Exploring the World of Fonts for Discovering the Most Standard Fonts and the Missing Fonts

Seiichi Uchida, Yuji Egashira, Kota Sato
Kyushu University
Fukuoka, Japan 819–0395
Email: uchida@ait.kyushu-u.ac.jp

***Abstract*—This paper has two contributions toward understanding the principles in font design. The first contribution of this paper is to discover the most standard font shape of each letter class by analyzing thousands of different fonts. For this analysis, two different methods are used. The first method is congealing for aligning multiple images based on a nonlinear geometric transformation model. The average of the aligned image is considered as a standard font shape. The second method is network analysis for representing font variations as a large-scale relative neighborhood graph (RNG) and then finding its center. The font corresponding to the center is considered as the standard font shape. Both of the standard font shapes given by the two methods are plain without decoration, serif, or slant, and thus give an objective reason why we consider the plain font as the typical font shape. The second contribution is to utilize the RNG and the pairwise congealing technique for discovering unexplored font designs and then generating totally new fonts automatically.**

I. Introduction

This paper tries to analyze *the world of fonts* (i.e., the font shape variations) by using thousands ($> 5,000$) of different fonts, toward understanding the principles in font design. This is an unexplored research direction of connecting two distant research fields, i.e., engineering and designing (or even fine art). In addition, it will contribute to even more practical scenarios, i.e., scene text detection and recognition, because scene texts are printed in various fonts.

The first contribution of this paper is to reveal the most standard font shape through several analyses of the thousands of fonts. This will give a top-down answer to the question, *what is "A"?*, which is the most fundamental question in not only OCR research and but also more general discussion on artificial intelligence [1]. While designing a new font, a designer first imagines her/his standard "A" and then make some original modification on it. The standard shape of "A", therefore, will provide us a solid ground for understanding human design principle.

The second contribution is to discover unexplored regions in the world of fonts and then generate totally new fonts, or missing fonts, by filling the regions. Even though we prepare thousands of fonts, they do not cover all possible designs. In other words, there will be many unexplored regions. We try to find the regions automatically by analyzing the current font set and then generate new fonts automatically. It is interesting to note that this gives a bottom-up answer to the above question (what is "A"?).

Both of the above contributions rely on two techniques. The first technique is congealing [2]–[4], which is a method of aligning multiple images simultaneously under a certain geometric transformation model. In order to deal with font images with huge variations, we introduce a nonlinear (i.e., nonrigid) transformation model based on a spline function that represents stroke outline shape.

The second technique is network analysis. In several analyses of the font shape variations, we will represent the thousands of fonts as a relative neighborhood graph (RNG). In RNG, each font image corresponds to one node and two font images neighboring in their feature space are two nodes connected by an edge. It is important to note that even two dissimilar font images might be treated as neighboring font images if there is no intermediate font between them. Later discussion proves that this characteristics of RNG is useful to find the unexplored font shape variations. In addition, RNG is a connected network and thus useful to find its "center," which might be a candidate of the most standard font shape.

Engineering font shapes is rather a new research direction and thus the number of related work is not so large. To the authors' best knowledge, there is no past trial on finding the most standard font image, except for [5]. Our paper is far advanced from the minimum spanning tree-based method of [5] by (i) not only using RNG representation to derive a more reliable center font as the standard font shape (without any unnatural constraint for representing the font distributions by a tree) (ii) but also employing the congealing method as a totally new way to discover the standard font shape. For font generation, we can find several past trials [6]–[9]. They generate a new font by interpolating a given pair of rather similar fonts. Thus, their purpose is different from ours, which tries to discover a missing font from a pair of rather dissimilar (but neighboring) fonts found automatically by RNG. The most active research related to font variations will be scene text detection and recognition [10], [11]; the past trials, however, neither tried to understand the nature of font shape nor generated new font shapes.

II. Thousands of Font Images

A. Collecting 7,000 Fonts

About 11,000 different fonts were first collected from *Ultimate Font* [1]. Then, about 4,000 fonts were excluded by manual inspection because they were almost unreadable by heavy decoration or just a standard font surrounded by some irrelevant illustration. Finally, about 7,000 fonts were selected as the target font and used in the later experiments. Figs. 1 and 2 show several examples of the selected and excluded fonts, respectively. The selected fonts of Fig. 1 show severe variations in their shape.

