Page Segmentation Using Script Identification Vectors: A First Look

Judith Hochberg, Michael Cannon, Patrick Kelly, and James White

Abstract

This paper explores the use of script identification vectors *in the analysis of multilingual document images. A script identification vector is calculated for each connected component in a document. The vector expresses the closest distance between the component and templates developed for each of thirteen scripts, including Arabic, Chinese, Cyrillic, and Roman. We calculate the first three principal components within the resulting thirteen-dimensional space for each image. By mapping these components to red, green, and blue, we can visualize the information contained in the script identification vectors.*

Our visualization of several multilingual images suggests that the script identification vectors can be used to segment images into script-specific regions as large as several paragraphs or as small as a few characters. The visualized vectors also reveal distinctions within scripts, such as font in Roman documents, and kanji *vs.* kana *in Japanese.*

Results are best for documents containing highly dissimilar scripts such as Roman and Japanese. Documents containing similar scripts, such as Roman and Cyrillic, will require further investigation.

1 Introduction

Document images in which different scripts, such as Chinese and Roman, appear on a single page pose a problem for optical character recognition (OCR) systems. Existing OCR capabilities for multi-script recognition are limited to identifying individual foreign characters, such as Roman characters within a page of Chinese. This approach is inappropriate for documents with larger numbers of possible foreign characters, e.g., Roman documents that contain Chinese characters. It also does not allow larger textual regions containing foreign scripts to be processed by more powerful OCR algorithms which incorporate contextual and linguistic information.

One possible solution to this problem is to extend the user-guided page segmentation approach currently used by OCR packages to process multilingual documents printed in a single script. That is, the user could make a separate scan of the document for each script, each time manually selecting the region(s) to be recognized. It would be preferable to develop an automatic means of segmenting multi-script document images. This could be the first step in a document processing stream for multi-script documents that includes language identification as well as content-based steps, such as machine translation, information retrieval, indexing, etc.

We hypothesized that a good starting point for such an algorithm would be the *script identification vectors* produced by our published algorithm for performing script identification on single-script documents [1]. Our current implementation encompasses thirteen scripts: Arabic, Armenian, Burmese, Chinese, Cyrillic, Devanagari, Ethiopic, Greek, Hebrew, Japanese, Korean, Roman, and Thai. It requires minimal preprocessing of the document image, and identifies the script in which a document is printed with a high degree of accuracy.

The algorithm is based on cluster analysis. Connected components from a training set of documents are clustered in order to determine the most frequent character types in each script. A representative template is chosen for each cluster. New documents are classified by comparing a subset of their connected components to the templates for each script, and choosing the script whose templates provide the best match.

As part of this process, a thirteen-element script identification vector is developed for each connected component in the test document. The first element is the Hamming distance between the component and its most similar Arabic template, the second is the distance between the component and its most similar Armenian template, and so on for all thirteen scripts currently in our system.

Each of these 'most similar' templates has an associated reliability statistic that was calculated in a second pass through our training set, as part of the training procedure described in [1]. As shown in Figure 1, reliable templates are true hallmarks of a script; unreliable templates tend to be blobs or other uninformative shapes.

