Available online at www.sciencedirect.com
ScienceDirect
Image and Vision Computing 25 (2007) 671–685
www.elsevier.com/locate/imavis

Text segmentation in color images using tensor voting

Jaeguyn Lim *, Jonghyun Park *, Gérard G. Medioni

Institute for Robotics and Intelligent Systems, University of Southern California, Los Angeles, CA 90089-0273, USA

Received 18 June 2005; received in revised form 18 March 2006; accepted 16 May 2006

Abstract

In natural scene, text elements are corrupted by many types of noise, such as streaks, highlights, or cracks. These effects make the clean and automatic segmentation very difficult and can reduce the accuracy of further analysis such as optical character recognition. We propose a method to drastically improve segmentation using tensor voting as the main filtering step. We first decompose an image into chromatic and achromatic regions. We then identify text layers using tensor voting, and remove noise using adaptive median filter iteratively. Finally, density estimation for center modes detection and K-means clustering algorithm is performed later for segmentation of values according to hue or intensity component in the improved image. Excellent results are achieved in experiments on real images.

Keywords: Tensor voting; Text segmentation; Mean shift-based density estimation; Adaptive median filter; Color component analysis

1. Introduction

Diverse approaches for common image segmentation have been investigated for a long time. Some segmentation algorithms only deal with gray scale images [1]. Other algorithms perform segmentation of color images in the RGB color space [2]. The segmentation is sensitive to illumination, so results are somewhat poor. Image segmentation in the HSI color space, as proposed by C. Zhang and P. Wang, produces better results [3]. HSI space is therefore preferred in natural scenes easily influenced by illuminations [4–6]. Moreover, the decision about achromatic region in the given image is improved with our observation.

Natural scenes have diverse objects and, among them, characters are important objects since they convey important meanings for image understanding. The fact has inspired many efforts on text recognition in static images, as well as video sequences [7–12]. In [13,14], Yang et al. develop a machine translation system which automatically detects and recognizes texts in natural scenes. In [15], Ye et al. use Gaussian mixture models (GMMs) in HSI color space with spatial connectivity information to segment characters from a complex background. They do not explicitly take into account the fact that characters in natural scenes can be severely corrupted by noise. In such cases, characters may not be segmented as separate objects due to the corruption of strokes which may cause errors when used as input in optical character recognition (OCR), as mentioned in the future work in [16]. Therefore, we propose a newly improved approach to text segmentation for recognition tasks, which is the step before OCR.

Fig. 1 shows the proposed location of our text segmentation module, which is enclosed by the dashed-line box, in a text recognition application. Assuming the text regions in a natural scene are detected by text features in [17] or edge sets in [18], we can use the regions as input data for text segmentation in HSI space. These regions, however, may be corrupted with considerable noise which must be removed for successful text segmentation. Noise makes conventional text segmentation approaches [13–16] difficult to do segment text as distinct objects exactly. Therefore, we propose to use the tensor voting framework for detection and removal of noise, which results in accurate segmentation.

* Corresponding authors. Tel./fax: +82 31 280 9473.
E-mail addresses: jaelim@korea.ac.kr, jaelim@usc.edu (J. Lim), jonghyun@iris.usc.edu (J. Park), medioni@iris.usc.edu (G.G. Medioni).

0262-8856/$ - see front matter © 2006 Elsevier B.V. All rights reserved.
doi:10.1016/j.imavis.2006.05.011

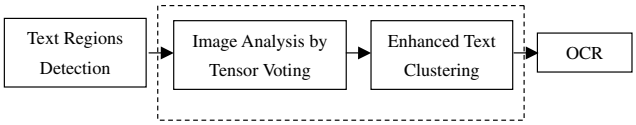


Fig. 1. Our text segmentation in common text recognition scheme.

Tensor voting was proposed by Medioni et al. in [19,20] and has been applied to diverse research fields such as the inference of object boundaries [21,22]: as a consequence, its use can explain the presence of noise based on surface saliency in the image feature space. This noise can be then removed by an adaptive median filter. The improved image is then segmented by clustering characters. Clustering requires parameters such as the number or centroid of modes [23,24], which are generally not known a priori. We use mean shift-based density estimation for automatic mode detection and use these modes as seed values for K-means clustering of characters. Characters are finally segmented as respective objects.

The rest of this paper is organized as follows: Section 2 gives an overview of our method. In Section 3, we discuss an important preprocessing step involving color analysis in HSI color space from one threshold based on RGB color space. In Section 4, we then describe how tensor voting is used to analyze noise on an image and adaptive median filter is applied to fill new values in noise regions. Section 5 details the clustering algorithm for text image segmentation. Experimental results are presented in Section 6. Finally, we draw our conclusions in the last section.

2. Overview of the method

Regarding the text regions in images of natural scenes, we assume the following:

- An original character is homogeneous in color because it is usually painted by one kind of color.
- A character can be corrupted by noise due to changes in illumination, dirt, or corrosion.

Input images to this method can be grayscale, pure color (no grayscale regions), or color images with some number of grayscale regions. Color (chromatic) regions are better described using the hue component of the pixels, whereas grayscale (achromatic) regions are distinguished using intensity only. Each pixel must therefore be labeled as either chromatic or achromatic to determine an appropriate feature space (hue or intensity) for analysis. As characters are nearly homogeneous in the appropriate feature space, the labeling process approximately divides the image into layers. Noise, however, is a deviation from the homogeneous surroundings and will therefore lie outside of the text layers. In our approach, we use the tensor voting framework in 3D to extract homogeneous layers as continuous regions with high surface saliency, and characterize noise as pixels with low surface saliency. Noise pixels are then removed using an adaptive median filter.

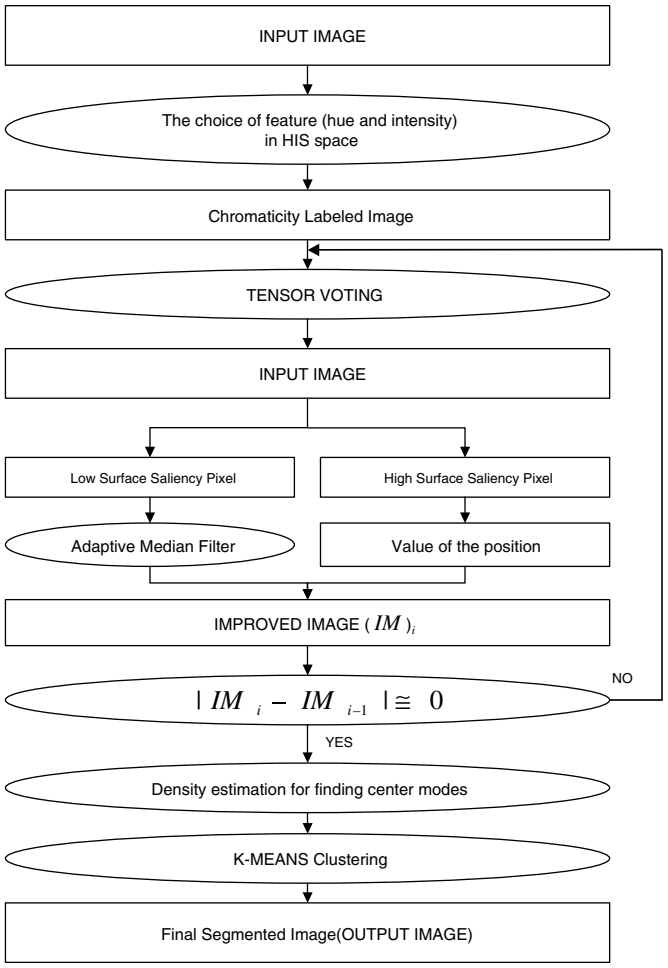


Fig. 2. The overall framework of our proposed method.

The improved image is segmented using K-means clustering algorithm. Automatic selection of clustering modes is performed using mean shift-based density estimation. Fig. 2 below shows the overall framework of our proposed method for text segmentation.

3. Color components analysis

3.1. Color space analysis

Color can be an important clue to extract some information from an image. There are many common choices of color spaces for analysis of images, each with distinct advantages for. The color in a given image can be represented in RGB, HSI, CMYK, CIE-L*a*b, and CIE-L*a*b space. In this paper, we use HSI (Hue, Saturation and Intensity) space converted from the original RGB space of the input image. In HSI space, the numerical value about chromatic and achromatic region can be adapted to the tensor voting framework, which originally classifies pixels into perceptual structures based on pixels' orientation. Instead of finding the orientation of pixel by edges' orientation (or direction), hue component's angle value is used in the tensor voting framework. In addition, intensity values are also represented

like the angle of hue component. Moreover, the hue component, which is determined by the reflective property of one object surface, is invariant to certain types of highlights, shading, and shadows providing considerable stability over the original RGB space analysis. In achromatic regions, the hue component may be unstable or meaningless, such as in a gray level region or pure grayscale image. In such achromatic regions, the intensity component is used for distinguishing objects in an image [3,25].

