Logo Detection in Document Images Based on Boundary Extension of Feature Rectangles

Hongye Wang
Room F404, Graduate School at Shenzhen, Tsinghua University, Shenzhen, 518055, China
hy-wang07@mails.tsinghua.edu.cn

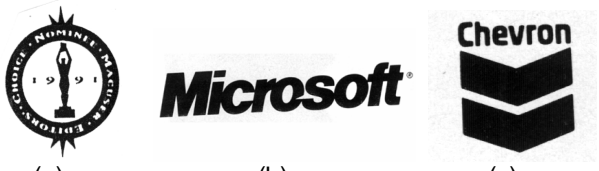
Youbin Chen
chenyb@sz.tsinghua.edu.cn

Abstract

A new method of logo detection in document images is proposed in this paper. It is based on the boundary extension of feature rectangles of which the definition is also given in this paper. This novel method takes advantage of a layout assumption that logos have background (white spaces) surrounding it in a document. Compared with other logo detection methods, this new method has the advantage that it is independent on logo shapes and very fast. After the logo candidates are detected, a simple decision tree is used to reduce the false positive from the logo candidate pool. We have tested our method on a public image database involving logos. Experiments show that our method is more precise and robust than the previous methods and is well qualified as an effective assistance in document retrieval.

1. Introduction

Logo is a key visual feature for readers to distinguish the origin or ownership of a document along with other features such as title and seal. In the applications of automatic document image processing, the main focus of logo detection is to find and extract logos with high speed and reliability. Figure 1 shows some logos from the University of Maryland logo database [1].



(a) (b) (c)

Figure 1. Examples from UMD logo database. (a) Graphic logo; (b) Text logo; (c) Mixed logo.

Most former research related to logos in document images focuses on logo recognition but not detection [3, 4, 5, 6]. S.Seiden et al. [7] used a top-down, hierarchical, X-Y tree segmentation scheme [8] to analyze a binary document. Once the segments are obtained, a set of features are generated and used in distinguishing between logos and non-logos. However, this method is not adaptive because X-Y cut does not always work well. Pham [9] developed another simple algorithm for detecting logos in grayscale document images. The detection is based on the assumption that the spatial density of a logo is greater than those of non-logo regions. However, this assumption does not always hold. G.Zhu et al. [10] proposed a method combining logo detection and extraction in a unified framework. First an initial coarse scale is chosen to reduce the resolution of the image so that the logo region becomes a grayscale blob. In this way each connected component contained in the logo is naturally merged. Then the connected components of the grayscale image with reduced size are computed. Each connected component, regarded as a sub-logo candidate region, is taken an ulterior step using a successive cascade of classifiers. However, this method also has obvious shortcomings. Since the algorithm is connected-component-based, it is very time consuming. Moreover, when the gap between a logo and other parts of a document image is small, it will merge the logo with non-logos together. All the three approaches do not use prior knowledge such as logos usually have background surrounding it and are separated from other parts of a document image with white spaces.

This paper proposes a novel logo detection method based on boundary extension of feature rectangles which will be defined in section 2. Our goals include the following:

(1) high precision and accuracy;
(2) insensitive to logo shapes;
(3) high speed and efficiency;
(4) few missing logos from the detection but keep

978-0-7695-3725-2/09 $25.00 © 2009 IEEE
DOI 10.1109/ICDAR.2009.129

the false positive rate as small as possible.

The rest of this paper is organized as follows. In section 2, the definition of feature rectangle is given. In section 3, our proposed approach of automatic logo detection is described in detail. In section 4, experimental results are given. Finally, section 5 summarizes this paper.

2. Feature Rectangles

2.1. Definition of Feature Rectangle

Definition 1. Feature Rectangle

A feature rectangle is a minimum virtual rectangle which fully embraces at least one foreground pixel (black) with four edges consisting of all background pixels (white) and has minimum inner area.

Definition 2. Edge Width of a Feature Rectangle

The edge width of a feature rectangle should be no less than one pixel as long as the edges consist of background pixels only.