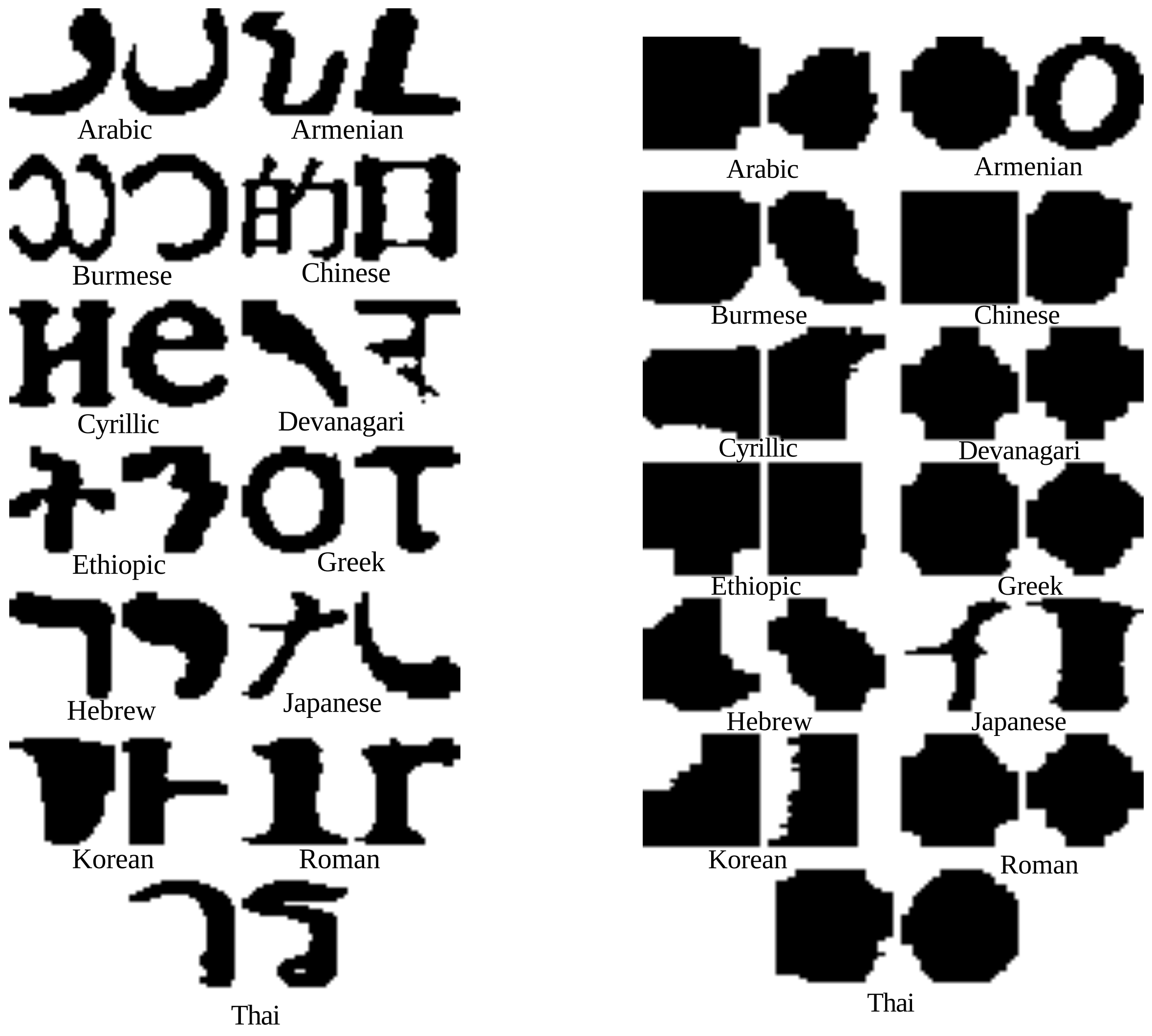


Fig. 1. Some frequent reliable (left) and unreliable (right) templates in thirteen scripts (from [1])

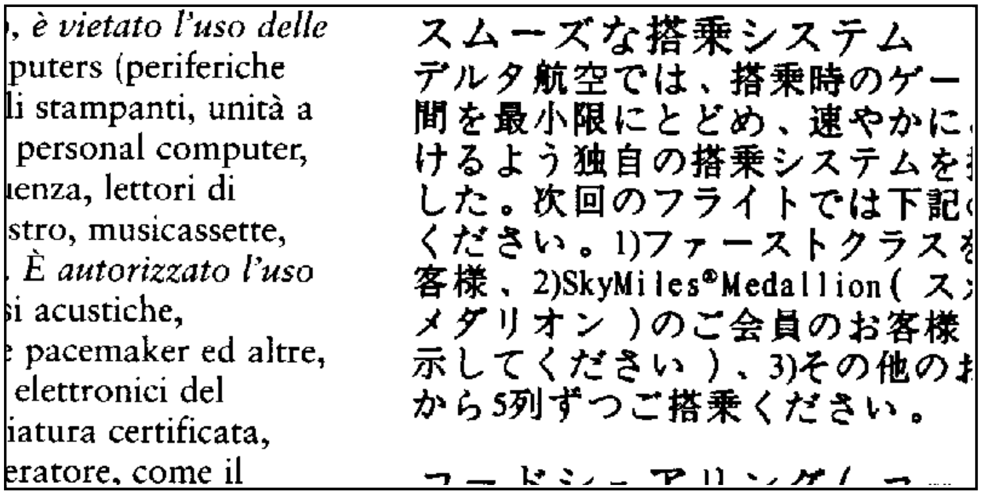
The script identification vectors were not intended to be accurate on an individual component basis. For example, a particular component from an Arabic document might have a higher matching score for Roman than for Arabic. Therefore, when using the vectors to perform script identification on a document as a whole, we considered several connected components simultaneously; our best results were obtained by examining more than 75 components per image. Combining information across components filtered out the noise and led to an accurate classification for the image. The question remained whether the information contained in the vectors was accurate enough to perform reliable script identification for textual units smaller than a page, such as a line or paragraph, or even an individual word.

