

2D and 3D Video Scene Text Classification

Jiamin Xu¹, Palaiahnakote Shivakumara², Tong Lu¹, and Chew Lim Tan³

¹ National Key Lab for Novel Software Technology, Nanjing University, Nanjing, China

² Faculty of Computer Science and Information Technology, University of Malaya, Kuala Lumpur, Malaysia

³ School of Computing, National University of Singapore

superxjm@yeah.net, hudempsk@yahoo.com, lutong@nju.edu.cn, and tancl@comp.nus.edu.sg

Abstract—Text detection and recognition is a challenging problem in document analysis due to the presence of the unpredictable nature of video texts, such as the variations of orientation, font and size, illumination effects, and even different 2D/3D text shadows. In this paper, we propose a novel horizontal and vertical symmetry feature by calculating the gradient direction and the gradient magnitude of each text candidate, which results in Potential Text Candidates (PTCs) after applying the k-means clustering algorithm on the gradient image of each input frame. To verify PTCs, we explore temporal information of video by proposing an iterative process that continuously verifies the PTCs of the first frame and the successive frames, until the process meets the converging criterion. This outputs Stable Potential Text Candidates (SPTCs). For each SPTC, the method obtains text representatives with the help of the edge image of the input frame. Then for each text representative, we divide it into four quadrants and check a new Mutual Nearest Neighbor Symmetry (MNNS) based on the dominant stroke width distances of the four quadrants. A voting method is finally proposed to classify each text block as either 2D or 3D by counting the text representatives that satisfy MNNS. Experimental results on classifying 2D and 3D text images are promising, and the results are further validated by text detection and recognition before classification and after classification with the existing methods, respectively.

Keywords—Video text frames, Horizontal and vertical symmetry, Video potential text candidates, Dominant potential text candidates, 2D and 3D text video classification

I. Introduction

Convergence of the technologies from computer graphics, computer vision, multimedia and other related fields has enabled the development of advanced types of visual media and devices such as 3D video (3DV) and free viewpoint video (FVV), which expand user's sensation beyond what is offered by the traditional 2D video [1]. As a result, in the future, video simultaneously containing 2D and 3D texts will become quite common and we can see 3D TV at everyone's home. Currently, Google Street View and iTowns have generated a huge amount of images and videos that contain both 2D and 3D scene texts [2]. Many potential applications, such as traffic monitoring, geographic information systems, road navigation and scene understanding can use the videos captured by the iTown imaging vehicle on which a camera is fixed at its top. For instance, to locate the address of a store, the user is offered 3D view of the location, created by suitable projection of pre-stitched image mosaics. A project like iTown could easily generate hundreds of thousands of such mosaics in a single city. The manual annotation of all these images with the visible textual information would be very time consuming and probably impractical [2]. Therefore, there is a great demand of automatic algorithms for detecting and recognizing both 2D and 3D texts with a good accuracy.

There are methods in literature which work well for video or images containing 2D texts [3]. However, when given video which contain both 2D and 3D texts as an input, the performance of the

methods degrades drastically [4, 5] because of the variations in edge pattern and strength. For instance, In Figure 1, (a) shows 2D characters chosen from video, (b) shows a 3D character from video but it is on its frontal view, and (c) shows a 3D character from video from its side view, where we can see 3D effect in the form of extra edges. It is observed from Figure 1 that OCR fails to recognize the 3D character on side view. Therefore, we consider all the 2D texts from video and the 3D texts on frontal view as 2D texts, while the 3D texts on side view from video are considered as 3D texts. Then there are two ways to achieve the accuracy: (1) developing a unified algorithm which works well for both 2D and 3D texts in video, (2) classifying 2D and 3D texts in video such that a separate algorithm can be proposed for 3D texts and the existing methods which give good accuracy for 2D text can be used separately. In this work, we focus on the second way to improve the accuracy because this way can make use of the existing 2D text detection methods, rather than developing a unified method which will be relatively hard. However, all the 3D texts that appear on iTown and urban videos are generally scene texts. This makes the problem of classification challenging and complex because scene text is a part of the image captured by camera and it poses virtually unlimited range of sizes, shapes and colors [3]. For comparison, graphics texts are artificially added to video frames to supplement visual or audio content. Therefore, the presences of both graphics texts and scene texts in a video frame bring another difficulty in classifying 2D and 3D video texts.

