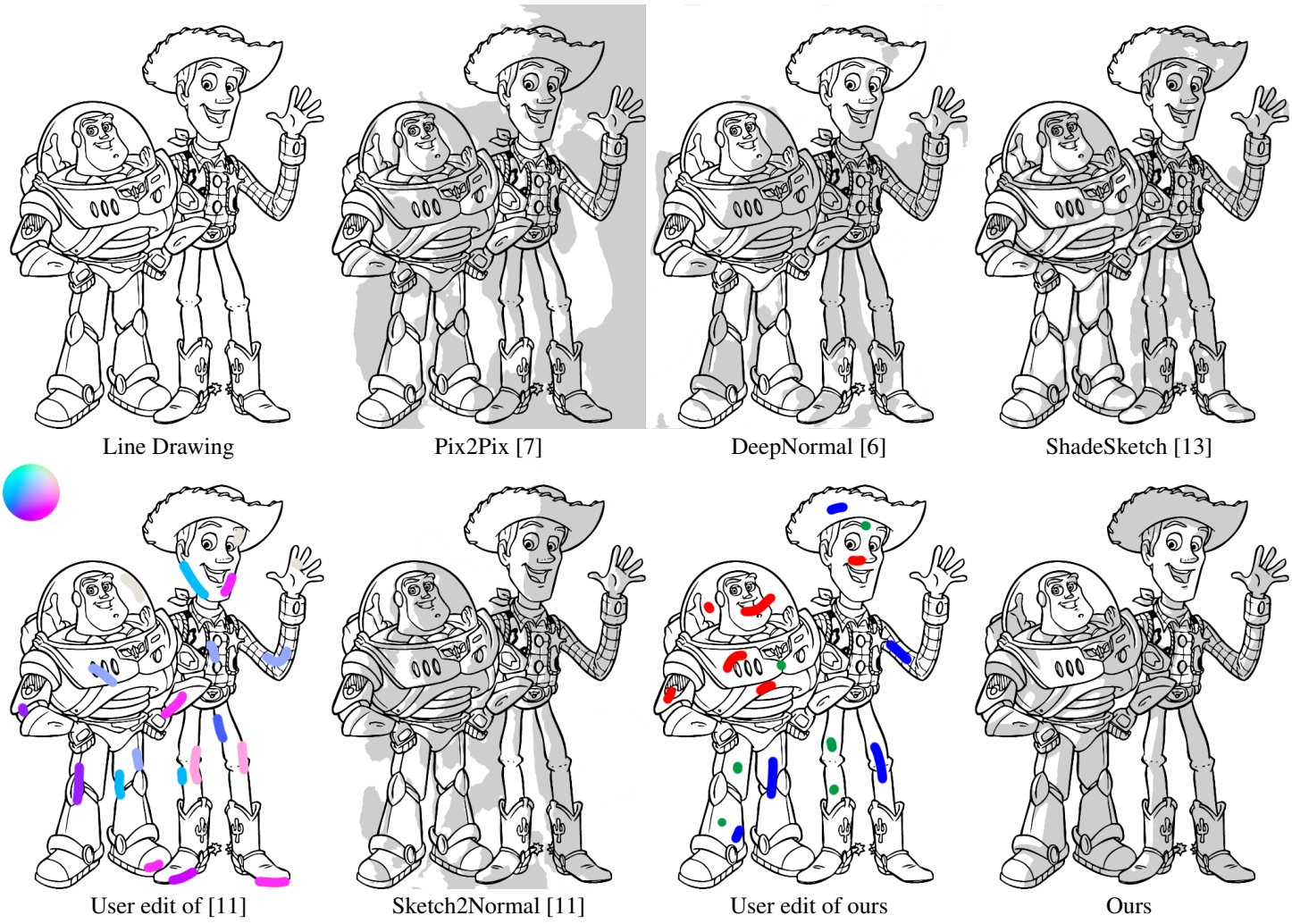
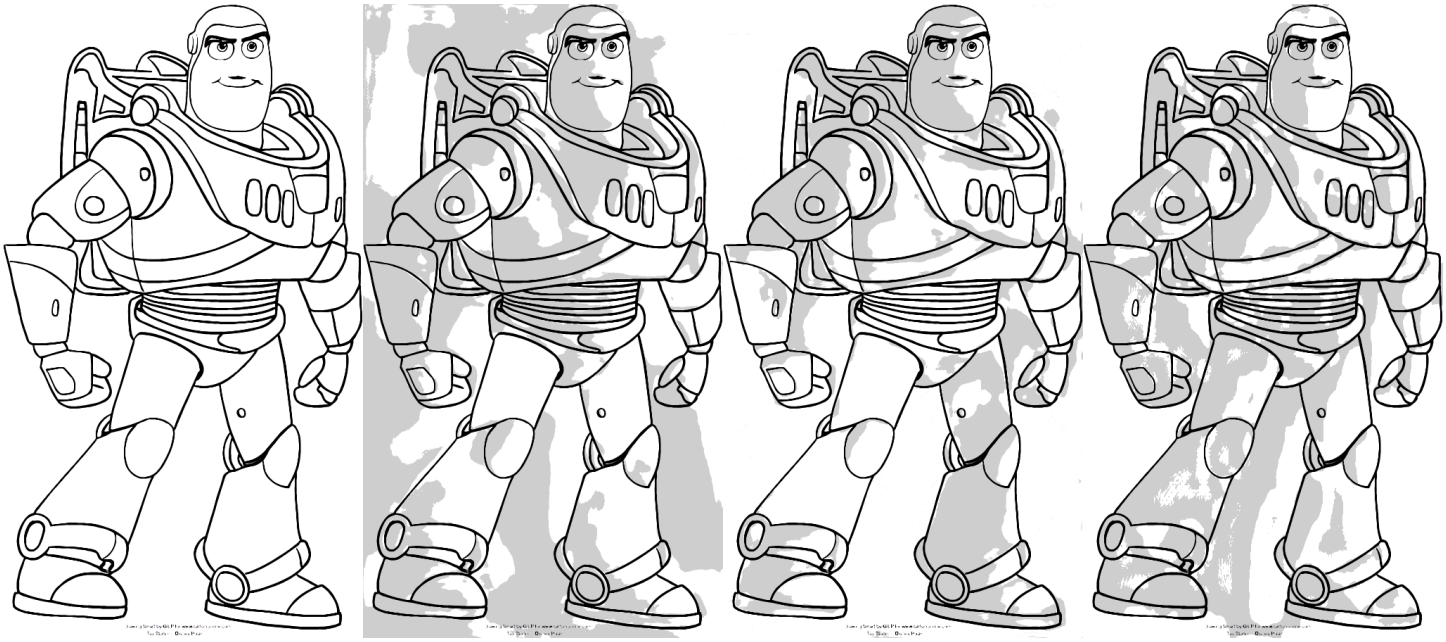


Figure 34: **Additional visual comparison #4.** We compare our approach to other possible alternatives. In the first row are automatic methods, and interactive methods are in the second row. The user scribbles are precisely one-pixel width and we dilated the scribbles for clearer presentation. *Artworks used with artist permissions.*



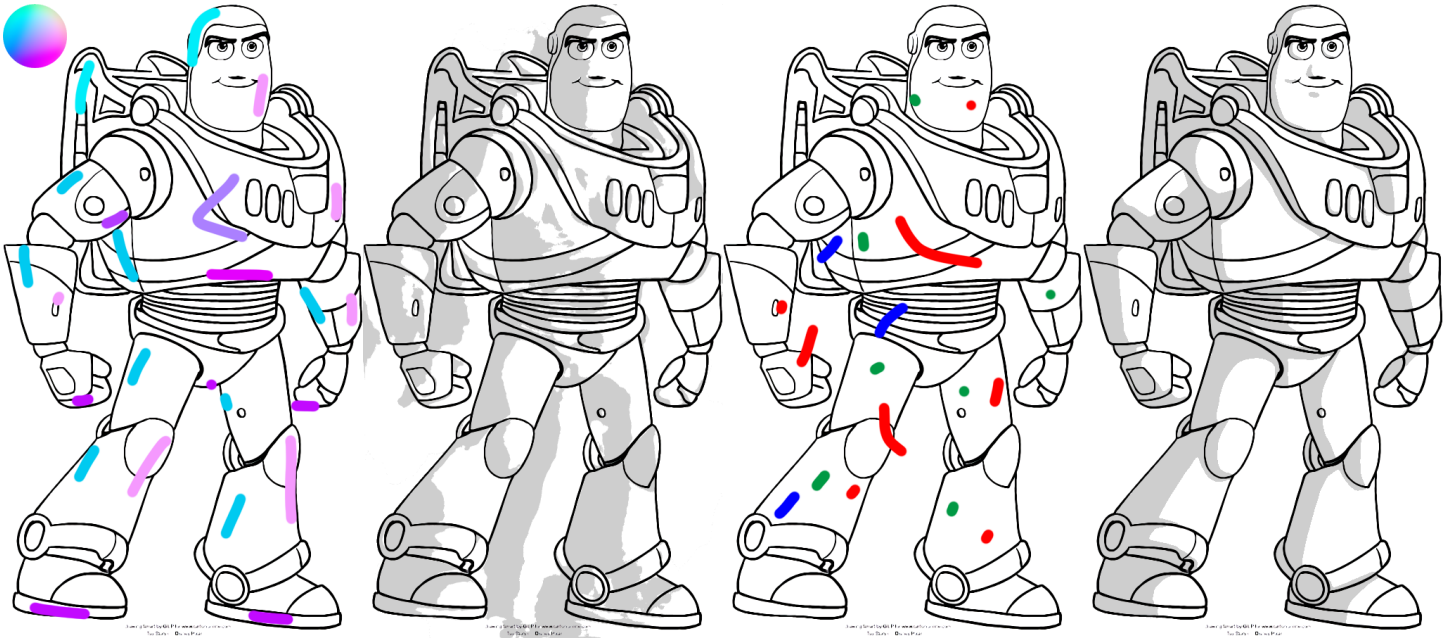


Figure 35: **Additional visual comparison #5.** We compare our approach to other possible alternatives. In the first row are automatic methods, and interactive methods are in the second row. The user scribbles are precisely one-pixel width and we dilated the scribbles for clearer presentation. *Artworks used with artist permissions.*

