Content-Based Retrieval of Historical Ottoman Documents Stored as Textual Images *

Ediz Şaykol[1], Ali Kemal Sinop[1], Uğur Güdükbay[1],
Özgür Ulusoy[1], and A. Enis Çetin[2]
[1] Department of Computer Engineering, Bilkent University,
06533 Bilkent, Ankara, Turkey
{ediz,gudukbay,oulusoy}@cs.bilkent.edu.tr,
[2] Department of Electrical and Electronics Engineering,
Bilkent University,
06533 Bilkent, Ankara, Turkey
cetin@ee.bilkent.edu.tr

June 22, 2004

Abstract

There is an accelerating demand to access the visual content of documents stored in historical and cultural archives. Availability of electronic imaging tools and effective image processing techniques makes it feasible to process the multimedia data in large databases. In this paper, a framework for content-based retrieval of historical documents in the Ottoman Empire archives is presented. The documents are stored as textual images, which are compressed by constructing a library of symbols occurring in a document, and the symbols in the original image are then replaced with pointers into the codebook to obtain a compressed representation of the image. The features in wavelet and spatial domain based on angular and distance span of shapes are used to extract the symbols. In order to make content-based retrieval in historical archives, a query is specified as a rectangular region in an input image and the same symbol-extraction process is applied to the query region. The queries are processed on the codebook of documents and the query images are identified in the resulting documents using the pointers in textual images. The querying process does not require decompression of images. The new content-based retrieval framework is also applicable to many other document archives using different scripts.

*This paper has appeared in IEEE Trans. on Image Processing, Vol. 13, No. 3, 314–325, March 2004.

1 Introduction

The amount of multimedia data captured, produced and stored is increasing rapidly with the advances in computer technology. Availability of electronic imaging tools and effective image processing techniques makes it feasible to process the multimedia data in large databases. It is now possible to access historical and cultural archives according to the visual content of documents stored in these archives.

The Ottoman Empire had lasted more than six centuries until its breakdown in World War I. There exist more than thirty independent nations today within the borders of the Ottoman Empire, which had spread over three continents. A content-based retrieval system can provide efficient access to the documents from the Ottoman Empire archives, which contain more than a hundred million handwritten files. Historians want the documents in historical archives to be stored in image form because the documents include not only text but also drawings, portraits, miniatures, signs, ink smears, etc., which might have an associated historical value. In order not to loose any details, these documents are suggested to be stored in image form. Another reason for the digitization of these documents is that the hard copies of some of the documents are deteriorating as time passes.

In this paper, a content-based retrieval system for the documents in the Ottoman Empire archives is presented. The Ottoman script is a connected script based on the Arabic alphabet. A typical word consists of compounded letters as in handwritten text. Therefore, an ordinary textual image compression scheme for documents containing isolated characters cannot encode Ottoman documents. The compression scheme used is an extended version of the textual image compression scheme developed by Ascher and Nagy [1], and Witten et al. [2]. The key idea in this special purpose image compression scheme is to extract textual data from the document image, and compress the textual data and the background separately. In our framework, the document images are compressed by constructing a library of shapes that occur in a document, and the compressed textual images contain pointers into the codebook for each occurrence of these shapes together with the coordinate information. The documents are compressed based on the extracted codebook, which is constructed by a global symbol extraction process. In this process, each new symbol is checked to find out whether it is similar to an existing symbol both as a whole or as a region within the symbol in a scale-invariant manner.

In order to have a scale-invariant symbol extraction and correlation process, three features are used: *distance span* and *angular span* based features are extracted from the spatial domain. These features also enable rotation-tolerance up to a desired angle to detect the slanted symbols. Scale-invariant features are also extracted from the extrema of the wavelet transform computed by the Adaptive Subband Decomposition (ASD).

In the content-based retrieval system, the query specification is performed by identifying a region containing a word, a phrase, or a sentence in a document image. Then, the features of the query region in wavelet and spatial domain are extracted. This process is performed by comparing the pointers to the codebook in the query image with the pointers maintained for each compressed image in the archive. The resulting documents are ranked in the decreasing order of their similarity to the query image, which is determined by the total number of symbols matched with the query region. There is no limitation on the number of lines in the query region, hence any portion of the query image can be queried successfully. The resulting documents are presented by identifying the matched region of each document in a rectangle. An important feature of the querying process is that it does not require decompression of images.

In addition to being the first complete content-based retrieval system for Ottoman archives, the system has the following innovative features:

- The system presents the resulting document images in the decreasing order of similarities with the help of splitting the query image into symbols which may or may not correspond to characters. Ordering of document images based on symbol-wise similarities yields more realistic partial matches than whole picture comparisons, because the number of matching symbols is taken into consideration.

- The document images queried do not have to be binary images, they can be gray level or color images. This makes the system capable of handling all possible documents in original forms without any possible distortions due to binarization.

- Symbols in the codebook are extracted from the document images by a scale-invariant process, which enables effective and consistent compression for all historical and cultural documents in textual image form.

- The techniques presented in this paper are not specific to the documents using the Ottoman script. They can easily be tailored to other domains of archives containing printed and handwritten documents for not only image-based document compression but also content-based retrieval.

The rest of the paper is organized as follows: Section 2 presents an overview on textual image compression schemes detailing the character extraction and matching steps. The proposed compression scheme that is applied to Ottoman documents is discussed in Section 3 and the content-based retrieval process is explained in Section 4. In order to evaluate effectiveness of the content-based retrieval process, precision and recall analysis was carried out on a sample Ottoman archive containing 102 document images from different sources. Promising results obtained for the retrieval effectiveness are presented in Section 5. Finally, we conclude the paper in Section 6.

2 Related Work on Textual Image Compression

Efficient compression of binary textual images is an important problem in document archives. A textual image typically consists of repeated patterns corresponding to mostly characters and punctuation marks. Witten et al. [2], described a textual image compression method that identifies the locations of the characters in the textual images and replaces them by pointers into a codebook of characters. The method can compress images in both lossy and lossless manner. The main steps of the lossy compression method can be summarized as follows:

1. Find and extract a character in the image.
2. Compare it with the symbol library (codebook) consisting of the separate character images.
3. If the character exists in the codebook, take the location only, otherwise add it to the library.
4. Compress the constructed codebook and the character locations.

The lossless compression is obtained by encoding the residue image produced by subtracting the compressed image from the original one [3]. Gerek et al. [4] proposed a subband domain textual image compression method based on [2]. In this method, the textual image is first decomposed into sub-images using a binary subband decomposition structure. Then, the repetitions of character images in the subband domain are encoded.

Another compression technology, called DjVu [5, 6], has been developed to allow the distribution of the scanned document images, digital documents and photographs at very high resolution leading to huge file sizes to provide the readability of the text and to preserve the quality of the images. Reducing resolution to achieve reasonable download speed during image distribution means forfeiting quality and legibility. DjVu provides a medium for content developers to scan and distribute high-resolution document images with acceptable download speeds. One of the main technologies behind DjVu is the ability to separate an image into a background layer and foreground layer. The former corresponds to the texture and pictures in the document image whereas the latter corresponds to the text and line drawings. By the help of this foreground/background separation, the text can be kept at high resolution while the background pictures can be compressed at lower resolution with a wavelet-based compression technique.

2.1 Character Extraction and Matching

A crucial step in a textual image compression scheme is the character extraction process, as the codebook is populated according to the extracted characters or symbols [7]. Extracting symbols or characters from binary images is a relatively easy task compared to extraction from gray level or color images. A typical binarization method is presented in [8] to facilitate the character extraction process, in which a gray level image is transformed into a binary image with the help of global or locally adaptive techniques.

