Determination of the Script and Language Content of Document Images

A. Lawrence Spitz, Member, IEEE

Abstract—Most document recognition work to date has been performed on English text. Because of the large overlap of the character sets found in English and major Western European languages such as French and German, some extensions of the basic English capability to those languages have taken place. However, automatic language identification prior to optical character recognition is not commonly available and adds utility to such systems.

Languages and their scripts have attributes that make it possible to determine the language of a document automatically. Detection of the values of these attributes requires the recognition of particular features of the document image and, in the case of languages using Latin-based symbols, the character syntax of the underlying language.

We have developed techniques for distinguishing which language is represented in an image of text. This work is restricted to a small but important subset of the world’s languages. The method first classifies the script into two broad classes: Han-based and Latin-based. This classification is based on the spatial relationships of features related to upward concavities in character structures. Language identification within the Han script class (Chinese, Japanese, Korean) is performed by analysis of the distribution of optical density in the text images. We handle 23 Latin-based languages using a technique based on character shape codes, a representation of Latin text that is inexpensive to compute.

Index Terms—Multilingual, script classification, machine printed OCR, language classification, Han-based languages, Latin-based languages, Asian scripts.

1 INTRODUCTION

Worldwide, there are many different languages in common use and many different scripts in which these languages are typeset [11]. In a document processing system that includes automated document recognition capabilities, early detection of the language or languages present in a document has implications both in the selection of the proper character recognition service and in the resolution of errors produced by character recognition.

The capability of recognizing multilingual documents is both novel and useful. With such capability, many potential applications can be supported including multilingual access to patent, business and regulatory information, document sorting in support of character recognition, translation, and keyword finding in document images. Dealing with multilingual documents raises many challenges including script identification, language determination, text reading direction, and differing character sets. This paper reports on algorithms that enable document recognition systems to recognize a mixture of documents in different languages, or individual documents incorporating more than a single language.

1.1 Previous Work on Script Classification

In earlier work, Spitz [18] described a method of classifying individual text lines as being either English or Japanese in documents restricted to contain only those two languages; hence the technique was really a script classifier rather than a language classifier. This technique is based on the distribution of an index of optical density and relies on the fact that Japanese contains Kanji which tend to be complex, and therefore optically dense, and Kana which tend to be simple and therefore optically light, while the Roman script that is used to set English has a more consistent optical density, character by character.

More recently Hochberg et al. have described a technique for identifying 13 scripts, including highly connected ones, where single connected components may span the entire horizontal extent of words [5]. Their algorithm has the distinct advantage of not requiring tuning to be able to accept a new script and is based upon identifying frequently occurring connected component templates which have been scale-normalized.

Using filtered pixel projection profiles, Wood et al. describe a technique that identifies which of five scripts is present based on the response to tuning the parameters of the filter [25].

1.2 Previous Work on Language Classification

Various techniques have been developed for the identification of the language (usually with the input restricted to a small set of common languages) when the input data is character-coded. Though other techniques have been used in the past, n-gram analysis appears to be the technology of choice at the current time. For example, Cavnar and Trenkle perform an n-gram analysis of ASCII text found in Usenet newsgroups [2]. Dunning likewise applies n-gram analysis to extremely short character-coded sequences [4]. Souter et al. describe corpus-based training of classifiers based on n-gram frequency [17].

Sibun and Reynar [16] survey several techniques applied to character-coded documents.
1.3 Current Work

In this paper, we concentrate on the Asian languages Chinese, Japanese, and Korean, which we will refer to as Han-based, and languages set in Roman (Latin) type (which include both European languages and non-European languages such as Swahili and Vietnamese). In Section 2, we describe some of the effects of the language of the document on the form of the document. Section 3 contains an overview of the algorithm.

We describe the preprocessing performed on page images in Section 4, including determination of text orientation, segmentation of individual text lines, registration of those lines, measurement of important line parameters, segmentation of words and segmentation of character cells. We also describe the different representations of, and fiducial points in, the image.

In Section 5 we describe the process of determining which of the two script classes is present in the document. In Section 6 we describe the process of language identification within the two basic script classes. In Section 7 we report on the accuracy of the algorithms described in Section 5 and Section 6. In Section 8 we describe the ramifications of the multilingual nature of documents on the process of optical character recognition.

2 LANGUAGE AND DOCUMENT IMAGES

The language of a document is reflected in the image of that document in at least two ways: the script, or character set and the writing conventions of the language in question. In some instances, identification of the script is sufficient to classify the document by language. For example, presence of the Hangul script unambiguously identifies the document as containing Korean; conversely, detection of the presence of Roman script is only the first step in language classification.

