

Language Identification in Degraded and Distorted Document Images

Shijian Lu, Chew Lim Tan, and Weihua Huang

School of Computing, National University of Singapore, 117543, Singapore
{lusj, tancl, huangwh}@comp.nus.edu.sg
WWW home page: <http://www.comp.nus.edu.sg/labs/chime/>

Abstract. This paper presents a language identification technique that differentiates Latin-based languages in degraded and distorted document images. Different from the reported methods that transform word images through a character shape coding process, our method directly captures word shapes with the local extremum points and the horizontal intersection numbers, which are both tolerant of noise, character segmentation errors, and slight skew distortions. For each language studied, a word shape template and a word frequency template are firstly constructed based on the proposed word shape coding scheme. Identification is then accomplished based on Bray Curtis or Hamming distance between the word shape code of query images and the constructed word shape and frequency templates. Experiments show the average identification rate upon eight Latin-based languages reaches over 99%. . . .

1 Introduction

With the widespread use of the document capture facilities including document scanner and digital camera, language identification from document images becomes more and more important for applications such as multilingual OCR, multilingual information retrieval, and library digitalization. Traditionally, language identification is frequently addressed in natural language processing areas where languages are differentiated based on the character-coded text [1, 2] or OCR results [3]. N-grams, which represent n adjacent text symbols, are normally utilized for language identification.

Some works have also been reported to identify languages in document images scanned through a document scanner. Unlike various scripts that hold different alphabet structures [4, 5] and texture features [6], Latin-based languages are all printed in the same set of Roman letters and so have similar texture features. As a result, they cannot be differentiated based on alphabets or texture features. Letter sequences, which are generally organized in different fixed patterns (words) for different languages, are therefore widely exploited for Latin-based language identification.

The reported language identification techniques normally begin with a character categorization process, which transforms character images to a number of categories with different codes. Character coding is normally implemented based

on the character shape characteristics including character ascender and descender and character ascent and descent. For example, the works in [5, 7, 8] propose to first group character and other text symbol images into six, ten, and thirteen categories, respectively. With the character categorization results, a set of word shape codes (WSCs) are then created based on the word segmentation results. Finally, languages are differentiated according to the WSC frequency profiles of a single word [5, 7], word pair, and word trigram [8]. Linear discriminate analysis (LDA) [5] and rule based systems [7, 8] are exploited for the final language identification. The reported identification rates reach around 90%.

Though promising identification results have been achieved, some problems still exist. The problems include: 1) Nearly all existing language identification techniques assume that character images are perfectly segmented. They cannot work well with the degraded documents that contain a large number of broken or touching character components. 2) Nearly all existing methods assume that text images are noise free and character ascent and descent can be correctly detected. Unfortunately, noise and character ascent and descent may not be differentiated correctly in lots of cases. 3) Nearly all existing methods require deskew before word shape coding. 4) Nearly all existing methods assume that document images contain a large number of words. Very few works handle language identification in text images that contain just a few word images.

In this paper, we propose a Latin-based language identification technique that is tolerant of noise, segmentation errors, document distortion, and word number problems. For distorted text images, we assume that the skew angle is within 20 degrees, which is quite reasonable for control during the capturing process. For each language studied, a word shape template and a word frequency template are firstly constructed through a WSC training process. Word shape coding is carried out based on the local extremum points [9] and the horizontal intersection numbers, which are both robust to noise, character segmentation errors, and the slight document distortion. Word shape vector and word frequency vector of query images are then constructed based on the same word shape coding scheme. Lastly, languages are identified based on Hamming or Bray Curtis distance between the word shape and frequency vectors of the query images and multiple trained word shape and frequency templates.

2 Word Shape Coding

The proposed language identification technique is presented in this section. In particular, we divide this section into a few subsections, which deal with the image preprocessing, feature extraction, and word shape coding respectively.

2.1 Image Preprocessing

Some preprocessing operations are required before the word shape coding. Firstly, document text must be located and segmented from the background. A number

of text detection and segmentation techniques have been reported in the literature. In this paper, we assume that document images are binarized and contain text with noise, segmentation errors, and slight skew distortion with skew angle smaller than 20 degrees.

