DOI: 10.1111/cgf.14800

COMPUTER GRAPHICS *forum*
Volume 42 (2023), number 6, e14800

Reference-based Screentone Transfer via Pattern Correspondence and Regularization

Zhansheng Li,[1] Nanxuan Zhao,[2] Zongwei Wu,[1] Yihua Dai,[1] Junle Wang,[3] Yanqing Jing[3] and Shengfeng He[4]

[1]South China University of Technology, Guangzhou, China
[2]Adobe Research, USA
[3]Tencent, Shenzhen, China
[4]Singapore Management University, Singapore
shengfenghe@smu.edu.sg

Abstract

Adding screentone to initial line drawings is a crucial step for manga generation, but is a tedious and human-laborious task. In this work, we propose a novel data-driven method aiming to transfer the screentone pattern from a reference manga image. This not only ensures the quality, but also adds controllability to the generated manga results. The reference-based screentone translation task imposes several unique challenges. Since manga image often contains multiple screentone patterns interweaved with line drawing, as an abstract art, this makes it even more difficult to extract disentangled style code from the reference. Also, finding correspondence for mapping between the reference and the input line drawing without any screentone is hard. As screentone contains many subtle details, how to guarantee the style consistency to the reference remains challenging. To suit our purpose and resolve the above difficulties, we propose a novel Reference-based Screentone Transfer Network (RSTN). We encode the screentone style through a 1D stylegram. A patch correspondence loss is designed to build a similarity mapping function for guiding the translation. To mitigate the generated artefacts, a pattern regularization loss is introduced in the patch-level. Through extensive experiments and a user study, we have demonstrated the effectiveness of our proposed model.

Keywords: manga, screentone, Reference-based, patch correspondence, pattern regularization

CCS Concepts: • Computing methodologies → Computer vision tasks; Neural networks; Unsupervised learning; Image processing

1. Introduction

Manga (a.k.a. Japanese comic) has become a worldwide popular art and entertainment because of its unique styles. Designers often apply screentone over initial line drawings to create particular textures or shadings, which is a key component contributing to the style of manga. Laying screen patterns is not an easy task, taking many factors into consideration, such as texture and underlying material property. It can be a tedious and time-consuming task, especially for novices. Thus, an automatic tool for screentoning on line drawings is in high demand for novices to add screentone easily. Though manga has attracted a lot of research works recently, such as manga colouration [QWH06, FHOO17], screen patterns removal [LLW17], and manga generation [PQW*08, XLLW20, ZWF*21], there are few works on screentone generation from line drawings. The closest research line is generating manga from colourful photographs or western comics [QPWH08, XLLW20]. However, these works rely on colour and texture affinity to segment regions for assigning the patterns, which is not applicable to sketch lines lacking texture information.

In this work, we aim to design a tool for generating screentone from line drawings. Rather than creating everything from scratch in a deterministic way, as there may exist plenty of possible solutions for a single line drawing, we choose to use example manga to aid the process. Taking reference manga as input provides flexible user controllability on the generated results without much effort. This shares some similarity to existing style transfer methods [GEB16,

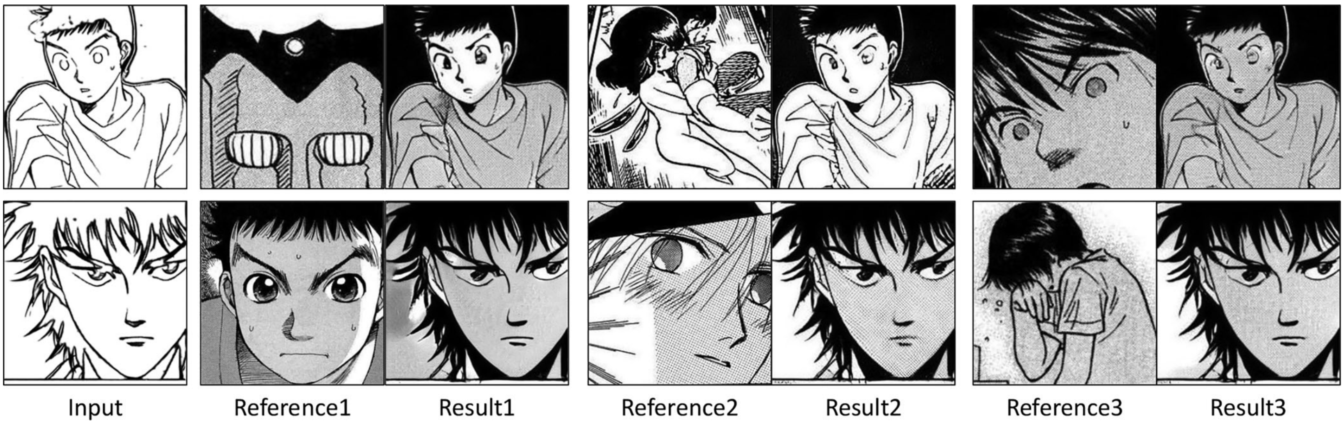

Figure 1: *Reference-based screentone transfer results by our method. Given a line drawing (i.e. input), and a reference manga image, our model can transfer the screentone pattern from the reference to the line drawing. We show each group of the results with three different reference manga images per row. (Better viewed in a digital version with zoom function.)*

LYY*17, ZPIE17, LKL*20, ZZZ*21]. But with the special structure of screentone, directly using these methods may easily generate serious artefacts.

However, example-based screentone transfer from line drawing imposes many challenges. First, how to extract accurate screentone style from the reference manga is unknown. The reference usually contains multiple different screen patterns, and the patterns are interleaved with the initial sketch lines. Second, it is difficult to measure the similarity between the reference and the target line drawing, especially the reference image already contains the screen patterns. Besides, the semantics gap exists as it is hard to always find a perfect match. Third, the quality of screentone is very sensitive to small variations, such as the level of grayscale and incomplete coverage. In this way, the transfer method needs to be taken carefully to keep the same details as the reference and achieve satisfying results.

To address the above challenges, we propose a novel framework, called Reference-based Screentone Transfer Network (**RSTN**), transferring the screentone from a reference manga to the input line drawing with high quality (see Figure 1). We first disentangle the screentone style from the 2D reference image through a 1D *stylegram*. Just like colour histogram, stylegram encodes necessary pattern style, discarding the disturbed spatial information. Stylegram can control the generated result in a meaningful way, demonstrating its flexibility and superiority through our experiments. To ensure a correct relationship between the reference and input line drawing, we introduce a method to find patch-correspondence in the deep feature space. Accordingly, we first synthesize the line drawings from existing manga images during training. Instead of building correspondence directly on the input pair (i.e. the reference and the line drawing), we calculate the patch-correspondence between the reference and the original manga image used to generate the line drawing. In other words, our model learns semantic correspondence implicitly rather than explicitly in a data-driven manner. Such implicit learning makes the model more robust for any reference-input pairs, without being constrained by scenes and hand-crafted rules. Directly transferring the screen patterns induces artefacts and inconsistent results, and we further propose a pattern regularization loss to mitigate this problem.

Through extensive experiments, we have demonstrated the effectiveness of our framework and design choices. To show the flexibility of our framework, we have exhibited that diverse high quality manga results can be generated by altering the reference. Our contributions can be summarized as follows.

- To the best of our knowledge, we are the first work on example-based manga screentone generation from line drawings. We propose a Reference-based screentone transfer network (RSTN), allowing users to control the screentone generation through references.
- To maintain good results, we design a stylegram method for extracting pattern style from the references; a patch-correspondence method for finding the correlation between input pairs; and a pattern regularization loss.
- To validate our framework, we have done a set of quantitative and qualitative experiments including a user study. The proposed RSTN effectively transfers the screentone to the target line drawings and is flexible to be used for manipulating the results based on the reference.

2. Related Work

2.1. Manga generation

Traditional manga production is mainly artificial. Artists first draft outlines and structural lines, then manually select appropriate pre-print screen sheets to fill regions to get the final manga. Many works have been done to shorten this workflow. Qu et al. [QPWH08] imitated manga production workflow to generate manga from a colour photograph by screen matching automatically. With the development of deep learning and the publication of large-scale manga dataset [AFO*20], many automatic manga generation methods are proposed. Xie et al. [XLLW20] proposed a screentone variational autoencoder to generate manga from western colour comics. Zhang et al. [ZWF*21] generated manga from illustration via mimicking manga creation workflow. Different from their works on generating manga from a colour image, our proposed method generates manga from drawing lines which are information-scarce and has its unique challenges.