Before trying to develop an algorithm to perform this task, we decided to use visualization as a tool to unpack the information inside the vectors. If the visualization showed good separation between script regions, this would be our go-ahead to develop script segmentation algorithms based on the vectors.

An eventual algorithm would use all thirteen elements from the script identification vectors. For the first look described in this paper, we reduced the vectors to three elements, using principal components analysis, so that we could easily visualize the information they contained. This is a lossy approach, yet, as it turns out, sufficient for most script combinations we analyzed.

2 Method

We collected an initial corpus of seven multi-script documents. Image sources were an airline magazine (Fig. 2), a book about manual script identification (Fig. 3, [2]), and bilingual dictionaries (e.g., Fig. 4). Script regions in these images varied in size from individual characters to entire paragraphs. All images were scanned as line art (black and white), with a resolution of 200 dpi, using an Agfa scanner equipped with StudioScan II software.

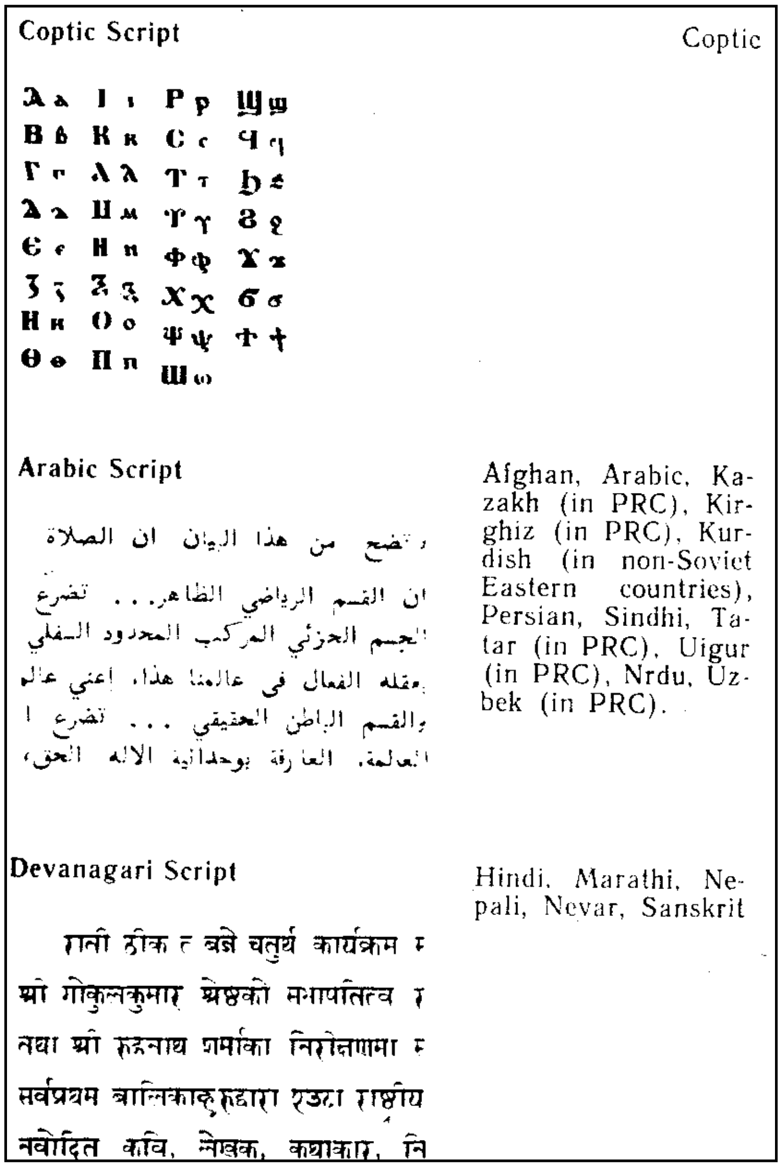


, è vietato l'uso delle
puters (periferiche
li stampanti, unità a
personal computer,
ienza, lettori di
stro, musicassette,
. *È autorizzato l'uso*
si acustiche,
pacemaker ed altre,
elettronici del
iatura certificata,
eratore, come il

スムーズな搭乗システム
デルタ航空では、搭乗時のゲー
間を最小限にとどめ、速やかに
けるよう独自の搭乗システムを
した。次回のフライトでは下記
ください。1)ファーストクラスを
客様、2)SkyMiles®Medallion(ス
メダリオン)のご会員のお客様
示してください)、3)その他のお
から5列ずつご搭乗ください。

コードシェアリング/コ

Fig. 2. Fragment of a multilingual image from an airline magazine



Coptic Script — Coptic

Ⲁⲁ	Ⲓⲓ	Ⲣⲣ	Ϣϣ
Ⲃⲃ	Ⲕⲕ	Ⲥⲥ	Ϥϥ
Ⲅⲅ	Ⲗⲗ	Ⲧⲧ	Ϧϧ
Ⲇⲇ	Ⲙⲙ	Ⲩⲩ	Ϩϩ
Ⲉⲉ	Ⲛⲛ	Ⲫⲫ	Ϫϫ
Ⲍⲍ	Ⲝⲝ	Ⲭⲭ	Ϭϭ
Ⲏⲏ	Ⲟⲟ	Ⲯⲯ	Ϯϯ
Ⲑⲑ	Ⲡⲡ	Ⲱⲱ	

Arabic Script

يتضح من هذا البيان ان الصلاة
ان القسم الرياضي الظاهر. . . تضرع
الجسم الجزئي المركب المحدود السفلي
بعقله الفعال في عالمنا هذا، اعني عالم
والقسم الباطن الحقيقي . . . تضرع ا
العالمة. العارفة بوحدانية الاله الحق،

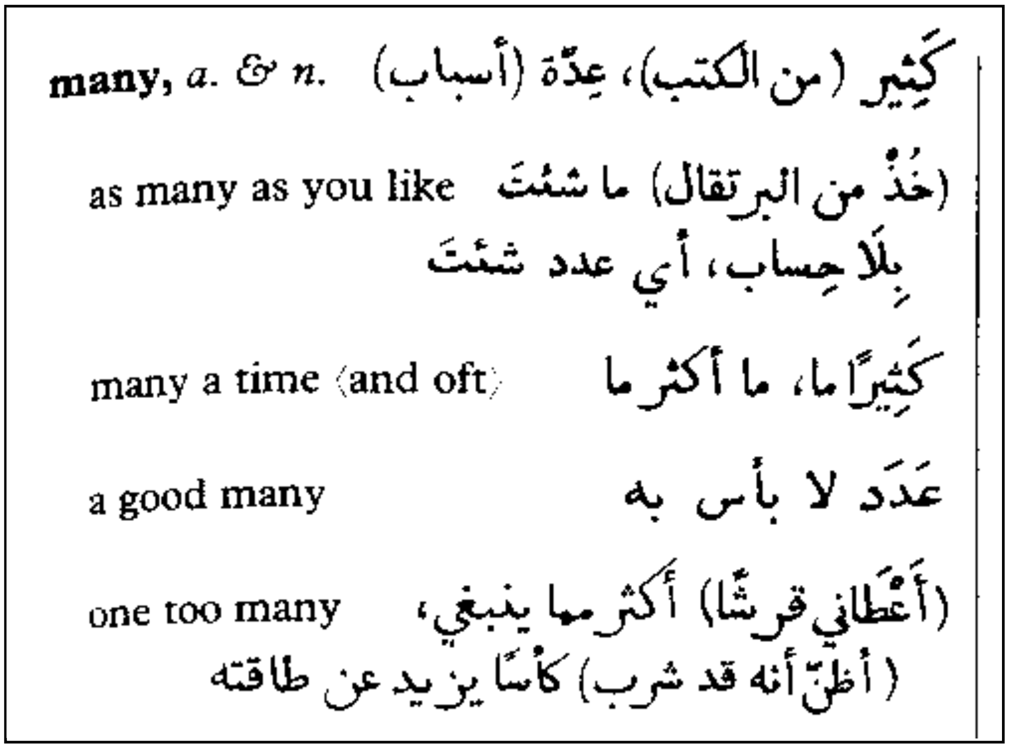
Afghan, Arabic, Kazakh (in PRC), Kirghiz (in PRC), Kurdish (in non-Soviet Eastern countries), Persian, Sindhi, Tatar (in PRC), Uigur (in PRC), Nrdu, Uzbek (in PRC).

