2012 10th IAPR International Workshop on Document Analysis Systems

Toward Part-based Document Image Decoding

Wang Song, Seiichi Uchida
Kyushu University, Fukuoka, Japan
wangsong@human.ait.kyushu-u.ac.jp, uchida@ait.kyushu-u.ac.jp

Marcus Liwicki
DFKI, Kaiserslautern, Germany
Marcus.Liwicki@dfki.de

***Abstract*—Document image decoding (DID) is a trial to understand the contents of a whole document without any reference information about font, language, etc. Typically, DID approaches assume the correct segmentation of the document and some a priori knowledge about the language or the script. Unfortunately, this assumption will not hold if we deal with various documents, such as documents with various sized fonts, camera-captured documents, free-layout documents, or historical documents. In this paper, we propose a part-based character identification method where no segmentation into characters is necessary and no a priori information about the document is needed. The approach clusters similar keypoints and groups frequent neighboring keypoint clusters. Then a second iteration is performed, i.e., the groups are again clustered and optionally pairs frequent group clusters are detected. Our first experimental results on multi font-size documents look already very promising. We could find nearly perfect correspondences between characters and detected group clusters.**

***Keywords*-Document image decoding; part-based;**

I. Introduction

Document image decoding (DID) is an approach to recognize a given document as a signal sequence [1], [2]. DID is a kind of deciphering process. Without character templates and precise segmentation, DID tries to recognize the entire printed text. For a very simple example, if a frequent pattern is identified in an English document image, this pattern can be guessed as "e". A typical DID system is often accompanied with a language model [3]. From the document analysis and recognition viewpoint, DID is a very promising strategy because DID can recognize texts in printed documents having various or even unknown fonts.

In the past research there have been some trials which aim to improve the DID to a font-free or even language-free system. In [4], the authors try to extract character templates from the original document image. Thus this method can be seen as a template-free method, i.e., a font-free OCR system. The method in [4] starts from finding occurrences of the word "a". This is because "a" is a single word in English and it is also frequently used. With the template "a" and the language model the rest templates of other letters are extracted. This method, however, have limitations. One obvious limitation is it requires a significant word like "a". Furthermore, the method is not robust against changes in the font size.

Another and more severe limitation is that it needs not only text line segmentation but also a rough inline segmentation (hopefully, into characters). This limitation can be found in many other DID systems. In fact, the identification of the same character often starts from segmentation and it requires a clear gap between every neighboring character pair and/or prior knowledge about the document (language, font, font size, etc.). However, these conditions are difficult to be satisfied especially for the target images of DID systems. Although the DID system is robust to some segmentation errors or incorrect template matches [5], up to now, not too much attention is paid to it.

Consequently, if we can develop a character identification method which is free from the above conditions, the DID system can deal with various documents, such as documents with multi sized fonts, camera-captured documents, free-layout documents, different languages, or historical documents. The DID system will be beneficial for the typical OCR systems because they always suffer from the problem of various document recognition.

Recently, some part-based method for handwritten character recognition was presented(e.g., [6]). This kind of method uses parts of a character for recognition. Based on the character parts, it is possible to design preprocessing-free and segmentation-free character recognition system.

Inspired by above character recognition method, in this paper, we will present a part-based character identification method for DID. This method is supposed to be preprocessing-free, segmentation-free, font-free and even language-free. Of course, like the other DID systems, it does not use any character template database. Given a document image, this method will find out similar characters based on the character parts (later also called keypoints) and then generate outputs of the locations of the same character. Another strength of the proposed method is that it can deal with multiple font-sizes; this important and novel property is realized by the recent development of scale-invariant keypoint extraction methodology. In our experiments, we use multi font-size documents as a test bed and show promising identification results.

