Text Regions Extracted from Scene Images by Ultimate Attribute Opening and Decision Tree Classification

Wonder Alexandre Luz Alves and Ronaldo Fumio Hashimoto
Department of Computer Science
Institute of Mathematics and Statistics, University of São Paulo, Brazil
Email: {wonder, ronaldo}@ime.usp.br

***Abstract*—In this work we propose a method for localizing text regions within scene images consisting of two major stages. In the first stage, a set of potential text regions is extracted from the input image using residual operators (such as ultimate attribute opening and closing). In the second stage a set of features is obtained from each potential text region and this feature set will be later used as an input to a decision tree classifier in order to label these regions as text or non-text regions. Experiments performed using images from ICDAR public dataset show that this method is a good alternative for problems involving text location in scene images.**

***Keywords*-scene-text localization, connected component approach, text information extraction, residual operator.**

Figure 1. Scene images taken from ICDAR database [6].

I. Introduction

The growing availability of online digital media has spurred interest in studies on how to index them automatically and then how to efficiently browse, search and manipulate such information. Owing to this fact, it has been observed in recent years an increasing effort to create mechanisms for content-based digital media indexing. However, on one hand, low level descriptors (such as, interest points or texture features) used for indexing are not sufficient for dealing with large and generic amounts of data. On the other hand, texts present within scenes are usually associated to digital media semantic contexts and may constitute important index content-based descriptors.

Based on these considerations, several algorithms for *text information extraction* (TIE) from scene images and videos have been proposed in the literature for general purposes (such as, text extraction from urban scenes [1] and content-based image and video indexing [2]) and also for specific purposes (such as, vehicle license plate number extraction [3] and postcard address extraction [4]). In spite of such studies, it is not an easy task to design a general-purpose TIE system [5] for scene digital media, since texts present within these type of media (scenes) are considered as an integrating part of the scene and their presence are usually accidental and non-intentional. Thus, text occurrences in scene digital media can be significantly different from one another with respect to slopes, sizes, font styles, illumination and also they can be partially occluded. In Fig. 1, we show some examples of images that belong to this category.

Despite the TIE problem classification given in [5], there is a certain confusion between text detection and localization concepts, since many authors use both terms to indicate the problem whose main objective is to locate text regions within an input image. Our work focuses on text localization problems which can be divided into two categories based on the method used for solving them: texture-based [7], [8] and region-based [9], [10], [11], [12], [13] methods.

Texture-based methods [7], [8] assume that characters within a text region have texture attributes which are different from their background making them possible to be discriminated. Techniques such as Gabor filter, wavelets, fast Fourier transform can be used to extract texture attributes from potential text regions in order to classify them.

Region-based methods [9], [10], [11], [12], [13] can still be divided into two approaches: edge-based [9], [10], [13] and connected component (CC)-based [11], [12] methods. Edge-based methods work in a top-down fashion by emphasizing high contrast between texts and their background in order to identify regions and classify them into text or non-text regions. CC-based methods work in a bottom-up fashion by identifying and combining sub-structures such as CCs and classify them into text or non-text regions. In fact, for the latter approach, it is common to use morphological filters to identify text region candidates [9], [10], [11], [12], [13] and then applying heuristics [9], [10], [12] or classifiers [11], [13], [14] to label them as text or non-text regions. In fact, recently, it has been proposed a region-based method to label regions, extracted from scene images by morphological filters (opening and closing top-hats), into text or non-text region by using a decision tree classifier [13]. Experimental results presented in this work have shown a better performance than Wu's work [9].

With the previous considerations in mind, the main objective of this paper is to provide an improvement of the region-based method published in [13] by performing the following steps: (i) use ultimate attribute openings and closings (as in [11]) to identify CCs corresponding to potential characters; (ii) combine some of these CCs in order to obtain a set of text region candidates; (iii) select a subset of text region candidates in order to form a set of potential text regions; (iv) extract a set of textual attributes from these potential text regions; (v) and finally, apply a decision tree classifier in order to label them as text or non-text regions. Steps from (i) to (iii) are different from work [13]. Steps from (ii) to (v) are different from work [11].

The remainder of this paper is organized as follows. In Section II, we give a brief description of ultimate attribute opening (and closing). The proposed method is presented in Section III and some experimental results are provided in Section IV. Finally, we conclude this work in Section V with some future perspective for this research.

