Fontender: Interactive Japanese Text Design with Dynamic Font Fusion Method for Comics

Junki Saito[1] and Satoshi Nakamura[1]

[1] Meiji University, 4-21-1 Nakano, Nakano-Ku, Tokyo, Japan
jonki.fms@gmail.com

Abstract. Comics consist of frames, drawn images, speech balloons, text, and so on. In this work, we focus on the difficulty of designing the text used for the narration and quotes of characters. In order to support creators in their text design, we propose a method to design text by a font fusion algorithm with arbitrary existing fonts. In this method, users can change the font type freely by indicating a point on the font map. We implement a prototype system and discuss its effectiveness.

Keywords: Font, Text Design, Font Fusion, Comic.

1 Introduction

Since the emergence of tablet PCs that can be used with a stylus, the creation and transmission of comics by individuals have become popular. In Japan, the number of services that can easily distribute comics created by individuals, such as MediBang [1] and LINE Manga [2], are increasing. The key elements of comic creation include drawing characters and backgrounds and designing text balloons and banners. Another element is the text design, which greatly influences the greatness of comics.

When a comic creator designs texts with the atmosphere of the scene or the characters' personality in mind, great care is taken to select a font from a list of available ones that fits the desired image. However, there are fewer types of Japanese fonts compared to European and American ones because Japanese fonts require over 7000 characters. This fact significantly narrows the choices of Japanese comic creators and makes it difficult for them to come up with text design that suits the situation.

There have been several studies on transforming existing fonts and acquiring new glyph shapes. Suveeranont et al. [3] proposed a system that creates a new font by fusing an arbitrary font with a character handwritten by the user. However, when this system merges the handwritten character and the font, the users have to adjust the corresponding stroke. Campbell et al. [4] proposed a method that detects corresponding strokes by placing fonts on two-dimensional flat surface so that fonts with similar shapes are placed close to each other. The problem with this method is that the combination of fusible fonts is limited.

We propose three methods in this work. The first is a dynamic font fusion method featuring an averaging technique. The second is a method of generating Japanese fonts

interactively by indicating a point in two-dimensional space, thus enabling users to design texts that match any scene in the comic. The third is a method to acquire new fonts by arbitrarily combining multiple fonts. We also implement a prototype system to determine how well the proposed methods can be used to design the texts used in comics.

2 Fontender

We propose a method called "Fontender", a new font generation method that uses arbitrary existing fonts and freely changes the shape of the fonts. This method sets existing fonts on a two-dimensional surface with the impression words input by the user as its axis, and it fuses multiple fonts according to the position on the space selected by the user with our font fusion algorithm.

2.1 Algorithm for Blending Fonts

We grasp information on the core and thickness of the font as a mathematical expression and blend the font by fusing the mathematical expressions. In this work, we assume that letters can be formed with a locus of a circle whose radius changes, and express a font by drawing a locus of the circle so that the point on the core line is the center coordinates.

The procedure for formulating a font is shown in Fig. 1. First, our method locates the locus of the circle expressing the font. Next, in order to express the font smoothly, cubic spline interpolation connects sets of the circles as much as possible and circles occupying the space are generated. Next, our method locates circle sets of the closed curves by folding back sets of the circles to which spline interpolation is applied. The reason for making it a closed curve is that if the starting point and the ending point of mathematical expression are separated by Fourier series expansion, the curve will be wavy near both ends trying to connect both ends. We denote the parametric representation of plane curve passing through this circle sets as $(\mathrm{x}, \mathrm{y}, \mathrm{z}) = \big(f(t), g(t), h(t)\big)$ $(-\pi \leq t \leq \pi)$. $(f(t), g(t))$ at t is the point on the core line of the font, and $h(t)$ is the largest radius of inscribed circle at t in the font. Although $f(t)$, $g(t)$, and $h(t)$ are not a periodic function, we define them as periodic function by defining them like the formula shown in (1).

$$f(t) = f(t + 2n\pi) \qquad n = 0, 1, 2 \ldots \tag{1}$$

Furthermore, $f(t)$, $g(t)$, and $h(t)$ can be expressed by a Fourier series by considering the corner of the character as a curved line approximating steeply bent. Therefore, $f(t)$ can be expressed by an equation as shown in (2).

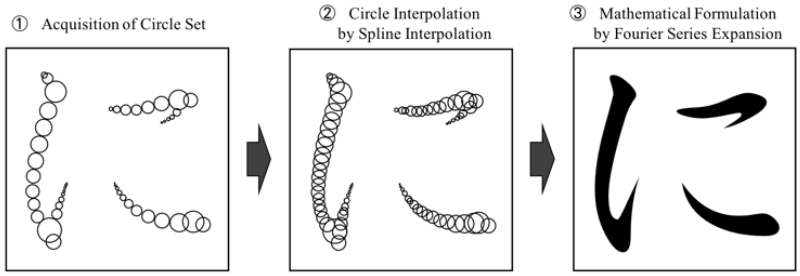
$$f(t) = \frac{a_0}{2} + \sum_{n=1}^{\infty}(a_n \cos nt + b_n \sin nt) \tag{2}$$

a_n and b_n are obtained by the formulas as shown in (3).