II. Methodology

Figure 1 shows the overall framework of the part-based character identification method. Since we use the speeded-up robust features (SURF) [7] as the feature extraction method, each character part is represented as a SURF keypoint. There are two clustering processes. One is keypoint clustering and

978-0-7695-4661-2/12 $26.00 © 2012 IEEE
DOI 10.1109/DAS.2012.90

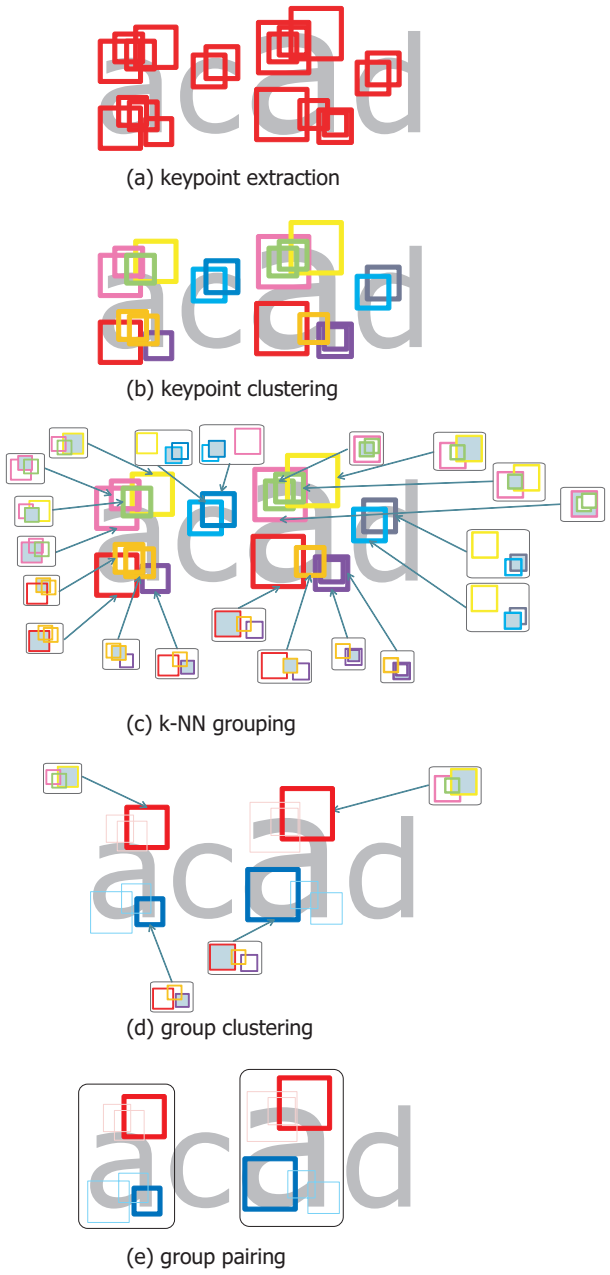


Figure 1. The part-based character identification method.

the other is group clustering where a "group" is neighboring keypoints. After group clustering, we can have a character identification result.

A. Keypoint extraction

SURF has been widely used in image processing and classification. A merit of SURF is that it is scale-invariant and can deal with multi-font sizes. Figure 2 shows the process of SURF. SURF first detects keypoints (i.e., location

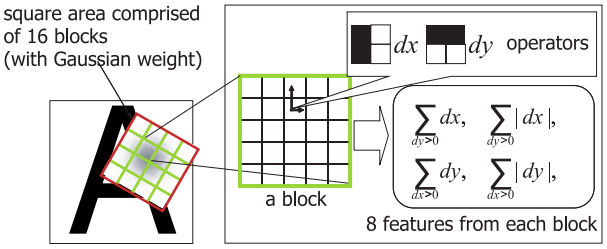


Figure 2. The process of SURF.

of local parts) as local maxima of approximate Hessian values in scale space. Second, SURF describes each local part as a 128-dimensional feature vector. The element of the vector is a local directional feature value. Although the original SURF keypoint is rotational as shown in the Fig. 2, we will use the non-rotational (i.e., upright) SURF keypoint just because of simplicity.

B. Keypoint clustering

All the extracted keypoints are then subjected to a density-based clustering process for finding similar and frequent keypoints in the SURF vector space. The density-based clustering is useful for finding out clusters comprised by the frequent keypoints. (In fact, k-means often have many clusters with small populations.) The process of density-based clustering has three steps as followings.

