Interactive Edge-Aware Segmentation of Character Illustrations for Articulated 2D Animations

Takumi Yajima Yoshihiro Kanamori Yuki Endo Jun Mitani
University of Tsukuba, Japan
Email: takuyajima0@gmail.com, {kanamori,endo,mitani}@cs.tsukuba.ac.jp

***Abstract*—Fascinating articulated animations can be created from character illustrations using commercial tools such as Live2D or spine. However, to compose articulated deformable models, artists have to draw each body part separately or segment input illustrations into several body parts manually. In this paper, we present an efficient technique for the segmentation of character illustrations with simple user interactions. The user first draws strokes (i.e., scribbles) on an input illustration to roughly specify body parts, and then the system segments the input character accordingly. Unlike an existing method for this interactive segmentation, our method has the following three advantages. First, our method recognizes weak colored edges as body-part boundaries, which can be missed by the existing method. Second, it lets the user directly specify body-part boundaries around articulations, which are not necessarily image edges and thus often missed by the existing method. Third, our method automatically enlarges body parts around articulations so that adjacent body parts have some overlaps and the resultant character models can be animated without exhibiting gaps between adjacent body parts. We demonstrate that the proposed system enables more efficient segmentation of character illustrations than the conventional technique.**

***Keywords*-multi-label segmentation; graphcut; 2D animation;**

I. Introduction

The conventional production of hand-drawn 2D animations imposes labor-intensive work of drawing hundreds or thousands of illustrations on animation artists. Recent commercial tools such as Live2D or spine have enabled more efficient production of vectorized 2D animations from character illustrations. However, for such vectorized animations, artists must draw each body part separately or decompose input illustrations into several body parts manually to compose articulated deformable models.

The decomposition process is essentially a task of *multi-label segmentation* (also known as *multi-class classification* in the field of machine learning), and can be assisted using an existing interactive technique *LazyBrush* [10]. In LazyBrush, the user draws colored strokes (i.e., scribbles) on an input illustration to roughly specify segments, and then the system segments the illustration accordingly by assigning user-specified colors to pixels. However, if we apply LazyBrush to character illustrations in order to segment body parts for articulated animations, we encounter the following three problems. First, LazyBrush recognizes image edges as segment boundaries, but often misses weak image edges in color illustrations. Second, image edges are not necessarily located at articulations of the input character, and thus LazyBrush often mis-detects body-part boundaries. These two problems cause inappropriate segmentations for articulated models. Third, for articulated animations, body parts should be slightly larger around articulations than the original segmentation so that the adjacent body parts have some overlaps and the resultant models can be animated without exhibiting gaps between body parts, which LazyBrush does not account for.

In this paper, we propose an interactive segmentation method that addresses the abovementioned problems as follows. First, our method enhances the weak image edges using histogram equalization while reducing noise via additional smoothing. Second, our system lets the user directly draw body-part boundaries with strokes. Third, it automatically generates slightly dilated shapes of body parts around articulations, by calculating additional circular shapes around body-part boundaries. We demonstrate that the proposed system enables more efficient segmentation of character illustrations than the conventional technique.

II. Related work

Image segmentation is an essential task in image processing, and has been long studied in computer vision and graphics. Particularly, foreground-background segmentation has been rigorously studied. It is further categorized to hard segmentation [6], [1] and soft segmentation [2], [8] (also known as *image matting*). Hard segmentation makes binary decisions to pixels by assigning foreground or background labels. On the other hand, soft segmentation allows transitions between foreground and background, and pixels have real values indicating the likelihood of being foreground or background. While these methods are for binary segmentation, we require multi-label segmentation to separate several body parts in character illustrations.

