Morphology-based Text Line Extraction

Jui-Chen Wu, Jun-Wei Hsieh*, and Yung-Sheng Chen
Department of Electrical Engineering
Yuan Ze Universit, Taiwan
**shieh@saturn.yzu.edu.tw*

ABSTRACT

This paper presents a morphology-based text line extraction algorithm for extracting text regions from cluttered images. First of all, the method defines a novel set of morphological operations for extracting important contrast regions as possible text line candidates. The contrast feature is robust to lighting changes and invariant against different image transformations like image scaling, translation, and skewing. In order to detect skewed text lines, a moment-based method is then used for estimating their orientations. According to the orientation, an x*-projection technique can be applied to extract various text geometries from the text-analogue segments for text verification. However, due to noise, a text line region is often fragmented to different pieces of segments. Therefore, after the projection, a novel recovery algorithm is then proposed for recovering a complete text line from its pieces of segments. After that, a verification scheme is then proposed for verifying all extracted potential text lines according to their text geometries. Thus, different desired text lines can be well and correctly detected from images or video frames for various document analyses. Different from traditional training methods which use an exhaustingly searching way to find all possible skewed text lines, the proposed technique can locate text lines very efficiently no matter what orientations they have. The average accuracy of the proposed system is 95.4%. Experimental results show that the proposed method improves the state-of-the-art work in terms of effectiveness and robustness for text line detection*

1: Introductions

Extracting text lines from images or videos is an important problem in many applications like document processing [1], [2], image indexing, video content summary [3]-[4], video retrieval [5], video understanding [6], and so on. Usually, texts embedded in an image or a frame capture important media contexts such as player's name, title, date, story introduction, and so on. Therefore, the task can provide various advantages for annotating an image or a video and thus improves the accuracy of a content-based indexing system to search desired media contents. In addition, the information can be used for content filtering so that commercial programs can be found and removed out for video summary. Moreover, when analyzing video audios, the recognition result of text line can provide extra refinements for correcting the errors of speech recognition.

Texts usually have different appearance changes like font, size, style, orientation, alignment, texture, color, contrast, and background [1]. All the changes will make the problem of automatic text extraction become complicated and difficult. There are many researchers who have devoted themselves to investigating different methods for tackling the above problems [1]-[6]. For example, in [3], Sato *et al*. proposed a caption detection system to detect and recognize text characters embedded in video captions. Zhong *et al*. [7] assumed that the characters embedded in an image usually have similar color, and then proposed a color reduction technique for locating them from complex images. In [8], Lienhart *et al*. made another assumption that text lines usually have high contrast to background, and then applied a motion analyzer to extract text line locations. In addition to the above character properties, the change between character boundaries also forms a good feature for text line extraction. In [9], Hasan and Karam used several morphological operations to extract this feature for text line localization. In [10], Wong and Chen computed this feature by accumulating the maximum gradient differences of pixels line by line for obtaining all potential text segments in the processed image. This feature can also be extracted from frequency domain. For example, in [11], Sin *et al*. took advantages of Fourier spectrum to extract high frequency components for text line detection. In [12], Mao *et al*. used wavelet transform to obtain edge maps at different resolutions for locating all possible text lines. The feature can also be obtained using a training process. For example, Kim *et al*. [13] used support vector machines (SVM) to learn important text features. In [14], Xiangrong *et al*. used adaboosting algorithm to build a stronger classifier for text line detection. The training scheme has superiorities in detecting normal text lines but often fails to detect skewed text lines. The training approach needs lots of training samples to train each configuration and becomes inefficient when more orientations of text line are detected.