2.2. Reference-based image translation

Automatic manga production has a limitation that users cannot manipulate the results with their desired manga style. Unfortunately, there is no such work available. We can regard our task as a reference-based image translation task that translates the input image based on a reference image. Gatys et al. [GEB16] proposed a neural algorithm to translate the style from one image to another one by online optimization, which takes time to process. Huang et al. [HB17] introduced an adaptive instance normalization (AdaIN) layer that aligns the mean and variance of the content features with those of the style features to translate style in real-time. On the topic of colourization, there are some works [HCL*18, SLWW19, LKL*20], that use an existing colorful image as a reference to colourize a greyscale image. Zhang et al. [ZZC*20] and Zhou et al. [ZZZ*21] propose a general framework for exemplar-based image translation, which translates abstract semantic content (edge map or pose keypoints) to a photo-realistic image. However, these two methods require the reference image with the same semantic content. If the input and the reference images have a distinct difference in semantic content, the translated results can generate serious artefacts. There are also some works to study image translation in specific domains [HLBK18, LTM*20], but unfortunately, they cannot be adapted to our screentone translation easily.

2.3. Segmentation-based manga manipulation

Many previous works rely on screentone segmentation techniques for manga manipulation, retrieval, and colourization [YHL*16, TIA19, CC16, QWH06]. One of the most common ways is to utilize procedural rules for segmentation, such as clustering based on binarization image or continuity of solid regions. After obtaining screentone segmentation, Yao et al. [YHL*16] demonstrated the applications including pattern manipulation, deformation, and lighting addition. Tsubota et al. [TIA19] trained a screentone label generator and filled screentone after obtaining screentone regions. Chu et al. [CC16] proposed two screentone features and combined them with panel features for manga retrieval. Qu et al. [QWH06] colourized the segmented regions using various methods. These works produce relatively good results but can only limit to a certain design scenario. In our work, we obtain segmentation implicitly which is more flexible and more suitable for finding semantic correspondence.

3. Overview

We denote the target line drawing and its corresponding manga image as x_1, y_1, and the reference as y_2. Our model (Figure 2) aims to transfer the screentone from reference y_2 to the target line drawing x_1. The result $\hat{y}_2$ should follow the screen style distribution of reference while filling in a semantically meaningful way into the regions of line drawing. Especially, except for the structural lines and solid white, we regard any part of manga as screentone, including solid black [LLW17]. However, asking designers for creating and sharing a large set of paired data (X, Y) is impractical. How to extract the screentone style and map to the target drawing is unknown a prior and cannot be trained in a fully-supervised way. Treating it as an unpaired image translation task [ZPIE17, LBK17] is inappropriate since it can only generate a determined version without further control by the given example.

To mitigate this issue, we synthesize a dataset by extracting the line drawings from existing manga images. Note that during training, each manga image y_1 has its corresponding line drawing image x_1, as x_1 is synthesized from y_1 using the latest manga screentone removal method [LLW17]. In this way, our model is possible to learn how to add screentone based on semantic content (paired path, the red line in Figure 2) while learning how to transfer the pattern style from the reference (unpaired path, the green line in Figure 2). To achieve the goal of our problem, we address three major challenges.

Our first step is to extract the screentone from the reference. The screen patterns are fused together with the sketch lines, making it hard to extract them in a faithful way. Besides, our extracted code should be disentangled from the spatial location and low-level content details as they can be very different between the reference and the target. To deal with this problem, we propose to extract the screen patterns through stylegram, which is a 1D vector encoding spatial-independent style information.

The next step is to apply screen patterns and the key is to know the mapping between the reference y_2 and target x_1 for finding suitable locations. Directly calculating the mapping based on line drawing without screentone and manga with screentone is difficult under different distributions. We thus turn to a synthetic pair for help by measuring the similarity between the reference y_2 and the corresponding manga of target y_1 in the same domain based on a patch-correspondence approach.

The patterns of screentone are very delicate and subtle, and direct transfer can generate serious artefacts on the textual details. To solve this problem, we add a novel pattern regularization loss to force model for transferring consistent screentones. In the next section, we will introduce our Reference-based Screentone Transfer Network in detail.

4. Reference-based Screentone Transfer Network

The framework of our Reference-based Screentone Transfer Network (RSTN) is shown in Figure 2. The model mainly consists of three components: a Content Encoder (CEnc), a Stylegram Encoder (SEnc), and a Stylegram Integration Decoder (SIDec). We follow Swapping AutoEncoder [PZW*20] to design our CEnc and SEnc, and StyleGAN2 [KLA*20] to design our SIDec. We first delve into the extraction and fusion of stylegram, then introduce the elaborated loss used for training.

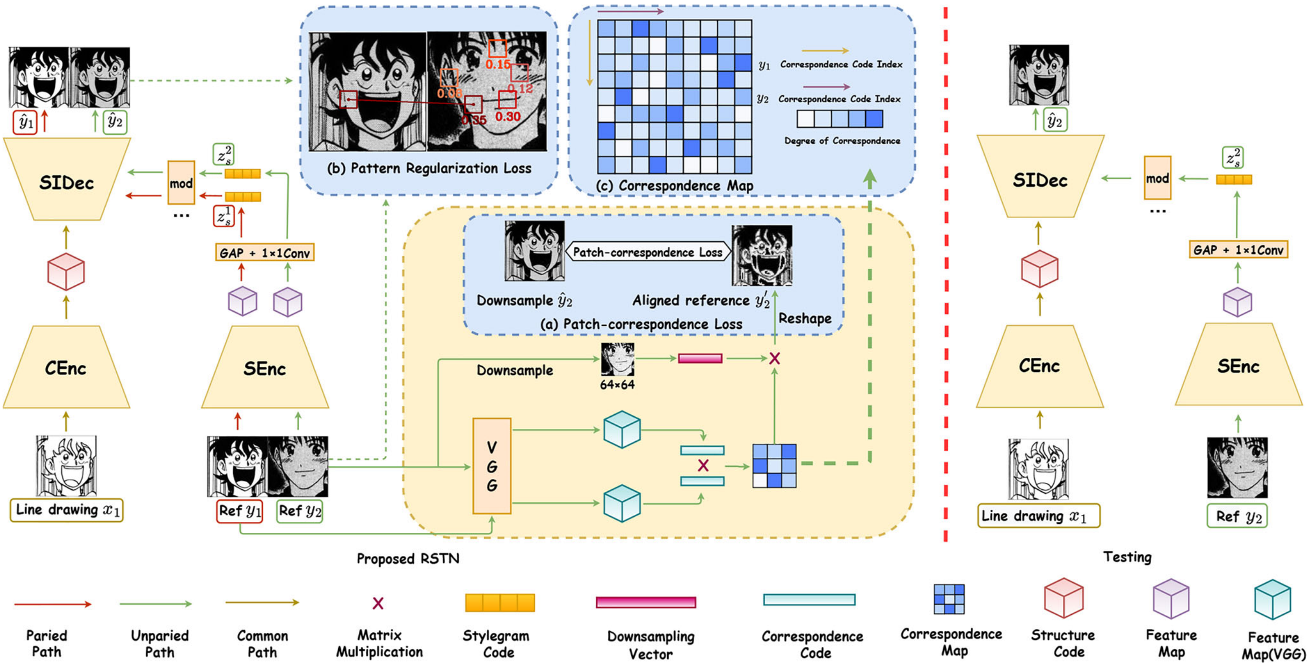

Figure 2: *The framework of our proposed Reference-based Screentone Transfer Network (RSTN). During training, RSTN is trained in a hybrid way using both paired data (flow indicated by red arrows) and unpaired data (flow indicated by green arrows). Given a manga image y_1 with the reference y_2, we first synthesize the input line drawing x_1 from y_1 and encode x_1 through* CEnc *for extracting the content code. Then y_1 and y_2 are sent to* SEnc *for extracting stylegram code z_s^1 and z_s^2. The content code and these two stylegram codes are fed into* SIDec *to get transferred results $\hat{y}_1$ and $\hat{y}_2$. A patch-correspondence loss (a) and a pattern regularization loss (b) are proposed to ensure the quality of generated results. (c) is the correspondence map. During testing, users only need to provide the input line drawing x_1 and the reference y_2 for obtaining the final result. Though there is no corresponding y_1, our* SEnc *has learned how to extract meaningful stylegram code from the reference manga y_2 directly.*

4.1. Stylegram extraction and fusion

Given the input line drawing x_1 and the reference manga y_2, our model aims at learning a mapping function $G(x_1, y_2)$ to transfer screentone style from reference to the input line drawing. There are multiple ways to fuse the inputs together. Recalling the approaches of works taking multiple images as inputs, the most common way is to concatenate inputs along the channel dimension before sending them into the encoder or generator [LRS*21, XSA*18, HYFW19]. However, as we mentioned in the last section, blending spatial information deteriorates the pattern style we want to obtain. Rather than using the raw reference image in 2D dimension to deliver style features, we encode the style as a 1D vector (i.e. stylegram) through an encoder before fusing it with content information extracted from the target line drawing.