II. Ultimate Attribute Opening

The ultimate attribute opening can be seen as a non-parametric method based on numerical residues [15] obtained by morphological operators. One application of this method is to extract CCs with no prior information about their sizes [11]. The organization of this section is the following: Section II-A presents an overview of ultimate opening; Section II-B gives a brief recall of attribute opening; and, finally, Section II-C, using the previous concepts, presents the definition of the ultimate attribute opening.

A. Ultimate Opening

The ultimate opening θ (and the ultimate closing ρ, by duality) is a residual operator [15] which successively (i) applies openings $\gamma_i(f)$ with increasing structuring elements (SEs) of size i, $i = 1, \dots, n$; (ii) computes the residues r_i obtained by the difference of two successive openings γ_i and γ_{i+1} (that is, $r_i = \gamma_i - \gamma_{i+1}$); and finally (iii) calculates the supremum residue. More formally, given an input image f, the ultimate opening θ provides two types of information stored in two images:

$$R_\theta(f) = \sup_{0<i<n} \{r_i(f) : r_i(f) = \gamma_i(f) - \gamma_{i+1}(f)\},$$
$$[q_\theta(f)](x) = \max_{0<i<n} \{i : [r_i(f)](x) \geq [R_\theta(f)](x) \text{ and } [r_i(f)](x) > 0\}.$$

That is, for each pixel x, $[R_\theta(f)](x)$ gives the maximal residue, while $[q_\theta(f)](x)$ gives the size of the opening that leads this residue. (By duality, we can define the ultimate closing ρ that provides $R_\rho(f)$ and $q_\rho(f)$.) With this definition in mind, we can conclude that one objective of the ultimate opening is to highlight patterns with higher contrast.

B. Attribute Opening

A binary attribute opening of a binary image $X \subset \mathbb{Z}^2$ consists of an opening by reconstruction associated to an increasing criterion κ (such as area, width, height, etc.) applied to each CC of X [16] in order to keep or to reject it into the output image. For gray-level images, the attribute opening γ_κ can be defined as

$$[\gamma_\kappa(f)](x) = \max\{h : x \in \Gamma_\kappa(T_h(f))\},$$

where $T_h(f)$ is the binary image obtained by thresholding f at level h and Γ_κ is a binary attribute opening using criterion κ.

C. Ultimate Attribute Opening

In the ultimate opening definition, it is possible to use attribute openings to build the set of openings with increasing size. In this case, this operator is called *ultimate attribute opening*. The chosen attribute for our application was the *height of the CC*, defined as the maximum difference of vertical coordinate among pixels belonging to that CC.

In this work, we use the efficient implementation of ultimate attribute opening proposed in [17] and the height criterion to extract a set of CCs from the input image. Fig. 2 presents an example of the application of the ultimate attribute opening.

(a) Input image f

(c) $q_\theta(f)$

(b) $R_\theta(f)$

Figure 2. Example of the application of the ultimate attribute opening.

III. Proposed Method

The proposed method consists of two major stages (see Fig. 3) which comprise *potential text regions extraction* and *labeling these regions into text or non-text regions* by using a decision tree classifier. In the first stage (see Section III-A), a set of potential text regions is extracted from the input image using morphological filters and it is briefly comprised by the following steps: (i) use ultimate attribute openings and closings to identify CCs corresponding to potential characters; (ii) combine some of these CCs in order to obtain a set of text region candidates; (iii) select a subset of text region candidates in order to form a set of potential text regions.

The second stage (see Section III-B) of the method consists of (iv) obtaining a set of features from the potential text regions that will be later used as an input to a decision tree classifier in order to (v) label them as text or non-text regions.

This work, as well as others [9], [10], [11], [13], [14], assumes that text regions consist of at least three characters, and also assumes as true some hypotheses that take account the contrast between text regions and their background and the geometric regularity of font character within text regions. Such hypotheses (called here as *text region hypotheses*) are classified as following:

1) *Contrast*
   a) There is a contrast between characters of text region and its background.
   b) Text gray levels within the same region are similar.
2) *Font Geometry*
   a) Characters within the same region
      - have similar heights and widths.
      - are aligned along a line.
      - have similar distances between any two adjacent characters.

A. First Stage: Potential Text Regions Extraction

The extraction procedure for potential text regions from the input image is illustrated in the flow chart presented in Fig. 4.

The extraction procedure of potential text regions can be divided into 8 steps. Just to make the first stage procedure clearer, we should remark that Steps from 1 to 4 correspond to text region candidates extraction, while Steps from 5 to 8 correspond to potential text regions extraction.

