Expert Systems With Applications 80 (2017) 210–231

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Efficient character segmentation approach for machine-typed documents

Vladan Vučković*, Boban Arizanović

Faculty of Electronic Engineering, Computer Department, Aleksandra Medvedeva 14, 18000 Niš, Serbia

ARTICLE INFO

Article history:
Received 27 November 2016
Revised 11 March 2017
Accepted 12 March 2017
Available online 16 March 2017

Keywords:
Character segmentation
Character recognition
Machine-typed documents
Machine-printed documents

ABSTRACT

In this paper an efficient approach for segmentation of the individual characters from scanned documents typed on old typewriters is proposed. The approach proposed in this paper is primarily intended for processing of machine-typed documents, but can be used for machine-printed documents as well. The proposed character segmentation approach uses the modified projection profiles technique which is based on using the sliding window for obtaining the information about the document image structure. This is followed by histogram processing in order to determine the spaces between lines, words and characters in the document image. The decision-making logic used in the process of character segmentation is describes and represents the most an integral aspect of the proposed technique. Beside the character segmentation approach, the ultra-fast architecture for geometrical image transformations, which is used for image rotation in the process of skew correction, is presented, and its fast implementation using pointer arithmetic and a highly optimized low-level machine routine is provided. The proposed character segmentation approach is semi-automatic and uses threshold values to control the segmentation process. Provided results for segmentation accuracy show that the proposed approach outperforms the state-of-the-art approaches in most cases. Also, the results from the aspect of the time complexity show that the new technique performs faster than state-of-the-art approaches and can process even very large document images in less than one second, which makes this approach suitable for real-time tasks. Finally, visual demonstration of the proposed approach performances is achieved using original documents authored by Nikola Tesla.

1. Introduction

Character segmentation, namely the process of extracting the individual characters from the digital image, either individually or as a part of a complex system, has long been the subject of scientific research. This subject, together with the recognition of the characters, is still a very challenging task. Diverse approaches for character segmentation have been presented in the past (Casey & Lecolinet, 1996). Usually, the process of character segmentation and its importance in complex systems, such as OCR systems (Bourbakis, Pereira, & Mertoguno, 1996; Grafmüller & Beyerer, 2013; Mao & Mohiuddin, 1997; Vamvakas, Gatos, Stamatopoulos, & Perantonis, 2008), is unduly underestimated (Lu, 1995; Lu & Shridhar, 1996). Related works can be classified based on a couple of characteristics. Some of them analyze approaches for character segmentation in natural images (González & Bergasa, 2013; Karatzas & Antonacopoulos, 2007; Lim, Park, & Medioni, 2007) and others deal with character segmentation on document images. This second group has a clear line between machine-printed and machine-typed documents (Antonacopoulos & Karatzas, 2005; Kise, Sato, & Iwata, 1998; Lu, 1995; Min-Chul, Yong-Chul, & Srihari, 1999; Nikolaou, Makridis, Gatos, Stamatopoulos, & Papamarkos, 2010; Park, Ok, Yu, & Cho, 2001) where a document structure and shape of its elements is regular, and handwritten documents where character segmentation is a real challenge due to irregular document structure (Choudhary, Rishi, & Ahlawat, 2013; Kovalchuk, Wolf, & Dershowitz, 2014; Lee & Verma, 2012; Lu & Shridhar, 1996; Manohar, Vitaladevuni, Cao, Prasad, & Natarajan, 2011; Rehman & Saba, 2011; Stamatopoulos, Gatos, Louloudis, Pal, & Alaei, 2013; Xu, Yin, & Liu, 2010; Younes & Abdellah, 2015). Old machine-printed and machine-typed documents are of particular research interest because of important historical documents (Bar-Yosef, Mokeichev, Kedem, Dinstein, & Ehrlich, 2009; Garz, Fischer, Sablatnig, & Bunke, 2012; Gupta, Jacobson, & Garcia, 2007; Vamvakas et al., 2008). The

* Corresponding author.
E-mail addresses: vladanvuckovic24@gmail.com (V. Vučković), bobanarizanovic@hotmail.com (B. Arizanović).

http://dx.doi.org/10.1016/j.eswa.2017.03.027
0957-4174/

most frequent problem in character segmentation of handwritten documents are touching characters and the variety of works deal with that (Kumar, Kang, Doermann, & Abd-Almageed, 2011; Saba, Sulong, & Rehman, 2010; Yang, Yan, & Zhao, 2009). Also, it should be mentioned that character segmentation and recognition of license plates is a popular research topic (Sedighi & Vafadust, 2011; Shan, 2010; Yoon, Ban, Yoon, & Kim, 2011).

Recent research of character segmentation includes all levels of this process. Analyses on image binarization parameters used in document image pre-processing showed that Otsu method and other Otsu-based methods give the best results on average (Gupta et al., 2007). A learning-based approach for finding the best binarization parameters was presented by Fernández-Caballero, López, and Castillo (2012). Venkateswarlu and Boyle (1995), proposed various efficient techniques for character segmentation providing a comparison of these techniques and other techniques. Zheng et al. (2012) exploited the approach based on searching for connected regions in the spatial domain performed on a binary image. Another method uses Bayes theorem for segmentation in order to use the prior knowledge and is adapted for real-time tasks (Grafmüller & Beyerer, 2013). Wang, Huang, and Liu (2011) proposed a method which uses the Gaussian distribution of the pixel colors in order to perform the segmentation. Manohar et al. (2011) proposed a graph clustering-based approach for handwritten text line segmentation. A segmentation approach for historical handwritten documents which consists of text zone detection followed by a text line segmentation method is proposed by Gatos, Louloudis, and Stamatopoulos (2014). Nomura, Yamanaka, Katai, Kawakami, and Shiose (2005), proposed an adaptive approach for character segmentation and feature vector extraction of degraded images, which uses mathematical morphology. Conjoined characters in handwritten documents usually present challenges for segmentation. To address this, supervised learning can be used in combination with Markov Random Field (MRF) (Yang et al., 2009). Xu et al. (2010) proposed character segmentation method for Chinese handwriting documents which uses contour analysis and dynamic time warping (DTW). Tan, Lai, Wang, Wang, and Zuo (2012) proposed a method for character segmentation of handwritten documents based on nonlinear clustering methods. Surinta, Karaaba, Schomaker, and Wiering (2015) proposed a method based on the usage of local gradient feature descriptors for character recognition of handwritten documents, which also uses machine learning algorithms, the k-nearest neighbor and the support vector machine. Kumar et al. (2011) proposed an approach for extraction of the text lines from handwritten document images based on combining the local and global techniques. Roy, Pal, Lladós, and Delalandre (2012) presented a method for segmentation of multi-oriented touching characters, which uses a dynamic programming algorithm. Other approaches for segmentation of connected handwriting characters exploit self-organizing maps and SVM classifiers (Lacerda & Mello, 2013; Roy et al., 2012). Evaluation of performances of the existing segmentation approaches for handwritten documents was presented by Stamatopoulos et al. (2013). For natural images, some segmentation approaches are based on tensor voting and the usage of the three-color bar code (Lim, Park, & Medioni, 2007; Starostenko, Cruz-Perez, Uceda-Ponga, & Alarcon-Aquino, 2015). Kavallieratou, Stamatatos, Fakotakis, and Kokkinakis (2000) proposed the transformation-based learning method for character segmentation of unconstrained cursive handwritten text. A character segmentation method for license plates based on Gaussian low-pass filter and innovational Laplace-like transform is proposed by Sedighi and Vafadust (2011). Shan (2010), proposed a method for segmentation and recognition of the license plates which uses the radial basis function (RBF) neural network. Another character segmentation method for automatic license plate recognition which is based on blob extraction was proposed by Yoon et al. (2011). Beside these character segmentation approaches which perform on document images, there are also video character segmentation approaches. Phan, Shivakumara, Su, and Tan (2011) proposed a video character segmentation method based on gradient vector flow. Since the video segmentation and recognition process is challenged by low resolution frames, a specific solution based on a new gradient based method for segmentation of words and characters is proposed by Shivakumara, Bhowmick, Su, Tan, and Pal (2011).

The technique represented here uses modified projection profiles which is based on histogram processing of previously obtained concentrations of black pixels in areas of interest. The main sliding window based method for obtaining the concentration of black pixels is used, with its variations, on all levels of segmentation. The proposed approach itself is semi-automatic since it uses the threshold values for necessary parameters. In general, the important gain in choosing this approach lies in time complexity. Since the proposed approach uses a one-pass processing of the pixel intensity values for each level of segmentation, processing time of the technique is less than one second even for large document images. It allows the use of this approach in real-time systems. Results using the new technique are presented focusing on the aspect of segmentation accuracy and the aspect of the time complexity, and are analyzed and compared with state-of-the-art approaches. It is shown that the proposed approach outperforms state-of-the-art approaches in most cases considering the segmentation accuracy. When it comes to the time complexity, the proposed approach proved to be more efficient than state-of-the-art approaches which makes the proposed approach suitable for real-time tasks. Evaluation of the technique's performance is achieved using the sets of the ground-truth historical machine-printed documents, and visual demonstration of the segmentation performance is achieved using the original machine-typed Nikola Tesla's documents.

This paper is organized as follows: in Section 2 the description of the related works is given. Section 3 offers the flowchart of the technique and short a explanation of each flowchart block. In Section 4 the complete mathematical background of the proposed approach is presented, including the complete pseudo-code. In Section 5 the new method is compared with state-of-the-art approaches and the numerical results are provided. Finally, discussion about the proposed approach, results, and the future work is given in Section 6.

2. Related works

This section provides a description of the approaches closely related to the new technique. It includes a description of the character segmentation approaches for machine-printed and machine-typed documents, since the proposed approach can be also applied to machine-printed document images.

Antonacopoulos and Karatzas (2005) presented a character segmentation approach for typewritten historical documents based on the usage of the horizontal projection profile of each word segment. First separator is predicted based on the expected character box width and is refined, taking into account the location and strength of the projection minima. Next separators are determined using the previous separator.

Nikolaou et al. (2010) presented an approach for segmentation of historical machine-printed documents. This approach uses an Adaptive Run Length Smoothing Algorithm (ARLSA) in order to solve the problem with document layout. Other parts of this

approach include detection of noisy areas and punctuation marks, detection of possible obstacles formed from background areas and the usage of skeleton segmentation paths for solving the problem of connected characters. This approach is compared with other segmentation approaches and results show that this technique provides better segmentation results.

Vamvakas et al. (2008) proposed an OCR methodology for recognition of the machine-printed and handwritten documents. The character segmentation aspect of this methodology exploits the standard image filtering algorithms, which are followed by top-down segmentation approach in order to detect text lines, words, and characters. Finally, a clustering scheme is used for grouping the characters of similar shape.

Min-Chul et al. (1999) proposed the character segmentation approach for machine-printed documents. This uses the recognition-based segmentation, combined with heuristic and holistic methods in order to separate touching characters. Line adjacency graph (LAG) is exploited for finding the blobs as connected components of the LAG.

Park et al. (2001) proposed an approach for extraction of the characters from machine-printed document images. Characters are extracted using several features such as size, elongation, and density. Classification is obtained using the run-length frequency of the image component.

Gupta et al. (2007) presented an approach which relates to the image binarization which is used in document image processing in order to search historical printed document images by keywords. A variety of binarization methods are used in order to construct this approach. It is shown that the Otsu method and other Otsu-based methods give the best results on average.

Younes and Abdellah (2015) proposed three methods for segmentation of handwritten text into lines. They used segmentation based on line classification, segmentation using the sliding window method, and segmentation using the hybrid method. Taking both, time complexity and segmentation accuracy results into account, the results show that a hybrid method gives best performance overall. The authors exploit the sliding window based method which is also used in this work.