One of the main problems in character extraction is processing characters from handwritten text or texts composed of connected scripts. In historical documents including the Ottoman documents, it may not be possible to isolate characters successfully so that one can apply a character recognition algorithm to recognize handwritten text or connected scripts [2].

Binary Template Matching is one of the candidates for calculating the similarity between characters. In [9], Tubbs presented eight different distance measures for determining character similarity, and found the best ones as the Jaccard distance (d_J) and the Yule distance (d_Y). In order to overcome the shape variance of the matching, probabilistic weights are assigned to the pixels with respect to their positions. Gader et al. [10] proposed a training-based approach in which the training phase is fuelled with a set of templates for each character class. However, providing a comprehensive set for each handwritten character is a tedious task and may not be suitable for connected scripts of various languages.

Projection histograms [11] is another approach for not only character matching but also character segmentation. The number of pixels for the horizontal and vertical projections of a character is stored in the horizontal and vertical projection histograms, respectively. These histograms can be normalized with respect to the total number of pixels of a character in order to satisfy scale invariance. However, the method is rotation variant and may not be helpful for some domains.

Local Template Matching [2, 3] uses local characteristics instead of global characteristics. The overall dissimilarity between two characters or symbols is calculated as follows: each pixel-wise error is computed and then summed up. The local template matching method is scale variant, thus, the characters have to be re-scaled to a fixed size to compare them.

3 Compression of Ottoman Documents Stored as Textual Images

It is recommended that documents in the Ottoman archives are stored as textual images in compressed form. To achieve this goal, a textual image compression scheme is proposed, which encodes both the textual information and background separately. While processing these document images, no noise filtering or neighborhood ranking methods are employed because of the fact that they might possibly damage the original forms of the document images, which have a significant historical value.

The Ottoman script is based on the Arabic alphabet. The image encoding scheme developed in [4] for Ottoman documents is a multi-pass method. The character-wise partitioning of the documents written in the Ottoman script is very difficult. This is due to the fact that there are isolated symbols corresponding to single letters as well as long connected symbols which are combination of letters in Ottoman documents. Usually, longer symbols include some letters that can also be found separately inside the document. When the smaller isolated symbols are encoded and removed from the document, longer symbols split into smaller symbols. Each of these smaller symbols may also be contained in other connected symbols. Therefore, they will later cause other connected symbols to split further. Due to this reason, it would be better to use the term "symbol" instead of "character" in Ottoman document image compression. Figure 1 presents the character extraction algorithm.

```
1. While there are isolated symbols in the document,
  1.1 find and extract an isolated symbol from the document,
  1.2 compare it with the symbol library consisting of the separate symbol images,
  1.3 if the symbol does not exist in the symbol library, insert it to the library,
  1.4 take the locations of all occurrences of the symbol and remove all
   occurrences of the symbol from the document.
2. Compress the constructed library and the symbol locations.
```

Figure 1: *Symbol extraction algorithm.*

The accuracy of this method depends on the fact that there can be enough isolated symbols to split longer connected symbols in a document which turns out to be a valid assumption in Ottoman documents. Usually, the longer symbols include some symbols that can be found separately inside the document. The symbols extracted in the first pass of the method are encoded and removed from the document, then the longer symbols split into smaller symbols for the succeeding passes of the method. The deletion of the extracted symbols from the document image is performed by sliding the symbol image throughout the document image [2, 4] (see Figure 2 for an example). The algorithm stops when all of the symbols are extracted from the document image during these multiple passes. The resulting codebook is found to be consisting of basic compound structures, curves and lines that eventually form the Ottoman script.

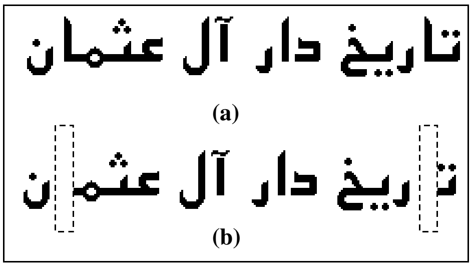


Figure 2: *Deletion of a symbol from a script image corresponding to a date: (a) image before deletion, (b) image after deletion of an L-like symbol, which appeared as an isolated symbol extracted earlier (shown with dashed rectangular regions).*

3.1 Scale-Invariant Symbol Extraction

The algorithm in [4] reviewed above is not scale-invariant. In order to make the symbol extraction process scale-invariant, aspect ratios of the symbols already in the library are considered with the help of feature vectors defined in Sections 3.2 and 3.3. Moreover, all isolated symbols in a document (i.e., symbol candidates) are extracted in the first pass. Then, the symbol-wise comparisons and deletions are carried out within these isolated symbols. These modifications make the codebook library more efficient for retrieval purposes. Figure 3 demonstrates the scale-invariant symbol extraction on a sample document. Two symbols are appearing in a sample document and the left one, which is bigger in size, completely resides in the right one.

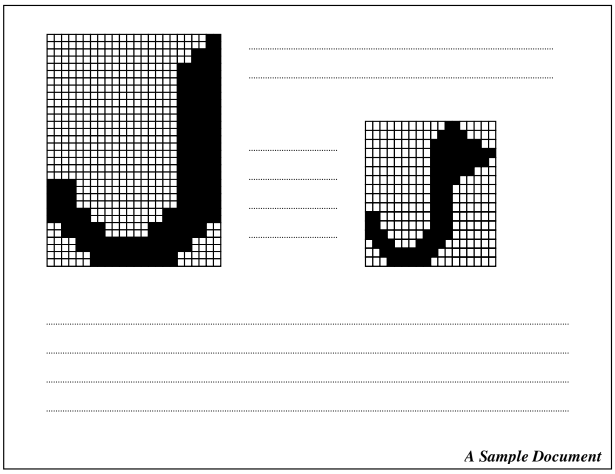


Figure 3: *Scale-invariant symbol extraction process can detect the symbol on the left within the symbol on the right, which cannot be detected by the ordinary pixel-wise sliding window symbol extraction process. The symbol on the right is split further by the symbol on the left, which was already in the codebook library.*

The new symbol extraction algorithm also solves the problem of having a new symbol occurring as a part of the previously extracted symbol. This problem is especially important for connected scripts because the symbols extracted earlier may be a combination of other symbols.

There is one codebook for each file containing writings of the same writer, or a booklet, or a book. The document compression algorithm used for the historical Ottoman documents is as follows: Let D denote the document to be processed, CCD denote the set of isolated symbols of D, CD be the compressed textual image representation of D to be used in querying and retrieval, and SL be the existing codebook at the time of processing D. Figure 4 presents

```
1. Extract all isolated symbols from D, and put them in CCD.
2. If SL is not empty, split further the items in CCD by checking each element in SL,
and augment CCD with the new symbols formed after split.
3. Pick an item S from CCD.
4. By keeping the aspect ratio of S, check the similarities within the elements of CCD.
5. For every found highly correlated region within another element S' in CCD, clip that
region from S' and encode the location of S in S' into CD.
6. Add S to SL, and remove S from CCD.
7. If CCD is not empty, go to Step 3.
```