[1] http://www.ultimatefontdownload.com/

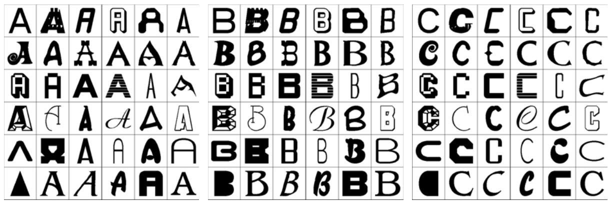

Fig. 1. Examples of the selected fonts of “A”, “B” and “C”.

Fig. 2. Examples of the excluded fonts of “A”.

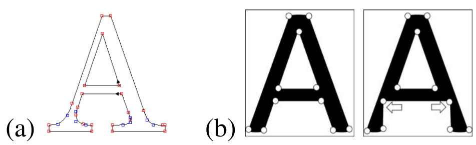

Fig. 3. (a) Control points of a certain “A”. On-curve and off-curve points are depicted in red and blue, respectively. (b) An illustrative example of font deformation by moving on-curve control points.

B. Outline fonts

All of those fonts are outline fonts (so-called TrueType fonts), which are not a bitmap but vector graphics. Each of font is represented as a set of contours and each contour is a spline curve specified by a certain number of control points. By moving the control points, it is possible to deform the contour. In other words, the font shape is totally represented by the location of the control points.

There are two types of control points in outline fonts — on-curve point and off-curve point. Fig. 3 (a) shows the control points of a certain font shape. The on-curve point is located at a corner of a stroke contour. As shown in Fig. 3 (b), if the location of a on-curve point is changed, the location of the corresponding corner is also changed. The off-curve point is located around a curve for regulating the curvature of the curve. In general, a complex font with more corners and curves have more control points.

III. DISCOVERING THE MOST STANDARD FONT SHAPE BY CONGEALING

A. Multiple image alignment by congealing

Congealing [2]–[4] is a method of aligning multiple images simultaneously under a certain geometric transformation (i.e., deformation) model. Assume a case of aligning N binary images $I_1, \ldots, I_N$ by optimizing their deformation parameters $\theta_1, \ldots, \theta_N$. Let $\tilde{I}_n$ denote the image I_n deformed by θ_n. In [2], the objective function F is defined as the sum of the pixel value entropy $\mathcal{H}_{x,y}$ over all pixels, i.e., $F = \sum_{x,y} \mathcal{H}_{x,y}$, where $\mathcal{H}_{x,y}$ is the pixelwise entropy of $\tilde{I}_1(x,y), \ldots, \tilde{I}_N(x,y)$ and achieves its minimum value 0 when $\tilde{I}_1(x,y) = \cdots = \tilde{I}_N(x,y) = 1$ or 0. Consequently, the minimization of F will result in the best alignment of the N images.

Congealing employs a non-global but efficient optimization scheme. This is because global optimization of all the parameters $\Theta = \{\theta_1, \ldots, \theta_N\}$ is computationally intractable for a larger value of N and a higher dimensionality of θ_n. (In our case, $N > 5,000$ and the dimensionality of θ_n becomes ~ 80 on average and thus the dimensionality of Θ is $> 400,000$.)

Specifically, the optimization scheme is based on simple perturbation. That is, each parameter value is first perturbed. Then, the perturbed value is accepted if it decreases the objective function F. This perturbation is repeated for all parameters of Θ until convergence or for a fixed amount of iterations. Fortunately, this scheme works well in spite of its simplicity because only a single image is deformed to fit to all the other $N-1$ images as possible at once. In other words, a bad perturbation that makes the image different from the others will not be accepted by a serious disagreement with the other $N-1$ images.

B. Congealing for discovering a standard font shape

The congealing method was applied for aligning thousands of different font images. The deformation parameter θ_n of the font image I_n was the locations of P_n control points. (That is, the dimensionality of θ_n equals to $2P_n$.) Due to a limit on computational resource, 1,600 fonts with $P_n > 100$ were excluded among 7,000 fonts. The following experiment on congealing, therefore, was done with 5,400 font shapes for each class. Note that the average of P_n of the 5,400 fonts was 41.6.