In this paper, a novel text line detection scheme is proposed for locating different text lines from cluttered images. The major contribution of this paper is to devise a novel morphology-based technique for extracting important text contrast features from the processing images. The feature is invariant against various geometrical image changes like translation, rotation, and scaling. Even though the lighting condition or text color has changes, the feature is still not changed. Thus, the proposed morphology-based method works robustly under different image alterations. After that, a coarse-to-fine text verification scheme is proposed to verify each text-analogue segment. The coarse scheme uses two constraints including the size and the width-to-height ratio between texts to filter out all impossible text candidates. Then, a finer verification scheme is applied to verify all remained text candidates using their detailed character features. Since each text line has different orientations, this paper first uses a moment-based method to find its longest axis. Then, we can benefit from an *x*-projection technique to find text character geometries for the finer verification. After text verification, due to noise, some character regions still will be missed or a complete text is fragmented into pieces of segments. To avoid the text missing or fragmentation problem, a recovery algorithm is then proposed to adjust text boundary so that all missed text components can be recovered. Without any training process, all possible text lines can be correctly verified. The proposed method can well detect various text lines even though they are skewed. In addition, no matter how cluttered the background is, all desired text regions can be very accurately located. The average accuracy of text line detection is 95.4%. Experimental results have shown the superiority of the proposed method in text line detection.

2: OVIEW OF THE PROPOSED SYSTEM

This paper presents a novel technique for automatically locating text lines from cluttered images. Fig. 1 shows the flowchart of the proposed system. The system consists of three major parts, *i.e.*, feature extraction, candidate selection, and verification. In what follows, details of each part are described.

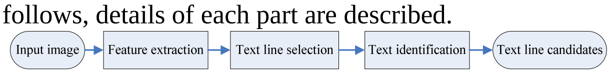


Fig. 1 Flowchart of the proposed system.

Feature Extraction: Text lines embedded in images often have quite high contrast to the background. The relative contrast between texts and their background is an important feature for text line detection. Thus, this paper proposes a novel morphology-based scheme for extracting the high contrast feature for locating all possible text lines.

Text Candidate Selection: After feature extraction, we can use a labeling technique to select all possible text lines from the analyzed image. For overcoming skewed text lines, a moment-based method is then applied to find their orientations. After that, a novel rule-based scheme is proposed to select all potential segments.

Candidate Verification: Once all potential text lines have been selected, a verification procedure will be then proposed for filtering out all impossible candidates. After verification, an extending technique is further used for adjusting text boundaries so that all the missed text pixels can be well recovered.

3: Feature Extraction

In order to well detect desired text lines from the cluttered background, a novel morphology-based approach will be presented in this section to extract high contrast regions as text candidates. The used high-contrast feature is invariant against text orientations and highly tolerant to noise. Before introducing the proposed method, some morphological operations should be first described.

Let $S_{m,n}$ denote a structuring element with size $m \times n$, where m and n are odds and larger than zero. Let *I(x,y)* denote a gray-level input image. Besides, let $\oplus$ denote a dilation operation, and $\odot$ denote an erosion operation. According to $S_{m,n}$, we define several useful morphological operations as follows:

Closing Operation:

$$I(x,y) \bullet S_{m,n} = (I(x,y) \oplus S_{m,n}) \odot S_{m,n};$$

Opening Operation:

$$I(x,y) \circ S_{m,n} = (I(x,y) \odot S_{m,n}) \oplus S_{m,n};$$

Smoothing Operation:

$$E_{S_{m\times n}}(I(x,y)) = \frac{1}{mn} \sum_{i=-n/2}^{n/2} \sum_{j=-m/2}^{m/2} I(x+i, y+j) S_{m\times n}(i,j);$$

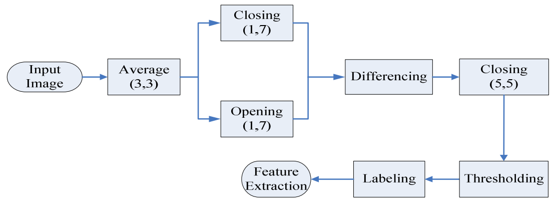


Fig. 2 Flowchart of the proposed method to extract contrast features for text line detection.

As mentioned previously, text region is a pattern which has highly relative contrast to the background. Fig. 2 shows the whole procedure of our morphology-based technique to extract the contrast feature. Firstly, in order to eliminate noise, a smoothing operation with a structure element $S_{3,3}$ is first applied. Then, the closing and opening operations with a structure element $S_{1,7}$ are performed into the smoothed image so that the output images I_c and I_o can be obtained, respectively. Furthermore, for detecting text boundary edges, a differencing operation is further applied into I_c and I_o. In order to make these edges more compactly and closely, a closing operation is then used so that all characters embedded in a text line can form a connected segment. After that, a thresholding operation is used for converting the analyzed image into a binary map. Then, a labeling process is executed to extract the text-analogue segments. After that, a set of potential text lines can be obtained from further verification.

