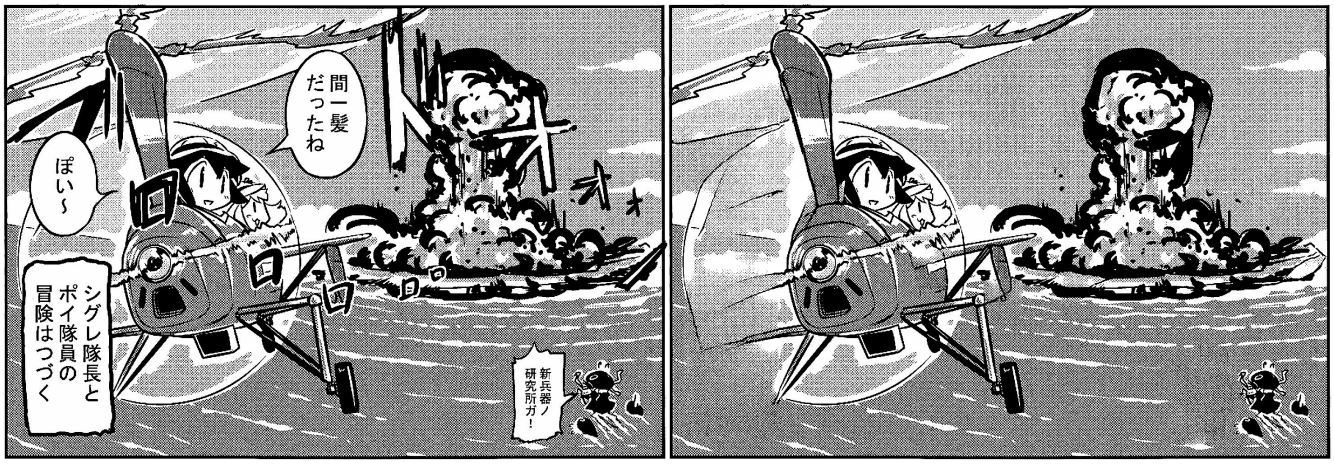
# Seamless Manga Inpainting with Semantics Awareness

MINSHAN XIE, The Chinese University of Hong Kong
MENGHAN XIA, The Chinese University of Hong Kong
XUETING LIU, Caritas Institute of Higher Education
CHENGZE LI, Caritas Institute of Higher Education
TIEN-TSIN WONG, The Chinese University of Hong Kong




(a) Original manga (b) Inpainted manga

Fig. 1. Our method automatically fills up the disoccluded regions with meaningful structural lines and seamless screentones. "She Great" ©SEIRAN

Manga inpainting fills up the disoccluded pixels due to the removal of dialogue balloons or "sound effect" text. This process is long needed by the industry for the language localization and the conversion to animated manga. It is mostly done manually, as existing methods (mostly for natural image inpainting) cannot produce satisfying results. Manga inpainting is more tricky than natural image inpainting because its highly abstract illustration using structural lines and screentone patterns, which confuses the semantic interpretation and visual content synthesis. In this paper, we present the first manga inpainting method, a deep learning model, that generates high-quality results. Instead of direct inpainting, we propose to separate the complicated inpainting into two major phases, semantic inpainting and appearance synthesis. This separation eases both the feature understanding and hence the training of the learning model. A key idea is to disentangle the structural line and screentone, that helps the network to better distinguish the structural line and the screentone features for semantic interpretation. Both the visual comparison and the quantitative experiments evidence the effectiveness of our method and justify its superiority over existing state-of-the-art methods in the application of manga inpainting.

CCS Concepts: • **Applied computing → Fine arts**.

Additional Key Words and Phrases: Manga production, Screentone, Manga inpainting

**ACM Reference Format:**
Minshan Xie, Menghan Xia, Xueting Liu, Chengze Li, and Tien-Tsin Wong. 2021. Seamless Manga Inpainting with Semantics Awareness. *ACM Trans. Graph.* 40, 4, Article 96 (August 2021), 11 pages. https://doi.org/10.1145/3450626.3459822

## 1 INTRODUCTION

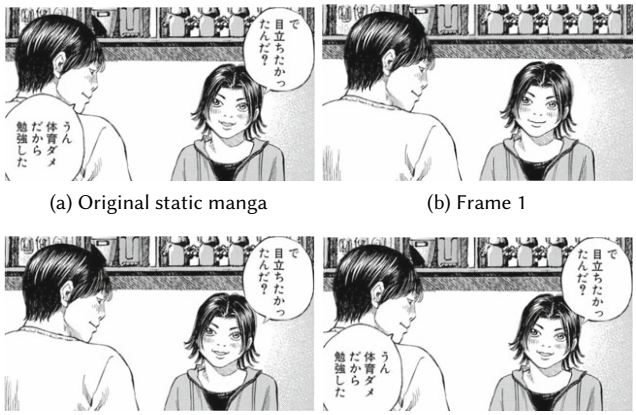
Manga inpainting is usually needed in the language localization of manga and the creation of animated manga. Fig. 2 demonstrates an example of translating the large "sound effect" text from Japanese to English, for the language localization purpose. Fig. 3(b)-(d) show three frames from an animated manga (electronic manga with Powerpoint-like animation, instead of smooth cartoon), converted from the original static manga (Fig. 3(a)). In both scenarios, the content behind the disoccluded regions is not available. Because both the dialogue balloon and the "sound effect" text are in place during the original layout design and never drawn by the manga artists, no matter the original manga is prepared digitally or on paper. Hence,

Authors' addresses: Minshan Xie, The Chinese University of Hong Kong, msxie@cse.cuhk.edu.hk; Menghan Xia, The Chinese University of Hong Kong, mhxia@cse.cuhk.edu.hk; Xueting Liu, Caritas Institute of Higher Education, tliu@cihe.edu.hk; Chengze Li, Caritas Institute of Higher Education, czli@cihe.edu.hk; Tien-Tsin Wong, The Chinese University of Hong Kong, ttwong@cse.cuhk.edu.hk.


0730-0301/2021/8-ART96 $15.00
https://doi.org/10.1145/3450626.3459822

(a) Manga (b) Inpainted manga (c) Localized manga

Fig. 2. The large "sound effect" text has to be modified during the language localization. "She Great" ©SEIRAN



(a) Original static manga (b) Frame 1

(c) Frame 2 (d) Frame 3

Fig. 3. Three frames from the animated manga, that is manually converted from the original static manga. "Give My Regards to Black Jack" ©Shuho Sato

manga inpainting has to be performed to seamlessly fill up the disoccluded regions. Unfortunately, even with the state-of-the-art natural image inpainting techniques, the results of applying them to manga inpainting are far from satisfactory. Manual inpainting is still the main practice in the industry.

Classical inpainting methods, such as PatchMatch [Barnes et al. 2009], synthesize the missing pixels with the similar visual appearance as the surrounding pixels from the same image. Recent deep neural network methods advance the natural image inpainting, with its higher-level semantic understanding and the utilization of pixels not just from the same image but also from the training images [Iizuka et al. 2017; Yu et al. 2018]. Unfortunately, both traditional and learning-based inpainting methods are designed for natural images, and fail to generate satisfying results for manga images, even those learning-based models are retrained with manga data. This is because the bitonal and pattern-rich natures of manga images can easily confuse existing methods in identifying the region boundary (Fig. 4(c)). In natural images, the region boundary and the texture/content usually exhibit significantly different characteristics, and hence ease their distinction, in which existing methods implicitly rely on. In sharp contrast, such distinction is no longer

(a) Input (b) [Barnes et al. 2009] (c) [Yu et al. 2018] (d) Ours

Fig. 4. Comparison of existing image inpainting methods. KaerimichiNo-Majo ©A-10

obvious in bitonal manga images as demonstrated in Fig. 4(c), the structural outline and the screened regions may share the same line width and appearance.

In this paper, we make the first attempt to generate high-quality manga inpainting. Instead of directly inpainting the final result, we propose a deep neural network model consists of two sub-networks, *semantic inpainting network* and *appearance synthesis network*. The semantic inpainting network predicts a pair of inpainted structural line map and inpainted screentone map. The appearance synthesis network takes this pair of predictions as guidance and synthesizes the disoccluded region with pixels from the original manga, using a contextual attention design. This indirect design allows us to generate high-quality inpainted content that can be seamlessly combined with the surrounding pixels in the original manga.

