Serdica J. Computing **5** (2011), 183–198

Serdica
Journal of Computing

Bulgarian Academy of Sciences
Institute of Mathematics and Informatics

PROCESSING OF BYZANTINE NEUME NOTATION IN ANCIENT HISTORICAL MANUSCRIPTS*,†

Lasko Laskov

Abstract. Byzantine neume notation is a specific form of note script, used by the Orthodox Christian Church since ancient times until nowadays for writing music and musical forms in sacred documents. Such documents are an object of extensive scientific research and naturally with the development of computer and information technologies the need of a software tool which can assist these efforts is needed. In this paper a set of algorithms for processing and analysis of Byzantine neume notation are presented which include document image segmentation, character feature vector extraction, classifier learning and character recognition. The described algorithms are implemented as an integrated scientific software system.

1. Introduction. During the last decade special attention has been paid to the preservation of scientific and cultural heritage. With the development of computer and information technologies new methods emerge for preservation and

---

ACM Computing Classification System (1998): I.7, I.7.5.

Key words: historical document image processing, optical character recognition, neume notation.

*This article presents the principal results of the doctoral thesis "Recognition of neume notation in historical documents" by Lasko Laskov (Institute of Mathematics and Informatics at Bulgarian Academy of Sciences), successfully defended before the Specialized Academic Council for Informatics and Mathematical Modelling on 07 June 2010.

†This work has been partly supported by Grant No. DTK 02/54, Bulgarian Science Fund, Ministry of Education, Youth and Science.

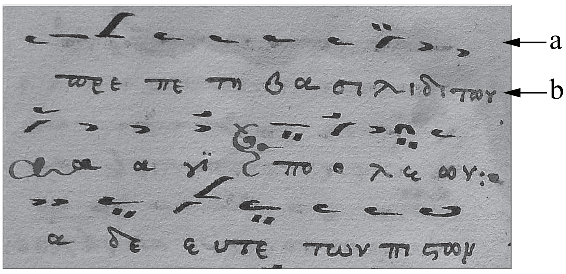


Fig. 1. A manuscript fragment which contains Byzantine neume notation. The neumes (a) are written on top of the ancient Greek text lines (b), specifying the way they have to be performed musically

distribution of ancient historical documents, such as digital libraries and databases containing digital copies of the documents accompanied with the required meta-data information. Digital libraries and historical manuscript databases provide a significantly wider circle of scientists with access to the documents needed for their research without any document preservation issues since no physical contact with the requested document is required. An object of such digitalization projects are the historical documents as a whole, as well as the ancient manuscripts containing Byzantine neume notation.

The digital copies of the manuscripts containing neume notation accompanied by the required meta-data are an important source of information for the scientific research of these documents. Most of the studies in this field are directly connected with the contents of the manuscripts, including search for similar fragments and combinations of neumes, comparisons between them, search for similarities in different documents, etc. These technical activities are a good reason for the development of a software tool which will help the automation and hence expedite the work of the researchers of the Byzantine neume notation. Such software should be capable to extract the required information contained in the document images, so the above automation will become possible.

During each historical period of the development of the neume notation it has been an alphabet composed by a finite number of symbols, written on top of the lines of the religious text, indicating the way it has to be performed musically (see Fig. 1). Because of its relatively simple linear structure of writing and the finite number of symbols, the neume notation suggests an OCR (optical character recognition) approach to the solution of the problem of automatic information extraction from the manuscripts images.

The processing and recognition of ancient manuscripts is a problem which cannot be solved using the standard OCR techniques, applicable in the case of standard texts written in contemporary alphabets such as Latin, Cyrillic, Greek, Chinese, etc. There are a number of specific features of the ancient documents, such as the presence of noise, destroyed or faint document parts, ancient and even unknown alphabets and languages. This is the reason to search for specific document image processing algorithms applicable for the precise problem being solved, as in the case of the ancient manuscripts containing neume notation.

In the literature there are not many attempts to develop a software system for processing and recognition of neume notation. Maybe one of the first works is the one by Gezerlis and Theodoridis [9] where an off-line recognition system is presented which processes the contemporary Byzantine neume writing. It is based on a nearest neighbor classifier where the feature space is built from the wavelet transform of the symbols, their projective profiles, Euler number, etc. A structural analysis of groups of neumes is presented which is based on a database of all such possible groups. Very good results are reported in the case of printed text and for two cases of carefully hand-copied texts. In this work the problem of document image segmentation is not discussed in detail despite the specific structure of these documents.

