Font Distribution Observation by Network-Based Analysis

Chihiro Nakamoto[1], Rong Huang[1], Sota Koizumi[1], Ryosuke Ishida[1], Yaokai Feng[2], and Seiichi Uchida[2]

[1] Graduate School of Information Science and Electrical Engineering, Kyushu University, Fukuoka 819-0395, Japan
{nakamoto,rong,koizumi,ishida}@human.ait.kyushu-u.ac.jp
[2] Faculty of Information Science and Electrical Engineering, Kyushu University, Fukuoka 819-0395, Japan
{fengyk,uchida}@ait.kyushu-u.ac.jp
http://human.ait.kyushu-u.ac.jp/index.html

Abstract. The off-the-shelf Optical Character Recognition (OCR) engines return mediocre performance on the decorative characters which usually appear in natural scenes such as signboards. A reasonable way towards the so-called camera-based OCR is to collect a large-scale font set and analyze the distribution of font samples for realizing some character recognition engine which is tolerant to font shape variations. This paper is concerned with the issue of font distribution analysis by network. Minimum Spanning Tree (MST) is employed to construct font network with respect to Chamfer distance. After clustering, some centrality criterion, namely closeness centrality, eccentricity centrality or betweenness centrality, is introduced for extracting typical font samples. The network structure allows us to observe the font shape transition between any two samples, which is useful to create new fonts and recognize unseen decorative characters. Moreover, unlike the Principal Component Analysis (PCA), the font network fulfills distribution visualization through measuring the dissimilarity between samples rather than the lossy processing of dimensionality reduction. Compared with K-means algorithm, network-based clustering has the ability to preserve small size font clusters which generally consist of samples taking special appearances. Experiments demonstrate that the proposed network-based analysis is an effective way to grasp font distribution, and thus provides helpful information for decorative character recognition.

Keywords: font distribution, minimum spanning tree, centrality criterion, network-based clustering

1 Introduction

Optical Character Recognition (OCR) techniques have achieved great success in the field of scanner-based document image analysis. However, as demonstrated by Epshtein *et al.* in [1], OCR engines were thwarted in the scene character

Fig. 1: Scene characters captured by camera.

recognition. This is because the scene character appearing on signboards, notice signage, wrapper, etc. (see Fig.1), is usually designed by special decoration with the intent to attract people's attention. On the other hand, since camera is far handier than scanner, camera-based OCR, which focuses on recognizing characters captured by camera, will not only extend new applications of OCR but also brings convenience to us in our daily life. In view of this prospect, it has become an imperative demand to develop the camera-based OCR. However, realization of a high-performance camera-based OCR is still a hard task, although numerous impressive methods have been proposed [2–4]. As just mentioned, one of the challenges for scene character recognition lies in the unconstrained appearances with various decorations. Therefore, one possible strategy towards the camera-based OCR is to extract a topological structure that is nearly invariant to decorations.

Along this line of thought, several font-related methods have been elaborately designed as efforts to narrow the gap between OCR and the decorative character. Zhu *et al.* [5] presented a font recognizer by using multichannel Gabor filters and weighted Euclidean distance classifier. Omachi *et al.* [6] detected ridges and ravines from multi-scale images to extract an essential structure of the decorated character. As a subsequent work, Omachi *et al.* [7] matched the graphs of the above-extracted structure and standard patterns to recognize a character image. Unlike relying on the global structure, Wang *et al.* [8] proposed a series of part-based methods which were characterized by the robustness against various appearances of a character.

Although the above methods fulfilled the decorative character recognition to some extents, the performance was far from ideal. A straightforward solution for performance improvement is to collect or enumerate all types of fonts. Unfortunately, this idea is impossible since he/she always can design a new font which takes remarkably different shapes compared with the members of the collected set. As a remedy, we can investigate and analyze a large-scale font set under a certain type of data structure like tree, graph, cluster or network, to grasp the font distribution so that we can approach the ideal effect of brute-force enumeration.