Olszewska (2015) proposed a new approach for real-time segmentation and recognition of digits in images and videos. The method consists of two stages. In the first stage, characters are extracted from real-world scenes using the active contours. The second stage represents the recognition of previously extracted characters using the template matching. This system performances proved to be better than state-of-the-art methods in case of automated identification of players' numbers in sport datasets.

Li, Li, Pan, Chu, and Roddick (2015) presented a two-stage procedure for character segmentation. The first stage has a goal to extract texture features of each block based on Gabor filter. Afterwards, the second stage should classify the previously extracted texture features using the Fisher classifier.

3. Efficient character segmentation approach

In this section a complete efficient character segmentation approach for machine-typed documents is presented in short. The algorithm consists of three main stages: manual skew correction, using a new general, ultra-fast architecture for geometrical image transformations, image filtering, which represents a pre-processing stage and exploits already well known literature algorithms for image binarization, and noise reduction (Pratt, 2006; Russ, 2009), and segmentation logic, which uses histogram processing (Gonzalez & Woods, 2008; Younes & Abdellah, 2015) in the process of determining character positions in the given document image. The flowchart of the complete proposed character segmentation approach is shown in Fig. 1.

The input to the proposed character segmentation technique are a document image and threshold values which are used in order to control the segmentation process, since the proposed character segmentation approach is semi-automatic. Threshold value T_{hswl} determines the segmentation criteria in the process of line segmentation. Threshold value T_{hsww} is used to determine the segmentation criteria in the process of word segmentation. T_{hswc} is used to control the segmentation criteria in the first part of segmentation inside words. The second part uses the threshold value T_{hcw} which represents the average estimated character width. Each block from the flowchart is described in short in the following subsections and detailed representation of the new technique, including the mathematical background, is given in Section 4.

3.1. Manual skew correction

The first important stage in the proposed character segmentation is skew correction of the document image. This step is a crucial for further processing since the further processing is performed exclusively on de-skewed document images and would be impossible without this stage. In order to perform this task, a generalized ultra-fast approach for geometrical image transformations is proposed and fast implementation is achieved using pointer arithmetic and a new highly optimized low-level machine code routine. This implementation is adapted for image rotation, which is used for the skew correction. Skew correction is applied manually, since the existing skew estimation approaches are not suitable for real-time tasks.

3.2. Grayscale conversion

First block in the image filtering stage represents the process of image conversion to grayscale image as the necessary step in the process of image binarization. In the proposed approach, the simplest standard algorithm for grayscale conversion is applied since the choice of the grayscale algorithm does not affect the further processing (Pratt, 2006; Russ, 2009).

3.3. Binarization

After the grayscale conversion, binarization is performed using the gray-level transformation function. This function is thresholding function and decides whether the pixel will take the intensity value equal to 0 or equal to 1 (Fernández-Caballero et al., 2012; Gonzalez & Woods, 2008; Gupta et al., 2007).

3.4. Noise reduction

Noise reduction is the process of removing the undesirable isolated black pixels in the previously obtained binary image. The proposed approach exploits the algorithm which eliminates only the black pixels which do not have other black pixels as 8-neighbors (Gonzalez & Woods, 2008). This noise reduction algorithm proved to be sufficient for further processing.

3.5. Line segmentation

The process of document image segmentation starts with a line segmentation, namely with separation of the given document image into text lines. The sliding window based method with vertical sliding is used for this task (Younes & Abdellah, 2015). After the line borders are obtained, the process of correction is applied in order to save the character data.

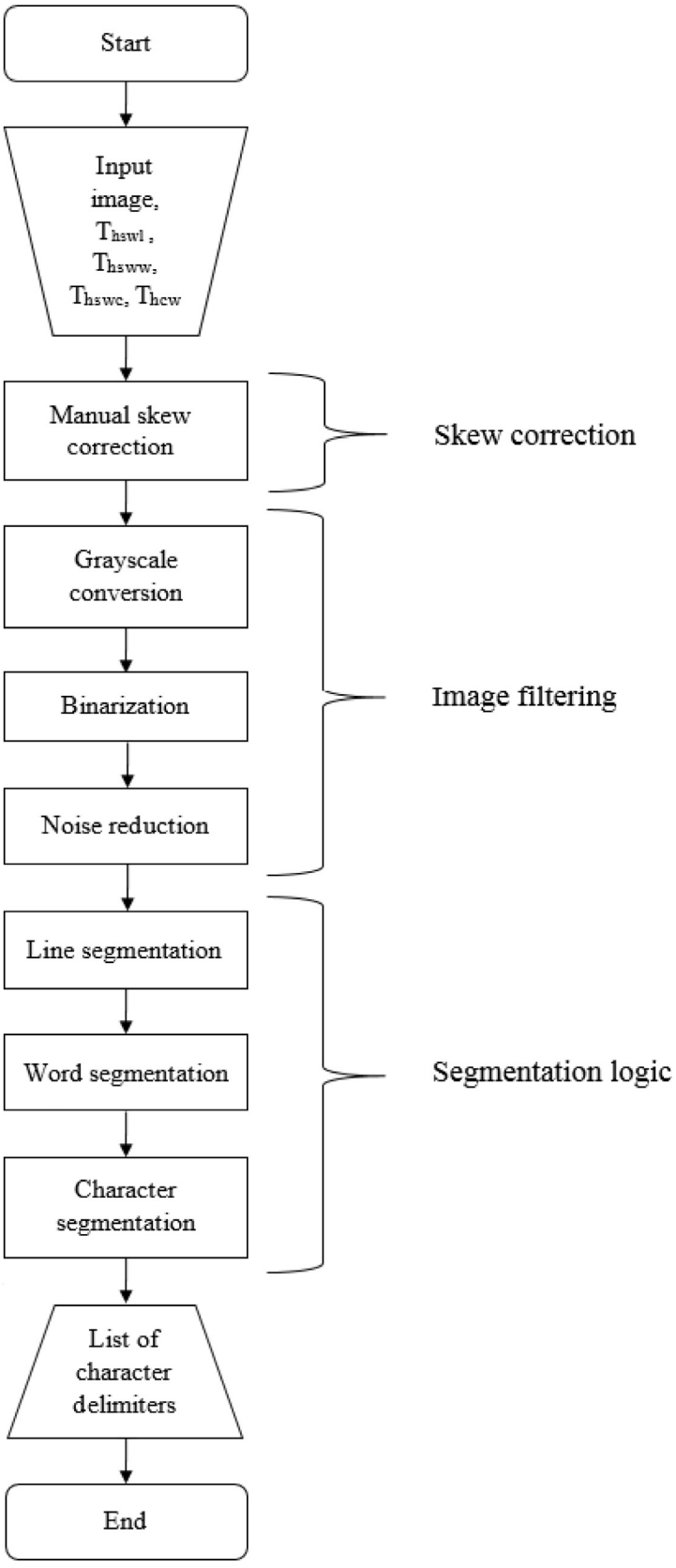


Fig. 1. The proposed efficient character segmentation approach.

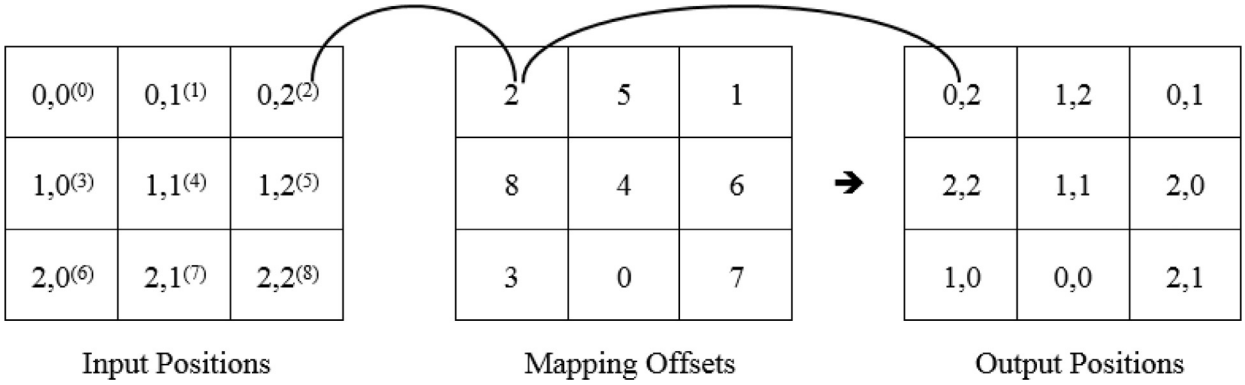


Fig. 2. Ultra-fast architecture for geometrical image transformations.

3.6. Word segmentation

Word segmentation is performed for each determined document line using the sliding window based method with horizontal sliding, which is followed by histogram processing (Gonzalez & Woods, 2008; Russ, 2009). The idea of this approach is to find bigger spaces inside lines, which will represent the spaces between words. The size of targeted spaces is controlled by the threshold value. As a part of the word segmentation, the word alignment is performed since documents can have dislocated lines.

3.7. Character segmentation

The final process represents the separation of the individual characters inside already separated words. The first part of this stage is identical to the previous process of word segmentation, except that it uses a different threshold value for sliding window size since the spaces between characters are smaller than spaces between words (Gonzalez & Woods, 2008; Russ, 2009). The second part of the character segmentation process represents the proposed decision-making logic and its task is to decide which of the potential delimiters between characters will be chosen as delimiters.

4. Mathematical background of the proposed approach

This section provides a detailed description of the proposed approach. As previously mentioned, the proposed approach consists of three stages. The manual skew correction stage exploits the proposed generalized ultra-fast architecture for geometrical image transformations. This approach, as a part of the proposed character segmentation approach, proved to be very efficient and its implementation is highly optimized and adapted for image rotation. The image filtering stage uses the common techniques in order to obtain a suitable binary image which is the input to the segmentation stage. Finally, the segmentation stage, which is the most complex stage, exploits the projection profiles technique which uses the sliding window for obtaining the initial information which are processed using the histogram processing techniques. The stages are thoroughly discussed in the following subsections.

4.1. Manual skew correction

The proposed approach uses the manual skew correction since the skew estimation techniques are generally slow for real-time

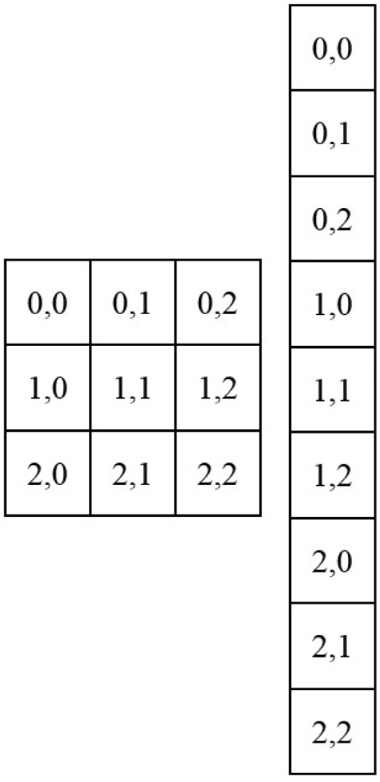


Fig. 3. Linear image representation.

tasks. In order to perform the manual skew correction in real-time, ultra-fast architecture for geometrical image transformations is used. This architecture is generalized and can also be used for other spatial transformations. The architecture scheme is shown in Fig. 2.

Image rotation is implemented using this architecture. The architecture is based on the usage of mapping offsets which represent the transformation matrix for chosen transformation. The transformation matrix is a matrix of offsets where each offset represents the offset relative to the first element of the input matrix. Using the mapping offsets each input position is mapped to the specific output position. In practice, this architecture provides that each pixel in the input image can be mapped to the pre-computed position in the output image. Calculation of the mapping offsets for chosen transformation with specific parameters is performed at

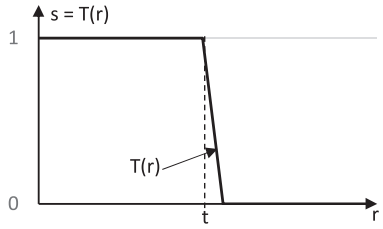


Fig. 4. Thresholding function.