Figure 4: *Document compression algorithm.*

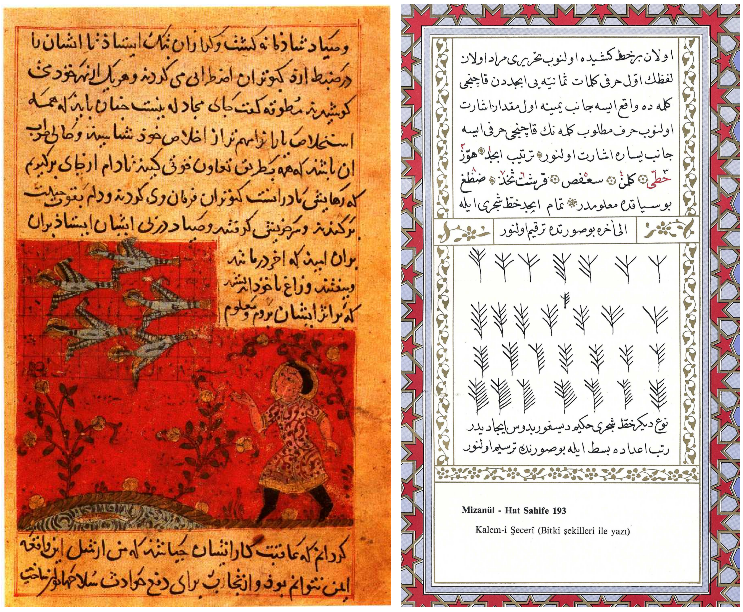
the overall algorithm. The extracted isolated symbol can be manipulated by a linked-list structure simply as well as the symbols in the codebook library. At step 3, while picking a isolated symbol from CCD, the smallest symbol heuristic can be used, which relies on the tendency that smaller sized symbols will divide other symbols more likely. The similarity check in Step 4 is performed for all possible alternatives for the width of Symbol S. The algorithm proposed here is not only applicable to binary textual images, but also symbols can be extracted from colorful images containing both drawings, figures, minor shapes, etc. Figure 5 shows example textual images, which can be processed by the algorithm successfully. An example textual image is given in Appendix A in compressed form.

The symbol comparison method used during the retrieval process has to be scale-invariant as well. Otherwise, it is impossible to detect the different occurrences of the same symbol with different sizes. As opposed to most of the shape comparison techniques, rotation invariance is not required for symbol comparison. However, rotation invariance up to a desired degree can be applied to detect minor rotations, e.g., slanted letters.

3.2 Spatial Domain Features

In order to have a scale-invariant and rotation-tolerant symbol correlation method, *distance span* and *angular span* of symbols are estimated. In this subsection, it is assumed that the symbols in the library of a document are binary. However, both features can be computed for gray scale symbols as well. These features are simple variations of the *distance* and *angle histograms* in [12, 13], in which they are used as shape descriptors of the objects for content-based retrieval. The distance and angular spans of a symbol are computed with respect to the center of mass (c_m).

Angular Span of a symbol is a vector whose entries are the number of black pixels in θ-degree slices centered at c_m with respect to the horizontal axis. The entries are normalized by the area, which is the total number black pixels, of the symbol. In Figure 6 (a), a symbol is shown. In Figure 6 (b), the angular span vector containing 8 entries is computed for $\theta = 45°$.


Mizanül - Hat Sahife 193
Kalem-i Şecerî (Bitki şekilleri ile yazı)

(a) (b)

Figure 5: *Examples of handwritten Ottoman textual images: (a) textual image containing a miniature painting, (b) a document containing plant sketches.*

Distance Span of a symbol is also a vector whose entries are the number of black pixels in between the concentric circles centered at c_m with radius r, 2r, 3r, ... The entries are normalized according to the distance of the farthest two pixels on the symbol. In Figure 6 (c), the distance span vector of the symbol shown in Figure 6 (a) is computed with four entries.

The correlation between two symbols is computed with respect to these two vectors instead of actual pixel values in both image compression and retrieval.

The rotation-tolerance can be also achieved with the help of the angular span vector. Rotating the symbols θ degrees is equivalent to circular shifting the angular span vector one slice. Therefore, the comparison of the circularly shifted angular span vectors of symbols is used to achieve rotational invariance.

3.3 Extracted Features from the Wavelet Domain

The template query image is also analyzed by Adaptive Subband Decomposition (ASD) to extract wavelet domain features. The features are scale invariant as well. In the next paragraph basic principles of Adaptive Wavelet Transform (AWT) is reviewed.

In feature extraction the so-called high-low and low-high images obtained from AWT analysis are used. The AWT structure is obtained by using adap-

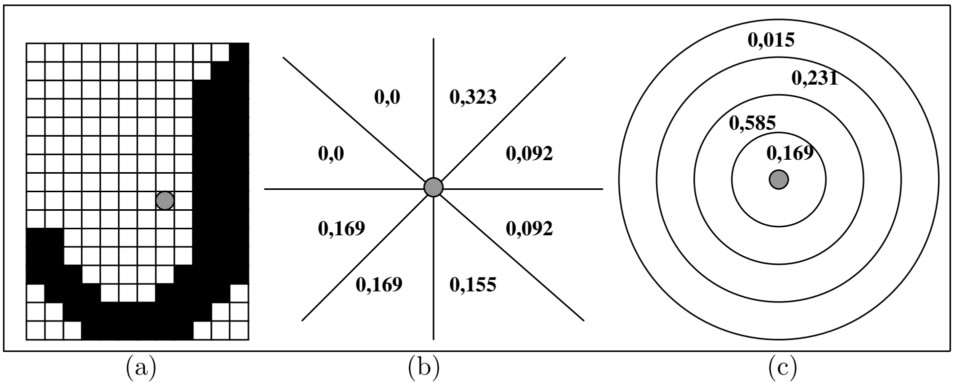


Figure 6: *(a) A symbol from the library (the gray spot is the center of mass), (b) the angular span of the symbol, for $\theta = 45$ degrees, (c) the distance span of the symbol with four entries.*

tive filters instead of fixed filters in the lifting based wavelet transform. One dimensional adaptive subband decomposition structure [14, 15, 16, 17] is shown in Figure 7. Its extension to two-dimensional (2-D) signals is straightforward using the row by row and column by column filtering methods as in ordinary 2-D separable subband decomposition (or wavelet transform). In Figure 7, the first subsignal v_1 is a downsampled version of the original signal v, a one dimensional signal which is either a column or a row of the input image. As v_1 is obtained after down-sampling, it contains only the even samples of the signal v. The sequence v_2 is a shifted and downsampled version of v, containing only odd samples of v. The subsignal v_2 is predicted from the samples of v_1 and the prediction error is the subsignal v_h which contains unpredictable regions such as edges of the original signal. Various adaptation schemes can be used for the predictor P_1 [14].

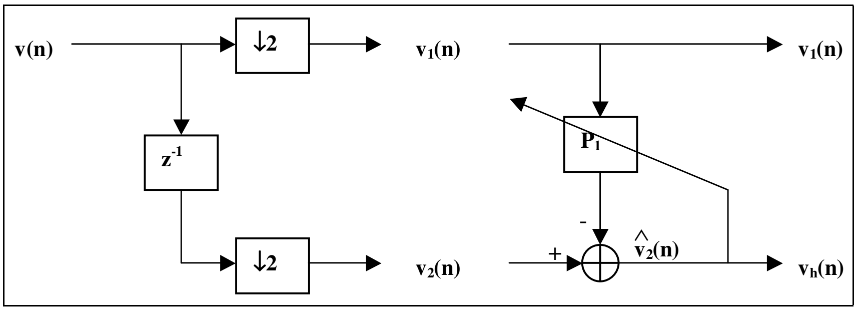


Figure 7: *One dimensional adaptive subband decomposition structure.*

In our work, a non-linear adaptive FIR-type estimator is used. This predictor was observed to perform well for images that contain very sharp edges and text [14]. The data within the analysis window is first α-trimmed, i.e., the highest and the lowest α-percent of the samples within the analysis window are removed and then an adaptive FIR-type estimator is used to predict the odd samples $\upsilon_1(n)$ from the even samples $\upsilon_2(n)$ as follows:

$$\widehat{\upsilon}_2(n) = \sum_{k=-N}^{N} \omega_{n+k}\upsilon_1(n-k) = \sum_{k=-N}^{N} \omega_{n+k}\upsilon(2n-2k) \tag{1}$$