In this paper, we propose a network-based method built on a large-scale font set, which allows us to analyze the font distribution in the feature space. Specifically, the so-called font network is constructed by Minimum Spanning Tree (MST) algorithm taking each font sample as a node. The dissimilarity between two font samples is measured by the Chamfer distance which has been widely adopted in the field like template matching [9] and handwritten Chinese recognition [10]. Merits of our proposal lie in that (1) unlike the well-known Principal Component Analysis (PCA) which lossily projects feature points onto low-dimensional space for distribution visualization, the proposed font network built by linking neighbors can represent the actual font distribution without information loss or distortion; (2) compared with the conventional K-means algorithm, network-based clustering can generate more reliable font cluster and typical samples by introducing some clustering criterion (as introduced in 3.4). This is because K-means is equivalent to Maximum a Posterior (MAP) estimation of a Gaussian mixture distribution while the font distribution is neither a Gaussian nor a Gaussian mixture; (3) along a path of the font network, we can understand the font shape transition between any two samples, which is useful to create new fonts or recognize various scene characters. For example, for a given decorative character, we can find its neighbours along transition paths of the font network, and then combines multiple recognition results for a final decision. All above mentioned merits are demonstrated by the subsequent experiments.

The remainder of the paper is organized as follows. Section 2 gives an introduction about large-scale font set preparation. Section 3 elaborates the detail procedure of font network construction. In section 4, we conduct experiments and analysis. Section 5 concludes the whole paper and outlines our future works.

2 Large-Scale Font Set Preparation

This section is devoted to a description of large-scale font set preparation. To simplify the problem, the proposed font network is only targeted at the capital alphabet “A” in the current trial. Note that since the process of font network construction is independent of alphabet class, it is feasible and tractable to further accommodate arbitrary alphabet classes. We totally collected 6930 “A”s without font duplication, and normalized each one of them to a 200×200 binary image. Figure 2 (a) shows 140 normalized font samples. It should be mentioned that we manually excluded several highly decorative font samples as well as the ones whose main character parts are normal but decorated with various surroundings. We deem this pre-filtering manipulation impartial since even humans may also be hard to make an explicit judgement whether they belong to alphabet or not. See the examples of excluded ones in Fig.2 (b).

Fig. 2: (a) Examples of normalized font samples. (b) Examples of excluded font samples.

3 Font Network Construction

This section introduces the procedure of font network construction based on the prepared large-scale font set. We adopt MST algorithm to build network and Chamfer distance to measure the dissimilarity between two font samples. To grasp the font distribution, font clusters and typical font samples on the network are then extracted via introducing a distance threshold and some clustering criterion.

3.1 Minimum Spanning Tree

Minimum Spanning Tree (MST) also called minimum weight spanning tree is a pivotal concept in graph theory. Given a connected, undirected graph $G(V, E)$ with vertices $v \in V$ and edges $(v_i, v_j) \in E$ corresponding to pairs of neighboring vertices, a spanning tree $T(V, E')$ of that graph G is a subgraph, namely $E' \subseteq E$, so that all pairs of the vertices are connected by one and only one path. Obviously, a graph can generate many different spanning trees. By assigning a weight $w(v_i, v_j)$ to each edge, MST can then be defined as a spanning tree that has the minimal sum of the weights of the edges E'. In our proposal, font samples in the large-scale set serve as vertices, and the Chamfer distance (see details in 3.2) between v_i and v_j is regarded as the weight $w(v_i, v_j)$. We adopt Prim's algorithm [11] to construct MST on the large-scale font set, which iteratively

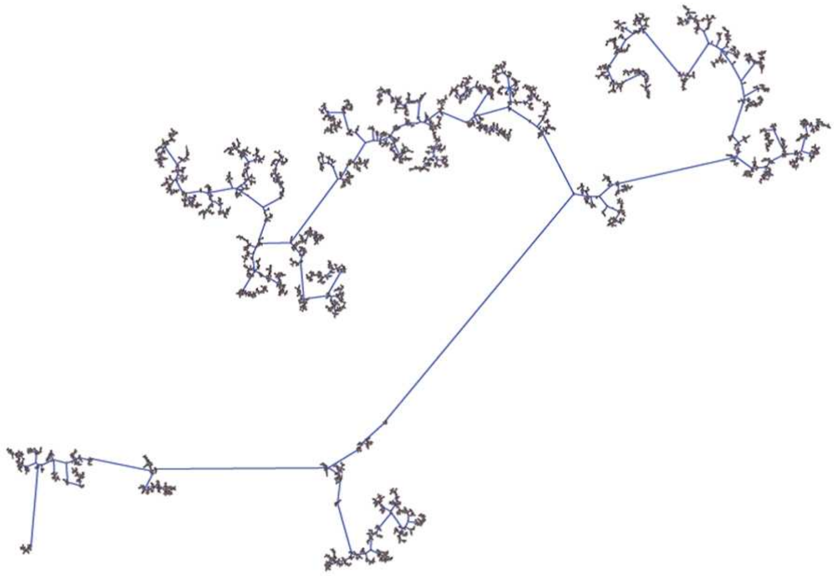


Fig. 3: Visual structure of a font network (MST).