1	1	1
1	0	1
1	1	1

Fig. 5. 3×3 filter mask used for noise reduction.

the start and whenever the transformation needs to be applied, already calculated mapping offsets are used. In this case, mapping offsets are calculated using the standard rotation transformation pair:

$$S_x = xcos(\theta) + ysin(\theta) \tag{1}$$

$$S_y = -xsin(\theta) + ycos(\theta) \tag{2}$$

where angle θ is desired angle which is set manually. Since the mapping offsets are valid for a given angle of rotation, it would be necessary to calculate the mapping offsets for different angles to make it possible to combine different mapping offsets for achieving a rotation for desired angle. This approach proved to be very efficient and does not depend on the type of image transformation.

The disadvantage of the presented ultra-fast architecture for geometrical image transformations is the memory usage. In order to achieve the fastest possible computational performances, the mapping offsets and other support lookup tables are loaded at start-up and perpetually held in memory.

In order to achieve the fast implementation of this approach for geometrical image transformations, the image is represented as a one dimensional array. This linear image representation is shown in Fig. 3. This image representation provides the direct memory access to the pixel intensity values using pointer arithmetic. Furthermore, implementation of the image rotation is achieved using highly optimized low-level machine code.

Beside the fact that the image is represented as a one-dimensional array of pixel intensity values, in order to achieve faster memory access, each pixel intensity value is also represented as 32-bit integer value. This value is determined as follows:

$$P_{VAL} = R * 256^2 + G * 256^1 + B * 256^0 \tag{3}$$

where R, G, and B are pixel intensity value components.

Implementation of both parts of the skew correction approach exploit the lookup support tables for values of trigonometric functions. This is a common way to avoid multiple calculations using the same parameters inside big loops. Image rotation is performed using the previously described ultra-fast architecture for geometrical image transformations. This approach proved to be very efficient and performing 50 times faster than the standard approach for image rotation. In order to achieve the highest computational performances, highly optimized low-level machine code implementation is used. The following listings show the Pascal implementation of the ultra-fast image transformation architecture adapted for image rotation

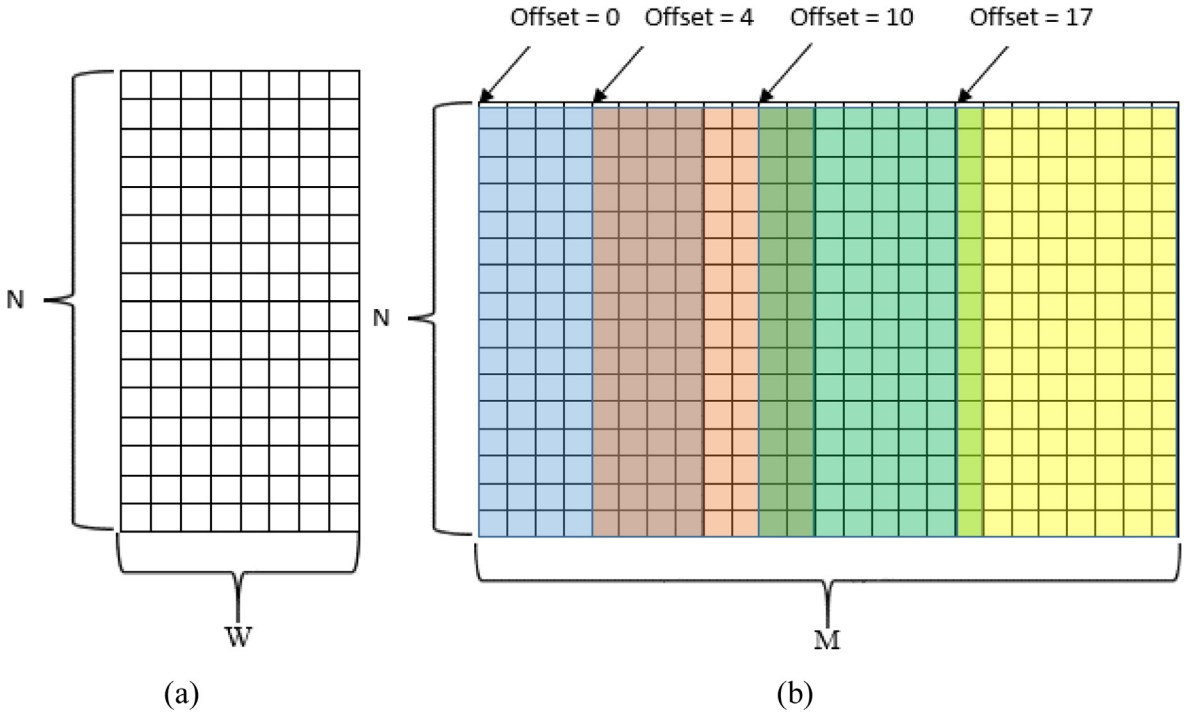


Fig. 6. Illustration of the sliding window propagation along the image: (a) Sliding window, (b) Sliding window moving from left to right with different offsets.

and optimized machine routine for this ultra-fast architecture:

```
for J := 0 to N do                                  {Number of transformation arrays}
begin
  SetLength(R[J], Count);                           {Set each transformation array to be equal
                                                    to the number of pixels}
  S := DPtr^;                                       {Get value for Sin from lookup table}
  Inc(DPtr);                                        {Increment pointer}
  C := DPtr^;                                       {Get value for Cos from lookup table}
  Inc(DPtr);                                        {Increment pointer}
  RPtr := @R[J, 0];                                 {Pointer to current transformation array}

  X := Trunc(HTemp * C + WTemp * S);                {Determine the coordinates of the image
  Y := Trunc(-HTemp * S + WTemp * C);               center after rotation}
  OffsetX := WTemp - Y;                             {Determine the offset from image center
  OffsetY := HTemp – X;                             for both coordinates}

  ImageStartPtr := @Image[0];                       {Source image pointer}
  Ptr1 := @PosMap[0];                               {Pointer to support lookup table}
  Ptr2 := @PosMap[1];                               {Pointer to support lookup table}

  for I := 0 to Count - 1 do
  begin
    X := Trunc(Ptr1^ * C + Ptr2^ * S) + OffsetY;    {Determine both coordinates after rotation}
    Y := Trunc(-Ptr1^ * S + Ptr2^ * C) + OffsetX;
                                                    {If coordinates are valid}
    if (X >= 0) and (X < Height) and (Y >= 0)
and (Y < Width) then                                {Store offset to transformation array}
      RPtr^ := X * Width + Y
    else                                            {Store -1}
      RPtr^ := -1;
                                                    {Increment Pointer}
    Inc(RPtr);                                      {Increment Pointer}
    Inc(Ptr1, 2);                                   {Increment Pointer}
    Inc(Ptr2, 2);
  end;
end;
```

```
  asm
    pushad                          {Push all registers to stack}
    mov ecx,Count                   {Number of pixels to process}
    mov esi,RPtr                    {Pointer to R transformation array}
    mov ebx,ImageSrcPtr             {Source image pointer}
    mov edi,ImageDstPtr             {Destination image pointer}
  @main:                            {Main loop}
    LODSD                           {Load current offset from R transformation array}
    mov edx,eax                     {Save current offset}
    or eax,eax                      {Is it -1?}
    js @init                        {If true, jump to label @init}
    shl edx,2                       {Offset * 4}
    mov eax,[edx+ebx]               {Calculate the final offset and load value to EAX}
    STOSD                           {Store loaded value from EAX to destination}
    dec ecx                         {Decrement counter}
    jnz @main                       {If not zero, loop again through ECX}
    jmp @ex                         {Else, jump to label @ex}
  @init:                            {Label @init}
    mov eax,WHITE_COLOR             {Store white color definition to EAX}
    STOSD                           {Store value from EAX to destination}
    dec ecx                         {Decrement counter}
    jnz @main                       {If not zero, loop again through ECX}
  @ex:                              {Label @ex}
    popad                           {Pop up all registers from stack}
  end;
```

4.2. Image filtering

Image filtering is a common stage in image segmentation tasks. The proposed character segmentation approach uses standard methods in the spatial domain for obtaining the binary image. The mathematical background of the methods which are used in the image filtering stage is provided in the following subsections.

4.2.1. Grayscale conversion

In the process of image binarization, grayscale conversion is performed on the original image in order to obtain a grayscale image with 256 Gy color levels, ranging from 0, which represents

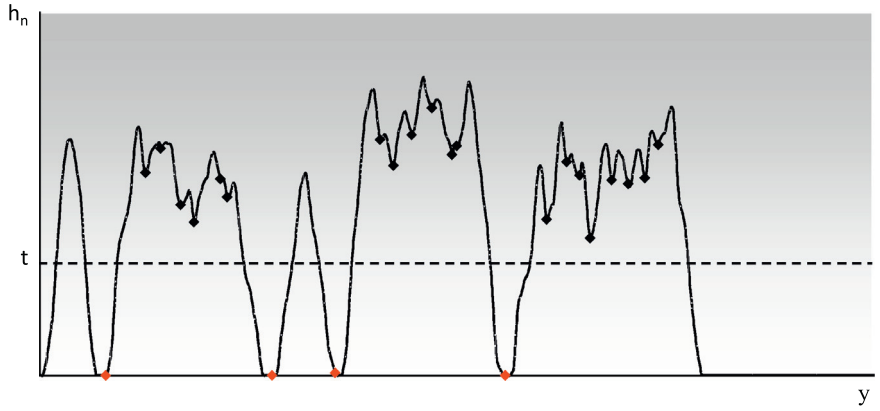


Fig. 7. Example of histogram used in the process of decision making.

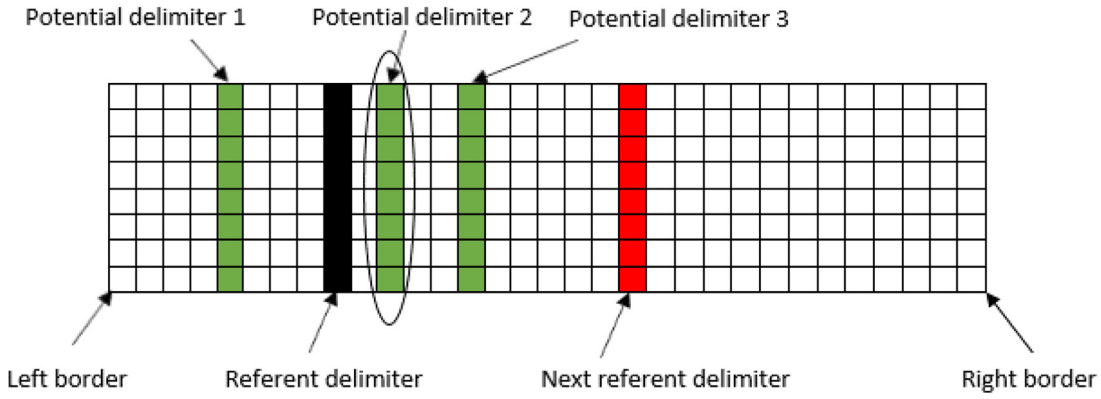


Fig. 8. Decision-making logic for character segmentation.

completely black, to 255, which represents completely white. Suppose that f is the starting 24-bit image of size $M \times N$, and that g is the 8-bit image of the same size. Each pixel g (x, y) in resulting grayscale image g will have the following intensity value:

$$g(x,\ y) = \max\left(\mathrm{f}(\mathrm{x},\ \mathrm{y})_{\mathrm{red}}, \mathrm{f}(\mathrm{x},\ \mathrm{y})_{\mathrm{green}}, \mathrm{f}(\mathrm{x},\ \mathrm{y})_{\mathrm{blue}}\right) \tag{4}$$

for $x = 0, 1, 2,\ldots, M-1$ and $y = 0, 1, 2,\ldots, N-1$, where max determines the maximum of three components of the pixel intensity value. The type of the grayscale conversion does not significantly affect further processing, thus the choice of the grayscale conversion for the proposed approach is not of significant importance.