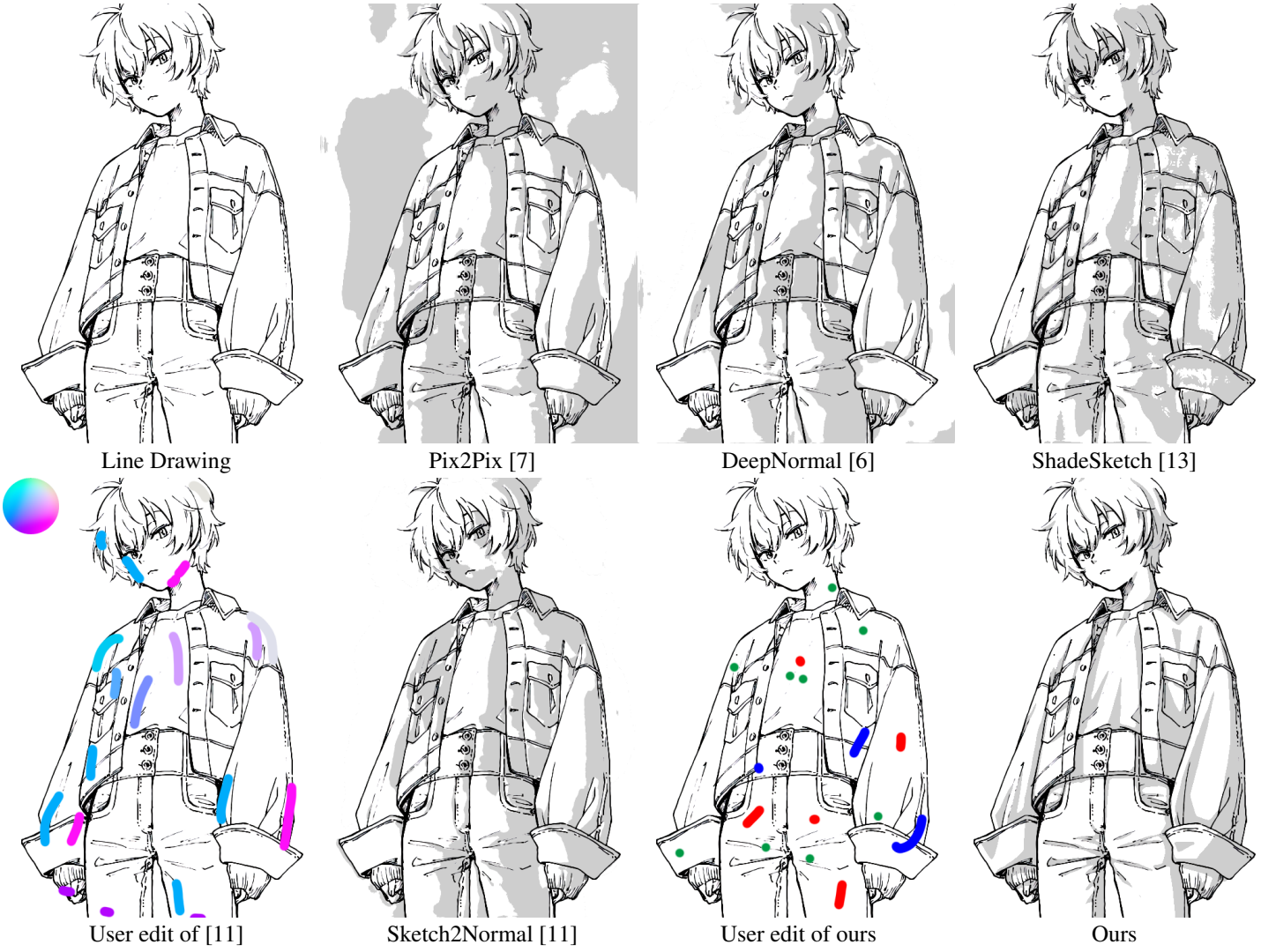


Figure 36: **Additional visual comparison #6.** We compare our approach to other possible alternatives. In the first row are automatic methods, and interactive methods are in the second row. The user scribbles are precisely one-pixel width and we dilated the scribbles for clearer presentation. *Artworks used with artist permissions.*

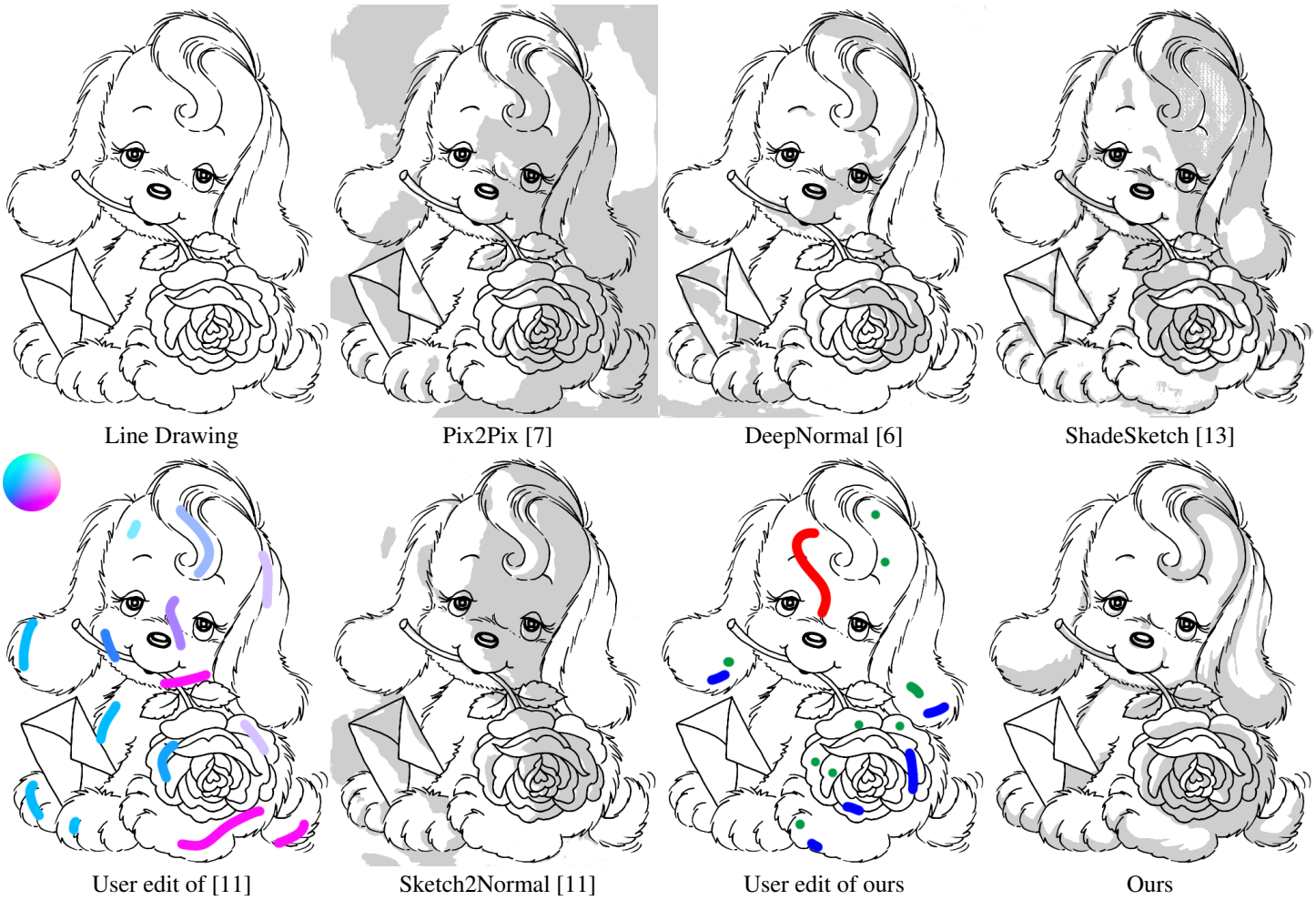


Figure 37: **Additional visual comparison #7.** We compare our approach to other possible alternatives. In the first row are automatic methods, and interactive methods are in the second row. The user scribbles are precisely one-pixel width and we dilated the scribbles for clearer presentation. *Artworks used with artist permissions.*

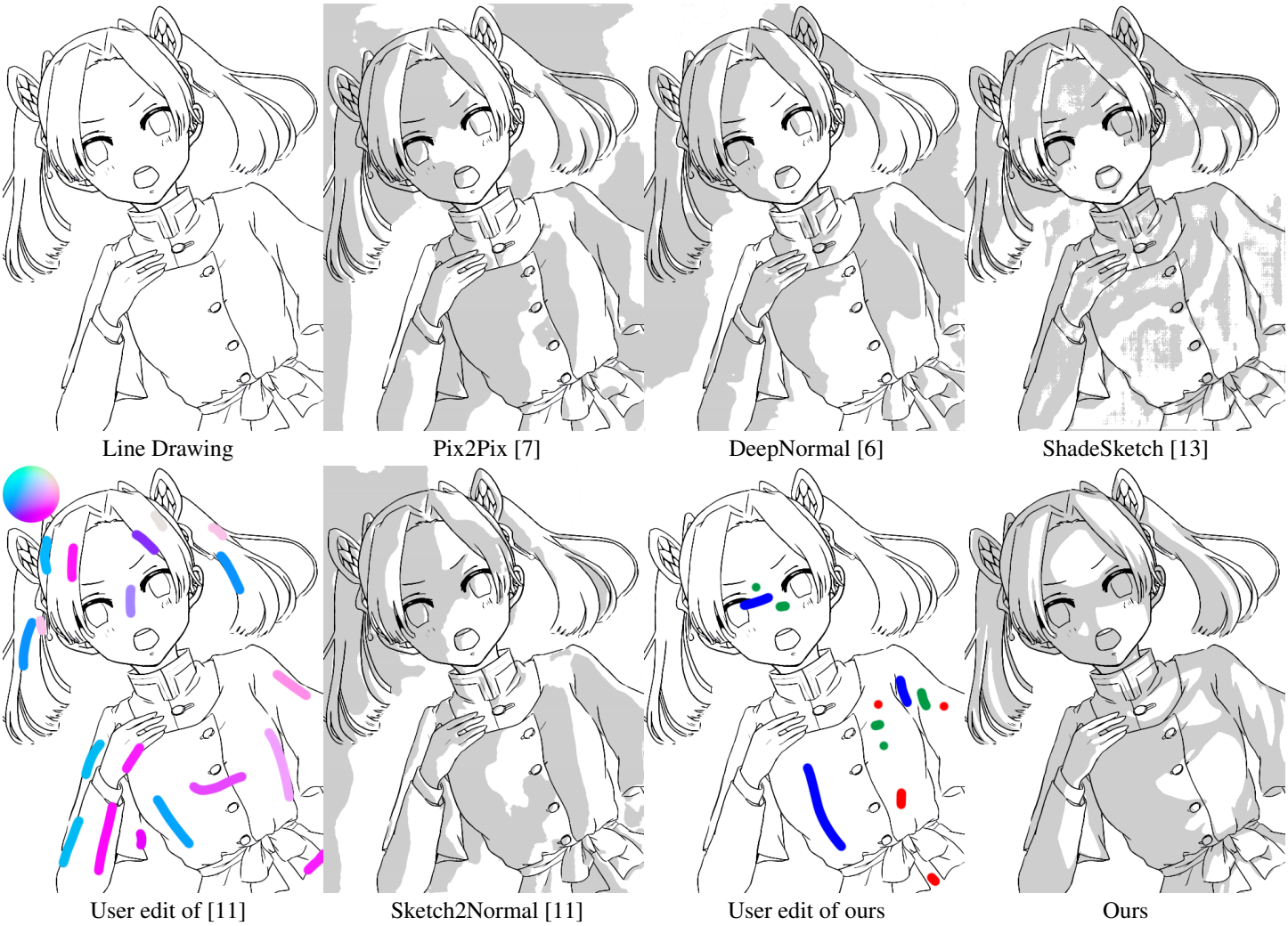


Figure 38: **Additional visual comparison #8.** We compare our approach to other possible alternatives. In the first row are automatic methods, and interactive methods are in the second row. The user scribbles are precisely one-pixel width and we dilated the scribbles for clearer presentation. *Artworks used with artist permissions.*