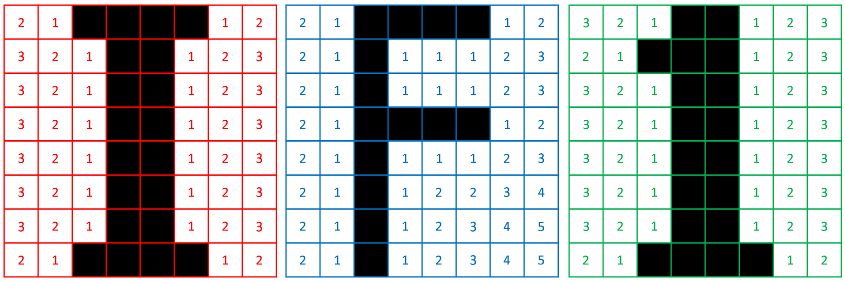
adds edges with smallest weights in a greedy matter, and runs in polynomial time.

The advantages of using MST can be summarized as follows: (1) MST not only reflects the global structure of the set via spanning all font samples, but also naturally guarantees that each local edge connects two font samples which are most similar to each other; (2) the path between two vertices allows us to observe the font shape transition; (3) without using dimension reduction projection, MST well preserves the dissimilarity between two font samples and reliably provides a visual network structure of the font feature space (see Fig.3); (4) as one type of network structure, MST is compatible with general graph or network theory.

3.2 Chamfer Distance

In this proposal, Chamfer distance [12] is employed to reliably measure the dissimilarity between two font samples and the resulting value serves as the weight $w(v_i, v_j)$ for MST construction. Unlike some naive distance measurement which directly accumulates the absolute difference of pixel intensities, Chamfer distance is average nearest distance from one image to another one so that it is more applicable to shape matching. In the practical algorithm implementation, distance transform is employed to reduce the computational cost of calculating Chamfer distance. This is because distance transform directly stored the wanted nearest distance by labeling each pixel of the image with the distance to the nearest boundary pixel as displayed in Fig.4. More specifically, given two font

(a)



(b)

Fig. 4: Examples of distance transform. (a) An actual example (left = original image; right = distance map). (b) Artificial examples where the numbers in the table stand for distance values (left = a capital alphabet “I”; middle = a capital alphabet “F”; right = a number “1”).

samples P and Q which are binary images of size 200×200, p and q denote the corresponding distance maps, respectively. The Chamfer distance is computed as follows:

$$D_{\text{Chamfer}}(P, Q) = \max(d_\alpha, d_\beta)$$

$$d_\alpha = \frac{\sum d(P(i,j);q)}{B(P)} \quad \text{and} \quad d_\beta = \frac{\sum d(Q(i,j);p)}{B(Q)}$$

where $B(P)$ counts the number of black pixels of an image P. The operator $d(P(i,j);q)$ is defined as

$$d(P(i,j);q) = \begin{cases} q(i,j) & \text{if} \quad P(i,j) = 0 \text{ (black pixel)} \\ 0 & \text{otherwise} \end{cases}$$

where $P(i,j)$ represents the pixel value in the position (i,j) of an image P. As we can see, the nearest distance can be obtained by visiting the same position of the corresponding distance map. Thus, distance transform helps reduce the computational complexity to the level of look-up table in the course of calculation.