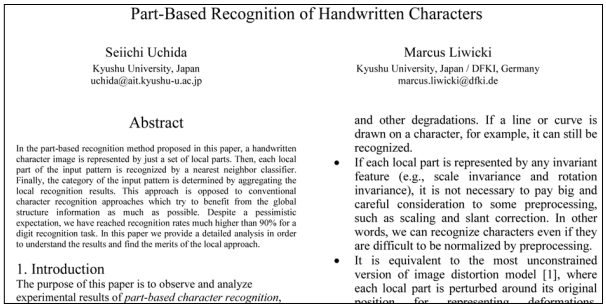
- First, each extracted keypoint is seen as a center keypoint and then its neighbors in SURF space are found. The distance (Euclidean distance) between this keypoint and its neighbors should be less than the radius parameter. The number of those nearest neighbors including this center keypoint is seen as the density of this keypoint.
- Second, all the keypoints are sorted by their density.
- Third, the keypoints which are close to each other are combined to one cluster.

C. Grouping

Since a keypoint covers only a local part of a character, grouping of neighboring keypoints is necessary. Specifically, we will find k-nearest-neighbor (k-NN) of a keypoint (called a target keypoint) from the image to create a group. The target keypoint is the center of its k-NN group. Note that the positional distance between two keypoints is used for measuring the neighborliness. With different k, the size of the group is different. One criterion of tuning k is to find group representing a discriminative part of the letter.

D. Group clustering

Group clustering is then performed for findings similar groups, which will represent the same part of a character.



Part-Based Recognition of Handwritten Characters

Seiichi Uchida
Kyushu University, Japan
uchida@ait.kyushu-u.ac.jp

Marcus Liwicki
Kyushu University, Japan / DFKI, Germany
marcus.liwicki@dfki.de

Abstract

In the part-based recognition method proposed in this paper, a handwritten character image is represented by just a set of local parts. Then, each local part of the input pattern is recognized by a nearest neighbor classifier. Finally, the category of the input pattern is determined by aggregating the local recognition results. This approach is opposed to conventional character recognition approaches which try to benefit from the global structure information as much as possible. Despite a pessimistic expectation, we have reached recognition rates much higher than 90% for a digit recognition task. In this paper we provide a detailed analysis in order to understand the results and find the merits of the local approach.

1. Introduction

The purpose of this paper is to observe and analyze experimental results of *part-based character recognition*,

and other degradations. If a line or curve is drawn on a character, for example, it can still be recognized.

- If each local part is represented by any invariant feature (e.g., scale invariance and rotation invariance), it is not necessary to pay big and careful consideration to some preprocessing, such as scaling and slant correction. In other words, we can recognize characters even if they are difficult to be normalized by preprocessing.
- It is equivalent to the most unconstrained version of image distortion model [1], where each local part is perturbed around its original

Figure 3. Extract of the multi font-size document used in our experiments.

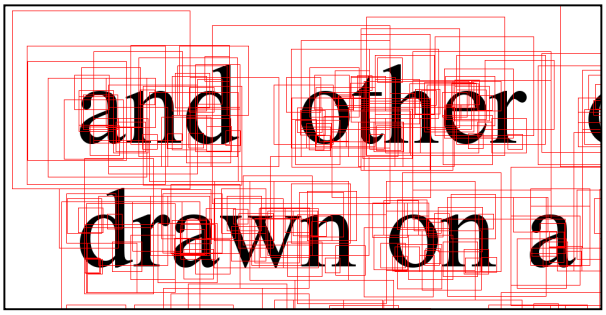


Figure 4. SURF keypoint extraction. Only 1/5 of the keypoints are shown.

We again use the density-based clustering method with the following distance d between two groups:

$$d = k + 1 - s, \tag{1}$$

where s is the number of labels of the largest common subset that two groups share. For example, if $k = 4$, group 1 contains the labels of keypoint as $(1, 2, 3, 4, 5)$ and group 2 contains the labels as $(1, 2, 3, 8, 9)$, then their largest common subset is $(1, 2, 3)$. Consequently, $s = 3$, so $d = k + 1 - s = 2$. A radius parameter is also needed in the group clustering. As mentioned above, by using large radius, we can have flexible comparison of the groups. In experiments reported in Section III, we will use $k = 49$ along with a radius of 25.

E. Group pairing

Group pairing is an option of our character identification method. Some group cluster may represent a part which is shared by different letters. For example, the "v", "y", and "w" all have similar parts on the top. This may not matter for the final DID results if we care the existence of such group clusters. If we perform group pairing which combines neighboring and frequent group pairs to be a new group, it is possible to make individual groups for "v", "y" and "w".