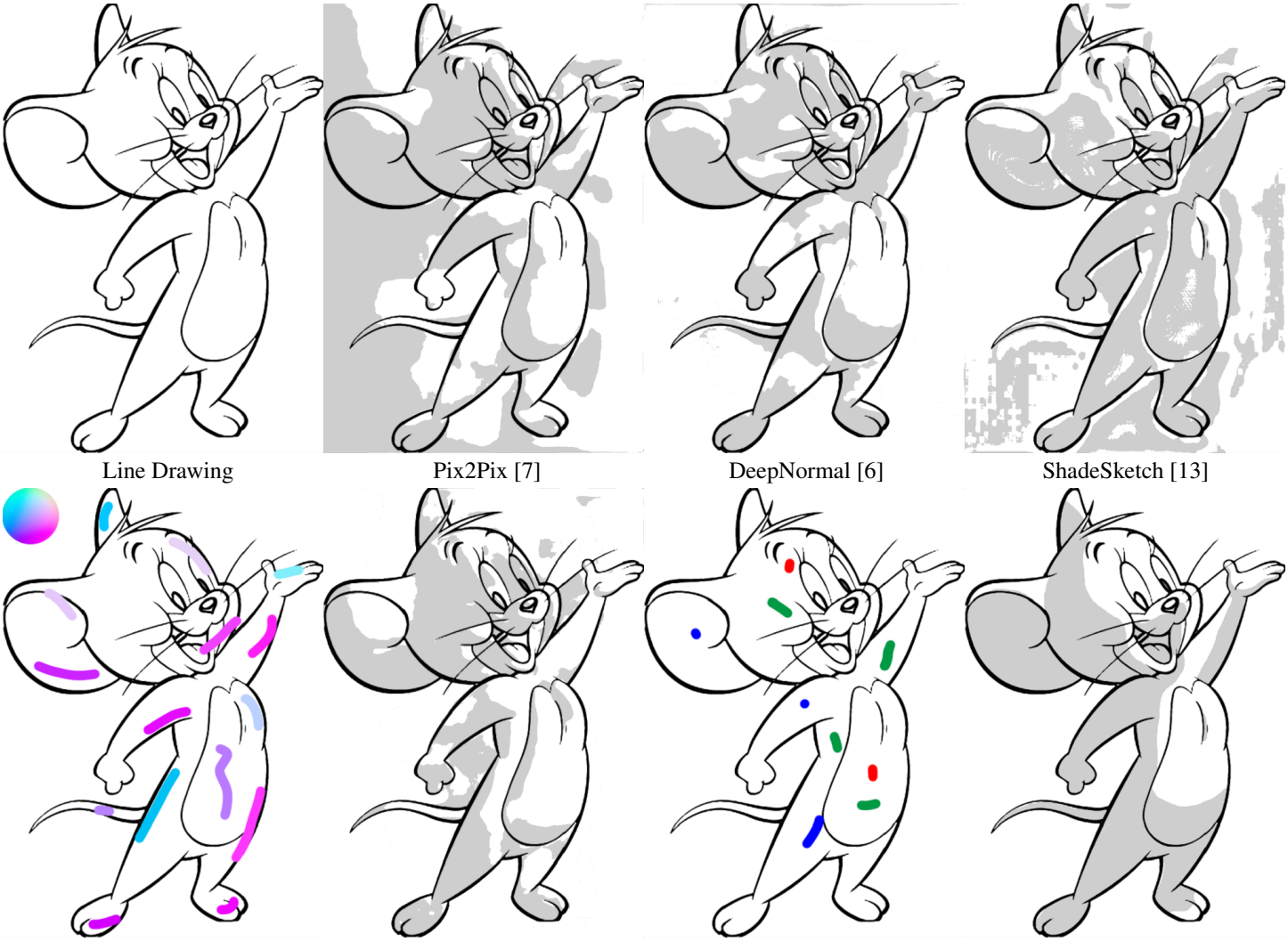


Figure 39: **Additional visual comparison #9.** We compare our approach to other possible alternatives. In the first row are automatic methods, and interactive methods are in the second row. The user scribbles are precisely one-pixel width and we dilated the scribbles for clearer presentation. *Artworks used with artist permissions.*

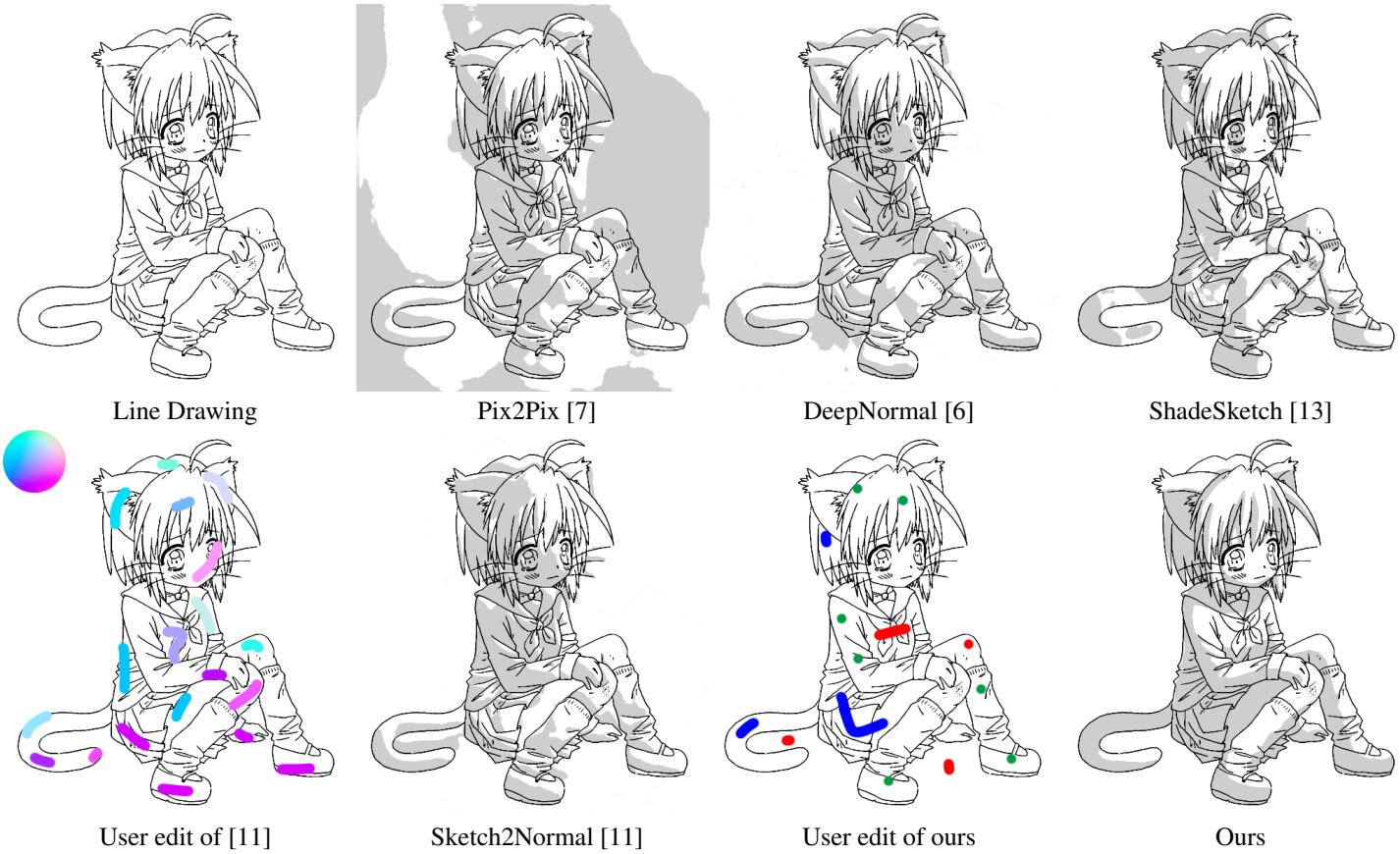


Figure 40: **Additional visual comparison #10.** We compare our approach to other possible alternatives. In the first row are automatic methods, and interactive methods are in the second row. The user scribbles are precisely one-pixel width and we dilated the scribbles for clearer presentation. *Artworks used with artist permissions.*