Interactive colorization of grayscale images can be regarded as multi-label segmentation where labels are colors specified with user scribbles. Colorization techniques for natural images [5], [3], [11] have been widely studied, but they often suffer from handling illustrations due to no-texture regions, high-contrast edges, or repetitive patterns like screentones. Sýkora et al. [9] proposed a method for colorizing black-and-white cartoons, but colors might be leaked from edge gaps and spilt out to unintended pixels. The method by Qu et al. [7] solves the color-leakage problem, but fails to propagate colors over image edges. LazyBrush [10] overcomes this problem by handling images edges not as hard constraints but as soft constraints for propagating user-specified colors.

Our task is not simple multi-label segmentation like grayscale colorization, but interactive segmentation with semantics regarding to body parts in character illustrations. We add simple extensions to LazyBrush, targeting to segmentation for articulated 2D animations.

III. Baseline method

As our method is based on LazyBrush [10], we briefly review LazyBrush in this section.

A. Algorithm of LazyBrush

LazyBrush is an interactive colorization method for grayscale illustrations, e.g., cartoons. The user specifies colors with scribbles, and the colors are then propagated via iterative graphcut-based optimization.

Suppose that the user assigns a set of colors $\mathcal{C}$ with scribbles to a set of pixels $\mathcal{P}$ in an input grayscale image I. LazyBrush assigns each un-colored pixel $p \in \mathcal{P}$ with a user-specified color $c_p \in \mathcal{C}$ by minimizing the following energy function

$$E(\mathcal{C}) = \sum_{\{p,q\}\in\mathcal{N}} V_{p,q}(c_p, c_q) + \sum_{p\in\mathcal{P}} D_p(c_p), \qquad (1)$$

where $\mathcal{N}$ indicates 4-connected neighbors, $V_{p,q}$ is the smoothness term, and D_p is the data term.

Intuitively, the smoothness term $V_{p,q}$ should be small at image edges so that different colors will be assigned on each side of an image edge. While grayscale illustrations often have pixels with small brightness values along black lines, using pixel values directly as the smoothness term is problematic because grayscale illustrations often contain large dark regions representing shading of objects. LazyBrush therefore employs edge detection using a Laplacian of Gaussian (LoG) filter to the input image I in order to obtain a filtered image $I_f = 1 - \max(0, s \cdot \mathrm{LoG}(I))$, where the negative response of the LoG filter is clamped to zero and positive values are scaled by s to match an interval of $[0, 1]$. Finally, the values in I_f is linearly mapped to an interval $[1, K]$ (where K is the perimeter of I) and used as the smoothness term $V_{p,q}$. The data term is simply set as $D_p(c_p) = \lambda \cdot K$, where λ is a user-specified constant. By setting λ smaller than 1, user-specified scribbles become no longer hard constraints, making LazyBrush insensitive to inaccurate user inputs.

Also, assuming that pixels along image borders are background and should be white, LazyBrush adds a white scribble by default along image borders.

B. Problems in Segmentation of Character Illustrations

Our goal is to segment colorful character illustrations into body parts suited for articulated animations. If we use LazyBrush as-is, we encounter the following problems.

Weak color edges: As LazyBrush targets at grayscale illustrations, colorful images must be converted to grayscale. However, image edges along regions with bright colors become weak edges, which causes edge detection in LazyBrush to fail. Figure 1 illustrates an example. While the green scribble is successfully propagated to the head and scarf, the red scribble is ignored and overwhelmed by the background white scribble (see the last paragraph in the previous subsection), leaving the input image unsegmented, i.e., the whole image is recognized as background.

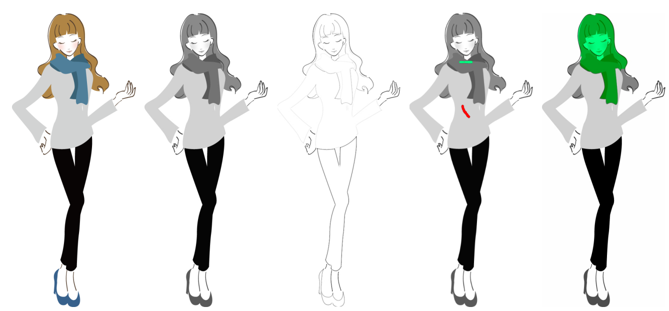

Figure 1. Failure case with weak color edges. The (a) input image is converted to (b) grayscale, and then LazyBrush applies (c) Laplacian of Gaussian filter. (d) The green scribble is appropriately handled to segment (e) the head and scarf. However, the color edges around the sweater are too weak, and thus (d) the red scribble is ignored. As a result, (e) the sweater is left unsegmented, i.e., recognized as background.