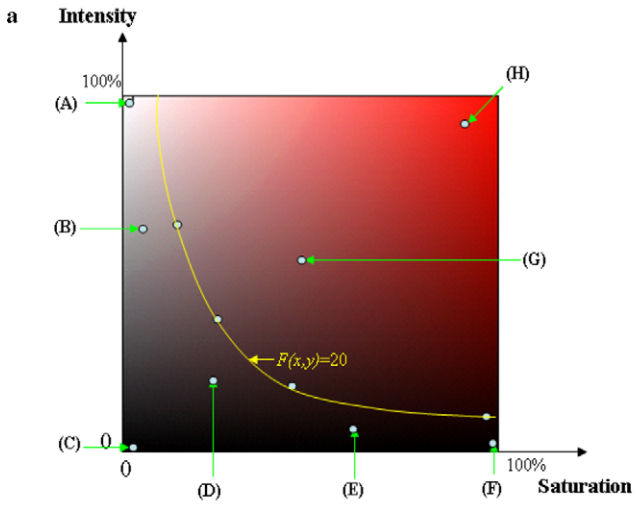
3.2. The choice of feature (hue and intensity) in HSI space

An image may contain achromatic as well as chromatic regions. Specially, in our case of focusing on characters, one object is mainly classified in either perceptually chromatic or achromatic region. As previously mentioned, using only the hue component for segmentation can be unstable in achromatic regions. In such regions, it is more appropriate to use intensity for segmentation. This section details a decision function for classifying a pixel as chromatic or achromatic such that the appropriate feature (hue or intensity) is used in segmentation. In [26], Sural et al. used the saturation value to determine the relative dominance of hue and intensity. Thresholding on saturation, however, is not illumination invariant. When a chromatic region is illuminated brightly, the saturation value in the region is likely to be low compared to the same chromatic region with lower illumination. The low saturation incorrectly indicates an achromatic region.

We propose an alternative decision function in RGB space that is independent of illumination. Instead of thresholding on saturation, we derive a chromaticity measure based on the sum of differences of R (red), G (green), and B (blue) components at each pixel (x, y) in RGB space.

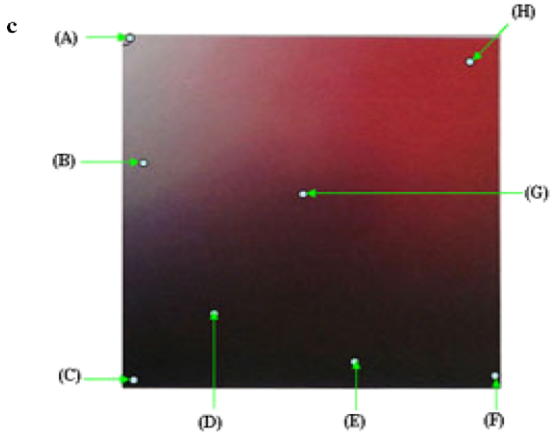
$$F(x,y) = \frac{|R(x,y) - G(x,y)| + |G(x,y) - B(x,y)| + |R(x,y) - B(x,y)|}{3}. \quad (1)$$

From our experimental observation, the smaller the sum, the closer the related position is to the achromatic regions.



b

Position	Red	Green	Blue	F(x,y)
(A)	248	243	242	**4**
(B)	140	131	128	**8**
(C)	5	5	5	**0**
(D)	60	44	41	**12.7**
(E)	33	16	16	**11.3**
(F)	4	1	0	**2.7**
(G)	142	72	68	**49.3**
(H)	240	32	24	**144**



d

Position	Hue in (a)	Hue in (c)	Intensity(%) in (a)	Intensity(%) in (c)
(A)	0	43	99	69
(B)	0	5	67	52
(C)	0	120	3	9
(D)	0	52	20	11
(E)	0	270	5	9
(F)	0	150	4	11
(G)	0	2	50	36
(H)	0	2	88	63

Fig. 3. The analysis of a color component: the value of hue is 0 and the range of RGB is [0–255]. (a) The analysis of a red (hue = 0). (b) The data of position in (a). (c) Photograph captured from (a). (d) Comparison between (a) and (c).

Fig. 3 shows how one fixed hue component is affected by both intensity and saturation components. It shows that saturation varies with illumination. We can observe that some parts such as position (F) in Fig. 3(a) as perceptually achromatic regions have high saturation. Meanwhile, the sum in (1) is low in all perceptually achromatic regions as well as the position (F) and high in all chromatic regions. Moreover, we have observed that a photograph by a digital camera should be analyzed depending on perceptual property rather than originally painted hue values. Although a synthetic red image is painted by one hue component in Fig. 3(a), achromatic regions (A, B, C, D, E, and F) in the photograph have random values of hue as shown in Fig. 3(d). On the other hand, the intensity values are fairly closer to original values.

The level of chromaticity is proportional to $F(x,y)$ in (1). A threshold value TH1 = 20 is used (determined heuristically) to classify a pixel with RGB components in the range [0–255]. Values below TH1 are classified as being achromatic and analyzed using the intensity component ($\boldsymbol{Int}(\boldsymbol{x},\boldsymbol{y})$) in HSI space. The remaining pixels are chromatic, and analyzed using the hue component ($\boldsymbol{Hue}(\boldsymbol{x},\boldsymbol{y})$) as shown in Fig. 4.1.

Hue and intensity values are defined in different ranges as shown in Fig. 4.1. If both hue and intensity values are normalized in the same range [0.0–1.0], as in Fig. 4.2, there is a possibility that the two regions (A and B) in Fig. 4.2(d) cannot be distinguished in a chromaticity labeled image, as shown in Fig. 4.2(e). Assuming the maximum intensity value to be 255, intensity values are normalized in the range [0–0.4]. Hue values in their original range [0–2π] are normalized in the range [0.6–1.0]. The use of two separate ranges maintains distinct regions in the feature space so that the chromaticity labeled image shows two distinct regions (A and B) in Fig. 4.3(e).

In the chromaticity labeled image, hue components are still values normalized from angles, which we take into account later. The values near 0.6 and 1.0 are clustered as one mode due to the cyclic property of hue component. In addition, leaving a gap between two feature ranges prevents that achromatic and chromatic regions are overlapped during clustering. The final values of a chromaticity labeled image are distributed in the range [0–1]. The values corresponding to one image are applied to the tensor voting framework in 3D.

Fig. 5 demonstrates the results of labeling pixels with respect to their chromaticity level, as an important segmentation preprocessing step in our proposed algorithm. The image in Fig. 5(b) is characterized by intensity components only. The intensity values of the text "Bus" are inhomogeneous due to illumination, so that proper segmentation is difficult. The image in Fig. 5(c) using hue components

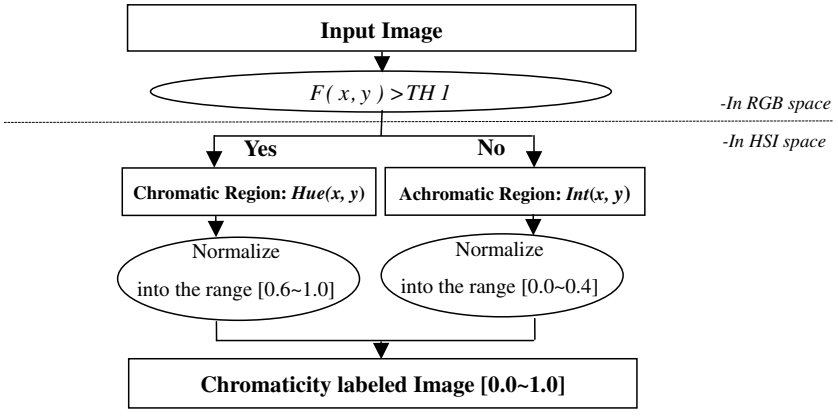


Fig. 4.1. Flow of color component selection in HSI space, where $\boldsymbol{Hue}(\boldsymbol{x},\boldsymbol{y})$ and $\boldsymbol{Int}(\boldsymbol{x},\boldsymbol{y})$ indicate the hue and intensity components, respectively.