2.1 Script

There is a complex relationship between a language and its script.

The Han script is strongly identified with the Chinese language but is widely used in other languages such as Japanese or Korean. But both Japanese and Korean incorporate the use of other scripts as well.

Japanese uses four scripts: Kanji (Chinese), two phonetic scripts (syllabaries), and Roman. One syllabary, Hiragana, is used for native Japanese words and for inflecting Chinese and Japanese words. The other syllabary, Katakana, is used for foreign words and for emphasis. Collectively, Hiragana and Katakana are known as Kana. Fig. 1 shows the inter-mixture of scripts in Japanese text.

Katakana  
<table>
<thead>
<tr>
<th></th>
<th></th>
<th>Kanji</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(He), リチウム (Li), で, て, をを原子番号といい,</td>
<td>Hiragana</td>
</tr>
</tbody>
</table>

Fig. 1. A fragment of Japanese text showing the inter-mixture of scripts.

2.2 Text Orientation and Spacing

Chinese documents are set mono-spaced. Japanese documents are set with the native Japanese text on a rigid grid but often interpolate words or phrases in Latin type; these words are usually set in proportional type. Korean and Latin-based documents are usually set proportionally spaced, particularly since the use of typewriters and chain type line printers have given way to laser printing.

Latin-based documents incorporate horizontal text. The lines of text are read left-to-right and the lines are read in top-to-bottom sequence. Asian documents may be set either horizontally or vertically. If they are horizontal, the individual lines are read as are Latin-based documents. If vertical, the individual lines are read top-to-bottom and the sequence of lines is read from right-to-left.

2.3 Capitalization

In Latin-based documents, the first words of sentences and proper nouns and their associated adjectives are capitalized. In German, all nouns are capitalized but adjectives associated with proper nouns are not. There is no rendering in Han-based scripts that is analogous to capitalization.

2.4 Interpolations

Small amounts—isolated words or phrases—of foreign language may appear in documents of a single predominant language. In Latin-based documents, the foreign nature of these occurrences is often signaled by a typeface change, typically to italic. In Japanese texts, foreign words are indicated by the character set used: either Roman (called Romaji) or Katakana, instead of the Kanji and Hiragana used for native Japanese.

Identification of these small chunks of foreign language is a challenge to the granularity limits of the techniques described below.

3 OVERVIEW OF THE ALGORITHM

All document images are subjected to some image processing to enhance the operation of downstream processes. A list of the connected components in the image is developed and these connected components are used to establish the reference coordinate system for analysis of compressed
images and as a basis for generalized line, word and character cell (LWC) segmentation. Once this segmentation is performed, the connected components are transferred into the LWC structure.

The script is classified as being either Han- or Latin-based on the vertical position distribution of upward concavities found in the image. Language determination in Han images is based on optical density distribution. Language determination in Latin images is based on the most frequently occurring word shapes characteristic of the languages.

The flow of information in the process of script and language classification is shown in Fig. 3. Note that script classification can be performed on compressed representations, if available, or on connected components.

![Fig. 3. Schematic diagram of information flow showing image processing functions followed by gross script classification and language classification.](image)

4 **TEXT IMAGE PROCESSING**

In general, for document recognition, some image processing is required. Discussion will be limited here to that image processing specifically required to support the particular techniques of script and language identification. These techniques may also be useful outside of this application. It is presumed that text/graphics segmentation, skew detection and correction, and other image-quality related processing required for single language documents is part of the system.

4.1 Pass Codes

The CCITT Group 4 compression algorithm [3], [7] is two-dimensional in nature, that is, each color transition position on a raster line is encoded with respect to those on the previous line. When there are no related transitions on the coding line with respect to the reference line, a pass code is generated. As described in Spitz [19], it is possible to locate the position of CCITT pass codes (see Fig. 4) in the compressed image representation and to label each pass code to indicate whether white or black pixels are being processed when the pass code is encountered. Using the CCITT black pass positions takes advantage of this pre-calculated transform of the image to mark the positions of the bottoms of upward concave black structures. The positions of these points can be determined at much lower computational expense than required by connected component analysis since they can be calculated directly from the compressed document representation. Since black passes are indicative of the bottoms of white structures, they are analogous to finding the upward concavities in connected components.

![Fig. 4. CCITT Group 4 compression technique showing reference and coding lines, notations for the transition positions on those lines, and the pass mode encoding. (Adapted from [3].)](image)

The reference coordinate system used for defining the spatial distribution of CCITT pass codes is the connected component centroid position.

