INTL JOURNAL OF ELECTRONICS AND TELECOMMUNICATIONS, 2024, VOL. 70, NO. 1, PP. 95–102
Manuscript received November 15, 2023; revised March, 2024. doi: 10.24425/ijet.2024.149519

Direct Tensor Voting in line segmentation of handwritten documents

Tomasz Babczyński, and Roman Ptak

***Abstract*—In the vast archives and libraries of the world, countless historical documents are tucked away, often difficult to access. Thankfully, the digitization process has made it easier to view these invaluable records. However, simply digitizing them is not enough – the real challenge lies in making them searchable and computer-readable. Many of these documents were handwritten, which means they need to undergo handwriting recognition. The first step in this process is to divide the document into lines. This article introduces a solution to this problem using tensor voting. The algorithm starts by conducting voting on the binary image itself. Then, using the local maxima found in the resulting tensor field, the lines of text are precisely tracked and labeled. To ensure its effectiveness, the algorithm's performance was tested on the data-set delivered by the organizers of the ICDAR 2009 competition and evaluated using the criteria from this contest.**

***Keywords*—document image analysis; off-line cursive script recognition; reliability of handwriting processing; handwritten text line segmentation; Tensor Voting; ICDAR09 competition**

I. Introduction

THIS paper is an extended version of the [1] publication, offering a more comprehensive literature review and presenting additional experimental results.

The transition to digital documents has not entirely supplanted conventional paper-based records, many archival materials and publications still exist in paper form. In addition to paper, other materials like for example parchment were also used for writing. Researchers, including historians, who frequently examine handwritten inscriptions, find these documents significant. However, accessing these records is often challenging as they are stored in archives and libraries worldwide. Digitization has made accessing them easier, but the documents remain unsearchable until their contents are rendered in a machine-readable format. This necessitates document image processing, with text line segmentation being a key aspect. Of course, the primary objective is to identify handwritten text, but the accurate and reliable segmentation of text lines is essential for the recognition process. In the context of text segmentation, localizing the text is often the first step. This involves identifying the boundaries of the text region within an image. Once the text is localized, the next step is to segment it into individual lines. This process separates the lines of the text from each other, making it easier to analyze and process them. After the line segmentation, the text can be further divided into words, and then into individual letters for character recognition purposes. This sequential approach allows for a systematic and accurate analysis of the text content.

In this article, we focus on the problem of text line segmentation using the Tensor Voting method.

The rest of the paper is arranged as follows: Section II discusses related works on text line segmentation, Section III presents the proposed method, Section IV showcases our experimental results, and finally, Section V provides the conclusions.

T. Babczyński and R. Ptak are with Department of Computer Engineering, Wrocław University of Science and Technology, Wrocław, Poland (e-mail: {tomasz.babczynski, roman.ptak}@pwr.edu.pl).

II. Background research

Various methods were employed to resolve the text lines segmentation issue, and it is valuable refer to articles: [2], [3] for a comprehensive understanding of the diverse solutions implemented.

Moreover, you may find more recent reviews in the papers [4], [5]. In the first case, unfortunately, details of the experiments were not given. The second work is more extensive and uses more references.

In the field of segmentation, there are two main types of algorithms: top-down and bottom-up. The focus of this review is on the top-down methods, specifically those that involve accumulating data or voting. Additionally, other approaches are discussed, particularly those utilized in the ICDAR 2009 *Handwriting Segmentation Contest* [6], which will be described in more detail later on. The performance of our algorithm is compared with the results obtained by the participants in this competition.

Among the prevalent methods employed for handwriting segmentation, projection profile-based techniques stand out. These techniques entail the aggregation of pixel data along specified paths, often by summing the values of foreground pixels in a binary image marked as 'black.' Horizontal or vertical projection is commonly employed for segmenting lines or words. Both older publications like [7], [8] and newer papers like [9], [10] employ these methods. Some algorithms also utilize piece-wise projection profiles, as seen in [11], [12], which can effectively handle instances of overlapping or slanted lines in handwritten documents.

The classical 1-pixel Hough transform is an alternative concept within the domain of document analysis that falls under the category of accumulating methods. It involves collecting the pixel coordinates or centroids of connected elements, with pixels or determined centers acting as voters [13]. This technique is commonly used to find straight elements in images,

enabling tasks such as slope determination, skew and slant detection, as well as text line segmentation (e.g. [14], [15]). Both pixel-based and block-based Hough transform methods can be used for text line segmentation [16]. These methods are capable of handling documents exhibiting variations in skew among their lines, as demonstrated in previous studies [17], [18].

The Tensor Voting (TV) based procedure shares a similar concept with the Hough transform, as they both belong to the accumulating methods group in image recognition. However, the TV technique differs in its approach by working more locally, with each voter casting votes in a limited neighborhood. By employing tensor representation of image features and non-linear voting, it enables the detection of straight lines as well as curves of the second order. For line segmentation, the first tensor field is usually generated from central points of connected regions of foreground pixels. The TV procedure is then applied to identify points that are more likely to belong to actual lines, forming the basis for constructing line chains and segmenting the document.

The TV method finds common application in estimating the non-uniform skew of text lines within printed documents. In the original work conducted by the authors of [13], this approach employed the centroids of connected components for the construction of the initial tensor field, subsequently employing a double voting process. This approach performs well on documents with clear letter separation, even when distortion occurs during the digitization process. However, the approach does not handle handwritten texts correctly.