Filter coefficients ω_{n+k} are updated using an LMS type algorithm as follows:

$$\widehat{\mathrm{w}}(n+1) = \widehat{\mathrm{w}}(n) + \mu \frac{\overline{\mathrm{v}}_n e(n)}{\|\overline{\mathrm{v}}_n\|^2}. \tag{2}$$

where $\widehat{\mathrm{w}}(n) = [\omega_{n-N}, ..., \omega_{n+N}]$ is the weight vector at time instant n, $e(n) = \upsilon_h(n) = \upsilon_2(n) - \overline{\mathrm{v}}^T(n)\overline{\mathrm{w}}(n)$, and $\overline{\mathrm{v}}_n = [\upsilon_1(n-N), \upsilon_1(n-N+1), ..., \upsilon_1(n+N-1), \upsilon_1(n+N)]^T$. The highband subsignal υ_h is given by

$$\upsilon_h(n) = \upsilon_2(n) - \widehat{\upsilon}_2(n). \tag{3}$$

This structure is the simplest adaptive filterbank. Other adaptive filterbanks in which the low-band subsignal is a lowpass filtered and downsampled version of the original signal can be found in [14].

The extension of the adaptive filterbank structure to two dimensions is straightforward. We first process the image x row-wise then column-wise, and obtain four subimages, x_{ll}, x_{lh}, x_{hl} and x_{hh}. We use the subimages x_{lh} and x_{hl} to extract features. It is well-known that the local extrema in these images correspond to edges in the original image [18, 19]. Let us define a pixel $x_{lh}(m_o, n_o)$ satisfying $|x_{lh}(m_o, n_o)| > |x_{lh}(m_o \pm 1, n_o)|$ a horizontal extremum of x_{lh}. A vertical extremum of x_{hl} is defined in a similar manner. We simply determine the number of significant horizontal and local maxima in three horizontal and vertical lines in subimages $|x_{lh}|$ and $|x_{hl}|$, respectively. The distance between horizontal and vertical lines are selected so that each symbol in the codebook library is divided almost equally. For each template image, we use these numbers to form a feature vector. In order to make this vector scale invariant, three horizontal (vertical) lines divide $|x_{lh}|$ ($|x_{hl}|$) subimage into three equal size regions according to the width (height) of the $|x_{lh}|$ and $|x_{hl}|$ subimage.

The main advantage of the adaptive wavelet transform over ordinary wavelet transform is that no ringing artifacts appear at the boundaries of the letters. In ordinary subband filters ringing occurs and this may lead to incorrect results in estimating feature parameters which are based on extrema in subband images. In binary images, binary subband decomposition methods [4] can be used to extract features. Definition of an extrema is the same as above.

3.4 Discussion on Selected Features

Features from the spatial domain are based on *distance* and *angular* spans. These are two basic vectors that encode information on the pixel distribution within a symbol. The angular span encodes information with respect to the center of mass and the x-axis. The distance span encodes information with respect to the center of mass only. This can be better described by an example. Consider two Latin characters ‘n’ and ‘u’. The distance span of the two symbols are the same, but their angular span is quite different. Obviously, there are situations where the angular span of the symbols are very close. Such a situation may occur when an ‘o’-like symbol is written in two forms with different circularity. Since we consider handwritten and connected scripts, the two features from the spatial domains are not adequate by themselves to differentiate nor uniquely specify symbols.

The wavelet extrema feature vector handles the problem in a different way. For example, consider a ‘p’-like symbol. Assume that the symbol has two occurrences in one of which the vertical line of the symbol is a bit longer than the other one. Then, both of the features from the spatial domain give different values for these two symbols. However, their wavelet extrema vectors are similar enough to classify these two symbols as equal to each other.

As a result, since each of the features has some limitations in various cases, we decide to employ a combination of these three vectors in computing the correlation between two extracted symbols. However, the system might still fail for some cases. For example, if a writer tries to fit words at the end of lines then this leads to distortions of some words. Semantically, the written words are the same, but morphologically they look like two different words.

4 Content-Based Retrieval of Ottoman Archives

In our content-based retrieval system, the query specification is performed by identifying a region containing a word, a phrase, or a sentence in the document image. Any keyword or pattern search algorithm can be carried out over the codebooks of the document images. The template image is analyzed, and distance and angular span vectors in spatial domain, and wavelet extrema information are determined. Once the symbols forming the template keyword image are identified according to the extracted features, the locations of the symbols within the documents are determined. If the symbols of a keyword image appear consecutively in the codebook of a document image, then that document is a match for the query. In our system, the documents in the archive are ranked according to the *partial symbol-wise matching* scheme. As the name of this method implies, the similarity of a document to the query image is calculated symbol-wise. In other words, the system orders the resulting document images in the decreasing order of similarities with respect to the symbols, since the query image is also split into symbols.

For a query image of N extracted symbols, a match between the query image and a document D may contain m symbols, where $\frac{N}{2} \leq m \leq N$ by default. The user may alter the lower bound, but the system does not allow lower bounds to be less than $\frac{3N}{10}$, because otherwise unsimilar documents can be retrieved. The similarity of a retrieved document D is denoted as $\frac{m}{N}$ for the match. If there are more than one such matches for a document, then the individual similarity measures are summed up to obtain the final similarity measure.

Ordering of retrieved document images based on symbol-wise similarities yields more efficient results than pixel-wise pattern comparisons, because this approach allows the query to be carried out in the compressed form of a document image. In order to compute the similarities of the symbols with respect to the extracted feature vectors, both in extraction and matching, *histogram intersection* technique [20] is employed, which is used by Swain and Ballard for color descriptor comparisons. In this technique, two vectors are intersected as a whole to obtain a similarity value. Let $H_1[i]$ and $H_2[i]$ denote the i-th entries of two vectors of size n, and S_{H_1,H_2} denote the similarity value between H_1 and H_2. The l_1 norm of the vectors are denoted as $|H_1|$ and $|H_2|$, respectively. Then, the similarity S_{H_1,H_2} can be expressed as:

$$S_{H_1,H_2} = \frac{\sum_i^n \min(H_1[i], H_2[i])}{\min(|H_1|, |H_2|)}. \tag{4}$$

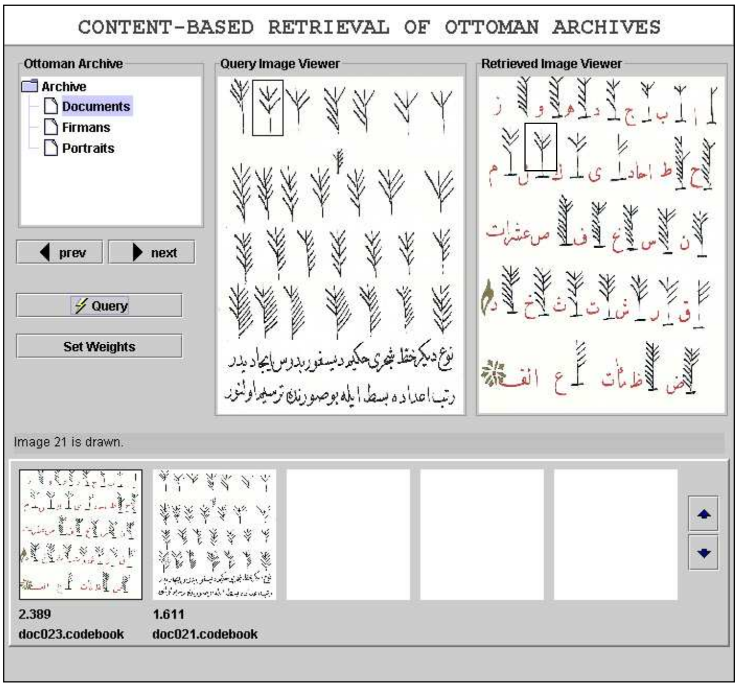
S_{H_1,H_2} takes values between 0 and 1, and if H_1 and H_2 are equal to each other $S_{H_1,H_2} = 1$. The sizes of the feature vectors are 8, 6, and 6 for angular and distance span, and wavelet extrema vectors, respectively. In our case, the above histogram intersection method is used to obtain similarity values for distance and angular span, and wavelet extrema feature vectors. A global similarity value can be obtained by combining these three partial similarity values with appropriate feature weights.