In order to semantically differentiate the structural line and the screened regions, the semantic inpainting network is subdivided into two sub-tasks, *structural line inpainting* for connecting structure within the disoccluded region, and *screentone inpainting* for filling up the screentone. The backward propagated training of our neural network ensures the consistency between the inpainted structural line and screentone guidance maps, as well as the utilization of knowledge from both sides. In order to handle the large disoccluded regions due to the removal of dialogue balloons, a recurrent updating approach is adopted in designing the semantic inpainting network to gradually inpaint from the outer ring of the mask towards its interior. This provides more pleasant results than that from one-step inpainting. In addition, screentone inpainting is performed in the feature domain extracted by a Screentone Variational AutoEncoder (ScreenVAE) [Xie et al. 2020], namely the ScreenVAE map. This allows the network to better understand the region-wise characteristics of screentones and eases the convergence.

We apply our manga inpainting method on various real-world manga to inpaint the dialogue balloons as well as the "sound effect" text. Extensive qualitative and quantitative experiments are conducted. Convincing results are obtained in terms of visual quality and quantitative evaluation. Experiments show that we outperform existing state-of-the-art inpainting techniques that are originally designed for natural images. Our contributions can be summarized as follows:

- We are the first to tackle the challenging manga inpainting task and obtain high-quality results.

- We propose to subdivide the manga inpainting task into structural line inpainting and ScreenVAE map inpainting to better address, and hence learn, the semantic difference between structure and screentone.

## 2 RELATED WORK

### 2.1 Image Inpainting

Image inpainting has been a classic problem in computer vision and computer graphics for decades, which aims to fill obstacle or contaminated regions with visually plausible synthetic contents. The existing image inpainting methods can be categorized into two branches, traditional methods and deep neural network methods.

The traditional methods mainly utilize the neighborhood structural features to search for similar patches to fill the target regions [Barnes et al. 2009; Criminisi et al. 2004; Darabi et al. 2012]. To minimize the transition discontinuities, image blending is usually performed between filled regions and source regions [Darabi et al. 2012; Huang et al. 2014]. However, these procedural methods are computationally expensive since they have to calculate the similarity scores for all possible target-source pairs. PatchMatch[Barnes et al. 2009] solved these problems by adopting a fast nearest-neighbor field algorithm. However, it fails to generate semantically meaningful results when plausible objects do not exist in the surrounding pixels or the low-level feature matching is interfered with by illumination or perspective distortion. Due to the difficulty in crafting high-level semantic features, the traditional methods generally assume that the target content exists in surrounding pixels.

Recently, deep learning achieves great success in image interpretation and generation. Thanks to the powerful representation capability, various convolutional neural networks (CNN) models have been proposed and push forward the state-of-the-art of natural image inpainting. To generate semantic meaningful content, the generative adversarial network [Goodfellow et al. 2014] is generally used in the existing image inpainting models to provide visual perceptual guidance. Pathak et al. [2016] made the first attempt in bringing adversarial training [Goodfellow et al. 2014] to image inpainting. Iizuka et al. [2017] introduced local and global discriminators, assisted by dilated convolution [Yu and Koltun 2015] to improve the image quality. However, these methods often generated content with distorted structural lines or blurry textured regions. Yu et al. [2018] and Song et al. [2018a] adopted a coarse-to-fine architecture to first inpaint the coarse semantic content and then generate the fine details by searching for the most similar patches from the existing pixels. Liu et al. [2019] proposed coherent semantic attention, which considers the feature coherency in the target regions to guarantee pixel continuity in image level. Zeng et al. [2019] attempted to inpaint the target region in an iterative approach from high-level semantics to low-level pixels to generate fine-grained image patches. However, these methods often suffer from either over-smoothed boundaries or texture artifacts. To tackle the problem, two-stage approaches introduce another image completion stage over the additional prior structure conditions, e.g. edge information [Nazeri et al. 2019], segmentation mask [Song et al. 2018b], etc. Ren et al. [2019] introduced a structure-aware network, that splits the inpainting task into the structure reconstruction and the texture generation sub-tasks. It uses appearance flow to yield image details from relative regions. Recently, image inpainting with recurrent updating scheme has been investigated to inpaint from the hole boundary to the center in an iterative manner. Oh et al. [2019] used an onion-peel scheme to progressively inpaint from the hole boundary on video data, using content from reference frames. Zeng et al. [2020] proposed to refer to a confidence map to iteratively revise the unsatisfactory regions. However, these methods generally fail to generate satisfying results for manga images due to the distinctive appearances of bitonal screentone patterns, which is much more tricky to be inpainted with seamless structure and screentone. Additionally, these methods cannot well distinguish the semantic components such as structure and textural regions. In comparison, our proposed method generates high-quality inpainting results by identifying the semantic difference between structure and screentone.

### 2.2 Manga Analysis and Processing

A few attempts have been proposed to identify and extract the content in manga, such as screentones, structural lines, text, balloons, and characters. Ito et al.[2015] proposed a method to separate the line drawings and screentones in manga. Liu et al [2017] proposed a method for segmenting the textures in manga. Recently, deep learning approaches are adopted in manga analysis and processing, and achieved high performance. Li et al.[2017] proposed a deep learning method for extracting the structural lines from the screened manga. Aramaki et al.[2016] combined connected-component and region-based classifications to detect the text in manga. To the best of our knowledge, none of the existing methods is tailored for manga inpainting. The most related one is the work proposed by Xie et al. [2020], which attempted to convert the filling style between manga and western color comics by interpreting the discrete screentones into a smooth representation. However, this method still fails to generate well-matched screentones in the target region when directly applied in manga inpainting. They also fail to inpaint the structural lines when the target region is of large size. Sasaki et al.[2017] proposed a deep learning approach for automatically detecting and completing gaps in line drawings. However, their method can only work well when the gap is small and generally fail when the gap is large. In this paper, we made the first attempt to tackle the challenging manga inpainting task and obtain high-quality results even for large disoccluded regions.

## 3 OVERVIEW

Our manga inpainting framework is illustrated in Fig. 5, which consists of a semantic inpainting network $G_{\text{inp}}$ and an appearance synthesis network $G_{\text{syn}}$. Given an input manga image $\mathbf{I}$ and a mask of the region to be inpainted $\mathbf{M}$, we first decompose the image into a structural line map $\mathbf{L}$ and a screentone image $\mathbf{I}_s$ using manga tailored method [Li et al. 2017]. The screentone image is further encoded as the ScreenVAE map $\mathbf{S}$ with smooth values within each screentone region [Xie et al. 2020]. Correspondingly, we also decompose the masked manga $\mathbf{I}_\mathbf{M} = \mathbf{I} \odot \mathbf{M}$ into a masked structural line map $\mathbf{L}_\mathbf{M}$ and a masked ScreenVAE map $\mathbf{S}_\mathbf{M}$. In particular, the masked regions in manga are set to be in white color to maintain the bitonal

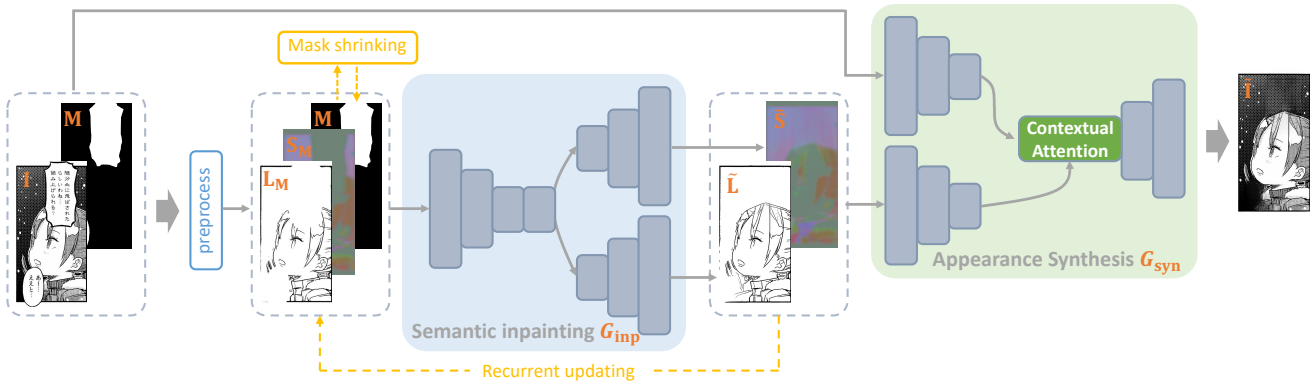



Fig. 5. Overview diagram of our manga inpainting system. Given a manga image $\mathbf{I}$ and the region to inpaint indicated by a mask $\mathbf{M}$, we first decompose $\mathbf{I}$ into the structural line $\mathbf{L}$ and the ScreenVAE map $\mathbf{S}$, which are then inpainted by the semantic inpainting network $G_{\text{inp}}$. The inpainted structural line $\tilde{\mathbf{L}}$ and ScreenVAE map $\tilde{\mathbf{S}}$ are further fed into the appearance synthesis model $G_{\text{syn}}$ as semantic guidance maps to utilize known pixels to predict the occluded pixels $\tilde{\mathbf{I}} \odot \mathbf{M}$. Besides, to handle large occlusions, the semantic inpainting runs in a recurrent updating fashion to gradually infer content from outermost contours to interior. KaerimichiNoMajo ©A-10