To address this limitation, the algorithm has been adapted for handwritten text analysis and presented in the publication [19]. This adaptation incorporates 2D Tensor Voting and uses the voting results to eliminate unwanted center points of connected handwriting components. Additionally, the application of the method to Chinese handwritten script is presented in the paper [20].

Barrett and Kennard [21] introduced an interesting approach centered around local data accumulation. This technique primarily involves the computation and binarization of a transition count map between the foreground and background. Subsequently, the connected components in this binarized map are scrutinized and partitioned using a min-cut/max flow graph cutting algorithm as described in [22]. This process effectively isolates connected lines of text, thereby enhancing segmentation accuracy.

Certain algorithms mentioned above initiate their process with connected components. Other approaches that also employ connected regions can be classified as bottom-up methods. In these methods local elements such as pixels are connected into bigger structures and finally handwriting lines.

Furthermore, other chosen methods are showcased including those employed in the ICDAR 2009 *Handwriting Segmentation Contest*. These methods employ distinct approaches.

The winner of the above mentioned competition — referred to as CUBS is based on *Adaptive Local Connectivity Map*, which is defined as a convolution. This algorithm is actually a projection in a moving window. A steerable directional filter is used to apply certain orientations of the projection [23].

Morphological operators have also found application in image segmentation, including line segmentation in handwritten documents (see e.g. [24]). The paper [25] introduced an approach for handwritten documents utilizing binary morphology. This technique utilizes dilation and morphological opening operations. Dilation is used to identify text line elements by connecting areas that are close to each other or overlap horizontally. The choice of the structuring element in the opening operation is made to avoid merging pixels in the vertical direction.

For the task of text segmentation in the case of printed documents, smearing methods can also be applied. One illustration of such an algorithm is the *Run-Length Smoothing Algorithm* (RLSA) [26]. If the distance between black pixels, which represent the foreground in the binary image of the document, falls below a predefined threshold, they are connected along the horizontal direction. The smearing direction should align with the orientation of the text lines, typically horizontal. A modification of this method, adopted to gray level images, is detailed in [27].

The paper [28] proposed a method called “water flow” for segmenting text lines. The assumption is that theoretical water flows from both sides of the image area. The sections in the image where the water does not reach are identified as the regions for extracting text lines. This algorithm was extended to improve the segmentation performance, see the following papers for details [29]–[31].

The authors of the paper [32] proposed an algorithm for recognizing baselines and centerlines. This approach relies on identifying local minima within connected components. It progressively recognizes the text line from its segments, enabling it to handle closely spaced or even touching lines.

Probability theory finds extensive application across diverse fields, including the segmentation of handwritten documents. The approach outlined in reference [33] introduces a robust method grounded in density estimation. This technique involves estimating a probability map for a document image, indicating the likelihood that a given pixel belongs to a text line.

Dynamic programming methods are also used. In [34] the image is handled as a graph and the text lines are determined by an efficient dynamic programing algorithm. The authors introduced a new approach for the automated detection of music staff lines. In [35] the handwriting text lines detection is using the well-known Viterbi algorithm.

In the segmentation process, a deep learning approach is employed. For instance, artificial neural networks can be utilized to detect lines within the text. The papers [36] and [37] demonstrate the utilization of a fully convolutional neural network for segmenting handwritten document images into lines.

III. Proposed method

A. An overview of Tensor Voting method in two-dimensional space

The Tensor Voting technique, as initially presented in 1995 and formally outlined in [38], has been widely used in pattern recognition. Inspired by Gestalt psychology, which suggests

that humans perceive shapes such as straight or curved lines even when confronted with only a partial array of points, the Tensor Voting method undertakes the task of instructing computers to establish connections between points within an image to synthesize coherent shapes, mimicking human perception.

The Tensor Voting method belongs to a group of accumulating methods. It is slightly similar to the Hough transformation but works more locally thanks to decay function and it is capable to find second order curves.

Utilizing symmetric, positive semi-definite tensors as the fundamental element for data manipulation, tensors in 2D space are expressed through symmetrical 2×2 matrices, as illustrated in Equation (1). The eigenvalues λ_1 and λ_2 of the tensor provide information about its size and shape[1], while the eigenvectors $\vec{e_1}$ and $\vec{e_2}$ form the orthonormal base providing the orientation. We assume that $\lambda_1 \geq \lambda_2$ and both are non-negative.

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

The method decomposes the tensor into *stick* and *ball* parts as in (3), representing line-like and area-like structures in the image, respectively. The orthogonal decomposition shown in (2) is not used in the presented method. The Tensor Voting method starts from a tensor field built from the information obtained from the input picture and then uses a voting procedure to connect tokens and form shapes.

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

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

The eigenvalue λ_2 can be interpreted, in the domain where TV is used, as a quantification of the tensor's circular attributes, also referred to as its *anisotropic saliency*. This value encapsulates information regarding intersections, regions, and the presence of noise within the image. Conversely, the value $\lambda_1 - \lambda_2$ serves as an indicator of the tensor's *curvilinear saliency* or its tendency to resemble a slender, stick-like structure, signifying the confidence in the existence of a line passing through the specific image point, oriented perpendicular to the $\vec{e_1}$ eigenvector.

In most cases, a 2D tensor is visually portrayed as an ellipse, wherein the lengths of the primary semiaxes are directly proportional to the eigenvalues. This graphical representation, in conjunction with the separation between stick-like and ball-like characteristics, is depicted in Figure 1.