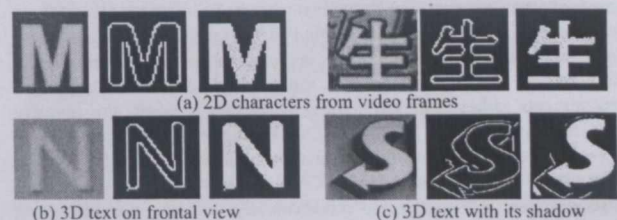


Fig. 1. Illustration of 3D effect in video text recognition: for characters “M”, “生”, “N”, the Tesseract OCR recognizes as “M”, “生” and “N”, respectively, while for “S” the OCR cannot recognize it correctly due to the extra edges caused by the 3D effect.

In literature, there are a plenty of text detection methods [6-8] by using connected component analysis, texture analysis, edge and gradient analysis. However, these methods generally consider the videos containing only 2D texts but without 3D ones. We can also see several methods which use temporal information in video for text detection [9-13]. For example, Bouaziz et al. [12] proposed a similarity criterion to find text appearance based on frame differences. However, the similarity criterion requires a threshold value to identify the sudden differences, and the focus of the method is only on 2D graphics text detection but not 3D text detection. Huang et al. [13] proposed automatic detection and localization of natural scene texts in video based on edge and stroke details. Similarly, these features may work well for 2D scene texts but not for 3D ones because the latter may not

This process basically finds the stable PTCs which are presented in all the consecutive frames until the iterative process terminates. In this way, the method continues the iterative process to filter out those unstable PTCs until the converging criterion is met. The final stable PTCs can be seen in Figure 6(f). This is nothing but getting DP_{final} by the iterative process. To define the terminating condition, we estimate the proximity matrix which indicates the distances between the PTCs in $DP_1, DP_2, \dots, DP_{final}$ as defined in equation (4). This is valid because the PTCs in the text area are more or less stable than the non-text PTCs in consecutive frames. It is observed that as iteration increases the standard deviation of the proximity matrix of DP_i decreases and after certain iterations, the standard deviations of the previous iteration and the current iteration have become almost equal as defined in equation (5). Figure 7 shows that after the 9th iteration the curve becomes flat from iteration 10 to iteration 12. This is the terminating point as defined in equation (5). It is because the unstable PTCs have been eliminated at the iterations. The outputs are called Stable Potential Text Candidates (SPTCs). Similarly, the same procedure is used for 3D text video to obtain SPTCs.

$$PM_i = \begin{bmatrix} Dist(P_1, P_1) & \dots & Dist(P_1, P_K) \\ \vdots & \ddots & \vdots \\ Dist(P_K, P_1) & \dots & Dist(P_K, P_K) \end{bmatrix} \quad (4)$$

$$DP_i = \{P_1, P_2, \dots, P_K\}, K = \|DP_i\|$$

$$\|Std(PM_{i+1}) - Std(PM_i)\| < 0.2 \quad (5)$$

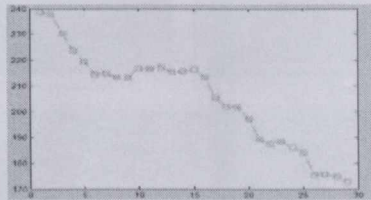


Fig. 7. Terminating condition for iterative process (Number of iterations vs standard variance of PM_i)

D. 2D and 3D Text Block Classification

For each SPTC in Figure 6(f), the method extracts edge components from the Canny edge image of the input frame to study the behavior of the SPTC, which results in Text Representatives (TPs). Figure 8(a) shows the TPs that are obtained from the Canny edge image (Figure 8(b)). Similarly, for 3D text frame, the method obtains TPs as shown in Figure 8(c), which are obtained from the Canny edge image in Figure 8(d). We use grouping criterion based on the nearest neighbor technique to merge all the TPs by referring the Canny edge image of the input frame. Components that have less than three pixels are eliminated because they do not contribute to text. More details for boundary growing and merging can be found in [14]. The output of this step is considered as text block segmentation. In this way, the method segments text lines of both graphics and scene texts irrespective of 2D and 3D.

For each segmented text block, the method studies the behavior of each TP in both 2D and 3D text blocks to classify it as either a 2D or a 3D block by using stroke width distances. The method divides the whole TP into four quadrants (top left, bottom left, top right, bottom right) at the center of the component as shown in Figure 9 (see the center with red color lines). Then for each TP of each quadrant, the method finds stroke width distance as suggested in [15]. The method performs the histogram operation on stroke width distances for each quadrant as shown in Figure 9, and chooses the dominant stroke width distances (highest frequency) from each histogram as the shown values. The values 4, 4, 7, and 7 are representing dominant stroke width distances of the four quadrants in clockwise direction, respectively. As we discussed in the proposed methodology section, if a character is from a 2D text frame then it must satisfy the symmetry like human face

