Key Frame Extraction for Text Based Video Retrieval Using Maximally Stable Extremal Regions

Werachard Wattanarachothai, Karn Patanukhom
Visual Intelligent and Pattern Understanding Laboratory
Department of Computer Engineering
Chiang Mai University, Thailand
karn@eng.cmu.ac.th

***Abstract*— This paper presents a new approach for text-based video content retrieval system. The proposed scheme consists of three main processes that are key frame extraction, text localization and keyword matching. For the key-frame extraction, we proposed a Maximally Stable Extremal Region (MSER) based feature which is oriented to segment shots of the video with different text contents. In text localization process, in order to form the text lines, the MSERs in each key frame are clustered based on their similarity in position, size, color, and stroke width. Then, Tesseract OCR engine is used for recognizing the text regions. In this work, to improve the recognition results, we input four images obtained from different pre-processing methods to Tesseract engine. Finally, the target keyword for querying is matched with OCR results based on an approximate string search scheme. The experiment shows that, by using the MSER feature, the videos can be segmented by using efficient number of shots and provide the better precision and recall in comparison with a sum of absolute difference and edge based method.**

Keywords-CBVR; text-based video retrieval; key frame extraction; shot boundary; MSER.

I. INTRODUCTION

A content based video retrieval system (CBVR) becomes more important nowadays because the number of video databases is rapidly growing. The CBVR [1]-[7] have been developed for many purposes. Several contents can be extracted from the videos depending on the applications such as human detection in surveillance camera based on the target characteristics (skin color, body size, hair and height) [1], shot classification for TV news [2], or event analysis in the sport videos [3].

In this paper, we focus on text information which is one of most useful content in the videos. Text localization and retrieval systems have been widely developed for many applications. License plate detection and recognition systems [4], [5] is one of the examples of text based video retrieval. The systems allow users to search for the vehicles in the video database based on their plate numbers. H.Yang et al. [6] developed an automated indexing system for video retrieval within large lecture video archives by using speech and text information. K. Choros [7] presented a method of automatic detection of sports news headlines which can be used for content-based indexing of TV sports news or recognizing sports disciplines.

For the text-based video retrieval (TBVR) system, the problem can be decomposed into three main parts that are key frame extraction, text localization, and text recognition. The key frame extraction process is a process for reducing redundancies among consecutive frames. In order to extract the key frame location, the videos are firstly segmented into shots which can be defined as continuous sequences of related frames. The key frames are considered to be the optimal set of frames that can represent the shot contents. There are many approaches developed to detect the shot boundaries [8]-[14]. Some examples of the features which have been developed for shot boundary detection are colors (color histogram [8], local color information [9]), edges (edge histogram [10], local edge information [11]), motions (optical flow), or textures (Gabor wavelet filter [12]). To detect the shot boundaries, the features are extracted for every frame. Similarities or feature distances between frames are observed and the shot boundaries can be determined by using thresholding schemes [13] or other classification scheme [14]. After the shot boundaries have been determined, the key frames can be located by optimizing the feature differences between the key frames and every frame in the entire shots [15]-[18].

In order to extract text information from the video frames, text locations have to be firstly detected. B.Epshtein et al. [19] proposed a method to locate texts in the natural scenes using a stroke width transform (SWT). The SWT is used for finding candidates of character components by connecting the pixels with similar values of stroke width. Then, the character candidates are grouped together based on similarities in stroke width, letter width and height, and space between letters or words. X. C. Yin et al. [20] developed a text localization scheme based on Maximally Stable Extremal Regions (MSERs). The MSER algorithm is used for selecting candidates of character regions. The character candidates are grouped together to construct the text candidates based on their similarities in interval, width, height, alignments, color, and stroke width. Then the text candidates are classified by height, width, aspect ratio, boundary smoothness, mean and variance of stroke width.