During the alignment, a one-pixel perturbation in horizontal or vertical direction was applied to each of the $2P_n$ elements of all I_n. This perturbation process is repeated 100 times — this was practically enough to achieve a quasi-convergence where most fonts do not move anymore.

Fig. 4 shows 36 congealing results among 5,400 fonts of “A”, “B”, and “C”. Comparing to huge variations in their original shape of Fig. 1, the alignment results show a strong similarity to each other. A closer look will reveal that decorations are often reduced to very thin lines, not to disturb the similarity in the alignment result. In addition, one font of “A” (in the rightmost column) just became thinner without large change in its shape; this is because its original stroke position was very different from that of other fonts and no local perturbation for “relocating” the stroke was successful. Consequently, the perturbation for “fading” the stroke was chosen for making F smaller [2]. Fig. 5 shows the congealing results of all 5,400 fonts of “A”. Regardless of huge variations in original font shapes, the converged results show a significant similarity to each other. This fact proves the stability of the congealing method with the perturbation of control points.

Figs. 6 (a) and (b) show the average images before and after the multiple shape alignment by congealing, respectively. Since the original font shapes have huge variations as shown in Fig. 1, the simple average images of Fig. 6 (a) are very blurred. In contrast, Fig. 6 (b), i.e., the average images after the alignment, shows clear edges around their strokes. This fact proves that very different font shapes becomes similar to each other by congealing (as also proved by Fig. 5) at all classes.

[2]It might be possible to avoid this “faded-out” case by increasing the degree of perturbation. This remedy, however, will make the congealing process not only unstable and computationally prohibitive.

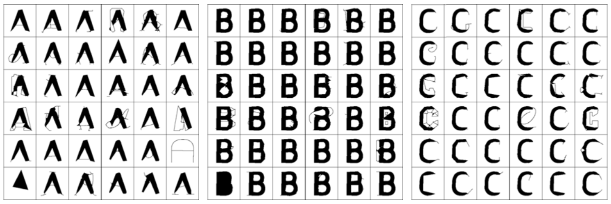

Fig. 4. Congealing results of the fonts shown in Fig. 1.

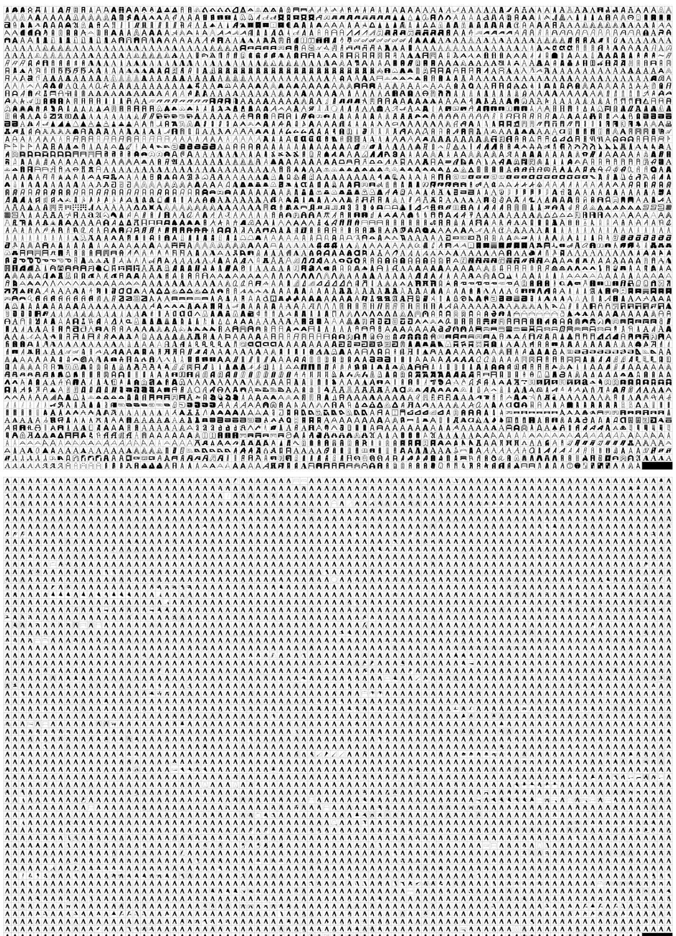

Fig. 5. 5,400 different font images of "A" (upper) and their alignment result by congealing (lower). A closer view will be available by digital zooming on the PDF file.