In **[9]**, Hasan and Karam also used several morphological operators to detect text lines. However, there are many differences between our method and their approach. Firstly, they used only the dilation and erosion operations to extract edges as text features. Instead of using these operations, we use the closing and opening operators to detect text lines. In addition, we use a horizontal structure element to extract contrast feature. Compared with their method, our approach performs more robustly in the abilities to deal with noise, cluttered background, and text fragmentation. Secondly, their approach used the closing operation after differencing. But our method uses the closing operation before differencing. Our approach will make text features more compactly and closer to each other. Thirdly, their approach did not discuss the problems when text lines are fragmented, missed, or skewed. All these problems will lead to the failure of text line detection and will be tackled later in this paper.

4: Text Candidate Selection and Merging

In Section 3, a novel morphology-based scheme has been proposed to extract high-contrast regions as potential text segments. However, a text line is not always horizontally aligned and sometimes fragmented into several small segments. In order to detect a skewed text line, in what follows, a moment-based method is first used for estimating its orientations. Then, a merging technique is proposed to link all the missed fragments.

4.1: Moment-based Orientation Estimation

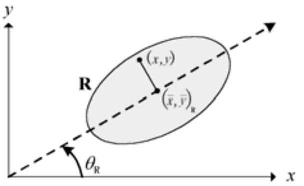


Fig. 3 The gravity center $(\bar{x},\bar{y})_R$ and orientation θ_R of an object R.

Given a binary region $R(x,y)$, the central moments of R can be defined as

$$(\mu_{p,q})_R=\sum_{(x,y)\in R}(x-\bar{x})^p(y-\bar{y})^q,$$

where $(\bar{x},\bar{y})=(\frac{1}{|R|}\sum_{(x,y)\in R}x,\frac{1}{|R|}\sum_{(x,y)\in R}y)$ and $|R|$ is the area of R. Here, if a pixel (x, y) belongs to R, the value of $R(x,y)$ is one; otherwise, its value is zero. Then, as shown in Fig. 3, the orientation θ_R of R can be obtained using the equation:

$$\theta_R=arg\ \min_{\theta}\sum_{(x,y)\in R}\left[(x-\bar{x})\sin\theta-(y-\bar{y})\cos\theta\right]^2. \quad (1)$$

Setting t to zero, we can get

$$\theta_R=\frac{1}{2}\tan^{-1}\left[\frac{2\mu_{1,1}}{\mu_{2,0}-\mu_{0,2}}\right]. \quad (2)$$

Then, according to Eq.(2), the orientation of R can be obtained.

4.2: Text Candidate Selection

Let R be a potential text line extracted from our morphology-based scheme. Due to noise, many impossible non-text regions will be also extracted. This section will use a rule-based scheme to coarsely eliminate impossible candidates. Then, in Section 5, a finer verification scheme will be proposed to more accurately verify all the remained text segments. Assume that R^{θ_R} is the rotated version of R with its orientation θ_R. In addition, $w_{R^{\theta_R}}$ and $h_{R^{\theta_R}}$ denote the width and height of R^{θ_R}, respectively. Since a text line has a longer width than its height, the first rule requires the ratio between $w_{R^{\theta_R}}$ and $h_{R^{\theta_R}}$ being larger than 1.5. In addition, the density of R should be larger enough, *i.e.*, $den=\text{Area of } R \big/ w_{R^{\theta_R}}\times h_{R^{\theta_R}}>0.1$. The final rule requires the area of R being not too small. According to the above requirements, R is a text line candidate if it satisfies the following three rules:

Rule 1: *den* should be larger than 0.1;

Rule 2: the ratio between $w_{R^{\theta_R}}$ and $h_{R^{\theta_R}}$ should be larger than 1.5;

Rule 3: the area of R should be larger than a threshold, *i.e.*, 2.5 pixels.