Step 1: If the input f is a color image, then it is converted into a gray scale image using the the luminance Y component of YIQ color model [18]:

$$f(x) = \lfloor 0.299 \cdot f_r(x) + 0.587 \cdot f_g(x) + 0.114 \cdot f_b(x) \rfloor,$$

where f_r, f_b, f_g are the three RGB components of the input image f. The choice for this model is because the luminance component Y provides more contrast information than the intensity component I from HSI color model. This is very important in our method, since our approach depends on the contrast between text characters and their background.

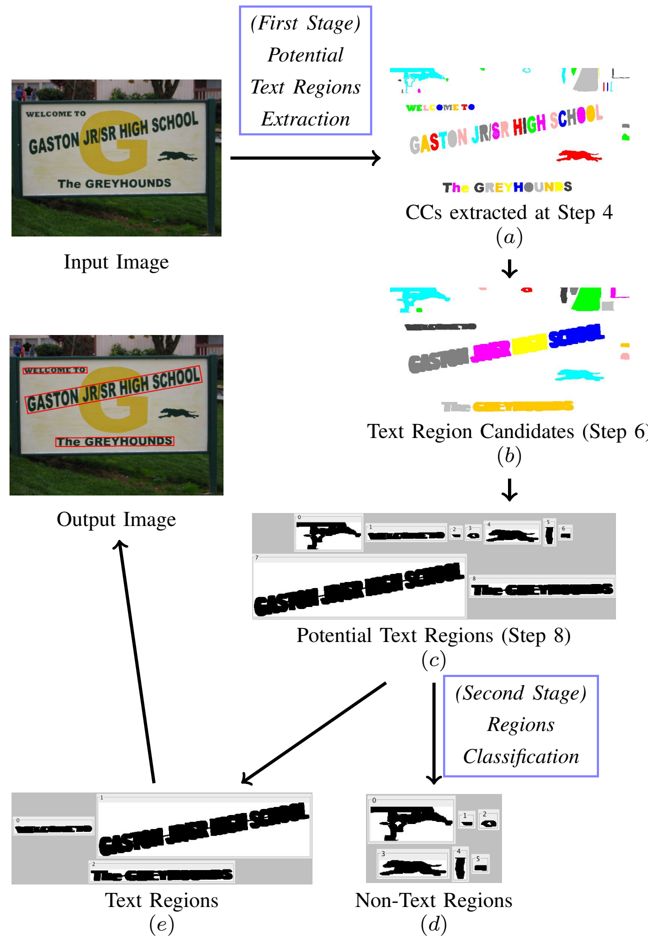

Figure 3. Overview of the proposed method. First stage: potential text regions extraction. (a) Image F obtained at the end of Step 4; (b) Image F_s obtained at the end of Step 6; (c) Set ζ obtained at the end of Step 8; (d) and (e) Second stage: classification of potential regions into text or non-text regions.

Step 2: Then, the ultimate attribute opening and closing are applied to the image f, using criteria of height (in this case $1/3$ of height f), producing four output images R_θ, q_θ, R_ρ and q_ρ, respectively.

Step 3: Then, the thresholdings $\tau(R_\theta)$ and $\tau(R_\rho)$ are computed. For a given image g, the thresholding $\tau(g)$ is based on toggle mapping operator [19] and is computed in the following way:

$$[\tau(g)](x) = \begin{cases} 1, & \text{if } [\delta_B(g)](x) - [\varepsilon_B(g)](x) \geq \alpha \text{ and} \\ & g(x) - [\varepsilon_B(g)](x) \leq [\delta_B(g)](x) - g(x) \\ 0, & \text{if } [\delta_B(g)](x) - [\varepsilon_B(g)](x) \geq \alpha \text{ and} \\ & g(x) - [\varepsilon_B(g)](x) > [\delta_B(g)](x) - g(x) \\ \alpha, & \text{otherwise,} \end{cases}$$

