Ikica and Peer *EURASIP Journal on Advances in Signal Processing* 2013, **2013**:95
http://asp.eurasipjournals.com/content/2013/1/95

EURASIP Journal on Advances in Signal Processing
a SpringerOpen Journal

RESEARCH Open Access

SWT voting-based color reduction for text detection in natural scene images

Andrej Ikica* and Peter Peer

Abstract

In this article, we propose a novel stroke width transform (SWT) voting-based color reduction method for detecting text in natural scene images. Unlike other text detection approaches that mostly rely on either text structure or color, the proposed method combines both by supervising text-oriented color reduction process with additional SWT information. SWT pixels mapped to color space vote in favor of the color they correspond to. Colors receiving high SWT vote most likely belong to text areas and are blocked from being mean-shifted away. Literature does not explicitly address SWT search direction issue; thus, we propose an adaptive sub-block method for determining correct SWT direction. Both SWT voting-based color reduction and SWT direction determination methods are evaluated on binary (text/non-text) images obtained from a challenging Computer Vision Lab optical character recognition database. SWT voting-based color reduction method outperforms the state-of-the-art text-oriented color reduction approach.

Keywords: Text detection, Natural scene images, Stroke width transform, SWT, Color reduction, SWT voting, SWT direction determination, SWT profiles, CVL OCR DB

1 Introduction

Text detection in natural scene images is a very challenging task, far from being completely solved. Complex backgrounds, uneven illumination, and presence of almost unlimited number of text fonts, sizes, and orientations pose great difficulties even to state-of-the-art text detection methods. Unlike document images, where text is usually superimposed on either blank or complex backgrounds and is therefore more distinct [1-3], natural scene images deal with scene text, which is already a part of the captured scene and is often much less distinct. Nevertheless, text detection has become a very popular research area due to its enormous potential in many applicative areas such as sign translation, content-based web image searching, and assisting the visually impaired.

State-of-the-art literature distinguishes between two major text detection approaches: texture-based and region-based. Texture-based methods [4-7] scan images at different scales, inspect area under the sliding window for text-like features, and classify it as text/non-text. They often lack precision and are relatively slow due to their scale-space approach. Region-based methods [8-12], on the other hand, work in a bottom-up fashion by selecting pixels (or regions) with typical text properties and grouping them into connected components that are further geometrically filtered and grouped into text lines and/or words. Region-based methods are not limited to text size/orientation and are (compared to texture-based methods) faster. Flowchart of a typical region-based method is depicted in Figure 1. Besides the aforementioned approaches, hybrid approaches exist, which exploit advantages of both texture-based and region-based approaches [13].

In this article, we propose a stroke width transform (SWT) voting-based color reduction method. It reduces the number of initial colors in the original image to only a few, typically less than 10, while preserving all dominant text colors. SWT voting-based color reduction corresponds to the first two stages of the text detection flowchart (see yellow rectangles in Figure 1). Two spatially connected pixels of a color-reduced image that belong to the same color class correspond to the same connected component as well.

*Correspondence: andrej.ikica@fri.uni-lj.si
Computer Vision Laboratory, Faculty of Computer and Information Science, Ljubljana, Slovenia

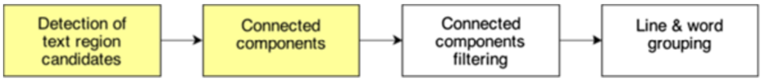
Detection of text region candidates
Connected components
Connected components filtering
Line & word grouping

Figure 1 Flowchart of a typical region-based text detection method. Yellow rectangles correspond to stages covered by our proposed method.

The proposed method improves the state-of-the-art color reduction approach for text detection by Nikolaou and Papamarkos [14] with additional SWT information [8]. Since SWT pixels most likely belong to text regions, they are mapped to color space, where they supervise the color reduction process, more specifically, the mean-shifting stage. When a particular color receives a high SWT vote, it is blocked from being mean-shifted away.

The choice of selected methods is reasonable since both the SWT [8] and Nikolaou color reduction-based text detection methods [9,14] achieve state-of-the-art results. SWT is a very robust method for detecting parallel-like structures such as text strokes. Unfortunately, it often fails to detect whole characters since strokes are not necessarily completely parallel (see Figure 2b). On the other hand, color reduction can successfully segment whole characters. However, text colors close to the background colors are often mean-shifted away (see Figure 2c).

Popular text detection datasets such as the International Conference on Document Analysis and Recognition (ICDAR) 2003 dataset [15] and ICDAR 2011 dataset [16] are inappropriate for evaluating our method since they are annotated with word rectangles. To evaluate the performance of our color reduction method, a per-character evaluation is necessary. Thus, binary ground truth images obtained from Computer Vision Lab Optical Character Recognition DataBase (CVL OCR DB) [17] are used for evaluation. Text and background pixels in ground truth images correspond to non-zero and zero values, respectively.

Text detection literature does not directly address the problem of finding correct SWT search direction. Typically, methods based on SWT execute the SWT method in both gradient and counter-gradient directions and combine the results of both directions. This, however, results in detecting inter-character and non-text areas. Thus, besides SWT voting, our contribution is an adaptive SWT direction determination method that uses SWT profiles to partition an image into subblocks and analyzes their SWT histograms of both SWT search directions.

The rest of the paper is organized as follows. Section 2 describes the proposed method in general. Sections 3 and 4 give a detailed description of both SWT direction determination and SWT voting-based color reduction methods. Experimental results are presented in Section 5. The article is concluded in Section 6.

2 Proposed method

Text in natural scene images is distinguished from other image structures and background by its characteristic shape (character strokes are more or less parallel) and color uniformity. Unlike many other text detection methods that analyze either shape or color, the proposed method combines both by integrating the SWT [8] and Nikolaou text-oriented color reduction [14] methods.

SWT method proposed by Epshtein et al. [8] is a region-based text detection method. It follows the stroke width constancy assumption, which states that stroke widths remain constant throughout individual text characters. After obtaining an edge map of an input image, SWT method locates pairs of parallel edge pixels in the following fashion: for each edge pixel p a search ray in the edge gradient direction is generated, and the first edge pixel q along the search ray is located. If p and q have nearly opposite gradient directions, an edge pair is formed and the distance between p and q (called stroke width) is computed. All pixels lying on the search ray between

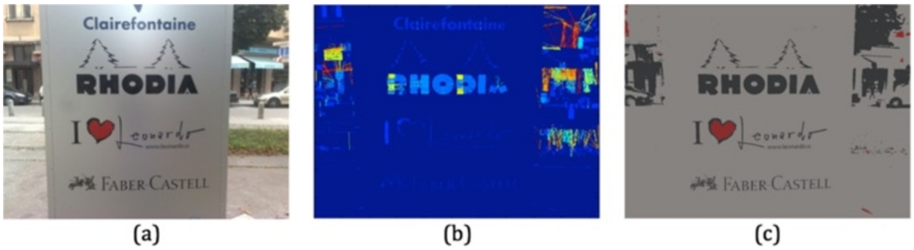

Figure 2 SWT and color reduction example. (**a**) Original image. (**b**) SWT image. SWT fails to detect whole characters. Letter 'A' in 'RHODIA' is not detected completely. (**c**) Color-reduced image. Color reduction fails to find 'Clairefontaine' text, since the blue color is mean-shifted towards gray.

p and q (including p and q) are assigned a corresponding stroke width. After assigning stroke widths to all image pixels, the SWT method groups pixels with similar stroke widths into connected components and filters out those that violate geometrical properties of the text. When the edge threshold is sufficiently low, SWT typically finds all characters in the image or at least small portions of each of them. However, it often fails to detect whole characters and leaves parts of them undetected (see letter 'A' in 'RHODIA' in Figure 2b). Another SWT drawback is the detection of non-text structures with nearly parallel edges.