III. Experiments

The target of our experiments is to evaluate if the part-based system could produce useful results on a given multi font-size document. Therefore we generated a document using Microsoft Word containing different font-sizes from 8 to 15. The top part of this document is depicted in Fig. 3.

This document is first transformed into gray-scale image and then enlarged to $10,200 \times 13,200$ in order to extract enough keypoints. Figure 4 shows the keypoint extraction results. As can be seen most keypoints are located in the character area.

Figure 5 shows the keypoints of three clusters after keypoint clustering on the document. As expected, most of the keypoints have a very similar shape. However, often the

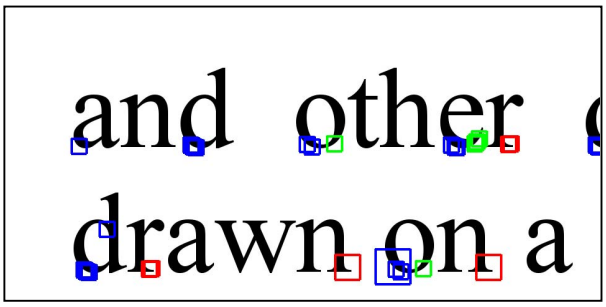


Figure 5. Three keypoint clusters after keypoint clustering. Each cluster is denoted by a different color.

keypoints of same cluster correspond to different characters. This effect will be reduced after the next clustering process.

The next step is the generation of groups as described in Section II-C. After trying several parameters for k on a sample document, we have chosen $k = 49$ to create the groups. Figure 6 shows randomly selected target keypoints after grouping. As can been seen in the figure, there are various sizes of keypoints in one group.

Figure 7 shows a cluster after group clustering. In this figure each red box denotes the target keypoint of a group. Note that for easier assessing the results, the sizes of the boxes are unified. As can be seen, group clusters often belong to the same letter, e.g., the lower parts of "g" for this group cluster. Note that the group clusters are again sorted by their frequency.

It is an interesting observation that many of the group clusters belong to the same letter category (although some others are from multiple letters). Also, sometimes two or more group clusters belong to the same letter but at another position.

To analyze this behavior more detailed we selected some of the most frequent group clusters which mainly belong to one letter.[1] The results are shown in Table I. For each cluster the corresponding letter and the percentage of keypoints

[1]most of the most frequent clusters belong to instances of "e". Due to space limitations, they are omitted here.

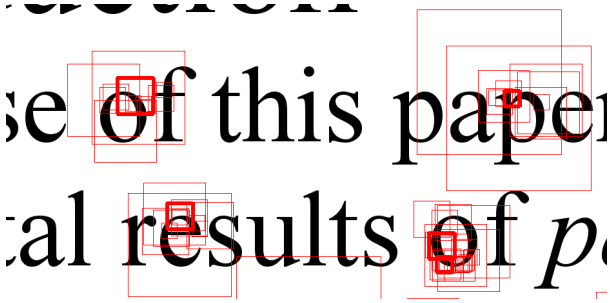


Figure 6. Result of the grouping for randomly target keypoints. The thick box denotes the target keypoint and the thin boxes represent a subset of the k-NN keypoints.

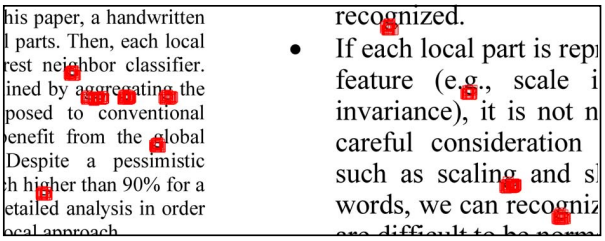


Figure 7. Cluster 8 after group clustering.

belonging to that letter are shown. Figure 8 depicts the corresponding occurrences of these keypoints in selected document region. It can be observed that most of the groups belong to parts of a single letter and the noise is very low. Only for the letter "y" there is some mislabeling. This is due to the fact that the group cluster 13 is located at the upper part of the "y" and this region is very similar to "w" and "v". We expect to overcome this problem after group pairing.