At the start, the input image undergoes encoding, transforming it into a tensor field. The tensors belonging to it are called *tokens*. The exact encoding method differs based on the specific problem being addressed. In our case, the encoding process is straightforward and limited to vertical stick tensors exclusively.

After the primary tensor field is generated, the voting process takes place. Tokens cast their votes within their local vicinity, either on other tokens through *sparse voting*, or across all positions via *dense voting*.

[1]The "size" and "shape" refer to the graphical representation as an ellipse.

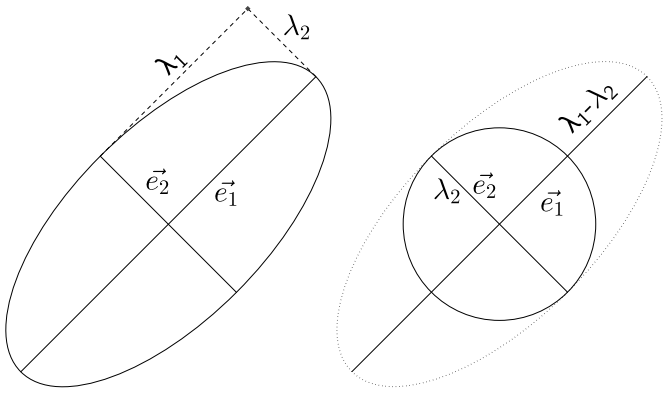


Fig. 1. Possible tensor decompositions.

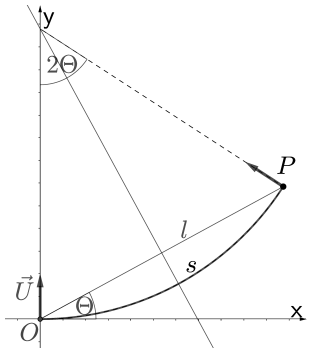


Fig. 2. 2D stick vote between two tokens.

a) Direction of the vote: Let's examine Figure 2 to understand the concept. In this figure, we have two tokens: **O** representing the *voter*, and **P** representing the *votee*. Usually, a continuous curve that traverses these two points will coincide with the osculating circle. In this particular case, we represent the curve as the arc denoted by s. When the tensor at point **O** takes the form of a pure stick tensor, signifying the presence of a nonzero eigenvector $\vec{U}$ (often referred to as the stick tensor's orientation), the stick tensor at the voting position will likewise exhibit an orientation perpendicular to the arc s.

b) Analyzing voting strength: In the Tensor Voting method, the intensity of the vote is determined through a consideration of both spatial separation and the angular disparity between tensors. Diverse adaptations of the Tensor Voting method exist, all rooted in the concept of distance decay with an exponential profile; however, they diverge in terms of how they penalize curvature and assess the spatial extent.

For instance, in the case of the Original Tensor Voting (OTV), distance is quantified as the arc length, while curvature penalties hinge upon the curvature value κ, as exemplified by the equation presented in (4).

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

Here, several key parameters come into play: s, denoting the arc length between the positions of the tensors under scrutiny; κ, a metric of curvature, calculated as $\kappa = 1/r$, with r representing the radius of the osculating circle determined by $r = l/(2\sin\Theta)$; and finally, c, a constant meticulously derived by the creators of OTV. The sole free parameter within

the method is denoted as σ, known as the *scale of voting*. In its original conception, the method prunes tensors associated with angles Θ greater than $45°$, as well as the central tensor at the origin.

c) Exploring Steerable Filters in TV: In the Steerable filters adaptation of the Tensor Voting method (STV), introduced in [39], the distance measurement is based on the straight-line distance. Instead of the curvature component, it employs a power of the cosine function to penalize deviations from straight lines. STV does not engage in conventional voting; rather, it performs convolutions of scalar fields within the complex number domain to compute three real-valued scalar fields, namely *saliency*, *ballness*, and *orientation*. While STV doesn't explicitly calculate tensor fields, it has the capability to generate them when necessary. The decay function for the STV kernel is illustrated in (5).

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

In this equation, l denotes the Euclidean distance between token positions, σ represents the voting scale, and $n = 2$ serves as the curvature penalization parameter, a value resulting in the simplest expressions defined in the method as suggested by the variant's creators. Figure 2 provides a visual depiction of these parameters.

It's worth noting that the STV method is specialized for stick voting within 2D space and does not support ball voting. While these limitations may initially seem restrictive, they align perfectly with our algorithm's specific requirements, which only necessitate stick voting in a 2D context. A significant advantage of the STV formulation lies in its speed, outperforming the OTV in terms of voting efficiency. Furthermore, its speed remains consistent, regardless of the number of tokens in the initial field, rendering dense voting highly efficient. This enables direct voting on images without the need to generate a sparse field, distinguishing it from the OTV and other established Tensor Voting formulations.

In the Figure 3 we present an example of the STV voting kernel. The contour lines show the half decays of the tensors' saliency and are plotted for values $1/2^n$ where $n = 1 \div 6$. Short lines show the directions of tensors in each point. The kernel is normalized – the saliency of the tensor in the center is equal to 1.

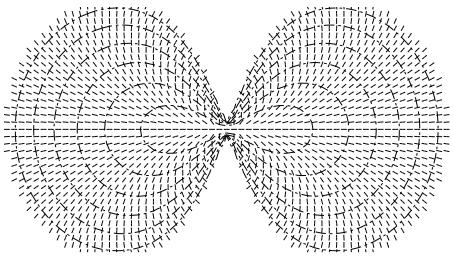


Fig. 3. Voting kernel for Tensor Voting using Steerable filters.