$$\begin{cases} a_n = \frac{1}{\pi}\int_{-\pi}^{\pi} f(t)\cos nt\, dt \\ b_n = \frac{1}{\pi}\int_{-\pi}^{\pi} f(t)\sin nt\, dt \end{cases} \tag{3}$$

Finally, using these formula, n kinds of fonts are blended at arbitrary ratios. When we denote the formula of each font's stroke as $(\mathrm{x},\mathrm{y},\mathrm{z}) = (f_i(\mathrm{t}), g_i(t), h_i(t))$ and the blend ratios of each font as $\alpha_1 \sim \alpha_n$, the formula of the stroke of the blended font can be expressed as in (4). Changing this blending ratio makes it possible to create arbitrary fonts.

$$\begin{cases} x = \sum_{i=1}^{n} \alpha_i f_i(t) \\ y = \sum_{i=1}^{n} \alpha_i g_i(t) \\ r = \sum_{i=1}^{n} \alpha_i h_i(t) \end{cases} \tag{4}$$



① $(x, y, r) = \{(x_1, y_1, r_1), (x_2, y_2, r_2), \ldots, (x_n, y_n, r_n)\}$

② $(x', y', r') = \{(x_1, y_1, r_1), (x_{1,1}, y_{1,1}, r_{1,1}), \ldots (x_{1,m}, y_{1,m}, r_{1,m}), (x_2, y_2, r_2), \ldots, (x_n, y_n, r_n)\}$

③ $X_1 = f_1(t), Y_1 = g_1(t), R_1 = h_1(t)$
$X_2 = f_2(t), Y_2 = g_2(t), R_2 = h_2(t)$
$X_3 = f_3(t), Y_3 = g_3(t), R_3 = h_3(t)$

Fig. 1. Procedure to display a font using formulas.

2.2 Impression Estimation for Font

In Fontender, in order to make the impression word of the font usable as input, it is necessary to obtain the impression value of the existing font and the factor structure of the impression people receive.

In order to obtain the impression value and the impression factor structure, we asked 17 experimental participants (university students aged 20 to 23 years old) to make an impression evaluation of the font. A total of 18 types of fonts were evaluated (see Fig. 2). We selected 35 adjective pairs used in an earlier impression study [5][6] of Japanese fonts and asked the participants to perform a seven-step evaluation by the semantic differential method.

Fig. 2. Font list to be evaluated.

In order to confirm what type of impression structure is formed from the obtained data, we performed a factor analysis by the main factor method and Promax rotation for all 18 types of fonts. After excluding an adjective pair whose commonality was less than 0.35 and an adjective pair belonging to a plurality of factors, 18 pairs of adjectives (hard-soft, square-round, dopey-clear, loose-tense, formal-casual, sharp-dull, harsh-mild, imitative-creative, unstable-stable, mature-childish, bad-good, uncomfortable-comfortable, dislike-like, ugly-beautiful, painful-pleasant, gloomy-happy, feeble-powerful, sober-flashy) and four factors (mild, attractive, optimistic, active) were extracted. In the prototype system described in section 3, it is possible to designate the four factors extracted by factor analysis along with the 18 adjective pairs.

3 Prototype System

We implemented a prototype system[1] for character design support in comics (see Fig. 3). This system consists of two parts: one for generating a font and one for previewing the generated font. This system was implemented by Processing.

The proposed system arranges existing fonts on a two-dimensional surface based on two impression words input by the user after viewing the image. Since we used factor names and adjective pairs (obtained by factor analysis in section 2) as input words, we use a value obtained by normalizing the value obtained by averaging the evaluation values of experimental collaborators to –1 to +1 as an impression values of the font. In addition, we normalized the impression values of each factor was –1 to +1.

The user can set impression words using the factor name or the adjective selection interface at the bottom of the font generating part. In addition, we installed a toggle button so that the user can switch between the mode in which factor names can be selected and the mode in which adjectives can be selected. The impression word selected first by the user is the horizontal axis of the two-dimensional plane, and the second impression word is the vertical axis of the two-dimensional plane so that further selection is not accepted. When the user selects impression words, 18 kinds of fonts are arranged according to the impression value with the two impression words selected by the user as the axis.

[1] https://www.youtube.com/watch?v= 6tLaz4dCfTE

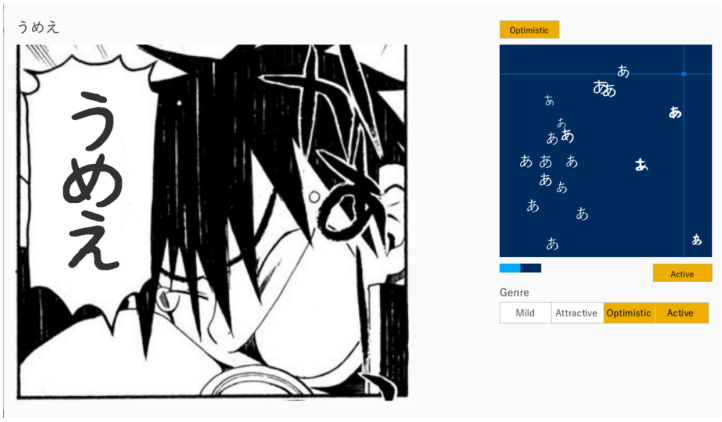


Fig. 3. A screen shot of the prototype system. In this screen shot, the user designs the text "うめえ (delicious in Japanese)" in the speech balloon. The user generates a new font by pointing in the right area of this window.