4.1 The User Interface

The main user interface of the content-based retrieval system[1] is shown in Figure 8. The query image viewing part enables browsing the images within the active cluster, which can be set easily from the archive hierarchy on the upper left part of the interface. The query selection is performed by defining the query image region as a rectangle from the active image. This query specification process is simply performed by dragging and dropping the mouse over the query image viewer. The weights of each of the three features (angular span, distance span, and wavelet extrema) can be set separately, which provides content-based retrieval by stressing on features in a desired way (cf. Section 5).

The documents in the archive are clustered based on the types of digitized textual images (e.g., documents, firmans, portraits). Images in other clusters can be also queried in addition to the active folder, and the resulting textual images are presented in the bottom part of the screen. The resulting documents are ranked with respect to the partial symbol-wise matching scheme described above. All of the retrieved documents can be viewed with the help of scrolling buttons, and any textual image can be enlarged in the upper right part of the interface. In this system, the query region(s), as well as the partial matches, are identified as rectangles to clarify the content-based retrieval.

[1] The system is available at `http://www.cs.bilkent.edu.tr/∼ediz/bilmdg/ottoman/webclient.html`


CONTENT-BASED RETRIEVAL OF OTTOMAN ARCHIVES

Ottoman Archive

- Archive
  - Documents
  - Firmans
  - Portraits

Image 21 is drawn.

2.389
doc023.codebook

1.611
doc021.codebook

Figure 8: *The user interface of the content-based retrieval system. The query is identified by drawing a rectangle from the active image. The retrieved documents, ranked by partial symbol-wise matching scheme, can be browsed by the help of the scrolling buttons, and any textual image can be viewed in enlarged form.*

4.2 Query Examples

In this section, three query examples are discussed. The sample queries are selected among the document images and the retrieval analysis is performed manually by evaluating the results.

In Figure 9, a frequent keyword is queried and the system responses to this query, as shown in Figure 9(a). The system returns 21 documents 5 of which are irrelevant. All of the relevant documents are retrieved and the first two of them are shown in Figure 9(b) and (c). In order to visualize the matched keywords better, the retrieved document can be displayed in a separate window in its original size, while it can also be displayed in the main user interface.

In Figure 10, a keyword that only appears in a single document is queried. The query image has three occurrences within the document. The system returns that single document and all of the three occurrences are found correctly within the retrieved document, as shown in Figure 10(b).

Figure 11 shows another query example with a keyword not as frequent as the first query image. The system returns three documents but in the document archive, there exist five relevant documents for this query. The system fails to retrieve the remaining two documents. The reason for this result is the fact that the query image exists in a somewhat distorted way within the documents that are not retrieved. The distortion is caused by two factors. First, the query keyword is written at the end of a line to fit into the space left. Second, small symbols that are not extracted due to the size limitations are written so close that the system perceives them as new symbols.

The query examples in Figures 9 and 11 also show the scale-invariance property.

5 Performance Evaluation

We constructed a sample Ottoman archive containing documents, portraits, and firmans as well as textual images mixed with figures corresponding to 102 document images. The document images were collected from the books [21, 22, 23, 24] and the Web site of the *Topkapı Palace Museum* [25]. The average resolution of the scanned document images is 415 × 345 pixels. Relying on the Ottoman script symbols, the connected components less than 100 pixels are discarded during the symbol extraction process. In most Ottoman documents, there are no small symbols above or below regular characters Arabic and Persian documents. Therefore, there is no retrieval accuracy loss due to discarding small symbols which may be due to noise. Thus, the average symbol count for the archive is 31 symbols per document image. The number of different symbols among the documents in the archive is 228. The average symbol extraction process is 0.39 seconds on a 2000 MHz PC.

In order to evaluate the effectiveness of the retrieval system, two well-known metrics, *precision* and *recall*, are estimated [26]. Precision is the ratio of the number of retrieved images that are relevant to the number of retrieved images. Recall is the ratio of the number of retrieved images that are relevant to the total number of relevant images. Prior to the experiments, the relevance degrees (1 for relevance, 0 for irrelevance) are subjectively assigned to the images in order to signify the relevance of an image to the query image. Moreover, the average case accuracy of the retrieval is examined for both each feature separately and all of the features integrated.

In the experiments, the query images are randomly picked from the archive among the set of candidate query images. We have determined this set of candidate query images as the image regions corresponding to some keywords appear more than once among the documents. There are 30 candidate query images in our set, and 16 query images are tested for the experiments. The effectiveness is evaluated as the average of the results calculated for each query separately. As in most information retrieval systems analysis [26], the individual precision values are interpolated to a set of 11 standard recall levels (0, 0.1, 0.2, ..., 1) in order to facilitate the computation of average of precision and recall values.

Figure 12 shows the interpolated precision-recall graph to evaluate the overall retrieval effectiveness. The retrieval process is performed by integrating all feature vectors. To determine the similarity of a query, a global metric can be obtained by linear combination of three partial similarities with appropriate weights. A possible set of weights can be determined by performing similarity calculations for each feature separately [27], which also provides more effective results by reflecting the characteristics of each feature. The overall similarity value S_T is defined as follows:

$$S_T = \frac{S_A \times w_A + S_D \times w_D + S_{WE} \times w_{WE}}{w_A + w_D + w_{WE}}, \tag{5}$$

where S_A, S_D and S_{WE} denote similarity measures for angular span, distance span and wavelet extrema features, respectively. In Equation (5), angular span, distance span and wavelet extrema features are linearly combined with prespecified weights w_A, w_D and w_{WE}.

Figure 13 presents the retrieval effectiveness of each feature separately. It can be seen that all of the three features have very close average precision values for the standard recall levels. Wavelet extrema information and angular span produce better results compared to distance span in medium level recall levels. To see the effect of the retrieval accuracy, a graph for the number of retrieved relevant documents versus the number of documents is plotted in Figure 14. By the help of this experiment, the feature weights are estimated to yield more effective results in querying and retrieval. All of the features were observed to find a relevant document in the first response, and the accuracy is at least 92% when first 5 responses are considered. The features are assigned almost equal weights, and the overall retrieval performance is obtained with weights w_A, w_D and w_{WE} as $0,35$, $0,30$, and $0,35$, respectively.

6 Conclusions

In this paper, a content-based retrieval system for the Ottoman documents is presented. Documents in the database are compressed using an improved version of the textual image coding scheme developed in [2]. An important characteristic of the symbol extraction process used in data compression is its scale invariance, which enables to query any type of textual image region successfully.

