Pattern Recognition Letters 30 (2009) 729–744

Contents lists available at ScienceDirect

Pattern Recognition Letters

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

Morphological preprocessing method to thresholding degraded word images

Shigueo Nomura [a,*], Keiji Yamanaka [b], Takayuki Shiose [a], Hiroshi Kawakami [a], Osamu Katai [a]

[a] Department of Systems Science, Graduate School of Informatics, Kyoto University, Yoshida Honmachi, Sakyo-ku, Kyoto 606-8501, Japan
[b] Faculty of Electrical Engineering, Federal University of Uberlândia, Uberlândia 38400-902, Brazil

ARTICLE INFO

Article history:
Received 21 July 2006
Received in revised form 22 February 2009
Available online 24 March 2009

Communicated by H. Sako

Keywords:
Degraded word image
Critical shadow location
Critical shadow lightening
Mathematical morphology
Morphological preprocessing
Optical character recognition

ABSTRACT

This paper presents a novel preprocessing method based on mathematical morphology techniques to improve the subsequent thresholding quality of raw degraded word images. The raw degraded word images contain undesirable shapes called critical shadows on the background that cause noise in binary images. This noise constitutes obstacles to posterior segmentation of characters. Direct application of a thresholding method produces inadequate binary versions of these degraded word images. Our preprocessing method called Shadow Location and Lightening (SL*L) adaptively, accurately and without manual fine-tuning of parameters locates these critical shadows on grayscale degraded images using morphological operations, and lightens them before applying eventual thresholding process. In this way, enhanced binary images without unpredictable and inappropriate noise can be provided to subsequent segmentation of characters. Then, adequate binary characters can be segmented and extracted as input data to optical character recognition (OCR) applications saving computational effort and increasing recognition rate. The proposed method is experimentally tested with a set of several raw degraded images extracted from real photos acquired by unsophisticated imaging systems. A qualitative analysis of experimental results led to conclusions that the thresholding result quality was significantly improved with the proposed preprocessing method. Also, a quantitative evaluation using a testing data of 1194 degraded word images showed the essentiality and effectiveness of the proposed preprocessing method to increase segmentation and recognition rates of their characters. Furthermore, an advantage of the proposed method is that Otsu's method as a simple and easily implementable global thresholding technique can be sufficient to reducing computational load.

1. Introduction

In word image analysis (Nagy, 2000), thresholding aims at converting the analyzed image to symbolic forms for modification, storage, retrieval, reuse, and transmission (Nagy, 2000). So, thresholding (Alcorn and Hoggar, 1969; Bernsen, 1986) consists of a crucial image-to-image transformation to help to extract binary characters and recognize printed characters (Kamel and Zhao, 1993; Abak et al., 1997). Thresholding is a simple but effective tool to distinguish word areas from background ones (Sankur and Sezgin, 2004) and its performance affects quite critically the degree of success on a subsequent character segmentation and recognition in degraded document processing (Gatos et al., 2006).

In optical character recognition (OCR) applications, it is common practice to use binary characters as input data saving memory space and computational effort (Impedovo et al., 1991) for dealing with real-time processing requirements in industrial-like environments (Sauvola and Pietikäinen, 2000). So, OCR applications, like most commercially available systems, require high quality binary images as input data because their character recognition rate depends on this quality. A raw degraded word image acquired by an unsophisticated imaging system has been the key problem in many OCR applications such as mobile reading system for visually impaired, mobile translation device for foreign visitors in any country, supporting system with embedded imaging device to recognize text information all along the road for car drivers (Thillou and Gosselin, 2005).

Usually, the scanned and digitized image represents only a degraded version of the original scene because unsophisticated imaging systems as well as imaging conditions are imperfect or inappropriate (Flusser and Suk, 1998). Degradation in raw images may occur due to several reasons which range from the acquisition of source type to environmental conditions (Gatos et al., 2006). Problems include the appearance of variable background intensity caused by non-uniform intensity, shadows, smear, smudge and low contrast (Gatos et al., 2006). Raw degraded images with problems of changes in color or size, very low contrast, low quality of focalization, and mainly poor and non-uniform illumination causing shadows are verified by the sample presented in Fig. 1. Because

* Corresponding author. Tel.: +81 75 753 3592; fax: +81 75 753 5042.
E-mail addresses: shigueo@sys.i.kyoto-u.ac.jp (S. Nomura), keiji@ufu.br (K. Yamanaka), shiose@i.kyoto-u.ac.jp (T. Shiose), kawakami@i.kyoto-u.ac.jp (H. Kawakami), katai@i.kyoto-u.ac.jp (O. Katai).

0167-8655/$ - see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.patrec.2009.03.008

Fig. 1. Sample of degraded word images scanned and extracted from poor quality photos.

of this degradation problem, it is quite complicated to obtain the appropriate ubiquitous characters (Wang and Kangas, 2003) using current thresholding methods for raw degraded images of traffic signs, indicators, text documents to feed OCR systems. Thresholding of such degraded images is not a trivial task, even though binary characters with good preservation of edges and corner are vital to OCR performance (Trier and Jain, 1995).

1.1. Related works

There are several methods available to threshold images in order to produce binary versions. Trier and Taxt (1995) experimentally evaluated performances of several such techniques. These methods can be classified into two classes, which are global and local.

Sankur and Sezgin (2004) conducted an exhaustive survey of forty image thresholding methods, both global and local. After a quantitative performance evaluation, the survey has shown that local methods perform better, but they require hard task to implement them and high computational effort to execute their application.

A global thresholding method calculates a single threshold value for the entire image (Trier and Jain, 1995). Sahoo et al. (1988) evaluated more than twenty global thresholding methods. They concluded that Otsu's class separability method provides the most satisfactory thresholding results. Otsu's method (Otsu, 1979) automatically selects the proper threshold of a grayscale image, based on a global histogram of 256 gray-levels. We have adopted this method to threshold our preprocessed grayscale images in this work.

On the other hand, a local adaptive thresholding method computes a threshold for each pixel on the basis of information contained in a neighborhood of the pixel (Trier and Jain, 1995). We adopted the following four adaptive thresholding methods:

Adaptive Lightning Method (ALM) proposed by Nomura et al. (2004): It automatically detects the luminance intensity of grayscale versions by calculating the gray-level average of their backgrounds. And, it adjusts the contrast between the relevant objects and the image background by lightning each sliced region of this image.

Mo and Mathews's method (Mo and Mathews, 1998): Mo and Mathews proposed an adaptive powerful algorithm based on a quadratic filter system model to preprocess the input degraded image before its thresholding. Then, they employed a non-linear quadratic filter (Mo and Mathews, 1998) to develop the algorithm for achieving edge enhancement and noise reduction.

Yanowitz and Bruckstein's method (Yanowitz and Bruckstein, 1989): It calculates a threshold surface that varies over different image regions so as to fit the spatially changing background and lighting conditions.

Niblack's method (Niblack, 1986): The idea of this method is to vary the threshold across the image, based on the local mean and local standard deviation.

These methods were selected because they have been frequently referred to in the literature and appeared to be promising to solve thresholding problems in degraded images. They work well for a slowly changing background of degraded images. However, they have not provided the expected enhancement results on binary versions of the degraded images that include the critical shadows in this work. Inappropriate binary images like the one in Fig. 3 can seriously hinder the subsequent segmentation (Lu, 1995; Nomura et al., 2005) and extraction of characters from words, and cause low performance on eventual OCR system.

1.2. Rationale and motivation for the work

A generalized preprocessing method for dealing with any degradation problem in digital word images is desirable. However, due to current hardware limitations, we have to develop methods for solving key degradation problems in order to gain speed and quality in real-time applications. Also, it is difficult to obtain satisfactory thresholding results from various kinds of document images (Tsai and Lee, 2002). To maintain OCR accuracy with decreasing quality of page image composition, production, and digitization, it is essential to develop methods for document-specific OCR systems (Xu and Navy, 1999). Existing thresholding methods (Sankur and Sezgin, 2004) have ignored the heterogeneity and diversity of the background for real-time images. So, these methods are not able to partially remove noise due to degradation in the background. In this work, the preprocessing constitutes the theory and its application of recovering the symbol structure (Nagy, 2000) of degraded digital word images extracted from poor quality photos taken in real-time.

This work devoted attention to the degradation due to uneven illumination that yields "ghost" objects called critical shadows as key problem. Technically, these critical shadows on the background and the relevant objects on the foreground present very close or quite equal pixel values that cause binary noise after applying a conventional thresholding process. Fig. 2 shows an example of degraded grayscale image with the undesirable critical shadows. For instance, the application of a global thresholding method (Otsu, 1979) may result in noisy and distorted binary image verified in Fig. 3 because of these critical shadows.

The critical shadows are classified according to corresponding noise place in the binary image. In Fig. 2, we have three types of critical shadows to be preprocessed:

Type 1: Isolated shadows are located on the middle area of a grayscale image.
Type 2: Isolated shadows are located around the border of a grayscale image.

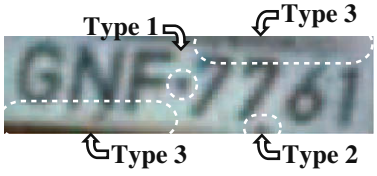

Fig. 2. Example of a digitized grayscale image with critical shadows (types 1, 2, and 3) due to a scenario under poor illumination conditions.

Fig. 3. Noisy binary image with a conventional thresholding method.

Type 3: Complex type whose shadows are located around the border, and they cause noise connected to relevant objects (characters).

The strategy of this work is the preprocessing of the critical shadows on grayscale degraded images by locating and lightening these image degradation problems before eventual thresholding process. This preprocessing method is named Shadow Location and Lightening (SL*L).

The goal of this method is to produce enhanced binary versions of degraded grayscale word images that retain most of the useful information while reducing the effect of noise. Our work's motivation is to solve this non-linear thresholding problem by preprocessing the critical shadows based on powerful morphological operations. In this way, binary images without unpredictable and inappropriate noise can be provided to segmentation process of their objects (characters). Subsequently, the process can segment and extract adequate binary characters in a high segmentation rate with low computational load. Also, these binary characters can be used as input data to commercially available optical character recognition (OCR) applications saving memory as well as computational effort and increasing recognition rate.

In the following section we present the reasons for adopting mathematical morphology techniques and the algorithms based on these techniques. In Section 3, we describe the SL*L method with its parts. An experimental system using the SL*L method is presented and described in Section 4. In Section 5, we evaluate the experimental results of the SL*L method application with those of Otsu's method (Otsu, 1979). Section 7 states the conclusion of this work.

2. Theoretical foundation for the SL*L method

We present some reasons for adopting mathematical morphology techniques (Serra, 1982; Heijmans, 1992) in order to solve the thresholding non-linear problem in this work:

- Mathematical morphology is a powerful methodology for processing and analyzing the shape (form of objects) in images (Talbot and Beare, 2002), and it is based on set-theoretical, geometrical and topological concepts (Serra, 1982). The methodology has achieved a status as a powerful method for image processing, and it is useful for analyzing the geometrical structure in an image (Heijmans, 1992).
- It is a versatile and powerful tool for extracting useful image components in the representation and description of regions, such as boundary, size, area, shape, or location in binary images (Gao et al., 2001).
- The language of mathematical morphology (Serra, 1982; Gonzalez and Woods, 1993) is the set theory that represents objects in an image, thus it can be used as a tool to locate, segment, and extract the objects from the images. The images can be subsequently processed to perform enhancement, edge detection, thinning, thickening, or segmentation of their objects.