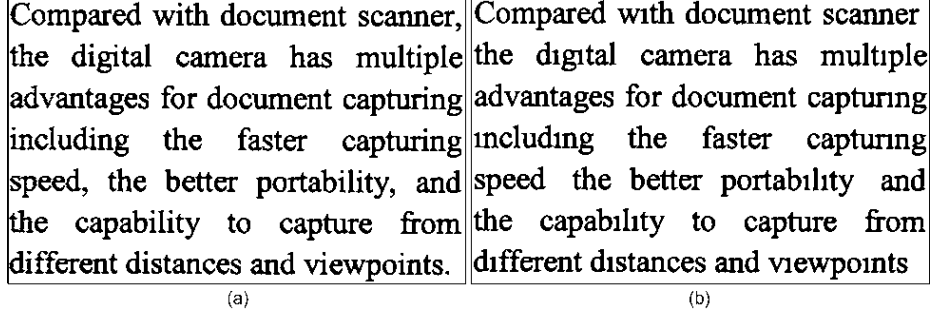


Fig. 1. (a) Binarized text image; (b) filtered text image.

Preprocessing is thus accomplished through two rounds of size filtering. Noise of small sizes is firstly removed through the first round filtering. We set threshold at 10 because nearly all labeled character components contain much more than 10 pixels. $Size_{mdn}$, the median size of the remaining character components, is then determined through a simple size sorting process. The document components with smaller size including the punctuation, the top part of characters “i”, and “j”, and character ascent and descent can be further removed through the second round size filtering. The threshold can be determined based on the $Size_{mdn}$:

$$T = k_t \cdot Size_{mdn} \quad (1)$$

where parameter k_t normally lies between 0.2-0.4. We set it at 0.3 in our implemented system. For the binarized text image given in Figure 1(a), Figure 1(b) shows the preprocessing result where small document components have been removed. In later discusses, all character components refer to the ones after these two rounds of size filtering.

Nearly all existing language identification techniques depend heavily on the small document components including the top part of character “i” and “j” and the character ascent and descent such as “é” and “ú” for character shape coding. Therefore, it is quite difficult to choose a proper threshold for noise removal. As a result, the generated WSCs normally contain lots of errors because these small document components cannot be differentiated from noise of similar size. As our proposed coding scheme does not require these small document components, the preprocessing is able to remove them together with the noise of similar sizes and generate a cleaner text image for ensuing word shape coding.

2.2 Feature Extraction

Two features are exploited for word shape coding. The first refers to the local extremum points that are extracted from upward and downward text boundary. The second is the horizontal intersection number, which counts the intersections between character strokes and the middle line of text lines.

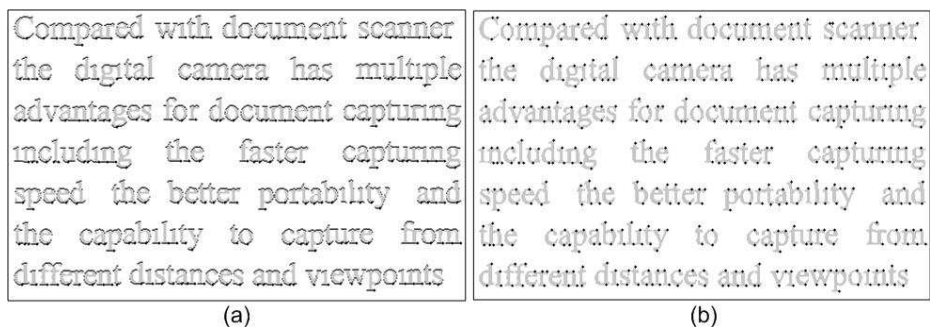


Fig. 2. (a) Upward and downward text boundary; (b) local maximum and minimum boundary points.

For each labeled character components, the upward and downward character boundary can be determined with a vertical scan line that traverses across the character component from left to right. The first and last character pixels of each scanning round, which correspond to the highest and lowest character pixels, are determined as character upward and downward boundary points. For documents with slight distortion, as we restrict the skew angle within 20 degrees, the boundary of characters ascender and descender such as “*b*” and “*p*” will not cover that of character strokes between x-line and baseline of text lines. Therefore, the extraction of the upward and downward character boundary is tolerant of slight document distortion. For the sample image given in Figure 1, Figure 2(a) shows the extracted text boundary where text is printed in light gray color to highlight the extracted text boundary.