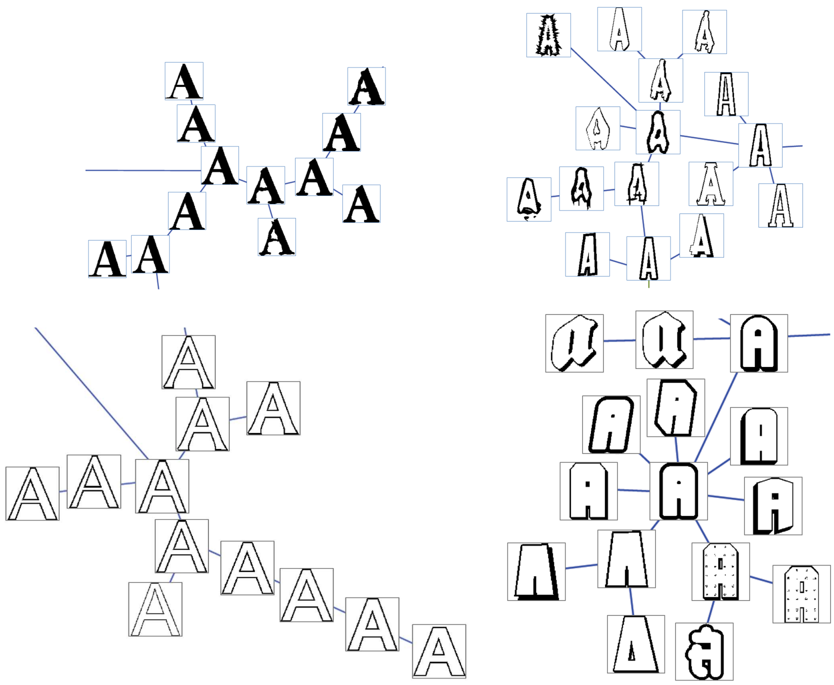


Fig. 5: The internal structures of four clusters.

3.3 Network-Based Clustering

To well grasp the font distribution, we propose a network-based clustering, by which font samples with short Chamfer distances to each other are grouped together. In doing so, a coarser overview of the original font network (MST) can be obtained that allows us to investigate its global configuration on different scales. The local details are reflected in the fact that font samples within a cluster share similar shape. Note that both the global configuration and the local details indicate the distribution of the large-scale font set. Moreover, to effectively represent and observe font clusters, we utilize some centrality criterion (see details in 3.4) to extract typical font samples.

The clustering starts with setting a distance threshold T_D. Then, traverse the vertices throughout the font network. If $w(v_i, v_j) \leq T_D$, the vertices v_i and v_j are grouped into a same font cluster.

Network-based clustering preserves the internal structure of each cluster as shown in Fig.5, and thus allows us to locally observe the font shape transition even after the clustering. More importantly, small size clusters survive from the network-based clustering as long as these clusters are essentially away from others in terms of Chamfer distance. On the contrary, K-means algorithm is approximately equally divided the whole feature space into several clusters so

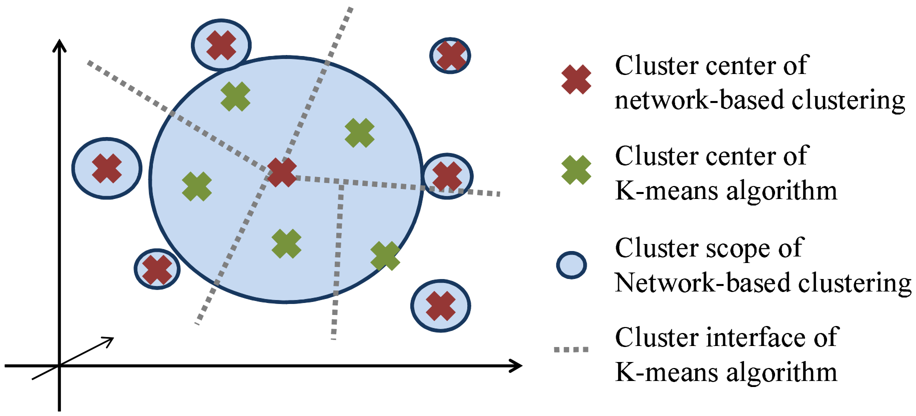


Fig. 6: An illustration indicating the different performance between the network-based clustering and the K-means algorithm. Experimental results of comparative study can be found in Table 1.

that scattered font samples with special shape are forced to merge into dissimilar large clusters. This contrast is illustrated in Fig.6 and a subsequent experiment. Here, the size of a cluster refers to the number of samples belonging to that cluster.

3.4 Typical Font Sample Extraction by Centrality Criterion

To extract a typical font sample from each cluster, this subsection introduces three centrality criteria, namely closeness centrality, eccentricity centrality and betweenness centrality, which are widely employed in the network analysis [13]. The centrality criterion estimates the degree of center for each vertex, and returns comparable scores. In this proposal, we adopt the centrality criterion to extract the typical font sample from each cluster. In the following, three centrality criteria are explained in turn.