nature. Then, the masked structural line map $\mathbf{L_M}$ and the masked ScreenVAE map $\mathbf{S_M}$ are fed to the semantic inpainting network $G_{\text{inp}}$ to generate two semantic guidance maps, the inpainted structural line map $\tilde{\mathbf{L}}$ and the inpainted ScreenVAE map $\tilde{\mathbf{S}}$. However, $\tilde{\mathbf{L}}$ and $\tilde{\mathbf{S}}$ alone cannot be directly used for the final inpainting, because the screentones generated this way usually does not match well with the surrounding screentones (Fig. 8(c)). Instead, we propose an appearance synthesis network $G_{\text{syn}}$ with a contextual attention design. The goal is to utilize the original surrounding pixels, through learning from the contextual with the guidance of $\tilde{\mathbf{L}}$ & $\tilde{\mathbf{S}}$, to synthesize the inpainting with semantic coherence and seamless appearance (especially screentones) to the surrounding pixels. The whole process can be formulated as:

$$\tilde{\mathbf{L}}, \tilde{\mathbf{S}} = G_{\text{inp}}(\mathbf{L_M}, \mathbf{S_M}, \mathbf{M}), \quad (1)$$

$$\tilde{\mathbf{I}} = G_{\text{syn}}(\mathbf{I_M}, \tilde{\mathbf{L}}, \tilde{\mathbf{S}}, \mathbf{M}). \quad (2)$$

Importantly, to better understand the manga semantic information, our semantic inpainting network is subdivided into two distinctive but correlated sub-tasks: structural line inpainting for connecting structures within the disoccluded region, and region-wise screentone inpainting for filling up proper screentone patterns. It helps to distinguish two visually close but functionally different manga elements and hence promotes the effectiveness of semantic generation. Furthermore, to circumvent the challenge of inpainting large disocclusion, a recurrent updating approach is adopted to infer the content in an iterative manner. The semantic content of missing regions is progressively inpainted from outer ring of mask to center to simplify the task. Then the mask is gradually shrunk for the next iteration. The number of iterations is fixed, and currently fixed to 5 iterations.

Supervised by the groundtruth, the semantic inpainting model and the appearance synthesis model are first trained separately, and then jointly trained for fine-tuning. The detailed model architectures and loss functions are described in Section 4.

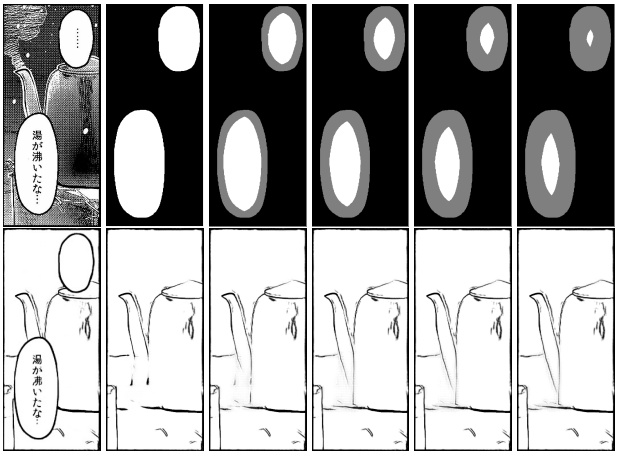
## 4 APPROACH

We divide the manga inpainting problem into two major steps. Firstly, we predict the semantic elements in the disoccluded regions. Then, we synthesize the appearance conditioned by semantic guidance. Comparing to existing direct inpainting solutions that usually fail to predict reasonable disoccluded manga regions, our indirect framework divides and conquers the problem by disentangling semantic interpretation and appearance synthesis. Below, we elaborate on the two steps in detail.

### 4.1 Semantic Inpainting

Each manga contains two key components, structural line and screentone. They play different roles in semantic illustration. Hence, we first decompose the input manga image $\mathbf{I_M}$ into the structural line $\mathbf{L_M}$ and the screentone components $\mathbf{S_M}$. In particular, we employ the manga structural line extractor [Li et al. 2017] to extract the structure lines from the input manga image, and the ScreenVAE method [Xie et al. 2020] to represent the manga image as a ScreenVAE map. The ScreenVAE representation serves to convert the region-wise bitonal screentone to a point-wise 4-channel ScreenVAE representation to ease the feature learning and translation. Similarly, the groundtruth structural line map $\mathbf{L}$ and the groundtruth ScreenVAE map $\mathbf{S}$ can be extracted from the original manga $\mathbf{I}$. The two separated components $\mathbf{L_M}$ and $\mathbf{S_M}$ are then concatenated and fed into our semantic inpainting network $G_{\text{inp}}$ to predict the disoccluded structures $\tilde{\mathbf{L}}$ and screentones $\tilde{\mathbf{S}}$. Our semantic inpainting network features a dual-branch structure, where the structural lines and screentones contribute to consistent semantic maps by benefiting each other. The network architecture is detailed in the supplementary material.

One key challenge in region-filling of our semantic inpainting task is the huge size of the disoccluded regions, especially for dialogue balloons. We miss almost a significant portion of prior knowledge to recover the semantic information within the region. To handle this problem, we adopt a recurrent updating approach in the semantic inpainting network, as depicted by the yellow dashed arrow in



(a) Input (b) i=1 (c) i=2 (d) i=3 (e) i=4 (f) i=5

Fig. 6. The inpainted results generated after each of the 5 iterations. Here, we use mask (each pixel can only be 0, 0.5 or 1) to allow the inpainted outermost ring regions to be refined in the whole iterative process. KaerimichiNoMajo ©A-10

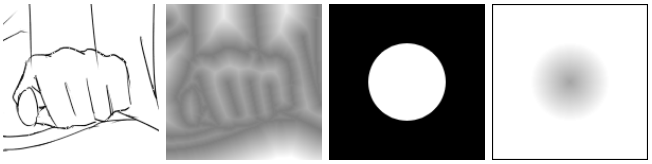
Fig. 5. Specifically, our semantic inpainting network takes a mask $\mathbf{M}$ as input to specify the iteratively changed inpainting focus. The pixel values in the mask only might be 0, 0.5, or 1, where 1 denotes the highest inpainting priority, 0 means no constraint, and 0.5 is in between. The upper row of Fig. 6 visualizes the masks of all iterations. The first mask is filled with value 1. After each iteration, we shrink the mask by filling the current outermost $\frac{1}{N}$ ring region with value 0.5, where $N$ is the total number of iterations. This approach allows us to train the network to focus on different regions for different iteration. Empirically, $N = 5$ is enough to achieve visually satisfactory results.

*Loss Function.* Our semantic inpainting model is optimized under the loss function composed of five loss terms, structural line reconstruction loss $\mathcal{L}_{\text{lrec}}$, ScreenVAE map reconstruction loss $\mathcal{L}_{\text{srec}}$, adversarial loss $\mathcal{L}_{\text{adv}}$, feature matching loss $\mathcal{L}_{\text{fm}}$ and binarization loss $\mathcal{L}_{\text{bin}}$, as formulated respectively below.