else not always. To extract this property, we propose Mutual Nearest Neighbor Symmetry (MNNS) to classify the TPs as representing 2D text or 3D text. The MNNS procedure first selects the Maximum and the Minimum from the four dominant stroke width distances (7 and 4) as defined in equation (6) and (7), respectively. Then it compares the remaining two dominant stroke width distances (7 and 4) with the Maximum and the Minimum, and classifies them into a Maximum cluster if the distance is close to the Maximum distance, otherwise classifies it into a Minimum cluster. This results in two equal clusters containing an equal number of distance values. If the dominant stroke width distances of TP satisfy MNNS then it is considered as a 2D TP else a 3D TP. In order to classify the whole text block as 2D, we consider a voting criterion that counts the number of 2D TPs and 3D TPs in text blocks of the images. If the count which represents 2D TPs is more than the count which represents 3D TPs in the text block, then the method considers the text block as a 2D text one else a 3D text one. Figure 10 shows the TPs that satisfy MNNS (red color components) for 2D text image, and the TPs that satisfy MNNS for 3D text image (red color components), respectively. It is noticed from Figure 9 that the number of the TPs that satisfy MNNS are more in the case of 2D text images while less in the case of 3D text images. Therefore, the two text blocks in the first image are classified as 2D and the three text blocks in the second image are classified as 3D. The main advantage of this method is that it can classify 2D and 3D texts even when a single frame contains both 2D and 3D text blocks.

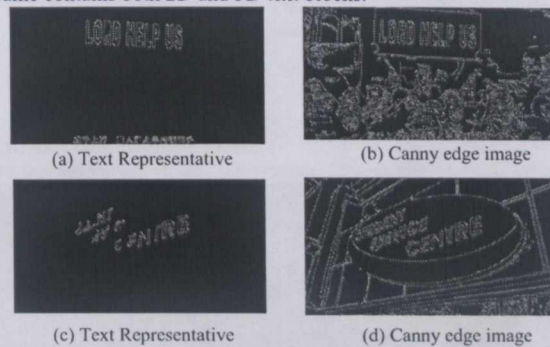


Fig. 8. Text Representative and Canny edge image for 2D and 3D frame

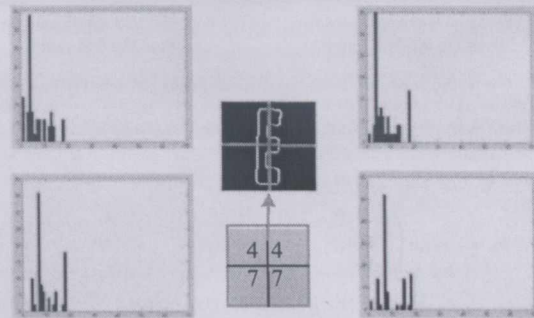


Fig. 9. Histograms for dominant stroke width distances for four quadrants

$$DSW_{max} = \max\{dsw(P_i) \mid i = 1,2,3,4\} \quad (6)$$

$$DSW_{min} = \min\{dsw(P_i) \mid i = 1,2,3,4\} \quad (7)$$

For each $P_i (i = 1,2,3,4)$,

$$\begin{cases} \text{if } \|dsw(P_i) - DSW_{min}\| \geq \|dsw(P_i) - DSW_{max}\|, \\ \quad \text{Add } DSW_i \text{ to Cluster}_{max} \\ \text{if } \|dsw(P_i) - DSW_{min}\| < \|dsw(P_i) - DSW_{max}\|, \\ \quad \text{Add } DSW_i \text{ to Cluster}_{min} \end{cases} \quad (8)$$

provide a constant stroke width or edge density as expected. Therefore, the method is not suitable for 3D text detection in video.

Hence, in this work, we propose a novel method for 2D and 3D text classification to improve text detection and recognition accuracy.

II. Proposed Method

It is noted from the work presented in [14] for video text detection that gradient operation on video frames is useful for increasing the contrast of text pixels. As motivated, we initially perform the gradient operation to enhance text pixels in this work. After text pixels are enhanced, we propose k-means clustering algorithm with $k=3$ to remove noisy pixels that have low gradient values as they may not contribute to text. The rest two clusters which have high mean values are considered as text clusters. This results in text candidates for each video frame. Due to complex background of video, there are chances of misclassifying false text candidates as text candidates. As we are inspired by the observation that characters usually have double edges with a constant stroke width distance [15], we propose a novel horizontal and vertical symmetry feature based on the gradient directions and the gradient magnitudes of each text candidate. For each canny edge pixel, the method [15] moves in its gradient direction until it reaches another edge pixel. The distance between the starting edge pixel and the reached pixel is called stroke width distance. As mentioned, the method explores constant stroke width property to identify text candidates. Then the method uses characteristics of text candidates for final text detection. In summary, the method is developed for text detection but not classification. The symmetry extracts the two facts that double edges have parallel directions, and text candidates have a high gradient magnitude at near edges and on edges but a low magnitude in between parallel edges [14]. In other words, our horizontal symmetry uses the gradient direction and the magnitude values between edges, while the vertical symmetry uses the direction of parallel edges, which is perpendicular to the gradient direction of a text candidate. This outputs Potential Text Candidates (PTCs).