(a)

(b)

Fig. 6. Average images (a) before and (b) after the multiple shape alignment by congealing.

More importantly, the average images of Fig. 6 (b) have *plain* shapes without any decoration, serif, slant, thick stroke, or thin stroke, and thus will coincide with our subjective standard font shapes, except for that of "A". Conversely, this result gives an objective reason why we often think that a plain upright sanserif font as a standard shape of each letter class — such a font is the average of all shape variations.

The reason that the horizontal stroke of "A" disappears in its average image will be explained by its less ambiguity to the other letters; even if there is no horizontal stroke in "A", it has no risk to be misrecognized as another Latin letter or digit. Thus, it is possible to say that the essence of "A" is its Λ-shaped part. In fact, sometimes there are "A"s without the horizontal stroke. The leftmost column of Fig. 1 shows two "A"s without horizontal stroke. It is interesting to note that the horizontal strokes of "E", "F" and "H" are not canceled out – this is because disappearance of them will cause misrecognitions, such as "E"→"C/F/t", "F"→ "T/t", and "H"→"I+I", and thus font designers preserve those horizontal strokes in most cases.

IV. Discovering the most standard font shape by network analysis

A. Font-network by relative neighborhood graph (RNG)

RNG (Relative Neighborhood Graph) is an undirected graph connecting neighboring nodes by an edge. Roughly speaking, neighboring nodes of RNG has no neighboring node between them. More formally, the neighboring nodes are defined as two nodes having no "intermediate" node v, which is a node in the intersection of two hyper-spheres whose radii equal to the distance between u and w and centers are at u and w respectively. As noted before, even two distant nodes u and w can be connected by a (long) edge[3]. Also note that RNG is a super-graph of the (directed) nearest neighbor graph and also the minimum spanning tree.

We will use RNG for representing the world of fonts. Specifically, each font image corresponds to one node of RNG. Nodes are connected by an edge if their corresponding fonts are neighboring to each other. The largest merit of using RNG is that there is neither loss nor error on representing the neighboring relationship among fonts. This merit is not available by other typical representation methods for high-dimensional data variations, such as low-dimensional representation by principal component analysis or multidimensional scaling or manifold learning. Another merit is that it is possible to know a "center" of RNG for understanding a standard font shape by using some network centrality, as discussed later.

For constructing the RNG for a font set, we need to specify a distance metric for evaluating dissimilarity between a pair of font images. Since font images are binary images, the simple Hamming distance might be the first choice among various metrics [12]. It, however, is very sensitive to the difference in stroke location. In addition, dissimilarity between a pair of fonts with very thin strokes is underestimated and thus quite different from our subjective dissimilarity. Chamfer distance [13], [14] is an alternative but also has a problem that dissimilarity with a heavily decorated font with dense edges is always underestimated [4].

We, therefore, propose a *pseudo-Hamming distance* as the distance metric. The pseudo-Hamming distance between

[3]As an intuitive example, let us imagine the RNG which connects million cities on the earth as nodes. Then, there will be a long edge between *Tokyo* and *Los Angeles* because there is no million city in north pacific ocean.

[4]Accordingly, the heavily decorated fonts are similar to any fonts and thus becomes a center of RNG.

two binary images I_1 and I_2 (white background = 0, black foreground = 1) is defined as follows:

$$D(I_1, I_2) = \sum_{x,y} d(I_1(x,y), I_2(x,y)),$$

$$d(I_1(x,y), I_2(x,y)) = \begin{cases} I_1^d(x,y), & \text{if } I_1(x,y)=0,\ I_2(x,y)=1, \\ I_2^d(x,y), & \text{if } I_1(x,y)=1,\ I_2(x,y)=0, \\ 0, & \text{otherwise,} \end{cases}$$

where I_n^d is the distance transform image of I_n and thus its pixel value is the distance from the nearest foreground pixel. The property of the pseudo-Hamming distance inherits the good properties of Hamming distance and chamfer distance. In fact, the pseudo-Hamming distance becomes zero at the pixel with the same binary pixel value like Hamming distance and has the robustness to the difference in stroke location like chamfer distance (because of the use of distance transform).