4.2.2. Binarization

The image binarization process is performed on grayscale image using a thresholding function T, which is a gray-level transformation function of the form:

$$T(r) = \begin{cases} 1, & 0 \le r \le t \\ 0, & t < r \le r_{\max} \end{cases} \tag{5}$$

where t is a binarization threshold value and r_{max} is a maximal value of the pixel intensity. In the concrete case, suppose that input to this part of the approach is an 8-bit grayscale image *f* with pixel intensity values in range [0, 255], and *g* is a binary image where 0vvalues represent white pixels (image foreground), and 1 values represent black pixels (image background). Also, suppose that the threshold value used for obtaining pixel intensity values in the resulting image is equal to t. Each pixel in image g will take the intensity value:

$$g(x,\ y) = \begin{cases} 1, & 0 \le f(x,\ y) \le t \\ 0, & t < f(x,\ y) \le 255 \end{cases} \tag{6}$$

for $x = 0, 1, 2,\ldots, M-1$ and $y = 0, 1, 2,\ldots, N-1$, and t being a binarization threshold value. The thresholding function used in this process is shown in Fig. 4.

The threshold value that gives the best results depends primarily on the quality of the document image. If the document image is of low quality, a higher threshold value is required. The threshold value used in image binarization is not considered as an input, since it can be constant in most cases.

4.2.3. Noise reduction

In the end of the image filtering process, a simple noise reduction algorithm is applied in order to eliminate undesirable isolated black pixels that could cause problems in further processing. This noise reduction algorithm eliminates isolated black pixels which do not have other black pixels as 8-neighbors. Suppose that f is a binary image of size $(M+2)\times(N+2)$ obtained after the binarization process and expanded with zeros on all four sides due to border processing. Suppose that w is a filter mask of size $m \times n$, where $m = 2a+1$ and $n = 2b+1$, where a and b are positive integer val-

ues. Each pixel g (x, y) will have the following intensity value:

$$g(x,\ y) = \begin{cases} f(x,\ y), & \sum_{s=-a}^{a}\sum_{t=-b}^{b} w(s,\ t) f(x+s,\ y+t) > 0 \\ 0, & \sum_{s=-a}^{a}\sum_{t=-b}^{b} w(s,\ t) f(x+s,\ y+t) = 0 \end{cases} \quad (7)$$

In this case, the filter mask w is 3×3 in size, with 1 s as edge elements, where a and b are also equal to 1, noting that the middle element of the filter mask is w (0, 0). The filter mask is shown in Fig. 5.

The dimensions of the filter mask and position of the 1 s determine the noise reduction method. The 1 s determine which of the pixel intensity values will be taken into consideration when the filter mask is applied to the given image block. The filter mask shown in Fig. 5 is chosen because it checks only the closest neighbor pixels to the given pixel in the middle of a filter mask, and based on their intensity values, determines if it is noise or not. Thus, this filter mask preserves the important document image pixels and removes only the isolated black pixels which are probably not useful. For old low quality historical documents into account, it is expected that this noise reduction method risks removing some useful image data, but in general, it is suitable for the proposed character segmentation approach.

4.3. Segmentation logic

The third stage of the proposed approach is the segmentation logic and it is the core of the algorithm. This part of the technique is based on histogram processing. The histogram is computed using the values obtained after performing the projection profiles technique using the sliding window for calculation of the concentration of black pixels in the area of interest (Younes & Abdellah, 2015). This method represents the key part of the proposed approach since it is used on all levels of segmentation. In Fig. 6, the sliding window is shown in image (a) and the sliding window propagation along the area of interest is shown in image (b).

The window slides from left to right by 1 pixel and the sum of black pixels in the window is calculated for each slide. Using these values the histogram is computed. The main importance of this method is the possibility of determining the image areas with the lowest concentration of black pixels. Considering this fact, it is clear that this method can be applied in order to determine the positions of spaces between lines, words, and characters. Fig. 6 describes horizontal sliding of the sliding window, which is used for determining the spaces between the words and characters. In contrast, vertical sliding of the window is used for determining the spaces between the document lines.

4.3.1. Line segmentation

One of the central tasks which the new technique must perform is the line segmentation of the document image. The approach used here exploits sliding window with vertical sliding. Suppose that the sliding window is size of $N \times H$, where N is the width of the binary image f, and H is the height of the sliding window. The concentration of black pixels in sliding window is calculated as:

$$s_n = \sum_{s=0}^{H-1}\ \sum_{y=0}^{N-1} f(x+s,\ y) \quad (8)$$

Values s_n represent the values in the array of all sliding window concentrations of black pixels, S. Deciding which offsets will used as potential delimiters between lines is made based on:

$$d_x = \begin{cases} x + \left[\frac{H}{2}\right], & s_n < T_{hswl} \\ -1, & \text{otherwise} \end{cases} \quad (9)$$

where d_x is offset relative to the top of the image, H is the height of the sliding window, and $\lfloor H/2 \rfloor$ represents integer division. More precisely, values d_x are offsets of the middle lines of the windows which slide vertically along the document image. Only the offsets which belong to the sliding window with concentration of black pixels lower than the threshold value T_{hswl} are taken into consideration.

The borders of potential document image lines are obtained by performing top down image analyses, more precisely by analyzing the offsets d_x obtained in the previous step. Using this offset analyses and considering the fact that document line height is greater than d pixels, the closest offsets on distance greater than d are determined. Processing is performed starting from the first offset. Once the next offset on distance greater than d is found, both offsets are taken as upper and lower borders of the document line. The process is repeated until the last offset. In the end of this step, the segments which represent the document line borders are obtained.

Sometimes a line segmentation can cut document characters due to a low concentration of black pixels in upper or lower part of the character. This happens in situations when lines are too close to each other and a higher value for the threshold is necessary to separate them. For that reason, it is necessary to spread previously obtained line borders and save character data. This is achieved by reducing the distance between the lower border of the previous line and upper border of the next line to 1, as follows:

$$U = U - \left[\frac{U-L}{2}\right] \quad (10)$$

$$L = L + \left[\frac{U-L}{2}\right] + (U-L)\ mod\ 2 - 1 \quad (11)$$

where U represents the offset of the upper border of the given line and L represents the offset of the lower border of the previous line. Finally, the upper border of each line is moved down to the top of the closest character and the lower border is moved to the bottom of the closest character.

4.3.2. Word segmentation

The key part of the proposed approach is focused on analysis of each detected document image line and segmentation of lines to words and after that to characters. This part also includes the detection of the punctuation characters, some of which may be even recognized by the concentration of black pixels in specific areas of the line.

In this part of the proposed approach, the sliding window with horizontal sliding is used. Suppose that sliding window size of W x H is taken, where W is the width of the sliding window and H represents the height of the current line. Concentration of black pixels in the sliding window is calculated as:

$$h_n = \sum_{x=H1}^{H2}\ \sum_{t=0}^{W-1} f(x,\ y+t) \quad (12)$$

where h_n is a number of black pixels in the given sliding window which slides horizontally along the document image line. After the values h_n are calculated and successive repeating values merged into one value at the position of the rightmost value in that group, the histogram is computed for each document image line. The key of the histogram analyses is in determination of the local minima which represent the valleys in the histogram, in this case the minima of the concentration of black pixels in a given line. The local minima less than threshold value t are taken. An example of histogram is shown in Fig. 7.

These values represent delimiters between the words. Suppose that $h_1, h_2, \ldots, h_n$ is a sequence of histogram values. The set of all

histogram valleys is described as follows:

$$V = \{h_i | h_{i-1} > h_i \wedge h_{i+1} > h_i\} \quad (13)$$

In the end of word segmentation process, obtained delimiters which are located between the words are moved to the left and to the right to the closest characters, and that way precise word positions are determined.

4.3.3. Character segmentation based on decision-making logic

Once the word segmentation is performed, it is followed by character segmentation. The approach used here consists of a word alignment process, the already described histogram based method, and the proposed decision-making logic. Due to the use of old typing machines, sometimes there are anomalies in typed documents such as line curvatures, where usually one part of a pixel in a given line is dislocated. This anomaly can cause problems with the output, because some characters could be sent to the recognition stage with a loss of information.

As it was the case with word segmentation, the sliding window based method with horizontal sliding is applied for each previously separated word and the histogram is computed. This time the sliding window of smaller width is used than that used for word segmentation. In fact, a longer sliding window width is used for determining larger areas with a small concentration of black pixels (bigger spaces). On the other hand, sliding window with smaller width is used for determining the smaller spaces and is suitable for character segmentation within a word. After the determination of the local minima, which represent potential delimiters between characters, the threshold value T_{hcw} for document letter width is used for determining the word length. The average letter width is calculated as follows:

$$C_n = \left[\frac{W_w}{T_{hcw}}\right] \quad (14)$$

$$C_{wavg} = \left[\frac{W_w}{C_n}\right] \quad (15)$$

where C_n is the assumed number of characters in a word, calculated using the threshold value T_{hcw}, W_w is word width in pixels of the given word and C_{wavg} represents the average character width used in further processing. Word length is necessary because it can be used to determine the number of delimiters in a word. Since we know the assumed word length calculated using a threshold value T_{hcw}, it is possible to determine the average character width, but this time with a known word length.

The given word is separated starting from the left border and taking delimiter at distance equal to obtained average character width as referent delimiter. The crucial part in character segmentation is the proposed decision-making logic. The algorithm has to decide which one of the potential delimiters should be chosen for a delimiter. This process is illustrated in Fig. 8.

The proposed decision-making logic looks for the potential delimiter that is closest to the referent delimiter. Searching for a delimiter is limited by maximal offset allowed. If there are no other potential delimiters surrounding referent delimiter on allowed distance, referent delimiter will be chosen to be the delimiter and the next referent delimiter will be set on distance equal to the average character width from the chosen delimiter. Suppose that d_1, d_2,...,d_n is the sequence of potential delimiter offsets, where each of them is calculated as:

$$d_i = h_i + \left[\frac{W}{2}\right] \quad (16)$$

where h_i represents a given histogram valley and W is the width of the sliding window. The offset of the chosen delimiter is determined as follows:

$$j = \underset{i}{argmin}\left(\left|d_i - d_{ref}\right|\right) \quad (17)$$

$$d = \begin{cases} d_j, & \left|d_j - d_{ref}\right| \le t \\ d_{ref}, & \left|d_j - d_{ref}\right| > t \end{cases} \quad (18)$$

where j is the index of the chosen delimiter in a set of all potential delimiters, d_{ref} is the referent delimiter and t is the threshold value for maximal allowed distance between the closest potential delimiter and the referent delimiter.

In case of the punctuation marks, the position of black pixels inside the sliding window is considered, e.g. in case of periods and commas, the important feature is that most of the black pixels are concentrated in the lower half of the document line and in case of dashes, the important feature is that black pixels are concentrated in the second third of the document line.

