Int. J. Appl. Math. Comput. Sci., 2020, Vol. 30, No. 3, 585–596
DOI: 10.34768/amcs-2020-0043

LINE SEGMENTATION OF HANDWRITTEN TEXT USING HISTOGRAMS AND TENSOR VOTING

TOMASZ BABCZYŃSKI [a,*], ROMAN PTAK [a]

[a]Department of Computer Engineering
Wrocław University of Science and Technology
Wybrzeże Wyspiańskiego 27, 50-370 Wrocław, Poland
e-mail: {tomasz.babczynski,roman.ptak}@pwr.edu.pl

There are a large number of historical documents in libraries and other archives throughout the world. Most of them are written by hand. In many cases they exist in only one specimen and are hard to reach. Digitization of such artifacts can make them available to the community. But even digitized, they remain unsearchable, and an important task is to draw the contents in the computer readable form. One of the first steps in this direction is to recognize where the lines of the text are. Computational intelligence algorithms can be used to solve this problem. In the present paper, two groups of algorithms, namely, projection-based and tensor voting-based, are compared. The performance is evaluated on a data set and with the procedure proposed by the organizers of the *ICDAR 2009* competition.

Keywords: document image processing, handwritten text line segmentation, projection profile, text string, off-line cursive script recognition, *ICDAR 2009* competition.

1. Introduction

Image processing is one of the important tasks of computational intelligence. Medical images, photos of objects and people or information forming scenes for robots are processed most often. Various manuscripts and paper documents, including historical primary sources, are also examined frequently. There are various aims in document processing. Obviously, the recognition of the manuscript text is the main purpose. In this case, algorithms focus on obtaining the contents of handwritten text. Another task related to the manuscript is the writer identification. The writer identification is often carried out for the forensic purposes.

Among numerous tasks in the process of script recognition, it is important to perform proper text line localization and text line segmentation. The localization is often the first stage of text line segmentation. In turn, the line segmentation often precedes segmentation into words. The text may be further divided into letters and then the actual recognition can be performed. The last segmentation is omitted in some kinds of algorithms (e.g., hidden Markov models) but the line segmentation is performed in all cases.

In the present article, we focus on the problem of the text lines segmentation. Many segmentation algorithms, which achieve better or worse accuracy, are described in the literature. None of them are perfect, so there is a need to invent new procedures, to improve the existing ones, or to tune the parameters of the algorithms to enhance their power. Furthermore, many papers do not discuss the segmentation stage in sufficient detail. The purpose of the present article is to verify the behavior of some existing algorithms and to improve them. Our attention is focused on two specific procedures belonging to the accumulating type. The first algorithm is projection based and is a result of our previous work (Ptak *et al.*, 2017). Here it is examined to what extent the results may be improved by tuning the parameters and by introducing a different labeling scheme. The second algorithm is based on tensor voting and is a modification of the one by Nguyen Dinh and Lee (2011). We want to verify its sensitivity to the parameter changes and to the alteration of the shape of the voting kernel. The aim is to identify the drawbacks of the evaluated algorithms and to determine the direction of their future development.

In many language systems the text is horizontally

*Corresponding author

aligned. The text is running from left to right or right to left. Then the block of lines runs from top to bottom. However, many Asian languages can be written both vertically and horizontally. There may also be quite different cases, e.g., a diagonal script. Here we concentrate on the case of scripts where text runs horizontally from left to right such as in Latin, Greek and similar writing.

The horizontal projection method consists of calculating, for each horizontal line of pixels, the number of foreground pixels. In such a data representation—the horizontal projection profile diagram of an image (also presented as the histogram)—the amount of information is reduced. The histograms represent density distributions of handwriting. It is possible to perceive areas of letter concentration. In the basic case a thresholding is used to separate text lines.

Tensor voting (TV) is an example of accumulation methods in image processing. TV is based on the tensor representation of image features and non-linear voting. In the problem of line segmentation, an initial tensor field is built from points close to central points of connected regions of foreground pixels. TV is applied to obtain a set of points more likely belonging to actual lines. This set is then used to construct line chains.

2. Related work

The problem of text lines segmentation was investigated using various approaches. Likforman-Sulem *et al*. (2007) presented a good survey of the methods applied to solve this issue. Comparison of algorithms can also be found in the work of Razak *et al*. (2008). A newer comprehensive survey of mostly document segmentation algorithms can be found in the paper by Eskenazi *et al*. (2017). In turn, Naz (2015) presented an overview of segmentation techniques for Arabic-like scripts.

Generally, two strategies are used in segmentation: top-down and bottom-up. Hybrid methods combining both strategies are also used. Some of the top-down methods can be categorized as accumulating data or voting.

A fast but rather simple method is global horizontal projection of the pixels and analyzing the resulting histogram. Methods of this kind are quick but not very smart. One of them can be found in the paper by Ptak *et al*. (2017). The authors proposed a text line segmentation algorithm based on the projection profile with a variable threshold. The threshold in the method was adaptively tuned and was different for each peak being proportional to its height.

Methods of this kind usually maintain the global projection profile but there are also methods applying the piece-wise projection profile of the document (Arivazhagan *et al*., 2007). In this case the accumulation range is smaller. The method proposed there is robust to handwriting documents with lines running into each other.

The projection profile methods accumulate data from pixels along a given direction. In the Hough transform, the pixels or the center points of connected components are accumulated in many directions in the whole image. In TV the accumulation is performed more locally. These procedures will be explained below.

An interesting approach, based on the type of local data accumulation, has been proposed by Kennard and Barrett (2006). The core of the approach is calculating and binarizing a "black/white transition count map". Next, the connected components in the resulting binarized map are analyzed and split using a min-cut/max flow graph cutting algorithm (Boykov and Kolmogorov, 2004). This separates connected lines of text improving segmentation accuracy.

For a printed document, smearing can be applied. An example of this type of algorithm is the run-length smoothing algorithm (RLSA) (Wong *et al*., 1982). The black pixels, representing foreground in the binary image of the document, are linked together along the horizontal direction if their distance is below a predefined threshold. The direction of smearing should be consistent with the direction of the line of text. A variant of this method adopted to gray level images is described by LeBourgeois (1997). There are also modifications of the RLSA algorithm used for handwriting recognition (Sarkar *et al*., 2011).

In document image analysis, morphological filters have been also used for the segmentation. Methods of this kind can be categorized as bottom-up. For example, Wu *et al*. (2008) describe a text line extracting method from cluttered images. It was applied to printed texts and defined a novel set of morphological operations—a combination of closing, opening, differentiating, etc. Thanks to them, important contrast regions with possible text are extracted. The contrast feature is robust to lighting changes and invariant against image transformations.