To validate the PTCs, we explore temporal redundancy in video. It is observed that texts in video usually have constant movements along a particular direction while the background does not as stated in several methods [9-13]. Inspired by this observation, we propose an iterative method that studies the neighbor information of each PTC in consecutive frames to identify stable PTCs. The reason for considering neighbor information of PTCs is to tolerate the arbitrary text movements because sometimes, video may contain arbitrary text movements rather than static movements. As a result, the iterative process gives stable PTCs by finding the PTCs which exist in consecutive frames, until the iterative process stops. Thus the iterative process in the study of stable PTCs from consecutive frames serves two purposes: (1) it helps in automatically deciding the number of interested frames out of 30 frames per second because it is a research issue for the existing methods [9-13] which assume a fixed number of five, ten etc., (2) it helps in identifying stable PTCs in spite of arbitrary text movements by throwing out non-stable ones which are likely non-text components. We call the output of the iterative process as Stable Potential Text Candidates (SPTCs).

For each SPTC, we extract its edge components, which we call text representatives, from the Canny edge image of the input frame. For each text representative, we divide the whole representative into four quadrants and then extract dominant stroke width distances for each quadrant. The stroke width distances are calculated according to the method in [15]. It is true that characters generally exhibit symmetry like faces of human when we divide into equal halves at the center point. Based on this observation and as it is explored in [16] for defining Mutual Nearest Neighbor Symmetry (MNNS) at block level

to classify text frames using wavelet and moments features, we further propose MNNS for classifying 2D or 3D text representatives. Namely, if the dominant stroke width distances of each quadrant form clusters which can satisfy MNNS then it is considered as a 2D text representative else a 3D text representative. This is valid because we can expect such symmetry for 2D characters due to double edges and parallel edges but for 3D character, it does not due to extra edges and the loss of edges caused by perspective distortion and complex background.

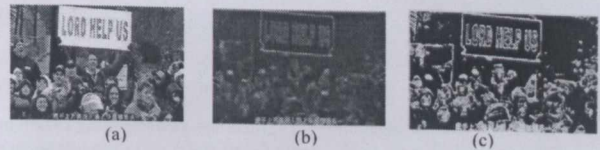


Fig. 2. Text candidates for 2D text frame: (a) Gray frame, (b) Gradient first frame (c) Output of k-means clustering algorithm (binary)

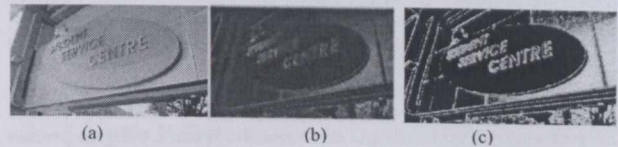


Fig. 3. Text candidates for 3D text frame: (a) Gray frame, (b) Gradient frame and (c) Output of k-means clustering algorithm (binary)

A. Text Candidates Selection

For the first video frame as shown in Figure 2(a), where 2D scene texts are embedded with different orientations and graphics texts are at the bottom of the frame, the method obtains the gradient image as shown in Figure 2(b), from which we can notice that text pixels are brightened compared to the pixels in Figure 2(a). Therefore, the method applies k-means clustering algorithm with $k=3$ on the gradient image shown in Figure 2(b) to classify text candidates as shown in Figure 2(c), where one can see all the high contrast pixels are classified as text candidates including text pixels. In the same way, for the video frame shown in Figure 3(a) where 3D scene texts appear on a building background with different orientations, the method obtains the gradient image as shown in Figure 3(b), and the text candidates by the k-means clustering algorithm on the gradient image in Figure 3(b) are shown in Figure 3(c). It is observed from Figure 2(c) and Figure 3(c) that the 3D texts appearing in Figure 3(c) are brighter than the 2D texts in Figure 2(c). This is due to extra edges and thickness of the strokes given by the 3D effect as illustrated in Figure 1. As a result, there is no guarantee that a 3D character always exhibits symmetry like human face and provides parallel edges as in 2D texts. This observation leads to exploring the new features like symmetry and direction of parallel edges to classify 2D and 3D texts in this work.