Another more recent work is [3] in which a software system is presented for processing the contemporary Byzantine neume notation. The described algorithms include document image segmentation, separation of the neume lines from the text lines, symbol recognition and symbol grouping. k-th nearest neighbor classifier is used and the implementation of the system is based on the Gamera framework [6]. For the feature space definition a combination of symbols' characteristics are used, such as minimal wrapping window, aspect ratio, moments, volume, etc. Again, very good results are reported in the case of printed texts.

Both works mentioned above assume the following preliminary conditions:

1. The systems are designed to work with printed texts, which assumes a *relatively good quality* of the input document images.

2. The writing being recognized is the contemporary Byzantine neume notation from which follows that *all the symbols and the rules for their combining are known*. It is assumed that in all documents the same standard notation is used which allows the application of classifiers which have passed the learning step beforehand.

In the presented problem this two conditions are not applicable since in the case of ancient manuscripts the document image quality is one of the main issues. Also, for the earlier documents the notation and the rules for symbols combination are not always known. This is the reason why the approach presented in this paper differs from the standard OCR approach and the neume notation is considered as built of an unknown alphabet and the presented algorithms are not bound to a concrete neume writing.

This paper is organized as follows. In Section 2 the problem of document image segmentation is examined in the case of the ancient neume script. Section 3 is dedicated to the neume symbols' feature vector extraction and self-learning classifier design. Finally, in Section 4 conclusions and some guidelines for future work are given.

2. Segmentation of neume notation in ancient manuscripts. In this section two methods for segmentation of Byzantine neume notation in ancient manuscripts are presented. The first method is based on the color information contained in the documents which often holds semantic meaning [11]. The second one is based on document structure analysis and assumes that the color information of the document has been extracted in advance [12].

2.1. Color segmentation of neume notation. In the literature the segmentation of the object pixels from the background pixels is often referred to as *binarization* assuming that the image pixels have to be distributed in two classes – object pixels and background pixels. Thus binarization is the process of conversion of a grayscale image to a binary one in which the black pixels represent the objects and the white pixels represent the background.

In the case of historical manuscripts containing neume notation there are meaningful objects in different colors where the color information has some semantic meaning. In the examined data pull the following objects are observed: (i) neume notation and concomitant text written in dark color; (ii) additional notes in red and neumes written in red; (iii) achromatic background or background with specific color which is highly dependent on the age, type and preservation of the parchment.

The presented approach for separation of the abovedefined objects is based on histogram approaches and Otsu's method [13], [10] is adopted for finding a set of statistically optimal thresholds. The two presented color segmentation algorithms are based on histograms in the HSV (hue, saturation, value) color space instead of the classical RGB.

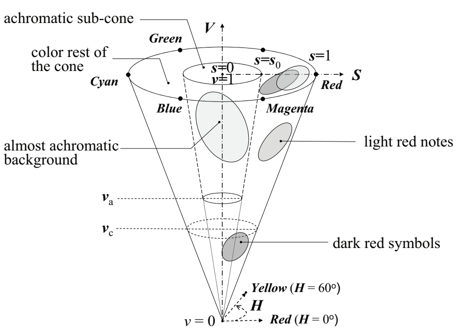
2.1.1. HSV histogram analysis. The HSV color space is usually represented as a cone [7], where: (i) hue is the color parameter $h \in [0, 360^\circ)$ which consecutively represents the colors from red, yellow, green, cyan, blue to magenta; (ii) saturation $s \in [0, 1]$ varies from unsaturated gray to fully saturated colors; (iii) value $v \in [0, 1]$ represents the brightness. Based on the HSV color space, the following three operations are defined:

Operation 1. **S-histogram segmentation.** Otsu's method is applied on the S-histogram which is accumulated along the S-axis of the HSV color space. The statistically optimal threshold s_0 divides the HSV cone in two parts: a nearly achromatic part $S_0(s)$, $s < s_0$, and a part which contains most of the color information of the image, $S_1(s)$, $s_0 \leq s \leq 1$.

Operation 2. **V-histogram segmentation.** Again, Otsu's method is applied to calculate a threshold v_0 on the V-histogram which divides the HSV cone in two separate parts: a dark part $V_0(v)$ for which $0 \leq v \leq v_0$ and a light part for which $v_0 < v \leq 1$.