The Ottoman documents compressed in this special textual form are effectively queried by specifying rectangular regions in document images without any limitation on the size of the query region. The querying takes place using only the compressed data. Our symbol matching scheme ranks the resulting document images in the decreasing order of similarities with respect to the number of symbols matched. The total similarity is computed by comparing the symbols in the query region with the pointers to the codebook. The resulting documents are presented by identifying the matched region of each document in a rectangle.

The symbol extraction and correlation process is scale-invariant and three features from spatial and wavelet domain are extracted. Angular span is a vector whose entries are the number of black pixels in θ-degree slices centered at the center of mass of a symbol with respect to the horizontal axis. Distance span is also a vector whose entries are the number of black pixels in concentric circles centered at the center of mass and separated by equal distances. These span vectors are normalized to provide scale invariance. Scale-invariant features are also extracted from the extrema of the wavelet transform computed by the Adaptive Subband Decomposition (ASD).

Our framework ensures the storage and content-based retrieval of documents written in ordinary and connected scripts, or even documents containing text mixed with pictures, figures, signs, etc. Our symbol-extraction scheme allows us to process gray level textual images without a binarization process for symbol-extraction which may disturb the originality of the historical documents. The proposed framework for content-based retrieval is also applicable to other historical and cultural archives using different scripts.

Effectiveness of the retrieval process was evaluated using interpolated precision-recall graphs for each of the features separately, and better results are obtained by linearly combining the three features. Experimental studies show that the system successfully retrieve documents containing the keyword image.

References

[1] R.N. Ascher and G. Nagy, "A means of achieving a high degree of compaction on scan-digitized printed text," *IEEE Trans Computers*, vol. C–23, no. 11, pp. 1174–1179, 1974.

[2] I.H. Witten, A. Moffat, and T.C. Bell, *Managing Gigabytes*, San Mateo, CA: Morgan Kaufmann, 1999.

[3] I.H. Witten, T.C. Bell, H. Emberson, S. Inglis, and A. Moffat, "Textual image compression: Two-stage lossy/lossless encoding of textual images," in *Proceedings of IEEE*, 1994, vol. 82, pp. 878–888.

[4] Ö.N. Gerek, E. Çetin, A.H. Tewfik, and V. Atalay, "Subband domain coding of binary textual images for document archiving," *IEEE Transactions on Image Processing*, vol. 8, no. 10, pp. 1438–1446, 1999.

[5] L. Bottou, P. Haffner, P.G. Howard, P. Simard, Y. Bengio, and Y. Le Cun, "High quality document image compression with DjVu," *Journal of Electronic Imaging*, vol. 7, no. 3, pp. 410–425, SPIE, 1998.

[6] P. Haffner, L. Bottou, P.G. Howard, and Y. Le Cun, "DjVu: analyzing and compressing scanned documents for internet distribution," in *Proceedings of the International Conference on Document Analysis and Recognition*, 1999, pp. 625–628.

[7] O.D. Trier, A.K. Jain, and T. Taxt, "Feature extraction methods for character recognition - a survey," *Pattern Recognition*, vol. 29, no. 4, pp. 641–662, 1996.

[8] O.D. Trier and T. Taxt, "Evaluation of binarization methods for document images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, pp. 312–315, 1995.

[9] J.D. Tubbs, "A note on binary template matching," *Pattern Recognition*, vol. 22, no. 4, pp. 359–365, 1989.

[10] P. Gader, B. Forester, M. Ganzberger, A. Gillies, M. Whalen, and T. Yocum, "Recognition of handwritten digits using template and model matching," *Pattern Recognition*, vol. 24, no. 5, pp. 421–431, 1991.

[11] M.H. Glauberman, "Character recognition for business machines," *Electronics*, vol. 29, pp. 132–136, 1956.

[12] E. Şaykol, U. Güdükbay, and Ö. Ulusoy, "A histogram-based approach for object-based query-by-shape-and-color in multimedia databases," *submitted paper (an older version is available as Tech. Rep. BU-CE-0201, Bilkent University, Comp. Eng. Dept.,* `http://www.cs.bilkent.edu.tr/tech-reports/2002/BU-CE-0201.ps.gz`*)*, 2002.

[13] M.E. Dönderler, E. Şaykol, Ö. Ulusoy, and U. Güdükbay, "BilVideo: A video database management system," *IEEE Multimedia*, vol. 10, no. 1, pp. 66–70, January/March 2003.

[14] Ö.N. Gerek and E. Çetin, "Adaptive polyphase subband decomposition structures for image compression," *IEEE Transactions on Image Processing*, vol. 9, pp. 1649–1660, 2000.

[15] F.J. Hampson and J.C. Pesquet, "A nonlinear subband decomposition structure with perfect reconstruction," in *Proceedings of the IEEE Int. Conf. on Image Proccessing*, 1996.

[16] W. Sweldens, "The lifting scheme: A new philosophy in biorthogonal wavelet constructions," in *Proceedings of the SPIE*, 1995, vol. 2569, pp. 68–79.

[17] G. Piella, B. Pesquet-Popescu, and H. Heijmans, "Adaptive update lifting with a decision rule based on derivative filters," *IEEE Signal Processing Letters*, pp. 329–332, October 2002.

[18] S. Mallat, *A Wavelet Tour of Signal Processing*, ISBN: 0-12-466606-X, Academic Press, 1999.

[19] A.E. Çetin and R. Ansari, "Signal recovery from wavelet transform maxima," *IEEE Transactions on Signal Processing*, vol. 42, no. 1, pp. 194–196, January 1994.

[20] M.J. Swain and D.H. Ballard, "Color indexing," *International Journal of Computer Vision*, vol. 7, no. 1, pp. 11–32, 1991.

[21] A. Alparslan, *Osmanlı Hat Sanatı Tarihi*, (in Turkish), Yapı Kredi Yayınları, 1999.

[22] M.U. Derman, *Hat Koleksiyonundan Seçmeler*, (in Turkish), Sakıp Sabancı Müzesi, Sabancı Üniversitesi, 2002.

[23] Hakkak-zade Mustafa Hilmi Efendi, *Mizanü'l-hatt*, (in Turkish), Osmanlı, İstanbul, 1986.

[24] M. Ülker, *Başlangıçtan Günümüze Türk Hat Sanatı*, (in Turkish), Türkiye İş Bankası Kültür Yayınları, 1987.

[25] E. Çetin, Ö.N. Gerek, and A.H. Tewfik, "The Topkapı Palace Museum," *Museum International*, vol. 1, no. 1, pp. 22–25, 2000, the Web site is available at `http://www.ee.bilkent.edu.tr/~history/topkapi.html`.

[26] K.S. Jones, *Information Retrieval Experiment*, Butterworth and Co., 1981.

[27] A.K. Jain and A. Vailaya, "Image retrieval using color and shape," *Pattern Recognition*, vol. 29, no. 8, pp. 1233–1244, 1996.

