Unpaired Sketch-to-Line Translation via Synthesis of Sketches

Gayoung Lee*
NAVER Corp. Clova OCR
gayoung.lee@navercorp.com

Dohyun Kim*
Chung-Ang University
ppooiiuuyh@naver.com

Youngjoon Yoo
NAVER Corp. Clova AI
youngjoon.yoo@navercorp.com

Dongyoon Han
NAVER Corp. Clova AI
dongyoon.han@navercorp.com

Jung-Woo Ha
NAVER Corp. Clova AI
jungwoo.ha@navercorp.com

Jaehyuk Chang
NAVER WEBTOON Corp.
jaehyuk.chang@webtoonscorp.com

ABSTRACT

Converting hand-drawn sketches into clean line drawings is a crucial step for diverse artistic works such as comics and product designs. Recent data-driven methods using deep learning have shown their great abilities to automatically simplify sketches on raster images. Since it is difficult to collect or generate paired sketch and line images, lack of training data is a main obstacle to use these models. In this paper, we propose a training scheme that requires only unpaired sketch and line images for learning sketch-to-line translation. To do this, we first generate realistic paired sketch and line images from unpaired sketch and line images using rule-based line augmentation and unsupervised texture conversion. Next, with our synthetic paired data, we train a model for sketch-to-line translation using supervised learning. Compared to unsupervised methods that use cycle consistency losses, our model shows better performance at removing noisy strokes. We also show that our model simplifies complicated sketches better than models trained on a limited number of handcrafted paired data.

CCS CONCEPTS

• **Computing methodologies → Neural networks**; **Computational photography**; **Image processing**.

KEYWORDS

image-to-image translation, sketch simplification, neural networks, data synthesizing

ACM Reference Format:
Gayoung Lee, Dohyun Kim, Youngjoon Yoo, Dongyoon Han, Jung-Woo Ha, and Jaehyuk Chang. 2019. Unpaired Sketch-to-Line Translation via Synthesis of Sketches. In *SIGGRAPH Asia 2019 Technical Briefs (SA '19 Technical Briefs), November 17–20, 2019, Brisbane, QLD, Australia.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3355088.3365163

*The work is done when the authors were in NAVER WEBTOON Corp.

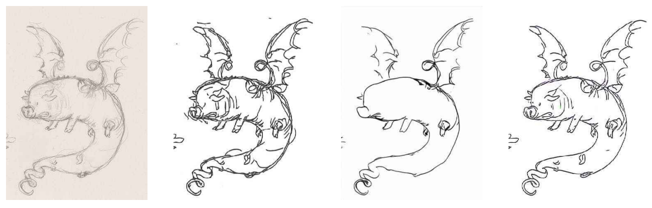

Figure 1: Visual comparison between MUNIT [Huang et al. 2018], MS [Simo-Serra et al. 2018a] and our method. MS and our method require only unpaired sketch and line image data. Image is copyrighted by David Revoy www.davidrevoy.com and licensed under CC-by 4.0.

1 INTRODUCTION

Sketching allows artists to quickly express artistic concepts without worrying details. It plays a fundamental role in the early stage for many design works such as cartoons, movies and product designs. After sketch designs are complete, artists need to simplify the sketches into clean lines due to coarseness and redundancy of sketch strokes, which is time consuming and hinders productivity. While this process seems an obvious task for human, automatic sketch simplification is challenging because there must be good reasoning to distinguish between detailed expressions to be preserved and noisy strokes to be removed.

Recently, Simo-Serra *et al.* [Simo-Serra et al. 2018a,b, 2016] presented deep learning-based methods that can directly simplify raster sketch images. Their methods can learn useful features automatically from data, and show robust performance on various sketch images. However, to acquire paired training data for supervised learning, they asked artists to inversely generate sketch images from line images, which is expensive and time consuming.

Meanwhile, unpaired sketch and line images are much easier to acquire than paired one from the web. Many prior works presented unpaired image-to-image translation models [Huang et al. 2018; Zhu et al. 2017] using unsupervised learning. These methods allow to learn desired image-to-image translation functions without expensive tasks of obtaining paired dataset. However, in sketch-to-line translation task, we found that cycle consistency losses used in these methods lead models to preserve undesired noisy sketch lines as shown in Figure 1 (b).