The pseudo-code that describes the complete proposed approach is as follows:

```
SLIDING-WINDOW-METHOD
Input:
    Area A, parameter Thsw.
Output:
    List of histogram minimums' positions V;
 1: S ← Ø, V ← Ø
 2: while not end of A do
 3: Compute concentrations Si of black pixels in sliding window size of
WIDTH(A) x Thsw in the given area A in binary image f
 4: end while
 5: Compute histogram using values Si
 6: REMOVE-SUCCESSIVE-DUPLICATES(S)
 7: for each value Si do
 8: if Si < Si-1 and Si < Si+1 then
 9: V ← I
10: endif
11: end for
12: return V

CHARACTER-SEGMENTATION-APPROACH
Input:
    Image f, parameters Thswl, Thsww, Thswc, Thcw.
Output:
    List of delimiters DELIMITERS;

 1: DELIMITERS ← Ø
 2: for each pixel f(x, y) do
 3: f (x, y) = MAX(RED[f (x, y)], GREEN[f (x, y)], BLUE[f (x, y)])
 4: if f(x, y) ≤ Thb then
 5: f(x, y) = 1
 6: else
 7: f(x, y) = 0
 8: endif
 9: end for
10: temp = f
11: for each black pixel temp (x, y) do
```

```
Z = SUM(8NEIGHBORS(temp(x, y)))
if Z == 0 then
f(x, y) = 0
endif
end for
for each horizontal scan line at position I in image f do
Calculate number of black pixels Z
if Z < T_hswl then
BORDERS ← I
endif
end for
FIX-BORDERS(BORDERS)
for each document image line L in BORDERS do
WORDS ← Ø
WORDS ← SLIDING-WINDOW-METHOD(L)
FIX-WORD-DELIMITERS(WORDS)
S ← Ø, V ← Ø
for each word W in WORDS do
D ← Ø
CHARACTERS ← Ø
CHARACTERS ← SLIDING-WINDOW-METHOD(W)
W_L = W_w div T_hcw
C_wavg = W_w div W_L
D_ref = START-POSITION(W) + C_wavg
while COUNT(D) ≠ W_L - 1 do
J = FIND-CLOSEST-DELIMITER-POSITION (CHARACTERS, D_ref)
if ABS(CHARACTERS [J] - D_ref) < T_hoffset then
D ← CHARACTERS [J]
Check if detected character between delimiters D_ref and D is punctuation mark [OPTIONAL]
D_ref = CHARACTERS [J] + C_wavg
else
D ← D_ref
Check if detected character between delimiters D_ref and D is punctuation mark [OPTIONAL]
D_ref = D_ref + C_wavg
endif
end while
DELIMITERS ← D
end for
end for
return DELIMITERS
```

The sign ← in pseudo-code represents the assignment operator for list.

5. Experiments

For the purpose of testing the performances of the proposed character segmentation approach, historical machine-printed and machine-typed documents are used. Document images used for evaluation of the character segmentation approaches?> and obtaining the numerical results which are used for comparison, consists of historical English, French, German, Polish, Spanish, and other machine-printed documents. Although the proposed approach is intended for machine-typed documents, it performs equally on both types of documents. Since the proposed character segmentation approach is designed as a part of a real-time OCR system for the needs of "Nikola Tesla Museum" in Belgrade, the original Nikola Tesla's documents are used for visual demonstration of performance. Numerical results are provided from the perspective of segmentation accuracy and the perspective of the time complexity, and the proposed approach is compared with state-of-the-art techniques for character segmentation. Since time complexity is analyzed, it is necessary to mention that experiments are conducted on PC machine with an AMD Quad Core Processor running at 3.1 GHz and 4GB RAM and Operating System Windows 8.1.

As it is already mentioned, the proposed approach is compared with state-of-the-art character segmentation techniques. Although the new method is based on using the projection profiles technique, the standard approach, which exploits the projection profiles technique on all levels of segmentation is also used for comparison. Also, the run length smearing (RLSA) based approach is implemented and the results from the perspective of character segmentation accuracy and the perspective of the time complexity are provided. The RLSA technique analyzes each scan line in the area of interest and each occurrence of the consecutive background (white) pixels, whose number is less than threshold value, is corrected and intensity values of those pixels are set to the intensity value of foreground (black) pixels. The RLSA technique can be used in different ways. It can be applied horizontally and vertically, or can be used in combination with the projection profiles technique. In the second case, RLSA has a goal to strengthen the histogram, more precisely to eliminate the local minimums with a low possibility of becoming the delimiters. Beside these techniques, the commercial products FineReader 12 (ABBYY FineReader OCR) and the Open Source OCROpus software (The OCROpus open source document analysis and OCR system) are used for comparison with the proposed character segmentation approach.

Character segmentation approaches are evaluated using the ground-truth sets of images which are manually chosen for each segmentation level. For text line segmentation 74 images (4363 text line segments), for word segmentation 58 images (20,584 word segments), and for character segmentation 22 images (34,623 character segments) are chosen.

The evaluation of the performances is performed using the matching score metric (Nikolaou et al., 2010; Phillips & Chhabra, 1999). This metric represents the ratio between the number of pixels in the segmented text line, word, or character region which belong to both the resulting image processed using the proposed approach and the ground-truth image, and total number of pixels in the determined segmented region in the ground-truth image. Using the threshold acceptance value for matching score, the segmentation accuracy can be controlled. The higher threshold value means a more rigid evaluation criteria. Table 1 shows comparative results for all segmentation levels when the acceptance threshold is set to 90.

Segmentation results show that the proposed approach, in most cases, perform better than state-of-the-art approaches. In case of text line segmentation, the new method uses the extended projection profiles technique and performs better than the standard projection profiles based approach. Taking the word segmentation results into account, the proposed approach based on adaptive projection profiles technique, ensuring better results than the standard projection profiles based approach. Character segmentation performed using the proposed approach proved to be much better than state-of-the-art techniques, since the proposed approach uses the projection profiles technique just as a pre-processing stage. The most important part for character segmentation is the proposed decision-making logic which gradually eliminates the possibility of big segmentation errors.

In order to provide further evaluation of the proposed approach performance, segmentation results are obtained for different categories of segmentation problems. These segmentation problems appear when specific documents are being processed. For this evaluation, multi-column documents, noisy documents, documents with non-constant spaces between text lines, word, and characters, documents with marginal text, documents with various font sizes, documents with ornamental characters and graphical illustrations, and warped and/or skewed documents are used. Tables 2–6 show detection rates for different approaches applied to different categories of segmentation problems.

The results show that in case of text line segmentation, the proposed approach gives better results than all state-of-the-art approaches. Considering the text line segmentation method which is used, segmentation problems can appear in the case of doc-

Fig. 9. Example of machine-printed historical documents used for evaluation of the proposed approach performances: (a) English document, (b) French document, (c) German document, (d) Polish document, (e) Spanish document.

Table 1
Comparison of the segmentation results for different approaches using the chosen sets of the ground-truth document images.

	Detection rate (%)		
	Text line segmentation	Word segmentation	Character segmentation
Projection profiles based approach	73.39	71.64	72.56
RLSA based approach	72.57	74.85	72.36
FineReader	69.35	76.18	64.82
OCROpus	75.18	79.48	74.43
Proposed approach	**80.58**	**77.57**	**86.14**

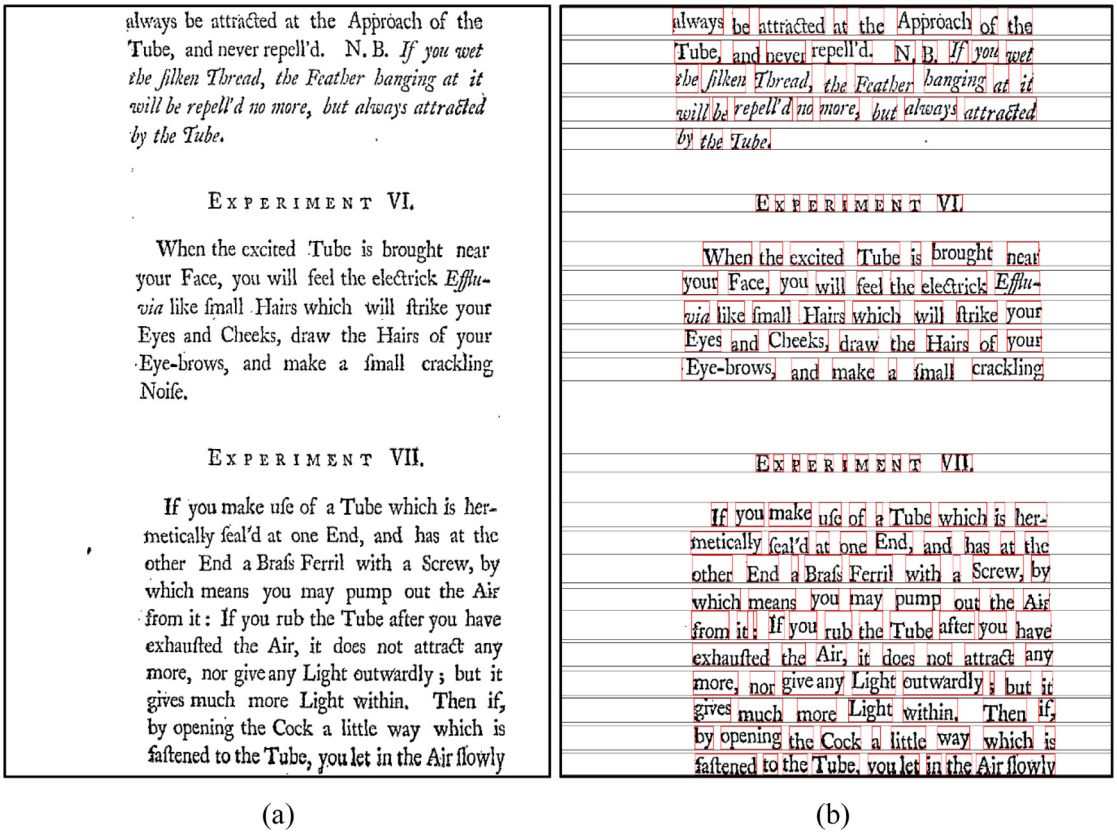


Fig. 10. Text line segmentation and word segmentation results: (a) Original document image, (b) Processed document image.

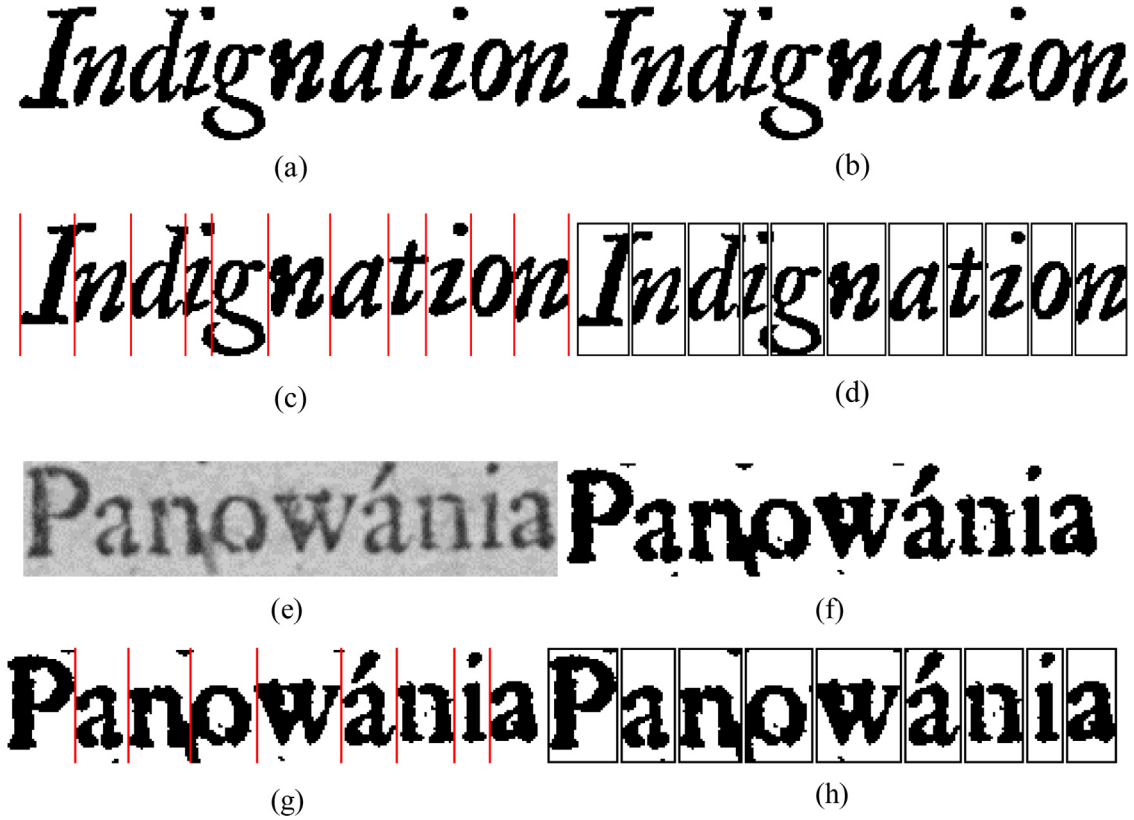


Fig. 11. Character segmentation in case of touching and overlapping characters (English and Polish words): (a) Original first word, (b) Binarized first word, (c) First word after segmentation, (d) Extracted characters from the first word, (e) Original second word, (f) Binarized second word, (g) Second word after segmentation, (h) Extracted characters from the second word.