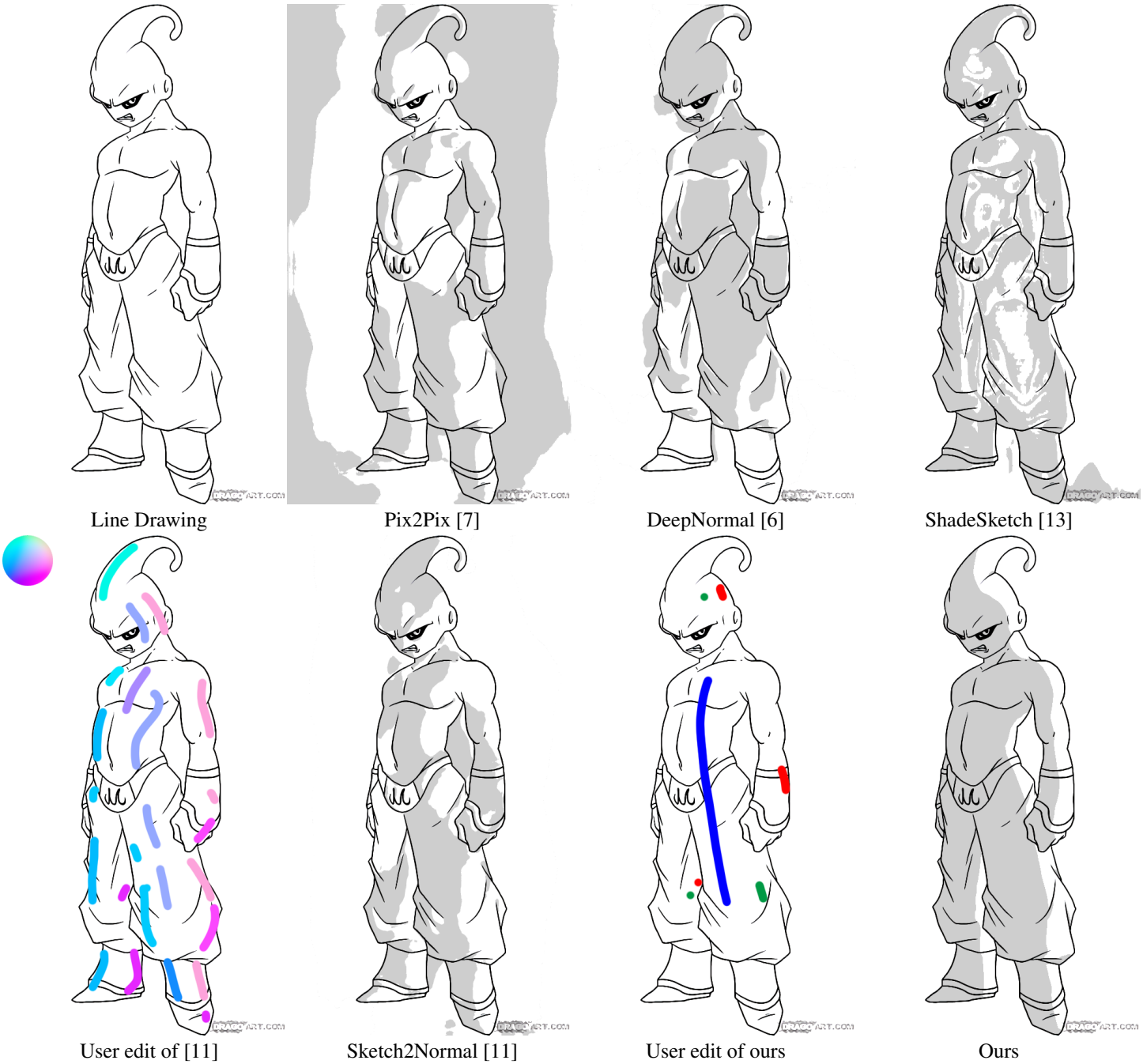


Figure 41: **Additional visual comparison #11.** We compare our approach to other possible alternatives. In the first row are automatic methods, and interactive methods are in the second row. The user scribbles are precisely one-pixel width and we dilated the scribbles for clearer presentation. *Artworks used with artist permissions.*

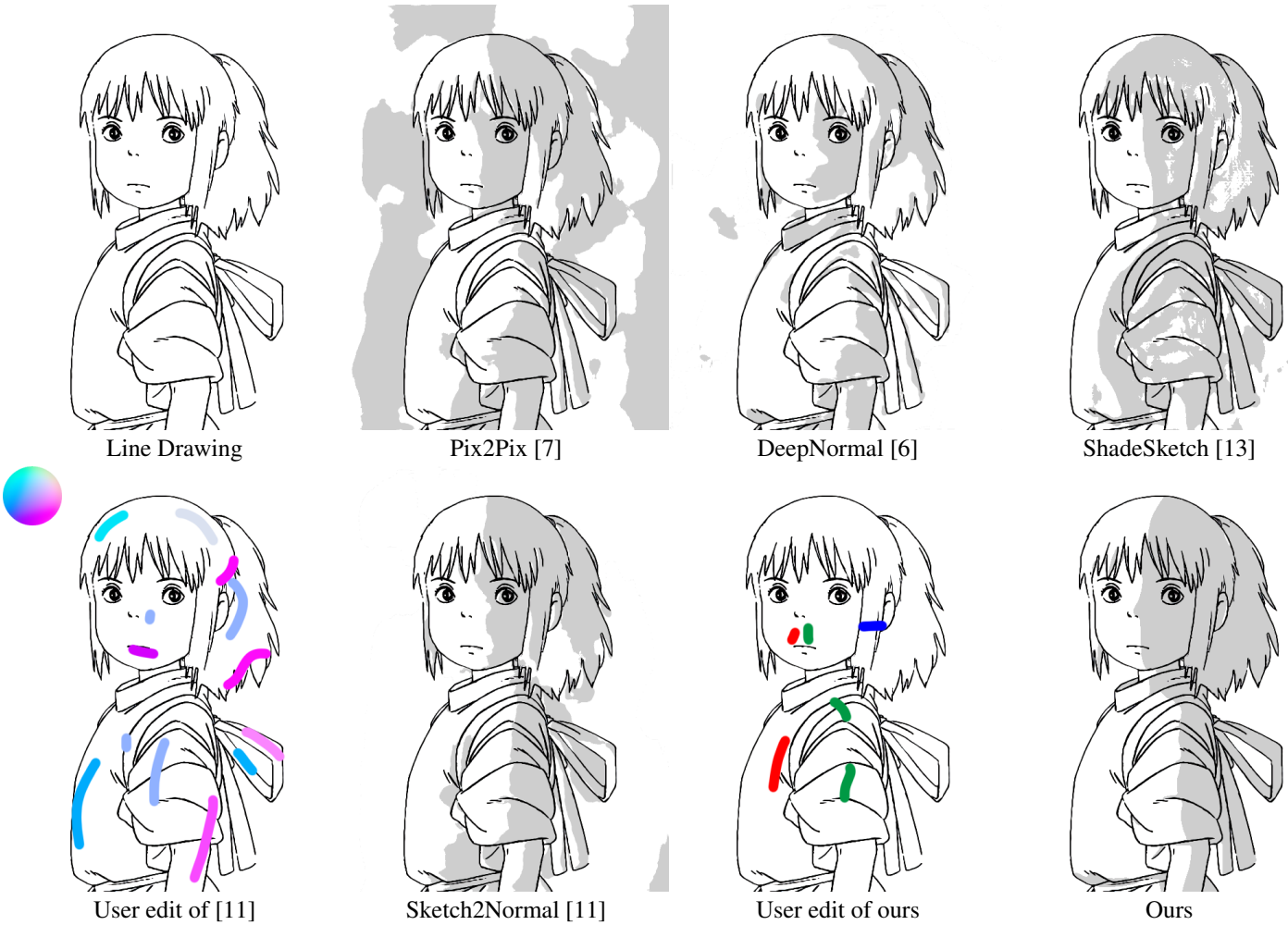


Figure 42: **Additional visual comparison #12.** We compare our approach to other possible alternatives. In the first row are automatic methods, and interactive methods are in the second row. The user scribbles are precisely one-pixel width and we dilated the scribbles for clearer presentation. *Artworks used with artist permissions.*

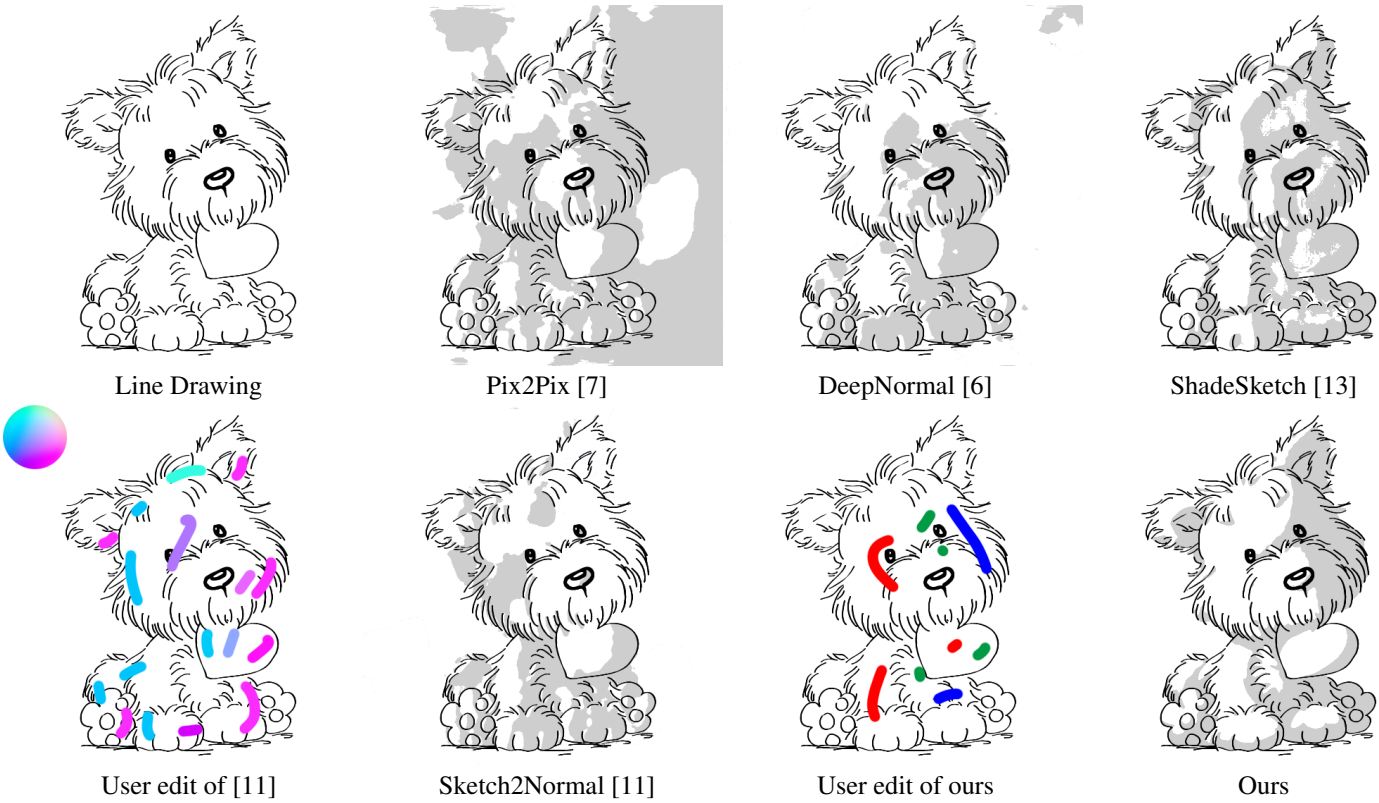


Figure 43: **Additional visual comparison #13.** We compare our approach to other possible alternatives. In the first row are automatic methods, and interactive methods are in the second row. The user scribbles are precisely one-pixel width and we dilated the scribbles for clearer presentation. *Artworks used with artist permissions.*