Text-oriented color reduction method proposed by Nikolaou and Papamarkos [14] and applied in [9] successfully deals with the problem of partially detected characters (see Figure 2c). The idea behind the method is to reduce colors in an image to only a few dominant image colors thus making text detection much easier. The method starts by creating an RGB histogram h_{RGB} of an image. Next, initial color cubes of fixed size are randomly generated inside the h_{RGB} until they completely cover all non-zero h_{RGB} cells. To further reduce the number of colors, the initial color cubes undergo mean-shift stage and are shifted towards dominant gravity centers in h_{RGB}. Additionally, if particular color cubes appear close enough to each other, they are merged together. Centers of the resulting color cubes correspond to the final colors C. Finally, color-reduced image is generated by replacing image colors with their closest match in C. When particular text colors cover only a small portion of an image and are not sufficiently far from other colors in the RGB color space, they are often mean-shifted away – in worst cases, towards background colors. In such scenarios, the color reduction is unable to detect text properly (see missing 'Clairefontaine' text in Figure 2c). Decreasing the cube size could preserve lost text colors, but would also increase the number of final colors, which is unacceptable.

To overcome the problems of partially detected characters and lost text colors, we propose the following strategy (see Figure 3): we expand the Nikolaou text-oriented color reduction method [14] with additional stages, i.e., SWT filter and SWT direction determination, and upgrade the mean-shift stage with SWT voting. After obtaining the SWT image, all pixels with non-zero SWT values are mapped to the RGB color space using SWT lookup table (see Section 4 for details). In order not to mean-shift true text colors, SWT voting is performed at each mean-shift iteration. If the source and target cubes receive high and low numbers of SWT votes, respectively, the color is probably leaving the safe text zone and is blocked from shifting any further. In other cases, mean-shifting is allowed.

The original SWT method [8] searches for parallel edges in edge gradient directions. In case of a dark text on a light background the gradient assumption is true, since search rays always bump into the parallel character strokes. However, in the case of a light text on a dark background, the gradient direction points out of the character, and search rays bump into undesirable structures (as shown in Figure 4e). In order for SWT voting to work correctly, it is important that SWT image corresponds to true text characters. To deal with both dark and light text scenarios, the original SWT implementation runs the whole text detection flowchart twice – in gradient and counter-gradient directions – and merges the results of both directions. Besides being questionable, such approach is unacceptable to us since SWT voting demands correct SWT directions. Thus, we propose a SWT direction determination method, which provides correct SWT image to SWT voting stage.

We implemented the Nikolaou color reduction method with modifications. The initial cubes are not selected randomly as in [14] but in the following manner: RGB histogram bins are sorted in descending order and initial cube centers are always selected from the top of the unvisited bins list. This way, we guarantee that algorithm works in deterministic fashion. HSL color model is used for generating color-reduced images from the final color clusters. We use HSL distance metrics defined in [18].

3 SWT direction determination

3.1 SWT profile-based sub-block partition and selection

The left column in Figure 4 depicts the original image (a), its gradient SWT image (c), and counter-gradient SWT image (e). We'll refer to them as SWT^{+} and

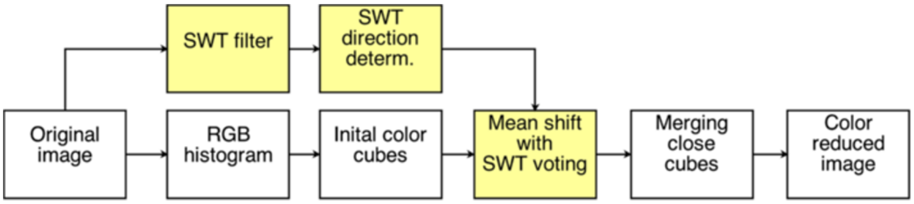

Figure 3 Flowchart of the proposed method. Yellow rectangles denote our contribution.

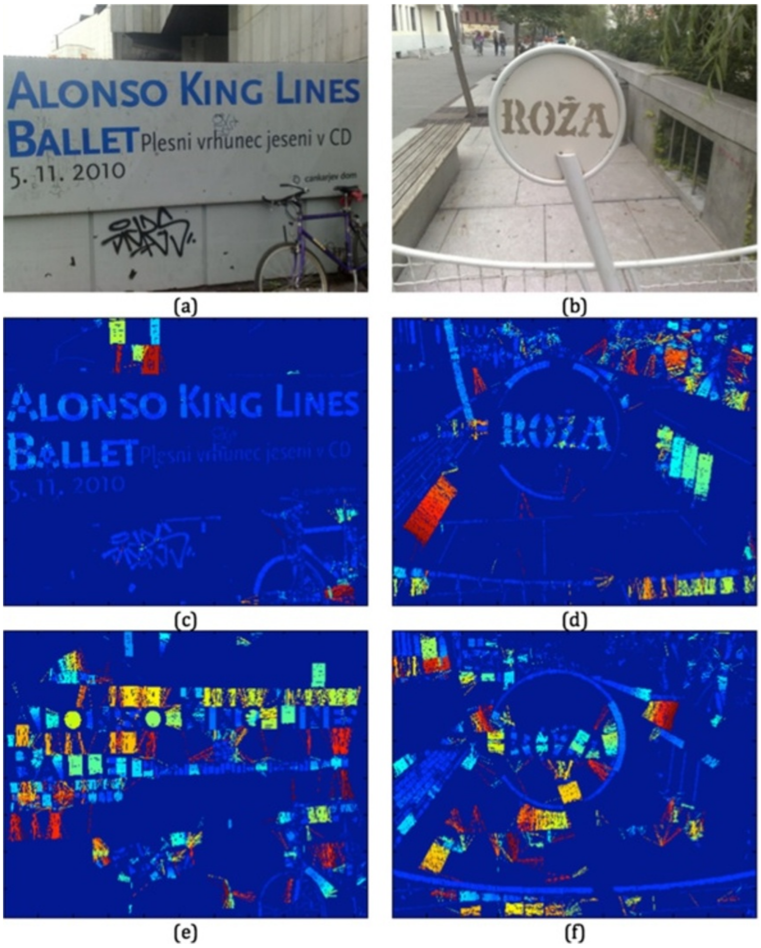

Figure 4 SWT direction examples. (**a**) Original image and corresponding SWT$^+$ (**c**) and SWT$^-$ (**e**) images. (**b**) Another image example with corresponding SWT$^+$ (**d**) and SWT$^-$ (**f**) images.