As the major aim of this paper is to detect letters, we have analyzed the behavior of the group clusters from

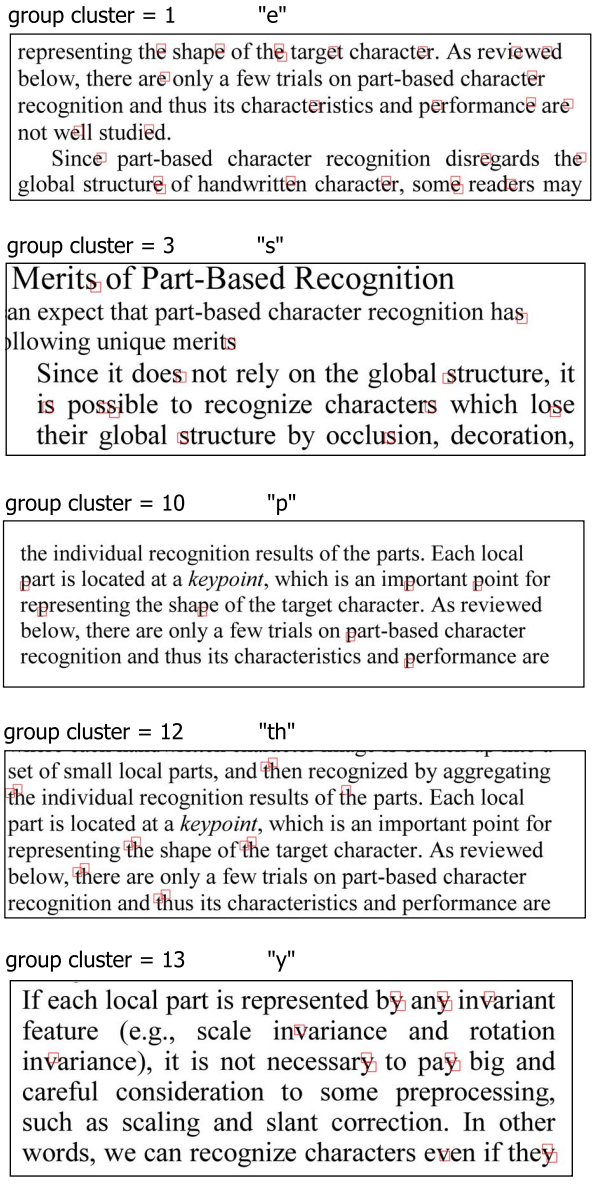


Figure 8. Output of group clusters.

Table I
CORRESPONDENCES OF THE MOST FREQUENT CLUSTERS TO SINGLE LETTERS.

Group cluster	Major letter	Frequency (%)
1	"e"	99.98
3	"s"	99.98
5	"e"	99.99
10	"p"	99.99
11	"w"	100
12	"th"	100
13	"y"	64.71

Table II
GROUP CLUSTERS FOR SELECTED LETTERS

Letter	Group cluster	Total	Extracted	Ratio(%)
"e"	1,5,6,15,17	452	367	81.19
"g"	8,19,20	84	83	98.81
"w"	11	35	35	100

the perspective of the letters as well. Table II shows the result of three letter categories. It is a very encouraging result that almost all instances of "g" and "g" are detected. Unfortunately, the method failed at most "e"s in the abstract. This is due to the fact that at those positions less keypoints are extractive. In other words, our method is sensitive to the parameter k. In future we will overcome this problem by using multiple values for k and combine the results. Note that group cluster 8 covers already a major part of these letters; for "w" all letters are covered by the group cluster 11.

As suggested in Section II-E, we further plan to pair the group clusters and finally hope to detect complete character

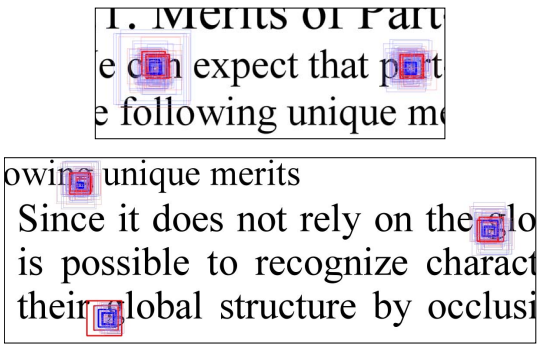


Figure 9. Group pairs.