4.2 Connected Components

Connected components [13] are a lossless representation of the image. That is, the original bitmap can be reproduced without error from the list of connected components, including constituent pixel-run information as well as bounding box position and dimensions. The computational overhead of computing the connected components is more than compensated for by the efficiency of downstream processes which operate on the components instead of the bitmap.

The set of eight-connected components in the document image is calculated. The representation of each connected component includes the coordinates and dimensions of the bounding box and a list of the individual runs of black pixels that make up the component. With the exceptions of the skew detection process and an optional method of finding upward concavities based on the CCITT representation, described below, connected components are used as the basic image representation throughout the recognition process.

The list of connected components provides an efficient data structure on which to perform image processing operations - not only those operations related to the positions of the components in different reference frames, but also the calculation of upward concavity and optical density required for script and Han-based language identification. Except for the determination of concavities, all of the processing is based on the bounding box information.
4.3 Upward Concavity

It is trivial to examine sets of runs within the connected component to determine the presence and location of upward concavities. Where two runs of black pixels appear on a single scan line of the raster image, if there is a run on the line below that spans the distance between these two runs, an upward concavity is formed on the line (Fig. 5). The reference coordinate system used for defining the spatial distribution of concavities is the character cell baseline.

Fig. 5. Upward concavity defined by the spanning of a single run of the gap between a pair of runs on the scan line above.

Fig. 6a shows the positions of upward concavities in a single Han character. Fig. 6b shows the word Laboratory and the positions of the upward concavities as signaled by the presence of a black pass. The a show two upward concavities near the baseline, while the b shows one concavity near the baseline and one well above the baseline.

Fig. 6. (a) Locations of black passes in a single, relatively complex, Han character. (b) Locations of black passes in a word of Roman script text.

4.4 Line Parameters

Four horizontal lines define the boundaries of three significant zones on each text line (see Fig. 7). The area between the bottom and the baseline is the descender zone; the area between the baseline and the top of characters such as x is the x-zone; and the area above the x-height level is the ascender zone.

Fig. 7. A text image showing the text line parameter positions: Top, x-height, baseline, and bottom, and the zones defined by the vertical positions of these fiducial points.

Three spatial distributions are calculated: the vertical projection profile of connected components within the text line; the positions of the tops of connected components; and the positions of the bottoms of the connected components. These are shown, along with the pixel projection profile in Fig. 8.

Starting from any vertical position within the peak value of the connected component distribution (which roughly corresponds to the vertical range from baseline to x-line) we search downward for a peak in the bottom distribution and label that peak position as the baseline position. This is in contrast to Kanai's method that relies on character prototypes for baseline prediction [10]. Note that in a text line that does not contain character descenders (or punctuation such as comma or semicolon that descend below the baseline), the baseline and the line bottom positions may be the same.

Starting from the baseline, we search upward for the peak in the connected component top distribution and label this position as the x-line or x-height position. Note that in the case of a text line containing a preponderance of capital letters, digits or lower case characters with ascenders, it may not be possible to accurately determine the x-height of an isolated text line.

4.5 Text Line Processing

Techniques for the processing of images to enhance recognition processes are described in detail in [19] and [21]. Compensating for the dominant skew angle in an entire page image may not be sufficient adjustment to allow accurate text parameterization. Sometimes individual lines or small groups of lines have a skew angle relative to the orientation of the entire page. Additionally, artifacts often arise when processing photocopies of bound materials. In particular, text lines may curve toward the binding where the printed portion of the page does not contact the photocopier platen.

Registration is a process that aligns the baselines of characters on a text line. Since the actual desired baseline is difficult to characterize in the presence of descenders, we first calculate the modal bottom position for connected components. Components that do not lie at the bottom of their respective character cells (e.g., accents, i and j dots, and the upper components of ?,!;) are classified as non-baseline components. For a connected component to qualify as a non-baseline component there must be another connected component below it. Quotation marks and apostrophes are, therefore, considered to be baseline components. While this may seem counter-intuitive since the bottoms of these connected components are far above the baseline, this classification has not had any adverse effects on the performance of the algorithm. A left-to-right sequence of baseline component bottom positions constitutes a baseline profile.

A least-squares regression analysis of the bottom positions of baseline components yields a slope which is used to linearize the text baseline.
The next step may be thought of as analogous to high-pass filtering of the original baseline profile, allowing sharp discontinuities (due to adjacency of descenders and non-descenders) while eliminating the small variations in baseline position. For each baseline component, we measure and retain the relative vertical offset between the bottom of the connected component and its left baseline component neighbor. At the beginning of the line, we use the modal bottom position in place of the missing neighbor.