More specifically, SEnc encodes a reference image as stylegram code which represents screentone style. And the stylegram code is disentangled from the spatial location and low-level content details as they can be very different between the reference and the target line drawing. Given a reference manga image $y_2 \in \mathbb{R}^{H \times W}$, where H and W represent the reference manga image's height and width, respectively, the SEnc outputs a stylegram code $z_s^2 \in \mathbb{R}^{1 \times d}$, where d denotes the length of stylegram code:

$$z_s^2 = SEnc(y_2). \tag{1}$$

Since the stylegram code is designed to be agnostic to positional information, stylegram is derived by convolutional layers using reflection padding or no padding, followed by a global average pooling and a dense layer. After extracting content code from CEnc, the stylegram code is fused with the features $\mathcal{F}$ from SIDec using the weight modulation (i.e. "mod" in Figure 2) to inject screentone style.

4.2. Patch-correspondence loss

We first build the correspondence between the paired manga y_1 and the reference y_2 to provide guidance for screen pattern transfer. Existing style transfer methods often rely on perceptual loss [JAFF16, HB17] for controlling style. However, such a global constraint does not consider the locality of the screentone pattern, which is not suitable for our task. Since the screentone pattern is often used in a local manner (e.g., the pattern used in the hair region is different from the pattern used on the face), we choose to adopt a patch correspondence way to keep the style consistency between the result and the reference.

We find that the feature from a pretrained VGG-16 can model the correspondence well. We thus take the feature $f \in \mathbb{R}^{h \times w}$ from the `relu3-1` layer of VGG-16, where h and w represent the feature map's height and width respectively, and each entry in the feature

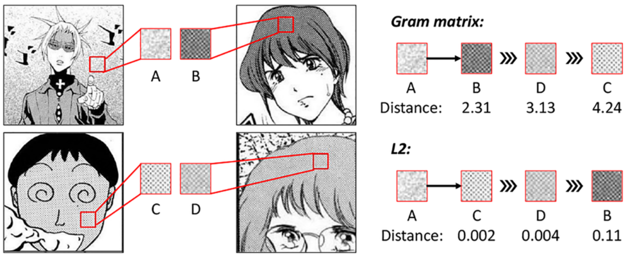

Figure 3: *Gram Matrix versus L_2 for measuring screentone pattern similarity. We use patch A to retrieve patches in the set of B, C, D, and display the distance calculated by the two metrics below the index. Gram matrix fails to capture intensity-aware similarity among patterns.*

map has a receptive field with the resolution of 24 × 24. We compute a correlation matrix $\mathcal{M} \in \mathbb{R}^{hw\times hw}$ between the feature f_1 of y_1 and f_2 of y_2:

$$\mathcal{M}(u,v) = \frac{f_1(u)^T f_2(v)}{||f_1(u)||||f_2(v)||}, \tag{2}$$

where $f_1(u)$ and $f_2(v)$ represent the channel-wise features of f_1 and f_2 in position u and v. The correlation matrix is normalized along the second dimension (i.e. v) with a softmax operation. We then obtain an aligned reference y_2' by multiplying the normalized correlation matrix with a downsampled y_2:

$$y_2'(u) = \sum_v \mathrm{softmax}_v(\mathcal{M}(u,v))DownSample(y_2(v)). \tag{3}$$

In other words, each pixel in y_2' is a linear combination of y_2 based on the normalized correlation matrix to y_1. In this way, the aligned reference y_2' keeps both the structure of y_1 and the screentone distribution of y_2 (Figure 2). We treat it as a fake ground truth to guide the prediction with patch-correspondence loss:

$$\mathcal{L}_{pc} = ||DownSample(\hat{y}_2) - y_2'||_1. \tag{4}$$

4.3. Pattern regularization loss

As the training goes on, the screentone transferred on the line drawing becomes more apparent. However, it still lacks a lot of details. To ensure the quality of generated screentone, we add a pattern regularization loss at the midpoint τ of the training process. For each patch $p_i \in \hat{y}_2,\ i = \{1,2,\dots,n\}$ of the generated result, where n is a positive integer, we aim to seek the most similar patch in the reference for regularization. The most straightforward way is using a gram matrix as it is a common way to measure style similarity [GEB16]. However, we find the L_2 distance is a more reasonable metric, as the gram matrix is used to measure the relative relationship among pixels. For the screentone pattern, whose local intensity is the same within a local patch, the gram matrix will overlook the intensity, while L_2 is intensity-aware, better modelling the style similarity in our case. To validate this, we show an example in Figure 3, and we can find that L_2 can better measure style similarity. We thus find the most similar patch denoted as $\{p_i'\}$ through:

$$p_i' = \underset{j}{\mathrm{argmin}}\, ||p_j - p_i||_2, \tag{5}$$

where $\{p_j\}$ are all patches from y_2.

We then use this most similar patch p_i' to guide the prediction and define the loss as:

$$\mathcal{L}_{pr} = \sum_{i=1}^{n} ||p_i - p_i'||_1. \tag{6}$$

This loss can force the details of the generated patch to be consistent with the reference patch.

4.4. Full objective

There are some other losses that are necessary to guarantee the performance of our model.

4.4.1. *Reconstruction loss*

If the model receives the corresponding pair x_1, y_1, it should transfer the style correctly and obtain the exactly same manga image as y_1. We use the pixel-wise L_2 loss for reconstructio n, formulated as:

$$\mathcal{L}_{rec} = ||y_1 - \hat{y}_1||_2. \tag{7}$$

4.4.2. *Adversarial loss*

To make sure that the reconstructed manga and the transferred results look realistic, aligning with the distribution of the manga domain. We also add the adversarial loss, defined as:

$$\mathcal{L}_{GAN,rec}(G) = \mathbb{E}_{x_1\sim X, y_1\sim Y}\big[-\log\,(D(\hat{y}_1)\big], \tag{8}$$

$$\mathcal{L}_{GAN,ref}(G) = \mathbb{E}_{x_1\sim X, y_2\sim Y}\big[-\log\,(D(\hat{y}_2)\big], \tag{9}$$

where D denotes discriminator.

4.4.3. *Patch adversarial loss*

Following the work [PZW*20], we also use the co-occurrent patch adversarial loss to encourage the network to learn a factored screentone style representation, which can be presented as:

$$\begin{aligned}&\mathcal{L}_{GAN,patch} = \\ &\mathbb{E}_{x_1\sim X, y_2\sim Y}\big[-\log\big(D_p(crop(\hat{y}_2), crops(y_2))\big)\big],\end{aligned} \tag{10}$$

where *crop* selects a random patch and *crops* is a collection of multiple patches, and D_p denotes the patch discriminator. The underlying idea is that no matter what the content sketches are, the screentone style distribution stays the same as the reference, and makes patch discriminator hard to differentiate.

4.4.4. *Sketch Line Loss*

To maintain the sketch lines in the generated result the same as the input, we add a pixel-wise sketch line loss:

$$\mathcal{L}_{sl} = ||x_1 * B - \hat{y}_2 * B||_1, \tag{11}$$

where B denotes a binary mask of x_1, and it sets to 1 if the corresponding pixel in x_1 is 'black', 0 if it is 'white', * denotes the element-wise multiplication.

4.4.5. *Total loss*

We obtain our final loss for training as:

$$\begin{aligned}\mathcal{L}_{total} &= \lambda_1\mathcal{L}_{pc} + \lambda_2\mathcal{L}_{pr} + \lambda_3\mathcal{L}_{rec} + \lambda_4\mathcal{L}_{GAN,rec} \\ &+ \lambda_5\mathcal{L}_{GAN,ref} + \lambda_6\mathcal{L}_{GAN,patch} + \lambda_7\mathcal{L}_{sl},\end{aligned} \tag{12}$$

where $\{\lambda_i\}$ denotes the weights for balancing loss terms and the default of $\{\lambda_i\}$ is 1.

5. Experiments

In this section, we first describe the experimental setting. We then compare our model with state-of-the-art methods in related domains both qualitatively and quantitatively. Ablation studies have been conducted to examine the effectiveness of each key design choice. Finally, we show the flexibility of our model on several design scenarios.