In this paper, we propose a method for automatic sketch-to-line translation using unpaired sketch and line images via synthesis of pair sketches. Instead of learning mapping functions between

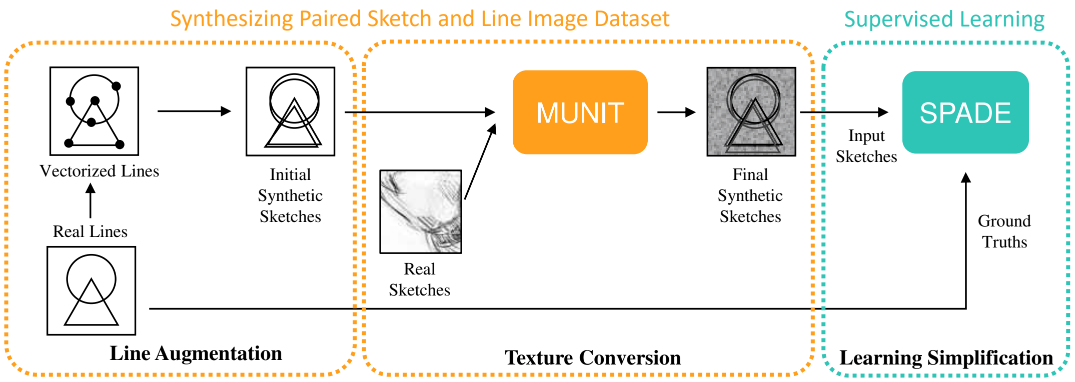

Figure 2: Overview of our method. Our method consists of two steps: synthesizing paired sketch and line images, and learning to simplify using the synthetic data.

unpaired domains at one go, our method consists of two steps: synthesizing sketches for paired data generation and supervised model learning with the synthetic data. For the synthesis of sketches, we first generate initial synthetic sketches by rule-based line augmentation that perturbs each Bézier curve individually by adding noises to its control points. We then refine the initial synthetic sketches using a multi-model unsupervised image-to-image translation (MUNIT) model to bridge the gap between textures in synthetic and real sketch images. Finally, we train a recent image-to-image translation model [Park et al. 2019] with our synthetic paired data to perform the sketch-to-line translation.

We show the effectiveness of our framework by comparing our method with existing sketch simplification methods and unsupervised image-to-image translation models. With extensive user studies and perceptual metrics, we show that our model can simplify complex actual sketches better than the prior works.

2 PROPOSED METHOD

In this section, we introduce the proposed method for unpaired sketch-to-line translation. Our method consists of two steps: synthetic pair sketch generation using unpaired sketch and line images, and training a model for sketch simplification. For synthetic sketch generation, we first create initial sketches using rule-based line augmentation, and then refine the sketches using the unsupervised image-to-image translation model [Huang et al. 2018] for realistic sketch textures. Using the synthetic paired sketch and line images, we train a slightly modified version of the Spatially-Adaptive Normalization(SPADE) model [Park et al. 2019]. Figure 2 visualizes the overview of the proposed method.

2.1 Rule-based Line Augmentation

In real sketches, each line segment is often represented by multiple noisy strokes whose location and slope are slightly differ with the line segment. To generate realistic and diverse sketch patterns, we propose an algorithm that simulates sketching process by augmenting each curve with noises individually. Figure 3 visualizes the flow of our line augmentation method.

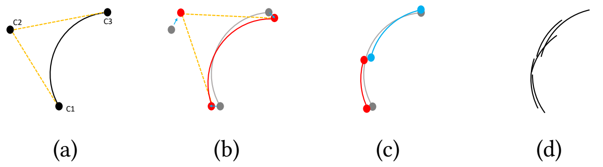

Figure 3: Flow of generating initial sketches described in Section 2.1. (a) Input curve represented by the three control points. (b) the results of the control points perturbation. (c) the case if the curve is divided into multiple smaller strokes. (d) the result after repeatedly drawing the sketch strokes.

We first vectorize a clean line image to get the list of the parametric curves using existing vectorization methods[Noris et al. 2013]. These methods work well with clean line images while they often fail to vectorize complicated sketch images. We use simple quadratic Bézier curve [Carter 1997] which can model curve shapes by three control points as follows:

$$B(c_1, c_2, c_3) = (1-t)^2 c_1 + 2(1-t)tc_2 + t^2 c_3,$$

where $t \in [0, 1]$.

