

Interactive Segmentation for Manga using Lossless Thinning and Coarse Labeling

Yuji Aramaki*, Yusuke Matsui*, Toshihiko Yamasaki*, Kiyoharu Aizawa*

The University of Tokyo, Tokyo, Japan

* E-mail: {aramaki, matsui, yamasaki, aizawa}@hal.t.u-tokyo.ac.jp Tel/Fax: +81-3-5841-6761

Abstract—*Manga* (Japanese comics) are popular all over the world. Applications using manga, such as automatic layout suggestion and character composition, have been proposed recently. To implement these applications, important but time-consuming manga segmentation is required. We propose a scribble-based interactive segmentation method for manga images. Based on Smart Scribbles, we propose (1) *lossless thinning*, which enables the handling of complex raster images such as manga, and (2) *coarse labeling*, which drastically reduces the computational cost of labeling and enables accurate segmentation in real time. Our experimental results show that the proposed method efficiently produces much more accurate results than does the Quick Selection Tool in Adobe Photoshop® or the connected-component labeling method.

I. INTRODUCTION

Manga (Japanese comics) are read widely in many countries. As more and more manga have been issued recently in electronic form, research on them is producing new applications. Kopf and Lischinski [5] proposed a method for vectorizing old printed comics and demonstrated a new application for re-rendering the scenes of comics from different viewpoints. Matsui et al. [6] proposed a retargeting method tailored for manga, whereby a manga image is resized naturally with characters preserved and unnecessary backgrounds discarded. Colorization of manga has been proposed to colorize black-and-white manga images [9]. Cao et al. produced several methods for the production of stylistic manga layouts [3], and the natural composition of manga elements [4]. They constructed a manga database, annotated the contents of each title, generated a probabilistic model of the structure of manga outlines, and generated algorithms that produce attractive new manga layouts or compositions. As an example of a commercial service, MANGAPOLO¹, provides a video-style manga, whereby manga images are divided into frames to be shown in the form of a video.

In any of these applications, the first and important step is to segment a target object such as a frame or a character. Currently, general image-processing applications such as Adobe Photoshop are used for this purpose. As a result, the segmentation of manga images is a manual and time-consuming process. This is particularly critical whenever we wish to process many manga images. Therefore, efficient manga segmentation is an essential requirement for many applications.

Because manga images are black-and-white line drawings and do not have color distributions or gradients, conventional interactive segmentation methods such as GrabCut [10] are not effective for processing manga images. There are only a few effective segmentation methods designed for line drawings. Though the simplest method involves label-based interactive selection [6], where all the pixels across an input scribble are labeled simply by considering the connectivity of black pixels using connected-component labeling [11], this method is too simple for application to the complex scenes that are typical of manga images. Another one is as a part of a colorization method [9], but it cannot be used for other purposes such as character extraction, because the method does not specify the line segments of objects but do the flat areas of objects. Smart Scribbles [7] is a state-of-the-art interactive segmentation method for line drawings. It produces satisfactory results for relatively simple line drawings. It can accept only image that meets three conditions: being a vector image, being composed of a relatively small number of strokes, and containing strokes with temporal information. It is, therefore, difficult to apply Smart Scribbles directly to manga images, because they involve raster images with many strokes that do not have temporal information. For example, even after vectorizing an 800×400 manga image, we found that Smart Scribbles without temporal information required 8.6 seconds per scribble.

In this paper, we propose a scribble-based interactive segmentation method for manga images. Our method can complete all the required processes for a single scribble in real time. Though our method is the same as that proposed in a previous work [2], we conducted more usability tests for the method. The experimental results show that, when using our method, the outcomes were more satisfactory for users than when they used the selection based on connected-component labeling or the Quick Selection Tool in Adobe Photoshop.

II. PROPOSED METHOD

To achieve the segmentation of manga images, we propose two novel approaches: *lossless thinning*, which is based on a form of thinning [12] that enables vector-raster reversible conversion, and *coarse-to-fine labeling*, which drastically reduces the computational cost of labeling and enables accurate segmentation in real time.