Now, we present two algorithms developed by using mathematical morphology basic operations for dilation, erosion, and "hit-or-miss" transform of objects. The first is the thickening algorithm to thicken the objects (characters and noise) in a noisy binary image. The second is the pruning algorithm to appropriately "clean-up" the thickened image with parasitic branches to obtain an "image without spurs."

2.1. Thickening algorithm

Thickening (Serra, 1982) is a morphological operation that works to grow selected regions of foreground pixels in binary images. In this work, the thickening algorithm is used to determine boundaries of these regions that cover characters and noise in binary images. Using Eq. (1), the thickening (Soille, 1999) of a binary image X by a structuring element B is denoted by $X \otimes B$ and defined as the union between X and the hit-or-miss transform of X by B $(X * B)$:

$$X \otimes B = X \cup (X * B). \tag{1}$$

2.2. Pruning algorithm

The pruning algorithm is an essential complement to the thickening algorithm. It is used to clean up parasitic components from 8-connected objects in this work. The solution is based on suppressing a parasitic branch by successively eliminating its end point (Gonzalez and Woods, 1993).

The pruning operation is denoted by $\ominus$, and its algorithm is implemented by using the following equation as its mathematical foundation (Gonzalez and Woods, 1993):

$$S \ominus B = S \cup \left(\bigcup_{i=1}^{n} (S * B^i) \oplus H \right) \cap X, \tag{2}$$

where S is a skeleton of the binary image X obtained by the thickening algorithm; B is a sequence of structuring elements to thicken and to restore the image to its original form; H is a structuring element to dilation of end points conditioned on X; n is the number of structuring elements; $*$ denotes hit-or-miss transform; $\oplus$ denotes dilation.

3. Proposed SL*L method

Our proposed SL*L method is basically composed of four processes according to the flow chart of its architecture shown in Fig. 4:

Process 1: Obtaining an intermediate binary image.
Process 2: Obtaining a thickened image.
Process 3: Obtaining an image without spurs (IWS).
Process 4: Lightening the critical shadows of a degraded grayscale image.

Processes 1–3 are to adaptively delimitate the shadowy regions for the critical shadows in grayscale images by applying mathematical morphology techniques to corresponding intermediate binary images. Process 4 is to lighten the critical shadows delimitated by the IWS.

To describe each process of our method, we adopted the following conventions to represent the images:

- A matrix of real numbers is used to represent digital images.
- A digital image function $f(x,y)$ is represented by an array of M rows and N columns where the values of the coordinates (x,y) are discrete quantities.

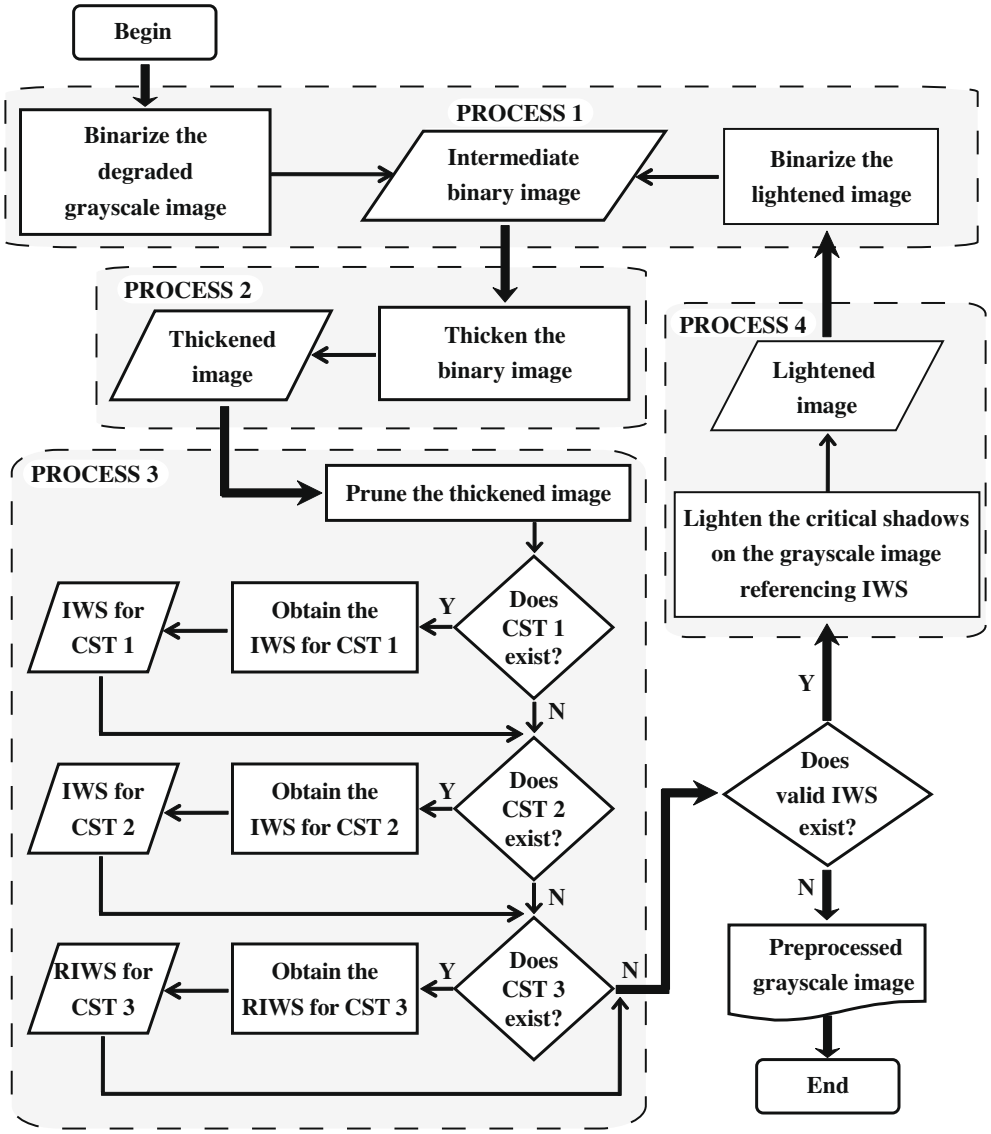

Fig. 4. Flow chart of the SL*L method architecture, where CST, IWS, and RIWS respectively mean critical shadow type, image without spurs, and repaired image without spurs.

- Let $\mathbb{Z}$ denote the set of integer values. The discrete coordinates are integers from the Cartesian product $\mathbb{Z} \times \mathbb{Z}$ (also written as $\mathbb{Z}^2$) where x coordinate ranges from 1 to M, and y coordinate from 1 to N, in integer increments.

3.1. Process 1: Obtaining intermediate binary images

This process consists of obtaining an intermediate binary image. Basically, we threshold a grayscale version of a raw degraded word image or previous shadow lightened image. A global thresholding process to obtain a binary image IX can be described as follows:

$$IX = \left\{ f(x,y) \in \{0,1\} \middle| \begin{matrix} f(x,y) = 1 \text{ if } g(x,y) \leqslant Tr \\ f(x,y) = 0 \text{ otherwise} \end{matrix} \right\}, \tag{3}$$

where $g(x,y) \in IG$; IG is a grayscale image; Tr is a thresholding value.

Fig. 5 presents a typical degraded grayscale image with the critical shadows to be lightened and Fig. 6 presents the binary version resulted from this process. The resulting binary version is to work as input data to the next process.

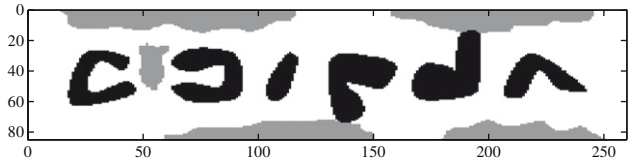

Fig. 5. Typical grayscale image degraded by critical shadows.

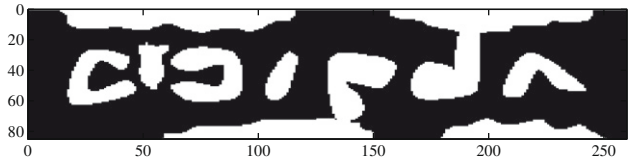

Fig. 6. Binary version of the grayscale image presented in Fig. 5.

3.2. Process 2: Obtaining thickened images

In this process, the morphological thickening operation defined by Eq. (1) is applied to the binary image IX. A thickened image IY is obtained as follows:

$$IY = IX \otimes SA, \tag{4}$$

where SA is a set of structuring elements to thicken the input image IX.

This process is crucial in making the method able to locate critical shadows on degraded grayscale images. The resulting thickened image (skeleton) provides reference data to make a rough detection of boundaries for likely shadowy regions on the degraded grayscale image. Fig. 7 shows the skeleton of the binary image presented in Fig. 6.

3.3. Process 3: Obtaining images without spurs (IWS)

The pruning operation presented in Section 2.2 is applied to the thickened image IY to obtain the pruned image IWS as follows:

$$IWS = IY \ominus SB, \tag{5}$$

where SB is a set of structuring elements to prune the thickened image IY.

The structuring elements shown in Fig. 8 have been created to detect an interconnecting pixel. This pixel must belong to the 8-connected object, that is, it must not be an ending point.

The definition of a relevant pixel is used to eliminate a parasitic branch. If a pixel satisfies one of the two following conditions, then it is considered relevant and must not be removed:

(1) The pixel belongs to the border of the image.
(2) It is an interconnecting pixel.

In addition, all eight neighbors of each non-relevant pixel (non-zero) are analyzed. It is then possible to maintain the relevant pixels so that only a parasitic branch is eliminated. Table 1 presents the pruning algorithm to obtain an IWS as a pruned version for the thickened image. Fig. 9 shows the sketch of the pruned image after applying this algorithm to the thickened version of Fig. 7.

In particular, there are situations in this process where a branch must not be suppressed. Then, the pruning algorithm must be adapted to provide the appropriate IWS to deal with one of three types of critical shadows mentioned in Section 1.2 and illustrated by the sketch in Fig. 10:

- Critical Shadow Type 1 abbreviated as CST 1
- Critical Shadow Type 2 abbreviated as CST 2
- Critical Shadow Type 3 abbreviated as CST 3

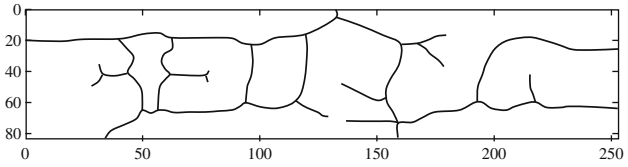

Fig. 7. Sketch of the thickened version for the binary image presented in Fig. 6.

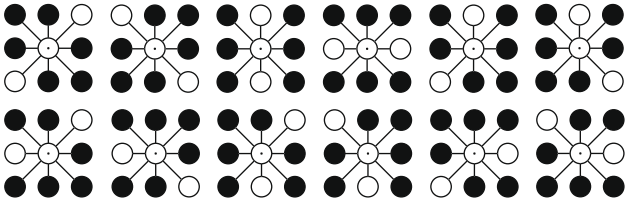

Fig. 8. Structuring elements created to detect interconnecting pixels in an "image without spurs". Where "○" represents the pixel of the 8-connected white object; "•" represents the pixel of the black background or 8-connected white object; the origin of each structuring element is at its center.