By manipulating the control points c_1 to c_3, the Bézier curve can intuitively deform the shape of the line. We perturb the parameters of each Bézier curve by adding noises to the locations of the control points to simulate noisy sketch strokes. We sample the noises ϵ from the Gaussian distribution with zero mean and the standard deviation $\alpha \cdot \log(length(s))$, where s is a given clean stroke to be perturbed, and α is a hyperparameter value. We set the perturbation noises proportional to the length of the line segment, because the absolute scales of noises in sketch lines tend to increase proportionally to the line lengths in real sketches.

We also randomly divide long curves into multiple shorter curves, to enhance the variety of synthetic sketches and mimic the common behavior of artists that expressing a long line with multiple short lines. For this purpose, we evenly divide the segment to M_c curves and add noises to each curve independently. Finally, since artists often reinforce sketches by drawing strokes multiple times, we also

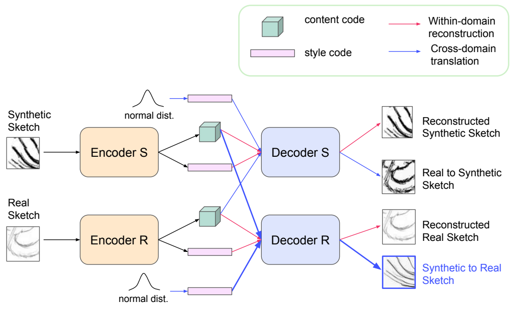

Figure 4: Description of Multi-modal Unsupervised Image-to-Imaget Translation model [Huang et al. 2018] used in our method. The model convert the textures of the initial synthetic sketches into the textures of real sketches.

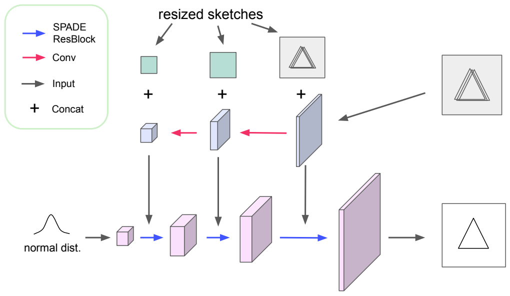

Figure 5: Description of our modified version of Spatially-Adaptive Normalization(SPADE) model [Park et al. 2019]. We feed the generator with the feature maps encoded by an additional convolutional neural network.

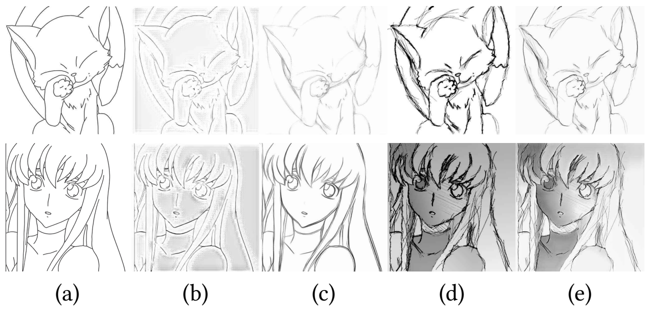

Figure 6: Comparison of synthetic sketches by various methods. (a) Input image. Synthetic sketches by (b) MUNIT [Huang et al. 2018], (c) MS [Simo-Serra et al. 2018a], (d) Our line augmentation, and (e) Our full methods. Images are copyrighted by nakamura02 https://www.deviantart.com/nakamura02.

repeatedly generate noisy strokes, where the number of repeats is another hyperparameter M_s. The choices of hyperparameters will be described in detail in Section 3.1.

Finally, we augment the synthetic sketches by changing background color and illumination. We also simulate areas made of lines that are common in sketches by running the watershed segmentation and filling some regions with lines with a regular interval. Figure 6 (d) shows some examples of the generated initial synthetic sketch images.