For each labeled character component, its upward or downward boundary actually forms an arbitrary curve that can be characterized by a function $f(x)$. The extrema of the function $f(x)$, which correspond to the local extremum points, can be mathematically defined as below:

Definition : Given an arbitrary curve $f(x)$:

1. We say that $f(x)$ has a relative (or local) maximum at $x = c$ if $f(x) \leq f(c)$ for every x in some open interval around $x = c$.
2. We say that $f(x)$ has a relative (or local) minimum at $x = c$ if $f(x) \geq f(c)$ for every x in some open interval around $x = c$.

For document images scanned through a document scanner, the local extremum points normally take six boundary patterns as illustrated in Figure 3.

The downward text boundary also takes six patterns, which are actually 180 degree rotation of the six upward patterns. For the extracted text boundary shown in Figure 2(a), the black dots in Figure 2(b) show the detected extremum points. It should be clarified that some single pixel concave and convex along the text boundary may affect the extremum point detection. These concave and convex with single pixel can be removed using certain logical or morphological operators beforehand.

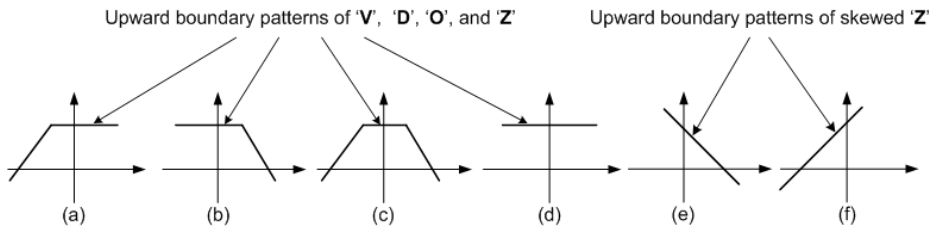


Fig. 3. (a-f): Six upward maximum patterns.

The local extremum points are tolerant of most segmentation errors and slight document distortions. Figure 4 illustrate these two properties. For word image “*language*” given in Figure 4(a), characters “*g*”, “*u*”, and “*a*” are falsely connected after the binarization process. With traditional character shape coding technique, these three characters will be treated as one and the resulting WSC will be totally different from the actual one. But the local extremum points are able to capture the word shape correctly while characters are connected as shown in Figure 4(a). Similarly, local extremum points can also be detected correctly in presence of slight skew distortion as illustrated in Figure 4(b).

In addition to the local extremum points, another feature we exploit is horizontal intersection number, which refers to the number of intersections between character strokes within the same word and the middle line of the related text line. For example, the horizontal intersection number of word image “*the*” is 4 because there are 4 intersections between the character strokes and the related middle line. Similar to the local extremum points, the horizontal intersection number is tolerant of most segmentation errors and slight skew distortions as well. For sample image “*language*” given in Figure 4, 14 horizontal intersections can be correctly counted in the presence of character segmentation errors and slight document distortion.

2.3 Word Shape Coding

With the local extremum points and horizontal intersection number, each word image can be transformed into a set of electronic WSCs.

Before the word shape coding, it is desired to extract text lines first to facilitate word segmentation and extremum point classification. We extract text

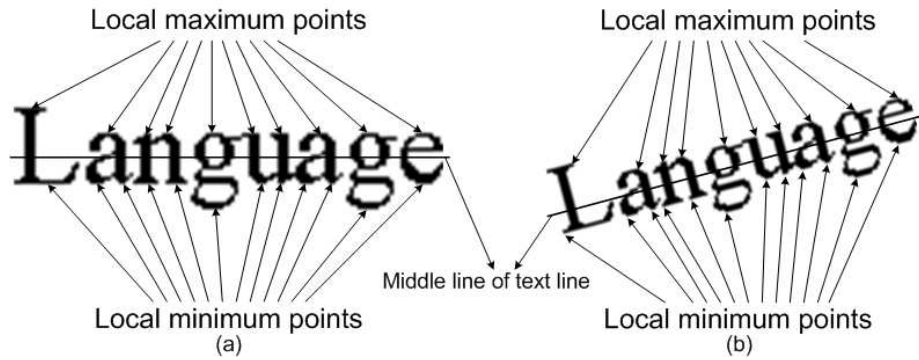


Fig. 4. local maximum and minimum points and horizontal intersection number in degraded and distorted word image.