Now, we adjust the positions of the baseline connected components to the modal baseline which provides a perfect baseline alignment but temporarily masks the attribute of descending characters and apostrophes. Non-baseline components are moved distances equivalent to those baseline components with which they share character cells. Next we refer to the relative vertical offset information. In general, the relative offsets are distributed such that small values result from printing and scanning artifacts and font variations, but exceptional large values arise from the adjacency of descender characters to non-descenders. We take these large values and apply them to the vertical positions of the connected components associated with the descender characters, then reconstruct the textline registered to the baseline. We show an example in Fig. 9. Note that individual connected components are still rotated relative to the coordinate system.

**Fig. 9.** Distorted text line with modal baseline. Baseline-aligned and registered text line with computed baseline and x-line.

We use a local implementation of Ittner’s technique [8], [1] to find text line orientation. Ittner’s algorithm is based on the observation that inter-character spacing is, in general, smaller than inter-line spacing. He develops a minimal spanning tree connecting the center of each connected component to its nearest neighbor, and calculates the predominant direction of the connecting branches.

### 4.6 Character Cell Processing

Character cells are isolated within the spatial boundaries of each word. Vertical paths of white space divide the word rectangle into preliminary character cells that extend from the line top to the line bottom and are bounded on the left and right by inter-character spaces. These preliminary cells are later expanded to fully contain constituent components. Note that in a small number of instances where the character is horizontally disjoint, such as double quote or some Chinese:

```
那 or Hangul 에
```

characters, more than one character cell will be generated per character. So far, this has not caused problems. In Chinese or Japanese, since text is set mono-spaced, it is trivial to join adjacent cells into a single cell where aberrant spacing is initially detected. In Hangul, the inter-component space is much smaller than the inter-character space.

### 5 Script Classification

Other methods of script classification were discussed in the introduction. In this paper, we approach the problem somewhat differently; we divide the scripts into two broad classes: Latin-based and Han-based. This classification is accomplished on the basis of the vertical distribution of upward concavities. The position of these concavities may in turn be derived from examination of the individual runs that comprise the connected component or from the position of the black pass codes referenced to connected component centroids.

The connected component-based method requires knowledge of the text line layout and orientation to provide a reference frame for the vertical distribution measurement. Character sizes are assumed to be homogeneous within a single text line. The pass code based method can be applied without any knowledge of the layout of the page, text flow direction or any restriction on the mixture of character sizes in the image.

For each connected component the positions of all associated black passes are calculated. Note that fully or partially overlapping connected component bounding boxes will result in some passes being associated with more than one connected component.

We observed that the distribution of upward concavity with respect to the corresponding fiducial point is significantly different for Han and Latin-based script [22]. Because of its more complex characters, incorporating more instances of enclosed white space, there are many more concavities per character in Han script. This difference is so obvious from visual inspection of the distributions that it was chosen as the feature on which the script classification decision is made.

Fig. 10 shows vertical profiles of upward concavity population for typical Latin-based and Han-based documents. To accommodate different type sizes, the units above and below the centroid are normalized to vertical extent of the aligned connected components in the instance of pass-based classification and to character cell height in the instance of text line-based classification. The distributions are normalized to the mean. Note that for Han-based documents the vertical distribution of these concavities is more random than that seen in Latin-based documents where the bimodal distribution reflects the high concentrations of occurrences a stroke thickness above the baseline and at the x-line. In this figure a single document sample is shown for each script.

**Fig. 10.** Spatial distributions of upward concavity vertical distance with respect to baseline position for Latin-based and Han-based documents are characteristically different.
The training set comprised 15 full page documents in each of six languages. Care was taken that each of the documents did not contain any fragments of other languages or specialized symbols such as mathematics. Fig. 11 shows the normalized distributions of the document images in the training set by language. The left column shows the Latin-based languages and the right column shows the Han-based languages. In this figure, multiple document samples are shown for each script, demonstrating the distribution characteristic of the underlying scripts and languages.

Discriminating between the two classes of distribution is relatively easy: We use a simple measure of variance. Latin-based script distributions show much greater variance than Han-based, reflecting the clustering of vertical position of upward concavities. A simple heuristically determined threshold value of variance is sufficient to provide accurate discrimination (see Section 7). The variance measure is insensitive to absolute offset between the distributions.