where δ_B and ε_B are dilation and erosion operators [19] by

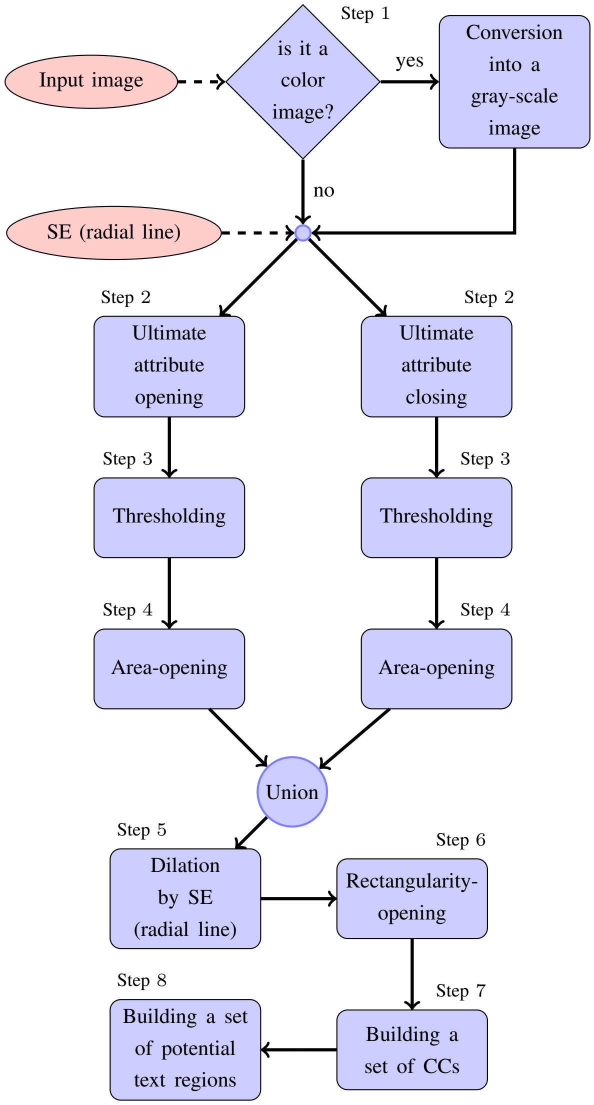

Figure 4. A simple flow chart showing our procedure for extraction of potential text regions (first stage).

the 3×3 square SE centered at the origin, respectively; and

$$\alpha = \begin{cases} [T_{\text{Otsu}}(g)](x), & \text{if } T_{\text{Otsu}}(g) > 10 \\ [T_{10}(g)](x), & \text{otherwise,} \end{cases}$$

where T_{Otsu} is the threshold obtained by Otsu's method [20] and T_{10} is the threshold at a fixed level 10.

Step 4: Then, in order to eliminate small CCs that have been come from noises within the input image f, an area-opening (attribute opening) is applied to the images $\tau(R_\theta)$ and $\tau(R_\rho)$, obtaining the images F_o and F_c, respectively. After that, by taking the union between F_o and F_c, we obtain the binary image $F = F_o \cup F_c$. The binary image F (see Fig. 3a for an example) may contain CCs corresponding to characters extracted from zones with some level of contrast. These CCs will be latter used to build the text region candidates.

Step 5: Now, we form regions by applying dilations by radial lines [19] to the binary image F obtaining a binary image F_R containing text region candidates.

Step 6: Heuristics based on the region rectangularity are highly employed in text localization problems to eliminate CCs that are considered impossible to be within text regions [9], [10], [13]. Thus, after forming text region candidates within the image F_R, it is applied an attribute opening to F_R using the criterion κ based on the rectangularity of the CCs in order to eliminate small and large CCs which do not have rectangular shape. The κ criterion is defined for each CC R belonging to F_R in the following way:

$$\kappa(R) = \frac{|R|}{W_R \cdot H_R} > 0.2 \quad \text{and} \quad \frac{\max\{W_R, H_R\}}{\min\{W_R, H_R\}} > 1.5,$$

where $|R|$ denotes the area of the CC R; W_R and H_R are the width and height of R, respectively. In this way, the image F_s is obtained by the application of this binary attribute opening, that is, $F_s = \Gamma_\kappa(F_R)$. Thus, F_s contains text region candidates of the input image f.

Step 7: Then, the image F_s is decomposed into n CCs, that is, $\Lambda(F_s) = \{R_1, R_2, ..., R_n\}$, and each element $R \in \Lambda(F_s)$ corresponds to a text region candidate.

Step 8: At this time, some regions in $\Lambda(F_s)$ may be small fragments of a larger region. Thus, we will perform a geometric analysis to merge some regions which may belong to the same text region. For this step, consider initially that $\zeta \leftarrow \{\{R_i\} : R_i \in \Lambda(F_s)\}$. Given an element $\mathcal{R} \in \zeta$, let $\omega_\mathcal{R}$ be the angle of $\mathcal{R}$ along its longest axis computed from the moment-based orientation estimation [18]. In addition, let $\mathcal{R}^\omega$ denote the rotated version of $\mathcal{R}$ by the angle $\omega_\mathcal{R}$ along its longest axis [18].