Optical character recognition (OCR) has been widely studied for a long time. There are many publications [21]-[23] and commercial products for the OCR applications. Tesseract [21] is one of the high-accuracy open source OCR engines that supports for various languages such as English, Arabic, Chinese, Japanese, etc. Tesseract was originally developed by

INISCom 2015, March 02-04, Tokyo, Japan

DOI 10.4108/icst.iniscom.2015.258410

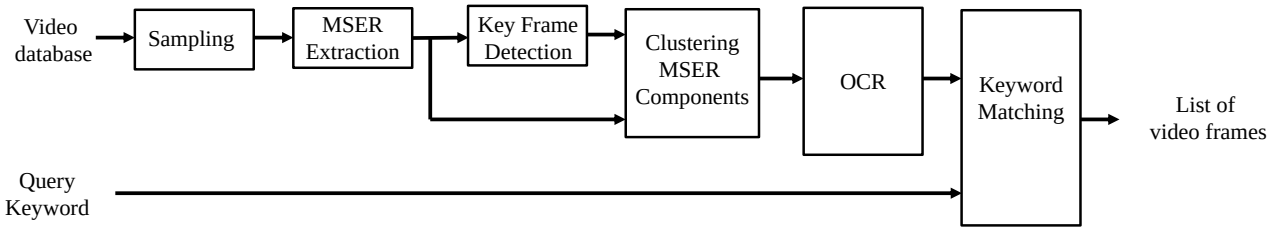

Figure 1. Overview of the proposed system.

Figure 2. Examples of MSER extraction, (top) original images, (bottom) MSERs shown in different colors.

Hewlett Packard. It was released as the open source in 2005 and has been sponsored by Google since 2006. Tesseract uses two-pass architecture for a word recognizer. On the first pass, all words in the image are recognized. Only recognized words that are in a dictionary and are not too ambiguous are passed to train an adaptive classifier. On the second pass, the words that were not satisfied on the first pass are recognized again by using the classifier that may have learned something useful from the first pass. In the recognition process, the words are segmented into characters where an initial segmentation is based on only geometry. However if the recognition result from that word is unsatisfactory and there are still some blobs can be chopped or merged, Tesseract will chop or merge the blobs and recognizes again until result is satisfying or there is no blobs to be chopped or merged. Tesseract uses a polygon approximation of character outline as a feature for classifying the characters. Currently, Tesseract also applies a Convolutional Neural Network (CNN) for the character classifier in a Cube mode. Tesseract also includes the processes of adaptive thresholding, page layout analysis, and text line analysis.

In this work, we present the MSER based key frame extraction scheme for text based video search. For the scene text recognition purpose, the videos should be segmented into shots with the same text contents. Ideally, the best key frames are the frames in the corresponding shot that cover all text contents and can provide the accurate OCR results. As a result, we proposed to detect the shot boundaries and the key frames based on the number of MSER components in the frames. The remaining of this paper is organized as follows. Section II describes the proposed method for retrieving the video frames from the query keywords. Section III and IV present the experimental results and conclusions of this paper, respectively.

II. The Proposed Method

Structure of the proposed scheme for the text based video search is demonstrated in Fig. 1. As shown in Fig. 1, inputs of the system are the video database and the query keyword while an output of the system is a list of video frames where the query keyword is found. The process can be considered as three stages that are (1) the key frame extraction stage, (2) the text localization stage and (3) the keyword matching stage.

The key frame extraction is starting from temporal down-sampling of the video frames. The sampling period must be chosen to ensure that all texts in the videos appear longer than the sampling period. The MSER algorithm is applied in every frame to extract the character candidates. The number of character candidates in the frame sequence is used to determine the shot boundaries and the key frame locations.

In the second stage, the system performs the text localization process only in the key frames. The character candidates are grouped into the text regions based on the similarity. To measure the similarity, RBG color difference, hue difference, width difference, height difference, stokes width difference, and geometric distances are used as features. This feature set is modified from the method of X.C.Yin [20] by removing a horizontal text alignment constraint and increasing a robustness of color similarity measuring by using both RGB and Hue.

In order to match the target keyword with the texts appearing in the scene, every region obtained in the previous stage is recognized. In this work, we use Tesseract OCR engine [21] for recognizing the texts. The proposed scheme inputs four different pre-processed images to the OCR engine. The recognition results can be different depending on how we process the images. In addition, the recognition results can be analyzed and used to remove the non-text regions. Then, the system determines the distances between the target keyword and every text region from every input type by using the Levenshtein distance [24]. The text regions are considered to be matched with the target keyword if the distances between them are lower than a threshold of sensitivity. The shots of the key frames that have at least one text region matched with the keyword are output as the searching result. The details of each stages of the proposed scheme are given in the following sections.