6 Language Classification

We deal with the problems of classifying Han-based and Latin-based languages in completely separate ways.

6.1 Han-Based Language Classification

Examples of Chinese, Japanese, and Korean text are shown in Fig. 12. Note that the Japanese characters are a mixture of relatively light characters (typically Kana) and relatively dense characters (typically Kanji). The Chinese text is composed of (predominantly relatively dense) Hanzi. The Korean text, like the Japanese, is made up of both light and dense characters, but with a significantly different distribution, as will be shown.

6.1.1 Optical Density

A function is derived that reflects the perceived optical density of the character cells as a function of reading order, that is, across each character cell within each text line under consideration.

The distribution of perceived optical density of the character cells is calculated. Within each character cell, the number of black pixels is counted by summing the lengths of the runs that comprise all of the connected components within the cell. This sum represents the mass of the character, and its value is represented across the entire horizontal span of the character cell. The inter-component and inter-character spaces, and the inter-word spaces found in Hangul, are ignored. The resulting function reflects the reading order distribution of optical density. This function is insensitive to font size but implicitly relies on a single font size within each text line.

Re-examining the Japanese text sample, digitized at 300 pixels per inch, we derive the optical density function for a fragment of that image as shown in Fig. 13. The vertical axis represents the number of black pixels in the character cell, while the horizontal axis reflects the distance (not counting spaces) along the assemblage of text lines, in units of pixels.

Normalized optical density distributions of Chinese, Japanese and Korean documents are characteristically different. Note that the optical density function for Korean documents exhibits a distinct bimodal nature, with the low density mode smaller than the high density mode. The distributions for the Japanese documents might also be characterized as bimodal, but in this instance the relative heights of the modes are reversed: the low density mode is greater than the high density one. In the Chinese document there is only one significant mode. See Fig. 14.

Classification is performed by measuring the Euclidean distance in linear discriminate analysis (LDA) space from the centroids of the clusters formed by the points derived from documents of the various languages.
In LDA, the $n$th discriminant variable is the linear combination of the original variables that maximizes the separation between groups, subject to the constraint that it must be orthogonal to all the previous variables. LDA examines a training set of data (in this case three documents each in Chinese, Japanese, and Korean, each in a different font) and uses it as a model to build a classification rule. Each document consists of a data vector that is based on the density histogram and a variable that correctly identifies its language. LDA transforms the data vector to a new coordinate space where the variables have equal variance and are uncorrelated. The mean data vector for each language is then calculated. In general, new documents are identified by converting them to the new coordinate system and assigning them to the language group with the closest mean vector. LDA operates under the assumption that all language groups have roughly equal covariance matrices. The effectiveness of this method depends on how well this assumption is satisfied.

There remains the problem of choosing the right set of variables to represent each document. The obvious choice would be to apply LDA to the density distributions directly. However, this is not ideal, since the distribution contains a large number of highly correlated variables. It is difficult to accurately estimate a large covariance matrix with a training set of limited size. A common solution to this problem is to perform a principal components decomposition on the data matrix and replace a large number of variables by a smaller number of principal components. The first $k$ principal components are linear combinations of the original variables that maximize the variance that can be explained in $k$ dimensions.

A visual examination of the data reveals a second, more efficient approach. Clearly, the density distributions of the Asian languages are characterized by the relative areas in distinct regions of the distribution. A simple lower-dimensional summary of these distributions can be obtained by using the areas under the curve in three regions: $a_1$, $a_2$, and $a_3$ roughly corresponding to the lower peak in the Korean and Japanese scripts, the null in the Korean script, and the peak in the Chinese and Japanese. We integrate the areas under the curves in these three regions (Fig. 14). This method provides an obvious three-dimensional summary of the data that is ideal for LDA. This method is likely to provide a good summary of the profile in much fewer dimensions than principal components, since font differences shift the density distribution within a language group. Principal components analysis is sensitive to this difference while the technique based on peak height is not. We apply LDA to the multivariate area data to resolve into multiple language classes. Because we are attempting to select one of three classes, two LDA variables ($V_1$, $V_2$) are needed (Fig. 15).

Classification is performed by selecting the shortest Euclidean distance to a cluster centroid in LDA space.

6.2 Latin-Based Language Classification

A number of Latin-based language classification techniques using character-coded information are described in the introduction. Identification of page images using such methods on OCR output can be considerably more difficult since recognition processes are prone to systematic error which corrupts the character-coded representation, particularly when presented with an abundance of diacritics.