B. Center node as standard font shape

As noted in IV-A, we can assume that the "center" node in the font-network shows the most standard font shape. There are several criteria for defining the center node for a large-scale network like our font-network. For our purpose, *closeness centrality* and *betweenness centrality* are useful [15]. The center node by the closeness centrality is the node which has the minimum total distance to all other nodes. On the other hand, the center node by the betweenness centrality is the node which lies on the shortest path connecting two arbitrary nodes most frequently.

Figs. 7(a) and (b) show the fonts that correspond to the node with maximum closeness centrality and betweenness centrality, respectively, in the RNG font-network with 7,000 fonts. For both of the centrality metrics, the derived center nodes are plain fonts without decoration, serif, or slant. The standard font shape *selected* as the center of the font-network is an existing font and thus different from the standard font shape *generated* by congealing (Fig. 6 (b)). We, however, can see that those standard shapes are still similar to each other (except for the disappearance of the horizontal stroke of "A" in Fig. 6 (b)) and thus get the objective reason why we think such a plain font as a standard font shape.

V. "Missing font" generation using font-network and pairwise congealing

A. Where are "missing fonts"?

Neighboring font images connected by the RNG-based font-networks are not always very similar to each other as noted in IV-A. That is, two dissimilar font images are connected if there is no intermediate font between them. Fig. 8 shows the histogram of pseudo-Hamming distance between neighboring fonts on the RNG and proofs the existence of neighboring but dissimilar fonts.

It is possible to consider that the region between two neighboring but dissimilar fonts is an unexplored region of font design. In other words, if we can generate any intermediate font from those fonts, it can be a "missing font" and may provide an idea for designing new font shapes. An important note is that we can automatically discover the region for the

(a)

(b)

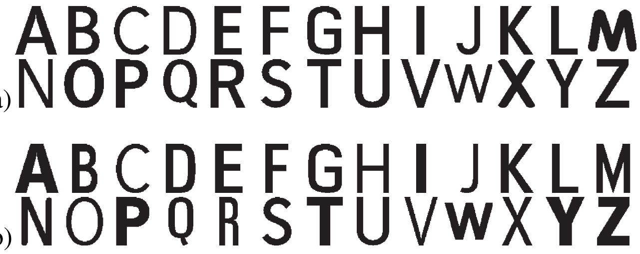

Fig. 7. The font with (a) maximum closeness centrality and (b) maximum betweenness centrality among 7,000 fonts at each class.

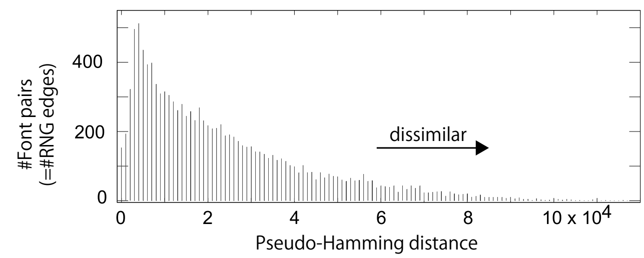

Fig. 8. Histogram of distance between neighboring fonts.

missing fonts, just by watching the edge distance of the RNG-based font-network.

B. Font generation by pairwise congealing

Given two neighboring but dissimilar fonts, we can generate their intermediate font by pairwise congealing. Unlike the original congealing method for aligning multiple font images simultaneously, the pairwise congealing method aligns a pair of font images I_1 and I_2 by perturbing the control points of the images. The pixelwise entropy $\mathcal{H}_{x,y}$ of Section III is no longer appropriate because it can take only two values for aligning two binary font images, i.e., $\mathcal{H}_{x,y} = 0$ (for $(I_1(x,y), I_2(x,y)) = (0,0)$ or $(1,1)$) and $\mathcal{H}_{x,y} = 1$ (for $(1,0)$ or $(0,1)$). We, therefore, use the pseudo-Hamming distance between $\tilde{I}_1$ and $\tilde{I}_2$ as the objective function F of the pairwise congealing. Note that the pairwise congealing method provides a non-globally optimal solution of the nonlinear image alignment problem and therefore we can replace it by some more reliable (but time-consuming) nonlinear image alignment method [16], if necessary.