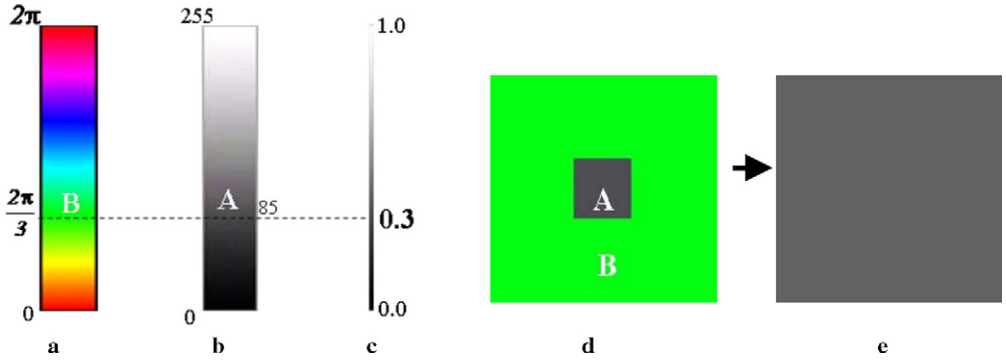


Fig. 4.2. Normalization in the same range [0–1] of hue and intensity values: (a) Original hue range [0–2π]. (b) Original intensity range [0–255]. (c) Normalization in the range [0–1]. (d) A synthetic image. (e) Chromaticity labeled image.

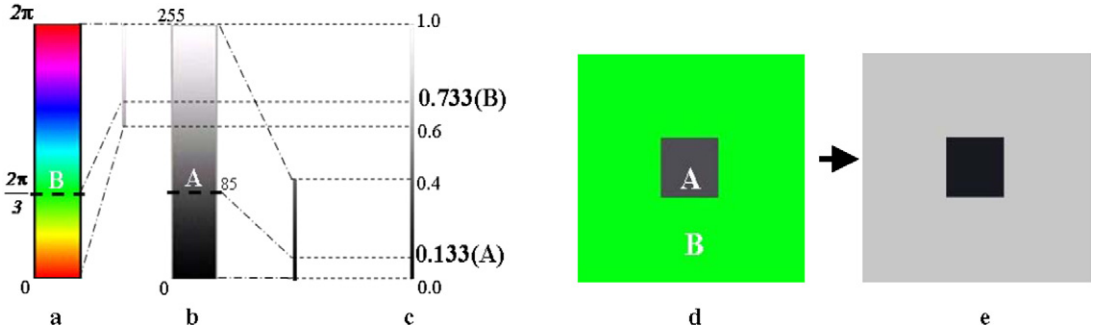


Fig. 4.3. Normalization in the respective range for intensity and hue values: (a) Original hue range [0–2π]. (b) Original intensity range [0–255]. (c) Normalization in the range [0–0.4] for intensity and [0.6–1.0] for hue. (d) A synthetic image. (e) Chromaticity labeled image.

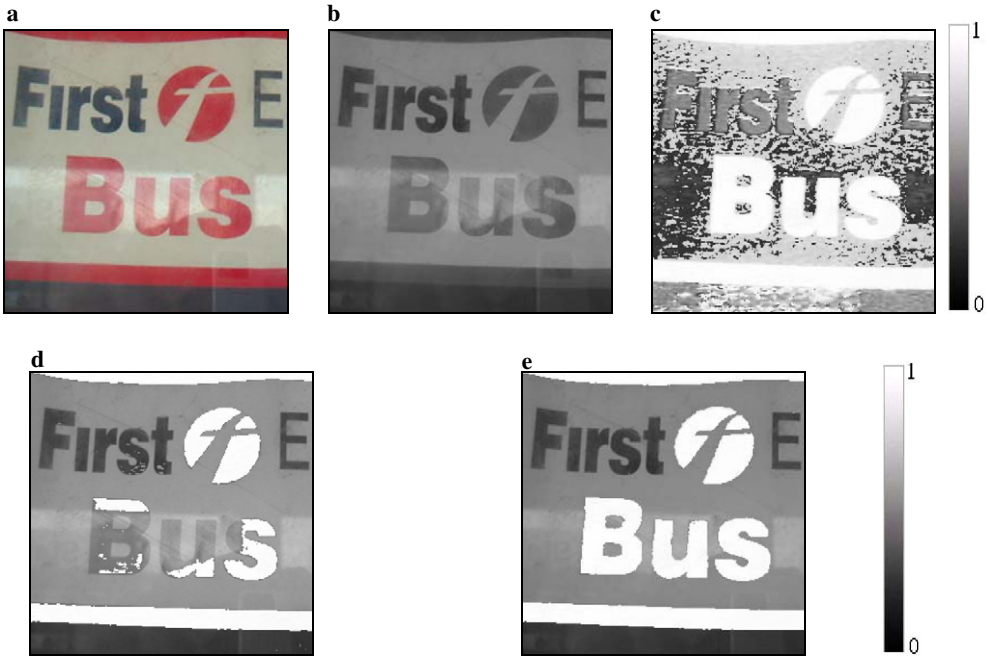


Fig. 5. One image example analyzed by hue, intensity, and our method. (a) Input image. (b) Intensity domain. (c) Hue domain. (d) Hue and intensity by saturation. (e) Our proposed method.

clearly improves the uniformity of the "Bus" region. However, achromatic regions such as the background and text "First" and "E" appear to become less homogeneous. In Fig. 5(d), the hue or intensity value of each pixel is shown dependent upon the classification of chromatic or achromatic by thresholding on the saturation value as proposed by Sural et al. in [26]. As the result, the texts "Bus" or "First" may not have proper data for text segmentation. However, Fig. 5(e) classifies pixels using our decision function as discussed in the previous section.

4. Natural scene analysis using tensor voting in 3D

4.1. Review of tensor voting

In our approach, the tensor voting framework in 3D [19,27] is used to detect the presence of noise in the chromaticity labeled image, which is relatively small or isolated from neighboring pixels in a natural scene. Tensor voting can classify pixels into perceptual structures such as corners, curves, or surfaces with respective saliency values. Noise is characterized as structures with low saliency. This section reviews the principles behind the tensor voting framework as applied to image segmentation.

Tensor voting is a local method to aggregate and propagate information. An initial set of input tokens at specific locations, broadcast their information in a fixed size neighborhood, by applying a "field" which encodes local smoothness. All sites aggregate the received votes to produce a local estimate of structure, such as curves or surfaces. A local marching process can then extract the most salient structures.

4.1.1. Data representation

Each pixel in an image may belong to some perceptual structure such as a corner, curve, or surface. To capture

the perceptual structure of input sites, tokens are defined and used. The tokens are represented by a second order symmetric non-negative definite tensor encoding perceptual saliency. The tensor can indicate its preferred tangent, normal orientation as well as saliency corresponding to its perceptual structures and be visualized as an ellipse in 2D and an ellipsoid in 3D. Such information is collected by a communication between input sites: tensor voting.

4.1.2. Tensor voting

Input tokens encoded as tensors cast votes computed through a voting field (2) to their neighborhood. The voting field explains how the tokens relate their information, such as orientation and magnitude, to their neighborhood to ensure smooth continuation. All voting fields are based on the fundamental 2D stick voting kernel, the decay function of which is:

$$\mathrm{DF}(s,k,\sigma) = \mathrm{e}^{-\left(\frac{s^2+ck^2}{\sigma^2}\right)}, \quad \text{where } s = \frac{l\theta}{\sin(\theta)},\ k = \frac{2\sin(\theta)}{l}. \tag{2}$$

The parameter s is the arc length OP, k is the curvature, c is a constant, and σ is the scale of voting field controlling the size of the voting neighborhood and the strength of votes in Fig. 6. Moreover, as seen in Fig. 6, the orientation of the stick vote is normal to the smoothest circular path connecting the voter and receiver.

4.1.3. The analysis of tensor voting

All tokens accumulate votes from the neighborhood and their collected information is computed as a covariance matrix S by the second order tensor sums (where $[v_x, v_y]$ is a vector vote generated by the neighbor pixel for center pixel.):

$$S = \begin{bmatrix} \sum v_x^2 & \sum v_x v_y \\ \sum v_y v_x & \sum v_y^2 \end{bmatrix}, \tag{3-1}$$

$$S = \begin{bmatrix} \sum \cos^2\theta & \sum \cos\theta \sin\theta \\ \sum \cos\theta \sin\theta & \sum \sin^2\theta \end{bmatrix}. \tag{3-2}$$

While (3-1) is the conventional notation for the analysis of tensor voting, (3-1) can also be expressed by (3-2) ($v_x = \cos\theta, v_y = \sin\theta$). For our approach, in (3-2), the θ is the numerical value of pixels on the chromaticity labeled image [0.0–1.0] based on the angle given in Section 3. Given its eigensystem, consisting of two eigenvalues (λ_1, λ_2) and two eigenvectors $(\hat{e}_1, \hat{e}_2)$, the matrix can be rewritten as:

$$S = (\lambda_1 - \lambda_2)\hat{e}_1\hat{e}_1^{\mathrm{T}} + \lambda_2\left(\hat{e}_1\hat{e}_1^{\mathrm{T}} + \hat{e}_2\hat{e}_2^{\mathrm{T}}\right), \tag{4}$$

where $\hat{e}_1\hat{e}_1^{\mathrm{T}}$ and $\hat{e}_1\hat{e}_1^{\mathrm{T}} + \hat{e}_2\hat{e}_2^{T}$ indicate a stick and ball tensor, with respective saliencies $\lambda_1 - \lambda_2$ and λ_2. Examining the eigensystem, we can infer the most likely perceptual structure of the token as either a surface, a curve, or a corner.