• **Structural line reconstruction loss.** The structural line reconstruction loss $\mathcal{L}_{\text{lrec}}$ ensures the generated structural line $\tilde{\mathbf{L}}$ to be similar to the groundtruth structural line $\mathbf{L}$. However, naive pixel-wise measurement on $\tilde{\mathbf{L}}$ and $\mathbf{L}$ is inappropriate because slight misalignment may not affect the visual fidelity but come with a great pixel-wise difference. Instead, we propose to compute a distance field map $\mathbf{M}_{\text{d}}$ [Frisken et al. 2000] computed from the structural line maps, to penalize the structural lines away from the target ones in a continuous manner. The weight value in $\mathbf{M}_{\text{d}}$ at point $p$ is computed as:

$$\mathbf{M}_{\text{d}}(p) = \frac{\|p - \mathcal{P}_S(p)\|}{\sqrt{H * W}} + \epsilon, \tag{3}$$

where $S$ is the 2D point set of the structural lines, and $\mathcal{P}_S(p)$ is the projection of $p$ in set $S$. It is normalized by the factor $\sqrt{H * W}$. $\epsilon = 0.5$ is used as base value. Fig. 7 shows an example of the distance field map. We can see that the penalty grows as the position is away from the groundtruth. Like the natural image inpainting tasks, the interior regions are more ambiguous to be learned than those outer regions since no immediate surrounding connections are available. This difference will be amplified in the case of sparse structural lines. To alleviate the learning ambiguity, we introduce a confidence map $\mathbf{M}_{\text{c}}$ that has radically decreasing values away from the mask boundary to help the network to focus more on those regions with stronger context prior. Then, the structural line reconstruction loss is computed as:



(a) Structural line (b) Distance field for line $\mathbf{M}_{\text{d}}$ (c) Mask (d) Confidence map for mask $\mathbf{M}_{\text{c}}$

Fig. 7. Distance field map of the structural line and the confidence map of the mask.

$$\mathcal{L}_{\text{lrec}} = \sum \{\|(\tilde{\mathbf{L}} - \mathbf{L}) \odot \mathbf{M}_{\text{d}} \odot \mathbf{M}_{\text{c}}\|_2\} \tag{4}$$

• **ScreenVAE map reconstruction loss.** The ScreenVAE map reconstruction loss $\mathcal{L}_{\text{srec}}$ ensures the inpainted ScreenVAE map $\tilde{\mathbf{S}}$ to be similar to the groundtruth ScreenVAE map $\mathbf{S}$. We adopt pixel-wise mean square error (MSE) to regularize the similarity with a confidence map $\mathbf{M}_{\text{c}}$ encourage to learn from more reliable regions. The reconstruction loss for ScreenVAE map can be computed as:

$$\mathcal{L}_{\text{srec}} = \sum \{|(\tilde{\mathbf{S}} - \mathbf{S}) \odot \mathbf{M}_{\text{c}}\|_2\} \tag{5}$$

• **Adversarial loss and feature matching loss** The adversarial loss $\mathcal{L}_{\text{adv}}$ and feature matching loss $\mathcal{L}_{\text{fm}}$ are to constrain the distribution of the generated structural line or screentone to be as indistinguishable as possible from the real ones. Note that, the structural line and ScreenVAE map are correlated, i.e. structural line usually locates at the place with abrupt change of screentone types. Here, we use an adversarial loss [Goodfellow et al. 2014] on the inpainted guidance maps. Specifically, we adopt a discriminator $D_{\text{ls}}$ with 5 strided downscaling blocks. To stabilize the adversarial training, we also encourage the similarity between the extracted discriminative features of the generated results and the groundtruth. The adversarial loss and feature matching loss is defined as:

$$\mathcal{L}_{\text{adv}} = \sum \{\log D_{\text{ls}}(\mathbf{L}, \mathbf{S}) + \log(1 - D_{\text{ls}}(\tilde{\mathbf{L}}, \tilde{\mathbf{S}}))\} \tag{6}$$

$$\mathcal{L}_{\text{fm}} = \sum \|\phi^i(\mathbf{L}, \mathbf{S}) - \phi^i(\tilde{\mathbf{L}}, \tilde{\mathbf{S}})\|_2 \tag{7}$$

where $\phi^i(.,.)$ is the extracted feature maps by the discriminator $D_{\text{ls}}$ on i-th layer. We empirically use the layers $i \in \{3, 4, 5\}$.

• **Binarization loss** With the above losses alone, we find that the generated structural line tends to be blurry, so we introduce the binarization loss $\mathcal{L}_{\text{bin}}$ to encourage the network to generate black-and-white pixels, which is defined as:

$$\mathcal{L}_{\text{bin}} = \sum \||\tilde{\mathbf{L}} - 0.5| - 0.5\|_2, \tag{8}$$

where $|\cdot|$ means to compute the element-wise absolute value while the $\tilde{\mathbf{L}}$ ranges in $[0, 1]$.

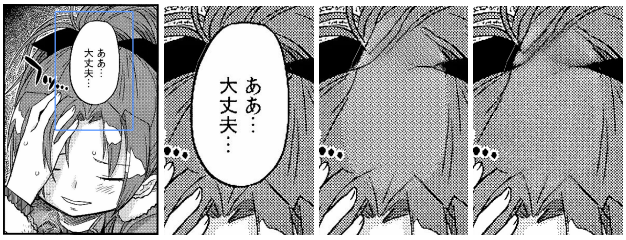
Overall, the loss function of our semantic inpainting network is defined as the weighted sum of the five loss terms:

$$\mathcal{L}_{\text{inp}} = \alpha_{\text{lrec}}\mathcal{L}_{\text{lrec}} + \alpha_{\text{srec}}\mathcal{L}_{\text{srec}} + \alpha_{\text{adv}}\mathcal{L}_{\text{adv}} + \alpha_{\text{fm}}\mathcal{L}_{\text{fm}} + \alpha_{\text{bin}}\mathcal{L}_{\text{bin}} \quad (9)$$

where $\alpha_{\text{lrec}} = 20$, $\alpha_{\text{srec}} = 20$, $\alpha_{\text{adv}} = 1$, $\alpha_{\text{fm}} = 5$ and $\alpha_{\text{bin}} = 1$ are empirically adopted to balance these terms.

### 4.2 Appearance Synthesis

Given the inpainted semantic maps, $\tilde{\mathbf{L}}$ and $\tilde{\mathbf{S}}$, an intuitive idea is to directly reconstruct the output manga from the maps with the ScreenVAE model [Xie et al. 2020]. However, as results shown in Fig. 8, the reconstructed screentones present noticeable visual inconsistency to the surrounding screentone in the original manga. One possible reason is that the learned ScreenVAE cannot represent all existing screentones equally well due to dataset bias. Instead, we only utilize the semantic maps as correlation guidance and fill up the disocclusion regions by borrowing features from known surrounding regions in the non-masked area. This design shares the same spirit with the contextual inpainting method [Yu et al. 2018] and improves the visual conformity to the original appearance as well as retaining plausible semantic meanings.




(a) Manga (b) Blown-up region (c) Screentone decoded by ScreenVAE (d) Ours

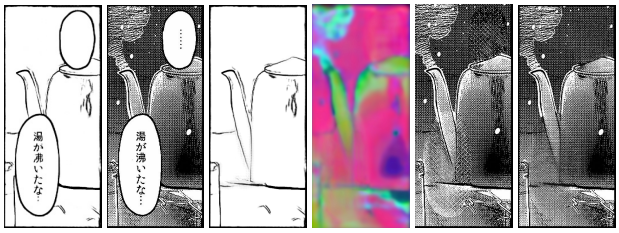
Fig. 8. Screentone directly decoded from the ScreenVAE [Xie et al. 2020] usually does not match well with the surrounding screentone in the input. KaerimichiNoMajo ©A-10

*Loss Function.* We formulate four loss terms to train the appearance synthesis network, i.e. manga reconstruction loss $\mathcal{L}_{\text{mrec}}$, ScreenVAE map loss $\mathcal{L}_{\text{scr}}$, adversarial loss $\mathcal{L}_{\text{adv}}$, and binarization loss $\mathcal{L}_{\text{line}}$.

• **Manga reconstruction loss** The manga reconstruction loss encourages the model to generate results that are similar to the groundtruth screened manga. Similar to the technique used in $\mathcal{L}_{\text{lrec}}$ to suppress the ambiguity, we also adopt a confidence map to emphasis more near the mask boundary to guarantee a natural transition. Fig. 9 demonstrates the superiority of this design. The manga reconstruction loss is then defined as:

$$\mathcal{L}_{\text{mrec}} = \sum \{\|(\tilde{\mathbf{I}} - \mathbf{I}) \odot \mathbf{M}_c\|_2\} \quad (10)$$

• **ScreenVAE map loss** The ScreenVAE map loss ensures that the generated manga image is inpainted with the same screentone types as the groundtruth. With the manga reconstruction loss alone, the model may generate screentones with a similar tone but different screentone types. Such inconsistency in screentone patterns significantly hurts the visual quality. So, the ScreenVAE map loss is then formulated as the difference between the screenVAE map of generated manga and the groundtruth ScreenVAE map $\mathbf{S}$.