A flowchart for the proposed method is shown in Figure 1. The input source image is vectorized automatically (see

¹<https://www.youtube.com/user/MANGAPOLO>

©Ken Akamatsu

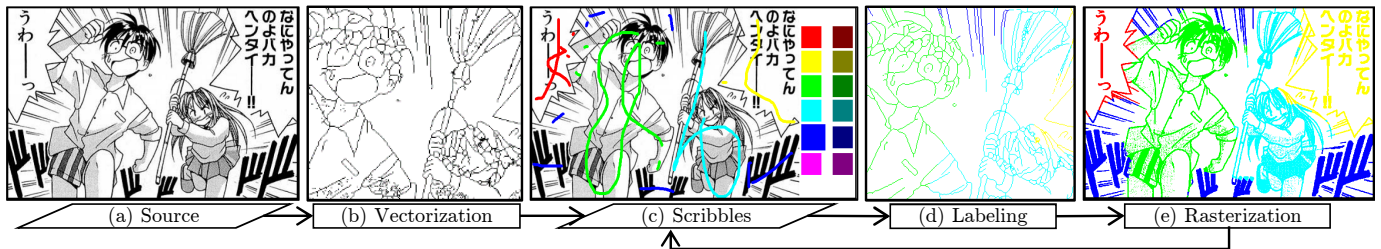


Fig. 1. A flowchart of the proposed method. (a) A source manga image. (b) The thinned and vectorized image visualized in terms of line segments. (c) After selecting a label as a color in a palette, the user draws scribbles with that label over the image. (d) Using the scribbles, the labels are assigned to the line segments. (e) The labels are propagated to all the pixels and the segmented result is shown to the user.

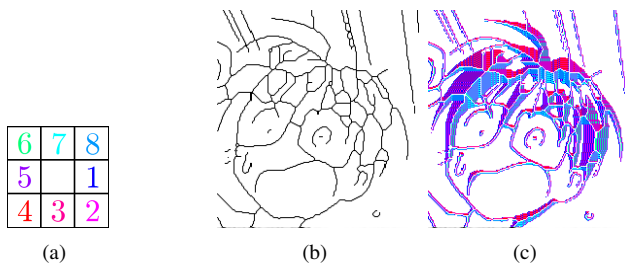


Fig. 2. (a) The direction numbers and colors. (b) The thinned image for the face of the boy in Figure 1. (c) The thinning direction map for the face.

Figure 1(b)). Next, the user selects a label by picking a color from a color palette on the side, and draws scribbles with that label (color) over the image (see Figure 1(c)). The labeling and rasterization are then performed automatically, giving a (temporary) result for the segmentation, as shown (see Figure 1(d)). The user repeats the scribble drawing until satisfactory convergence is achieved, after which the final segmented result is produced (see Figure 1(e)). We describe our method step by step in the following subsections.

A. Vectorization

To apply a segmentation method for vector line drawings to a raster manga image, we first need to vectorize them before the next (labeling) step. Given a gray-scale manga image (see Figure 1(a)), we apply lossless thinning. The source raster image is first converted to a binary image using a binarization method [8]. We then apply thinning [12] to the binary image, whereby each line in the source image is thinned to one-pixel width (see Figure 2(b)). At the same time, we compute a thinning direction map (see Figure 2(c)). Here, each pixel is assigned a number representing the direction in which black pixels were shrunk during the thinning. In Figure 2(c), the direction is visualized by the colors defined in Figure 2(a). The thinning direction map is crucial to the accurate assignment of labels to all pixels in the rasterization. Next, the thinned image is vectorized into line segments by considering the connectivity of thinned lines (see Figure 1(b)).

B. Labeling

After vectorizing the image, the user draws scribbles with labels identifying objects over the source image (see Figure

1(c)). Though the user can see only the source image, the labeling is applied to the line segments which are maintained internally. We use Smart Scribbles to assign the labels finely, but its computational complexity is $O(n^3)$, where n is the number of the line segments. This is because it has to solve a graph-cut problem for a complete graph, each node of which corresponds to a line segment. Because applying it directly to the whole line segments can take excessive time, we apply our coarse labeling in advance, whose computational complexity is only $O(n)$. The combination of the coarse but fast labeling and the fine but expensive one enables accurate selection in real time.