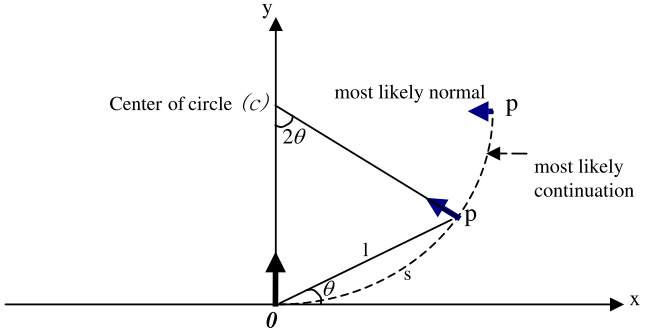


Fig. 6. The generation of tensor voting field in 2D.

4.2. Tensor voting and analysis in 3D

In our case, input tokens are first encoded as 3D ball tensors in a 3-dimensional space ($x, y,$ *value of position in a chromaticity labeled image*). These initial tensors communicate with each other to understand the most preferred orientation information at each position. Votes are accumulated at all positions by tensor addition based on the voting field. The result of one position is given in matrix form by:

$$S_{3\mathrm{D}} = [\,\hat{e}_1 \quad \hat{e}_2 \quad \hat{e}_3\,] \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{bmatrix} \begin{bmatrix} \hat{e}_1^{\mathrm{T}} \\ \hat{e}_2^{\mathrm{T}} \\ \hat{e}_3^{\mathrm{T}} \end{bmatrix} \tag{5}$$

or equivalently

$$S_{3\mathrm{D}} = (\lambda_1 - \lambda_2)\hat{e}_1\hat{e}_1^{\mathrm{T}} + (\lambda_2 - \lambda_3)\left(\hat{e}_1\hat{e}_1^{\mathrm{T}} + \hat{e}_2\hat{e}_2^{\mathrm{T}}\right) + \lambda_3\left(\hat{e}_1\hat{e}_1^{\mathrm{T}} + \hat{e}_2\hat{e}_2^{\mathrm{T}} + \hat{e}_3\hat{e}_3^{\mathrm{T}}\right). \tag{6}$$

For surface inference, surface saliency is then given by $\lambda_1 - \lambda_2$, with normal estimated as $\hat{e}_1$. Moreover, curves and junctions are inferred from the curve and junction saliencies given by $\lambda_2 - \lambda_3$ and λ_3.

4.3. Text analysis in natural scenes

Characters are usually made to appear as regions of homogeneous color. However, the image may also be noisy, as the physical surface of the sign degrades due to corrosion, dirt, intentional defacing, etc. Noise such as cracks is more inhomogeneous, so that the noise regions are comprised of severely different values. Even though the noise regions appear with similar values, their regions size is small. In our experiments, we use 200×200, 256×256, or 512×512 as the size of input image. Noise regions can be regarded to be smaller than 10×10 (this can be adjusted according to image size) and the size of characters by experimental experiences. The same result is achieved by tensor voting.

In the tensor voting framework, one image can be represented with $[x, y, H(x,y)]$. x and y indicate the positions in the image and $H(x,y)$ is the values corresponding to respective positions in a chromaticity labeled image, which is obtained in the previous step. The values $H(x,y)$ are shown on its Z axis on Fig. 7(b) with a synthetic image. Due to the homogeneous property of characters, in the absence of noise, character features will

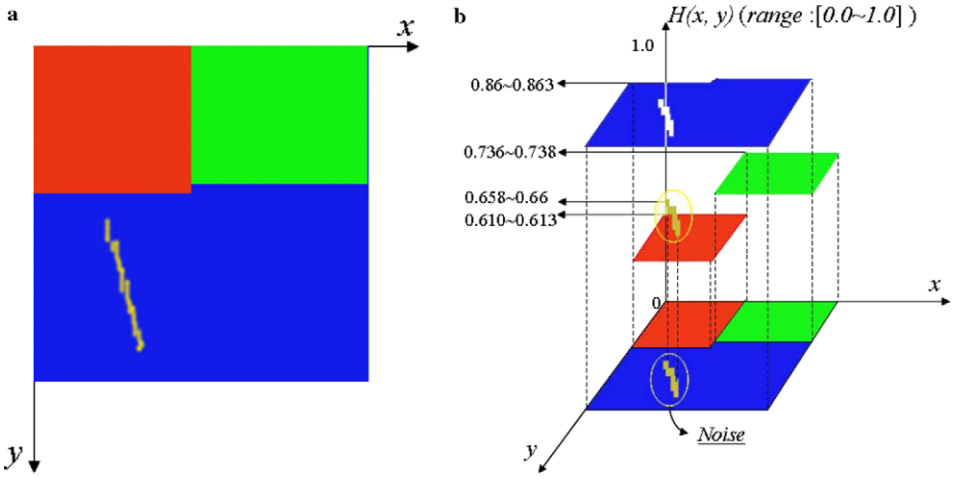
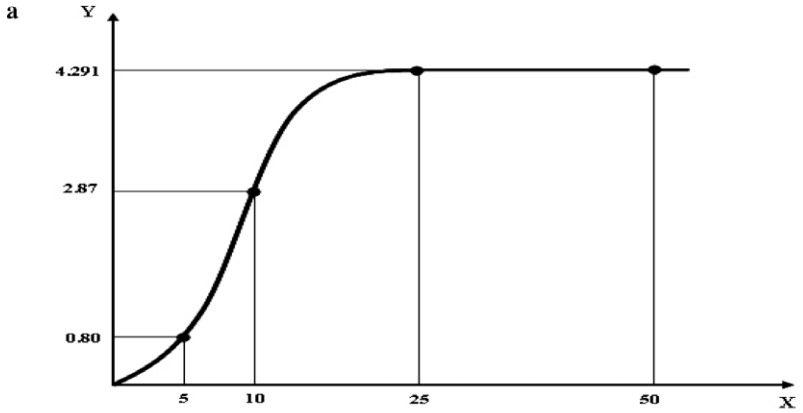


Fig. 7. The value representation of chromaticity labeled image in the tensor voting framework. (a) A synthetic image in the xy planes. (b) Analysis in the tensor voting framework.

also form layers in the chosen feature space (Hue or Intensity) as shown in Fig. 7(b). Layers are characterized by strong surface saliency in tensor voting. Meanwhile, noise regions are either inhomogeneous, or are small, and hence carry low surface saliency compared to character objects.

In Fig. 8, homogeneous regions occupying the size of more than 10×10 show strong surface saliency based on that the $\sigma = 8$ (the scale of voting field). The empirical surface saliency analysis of pixels can be used to select a threshold for determining noise regions. As we mentioned previously, we have observed that noise is shown in



c

Position	Surface Saliency
(A)	1.3
(B)	4.291
(C)	4.291
(D)	0.78
(E)	0.89
(F)	0.76
(G)	4.291

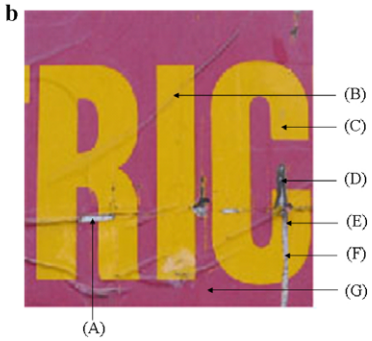


Fig. 8. Experimental result of surface saliency by tensor voting. (a) Experimental result of surface sliency values (Y) corresponding the size of homogeneous layer (X) with $\sigma = 8$. (b) Natural image. (c) Surface saliency of each position in (b)-image.

homogeneous regions smaller than 10×10. From such an observation and the experimental data in Fig. 8(a), we figure out that the threshold ($TH2$) can be selected between [0–2.87] (here we use $TH2 = 0.9$) when the maximum value of surface saliency in a given image is 4.291 based on that $\sigma = 8$. In addition, if the value of a selected threshold is close to 0, noise occupying broad homogeneous regions may not be removed clearly and then the remaining noise can be removed through iteration.