(a) Input (b) Inpainted guidance (c) w/o $\mathbf{M}_c$ (d) w $\mathbf{M}_c$

Fig. 9. The inpainted results generated with and without the weight masks. The guidance ScreenVAE map is visualized after PCA. KaerimichiNoMajo ©A-10

$$\mathcal{L}_{\text{scr}} = \sum \|\text{SVAE}(\tilde{\mathbf{I}}) - \mathbf{S}\|_2 \quad (11)$$

where SVAE denotes the pretrained ScreenVAE model [Xie et al. 2020] that computes the ScreenVAE map from the input manga image.

• **Adversarial loss** The adversarial loss encourages the generated manga to follow the information provided by the inpainted structural line and inpainted ScreenVAE map. We adopt Conditional GAN [Mirza and Osindero 2014] to impose this constraint. A discriminator $D_{\text{mg}}$ with 5 strided downscaling blocks is introduced. The adversarial loss is defined as:

$$\mathcal{L}_{\text{adv}} = \sum \{\log(1 - D_{\text{mg}}(\tilde{\mathbf{I}}, \mathbf{L}, \mathbf{S})) + \log D_{\text{mg}}(\mathbf{I}, \mathbf{L}, \mathbf{S})\} \quad (12)$$

• **Binarization loss** The binarization loss is introduced to encourage bitonal appearance of the synthesized manga image $\tilde{\mathbf{I}}$. The binarization loss $\mathcal{L}_{\text{bin}}$ is defined as:

$$\mathcal{L}_{\text{bin}} = \sum \|\|\tilde{\mathbf{I}} - 0.5| - 0.5\|_2, \quad (13)$$

The overall loss function for the appearance synthesis network is defined as the weighted sum of the four terms:

$$\mathcal{L} = \beta_{\text{mrec}}\mathcal{L}_{\text{mrec}} + \beta_{\text{scr}}\mathcal{L}_{\text{scr}} + \beta_{\text{adv}}\mathcal{L}_{\text{adv}} + \beta_{\text{bin}}\mathcal{L}_{\text{bin}} \quad (14)$$

where $\beta_{\text{mrec}}$, $\beta_{\text{scr}}$, $\beta_{\text{adv}}$ and $\beta_{\text{bin}}$ are the weight coefficients for different loss terms. We empirically set $\beta_{\text{mrec}} = 10$, $\beta_{\text{scr}} = 100$, $\beta_{\text{adv}} = 1$, and $\beta_{\text{bin}} = 1$ in our experiments.

## 5 EXPERIMENTAL RESULTS

### 5.1 Data Preparation and Implementation Details

*Dataset.* Currently there is no publicly available high-resolution manga dataset that we can directly use. So, we manually collect 20,000 screened manga of resolution 2,048× 1,536 to train our model. For each screened manga, we extract the structural lines using the line extraction model by Li et al. [2017] and the ScreenVAE map using the model proposed by Xie et al. [2020].

For our experiments, we use two types of image masks, regular-shaped and irregular-shaped, to imitate the balloon and sound effect text, respectively. We configure the regular-shaped masks as rectangle masks covering about 4% to 25% of the image at a random location. For the irregular-shaped masks, we gather them from the work of Liu et al. [2018]. They are generated based on their sizes

relative to the entire image in the increment range of 5% to 20%. Meanwhile, we further dilate the masks to simulate the sound effect text in manga. Overall, the irregular masks cover about 10% to 50% of the image.

*Training.* We trained our model in the PyTorch framework [Paszke et al. 2017]. The network weights are randomly initialized using the method of [He et al. 2015]. When training the semantic inpainting model, we empirically set parameters as $\alpha_{\text{lrec}} = 20$, $\alpha_{\text{srec}} = 20$, $\alpha_{\text{adv}} = 1$, $\alpha_{\text{fm}} = 5$, and $\alpha_{\text{bin}} = 1$. To train the appearance synthesis model, we set $\beta_{\text{mrec}} = 10$, $\beta_{\text{scr}} = 100$, $\beta_{\text{adv}} = 1$, and $\beta_{\text{line}} = 1$. Adam solver [Kingma and Ba 2014] is applied to two models with a batch size of 4 and an initial learning rate of 0.0001.

## 5.2 Comparision

To evaluate our method, we compare our inpainted results against the-state-of-art image inpainting methods. There are three categories, including PatchMatch [Barnes et al. 2009] using low-level image features, four feed-forward generative models with deep convolutional networks [Nazeri et al. 2019; Xie et al. 2020; Yu et al. 2018; Zeng et al. 2019] and a commercial software, Photoshop [2021].

The visual comparison can be seen in Fig. 10. When inpainting the disoccluded regions in manga, we need to guarantee both the meaningfully structural lines and consistent screentones. We can see that, in general, our method shows plausible results which not only contains clear and pleasant structural line but also homogenous screentones over regions to show great visual impression. Although the results inpainted by PatchMatch [Barnes et al. 2009] can have realistic patterns, it failed to generate semantically meaningful content as it only considers the low-level features. We also compare our inpainted results against a commercial software Photoshop [2021] with content-aware filling option. Although they can generate plausible patches over small regions, they fail to generate acceptable structures over large regions.

We also performed a comparison against 4 deep learning methods. Note that three of them are originally designed for natural image inpainting [Nazeri et al. 2019; Yu et al. 2018; Zeng et al. 2019] and we tested with their models retrained with our training data. From Fig. 10, we can see that all the methods tailored for natural image inpainting fail to understand the content of manga and synthesize the appearance of screentones. We also compared to the manga filling style translation method proposed by Xie et al. [2020] which has shown their potential on manga inpainting. However, their method cannot generate screentone that matches well with the surrounding non-masked screentone in the original manga. Moreover, they do not generate the structural line inpainting, as shown in Fig. 10.

Other than the qualitative comparison, we also evaluate the performance of all these methods with quantitative measurements. We first collected 500 manga images and then generate random masks. These images and masks are further inpainted by these methods. The quantitative evaluation is listed in Table 1. We adopt 3 metrics to evaluate the quality of these inpainted images, including Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index Measure (SSIM) [Wang et al. 2004], and Learned Perceptual Image Patch Similarity (LPIPS) [Zhang et al. 2018]. For the PSNR and SSIM measurement, we only measure the masked regions, non-masked area is excluded. Note that the PSNR and SSIM scores have limitations in evaluating inpainted content. A slight shift of content that is visually acceptable, can be over-penalized by these metrics, as they are alignment-sensitive. Besides the PSNR and SSIM, we adopt a perceptual-associated metric, Learned Perceptual Image Patch Similarity (LPIPS)[Zhang et al. 2018], to evaluate the distance between image patches. The LPIPS is often used as a deep image quality assessment for image synthesis. In this paper, the LPIPS is evaluated based on the VGG16 [Simonyan and Zisserman 2014] model. The lower the LPIPS score is, the better the visual quality of the inpainted content is. From the Table 1, we can see that our results outperform all the existing methods, in terms of all metrics.

Table 1. The quantitative comparison to various state-of-the-art methods.