2.2. Characteristics of Feature Rectangles

Theorem 1. Any foreground pixel has a unique feature rectangle surrounding it. This is called the uniqueness of feature rectangle.

Theorem 2. If a foreground pixel's feature rectangle is and another foreground pixel inside a has its feature rectangle , then ba.

Theorem 3. All foreground pixels in one connected component share the same feature rectangle.

3. The Proposed Logo Detection Method

Logos can be classified into three categories: graphic logo, text logo and mixed logo (See Figure 1). Although a logo may have text element, this text is not treated the same as the regular text in a document page. Therefore, we do not take text in logos as normal text but regard a whole logo as a picture which consists of one or several parts. Also a logo should be easily separated from other elements so as to keep its conspicuousness. Based on our observation on document images with logos, almost all documents with logos keep a relatively larger distance for the logo from other parts in the documents.

Briefly, the logo detection proposed in this paper has two steps: first, a foreground pixel is taken as a seed with a 3x3 window as a feature rectangle candidate. Then this feature rectangle candidate will grow with boundary extension until the final feature rectangle is got or the growing is stopped. All objects as a whole embraced by the generated feature rectangles in this step are a logo candidate. Second, a tree classifier is employed to quickly discard the candidates whose likelihood to be logos is very small. After this two-stage processing, most false positives are removed from the logo candidate pool, leaving only a small number of logo candidates to be verified with the logo database in further logo recognition or verification phase. Our two-stage approach exhibits superb performance on the Tobacco-800 [2] dataset which will be described in section 4.

3.1. The Layout of Document with Logos

Based on our observation on documents with logos, we have found that each logo occupies a rectangular area with white space separating it from other elements, no matter what shape the logo is. This becomes the basis of our feature-rectangle-based logo detection.

3.2. Boundary Extension of Feature Rectangle

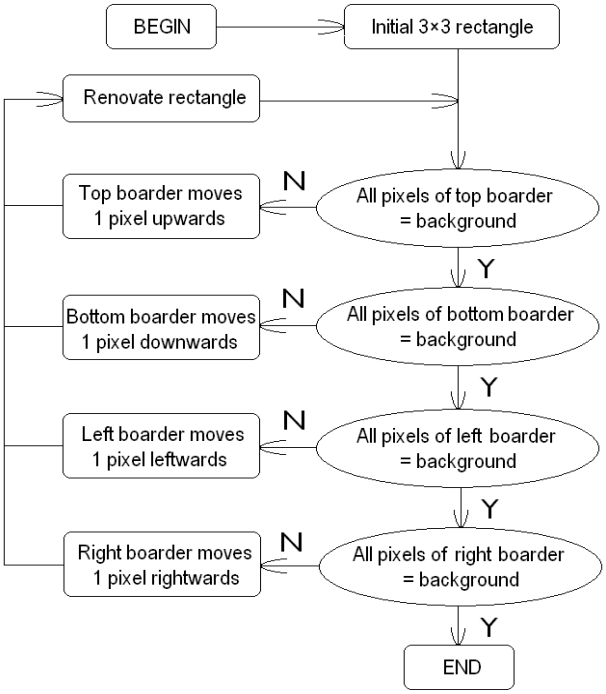


Figure 2. Workflow of our proposed method on the boundary extension of a feature rectangle.

The input image is searched from top to bottom and left to right for foreground pixels. Once a foreground pixel is detected, it is considered as a seed pixel and its 3x3 neighbor window is considered as the feature

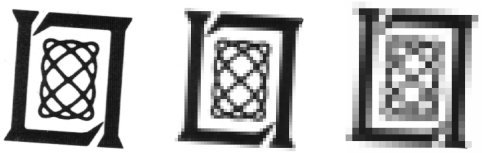
rectangle candidate of this foreground pixel. Then this feature rectangle candidate grows itself by extending its boundary (four edges) until the feature rectangle of this pixel is got or the edge growing has reached the image boundary. As defined in section 2 about feature rectangle, the four edges of a feature rectangle consist of background pixels only. So, when a feature rectangle is got, it means that a virtual rectangle embracing some objects of foreground pixels has been found. The objects embraced by this feature rectangle are considered in total as a logo candidate.