4.3: Text Line Merging

After filtering out several impossible text regions using the above rule-based approach, this section propose a novel merging scheme to deal with the problem of text fragmentation. Given two regions R_i and R_j, if they belong to the same text line, their heights and orientations should be similar. In addition, their centroids and major axes are close to each other. Therefore, if R_i and R_j belong to the text line, the first criterion requires the orientation difference between them being less than 10°, *i.e.*,

$$d_\theta(R_i,R_j)=\min(|\theta_{R_i}-\theta_{R_j}|,|\theta_{R_i}-\theta_{R_j}+360^\circ|,|\theta_{R_i}-\theta_{R_j}-360^\circ|)<10^\circ, \quad (3)$$

where θ_{R_i} and θ_{R_j} are the major orientations of R_i and R_j, respectively. In addition, the heights of R_i and R_j should be similar enough, *i.e.*,

$$d_h(R_i,R_j)=\frac{2|h_{R_i^{\theta_{R_i}}}-h_{R_j^{\theta_{R_j}}}|}{h_{R_i^{\theta_{R_i}}}+h_{R_j^{\theta_{R_j}}}}<0.15. \quad (4)$$

Let Cen_{R_i} and Cen_{R_j} be the centroids of R_i and R_j, respectively. The third criterion regions Cen_{R_i} and Cen_{R_j} being closer to each other; that is,

$$|\mathrm{Cen}_{R_i}-\mathrm{Cen}_{R_j}|<2(h_{R_i^{\theta_{R_i}}}+h_{R_j^{\theta_{R_j}}}). \quad (5)$$

Let L_R be the longest axis of R denoted by this equation: $y=m_Rx+b_R$. Then, the distance between the major axes L_{R_i} and L_{R_j} of R_i and R_j can be defined as follows:

$$d_L(R_i,R_j)=\frac{|y_{Cen_{R_j}}-m_{R_i}x_{Cen_{R_j}}-b_{R_i}|}{2\sqrt{1+m_{R_i}^2}}+\frac{|y_{Cen_{R_i}}-m_{R_j}x_{Cen_{R_i}}-b_{R_j}|}{2\sqrt{1+m_{R_j}^2}}, \quad (6)$$

where x_{Cen_R} and y_{Cen_R} are the coordinates of Cen_R in the x and y directions. The fourth criterion requires the distance $d_L(R_i,R_j)$ being less than 5 pixels,

$$d_L(R_i,R_j)<5. \quad (7)$$

Based on Eqs.(3)-(5), and (7), we can determine whether R_i and R_j should be merged or not. Thus, different text candidates can be well selected for further verification.

5: Text Line Verification

Once all potential text lines have been extracted, this section will propose a finer verification process for verifying all the remained text candidates.

5.1: *X*-projection

Before verification, each gray text region should be binarized into a binary map using a threshold T_R. In this paper, the "*minimum within-group variance*" dynamic thresholding method [15] is adopted for finding T_R. Then, given a text candidate *R*, after binization an *x* projection technique is used to find its various text geometries. In practice, for all characters embedded in *R*, they should satisfy the following requirements:

A1: their widths should be similar;

A2: their heights should be similar;

A3: their centers are aligned along a straight line.

The projection technique can well separate *R* into different characters if *R* is a real text line. The technique tries to project all text pixels of *R* on its longest axis and then accumulates the number of character pixels column by column. Like Fig. 4, (b) is the result of *x*-projection got from (a). The minimum valleys can provide important information for separating *R* into different characters C_i like 'g', 'u', 'i', 'd', and 'e'. Assume that $\overline{W_R}$ and $\overline{h_R}$ are the average width and height of these characters C_i in R. Then, the width and height variances of characters in R can be calculated, respectively, as follows

$$\sigma_{W,R}^2 = \frac{1}{N_R^C} \sum_{C_i \in R} \left(W_{C_i} - \overline{W_R} \right)^2 \text{ and } \sigma_{h,R}^2 = \frac{1}{N_R^C} \sum_{C_i \in R} \left(h_{C_i} - \overline{h_R} \right)^2, \quad (8)$$

where N_R^C is the number of characters in *R*. In addition, all the character centers in R form a line and satisfy the equation:

$$y = m_R^C x + b_R^C. \quad (9)$$

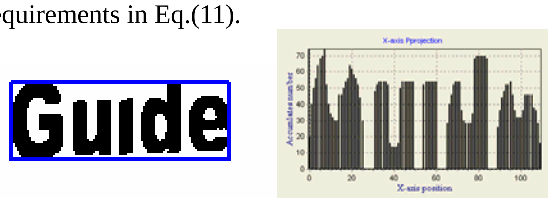
The values of m_R^C and b_R^C can be easily obtained using a line fitting technique [16]. Then, the linearity of *R* can be measured by:

$$L_{inearity(R)} = \frac{1}{\left| N_R^C \right|} \sum_{C_i \in R} \frac{\left| y_{C_i} - m_R^C x_{C_i} + b_R^C \right|}{\sqrt{\left(m_R^C \right)^2}}. \quad (10)$$

Thus, if R is a text line, it should satisfy

$$\sigma_{R,w}^2 < T_w,\ \sigma_{R,L}^2 < T_h, \text{ and } linearity(R) < T_l. \quad (11)$$

The values of T_w, T_h, and T_l can be obtained from thousands of training text samples. Then, different text lines can be correctly verified if they satisfy all requirements in Eq.(11).



(a) (b)

Fig. 4 Character analysis. (a) Original image; (b) Result of the *x*-projection obtained from (a).

5.2: Text Line Recovery

Due to noise or lighting changes, some characters still would be missed from the above text analysis. Therefore, this section will propose a novel recovery algorithm for adjusting the boundaries of a text line *R* so that its all missed data can be well recovered.

Assume that *R* is the incomplete text region and R^{θ_R} is its rotated version with the major orientation θ_R of *R*. The whole recovery algorithm is performed on R^{θ_R} by considering the major axis of *R* as the *x* direction and the center of R^{θ_R} as the original. With the transformation, all the text characters in R^{θ_R} will be horizontally aligned. Then, a horizontally extending technique can be used for recovering *R* to a complete one. Let l_{R^θ}, r_{R^θ}, t_{R^θ}, and b_{R^θ} denote the most left, right, top, and bottom coordinates of R^{θ_R} in the *x* and *y* directions, respectively. In addition, T_R is the threshold used to binarize R to a binary map. The recovery algorithm is iteratively performed. At each iteration, we create two new regions extended from both sides of R^{θ_R} with the same height of R^{θ_R} and a fixed width. The retrieved region will be the final desired text line. In what follows, the details of text line recovery algorithm are described.

Text Line Recovery Algorithm

Input: a region *R*, the average character width $\overline{w}_{R,C}$ and height $\overline{h}_{R,C}$ in *R*, the threshold th_R to binary *R*;

Output: a new recovered region $\overline{R}$.

Step 1: According to the major orientation of *R*, obtain its rotated version R^{θ_R}.

Step 2: Obtain the boundary coordinates of R^{θ_R}, *i.e.*, l_{R^θ}, r_{R^θ}, t_{R^θ}, and b_{R^θ}, respectively by considering the major axis of R^{θ_R} as the *x* direction.

Step 3: // left extension

S3.1: Create a new region R_{left}^{New} with the coordinates: l = max (0, l_{R^θ} - 5 $\overline{w}_{R,C}$), r = l_{R^θ}, t = max (0, t_{R^θ} - $\overline{h}_{R,C}$ /5), and b = min ($b_{R^\theta} + \overline{h}_{R,C}$ /5, H_I).

S3.2: Binarize R_{left}^{New} using the threshold th_R.

S3.3: Check whether there are isolated characters in R_{left}^{New} using a connected component analysis.

S3.4: If any isolated characters exist, l_{R^θ} = the most left x coordinate of pixels in them and go to Step 3; otherwise, go to step 4.

Step 4: // right extension

S4.1: Create a new region R_{right}^{New} with the coordinates: l = r_{R^θ}, r = min(r_{R^θ}+5$\overline{w}_{R,C}$, W_I), t = min (t_{R^θ} - $\overline{h}_{R,C}$ /5, 0), and b = max ($b_{R^\theta} + \overline{h}_{R,C}$ /5, W_I).