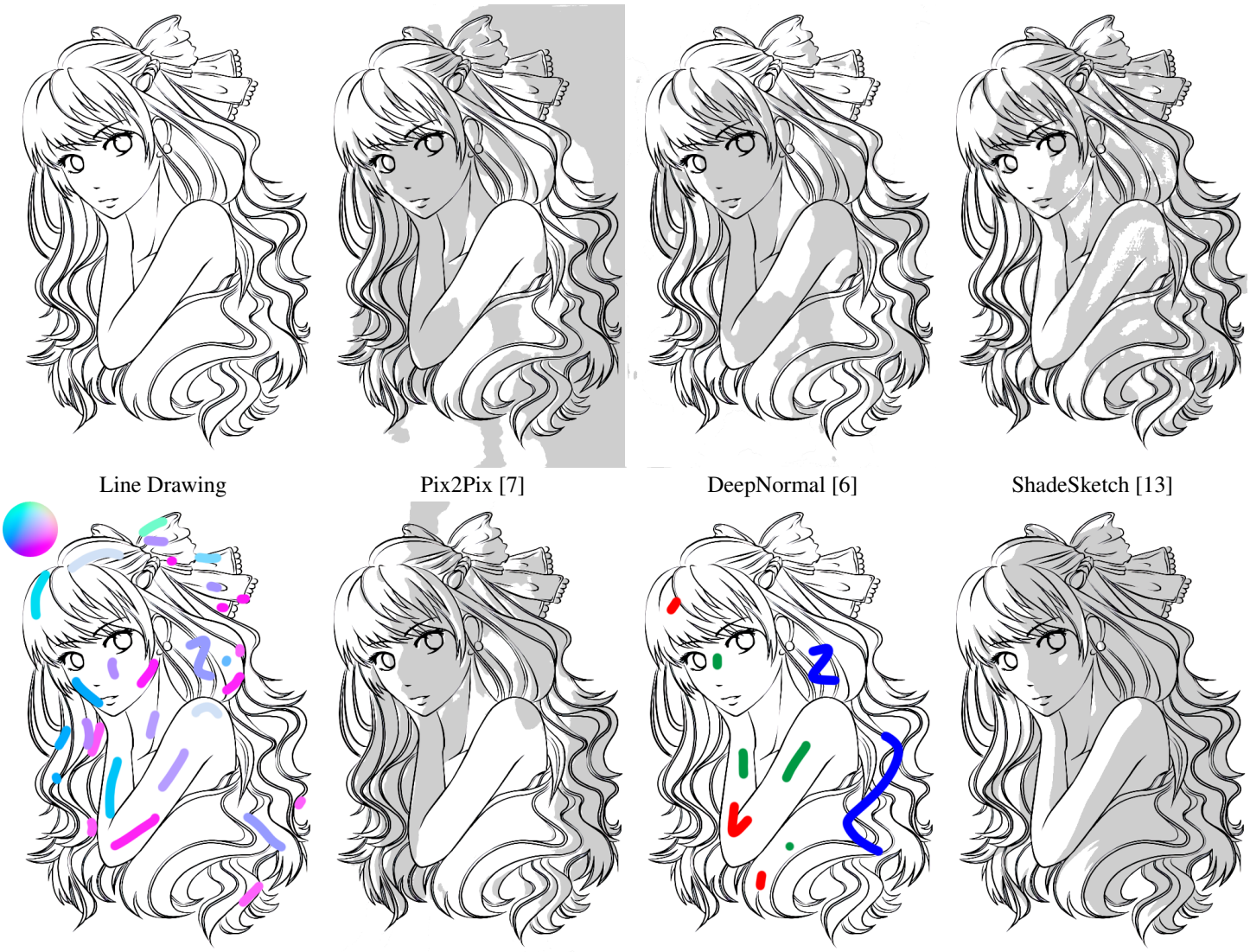


Figure 44: **Additional visual comparison #14.** We compare our approach to other possible alternatives. In the first row are automatic methods, and interactive methods are in the second row. The user scribbles are precisely one-pixel width and we dilated the scribbles for clearer presentation. *Artworks used with artist permissions.*

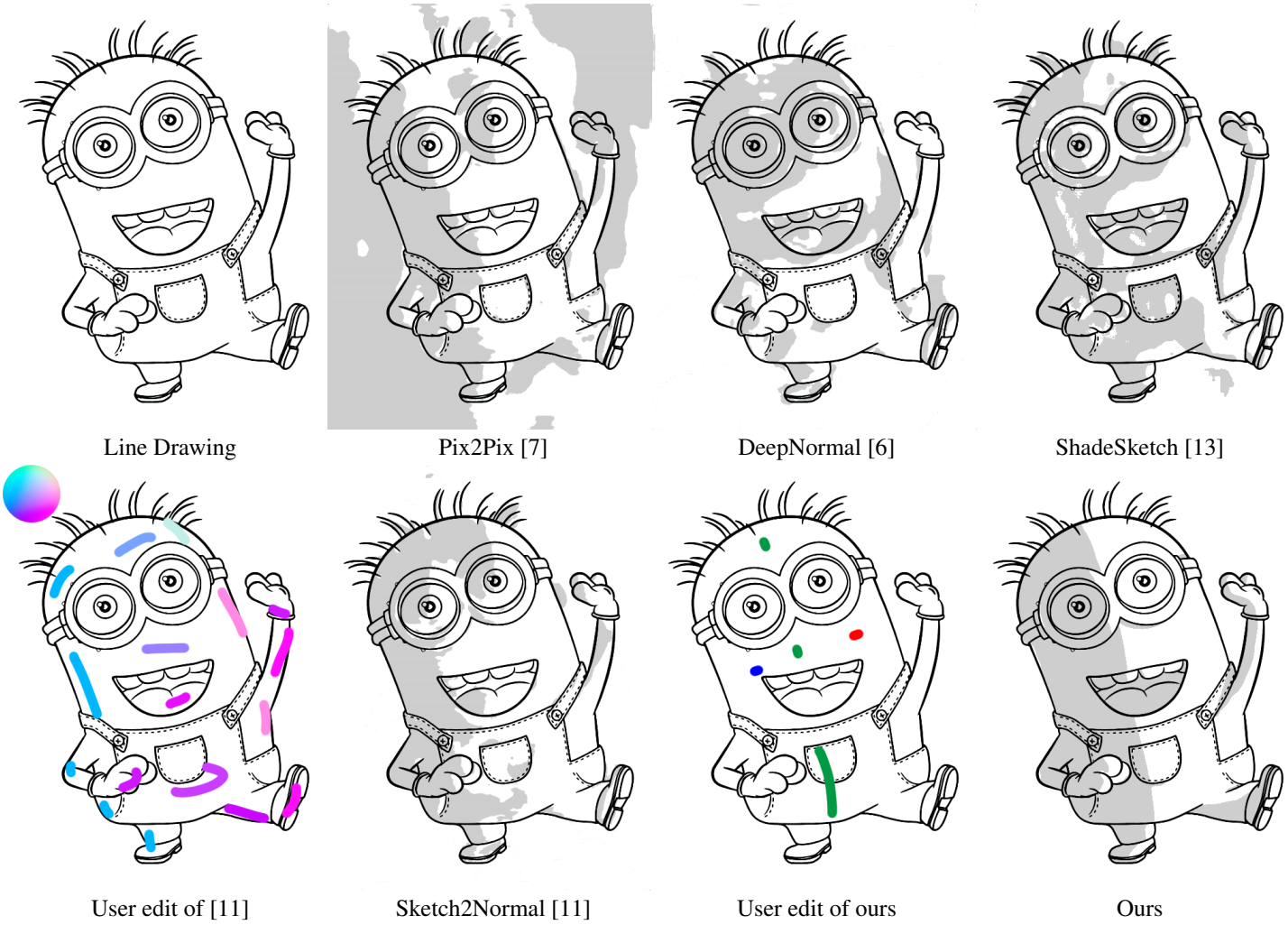


Figure 45: **Additional visual comparison #15.** We compare our approach to other possible alternatives. In the first row are automatic methods, and interactive methods are in the second row. The user scribbles are precisely one-pixel width and we dilated the scribbles for clearer presentation. *Artworks used with artist permissions.*

2 Additional user study details

We report on the user study conducting during the evaluation of our approach.

2.1 Design

We sample 52 unseen line drawings from Pixiv [9], and then assign each line drawing to 3 random users targeted to 3 methods:

- commercial tool (Adobe PhotoShop);
- baseline interactive method Su [11]; and
- our approach (interactive mode).

We also use 5 fully-automatic methods:

- DeepNormal [6];
- Sketch2Normal [11] (automatic mode);
- Pix2Pix [7];
- ShadeSketch [13]; and
- our approach (automatic mode).

We ensure that any image is assigned to each user at most once to avoid users being trained for specific instances. We study the drawing time and output quality of those methods.

2.2 Hardware (Wacom) and compared commercial software (Adobe PhotoShop)

We perform the user test using a Wacom MobileStudio Pro 16, which is a professional mobile pen computer designed for illustration. It comes both with a pen with pressure sensitivity and employs a large 16" touch screen. We note that while Adobe PhotoShop uses this pressure sensitivity, our approach does not.

We perform the user test using the commercial software Adobe PhotoShop, which is one of the most widely used digital painting software in related industry. It contains extensive features for shadow drawing, *e.g*, smart selection, smart filling, smoothed pens, *etc*, and it particular, drawing shadows on cartoon or comic line drawings. This professional software package is what we compared SmartShadow to and we abbreviate as Commercial Tools (CT).

2.3 User explanation

We inform users that "your time consumption will be recorded and please draw at your normal speed" when they are drawing shadows interactively, so as to capture reliable time data.

After they are finished, the users are shuffled and assigned to rank the shadow results of fully-automatic methods [6, 7, 13] and the automatic outputs of our method. The asked question is "Which of the following shadow do you prefer most to use in your daily digital painting? Please rank according to your preference."

2.4 Raw experimental data

We use the Time Consumption (TC) as speed metric, *i.e*, we record the precise drawing minutes. We also use the Average Human Ranking (AHR) as preference metric. For each line drawing, the users rank the results of the 5 methods from 1 to 5 (lower is better). We present the raw time data in Table 1 and raw user preference data in Table 2.