B. Horizontal and Vertical Symmetry for Potential Text Candidates

It is observed that the method presented in Section A misclassifies false text candidates as text candidates as shown in Figure 2(c) and Figure 3(c). Therefore, we propose a novel horizontal and vertical symmetry feature for identifying PTCs. For each text candidate as shown in Figure 4(a), the method considers a 3×3 window and computes the mean gradient values for the window, which tolerates little distortions and text movements. It moves in both the positive and the negative gradient directions of a text candidate (P0 in Figure 4(a)) and while moving, the method checks the mean gradient values as defined in equation (1) and equation (2) until the condition is met, which we call the horizontal symmetry. This is illustrated in Figure 4(b)

dsw returns the dominant stroke width.

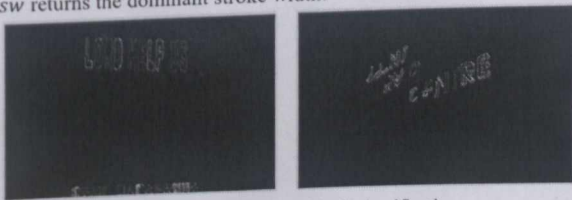


Fig. 10. 2D and 3D text block classification

III. Experimental Results

As it is the first work on classification of 2D and 3D texts in video, there is no benchmark or standard dataset for evaluating the proposed method. There are standard databases, namely, ICDAR and SVT that contain natural scene text but not video text. Therefore, we create video data comprising 500 video clips, which includes 200 3D text video and 300 2D text video clips that are captured by our own video camera at different places such as urban scenes, shops, markets and buildings. This dataset contains the texts of different orientations, scripts, fonts, font sizes, etc. Each video clip may last less than ten seconds. To evaluate the method, we consider the measures, namely, recall, precision and F-measure for text line segmentation, classification rate for the classification of 2D and 3D text frames and character recognition rate for the recognition results. These are the standard measures to evaluate the methods. To show the effectiveness of the method, we implement two existing methods [12, 13] which use temporal redundancy, edges and stroke information. Similarly, to validate the classification in terms of recognition rate, we implement three baseline binarization methods that are Niblack [17] and Sauvola [18] methods which use thresholds for binarizing the images, and one more recently developed method [19] for video text binarization based on Wavelet-Gradient Fusion (WGF) criterion.

A. Experiments for Text Block Segmentation

Sample qualitative results of the proposed and the existing methods [12, 13] for text block segmentation are shown in Figure 11, where the first row shows the input frames having 3D text, 2D text and 2D text of Chinese script. The second row shows the results of the proposed method which successfully detects almost all the texts in the input frames. The third and the fourth rows show the results of Huang et al [13]. and Bouaziz et al [12] methods, respectively. The existing methods detect 2D texts well but fail to detect 3D texts. The main reason for the poor accuracy of the existing methods is that the methods developed for 2D text detection but not 3D text detection and the features used are sensitive to 3D texts. The quantitative results of the proposed and existing methods are reported in Table I, where both the existing methods give poor accuracies compared to the proposed method in terms of recall, precision and F-measure. Therefore, it can be concluded that the proposed method outperforms the existing methods for text line segmentation.

TABLE I. TEXT BLOCK DETECTION RESULTS OF THE PROPOSED AND EXISTING METHODS (IN %) ON BOTH 2D + 3D VIDEOS

Method	R	P	F
Proposed method	86.0	83.0	84.5
Huang et al.[13]	57.0	51.5	54.0
Bouaziz et al.[12]	50.0	35.0	41.0

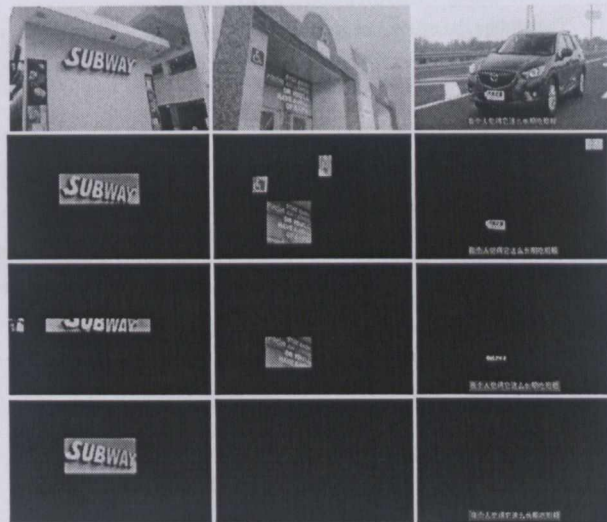
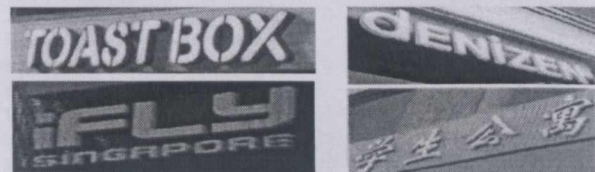


