

A Multi-Level Component Grouping Algorithm and Its Applications

Bo Yuan^{1,2} and Chew Lim Tan²

¹Centre for Remote Imaging, Sensing and Processing

²Department of Computer Science, School of Computing

National University of Singapore

¹yuanbo@nus.edu.sg, ²tancl@comp.nus.edu.sg

Abstract

This paper describes a fast and effective component grouping algorithm for some layout analysis problems of textual document images. Based on an empirical function, it discovers neighborhood relations at the component level. Time-consuming algorithms that operate globally among components may obtain significant performance boost from the localization information that this grouping algorithm provides. Some of the applications using this grouping function are also demonstrated.

1. Introduction

Connected-component based methods for textual document images analysis require information about the physical structures or layout of the documents. There are several popular approaches for layout analysis. The bottom-up approaches starts from the detected connected components and move up to higher-level structures of word, text line and paragraph [1][2]. The top-down approaches work along the opposite direction from the page level down to lower-level structures [3][4]. Applications may need to work on only some levels of structures.

This paper describes a fast and effective component grouping algorithm that provides some of the components localization information that the full-fledged page segmentation algorithms offers, such as the identification of word, text lines, and paragraph or column of textual documents.

The proposed algorithm can do multiple levels of component grouping in either top-down or bottom-up fashion. In principle, this grouping algorithm checks available pairs of connected components on a page to determine whether they belong to the same group. Its empirical grouping function depends not only on the distances in between the component pairs, but also their

sizes and the size differences. There is only one adjustable parameter in this function, which enables it working on various levels of structures, in a fixed or recursive style. Its value is determined only once for a batch processing of documents of similar properties, such as images of printed magazines in certain period of time.

2. The Grouping Function

The grouping function f is chosen as

$$f(s_1, s_2) = \sqrt{\frac{ks_1s_2}{s_1 + s_2}}$$

where s_1, s_2 are the sizes of two connected components and k is an adjustable parameter that determines the grouping level. This grouping function has several desirable properties:

- It is a distance measure which is proportional to the square root of component size;
- It is symmetric for any two components;
- It is rotation invariant;
- It has strong tendency towards components of similar sizes, which is similar to the effective resistance of paralleled resistors. This also makes the grouping process more tolerant to the interferences from graphical elements and other source of noises.

To group a set of connected components, a pair of components is chosen and the f value is computed with a given parameter k . If the f value is no larger than the distance between the pair, the two components are considered belonging to the same group. This test continues until all unique pairs are tested.

The grouping function has a time complexity that is better than $n(n-1)/2$, where n is the total number of components being tested for grouping. This because when components A and B are in the same group, if component C is tested to be in the same group as component B , then

there is no need to test the components *A* and *C*. In other words, given 3 components, only 2 tests are needed rather than 3.

The grouping function can work at multiple levels by choosing different parameter *k*. As Figure 1 shows, when *k* starts from 0, components are all isolated. When *k* becomes larger, components start to get clustered, so the size of the largest group rises. When *k* reaches 7, words are formed. When *k* becomes even larger, text lines are formed. When *k* increases to 17 and above, major paragraphs are formed as in Figure 2. This process continues until *k* is so large that the whole page becomes a single group.

The grouping process can be either top-down with descending *k* or bottom-up with ascending *k*. From the implementation point of view, however, successive grouping with incremental *k* can be much more efficiently implemented in either top-down or bottom-up manners. When in top-down manner, all further split of components is only detected within the components of the current groups. No cross-group detections are needed. When in bottom-up manner, all further aggregation of components is only detected among the components of the current groups. When two components from two current groups are found to satisfy the grouping function *f*, the two groups are merged into one group in the next level. This multi-level grouping can have variable increments in *k*. The whole multi-level grouping process has an irregular pyramid look. The inter-group and inter-level links can be maintained in a multi-dimensional array. One of the purposes of using successive (recursive) grouping is to correct over- or under-grouping by a single *k*.