S4.2: Binarize R_{right}^{New} *using* the threshold th_R.

S4.3: Check whether *there* are isolated characters in R_{right}^{New} using a connected component analysis.

S4.4: If any isolated characters exist, r_{R^θ} = the most right *x* coordinate of pixels in them and go to Step 4; otherwise, go to step 5.

Step 5: Obtain $\overline{R}$ with the new boundary coordinates: l_{R^θ}, r_{R^θ}, t_{R^θ}, and b_{R^θ}.

6: Experimental Results

For well testing our method, images having various appearance changes like contrast changes, complex backgrounds, lightings, different fonts, and sizes were used. Fig. 5 shows the case of text line extraction when text regions have low contrast to the background. Fig. 6 is the detection result when text lines were embedded in a cluttered background. In (a) and (b), the backgrounds are colorful and with textured background. Clearly, no matter what the background is, our method works very well to detect all desired text regions.

(a) (b) (c)

Fig. 5 Results of text line extraction when text regions had low contrasts to the background.

In real conditions, the changes of font and size will also affect the accuracy of text line detection. Fig. 7 shows the results of text line detection when different fonts and sizes were found in the processed images. A more challenging work is to detect text line from video sequences since text lines have quite distortions after compression. Fig. 8 shows the results of text line detection when video frames were handled. Fig. 9 shows another difficult task when text lines had different orientations. Clearly, no matter what cases are handled, all the described text lines can be well extracted using our proposed methods.

(a) (b)

Fig. 6 Results of text line detection when cluttered backgrounds were handled.

(a) (b)

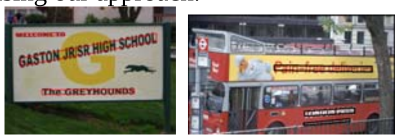
Fig. 7 Results of text line extraction when text lines had different fonts and sizes.

In another set of experiments, we also compared our proposed approach with two other approaches, *i.e.*, Hasan and Karam [9] and the maximum gradient approach [10]. Fig. 10 shows the comparison results among the three methods. Hasan and Karam's method [9] also used several morphological operators to detect text lines. However, they used only the dilation and erosion operators to extract text features. In addition, their verification algorithm depends strongly on the geometries and sizes of individual characters. Therefore, their method is very sensitive to noise, cluttered backgrounds, and text fonts. As to Wong and Chen's approach [10], their maximum gradient technique is sensitive to complicated background and text fonts so that many incorrect regions would be extracted and lead to may false alarms. As to our method, even though the processed text lines have different fonts and sizes, it still works very well to detect all desired text lines.

(a) (b) (c)

Fig. 8 Results of text line extraction when video frames were handled.

Fig. 10 shows another set of performance comparison when skewed text lines were handled. (a) is the detection result using Hasan and Karam's approach [9]. Their verification method is easily affected by noise and doesn't deal with fragmented text lines. Therefore, text fragments tended to be detected. (b) is the result obtained using Wong and Chen's method [10]. Their method used a horizontally scanning method to search all regions with local maximum gradients as potential text candidates. The local maximum gradient is very sensitive to complicated background and text orientations. Therefore, although the text lines were correctly located, many false alarms were also detected. However, since our method is invariant to text orientations, all desired text lines were successfully detected using our approach.



(a) (b)

Fig. 9 Results of text line extraction when skewed text lines were handled.

For comparing the accuracy of our approach to other methods, we collected 254 normal text lines and 106 skewed text lines in the simple background category. In addition, 52 normal text lines and 49 skewed text lines were collected in the cluttered background category. Table 1 shows the overall accuracy comparisons among the above three methods. Since Hasan and Karam's method didn't well deal with fragmented text lines, their method performed the worst among the three methods. As to Wong and Chen's approach [10], they used a line scanning method to find pixels having maximum gradients as potential text lines. Many false alarms would be caused when complicated backgrounds or skewed text lines were handled. In this experiment, our method performed the best among all the above comparisons. In addition to the recall-precision analysis, we also use the well-known probability of error (*PE*) to estimate the error rate of each method. Table 2 summaries the PE values of the three methods. Clearly, our method got the lowest error rates. All the experiments have proved the superiority of our proposed method in text line detection.