Two subsets $\mathcal{R}, \mathcal{S} \in \zeta$ belong to the same text region, if their orientations, heights, placements and alignments are very similar [9], [13]. So, if $\mathcal{R}$ and $\mathcal{S}$ belong to the same text region, the both criteria described below must be satisfied.

1) The orientations of $\mathcal{R}$ and $\mathcal{S}$ must be similar:

$$\begin{aligned} d_{\text{orientation}}(\mathcal{R}, \mathcal{S}) \quad = \quad & \min\{|\omega_\mathcal{R} - \omega_\mathcal{S}|, \\ & |\omega_\mathcal{R} - \omega_\mathcal{S} + 360|, \\ & |\omega_\mathcal{R} - \omega_\mathcal{S}| - 360\} < 15. \end{aligned}$$

This criterion comes from [9].

2) The heights of $\mathcal{R}$ and $\mathcal{S}$ must be similar:

$$d_{\text{height}}(\mathcal{R}, \mathcal{S}) = |H_{\mathcal{R}^\omega} - H_{\mathcal{S}^\omega}| < \min\{H_{\mathcal{R}^\omega}, H_{\mathcal{S}^\omega}\},$$

where $H_{\mathcal{R}^\omega}$ and $H_{\mathcal{S}^\omega}$ are the heights of the rotated components $\mathcal{R}^\omega$ and $\mathcal{S}^\omega$, respectively. This criterion was inspired from the work [11].

3) The centroids of $\mathcal{R}$ and $\mathcal{S}$ must be similar and aligned along a line:

$$\begin{aligned} d_{\text{centroid}}(\mathcal{R}, \mathcal{S}) &= \|\text{Cen}(\mathcal{R}) - \text{Cen}(\mathcal{S})\|^2 \\ &< \max\{W_{\mathcal{R}^\omega}, W_{\mathcal{S}^\omega}\}, \end{aligned}$$

where $\text{Cen}(\mathcal{Q})$ is the position of the centroid of the component $\mathcal{Q}$ and $W_{\mathcal{Q}^\omega}$ is the width of the rotated component $\mathcal{Q}^\omega$. This criterion was inspired from the work [9].

4) Let $L_\mathcal{Q}$ be the longest axis of $\mathcal{Q}$ obtained by the equation $y(\mathcal{Q}) = m_\mathcal{Q} \cdot x + b_\mathcal{Q}$. Then, the distance between the axes $L_\mathcal{R}$ and $L_\mathcal{S}$ of $\mathcal{R}$ and $\mathcal{S}$, respectively, can be defined as:

$$\begin{aligned} d_{\text{linearity}}(\mathcal{R}, \mathcal{S}) &= \frac{|y_{Cen_\mathcal{S}} - m_\mathcal{R} \cdot x_{Cen_\mathcal{S}} - b_\mathcal{R}|}{2\sqrt{1 + m_\mathcal{R}^2}} \\ &+ \frac{|y_{Cen_\mathcal{R}} - m_\mathcal{S} \cdot x_{Cen_\mathcal{R}} - b_\mathcal{S}|}{2\sqrt{1 + m_\mathcal{S}^2}}, \end{aligned}$$

where $x_{Cen_\mathcal{Q}}$ and $y_{Cen_\mathcal{Q}}$ are the coordinates of the centroid $Cen_\mathcal{Q}$. This criterion requires that $d_{\text{linearity}}(\mathcal{R}, \mathcal{S}) < \min\{H_{\mathcal{R}^\omega}, H_{\mathcal{S}^\omega}\}$. This criterion was inspired from the work [9].

Based on these four criteria, it is possible to decide whether $\mathcal{R}$ and $\mathcal{S}$ will be merged or not, that is, if $\mathcal{R}' \leftarrow \mathcal{R} \cup \mathcal{S}$ will be placed into ζ and, at the same time, $\mathcal{R}$ and $\mathcal{S}$ will be removed from ζ. At the end of Step 8, set ζ contains all the potential text regions (see Fig. 3c for an example).