5.1. Implementation details

We use Manga109 Dataset [AFO*20] for training and evaluation. We use the resolution of 256×256 for all our inputs. Especially, in our experiment, we set $H = 256$, $W = 256$, $d = 2048$, $h = 64$, $w = 64$, and $n = 16$. Meanwhile, we follow StyleGAN2 [KLA*20] to design our discriminators and Swapping AutoEncoder [PZW*20] to design the patch-discriminator. We only use the *pattern regularization loss* at the last 50% of iteration during training (i.e. $\tau = 400{,}000$). Moreover, we choose Adam [KB15] optimizer with a learning rate of 0.002 for optimization and set the weights of each loss term to 1.0. We train the model with the batch size of 1 on a single NVIDIA GeForce RTX 2080Ti GPU for about 3 days.

5.2. Network architecture details

Our RSTN contains mainly three subnets: Content Encoder, Stylegram Encoder, and Stylegram Integration Decoder. We introduce them separately next.

5.2.1. *Encoders*

Our encoders consist of Content Encoder and Stylegram Encoder. For the Content Encoder, we follow Swapping AutoEncoder [PZW*20] by designing 4 downsampling residual blocks and two convolutional layers. Given a 256 × 256 line drawing, our Content Encoder outputs a content code that is of dimension 8×32×32. For the Stylegram Encoder, we design 4 downsampling residual blocks, two convolutional layers, a global average pooling layer and a 1×1 convolutional layer. Given a 256 × 256 reference manga image, our Stylegram Encoder outputs a stylegram code that is of dimension 1×1×2048.

5.2.2. *Stylegram integration decoder*

We follow StyleGAN2 [KLA*20] to design the architecture of our Stylegram Integration Decoder. Content code is passed into the decoder which consists of 4 residual blocks and 4 upsampling residual blocks and stylegram code is injected into the decoder by the modulation layer [KLA*20].

5.3. Dataset processing

To process the data for training, we collected 80 manga books from 109 after removing those including too many blank spaces. We then followed the annotation process of Manga109 Dataset [AFO*20] by cropping manga images with the label "body" after filtering out pages with text. By discarding the monochrome and low-resolution images, we obtained 5200 manga images in total. Besides, we used the latest manga screentone remove method [LLW17] to get paired line drawings. And we split the dataset into 4800 and 400 for training and testing, respectively.

5.4. Compared to state-of-the-art methods

5.4.1. *Compared methods*

As far as we know, there is no existing work on generating manga from line drawings. Therefore, we chose six state-of-the-art works in the related domains for comparison, including the traditional neural style transfer method [GEB16], unsupervised image2image translation [ZPIE17, LTM*20], and exemplar-based image2image translation [HB17, HLBK18, LKL*20]. We compare our method against them both qualitatively and quantitatively. All compared methods are re-trained using officially released code on the same dataset for fair comparisons.

5.4.2. *Qualitative comparison*

We show the qualitative comparison in Figure 4. For neural style transfer methods NST [GEB16] and Adain [HB17], they can only transfer screentone in a global manner, without considering screen details of the reference or semantic content of the target line drawing. For CycleGAN [ZPIE17], since it can only generate a specific type of style, there is no adaption to the input references. Though methods, such as MUNIT [HLBK18] and ASCFT [LKL*20], can vary the style of results based on the input references, the transferred manga fails to satisfy the requirement of semantics. For example, in the second last row of Figure 4, the screentone is taken on the girl's face which is unsatisfied. This is also the case for DRIT++ [LTM*20]. Besides, these methods often generate incomplete screentone in a certain region. In the same example of the second last row, the screentone is missed in the sub-region of the girl's hair.

5.4.3. *Quantitative comparison*

We first compute two metrics: PSNR and FID, that are commonly used in existing works on image generation and translation [KLA19, DLHT15, NT20]. Since the manga image can be reconstructed by taking itself as the reference, we send the manga image with its line drawing as inputs, and compare the output with the manga image

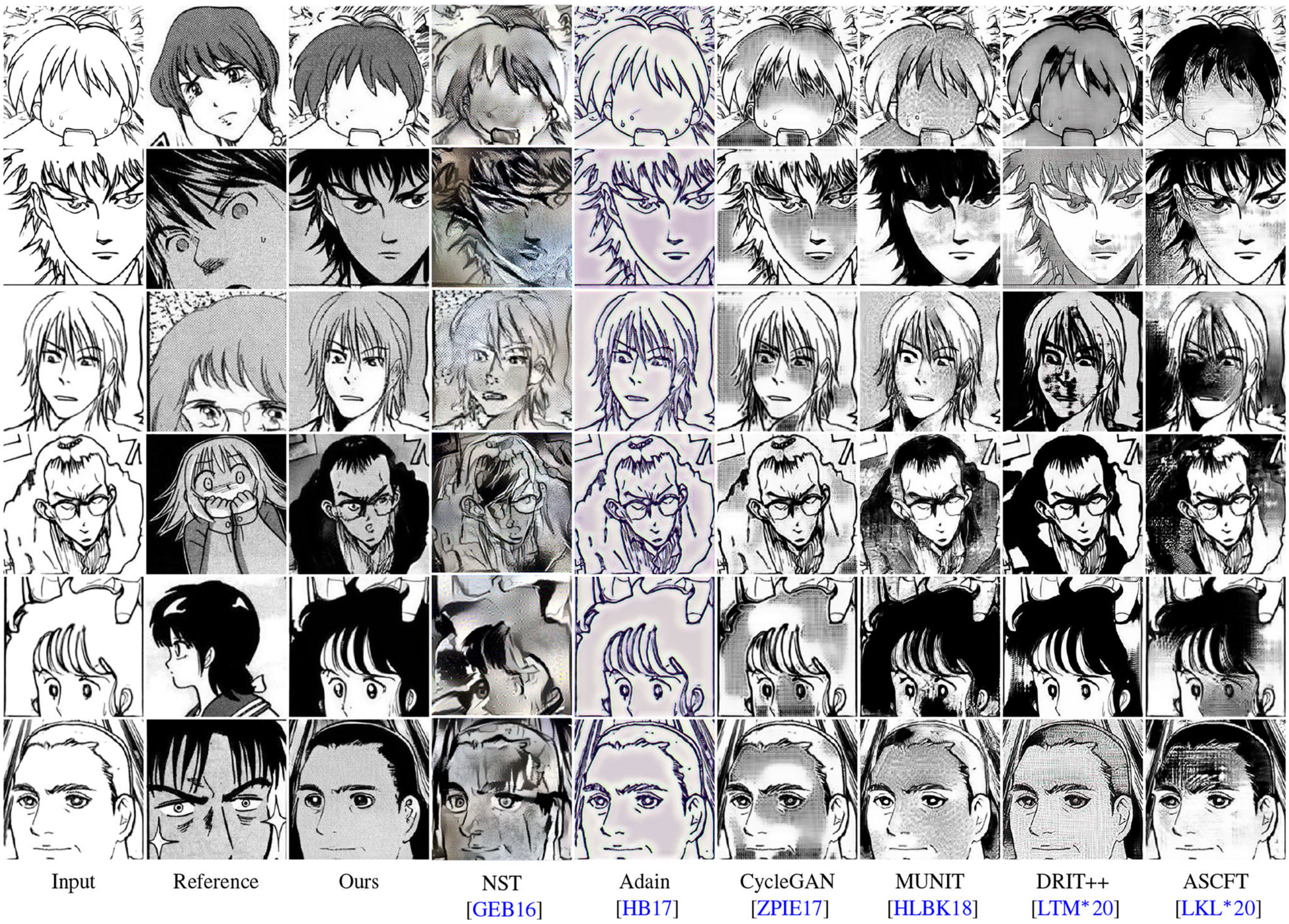

Figure 4: *Qualitative comparison of our method with state-of-the-art methods. Notice that results from other methods have serious artefacts on either structure deformation or appearance consistency.*

itself for calculating the PSNR. A higher PSNR score indicates a better performance.

Besides, there is no existing metric can be used to measure whether the transferred results have consistent screentone style with the reference, which is one of the key factors for our model design. We thus introduce a novel metric for measuring the screentone consistency between two manga images. Benefiting from our stylegram code, we can design a metric called style consistency score (*SC_score*) to measure the screentone style consistency. More specifically, given two manga images y_1 and y_2, we can compute the *SC_score* with the distance over stylegram codes. The metric can be defined as:

$$\begin{aligned} SC\_score &= ||SEnc(y_1) - SEnc(y_2)||_2 \\ &= ||z_s^1 - z_s^2||_2. \end{aligned} \tag{13}$$

In our paper, we compute *SC_score* between the transferred screentone result and the reference image.

We show the quantitative of different methods over three metrics in Table 1. We can see that our model outperforms other methods by a large margin, which indicates our transferred results are more realistic and consistent with reference images than results generated by other methods.