Table 1
Algorithm for obtaining an IWS as the pruned image.

1: Input the thickened image IY resulted from Eq. (4)
2: Input the structuring elements presented in Fig. 8 to detect the interconnecting pixels
3: For each non-zero pixel from IY
  a: if the pixel does not belong to the border of IY and it is not an interconnecting pixel according to the structuring elements, then
  b: remove this pixel
4: Output the pruned image IWS

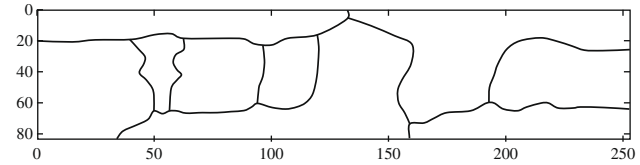

Fig. 9. Sketch of the pruned version for the thickened image presented in Fig. 7.

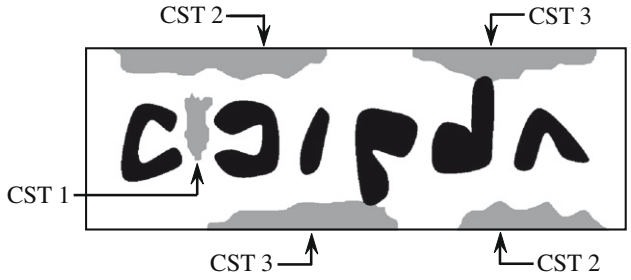

Fig. 10. Sketch of the grayscale image presented in Fig. 5 with three types of critical shadows.

3.3.1. Obtaining the IWS for CST 1

In this type of critical shadow, the IWS must have an "enclosing boundary" surrounding a shadowy region located on the middle area (critical zone) of the sketch shown in Fig. 11. This critical zone delimitates the following shadowy region μ to locate CST 1:

$$\mu = \left\{ (x,y) \in \mathbb{Z}^2 \middle| \begin{matrix} x_{min} < x < x_{max}, y_{min} < y < y_{max} \\ f(x,y) \in IWS \end{matrix} \right\}, \tag{6}$$

where IWS represents the pruned image; x_{min} and y_{min} are the minimum coordinates delimitating the critical zone; x_{max} and y_{max} are the maximum coordinates delimitating the critical zone.

In this case, the pruning algorithm presented in Table 1 should be adapted to provide the IWS using only the following condition: the analyzed pixel is an interconnecting pixel. Executing the adapted pruning algorithm for the critical zone delimitated by Eq. (6), we can obtain the sketch presented in Fig. 12. It corresponds to the IWS for locating CST 1 sketched in Fig. 10.

3.3.2. Obtaining the IWS for CST 2

When the critical shadows are disconnected from relevant objects (for example, characters) and located near to the upper or lower border of a grayscale image, we have the situation shown in Fig. 13. In this case, the IWS must provide appropriate boundaries to enclose eventual critical shadows on the shadowy region Ⓐ (upper) or Ⓑ (lower) of the grayscale image.The critical zones delimitate two regions as follows:

$$\mu_A = \{(x,y) \in \mathbb{Z}^2 \mid 0 < x < x_{at},\ 0 < y \leqslant N,\ f(x,y) \in IWS\} \tag{7}$$

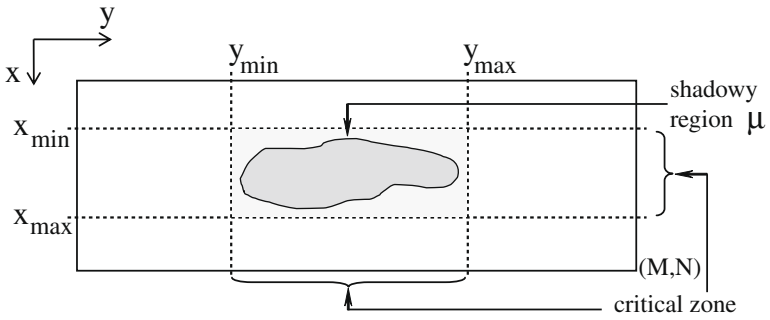

Fig. 11. Sketch of the IWS for locating a CST 1 on the shadowy region μ delimitated by Eq. (6).

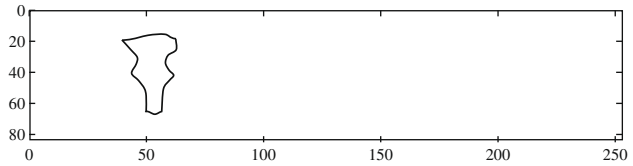

Fig. 12. Sketch of the IWS for locating CST 1 of the image in Fig. 10.

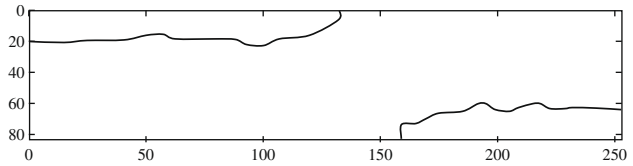

Fig. 14. Sketch of the IWS for locating CST 2 sketched in Fig. 10.

for shadowy region Ⓐ and

$$\mu_B = \{(x,y) \in \mathbb{Z}^2 \mid x_{pt} < x \leqslant M,\ 0 < y \leqslant N,\ f(x,y) \in IWS\} \quad (8)$$

for shadowy region Ⓑ, where *IWS* represents the pruned image; x_{at} is the maximum coordinate delimitating the upper critical zone; x_{pt} is the minimum coordinate delimitating the lower critical zone; M is the total quantity of rows of the IWS; N is the total quantity of columns of the IWS.

The pruning algorithm is applied to the variation of coordinates defined by Eqs. (7) and (8). The resulting sketch of the IWS to locate CST 2 of the image in Fig. 10 is presented in Fig. 14.

3.3.3. Obtaining the IWS for CST 3

This is a complex type whose critical shadows are connected to one or more relevant objects (for example, characters). Because of this connection, the pruned image determined by process 3 in the flow chart of Fig. 4 presents an interrupted or non-enclosing boundary for the critical shadow. This interruption problem causes an unideal IWS with untrue shadowy regions that cannot adequately locate the critical shadows. These untrue regions tend to trespass the critical x coordinates (x_{at} and x_{pt}). They invade areas of relevant objects creating an overlapping region as illustrated by the sketch of Fig. 15. Our idea comes to repair the interruption problem in two steps by providing the repaired IWS to locate CST 3 as follows:

Step 1: Detecting boundaries of upper untrue shadowy regions Ⓐ① and Ⓐ② as well as lower shadowy regions Ⓑ① and Ⓑ② shown in Fig. 15.

Step 2: Determining line segments delimitated by the minimum points (PA_1 and PA_2) as well as maximum points (PB_1 and PB_2) verified in Fig. 16.

3.3.3.1. Step 1: Boundary detection for untrue shadowy regions. Basically, these boundaries are detected by analyzing each object pixel of the unideal IWS in Fig. 15 under the following conditions:

C-1 All the pixels must belong to the critical zone (upper or lower) of the unideal IWS. Mathematically, the coordinates (x,y) of these pixels must belong to the upper shadowy region given by Eq. (7) or lower shadowy region given by Eq. (8).

C-2 The boundary for the second shadowy region (Ⓐ② or Ⓑ②) must be determined after detecting the boundary for the first shadowy region (Ⓐ① or Ⓑ①).

C-3 The detected boundary must have at least three consecutive pixels located at different y coordinates of the unideal IWS.

C-4 When a local minimum or maximum as the extreme point belongs to the border of the unideal IWS in Fig. 15, this extremum of the detected boundary assumes one of the following equalities:

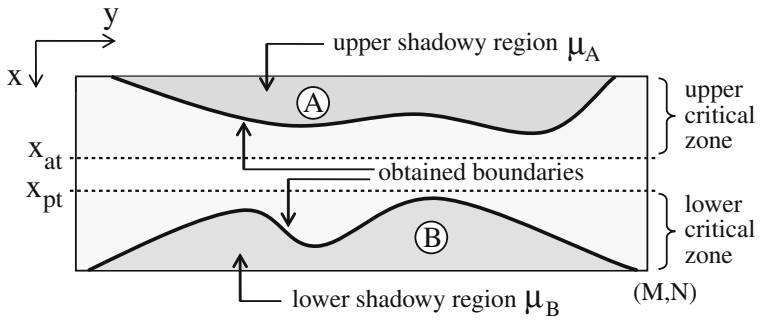

Fig. 13. Sketch of the IWS for locating a CST 2 in the upper shadowy region μ_A delimitated by the coordinates from Eq. (7) or lower shadowy region μ_B delimitated by the coordinates from Eq. (8).

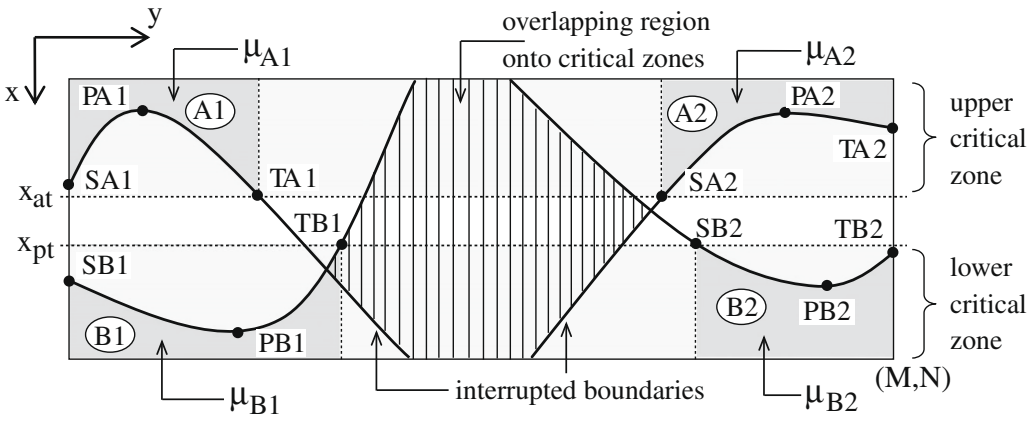

Fig. 15. Sketch of the unideal IWS (UIWS) with upper untrue shadowy regions Ⓐ1 and Ⓐ2 as well as lower untrue shadowy regions Ⓑ1 and Ⓑ2, where μ_{A_1} is from Eq. (17) and delimitates the upper shadowy region above the boundary defined by points SA_1, PA_1, and TA_1; μ_{A_2} is from Eq. (19) and delimitates the upper shadowy region above the boundary defined by points SA_2, PA_2, and TA_2; μ_{B_1} is from Eq. (18) and delimitates the lower shadowy region below the boundary defined by points SB_1, PB_1, and TB_1; μ_{B_2} is from Eq. (20) and delimitates the lower shadowy region below the boundary defined by points SB_2, PB_2, and TB_2.