Insensitivity to articulations: To animate arms and legs of 2D characters, we would like to separate body parts around articulations such as knees and elbows so that upper/lower arms/legs remain nearly rigid and each limb bends only around its articulation. However, unless the user specifies scribbles relentlessly, segment boundaries generated by LazyBrush are image edges only, which are not necessarily located at articulations. Figure 2 shows such failure case, where the segment boundaries are not located at knees or elbows.

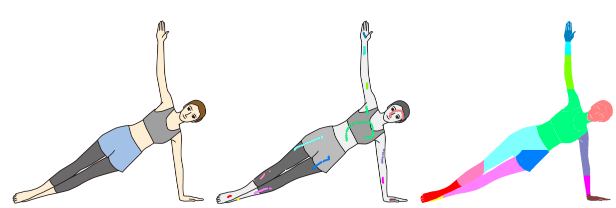

Figure 2. Failure case without explicit specifying body-part boundaries. Because there are not image edges at knees or elbows in (a) the input image, (c) the resultant body parts specified by (b) the user scribbles are not separated at the articulations.

Overlaps between adjacent body parts: Also, adjacent body parts should have some overlaps so that they do not exhibit gaps when animated, which LazyBrush does not account for. An example of body parts having some overlaps is shown in Figure 3.

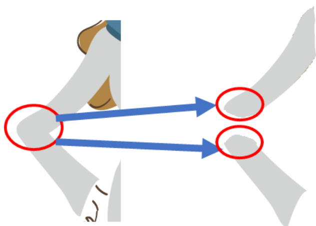

Figure 3. Segmentation of body parts with overlaps. To avoid a gap at the elbow (left) during animation, the segmented upper/lower arms (right) are slightly enlarged so that they have overlaps.

IV. Our Method

In this section, we propose solutions to the problems introduced in the previous section.

A. Filtering with edge enhancement

To better recognize weak color edges explained in Section III-B, we enhance such edges. Namely, as one of the simplest techniques, we employ histogram equalization for edge enhancement. Because histogram equalization also enhances noise, we additionally apply a Laplacian of Gaussian (LoG) filter. Figure 4 shows an example.

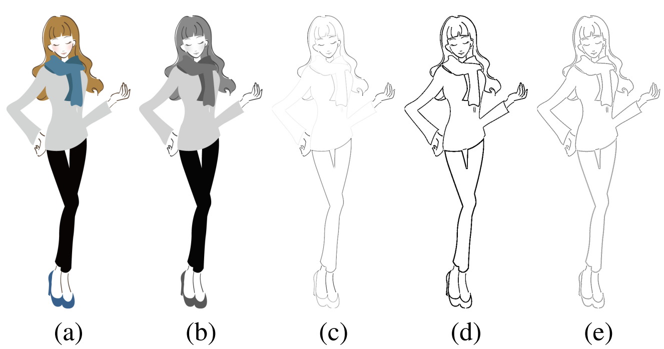

Figure 4. Edge enhancement with histogram equalization. (c) The filtered image of (b) the grayscale version of (a) the input image has weak edges around the sweater. We apply (d) histogram equalization to (c) enhance the edges, and further apply (e) an LoG filter to reduce noise.

B. Explicit inputs of body-part boundaries

We let the user explicitly draw body-part boundaries with strokes. These strokes do not have to be accurate, but rough strokes often suffice thanks to LazyBrush's insensitivity to inaccurate user inputs (see Section III-A). Specifically, user strokes are added to the filter image I_f as black edges, and iterative graphcut-based optimization is performed again with the updated smoothness term. Figure 5 illustrates an example. Note that the filtered image is not displayed to the user.