Table 1: *Quantitative comparison of our method with state-of-the-art methods. ↓ denotes the lower the better and vice visa. The best results are marked in **bold**. (SC_score: style consistency score.)*

Methods	*SC_score*↓	FID↓	PSNR↑
NST [GEB16]	148.627	241.64	11.94
Adain [HB17]	147.546	266.76	11.02
CycleGAN [ZPIE17]	117.768	207.16	11.04
MUNIT [HLBK18]	112.146	214.33	9.89
DRIT++ [LTM*20]	130.414	204.02	9.37
ASCFT [LKL*20]	97.525	213.56	10.99
Ours	**56.852**	**183.64**	**14.20**

5.5. User study

To further demonstrate the effectiveness of our model, we introduce a user study to evaluate our results with both novices and profes-

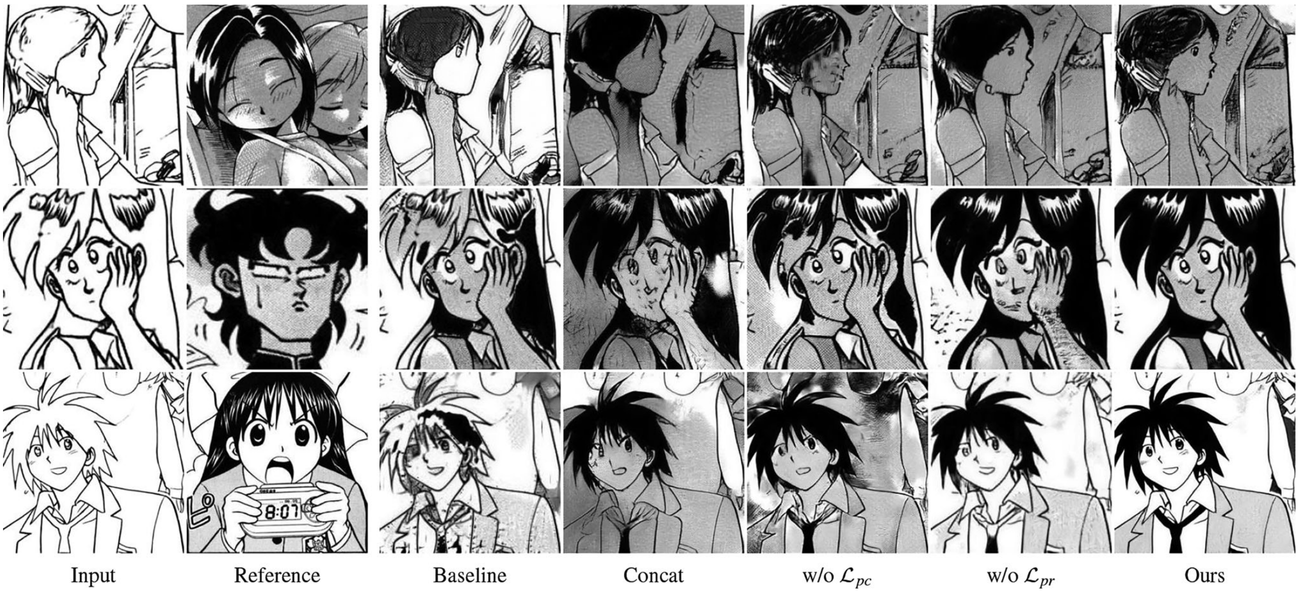

Figure 5: *User study results. The statistics of the user study from novices and professionals, respectively, demonstrate our method surpasses all the competitors by a large margin in terms of screentone quality, screentone consistency, and sketch consistency, from both the novice and the professional perspective.*

sionals. There are mainly three questions. (1). **Screentone Quality**: it cares about the perceptual quality of transferred results, evaluating the visual quality solely on the generated images. (2). **Screentone Consistency**: it evaluates the screentone style consistency between transferred results and reference images. Given a reference input, participants are asked to choose the most consistent one in screentone style from transferred results acquired by different methods. (3). **Sketch Consistency**: it evaluates the sketch lines consistency between transferred results and input images.

5.5.1. *Evaluate with novices*

We invited 30 participants with majors in Computer Science who have read many mangas but have no design experience. Every task contains 20 design cases and each with the three above questions for evaluation. Transferred results from different methods in each design case are presented in a random order for a fair comparison. For the study, we got 600 answers from novices in total.

5.5.2. *Evaluate with professionals*

We invited 20 participants with majors in Art who have professional painting experience. They were also asked to answer the same three questions above. And we got 400 answers from professionals in total.

We show the results in Figure 5. Both novices and professionals prefer our method over the others by a large margin, demonstrating the effectiveness of our method in practical usage. We also interviewed three participants: P1 (Novice), P2 (Novice), and P3 (Professional with painting experience). These are some comments for our work. P1: *"This is an attractive work."*; P2: *"I want to use it to transfer screentone to my line drawings."*; P3: *"The automatic screentone transfer still has a lot of room for progress, but it is an interesting attempt."* The above comments also demonstrate the effectiveness of our method.

5.6. Ablation study

In this subsection, we analyse the efficacy of the main components in the network architectures and the losses design of the proposed method. We first set a baseline (**Baseline**) by retraining our model without the proposed key losses (i.e. $\mathcal{L}_{pc}$ and $\mathcal{L}_{pr}$). Then, to demonstrate the importance of our proposed stylegram code, we also set up a method by directly concatenating the inputs together before sending them into the encoder, called **Concat**. Besides, we compare our full model with two other variants: (1)**w/o** $\mathcal{L}_{pc}$, by removing the patch-correspondence loss; (2) **w/o** $\mathcal{L}_{pr}$, by removing the pattern-regularization loss.

The qualitative comparisons of different variants are shown in Figure 6. From the results, we can see that the performance degraded with serious artefacts or misalignment to reference if removing any of our key components. Meanwhile, if we use the concatenation method to extract the screentone style of reference rather than our proposed stylegram code, the transferred result is not good with deformable sketch lines and less-vivid pattern style. Besides, the image transferred by the variant (**w/o** $\mathcal{L}_{pc}$) presents the unmatched screentone result, showing that our patch-correspondence loss can build correct correspondence between input and reference image. Similarly, without the $\mathcal{L}_{pr}$, the transferred result becomes less consistent with the reference.

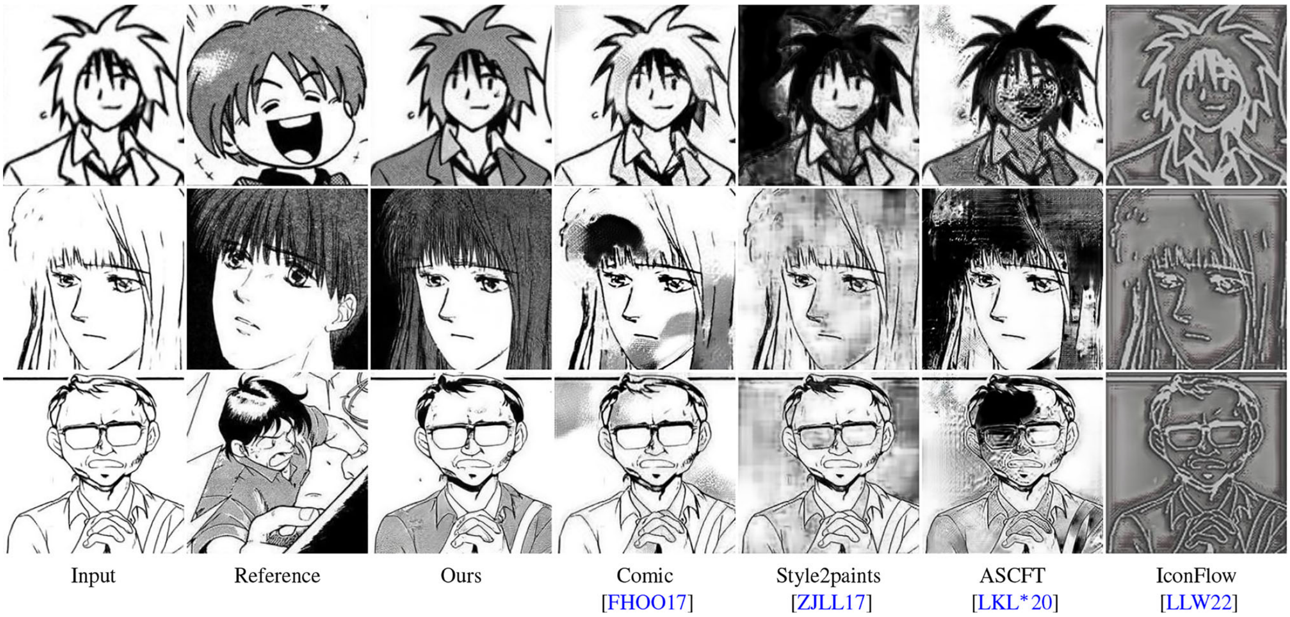

Figure 6: *Ablation study on network architecture(i.e. "**Baseline**" and "**Concat**") and loss function designs (i.e. "w/o $\mathcal{L}_{pc}$" and "w/o $\mathcal{L}_{pr}$"). Each of the components makes an essential contribution to the final quality of the results.*

5.7. Comparisons with colourization methods

Our work has a distinct focus on colourization. For reference-based colourization, the focus of the reference image is to ensure the consistency of global colours, using the histogram [FHOO17, LLW22]. Although regional semantic mappings are taken into account, the input and the inference do not enforce semantic correlations, and the goal is to assign colours correctly [LKL*20]. For our task, we need to consider the regularization of the texture. Locally, we need to make sure that the texture is consistent with reference, which is an important challenge for transferring screentone. The simple colourization model does not take this into account [ZJLL17]. For this reason, the problem of regularization cannot be solved simply by using reference-based colourization techniques. The comparisons are shown in Figure 7.