SWT$^-$, respectively. Each color in the SWT image corresponds to a particular stroke width; therefore, pixels sharing the same stroke widths are represented with the same color. If we carefully observe both SWT images, we can see that SWT$^+$ is more compact and contains less colors compared to SWT$^-$. This is reasonable since SWT$^+$ corresponds to the actual text with uniform stroke widths. On the other hand, SWT$^-$ corresponds to non-text areas with randomly distributed 'strokes'. However, if we look at the right column in Figure 4, the distinction between SWT$^+$ and SWT$^-$ is not so clear anymore. But still, the given assumption holds for the central region with text 'ROŽA'.

Motivated by the observations above, we propose a method to determine the correct SWT direction. We compute SWT$^+$ and SWT$^-$ of an input image I and obtain a merged SWT* image by superimposing SWT$^-$ on top of SWT$^+$. Since SWT$^+$ and SWT$^-$ are disjoint (non-zero pixels of SWT$^+$ do not overlap with non-zero pixels of SWT$^-$, and vice versa), the superimposition is performed by taking SWT$^+$ as foundation and placing SWT$^-$ pixels at corresponding locations in SWT$^+$. Thus, the SWT* image can be thought of as the union of SWT$^+$ and SWT$^-$. An example of a SWT* image is depicted in Figure 5a. Afterwards, we split SWT* image into sub-blocks. Similar to [19] and [20], where edge profiles are used to find text peaks in images, we use custom profiles to split the image. Since we count occurrences of SWT pixels instead of edges, we'll refer to them as SWT profiles. The SWT* image is first split vertically by identifying peak regions of its horizontal SWT profile (see Figure 5b). Each vertical block (Figure 5c) is further split horizontally into sub-blocks by identifying peak regions of the corresponding vertical SWT profile (Figure 5d).

After partitioning SWT* image into sub-blocks, each sub-block is independently analyzed for correct SWT direction. For a particular sub-block (SB), a sub-block

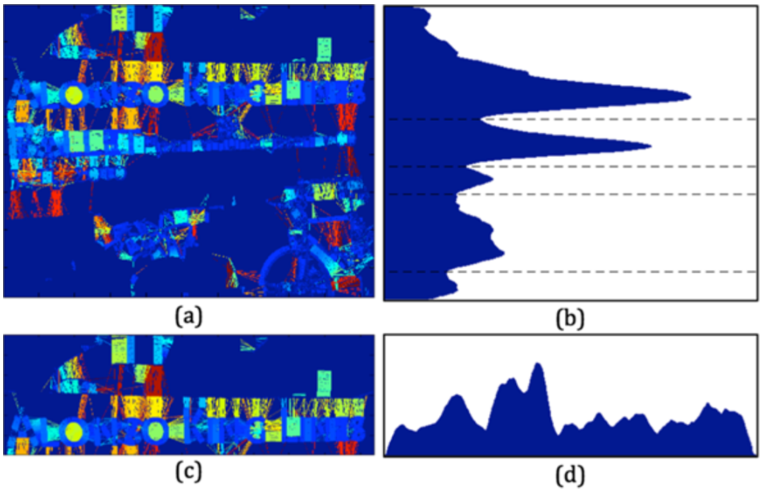

Figure 5 SWT profiles. (**a**) SWT* image merged from SWT$^+$ and SWT$^-$ in Figure 4c, e, respectively. (**b**) Horizontal SWT profile of (a) and corresponding vertical blocks marked with dashed lines. (**c**) The top vertical block of (a). (**d**) Vertical SWT profile of (c).

$\mathrm{SWT}^+_{\mathrm{SB}}$ with the same coordinates as SB is extracted from SWT^+ image and a sub-block $\mathrm{SWT}^-_{\mathrm{SB}}$ with the same coordinates as SB is extracted from SWT^- image. Examples of $\mathrm{SWT}^+_{\mathrm{SB}}$ and $\mathrm{SWT}^-_{\mathrm{SB}}$ sub-blocks are depicted in Figure 6a,b, respectively. For both sub-blocks, SWT histograms h^+ and h^- with N_b number of bins are generated:

$$\begin{aligned} h^+(i) &= \#\{p \in \mathrm{SWT}^+_{\mathrm{SB}} \mid \mathrm{SWT}^+_{\mathrm{SB}}(p) > 0 \wedge \\ &\quad \times (i-1)\frac{max_{\mathrm{SWT}}}{N_b} < \mathrm{SWT}^+_{\mathrm{SB}}(p) \leq i\frac{max_{\mathrm{SWT}}}{N_b}\} \\ h^-(i) &= \#\{p \in \mathrm{SWT}^-_{\mathrm{SB}} \mid \mathrm{SWT}^-_{\mathrm{SB}}(p) > 0 \wedge \\ &\quad \times (i-1)\frac{max_{\mathrm{SWT}}}{N_b} < \mathrm{SWT}^-_{\mathrm{SB}}(p) \leq i\frac{max_{\mathrm{SWT}}}{N_b}\}, \end{aligned} \tag{1}$$

where i is the bin index, p is the image pixel, and max_{SWT} is the maximum SWT value of both $\mathrm{SWT}^+_{\mathrm{SB}}$ and $\mathrm{SWT}^-_{\mathrm{SB}}$ sub-blocks. Simply put, each SWT histogram bin contains a number of pixels with stroke widths in a given bin range. After obtaining SWT histograms, both histograms are sorted in ascending order. Examples of sorted SWT histograms are depicted in Figure 6.

While examining SWT histograms of several SWT images, we came across some interesting observations. SWT histograms corresponding to true text are usually steeper, more compact, and edgier as opposed to the non-text histograms, which are typically wider and ascend in a more continuous fashion (see Figure 6). Our empirical observations seem to be reasonable since the text usually contains equal stroke widths. On the other hand, non-text regions contain more SWT noise; therefore, stroke widths are more evenly distributed over the whole spectrum.

To favor the histograms that correspond to true text regions, we propose an f measure, which is an average bin difference between adjacent non-zero bins:

$$f(h) = \frac{1}{N_{nz}} \sum_{i=2}^{N_b} (h(i) - h(i-1)), \tag{2}$$

where h is a SWT histogram, and N_{nz} is number of non-zero bins. The f measure favors edgier histograms. Averaging with the number of non-zero bins is crucial since it favors narrower histograms. The SWT sub-block with higher f value is chosen as the correct sub-block:

$$\mathrm{SWT}_{\mathrm{SB}} = (\arg\max_{swt} f(h(swt)) \,|\, swt \in \{\mathrm{SWT}^+_{\mathrm{SB}}, \mathrm{SWT}^-_{\mathrm{SB}}\}). \tag{3}$$

After obtaining all sub-blocks $\mathrm{SWT}_{\mathrm{SB}}$ they are glued together into the final $\mathrm{SWT}_{\mathrm{RES}}$ image of the same size as the input image *I*.