Closeness centrality is based on the natural distance metric between all pairs of vertices as given below.

$$C_C(v_i) = \frac{n-1}{\sum_{j=1}^{n} d(v_i, v_j)}, \quad i = 1, 2, \cdots, n,$$

where n denotes the size of a font cluster. The distance $d(v_i, v_j)$ takes the sum of $w(v_i, v_j)$ along the path from v_i to v_j. Recall that $w(v_i, v_j)$ is the Chamfer distance between two directly connected vertices. A vertex v_i with largest $C_C(v_i)$ is considered as the typical font sample.

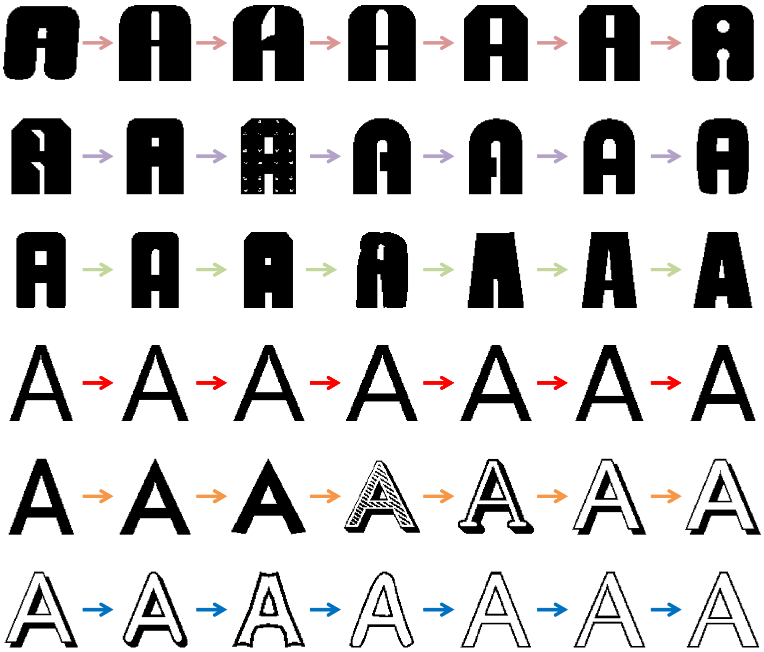


Fig. 7: Font shape transition along paths.

Eccentricity centrality selects the center sample by comparing all pairs of maximum distances. That is

$$C_E(v_i) = \frac{1}{\max d(v_i, v_j)}, \ \ i, j = 1, 2, \cdots, n.$$

The denominator $\max d(v_i, v_j)$ can be defined as the degree of eccentricity so that a larger $C_E(v_i)$ indicates a more compact extent that the vertices v_j (where $j = 1, 2, \cdots, n$) gather around v_i.

Betweenness centrality quantifies the number of times that a vertex acts as a bridge along the path between two other vertices. More specifically, the betweenness centrality can be represented as follows.

$$C_B(v_i) = \sum\nolimits_{s \neq i \neq t = 1}^{n} \frac{\sigma_{v_s v_t}(v_i)}{\sigma_{v_s v_t}}, \ \ i = 1, 2, \cdots, n,$$

where $\sigma_{v_s v_t}$ is total number of edges from v_s to v_t ($\forall s \neq t \in \{1, 2, \cdots, n\}$) and $\sigma_{v_s v_t}(v_i)$ accumulates the number of times that all these paths pass through v_i. The betweenness centrality relies on the natural fact that the center vertex has a greater opportunity to be passed through by paths. Therefore, the vertex v_i having the largest $C_B(v_i)$ corresponds to the typical font sample.