5.8. Reference-based screetone transfer

As a reference-guided framework, the results should be adapted and controlled by the reference in a meaningful way. We thus conduct an experiment by transferring a set of reference manga images to the same target line drawing and show the results in Figure 8. From the results we can see our model is able to extract different screentone patterns from different reference manga images and integrates screentone style with drawing lines well. This verifies the superiority of our proposed method.

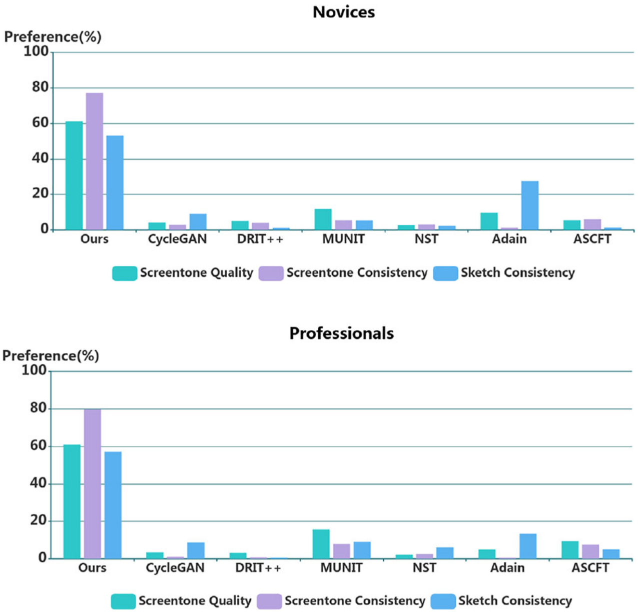

Figure 7: *Qualitative comparison of our method with state-of-the-art reference-based colourization methods. Notice that results from other colourization methods have serious artefacts either on structure deformation or on appearance consistency.*

5.9. Stylegram interpolation

To demonstrate the flexibility of our stylegram, we conduct an experiment on style interpolation with our stylegram codes. Since we use a stylegram code z_s to represent the screentone style of the manga, the style can be easily controlled by manipulating the stylegram code. Given two reference manga images, we first extract two stylegram codes, respectively. Then we apply a linear interpolation

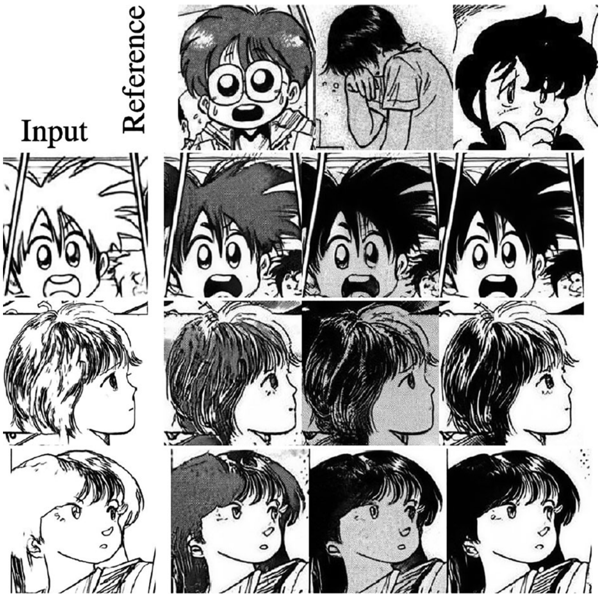

Figure 8: *Controlling screentone generation by reference images. Our method provides the capability of generating multiple plausible screentones by varying the references.*

between these two codes to form a new stylegram code for controlling the screentone. Figure 9 shows the interpolation results from two reference images. We can see that our model can control screentone transfer in a smooth and accurate way.

5.10. Generalization of our method

To demonstrate the generality of our model, we conduct experiments on different types of line drawings and references. First, we transfer the screentone to a larger scale of image, rather than the head only, as shown in Figure 10. The images contain a full body and more background semantics. However, our model can still achieve this task and achieve high-quality results with screentone patterns applied locally to suitable regions. Second, to examine that our method not only works on screentone patterns, we also test on manga images with unique abstract manga effects, and show the results in Figure 11. Though the patterns are dramatically different from those in the main paper and Figure 10, our model obtains high-quality results and transfers this unique style properly to the line drawings. Third, we test on line drawings containing more content not limiting to the human part. Especially, the animal line drawings are real. And the results in Figure 12 show that our model can achieve high-quality results on flower and animal line drawings. Besides, we also show the failure case in the last column of Figure 12. Our model does not explicitly consider the orientation so the orientation of screentone between input and reference is not consistent.

6. Conclusion and Limitation

In summary, we introduce a practical but underdeveloped research problem, that is, reference-based screentone transfer. To deal with this challenging task, we propose a novel Reference-based Screentone Transfer Network (RSTN), by encoding the screentone pattern from the reference as a *stylegram*. A set of calibrated loss functions, especially patch correspondence loss and pattern regularization loss, are defined to enable the training of our model. We have demonstrated the quality and controllability of our model through comprehensive experiments both qualitatively and quantitatively.

Though we have achieved promising results, there are still a few directions that can be studied in future works. First, as a data-driven method, the performance of our model is bounded by the scale of the dataset. The model may fail to obtain faithful patch correspondences and results if the semantic contents and screentones seldom appeared or screentones are complicated during training, such as highlight rendering and shadows. However, with the worldwide popularity, the manga community grows very fast, contributing more and more manga under common creative license. Since our model does not rely on manually created pairs for training, it is easily extended as the data grows. Second, our method cannot deal with high-resolution scenes (larger than 512 × 512) containing various types of screentone patterns, due to our implicit transfer setting. This problem can be solved with explicit region-level semantic cor-

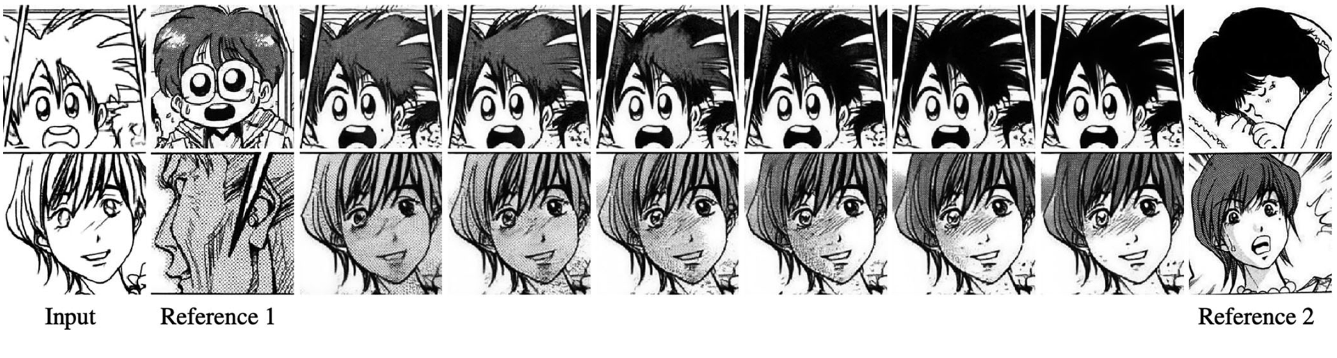

Figure 9: *Interpolation between two reference stylegram codes. We extract the stylegram codes from both Reference 1 and 2, and conduct linear interpolation to generate new code for controlling the screen patterns in between.*