For more information on the OTV formalism, interested readers can refer to [40], [41]. The STV is extensively discussed in [39]. The paper [42] provides a comprehensive overview of the most advanced developments in the TV domain, showcasing various variants of the formalism. Since the publication of that survey, the validity of the challenging closed-form solution has been proven. Additionally, a significant new result is the development of the probabilistic Tensor Voting which incorporates its own voting kernel shape and the ability to handle inaccuracies in the initial tensor positions.

B. Algorithm

The proposed algorithm assumes a binary image as the input and does not involve the binarization process. The procedure can be summarized as follows (Algorithm 1). Each step of the algorithm will be explained in detail in the subsequent paragraphs.

Algorithm 1 Summary of the Algorithm

Input: I_{in} {BW picture}
{**Parameters:**
σ – scale of the TV
ϕ – angle deviation limit
ρ – threshold of the saliency value}
Output: I_{out} {segmented image}

1: Calculate average character height;
2: Create initial tensor field;
3: Perform the dense voting;
4: Calculate the gradient;
5: Select points probably belonging to lines;
6: Calculate lines;
7: Assign labels to pixels;
8: **return** I_{out}

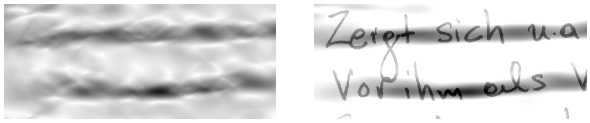
Step 1 The character's average height is determined by computing the mean height of connected components in the whole document. This parameter, represented as $\overline{H}$, holds significance in subsequent stages, notably in step 6. In cases where text lines frequently overlap, $\overline{H}$ might register as elevated, although the algorithm maintains robustness to variations in this value.

Step 2 Within the initial tensor field, every foreground pixel of the input image is symbolized by a unit stick tensor. These tensors uniformly align horizontally, reflecting the presumption that text lines predominantly exhibit horizontal orientation.

Step 3 In the case of STV, instead of traditional voting, a unique process is employed, driven by a scale parameter σ that can be adjusted during experiments. This process results in the generation of the saliency, ballness, and orientation fields. During this phase, the reconstruction of the tensor field is unnecessary as the saliency and orientation fields prove adequate for ensuing steps. Figure 4a provides a depiction of the saliency field for a segment encompassing two lines of text, with darker points signifying areas of heightened line saliency.

Step 4 To refine the saliency field, a Gaussian smoothing operation is employed, utilizing dispersion values of $\sigma_x = 30$ and $\sigma_y = 3$. This anisotropic adjustment stems from the underlying presumption of horizontal lines. The specific values of σ were determined through preliminary experiments on a

set of documents. The impact of this filtering process can be observed in Figure 4b. Next, the smoothed saliency field (s) undergoes computation of its vertical gradient component, achieved through the central differences method, denoted as $G_{:,y} = (s_{:,y+1} - s_{:,y-1})/2$.



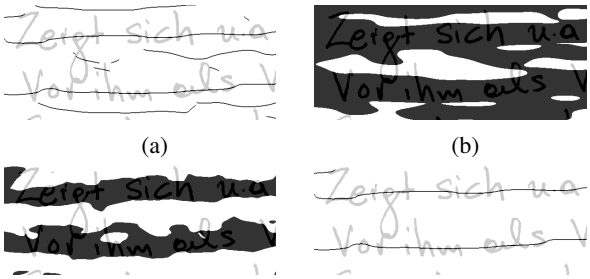
(a) (b)

Fig. 4. A segment of the (a) post-voting saliency field and (b) smoothed field, superimposed with the original text

Step 5 Points are now selected based on whether they meet all conditions in (6). These points are considered to be the most likely ones belonging to the text lines.

$$\begin{cases} G_{x,y-1} \text{ negative} \\ G_{x,y+1} \text{ positive} \\ s_{x,y} > \rho\,\overline{s} \\ |o_{x,y}| < \phi \end{cases} \tag{6}$$

In our notation, G corresponds to the vertical gradient component. The s symbolizes the smoothed saliency field, with $\overline{s}$ representing its average value. The saliency threshold is denoted as ρ. Furthermore, o represents the orientation field. We use the notation $|.|$ to denote the absolute value of a number, and the angle limit is expressed as ϕ, a parameter we varied in our experiments. Figure 5 illustrates regions where specific conditions have been satisfied. The outcome of this stage is illustrated in Figure 5d, which shows the areas selected as likely belonging to the text lines.

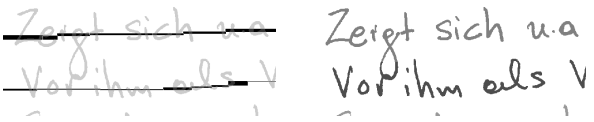


(a) (b) (c) (d)

Fig. 5. Document section featuring areas where the conditions specified in (6) are met, depicted in (a) 1 and 2, (b) 3, (c) 4, and (d) all conditions

Step 6 To generate line chains for each text line from a set of points, we initiate the process by searching for starting points. This search is conducted first in a downward direction and then from left to right. Every discovered point (P) serves as the origin of a polyline. The construction of this polyline involves selecting subsequent points within a moving window, which spans dimensions of $2{\cdot}\overline{H}\times 70$. This window is centered vertically on point P and commences from P's x position. Linear regression is employed to determine the direction of the line emanating from P. The next point in the line chain is identified as the farthest point within the window, provided its distance from the line is less than 5. Any points to the left of this selected point are removed, and the window is repositioned accordingly. This iterative process persists until either the image's right edge is reached or there are no points remaining within the window.