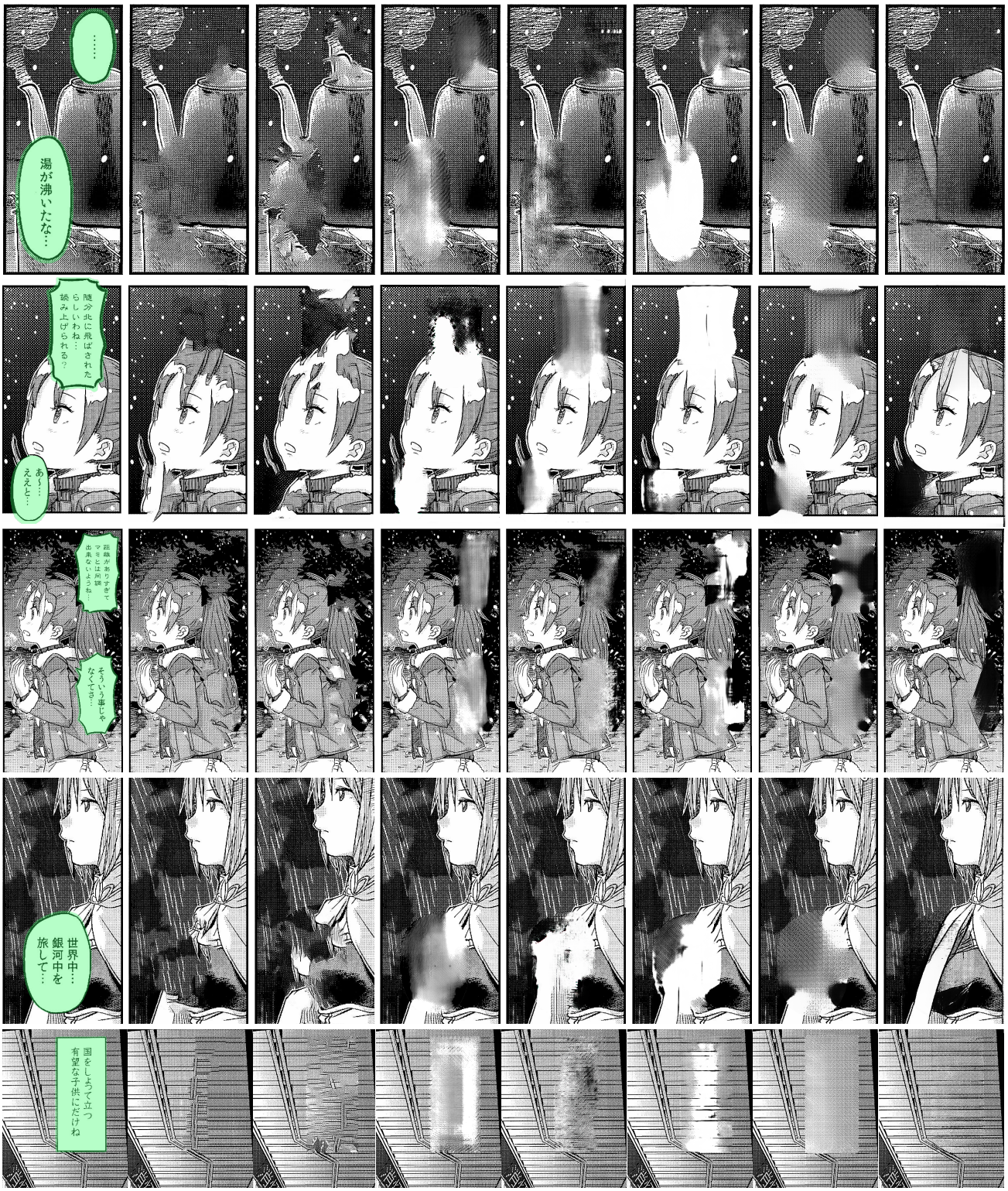
| Methods | PSNR | SSIM | LPIPS |
|---|---|---|---|
| [Barnes et al. 2009] | 8.6844 | 0.3412 | 0.1107 |
| [Yu et al. 2018] | 9.0901 | 0.4101 | 0.1127 |
| [Zeng et al. 2019] | 8.3175 | 0.3497 | 0.1217 |
| [Nazeri et al. 2019] | 8.4133 | 0.3895 | 0.1141 |
| [Xie et al. 2020] | 7.8237 | 0.2477 | 0.1318 |
| Ours | **10.8302** | **0.5373** | **0.0706** |

## 5.3 Ablation Study

To verify the effectiveness of individual loss term, we conduct ablation studies for each module by visually and quantitatively comparing the generated output.

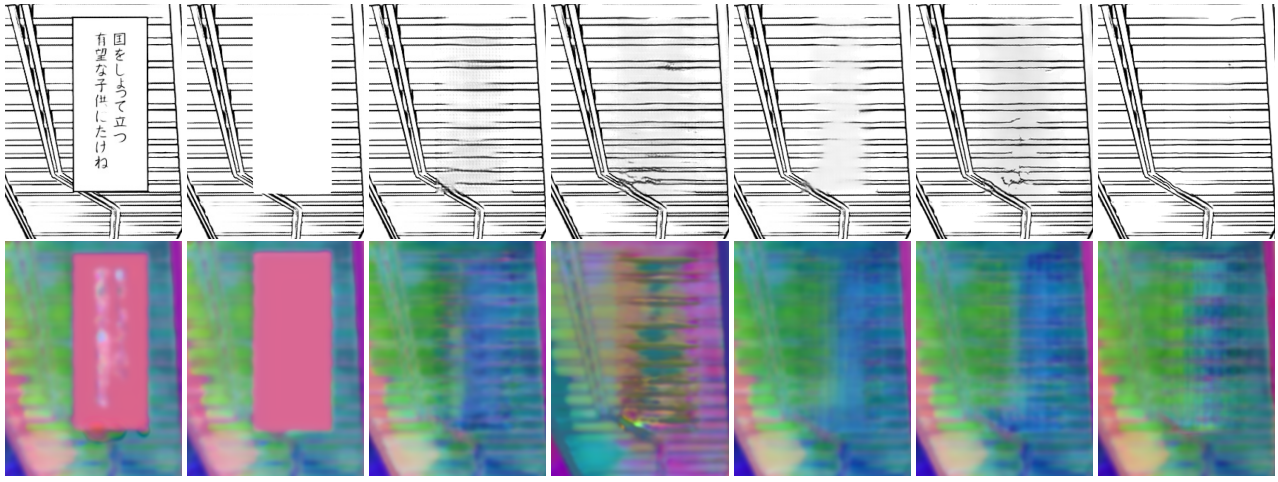
*Semantic inpainting model.* Fig. 11 shows the generated structural lines and ScreenVAE map of the trained model without individual loss term. The top row shows the inpainted structural line map while the bottom row visualizes the inpainted ScreenVAE map. Without the structural line loss $\mathcal{L}_{\text{lrec}}$, some fine structural line may not be generated (top row of Fig. 11(c)). Similarly, the ScreenVAE map $\mathcal{L}_{\text{srec}}$ cannot be inpainted well without the ScreenVAE map loss (bottom row of Fig. 11(d)). Without the adversarial loss $\mathcal{L}_{\text{adv}}$ and feature matching loss $\mathcal{L}_{\text{fm}}$, the network may fail to be aware of the high-level semantic features, so the generated results for these large regions are usually not recovered (Fig. 11(e)). The binarization loss $\mathcal{L}_{\text{bin}}$ will avoid the generated structural line to be smoothed out (top row of Fig. 11(f)). In comparison, the combined loss can help the network to generate meaningful structural line as well as continuous and aligned ScreenVAE map for disoccluded region (Fig. 11(g)). The quantitative evaluation in Table 2 also shows that the combined loss quantitatively outperforms the others.

*Appearance synthesis model.* Fig. 12 shows the generated screened manga of the trained model without individual loss term. Without the manga reconstruction loss $\mathcal{L}_{\text{mrec}}$, unconsistent results may be generated (Fig. 12(d)). When we remove the ScreenVAE map loss $\mathcal{L}_{\text{srec}}$, the model is not restricted to generate consistent screentones for disoccluded regions. As we can see in Fig. 12(e), the disoccluded regions may fail to be filled with screentones. Without the adversarial loss $\mathcal{L}_{\text{adv}}$, the network may fail to generate homogenous screentones, so the generated screentones for these regions may be noisy (Fig. 12(f)). Note that the difference between $\mathcal{L}_{\text{srec}}$ and




(a) Input (b) Photo-shop[2021] (c) [Barnes et al. 2009] (d) [Yu et al. 2018] (e) [Zeng et al. 2019] (f) [Nazeri et al. 2019] (g) [Xie et al. 2020] (h) Ours

Fig. 10. The inpainted results generated by various state-of-the-arts inpainting methods. The mask is labeled in green. KaerimichiNoMajo ©A-10



(a) Input (b) Masked input (c) w/o $\mathcal{L}_{\text{lrec}}$ (d) w/o $\mathcal{L}_{\text{srec}}$ (e) w/o $\mathcal{L}_{\text{adv}}$ and $\mathcal{L}_{\text{fm}}$ (f) w/o $\mathcal{L}_{\text{bin}}$ (g) Ours

Fig. 11. The inpainted results generated by semantic inpainting model with and without different loss terms. The ScreenVAE map is visualized after PCA.