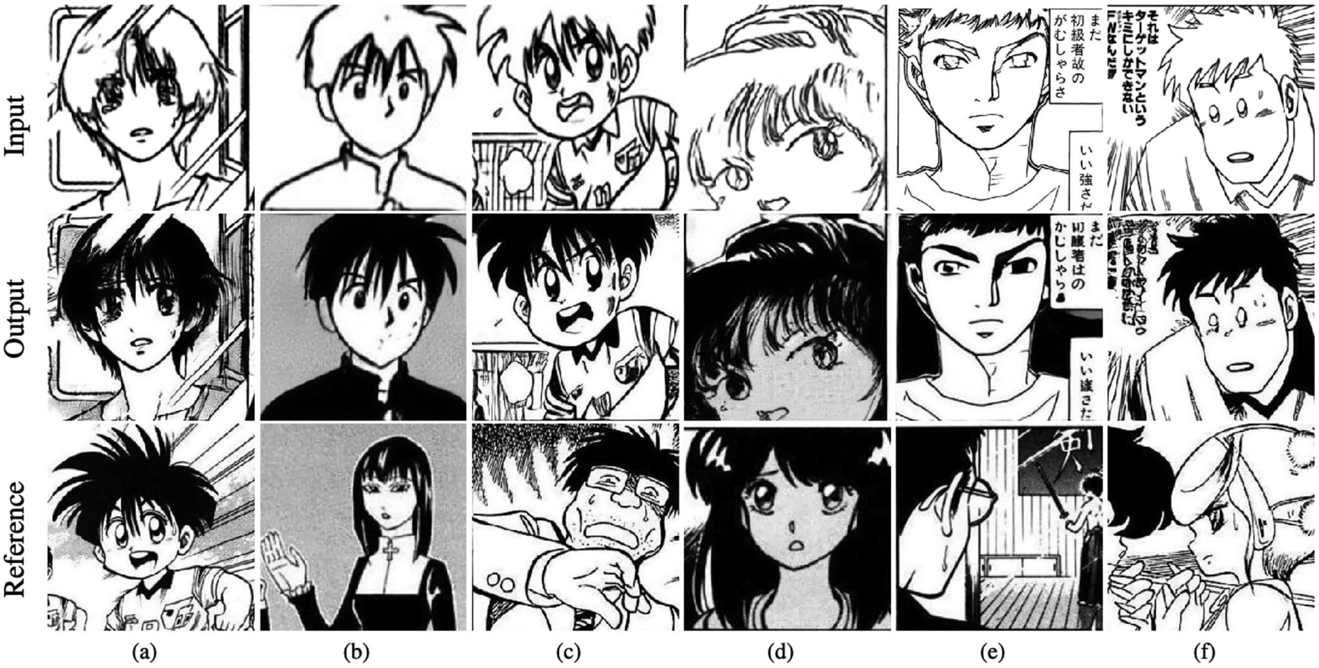

Figure 10: *Example results on line drawings containing more complex content rather than a body part (e.g. head only).*

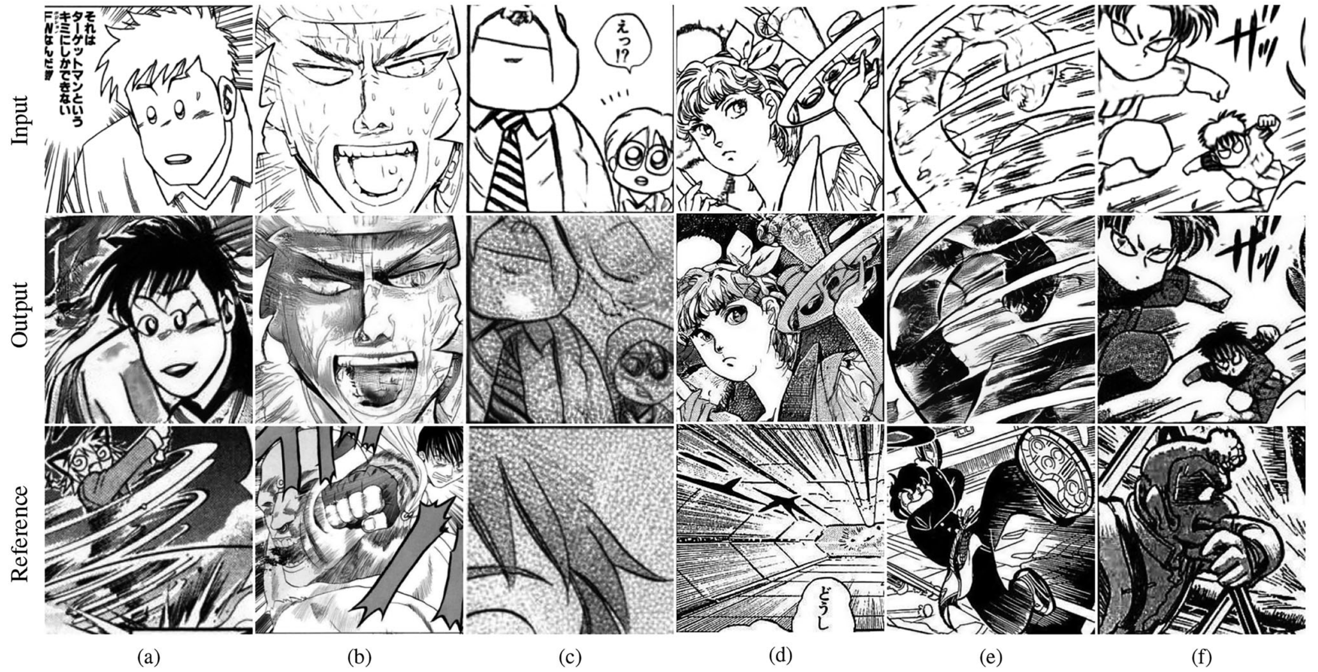

Figure 11: *Example results on images with abstract manga effects.*

respondences and then decomposing the large scenes into several sub-transfer tasks. We leave it in our future work.

As the first step to explore fully automatic screentone transfer, although our model does not meet the requirements of professional designers in terms of the degree of detail of effect, it is still very effective for amateur painters to simply add screentone as validated through experiments and the user study. When our work is applied to mobile applications, ordinary users can simply try adding screentone to their sketches. We think our work is a very valuable first

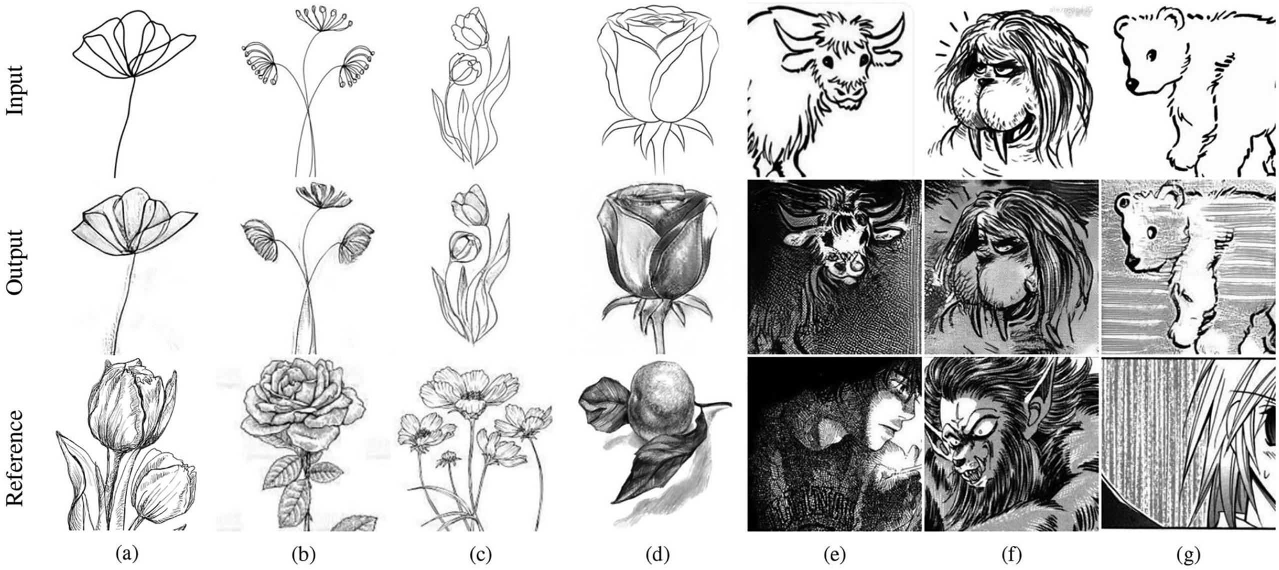

Figure 12: *Example results on non-face images. Especially, (a), (b), and (c) are synthesized line drawings, (d), (e), and (f) are real line drawings. Failure case is shown in (g) column.*

step. And we hope that it can inspire more future works in this research field.