In practice, we often detect an excessive number of line chains, necessitating a refinement step. During this step, lines that either do not intersect any connected component of the original document or exclusively intersect components already touched by another line chain are eliminated. An example of a line to be removed is visible in the upper left corner of Figure 5d. Following this elimination process, line chains that are in close proximity to each other, with a maximum vertical distance lower than $\overline{H}$, are merged and assigned the same order number. The result of this merging process is demonstrated in Figure 6a.



(a) (b)

Fig. 6. (a) Line chains for two lines of text, (b) results of the segmentation

When we have determined the set of line chains, the process of text labeling ensues. This process involves the individual analysis of each connected region within the image. If a region is exclusively intersected by a single line chain, all the points within that region are labeled with the number corresponding to that chain. Conversely, in cases where a region remains untouched by any chain or is intersected by multiple chains, each point within the region is assigned the label of the nearest chain. The proximity is determined using the straight-line distance. The outcomes of the entire algorithm are showcased in Figure 6b. Nevertheless, it's crucial to acknowledge that the algorithm occasionally faces challenges in distinguishing overlapping letters from two consecutive lines, especially when only one of them has a descender or ascender. An example of this can be observed in the case of the letter "g", which tail is incorrectly assigned to the second line.

At the conclusion of the process, the labeled image is subjected to a comparison with the manually annotated image, which serves as the ground truth. A comprehensive explanation of this comparison can be found in section IV.

The complete code of the algorithm and evaluation procedure can be found in a public repository [43].

IV. Experiments

A. Data-set

Our algorithm underwent testing using a collection of handwritten documents sourced from the *Handwriting Segmentation Contest*, which was associated with the ICDAR 2009 conference [6]. The data-set was partitioned into two distinct sections: the training set and the benchmark set. The benchmark set encompassed 200 one-page handwritten documents, spanning multiple languages, namely English, French, German, and

Greek. These documents were authored by various individuals, resulting in a total of 4034 lines across all the documents. In contrast, the training set consisted of 100 documents. These documents, while similar, exhibited variations in script, size, and layout. They were sourced from the ICDAR 2007 competition and served as the training data for the ICDAR 2009 contest. We have also performed additional experiment taking some documents from the benchmark set and using them to train our algorithm. All images in the data-set were in black-and-white. To establish a basis for evaluation, the competition organizers manually annotated each document, creating the *ground truth*. This ground truth data-set was subsequently used to assess the results achieved by participants. Notably, each pixel within the image was assigned a label corresponding to its respective line.

B. Evaluation methodology

The assessment of outcomes followed a one-to-one correspondence method, as outlined in [44]. The MatchScore table served as the primary tool for this evaluation.

To construct the table, denoted as (7), we initiate with the set of foreground pixels, referred to as I. We define R_i as the set of pixels identified as belonging to the i^{th} class, and G_j as the set of pixels in the j^{th} class of the ground truth. The function $T(s)$ counts elements of the set s. Within the MatchScore table, values are assigned in the inclusive range of 0 to 1.

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

We define a one-to-one match between a recognized line i and a ground truth line j when the $\mathrm{MatchScore}(i,j) > 0.95$, a value accepted within the ICDAR challenge. Here, we denote M as the count of recognized lines, N as the count of lines in the ground truth, and $o2o$ as the count of one-to-one matches. The metrics of detection rate (DR) and recognition accuracy (RA), along with the value FM employed in constructing the rank table of applications during the competition, are defined in equation (8). For a more comprehensive evaluation of the results, please refer to the report published after the competition [6].

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

Later contests, such as those at ICFHR 2010 [45] and ICDAR 2013 [46], were also conducted, but they used more challenging data-sets for segmentation. In this study, we chose the simplest data-set to facilitate comparison with various methods, including the approach outlined by [19] and our earlier work [10].

C. Experimental results

During the evaluation phase, we utilized the aforementioned data-set of manuscripts. To evaluate the algorithm's performance outlined in section III-B, we conducted a comparative analysis by varying the algorithm's parameters. Most of these parameters are defined in (6). In our initial experiments, we explored the effect of adjusting the threshold ρ within the range of 0 to 0.9. This allowed us to observe the algorithm's behavior when accepting a wide spectrum of saliency values or focusing only on the largest ones. Surprisingly, our method proved to be highly robust to changes in this parameter, leading us to present results for a single value: $\rho = 0.5$. Additionally, we examined the impact of modifying the voting scale σ (from 30 to 150) and the angle ϕ (from 0 to 50°).