B. Second Stage: Labeling Regions into Text or Non-Text Regions

After building the set ζ that contains all potential text regions, the next step consists of using a decision tree classifier to discriminate whether a given element $\mathcal{R} \in \zeta$ is a text or a non-text region. Most ideas presented in this section have been inspired from works [9], [10], [13]. Differently from [9], [10], our work has distinct bounds or inequalities.

1) Feature Extraction: In the following, we will describe how to obtain the features set from $\mathcal{R} \in \zeta$ that will be later used for classification. Given an element $\mathcal{R} \in \zeta$ and its rotated version $\mathcal{R}^\omega$, the x-projection technique [9] projects onto a line all pixels within the region $\mathcal{R}^\omega$, column by column. (see Fig. 5 for an example).

If $\eta(\mathcal{R}^\omega)$ is the vector that stores the x-projection of $\mathcal{R}^\omega$, the deepest valleys of $\eta(\mathcal{R}^\omega)$ provide an important information about the segmented elements (which will simply call from this moment on by *characters*) within $\mathcal{R}^\omega$ and they can be detected by a simple threshold.

Here, we should remark that the number of CCs inside $\mathcal{R}^\omega$ may be different from the number of characters. For a simple example, just consider the letters 'i' or 'j': although there are 2 (two) CCs, the x-projection will provide just 1 (one) character.

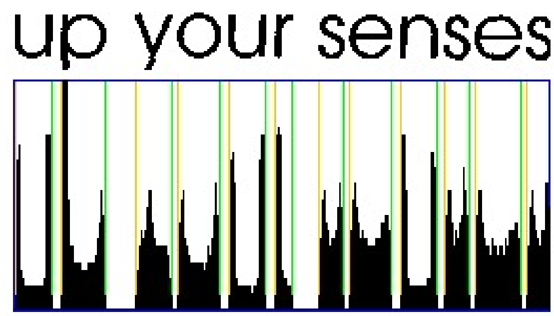

Figure 5. Example of the application of x-projection (bottom) to a potential text region (top). The Yellow and green lines indicate, respectively, the beginning and the end of a character segmentation.

Given an element $\mathcal{R} \in \zeta$, let $\overline{W_{\mathcal{R}^\omega}}$ and $\overline{H_{\mathcal{R}^\omega}}$ be, respectively, the average width and height of all characters (just a reminder, elements extracted from the x-projection technique) $C \in \mathcal{R}^\omega$. Then the width and height variances are, respectively,

$$\sigma^2_{W,\mathcal{R}^\omega} = \frac{1}{N_C} \cdot \sum_{C \in \mathcal{R}^\omega} (W_C - \overline{W_{\mathcal{R}^\omega}})^2, \tag{1}$$

$$\sigma^2_{H,\mathcal{R}^\omega} = \frac{1}{N_C} \cdot \sum_{C \in \mathcal{R}^\omega} (H_C - \overline{H_{\mathcal{R}^\omega}})^2, \tag{2}$$

where N_C is the number of characters in $\mathcal{R}^\omega$.

Now, consider that all characters within $\mathcal{R}^\omega$ form a line satisfying the linear equation $y = m_{\mathcal{R}^\omega} \cdot x + b_{\mathcal{R}^\omega}$, where the coefficients $m_{\mathcal{R}^\omega}$ and $b_{\mathcal{R}^\omega}$ can be obtained from linear regression [21]. Thus, a way for measuring the linearity of all characters of $\mathcal{R}^\omega$ can be defined as

$$Lin(\mathcal{R}^\omega) = \frac{1}{N_C} \cdot \sum_{C \in \mathcal{R}^\omega} \frac{|y_C - m_{\mathcal{R}^\omega} \cdot x_C + b_{\mathcal{R}^\omega}|}{\sqrt{(m_{\mathcal{R}^\omega})^2 + 1}}. \tag{3}$$

Now, let us consider the gray levels of all characters of $\mathcal{R}^\omega$. Let μ be the average gray level of all characters $C \in \mathcal{R}^\omega$ using the corresponding gray levels of the input image f. Thus, the homogeneity of $\mathcal{R}^\omega$ can be defined by

$$Hom(\mathcal{R}^\omega) = \frac{1}{N_P} \cdot \sum_{C \in \mathcal{R}^\omega} \sum_{x \in C} (f(x) - \mu)^2, \tag{4}$$

where N_P is the number of pixels of all characters in $\mathcal{R}^\omega$.