ID	Commercial Tool (PhotoShop)	Ours (auto)
# 1	26 minutes 17 seconds	2 minutes 13 seconds
# 2	15 minutes 22 seconds	0 minutes 51 seconds
# 3	27 minutes 34 seconds	3 minutes 11 seconds
# 4	23 minutes 13 seconds	1 minutes 59 seconds
# 5	17 minutes 25 seconds	5 minutes 17 seconds
# 6	11 minutes 14 seconds	1 minutes 25 seconds
# 7	17 minutes 13 seconds	1 minutes 55 seconds
# 8	8 minutes 6 seconds	1 minutes 31 seconds
# 9	23 minutes 32 seconds	2 minutes 15 seconds
# 10	9 minutes 17 seconds	1 minutes 1 seconds
# 11	16 minutes 18 seconds	7 minutes 39 seconds
# 12	11 minutes 39 seconds	8 minutes 29 seconds
# 13	16 minutes 12 seconds	3 minutes 3 seconds
# 14	12 minutes 38 seconds	0 minutes 34 seconds
# 15	18 minutes 40 seconds	7 minutes 49 seconds
# 16	14 minutes 14 seconds	7 minutes 39 seconds
# 17	13 minutes 33 seconds	3 minutes 57 seconds
# 18	11 minutes 26 seconds	4 minutes 17 seconds
# 19	13 minutes 17 seconds	6 minutes 21 seconds
# 20	17 minutes 20 seconds	7 minutes 37 seconds
# 21	17 minutes 14 seconds	9 minutes 9 seconds
# 22	18 minutes 19 seconds	2 minutes 48 seconds
# 23	18 minutes 23 seconds	1 minutes 42 seconds
# 24	11 minutes 8 seconds	2 minutes 53 seconds
# 25	11 minutes 20 seconds	6 minutes 54 seconds
# 26	10 minutes 48 seconds	9 minutes 43 seconds
# 27	15 minutes 28 seconds	7 minutes 35 seconds
# 28	19 minutes 30 seconds	7 minutes 12 seconds
# 29	19 minutes 53 seconds	4 minutes 50 seconds
# 30	14 minutes 9 seconds	5 minutes 57 seconds
# 31	18 minutes 40 seconds	9 minutes 11 seconds
# 32	25 minutes 6 seconds	4 minutes 29 seconds
# 33	22 minutes 44 seconds	6 minutes 27 seconds
# 34	24 minutes 49 seconds	7 minutes 3 seconds
# 35	19 minutes 52 seconds	0 minutes 56 seconds
# 36	11 minutes 57 seconds	1 minutes 51 seconds
# 37	19 minutes 41 seconds	8 minutes 1 seconds
# 38	12 minutes 36 seconds	4 minutes 26 seconds
# 39	19 minutes 37 seconds	3 minutes 58 seconds
# 40	15 minutes 17 seconds	2 minutes 6 seconds
# 41	13 minutes 53 seconds	3 minutes 49 seconds
# 42	16 minutes 20 seconds	7 minutes 17 seconds
# 43	19 minutes 16 seconds	7 minutes 51 seconds
# 44	19 minutes 15 seconds	0 minutes 52 seconds
# 45	16 minutes 50 seconds	0 minutes 5 seconds
# 46	18 minutes 49 seconds	6 minutes 16 seconds
# 47	16 minutes 54 seconds	8 minutes 3 seconds
# 48	10 minutes 18 seconds	9 minutes 0 seconds
# 49	19 minutes 50 seconds	8 minutes 10 seconds
# 50	17 minutes 26 seconds	8 minutes 44 seconds
# 51	18 minutes 28 seconds	3 minutes 40 seconds
# 52	15 minutes 25 seconds	11 minutes 28 seconds
Mean	16 minutes 35 seconds	5 minutes 21 seconds
Std	9 minutes 43 seconds	4 minutes 25 seconds

Table 1: **Time Consumption (TC).** We compare the time consuming of a typical commercial tool (Adobe PhotoShop) and ours. We present the raw timing data of 52 cases.

ID	Pix2Pix [7]	DeepNormal [6]	S2N [11] (auto)	ShadeSketch [13]	Ours (auto)
# 1	4	3	5	2	1
# 2	5	2	4	3	1
# 3	5	3	4	2	1
# 4	4	3	5	2	1
# 5	4	5	3	2	1
# 6	5	4	2	3	1
# 7	5	3	4	2	1
# 8	3	4	5	2	1
# 9	4	3	5	2	1
# 10	4	3	5	1	2
# 11	5	2	3	4	1
# 12	4	2	5	3	1
# 13	5	3	4	2	1
# 14	5	4	3	2	1
# 15	5	2	4	3	1
# 16	3	4	5	2	1
# 17	5	2	3	4	1
# 18	4	2	5	3	1
# 19	5	4	3	2	1
# 20	5	4	3	2	1
# 21	4	3	5	2	1
# 22	4	3	5	2	1
# 23	5	2	4	3	1
# 24	5	3	4	2	1
# 25	4	3	5	2	1
# 26	5	3	4	2	1
# 27	4	3	5	2	1

ID	Pix2Pix [7]	DeepNormal [6]	S2N [11] (auto)	ShadeSketch [13]	Ours (auto)
# 28	4	3	5	2	1
# 29	5	3	4	2	1
# 30	5	2	4	3	1
# 31	4	2	5	3	1
# 32	5	3	4	2	1
# 33	5	2	4	3	1
# 34	5	2	4	3	1
# 35	4	2	5	3	1
# 36	4	3	5	2	1
# 37	5	2	4	3	1
# 38	5	2	4	3	1
# 39	5	2	4	3	1
# 40	3	2	5	4	1
# 41	5	2	4	3	1
# 42	5	3	4	2	1
# 43	5	3	4	2	1
# 44	4	3	5	2	1
# 45	5	3	4	2	1
# 46	4	2	5	3	1
# 47	5	3	4	2	1
# 48	5	3	4	2	1
# 49	5	4	3	2	1
# 50	5	4	3	2	1
# 51	4	2	5	3	1
# 52	5	2	4	3	1
Mean	4.53	2.81	4.19	2.44	1.01
Std	0.60	0.76	0.76	0.63	0.13

Table 2: **Average Human Ranking (AHR).** We present the raw data of the ranking results of the user study.

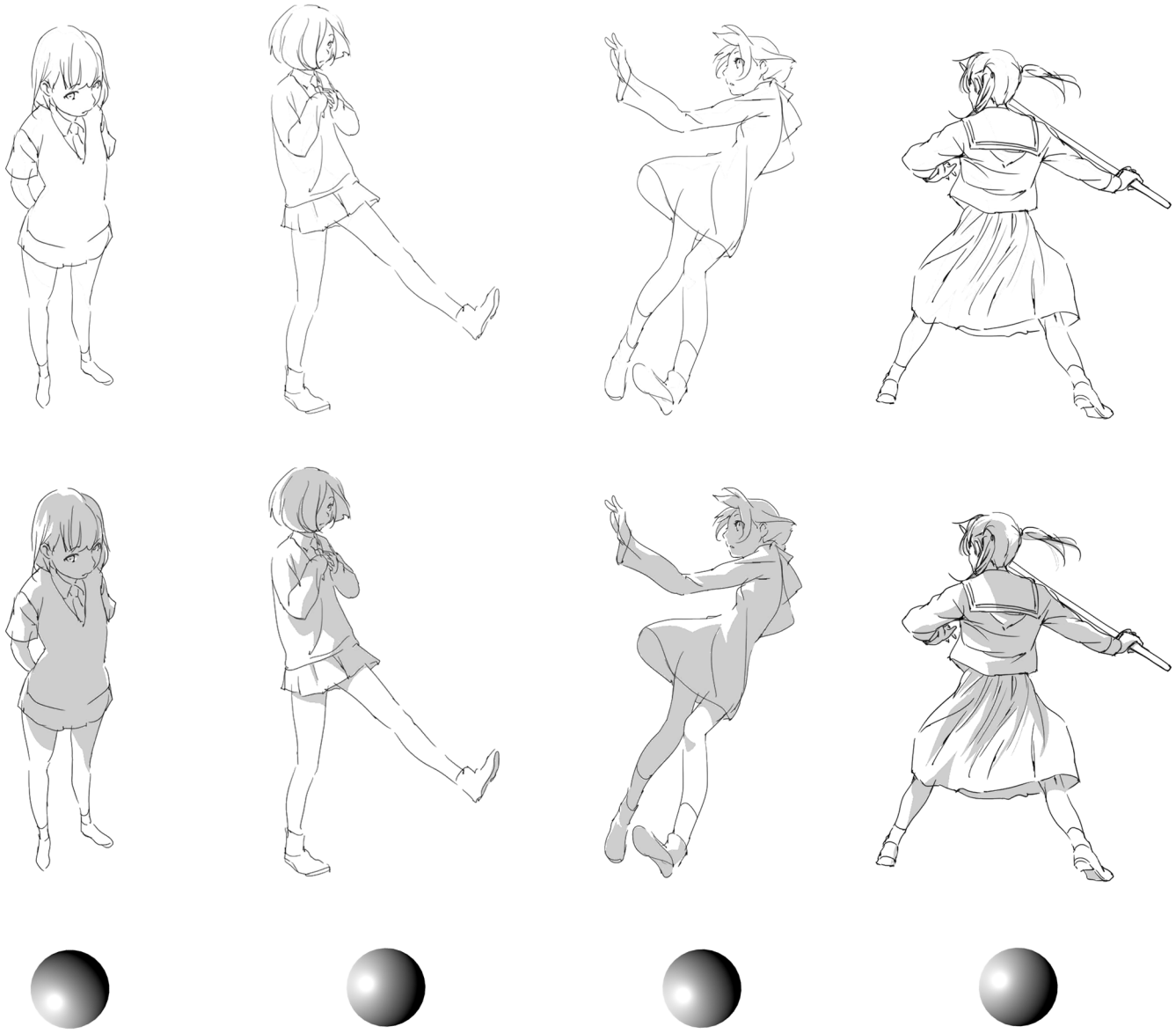


Figure 46: **Examples of data from real artists.** These real-artist shadow data are collected from internet artworks by searching "line drawing and shadow pairs" in illustration websites. These shadows are refined by our artists into usable data pairs for our dataset format. We present the annotated shadow direction below images.

3 Additional dataset details

3.1 Additional examples of presented data

We present additional examples of real shadow drawn by artists, rendered shadows, and extracted shadows in Fig. 46, 47, 48, and 49.

3.2 Algorithm details of shadow synthesizing

We present more detailed implementation about our shadow data synthesizing algorithms. These algorithms are not proposed in this paper. The verification, validation, justification of these methods can be found in related publications.