3.2 Upper SWT boundary

When searching in non-text direction, the SWT method often bumps into straight structures such as sign borders, walls, etc. This anomalous behavior often produces solid SWT regions with constant stroke widths that receive very high f measures, typically higher than those of text regions. Figure 7 depicts such scenario. Figure 7a,b represents $\mathrm{SWT}^+_{\mathrm{SB}}$ and $\mathrm{SWT}^-_{\mathrm{SB}}$ sub-blocks, respectively. The first sub-block receives much higher f measure than the

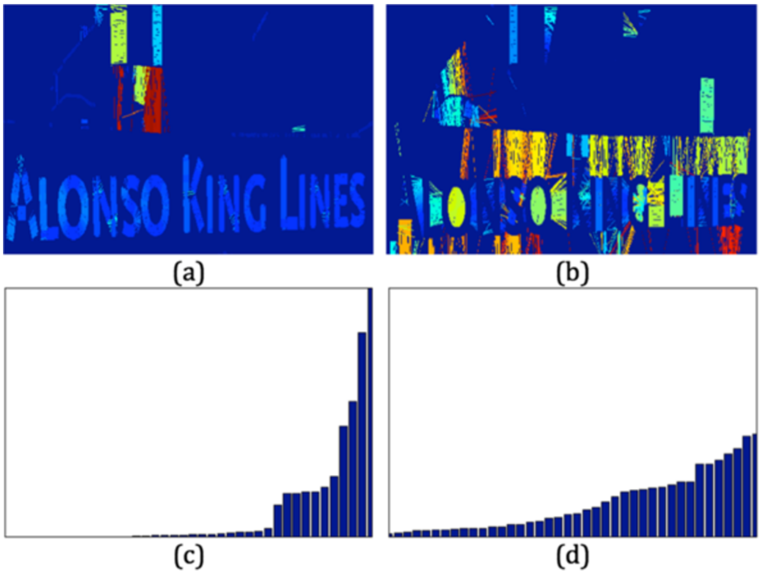

Figure 6 SWT sub-blocks and corresponding histograms. (**a**) $\mathrm{SWT}^{+}_{\mathrm{SB}}$ sub-block corresponding to the sub-block in Figure 5c. (**b**) $\mathrm{SWT}^{-}_{\mathrm{SB}}$ sub-block corresponding to the sub-block in Figure 5c. (**c**) SWT histogram of (a) with measure $f = 425.7$. (**d**) SWT histogram of (b) with measure $f = 111.5$.

second due to the compact non-text SWT region marked with yellow.

To avoid this issue, we define the upper SWT boundary ub_{SWT} and set SWT pixels with SWT values higher than ub_{SWT} to zero. Upper SWT boundary is defined as weighted average of mean SWT values of both SWT sub-blocks:

$$ub_{\mathrm{SWT}} = \alpha \cdot \min(\mu(\mathrm{SWT}^{+}_{\mathrm{SB}}), \mu(\mathrm{SWT}^{-}_{\mathrm{SB}})) + (1-\alpha) \cdot \max(\mu(\mathrm{SWT}^{+}_{\mathrm{SB}}), \mu(\mathrm{SWT}^{-}_{\mathrm{SB}})), \tag{4}$$

where $\mu(\cdot)$ corresponds to the matrix mean and $0 < \alpha < 1$. Since the distance between parallel edges of the character is usually shorter than the distance between the character and other structures in an image (such as signboards), Equation 4 serves for limiting the influence of the longer (usually non-text) distance. Note that longer distance does not always correspond to non-text area; therefore, correct SWT direction typically cannot be determined merely on the length observation basis. Nevertheless, the upper SWT boundary is a very useful addition to the SWT sub-block histogram analysis.

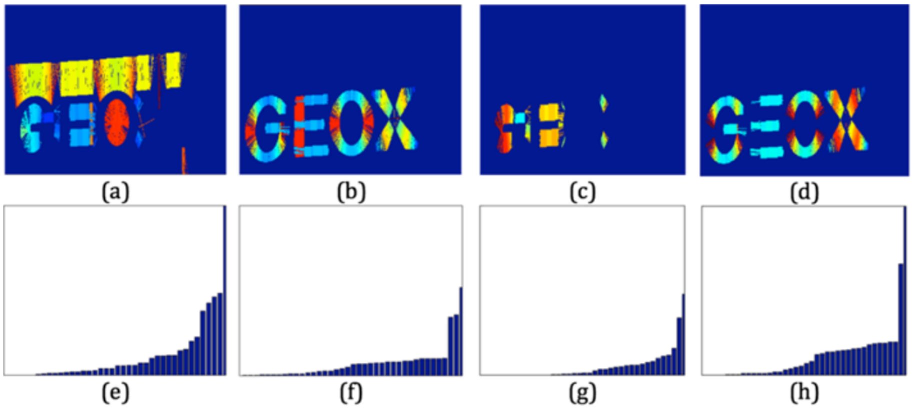

Figure 7 Examples of SWT sub-blocks with and without upper SWT boundary. Examples of (**a**) $\mathrm{SWT}^{+}_{\mathrm{SB}}$ and (**b**) $\mathrm{SWT}^{+}_{\mathrm{SB}}$ sub-blocks without upper SWT boundary and (**c**) $\mathrm{SWT}^{+}_{\mathrm{SB}}$ and (**d**) $\mathrm{SWT}^{-}_{\mathrm{SB}}$ sub-blocks with upper SWT boundary, where $\alpha = 0.33$. Corresponding SWT histograms are depicted in the bottom row: (**e**) $f = 263.2$, (**f**) $f = 119.1$, (**g**) $f = 85.9$, (**h**) $f = 133.1$.

Filtering SWT sub-blocks with upper SWT boundary before generating SWT histograms and computing the f measure successfully reduces the number of SWT outliers as shown in Figure 7c,d. In the sections that follow, we denote the final $\mathrm{SWT}_{\mathrm{RES}}$ image simply as SWT image.

4 SWT voting-based color reduction

The Nikolaou text-oriented color reduction method proposed in [14] works very well when image colors appear far from each other in the RGB histogram as shown in Figure 8a,b. Colors in natural scene images, however, usually appear much closer to each other (Figure 8c,d) so color reduction often fails to find non-dominant text colors, since they are mean-shifted towards the more dominant ones. Figure 9 depicts an example of such color reduction behavior. Initial color clusters, their corresponding colors, and initial color-reduced image of original image in Figure 8c are shown in Figure 9a,b,c, respectively. Since the blue 'Clairefontaine' color appears too close to the background's gray color, it is mean-shifted away as shown in Figures 9d,e,f.

4.1 SWT lookup table

To block probable text colors from being mean-shifted away, a correspondence between text shape and image colors must be established. A key to plausible integration of both lies in the SWT information, since image pixels corresponding to non-zero values in the SWT image most likely belong to text regions. Therefore, before executing a mean-shift stage, non-zero SWT pixels are mapped to the color space using SWT lookup table.

SWT lookup table is a three-dimensional table of the same size as the RGB histogram. Each table cell corresponds to a particular RGB triplet and contains a list of non-zero SWT values of all RGB occurrences in the original image. For instance, the SWT lookup table entry for RGB triplet (120,50,70) is generated by locating all pixels with $R = 120$, $G = 50$ and $B = 70$ in the original image and storing SWT values at corresponding locations in the SWT image. If a particular color is a text color, its SWT lookup cell contains more or less similar SWT values. On the other hand, non-text colors mostly contain very few SWT values (non-text regions mostly correspond to zero SWT values), which are randomly distributed.

The SWT lookup table represents an efficient mapping from stroke width space to color space and allows us to quickly obtain SWT information for a particular color.