First, we assign the label of the nearest scribble to all line segments “close” to the segments of a scribble:

$$\forall l_* \in L, \forall i \in S(l_*), \phi_i = l_* \quad (1)$$

Here, L is the set of the labels, $S(l_*)$ is the set of the line segments sufficiently close to the scribbles with the label l_* , and ϕ_i is the label for the line segment i being considered. The threshold for the closeness of $S(l_*)$ is determined dynamically:

$$threshold = D_i - n \cdot D_s + k \cdot D_t \quad (k = 0, 1, 2, \dots) \quad (2)$$

Here, D_i , D_s , and D_t are constant values, n is the number of input labels (i.e., the number of the colors used), and k is incremented until the number of unlabeled line segments becomes small enough. In this way, the range of labeled line segments is expanded gradually (see Figure 3 (a) and (b)). In our experiments, D_i , D_s , and D_t were set to 60, 5, and 5, respectively, and the stop-condition number k was fixed at 2,000.

Second, we assign the labels produced by Smart Scribbles to the line segments remaining after the previous labeling process. Smart Scribbles assigns labels in such a way as to minimize the energy function:

$$\sum_{i,j \in S'} V_{i,j}(\phi_i, \phi_j) + \lambda \sum_{i \in S'} D_i(\phi_i), \quad (3)$$

where S' is the set of the remaining line segments, $V_{i,j}$ is the cost of the labeling based on the similarity between two line segments i and j , D_i is the affinity between the scribbles and line segment i , and λ is the controlling parameter between two

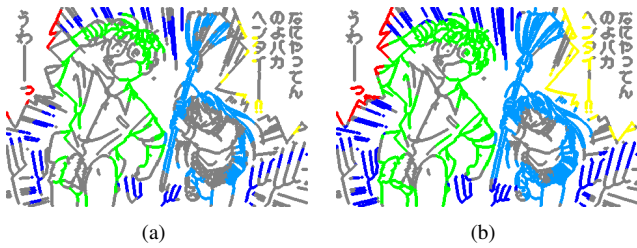


Fig. 3. Temporary labeling results for the input of Figure 1 at the stage for which (a) $k = 1$ and (b) $k = 5$.

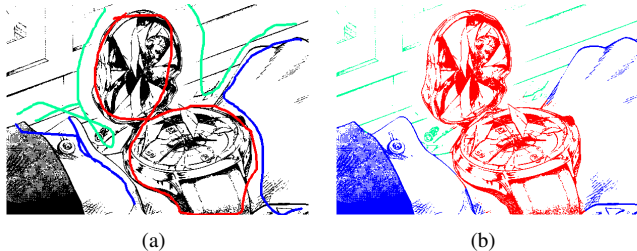


Fig. 4. An example of our method. (a) A source image with scribbles. (b) The result image. ©Ryusei Deguchi

terms, which is set to 4 in our experiments. $V_{i,j}$ and D_i are composed of the functions for distance and direction.

Following this step, all line segments in the vectorized image will be labeled. However, because Smart Scribbles is designed for relatively simple images, there may be some line segments to which the wrong label has been assigned. Empirically, most of the line segments i with a wrong label will satisfy these two conditions:

$$\begin{cases} \forall j \in S', V_{i,j}(\phi_i, \phi_j) = 0, \\ \forall \phi_i \in L, D_i(\phi_i) = 1. \end{cases} \quad (4)$$

$$\quad (5)$$

That is, they can equally easily be labeled the same as any other line segment in the image and can equally awkwardly be labeled the same as any scribbles. We modify these wrong labels by a simple procedure that assigns the label of the nearest line segment in the image to line segments with a wrong label.

C. Rasterization

Finally, we convert the labeled vector image back to a raster image. Because the labels are assigned to all the line segments by the labeling procedure, we simply assign the label to the pixels in the original raster image. All the black pixels in the thinned image will then be labeled (see Figure 1(d)).

However, even after rasterizing the labeled vector image, only the labeled thinned image is obtained. Our thinning direction map makes it possible to propagate labels to all unlabeled pixels in the source image. By using this map, we can trace the thinning process back and assign labels to the unlabeled pixels. As a result, all the pixels can be correctly labeled, even if there are some black-filled regions, such as the head of the boy in Figure 1(e). This is important because vector-based methods such as Smart Scribbles cannot handle the filled areas that often appear in manga images.