Acknowledgements

This project is supported by the National Natural Science Foundation of China (No. 61972162); Guangdong Natural Science Funds for Distinguished Young Scholar (No. 2023B1515020097); and Guangdong Natural Science Foundation (No. 2021A1515012625).

References

[AFO*20] Aizawa K., Fujimoto A., Otsubo A., Ogawa T., Matsui Y., Tsubota K., Ikuta H.: Building a manga dataset "manga109" with annotations for multimedia applications. *IEEE MultiMedia 27*, 2 (2020), 8–18.

[CC16] Chu W.-T., Cheng W.-C.: Manga-specific features and latent style model for manga style analysis. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (Shanghai, China, 2016), IEEE, pp. 1332–1336.

[DLHT15] Dong C., Loy C. C., He K., Tang X.: Image super-resolution using deep convolutional networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence 38*, 2 (2015), 295–307.

[FHOO17] Furusawa C., Hiroshiba K., Ogaki K., Odagiri Y.: Comicolorization: Semi-automatic manga colorization. In *SIGGRAPH Asia Technical Briefs*. ACM, 2017, pp. 1–4.

[GEB16] Gatys L. A., Ecker A. S., Bethge M.: Image style transfer using convolutional neural networks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (Las Vegas, NV, USA, June 2016), IEEE.

[HB17] Huang X., Belongie S.: Arbitrary style transfer in real-time with adaptive instance normalization. In *IEEE/CVF International Conference on Computer Vision (ICCV)* (Venice, Italy, 2017), IEEE, pp. 1501–1510.

[HCL*18] He M., Chen D., Liao J., Sander P. V., Yuan L.: Deep exemplar-based colorization. *ACM Transactions on Graphics (TOG) 37*, 4 (2018), 1–16.

[HLBK18] Huang X., Liu M.-Y., Belongie S., Kautz J.: Multimodal unsupervised image-to-image translation. In *European Conference on Computer Vision (ECCV)* (Munich, Germany, 2018), Springer, pp. 172–189.

[HYFW19] Hu X., Yang K., Fei L., Wang K.: Acnet: Attention based network to exploit complementary features for rgbd semantic segmentation. In *IEEE International Conference on Image Processing (ICIP)* (Taipei, Taiwan, 2019), IEEE, pp. 1440–1444.

[JAFF16] Johnson J., Alahi A., Fei-Fei L.: Perceptual losses for real-time style transfer and super-resolution. In *European Conference on Computer Vision (ECCV)* (Amsterdam, Netherlands, 2016), Springer, pp. 694–711.

[KB15] Kingma D. P., Ba J.: Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)* (San Diego, CA, USA, 2015).

[KLA19] Karras T., Laine S., Aila T.: A style-based generator architecture for generative adversarial networks. In *IEEE/CVF*

Conference on Computer Vision and Pattern Recognition (CVPR) (Long Beach, CA, USA, 2019), IEEE, pp. 4401–4410.

[KLA*20] KARRAS T., LAINE S., AITTALA M., HELLSTEN J., LEHTINEN J., AILA T.: Analyzing and improving the image quality of stylegan. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (Seattle, WA, USA, 2020), IEEE, pp. 8110–8119.

[LBK17] LIU M.-Y., BREUEL T., KAUTZ J.: Unsupervised image-to-image translation networks. In *Advances in Neural Information Processing Systems (NeurIPS)* (Long Beach, CA, USA, 2017), Curran Associates Inc., pp. 700–708.

[LKL*20] LEE J., KIM E., LEE Y., KIM D., CHANG J., CHOO J.: Reference-based sketch image colorization using augmented-self reference and dense semantic correspondence. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (Seattle, WA, USA, 2020), IEEE, pp. 5801–5810.

[LLW17] LI C., LIU X., WONG T.-T.: Deep extraction of manga structural lines. *ACM Transactions on Graphics (TOG) 36*, 4 (2017), 1–12.

[LLW22] LI Y.-k., LIEN Y.-H., WANG Y.-S.: Style-structure disentangled features and normalizing flows for diverse icon colorization. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (New Orleans, LA, USA, 2022), IEEE, pp. 11244–11253.

[LRS*21] LIN S., RYABTSEV A., SENGUPTA S., CURLESS B. L., SEITZ S. M., KEMELMACHER-SHLIZERMAN I.: Real-time high-resolution background matting. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (Nashville, TN, USA, 2021), IEEE, pp. 8762–8771.

[LTM*20] LEE H.-Y., TSENG H.-Y., MAO Q., HUANG J.-B., LU Y.-D., SINGH M., YANG M.-H.: Drit++: Diverse image-to-image translation via disentangled representations. *International Journal of Computer Vision 128*, 10 (2020), 2402–2417.

[LYY*17] LIAO J., YAO Y., YUAN L., HUA G., KANG S. B.: Visual attribute transfer through deep image analogy. *ACM Transactions on Graphics (TOG) 36*, 4 (Jul 2017).

[NT20] NIZAN O., TAL A.: Breaking the cycle-colleagues are all you need. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (Seattle, WA, USA, 2020), IEEE, pp. 7860–7869.

[PQW*08] PANG W.-M., QU Y., WONG T.-T., COHEN-OR D., HENG P.-A.: Structure-aware halftoning. In *SIGGRAPH*. ACM, Los Angeles, CA, USA, 2008, pp. 1–8.

[PZW*20] PARK T., ZHU J.-Y., WANG O., LU J., SHECHTMAN E., EFROS A. A., ZHANG R.: Swapping autoencoder for deep image manipulation. In *Advances in Neural Information Processing Systems (NeurIPS)* (BC, Vancouver, Canada, 2020), Curran Associates Inc.

[QPWH08] QU Y., PANG W.-M., WONG T.-T., HENG P.-A.: Richness-preserving manga screening. *ACM Transactions on Graphics (TOG) 27*, 5 (2008), 1–8.

[QWH06] QU Y., WONG T.-T., HENG P.-A.: Manga colorization. *ACM Transactions on Graphics (TOG) 25*, 3 (2006), 1214–1220.

[SLWW19] SUN T.-H., LAI C.-H., WONG S.-K., WANG Y.-S.: Adversarial colorization of icons based on contour and color conditions. In *27th ACM International Conference on Multimedia* (Nice, France, 2019), ACM, pp. 683–691.

[TIA19] TSUBOTA K., IKAMI D., AIZAWA K.: Synthesis of screentone patterns of manga characters. In *IEEE International Symposium on Multimedia (ISM)* (San Diego, CA, USA, 2019), IEEE, pp. 212–2123.

[XLLW20] XIE M., LI C., LIU X., WONG T.-T.: Manga filling style conversion with screentone variational autoencoder. *ACM Transactions on Graphics (TOG) 39*, 6 (2020), 1–15.

[XSA*18] XIAN W., SANGKLOY P., AGRAWAL V., RAJ A., LU J., FANG C., YU F., HAYS J.: Texturegan: Controlling deep image synthesis with texture patches. In *IEEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (Salt Lake City, UT, USA, 2018), IEEE, pp. 8456–8465.

[YHL*16] YAO C.-Y., HUNG S.-H., LI G.-W., CHEN I.-Y., ADHITYA R., LAI Y.-C.: Manga vectorization and manipulation with procedural simple screentone. *IEEE Transactions on Visualization and Computer Graphics 23*, 2 (2016), 1070–1084.

[ZJLL17] ZHANG L., JI Y., LIN X., LIU C.: Style transfer for anime sketches with enhanced residual u-net and auxiliary classifier gan. In *IAPR Asian conference on pattern recognition (ACPR)* (Nanjing, China, 2017), IEEE, pp. 506–511.

[ZPIE17] ZHU J.-Y., PARK T., ISOLA P., EFROS A. A.: Unpaired image-to-image translation using cycle-consistent adversarial networks. In *IEEE/CVF International Conference on Computer Vision (ICCV)* (Venice, Italy, 2017), IEEE, pp. 2223–2232.

[ZWF*21] ZHANG L., WANG X., FAN Q., JI Y., LIU C.: Generating manga from illustrations via mimicking manga creation workflow. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (Nashville, TN, USA, 2021), IEEE, pp. 5642–5651.

[ZZC*20] ZHANG P., ZHANG B., CHEN D., YUAN L., WEN F.: Cross-domain correspondence learning for exemplar-based image translation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (Seattle, WA, USA, 2020), IEEE, pp. 5143–5153.

[ZZZ*21] ZHOU X., ZHANG B., ZHANG T., ZHANG P., BAO J., CHEN D., ZHANG Z., WEN F.: Cocosnet v2: Full-resolution correspondence learning for image translation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (Nashville, TN, USA, 2021), IEEE, pp. 11465–11475.