2.2 Unsupervised Texture Conversion

Although the initial synthetic sketches generated by Section 2.1 mimic real sketches in shape, they do not express varying textures and colors of papers and pencils used in real sketches. To convert artificial textures of the synthetic sketches, we train a multi-modal unsupervised image-to-image translation(MUNIT) model [Huang et al. 2018] with the initial synthetic sketches and real sketches.

In the MUNIT [Huang et al. 2018], generators consist of encoders and decoders. By decoding the contents from the synthetic sketches using the decoder for real sketches, we can generate the images with the contents of the synthetic sketches and the textures of real sketches. Please refer the original paper for more detailed model descriptions. Figure 4 describes the overall MUNIT model.

Figure 6 (e) shows some completed synthetic sketches, and Figure 6 (b) shows the results without the line augmentation. Mastering Sketching(MS) [Simo-Serra et al. 2018a] also presented a method for line-to-sketch conversion, and their method requires real paired sketch and line images. We show some example sketches by their method in Figure 6 (c). We can see that more challenging and diverse patterns are produced with our methods.

2.3 Learning to Simplify using Synthetic Data

To learn sketch-to-line translation, we train a recently proposed Spatially-Adaptive Normalization(SPADE) model [Park et al. 2019]. The model requires paired dataset and shows prominent progress in semantic image generation task compared to popular Pix-to-Pix model [Isola et al. 2017]. We train the model with our synthetic paired sketch and line images created by Section 2.1 and Section 2.2.

Unlike semantic labels used in the original SPADE paper, sketch images need to be encoded first to being semantically meaningful. Therefore, we additionally feed the feature maps encoded by an additional convolutional neural network jointly trained as shown in Figure 5. Without the encoded features, we found the model suffers to find meaningful structures in sketch images. We used Instance Normalization for the generator. The overall architecture of our simplification model is described in Figure 5.

3 EXPERIMENTS

3.1 Implementation Details

We collected 130 line images and created 10 sketches per each image. To train the MUNIT model, we collected unpaired sketch and line images from Danbooru2018 Dataset [Anonymous 2019]. We collected 20,000 sketches and 1320 line arts. For the hyperparameters used in the synthetic sketch construction, we used $\alpha \in \{30, 40, 50\}$ and randomly sampled M_c and M_s values from the poisson distribution with $\lambda = 3$. We set the ratio of colored or filled with lines regions to 50% of the images. When training MUNIT and SPADE models, we used ADAM optimizer [Kingma and Ba 2015] with a

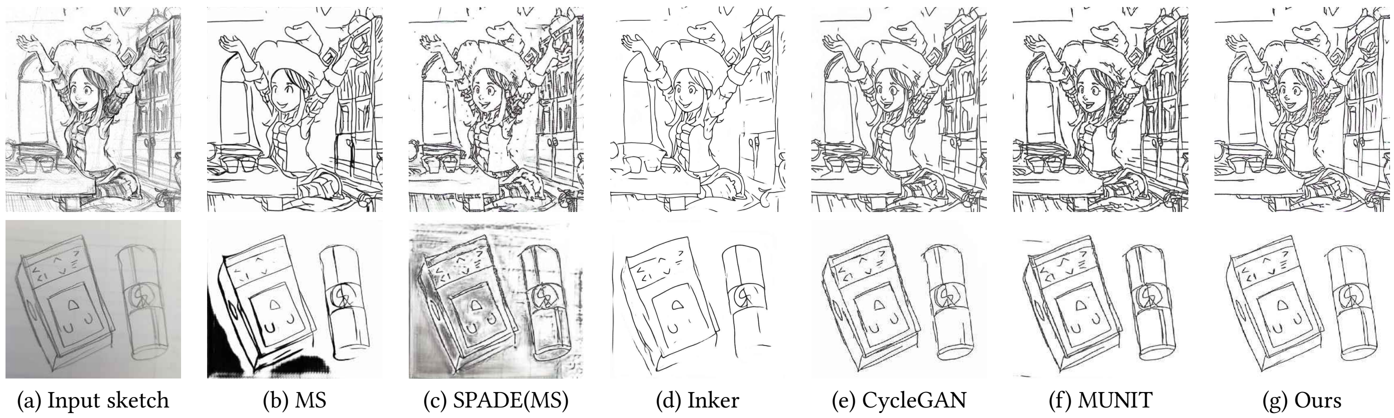

Figure 7: Visual comparison between various methods for sketch-to-line translation. The detailed references of each method are described in Section 3.2. Upper image is copyrighted by David Revoy www.davidrevoy.com and licensed under CC-by 4.0.

learning rate of 0.0001 with $\beta_1 = 0.5$ and $\beta_2 = 0.9$. During training, we first resize images into 512×512 and use 256×256 cropped patches. We trained the models 250,000 iterations with the batch size of 16. To prevent deformation of the line width of ground truth line images, we use line normalization module proposed in [Simo-Serra et al. 2018b].