Operation 3. **H-histogram segmentation.** Since the H-histogram is cyclic, an extension of Otsu's algorithm is applied which finds a pair of statistically optimal thresholds h_0 and h_1 which maximizes Otsu's criterion $\eta(h_0|h_1)$.

Based on the above operations two algorithms (*Algorithm A* and *Algorithm B*) for color segmentation of neume notation are proposed.



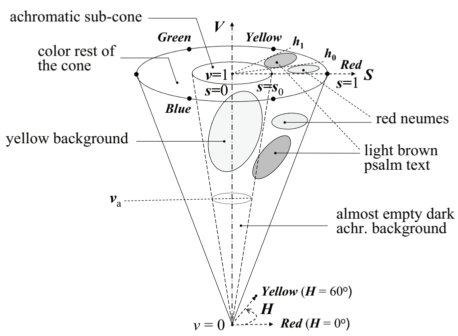


Fig. 2. The HSV scheme of the meaningful volumes for: (a) Algorithm A; (b) Algorithm B

2.1.2. Algorithm A.

Step 1. Operation 1 is applied. The resulting threshold s_0 divides the HSV cone into an achromatic part $S_0(s)$ and the remaining chromatic part $S_1(s)$.
Step 2. Operation 2 is applied on the achromatic cone part $S_0(s)$. The resulting threshold v_a separates the dark symbols from the achromatic component of the background, $S_0(s, v : v_a < v \leq 1)$.
Step 3. Operation 2 is applied again but this time on the chromatic part of the HSV cone $S_1(s)$. The resulting threshold v_c separates the dark symbols contained in neume and text lines from the light-red symbols of the additional

notes. Actually both sets of pixels, separated by v_c, have nearly the same color but they have different saturation.

2.1.2. Algorithm B.
Step 1. The same as Step 1 of Algorithm A.
Step 2. Similar to Step 2 of Algorithm A, but this time the interpretation of the threshold v_a is that it separates the dominating light part from the nearly empty dark part. In this case both parts of S are considered as background.
Step 3. Operation 3 is applied on the color part $S_1 s$ of the HSV cone. The resulting thresholds in the periodic H-histogram separate the light-red neume symbols from the darker text symbols.

The separation of the HSV cone into meaningful volumes by Algorithm A and Algorithm B is given respectively in Fig. 2(a) and Fig. 2(b).

2.2. Structural segmentation of neume notation. In the previous paragraph the problem of color segmentation was described on pixel level. The structural segmentation is the process of detection of document paragraphs, lines and symbols by analyzing the document on a structural level. It assumes that the color information was extracted in advance and the input images are binary.

Because of the specifics of the examined manuscripts, it can be considered that their lines are horizontal and the symbols are written carefully in calligraphic manner. This allows horizontal and vertical projection profiles to be used to segment the lines and symbols. Also, these documents have a specific structure – each even line is a text line and the line above it contains the neume notation which determines the musical interpretation of the text (see Fig. 1).

The proposed method is composed from three basic stages:

1. Initial line segmentation using the horizontal projective profile of the document image.
2. Initial symbol segmentation using the individual vertical projective profile for each separate line, found at the previous stage.
3. The collision solve of segmentation for the overlapping symbols which belong to different document lines.

Given the discrete image $I_{M\times N}(x,y)$ with M rows and N columns, the definition of horizontal and vertical projective profile is as follows:

Definition 1. *A horizontal projective profile of the image $I(x,y)$ is given by the function*

$$A(y) = \sum_{i=0}^{N-1} I(i,y), \quad y = 0,1,\ldots,M-1 \tag{1}$$

A vertical projective profile of $I(x, y)$ is given by the function

$$B(x) = \sum_{i=0}^{M-1} I(x, i), \quad x = 0, 1, \ldots, N-1 \tag{2}$$

2.2.1. Initial line segmentation. The initial line segmentation starts with the accumulation of the horizontal projection profile (1) of the document image. The local maxima of $A(y)$ correspond to the pivots of the document lines, while the local minima correspond to the gaps between the lines. $A(y)$ also contains a lot of "false" extrema which are caused by noise and symbols' parts which lie in the gaps between the lines. Thus the problem of line detection is reduced to the problem of determining these local extrema of $A(y)$ which correspond to the document lines. The method proposed here is based on one-dimensional discrete filter of the type *floating mean*.