Table 2
Segmentation results for different categories of segmentation problems (projection profiles based approach).

	Multi column	Noisy	Non-constant spaces	Marginal text	Various font sizes	Ornamental characters and graphical illustrations	Warped - skewed text
Text line segmentation	70.59	67.29	68.41	43.84	52.38	64.56	59.30
Word segmentation	66.37	58.14	62.67	44.63	58.62	60.12	55.23
Character segmentation	72.84	65.93	70.28	72.45	71.40	70.83	67.91

Table 3
Segmentation results for different categories of segmentation problems (RLSA based approach).

	Multi column	Noisy	Non-constant spaces	Marginal text	Various font sizes	Ornamental characters and graphical illustrations	Warped - skewed text
Text line segmentation	69.21	50.72	67.32	52.38	53.87	64.09	66.74
Word segmentation	68.17	52.73	64.71	55.28	62.44	68.78	62.39
Character segmentation	74.59	63.54	68.29	73.57	75.36	74.93	75.97

Table 4
Segmentation results for different categories of segmentation problems (FineReader).

	Multi column	Noisy	Non-constant spaces	Marginal text	Various font sizes	Ornamental characters and graphical illustrations	Warped - skewed text
Text line segmentation	67.55	55.63	59.31	50.56	70.52	66.27	57.43
Word segmentation	68.15	53.76	56.46	54.33	69.49	69.35	55.81
Character segmentation	70.26	44.03	58.39	65.80	70.34	66.89	64.73

Table 5
Segmentation results for different categories of segmentation problems (OCROpus).

	Multi column	Noisy	Non-constant spaces	Marginal text	Various font sizes	Ornamental characters and graphical illustrations	Warped - skewed text
Text line segmentation	71.13	73.86	68.44	47.52	69.81	72.38	62.65
Word segmentation	80.43	68.72	69.39	73.86	75.37	75.14	74.45
Character segmentation	72.67	64.66	66.60	70.62	72.56	73.78	73.67

uments with marginal text, skewed documents, and documents with graphical illustrations. This applies also to word segmentation, which can also have problems with noisy documents, but in most cases results are good. Character segmentation proved to be very accurate due to the nature of the proposed decision-making logic used for character segmentation.

Taking state-of-the-art approaches into account, projection profiles based approach is mainly affected by documents with marginal text and noisy documents. In case of a significant document skew, results can be unacceptable. RLSA based approach has problems with noisy documents and documents with non-constant spaces. Noisy documents are also a significant challenge

Table 6
Segmentation results for different categories of segmentation problems (proposed approach).

	Multi column	Noisy	Non-constant spaces	Marginal text	Various font sizes	Ornamental characters and graphical illustrations	Warped - skewed text
Text line segmentation	77.46	75.56	72.34	62.72	71.95	70.38	70.69
Word segmentation	75.24	71.49	73.37	52.18	68.64	72.32	65.36
Character segmentation	86.71	80.38	87.06	89.14	87.26	84.31	82.10

for FineReader technique, while OCROpus technique provides the worst results with documents containing marginal text.

As already mentioned, historical machine-printed documents are used for evaluation of the proposed approach performances. Some of these documents are shown in Fig. 9(a)–(e).

Text line segmentation is the most important part of the segmentation process since it is the entry point to the segmentation process and any segmentation error in this step would affect the further processing. Example of the text line and word segmentation using the proposed approach is shown in Fig. 10(a) and (b).

As can be observed from Fig. 10, word segmentation performs very well in most cases. Common problems appear in cases when words are close to each other. In this case, the word segmentation algorithm will treat such words as one single word. Thresholds which are used for controlling the word segmentation should be chosen carefully, since overly high values will fail to separate the words, and too low values could separate the characters inside words, which is erroneous.

Character segmentation is also very important part in the segmentation process. The new method uses the proposed decision-making logic which has proved to be quite accurate. The problem in character segmentation appears with touching and overlapping characters. Illustration of the character segmentation using the proposed approach is shown in Fig. 11(a)–(h).

Words shown in Fig. 11 contain touching and overlapping characters. The main characteristic of the proposed decision-making logic used for character segmentation is that this approach tends to eliminate possible segmentation errors from the start. The proposed decision-making logic uses information from the histogram previously computed, but it carefully chooses the delimiters which have a higher possibility of being correct. If correct threshold values are chosen, this method works very well in the case of touching characters. In the case of overlapping characters, it is not possible to avoid the loss of information since some characters will be cut in the process of character segmentation.

Another important aspect of the segmentation process is time complexity. Since the proposed approach is intended for use in real-time OCR system, the segmentation part should be efficient and able to perform segmentation tasks with low processing time. In order to achieve this requirement, the proposed approach is based on projection profiles which proved to be a relatively efficient technique. Projection profiles technique uses the sliding window in order to compute the histogram which is used for further processing. This process is used on all levels of segmentation, thus its processing time determines the overall processing time of the proposed approach. Fig. 12 shows the dependency of the processing time for the basic projection profiles technique used for one level of segmentation on the total number of pixels which are being processed. This graph shows that projection profiles technique processing time has a linear dependency of a total number of pixels which are being processed. Comparison of the processing time for different character segmentation approaches is given in Table 7.

Table 7
Comparison of the segmentation processing time for different approaches.

	Processing time (ms)		
	Projection profiles based approach	RLSA based approach	Proposed approach
1736×4872			
18 text lines	55.84976	128.36514	55.53348
29 text lines	89.12796	209.14872	88.03042
42 text lines	123.55864	304.77813	122.10422
56 text lines	164.13659	365.12239	162.74385
64 text lines	193.28441	405.28846	187.69245
5072×4312			
16 text lines	102.21373	233.44023	101.77729
25 text lines	181.36465	422.36867	180.22220
42 text lines	326.07428	736.58552	324.41320
50 text lines	377.49667	821.27488	374.59168
60 text lines	415.21295	912.07257	408.02283

Comparative results from the aspect of the time complexity show that the proposed approach is more efficient than state-of-the-art approaches. Results are given for two document images of different dimensions. Since the further segmentation highly depends on text line segmentation, results are provided for documents with different numbers of text lines. Based on results, the proposed method gives slightly better results than standard projection profiles based approach. The standard projection profiles based technique uses slightly slower text line segmentation and slightly faster character segmentation than the new algorithm. The word segmentation is the same from the aspect of the time complexity. The RLSA based approach used here is based on two-dimensional image processing. Since the image is analyzed in both directions, this approach gives worse results than both the standard projection profiles technique and the proposed approach. It should be mentioned that provided results for processing time are obtained using the optimized implementations. Visual demonstration of the proposed approach performances is achieved using original, machine-typed documents, authored by Nikola Tesla. These processed document images are shown in the Appendix section.

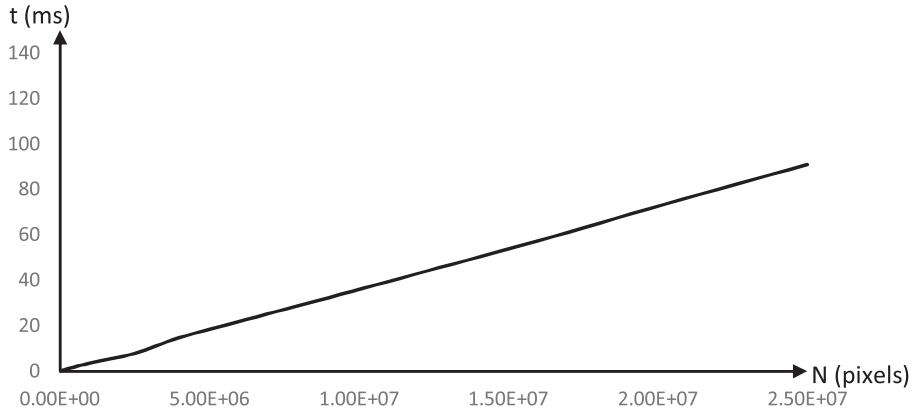


Fig. 12. Processing time for projection profiles technique as a function of a total number of pixels which are being processed.

6. Conclusions

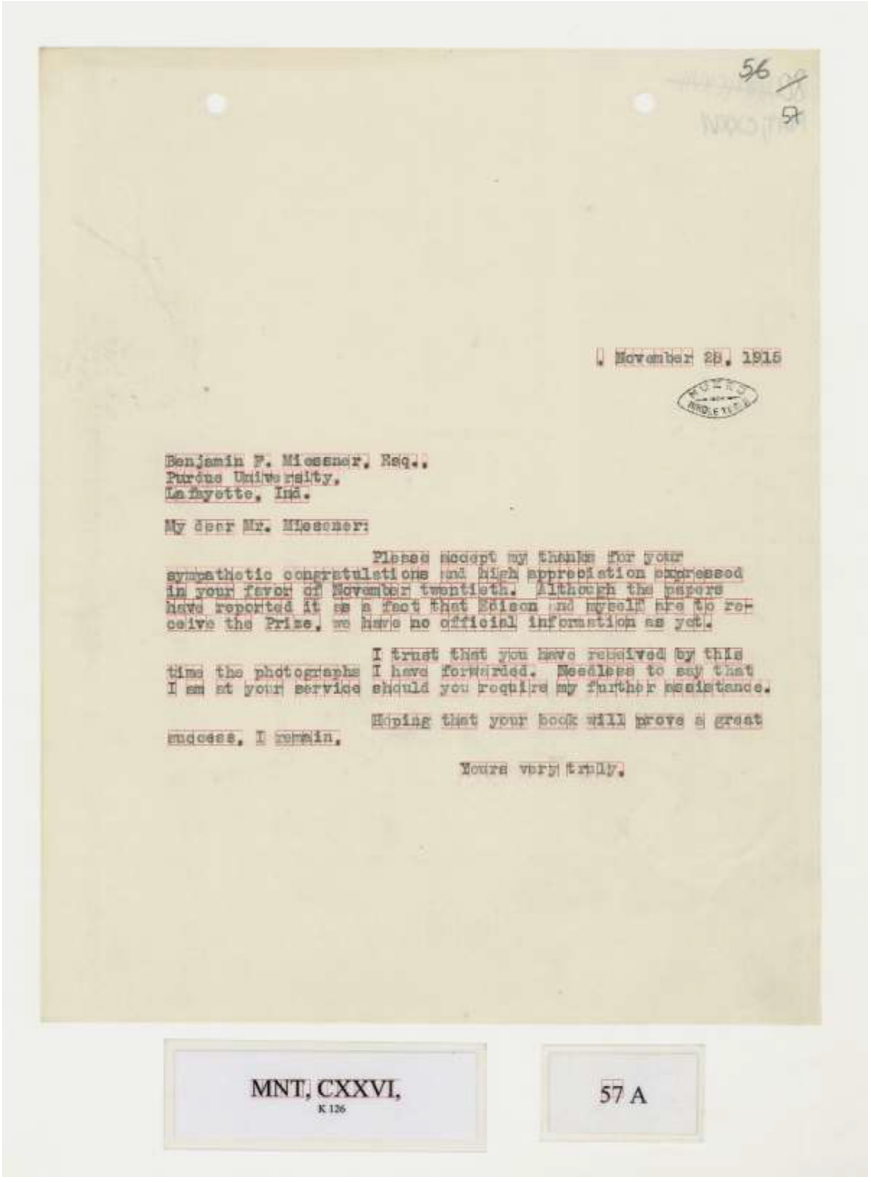
In this paper, an efficient approach for character segmentation of old machine-typed documents is presented. Although the main targets of the technique are machine-typed documents, the proposed approach can also be used for machine-printed documents. The method is semi-automatic, since threshold values are necessary to define some important parameters. In Section 2 a description about related works is given. In Section 3 flowchart of the proposed approach is shown and the functionality of its blocks is described. In Section 4 the mathematical background of the previously described flowchart blocks is explained in detail. This approach uses the image processing methods in the spatial domain and exploits the features of the document structure in the process of character segmentation. Three main stages of the proposed approach are manual skew correction, image filtering, and segmentation logic, which represents the core stage of the algorithm. Segmentation logic represents the modified projection profiles based approach which is based on calculation of the concentration of black pixels in the sliding window and making decisions using histogram processing. In Section 5 experimental results for historical machine-printed documents processed by the proposed approach and state-of-the-art approaches are given. The proposed method in most cases outperforms state-of-the-art approaches considering the segmentation accuracy. Furthermore, the proposed approach provides better results taking the time complexity into account. The new approach can process even very large document images in less than one second, which makes it to be suitable for real-time tasks. Beside numerical results, visual demonstration of the proposed approach performance is achieved using original Tesla documents from the "Nikola Tesla Museum" in Belgrade, since the method is designed for needs of the "Nikola Tesla Museum", namely for conversion of the original Nikola Tesla's documents to electronic form. Also, official evaluation of the proposed approach performances will be performed at the "Nikola Tesla Museum". All procedures for the proposed approach are optimized and adjusted to the old machine-typed documents, but the approach itself is general; namely the proposed approach can be used for any machine-printed documents. Our future work will be focused on improvement of the method, especially of segmentation logic stage, further optimization, including further automation (removing the manual stage), and its integration into the complete OCR system.