4.2 Mean-shift with SWT voting

To block text colors from being mean-shifted away, we propose the following solution: before each mean-shift step, SWT properties of source and target cubes are compared (source and target cube correspond to color cube before and after particular mean-shift step, respectively). When a source cube rich with SWT pixels

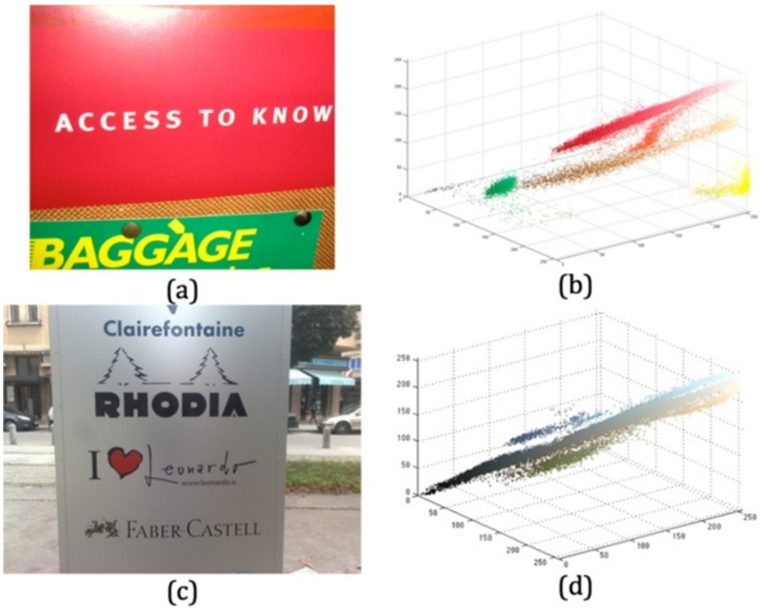

Figure 8 RGB representations of different images. (**a**) An image with five dominant colors. (**b**) RGB representation of (a). (**c**) A typical natural scene image. (**d**) RGB representation of (c).

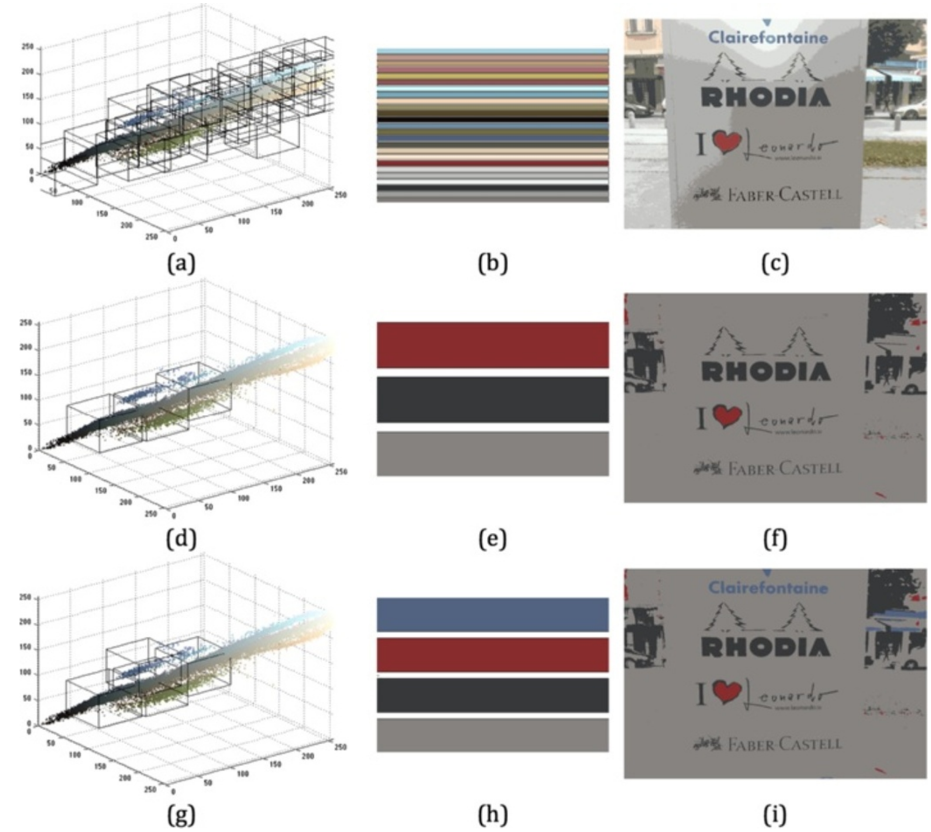

Figure 9 Initial and final color clusters. (**a**) Initial color clusters obtained from image in Figure 8c. (**b**) Initial colors corresponding to (a). (**c**) Initial color-reduced image corresponding to (a). (**d**) Final color clusters obtained by the Nikolaou color reduction method [14]. (**e**) Final colors corresponding to (d). (**f**) Final color-reduced image corresponding to (d). (**g**) Final color clusters obtained by SWT voting-based color reduction. (**h**) Final colors corresponding to (g). (**i**) Final color-reduced image corresponding to (g).

is about to be mean-shifted towards a target cube that is drastically poorer with SWT pixels, mean-shifting of a current color is blocked since the probable text/non-text transition has occurred. We call this process SWT voting since SWT pixels cast a vote to a color they are assigned to and determine whether it is a text or non-text color.

Let CB denote a color cube with center C_{CB} and edge length L_{CB}, and let LT_{SWT} denote a SWT lookup table. Before each mean-shift iteration, a smaller concentric SWT cube CB_{SWT} with edge length $L_{\mathrm{SWT}} = \beta \cdot L_{CB} (0 < \beta \leq 1)$ is generated inside the color cube. The following properties of the SWT cube are computed:

- **SWT density** D_{SWT}. SWT density is a ratio between number of non-zero SWT pixels and number of RGB triplets covered by SWT cube area:

$$D_{\mathrm{SWT}} = \frac{\#(LT_{\mathrm{SWT}}(CB_{\mathrm{SWT}}))}{\#(h_{\mathrm{RGB}}(CB_{\mathrm{SWT}}))}, \tag{5}$$

  where h_{RGB} is the RGB histogram of an original image.
- **Standard deviation of SWT lengths** SD_{SWTL}. SD_{SWTL} measures stroke width variance of SWT pixels covered by the SWT cube. Lower deviation indicates that SWT cube covers pixels of uniform stroke widths and therefore corresponds to a text color.
- **Standard deviation of SWT offsets** SD_{SWTO}. SD_{SWTO} indicates how scattered the SWT pixels are with respect to SWT cube's origin.