In the paper by Papavassiliou *et al*. (2010), a method based on binary morphology was proposed for handwritten documents. This method uses the morphological dilation and opening operations. The dilation is applied to determine text line components through joining regions that are close to each other or horizontally overlapping. The structuring element of the opening operation is chosen in a way preventing merging in the vertical direction.

The so-called "water flow" algorithm of text line segmentation was proposed by Basu *et al*. (2007). It assumes that hypothetical water flows from both sides of the image area. The stripes of areas left "unwetted" in the image are labeled for extraction of text lines. This algorithm was extended and further improved; see the works of Brodić and Milivojević (2011) or Brodić

(2012; 2015) for details.

Feldbach and Tönnies (2001) proposed an algorithm of baselines and centerlines recognition. The method is based on local minima detection of connected components. The text line is recognized progressively from line segments. This algorithm can deal with close or even touching lines.

Probability theory is also used to solve the problem of the segmentation of handwritten documents. The algorithm described by Li *et al*. (2008) presents a robust approach based on probability density estimation. For a document image, a map of probability that the underlying pixel belongs to a text line, is estimated.

Artificial neural networks are alterative computing systems used to recognize lines in the text. Segmentation of handwritten document image into lines applying a fully convolutional neural network is presented in the paper by Vo *et al*. (2018). Using artificial neural networks is an example of a deep learning algorithm.

The concept of the Hough transform is employed in the field of document analysis for many purposes such as skew and slant detection and text line segmentation (Likforman-Sulem *et al.*, 1995; Louloudis *et al.*, 2008; 2009; Alaei *et al.*, 2011). The Hough transform is a popular technique for finding straight elements in images. It can be used to determine the slope of elements. Pixel- and block-based Hough transforms can be employed to the task of text lines segmentation (Louloudis *et al*., 2008). The Hough transform based methods can cope with documents with variations in the skew between lines (Likforman-Sulem *et al*., 1995; Pu and Shi, 1999). Pach and Bilski (2014) proposed a robust method for the text line segmentation of medieval overlapping text. In this procedure, bounding boxes of the connected components of foreground pixels are divided into smaller parts. As a result, the nonrectangular zones between the lines are detected, better reflecting the askew text. This method uses projections and the Hough transform in its operation.

Although the Hough transform can find straight lines in the document, it works globally finding artifacts which are not actual lines of text. Better results are obtained by more locally working methods like, e.g., TV. This method has another advantage—it can find not only straight lines but also second order order curves.

In the paper by Han *et al*. (1997), TV was used to estimate nonuniform skewness of text lines of printed text. The double voting is performed on the centroids of the connected components. This method performs very well on documents with clearly separated letters. Unfortunately, handwritten text is treated incorrectly by that method.

Nguyen Dinh *et al*. (2010) as well as Nguyen Dinh and Lee (2011) adapted the algorithm for recognition of text lines in handwritten text. The algorithm is based on 2D TV. Zhang and Lee (2011) present the application of the procedure to Chinese handwritten script.

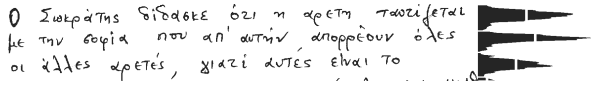


Fig. 1. Original text with histogram and thresholds levels.

3. Line segmentation

In the present paper we compare the performance of the projection based algorithm with the grouping algorithm which uses TV. Sections 3.1 and 3.2 contain an overview of these methods.

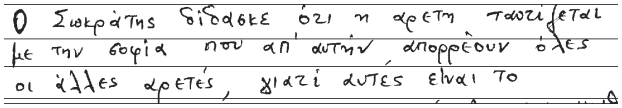
3.1. Projection algorithm. The algorithm works as follows (see Algorithm 1). We start from the binary image I_{in}. The horizontal projection profile H is counted and filtered using a moving average filter. All points of the projection profile are sorted by their values (Steps 1 and 2). The points are processed in a descending sequence starting from the point which has the maximum value. For each point the width of the peak to which it belongs is determined at a certain height. Its value in proportion to the height of the peak h_p is equal to the threshold $t_a = t \cdot h_p$. The parameter $t \in (0, 1)$ is global in the algorithm. The width of the peak is defined as the size of the range of arguments having values greater than the threshold t_a (Line 6). Results of these steps are presented in Fig. 1. If the range R does not overlap any of the previously determined ranges, it is accepted as a text line and added to the set P (Line 7). Otherwise, it is rejected to prevent the connection of overlapping ranges which would cause recognition of two or more text lines as one. All arguments in the range R are marked as checked (Line 8). The process terminates when the value of a given point is less than $\alpha = 0.1$ of the maximum value of the diagram (Line 3). Details can be found in the paper by Ptak *et al*. (2017).

In the present paper we compare two variants of the final part of the algorithm—line identification and labeling:

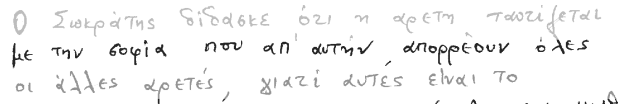
- using the borders,
- using the line chains.

The first of them was used in the cited paper while the other is introduced here to compare the performance.

3.1.1. Using borders. All ranges found in set P correspond to text line and the minimum values of the regions between them are adopted as text lines separators (Line 11). Then the labeling is performed. Each foreground pixel lying vertically between the separation lines i and $i + 1$ get the label i. The separators and the labeled image are shown in the output image I_{out} in Fig. 2.

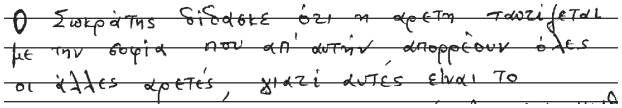


(a)

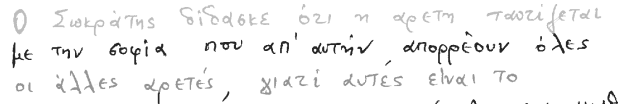


(b)

Fig. 2. Borders between lines of text (a), text labeled using the borders (b).



(a)



(b)

Fig. 3. Line chains of text (a), text labeled using the line chains (b).

3.1.2. Using line chains. The center position of each line of text is obtained as a mean value between separator lines enclosing the text line. The horizontal line of the length equal to the width of the image is placed at the position. Each of them is assigned an individual number.