Figure 2. shows the workflow of our proposed method on the boundary extension of feature rectangle.

3.3. Detection of Logo Candidates

As stated in section 3.2 above, the whole image is searched from top to bottom and left to right for foreground pixels. Once a foreground pixel is detected, it is considered as a seed pixel and its 3x3 neighbor window is considered as the feature rectangle candidate. Once a feature rectangle is got, it is put into the logo candidate pool. After that, all foreground pixels inside this feature rectangle will be considered as "already visited". However, all foreground pixels which have not been visited will be used as seed pixels to detect feature rectangles. This process will end until all foreground pixels have been visited. Once that's done, the logo candidate pool should include all the detected logo candidates.

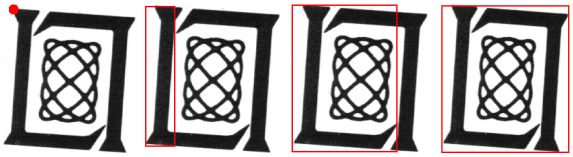
Compared with other methods on logo detection such as G.Zhu's work [10], one of the advantages of our method is that it is very fast and very simple. Figure 3 shows the logo detection process of [10] and Figure 4 shows our proposed logo detection process.



(a) (b) (c)

Figure 3. G.Zhu's merging process [10]. (a) original logo; (b) Logo merged at a small coarse scale; (c) Merging eventually completes at a large coarse scale.

Another advantage of our proposed method is that it is shape independent on logo detection. It can deal with logos consisting of isolated connected components as shown in Figure 5. This logo has three connected components: A, B and C. Even though feature rectangles of foreground pixels in components B and C cannot represent the whole logo, the feature rectangle of A can still be detected successfully and it is the feature rectangle we are looking for.



(a) (b) (c) (d)

Figure 4. Boundary extension process. (a) Seed pixel. (b)(c) Intermediate results of boundary extension of a feature rectangle. (d) Final detected feature rectangle of the seed.

(a) (b) (c) (d)

Figure 5. (a) A logo of 3 connected components. (b) Feature rectangle of connected component A. (c) Feature rectangle of connected component B. (d) Feature rectangle of connected component C.

3.4. Initial Classification of Logo and Non-Logo

As mentioned in section 1, we have four goals to reach about logo detection. So far, we have already realized goals (2), (3) and (4). Especially for the goal (4), we do not want to miss any real logo from the above step on boundary extension of feature rectangle. However, this will result in a lot of false positive candidates existing in the logo candidate pool. Therefore, our next step is a prescreening step to reduce the false positive candidates. Similar to S. Seiden's method [7] and G.Zhu's method [10], we use some prior knowledge such as the logo location, size and aspect ratio for prescreening. Figure 6 shows a simple decision tree which uses the possible logo location, appropriate logo width and height (size information) and the aspect ratio of logo feature rectangle. This is reasonable because in some real-life applications such as the logo detection in name cards and personal or business checks, the logos usually locate at the top or top-left corner of the input document images.

It's not hard to imagine that the next step for a real-life application system is to do the logo recognition or logo verification based on the size-reduced logo candidate pool and the existing known logo database which is considered as the recognition dictionary.