Devanagari Script

राती ठीक ८ बजे चतुर्थ कार्यक्रम
श्री गोकुलकुमार श्रेष्ठको सभापतित्व
तथा श्री रुद्रनाथ शर्माको निर्देशनमा
सर्वप्रथम बालिकाहरुद्वारा एउटा राष्ट्रिय
नवोदित कवि, लेखक, कथाकार, नि

Hindi, Marathi, Nepali, Nevar, Sanskrit

Fig. 3. Fragment of a multilingual image from a script identification manual [2]


many, *a. & n.* كَثِير (من الكتب)، عِدّة (أسباب)

as many as you like (خُذْ من البرتقال) ما شئْتَ بِلا حِساب، أي عدد شئتَ

many a time (and oft) كَثِيرًا ما، ما أكثر ما

a good many عَدَد لا بأس به

one too many (أَعْطاني قرشًا) أكثر مما ينبغي، (أظنّ أنه قد شرب) كأسًا يزيد عن طاقته

Fig. 4. Fragment of a multilingual image from an English-Arabic dictionary

The scripts represented in each image are presented in Table 1. Most of the scripts in these documents were among the thirteen included in the script identification templates. The exceptions were all found in images 4-5, from the script identification book [2].

Table 1: Sources and scripts in test images

Image #	Source	Scripts
1	dictionary	Arabic, Roman
2	airline magazine	Japanese, Roman
3	dictionary	Korean, Roman
4	script identification book	Roman, Coptic, Arabic, Devanagari, Armenian, Ethiopic, Bengali
5	script identification book	Roman, Chinese, Nasi pictorial script, Hebrew, Javanese, Avestan, Mayan hieroglyphics
6	dictionary	Greek, Roman
7	dictionary	Cyrillic, Roman

The first step in processing each image was to develop a script identification vector, using the approach described in [1]. This involved the following steps:

- identifying all connected components in the image that contained more than ten pixels and were less than 80 pixels in height;
- rescaling components to 30 x 30 pixels;
- comparing each component to the script identification templates for the thirteen scripts in order to find the template within each script that gave the closest match (judged by Hamming distance). This step created a script identification vector for each connected component consisting of thirteen Hamming distances.
- Finally, looking up the associated reliability statistic for each chosen template.

For each image, after creating the script ID vectors as described above, we performed a principal components analysis. This identified the main axes in the thirteen-dimensional space defined by the image's vectors. Reliability statistics were not taken into account in this analysis, although we intend to use them in the future.

We discarded all but the first three principal components, and normalized the values on each of the three to a range of 0 to 255. We then mapped the normalized values onto the colors red, green, and blue. Thus each connected component in the image was assigned a color that symbolized its location in a reduced, three-dimensional script identification space.

This process is demonstrated in Figures 5a-c, for the fragment of image 1 seen in Figure 4. The three figures illustrate the first three principal components in grayscale, with intensity indicating the value of the component. The second principal component (Fig. 5b) showed the sharpest separation between Arabic and Roman, with Arabic characters generally darker than Roman ones. The first and third components also showed some separation. For the first principal component (Fig. 5a), Arabic characters tended to be lighter than Roman ones. For the third principal component (Fig. 5c), most Roman characters had medium values, with Arabic characters darker or lighter.