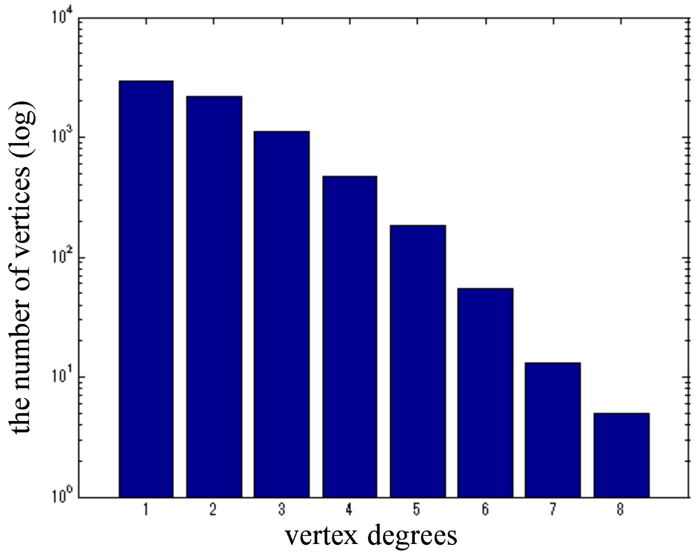


Fig. 8: Vertex degree histogram.

4 Experiment and Analysis

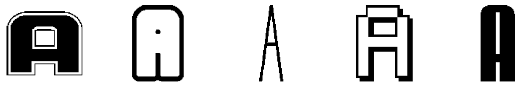
4.1 Font Network

In this experiment section, the MST algorithm and the Chamfer distance were applied to construct the font network of the large-scale alphabet "A" set. The global structure of the built network was displayed in Fig.3. Further, according to the metric of dissimilarity, the network-based clustering algorithm divided the feature space into several clusters without affecting their internal structures (see Fig.5). In addition, figure 7 exhibited the font shape transition between two vertices, which was useful to generate new fonts or recognize various scene characters. It was worthwhile to point out that we could still observe the font shape transition even after the clustering processing, which benefited from the above mentioned structure preservation property. Moreover, after introducing some centrality criterion like closeness centrality, eccentricity centrality or betweenness centrality, the typical font sample could be extracted from each cluster. Note that the typical font sample provided an effective representation for each font cluster. All above investigations allowed us to grasp the distribution of a large number of font samples.

4.2 Vertex Degree Histogram

The degree of a vertex is defined as the number of edges incident to the vertex. Figure 8 showed the histogram of vertex degree and provided another aspect of

(a)



(b)

Fig. 9: The hub font samples. (a) The font samples having degree 7; (b) the font samples having degree 8.

the font distribution. As we could see, the maximum vertex degree is eight. In addition, the number of vertices diminished with the increase of degree, which implied that the vertices having highest degree could be considered as the hubs of a network. See the hub font samples in Fig.9. Note that the hub font samples were different from the typical ones. The former was selected from the whole network and reflected the global distribution while the latter was the local representative extracted from each font cluster using some centrality criterion.

4.3 Font Cluster

Since the distance threshold T_D had a great impact on clustering, in this subsection, we analyzed the configuration evolution of font clusters. We displayed the evolution of the maximum size and the number of clusters versus the increase of T_D in Fig.10, which demonstrated the existence of the font cluster.

Table 1 listed the size of the top five largest font clusters with respect to each distance threshold T_D. Furthermore, the result of K-means algorithm ($K = 5$) was given in the last row of Table 1. The results listed in Table 1, also illustrated in Fig.6, indicated that the network-based clustering could preserve small size font clusters which contained samples taking special shapes, while K-means approximatively equally divided the feature space.

4.4 Typical Font Sample

Typical font samples were extracted by introducing some centrality criterion, namely closeness centrality, eccentricity centrality or betweenness centrality, as described in 3.4. The selected font samples from the top five largest clusters were exhibited in Fig.11. The digits printed above each sub-picture stood for the size of the cluster from which the typical font sample was extracted. As we could observe, when the distance threshold T_D was small, for example $T_D = 0.30$, the