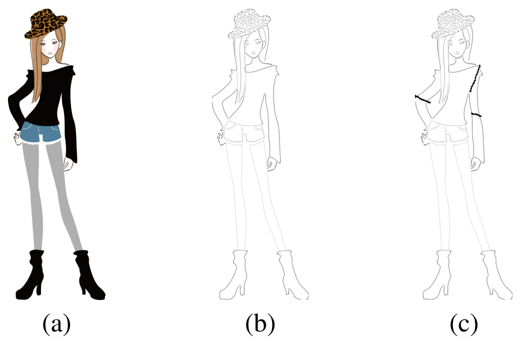

Figure 5. User-specified body-part boundaries. (b) The filtered image of (a) the input image does not have image edges at shoulders or elbows. (c) We let the user draw body-part boundaries explicitly (shown as black lines). Note that these filtered images are not visible from the user.

C. Overlaps between adjacent body parts

To ensure overlaps between adjacent body parts, our system automatically enlarges segmented body parts. Given body-part segmentation with user-specified body-part boundaries, our system calculates a circle whose diameter shares the endpoints of each user-specified body-part boundary, and then extracts the intersection of the circle and body-part segment as an overlapped region. Figure 6 shows an example.

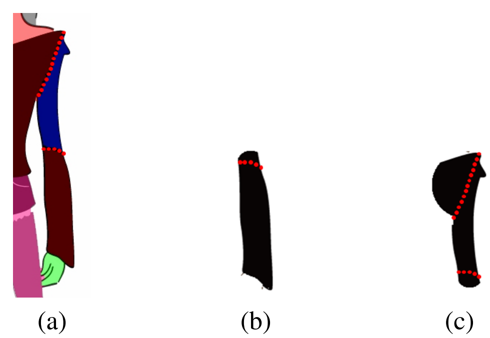

Figure 6. Overlaps in adjacent body parts. The dotted red lines indicate user-specified body-part boundaries. The segmented (b) lower and (c) upper arms are enlarged than (a) their original shapes so that they have overlaps with adjacent body parts.

V. Results

We implemented the baseline LazyBrush and our prototype system using C++, and conducted experiments on a PC with 4.20 GHz CPU and 32 GB RAM. The image resolution used in our experiments is about 1000×1000 pixels. We used the OpenCV library for histogram equalization. The computational burden caused by additional processing, e.g., histogram equalization, is negligible thanks to the optimized implementations in OpenCV.

A. Comparison with LazyBrush

Figure 7 shows comparisons with the baseline method LazyBrush and our method. From the left column to the right, we show input images, user scribbles (colored) with user-specified body-part boundaries (gray), results of the baseline LazyBrush, those with user-specified boundaries, those with edge enhancement, and our results with full extensions. In each row, not the same but as-close-as-possible scribbles were used as inputs for each segmentation, and each rectangular region is enlarged in the lower row. Note that the white thin lines in the segmentation results appear because LazyBrush does not propagate user-specified colors on image edges. These white thin lines are removed after segmentation automatically using a morphological operation.

The first row (and the enlarged images in the second row) in Figure 7 demonstrates a typical example of the issue with weak color edges. The shirt silhouette is not recognized in the filtered image of the baseline LazyBrush, and thus the input green scribble is ignored and overwhelmed by the background white scribble, making the shirt region fused to the background. With edge enhancement (in the fifth and sixth columns), the shirt region is successfully recognized, and by explicitly specifying body-part boundaries, the right arm is segmented appropriately at the elbow. Similarly, in the third and fifth rows, we can see that appropriate segmentations at articulations are accomplished with user-specified body-part boundaries.