Fig. 11. Sample results of the proposed, Huang et al. [13] and Bouaziz et al. [12]



(a) Sample 2D text lines classified by the proposed method



(b) Sample 3D text lines classified by the proposed method

Fig. 12. Sample 2D and 3D text blocks from our database

B. Experiments for Classification of 2D and 3D Text Blocks

Sample 2D and 3D text blocks that are successfully classified by the proposed method are shown in Figure 12(a) and (b), respectively. It is observed from Figure 12 that the proposed method works well for different types of texts and different scripts. The qualitative results of the proposed method are reported in Table II, where the confusion matrix gives promising results for 2D and 3D text classification. We can also see from Figure 12 that the proposed method classifies both graphics texts (most likely 2D) and scene texts (can be either 2D or 3D) correctly though the text lines are suffering from illumination, orientation, different fonts and contrasts.

TABLE II. CONFUSION MATRIX OF THE CLASSIFICATION METHOD FOR 2D AND 3D TEXT BLOCKS (IN %)

Type	2D text	3D text
2D text	85.5	14.5
3D text	21.0	79.0

where we can see the results P1 and P2 for P0. Then the method moves along the perpendicular direction to the gradient in both the two directions of down and up for P0, P1 and P2. It continues as long as the distance between P1 and P2 gives the same distance as shown in Figure 4(c). Let these pixels be P0U, P0D, P1U, P1D, P2U and P2D, respectively, as shown in Figure 4(d). The method computes the standard deviation of the gradient angles of those points as defined in equation (3). If P0 satisfies equation (3) then it is said to satisfy both the horizontal and the vertical symmetries, and is called as a PTC. All the PTCs from the 2D text frame in Figure 2(c) and the 3D text frame in Figure 3(c) can be seen in Figure 5(a) and (b), respectively, where most of the false text candidates have been removed. However, we can see still few false PTCs due to background complexity.

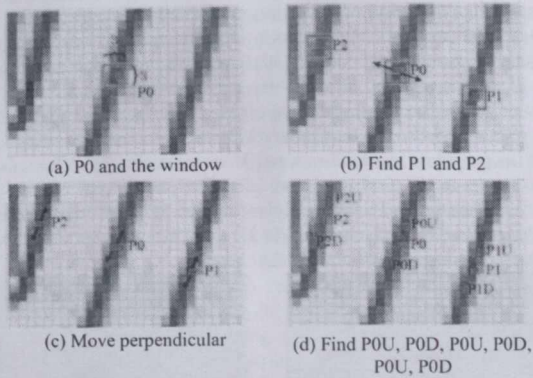


Fig. 4. The procedure of horizontal and vertical symmetry.



Fig. 5. Procedure for Potential text candidates selection (PTC)

More formally, gradient magnitude and mean gradient magnitude can be calculated as below.

Let $G_p :=$ The gradient of pixel P

$$V_p := \text{mean} \left(\begin{pmatrix} GM(P_{i-1,j-1}) & GM(P_{i-1,j}) & GM(P_{i-1,j+1}) \\ GM(P_{i,j-1}) & GM(P_{i,j}) & GM(P_{i,j+1}) \\ GM(P_{i+1,j-1}) & GM(P_{i+1,j}) & GM(P_{i+1,j+1}) \end{pmatrix} \right)$$

GM is the gradient magnitude

$$PriP1 := P1 - G_{P1}, \quad NextP1 := P1 + G_{P1}$$

$$V_{P1} > V_{PriP1} \&\& V_{P1} > V_{NextP1} \quad (1)$$

$$PriP2 := P2 - (-G_{P2}), \quad NextP2 := P2 + (-G_{P2})$$

$$V_{P2} > V_{PriP2} \&\& V_{P2} > V_{NextP2} \quad (2)$$

$\theta_p :=$ The gradient angle of pixel P , $\theta \in (-\pi/2, \pi/2]$

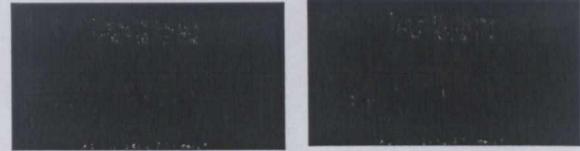
$$\begin{cases} P1, P2, P0U, P0D, P1U, P1D, P2U, P2D \text{ all exist} \\ Std(\theta_{P0}, \theta_{P0U}, \theta_{P0D}, \theta_{P1}, \theta_{P1U}, \theta_{P1D}, \theta_{P2}, \theta_{P2U}, \theta_{P2D}) \\ \in (1,10) \end{cases} \quad (3)$$