After tensor voting described in Fig. 9(b), a "surface saliency map" defines surface saliency at every pixel in a given image [19]. The map is able to indicate the presence of noise on characters as in Fig. 9 (c) by white regions. The white pixels can be replaced by applying an adaptive median filter to the saliency information.

4.4. The application of adaptive median filter

Tensor voting builds a surface saliency map for the image. In the map, low surface saliency regions are primarily considered noise and should be replaced with values of the high surface saliency neighbors. An adaptive median filter is used to remove such noise regions.

If a pixel is judged as noise from the tensor voting analysis, neighbors surrounding the pixel in a (3×3) window are initially examined to find high surface saliency with which to replace the pixel. If the noise region is broad, however, the 3×3 window may be insufficient to find high surface saliency defined by $TH2$. The size of window $((m + s) \times (n + v))$ is therefore increased until the proper number of high surface saliency pixels is detected (in our implementation we use the value $TH3 = 8$). Then, the median value among the high surface saliency pixels within the final window is selected. This process removes noise on the character and background regions making character region segmentation more effective for simple text recognition. The steps below represent the process of this adaptive median filter in detail.

STEP 1: Check the surface saliency of neighbor pixels in the $(m + s) \times (n + v)$ window surrounding the noise pixel:
$S(x, y)$, $1 \leqslant x \leqslant m + s$, $1 \leqslant y \leqslant n + v$
(As an initial value, $m = n = 3$ and $s = v = 0$.)
STEP 2: The number of high surface saliency pixel in the window:
IF $(S(x, y) \geqslant TH2)$
Count = Count + 1;
STEP 3: The size change of windows for median filter
IF (Count $< TH3$)
$s = s + 2$, $v = v + 2$ and GOTO STEP 1
ELSE

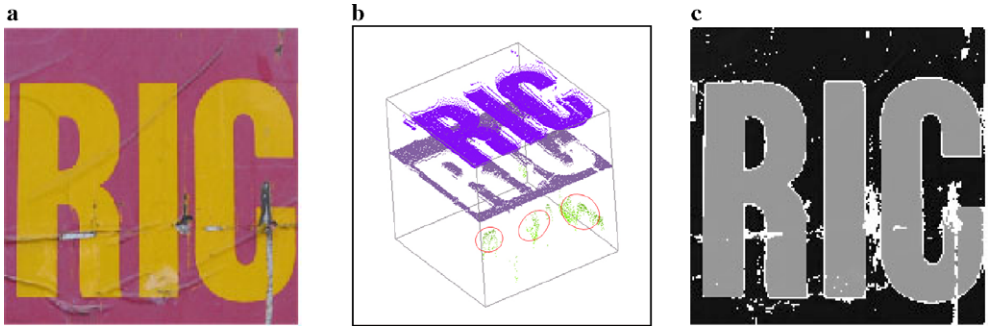


Fig. 9. The result from tensor voting: (a) original image, (b) the data in the tensor voting framework where red circle indicates noise, (c) the normalized representation of chromaticity labeled image in the range [0–1] with white regions denoting noise.

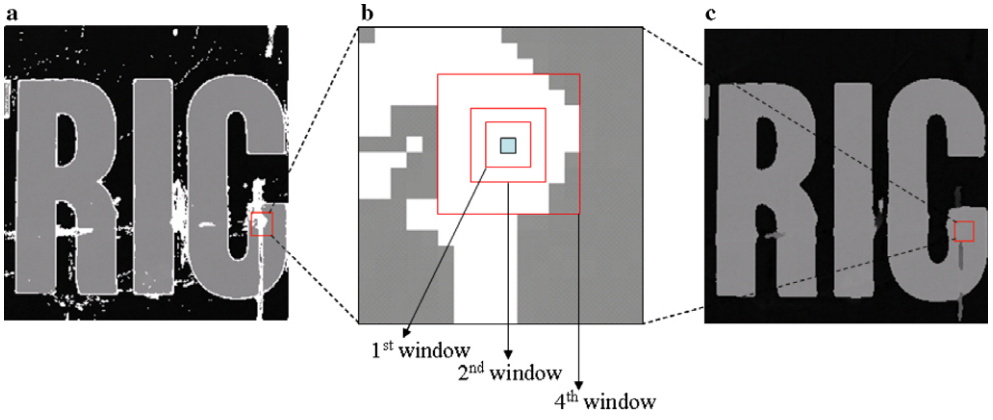


Fig. 10. The result from using adaptive median filter: (a) image analysis by tensor voting, (b) the size of filter windows, (c) enhanced 1st iteration result in the normalized range [0–1].

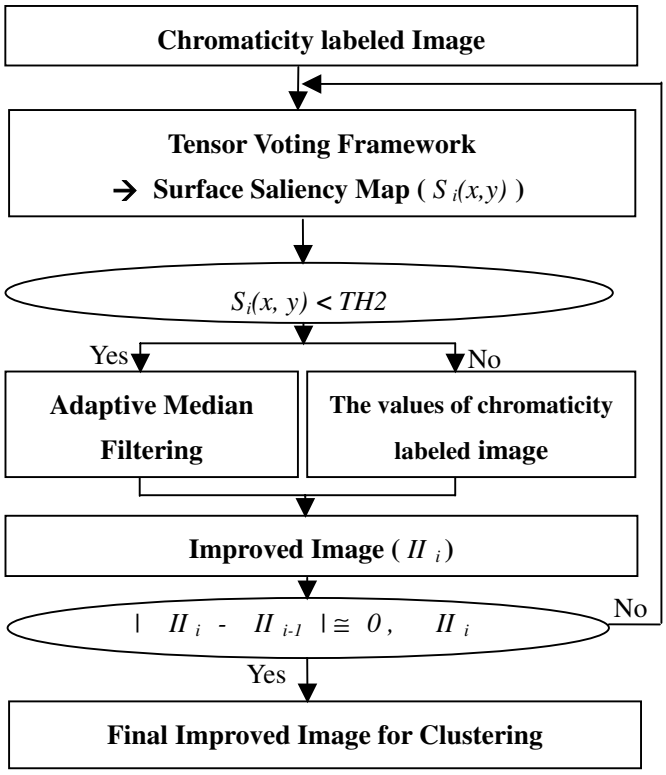


Fig. 11. Flow chart of the proposed method (i = the number of iteration).

GOTO STEP 4

STEP 4: Enumerate values of pixels corresponding high surface saliency in the increased window:

$H(x,y)$, $1 \leqslant x \leqslant m+s$, $1 \leqslant y \leqslant n+v$

STEP 5: Find the median value among the enumerated values and fill the median value in the noise pixels.

Fig. 10 briefly shows an example of the adaptive median filter procedure. Fig. 10(a) is the result of tensor voting. Noise is shown as white regions in the image. The remaining pixels are described by the values of chromaticity labeled image. In Fig. 10(b), a window is increased to find pixels corresponding to high surface saliency. Namely, the first and second windows of this filter do not contain such pixels and the increased 4th window has nine high surface saliency pixels. The pixels can be the candidate positions. The values in the candidate positions are enumerated and the median value among them is used to replace the noise pixels in Fig. 10(c).

4.5. Iteration with adaptive median filter

The surface saliency map by tensor voting indicates information where noise exists. An adaptive median filter is then used to remove noise. However, some noise may remain after a single pass of the tensor voting and median

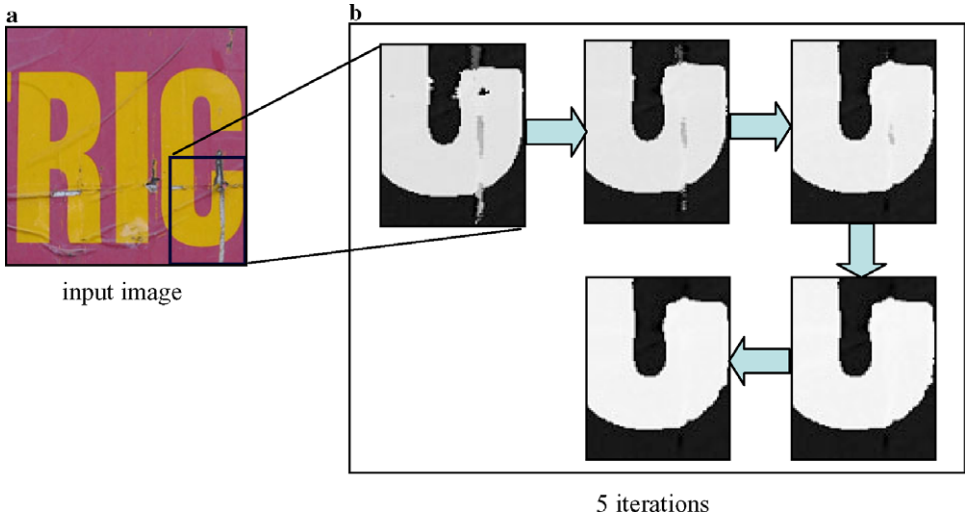


Fig. 12. One example of our proposed method.