Note that, with the original LazyBrush, the user can specify body-part boundaries by drawing different scribbles along target boundaries. However, such operation increases the number of input scribbles and the computational cost because the computational time of LazyBrush linearly depends on the numbers of scribbles and the number of pixels in the input image. For example, with our implementation, it took four seconds to segment an input illustration with five scribbles while more than eight seconds with more than 10 scribbles every time the user draws a new scribble. Additionally, our system allows the user to directly draw body-part boundaries at intended positions to obtain intended segmentation, which is also beneficial to reduce the computational time.

B. Effect of overlaps between adjacent body-parts

Figure 8 shows a comparison with and without overlaps between adjacent body-parts. The input image is segmented using our system with and without body-part overlaps. Without overlaps, we can see gaps at the knees and elbows in the screenshots of the animation sequence. Such gaps are not observed with overlaps.

C. User study

To compare the usability of the original LazyBrush and our system, we conducted an informal user study with three subjects where each subject was requested to use each tool to segment each illustration into 17 body parts (i.e., hair, face, trunk, as well as left and right upper/lower arms and legs, wrists, and ankles) after several minutes of practice.

Figure 9 shows the results. The time required for each segmentation with each tool is shown in the left. Subjects A and B were beginners of the image segmentation task, and spent more time using LazyBrush than using ours. With the original LazyBrush, the both subjects tried to segment body parts at their intended positions by drawing different scribbles along their intended boundaries. Unfortunately, such attempts did not make sense at the enlarged regions where silhouettes of the chest (in Subject A's results) and the left hand (in Subject B's results) were not recognized in the original LazyBrush. Subject C had an experience with the commercial animation tool, Live2D, and was skilled at this image segmentation. He spent almost the same time for each tool. However, with the original LazyBrush, he could not specify body-part boundaries at the shoulder or the elbow, while he could with explicitly specifying boundaries in our system.

VI. Conclusion and Future Work

In this paper, we have improved an existing method, LazyBrush, to segment character illustrations interactively for the use of articulated 2D animations. To solve the issue that weak color edges are not detected by a simple Laplacian of Gaussian (LoG) filter after RGB-to-grayscale conversion, we added histogram equalization to enhance filtered weak edges plus an LoG filter to reduce noise. By letting the user draw body-part boundaries explicitly, we allow the user to segment body parts at intended positions. Also, our system automatically enlarges segmented body parts so that adjacent body parts will have overlaps and they do not exhibit gaps during animation.

In future work, we would like to accelerate the iterative graphcut-based optimization, which is currently the computational bottleneck. Also, if we segment out a body part, the hidden pixel behind the body part becomes a blank space or a hole. While this can be filled using image retouching software, e.g., Adobe Photoshop, we would like to integrate a technique for automatic image completion, e.g., the method by Hui et al. [4].

References

[1] Rother Carsten, Kolmogorov Vladimira, and Blake Andrew. "GrabCut": Interactive Foreground Extraction Using Iterated Graph Cuts. *ACM Transactions on Graphics (TOG)*, 23(3):309–314, 2004.

[2] Yung-Yu Chuang, Brian Curless, David H Salesin, and Richard Szeliski. A bayesian approach to digital matting. In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2001)*, volume 2, pages II:264–II:271. IEEE, 2001.

[3] Yuki Endo, Satoshi Iizuka, Yoshihiro Kanamori, and Jun Mitani. DeepProp: Extracting deep features from a single image for edit propagation. *Computer Graphics Forum*, 35(2):189–201, 2016.

[4] Huang Hui, Yin Kangxue, Gong Minglun, Lischinski Dani, Cohen-Or Daniel, Ascher Uri, and Chen Baoquan. "Mind the Gap": Tele-Registration for Structure-Driven Image Completion. *ACM Transactions on Graphics (TOG)*, 32(6):174:1–174:10, 2013.

[5] Anat Levin, Dani Lischinski, and Yair Weiss. Colorization using optimization. *ACM Transactions on Graphics (TOG)*, 23(3):689–694, 2004.