Let CB_{SWT_1} and CB_{SWT_2} denote source and target SWT cubes, respectively. If the condition in Equation 6 is true, mean-shifting is stopped, and the final mean-shift location is set to CB_{SWT_1}:

$$\begin{aligned} &D_{\mathrm{SWT}_2} < D_{\mathrm{SWT}_1} \cdot \tau_D \\ &D_{\mathrm{SWT}_1} \geq D_{min} \\ &SD_{\mathrm{SWTL}_2} > SD_{\mathrm{SWTL}_1} \cdot \tau_L \\ &SD_{\mathrm{SWTO}_2} < SD_{\mathrm{SWTO}_1} \cdot \tau_O \end{aligned} \tag{6}$$

Let us explain this condition. When a significant drop in SWT density is detected (first row in Equation 6), the cube is probably undergoing a text/non-text transition; however, some additional checks need to be performed: first, the SWT density of a source cube must be relatively high. Otherwise, density drop can be a result of SWT noise present in low SWT density cubes (second row in Equation 6). Second, we empirically found out that SWT length deviation typically rises when text/non-text transitions occur (third row in Equation 6). Third, mean-shifting from text to background color is gradual, wherein transition is typically made of more than one mean-shift steps. To assure that mean-shifting is blocked in the first step and not in the intermediate steps, the fourth row in Equation 6 must be true. SWT cubes corresponding to dominant text color tints usually have SWT pixels spread all over the cube. When mean-shifting towards background colors, SWT pixels slowly vanish and appear only at SWT cube borders thus lowering SWT offset deviation. Due to the SWT and color noise, all four conditions in Equation 6 must be true. Otherwise, mean-shifting proceeds normally. Parameters τ_D, τ_L, and τ_O in Equation 6 determine sensitivity of the SWT voting. By lowering/raising them, the mean-shift stopping condition becomes more/less strict thus affecting how many color cubes are blocked from mean shifting any further.

Two mean-shift scenarios are shown in Figure 10. Scenario (a) depicts mean-shifting of gray background color. SWT density is very small, and there are no significant changes in standard deviations, so mean-shifting proceeds. On the other hand, scenario (b) depicts mean-shifting of the blue color (which corresponds to the ‘Clairefontaine’ text in Figure 8c). A significant drop in SWT density is detected; besides, standard deviations change considerably, so mean-shifting is blocked. Figure 9i and corresponding Figure 9g,h depict final SWT voting-based color reduction results on the image in Figure 8c. Unlike the Nikolaou color reduction method [14], the proposed method preserves the blue ‘Clairefontaine’ text.

5 Experimental results

Since the popular text detection datasets such as ICDAR [15,16] and Street View Text (SVT) [21] are annotated with word rectangles they are inappropriate for evaluating our color reduction method, which covers the first two stages of the text detection flowchart (Figure 1). Therefore, we created a dataset of binary text/non-text ground truth images obtained by manually binarizing the first 60 images of normal category from challenging CVL OCR DB text detection dataset [17]. We will refer to binary dataset as CVL OCR BIN DB. Both CVL OCR DB and CVL OCR BIN DB datasets are available for download [22]. Binarizing natural scene text images is a very difficult and time-consuming task, so we had to make a compromise between the reasonable binarizing effort and the size of the evaluation dataset. Thus, we chose a representative dataset of 60 images. Since our goal is to eventually binarize all CVL OCR DB images, we decided to

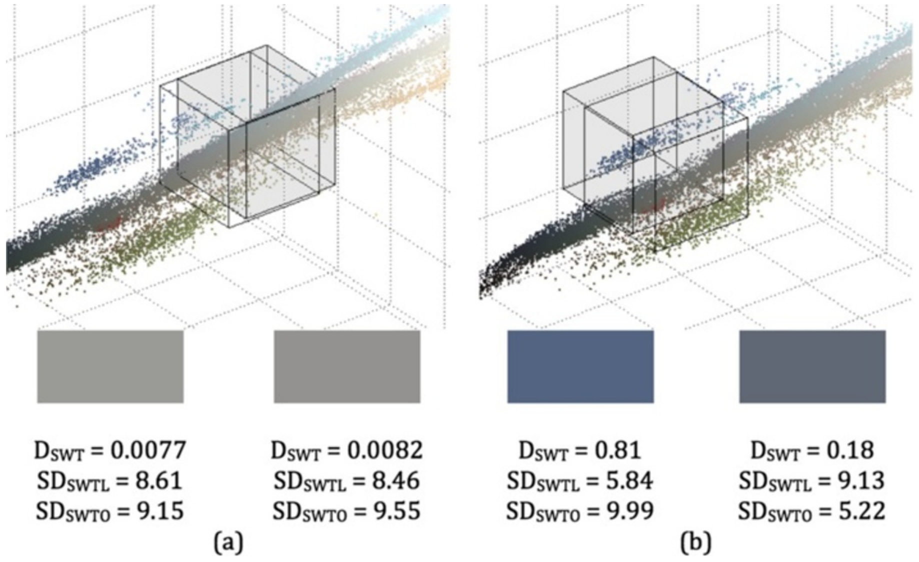

Figure 10 Mean-shift scenarios. (**a**) Mean shifting from gray (bottom left) to darker gray (bottom right) is allowed. (**b**) Mean shifting from blue (bottom left) to darker blue (bottom right) is blocked.

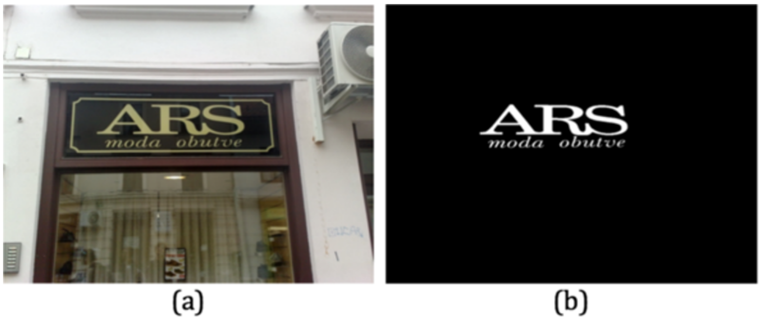

Figure 11 An example of CVL OCR BIN DB ground truth data. (**a**) Original image. (**b**) Binary text/non-text image.

approach binarization systematically and started to binarize from the beginning of the dataset. Figure 11 depicts an example of binary text/non-text ground truth data. We evaluated both SWT direction determination and SWT voting-based color reduction methods on CVL OCR BIN DB.

All experiments were carried out using the following empirically obtained parameter values: $N_{\mathrm{B}} = 40$, $\alpha = 0.33$, $\beta = 0.63$, $D_{min} = 0.18$, $\tau_{\mathrm{D}} = 0.70$,$\tau_{\mathrm{L}} = 0.80$, $\tau_{\mathrm{O}} = 0.80$. Other color reduction parameters were identical as in [14].

5.1 SWT direction determination

We evaluated both SWT direction determination methods (basic method and method with upper SWT boundary). Since both methods adaptively partition image into sub-blocks, we manually inspected each sub-block and checked whether true SWT direction was determined. Evaluation results are shown in Table 1.

It is important to stress out why the number of all the text blocks is different for the basic and the advanced methods. The upper SWT boundary shortens the SWT lengths resulting in sparser SWT images compared to SWT images obtained with the basic method. Therefore, sub-block image partitions are different. Examples of SWT direction determination are shown in Figure 12.

Relatively high detection rate in Table 1 indicates that the SWT direction determination method (in combination with upper SWT boundary) works well and fails to determine the correct SWT direction only in a few cases. Since these cases typically correspond to smaller parts of the scene text (such as words, word segments, or even single characters), they do not critically affect the succeeding SWT voting process. Enough color information is still available in the remaining (correctly determined) parts of the text.