lines using the character tracing technique proposed in [10]. With the extracted text lines, word images can be segmented based on the distance between the adjacent extremum points, as the distance between the extremum points adjacent to inter-word blank is much bigger than that between the adjacent extremum points within the same word.

Extremum points can thus be classified based on their positions relative to the x-line and baseline of text lines. In our proposed method, extremum points are classified into three categories with three different codes. The maximum points within the first category lie far above the x-line and they are coded with the number "3". The maximum or minimum points within the second category lie between the x-line and baseline and they are coded with number the "2". The minimum points within the third category lie far below the baseline and they are coded with the number "1". The x-line and baseline can be roughly fitted based on the extremum points extracted from the studied text line.

Coded local maximum and minimum	Horizontal intersection number	WSC occurrence number
---------------------------------	--------------------------------	-----------------------

Fig. 5. Word shape coding format.

Combined with the horizontal intersection numbers, word images can thus be transformed to the WSCs and Figure 5 illustrates the WSC format. The first part on the left in Figure 5 refers to the number sequence coded based on the local extremum points. Horizontal intersection number in the middle counts the number of intersections between word strokes and the middle line of text line. These first two parts form the WSC. WSC frequency on the right

denotes the occurrence number of the WSC within the studied document image. For example, the word images “*the*” and “*of*” can be coded with the number sequences “33224 m ”, “233 n ” respectively where parameters m and n refers to the occurrence number of the sample words. Number sequences “3322” and “23” are coded based on the local extremum points and the following numbers “4” and “3” record the horizontal intersection numbers.

The word images within the whole document can thus be coded based on the coding scheme as illustrated in Figure 5. Each word image is translated to a WSC and the whole document is thus transformed into a word shape vector. Each element in the word shape vector records a unique WSC number sequence. For a new WSC translated from a word image, the word shape vector is searched for the element with the same WSC. If such element exists, the occurrence number is increased by one. Otherwise, a new element is created and initialized with the new detected WSC and the occurrence number 1.

The word frequency vector is normalized to facilitate the language identification. For the i^{th} WSC element within the word shape vector, the corresponding frequency element within the frequency vector is defined as:

$$F_i = \frac{ON_i}{N_w} \quad (2)$$

where parameter N_w denotes the total number of words detected. Parameter ON_i refers to the occurrence number of the i_{th} WSC.

3 Language Identification

We use the proposed word shape codes for language identification. For each language studied, a word shape template and a word frequency template is first constructed through a training process. The language is then determined based on the distance between the word shape and frequency vectors of the query images and multiple trained word shape and frequency templates.

word shape and frequency templates are constructed through a learning process that accumulates WSCs and the related frequency from multiple training images. For each WSC translated from a word image within a specific training image, the word shape template is searched for the element with the same WSC. If the WSC pattern exists within the WSC template, its occurrence number is increased by one. Otherwise, a new element is created and initialized with the translated WSC. The training process stops automatically while the training images or the WSC patterns within the related template reaches a fixed number. With the constructed word shape template, word frequency template can accordingly be constructed based on the accumulated WSC occurrence numbers using Equation (2).

Language can thus be determined based on the distance between the word shape and frequency vectors of query images and the constructed word shape and frequency templates. For each query image, the word shape and frequency vector can be determined using the word shape coding method described in Section 2. N

frequency expectation vectors with dimension same as that of word shape vector can thus be constructed where parameter N refers to the number of languages studied. For each element within the word shape vector of query images, the same WSC pattern is searched throughout each word shape templates. If the same WSC pattern exists with the specific template, the element within the corresponding frequency expectation vector is initialized with the corresponding frequency element within the related word frequency template. Otherwise, the element of frequency expectation vector is set to zero.