Fig. 13. The comparison of our approach with a general median filter: (a) 3×3 median filter (b) 9×9 median filter (c) 17×17 median filter (d) the result by our approach.

filtering. We therefore repeatedly apply the filter to remove the remaining noise as described in Fig. 11. The saliency map is recomputed and median filtering applied repeatedly until the change between the current and pervious image is negligible. The resulting image is then used in the final clustering stage. Fig. 12 shows an example of this process for character in a natural scene. Four iterations of the tensor voting and median filtering process are performed. The fifth iteration provides little improvement and the termination criterion is met. The final improved image is provided as an input for the final density estimation and clustering procedure.

In Fig. 13, we additionally show that our approach produces better improved images than general median filter with different window sizes.

5. Text image segmentation using clustering algorithm

Following the preprocessing and noise removal in the previous stages, the image consists of an unknown number of distinct and homogeneous regions. To segment these regions, the K-means clustering algorithm can be used. Clustering is sensitive to initial values, which are generally selected manually. We therefore use mean shift-based density estimation to estimate both the number of regions as well as the seed values for K-means clustering.

5.1. Density gradient estimation

The image is interpreted as n data points in a d-dimensional space where n is the number of pixels in the image. The values of improved image are distributed in the range [0–1] and used directly, giving a 1-dimensional feature space. The initial values for distinct characters coincide with the modes of the data.

Mean shift-based density gradient estimation with sampling data finds the local maximum of the probability densities [28,29]. Let $\{\mathbf{X}_i\}$, $i = 1, \ldots, n$, be the set of n data points in a d-dimensional Euclidean space. The multivariate kernel density estimate obtained with kernel $K(\mathbf{x})$ and window radius for bandwidth h, computed at point $\mathbf{x}$ is defined as:

$$\hat{f}_K(\mathbf{x}) = \frac{1}{nh^d}\sum_{i=1}^{n} K\left(\frac{\mathbf{x}-\mathbf{X}_i}{h}\right). \tag{7}$$

Here, we are interested only in a class of radially symmetric kernels satisfying

$$K(\mathbf{x}) = c_{K,d}k(\|\mathbf{x}\|^2),$$

in which case it suffices to define the function $k(x)$ called the profile of the kernel, only for $x \geqslant 0$ and $c_{K,d}$ is the normalized constant which makes $K(\mathbf{x})$ integrate to one.

The differentiation of the kernel allows one to define the estimate of the density gradient as the gradient of the kernel density estimate:

$$\begin{aligned}\nabla\hat{f}_K(\mathbf{x}) &= \frac{1}{nh^d}\sum_{i=1}^{n}\nabla K\left(\frac{\mathbf{x}-\mathbf{X}_i}{h}\right)\\ &= \frac{2c_{K,d}}{nh^{d+2}}\sum_{i=1}^{n}(\mathbf{x}-\mathbf{X}_i)k'\left(\left\|\frac{\mathbf{x}-\mathbf{X}_i}{h}\right\|^2\right).\end{aligned} \tag{8}$$

We define the derivative of the kernel profile as a new function

$$g(x) = -k'(x),$$

and assume that this exists for all $x \geqslant 0$, except for a finite set of points. Now, if we use a function for profile, the kernel is defined as

$$G(\mathbf{x}) = c_{G,d}g(\|\mathbf{x}\|^2),$$

where $c_{G,d}$ is the corresponding normalization constant. In this case, the kernel $K(\mathbf{x})$ is called the shadow of kernel $G(\mathbf{x})$. If we use a function $g(x)$ in formula (8), then the gradient of the density estimator is written by

$$\begin{aligned}\nabla\hat{f}_K(\mathbf{x}) &= \frac{2c_{K,d}}{nh^{d+2}}\sum_{i=1}^{n}(\mathbf{x}-\mathbf{X}_i)g\left(\left\|\frac{\mathbf{x}-\mathbf{X}_i}{h}\right\|^2\right)\\ &= \frac{2c_{K,d}}{nh^{d+2}}\sum_{i=1}^{n}g\left(\left\|\frac{\mathbf{x}-\mathbf{X}_i}{h}\right\|^2\right)\left(\frac{\sum_{i=1}^{n}g\left(\left\|\frac{\mathbf{x}-\mathbf{X}_i}{h}\right\|^2\right)\mathbf{X}_i}{\sum_{i=1}^{n}g\left(\left\|\frac{\mathbf{x}-\mathbf{X}_i}{h}\right\|^2\right)}-\mathbf{x}\right).\end{aligned} \tag{9}$$

Here, this is given as the product of two terms having special meaning. The first term in the expression (9) is proportional to the density estimate at $\mathbf{x}$ computed with the kernel $G(\mathbf{x})$

$$\hat{f}_G(\mathbf{x}) = \frac{1}{nh^d}\sum_{i=1}^{n}G\left(\frac{\mathbf{x}-\mathbf{X}_i}{h}\right) = \frac{c_{G,d}}{nh^d}\sum_{i=1}^{n}g\left(\left\|\frac{\mathbf{x}-\mathbf{X}_i}{h}\right\|^2\right),$$

and the second term is defined as the mean shift vector

$$\mathbf{m}_G(\mathbf{x}) = \left(\frac{\sum_{i=1}^{n}g\left(\left\|\frac{x-X_i}{h}\right\|^2\right)X_i}{\sum_{i=1}^{n}g\left(\left\|\frac{x-X_i}{h}\right\|^2\right)}-x\right). \tag{10}$$

This vector is the difference between the weight mean using the kernel $G(\mathbf{x})$ for weights and the center of the kernel. Then, we can rewrite the expression (9) as

$$\nabla\hat{f}_K(\mathbf{x}) = \frac{2c_{K,d}}{h^2c_{G,d}}\hat{f}_G(\mathbf{x})\mathbf{m}_G(\mathbf{x}),$$

which yields

$$\mathbf{m}_G(\mathbf{x}) = \frac{1}{2}h^2c\frac{\nabla\hat{f}_K(\mathbf{x})}{\hat{f}_G(\mathbf{x})}. \tag{11}$$

The expression (5) shows the mean shift vector being proportional to the gradient of the density estimate at the point it is computed. As the vector points in the direction of maximum increase in density, it can define a path leading to a local density maximum which becomes a mode of

density. It also exhibits a desirable adaptive behavior, with the mean shift step being large for low-density regions and decreases as a point $\mathbf{x}$ approaches a mode. Each data point thus becomes associated to a point of convergence, which represents a local mode of the density in the d-dimensional space.

5.2. Mean shift-based mode detection

Let us denote by $\{\mathbf{y}_1, \mathbf{y}_2, \ldots\}$ the sequence of successive locations of kernel $G(\mathbf{x})$, where these points are computed by the following formula

$$\mathbf{y}_j = \frac{\sum_{i=1}^{n} g\left(\left\|\frac{\mathbf{x}-\mathbf{X}_i}{h}\right\|^2\right)\mathbf{X}_i}{\sum_{i=1}^{n} g\left(\left\|\frac{\mathbf{x}-\mathbf{X}_i}{h}\right\|^2\right)} \quad j = 1, 2, \ldots \tag{12}$$

This is the weighted mean at $\mathbf{y}_j$ computed with kernel $G(\mathbf{x})$ and $\mathbf{y}_1$ is the center of the initial position of the kernel, $\mathbf{x}$. The corresponding sequence of density estimates computed with shadow kernel $K(\mathbf{x})$ is given by

$$\hat{f}_K(j) = \hat{f}_K(\mathbf{y}_j), \quad j = 1, 2, \ldots$$

Here, if the kernel has a convex and monotonically decreasing profile, two sequence $\{\mathbf{y}_1, \mathbf{y}_2, \ldots\}$ and $\{\hat{f}_K(1), \hat{f}_K(2), \ldots\}$ converge and $\{\hat{f}_K(1), \hat{f}_K(2), \ldots\}$ is monotonically increasing. After that, let us denote by $\mathbf{y}_c$ and $\hat{f}_K^c$ the convergence points of their sequences respectively. Here, we can get two kinds of implications from the convergence result. First, the magnitude of the mean shift vector converges to zero. In fact, the j-th mean shift vector is given as

$$\mathbf{m}_G(\mathbf{y}_j) = \mathbf{y}_{j+1} - \mathbf{y}_j,$$

and this is equal to zero at the limit point, $\mathbf{y}_c$. In other words, the gradient of the density estimate computed at $\mathbf{y}_c$ is zero. That is,

$$\nabla \hat{f}_K(\mathbf{y}_c) = 0.$$

Hence, $\mathbf{y}_c$ is a stationary point of density estimate, $\hat{f}_K(\mathbf{x})$. Second, since $\{\hat{f}_K(1), \hat{f}_K(2), \ldots\}$ is monotonically increasing, the trajectories of mean shift iterations are attracted by local maximum if they are unique stationary points. That is, once $\mathbf{y}_j$ gets sufficiently close to a mode of density estimate, it converges to mode.