C. Generated fonts

Fig. 9 shows examples of fonts generated by pairwise congealing. Since pairwise congealing deforms both of I_1 and I_2 for making them similar to each other, we have two fonts from I_1 and I_2. Considering that the majority of the distance between neighboring fonts is less than 1.0×10^4 as shown by Fig.8, more distant pairs were intentionally selected as I_1 and I_2 in Fig. 9. This is because our aim is to generate new fonts in the unexplored regions, as noted in V-A.

The pairs of I_1 and I_2 selected in Fig. 9 prove that there are neighboring but dissimilar fonts in RNG. Especially, when the pseudo-Hamming distance $D(I_1, I_2)$ is more than 9.0×10^4, the dissimilarity becomes more obvious. Again, this means there is no existing font between these dissimilar fonts and thus their intermediate fonts should provide a hint of new fonts.

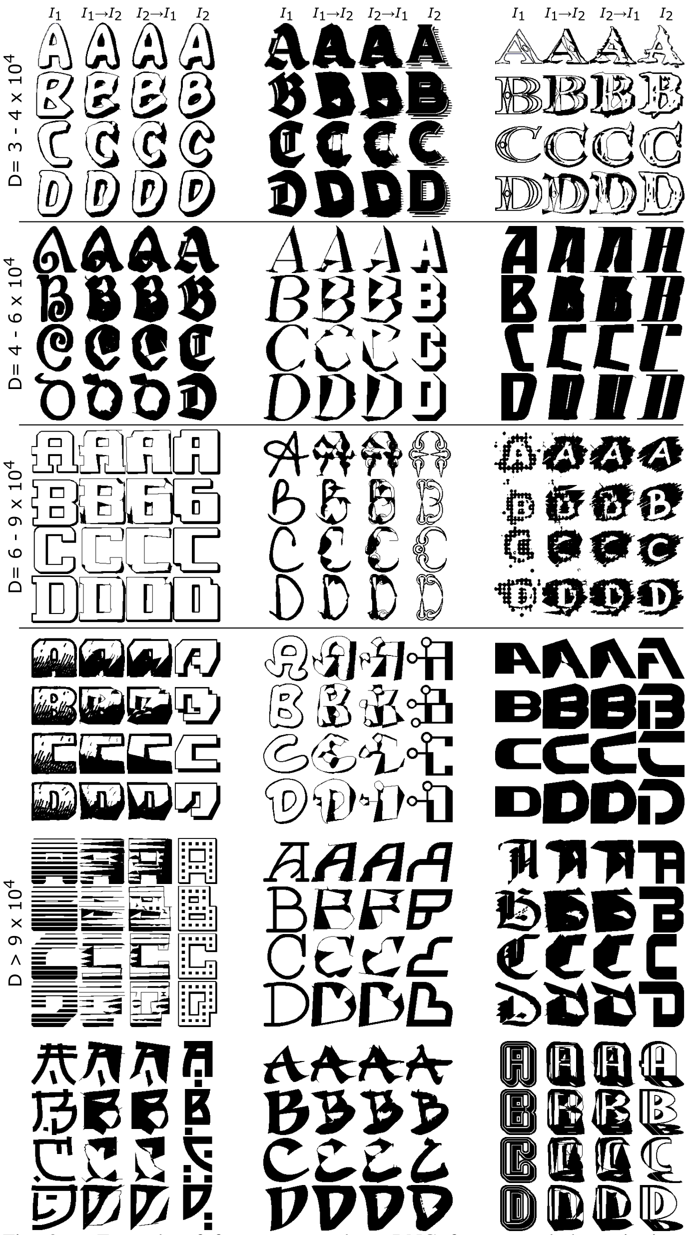

Fig. 9. Example of fonts generated on RNG font-network by pairwise congealing.