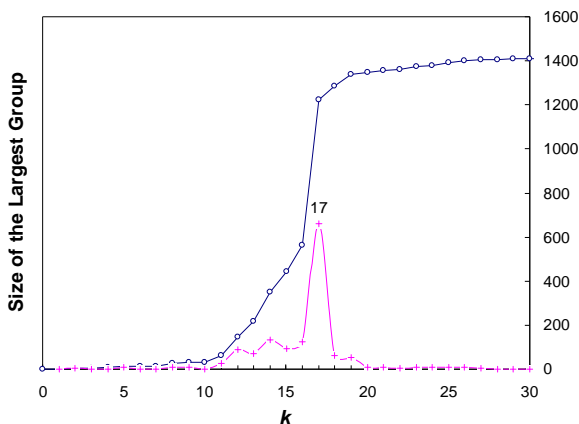


Figure 1. The size (number of components) of the largest group versus the parameter *k* for the image AOOL from UWDB-I. The lower curve is the first derivative of the upper one.

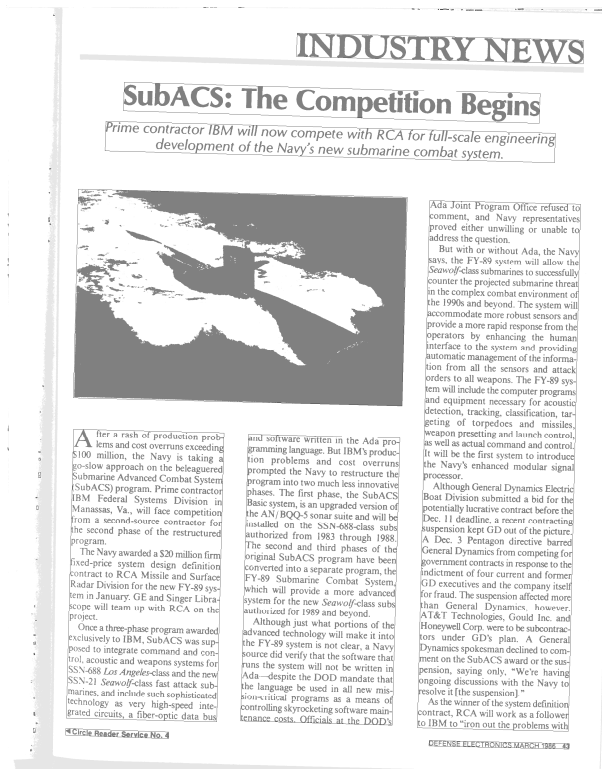


Figure 2. The grouping result of the image A00L from UWDB-I at *k* = 27. The bounds are fit along the skew angles by the proposed skew detector in this paper.

3. The Applications

The proposed component grouping function has some useful applications, such paragraph/column identification, text/graphics separation, skew estimation, etc.

3.1 A speed-up measure

The direct motivation of using such a grouping function is when the authors were involved in designing a skew detector for textual documents [5]. The skew estimation function traverses all component pairs to find co-linearity among them. The skew estimation function involves unavoidable time-consuming computations. If the components are pre-grouped, only the components inside a group are tested for skew; no cross-group tests are necessary. Actually, across-the-group results provide more often than not unreliable, even harmful skew information. With the pre-grouping of the proposed grouping algorithm, the performance obtains a 3 fold boost based

on the evaluation using the 979 real images from the University of Washington English Document Image Database I (UWDB-I). The chosen value for k is 27 based on the fact that for the skew estimation, a little over-grouping is safer than under-grouping. The speedup at this level makes the skew estimator achieve sub-second speed even on sub-GHz computers for the images of sizes larger than 6 Mega pixels. Figure 2 shows an example of the grouping function at work. The bounding boxes of the resultant groups are in black, the grouped components are in dark gray and the filtered out components are in light gray. Columns can be identified at this k value. This is exactly what is desired by the skew estimator whose expansive computation can be drastically reduced by the relatively cheap pre-grouping.

This grouping function serves not only as a speed-up measure, but also a reliability guard in processing some difficult samples. In Figure 3 the two facing pages have different skews. If not separated first, the skew detector may not give the correct result.