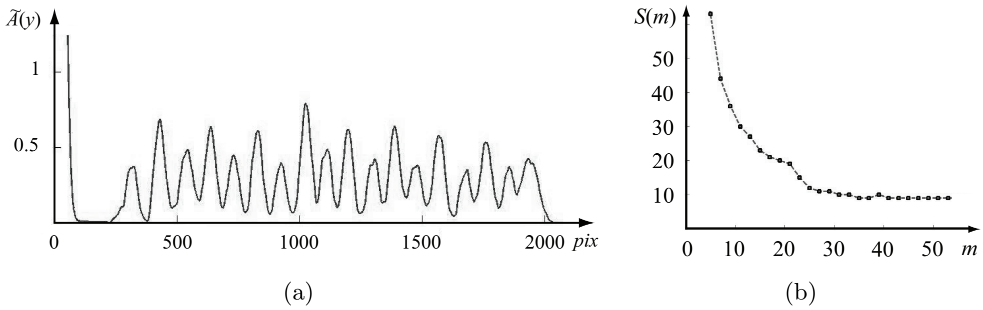


Fig. 3. (a) The projective profile $A(y)$ smoothened with the filter Φ_m, where m = 45. The resulting function $\tilde{A}(y)$ is smooth enough to indicate the local extrema which correspond to document lines. (b) The function $S(m)$ which determines the correspondence between filter size m and the number of local extrema of $\tilde{A}(y)$. In the interval $m \in [45, 55]$ the function $S(m)$ has constant value which indicates that the correct filter size is found

Definition 2. *Floating mean filter is the one-dimensional discrete integrating filter $\Phi_m(y)$:*

$$\Phi_m(y) = \begin{cases} 1, & |y| \leq m/2 \\ 0, & |y| > m/2 \end{cases} \tag{3}$$

where m is the domain width of the filter.

The filtered horizontal profile $\tilde{A}(y)$ can be formally expressed as a convolution of $A(y)$ and $\Phi_m(y)$:

$$(4)\qquad \tilde{A}(y) = (\Phi_m \circ A)(y) = \frac{1}{m}\sum_{i=-m/2}^{m/2} \Phi_m(i)A(y+i),\ \ y = 0, 1, \dots, M-1,$$

where $A(y)$ is filled with zeroes outside the interval $[0, M-1]$. Then the filtered profile $\tilde{A}(y)$ is expressed by:

$$(5)\qquad \tilde{A}(y) = \frac{1}{m}\sum_{i=-m/2}^{m/2} A(y+i),\ \ y = 0, 1, \dots, M-1$$

When $\tilde{A}(y)$ is smoothened with the filter Φ_m then some of the local extrema (mainly the ones which correspond to noise) will be removed depending on the size m of the filter (see Fig. 1(a)). If this process starts with a small filter size, for example $m = 3$, and continues with increasing of m, then the number of extrema of $\tilde{A}(y)$ is going to decrease until $\tilde{A}(y)$ becomes a nearly constant function. This leads to the following:

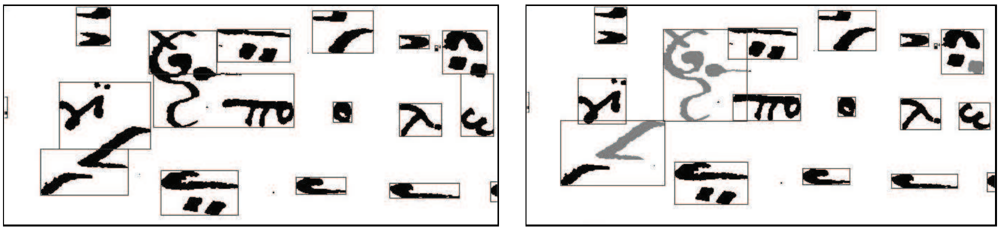
Definition 3. *Let n denote the number of local minima of $\tilde{A}(y)$. Then $S(m) : m \to n$ is a function which gives the correspondence between the size of the filter Φ_m and the number of local minima of the projective profile $A(y)$ smoothened with Φ_m.*

Since the local extrema of interest dominate over the noise extrema, it can be expected that when $\tilde{A}(y)$ is smooth enough the function $S(m)$ will have a nearly constant value which will correspond to the filter sizes for which $\tilde{A}(y)$ will contain only the local extrema which indicate document lines (see Fig. 3(b)).

It should be noted that the function $S(m)$ is not monotonous along its whole domain, especially for the big values of m which are greater than the values in the interval where $S(m)$ is constant.