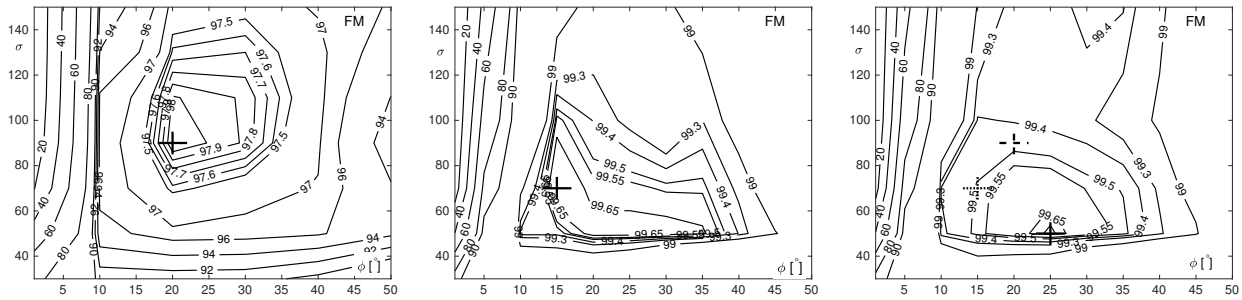
The figure 7a presents the outcomes of fine-tuning the algorithm using the training data-set. After extensive experimentation, the optimal solution was achieved with the parameter values of $\sigma = 90$ and $\phi = 20°$. To highlight this successful combination, a cross has been placed at the corresponding location on the figure.

The evaluation of our algorithm's performance on the benchmark data-set involved conducting experiments within the same range of attributes as employed earlier. This allowed us to assess the algorithm's potential across various parameter values. The outcomes of these experiments are showcased in Figure 7c. The figure clearly illustrates the robustness of our method in response to variations in the σ and ϕ parameters. Over a broad spectrum of parameter values, the quality metric FM consistently achieves a level exceeding 99%. The plateau near the peak in the figure is big and indicates a significant region of optimal performance. The dashed line cross marks the optimal parameter combination identified during tuning. In a competition scenario, this combination of parameters would have yielded a FM score of 99.43%. The documents from the benchmark set were very similar one to the other while the training set was not so coherent. To check the influence of this fact on the results, we provided additional experiment using the 50 documents selected from the benchmark set as the training data. Now, the $\sigma = 70$ and $\phi = 15°$ were the best. Results are shown in the figure 7b and marked in 7c with the cross indicated by the dotted line. For this pair of parameters the metric FM equals 99.51%. When evaluating achievements of our algorithm on a set of manuscripts, we compared the results with those of the ICDAR 2009 competition. Our algorithm demonstrated impressive performance, securing the second position with a negligible difference of merely 0.1 percentage point away from the top scorer's result ($FM = 99.53\%$). This outcome showcases the effectiveness of our algorithm and its potential to excel in similar competitions.

The values obtained in the experiments on the benchmark data-set are shown in the table I alongside the outcomes achieved by participants in the ICDAR 2009 competition. Our results are labeled:

1) **STV-t** for the parameters selected in the training phase on the data-set used during the contest, marked with a dashed cross in the figure 7c,
2) **STV-b** for the parameters obtained during the training phase on the selected documents from the benchmark set (dotted line cross in the figure 7c).

Of course, a comparison of the second case with the results obtained during the competition would be unfair, but it shows the potential of the algorithm. Additionally, we show the results presented in [19] with the algorithm also using Tensor



(a) (b) (c)

Fig. 7. *FM* at a threshold $\rho = 0.5$ (a) training data-set, (b) training on selected benchmark images (c) benchmark documents

Voting. That algorithm did not participate in the ICDAR competition. Their method is labeled *CTV*.

During the evaluation phase, it was observed that the algorithm performed perfectly when segmenting many documents. However, upon analyzing the eight inaccurately segmented documents, it was found that the algorithm had successfully identified the lines in most cases. The errors were primarily a result of inaccuracies in diacritic classification or the merging of characters in adjacent lines. As per the competition rules, such cases were rejected. Out of the total 4034 lines, only two lines were recognized completely incorrectly.

TABLE I.
ICDAR 2009 results

	DR[%]	RA[%]	FM[%]
CUBS	99.55	99.50	99.53
STV-b	**99.49**	**99.53**	**99.51**
CTV	*99.58*	*99.31*	*99.44*
STV-t	**99.43**	**99.43**	**99.43**
ILSP-LWSeg-09	99.16	98.94	99.05
PAIS	98.49	98.56	98.52
CMM	98.54	98.29	98.42
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
REGIM	40.38	35.70	37.90

V. Conclusions

The algorithm described in the paper proposes an innovative method for text lines segmenting in manuscripts. It utilizes a rapid variant of Tensor Voting that incorporates the concept of steerable filters. This modified version simplifies the preprocessing phase of the algorithm significantly. The procedure begins with a binary image, where each pixel is transformed into a tensor and constitutes the initial field for the voting process.

The experiments conducted in this study demonstrated that the proposed method for text line segmentation achieved very favorable results. When compared to the algorithms used by participants of the ICDAR 2009 competition, our method would have secured the second prize[2], indicating its effectiveness. This establishes our method as a reliable step in the comprehensive analysis of historical documents. Notably, the results were found to be robust against parameter changes, implying that the tuning process does not require precise adjustments to achieve satisfactory outcomes. However, further investigation of the parameter space revealed the potential for even better results, highlighting the need to refine the method of determining optimal parameters based on the document's specific characteristics.

The data-set used for testing the algorithm had simple layouts, with clear line spacing and minimal instances of lines touching each other. Such documents were segmented perfectly. However, the proposed method may not perform equally well on more complex layouts where lines are closely intertwined. It shows the need of improving the labeling stage to handle touching lines more effectively. Additionally, it would be important for future development of the algorithm to simplify and speed up the phase of constructing line chains, which is currently a complex and lengthy procedure. These potential areas of improvement are important for enhancing the algorithm's performance in real-world scenarios with more diverse document layouts.

[2] *CTV* was not presented during the competition. It is the method from [19].