The color image produced by mapping Figs. 5a-c to red, green, and blue, respectively, had Roman characters in green, shading to green-blue, and Arabic characters in red, purples, blues, and blacks. The touching characters *man* in the word *many* in the second row, which were unusually dark in the second component and unusually light in the third component, were bluer than the rest of the Roman in the color image.

This image, and the other six images described in this paper, can be viewed in color on the Internet at [3].

3 Script segmentation

We evaluated each image according to the degree of observed color separation between scripts. Images 1-3 had the sharpest separation. These images each contained Roman, plus a second script that was visually dissimilar to Roman: Arabic, Japanese, or Korean. In these images, each script was mapped to a distinct color or range of colors, indicating that the script identification vectors held the information required for script segmentation. Image 1 was described in the previous section. In image 2, Roman characters were in shades of red, purple, and red-blue, while Japanese characters were in greens and blues. In image 3, Roman characters were in greens and greenish browns, while Korean characters were in purple and purplish browns (recall that color images are available at [3]).

Within this set, images 1 and 2 had the sharpest script separation. In these images, individual words and phrases in one script stood out against a background from a contrasting script. Good examples of this were "SkyMiles® Medallion", near the top of the second column of image 2 (Fig. 2), and "dead march", near the bottom of the second column of image 1.

In images 1 and 2, the visualization also revealed differences within scripts. This happened most dramatically in image 2. In the Roman areas in this image, italicized characters were bluer, and bold characters pinker, than the other Roman characters. In the Japanese areas, *kanji* (Chinese root characters) were blue or blue-green, while *kana* (phonetic characters) were green or blue-green. In image 1, italicized Roman tended to be bluer than non-italicized Roman.