Once the local extrema which correspond to document lines are found, the lines are segmented as the image area between each two neighboring local minima of $\tilde{A}(y)$.

2.2.2. Initial symbol segmentation. After the line detection has been completed for each segmented document line the vertical projective profile $B(x)$ is calculated (see Definition 1). $B(x)$ is examined for the empty spaces between symbols. Since the images are binary the empty spaces between symbols result in $B(x) = 0$ which leads to whole intervals of gaps between symbols, which are reduced to their middle points. These coordinates are used for calculating the *minimal wrapping window* (MWW) for each symbol or group of symbols. The empty wrapping windows are discarded as noise.



(a) (b)

Fig. 4. (a) Initially segmented symbols in their MWWs; (b) The symbols in their MWWs after the collision solving stage. The previously broken symbols are marked with grey color

2.2.3. Solving segmentation collision. Because of the way of writing, noise and not precisely straight lines at the initial symbol segmentation stage can result in multiple vertical overlapping and breaking of symbols. At this stage of the proposed segmentation method these collisions are solved using the following algorithm.

Step 1. All broken symbols (symbols which are contained in more than one MWW) are detected by finding MWWs with a common horizontal boundary.

Step 2. The broken symbol is marked using a flood-fill algorithm [7] and its MWW is calculated. The initial two MWWs are recalculated without the broken symbol.

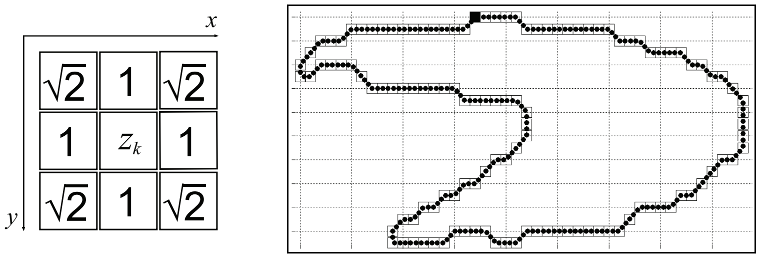
Step 3. If the three MWWs calculated in Step 2 are not empty, the MWW containing the broken symbol is merged to the closest initial window but with priority given to the neume document lines.

Examples of initially detected symbols and the same symbols after the collision solve are given in Fig. 4.

The coordinates of MWW of each segmented neume symbol are used to represent it as a separate raster image in a database of segmented neumes which is appropriate for the next stage of building a self-learning classifier.

3. Self-learning classifier of neume notation. The next stage of the described approach for processing Byzantine neume notation in ancient manuscripts is the design and development of a self-learning classifier of neume symbols. The proposed method is based on *Fourier descriptors* (FDs) [15], [2], [8], [1] of the outer contour of the symbols and the introduction of lexicographical order between the feature vectors composed by them. The method contains three basic stages:

1. Contour extraction for each neume symbol discovered during the segmentation step.
2. A one dimensional complex *Fourier transformation* (FT) of the symbol's contour, affine transformations normalization and reduced frequency representation. The result of these operations is the FDs of the contour.
3. Feature vector representation based on the FDs which is used as an index in a database of the discovered neume representatives for the given document. Introduction of lexicographical order between the feature vectors and sorting the database which reduce the problem to one-dimensional clusterization.



(a) (b)

Fig. 5. (a) The pixel z_k and the distance to its eight neighbors. (b) Original contour of a neume with pixel coordinates represented with squares where the initial pixel is marked in black and the approximated contour is represented with dots. The length of the approximating contour is 256 points and the distance between each pair of consecutive approximating points is 0.51762

3.1. Contour extraction. The first stage of the symbol contour extraction is the contour trace process. For this stage the classical algorithm of Pavlidis [14] is chosen. Given the symbol as a connected set of pixels in a raster image, the contour trace algorithm finds the boundary pixels starting from an initial pixel which in this case is chosen to be the upper leftmost pixel. Then it labels each next boundary pixel using a Freeman code of eight positions of neighborhood.

The resulting Freeman chain code is easily transformed into pixel coordinates and is used at the next stage of contour approximation with 2^n number of equally spaced points in $\mathbb{R}^2$, where $n \in \mathbb{N}$, $n \geq 2$. This stage is needed for two reasons: (i) in order to calculate FDs efficiently, a *fast Fourier transform* (FFT) algorithm is adopted which requires input of size 2^n number of points; (ii) the contour points are not equally spaced when eight positions of the neighborhood are used since the neighboring pixels which lie horizontally or vertically are at distance 1, while the pixels which lie diagonally are at distance $\sqrt{2}$ (Fig. 5(a)).