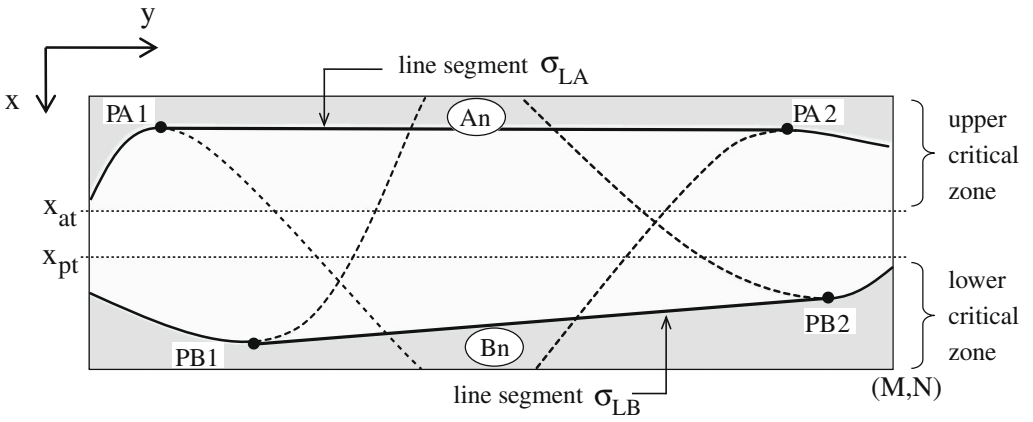

Fig. 16. Sketch of the repaired IWS (RIWS) with the new upper shadowy region Ⓐn and lower shadowy region Ⓑn for locating CST 3, where: σ_{LA} is the line segment delimitated by the local minimum points (PA_1 and PA_2), and calculated by Eq. (25); σ_{LB} is the line segment delimitated by the local maximum points (PB_1 and PB_2).

- $SA_1 \equiv PA_1$
- $PA_2 \equiv TA_2$
- $SB_1 \equiv PB_1$
- $PB_2 \equiv TB_2$

where S denotes a starting point; P denotes a point of local minimum or maximum; T denotes an ending point; A represents an upper region; B represents a lower region; 1 is the index to indicate the first untrue region of the pair; 2 is the index to indicate the second untrue region of the pair.

C-5 The first y coordinate corresponding to the first shadowy region (Ⓐ1 or Ⓑ1) boundary may be the first y coordinate of the unideal IWS, or the corresponding y coordinate of PA_1 or PB_1 which belongs to the border of the unideal IWS. Mathematically, the first y coordinate is given by

$$y = \begin{cases} 1 \text{ if } SA_1 \neq PA_1, \\ y_{PA_1} \text{ otherwise} \end{cases} \tag{9}$$

for upper shadowy region and

$$y = \begin{cases} 1 \text{ if } SB_1 \neq PB_1, \\ y_{PB_1} \text{ otherwise} \end{cases} \tag{10}$$

for lower shadowy region, where SA_1, SB_1, PA_1, and PB_1 are described in the condition C-4; y_{PA_1} is the y coordinate for PA_1; y_{PB_1} is the y coordinate for PB_1.

C-6 The last pixel corresponding to the first shadowy region boundary is defined by TA_1 or TB_1 points presented in Fig. 15. TA_1 corresponds to the point whose x coordinate is x_{at} at the boundary of the first upper shadowy region Ⓐ1. TB_1 corresponds to the point whose x coordinate is x_{pt} at the boundary of the first lower shadowy region Ⓐ2. Mathematically, the last pixel is given by the point with coordinates

$$(x, y) = (x_{at}, y_{TA_1}) \tag{11}$$

for upper shadowy region and

$$(x, y) = (x_{pt}, y_{TB_1}) \tag{12}$$

for lower shadowy region, where $y_{TA_1} > 1$ is the y coordinate for TA_1 representing the ending point for the first upper untrue region; $y_{TB_1} > 1$ is the y coordinate for TB_1 representing the ending point for the first lower untrue region.

C-7 The first pixel corresponding to the second shadowy region (Ⓐ2 or Ⓑ2) boundary of the sketch in Fig. 15 is defined by SA_2 or SB_2 points similarly to the condition C-6. Mathematically, the first pixel is given by the point with coordinates

$$(x, y) = (x_{at}, y_{SA_2}) \tag{13}$$

for upper shadowy region and

$$(x,y) = (x_{pt}, y_{SB_2}) \tag{14}$$

for lower shadowy region, where $y_{SA_2} < N$ is the y coordinate for SA_2 representing the starting point for the second upper untrue region; $y_{SB_2} < N$ is the y coordinate for SB_2 representing the starting point for the second lower untrue region.

C-8 The last y coordinate corresponding to the second shadowy region boundary may be the last y coordinate of the unideal IWS, or the corresponding y coordinate of PA_2 or PB_2 which belongs to the border of the unideal IWS. Mathematically, the last y coordinate is given by

$$y = \begin{cases} N \text{ if } PA_2 \neq TA_2, \\ y_{PA_2} \text{ otherwise} \end{cases} \tag{15}$$

for upper shadowy region and

$$y = \begin{cases} N \text{ if } PB_2 \neq TB_2 \\ y_{PB_2} \text{ otherwise} \end{cases} \tag{16}$$

for lower shadowy region, where PA_2, PB_2, TA_2, and TB_2 are described in the condition C-4; y_{PA_2} is the y coordinate for PA_2; y_{PB_2} is the y coordinate for PB_2.

In summary, the coordinates to detect the boundaries of the untrue shadowy regions given by the unideal IWS can be mathematically determined as follows:

(1) Coordinates of the first untrue shadowy region:

$$\mu_{A_1} = \left\{ (x,y) \in \mathbb{Z}^2 \middle| \begin{matrix} 0 < x \leqslant x_{at},\ 0 < y \leqslant y_{TA_1} \\ f(x,y) \in UIWS \end{matrix} \right\} \tag{17}$$

for upper region and

$$\mu_{B_1} = \left\{ (x,y) \in \mathbb{Z}^2 \middle| \begin{matrix} x_{pt} \leqslant x \leqslant M,\ 0 < y \leqslant y_{TB_1} \\ f(x,y) \in UIWS \end{matrix} \right\} \tag{18}$$

for lower region, where *UIWS* represents the unideal IWS (pruned image); x_{at} is the maximum x coordinate delimitating the upper critical zone; x_{pt} is the minimum x coordinate delimitating the lower critical zone; M is the total quantity of rows of the *UIWS*; y_{TA_1} is the y coordinate of the point defined by Eq. (11); y_{TB_1} is the y coordinate of the point defined by Eq. (12).

(2) Coordinates of the second untrue shadowy region:

$$\mu_{A_2} = \left\{ (x,y) \in \mathbb{Z}^2 \middle| \begin{matrix} 0 < x \leqslant x_at,\ y_{SA_2} \leqslant y \leqslant N \\ f(x,y) \in UIWS \end{matrix} \right\} \tag{19}$$

for upper region and

$$\mu_{B_2} = \left\{ (x,y) \in \mathbb{Z}^2 \middle| \begin{matrix} x_pt \leqslant x \leqslant M,\ y_{SB_2} \leqslant y \leqslant N \\ f(x,y) \in UIWS \end{matrix} \right\} \tag{20}$$

for lower region, where *UIWS* represents the unideal IWS (pruned image); x_{at} is the maximum coordinate delimitating the upper critical zone; x_{pt} is the minimum coordinate delimitating the lower critical zone; M is the total quantity of rows of the *UIWS*; N is the total quantity of columns of the *UIWS*; y_{SA_2} is the y coordinate of the point defined by Eq. (13); y_{SB_2} is the y coordinate of the point defined by Eq. (14).

Table 2 presents the algorithm to provide the unideal IWS with the boundaries of untrue shadowy regions due to the interruption problem from CST 3. Fig. 17 presents the sketch of the unideal IWS with the interrupted boundaries of untrue shadowy regions due to CST 3 illustrated by the sketch of Fig. 10.

3.3.3.2. Step 2: Line segment finding. In this step, a line segment is determined between two extreme points belonging to the interrupted boundaries of the untrue shadowy regions detected above. These extreme points are defined as follows:

- Local minimum points (PA_1 and PA_2) at which the x coordinate assumes its minimum value for the boundaries of upper untrue shadowy regions Ⓐ1 and Ⓐ2 of Fig. 15.
- Local maximum points (PB_1 and PB_2) at which the x coordinate assumes its maximum value for the boundaries of lower untrue shadowy regions Ⓑ1 and Ⓑ2 of Fig. 15.

Mathematically, the determination of the line segment between two extreme points is as follows:

(1) For coordinates of local minimum points PA_1 and PA_2 at the boundaries of upper regions Ⓐ1 and Ⓐ2 on the sketch of Fig. 15, the respective minimum value of the x coordinate is determined as follows:
   - $min_1(x)$ – starting from the point SA_1 and following the boundary of upper shadowy region Ⓐ1 toward the point TA_1

$$\mu_{PA_1} = \{(x,y) \in \mu_{A_1} \mid (x,y) = (x_{PA_1}, y_{PA_1})\}, \tag{21}$$

   where μ_{A_1} is the first upper untrue shadowy region delimitated by Eq. (17); $x_{PA_1} = min_1(x)$; y_{PA_1} is the corresponding y coordinate for the point PA_1 at the boundary of Ⓐ1.
   - $min_2(x)$ – starting from the point SA_2 and following the boundary of upper shadowy region Ⓐ2 toward the point TA_2

$$\mu_{PA_2} = \{(x,y) \in \mu_{A_2} \mid (x,y) = (x_{PA_2}, y_{PA_2})\}, \tag{22}$$

   where μ_{A_2} is the second upper untrue shadowy region delimitated by Eq. (19); $x_{PA_2} = min_2(x)$; y_{PA_2} is the corresponding y coordinate for the point PA_2 at the boundary of Ⓐ2.

(2) For coordinates of local maximum points PB_1 and PB_2 at the boundaries of lower regions Ⓑ1 and Ⓑ2 on the sketch of Fig. 15, the respective maximum value of the x coordinate is determined as follows:
   - $max_1(x)$ – starting from the point SB_1 and following the boundary of lower shadowy region Ⓑ1 toward the point TB_1

Table 2
Algorithm for detecting the boundaries of untrue shadowy regions.

1: Input the pruned image (IWS)
2: Calculate the coordinates for the boundary of the first untrue shadowy region according to the conditions (C-1, C-3, C-4, C-5, C-6) presented in Section 3.3.3.1 and using Eqs. (17) and (18)
3: Calculate the coordinates for the boundary of the second untrue shadowy region according to the conditions (C-1, C-2, C-3, C-4, C-7, C-8) presented in Section 3.3.3.1 and using Eqs. (19) and (20)
4: Output the unideal image without spurs (UIWS) with the boundaries of untrue shadowy regions

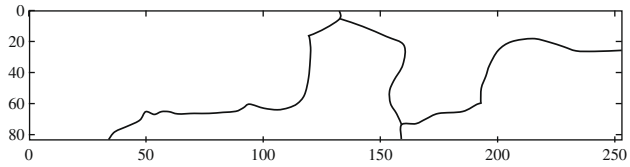

Fig. 17. Sketch of the UIWS with the boundaries of untrue shadowy regions due to CST 3 sketched in Fig. 10.

$$\mu_{PB_1} = \{(x,y) \in \mu_{B_1} \mid (x,y) = (x_{PB_1}, y_{PB_1})\}, \quad (23)$$

where μ_{B_1} is the first lower untrue shadowy region delimitated by Eq. (18); $x_{PB_1} = max_1(x)$; y_{PB_1} is the corresponding y coordinate for the point PB_1 at the boundary of ⑨①.