We show a result for the proposed method in Figure 4.

III. EXPERIMENTS

We conducted usability tests to compare our method to the selection based on connected-component labeling [6] and the Quick Selection Tool, making the assumption that users would prefer easier operations. Note that the original Smart Scribbles method was not included in the tests because it cannot accept raster images and was much slower than the other approaches. In the experiments, the participants were shown a source image and the colored ground truth for the image, and were asked to segment the source image to the same as the ground truth. Each participant performed this task for six images in succession. The images were selected for their variety in terms of line thickness, the number of objects, and the background tones. We implemented our system using a PC with a 1.8 GHz Intel Core i7 CPU and 8 GB RAM, as well as a 21-inch graphic tablet and pen (WACOM DTZ-2100) for the participants to input scribbles.

We counted the number of labeled pixels in the ground truth in advance, and measured the processing time, the number of labeled pixels in the result, and the number of pixels with the same label in the result and in the ground truth. Segmentation accuracy was obtained via the F-measure, which are defined by the three numbers.

Each experiment took one of two forms, depending on the type of application the tasks were assuming. The two types of task are described below.

A. Task aiming for accurate selection

For the first task, we assumed that it is necessary for users to segment images as accurately as possible, as is required for the colorization of manga. We asked four participants to segment target objects as accurately as possible. We did not set time constraints, and the participants finished the task when they were satisfied with their results. Here, the proposed method was compared with the Quick Selection Tool, because this task requires careful selection. All the participants were new to both our system and the Quick Selection Tool.

We could not find a clear difference between the proposed method and the Quick Selection Tool in terms of the time required for the task, as shown in Figure 5, whereas there were great differences between the participants. However, Figure 6 shows that the average F-measure for the proposed method exceeds that for the Quick Selection Tool for all source images.

B. Task aiming for fast selection

The second task assumed that users needed to segment images rapidly, but the results did not have to be very accurate, as in character annotation. For each image, we defined a limit to the processing time based on the difficulty of segmentation, ranging from 100 to 200 seconds. Seven participants, different from those performing the first task, were instructed to segment target objects as accurately as possible within the specified time. For this task, we compared our method with connected-component labeling, which is suited to this task because of its simplicity and intuitiveness.

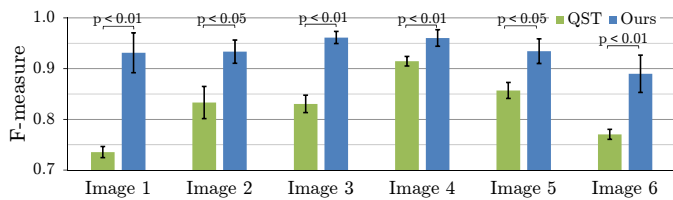


Fig. 5. The accuracy of the task aiming for accurate selection.

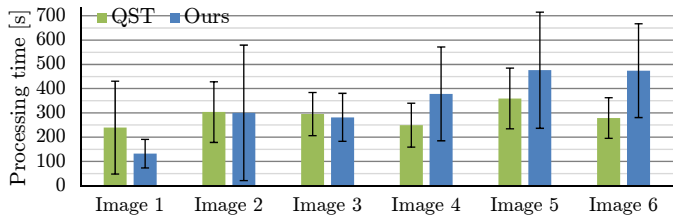


Fig. 6. The processing time of the task aiming for accurate selection.

As seen in Figure 7, our method performed more precisely than did the connected-component labeling. In addition, the variation of F-measures among the participants is smaller for our method than for the connected-component labeling. These results show that the proposed method enables accurate interactive segmentation even for novice workers. We show a time series for the F-measure in Figure 8. Our method achieved segmentation more quickly and had less fluctuation than did the connected-component labeling.

IV. CONCLUSIONS AND LIMITATIONS

We proposed an approach for segmenting manga images interactively that comprises vectorization, labeling, and rasterization. Our *lossless thinning* method enabled reversible vector-raster conversion, and our *coarse labeling* method made it possible to segment complicated manga images. We have verified that our proposed method can be more effective than previous tools and methods.

On the early stages of segmentation, because only a few scribbles are input, our method often produces very poor results. In addition, in a few cases where the tiny errors occur on the boundary between the objects, our method cannot modify the errors whatever scribbles are added.