3.2 Comparison with Existing Approaches

We compared our model with various state-of-the-art methods. Figure 7 shows the visual comparison between MS [Simo-Serra et al. 2018a], SPADE(MS), Inker [Simo-Serra et al. 2018b], CycleGAN [Zhu et al. 2017], MUNIT [Huang et al. 2018] and our model. SPADE(MS) model is trained with the sketches generated by the line-to-sketch model presented in MS. For the results of MS and SPADE(MS), we use the pretrained models provided by the authors, and for CycleGAN and MUNIT, we use the codes provided by their project sites. For Inker, we get the results from their demo site. We resized test images into 512×512.

As shown in Figure 7 (e) and (f), CycleGAN and MUNIT often suffer to remove noisy strokes, because they are forced to reconstruct noisy strokes from generated line images by cycle consistency losses. Compared to these models, our model is better at removing noises from strokes without worrying about the reconstruction. Also, our model works more robustly in difficult edge cases than models trained with a limited number of manually generated paired data. We believe that our model takes advantages of utilizing various and challenging sketch patterns synthesized by the proposed methods.

We report Fréchet Inception Distance(FID) between the distribution of results of each model and the distribution of test line images. We used 1000 sketch and 150 line images for the measurement. As shown in Table 1, our model achieves the lowest FID scores among the models. For human evaluation, we asked 20 users to choose the most visually appealing image. To reduce user burden, three models with the highest FID scores with randomly selected 20 test sketch images were compared. We report the percentage of predominant images in the table 1, and found that our method is the most preferred by the users.

Table 1: FID Scores(Lower is better) and the user study scores of various methods.

	FID Scores	User Study Scores
MS	100.70	25.0%
SPADE(MS)	109.59	-
CycleGAN	100.04	25.0%
MUNIT	101.21	-
Ours	95.38	50.0%

REFERENCES

Gwern Branwen Aaron Gokaslan Anonymous, the Danbooru community. 2019. Danbooru2018: A Large-Scale Crowdsourced and Tagged Anime Illustration Dataset. https://www.gwern.net/Danbooru2018
Matthew P Carter. 1997. Computer graphics: principles and practice.
Xun Huang, Ming-Yu Liu, Serge Belongie, and Jan Kautz. 2018. Multimodal Unsupervised Image-to-image Translation. *arXiv preprint arXiv:1804.04732* (2018).
Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. 2017. Image-to-Image Translation with Conditional Adversarial Networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017.* 5967–5976.
Diederik P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *ICLR*.
Gioacchino Noris, Alexander Hornung, Robert W Sumner, Maryann Simmons, and Markus Gross. 2013. Topology-driven vectorization of clean line drawings. *ACM Transactions on Graphics (TOG)* 32, 1 (2013), 4.
Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. 2019. Semantic Image Synthesis with Spatially-Adaptive Normalization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.*
Edgar Simo-Serra, Satoshi Iizuka, and Hiroshi Ishikawa. 2018a. Mastering sketching: adversarial augmentation for structured prediction. *ACM Transactions on Graphics (TOG)* 37, 1 (2018), 11.
Edgar Simo-Serra, Satoshi Iizuka, and Hiroshi Ishikawa. 2018b. Real-Time Data-Driven Interactive Rough Sketch Inking. *ACM Transactions on Graphics(SIGGRAPH)* 37, 4 (2018).
Edgar Simo-Serra, Satoshi Iizuka, Kazuma Sasaki, and Hiroshi Ishikawa. 2016. Learning to simplify: fully convolutional networks for rough sketch cleanup. *ACM Transactions on Graphics(SIGGRAPH)* 35, 4 (2016), 121.
Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. 2017. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *ICCV*.