[6] Eric N Mortensen and William A Barrett. Intelligent scissors for image composition. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pages 191–198. ACM, 1995.

[7] Yingge Qu, Tien-Tsin Wong, and Pheng-Ann Heng. Manga colorization. *ACM Transactions on Graphics (TOG)*, 25(3):1214–1220, 2006.

[8] Mark A Ruzon and Carlo Tomasi. Alpha estimation in natural images. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2000)*, volume 1, pages 18–25. IEEE, 2000.

[9] Daniel Sỳkora, Jan Buriánek, and Jiří Žára. Unsupervised colorization of black-and-white cartoons. In *Proceedings of the 3rd international symposium on Non-photorealistic animation and rendering (NPAR 2004)*, pages 121–127. ACM, 2004.

[10] Daniel Sýkora, John Dingliana, and Steven Collins. Lazy-Brush: Flexible Painting Tool for Hand-drawn Cartoons. *Computer Graphics Forum*, 28(2):599–608, 2009.

[11] Richard Zhang, Jun yan Zhu, Phillip Isola, Xinyang Geng, Angela S. Lin, Tianhe Yu, and Alexei A. Efros. Real-time User-guided Image Colorization with Learned Deep Priors. *ACM Trans. Graph. (TOG)*, 36(4):119:1–119:11, 2017.