The labeling in this strategy is performed in the following way. The connected regions in the image are identified and analyzed one by one. If the region is touched by one line then all points in this region are labeled with the number of this line. In the opposite case (i.e., none of the lines or more than one line touch the region) each point in the region is labeled with the number of the nearest (using the Euclidean distance) line.

3.2. Segmentation with tensor voting.

3.2.1. Tensor voting. Tensor voting (TV) was first introduced by Lee and Medioni (1997). Since then it has been applied in many fields of pattern recognition giving very good results. The fundamentals of that way lie in the psychological Gestalt perception rules which state that humans tend to perceive shapes where only some points are seen. The TV method is an attempt to teach the computer how to join points in the image into shapes in much the same way as the humans do. The method belongs to the accumulating methods group and is somewhat similar to the Hough transformation. The differences are that TV works locally using the decay function and that TV can find not only straight

Algorithm 1. Algorithm with the histogram and a variable threshold.

Input: I_{in} {binary image}
{**Parameters:**
t: relative threshold within the interval (0, 1) }
Output: I_{out} {segmented image}

1: Count and filter the projection profile H of I_{in} in the horizontal direction
2: Sort elements of H in descending order of H values
3: **for all** value of $H > 0.1 \max(H)$ **do**
4: **if** (current element is not checked) **then**
5: Count adaptive threshold t_a related to t
6: Determine the range R of the width of the peak
7: Add R to the set of intervals P corresponding to peaks
8: Add range R to checked points
9: **end if**
10: **end for**
11: Determine the set S of separators between text lines
12: Labeling {Two variants of labeling were used}
13: **return** I_{out}

lines but the second order curves. The TV method uses second order symmetric, nonnegative defined tensors as the fundamental portion of the data. Each tensor can be defined as

$$T = \begin{bmatrix} \vec{e_1} & \vec{e_2} \end{bmatrix} \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} \begin{bmatrix} \vec{e_1}^T \\ \vec{e_2}^T \end{bmatrix} \tag{1}$$

and, when applied to the 2D space, is stored in the form of the 2×2 symmetrical matrix. The eigenvectors $\vec{e_1}$ and $\vec{e_2}$ form the orthonormal basis of the tensor, while the eigenvalues λ_1 and λ_2 are the real numbers which can be interpreted as sizes in the directions of eigenvectors. It is assured that $\lambda_1 \geq \lambda_2$ and both are nonnegative. The tensor can be decomposed into two orthogonal parts

$$T = \lambda_1 \vec{e_1}\vec{e_1}^T + \lambda_2 \vec{e_2}\vec{e_2}^T, \tag{2}$$

but in the TV method it is decomposed into the stick and ball parts (3). According to this decomposition the first term of

$$T = (\lambda_1 - \lambda_2)\vec{e_1}\vec{e_1}^T + \lambda_2(\vec{e_1}\vec{e_1}^T + \vec{e_2}\vec{e_2}^T) \tag{3}$$

is called the *stick tensor* while the second term is the *ball tensor* and the actual tensor is their linear combination.

The value λ_2 is called the *ballness* or *anisotropic saliency* and encodes the junctions or the noise. The quantity $\lambda_1 - \lambda_2$ is called the *curve saliency* or the *stickness* and represents the certainty that the line indeed runs through the given point and its direction is normal to the vector $\vec{e_1}$. A tensor having $\lambda_1 = \lambda_2$ is called a *pure ball tensor* while one with $\lambda_2 = 0$ is termed a *pure stick tensor*.

Often the tensor is presented graphically in a 2D space as an ellipse whose main axes have lengths proportional to the eigenvalues. This representation and the stick/ball decomposition are shown in Fig. 4. Due to the geometrical representation of a tensor we can call λ_2 of a pure ball tensor, its radius, and λ_1 of a pure stick tensor its length.

In the described method, the initial image is first encoded as a tensor field. Tensors in this field are called *tokens*. The actual procedure of such an encoding depends on the problem to be solved. In our case only vertical unit stick tensors are used as described in the next section.

After generating the initial tensor field, voting is performed. Each token votes on its neighborhood—either the other tokens in the *sparse voting* or all positions in the *dense voting*. In this paper only sparse voting is applied.

Vote direction. Now, examine the two tokens: **O**, the *voter*, and **P**, the *votee* (see Fig. 5). The fundamental assumption in TV is that the most likely smooth curve going through the two points coincides with the osculating circle. In our example it is the arc s. If the tensor point **O** is purely the stick one with the only nonzero eigenvector $\vec{U}$ (called in short the direction of the tensor), so the direction of the stick tensor at the votee position is also normal to the arc s.

Vote strength. The strength of the vote depends on the distance between the positions and the angle between the tensors. There are some variants of TV. All of them use the exponential decay with the distance but the penalization of the curvature is treated in different and so is distance measuring. Only two variants used in the following experiments will be described.

In the original tensor voting (OTV) the distance is measured as the arc length and the deviation from the straight line is penalized using the curvature value κ in accordance with

$$\mathrm{DF}(l|\sigma) = \exp\left(-\frac{s^2 + c\kappa^2}{\sigma^2}\right). \tag{4}$$

Here $s = 2\Theta r$ is the arc length between the analyzed two points, $\kappa = 1/r$ is the curvature, $r = l/(2\sin\Theta)$ is the radius of the osculating circle, c is the constant calculated by the authors of OTV. The value σ is the only free parameter of the method and is called the *scale of voting*. In the original formulation the tensors for angles $\Theta > 45^\circ$ are cropped.

The voting by steerable filters (STV) is defined by Franken *et al.* (2006). In this variant the Euclidean distance is used, the power of the trigonometric function replaces the component with the curvature and no cropping is applied. In the formulation of STV the cosine is used because the authors expressed this as the tangent tensors instead of the normals and therefore the decay function looks like

$$\mathrm{DF}(l|\sigma) = \exp(-\frac{l^2}{2\sigma^2})\cos^{2n}(\Theta). \tag{5}$$

Here l is the Euclidean distance between taken positions, σ is again the scale of voting, and n is the parameter used for curvature penalization and have the value of 2 as proposed by Franken *et al.* (2006). All these parameters are shown in Fig. 5. It is noticeable that STV is defined only for stick voting and there is no ball voting. Another constraint is that it is defined only in a 2D space. Both constraints have no importance in the application presented here because only stick voting in 2D space is used. The big advantage of this formulation is its speed, the voting is very fast in comparison to the OTV.