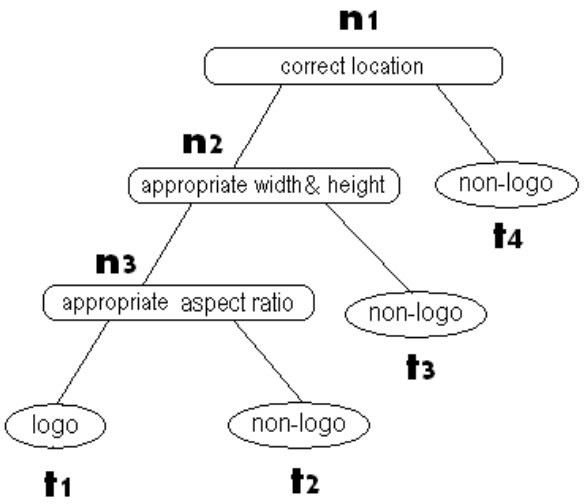


Figure 6. A simple decision tree to reduce the false positives from the logo candidate pool.

4. Experiments

4.1. Dataset

The Tobacco-800 dataset, a public subset of the IIT CDIP Test Collection [2] constructed by Illinois Institute of Technology, is used in our experiments. It is a realistic data set assembled from 42 million pages of documents (in seven million multi-page TIFF images) released by tobacco companies under the Master Settlement Agreement and originally hosted at UCSF [12]. Tobacco-800 is composed of 1290 document images collected based on a special collection building method [13]. Among them we find 416 images including logos, among which we selected 100 documents as the training set, and the rest 316 for testing.

4.2. Experiment and discussion

In accordance with G.Zhu's work [10] we define the success of logo detection as "*if and only if the detected region contains more than 75% pixels of a groundtruthed logo and less than 125% of the area of that groundtruthed logo*". For performance evaluation, we define the detection accuracy and precision metrics as:

$$\text{Accuracy} = \frac{\text{no. of correctly detected logos}}{\text{no. of logos in groundtruth}}$$

$$\text{Precision} = \frac{\text{no. of correctly detected logos}}{\text{no. of detected logos}}$$

The proposed boundary extension of feature rectangle is used for the initial logo detection. The logo candidate pool includes a lot of false positive logos. However, our goal in the first step is to guarantee there is not any missing logo. Then in the next step, a decision tree is used to eliminate the non-logos as more as possible and obviously this will reduce the false positive cases. The tree classifier was trained with a number of samples including both logos and non-logos. Figure 7 shows some logos from the Tobacco-800 database which are successfully detected by our method.

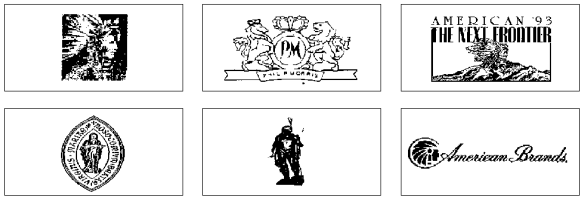


Figure 7. Examples of detected logos from the Tobacco-800 dataset.

Sometimes our proposed method has difficulties in detecting X-Y separated logos as shown in Figure 8 below. This is mainly because this kind of logos can not be simply embraced by one feature rectangle. However, we can use the concept of edge width of feature rectangle as defined in section 2. As long as the internal space a between two separated components of a logo (shown in Figure 8) is less than the external space b between a logo and non-logo, we can enlarge the edge width of feature rectangle to be slightly greater than a. In this way, the internal space will be ignored during the boundary extension of feature rectangle detection.

Figure 8. The X-Y separated logo "American Brands One" above a text line.

Our logo detection approach with a simple decision tree classifier performs superbly in both accuracy and precision as shown in Table 2. We trained each node of the decision tree with 50 logos and the same number of non-logos. As shown in Table 1, the decision rules are relatively loose as our main objective is to reduce false negatives. If we get more than one logo candidate, we only select the candidate with the maximum spatial density in order to increase the precision.

It is more likely to get false positives when the edge width becomes larger. However a proper edge width

can effectively eliminate a lot of X-Y separate logos as false negatives. By fully executing our strategy, we gain 80.4% on accuracy and 93.3% on precision in logo detection. This has demonstrated the success of our proposed method.

Table 1. Decision rules of each node of the tree classifier derived from training samples.