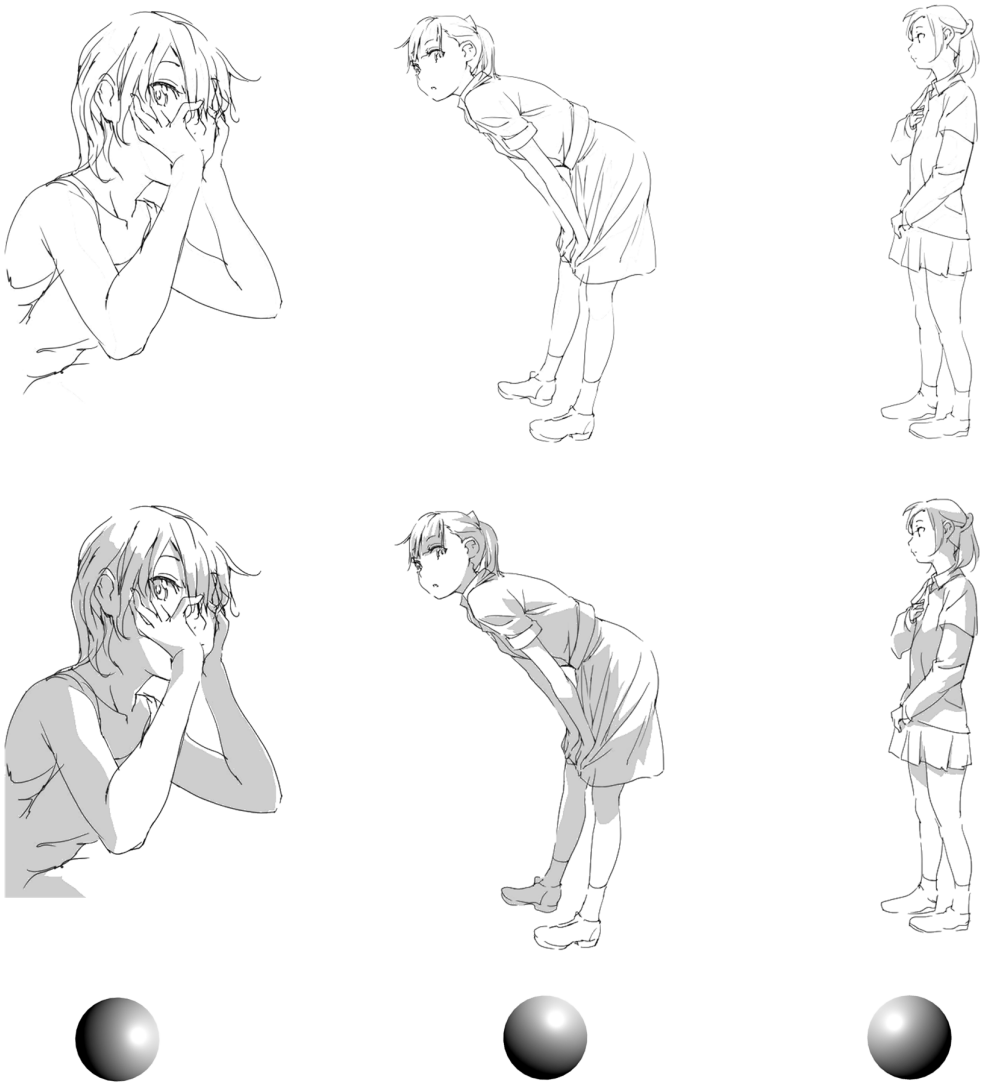


Figure 47: **Examples of data from real artists.** These real-artist shadow data are collected from internet artworks by searching "line drawing and shadow pairs" in illustration websites. These shadows are refined by our artists into usable data pairs for our dataset format. We present the annotated shadow direction below images.

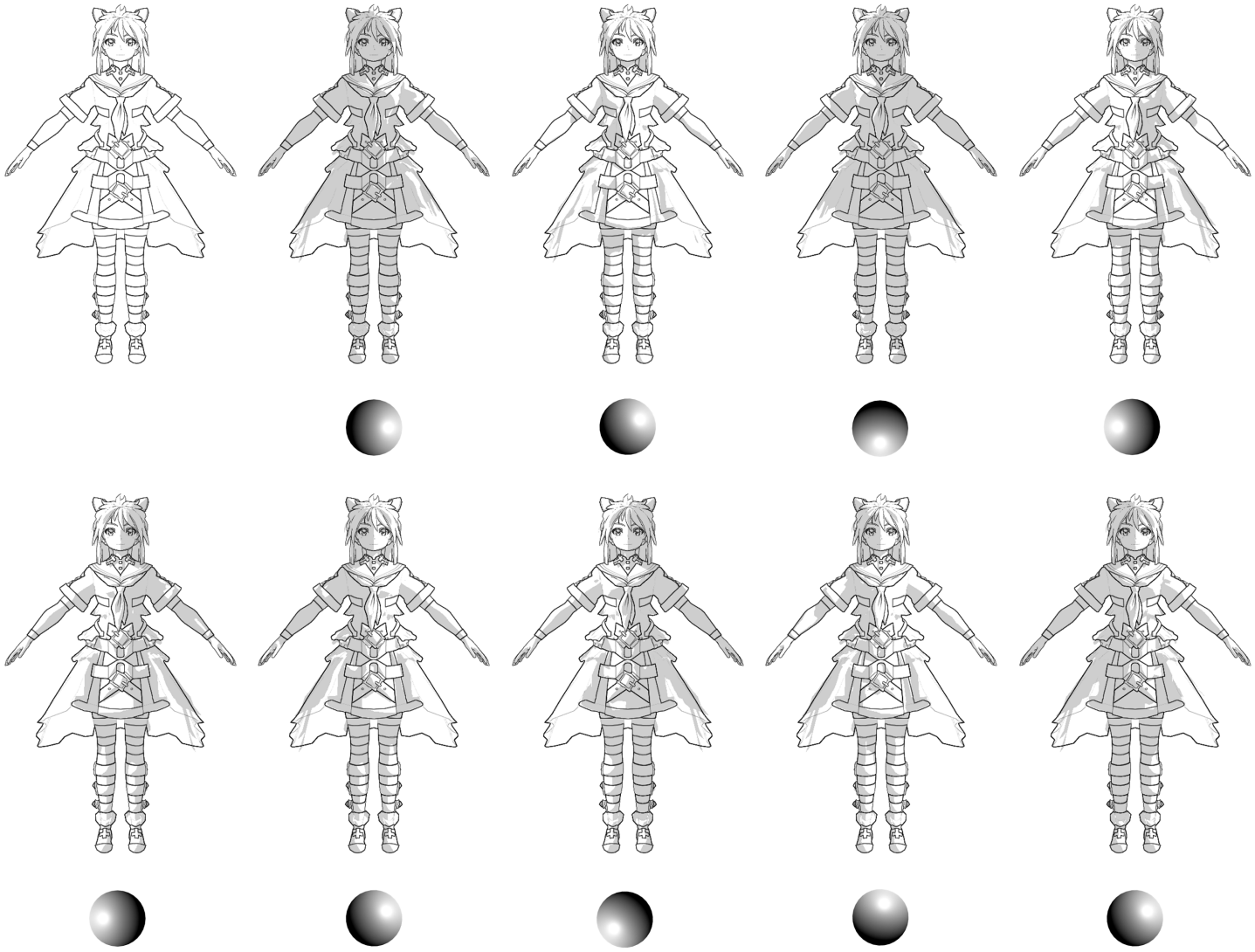


Figure 48: **Examples of data from rendering engine.** We present the shadow data rendered by blender. We present the recorded shadow direction below images. These data are only used in pre-training.

3.2.1 Real shadow drawn by artists manually

We invite artists to draw shadows on line drawings. We provide 1670 shadow samples drawn by 12 actual artists. We search the key word "line drawing" in illustration websites Pixiv [9] and Danbooru [5] to get 10,000 line drawings. Then the invited 12 artists select their interested line drawings and choose their preferred shadow directions. They draw the corresponding shadow according to their perceptual understandings. We collect 1670 high-quality shadow samples drawn by artists. Note that, to save artists' work, we also accept line drawing and shadow pairs from their daily jobs. Some artists work for animation companies and they own some line drawing and shadow pairs recorded in their previous animation projects. We also accept these data.

3.2.2 Rendered shadow and Non-Photorealistic Rendering (NPR)

We use Non-Photorealistic Rendering (NPR) techniques to render line drawings and shadows. We search the key word "free" in *Unity 3D Assets Store* and download 471 random 3D prefabs. We import them to the rendering engine Blender [4] and then use a NPR script to generate 25,413 line art and shadow pairs at random shadow directions. To be specific, we put one single lighting source in the scene, and the shadow direction is recorded as the opposite direction as the lighting direction.

Figure 49: **Examples of data from shadow extraction.** Note that these data are **only used in pre-training**, and these 291,951 shadow pairs do not need to have very high quality or very clear problem formulation because they only warm-up the learning before the actual training. See also the main paper for the details of our customized training schedule. Note that in this figure we have not binarized the shades. The binarization will be performed in the training time because the binarization is fast and sometimes we want to use dynamic binarization threshold values to get different types of shadows.

3.2.3 Extracted shadow and intrinsic imaging

We use shadow extraction method to extract shadows, and use line drawing extraction methods to extract line drawings. We sample 300,000 random digital paintings from Danbooru dataset [5] and Pixiv [9]. We use auto inking method [10] to extract line arts. To be specific, [10] is a sketch inking method, and we directly input black-and-white version of illustrations as sketches, and use [10] to extract boundary lines. We use intrinsic imaging method [2] to decompose reflectance and illumination maps. In particular, we enhance the [2] with the method [12] (a method that can make the illumination decomposition more adequate, we use the open-sourced codes of [12] to enhance [2]), and we also use the method [3] to get the colored illumination maps (we use the officially described steps in [3] to implement this component). We perform a shadow voting using OTSU algorithm [8] to obtain the binarized shadow (this is an adaptive threshold searching system and OpenCV has embedded functions). Finally, we use the Barron&Malik model [1] to estimate the shadow direction (we use the official implementation of [1] to perform this step), and note that the shadow direction is recorded as the opposite vector of the estimated lighting direction. Afterwards, we one-by-one check the quality of samples, and remove 8,049 pairs with obviously low quality. In this way, we acquire the remaining 291,951 pairs with acceptable quality.

4 Implementation details

4.1 Code implementation of the neural architecture

We provide codes of our neural network implementations.

4.1.1 Shadow direction model

We present a python (Keras style) implementation of the neural network architecture in our shadow direction model.