An example of stick voting kernels of the two TV variants with the same value σ is shown in Fig. 6. In both kernels only pure stick tensors appear so they can be displayed as if they were vector fields. The short lines show the orientations of tensors, their lengths are proportional to the λ_1 values. The orientations of the tensors are the same but both the kernels differ in tensors lengths because of different decay functions. Along the horizontal symmetry axis, the lengths decrease twice faster in OTV than in STV, while outside the axis they decrease faster in the case of STV.

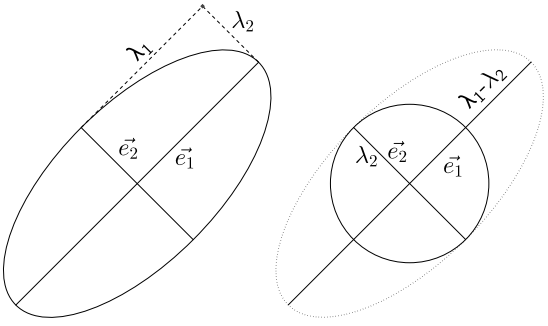


Fig. 4. Tensor decomposition.

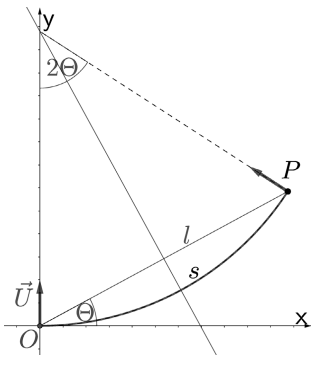


Fig. 5. 2D stick vote between two tokens.

(a) (b)

Fig. 6. Voting kernel for original TV (a) and steerable filters (b).

Nguyen Dinh and Lee (2011) used another variant, namely, the closed form solution introduced by Wu *et al*. (2012). This version has a decay function similar to that of STV.

The reader interested in the details of the OTV formalism can find more information in the books Medioni and Kang (2004) or Mordohai and Medioni (2006). The current state-of-the-art in the TV domain is well presented by Maggiori *et al*. (2014), who show some variants of the formalism. The survey is five years old, but it can still be considered up-to-date. Many papers concerning the utilization of TV have been published since then, but only in few of them the core of the method was touched. Probabilistic TV, introduced in 2012 and mentioned in the review, was actually developed later. The closed form solution had been challenged as incorrect and therefore it was only noted in the survey, but it was finally proven in 2016 to be valid.

3.2.2. Outline of the algorithm. In our research we generally follow the procedure shown by Nguyen Dinh *et al*. (2010) as well as Nguyen Dinh and Lee (2011) but with some modifications pointed out later. The main steps are shown in Algorithm 2.

In this algorithm we start again from the binary image. Our paper does not address the problem of proper binarization of gray scale or color images.

Step 1 is designed to obtain some parameters of the current picture such as the average height ($\bar{H}$) of the line and to produce a starting set of characteristic points used later to produce the initial tensor field. The task is done by performing the dilation with the horizontal line of length $\bar{H}$ as the structuring element. Next, the erosion with the horizontal line of the length $\bar{H} + \theta$ is performed. The parameter θ is equal to 10 in the cited paper but our evaluation showed that the value 6 gives the same final result but has a rationale—it is the average stroke width in the investigated documents. After the morphological operations the picture is cut into smaller pieces with vertical lines. The distance between them is $\bar{H}/2$. After the cut, the centroids of the connected components are calculated giving the sparse image of the text layout. Partial results of this step are presented in Fig. 7.

Next, *Step 2* is taken for token generation. The token is a tensor of the unit length and vertically oriented. Tensors are placed at the positions of centroids from the previous step. The vertical orientation is used to encode the silently made assumption that lines of text are approximately horizontal. The convention is used that tensor direction is normal to the explored feature direction.

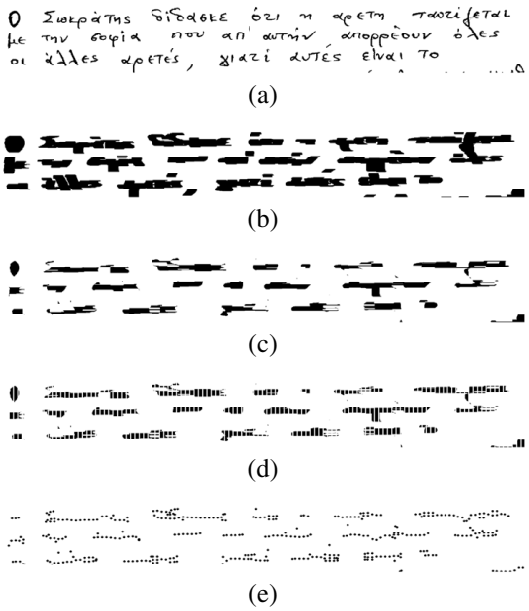
Afterwards the actual TV (described in Section 3.2.1) is performed, cf. *Step 3*. The voting is sparse—each token votes only for each other, not at each point of the image. The goal of this step is to find tokens which do not match the others and, in general, do not fit the assumed model of parallel horizontal lines.

The tokens identified as outsiders are eliminated in

Algorithm 2. Main steps of the TV algorithm.

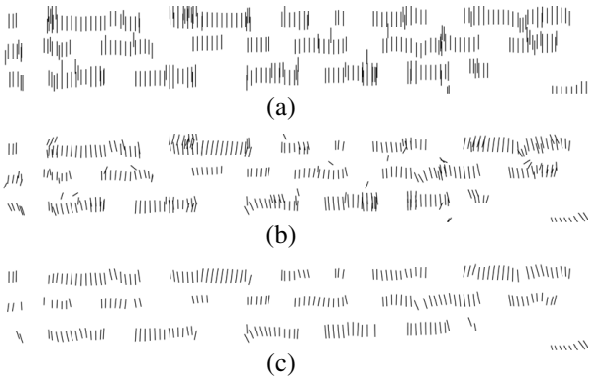
Input: I_{in} {binary image}
{**Parameters:**
σ: scale of the TV
ω: relative strength of feature tensor }
Output: I_{out} {segmented image}

1: Pre-processing;
2: Token generation;
3: Tensor voting (σ);
4: Removing outliers (ω);
5: Line chain generation;
6: Labeling;
7: **return** I_{out}



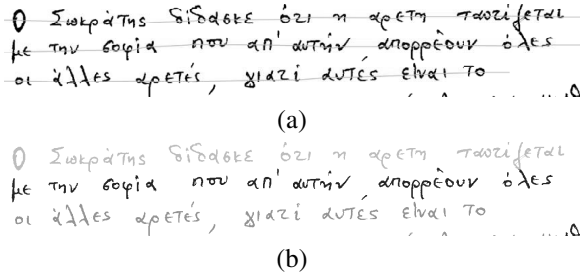
(a) (b) (c) (d) (e)

Fig. 7. Pre-processing of the given text into centroids: initial text (a), after dilation (b), after erosion (c), cut image (d), centroid positions (e).