Our technique for Latin-based language classification is based on the principle of finding characteristic word shapes in multilingual document images [20]. Others have developed similar features to assist in performing functions other than language classification. Tanaka and Torii describe a character position coding technique used for word-spotting in document images [24]. Schürmann et al. derive these features as a pre-classification step for OCR [14]. Hull et al. use features generated by similar word images to assist character recognition in degraded documents [6].

Nakayama and Spitz [12] developed a technique to accurately detect the language of a document (when that language was restricted to be one of English, French, or German) based on the frequency of occurrence of particular word shapes. Sibun and Spitz [15] extended this work to include more languages and to provide a mechanism for the automated selection of the optimal discriminating set of word shapes. Sibun and Reynar [16] adopt a relative entropy measure in order to use character shape coded (in addition to character coded) data for the identification of language in short texts.

6.2.1 Typography Used in This Section

The following examples use the following conventions: mono-spaced to represent input characters, boldface to represent the six character shape codes ($A$, $x$, $i$, $g$, $j$, $U$), and sans-serif to represent typographic conventions.

6.3 Character Shape Codes and Word Shape Tokens

Characterizations of the number and positions of connected components in a character cell define their coding. Thus most characters, including all of the alphabetic characters, can be readily mapped from their positions relative to the baselines and x-line to a small number of distinct codes (Table 1). Our method maps from the image to a shape-based representation. The basal representation is the character shape code (CSC) of which there are a small number. The decision tree defining the geometric relationships that
result in the CSCs is shown in Fig. 16. These shape codes are aggregated into word shape tokens (WSTs) that are delimited by white space or punctuation. An example of the transformation from character codes to CSCs is shown in Fig. 17.

### Table 1

<table>
<thead>
<tr>
<th>Character Shape Code</th>
<th>Character</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>A-Zbdfhklm0-9#$&amp;()[]@{}</td>
</tr>
<tr>
<td>x</td>
<td>acemnorsuwvxz</td>
</tr>
<tr>
<td>i</td>
<td>ìáëééèíóúñ</td>
</tr>
<tr>
<td>g</td>
<td>gpqyç</td>
</tr>
<tr>
<td>j</td>
<td>j</td>
</tr>
<tr>
<td>U</td>
<td>äüöü</td>
</tr>
</tbody>
</table>

### Table 2

<table>
<thead>
<tr>
<th>WST</th>
<th>Rank</th>
<th>Word</th>
<th>Language</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aax</td>
<td>1</td>
<td>the</td>
<td>English</td>
</tr>
<tr>
<td>xA</td>
<td>2</td>
<td>of</td>
<td>English</td>
</tr>
<tr>
<td>Ax</td>
<td>3</td>
<td>to</td>
<td>English</td>
</tr>
<tr>
<td>ix</td>
<td>4</td>
<td>is</td>
<td>English</td>
</tr>
<tr>
<td>xxA</td>
<td>5</td>
<td>and</td>
<td>English</td>
</tr>
<tr>
<td>xx</td>
<td>6</td>
<td>en</td>
<td>German</td>
</tr>
<tr>
<td>Axx</td>
<td>9</td>
<td>les</td>
<td>German</td>
</tr>
<tr>
<td>xxx</td>
<td>8</td>
<td>aux</td>
<td>German</td>
</tr>
<tr>
<td>gx</td>
<td>5</td>
<td>pas</td>
<td>German</td>
</tr>
<tr>
<td>Aix</td>
<td>4</td>
<td>die</td>
<td>German</td>
</tr>
</tbody>
</table>

### 6.3.2 Classification

Our method of language determination is a form of statistical classification: our system learns how to discriminate a set of languages; then, for any input document, the system determines to which language the document belongs.

We did the initial training by the transliteration of character-coded text corpora into CSCs. We analyzed approximately 15,000 words of each language. Our initial set of discriminable languages was English, French, and German [12]. There were two criteria for the initial selection of characteristic WSTs: relative high frequency in one language and relative low frequency in the others. For English, the choice is easy: AAx (corresponding to the or The) constitutes 7% of the WSTs in the English corpus and is quite rare in the other languages. In the German corpus, Aix (corresponding to die or Die) is selected even though it is not the most frequent WST because it is rare in English and French. While Ax is frequent in all corpora, it is overwhelmingly frequent in French (corresponding to la, le, du, etc.), where it makes up 11% of the WSTs (vs. 4% for English and 2% for German). These differences in the distribution of the characteristic WSTs in 10 documents in each of the three languages were shown to be sufficient to correctly identify the language of each document without error [12]. While it may seem fortuitous that in English AAx is virtually always a mapping of the, unique WSTs are more common in Latin-based languages than one might suppose (see Table 2).