For this stage a linear approximation scheme is developed which approximates the input discrete contour with another discrete contour composed of 2^n equally spaced points. Each of the approximating points lies on a line segment connecting two neighboring pixels in the input contour and the number of the output points is greater than the number of input pixels to prevent loss of contour shape information (Fig. 5(b)).

3.2. Affine invariant representation of neume symbols in the frequency domain. The representation of a planar shape, such as a neume symbol, by its outer contour allows its interpretation as a 1D discrete complex function by representing each point of the contour as a complex number. This permits 1D complex FT to be adopted for contour transformation into the frequency domain where a feature vector composed by FDs is extracted from the contour spectrum.

A planar closed contour z is given by the coordinates of its points starting from the initial point $z(0)$ in counterclockwise direction:

$$z(k) = (x(k), y(k))\,, \ \ k = 0, 1, \dots, K-1, \tag{6}$$

where K is the number of points. The contour by definition is a closed curve with equally spaced points. It can be represented in the complex plane $\mathbb{C}$ where x is the real and y is the imaginary component:

$$z(k) = x(k) + iy(k), \ \ k = 0, 1, \dots, K-1 \tag{7}$$

and $i = \sqrt{-1}$ is the imaginary unit. Then the contour spectrum is found using the well known 1D complex FT [1]:

$$\hat{z}(\omega) = \frac{1}{K} \sum_{k=0}^{K-1} z(k) e^{-i\frac{2\pi}{K}k\omega}, \ \ \omega = 0, 1, \dots, K-1 \tag{8}$$

In contour representation in the frequency domain, low frequencies contain the information about the global characteristics of the shape, while high frequencies correspond to finer details and noise. Hence, for the purposes of shape matching low frequencies can be considered more important than high frequencies. This leads to the following reduced spectrum representation of the contour:

$$\hat{r}(\omega) = \begin{cases} 0, & l \le \omega \le K-l-1 \\ \hat{z}(\omega), & \text{otherwise} \end{cases} \tag{9}$$

[1] In the computer implementation the FFT algorithm is used.

for some boundary frequency $\dfrac{2\pi}{n}l$, $0 < l \leq K/2$.

The methods for neume symbol segmentation and contour trace do not guarantee that the resulting contours will be placed on the same position with respect to the coordinate systems of the MWWs, with the same size, rotated at the same angle and traced from the same starting point. This is the reason why the contours must be normalized with respect to the operations of translation, scaling, rotation and change of the contour trace starting point.

The following normalization operations are used to calculate the affine invariant of the neume symbols in the frequency domain:

Translational normalization. Representing a discrete closed contour as a complex function, the zeroth harmonic of its FT corresponds to the mass center of the figure. Hence $\hat{z}(0) \equiv 0$ leads to translation of the beginning of the coordinate system of the MWW to the mass center of the symbol and thus to its translational normalization.

Scale normalization. Suppose that the contour $v(k)$ is the scaled contour $z(k)$ with coefficient α, where $\alpha > 0$:

$$v(k) = \alpha z(k), \ \ k = 0, 1, \ldots, K-1. \tag{10}$$

Then from (8) and (10), the spectrum of v will be scaled with the same coefficient:

$$\hat{v}(\omega) = \alpha \hat{z}(\omega), \ \ \omega = 0, 1, \ldots, K-1. \tag{11}$$

Hence, the invariance with respect to scaling can be achieved by division of the contour spectrum modulus of some non-zero linear combination of them. Without loss of generality it can be assumed that $|\hat{v}(1)| \neq 0$ and then the scale normalized contour $\hat{z}_s$ is given by:

$$|\hat{z}_s(\omega)| = \frac{|\hat{z}(\omega)|}{|\hat{z}(1)|}, \ \ \omega = 0, 1, \ldots, K-1. \tag{12}$$

Rotational normalization. Let $v(k)$ denote the contour $z(k)$ rotated at angle φ_0. If z and v are normalized with respect to translation in advance, then the rotation at angle φ_0 corresponds to multiplication of the complex contour by $e^{i\varphi_0}$, and:

$$v(k) = e^{i\varphi_0} z(k), \ \ k = 0, 1, \ldots, K-1. \tag{13}$$

From (8) and (13) follows that the spectrum of the contour will be rotated by the same angle:

$$\hat{v}(\omega) = e^{i\varphi_0} \hat{z}(\omega), \ \ \omega = 0, 1, \ldots, K-1. \tag{14}$$

The above shows that the rotation of the contour results in rotation of its spectrum phases and hance there are two approaches to achieving rotational invariance of the contour spectrum. The first approach is to ignore the spectrum phases which will ensure rotational normalization but also will result in loss of contour shape information.