(a) (b) (c)

Fig. 8. TV results: initial tensor field (a), tensor field after voting (b), after removing outliers (c).



(a) (b)

Fig. 9. Line chains of text (a), labeled text using the line chains (b).

Step 4. The outsider is the tensor which is too horizontal or too short. Both conditions are taken globally, the direction is compared with $\pi/4$, and the length (actually the stickness) with the mean value of that of all tensors. Formally, denote by $\vec{U}$ the vertical unit vector and by $\vec{X}$ the normal stickness vector ($\vec{X} = (\lambda_1 - \lambda_2)\vec{e_1}$) at a given point. Those tensors for which at least one of conditions

$$\begin{cases} |\vec{X}| < \omega \cdot \mathrm{mean}(|\vec{X}_i|), \\ |\cos(\vec{X}, \vec{U})| < \cos(\pi/4) \end{cases} \tag{6}$$

is fulfilled are removed. Here $\mathrm{mean}(|\vec{X}_i|)$ is the arithmetic mean of stickness values of all tensors and ω is a parameter of the algorithm. Nguyen Dinh and Lee (2011) took $\omega = 0.65$ but in our experiments also different values were taken (see Section 4). Additionally, all tokens but the maximal one in the moving window $2\bar{H} \times \bar{H}/2$ are removed in this step.

After the outlier removal is finished, the tensor field is no longer needed. Only the origins of tensors are saved as points for the next steps. A set of polygonal chains is constructed from these points as described by Nguyen Dinh and Lee (2011). In short, primarily we look for points from the left edge of the image. Next, points are selected in the moving window of the size $\bar{H} \times \sigma$, where σ is the voting scale, such that the nearest point in the vertical direction is taken. The procedure finishes at the right edge of the image or if there are no points in the window. After that the removing of excess lines and merging the continuations is carried out.

Next the labeling is performed in the way similar to that described in Section 3.1.2 as the line chain variant. The only difference is that now the polygonal chains are used instead of horizontal lines. The connected regions in the image are identified and analyzed one by one. If the region is touched by one polygonal chain then all points in this region are labeled with the number of this chain. In the opposite case (i.e., none of the chains or more than one chain touch the region) each point in the region is labeled with the number of the nearest (using the Euclidean distance) chain. The results of this step is shown in Fig. 9.

Finally the labeled image is compared with the manually annotated image—ground truth as described in Section 4.

4. Experiments

4.1. Data set and evaluation methodology. The handwritten documents to evaluate were taken from the materials of the challenge *ICDAR 2009: Handwriting Segmentation Contest* accompanying the *ICDAR 2009* conference (Gatos *et al*., 2011). The data set contained 200 one-page handwritten documents in four languages (English, French, German and Greek) written by many

writers. The images were binary ones. The challenge had two parts—line segmentation and word segmentation. We used only data of the former. Each document was manually annotated by the organizers of the competition to make the *ground truth* which was used to evaluate the participants' results. Each pixel of the image got a label informing to which line it belonged.

The evaluation of the result, during the challenge, was based on the MatchScore table using the one-to-one matching defined by Phillips and Chhabra (1999). A detailed description of the performance evaluation is contained in the postcompetition report (Gatos *et al.*, 2011). In our previous paper (Ptak *et al.*, 2017) we found the method as not adequate to grade the horizontal histogram algorithm and algorithms compared with it. The reason was the lack of annotated handwritten documents in the Polish language on which we tested the algorithms. Here, however we use the metric applied during the cited competition also to compare our previous results with the current results and the results obtained by Nguyen Dinh and Lee (2011) using documents from the *ICDAR* competition.

The MatchScore table

$$\text{MatchScore}(i,j) = \frac{T(R_i \cap G_j \cap I)}{T((R_i \cup G_j) \cap I)} \tag{7}$$

is constructed as follows. Let I be the set of foreground pixels in the image, R_i the set of pixels recognized as belonged to the i-th class, G_j the set of pixels in the j-th class of the ground truth. $T(s)$ is a function giving the number of elements in the set s. The MatchScore table takes values in the range $[0, 1]$.

Line i is treated as a one-to-one match with the ground truth line j only if $\text{MatchScore}(i,j)$ is greater then the threshold $T_a = 0.95$. That value was accepted during the *ICDAR* challenge. Let M be the number of recognized lines, N the number of lines in the ground truth, and $o2o$ the number of one-to-one matches. The detection rate (DR) and recognition accuracy (RA) metrics are defined

$$DR = \frac{o2o}{N}, \quad RA = \frac{o2o}{M}, \\ FM = \frac{2 \cdot DR \cdot RA}{DR + RA} \tag{8}$$

along with the aggregated value FM which was used to range applications during the competition.

4.2. Experimental results. In the experiments we compared the performance of the algorithms described in Sections 3.1 and 3.2 on the data set depicted in the previous section. Both algorithms have two variants. The projection algorithm labels lines using the border strategy (ProjB) described in Section 3.1.1 or center line strategy (ProjL) from Section 3.1.2. The TV algorithm uses original voting (OTV) or voting by steerable filters (STV). The results of the comparison can be found in Table 1 among of the attendees of the *ICDAR* competition. The results obtained by the algorithms evaluated in the present paper are marked in boldface. Additionally, two algorithms not attending the challenge are presented, one from the paper by Nguyen Dinh and Lee (2011), marked as CLTV and the *Projections* algorithm presented in the post competition report as a reference algorithm. They are marked in italics.

Table 1. *ICDAR 2009* results.