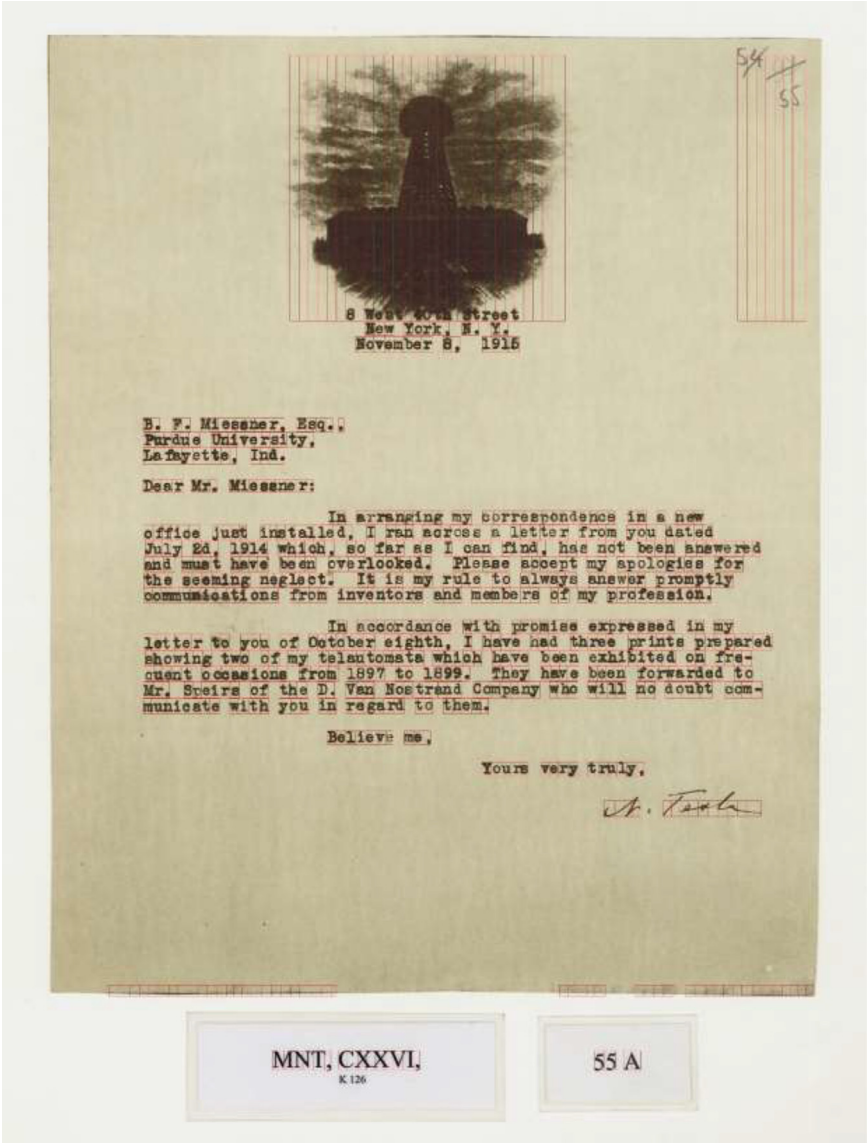
Acknowledgements

This paper is supported by the Ministry of Education, Science and Technological Development of the Republic of Serbia (Project III44006-10), Mathematical Institute of Serbian Academy of Science and Arts (SANU), The "Nikola Tesla Museum" (providing original typewritten documents of Nikola Tesla), and Pattern Recognition & Image Analysis Research Lab (PRimA) (providing ground-truth historical machine-printed documents). The authors greatly appreciate the anonymous referees for their very valuable and helpful suggestions on the earlier version of the paper. The authors also extend their thanks to Dr. Simon LE BLOND from Department of Electronic and Electrical Engineering at Bath University, UK, for professional English editing assistance.

Appendix

Illustration of the character segmentation results for the proposed approach using original Nikola Tesla's documents





54 / 55

8 West 40th Street
New York, N. Y.
November 8, 1915

B. F. Miessner, Esq.,
Purdue University,
Lafayette, Ind.

Dear Mr. Miessner:

In arranging my correspondence in a new office just installed, I ran across a letter from you dated July 2d, 1914 which, so far as I can find, has not been answered and must have been overlooked. Please accept my apologies for the seeming neglect. It is my rule to always answer promptly communications from inventors and members of my profession.

In accordance with promise expressed in my letter to you of October eighth, I have had three prints prepared showing two of my telautomata which have been exhibited on frequent occasions from 1897 to 1899. They have been forwarded to Mr. Speirs of the D. Van Nostrand Company who will no doubt communicate with you in regard to them.

Believe me,

Yours very truly,

N. Tesla

MNT, CXXVI,
K 126

55 A

58 60

May 23, 1916.

Mr. Nikola Tesla,
8 East 40th St.,
New York, N.Y.

My dear Mr. Tesla:

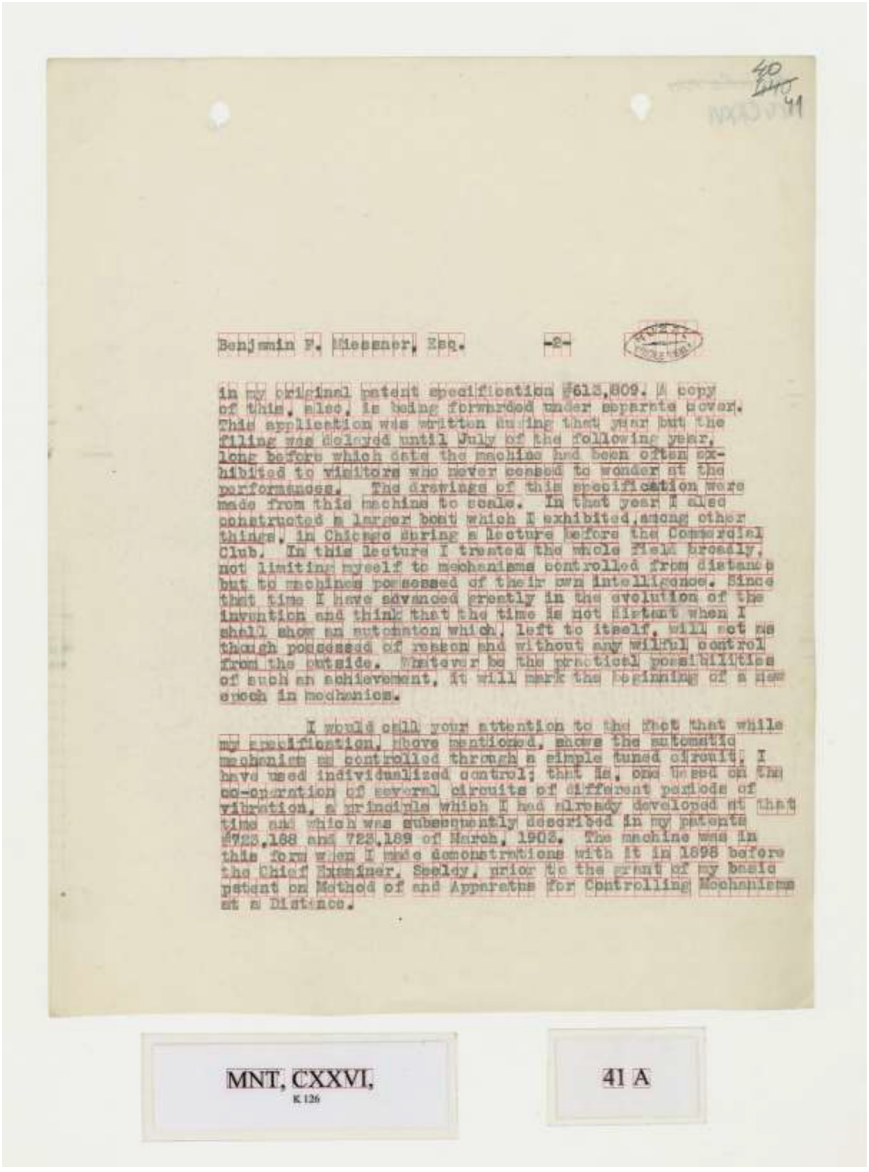
I have discontinued my work at the University and am now in search of employment.

I believe you know something of my qualifications. If you also know of any employer who may require such services as I am able to render you will favor me greatly by putting me in touch with him. I wish to thank you in advance for this favor.

Very truly yours,

MNT, CXXVI, K 126

60 A



Benjamin F. Miessner, Esq. -2-

in my original patent specification #613,809. A copy of this, also, is being forwarded under separate cover. This application was written during that year but the filing was delayed until July of the following year, long before which date the machine had been often exhibited to visitors who never ceased to wonder at the performances. The drawings of this specification were made from this machine to scale. In that year I also constructed a larger boat which I exhibited, among other things, in Chicago during a lecture before the Commercial Club. In this lecture I treated the whole field broadly, not limiting myself to mechanisms controlled from distance but to machines possessed of their own intelligence. Since that time I have advanced greatly in the evolution of the invention and think that the time is not distant when I shall show an automaton which, left to itself, will act as though possessed of reason and without any willful control from the outside. Whatever be the practical possibilities of such an achievement, it will mark the beginning of a new epoch in mechanism.