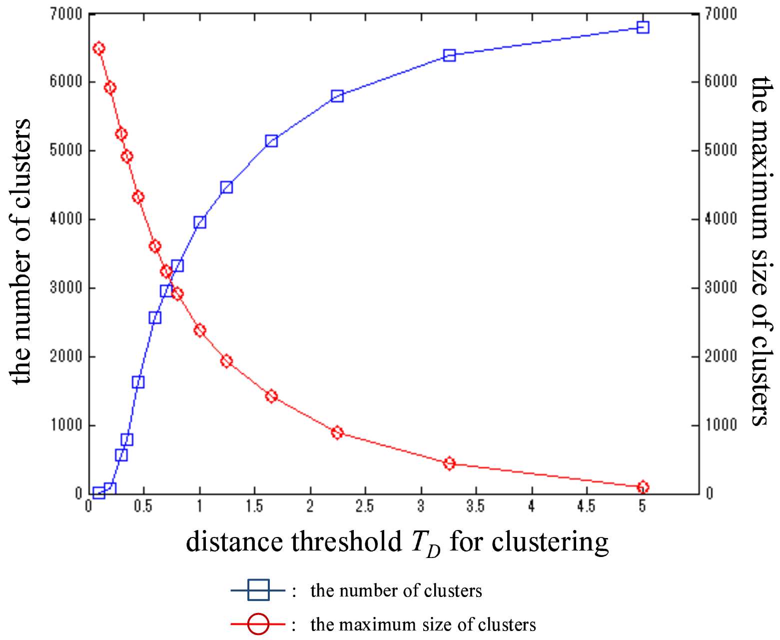
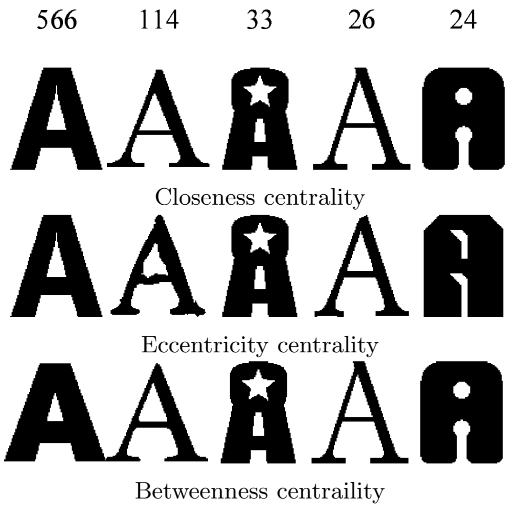


Fig. 10: The evolution of the maximum size and the number of clusters versus the increase of T_D.

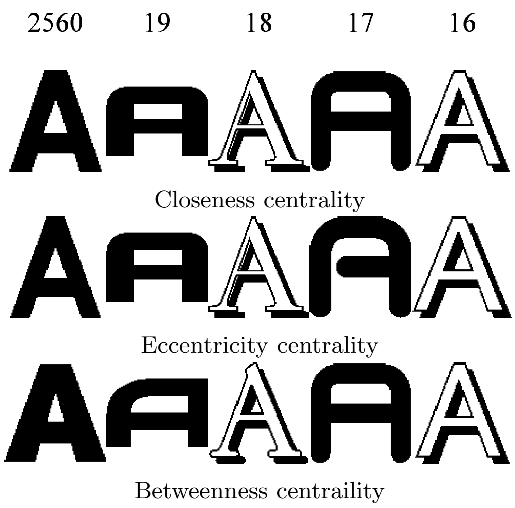
typical font samples shared small dissimilarity to each other. With the increase of T_D, the great dissimilarity among the typical font samples emerged as shown in Fig.11 (c). Moreover, the typical font sample from the largest cluster shared the similar standard shape regardless the change of T_D, which indicated that no matter how protean a font sample was, its appearance would hold approximately constant structures. In addition, applying three centrality criteria leaded to similar results. In other words, centrality criterion would not sharply affect the process of typical font sample extraction.

5 Conclusion

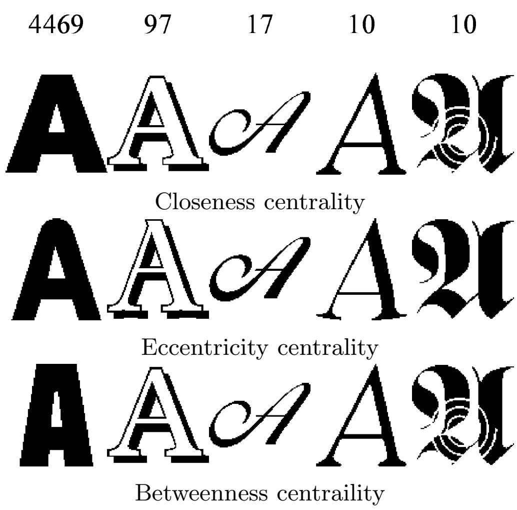
In this paper, we analyze the font distribution of a large-scale set by network, which opens a new door to the camera-based OCR engines. To construct the font network, we adopt MST algorithm under the dissimilarity measurement using Chamfer distance. Font clusters are formed though setting distance threshold. After that, we extract typical font samples from clusters by introducing some centrality criterion, namely closeness centrality, eccentricity centrality and betweenness centrality. Benefitting from the network structure, both the global configuration and the font shape transition can be observed. Compared with the



(a) Distance threshold $T_D = 0.30$.