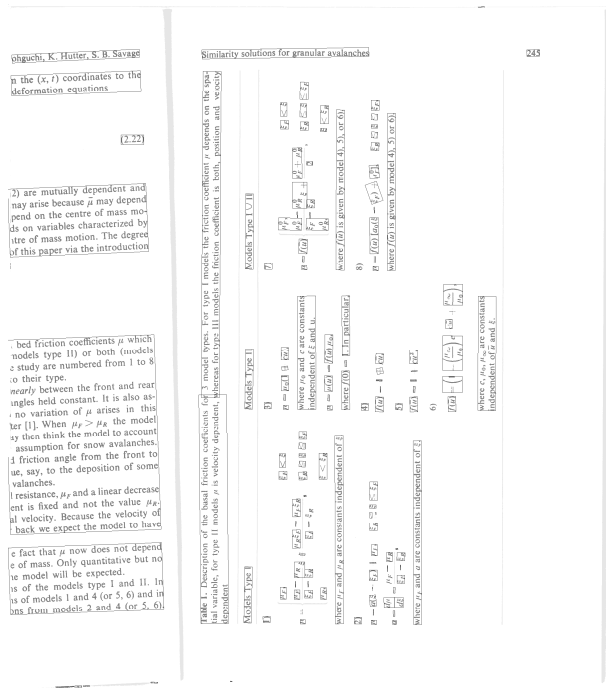


Figure 3. The grouping result of the image A002 from UWDB-I at $k = 27$. Note that the two facing pages have different skew angles, which can be observed from the best-fit bounds.

3.2 A text/graphics separator

Another application is the text/graphics separation. Normally, this is achieved by using filters based on various properties, such as size, aspect ratio of the bounding-box, density, etc. Figure 4 is a newspaper clip that have mixed text and graphics. With the use of the grouping algorithm, the major text and graphics are separated. Unlike the filter approach, the objects are also grouped. This may provides important hints for the processing stages that follow. Figure 4 also shows that the grouping algorithm can be applied to multi-lingual documents. In languages which have discrete alphabets or the printed text are not touched, the grouping algorithm works as usual, but maybe with different scale of k .

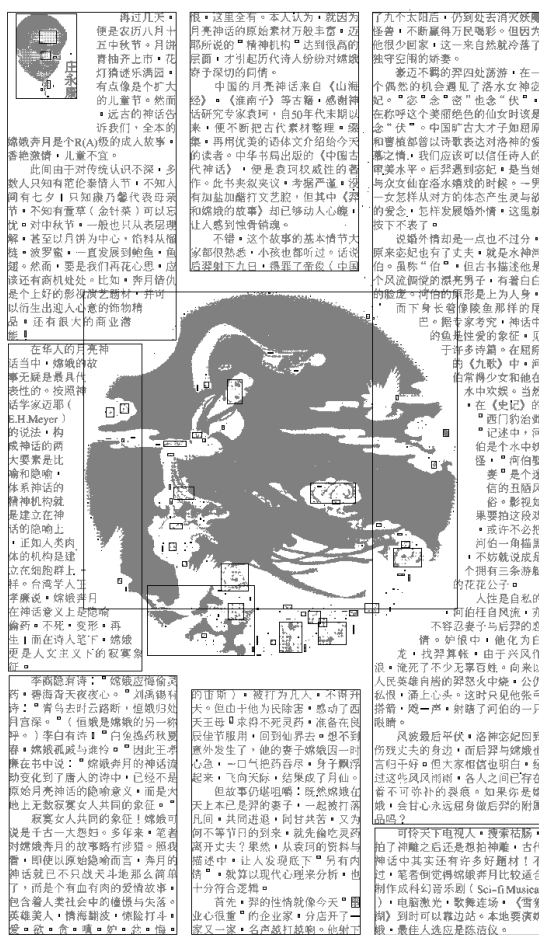


Figure 4. The grouping result of a Chinese newspaper clip at $k = 22$.