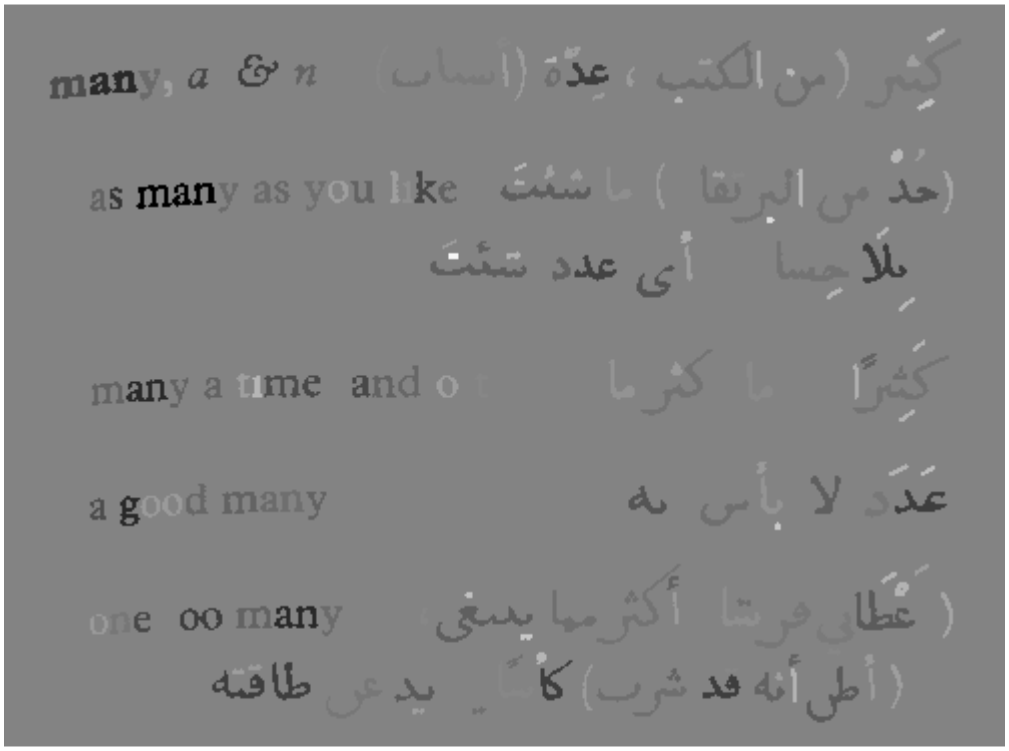


Figure 5a. First principal component for a fragment of image 1

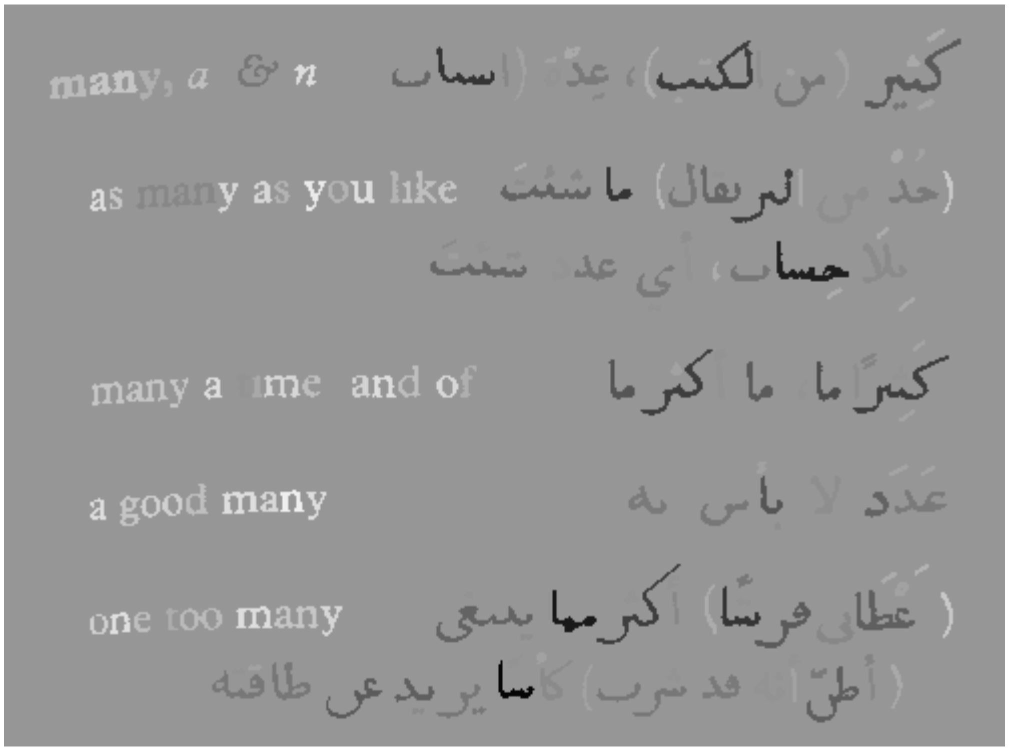


Figure 5b. Second principal component for a fragment of image 1

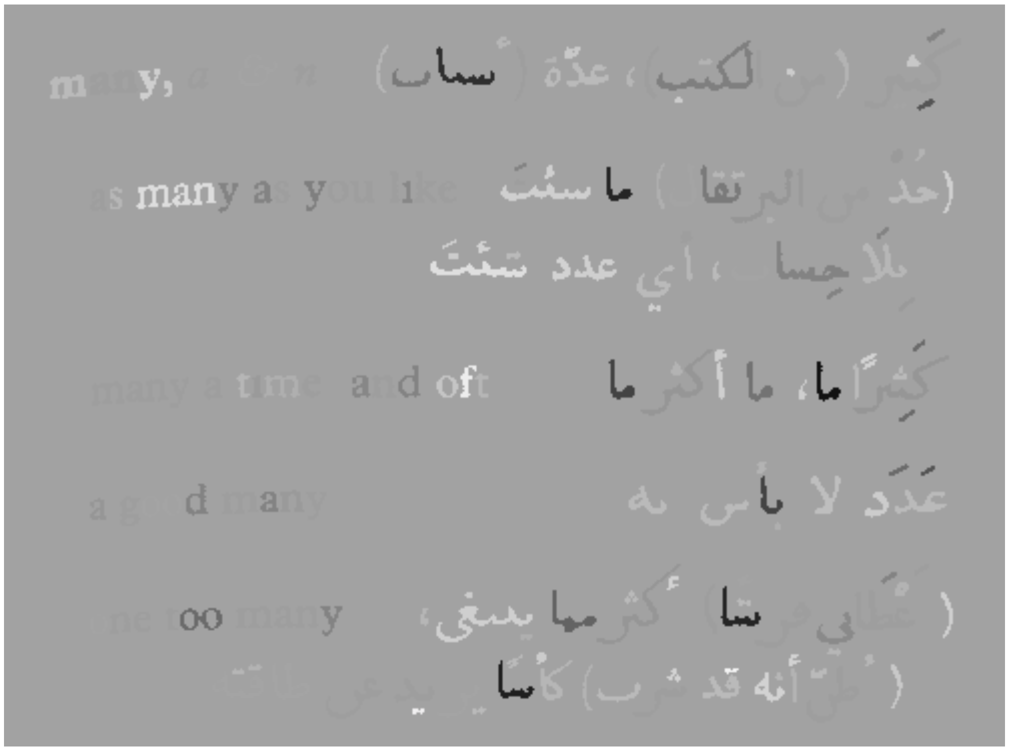


Figure 5c. Third principal component for a fragment of image 1

Images 4-5, which contained several scripts each, also showed clean script separation. The tendency in these images was for familiar scripts to be mapped to a single color, or narrow range of colors, while unfamiliar scripts were mapped to a mélange of colors. Thus in image 4, Roman was mapped to green, Arabic to blue/red, Armenian to brown, and Devanagari to dark blue. In image 5, Roman was mapped to dark reds and purples, Hebrew to green, and Chinese to blue, shading to brownish green. Coptic, Nasi, Javanese, Avestan, and Mayan, all unfamiliar scripts, were mapped to a mélange of colors.

Two scripts in images 4 and 5 differed from this general pattern. In image 4, Bengali was mapped to the same dark blue as Devanagari. This was a pleasing result given the visual similarity of the two scripts. Also in image 4, Ethiopic was mapped to a mélange of colors. This was surprising because Ethiopic was among our set of known scripts.

Images 6 and 7, which combined Roman with Greek or Cyrillic (both of which share several letters with Roman), showed the least script separation. In image 7, bright greens were generally Roman, and bright blues generally Greek; no other systematic differences were observable, and the overall effect was a mélange. In image 8, the only systematic separation was that bright greens were generally Roman. Most observable patterning pertained to individual characters: Cyrillic 'T' was always pale blue-green, and Cyrillic and Roman 'o' always red. Again, the overall effect was a mélange.

4 Conclusion