- $max_2(x)$ – starting from the point SB_2 and following the boundary of lower shadowy region ⑩ toward the point TB_2

$$\mu_{PB_2} = \{(x,y) \in \mu_{B_2} \mid (x,y) = (x_{PB_2}, y_{PB_2})\}, \quad (24)$$

where μ_{B_2} is the second lower untrue shadowy region delimitated by Eq. (20); $x_{PB_2} = max_2(x)$; y_{PB_2} is the corresponding y coordinate for the point PB_2 at the boundary of ⑩②.

(3) Coordinates of the line segment:

$$\sigma_L = \left\{ (x,y) \in \mathbb{Z}^2 \middle| \begin{array}{l} x = m * (y - y_1) + x_1 \\ y_1 \leqslant y \leqslant y_2 \\ f(x,y) \in UIWS \end{array} \right\}, \quad (25)$$

where $UIWS$ is the unideal IWS; $m = (x_2 - x_1)/(y_2 - y_1)$; $x_1 = x_{PA_1}$ ($\in \mu_{PA_1}$ from Eq. (21)) or x_{PB_1} ($\in \mu_{PB_1}$ from Eq. (23)); $x_2 = x_{PA_2}$ ($\in \mu_{PA_2}$ from Eq. (22)) or x_{PB_2} ($\in \mu_{PB_2}$ from Eq. (24)); $y_1 = y_{PA_1}$ ($\in \mu_{PA_1}$ from Eq. (21)) or y_{PB_1} ($\in \mu_{PB_1}$ from Eq. (23)); $y_2 = y_{PA_2}$ ($\in \mu_{PA_2}$ from Eq. (22)) or y_{PB_2} ($\in \mu_{PB_2}$ from Eq. (24)).

(4) Pruned image repaired by the line segment (repaired IWS abbreviated as RIWS):

$$RIWS = UIWS \bigcup \left[g(x,y) = \begin{cases} 1 \text{ if } (x,y) \in \sigma_L \\ 0 \text{ otherwise} \end{cases} \right], \quad (26)$$

where $UIWS$ is the unideal IWS; $g(x,y)$ is a pixel from the determined line segment; σ_L is from Eq. (25).

Table 3 presents the algorithm to determine the line segments delimitated by two extreme points at the interrupted boundaries of the UIWS.

After applying this algorithm, the resulting line segments must delimitate new true shadowy regions ⑧ and ⑨ of the repaired IWS (RIWS) as shown in Fig. 16. Fig. 18 presents the sketch of the RIWS with the line segments correcting the interrupted boundaries of the UIWS presented in Fig. 17.

Table 3
Algorithm for repairing the interrupted boundaries of the UIWS.

1: Input the unideal image without spurs (UIWS) obtained from the algorithm presented in Table 2
2: Determine the coordinates of local minimum points (PA_1 and PA_2 presented in Fig. 15) using Eqs. (21) and (22)
3: If the local minimum points exist then calculate the coordinates of the line segment delimitated by these two points using Eq. (25)
4: Determine the coordinates of local maximum points (PB_1 and PB_2 presented in Fig. 15 using Eqs. (23) and (24)
5: If the local maximum points exist then calculate the coordinates of the line segment delimitated by these two points using Eq. (25)
6: If the coordinates of a line segment exist then replace the pixels corresponding to these coordinates from the UIWS with the pixels from the line segment using Eq. (26)
7: Output the RIWS as the repaired IWS

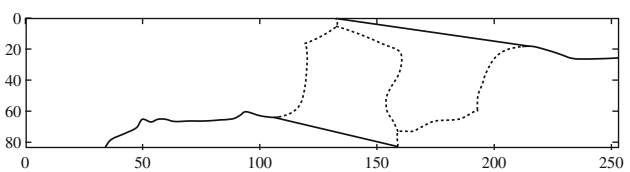

Fig. 18. Sketch of the repaired IWS (RIWS) for locating CST 3 of the image in Fig. 10.

3.4. Process 4: Lightening critical shadows

This process consists of lightening the critical shadowy regions for CST 1, CST 2, and CST 3 located by the previous process. The pixel intensity of the shadow to be lightened is automatically calculated (Nomura et al., 2004) as the gray-level average ($GLav$) for the background of a grayscale image IG:

$$GLav = \frac{\sum h(x,y)}{N_B}, \quad (27)$$

where $h(x,y) = \{f(x,y) \in IG \mid f(x,y) < threshold\}$; N_B is the total quantity of background pixels.

The shadow lightening algorithm uses the boundaries of shadowy regions represented by the IWS or RIWS as input data. The algorithm analyzes each pixel from the IWS or RIWS to detect the starting, the ending, and all the coordinates of the boundary that delimitates a shadowy region with the critical shadows. The detected boundary must provide appropriate data to lighten the eventual shadowy region. In this lightening process, a pixel within the shadowy region is replaced with the above calculated gray-level average ($GLav$) intensity. Then, the new lightened pixels replace the pixels from the critical shadows providing the preprocessed grayscale images with the lightened shadows before an eventual thresholding.

Table 4 presents the algorithm to lighten the shadowy regions with the critical shadows on degraded grayscale images.

4. Experimental system

Fig. 19 shows the steps of an experimental system employing the SL*L method for morphological preprocessing of raw degraded

Table 4
Algorithm for lightening the critical shadows.

1: Input the degraded grayscale image
2: Get the binary version using Eq. (3)
3: Get the thickened version using Eq. (4)
4: Calculate the gray-level average ($GLav$) for the grayscale image using Eq. (27)
5: Get the IWS for CST 1 using the shadowy region given by Eq. (6)
6: Get the IWS for CST 2 considering the areas delimitated by Eq. (7) for upper shadowy region and delimitated by Eq. (8) for lower shadowy region
7: Get the RIWS for CST 3 from the algorithm presented in Table 3
8: For each pruned image (IWS or RIWS) to locate each type of critical shadow (CST 1, CST 2, or CST 3)
  a: delimitate the shadowy region on the grayscale image
  b: replace the pixel from this shadowy region with the above calculated $GLav$
9: Output the preprocessed grayscale image with the lightened critical shadows

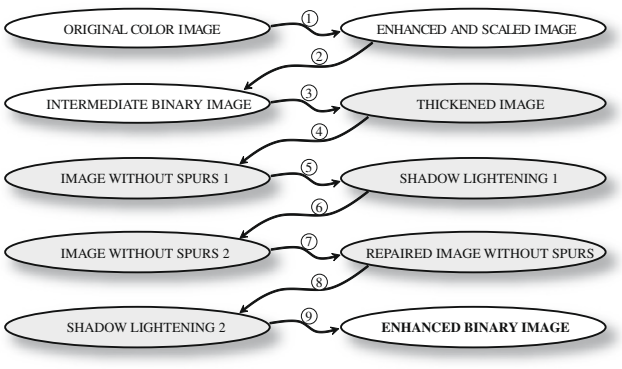

Fig. 19. Main steps of the experimental system based on the SL*L method.

grayscale images extracted from real photos. The steps of the experimental system proceed as follows:

4.1. Step 1: Obtaining pre-enhanced grayscale images

In this step, the following functions are executed:

- Converting an original color image into a grayscale version.
- Standardizing the size of each grayscale image.
- Pre-enhancing the scaled grayscale image.

The details of these functions are described in our previous work (Nomura et al., 2004). A resulting image of this step is shown in Fig. 20.

4.2. Step 2: Obtaining an intermediate binary image

The binary image in Fig. 21 is the result of global thresholding on the grayscale image in Fig. 20 with Otsu's method. This binary image is to detect the boundaries of shadowy regions for CST 1, CST 2, and CST 3 of the degraded grayscale image.

4.3. Step 3: Obtaining a thickened image

The thickened image in Fig. 22 is obtained by applying the morphological thickening algorithm to the image in Fig. 21. This thickened image presents parasitic components which are not useful for delimitating shadowy regions of the grayscale version.

4.4. Step 4: Obtaining the IWS for locating CST 1

The image in Fig. 20 shows a small CST 1 that corresponds to a separation dash between letters and digits in a word photo. Eqs. (28) and (29) define the critical zone where the shadowy region should be located on the experimental images:

$$\frac{1}{3}ncol < c < \frac{2}{3}ncol, \tag{28}$$

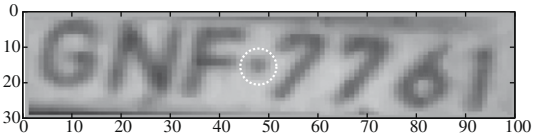

Fig. 20. Grayscale version after standardizing the size and pre-enhancing the original image in Fig. 2. An isolated critical shadow is located on the middle area.

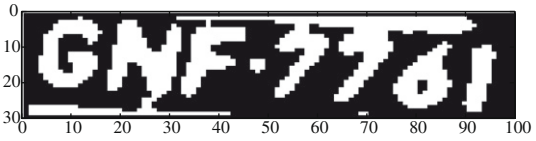

Fig. 21. Binary version after applying Otsu's method to the image in Fig. 20.

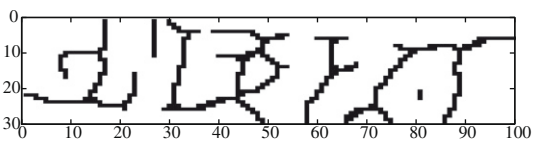

Fig. 22. Thickened version after applying the morphological thickening algorithm to the image in Fig. 21.

where c is the index for the columns of the shadowy region, and $ncol$ is the total quantity of columns of the IWS.

In addition,

$$\frac{1}{6}nrow \leqslant r \leqslant \frac{5}{6}nrow, \tag{29}$$

where r is the index for the rows of the shadowy region, and $nrow$ is the total quantity of rows of the IWS.

By considering this critical zone, the adapted pruning algorithm is applied to the image of Fig. 22. The IWS for locating the shadowy region corresponding to CST 1 (separation dash) is presented in Fig. 23. This IWS is used as input data to lighten the shadowy region that includes the separation dash as a critical shadow.

4.5. Step 5: Lightening the shadowy region for CST 1

In this step, the pixel intensities of the shadowy region which includes CST 1 (separation dash) are replaced with the gray-level average intensity of the image background.

The grayscale image in Fig. 24 is the result after lightening the shadowy region for CST 1.

Fig. 25 presents the binary version (without the separation dash) of the lightened grayscale image in Fig. 24.

4.6. Step 6: Obtaining the IWS for locating CST 2 and CST 3

Fig. 26 displays the thickened image after applying the thickening algorithm to the intermediate binary image in Fig. 25. The critical x coordinate was adopted as 50% of the total quantity of rows dividing the image into upper and lower critical zones, that is,

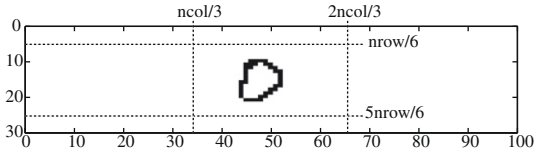

Fig. 23. Image without spurs (IWS) including the enclosing boundary of CST 1.