3.3 A basis for skew detection

Skew detection and page segmentation usually are the two interdependent document processing stages. A page segmentor may need the skew information of a document in order to work, while a skew detector may need the document to be properly segmented before it can detect the skews of the individual segments.

The grouping function proposed in this paper is skew independent. It is also able to produce higher-level groups that correspond to columns or paragraphs. Therefore, it forms a good basis for a skew detector to work on.

One of such a skew detector is shown in Figure 5 and Figure 6 that makes use of the results of the grouping function. It fits the identified component groups with the best bounds (bounding-boxes), and derives the skew angles of the component groups from the orientations of their bounds.

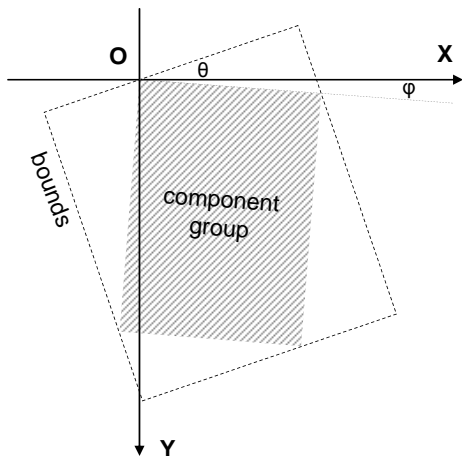


Figure 5. The principle of the proposed skew detector – finding the best bounds.

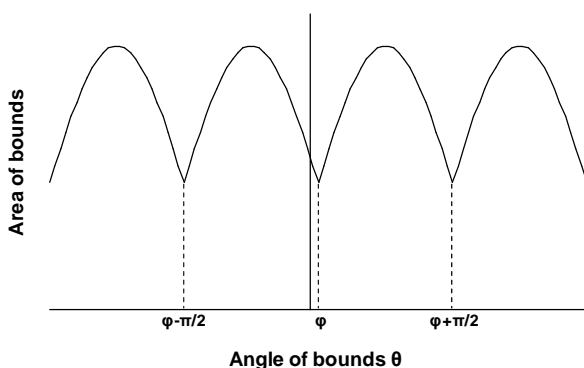


Figure 6. The determination of the skew angle ϕ .

The bounds drawn in Figure 2, Figure 3 and Figure 4 are all detected by this skew detector. In fact, this skew detector has been tested using the real images from UWDB-I. The results indicate that this detector is quite accurate and capable of dealing with samples of multiple skew, heavy noises, sparse text and some other difficulties.

4. Conclusions

This paper presents a effective and fast grouping function for applications that need a grouping method to improve their performance or robustness, where the full-fledged segmentation functions are either too complex to use or overkill for their specific purposes.

It should be pointed out that the grouping algorithm cannot match, and does not intend to have, the ability of accurate functional layout analysis that any capable full-fledged segmentation algorithm should provide.

Acknowledgment

This research is supported in part by A*STAR under grant R252-000-206-305 and NUS URC under grant R252-000-202-112.

References

- [1] L. O’Gorman, “The Document Spectrum for Page Layout Analysis”, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 15, No. 11, November, 1993, pp. 1162-1173.
- [2] L. A. Fletcher, and R. Kasturi, “A Robust Algorithm for Text String Separation from Mixed Text/Graphics Images”, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 10, No. 6, November, 1988, pp. 910-918.
- [3] A. K. Jain, and B. Yu, “Document Representation and Its Application to Page Decomposition”, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 20, No. 3, March, 1998, pp. 249-308.
- [4] M. Krishnamoorthy, G. Nagy, S. Seth, and M. Viswanathan, “Syntactic Segmentation and Labeling of Digitized Pages from Technical Journals”, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 15, No. 7, March, 1997, pp. 737-747.
- [5] B. Yuan, and C.L. Tan, “Skewscape: The Document Image Skew Detector”, *Seventh International Conference on Document Analysis and Recognition*, Edinburgh, Scotland, 3 - 6 August 2003, pp. 49-53.