As the frequency profile of the WSCs are normally quite similar for documents printed in the same language, Latin-based languages can thus be identified based on the Bray Curtis distance between the word frequency vector of the query image and N constructed frequency expectation vectors. Bray Curtis Distance given below has a nice property that its value always lies between zero and one.

$$D_i = \frac{\sum_{j=1}^n (|W E F_j - W F_j|)}{\sum_{j=1}^n (W E F_j) + \sum_{j=1}^n (W F_j)} \quad (3)$$

where parameter $W E F_j$ represents the j^{th} word expectation frequency within the i^{th} frequency expectation vector. Parameter $W F_j$ corresponds to the j^{th} element of the word frequency vector of query images. The distance D_i is the Bray Curtis Distance calculated for the i^{th} language. As a result, query image is determined to be printed in the language with the smallest D_i .

The identification performance described above may deteriorate while the number of word images becomes too small. For query images with small number of words, the WSC frequency is far different from the word frequency expectation in the related vector. Under such circumstance, the Bray Curtis Distances calculated may be quite close for different languages. We therefore propose to identify languages based on the normalized Hamming distance between word shape vector and the word shape templates while the word number is too small, say, smaller than 50.

The normalized Hamming distance is defined as:

$$H D_i = \frac{1}{N} H(W S V_{query}, W S T_i) \quad (4)$$

where parameter N is equal to the dimension of $W S V_{query}$, which denotes the number of the unique word shape patterns in the query image. $W S T_i$ represents the i^{th} WSC template. Function H count the number of WSCs in $W S V_{query}$ that do not exist in $W S T_i$. Therefore, as WSC in $W S V_{query}$ all appear in $W S T_i$, the normalized Hamming distance of the i^{th} template given in Equation (4) is 0. On the contrary, if no WSC in $W S V_{query}$ appears in $W S T_i$, the normalized Hamming distance is 1.

4 Experimental Results

800 training and testing images are prepared to evaluate the performance of our proposed method. Document texts printed in eight Latin-based languages

Table 1. WSC numbers learned from training images.

	English	German	French	Italian	Spanish	Portuguese	Swedish	Norwegian
English	6344	1332	1586	1494	1526	1404	1064	1076
German	1332	8280	1546	1448	1466	1400	1158	1230
French	1586	1546	7472	1620	1646	1552	1232	1210
Italian	1494	1448	1620	6381	1716	1708	1208	1194
Spanish	1526	1466	1646	1716	6811	1808	1278	1188
Portuguese	1404	1400	1552	1708	1808	6144	1124	1102
Swedish	1064	1158	1232	1208	1278	1124	8773	1418
Norwegian	1076	1230	1210	1194	1188	1102	1418	9147

including English, French, German, Italian, Spanish, Portuguese, Swedish, and Norwegian are tested. Four corpora of text images from different sources including books, articles, and web pages are constructed where the first one is prepared for the word shape and frequency training and the last three are for testing. Document images are scanned through a generic document scanner at different resolutions.

The first corpus contains 320 text images scanned at 400 ppi where every 40 are printed in one specific language. The corpus one is prepared for word shape and frequency training as described in Section 3. Table 1 gives the training results where the diagonal items give the numbers of the WSCs learned from the training images and the off-diagonal items give the numbers of WSCs that are shared by two related languages. As Table 1 shows, the average collision rate reaches around 10%. Therefore, the proposed technique can be exploited for language identification. Furthermore, as the trained WSCs contain a large number of short frequently appeared words, the 10% collision rate is actually much higher than the real one. It may be reduced greatly after more sample images are trained and some longer WSCs ones are collected.

The second corpus contains 160 text images with every 20 printed in one specific language. Text images in corpus two are scanned at a lower resolution (200 ppi) and so the binarized images contain more segmentation errors including broken or touching character components. At the same time, different from text images within corpus one where texts are all printed in Time New Roman, texts in corpus two are printed in several different fonts including Arial, Verdana, and Courier. Experimental results show the proposed technique is quite tolerant of text fonts and document degradation. For 160 text images studied, 159 are correctly identified with average identification rate reaching over 99%. Table 2 shows some typical Bray Curtis distances calculated using Equation (3) and four distances are listed for each language studied. As Table 2 shows, the Bray Curtis distances between word frequency vectors and the corresponding frequency expectation vectors are much smaller than those between word frequency vectors and other frequency expectation vectors.