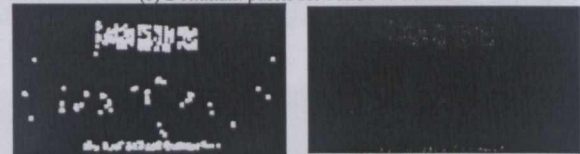
(a) Video sequence t, t+1... t+n



(b) t and t+1 frames as input for first iteration



(c) Dominant pixels for t and t+1 frames



(d) Mask M_{t+1}

(e) DP_1



(e) Stable potential text candidates after meeting converging criterion

Fig. 6. The iterative process for Stable Potential Text Candidates (SPTCs)

C. Temporal redundancy for Stable Potential Text Candidates Selection

It is noted from the results of the previous section that there still exist false PTCs in the resultant images as shown in Figure 5(a) and (b). To validate the PTCs, the method proposes an iterative process which explores temporal redundancy in video to identify Stable Potential Text Candidates (SPTCs). Let $t, t+1, \dots, t+n$ be the video sequence as shown in Figure 6(a). Here n denotes 30 frames per second. Initially, the method considers the first two consecutive frames, say t and $t+1$ as shown in Figure 6(b), and finds the corresponding PTCs as shown in Figure 6(c). For frame $t+1$, the method merges all the PTCs within the defined window of size 11×11 pixels centered at each PTC by a mask M_{t+1} operation. The results can be seen in Figure 6(d). Then the method removes all the other PTCs in t , which are not covered by the mask operation as shown in Figure 6(e), where the remaining PTCs after elimination are drawn. It is observed that the number of the PTCs in Figure 6(e) is less than the number of the PTCs in Figure 6(d). Let the results of filtering PTCs be DP_1 . Similarly, for the second iteration, the method gets the PTCs for the $t+2$ frame and again the same mask M_{t+2} operation is applied to filter out the PTCs in DP_1 . This leads to DP_2 .

C. Validation of Classification by Text Blocks

To validate the effectiveness of the proposed classification method, we compute recall, precision and F-measure after classification that is to give 2D text frames and 3D text frames as the input separately for the existing and the proposed methods. Table I gives the accuracies before classifying 2D and 3D texts, which give overall performances of the existing and the proposed methods. Table III shows the existing methods give better accuracies for 2D text but low accuracies for 3D texts. On the other hand, the proposed method gives better accuracies for both 2D and 3D texts compared to the existing methods. When we compare the accuracies of 2D and 3D texts for the proposed method, we find the accuracy of 3D texts is lower than that of 2D texts. This is because of the loss of information during classification. Therefore, we can assert that the proposed classification method makes difference in improving the accuracy for text detection when data is mixed with 2D and 3D texts.

TABLE III. TEXT BLOCK DETECTION RESULTS OF THE PROPOSED AND EXISTING METHODS (IN %) AFTER CLASSIFICATION

Method	2D Text video			3D Text video		
	R	P	F	R	P	F
Proposed method	89.0	84.0	86.5	81.5	82.0	82.0
Huang et al. [13]	65.0	58.5	61.5	42.0	39.0	41.0
Bouaziz et al.[12]	45.0	56.0	50.0	38.0	31.0	34.0

D. Validation of Classification by Recognition

To know the effectiveness of the proposed classification method in terms of the recognition rates before classification and after classification, we compare two baseline thresholding binarization methods [17, 18] and the recent method [19] of video text binarization. The method binarizes images and passes them to tesseract (Google OCR) [20] which is publicly available to calculate character recognition rate. The results are reported in Table IV, from which one can notice that the all the three binarization methods give poor accuracies for 3D text after classification compared to 2D text. The results of before classification are higher than those of after classification. The reason for the poor accuracies is that the methods are developed for 2D text binarization but not for 3D text. In addition, the methods require high contrast text images but not like video frames. Another reason may be the inherent limitations of the OCR which accepts only particular fonts, size, and clear shape characters. Hence, the classification is necessary to improve the accuracy.