Code 1. A Keras implementation of the shadow direction model architecture.

```
from keras.layers import Conv2D, Activation, Input, AveragePooling2D, GlobalAveragePooling2D, Concatenate, Reshape
from keras.models import Model

input_image = Input(shape=(None, None, 5))
conv1a = Conv2D(filters=32, strides=(1, 1), kernel_size=(3, 3), padding='same')(input_image)
conv1a = Activation('relu')(conv1a)
conv1b = Conv2D(filters=32, strides=(1, 1), kernel_size=(3, 3), padding='same')(conv1a)
conv1b = Activation('relu')(conv1b)
conv2a = AveragePooling2D(pool_size=(2, 2))(conv1b)
conv2a = Conv2D(filters=64, strides=(1, 1), kernel_size=(3, 3), padding='same')(conv2a)
conv2a = Activation('relu')(conv2a)
conv2b = Conv2D(filters=64, strides=(1, 1), kernel_size=(3, 3), padding='same')(conv2a)
conv2b = Activation('relu')(conv2b)
conv3a = AveragePooling2D(pool_size=(2, 2))(conv2b)
conv3a = Conv2D(filters=128, strides=(1, 1), kernel_size=(3, 3), padding='same')(conv3a)
conv3a = Activation('relu')(conv3a)
conv3b = Conv2D(filters=128, strides=(1, 1), kernel_size=(3, 3), padding='same')(conv3a)
conv3b = Activation('relu')(conv3b)
conv3c = Conv2D(filters=128, strides=(1, 1), kernel_size=(3, 3), padding='same')(conv3b)
conv3c = Activation('relu')(conv3c)
conv4a = AveragePooling2D(pool_size=(2, 2))(conv3b)
conv4a = Conv2D(filters=256, strides=(1, 1), kernel_size=(3, 3), padding='same')(conv4a)
conv4a = Activation('relu')(conv4a)
conv4b = Conv2D(filters=512, strides=(1, 1), kernel_size=(3, 3), padding='same')(conv4a)
conv4b = Activation('relu')(conv4b)
conv4c = Conv2D(filters=512, strides=(1, 1), kernel_size=(3, 3), padding='same')(conv4b)
conv4c = Activation('relu')(conv4c)
conv4d = Conv2D(filters=512, strides=(1, 1), kernel_size=(3, 3), padding='same')(conv4c)
conv4d = GlobalAveragePooling2D(conv4d)
shadow_direction = Conv2D(filters=3, strides=(1, 1), kernel_size=(3, 3), padding='same')(conv4c)
model = Model(inputs=input_image, outputs=shadow_direction)
```

The “input_image” node is the placeholder for the input image, and the “shadow_direction” is for the output shadow direction.

4.1.2 Shadow model

We present a python (Keras style) implementation of the neural network architecture in our shadow model.

Code 2. A Keras implementation of the shadow direction model architecture.

```
from keras.layers import Conv2D, Activation, Input, AveragePooling2D, UpSampling2D, Concatenate, Reshape
from keras.models import Model

input_image = Input(shape=(None, None, 5))
input_direction = Input(shape=(1, 1, 3))
conv1a = Conv2D(filters=32, strides=(1, 1), kernel_size=(3, 3), padding='same')(input_image)
conv1a = Activation('relu')(conv1a)
conv1b = Conv2D(filters=32, strides=(1, 1), kernel_size=(3, 3), padding='same')(conv1a)
conv1b = Activation('relu')(conv1b)
conv2a = AveragePooling2D(pool_size=(2, 2))(conv1b)
conv2a = Conv2D(filters=64, strides=(1, 1), kernel_size=(3, 3), padding='same')(conv2a)
conv2a = Activation('relu')(conv2a)
conv2b = Conv2D(filters=64, strides=(1, 1), kernel_size=(3, 3), padding='same')(conv2a)
conv2b = Activation('relu')(conv2b)
conv3a = AveragePooling2D(pool_size=(2, 2))(conv2b)
conv3a = Conv2D(filters=128, strides=(1, 1), kernel_size=(3, 3), padding='same')(conv3a)
conv3a = Activation('relu')(conv3a)
conv3b = Conv2D(filters=128, strides=(1, 1), kernel_size=(3, 3), padding='same')(conv3a)
conv3b = Activation('relu')(conv3b)
conv3c = Conv2D(filters=128, strides=(1, 1), kernel_size=(3, 3), padding='same')(conv3b)
conv3c = Activation('relu')(conv3c)
conv4a = AveragePooling2D(pool_size=(2, 2))(conv3b)
conv4a = Concatenate()([conv4a, Reshape(target_shape=(conv4a._shape[1], conv4a._shape[2], 3))(input_direction)])
conv4a = Conv2D(filters=256, strides=(1, 1), kernel_size=(3, 3), padding='same')(conv4a)
conv4a = Activation('relu')(conv4a)
conv4b = Conv2D(filters=512, strides=(1, 1), kernel_size=(3, 3), padding='same')(conv4a)
conv4b = Activation('relu')(conv4b)
conv4c = Conv2D(filters=512, strides=(1, 1), kernel_size=(3, 3), padding='same')(conv4b)
conv4c = Activation('relu')(conv4c)
conv4d = Conv2D(filters=512, strides=(1, 1), kernel_size=(3, 3), padding='same')(conv4c)
conv4d = Activation('relu')(conv4d)
conv4e = Conv2D(filters=256, strides=(1, 1), kernel_size=(3, 3), padding='same')(conv4d)
conv4e = Activation('relu')(conv4e)
conv5a = UpSampling2D(size=(2, 2))(conv4e)
conv5a = Conv2D(filters=128, strides=(1, 1), kernel_size=(3, 3), padding='same')(conv5a)
conv5a = Concatenate()([conv5a, conv3b])
conv5a = Activation('relu')(conv5a)
conv5b = Conv2D(filters=128, strides=(1, 1), kernel_size=(3, 3), padding='same')(conv5a)
conv5b = Activation('relu')(conv5b)
conv5c = Conv2D(filters=128, strides=(1, 1), kernel_size=(3, 3), padding='same')(conv5b)
conv5c = Activation('relu')(conv5c)
conv6a = UpSampling2D(size=(2, 2))(conv5c)
conv6a = Conv2D(filters=64, strides=(1, 1), kernel_size=(3, 3), padding='same')(conv6a)
conv6a = Concatenate()([conv6a, conv2b])
conv6a = Activation('relu')(conv6a)
conv6b = Conv2D(filters=64, strides=(1, 1), kernel_size=(3, 3), padding='same')(conv6a)
conv6b = Activation('relu')(conv6b)
conv7a = UpSampling2D(size=(2, 2))(conv6b)
conv7a = Conv2D(filters=32, strides=(1, 1), kernel_size=(3, 3), padding='same')(conv7a)
conv7a = Concatenate()([conv7a, conv1b])
conv7a = Activation('relu')(conv7a)
conv7b = Conv2D(filters=32, strides=(1, 1), kernel_size=(3, 3), padding='same')(conv7a)
conv7b = Activation('relu')(conv7b)
shadow_output = Conv2D(filters=1, strides=(1, 1), kernel_size=(3, 3), padding='same')(conv7b)
model = Model(inputs=[input_image, input_direction], outputs=shadow_output)
```

The “input_image” node is the placeholder for the input image, the “shadow_direction” is for the input shadow direction, and the “shadow_output” is for the output shadow.

4.2 Experimental details of compared methods

4.2.1 Pix2Pix [7]

We train Pix2Pix with their official codes implementations. Note that Pix2Pix itself does not support global shadow direction embedding, and we directly embed the shadow direction vector in the middle U-net layers of Pix2Pix. This model is trained on our dataset, using the same training schedule as proposed in this paper (large-scale pre-training and fine tuning with artists' data) to form a fair comparison.

4.2.2 DeepNormal [6]

We use the open-sourced official implementation of DeepNormal. Note that DeepNormal outputs a normal map and we directly use DeepNormal's recommended rendering codes to get the shadows. The lighting direction is opposite to the global shadow direction, and we obtain the lighting direction by computing the opposite value of the shadow direction vector. Note that DeepNormal requires masks and we use the official method to provide masks for this method.

4.2.3 Sketch2Normal [11]

Sketch2Normal has many versions of their implementations. Some of these implementations use hint points as user inputs and some other implementations use rough scribbles as user inputs. We use an "scribble-based" version as the compared method. Similar as Pix2Pix, we train their model using the same scribble shapes as our method to form a fair comparison. Note that Sketch2Normal requires masks and we use their official method to input masks.

In this method, users are given a "normal ball" and they can draw normal vectors as geometry indications. The method output normal maps, and we use their recommended methods to render shadows. We obtain the lighting direction by computing the opposite value of our global shadow direction vector.

4.2.4 ShadeSketch [13]

ShadeSketch is the current state of the art in the artistic shadow drawing problem. The authors of ShadeSketch has not only open-sourced their models but also released an user interface. We use the official models and the official user interface to form the comparison.

5 Copyrights and ethics

Fig. 1, 2, 3, 4, 5, 21, 22, 24, 26, 28, 31, and 32 are artworks used with artist permissions. We thank all artists who have provided the permissions for us to demonstrate their artworks.

5.1 Ethic statements

The involved artworks may contain commercial elements like the focused Asia-style illustration topics (*e.g*, flower, girl, boy, elf, magic, fantasy, love, *etc*), Asia-style female/male depiction (*e.g*, romantic, cute, gorgeous, elegant, *etc*), and so on. These artworks captures the real data distribution in the real market for commercial illustrations in recent years. The choice of artworks is fully based on technical merits, and those commercial elements are indispensable for our technical presentation to investigate the real targets or problems of the commercial illustrations that we actually want to assist. The choice of artworks is NOT on behalf of the preference or criteria of the researchers.

References

[1] Jonathan T. Barron and Jitendra Malik. Shape, illumination, and reflectance from shading. *TPAMI*, 2015.

[2] Sai Bi, Xiaoguang Han, and Yizhou Yu. An l1 image transform for edge-preserving smoothing and scene-level intrinsic decomposition. *ACM Trans. Graph.*, 34(4), July 2015.

[3] Robert Carroll, Ravi Ramamoorthi, and Maneesh Agrawala. Illumination decomposition for material recoloring with consistent interreflections. In *ACM Transactions on Graphics*. ACM Press, 2011.

[4] Blender Online Community. *Blender - a 3D modelling and rendering package*. Blender Foundation, Stichting Blender Foundation, Amsterdam, 2018.

[5] DanbooruCommunity. Danbooru2017: A large-scale crowdsourced and tagged anime illustration dataset, 2018.

[6] Matis Hudon, Rafael Pages, Mairead Grogan, and Aljosa Smolic. Deep normal estimation for automatic shading of hand-drawn characters. *ECCV*, 2018.

[7] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. *CVPR*, 2017.

[8] Nobuyuki Otsu. A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 9(1):62–66, jan 1979.

[9] pixiv.net. pixiv. *pixiv*, 2007.

[10] Edgar Simo-Serra, Satoshi Iizuka, Kazuma Sasaki, and Hiroshi Ishikawa. Learning to Simplify: Fully Convolutional Networks for Rough Sketch Cleanup. *ACM Transactions on Graphics*, 35(4), 2016.

[11] Wanchao Su, Dong Du, Xin Yang, Shizhe Zhou, and Hongbo Fu. Interactive sketch-based normal map generation with deep neural networks. *ACM on Computer Graphics and Interactive Techniques*, 1(1):1–17, jul 2018.

[12] Lvmin Zhang, Chengze Li, Yi JI, Chunping Liu, and Tien tsin Wong. Erasing appearance preservation in optimization-based smoothing. In *European Conference on Computer Vision (ECCV)*, 2020.

[13] Qingyuan Zheng, Zhuoru Li, and Adam Bargteil. Learning to shadow hand-drawn sketches. In *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.