The generated fonts in Fig. 9 are mostly natural and thus can be considered as new font shapes, as expected. Consequently, we can prove qualitatively that RNG and the pairwise congealing discovers "missing fonts". This fact was also proved quantitatively; that is, the distances among I_1, I_2, and the generated fonts ($I_1 \to I_2$ and $I_2 \to I_1$) were compared and then found that the generated fonts correctly fell into the intermediate region between I_1 and I_2. In other words, if we add one of generated fonts to the original font sets and rebuild a new RNG, we have two bridging edges connecting like I_1–the generated font–I_2.

VI. Conclusion

In this paper, we first used two techniques and thousands of different fonts for revealing the standard font shape, as an answer to a fundamental question, that is, *what is "A"?*. The first technique was the congealing method, which aligns multiple font images simultaneously with a non-rigid deformation model. By taking the average of the aligned font images, we got a standard font image. The second technique was the network analysis on relative neighborhood graph (RNG) whose node corresponds to a font image and two nodes are connected by an edge if the corresponding fonts are neighboring to each other. By finding the center of the RNG, we got another standard font image. Our results revealed that *both* standard fonts are plain without decoration, serif, or slant. These results prove an objective reason why we consider the plain font shape as the standard shape.

The RNG was further utilized for finding the unexplored regions in the current font variations and then the pair-wise congealing method was used for generating new fonts that lie on the unexplored region. Most of the generated fonts look natural (even though they are generated from a pair of dissimilar fonts) and thus are expected to provide some hints for human font designers.

Font engineering is a new research field to connect engineering and designing. In the past, font design has been done just by human; however, analyzing thousands of existing fonts will help to understand the principles of font design even for non-experts, and also will give many hints to deal with various fonts in natural scene.

Acknowledgment

This research was partially supported by JST-CREST and MEXT-Japan (Grant No. 26240024).

References

[1] D. R. Hofstadter, *Metamagical Themas: Questing for the Essence of Mind and Pattern*, Basic Books, 1985.

[2] E. Learned-Miller, "Data Driven Image Models through Continuous Joint Alignment," *TPAMI*, 28(2), 2006.

[3] M. Cox, et al., "Least Squares Congealing for Unsupervised Alignment of Images", *CVPR*, 2008.

[4] Y. G. Peng, et al., "RASL: Robust Alignment by Sparse and Low-Rank Decomposition for Linearly Correlated Images", *TPAMI*, 34(11), 2012.

[5] C. Nakamoto, et al., "Font Distribution Analysis by Network," *CBDAR*, 2013.

[6] R. Suveeranont and T. Igarashi, "Feature-Preserving Morphable Model for Automatic Font Generation," *SIGGRAPH ASIA*, 2009.

[7] R. Suveeranont and T. Igarashi, "Example-Based Automatic Font Generation", *Int. Symp. on Smart Graphics*, 2010.

[8] V. M. K. Lau, "Learning by Example for Parametric Font Design", *SIGGRAPH ASIA*, 2009.

[9] N. Campbell and J. Kautz, "Learning a Manifold of Fonts," *SIGGRAPH*, 2014.

[10] S. Uchida, "Text Localization and Recognition in Images and Video," *Handbook of Document Image Processing and Recognition*, Springer-Verlag, 2014.

[11] Q. Ye and D. Doermann, "Text Detection and Recognition in Imagery: A Survey," *TPAMI*, (To be published in 2015).

[12] J. D. Tubbs, "A Note on Binary Template Matching," *PR*, 22(4), 1989.

[13] G. Borgefors, "Hierarchical Chamfer Matching: A Parametric Edge Matching Algorithm," *TPAMI*, 10(6), 1988.

[14] D. Shi, S. Gumm, and R. Damper, "Handwritten Chinese Radical Recognition Using Nonlinear Active Shape Models," *TPAMI*, 25(2), 2003.

[15] T. Opsahl, F. Agneessens, and J. Skvoretz, "Node Centrality in Weighted Networks: Generalizing Degree and Shortest Paths", *Social Networks*, 32(3), 2010.

[16] S. Uchida and H. Sakoe, "A Survey of Elastic Matching Techniques for Handwritten Character Recognition", *IEICE Trans.*, E88-D(8), 2005.