I would call your attention to the fact that while my specification, above mentioned, shows the automatic mechanism as controlled through a simple tuned circuit, I have used individualized control; that is, one based on the co-operation of several circuits of different periods of vibration, a principle which I had already developed at that time and which was subsequently described in my patents #723,188 and 723,189 of March, 1903. The machine was in this form when I made demonstrations with it in 1898 before the Chief Examiner, Seely, prior to the grant of my basic patent on Method of and Apparatus for Controlling Mechanism at a Distance.

MNT, CXXVI, K 126

41 A

References

Antonacopoulos, A., & Karatzas, D. (2005). Semantics-based content extraction in typewritten historical documents. In *8th international conference on document analysis and recognition (ICDAR '05)* (pp. 48–53).

Bar-Yosef, I., Mokeichev, A., Kedem, K., Dinstein, I., & Ehrlich, U. (2009). Adaptive shape prior for recognition and variational segmentation of degraded historical characters. *Pattern Recognition, 42*(12), 3348–3354.

Bourbakis, N., Pereira, N., & Mertoguno, S. (1996). Hardware design of a letter-driven OCR and document processing system. *Journal of Network and Computer Applications, 19*(3), 275–294.

Casey, R. G., & Lecolinet, E. (1996). A survey of methods and strategies in character segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 18*(7), 690–706.

Choudhary, A., Rishi, R., & Ahlawat, S. (2013). A New character segmentation approach for off-line cursive handwritten words. In *Procedia computer science, first international conference on information technology and quantitative management: Vol. 17* (pp. 88–95).

Fernández-Caballero, A., López, M. T., & Castillo, J. C. (2012). Display text segmentation after learning best-fitted OCR binarization parameters. *Expert Systems with Applications, 39*(4), 4032–4043.

Garz, A., Fischer, A., Sablatnig, R., & Bunke, H. (2012). Binarization-free text line segmentation for historical documents based on interest point clustering. *2012 10th IAPR international workshop on document analysis systems (DAS)*.

Gatos, B., Louloudis, G., & Stamatopoulos, N. (2014). Segmentation of historical handwritten documents into text zones and text lines. *2014 14th international conference on frontiers in handwriting recognition (ICFHR)*.

González, Á., & Bergasa, L. M. (2013). A text reading algorithm for natural images. *Image and Vision Computing,, 31*(3), 255–274.

Gonzalez, R. C., & Woods, R. E. (2008). *Digital image processing* ((3rd ed.)). New Jersey: Prentice-Hall.

Grafmüller, M., & Beyerer, J. (2013). Performance improvement of character recognition in industrial applications using prior knowledge for more reliable segmentation. *Expert Systems with Applications, 40*(17), 6955–6963.

Gupta, M. R., Jacobson, N. P., & Garcia, E. K. (2007). OCR binarization and image pre-processing for searching historical documents. *Pattern Recognition, 40*(2), 389–397.

Karatzas, D., & Antonacopoulos, A. (2007). Colour text segmentation in web images based on human perception. *Image and Vision Computing, 25*(5), 564–577.

Kavallieratou, E., Stamatatos, E., Fakotakis, N., & Kokkinakis, G. (2000). Handwritten character segmentation using transformation-based learning. In *15th international conference on pattern recognition: Vol. 2* (pp. 634–637).

Kise, K., Sato, A., & Iwata, M. (1998). Segmentation of page images using the area Voronoi diagram. *Computer Vision and Image Understanding, 70*(3), 370–382.

Kovalchuk, A., Wolf, L., & Dershowitz, N. (2014). A simple and fast word spotting method. *2014 14th international conference on frontiers in handwriting recognition (ICFHR)*.

Kumar, J., Kang, L., Doermann, D., & Abd-Almageed, W. (2011). Segmentation of handwritten text lines in presence of touching components. *2011 international conference on document analysis and recognition (ICDAR)*.

Lacerda, E. B., & Mello, C. A. B. (2013). Segmentation of connected handwritten digits using self-organizing maps. *Expert Systems with Applications, 40*(15), 5867–5877.

Lee, H., & Verma, B. (2012). Binary segmentation algorithm for English cursive handwriting recognition. *Pattern Recognition, 45*(4), 1306–1317.

Li, J., Li, M., Pan, J., Chu, S., & Roddick, J. F. (2015). Gabor-based kernel self-optimization Fisher discriminant for optical character segmentation from text-image-mixed document. *Optik - International Journal for Light and Electron Optics, 126*(21), 3119–3124.

Lim, J., Park, J., & Medioni, G. G. (2007). Text segmentation in color images using tensor voting. *Image and Vision Computing, 25*(5), 671–685.

Lu, Y. (1995). Machine printed character segmentation - An overview. *Pattern Recognition, 28*(1), 67–80.

Lu, Y., & Shridhar, M. (1996). Character segmentation in handwritten words - An overview. *Pattern Recognition, 29*(1), 77–96.

Manohar, V., Vitaladevuni, S. N., Cao, H., Prasad, R., & Natarajan, P. (2011). Graph clustering-based ensemble method for handwritten text line segmentation. *2011 international conference on document analysis and recognition (ICDAR)*.

Mao, J., & Mohiuddin, K. M. (1997). Improving OCR performance using character degradation models and boosting algorithm. *Pattern Recognition Letters, 18*(11-13), 1415–1419.

Min-Chul, J., Yong-Chul, S., & Srihari, S. N. (1999). Machine printed character segmentation method using side profiles. In *Proceedings of IEEE SMC '99 conference on systems, man and cybernetics*.

Nikolaou, N., Makridis, M., Gatos, B., Stamatopoulos, N., & Papamarkos, N. (2010). Segmentation of historical machine-printed documents using adaptive run length smoothing and skeleton segmentation paths. *Image and Vision Computing, 28*(4), 590–604.

Nomura, S., Yamanaka, K., Katai, O., Kawakami, H., & Shiose, T. (2005). A novel adaptive morphological approach for degraded character image segmentation. *Pattern Recognition, 38*(11), 1961–1975.

Olszewska, J. I. (2015). Active contour based optical character recognition for automated scene understanding. *Neurocomputing, 161*(5), 65–71.

Park, H. C., Ok, S. Y., Yu, Y. J., & Cho, H. G. (2001). A word extraction algorithm for machine-printed documents using a 3D neighborhood graph model. *International Journal on Document Analysis and Recognition, 4*(2), 115–130.

Phan, T., Shivakumara, P., Su, B., & Tan, C. (2011). A gradient vector flow-based method for video character segmentation. *2011 international conference on document analysis and recognition (ICDAR)*.

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

Pratt, W. K. (2006). *Digital image processing: PIKS scientific inside* (4th ed.). New York: John Wiley & Sons.

Rehman, A., & Saba, T. (2011). Performance analysis of character segmentation approach for cursive script recognition on benchmark database. *Digital Signal Processing, 21*(3), 486–490.

Roy, P. P., Pal, U., Lladós, J., & Delalandre, M. (2012). Multi-oriented touching text character segmentation in graphical documents using dynamic programming. *Pattern Recognition, 45*(5), 1972–1983.

Russ, J. C. (2009). *The image processing handbook* (5th ed.). Florida: CRC Press.

Saba, T., Sulong, G., & Rehman, A. (2010). A survey on methods and strategies on touched characters segmentation. *International Journal of Research and Reviews in Computer Science, 1*, 113–114.

Sedighi, A., & Vafadust, M. (2011). A new and robust method for character segmentation and recognition in license plate images. *Expert Systems with Applications, 38*(11), 13497–13504.

Shan, B. (2010). License plate character segmentation and recognition based on RBF neural network.

Shivakumara, P., Bhowmick, S., Su, B., Tan, C., & Pal, U. (2011). A new gradient based character segmentation method for video text recognition. *2011 international conference on document analysis and recognition (ICDAR)*.

Stamatopoulos, N., Gatos, B., Louloudis, G., Pal, U., & Alaei, A. (2013). ICDAR 2013 handwriting segmentation contest. *2013 12th international conference on document analysis and recognition (ICDAR)*.

Starostenko, O., Cruz-Perez, C., Uceda-Ponga, F., & Alarcon-Aquino, V. (2015). Breaking text-based CAPTCHAs with variable word and character orientation. *Pattern Recognition, 48*(4), 1101–1112.

Surinta, O., Karaaba, M. F., Schomaker, L. R. B., & Wiering, M. A. (2015). Recognition of handwritten characters using local gradient feature descriptors. *Engineering Applications of Artificial Intelligence, 45*, 405–414.

Tan, J., Lai, J., Wang, C., Wang, W., & Zuo, X. (2012). A new handwritten character segmentation method based on nonlinear clustering. *Neurocomputing, 89*, 213–219.

Vamvakas, G., Gatos, B., Stamatopoulos, N., & Perantonis, S. (2008). A complete optical character recognition methodology for historical documents. *IAPR International Workshop on Document Analysis Systems, 1*, 525–532.

Venkateswarlu, N. B., & Boyle, R. D. (1995). New segmentation techniques for document image analysis. *Image and Vision Computing, 13*(7), 573–583.

Wang, X., Huang, L., & Liu, C. (2011). A novel method for embedded text segmentation based on stroke and color. *2011 international conference on document analysis and recognition (ICDAR)*.

Xu, L., Yin, F., & Liu, C. (2010). Touching character splitting of chinese handwriting using contour analysis and DTW. *2010 Chinese conference on pattern recognition (CCPR)*.

Yang, G., Yan, Z., & Zhao, H. (2009). Touching string segmentation using MRF. In *International conference on computational intelligence and security (CIS '09)* (pp. 520–524).

Yoon, Y., Ban, K., Yoon, H., & Kim, J. (2011). Blob extraction based character segmentation method for automatic license plate recognition system. *2011 IEEE international conference on systems, man, and cybernetics (SMC)*.

Younes, M., & Abdellah, Y. (2015). Segmentation of arabic handwritten text to lines. In *Procedia computer science, international conference on advanced wireless information and communication technologies (AWICT 2015): 73* (pp. 115–121).

Zheng, Z., Zhao, J., Guo, H., Yang, L., Yu, X., & Fang, W. (2012). Character segmentation system based on C# design and implementation. *Procedia Engineering, International Workshop on Information and Electronics Engineering, 29*, 4073–4078.

ABBYY FineReader OCR. 2017. <http://finereader.abbyy.com/>.

The OCROpus open source document analysis and OCR system. 2017. <http://code.google.com/p/ocropus>.

Vladan Vučković was born in Niš, Serbia, in 1970. He received the B.E. degree in electrical engineering from the University of Niš, Faculty of Electronic Engineering, Niš, Serbia, in 1994, and the M.Tech. and Ph.D. degrees in electrical engineering and computer science from the, University of Niš, Faculty of Electronic Engineering, Niš, Serbia, in 1997 and 2006, respectively. In 1995, he joined the Computer Department of Faculty of Electronic Engineering, University of Niš, as a Researcher, and in 2003 became an Assistant, in 2007 an Assistant Professor, and became an Associate Professor in 2012. His current research interests include theory of games, artificial intelligence, machine programming and optimization, information security, and 3D modeling, simulation and virtual reality. Dr. Vučković is a member of the International Computer Games Association (ICGA) since 2008. He is a leader of the national project "Advanced methods in 3D modeling and computer simulation of the original patents of Nikola Tesla" since 2009. He is Pupin prize (1995) and Tesla prize (2012) winner.

Boban Arizanović was born in Surdulica, Serbia, in 1991. He received the B.E. degree in electrical engineering from the University of Niš, Faculty of Electronic Engineering, Niš, Serbia, in 2015. Since 2016, for the purpose of academic research and faculty projects working on development of the real-time OCR system. Since 2016, actively researching the predictive modelling approaches for real-world examples. His current research interests include image processing, pattern recognition, artificial intelligence, machine learning, optimization problems, digital signal processing, and information security.