When 18 kinds of fonts are arranged, the user can generate blended fonts. Specifically, by changing the position of the mouse pointer on the two-dimensional surface, this system blends the four adjacent fonts by changing the fusion ratio according to the distance to each mouse pointer. The fusion ratio is the ratio of the inverse of the distance to the mouse pointer, and the closest font has the highest fusion ratio. Blending results are displayed in real time.

Fig. 4 shows how to design the expression "しあわせだ (happy in Japanese)" using the prototype system. Fig. 4 left is a text design in which the degree of "like" and "warm" is high. This text design is rounded and has a shape that expresses femininity. In another example, Fig. 4 right is a text design in which the degree of "like" and

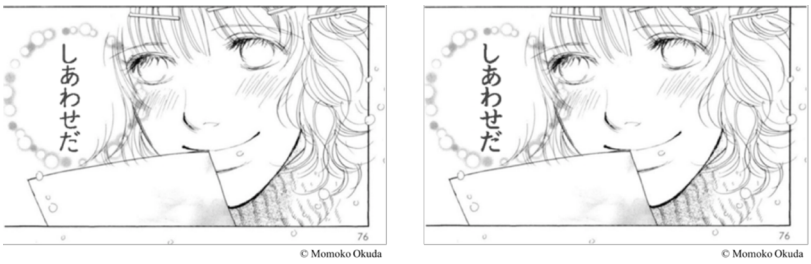


Fig. 4. Left: Font creation example with high "like" and "warm" degrees. Right: Font creation example with low "like" and "warm" degrees low. This image uses the excerpts from "Aosugiru Haru" written by Momoko Okuda in Manga 109 [7].

"warm" is low. This text design is thick and has a shape that makes you feel powerful. In the implementation at this point, since smoothing cannot be performed to change the thickness, some noise remains partly in the character.

To verify the usefulness of our system, we conducted experiments comparing the proposed method with two baseline systems. One is a pull down interface and the other is a map interface without font fusion. We asked 15 subjects (aged from 21 to 23) to design five times with three systems, and asked them to evaluate the degree of satisfaction after each design task by the five level Likert scale. As a result of two variance analysis of them, the main effect of the design task was not significant in either question, but the main effect of the method is the map arrangement form ($F [1, 4] = 4.95$, $p < 0.05$) and list format ($F [1, 4] = 6.54$, $p < 0.05$).

4 Conclusion and Future Work

To solve the inherent problems in text design, we proposed a method of blending existing fonts to match the impression of users' intended impression and generating a new font. We also implemented a prototype system for character design support using the proposed method and showed through examples under various conditions that this method is effective.

In the future work, we will improve the system so that it can support the design of letters used in comics in general. Specifically, we plan to implement a function that allows users to easily compare previous and next characters, creating a log of the font blend results, and an adjust function by outlining the blending results of multiple fonts or the blending results of fonts and handwriting.

Our method is not only for comics but also for every content design. For example, we can apply our system for poster design and digital publishing such as a name card, a handout, a presentation slide, a web page, a Christmas card, and so on. In the future, we will realize such system.

Acknowledgments. This work was supported in part by JST ACCEL Grant Number JPMJAC1602, Japan.

References

1. MediBang, https://medibang.com/creators, last accessed 2018/09/15.
2. LINE Manga, https://manga.line.me/indies/debut, last accessed 2018/09/15.
3. Suveeranont, R., Igarashi, T.: Example-Based Automatic Font Generation. Proceedings of Smart Graphics 2010, Lecture Notes in Computer Science, vol. 6133, pp. 127–138 (2010).
4. Campbell, N., Kautz, J.: Learning a Manifold of Fonts. In ACM Transactions on Graphics (SIGGRAPH), vol. 33, no. 4 (2014).
5. Inoue, M., Yoroizawa, I.: Impression Received from Letter Form and Examination of Quality Evaluation Factor. The IEICE Transactions, vol. J67-B, no. 3, pp. 328–335 (1984).
6. Inoue, M., Kobayashi T.: The Research Domain and Scale Construction of Adjective-Pairs in a Semantic Differential Method in Japan. The Japanese Journal of Educational Psychology, vol.33, no. 3, pp. 253–260 (1985).
7. Matsui, Y., Ito, K., Aramaki, Y., Fujimoto, A., Ogawa, T., Yamasaki, T., Aizawa, K.: Sketch-based Manga Retrieval using Manga109 Dataset. Multimedia Tools and Applications, Springer (2017).