Table 2. The quantitative evaluation by semantic inpainting model with and without different loss terms.

| Methods | $\tilde{\mathbf{L}}$ | | $\tilde{\mathbf{S}}$ | |
|---|---|---|---|---|
| | SSIM | LPIPS | PSNR | SSIM |
| w/o $\mathcal{L}_{\text{lrec}}$ | 0.6913 | 0.0780 | 22.7614 | 0.9224 |
| w/o $\mathcal{L}_{\text{srec}}$ | 0.6740 | 0.0833 | 22.1696 | 0.9157 |
| w/o $\mathcal{L}_{\text{adv}}$ and $\mathcal{L}_{\text{fm}}$ | 0.6288 | 0.0974 | 24.8754 | 0.9452 |
| w/o $\mathcal{L}_{\text{bin}}$ | 0.6185 | 0.1100 | 24.2036 | 0.9402 |
| Ours | **0.7443** | **0.0638** | **29.7900** | **0.9745** |

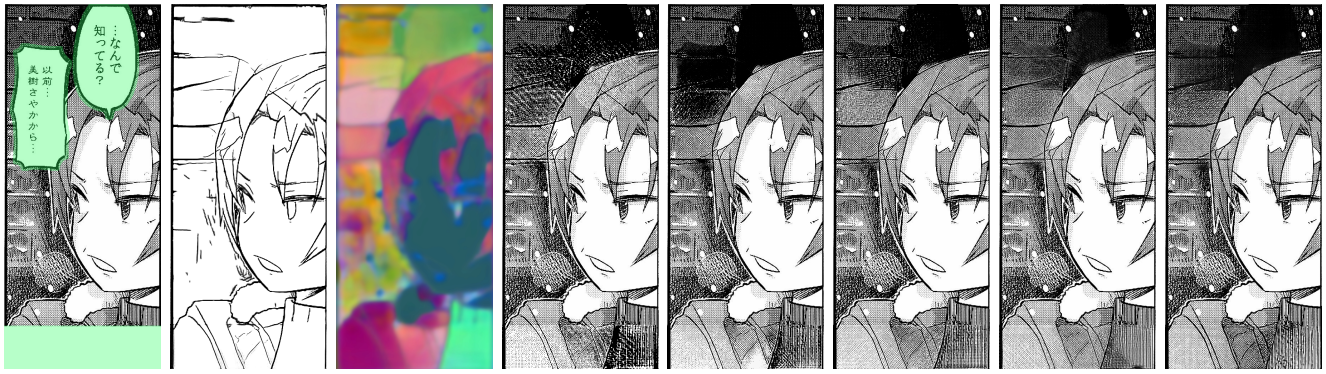
Table 3. The quantitative evaluation by appearance synthesis model with and without different loss terms.

| Methods | PSNR | SSIM | LPIPS |
|---|---|---|---|
| w/o $\mathcal{L}_{\text{mrec}}$ | 12.2863 | 0.6188 | 0.0585 |
| w/o $\mathcal{L}_{\text{srec}}$ | 14.3382 | 0.6905 | 0.0478 |
| w/o $\mathcal{L}_{\text{adv}}$ | 14.0485 | 0.6724 | 0.0460 |
| w/o $\mathcal{L}_{\text{bin}}$ | 13.4857 | 0.6270 | 0.0541 |
| Ours | **14.9836** | **0.6935** | **0.0449** |

$\mathcal{L}_{\text{adv}}$ is that $\mathcal{L}_{\text{srec}}$ encourages the generated results to be filled with screentones but the screentones may be a little noisy. In the space of ScreenVAE map, similar screentones can be projected to similar representations including the screentones with noise. $\mathcal{L}_{\text{adv}}$ can encourage the generated screentones to be homogenous over region. As we see in Fig. 12(g), when we remove the binarization loss $\mathcal{L}_{\text{bin}}$, blurry results may be resulted. In comparison, combining all losses can help the network to recognize different types of screentones and generate clear and consistent screentone for the disoccluded regions (Fig. 12(h)). We also quantitatively evaluate the models with different losses. As listed in Table 3, the combined loss achieves the best numerical performance.

*Recurrent updating module in Semantic inpainting model.* To verify the effectiveness of our recurrent updating module, we conduct an ablation study on the effect of using different number of iteration. Due to the limited GPU memory, we only compare the results with iteration count $t \in [1, 5]$ as larger $t$ requires to train with a smaller patch size which will inevitably degrade the performance. Table 4 shows the quantitative evaluations for different iteration counts. Other than evaluating the quality of the inpainted manga, we also evaluate the quality of structural line which is a visually apparent component in manga. SSIM and IPLPS are used to measure the inpainted quality. Note that SSIM metric is alignment-sensitive, while LPIPS is more reasonable as it is more perceptually related and alignment-insensitive. We quantitatively evaluate the results under three different resolutions including $256 \times 256$, $512 \times 512$ and $1024 \times 1024$. The masks are directly resized to the required resolution while the testing images are cropped on the same image set. From the statistics, we can see that with resolution $256 \times 256$, the content can be inpainted within an iteration. When the resolution increases, there is a trend that higher iteration count is needed to generate more plausible inpainted results, e.g. $N = 3$ under $512 \times 512$ and $N >= 3$ under $1024 \times 1024$. However, the optimal number of iteration is inconclusive from the statistics. Specifically, for resolution of $1024 \times 1024$, although the LPIPS may get worse with more iterations, we still prefer to inpaint with more iterations for larger missing regions, based on our experience. We empirically set $N = 5$ in this paper to inpaint with more structural lines as the testing images are with resolution about $1024 \times 1024$ and masked over 20% area.

Note that SSIM, PSNR, and LPIPS have limitations in evaluating the quality of inpainted content. They all cannot account for the non-uniqueness nature of inpainted content. As there may exist multiple visually plausible and reasonable inpainted content, any metric comparing to the groundtruth implicitly assumes that the inpainting must be unique. Moreover, the LPIPS model we used in



(a) Manga (b) Inpainted Structural line (c) Inpainted ScreenVAE map (d) w/o $\mathcal{L}_{\text{mrec}}$ (e) w/o $\mathcal{L}_{\text{srec}}$ (f) w/o $\mathcal{L}_{\text{adv}}$ (g) w/o $\mathcal{L}_{\text{bin}}$ (h) Ours

Fig. 12. The inpainted results generated by the appearance synthesis model with and without different loss terms. The mask is labeled in green. The ScreenVAE map is visualized after PCA. KaerimichiNoMajo ©A-10

Table 4. The quantitative comparison with different iterations.

| Resolution | Iterations | $\tilde{\mathbf{L}}$ | | $\tilde{\mathbf{I}}$ | |
|---|---|---|---|---|---|
| | | SSIM | LPIPS | SSIM | LPIPS |
| 256 × 256 | N=1 | **0.8432** | **0.0430** | **0.6352** | 0.0637 |
| | N=2 | 0.8362 | 0.0432 | 0.6318 | **0.0634** |
| | N=3 | 0.8290 | 0.0448 | 0.6247 | 0.0644 |
| | N=4 | 0.8227 | 0.0465 | 0.6152 | 0.0657 |
| | N=5 | 0.8168 | 0.0484 | 0.6049 | 0.0672 |
| 512 × 512 | N=1 | **0.7611** | 0.0692 | **0.5470** | 0.0725 |
| | N=2 | 0.7512 | 0.0650 | 0.5460 | 0.0713 |
| | N=3 | 0.7443 | **0.0638** | 0.5373 | **0.0706** |
| | N=4 | 0.7386 | 0.0639 | 0.5293 | 0.0725 |
| | N=5 | 0.7339 | 0.0642 | 0.5286 | 0.0729 |
| 1024 × 1024 | N=1 | **0.6844** | 0.0905 | 0.4573 | 0.0921 |
| | N=2 | 0.6743 | 0.0839 | 0.4578 | 0.0908 |
| | N=3 | 0.6743 | 0.0791 | 0.4504 | **0.0901** |
| | N=4 | 0.6706 | 0.0774 | **0.4579** | 0.0907 |
| | N=5 | 0.6669 | **0.0767** | 0.4392 | 0.0922 |

our experiments is trained based on natural photographs which may be substantially different from our manga images in nature.

### 5.4 Limitations

Our framework still suffers from some limitations. The currently trained model sometimes cannot generate rarely seen objects in the background. This is mainly due to the insufficiency of the data containing purely background. Most training images are covered with human characters. For example, in Fig. 14, the fence on the background cannot be well inpainted. Meanwhile, our proposed method might not be able to achieve high-quality content-rich inpainting if there is very little (or even no) contextural information provided. An example can be found in Fig. 15. Our model fails to recover the cabinet in the bottom-left while we manage to recover the top-right wall structure. We believe that this ill-posed problem of lacking prior contextural cues can be significantly relieved by providing additional constraints. Such additional constraints can be from a given template dataset or simply the user hint. We shall consider this as one of our future works.