ACKNOWLEDGMENT

Manga109 dataset [1] is used for the experiments. We thank authors of manga for the use of their images. This work was supported by the Strategic Information and Communications R&D Promotion Programme (SCOPE).

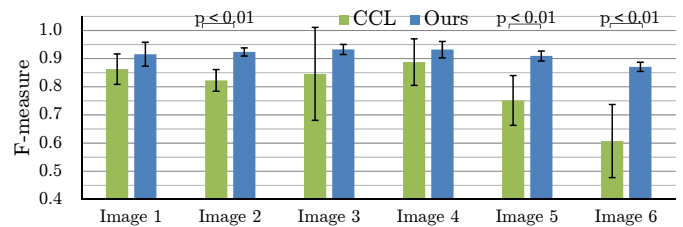


Fig. 7. The accuracy of the task aiming for fast selection.

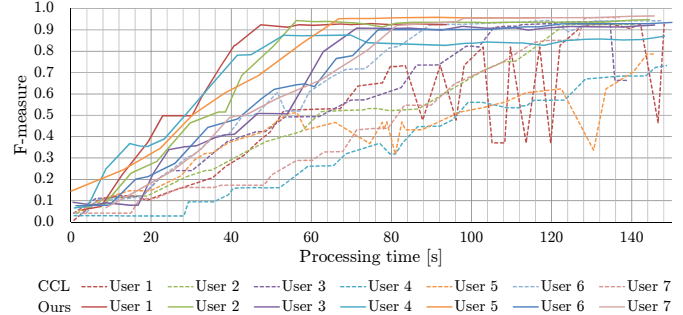


Fig. 8. The time series of the F-measure in the task aiming for fast selection of Image 4.

REFERENCES

- [1] <http://www.manga109.org/> (Manga dataset for academic research).
- [2] Yuji Aramaki, Yusuke Matsui, Toshihiko Yamasaki, and Kiyoharu Aizawa. Interactive segmentation for manga. In *ACM SIGGRAPH 2014 Posters*, page 66. ACM, 2014.
- [3] Ying Cao, Antoni B. Chan, and Rynson W. H. Lau. Automatic stylistic manga layout. In *ACM Transactions on Graphics*, volume 31, pages 140:1–140:10. ACM, 2012.
- [4] Ying Cao, Rynson W. H. Lau, and Antoni B. Chan. Look over here: Attention-directing composition of manga elements. In *ACM Transactions on Graphics*, volume 33. ACM, 2014.
- [5] Johannes Kopf and Dani Lischinski. Digital reconstruction of halftoned color comics. In *ACM Transactions on Graphics*, volume 31, pages 140:1–140:10. ACM, 2012.
- [6] Yusuke Matsui, Toshihiko Yamasaki, and Kiyoharu Aizawa. Interactive manga retargeting. In *ACM SIGGRAPH 2011 Posters*, page 35. ACM, 2011.
- [7] Gioacchino Noris, Daniel Šykora, Ariel Shamir, Stelian Coros, Brian Whited, Maryann Simmons, Alexander Hornung, Markus Gross, and Robert Sumner. Smart scribbles for sketch segmentation. In *Computer Graphics Forum*, volume 31, pages 2516–2527, 2012.
- [8] Nobuyuki Otsu. A threshold selection method from gray-level histograms. *Automatica*, 11(285-296):23–27, 1975.
- [9] Yingge Qu, Tien-Tsin Wong, and Pheng-Ann Heng. Manga colorization. In *ACM Transactions on Graphics*, volume 25, pages 1214–1220. ACM, 2006.
- [10] Carsten Rother, Vladimir Kolmogorov, and Andrew Blake. Grabcut: Interactive foreground extraction using iterated graph cuts. In *ACM Transactions on Graphics*, volume 23, pages 309–314. ACM, 2004.
- [11] Kenji Suzuki, Isao Horiba, and Noboru Sugie. Linear-time connected-component labeling based on sequential local operations. *Computer Vision and Image Understanding*, 89(1):1–23, 2003.
- [12] T.Y. Zhang and Ching Y. Suen. A fast parallel algorithm for thinning digital patterns. *Communications of the ACM*, 27(3):236–239, 1984.