5.2 SWT voting-based color reduction

State-of-the-art text-oriented color reduction method proposed by Nikolaou and Papamarkos [14] and the SWT voting-based color reduction methods were evaluated in the same way as in [14]. Each connected component CC_{GT} in particular ground truth image is compared to its best-detected match CC_{DET} in the color-reduced image. If two neighboring pixels in the color-reduced image share the same color, they belong to the same connected component. When the size ratio between CC_{DET} and CC_{GT}, as well as the size ratio between intersection area $CC_{\mathrm{DET}} \cap CC_{\mathrm{GT}}$ and the size of CC_{GT}, are both larger than a threshold T_{R}, a detection match is registered [14]. Evaluation results are shown in Tables 2 and 3. CC detection rate corresponds to a ratio between the number of correctly detected connected components and

Table 1 SWT direction determination results

	Number of text blocks with correct SWT directions	Number of all text blocks	Detection rate
Basic SWT direction method	80	100	80.00%
SWT direction method with upper SWT boundary	130	142	91.55%

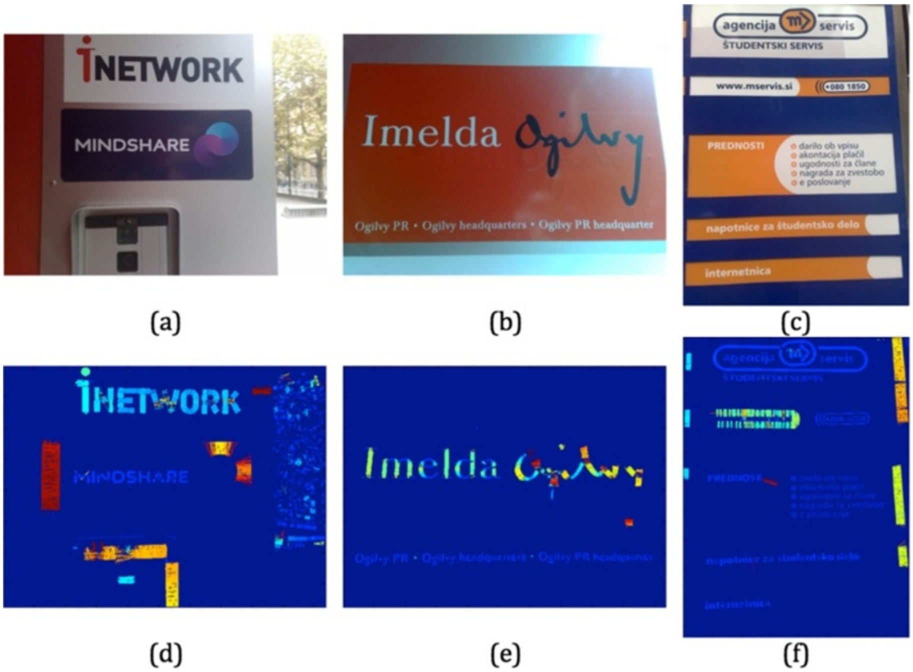

Figure 12 SWT direction determination examples. (**a**, **b**, **c**) Original images. (**d**, **e**, **f**) SWT directions obtained using the upper SWT boundary. Note how the method correctly determines the SWT direction of images (a) and (b) even when both dark text on light background and light text on dark background appear simultaneously. The method is unable to detect the 'www.mservis.si' part of image (c) since it lies inside the rectangular area with SWT rays bumping into it thus creating a character-like structures.

the number of all connected components in all images. Mean detection rate is an average CC detection rate per image [14].

Figure 13 depicts some color reduction results. The first column of the figure corresponds to the original images, while the second and the third columns correspond to the Nikolaou and SWT voting-based color reduction results, respectively. In the first three rows of Figure 13, the SWT voting-based method outperforms the Nikolaou color reduction method. The fourth row demonstrates how the SWT voting-based method preserves the red color in letter 'E' as opposed to the Nikolaou method. The fifth row depicts a case where both methods are unable to correctly segment the text due to problematic reflection in the left half of the image. But still, SWT voting is able to detect at least the first character 'C' in the 'Caffé'.

Results in Tables 2 and 3 indicate that SWT voting outperforms the state-of-the-art Nikolaou text-oriented color reduction method [14]. SWT regions, most likely belonging to text regions, contain valuable text color information, which is used to correctly supervise the mean-shifting in favor of text colors.

6 Conclusions

We presented a novel SWT voting-based color reduction method. First, an adaptive sub-block SWT direction determination method is described. By splitting an image into sub-blocks and analyzing corresponding

Table 2 CC detection rate

	Threshold T_R				
	0.5	0.6	0.7	0.8	0.9
Nikolaou color reduction [14]	69.02%	65.75%	61.86%	56.95%	48.53%
SWT voting-based color reduction	71.01%	67.48%	63.93%	59.84%	50.41%

Table 3 Mean detection rate

	Threshold T_R				
	0.5	0.6	0.7	0.8	0.9
Nikolaou color reduction [14]	77.71%	74.40%	71.06%	66.02%	57.40%
SWT voting-based color reduction	80.24%	76.72%	73.77%	69.12%	59.27%