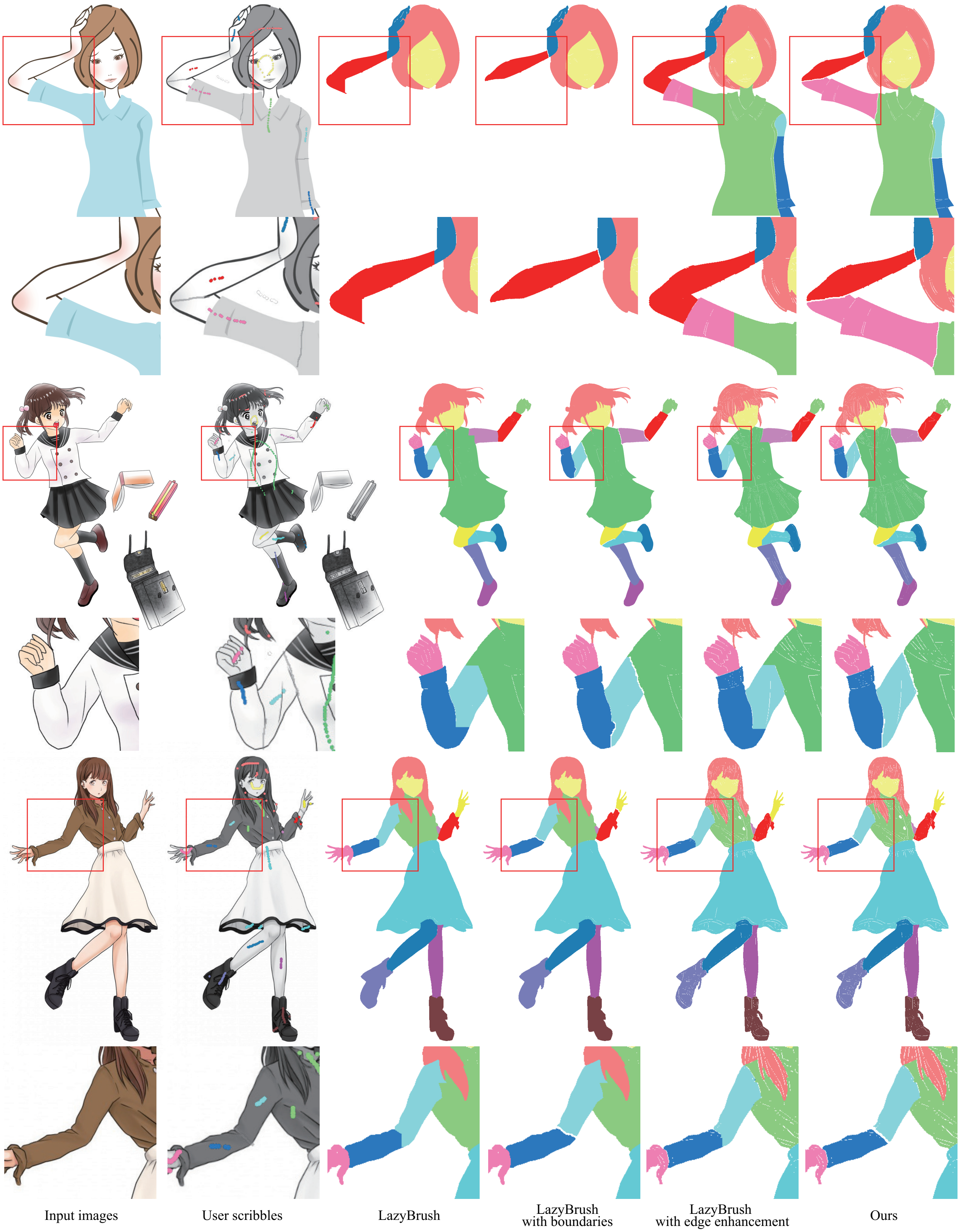

Figure 7. Comparisons of segmentation results with LazyBrush and our extensions. Note that the white thin lines in the segmentation results are removed automatically using a morphological operation.

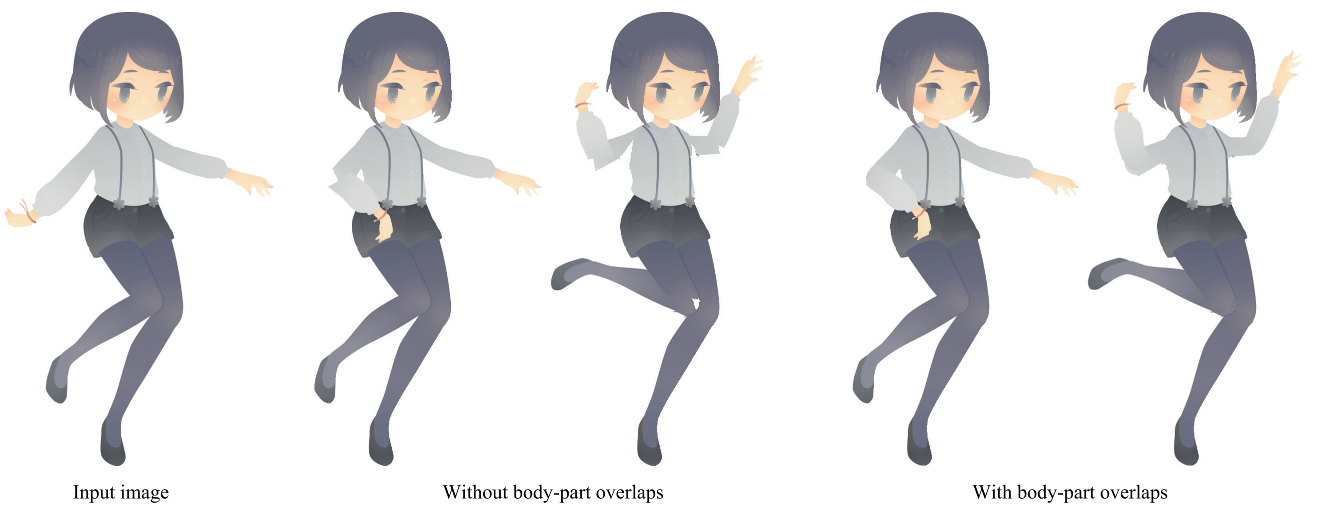

Figure 8. Screenshots of animation sequences with and without overlaps between adjacent body parts. Without overlaps, we can see gaps at the knees and elbows.