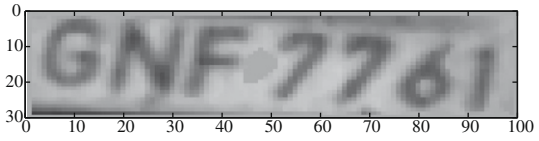

Fig. 24. Grayscale image with the lightened shadowy region for CST 1.

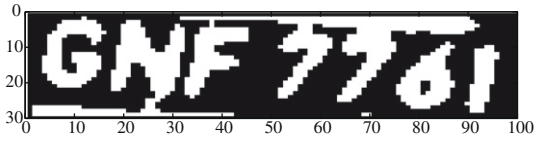

Fig. 25. Binary version of the image in Fig. 24 after processing CST 1.

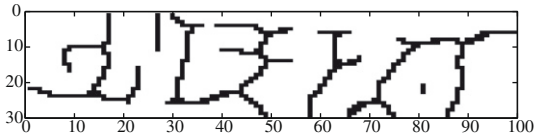

Fig. 26. Thickened version of the binary image in Fig. 25.

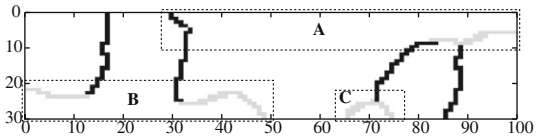

Fig. 27. Pruned version after applying the morphological pruning algorithm to the thickened image in Fig. 26.

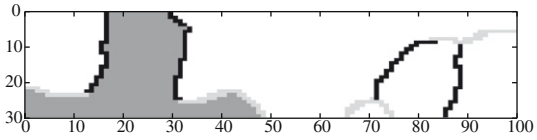

Fig. 28. Example of an overlapping shadowy region caused by the interrupted boundary line (gray lines on the area *B* of the image in Fig. 27).

$x_{at} = x_{pt} = 15$. The IWS in Fig. 27 is the pruned version after applying the morphological pruning algorithm to the image in Fig. 26.

In this IWS, the two types of critical shadows are as follows:

- CST 2: The resulting noise is isolated, and this noise is located around the lower border of the image in Fig. 27. The corresponding boundary of the shadowy region for this noise is indicated by the area *C* on the image of Fig. 27.
- CST 3: The corresponding binary noise in Fig. 25 is connected to one or more characters. So, the boundaries are interrupted resulting in untrue shadowy regions. The boundary interruptions are automatically repaired by determining a line segment that delimitates a new and correct shadowy region. An interruption is indicated by the area *A*, and another one is indicated by the area *B* on the unideal IWS in Fig. 27. The interruption problem located at the area *B* causes an overlapping shadowy region (grayscale part) as shown in the image of Fig. 28.

4.7. Step 7: Repairing the boundary line interruption

Fig. 29 presents the repaired version of the unideal IWS in Fig. 27 after applying the algorithms presented in Tables 2 and 3. The line segments, that repair the interruptions at the boundary lines, provide correct shadowy regions of CST 3 as input data to be lightened.

4.8. Step 8: Applying the algorithm to lighten shadows

The shadow lightening algorithm presented in Table 4 uses the repaired IWS in Fig. 29 as reference data to lighten the shadowy regions of the image in Fig. 24.

The resulting lightened version as the preprocessed grayscale image is shown in Fig. 30.

The image in Fig. 31 corresponds to the binary version of the lightened image in Fig. 30. We can verify in the image of Fig. 31 an isolated noise (superior zone) still to be removed. For locating this remaining noise, Fig. 31's image is input data to obtain another IWS in Fig. 32.

In fact, the steps 2–8 are repeated until the IWS is not generated more, that is, it does not exist valid IWS.

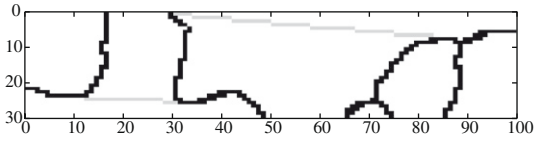

Fig. 29. Repaired version of the unideal IWS in Fig. 27. The line segments (gray color) repair the interruptions of the boundary lines and delimitate correct shadowy regions to be lightened.

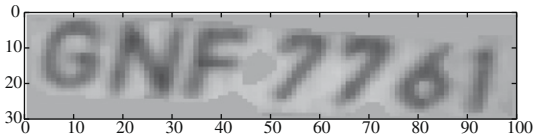

Fig. 30. Preprocessed grayscale version after applying the shadow lightening algorithm to the image in Fig. 24.

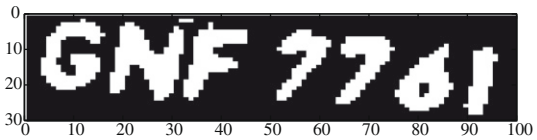

Fig. 31. Binary version for the lightened image in Fig. 30. There is an isolated noise still to be removed.

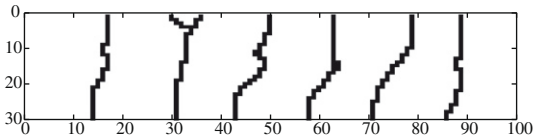

Fig. 32. Intermediate IWS based on the image of Fig. 31 to locate the remaining noise.

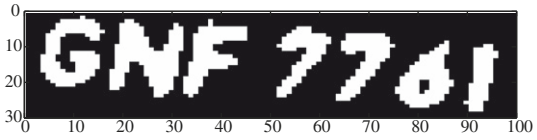

Fig. 33. Final version of the enhanced binary image. The stop condition (no more noise) of the method is satisfied.

4.9. Step 9: Obtaining the enhanced binary image

Finally, we have the enhanced binary word image in Fig. 33. In this step, the stop condition of the algorithm to lighten shadows is satisfied (it does not exist valid IWS for critical shadows), that is, there is no noise on the image of Fig. 33.

5. Experimental results

To evaluate the proposed preprocessing method (SL*L), a testing set of 1194 raw degraded word images was extracted from data of real photos automatically taken by unsophisticated cameras subjected to non-controlled environmental conditions (weather, luminosity, smog, smear, movement).

Two sets of binary images were generated from these raw degraded word images data as follows:

- First set: By directly applying Otsu's method to the grayscale versions of raw degraded word images.

- Second set: By applying the SL*L preprocessing method to the grayscale versions of raw degraded word images and subsequently applying Otsu's method to these preprocessed grayscale images.

The generated binary images were used as input data to segment their relevant objects (letters or digits). As validating criterion, the segmentation process was considered as correctly performed if all the objects were properly extracted from the original image and they matched the characters of this image in terms of their positions as well as quantities.

Table 5 presents segmentation performances using the binary versions (first set) of original degraded word images and the binary ones (second set) of these degraded word images preprocessed by the SL*L method as input data.

To inspect the effectiveness of the proposed preprocessing method, comparisons of the thresholding results for some sample data from the testing set of 1194 degraded word images are presented in Figs. 34–43. The listed images in each figure respectively correspond to (a) the grayscale version of a raw degraded word image, (b) the thresholding result applying only Otsu's method to the degraded word image, (c) the preprocessing result applying the SL*L method to the degraded word image, and (d) the binary

Table 5
Segmentation performances for different sets of thresholding results from degraded word images.

Thresholding results	Testing images	Segmented images	Segmented characters
First set (without preprocessing)	1194	none[a]	none[a]
Second set (with the SL*L preprocessing)	1194	1010	7098

[a] It was not possible to correctly segment and extract relevant objects from this set of binary images because of strong noise caused by critical shadows.

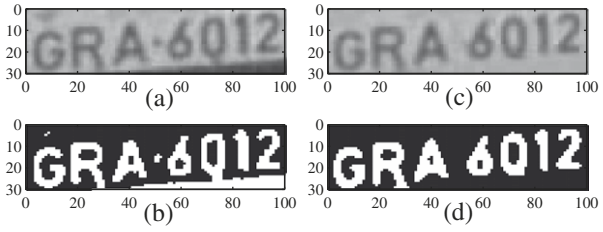

Fig. 34. Sample I: (a) degraded word image; (b) thresholding result by Otsu's method; (c) preprocessed image by the SL*L method; (d) enhanced binary version after applying Otsu's method to the SL*L preprocessed image.

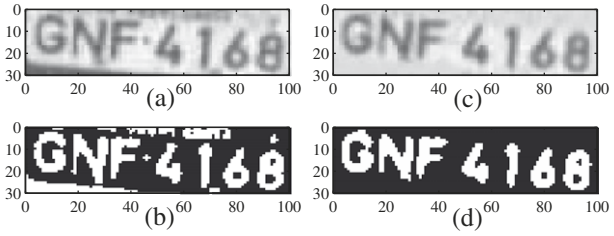

Fig. 35. Sample II: (a) degraded word image; (b) thresholding result by Otsu's method; (c) preprocessed image by the SL*L method; (d) enhanced binary version after applying Otsu's method to the SL*L preprocessed image.

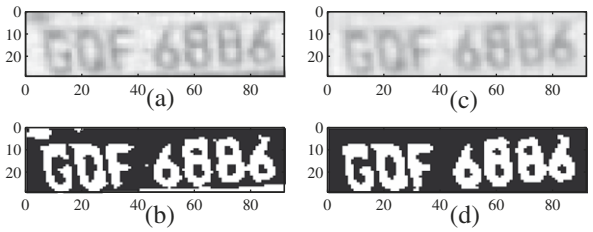

Fig. 36. Sample III: (a) degraded word image; (b) thresholding result by Otsu's method; (c) preprocessed image by the SL*L method; (d) enhanced binary version after applying Otsu's method to the SL*L preprocessed image.

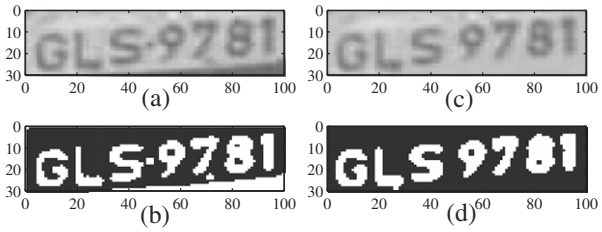

Fig. 37. Sample IV: (a) degraded word image; (b) thresholding result by Otsu's method; (c) preprocessed image by the SL*L method; (d) enhanced binary version after applying Otsu's method to the SL*L preprocessed image.

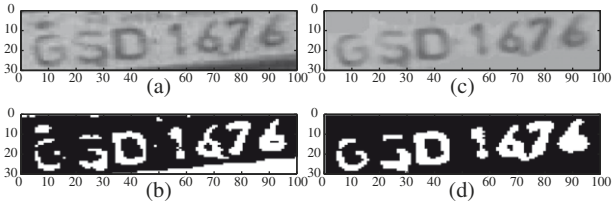

Fig. 38. Sample V: (a) degraded word image; (b) thresholding result by Otsu's method; (c) preprocessed image by the SL*L method; (d) enhanced binary version after applying Otsu's method to the SL*L preprocessed image.