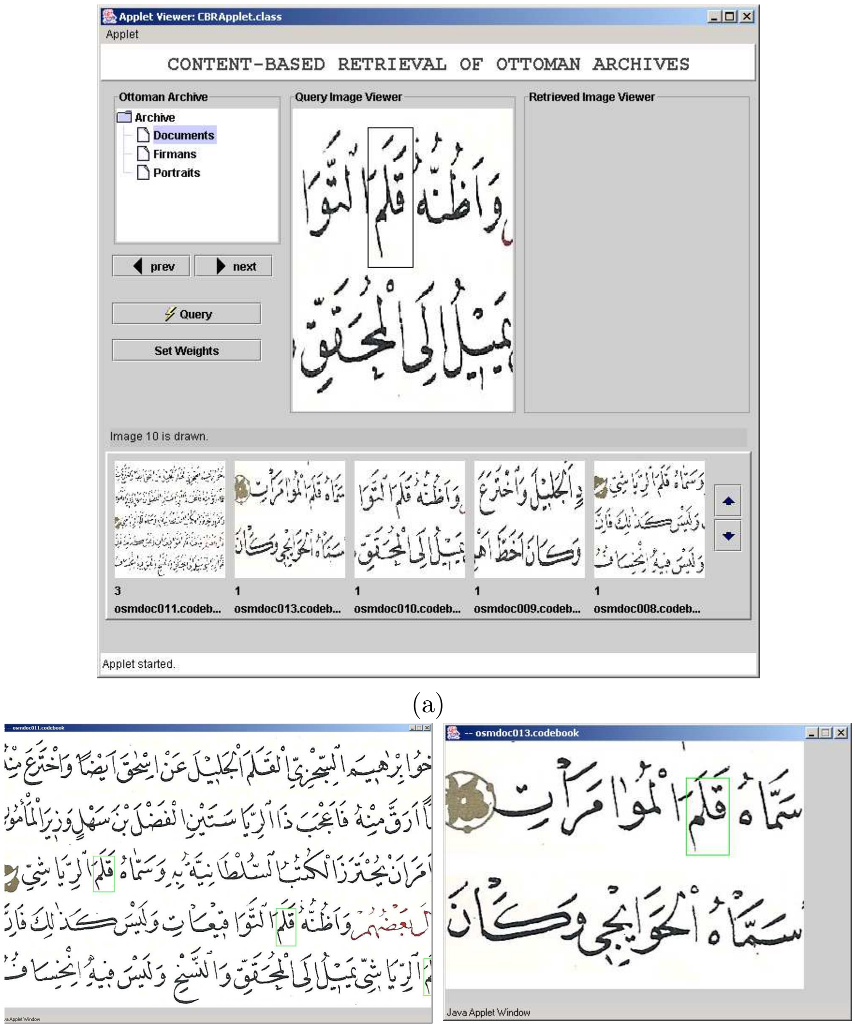


Image 10 is drawn.

Applet started.

(a)

(b)

(c)

Figure 9: *A query example. (a) Document image where the query image is specified by a rectangle. (b) First retrieved document with 3 matches. (c) Second retrieved document with 1 match. The matched keywords in (b) and (c) have different scales compared to the keyword in the query image.*

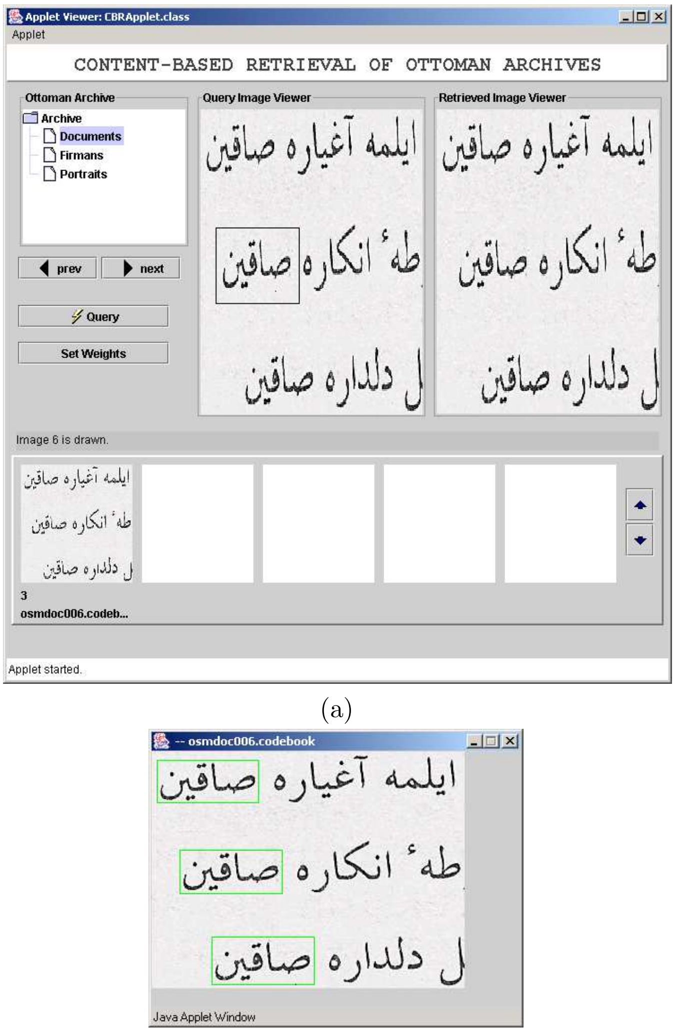


Image 6 is drawn.

Applet started.

(a)

(b)

Figure 10: *A query example. (a) Document image where the query image is specified by a rectangle. (b) First retrieved document with 3 matches.*

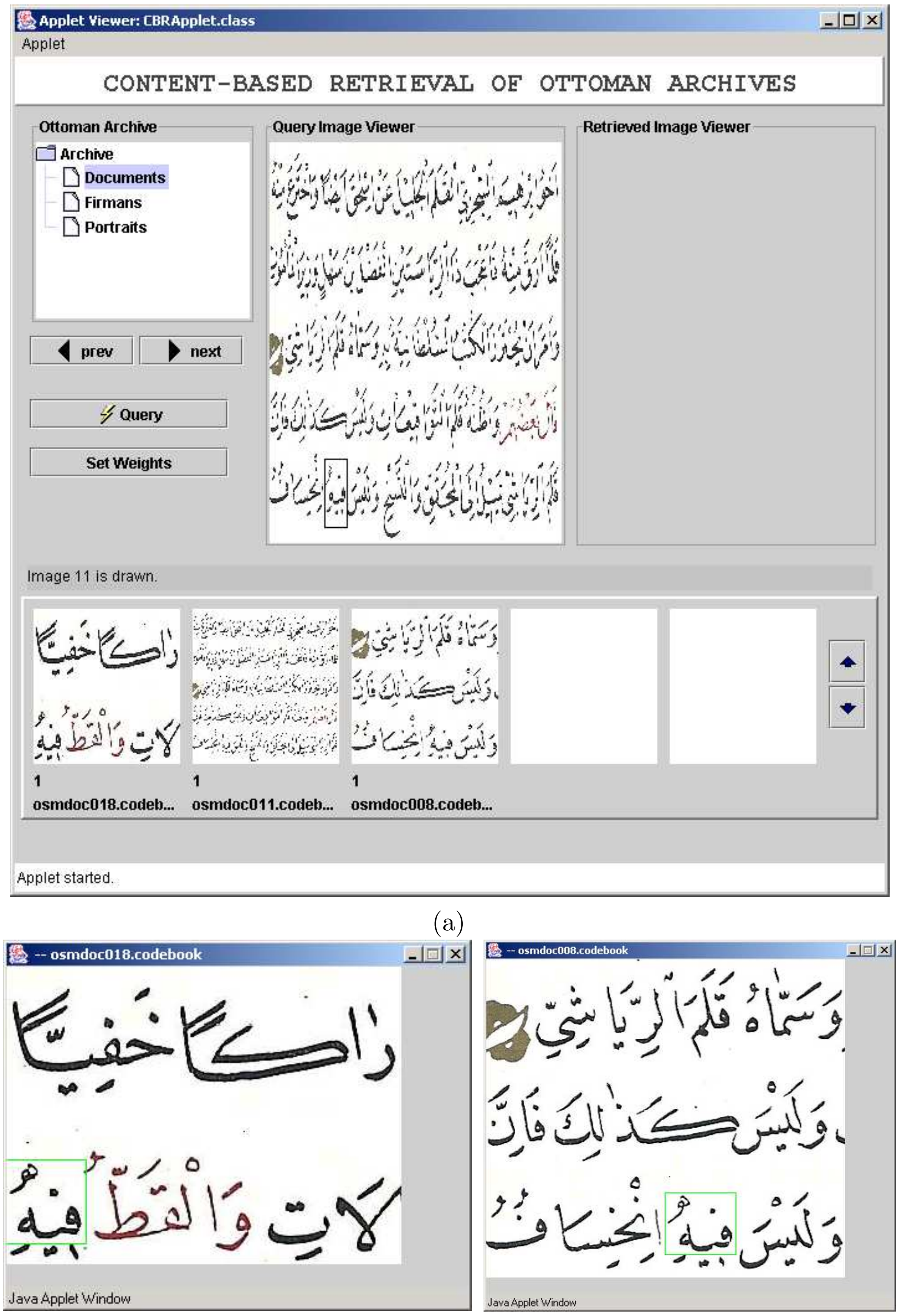


Image 11 is drawn.