(a) (b) (c)

Fig. 10 Comparison result of text line detection. (a) Result of Hasan and Karam [9]. (b) Result of Wong and Chen [10]. (c) Result of our proposed method.

Table 1 Recall and precision analysis

Methods	Simple background		Cluttered background	
	Recall	Precision	Recall	Precision
Our method	95.3%	99.4%	96.0%	95.1%
Hasan [9]	61.6%	95.7%	45.5%	79.3%
Wong [10]	92.5%	79.1%	83.2%	48.6%

Table 2 PE performance analysis

Methods	Probability of Error
Our method	0.022
Hasan and Karan [9]	0.226
Wong and Chen [10]	0.1595

7: Conclusions

The contributions of this paper are summarized as follows:

1. A morphology-based method was proposed for extracting high contrast areas as text line candidates. The feature is invariant to different image visual changes including lighting, rotation, and translation.
2. An *x*-projection was proposed for extracting different text properties from a text line. Since the projection was performed adaptively according to text line orientation, text lines even skewed can be well verified and detected.
3. A recovery algorithm was proposed for reconstructing a complete text line from its fragmented segments. Thus, the proposed scheme has high tolerances to noise.

Reference

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

[2] Z. Dekun and Y. Q. Shi, “Formatted text document data hiding robust to printing, copying and scanning,” *IEEE International Symposium on Circuits and Systems*, vol. 5, pp.4971-4974, 2005.

[3] T. Sato, *et al.*, “Video OCR for digital news archive,” *1998 IEEE International Workshop on Content-Based Access of Image and Video Database*, pp.52-60, Bombay India, 1998.

[4] M. R. Lyu, J. Song, and M. Cai, “A comprehensive method for multilingual video text detection, localization, and extraction,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 15, no. 2, pp.243-255, 2005.

[5] N. Zhang, *et al.*, “Modified LZW algorithm for efficient compressed text retrieval,” *International Conference on Information Technology, Coding and Computer*, pp.224-228, 2004.

[6] A. Hoogs, J. Mundy, and G. Cross, “Multi-modal fusion for video understanding,” *Proceeding 30th Applied Imagery Pattern Recognition*, pp.103-108, 2001.

[7] Y. Zhong, K. Karu, and A. K. Jain, “Locating text in complex color images,” *Pattern Recognition*, vol. 28, no. 10, pp.1523-1536, Oct. 1995.

[8] R. Lienhart and F. Stuber, “Automatic text recognition in digital videos,” *Proceeding of SPIE*, pp.180-188, 1996.

[9] Y. M. Y. Hasan and L. J. Karam, “Morphological text extraction from images,” *IEEE Transaction on Image Processing*, vol. 9, no.11, pp.1978-1983, 2000.

[10] E. K. Wong and M. Chen, “A new robust algorithm for video text extraction,” *Pattern Recognition*, vol. 36, no. 6, pp.1397-1406, 2003.

[11] B. Sin, S. Kim, and B. Cho, “Locating characters in scene images using frequency features,” *International Conference on Pattern Recognition*, vol. 3, Quebec, Canada, pp.489–492, 2002.

[12] W. Mao, F. Chung, K. Lanm, and W. Siu, “Hybrid Chinese/English text detection in images and video frames,” *International Conference on Pattern Recognition*, vol. 3, Quebec, Canada, pp.1015–1018, 2002.

[13] K. I. Kim, K. Jung, S. H. Park, and H. J. Kim, “Support vector machine-based text detection in digital video,” *Proceeding of Pattern Recognition*, pp.527–529, 2001.

[14] C. Xiangrong and A. L. Yuille, “Detecting and reading text in natural scenes,” *Proceeding of the IEEE Computer Vision and Pattern Recognition*, vol.2, pp.366-373, 2004.

[15] M. Sonka, V. Hlavac, and R. Boyle, *Image Processing, Analysis, and Machine Vision*, London, U. K., Chapman & Hall, 1993.

[16] W. H. Press, S. A. Teukolsky, W.T. Vetterling, and B. P. Flannery, *Numerical Recipes in C*, 1992.