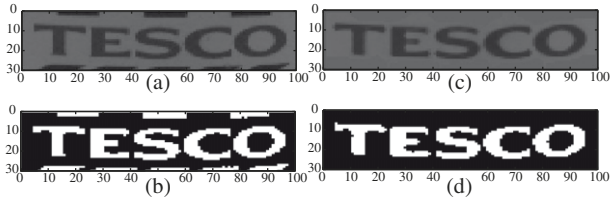

Fig. 39. Sample VI: (a) degraded word image; (b) thresholding result by Otsu's method; (c) preprocessed image by the SL*L method; (d) enhanced binary version after applying Otsu's method to the SL*L preprocessed image.

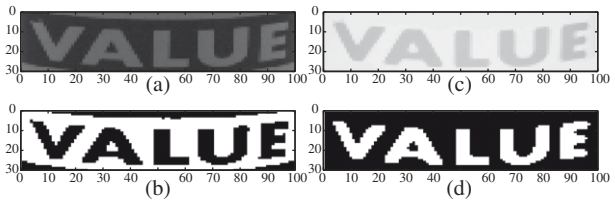

Fig. 40. Sample VII: (a) degraded word image; (b) thresholding result by Otsu's method; (c) preprocessed image by the SL*L method; (d) enhanced binary version after applying Otsu's method to the SL*L preprocessed image.

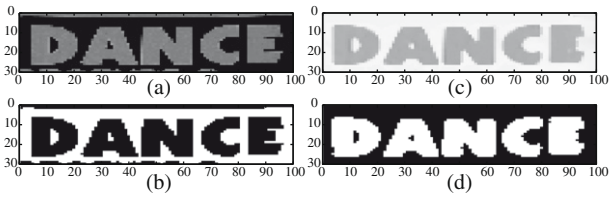

Fig. 41. Sample VIII: (a) degraded word image; (b) thresholding result by Otsu's method; (c) preprocessed image by the SL*L method; (d) enhanced binary version after applying Otsu's method to the SL*L preprocessed image.

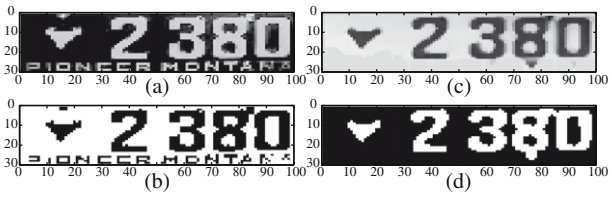

Fig. 42. Sample IX: (a) degraded word image; (b) thresholding result by Otsu's method; (c) preprocessed image by the SL*L method; (d) enhanced binary version after applying Otsu's method to the SL*L preprocessed image.

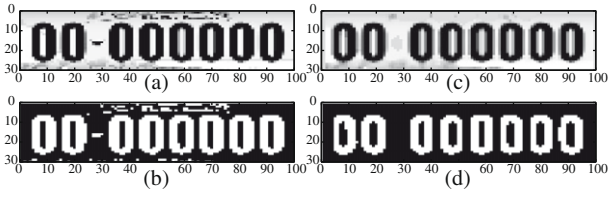

Fig. 43. Sample X: (a) degraded word image; (b) thresholding result by Otsu's method; (c) preprocessed image by the SL*L method; (d) enhanced binary version after applying Otsu's method to the SL*L preprocessed image.

version after applying Otsu's method to the grayscale image preprocessed by the SL*L method. Figs. 34–38 present thresholding results using some samples of raw degraded word images extracted from real photos automatically taken by unsophisticated imaging systems installed along the roads of Uberlândia City in Brasil (Nomura, 2002). The degraded word images of Figs. 39–41 were extracted from the ICDAR 2003 Robust Reading Dataset (ICDAR, 2003) used to text locating and page segmentation competitions. The degraded plate images of Figs. 42 and 43 were extracted from the number plate photos available on the Montana Department of Justice site (Montana Department of Justice, 2007).

Also, to confirm the essentialness of the SL*L preprocessing method, Figs. 44–53 show the segmentation results inputting the original degraded word images without preprocessing and degraded word ones preprocessed by the SL*L method for samples I–X. The implemented segmentation process was based on the adaptive morphological approach (Nomura et al., 2005) to segment characters from degraded images.

Furthermore, Table 6 shows the recognition results applying a state-of-the-art OCR engine to the original grayscale images and the binary images preprocessed by the SL*L presented in Figs. 34–43. As validating criterion, a character from the input image was considered as correctly recognized if its corresponding OCR result found the same code and position at the original image. In Table 6, the recognition rate was calculated by dividing the number of recognized characters by the total number of characters.

6. Discussion

In experiments for quantitative evaluation, using the first set of 1194 binary images generated by directly applying Otsu's method to the degraded images, the segmentation engine was not able to correctly segment and extract their relevant objects because of strong noise caused by critical shadows. On the other hand, using the second set of 1194 binary images enhanced by applying the SL*L preprocessing method to the degraded images before thresholding, the segmentation engine was able to correctly segment

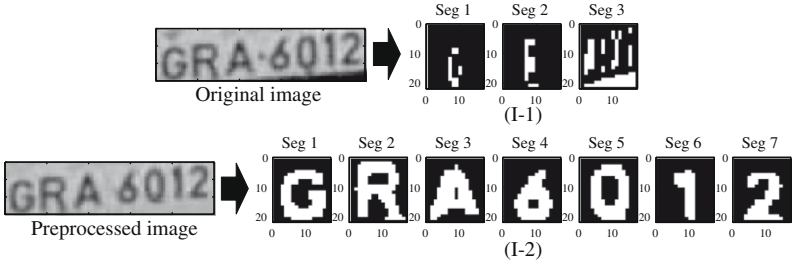

Fig. 44. Segmentation results for sample I: (I-1) using the original degraded word image as input data; (I-2) using the grayscale image preprocessed by the SL*L method as input data.

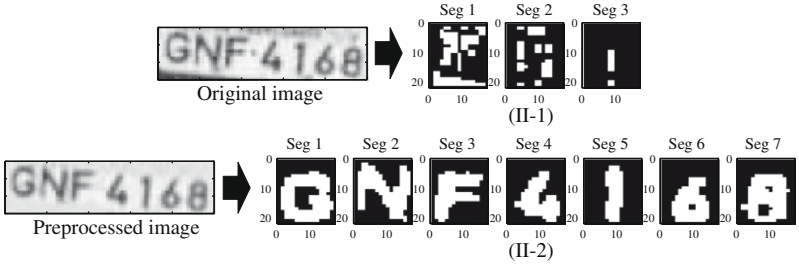

Fig. 45. Segmentation results for sample II: (II-1) using the original degraded word image as input data; (II-2) using the grayscale image preprocessed by the SL*L method as input data.

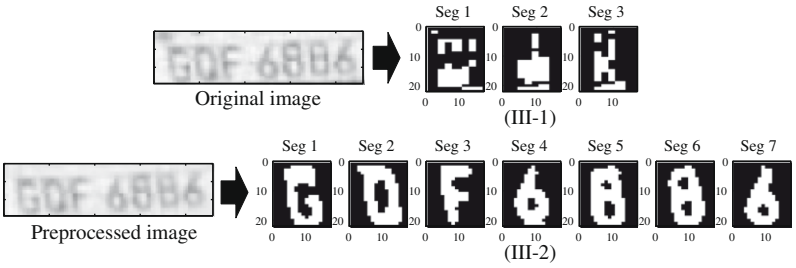

Fig. 46. Segmentation results for sample III: (III-1) using the original degraded word image as input data; (III-2) using the grayscale image preprocessed by the SL*L method as input data.

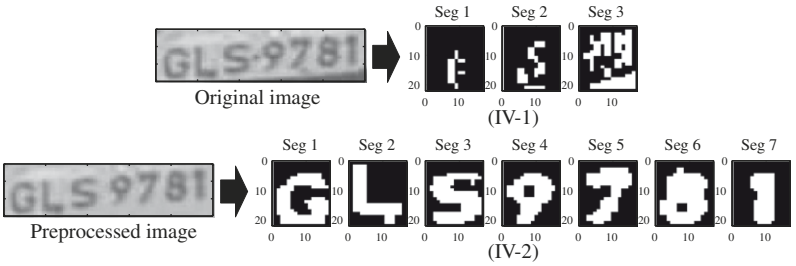

Fig. 47. Segmentation results for sample IV: (IV-1) using the original degraded word image as input data; (IV-2) using the grayscale image preprocessed by the SL*L method as input data.

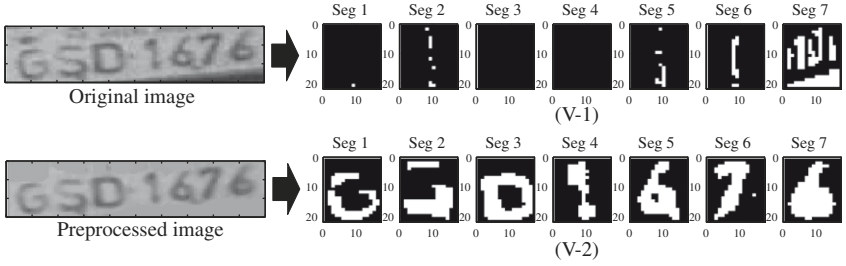

Fig. 48. Segmentation results for sample V: (V-1) using the original degraded word image as input data; (V-2) using the grayscale image preprocessed by the SL*L method as input data.

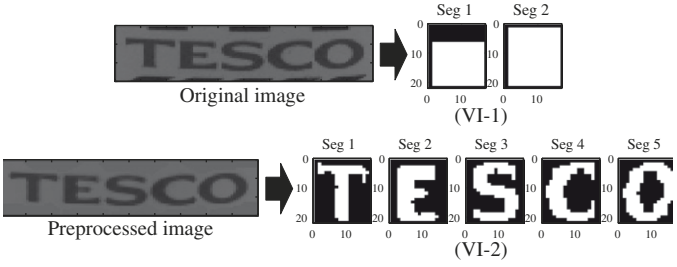

Fig. 49. Segmentation results for sample VI: (VI-1) using the original degraded word image as input data; (VI-2) using the grayscale image preprocessed by the SL*L method as input data.

and extract 7098 objects (characters) from 1010 binary versions of the testing set.

This segmentation performance shows that the SL*L preprocessing method was essential to generate enhanced binary images as appropriate input data.

The thresholding results in Figs. 34–43 show that Otsu's method applied to the degraded word image produced inappropriate noise due to the critical shadows. Whereas this noise was completely removed from binary versions after applying the SL*L preprocessing method to the degraded word images even using Otsu's method as the simplest thresholding method. A simple visual comparison between (b) and (d) thresholding results of Figs. 34–43 is practically enough to verify that the SL*L preprocessing method significantly contributed to remove noise and provide relevant features of characters from binary images.

Also, in Figs. 44–53 can be verified that the segmentation process was not able to correctly segment the binary characters of original degraded images thresholded by Otsu's method without preprocessing. However, it is verified that the characters of word images preprocessed by the SL*L method were adequately segmented according to the validating criterion for segmentation (all the extracted segments match the characters from the original word image regarding their positions and quantities).