Applet started.

(a)

(b)

(c)

Figure 11: *A query example. (a) Document image where the query image is specified by a rectangle. (b) First retrieved document with 1 match. (c) Third retrieved document with 1 match (second retrieved document is the query document itself.). This query example also shows the scale invariance property.*

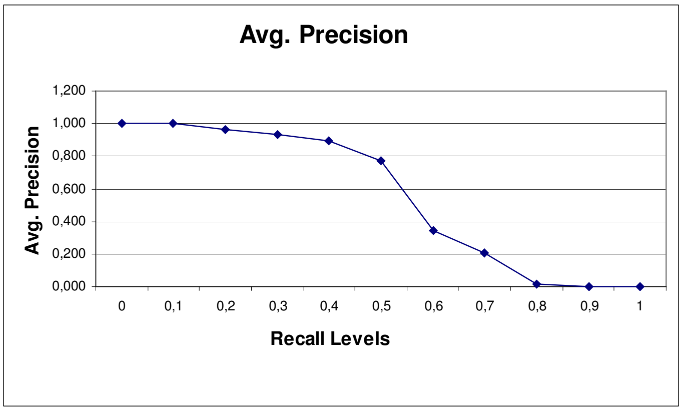


Figure 12: *The interpolated precision-recall graph of retrieval effectiveness with features integrated.*

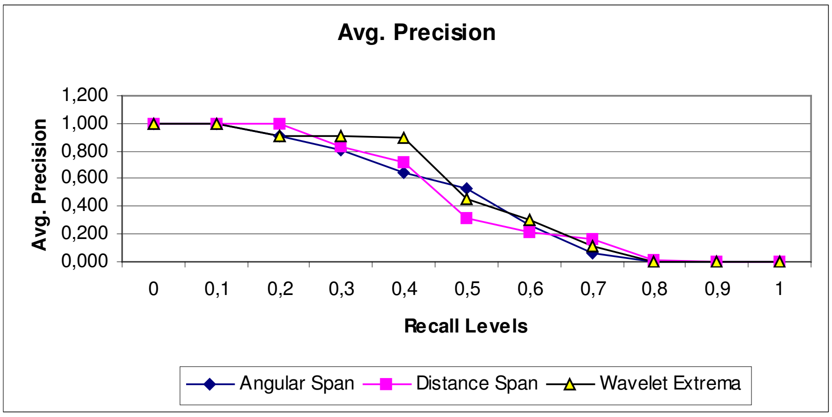


Figure 13: *The interpolated precision-recall graph of retrieval effectiveness of each feature.*

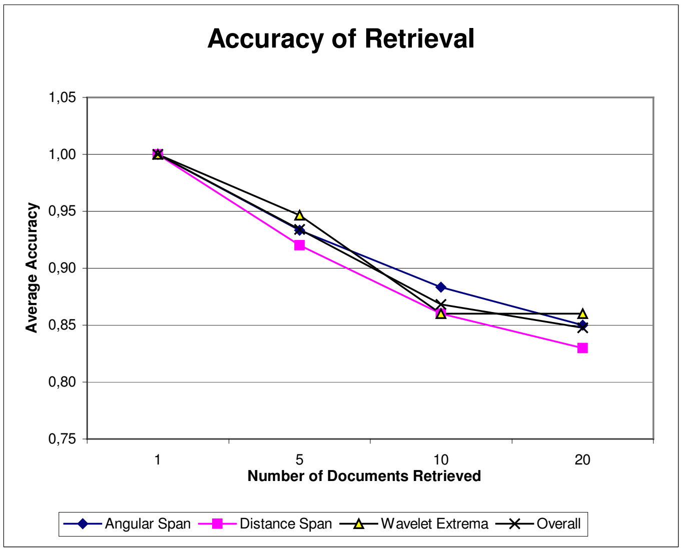


Figure 14: *The accuracy of retrieval for each feature and linear combination of all three features.*

A An Example Textual Image in Compressed Form

An excerpt from a document in textual form is given in this section. The first of five integers in a row is the symbol identifier, and the remaining four values are *x-offset*, *y-offset*, *symbol-width*, and *symbol-height*, respectively. The offset coordinates denote the Euclidean distance between the lower left corners of the two adjacent symbols.

```
984 16 83 7 8
556 2 -28 7 9
2322 0 62 7 8
1838 7 69 7 8
1802 5 -18 7 13
2258 5 -8 15 29
819 6 -43 11 12
682 21 -15 8 39
1802 6 23 8 16
1378 7 -5 8 46
2263 14 -29 7 44
1008 15 -13 13 26
1537 1 -19 7 9
1838 15 44 7 8
2098 7 30 8 8
2107 8 -18 8 8
14 1 -13 9 11
419 8 -15 10 12
1838 10 16 7 8
1226 1 22 8 18
984 6 -31 7 8
1092 2 24 7 8
756 15 -43 7 37
433 8 14 11 11
1607 2 64 8 9
1654 4 -14 9 21
1812 11 26 8 16
556 7 -16 7 9
1 7 -45 10 19
...
884 2 -54 14 20
2000 920 13 9
1187 6 -34 12 22
2384 9 -38 7 7
1711 7 0 10 15
97 15 2 7 8
2337 1 58 9 10
97 15 46 7 8
2456 0 3 11 27
2331 8 -8 14 33
1455 11 12 8 24
36 5 -24 12 7
2384 8 -7 7 7
1253 8 -47 10 11
1638 9 -6 13 12
1423 7 -38 7 10
1974 18 10 7 8
756 13 16 7 37
984 9 -2 7 7
1197 10 12 10 14
2337 7 -31 9 10
52 22 -27 12 19
1197 5 -12 7 9
2384 0 44 7 7
392 11 -31 7 8
433 4 -39 11 11
934 3 -42 12 20
1838 4 17 7 8
2384 4 62 7 7
...
1539 5 55 8 7
228 11 -45 10 13
1008 14 -35 13 21
1253 2 35 7 7
2062 5 -10 10 12
1624 0 13 8 8
1597 4 -35 10 13
433 1 -47 11 11
1804 36 3 7 31
2332 0 31 14 23
1838 8 17 7 8
1312 7 42 12 13
550 23 -28 11 38
546 0 17 13 11
33 5 19 7 7
33 1 40 7 7
1984 2 -34 8 39
1384 4 36 7 22
1035 4 -42 7 8
774 1 18 8 14
1540 1 27 7 12
2337 6 16 9 10
1624 7 0 8 8
1253 9 -66 7 7
1607 6 26 8 9
1339 11 16 11 43
1988 5 40 11 18
2108 11 2 10 24
1607 8 -2 8 9
...
```