The other approach is to normalize the spectrum phases with one of the harmonics, for example the first one, where without loss of generality it can be assumed that $\hat{z}(1) \neq 0$. Then the normalized contour with respect to the rotation $\hat{z}_r$ is given by:

$$\hat{z}_r(\omega) = \frac{\hat{z}(\omega)}{\hat{z}(1)}, \ \ \omega = 0, 1, \ldots, K-1. \tag{15}$$

Starting point normalization. Let $v(s)$ denote the contour $z(k)$ shifted by k_0 positions , where:

$$\begin{array}{l} v(s) = z(s+k_0), \ \ s = 0, 1, \ldots, n-k_0-1 \\ v(s) = z(s+k_0-n), \ \ s = n-k_0, n-k_0+1, \ldots, n-1 \end{array} \tag{16}$$

Then it can be proved (see [4]) that the relation between v and z in the frequency domain is:

$$\hat{v}(\omega) = e^{i\frac{2\pi}{K}k_0\omega}, \ \ \omega = 0, 1, \ldots, K-1. \tag{17}$$

Then the normalized contour spectrum with respect to the starting point shift $\hat{z}_m$ is given by:

$$\hat{z}_m(\omega) = \frac{\hat{z}(\omega)}{[\hat{z}(1)]^{\omega}}, \ \ \omega = 1, 2, \ldots, K-1. \tag{18}$$

3.3. FD feature vector. The feature vector $\mathcal{L}$ for each neume representative is built from the normalized version of the reduced spectrum representation of the contour (9) which are also called FDs. The FDs are ordered as elements of $\mathcal{L}$, starting with the one that corresponds to the lowest frequency, continuing in ascending order and ending with the FD which corresponds to the highest frequency, specified in (9):

$$\mathcal{L} = (FD_1, FD_{K-1}, FD_2, FD_{K-2}, \ldots, FD_{l-1}, FD_{K-l-1}), \tag{19}$$

where FD_k denotes the FD calculated from the k^{th} harmonic of the contour.

The feature vector defined in (19) is used as an index field in the database of neume representatives and is used to sort them in lexicographical order. This

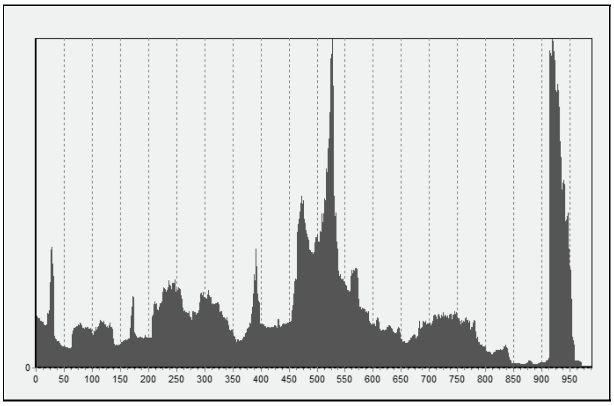


Fig. 6. Density histogram H_{dens} of the neume database containing 989 records. The accumulation interval is $\delta = 31$ and the accumulation step is $s = 1$

process groups the symbols with respect to their feature vectors, which brings the problem of unsupervised learning to a problem of one-dimensional clustering.

3.4. Sorted database histogram and histogram multi-thresholding. For the purposes of one-dimensional clustering a histogram-based approach is adopted. After the sorting of the database using the feature vectors (19) as an index field, the histogram of densities H_{dens} is calculated (see also Fig. 6):

$$H_{dens} = \frac{1}{\mathcal{L}_{j+\delta/2} - \mathcal{L}_{j-\delta/2}}, \tag{20}$$

where $\delta \in \mathbb{Z}, \delta > 0$ is the size of the accumulation interval which is chosen $\delta = \lfloor \sqrt{N} \rfloor$, N is the number of database records, and $j = 0, s, 2s, \dots, N/s$, with s being the accumulation step. At the beginning and the end of the database where the interval exceeds its boundaries, it is considered filled with empty records which are not taken into account in the calculations.