In Table 6, the recognition rates by applying a state-of-the-art OCR engine to the images preprocessed by the SL*L method were equal or far higher than the rates by applying this OCR engine to the original degraded images. From the total number of 72 characters to be recognized, the OCR engine was able to recognize 47 characters using the preprocessed images and only 18 ones using the original degraded images as input data. In other words, the

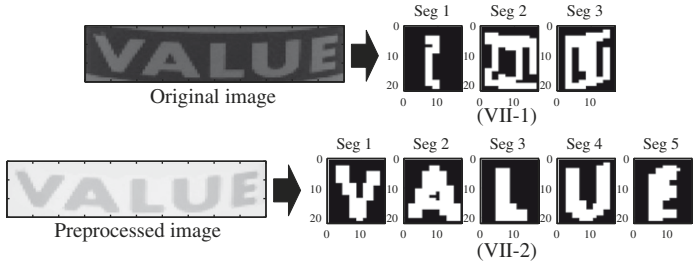

Fig. 50. Segmentation results for sample VII: (VII-1) using the original degraded word image as input data; (VII-2) using the grayscale image preprocessed by the SL*L method as input data.

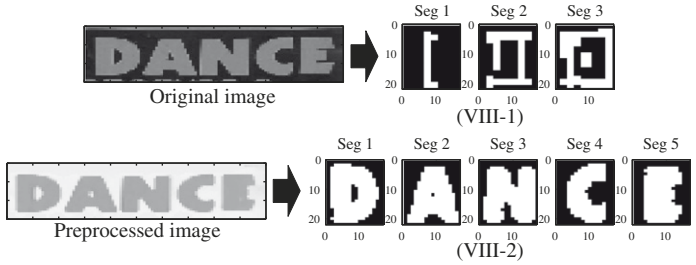

Fig. 51. Segmentation results for sample VIII: (VIII-1) using the original degraded word image as input data; (VIII-2) using the grayscale image preprocessed by the SL*L method as input data.

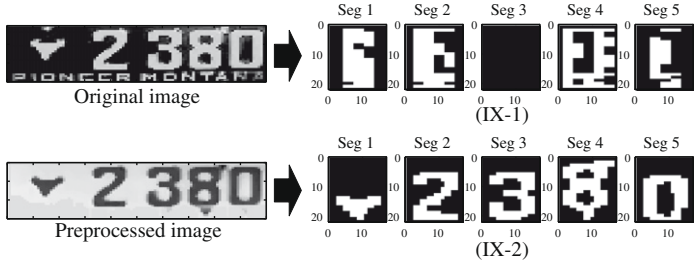

Fig. 52. Segmentation results for sample IX: (IX-1) using the original degraded word image as input data; (IX-2) using the grayscale image preprocessed by the SL*L method as input data.

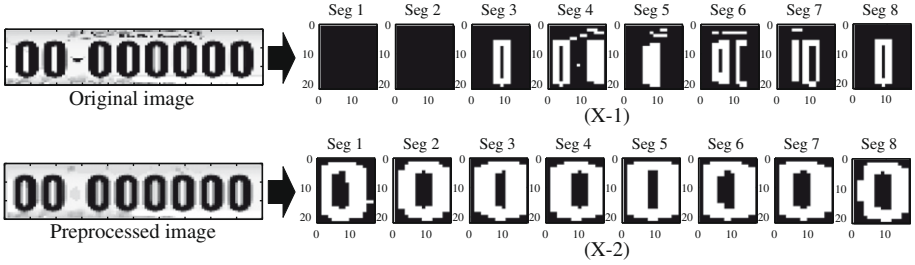

Fig. 53. Segmentation results for sample X: (X-1) using the original degraded word image as input data; (X-2) using the grayscale image preprocessed by the SL*L method as input data.

recognition rate increased 40% due to the SL*L method application. This increase is due to the high quality binary images provided by the SL*L preprocessing method and required by most commercially available OCR engines to work in real-time processing conditions.

7. Conclusion

A novel morphological method called Shadow Location and Lightening (SL*L) was proposed to preprocess raw degraded word images. In this method, powerful mathematical morphology operations were taken advantage to locate and lighten the critical shadows as the key problem that causes noise on thresholding results. We implemented the SL*L method on an experimental system to preprocess raw degraded word images and obtain enhanced binary versions without noise. Experimental evaluation confirmed that the preprocessing of raw degraded word images by the SL*L method was crucial and useful to provide more enhanced binary versions than the simple application of widely known thresholding methods. Otsu's method application to the degraded word images

Table 6
Comparison of recognition results applying different input images to a state-of-the-art OCR engine.

	Without preprocessing			With the SL*L preprocessing		
	Input image	OCR result	Rate	Input image	OCR result	Rate
I	Fig. 34a	6RA·6Q:lz2	0	Fig. 34d	GRA6012	1.00
II	Fig. 35a	9:IFFa'fis6	0	Fig. 35d	GNF"6e	0.57
III	Fig. 36a	t1MF6ses	0	Fig. 36d	GEF66e6	0.57
IV	Fig. 37a	GLS·gue	0.43	Fig. 37d	GLS97S1	0.86
V	Fig. 38a	1.3D·1{_7s	0	Fig. 38d	b=v167'	0.43
VI	Fig. 39a	TESCO	1.00	Fig. 39d	TESCO	1.00
VII	Fig. 40a	VALUE	1.00	Fig. 40d	VALUE	1.00
VIII	Fig. 41a	DANCE	1.00	Fig. 41d	DANCE	1.00
IX	Fig. 42a	2390	0	Fig. 42d	v2380	1.00
X	Fig. 43a	00·aooum	0	Fig. 43d	00000000	1.00

preprocessed by the SL*L method was enough to improve the quality of binary versions. We concluded that the SL*L method can produce appropriate thresholding results with useful features of characters while reducing the effect of noise, that is, the goal was achieved to solve the non-linear problem caused by critical shadows on degraded word images. Furthermore, the SL*L method provided a better utilization of significant degraded word images to be fed as input data to subsequent segmentation and recognition processes. These degraded images had to be discarded due to excessive noise and poor quality features of their binary versions generated by conventional thresholding methods without the SL*L preprocessing application. The SL*L method is expected to contribute to preprocessing and utilization of vital degraded word images for industrial-like environment systems represented by most commercially available OCR-based applications.

References

Abak, T., Baris, U., Sankur, B., 1997. The performance evaluation of thresholding algorithms for optical character recognition. In: Internat. Conf. on Document Analysis and Recognition, ICDAR'97, pp. 697–700.

Alcorn, T.M., Hoggar, C.W., 1969. Pre-processing of data for character recognition. Marconi Rev. 32, 61–81.

Bernsen, J., 1986. Dynamic thresholding of grey-level images. In: Proc. of the 8th Internat. Conf. on Pattern Recognition, pp. 1251–1255.

Flusser, J., Suk, T., 1998. Degraded image analysis: An invariant approach. IEEE Trans. PAMI 20, 590–603.

Gao, H., Siu, W., Hou, C., 2001. Improved techniques for automatic image segmentation. IEEE Trans. Circuits Syst. Video Technol. 11 (12), 1273–1280.

Gatos, B., Pratikakis, I., Perantonis, S.J., 2006. Adaptive degraded document image binarization. Pattern Recognition 39 (3), 317–327.

Gonzalez, R.C., Woods, R.E., 1993. Digital Image Processing. Addison-Wesley Publishing Company, Reading, MA.

Heijmans, H.J.A.M., 1992. Mathematical morphology: A geometrical approach to image processing. Nieuw Arch. Wiskd. Vierde Serie, Deel 10 (3), 237–276.

ICDAR, 2003. The 7th Internat. Conf. on Document Analysis and Recognition (ICDAR 2003) datasets available from <http://algoval.essex.ac.uk/icdar/Datasets.html>.

Impedovo, S., Ottaviano, L., Occhiegro, S., 1991. Optical character recognition – A survey. Internat. J. Pattern Recognition Artif. Intell. 5, 1–24.

Kamel, M., Zhao, A., 1993. Extraction of binary character/graphics images from grayscale document images. Graph. Models Image Process. 55 (3), 203–217.

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

Mo, S., Mathews, V.J., 1998. Adaptive, quadratic preprocessing of document images for binarization. IEEE Trans. Image Process. 7, 992–999.

Montana Department of Justice, 2007. The Montana Department of Justice antique plates available from http://www.doj.mt.gov/driving/platedesign/antique.asp.

Nagy, G., 2000. Twenty years of document image analysis in PAMI. IEEE Trans. Pattern Anal. Machine Intell. 22, 38–62.

Niblack, W., 1986. An Introduction to Digital Image Processing. Prentice Hall, New Jersey.

Nomura, S., 2002. New Methods for Image Binarization and Character Segmentation Applied to an Automatic Recognition System of Number Plates. M.S. Thesis (in Portuguese), Faculty of Electr. Eng., Federal Univ. of Uberlandia, Brazil.

Nomura, S., Yamanaka, K., Katai, O., Kawakami, H., 2004. A new method for degraded color image binarization based on adaptive lightning on grayscale versions. IEICE Trans. Inform. Systems E87-D (4), 1012–1020.

Nomura, S., Yamanaka, K., Katai, O., Kawakami, H., Shiose, T., 2005. A novel adaptive morphological approach for segmenting characters in degraded images. Pattern Recognition 38 (11), 1961–1975.

Otsu, N., 1979. A threshold selection method from gray-level histograms. IEEE Trans. Systems Man Cybernet. 9, 62–66.

Sahoo, P.K., Soltani, S., Wong, A.K.C., Chen, Y., 1988. A survey of thresholding techniques. Comput. Vision Graphics Image Process. 41, 233–260.

Sankur, B., Sezgin, M., 2004. Survey over image thresholding techniques and quantitative performance evaluation. J. Electron. Imaging 13 (1), 146–165.

Sauvola, J., Pietikäinen, M., 2000. Adaptive document image binarization. Pattern Recognition 33, 225–236.

Serra, J., 1982. Image Analysis and Mathematical Morphology. Academic Press, London.

Soille, P., 1999. Morphological Image Analysis: Principles and Applications. Springer-Verlag, Berlin.

Talbot, H., Beare, R., 2002. In: Proc. of the 6th Internat. Symp. on Mathematical Morphology. CSIRO Publishing, Sydney.

Thillou, C., Gosselin, B., 2005. Color binarization for complex camera-based images. Proc. of SPIE 5667, 301–308.

Trier, O.D., Jain, A.K., 1995. Goal-directed evaluation of binarization methods. IEEE Trans. PAMI 17 (12), 1191–1201.

Trier, O.D., Taxt, T., 1995. Evaluation of binarization methods for document images. IEEE Trans. Pattern Anal. Machine Intell. 17 (3), 312–315.

Tsai, C.M., Lee, H.J., 2002. Binarization of color document images via luminance and saturation color features. IEEE Trans. Image Process. 11, 434–451.

Wang, K., Kangas, J., 2003. Character location in scene images from digital camera. Pattern Recognition 36 (10), 2287–2299.

Xu, Y., Navy, G., 1999. Prototype extraction and adaptive OCR. IEEE Trans. PAMI 21 (12), 1280–1296.

Yanowitz, S.D., Bruckstein, A.M., 1989. A new method for image segmentation. Comput. Vision Graphics Image Process. 46 (4), 82–95.