	DR[%]	RA[%]	FM[%]
CUBS	99.55	99.50	99.53
CLTV	*99.58*	*99.31*	*99.44*
ILSP-LWSeg-09	99.16	98.94	99.05
PAIS	98.49	98.56	98.52
CMM	98.54	98.29	98.42
OTV	**97.61**	**98.34**	**97.97**
STV	**97.25**	**98.07**	**97.65**
CASIA-MSTSeg	95.86	95.51	95.68
PortoUniv	94.47	94.61	94.54
PPSL	94.00	92.85	93.42
LRDE	96.70	88.20	92.25
Jadavpur&Univ	87.78	86.90	87.34
ETS	86.66	86.68	86.67
AegeanUniv	77.59	77.21	77.40
ProjL	**81.45**	**73.67**	**77.37**
ProjB	**75.28**	**54.12**	**62.97**
Projections	*62.92*	*57.80*	*60.25*
REGIM	40.38	35.70	37.90

The projection based algorithm has one free parameter[1] i.e., the relative threshold $t \in [0, 1]$. During experiments, this parameter was varied from 0.1 to 0.9. The results are shown in Fig. 10. It is noticeable that the DR metric is close to a constant in the wide range of the parameter t for both labeling strategies. This means that also the number of perfect matches ($o2o$) is considerably independent of the threshold because the true number of lines $N = 4034$ is constant.

For the border strategy of labeling (see Fig. 10(a)) the RA value is maximal for the smallest values of the threshold ($t = 0.2$ yields $RA = 75.28\%$) and decreases for greater values. Such a result comes from the fact that for greater threshold values the number of recognized lines (M) is greater. Unfortunately, they are incorrectly recognized because for greater M values, $o2o$ remains at the same level. Here we can see that the RA metric is not appropriate for this algorithm because its relatively large

[1] In the paper by Ptak *et al.* (2017) also the second parameter was varied, namely, the window length of the histogram smoothing, but here only the optimal value found there is used.

values do not reflect the quality of the recognition. In fact, many of the recognized lines are identified correctly but the total number of well matched lines is small. The DR metric gives the quality of recognition more accurately in this case. DR gets the greatest value for $t = 0.6$ yields $DR = 57.26\%$. Taking the formal rules of the challenge, the maximal value of the FM is taken into account.[2] That value is $FM = 62.97\%$ for $t = 0.2$.

The center line labeling (Fig. 10(b)) gives better results than the previous strategy. Here also we can see that for the average values of t, the RA metric is greater than DR. This means that the algorithm recognizes fewer lines than their actual number in the document. The metric value of $RA = 83.78\%$ for $t = 0.4$ while the $DR = 74.42\%$ is maximized for $t = 0.7$. The average metric FM reaches the maximal value $FM = 77.37\%$ for $t = 0.5$.

This result would give our algorithm the glorious last but one position during the competition regardless of the labeling strategy. But still the result obtained here is better than that referenced as *Projections* by the challenge organizers. That algorithm was used as the version state-of-the-art example and was similar to *ProjB* presented here but with a constant threshold instead of the adaptive one.

The line segmentation algorithm using TV (Section 3.2) has many free parameters. The most obvious one is the scale of the voting kernel, σ. This parameter must be big enough to traverse over the letters and words forming the lines, but small enough so as not to bind the neighboring lines. The next parameter treated as a variable by Nguyen Dinh and Lee (2011) is ω, the threshold of the tensor saliency used during outsider removal. The other parameters such as the size of the structuring element during dilation and erosion, the angular threshold during outsider removal, the size of the window used in line string extraction or in string merging are treated as constants with values obtained from the cited paper.

The results of the algorithm applied to the data set described above are shown in Fig. 11. The *ICDAR* metric values are presented as contour plots of ω and σ. The dashed lines on contour plots show the parameters for which the best results are obtained. Panels (c) and (f) visualize the sections of these charts for the best σ values additionally showing the FM metric. Panels (a)–(c) refer to the original TV while (d)–(f) voting by steerable filters.

It can be seen that the performance of the algorithm is sensitive to changes in both the parameters. In a wide range of their values the ones are good but the best results are obtained only for precisely selected values. Especially for the $\sigma < 80$ the performance decreases dramatically because voting cannot traverse gaps between words.

[2]It is not a fair comparison because that maximal value is taken from the testing data without prior training on part of the data set, but we are interested in trends rather than the position in ranking.

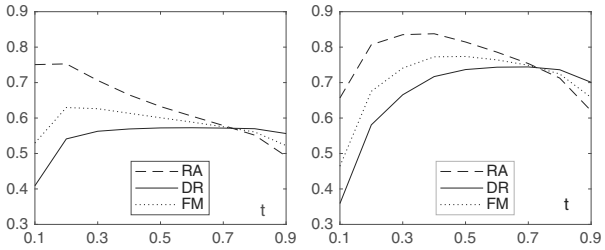


Fig. 10. Results evaluated using the projection algorithm with the border strategy (a) and the center line strategy (b).

Table 2. Labeling time of all *2009 ICDAR* documents using different algorithms.

	sec	min:sec
ProjB	26	00:26
ProjL	63	01:03
STV	880	14:40
OTV	2589	43:09

In this kind of algorithms all metrics imitate each other. In the whole range of parameters we have $DR > FM > RA$. This means that M, the number of detected lines, is greater than the actual number of lines but does not change much with changes in parameters. The best values of all metrics are observed for the same combination of σ and ω. This happens for both the variants of the algorithm. For the original voting, the metrics are maximized for $\sigma = 100, \omega = 0.58$. For the voting with steerable filters, the best results are obtained for $\sigma = 90, \omega = 0.54$. The optimal values of the metrics are shown in Table 1.

It is also interesting to compare the execution times of all compared algorithms. All of them were implemented in MATLAB 2018a and run under Linux on a machine with the Intel i7 4 core HT processor and 16 GB of DDR4 RAM. The clock was 2800 GHz and the time of processing (recognition and evaluation) of all 200 *ICDAR* documents was measured. The results are listed in Table 2. Nguyen Dinh and Lee (2011) claim that their algorithm needs "less than 10 seconds" for a document but on different hardware. We found that MATLAB scripts run 2 to 3 times faster on our computer than on the one comparable with theirs. This means that their algorithm would take less than 1000 seconds of processing the whole data set on our computer, i.e., the time similar to our *STV* algorithm. The projection based algorithms are way better in this comparison. They give decent results, but in a very short time, which may be meaningful in some applications.