TABLE IV. CHARACTER RECOGNITION OF THE BINARIZATION METHODS BEFORE AND AFTER CLASSIFICATION (IN%)

Methods	Before classification	After classification	
	2D + 3D	2D	3D
WGF [19]	56.5	75.5	34.0
Niblack [18]	37.0	50.5	22.0
Souvola [17]	12.5	19.0	5.0

IV. Conclusion and Future Work

In this paper, we propose a novel method for classification of 2D and 3D texts blocks. The method identifies text candidates with the help of k-means clustering algorithm on gradient images. Then horizontal and vertical symmetry based on gradient direction and gradient magnitudes of text candidates to identify potential text candidates. The potential text candidates are validated by iterative method which uses temporal redundancy and spatial proximity of the potential text candidates to identify stable potential text candidates. For

stable potential text candidates, the method proposes new mutual nearest neighbor symmetry to identify the 2D and 3D text components. Voting method is used to classify 2D and 3D texts in frames. We evaluate the text line segmentation, text detection and recognition before classification and after classification with the results of the existing methods. However, the proposed methods detect text regions regardless of script type, therefore, we are planning to develop a method for script identification of 3D and 2D scripts in video in the future.

Acknowledgements

The work described in this paper was supported by the Natural Science Foundation of China under Grant No. 61272218 and No. 61321491, the 973 Program of China under Grant No. 2010CB327903, and the Program for New Century Excellent Talents under NCET-11-0232.

References

- [1] A. Smolic, "3D video and free viewpoint video-from capture to display", Pattern Recognition, 2011, pp 1958-1968.
- [2] R. Minetto, N. Thome, M. Cord, N. J. Leite and J. Stolfi, "SnooperText: A text detection system for automatic indexing of urban scenes", Computer Vision and Image Understanding, 2013 (article in press).
- [3] R. Minetto, N. Thome, M. Cord, N. J. Leite and J. Stolfi, "Snoopertrack: Text Detection and Tracking for Outdoor Videos", In Proc. ICIP, pp 505-508, 2011.
- [4] R. Minetto, N. Thome, M. Cord, N. J. Leite and J. Stolfi, "T-HOG: An effective gradient-based descriptor for single line text regions", Pattern Recognition, 2013, pp 1078-1090.
- [5] C. M. Gracia, M. Mirmehdi, J. Sigut, J. L. G. Mora, "Fast perspective recovery of text in natural scenes", Image and Vision Computing, 2013, pp 714-724.
- [6] D. Doermann, J. Liang and H. Li, "Progress in Camera-Based Document Image Analysis", In Proc. ICDAR, 2003, pp 606-616.
- [7] J. Zang and R. Kasturi, "Extraction of Text Objects in Video Documents: Recent Progress", In Proc. DAS, 2008, pp 5-17.
- [8] M. R. Lyu, J. Song and M. Cai, "A Comprehensive Method for Multilingual Video Text Detection, Localization, and Extraction", IEEE Trans. CSVT, 2005, pp 243-255.
- [9] R. Wang, W. Jin and L. Wu, "A Novel Video Caption Detection Approach using Multi-Frame Integration", In Proc. ICPR, 2004.
- [10] H. Li, D. Doermann and O. Kia, "Automatic Text Detection and Tracking in Digital Video", IEEE Trans. IP, pp 147-156, 2000.
- [11] T. Q. Phan, P. Shivakumara, T. Lu and C. L. Tan, "Recognition of Video Through Temporal Integration", In Proc. ICDAR, pp 589-593, 2013.
- [12] B. Bouaziz, T. Zlitni and W. Mahdi, "AViText: Automatic Video Text Extraction" CoRR abs/1301.2173, 2013.
- [13] X. Huang and H. Ma, Automatic Detection and Localization of Natural Scene Text in Video", In Proc. ICPR, 2010, pp 3216-3219.
- [14] P. Shivakumara, T. Q. Phan, S. Lu and C. L. Tan, "Gradient Vector Flow and Grouping based Method for Arbitrarily-Oriented Scene Text Detection in Video Images", IEEE Trans. CSVT, 2013, pp 1729-1739.
- [15] B. Epshtein, E. Ofek, Y. Wexler, "Detecting text in natural scenes with stroke width transform," In: Proc. CVPR, 2010, pp. 2963-2970.
- [16] P. Shivakumara, A. Dutta, T. Q. Phan, C. L. Tan and U. Pal, "A Novel Mutual Nearest Neighbor based Symmetry for Text Frame Classification in Video", Pattern Recognition, 2011, pp 1671-1683.
- [17] W. Niblack, "An Introduction to Digital Image Processing", Prentice Hall, Englewood Cliffs, 1986.
- [18] J. Sauvola, T. Seppanen, S. Haapakoski and M. Pietikainen, "Adaptive Document Binarization", In Proc. ICDAR, 1997, pp 147-152.
- [19] S. Roy, P. Shivakumara, P. Roy and C. L. Tan, "Wavelet-Gradient-Fusion for Video Text Binarization", In Proc. ICPR, 2012, pp 3300-3303.
- [20] Tesseract. <http://code.google.com/p/tesseract-ocr/>.