We classified the language by applying LDA to the frequencies of occurrence of the characteristic WSTs.

### 6.3.3 Automated Determination for Many Languages

In order to expand the capability to more languages we assembled a database of 755 one-page document images in the 23 languages indicated in Table 4.

1. In retrospect, the use of LDA in this context seems to have been a mistake, or at least, an example of the use of a sub-optimal technique, since there is nothing about the occurrence of components of language (letters, words, CSCs, etc.) that can be considered to be normally distributed. Pragmatically, however, it works reasonably well.
To construct a set of discriminating features, we selected the five most frequent WSTs from each language. Because of overlap, this resulted in 24 WSTs. Some of these discriminating WSTs have a high frequency across languages; in fact, xx appears in the top five of 22 of the languages we examined and therefore is of little utility in discriminating between them. However, even when we consider 23 languages, there are eight WSTs appearing in the top five of single languages that do not appear in the top five of any others. (This does not mean, of course, that these WSTs do not appear in other languages at all, but simply that they are relatively much less frequent.) The 24 WSTs comprise the set: x, xx, xxx, xxxx, i, ix, xix, A, AxA, Ax, AxA, Axx, Axx, A, AxA, AxA, Axx, xx, Ai, Aix, g, gx, xg, xxg, jx.

Relatively large amounts of data are required for adequate training and, in general, we did not have character-coded text in most of these languages to transliterate. We therefore trained on data derived from the images themselves. In testing accuracy on any particular document, training data were derived from all documents in the set except for the document in question. As the document collection grows, and when not adding new languages, it will be possible to isolate a distinct training set.

7 Accuracy

7.1 Script Classification

The original test set for script classification contained 240 samples containing as little as two lines of text, 120 from each of the two script classes, Han and Latin. No classification errors were encountered. Moreover, this algorithm has been in continuous “production” use [23] and has never been observed to fail.

7.2 Han-Based Language Classification

Within Han-based script, separation of Hangul from Chinese or Japanese using the technique described in this paper becomes problematic at small granularity, as the optical density distributions are extracted from smaller and smaller samples. We are able to classify the language of a document image containing Han-based script with perfect accuracy on text samples as small as six lines. These multi-line samples were drawn from a corpus of 27 full page documents, nine from each language. Table 3 shows the increase in error rate as the number of lines decreases.

<table>
<thead>
<tr>
<th>Lines</th>
<th>Samples</th>
<th>Error rate (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>72</td>
<td>0.0</td>
</tr>
<tr>
<td>8</td>
<td>95</td>
<td>0.0</td>
</tr>
<tr>
<td>6</td>
<td>140</td>
<td>0.0</td>
</tr>
<tr>
<td>4</td>
<td>217</td>
<td>0.5</td>
</tr>
<tr>
<td>2</td>
<td>447</td>
<td>0.2</td>
</tr>
<tr>
<td>1</td>
<td>910</td>
<td>15.8</td>
</tr>
</tbody>
</table>

Each sample is a fragment of a document image of the specified number of text lines.

The abbreviations shown are defined in ISO 639:1988 and are used as indices in Table 5.

7.3 Latin-Based Language Classification

A statistical model of the language categorizations was built using LDA and tested by cross validation (see Table 5). Overall accuracy is better than 90%, while the accuracy for individual languages varies between 75% and 100%, with an outlier of 44% for Czech/Slovak. The likely reason that performance on the Czech and Slovak languages is so poor is that these languages do not have articles, which in the other languages comprise a small number of frequently-occurring distinctively shaped words.

<table>
<thead>
<tr>
<th>Detected Language</th>
<th>Detected Language</th>
<th>Detected Language</th>
<th>Detected Language</th>
</tr>
</thead>
<tbody>
<tr>
<td>en</td>
<td>de</td>
<td>fr</td>
<td>it</td>
</tr>
<tr>
<td>36</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>29</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>28</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>23</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>38</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>25</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>40</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>40</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>31</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>30</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>33</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>31</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>18</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>69</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Numbers on the major diagonal indicate the number of correct classifications for each language. Numbers off-diagonal show classification errors. Languages are identified using their ISO 639 abbreviation shown in Table 4.
Examination of misclassifications proves somewhat instructive as can be seen in the confusion matrix in Table 5. For example, Dutch and Afrikaans are closely related languages, and the only error in either language is the categorization of one Afrikaans document as Dutch. Among the five Romance languages—French, Italian, Spanish, Portuguese, Romanian—nine of the 10 classification errors are within that language family. For the Scandinavian language family—Danish, Norwegian, Swedish, and Icelandic—the pattern is less clear. Two Norwegian documents are classified as Icelandic, but the three other errors in that family are classifications outside of the family.