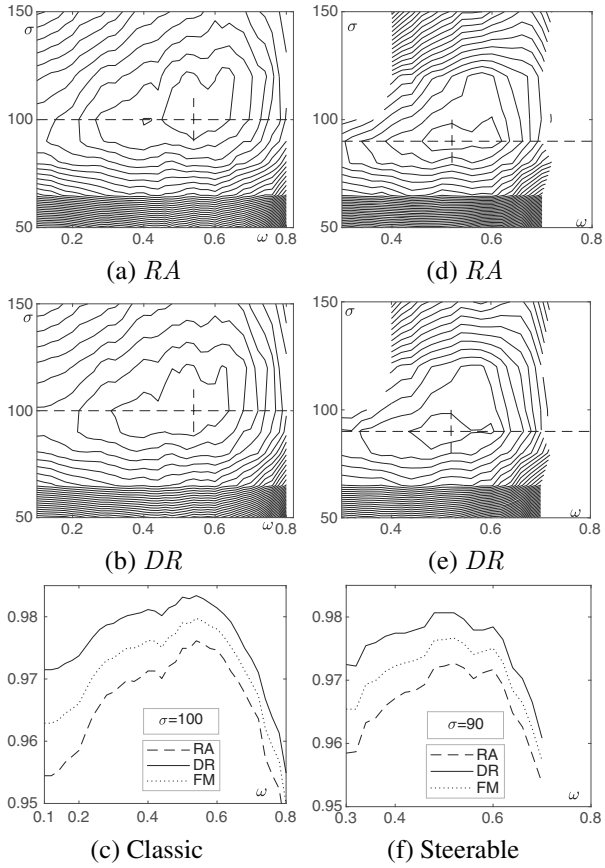


Fig. 11. Results evaluated using TV (original (a)–(c), by steerable filters (d)–(f)), with $\omega = 0.1 \div 0.8$ and RA for $\sigma = 50 \div 150$ ((a) and (d)), DR for $\sigma = 50 \div 150$ ((b) and (e)), profile for $\sigma = 100$ (c), profile for $\sigma = 90$ (f).

5. Conclusions

In the present paper, a comparison of two methods of line segmentation and labeling in handwritten documents is made. During experiments strengths and weaknesses of the algorithms were discovered. It was shown that the methods need a number of improvements. Some advice to developers is summarized below.

The TV-based solution proved its superiority in the studied case ($FM{\approx}98\%$). Our results were, however, slightly worse than those presented by Nguyen Dinh and Lee (2011). Different forms of TV used in our approach were unlikely to be the cause for the observed discrepancy. It was rather caused by an imprecise description of the algorithm used by Nguyen Dinh and Lee (2011) that resulted in the impossibility of its reconstruction. In addition, the procedure has many hidden parameters whose values may depend on the shape of the voting field. Analysis of the unsuccessful cases shows that a slight change in some of them can improve the segmentation of the particular document while spoiling that of another. The developer's attention should be directed to decreasing the number of such parameters or binding them with easily measurable properties of the analyzed document. Comparing the segmentation time with the accuracy for both the evaluated versions of the algorithm, we can state that the use of any novel calculation scheme of the voting is encouraged. The use of the closed form or voting by means of steerable filters causes a triple increase in the speed while the quality of the segmentation remains almost unchanged.

Both the compared versions of the projection-based algorithm presented here gave mediocre results ($FM{\approx}63\%$ or 77%). The modification of the line identification and labeling parts of the procedure improved it by 14% but at the cost of a triple slowdown. The improvement is significant but the achievement can hardly be called excellent. The relatively small improvement in the projection based algorithm may indicate its limitations. Projection across the entire horizontal line of pixels cannot separate slanted lines of text. A further improvement of this algorithm is likely to give only a limited enhancement of the results. Although the effect is not so good as in the previous case, this algorithm is really fast. In some applications, speed is the crucial demand while accuracy is less important. If exactness is more significant, more substantial modifications like projection in other directions or piece-wise profiles should be introduced.

It is shown here that both the groups of algorithms ought to be improved to give better results. Some possible directions for future work have been indicated. Another option for a TV based procedure is to take advantage of dense voting which is very ineffective in OTV but can be easily and efficiently performed by steerable filters. Also the line identification algorithm could be designed in a way closer to that of the spirit of TV thanks to the dense voting result. All ideas presented here deserve further investigation and will be examined in the future.

References

Alaei, A., Nagabhushan, P. and Pal, U. (2011). Piece-wise painting technique for line segmentation of unconstrained handwritten text: A specific study with Persian text documents, *Pattern Analysis and Applications* **14**(4): 381–394.

Arivazhagan, M., Srinivasan, H. and Srihari, S. (2007). A statistical approach to line segmentation in handwritten documents, *Document Recognition and Retrieval XIV* **65000**: 245–255.

Basu, S., Chaudhuri, C., Kundu, M., Nasipuri, M. and Basu, D.K. (2007). Text line extraction from multi-skewed handwritten documents, *Pattern Recognition* **40**(6): 1825–1839.

Boykov, Y. and Kolmogorov, V. (2004). An experimental comparison of min-cut/max-flow algorithms for energy

minimization in vision, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **26**(9): 1124–1137.

Brodić, D. (2012). Extended approach to water flow algorithm for text line segmentation, *Journal of Computer Science and Technology* **27**(1): 187–194.

Brodić, D. (2015). Text line segmentation with water flow algorithm based on power function, *Journal of Electrical Engineering* **66**(3): 132–141.

Brodić, D. and Milivojević, Z. (2011). A new approach to water flow algorithm for text line segmentation, *Journal of Universal Computer Science* **17**(1): 30–47.

Eskenazi, S., Gomez-Krämer, P. and Ogier, J.-M. (2017). A comprehensive survey of mostly textual document segmentation algorithms since 2008, *Pattern Recognition* **64**: 1–14.

Feldbach, M. and Tönnies, K. (2001). Robust line detection in historical church registers, *23rd DAGM Symposium on Pattern Recognition, Munich, Germany*, pp. 140–147.

Franken, E., van Almsick, M., Rongen, P., Florack, L. and ter Haar Romeny, B. (2006). An efficient method for tensor voting using steerable filters, *European Conference on Computer Vision, Graz, Austria*, pp. 228–240.

Gatos, B., Stamatopoulos, N. and Louloudis, G. (2011). *ICDAR 2009* handwriting segmentation contest, *International Journal on Document Analysis and Recognition* **14**(1): 25–33.

Han, S., Lee, M.-S. and Medioni, G. (1997). Non-uniform skew estimation by tensor voting, *Workshop on Document Image Analysis (DIA'97), San Juan, Puerto Rico*, pp. 1–4.

Kennard, D.J. and Barrett, W.A. (2006). Separating lines of text in free-form handwritten historical documents, *2nd International Conference on Document Image Analysis for Libraries (DIAL'06), Lyon, France*, pp. 12–23.