References

[1] T. Babczyński and R. Ptak, "Line segmentation of handwritten documents using direct tensor voting," in *Dependable Computer Systems and Networks*, W. Zamojski, J. Mazurkiewicz, J. Sugier, T. Walkowiak, and J. Kacprzyk, Eds. Cham: Springer Nature Switzerland, 2023, pp. 1–12.

[2] L. Likforman-Sulem, A. Zahour, and B. Taconet, "Text line segmentation of historical documents: a survey," *International Journal of Document Analysis and Recognition (IJDAR)*, vol. 9, no. 2, pp. 123–138, 2007.

[3] Z. Razak, K. Zulkiflee, M. Y. I. Idris, E. M. Tamil, M. Noorzaily, M. Noor, R. Salleh, M. Yaakob, Z. M. Yusof, and M. Yaacob, "Off-line handwriting text line segmentation: A review," *International Journal of Computer Science and Network Security*, vol. 8, no. 7, pp. 12–20, 2008.

[4] N. Mehta and J. Doshi, "Segmentation methods: A review," *International Journal for Research in Applied Science and Engineering Technology*, vol. 8, pp. 536–540, 10 2020. [Online]. Available: doi:10.22214/ijraset.2020.31939

[5] S. Joseph and J. George, "A review of various line segmentation techniques used in handwritten character recognition," in *Information and Communication Technology for Competitive Strategies (ICTCS 2021)*, A. Joshi, M. Mahmud, and R. G. Ragel, Eds. Singapore: Springer Nature Singapore, 2023, pp. 353–365.

[6] B. Gatos, N. Stamatopoulos, and G. Louloudis, "ICDAR2009 handwriting segmentation contest," *International Journal on Document Analysis and Recognition (IJDAR)*, vol. 14, no. 1, pp. 25–33, 2011.

[7] M. Shridhar and F. Kimura, "Handwritten address interpretation using word recognition with and without lexicon," in *1995 IEEE International Conference on Systems, Man and Cybernetics. Intelligent Systems for*

the 21st Century, vol. 3, 1995, pp. 2341–2346 vol.3. [Online]. Available: doi:10.1109/ICSMC.1995.538131

[8] E. Kavallieratou, N. Dromazou, N. Fakotakis, and G. Kokkinakis, "An integrated system for handwritten document image processing," *International Journal of Pattern Recognition and Artifical Intelligence*, vol. 17, pp. 617–636, 2003.

[9] R. Ptak, B. Żygadło, and O. Unold, "Projection–based text line segmentation with a variable threshold," *International Journal of Applied Mathematics and Computer Science*, vol. 27, no. 1, pp. 195–206, 2017.

[10] T. Babczyński and R. Ptak, "Line segmentation of handwritten text using histograms and tensor voting," *International Journal of Applied Mathematics and Computer Science*, vol. 30, no. 3, pp. 585–596, 2020. [Online]. Available: doi:10.34768/amcs-2020-0043

[11] M. Arivazhagan, H. Srinivasan, and S. Srihari, "A statistical approach to line segmentation in handwritten documents," in *Document Recognition and Retrieval XIV*, vol. 6500. International Society for Optics and Photonics, 2007, pp. 245–255.

[12] T. Babczyński and R. Ptak, "Handwritten text lines segmentation using two column projection," in *Advances in Intelligent Systems and Computing*. Springer, 2020, vol. 1173 AISC, pp. 11–20. [Online]. Available: doi:10.1007/978-3-030-48256-5{_}2

[13] S. Han, M.-S. Lee, and G. Medioni, "Non-uniform skew estimation by tensor voting," in *Document Image Analysis, 1997.(DIA'97) Proceedings., Workshop on*. IEEE, 1997, pp. 1–4.

[14] G. Louloudis, B. Gatos, I. Pratikakis, and C. Halatsis, "Text line and word segmentation of handwritten documents," *Pattern Recognition*, vol. 42, no. 12, pp. 3169–3183, 2009.

[15] A. Alaei, P. Nagabhushan, and U. Pal, "Piece-wise painting technique for line segmentation of unconstrained handwritten text: a specific study with persian text documents," *Pattern Analysis and Applications*, vol. 14, no. 4, pp. 381–394, 2011.

[16] G. Louloudis, B. Gatos, I. Pratikakis, and C. Halatsis, "Text line detection in handwritten documents," *Pattern Recognition*, vol. 41, no. 12, pp. 3758–3772, 2008.

[17] L. Likforman-Sulem, A. Hanimyan, and C. Faure, "A hough based algorithm for extracting text lines in handwritten documents," in *Document Analysis and Recognition, 1995., Proceedings of the Third International Conference on*, vol. 2. IEEE, 1995, pp. 774–777.

[18] Y. Pu and Z. Shi, "A natural learning algorithm based on hough transform for text lines extraction in handwritten documents," *Series in Machine Perception and Artificial Intelligence*, vol. 34, pp. 141–152, 2000.

[19] T. Nguyen Dinh and G. S. Lee, "Text line segmentation in handwritten document images using tensor voting," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. E94.A, no. 11, pp. 2434–2441, 2011.

[20] C. Zhang and G. S. Lee, "Text line segmentation in chinese handwritten text images," in *17th Korea-Japan Joint Workshop on Frontiers of Computer Vision (FCV)*, 2011, pp. 1–3.