Similar to the corpus two, the third corpus contains 160 text images as well with every 20 printed in one specific language. However, all sample images in

Table 2. Bray Curtis distances calculated for text images within corpus two.

	English	French	German	Italian	Spanish	Portuguese	Swedish	Norwegian
English1	0.3132	0.6872	0.7292	0.7844	0.7349	0.7819	0.7718	0.7398
English2	0.2438	0.5782	0.7266	0.7534	0.6922	0.8471	0.7491	0.6531
English3	0.3156	0.7275	0.6660	0.8337	0.8244	0.9040	0.6755	0.6988
English4	0.3676	0.7768	0.6720	0.8632	0.8703	0.8930	0.7405	0.7417
French1	0.7000	0.1972	0.5493	0.4485	0.3584	0.5605	0.7193	0.6596
French2	0.6843	0.1850	0.5565	0.4810	0.3626	0.5107	0.7466	0.6350
French3	0.7251	0.2098	0.6101	0.4113	0.4017	0.5594	0.7047	0.6589
French4	0.6944	0.1764	0.4932	0.4553	0.4196	0.4879	0.6553	0.6518
German1	0.6290	0.6959	0.2985	0.7418	0.8002	0.7275	0.6486	0.6763
German2	0.6821	0.7163	0.2541	0.7389	0.8274	0.7748	0.6814	0.6668
German3	0.7069	0.6823	0.2539	0.7353	0.7246	0.7700	0.6742	0.6925
German4	0.6665	0.6849	0.3458	0.7960	0.8050	0.7628	0.6067	0.5949
Italian1	0.7861	0.5149	0.6225	0.2530	0.5542	0.5505	0.7641	0.6149
Italian2	0.7431	0.3933	0.5000	0.1749	0.3943	0.5567	0.6213	0.5824
Italian3	0.6719	0.4873	0.6176	0.2541	0.4956	0.4942	0.7340	0.6464
Italian4	0.7521	0.4038	0.5816	0.2032	0.3984	0.5345	0.6213	0.5458
Spanish1	0.7365	0.3912	0.7740	0.4937	0.2594	0.4283	0.6994	0.6478
Spanish2	0.7167	0.4379	0.7062	0.5213	0.2583	0.5129	0.6658	0.6496
Spanish3	0.7861	0.4276	0.6362	0.5411	0.2282	0.4736	0.7298	0.6437
Spanish4	0.7784	0.4764	0.6873	0.6070	0.2408	0.5930	0.7258	0.7651
Portuguese1	0.7386	0.4841	0.5916	0.6663	0.4638	0.3131	0.7464	0.7562
Portuguese2	0.7693	0.4831	0.6959	0.6922	0.5414	0.1688	0.7951	0.7402
Portuguese3	0.7843	0.6242	0.6811	0.6877	0.6725	0.2458	0.8425	0.8094
Portuguese4	0.7807	0.5610	0.6346	0.7122	0.5952	0.2351	0.7695	0.7224
Swedish1	0.6685	0.6826	0.4480	0.6599	0.6689	0.7307	0.2712	0.4702
Swedish2	0.7032	0.7226	0.5339	0.7978	0.7514	0.7706	0.2617	0.5631
Swedish3	0.6785	0.6995	0.3475	0.6204	0.6389	0.7741	0.2461	0.5389
Swedish4	0.8598	0.8026	0.5334	0.7636	0.6986	0.7803	0.2840	0.5879
Norwegian1	0.7054	0.5514	0.5512	0.5310	0.5460	0.6588	0.4726	0.2134
Norwegian2	0.6220	0.5618	0.4487	0.5574	0.6345	0.6780	0.5000	0.1643
Norwegian3	0.7405	0.6091	0.4851	0.6438	0.6192	0.7750	0.4603	0.1807
Norwegian4	0.5934	0.5706	0.4547	0.5856	0.6129	0.6904	0.4200	0.1581

corpus three are coupled with slight skew distortion with skew angle controlled under 20 degree. Unlike some methods [7, 8] that require document restoration first, word images are transformed to WSCs directly based on our proposed word shape coding scheme. Experiment results show 157 text image are correctly identified with average identification rate reaching over 97%. Therefore, the proposed technique is quite tolerant of slight skew distortion.