The theoretical results obtained from the above implications suggest a practical algorithm for mode detection:

Step 1: Run the mean shift procedure to find the stationary points of density estimates.
Step 2: Prune these points by retaining only the local maximum.

This algorithm automatically determines the number and location of modes of estimated density function. We shall use the detected mode or cluster centers from the mean shift procedure to be manifestations of underlying components of the clustering algorithm for our image segmentation task.

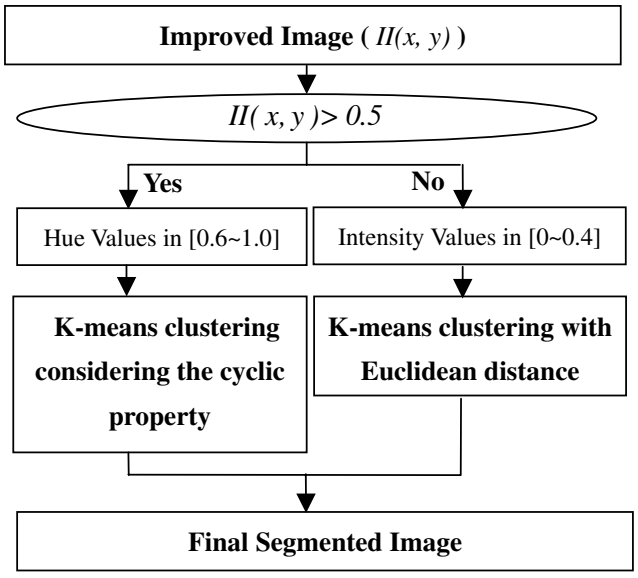


Fig. 14. Our segmentation scheme using K-means clustering algorithm.

5.3. Clustering

The number and centroid of modes selected in Section 5.1 are used as seed values in K-means clustering. K-means clustering is then applied to the values in the improved image to segment the character [3].

In our case, we should perform two different K-means clustering algorithm because intensity values are linear and hue values are characterized with the cyclic property as shown in Fig. 14. First, intensity values and their seed values fall in the range [0–0.4] as normalized in chromaticity labeled image as well as the improved image. Intensity values compute Euclidean distance between itself value and seed values to find the closet seed value without considering the seed values in the range [0.6–1.0]. The second K-means clustering algorithm should be used for hue values normalized into the range [0.6–1.0] so that the algorithm can account for the cyclic property. In that case, the values of every pixel find the closest one among seed values in the range [0.6–1.0] based on the approach in [3]. Zhang et al. in [3] show that values near the minimum (0.6) and maximum (1.0) are clustered as one mode. Two K-means clustering passes are therefore performed while maintaining both the linear property of intensity values in the range [0–0.4] and the cyclic property of hue values in the range [0.6–1.0].

6. Experimental results

We have tested our approach on natural scene images which are corrupted by noise. The images are of RAW type, with the sizes from 200×200, 256×256, and

512×512. After text region detection to define a region of interest, our approach is performed. In our experiment, text regions are manually detected and the selected regions are segmented using our method. Fig. 15 shows our experimental results. The first and third image contains nonlinear red components which can typically cause problems when using the hue component for image segmentation. The results show that our approach is considering the nonlinear parts in hue component as well as removing noise. In Fig. 16, we can see that our approach handles achromatic and chromatic images equally well. Moreover, Fig. 17 compare our approach with a state-of-the art segmentation approach such as EDISON [29].

Lastly, Fig. 18 shows a comparison of our approach to three other segmentation approaches (EDISON, by median filter, and GMM [30]) in respect of error rates. Fig. 18(a) illustrates image data extracted from original natural scenes and Fig. 18(b) shows results segmented in manually labeled ground truth images. Compared to the results segmented by K-means clustering in Fig. 18(b), we indicate errors as both FP and ND in Fig. 18(c). FP (false positive) indicates background pixels classified as character pixels in a segmented image and ND (no detection) indicates character pixels classified as background pixels or noise values in a segmented image. Fig. 18(d) illustrates one example of detected errors (FP and ND). The result of our approach shows the lowest errors among four approaches. To show the error rate (ER) as one numerical value, we calculate the similarity between results segmented from ground truth images and original noisy images by:

Fig. 15. Examples of experimental results: (a) original natural scenes, (b) segmented image.

Fig. 16. Examples of experimental results primarily containing achromatic regions: (a) original natural scenes, (b) segmented image.

Fig. 17. The comparison with an existent segmentation approach. (a) Natural image. (b) The result by EDISON. (c) The result by our approach.

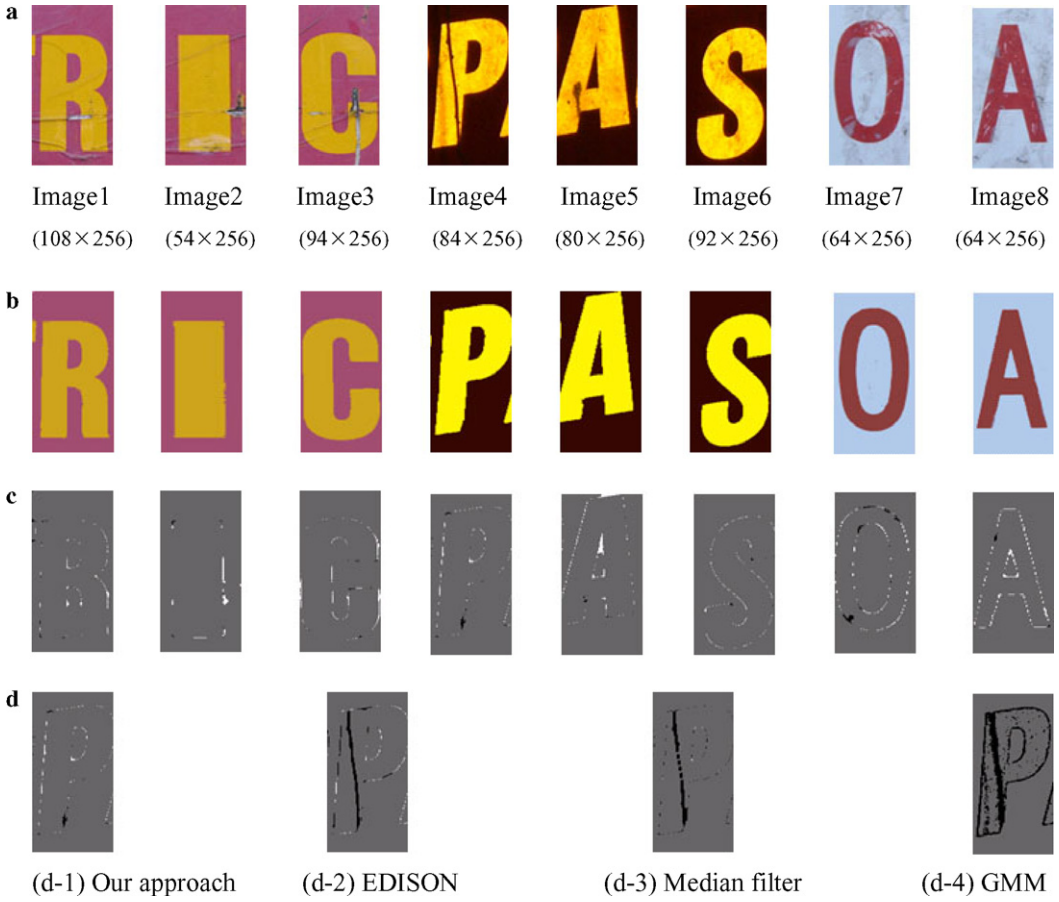


Fig. 18. Performance comparison of our approach to other segmentations. (a) Image data. (b) The segmentation in ground truth images. (c) The errors of our approach: FP (white) and ND (black). (d) The comparison of our approach to other regarding to errors: FP (white) and ND (black).

$$\text{Similarity} = \frac{\text{Result\_ni} \cap \text{Result\_gt}}{\text{Result\_ni} \cup \text{Result\_gt}} \tag{13}$$

$$\text{ER}(\%) = [1 - \text{Similarity}] \times 100, \tag{14}$$

where Result_ni is a character result segmented from an original noisy image and Result_gt is a character result segmented from the ground truth image.