8 Optical Character Recognition

At the present time, most commercial character recognizers are specifically designed for processing English text. Most are also able to handle a repertoire of other languages set in Roman font, but only on a document-by-document basis. There is also a small number of commercial products available for the processing of Japanese. Among commercial products, the ability to process documents that contain a mixture of Roman-font languages is rare; none can process mixtures of Han-based and Latin-based languages.

Many researchers have developed character recognizers tuned to specific applications, but multilingual capability has not received much attention. Complexity as represented in the number of classes of symbols to be detected is the enemy of robustness, therefore it is desirable to restrict the size of the symbol set as much as possible for particular applications. Morphological differences in character forms can be difficult or impossible to detect, particularly in the presence of printing and scanning artifacts.

It is desirable to know the language of a document, or portion of a document, before performing OCR in order to restrict the symbol set to be recognized and to enable post-classification checking against lexica. While not meeting this desideratum, in particular applications Ittner and Baird [9] are able to process extremely small granularity language fragments such as those found in a Russian-English dictionary by combining the symbol sets and detecting the locations of language changes by noting the sequence of recognized characters. This allows language determination to take place on a character-by-character basis except where the form of the symbol is indistinguishable between the Roman and Cyrillic character sets (e.g., ABCEHIMOP-STXYaceijspxy).

9 Conclusions

We hope to extend this work by developing tools for script classification of single line, and ultimately single word, text images, but since both Japanese and Chinese are set without word spaces, it is likely that text line granularity will be the limiting case. The discrimination between Chinese and Japanese relies upon recognition of Hiragana and Katakana as separate scripts, with the presence of these Kana used as an indicator that the text is Japanese. Moreover, since Japanese and Chinese both incorporate Kanji script, it is difficult to discriminate Chinese from Japanese if the input image contains only Kanji.

There is potential for increasing the number of languages classified by the techniques described here. The original work on Roman script language classification was limited to English, French and German. It has now been extended to more than 20 languages. The process of adding languages has been automated making it easy to add a new language once enough document images constituting a training set have been scanned into a database.

Extension of the techniques applied to Latin-based script is likely to be productive, perhaps with minor modification, if also applied to Cyrillic, Greek, or Hebrew scripts. Extension of the Asian language capability is also feasible, again possibly with minor modification, to be able to classify languages such as Cambodian.

Development of techniques to recognize highly connected languages has not been initiated. Handling of Arabic, for example, depends not only on handling connectedness, but also on independence from the effects of the horizontal elasticity, called keshide, found in Arabic print.

Multilingual document recognition is a new technology. However there is increasing interest in the design of systems that can handle documents in more than a single language. Driven by the demands of increasing world trade, we can expect that multilingual capabilities will become more common in both the academic and the commercial worlds. Identification of the script and language content of documents therefore becomes a crucial technology in the automation of multilingual document processing.

Some of the algorithms described above are protected by patents. In particular U.S. Patents 5,384,864 and 5,513,304 relate to the text line characterization, 5,375,176 to the character shape coding process 5,444,797 to Han/Latin classification, 5,425,110 to Han-based language classification, and 5,377,289 to Latin-based language classification.

Acknowledgments

The author would like to thank Jean-Marie de La Beaujardière, Arlene Holloway, David Hull, Takehiro Nakayama, Masaharu Ozaki, Jeffery Reynar, and Penelope Sibun for their invaluable contributions to the work reported here. The author would also like to thank Jonathan Hull and the anonymous reviewers of this paper for suggestions which improved its quality and readability.

References


A. Lawrence Spitz received degrees in Physics from the University of Delaware and in computer science from the University of Santa Clara. He started his career doing upper atmosphere research in polar regions applying image processing and pattern recognition to optical measurements of auroral displays. During this time he was a guest scientist at the National Research Council of Canada. This was followed by six years developing on-line systems for medical diagnosis and monitoring in the Cardiology Division of the Stanford University School of Medicine. For the past 18 years he has been involved with various aspects of image processing and recognition including document scanner design, reading machines for the blind, and document image database systems. He is currently Principal Scientist at Daimler Benz Research and Technology Center. His current interests are in multilingual documents, high-speed, lexically-driven character recognition, and analysis of compressed images of documents.