Based on the hypotheses described in Section III, the considered features are:

- *Contrast based feature*
  - Gray level homogeneity (Eq. 4).
- *Font geometric based features*
  - Number of characters of $\mathcal{R}^\omega$;
  - Height variance of characters in $\mathcal{R}^\omega$ (Eq. 1);
  - Width variance of characters in $\mathcal{R}^\omega$ (Eq. 2);

– Linearity of characters in $\mathcal{R}^\omega$ (Eq. 3).

All these values together form a vector $\nu(\mathcal{R}^\omega) \in \Re^5$

2) Classifier Design: A decision tree classifier is built using a training set $S = \{(p_i, c_i) \in \Re^5 \times \{1,2\} : i = 1, 2, \ldots, n\}$ of labeled feature vectors (labels 1 and 2 for text and non-text regions, respectively), obtained by the application of the measures described in the previous subsection. Similarly to [13], the C4.5 algorithm [22] (implemented in WEKA[1] software) was used for building the decision tree.

We have built a training set S for our decision tree containing about 1093 feature vectors extracted from all the 258 images of the training set of ICDAR dataset [6] in which about 466 feature vectors are from class 1 (text region), while about 637 feature vectors are from class 2 (non-text region). The training error obtained for the decision tree classifier is about 8.6% using the traditional k-fold-cross-validation (with $k = 10$) technique to evaluate the classifier performance [23].

IV. Experimental Results

In order to evaluate the proposed method, we have used all the 251 images from trial test set of the ICDAR dataset. Although there are many ways to evaluate the performance of text region localization from images (such as the one given by the ICDAR competition), we have chosen to measure the performance of the proposed method in terms of recall and precision rates as they have been used by the other methods [9], [10], [11], [13].

The 251 test images contain 737 text regions (total text regions) with at least 3 characters. Performed tests showed that 627 text regions were correctly classified (true positives) and only 94 were wrongly assigned as text regions (false positives). In this case, the recall and precision rates are 85% and 86.9%, respectively. Table 1 shows the obtained confusion matrix and Fig. 6 presents some images with text regions marked with green rectangles after the application of the decision tree classifier.

Table 1: Confusion matrix.

Classes	True positive	False positive
Text region	$627(85\%)$	$94(12.7\%)$
Non-text region	(87.3%)	$110(15\%)$

Table 2 presents a comparison between the results obtained by our method to the ones obtained by methods proposed in [9], [13] when applied to the same image dataset.

Table 2: Comparison.

Methods	recall	precision
Our method	85%	86.9%
Alves's method [13]	78.4%	86.6%
Wu's method [9]	74.6%	65.3%

[1]`http://www.cs.waikato.ac.nz/ml/weka/`

Just for a simple comparison, Fig. $7(a)$ and Fig. $7(b)$ show the results of an application of Wu's [9] and Alves's [13] methods to a scene image; while Fig. $7(c)$ shows the result using our method.

(a) (b) (c)

Figure 7. Scene images containing text regions with different scales. Results from the application of (a) Wu's method [9]; (b) Alves's method [13] and (c) our method.

V. Conclusion and Future Work

In this work we propose a method for localizing text regions within scene images consisting of two major stages. In the first stage, a set of potential text regions is extracted from the input image using residual operators and it can be

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

Figure 6. Scene images. $(a - f)$ The classified text regions are marked by green rectangles.

briefly described by the following steps: (i) use ultimate attribute openings and closings to identify CCs corresponding to potential characters; (ii) combine some of these CCs in order to obtain a set of text region candidates; (iii) select a subset of text region candidates in order to form a set of potential text regions. In the second stage a set of features is obtained from each potential text region and this feature set will be later used as an input to a decision tree classifier in order to label these regions as text or non-text regions.

In our experiments, the obtained results show a good performance of text region classification with the overall recall and precision rates equal to 85% and 86.9%, respectively. A comparison to other methods (see Table 2) has been shown that our method is superior and consequently it can be a good alternative for text localization in scene images.

For future work, we plan to perform a comparative analysis of our method with others using the metric proposed in [24]. We also envisage to study a new method for segmenting characters within potential text regions using ideas that combine x-projection technique and CCs segmentation.

ACKNOWLEDGMENT

We would like to thank the financial support of CNPq (www.cnpq.br) and FAPESP (www.fapesp.br).

REFERENCES

[1] S. Escalera, X. Bar, J. Vitri, and P. Radeva, "Text detection in urban scenes," *Catalan Association of Artificial Intelligence, 12th International Congress of the*, 2009.

[2] M. Cai, J. Song, and M. R. Lyu, "A new approach for video text detection," in *Image Processing, International Conference on*, 2002, pp. 117–120.