Similarity in (13) measures the ratio of pixels with the same assignment in the ground truth and the results by our approach. Exact matching and no-matching have an ER of 0% and 100%, respectively. Table 1 shows the statistical data of ER. Experimental result by our approach is the closest to 0% in Table 1. Our approach has performed better segmentation, potentially improving accuracy and reducing computational complexity of OCR algorithms.

And then, the processing time of the EDISON, GMM, Median filter, and our approach on three real images are shown in the Table 2. The processing time of our proposed method may not be efficient because our approach method has iteration procedure for restoring damaged regions. However, this approach provides a superior segmentation through reducing the noise remarkably from a damaged color text images.

Table 1
Performance comparison of four approaches with error rates (ER)

	Our approach (%)	EDISON (%)	By median filter (%) (5 × 5)	GMM (%)
Image 1	1.528	2.065	3.886	3.763
Image 2	1.696	9.602	2.93	6.725
Image 3	2.167	5.639	6.479	6.799
Image 4	2.314	8.482	7.332	21.67
Image 5	2.792	5.762	3.838	19.467
Image 6	1.562	2.51	1.037	22.639
Image 7	4.772	12.46	19.436	6.68
Image 8	3.623	6.2	13.875	7.78
Image 9	2.6	6.6	7.4	11.9

Table 2
Performance comparison of four approaches with processing time (s)

	Our approach (number of iteration)	EDISON	By median filter (5 × 5)	GMM
Image 1 (108 × 256)	7.5 (3 times)	2.4	2.0	2.1
Image 2 (54 × 256)	4.2 (3 times)	1.8	1.5	1.6
Image 3 (94 × 256)	9.4 (4 times)	2.1	2.0	1.9

7. Conclusions and future work

We have presented an approach for text segmentation in natural scenes. All pixels in the given image are defined with the corresponding hue and intensity component. Next, tensor voting is used for image analysis. This step can detect the presence of noise such as crack or scrawl in a given image. Adaptive median filter then provides proper values to replace the noise values which are present on characters. The improved image is used with a density estimation to find proper modes such that K-means clustering algorithm can get automatic seed values and perform text segmentation. Unlike other existent text segmentations, our approach can remove different kinds of noise well and segment one character as one object. The result can contribute to improving text recognition rate as well as reducing the complexity of final step OCR in text recognition. This approach can then be extended to handle text recognition in natural scenes.

Currently, in our experiment, one character should be classified as one either achromatic or chromatic region except noise. However, a character can be described artistically with both achromatic and chromatic region. Our future work includes even segmenting such a character as one object, and doing more experiments with natural scenes containing characters of diverse sizes.

Acknowledgments

This work was supported by the Post-doctoral Fellowship program of Korea Science & Engineering Foundation (KOSEF). Moreover, I have appreciated the advice of Gerard Medioni, Philippos Mordohai, and Douglas A. Fidaleo.

References

[1] N.R. Pal, S.K. Pal, A review on image segmentation techniques, Pattern Recognition 26 (9) (1993) 1277–1294.
[2] A. Moghaddamzadeh, N. Bourbakis, A fuzzy region growing approach for segmentation of color images, Pattern Recognition 30 (6) (1997) 867–881.
[3] Chi Zhang, P. Wang, A new method of color image segmentation based on intensity and hue clustering, IEEE International Conference on Pattern Recognition 3 (2000) 3617–3621.
[4] Xu Jie, Shi Peng-fei, Natural color image segmentation, IEEE International conference on Image Processing 1 (2003) 14–17.
[5] L. Lucchese, S.K. Mitra, Color image segmentation: a state-of-the-art survey, Proceedings of the Indian National Science Academy 67 (2001) 207–221.
[6] H.D. Cheng, X.H. Jiang, Y. Sun, Jingli Wang, Color image segmentation: advances and prospects, Pattern Recognition 34 (2001) 2259–2281.
[7] Datong Chen, Herve Bourlard, Jean-Philippe Thiran, Text identification in complex background using SVM, Proceedings Of the International Conference on Computer Vision and Pattern Recognition 2 (2001) 621–626.
[8] Y. Zhong, K. Karu, A.K. Jain, Locating text in complex color images, Pattern Recognition 28 (1995) 1523–1536.
[9] K. Jain, B. Yu, Automatic text location in images and video frames, Pattern Recognition 31 (1998) 2055–2076.
[10] Ismail Haritaoglu, Scene text extraction and translation for handheld devices, IEEE Conference on Computer Vision and Pattern Recognition (2001) 408–413.
[11] Chuang Li, Xiaoqing Ding, Youshou Wu, Automatic text location in natural scene images, International Conference on Document Analysis and Recognition (2001) 1069–1073.
[12] Color segmentation for text extraction, International Journal on Document Analysis and Recognition 6 (2004) 271–284.
[13] Jie Yang, Xilin Chen, Jing Zhang, Ying Zhang, Alex Waibel, Automatic detection and translation of text from natural scenes, IEEE International Conference on Acoustics, Speech, and Signal Processing 2 (2002) 2101–2104.
[14] Jing Zhang, Xilin Chen, Jie Yang, Alex Waibel, A PDA-based sign translator, IEEE International Conference on Multimodal Interfaces (2002) 217–222.
[15] Qixiang Ye, Wen Gao, Qingming Huang, Automatic text segmentation from complex background, IEEE International Conference on Image Processing 5 (2004) 2905–2908.
[16] S.M. Lucas, A. Panaretos, L. Sosa, A. Tang, S. Wong, R. Young, ICDAR 2003 robust reading competitions, IEEE International Conference on Document Analysis and Recognition (2003) 682–687.
[17] Qixiang Ye, Qingming Huang, Wen Gao, Debin Zhao, Fast and robust text detection in images and video frames, Image and Vision Computing 23 (2005) 565–575.
[18] Xilin Chen, Jie Yang, Jing Zhang, Alex Waibel, Automatic detection and recognition of signs from natural scenes, IEEE Transaction on Image Processing 13 (1) (2004) 87–99.
[19] G. Medioni, M.S. Lee, C.K. Tang, A Computational Framework for Segmentation and Grouping, Elsevier, 2000.
[20] Mi-Suen Lee, Gerard Medioni, Grouping ., -, →, ⋑, into regions, curves, and junctions, Computer Vision and Image Understanding 76 (1) (1999) 54–69.
[21] Jiaya Jia, Chi-keung Tang, Inference of segmented color and texture description by tensor voting, IEEE Transactions on Pattern Analysis and Machine Intelligence 26 (6) (2004) 771–786.
[22] Wai-Shun Tong, Chi-Keung Tang, Philippos Mordohai, Gerard Medioni, First order augmentation to tensor voting for boundary inference and multiscale analysis in 3D, IEEE Transactions on Pattern Analysis and Machine Intelligence 26 (5) (2004) 569–611.

[23] L. Lucchese, S.K. Mitra, Unsupervised segmentation of color images based on k-means clustering in the chromaticity plane, IEEE Workshop on Content-based Access of Image and Video Libraries (1999) 74–78.
[24] N. Ueda, R. Nakano, Deterministic annealing EM algorithm, Neural Networks 11 (1998) 271–282.
[25] Hanzi Wang, David Suter, Color image segmentation using global information and local homogeneity, Digital Image Computing: Techniques and Applications (2003) 10–12.
[26] Shamik Sural, Gang Qian, Sakti Pramanik, Segmentation and histogram generation using the hsv color space for image retrieval, IEEE International Conference on Image Processing 2 (2002) 589–592.
[27] Jiaya Jia, Chi-Keung Tang, Image repairing: robust image synthesis by adaptive ND tensor voting, IEEE Computer Vision and Pattern Recognition 1 (2003) 643–650.
[28] Hanzi Wang, David Suter, A novel robust method for large numbers of gross errors, International Conference on Control, Automation, Robotics and Vision (2002) 326–331.
[29] D. Comaniciu, P. Meer, Mean shift: a robust approach towards feature space analysis, IEEE Transaction on Pattern Analysis and Machine Intelligence 24 (5) (2001) 1–18.
[30] Carl Edward Rasmussan, The infinite Gaussian mixture model, Advances in Neural Information Processing Systems 12 (2000) 554–560.