Table III
CORRESPONDENCES OF THE MOST FREQUENT CLUSTERS TO SINGLE LETTERS FOR THE SECOND DOCUMENT.

Group cluster	Major letter	Frequency (%)
10	"e"	95.95
15	"s"	95.43

shapes. However, a complete analysis of the pairing behavior is beyond the scope of this paper. Due to space limitations we will only report on first promising observations, i.e., Fig. 9 shows the group pairs of the letters "a" and "g". The groups pairs are represented by a dark blue and a dark red box corresponding to one another. Note that some group pairs overlap and therefore several red and blue boxes are depicted.

In order to test the robustness of our method we have applied the same strategy on another document with a different font and font sizes ranging from 8 to 20. The corresponding results are shown in Tables III and IV, respectively. The results are also very promising. Note that the second document can be seen as a more challenging case since more variations are present.

IV. Conclusion and Future Work

In this paper we introduced a part-based character identification method. Our method first applies SURF keypoint extraction and then performs several clustering steps, i.e., a density-based clustering in the SURF descriptor space; a location-based nearest-neighbor grouping in the 2D document space; a second clustering of the groups; and a final (optional) pairing of group clusters.

In our experiments on multi font-size documents we observe an improvement of the identification at each stage.

Table IV
GROUP CLUSTERS FOR SELECTED LETTERS OF THE SECOND DOCUMENT

Letter	Group cluster	Total	Extracted	Ratio(%)
"e"	2,15	336	316	94.05
"s"	6,10	157	148	94.27

An in-depth analysis after group clustering shows that some characters like "g" and "w" are almost perfectly represented by one cluster. Other characters like "e" do not belong to a single group cluster. However, a combination of several group clusters results already in a good coverage. The group clusters themselves often belong mainly to a single character. Since group clusters only represent a part of a character, some of them belong to multiple characters, like "y", "w", and "v". We expect to overcome this problem after group pairing.

Therefore, our main aim for future work is to develop a group pairing method which is more robust to severe changes in font size, e.g., by simultaneously using different values of k for keypoint grouping and combining the results. Furthermore, instead of just generating pairs we plan to generate structures by connecting pairs according to the friend-of-a-friend approach.

In the future we will try to apply the proposed method on documents of different difficulties. One idea is to include more font sizes or to use degraded documents. It might also be interesting to apply the rotation-invariant version of SURF and analyze the behavior of the proposed method on warped documents. An interesting side-effect of our method is that the group cluster locations are often at the same positions of the characters which could result into an unconstrained baseline detection strategy.

Acknowledgment

This research was partially supported by JST, CREST, JSPS and the Research Grant (No. 23300072) of The Ministry of Education, Culture, Sports, Science and Technology in Japan.

References

[1] G.E. Kopec and P.A. Chou, "Document image decoding," Image Processing, 1994. Proceedings. ICIP-94., IEEE International Conference , vol.2, no., pp.36-40 vol.2, 13-16 Nov 1994.

[2] G.E. Kopec and P.A. Chou, "Document Image Decoding Using Markov Source Models," IEEE Transactions on Pattern Analysis and Machine Intelligence, pp. 602-617, June, 1994.

[3] Kris Popat, Dan Greene, Justin Romberg and S. Dan Bloomberg, "Adding linguistic constraints to document image decoding: Comparing the iterated complete path and stack algorithms," in Proceedings of IS&T/SPIE Electronic Imaging 2001: Document Recognition and Retrieval VIII, January 2001.

[4] A. Kae and E. Learned-Miller, "Learning on the Fly: Font-Free Approaches to Difficult OCR Problems," 10th International Conference on Document Analysis and Recognition, pp.571-575, 26-29 July 2009

[5] G.E. Kopec and M. Lomelin, "Supervised template estimation for document image decoding," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.19, no.12, pp.1313-1324, Dec 1997.

[6] S. Uchida and M. Liwicki, "Part-Based Recognition of Handwritten Characters," Proc. ICFHR, pp. 545-550, 2010.

[7] H. Bay, T. Tuytelaars, and L. V. Gool, "SURF: Speeded Up Robus Features," Proc. ECCV, 2006.