[3] J.-W. Hsieh, S.-H. Yu, and Y.-S. Chen, "Morphology-based license plate detection in images of differently illuminated and oriented cars," *Journal of Electronic Imaging*, vol. 11, no. 4, pp. 507–516, 2002.

[4] D. Menoti, D. L. Borges, J. Facon, and A. de Souza Britto, Jr, "Segmentation of postal envelopes for address block location: an approach based on feature selection in wavelet space," in *ICDAR '03: Proceedings of the Seventh International Conference on Document Analysis and Recognition*. Washington, DC, USA: IEEE Computer Society, 2003, p. 699.

[5] K. Jung, K. I. Kim, and A. K. Jain, "Text information extraction in images and video: a survey," *Pattern Recognition*, vol. 37, no. 5, pp. 977–997, 2004.

[6] S. M. Lucas, A. Panaretos, L. Sosa, A. Tang, S. Wong, and R. Young, "Icdar 2003 robust reading competitions," *Document Analysis and Recognition, Int. Conf. on*, vol. 2, p. 682, 2003.

[7] J. Gllavata, E. Qeli, and B. Freisleben, "Detecting text in videos using fuzzy clustering ensembles," *Multimedia, International Symposium on*, vol. 0, pp. 283–290, 2006.

[8] K. I. Kim, K. Jung, and J. H. Kim, "Texture-based approach for text detection in images using support vector machines and continuously adaptive mean shift algorithm," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 12, pp. 1631–1639, 2003.

[9] J.-C. Wu, J.-W. Hsieh, and Y.-S. Chen, "Morphology-based text line extraction," *Machine Vision Application*, vol. 19, no. 3, pp. 195–207, 2008.

[10] Y. M. Y. Hasan and L. J. Karam, "Morphological text extraction from images," *Image Processing, IEEE Transactions on*, vol. 9, no. 11, pp. 1978–1983, 2000.

[11] T. Retornaz and B. Marcotegui, "Scene text localization based on the ultimate opening," in *International Symposium on Mathematical Morphology, 8 (ISMM)*, vol. 1, 2007, Image processing, pp. 177–188.

[12] L. Gu, "Character region identification from cover images using dtt," in *ICADL*, 2004, pp. 388–397.

[13] W. A. L. Alves and R. F. Hashimoto, "Classification of regions extracted from scene images by morphological filters in text or non-text using decision tree," in *18th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision - WSCG 2010 Proceedings*, Plzen, Czech Republic, february 2010.

[14] R. Jiang, F. Qi, L. Xu, and G. Wu, "Detecting and segmenting text from natural scenes with 2-stage classification," in *ISDA '06: Proceedings of the Sixth International Conference on Intelligent Systems Design and Applications*, Washington, DC, USA, 2006, pp. 819–824.

[15] S. Beucher, "Numerical residues," *Image Vision Comput.*, vol. 25, no. 4, pp. 405–415, 2007.

[16] E. J. Breen and R. Jones, "Attribute openings, thinnings, and granulometries," *Comput. Vis. Image Underst.*, vol. 64, no. 3, pp. 377–389, 1996.

[17] T. Retornaz and B. Marcotegui, "Ultimate opening implementation based on a flooding process," *The 12th International Congress for Stereology*, 2007.

[18] R. C. Gonzalez and R. E. Woods, *Digital Image Processing (3rd Edition)*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 2006.

[19] P. Soille, *Morphological Image Analysis: Principles and Applications*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2003.

[20] N. Otsu, "A threshold selection method from graylevel histograms," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 9, pp. 62–66, 1979.

[21] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, *Numerical recipes in C: the art of scientific computing*. New York, NY, USA: Cambridge University Press, 1988.

[22] J. R. Quinlan, *C4.5: programs for machine learning*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1993.

[23] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, 2nd ed. Wiley-Interscience, 2000.

[24] S. M. Lucas, A. Panaretos, L. Sosa, A. Tang, S. Wong, R. Young, K. Ashida, H. Nagai, M. Okamoto, H. Yamamoto, H. Miyao, J. Zhu, W. Ou, C. Wolf, J. M. Jolion, L. Todoran, M. Worring, and X. Lin, "ICDAR 2003 Robust Reading Competitions: Entries, Results and Future Directions," *International Journal on Document Analysis and Recognition*, vol. 7, no. 2-3, pp. 105–122, 2005.