LeBourgeois, F. (1997). Robust multifont OCR system from gray level images, *Proceedings of the 4th International Conference on Document Analysis and Recognition, Ulm, Germany,* Vol. 1, pp. 1–5.

Lee, M.-S. and Medioni, G. (1997). Inferred descriptions in terms of curves, regions and junctions from sparse, noisy binary data, *3rd International Workshop on Visual Form, Capri, Italy*, pp. 350–367.

Li, Y., Zheng, Y., Doermann, D. and Jaeger, S. (2008). Script-independent text line segmentation in freestyle handwritten documents, *Pattern Analysis and Machine Intelligence, IEEE Transactions on* **30**(8): 1313–1329.

Likforman-Sulem, L., Hanimyan, A. and Faure, C. (1995). A hough based algorithm for extracting text lines in handwritten documents, *Proceedings of the 3rd International Conference on Document Analysis and Recognition, Montreal, Quebec, Canada*, Vol. 2, pp. 774–777.

Likforman-Sulem, L., Zahour, A. and Taconet, B. (2007). Text line segmentation of historical documents: A survey, *International Journal of Document Analysis and Recognition* **9**(2): 123–138.

Louloudis, G., Gatos, B., Pratikakis, I. and Halatsis, C. (2008). Text line detection in handwritten documents, *Pattern Recognition* **41**(12): 3758–3772.

Louloudis, G., Gatos, B., Pratikakis, I. and Halatsis, C. (2009). Text line and word segmentation of handwritten documents, *Pattern Recognition* **42**(12): 3169–3183.

Maggiori, E., Manterola, H.L. and del Fresno, M. (2014). Perceptual grouping by tensor voting: A comparative survey of recent approaches, *IET Computer Vision* **9**(2): 259–277.

Medioni, G. and Kang, S.B. (2004). *Emerging Topics in Computer Vision*, Prentice Hall, Upper Saddle River, NJ.

Mordohai, P. and Medioni, G. (2006). Tensor voting: A perceptual organization approach to computer vision and machine learning, *Synthesis Lectures on Image, Video, and Multimedia Processing* **2**(1): 1–136.

Naz, S. (2015). Segmentation techniques for recognition of Arabic-like scripts: A comprehensive survey, *Springer Journal of Education and Information Technologies* **21**(5): 1225–1241.

Nguyen Dinh, T. and Lee, G.S. (2011). Text line segmentation in handwritten document images using tensor voting, *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences* **E94.A**(11): 2434–2441.

Nguyen Dinh, T., Park, J.-H. and Lee, G.-S. (2010). Voting based text line segmentation in handwritten document images, *10th IEEE International Conference on Computer and Information Technology, Bradford, UK*, pp. 529–535.

Pach, J.L. and Bilski, P. (2014). Robust method for the text line detection and splitting of overlapping text in the Latin manuscripts, *Machine Graphics and Vision* **23**(3–4): 11–22.

Papavassiliou, V., Katsouros, V. and Carayannis, G. (2010). A morphological approach for text-line segmentation in handwritten documents, *2010 International Conference on Frontiers in Handwriting Recognition (ICFHR), Kolkata, India*, pp. 19–24.

Phillips, I.T. and Chhabra, A.K. (1999). Empirical performance evaluation of graphics recognition systems, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **21**(9): 849–870.

Ptak, R., Żygadło, B. and Unold, O. (2017). Projection-based text line segmentation with a variable threshold, *International Journal of Applied Mathematics and Computer Science* **27**(1): 195–206, DOI: 10.1515/amcs-2017-0014.

Pu, Y. and Shi, Z. (1999). A natural learning algorithm based on hough transform for text lines extraction in handwritten documents, *Advances in Handwriting Recognition* **34**: 141–150.

Razak, Z., Zulkiflee, K., Idris, M.Y.I., Tamil, E.M., Noorzaily, M., Noor, M., Salleh, R., Yaakob, M., Yusof, Z.M. and Yaacob, M. (2008). Off-line handwriting text line segmentation: A review, *International Journal of Computer Science and Network Security* **8**(7): 12–20.

Sarkar, R., Malakar, S., Das, N., Basu, S., Kundu, M. and Nasipuri, M. (2011). Word extraction and character segmentation from text lines of unconstrained handwritten Bangla document images, *Journal of Intelligent Systems* **20**(3): 227–260.

Vo, Q.N., Kim, S.H., Yang, H.J. and Lee, G.S. (2018). Text line segmentation using a fully convolutional network in handwritten document images, *IET Image Processing* **12**(3): 438–446.

Wong, K.Y., Casey, R.G. and Wahl, F.M. (1982). Document analysis system, *IBM Journal of Research and Development* **26**(6): 647–656.

Wu, J.-C., Hsieh, J.-W. and Chen, Y.-S. (2008). Morphology-based text line extraction, *Machine Vision and Applications* **19**(3): 195–207.

Wu, T.-P., Yeung, S.-K., Jia, J., Tang, C.-K. and Medioni, G. (2012). A closed-form solution to tensor voting: Theory and applications, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **34**(8): 1482–1495.

Zhang, C. and Lee, G.S. (2011). Text line segmentation in Chinese handwritten text images, *17th Korea–Japan Joint Workshop on Frontiers of Computer Vision (FCV), Ulsan, South Korea*, pp. 253–255.

Tomasz Babczyński received his MSc degree in electronics (industrial automation) from the Wrocław University of Science and Technology in 1989, and his PhD degree in computer science (software engineering) in 2001 from the same university. He works as an assistant professor in the Department of Computer Engineering, Faculty of Electronics, Wrocław University of Science and Technology. His scientific interests concentrate on the performance evaluation of computerized systems and mathematical and statistical methods of such an evaluation. He has applied the relevant methods to the safety and reliability analysis of transportation and power systems. His interests also include the analysis of historical maps and documents, especially for genealogical purposes.

Roman Ptak received his MSc and PhD degrees in computer science from the Wrocław University of Science and Technology in 1998 and 2006, respectively, and his MA degree in history from the University of Wrocław in 2001. He is an assistant professor in the Department of Computer Engineering, Wrocław University of Science and Technology. His current research focuses on image recognition and computational intelligence and their application, as well as on spatio-temporal databases and data warehouses. His other interests include the history of Silesia, historical maps, paleography and analysis of historical documents, also for forensic purposes.

Received: 24 December 2019
Revised: 12 May 2020
Re-revised: 1 July 2020
Accepted: 2 July 2020