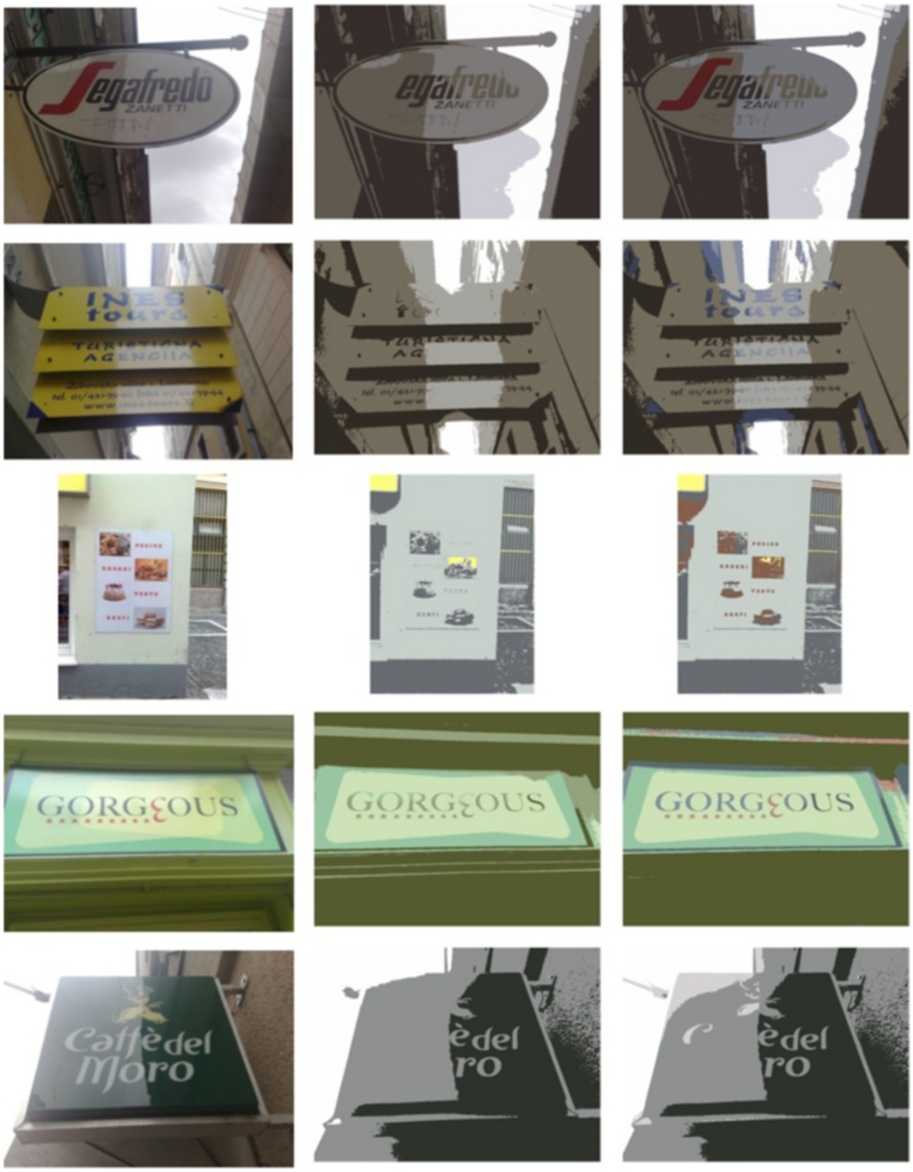

Figure 13 Color reduction examples. First column depicts original images while second and third columns correspond to the Nikolaou and SWT voting-based color reduction results.

SWT histograms of gradient and counter-gradient directions, the method is able to achieve 91% detection rate. Second, a SWT voting approach for color reduction is proposed. Colors rich with SWT pixels most likely belong to text characters and are therefore blocked from being mean-shifted away. Besides improving the state-of-the-art Nikolaou text-oriented color reduction approach, SWT information can be successfully applied in the connected component filtering stage. Only connected components with the SWT cover ratio higher than a predefined threshold are treated as text. All others are filtered out since they are not covered by enough SWT pixels and most probably do not correspond to text. SWT voting-based color reduction achieves up to 80% mean detection rate and up to 71% CC detection rate.

Thresholds in the SWT voting condition determine sensitivity of the text/non-text transition detection in the mean-shift stage. If the thresholds are relaxed, even more text colors can be preserved, but in this case text characters are often split into several text colors. Our future work will therefore focus on merging text colors that belong to the same text character.

Competing interests

Both authors declare that they have no competing interests.

Acknowledgements

This research is supported by the Public Agency for Technology of the Republic of Slovenia (TIA) – operation partly financed by the European Union, European Social Fund.

Received: 8 February 2013 Accepted: 15 April 2013
Published: 1 May 2013

References

1. YL Chen, BF Wu, A multi-plane approach for text segmentation of complex document images. Pattern Recognition. **42**(7), 1419–1444 (2009)
2. K Zagoris, SA Chatzichristofis, N Papamarkos, Text localization using standard deviation analysis of structure elements and support vector machines. EURASIP J. Adv. Signal Process. **2011**(47), 1-12 (2011)
3. S Kumar, R Gupta, N Khanna, S Chaudhury, SD Joshi, Text Extraction and Document Image Segmentation Using Matched Wavelets and MRF Model. IEEE Trans. Image Process. **16**(8), 2117–2128 (2007)
4. Q Ye, Q Huang, W Gao, D Zhao, Fast and robust text detection in images and video frames. Image and Vision Comput. **23**(6), 565–576 (2005)
5. X Chen, AL Yuille, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Detecting and reading text in natural scenes (IEEE, Washington, 2004), pp. 366–373
6. X Li, W Wang, S Jiang, Q Huang, W Gao, in *IEEE International Conference on Image Processing*. Fast and effective text detection (IEEE, Washington, 2008), pp. 969–972
7. J Lee, P Lee, S Lee, A Yuille, C Koch, in *International Conference on Document Analysis and Recognition*. AdaBoost for text detection in natural scene, Beijing, 18-21 September 2011 (IEEE, Washington, 2011), pp. 429–434
8. B Epshtein, E Ofek, Y Wexler, in *IEEE Conference on Computer Vision and Pattern Recognition*. Detecting text in natural scenes with stroke width transform, San Francisco, CA , 13-18 June 2010 (IEEE, Washington, 2010), pp. 2963–2970
9. C Yi, Y Tian, Text String Detection From Natural Scenes by Structure-Based Partition and Grouping. IEEE Trans. on Image Process. **20**(9), 2594–2605 (2011)
10. H Chen, SS Tsai, G Schroth, DM Chen, R Grzeszczuk, B Girod, in *IEEE International Conference on Image Processing*. Robust text detection in natural images with edge-enhanced maximally stable extremal regions, Brussels, 11-14 September 2011 (IEEE, Washington, 2011), pp. 2609–2612
11. TD Nguyen, J Park, G Lee, Tensor voting based text localization in natural scene images. IEEE Signal Process. Lett. **17**(7), 639–642 (2010)
12. L Neumann, J Matas, in *IEEE Conference on Computer Vision and Pattern Recognition*. Real-time scene text localization and recognition, Providence, RI , 16-21 June 2012 (IEEE, Washington, 2012), pp. 3538–3545
13. YF Pan, X Hou, CL Liu, A Hybrid Approach to Detect and Localize Texts in Natural Scene Images. IEEE Trans. Image Process. **20**(3), 800–813 (2011)
14. N Nikolaou, N Papamarkos, Color reduction for complex document images. Int. J. Imaging Syst. Technol. **19**(1), 14–26 (2009)
15. SM Lucas, A Panaretos, L Sosa, A Tang, S Wong, R Young, in *Proceedings of the Seventh International Conference on Document Analysis and Recognition*. ICDAR 2003 robust reading competitions, 3-6 August 2003 (IEEE, Washington, 2003), pp. 682–687
16. A Shahab, F Shafait, A Dengel, Beijing ICDAR 2011 robust reading competition challenge 2: reading text in scene images 18-21 September 2011, in *International Conference on Document Analysis and Recognition* (IEEE, Washington, 2011), pp. 1491–1496
17. A Ikica, P Peer, CVL OCR DB, an annotated image database of text in natural scenes, and its usability. Informacije MIDEM. **41**(2), 150–154 (2011)
18. RB Fisher, Change Detection in Color Images. [http://homepages.inf.ed.ac.uk/rbf/PAPERS/iccv99.pdf]
19. J Park, G Lee, E Kim, J Lim, SH Kim, HJ Yang, M Lee, S Hwang, Automatic detection and recognition of Korean text in outdoor signboard images. Pattern RecognitLett. **31**(12), 1728–1739 (2010)
20. A Ikica, P Peer, in *EUROCON*. An improved edge profile based method for text detection in images of natural scenes, Lisbon, 27-29 April 2011 (IEEE, Washington, 2011), pp. 1–4
21. K Wang, B Babenko, S Belongie, in *International Conference on Computer Vision*, vol. 2011. End-to-end scene text recognition, Barcelona, 6-13 November (IEEE, Washington, 2011), pp. 1457–1464
22. CVL OCR DB and CVL OCR BIN DB download. [http://lrv.fri.uni-lj.si/andreji/CVLOCRDB/]

doi:10.1186/1687-6180-2013-95
Cite this article as: Ikica and Peer: **SWT voting-based color reduction for text detection in natural scene images.** *EURASIP Journal on Advances in Signal Processing* 2013 **2013**:95.