Node #	Feature	Value range
1st node	$\frac{\text{y coordinate of center}}{\text{image height}}$	[0, 0.19]
2nd node	$\frac{\text{rectangle width}}{\text{image width}}$	[0.041, 0.38]
	$\frac{\text{rectangle height}}{\text{image height}}$	[0.034, 0.20]
3rd node	aspect ratio	[0.48, 4.55]

Table 2. Algorithm performance based on different edge width of feature rectangle.

Edge width of feature rectangle	Accuracy	Precision
1 pixel	76.6%	91.9%
Image height / 500	80.4%	93.3%

(a) (b) (c) (d)

Figure 9. Inaccurate detection results because of (a) the gap between the logo and the text line is too narrow; (b) Adhesion of logo and other elements; (c) Low image quality with much noise.

Figure 9 shows some inaccurate logo detection results in which logos and non-logos are combined together as the output of logo candidates. Figure 9 (a) and (b) show that the text lines which are too close to logos or sticky to logos are detected as a part of the logos. Figure 9 (c) shows that noise is detected as part of the logo because of bad image quality.

For next step all detected logo candidates will be matched with logo database so as to reduce the false positive. Obviously, this will improve the precision and get rid of the three cases shown in Figure 9.

5. Conclusion

In this paper, a novel approach of logo detection in document images is presented. A fast algorithm using boundary extension of feature rectangles performs initial detection of logo candidates. Then each logo candidate is further prescreened by a simple decision tree classifier so as to reduce the false positive. This approach achieves high success on a large collection of real-world documents.

Experiments show that the boundary extension of feature rectangles is very effective in logo detection. Compared with other methods on logo detection, our proposed method is simple and fast with high accuracy and precision.

Further research will be the logo recognition and matching with database to reduce the false positive and improve the detection precision.

References

[1] The University of Maryland (UMD) Logo Database, http://lampsrv01.umiacs.umd.edu/projdb/project.php?id=47.

[2] Tobacco800 Complex Document Image Database, http://www.umiacs.umd.edu/~zhugy/Tobacco800.html.

[3] D.Doermann, E.Rivlin, I.Weiss, Logo recognition using geometric invariants. Procee International Conference on Document Analysis and Recognition, pages 897-7, 1993.

[4] D. Doermann, E. Rivlin, and I. Weiss. Applying and differential invariants for logo recognition. Machine Vision and Application, 9(2), pages73–86, 1996.

[5] J. Neumann, H. Samet, and A. Soffer. Integration and global shape analysis for logo classification. Pattern Recognition Letters, 23(12), pages1449–1457, 2002.

[6] P. Suda, C. Bridoux, B. Kammerer, and G. Maderlechner. Logo and word matching using a registration. In Proc. Int'l Conf. Document Analysis and Recognition, pages 61–65, 1997.

[7] S. Seiden, M. Dillencourt, S. Irani, R. Borrey, and T.Murphy. Logo detection in document images. In Proc. Int'l Conf. Imaging Science, Sys, and Tech, pages 446–449, 1997.

[8] G. Nagy and S. Seth. Hieraroptically scanned documents. In Proc. Int'l Conf. Pattern Recognition, pages 347–349, 1984.

[9] T. Pham. Unconstrained logo detection in document images. Pattern Recognition, 36(12), pages 3023–3025, 2003.

[10] G.Zhu and D.Doermann. Automatic Document Logo Detection. In Conference on Document Analysis and Recognition, pages 864-868, 2007.

[11] G. Agam, S. Argamon, O. Frieder, D. Grossman, and Lewis, The IIT complex document image processing (CDIP) test collection, 2006. http://ir.iit.edu/projects/CDIP.html.

[12] The Legacy Tobacco Document Library (LTDL), University of California, 2007. http://legacy.library.ucsf.edu.

[13] D. Lewis, G. Agam, S. Argamon, O. Frieder, D. Grossman, and J. Heard, Building a test collection for complex document information processing. In Proc. Annual Int. ACM SIGIR Conference, pages 665–666, 2006.