## 6 CONCLUSION

In this paper, we made the first attempt to tackle the challenging manga inpainting task and obtain high-quality results. We proposed to subdivide the manga inpainting task into structural line inpainting and ScreenVAE map inpainting to better address, and hence learn, the semantic difference between structure and screentone. We further adopted recurrent updating scheme to tackle large disoccluded regions and improve the visual quality of the inpainting. Our method generates seamless high-quality inpainting results in terms of both plausible structures and well-matched screentones. Experiments show that we outperform existing state-of-the-art inpainting techniques that are originally designed for natural images. Our method may benefit various manga reproduction applications, such as manga localization and motion manga generation.

## REFERENCES

2021. Photoshop. https://www.photoshop.com.

Yuji Aramaki, Yusuke Matsui, Toshihiko Yamasaki, and Kiyoharu Aizawa. 2016. Text detection in manga by combining connected-component-based and region-based classifications. In *2016 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2901–2905.

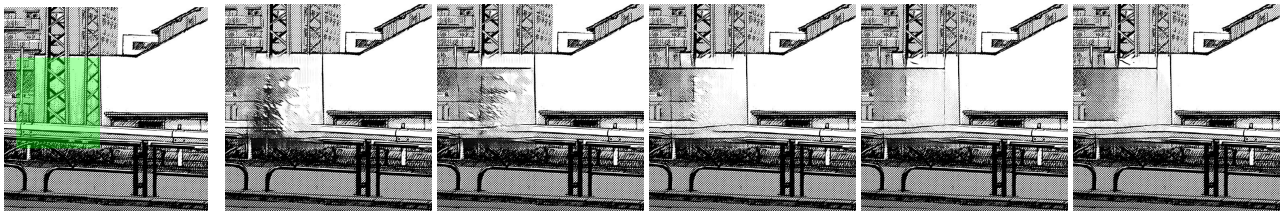
Connelly Barnes, Eli Shechtman, Adam Finkelstein, and Dan B Goldman. 2009. PatchMatch: A randomized correspondence algorithm for structural image editing. In *ACM Transactions on Graphics (ToG)*, Vol. 28. ACM, 24.

Antonio Criminisi, Patrick Pérez, and Kentaro Toyama. 2004. Region filling and object removal by exemplar-based image inpainting. *IEEE Transactions on image processing* 13, 9 (2004), 1200–1212.

Soheil Darabi, Eli Shechtman, Connelly Barnes, Dan B Goldman, and Pradeep Sen. 2012. Image melding: Combining inconsistent images using patch-based synthesis. *ACM Transactions on graphics (TOG)* 31, 4 (2012), 1–10.

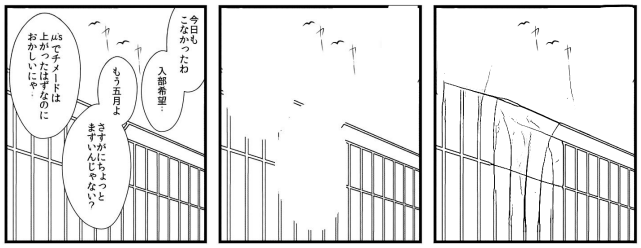
Sarah F Frisken, Ronald N Perry, Alyn P Rockwood, and Thouis R Jones. 2000. Adaptively sampled distance fields: A general representation of shape for computer graphics. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*. 249–254.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. *Advances in neural information processing systems* 27 (2014), 2672–2680.



(a) Input manga (b) N=1(LPIPS:0.1245) (c) N=2(LPIPS:0.1180) (d) N=3(LPIPS:0.1107) (e) N=4(LPIPS:0.1090) (f) N=5(LPIPS:0.1095)

Fig. 13. The inpainted results generated by semantic inpainting model with different iterations. Large masked region requires more iterations. "Give My Regards to Black Jack" ©Shuho Sato




(a) Structural line (b) Masked Structural line (c) Inpainted Structural line

Fig. 14. Some background may not be inpainted with meaningful structures, potentially due to the lack of sufficient training data.


(a) Manga (b) Inpainted structural line (c) Inpainted manga

Fig. 15. Our method may failed to generate complicated content within a large disoccluded region if not enough contextual information facilitate. "Give My Regards to Black Jack" ©Shuho Sato


Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*. 1026–1034.

Jia-Bin Huang, Sing Bing Kang, Narendra Ahuja, and Johannes Kopf. 2014. Image completion using planar structure guidance. *ACM Transactions on graphics (TOG)* 33, 4 (2014), 1–10.

Satoshi Iizuka, Edgar Simo-Serra, and Hiroshi Ishikawa. 2017. Globally and locally consistent image completion. *ACM Transactions on Graphics (ToG)* 36, 4 (2017), 1–14.

Kota Ito, Yusuke Matsui, Toshihiko Yamasaki, and Kiyoharu Aizawa. 2015. Separation of Manga Line Drawings and Screentones.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

Chengze Li, Xueting Liu, and Tien-Tsin Wong. 2017. Deep Extraction of Manga Structural Lines. *ACM Transactions on Graphics (SIGGRAPH 2017 issue)* 36, 4 (July 2017), 117:1–117:12.

Guilin Liu, Fitsum A. Reda, Kevin J. Shih, Ting-Chun Wang, Andrew Tao, and Bryan Catanzaro. 2018. Image Inpainting for Irregular Holes Using Partial Convolutions. In *The European Conference on Computer Vision (ECCV)*.

Hongyu Liu, Bin Jiang, Yi Xiao, and Chao Yang. 2019. Coherent semantic attention for image inpainting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 4170–4179.

Xueting Liu, Chengze Li, and Tien-Tsin Wong. 2017. Boundary-aware texture region segmentation from manga. *Computational Visual Media* 3, 1 (2017), 61–71.

Mehdi Mirza and Simon Osindero. 2014. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784* (2014).

Kamyar Nazeri, Eric Ng, Tony Joseph, Faisal Qureshi, and Mehran Ebrahimi. 2019. Edgeconnect: Generative image inpainting with adversarial edge learning. In *The IEEE International Conference on Computer Vision (ICCV) Workshops*.

Seoung Wug Oh, Sungho Lee, Joon-Young Lee, and Seon Joo Kim. 2019. Onion-peel networks for deep video completion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 4403–4412.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. Automatic differentiation in PyTorch. (2017).

Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. 2016. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2536–2544.

Yurui Ren, Xiaoming Yu, Ruonan Zhang, Thomas H Li, Shan Liu, and Ge Li. 2019. Structureflow: Image inpainting via structure-aware appearance flow. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 181–190.

Kazuma Sasaki, Satoshi Iizuka, Edgar Simo-Serra, and Hiroshi Ishikawa. 2017. Joint gap detection and inpainting of line drawings. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 5725–5733.

Karen Simonyan and Andrew Zisserman. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* (2014).

Yuhang Song, Chao Yang, Zhe Lin, Xiaofeng Liu, Qin Huang, Hao Li, and C-C Jay Kuo. 2018a. Contextual-based image inpainting: Infer, match, and translate. In *Proceedings of the European Conference on Computer Vision (ECCV)*. 3–19.

Yuhang Song, Chao Yang, Yeji Shen, Peng Wang, Qin Huang, and C-C Jay Kuo. 2018b. Spg-net: Segmentation prediction and guidance network for image inpainting. *arXiv preprint arXiv:1805.03356* (2018).

Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. 2004. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing* 13, 4 (2004), 600–612.

Minshan Xie, Chengze Li, Xueting Liu, and Tien-Tsin Wong. 2020. Manga filling style conversion with screentone variational autoencoder. *ACM Transactions on Graphics (TOG)* 39, 6 (2020), 1–15.

Fisher Yu and Vladlen Koltun. 2015. Multi-scale context aggregation by dilated convolutions. *arXiv preprint arXiv:1511.07122* (2015).

Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. 2018. Generative image inpainting with contextual attention. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 5505–5514.

Yanhong Zeng, Jianlong Fu, Hongyang Chao, and Baining Guo. 2019. Learning pyramid-context encoder network for high-quality image inpainting. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 1486–1494.

Yu Zeng, Zhe Lin, Jimei Yang, Jianming Zhang, Eli Shechtman, and Huchuan Lu. 2020. High-resolution image inpainting with iterative confidence feedback and guided upsampling. In *European Conference on Computer Vision*. Springer, 1–17.

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. 2018. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 586–595.