[21] D. J. Kennard and W. A. Barrett, "Separating lines of text in free-form handwritten historical documents," in *Second International Conference on Document Image Analysis for Libraries (DIAL'06)*, 2006, pp. 12–23. [Online]. Available: doi:10.1109/DIAL.2006.40

[22] Y. Boykov and V. Kolmogorov, "An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 9, pp. 1124–1137, 2004.

[23] Z. Shi, S. Setlur, and V. Govindaraju, "A steerable directional local profile technique for extraction of handwritten Arabic text lines," in *10th International Conference on Document Analysis and Recognition*, 2009, pp. 176–180. [Online]. Available: doi:10.1109/ICDAR.2009.79

[24] J.-C. Wu, J.-W. Hsieh, and Y.-S. Chen, "Morphology-based text line extraction," *Mach. Vis. Appl.*, vol. 19, pp. 195–207, May 2008. [Online]. Available: doi:10.1007/s00138-007-0092-0

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

[26] K. Y. Wong, R. G. Casey, and F. M. Wahl, "Document analysis system," *IBM journal of research and development*, vol. 26, no. 6, pp. 647–656, 1982.

[27] F. LeBourgeois, "Robust multifont ocr system from gray level images," in *Document Analysis and Recognition, 1997., Proceedings of the Fourth International Conference on*, vol. 1. IEEE, 1997, pp. 1–5.

[28] S. Basu, C. Chaudhuri, M. Kundu, M. Nasipuri, and D. K. Basu, "Text line extraction from multi-skewed handwritten documents," *Pattern Recognition*, vol. 40, no. 6, pp. 1825–1839, 2007.

[29] D. Brodić and Z. Milivojević, "A new approach to water flow algorithm for text line segmentation," *Journal of Universal Computer Science*, vol. 17, no. 1, pp. 30–47, 2011.

[30] D. Brodić, "Extended approach to water flow algorithm for text line segmentation," *Journal of Computer Science and Technology*, vol. 27, no. 1, pp. 187–194, 2012.

[31] ——, "Text line segmentation with water flow algorithm based on power function," *Journal of Electrical Engineering*, vol. 66, no. 3, pp. 132–141, 2015.

[32] M. Feldbach and K. Tönnies, "Robust line detection in historical church registers," in *Pattern Recognition, 23rd DAGM-Symposium*, 2001, pp. 140–147.

[33] Y. Li, Y. Zheng, D. Doermann, and S. Jaeger, "Script-independent text line segmentation in freestyle handwritten documents," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 30, no. 8, pp. 1313–1329, Sep. 2008. [Online]. Available: doi:10.1109/TPAMI.2007.70792

[34] J. S. Cardoso, A. Capela, A. Rebelo, and C. Guedes, "A connected path approach for staff detection on a music score," in *2008 15th IEEE International Conference on Image Processing*, 2008, pp. 1005–1008. [Online]. Available: doi:10.1109/ICIP.2008.4711927

[35] T. Stafylakis, V. Papavassiliou, V. Katsouros, and G. Carayannis, "Robust text-line and word segmentation for handwritten documents images," in *2008 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2008, pp. 3393–3396. [Online]. Available: doi:10.1109/ICASSP.2008.4518379

[36] Q. N. Vo, S. H. Kim, H. J. Yang, and G. S. Lee, "Text line segmentation using a fully convolutional network in handwritten document images," *IET Image Processing*, vol. 12, no. 3, pp. 438–446, 2018.

[37] Y. Baek, B. Lee, D. Han, S. Yun, and H. Lee, "Character region awareness for text detection," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 9365–9374.

[38] M.-S. Lee and G. Medioni, "Inferred descriptions in terms of curves, regions and junctions from sparse, noisy binary data," in *Proc. IEEE Int. Symp. Computer Vision*, 1995, pp. 73–78.

[39] E. Franken, M. van Almsick, P. Rongen, L. Florack, and B. ter Haar Romeny, "An efficient method for tensor voting using steerable filters," in *European Conference on Computer Vision*. Springer, 2006, pp. 228–240.

[40] G. Medioni and S. B. Kang, *Emerging topics in computer vision*. Upper Saddle River, N.J.: Prentice Hall PTR ; London : Pearson Education, 2004.

[41] P. Mordohai and G. Medioni, "Tensor voting: A perceptual organization approach to computer vision and machine learning," *Synthesis Lectures on Image, Video, and Multimedia Processing*, vol. 2, no. 1, pp. 1–136, 2006.

[42] E. Maggiori, H. L. Manterola, and M. del Fresno, "Perceptual grouping by tensor voting: a comparative survey of recent approaches," *IET Computer Vision*, vol. 9, no. 2, pp. 259–277, 2014.

[43] T. Babczyński and R. Ptak, "Line segmentation of handwritten documents; the MATLAB code," 2023. [Online]. Available: https://github.com/tbabczynski-openSource/HWlinesSTV

[44] I. T. Phillips and A. K. Chhabra, "Empirical performance evaluation of graphics recognition systems," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 9, pp. 849–870, 1999.

[45] B. Gatos, N. Stamatopoulos, and G. Louloudis, "Icfhr 2010 handwriting segmentation contest," in *2010 12th International Conference on Frontiers in Handwriting Recognition*. IEEE, 2010, pp. 737–742.

[46] N. Stamatopoulos, B. Gatos, G. Louloudis, U. Pal, and A. Alaei, "Icdar 2013 handwriting segmentation contest," in *2013 12th International Conference on Document Analysis and Recognition*. IEEE, 2013, pp. 1402–1406.