Our visualization efforts were encouraging. They suggested that, for all but closely related scripts, the script identification vectors, even as reduced by principal components analysis, contained the information needed to distinguish script regions on an individual page. For our best pairings, English/Arabic and English/Japanese, the vectors apparently also contained distinguishing font information.

Given this result, we expect that automatic segmentation algorithms based on the vectors will be fruitful in most cases. For more difficult script combinations, such as Roman/Greek and Roman/Cyrillic, we have two possibilities in mind for improving our current results. First, we plan to incorporate the reliability scores described in sections 1 and 2. These scores were crucial in making fine distinctions in our earlier work, and should help in the segmentation problem as well. The reliability scores could be used in visualization, by reducing the visual intensity of components whose best match overall was to a template with low reliability. Alternatively, each Hamming distance could be weighted by the reliability of the relevant template prior to any vector analysis. Second, in moving from visualization to an actual page segmentation algorithm, we plan to make use of the original script identification vector instead of the principal components. Perhaps the information lost in the principal components analysis is necessary for full-scale segmentation.

Acknowledgments

This work was performed under the auspices of the United States Department of Energy, contract W-7405-ENG-36.

References

[1] J. Hochberg, P. Kelly, T. Thomas, and L. Kerns, Automatic script identification from document images using cluster-based templates, *IEEE Trans. on Pattern Anal. and Mach. Intell.* **19** (1997) 176-181.

[2] R.S. Gilyarevsky and V.S. Grivnin, *Languages Identification Guide* (Nauka, Moscow, 1970).

[3] http://www.c3.lanl.gov/~judithh/LIFI/main.shtml