(b) Distance threshold $T_D = 0.60$.



(c) Distance threshold $T_D = 1.25$.

Fig. 11: The typical font samples extracted by centrality criterion.

Table 1: The size of font clusters.

	T_D	1st	2nd	3rd	4th	5th
Network-Based Clustering	0.10	10	8	7	6	5
	0.20	67	66	37	32	23
	0.30	566	114	33	26	24
	0.35	796	238	63	21	18
	0.45	1630	26	22	22	22
	0.60	2560	19	18	17	16
	0.70	2953	25	18	18	17
	0.80	3330	42	18	17	17
	1.00	3957	79	17	14	10
	1.25	4469	97	17	10	10
	1.65	5149	23	10	10	10
	2.25	5816	28	11	8	8
	3.25	6403	6	6	5	5
	5.00	6813	5	3	3	2
K-means	$K = 5$	2128	1444	1308	1128	922

conventional PCA, the proposed font network realizes distribution visualization through Chamfer distance rather than the process of dimensionality reduction. Moreover, as verified by experiments, the network-based clustering preserves small size font clusters, while K-means algorithm will produce an approximately equal division. The existence of font cluster and the effectiveness of network-based analysis are also demonstrated by experiments. Our future work is to extract the internal structures from font clusters, and to design regularization approaches based on the path of font shape transition.

Acknowledgment

The authors would like to thank the support of Creation of Human-Harmonized Information Technology for Convivial Society, which is a CREST project organized by Japan Science and Technology Agency (JST).

References

1. B. Epshtein, E. Ofek, and Y. Wexler, Detecting Text in Natural Scenes with Stroke Width Transform, IEEE Conference on Computer Vision and Pattern Recognition, 2010.
2. K. Wang and S. Belongie, Word Spotting in the Wild, European Conference on Computer Vision, 2010.
3. K. Wang, B.Babenko and S. Belongie, End-to-End Scene Text Recognition, IEEE International Conference on Computer Vision, 2011.
4. A. Mishra, K. Alahari and C. Jawahar, Top-Down and Bottom-Up Cues for Scene Text Recognition, IEEE Conference on Computer Vision and Pattern Recognition, 2012.
5. Y. Zhu, T. Tan, and Y. Wang, Font Recognition Based on Global Texture Analysis, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.23, no.10, pp.1192-1200, 2001.
6. S. Omachi, M. Inoue, and H. Aso, Structure Extraction from Decorated Characters Using Multiscale Images, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.23, no.3, pp.315-322, 2001.
7. S. Omachi, S. Megawa, and H. Aso, Decorative Character Recognition by Graph Matching, IEICE Transactions on Information and Systems, vol.E90-D, no.10, pp.1720-1723, 2007.
8. S. Wang, S. Uchida, and M. Liwicki, Part-Based Recognition of Arbitrary Fonts, International Conference on Document Analysis and Recognition, 2013.
9. G. Borgefors, Hierarchical Chamfer Matching: A Parametric Edge Matching Algorithm, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.10, no.6, pp.849-865, 1988.
10. D. Shi, S. Gumm, and R. Damper, Handwritten Chinese Radical Recognition Using Nonlinear Active Shape Models, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.25, no.2, pp.277-280, 2003.
11. T. Cormen, C. Leiserson, R. Rivest and C. Stein, Section 23.2: The algorithms of Kruskal and Prim, Introudction to Algorithms, Third Edition, MIT Press, pp.631-638, 2009.
12. H. Barrow, J. Tenenbaum, R. Bolles and H. Wolf, Parametric Correspondence and Chamfer Matching: Two New Techniques for Image Matching, International Joint Conference Artificial Intelligence, 1977.
13. T. Opsahl, F. Agneessens and J. Skvoretz, Node Centrality in Weighted Networks: Generalizing Degree and Shortest Paths, Social Networks, vol.32, no.3, pp.245-251, 2010.