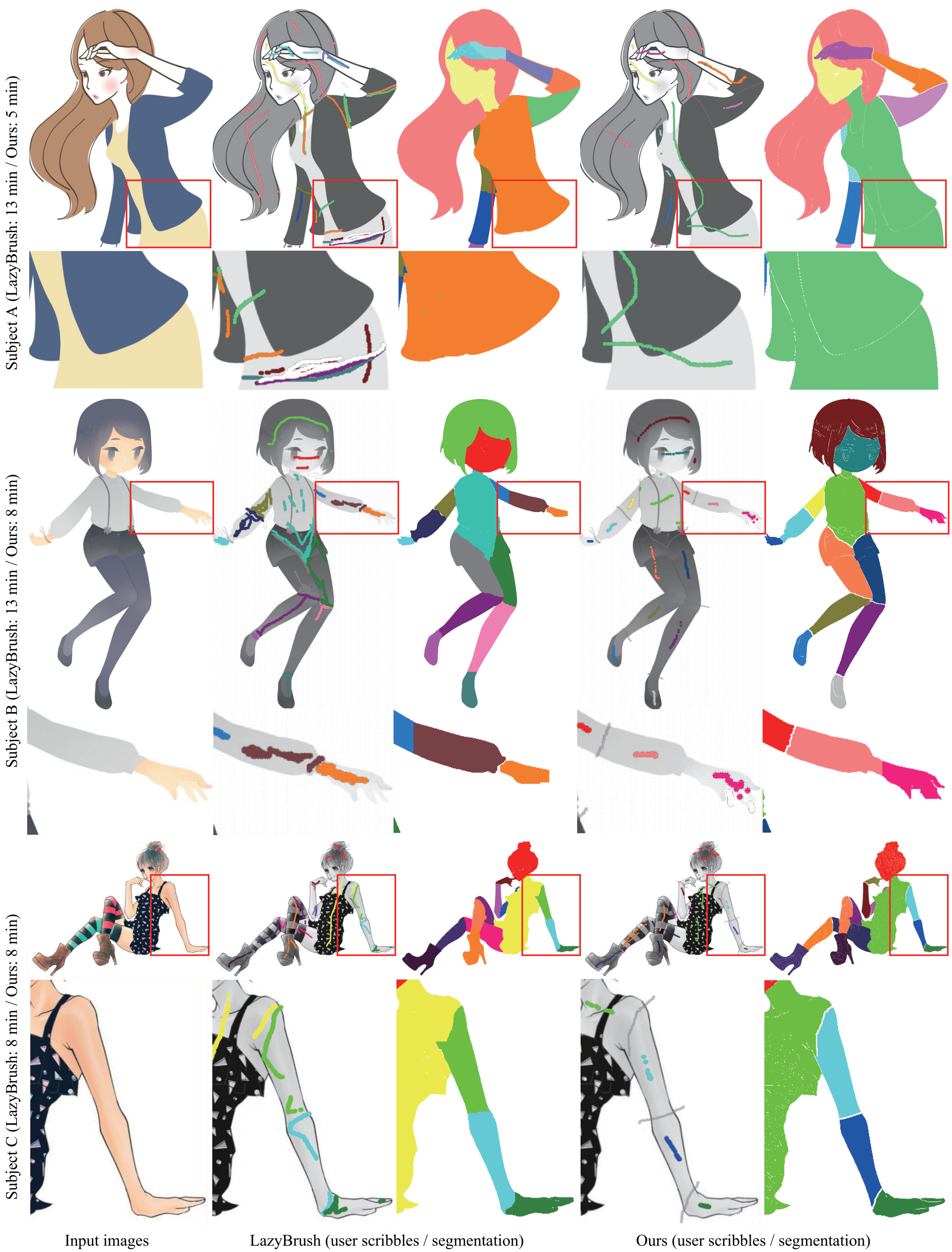

Figure 9. Summary of the user study. The time required for each image with each tool is shown in the left.