The neume classes are discovered using the multi-thresholding procedure described in [5]. This method is based on error minimization of approximation of the histogram with a preliminarily unknown number of Gaussian functions. Applied on the density histogram given in Fig. 6, this method discovers 23 classes of neume symbols. The inter-class similarity is calculated with the mean standard deviation $\sigma = 9.8591$. The quality of the clustering is given by Otsu's criterion and is calculated $\eta = 0.9985$. The mean class radius is 10.8008.

4. Conclusion.

In this paper a methodology is presented for processing, classification and recognition of Byzantine neume notation in ancient manuscripts. The problems for manuscript segmentation and structure analysis were addressed in detail because of the specific structure and characteristics of the processed documents. Also, the neume notation is treated as an unknown alphabet and from this point of view the presented approach for building of a self-learning classifier differs from the standard OCR methods.

As future work, an extension of the presented methods will be developed which will allow the interaction with an expert in the field of neume writing to evaluate and enhance the process of classifier learning. Another possible application is the development of a software instrument for similarity search in long strings of neumes in a database of digitalized manuscripts.

REFERENCES

[1] Chaker F., M. T. Bannour, F. Ghorbel. Contour retrieval and matching by affine invariant fourier descriptors. In: Proceedings of the IAPR Conference on Machine Vision Applications, May 16–18, 2007, Tokyo, JAPAN, 291–294.

[2] Cheng D., H. Yan. Recognition of handwritten digits based on contour information. *Pattern Recognition*, **31**(1998), No 3, 235–255.

[3] Dalitz C., G. K. Michalakis, C. Pranzas. Optical recognition of psaltic Byzantine chant notation. *International Journal on Document Analysis and Recognition*, **11**(2008), No 3, 143–158.

[4] Dimov D., L. Laskov. Invariant fourier descriptors representation of medieval byzantine neume notation. In: Proceedings of the 2009 joint COST 2101 and 2102 international conference on Biometric ID management and multimodal communication, BioID_MultiComm'09, Berlin, Heidelberg, 2009, Springer-Verlag, 192–199.

[5] Dimov D. T. Histogram optimal multi-thresholding. In: Proceedings of he 1st. Intern. Conf. on Application of Mathematics in Technical Sciences, American Institute of Physics Conf, 2009, 399–412.

[6] Droettboom M., K. MacMillan, I. Fujinaga. The gamera framework for building custom recognition systems. In: Symposium on Document Image Understanding, 2003, 275–286.

[7] Foley J., A. V. Dam. Fundamentals of Interactive Computer Graphics. Addison Wesley, Reading, MA, 1982.

[8] Folkers A., H. Samet. Content-based image retrieval using fourier descriptors on a logo database. In: Proceedings of the 16th Int. Conf. on Pattern Recognition, vol. III, Quebec City, Canada, August 2002, 521–524.

[9] Gezerlis V. G., S. Theodoridis. Optical character recognition of the orthodox hellenic byzantine music notation. *Pattern Recognition*, **35** (2002), No 4, 895–914.

[10] Kurita T., N. Otsu, N. N. Abdelmalek. Maximum likelihood thresholding based on population mixture models. *Pattern Recognition*, **25**(1992), No 10, 1231–1240.

[11] Laskov L., D. Dimov. Color image segmentation for neume note recognition. In :Proceedings of the Int.Conf. Automatics & Informatics'07, vol. III, 2007, 37–41.

[12] Laskov L., D. Dimov. Segmentation of ancient neumatic musical notation. In: Proceedings of the Int.Conf. Automatics & Informatics'08, vol. II, 2008, 21–24.

[13] Otsu N. A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man and Cybernetics*, **9**(1979), No 1, 62–66.

[14] Pavlidis T. Algorithms for Graphics and Image Processing. Springer, 1982.

[15] Zahn C. T., R. Z. Roskies. Fourier descriptors for plane closed curves. *IEEE Transactions on Computers*, **21**(1972), No 3, 269–281.

Lasko Laskov
New Bulgarian University
Informatics Department
21, Montevideo Str.
1618 Sofia, Bulgaria
e-mail: `llaskov@nbu.bg`

Received February 20, 2011
Final Accepted April 28, 2011