Lastly, most reported language identification techniques [5, 7, 8] cannot identify language in document images that contain just a few words. We therefore construct the fourth corpus to evaluate the performance of our proposed technique with respect to word number. 160 text images are prepared with every 20 printed in one specific language. Test images are directly cut from the 160 testing images in corpus three and each test image contains just one or two text lines with around 20 word images on average. Languages are identified based on the normalized Hamming distance given in Equation (4). Experimental results show that 151 text line images are correctly identified with average identification rate around 94%. The lower identification rate is mainly due to the small number of WSC patterns accumulated within the constructed word shape templates. The identification rates can be improved greatly after more text images are trained and more WSCs and the related frequency information are collected.

Though the proposed technique is able to identify languages from Latin-based text images, some problems still exist. Firstly, the proposed method cannot handle text images with big skew angle. While skew angle is bigger than 20 degrees, upward and downward text boundary and so the local extremum points may not be extracted properly. Under such circumstance, document deskew is normally required before the word shape coding. Secondly, the proposed technique can only handle the Latin-based language identification. For languages typed in different scripts such as Chinese and Arabic, the collision rate is quite high and the coded WSCs are heavily affected by text fonts. We will investigate these two issues in our future work.

5 Conclusion

A Latin-based language identification technique is presented in this paper. The proposed technique is able to identify languages from degraded and distorted text images scanned through a document scanner. Language identification is accomplished through a word shape coding scheme that transforms word images into a set of electronic codes. The local extremum points and the horizontal intersection numbers are exploited for word shape coding and they are both robust to noise, segmentation errors, and slight document distortions. With coded WSCs, languages are identified based on the Hamming or Bray Curtis distance between the word shape and frequency vectors of the query images and the related word shape and frequency templates. Experiments show the proposed technique is able to identify eight Latin-based languages with average identification rate over 99%.

References

1. W. Cavnar, J. Trenkle. N-Gram Based Text Categorization, *3rd Annual Symposium on Document Analysis and Information Retrieval*, Las Vegas, NV, pages 161–175, 1994.
2. T. Dunning. Statistical Identification of Language, Technical report, Computing Research Laboratory, New Mexico State University, 1994.
3. D. S. Lee, C. R. Nohl, and H. S. Baird. Language Identification in Complex, Un-oriented, and Degraded Document Images, *International Workshop on Document Analysis Systems*, Malvern, Penn-sylvania, pages 76–98, 1996.
4. J. Hochberg, L. Kerns, P. Kelly and T. Thomas. Automatic Script Identification from Images Using Cluster-based Templates, *IEEE PAMI*, vol. 19, No. 2, pages 176–181, 1997.
5. A. L. Spitz, Determination of the Script and Language Content of Document Images, *IEEE PAMI*, vol. 19, no. 3, pages 235–245, 1997.
6. T. N Tan, Rotation Invariant Texture Features and Their Use in Automatic Script Identifica-tion, *IEEE PAMI*, vol. 20, no. 7, pages 751–756, 1998.
7. N. Nobile, S. Bergler, C. Y. Suen, S. Khoury, Language Identification of On-Line Documents Using Word Shapes, *4th ICDAR*, Ulm, Germany, pages 258–262, 1997.
8. C. Y. Suen, S. Bergler, N. Nobile, B. Waked, C. P. Nadal, and A. Bloch, Categorizing Document Images Into Script and Language Classes, *International Conference on Advances in Pattern Recognition*, Plymouth, England, pages 297–306, November 1998.
9. R. K. Powalka, N. Sherkat, R. J. Whitrow, Word Shape Analysis for a Hybrid Recognition System, *Pattern Recognition*, vol. 30, no. 3, pages 421–445, 1997.
10. S. J. Lu, B. M. Chen, C. C. Ko, Perspective Rectification of Document Images Using Fuzzy Set and Morphological Operations, *Image and Vision Computing*, vol. 23, no. 5, pages 541–553, 2005.