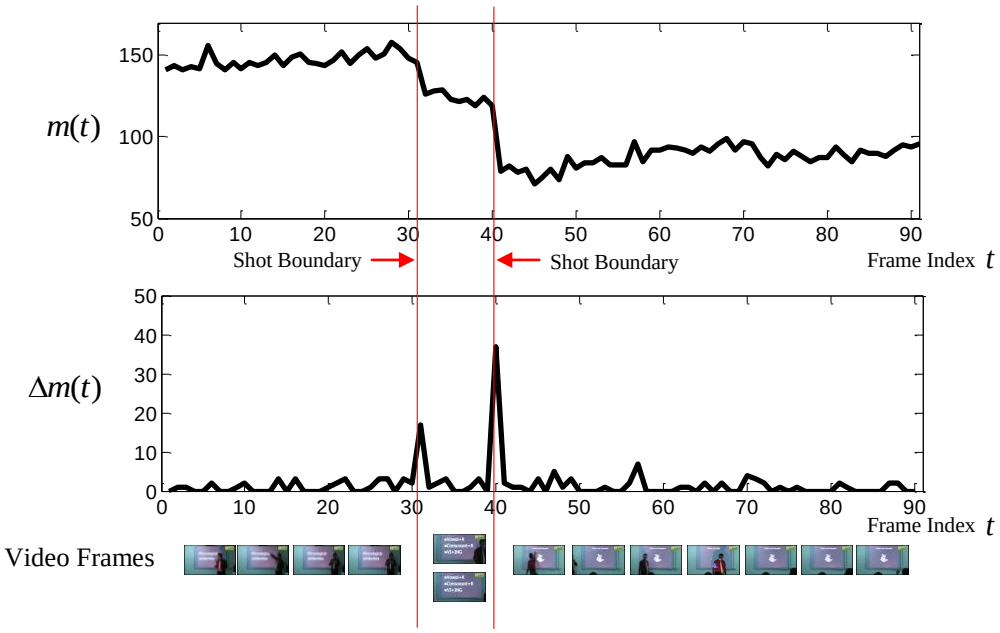

Figure 3. Example of the proposed shot boundary detection

A. *Maximally Stable Extremal Regions (MSERs)*

The MSER algorithm is an algorithm for extracting regions of interest that was firstly introduced by Matas et al. [25]. The MSER can be considered as a set of pixels with homogenous intensities that are higher or lower than their boundary pixels. The MSER algorithm can be applied to initialize the character candidate regions in the image [20]. To extract MSERs, binary images are extracted from the original image by varying the threshold levels step by step. Area of each connected component is compared between the current threshold level and the previous threshold level. If the area changes less than a maximum area variation limit, that connected component is considered as the MSER. The examples of MSER extraction are shown in Fig. 2. The results from MSER extraction still include both the character regions and a large number of the non-character regions. Then, the non-character regions can be eliminated in a clustering process which is described in Section II-C.

B. *MSER Based Key Frame Detection*

In order to segment the videos into shots with the same text contents, we propose a method that observes change of the number of MSER components in the frame sequence. Let $m(t)$ denotes the number of MSER components in the t-th frame. Parameters of the MSER extraction such as threshold step size, maximum area variation or component's size constraints must be set constantly for entire videos. The shot boundaries can be considered as transition edges of $m(t)$. To determine the transition edges, a first-order difference filter is applied as

$$\Delta m(t) = m(t) - m(t-1).$$

By using the thresholding scheme, the set of shot boundaries $\mathbf{B}$ can be defined as

$$\mathbf{B} = \{ t \mid \ |\Delta m(t)| \geq Th, |m(t)| \geq |m(t \pm 1)| \}.$$

The shot boundaries are the frames that provide local peaks of the differences of the number of MSER components and those differences are greater than the threshold. Smoothing filters can be employed to smooth $\Delta m(t)$ before the peak finding process to remove the small peaks. The threshold level Th and size of smoothing operators are used to control the number of segmented shots. The low threshold level or the small size of smoothing operators may cause the video to be over-segmented. On the other hand, some shots may be lost due to the high threshold level or the large size of smoothing operators. The optimal threshold and smoothing operator can be estimated by using a cross validation scheme. Illustration of the proposed shot boundary detection process is shown in Fig. 3.

Next, the key frames for each shot boundary are determined based on minimizing the following objective function as

$$z(t) = w_1 |\Delta m(t)| + w_2 \left| t - \frac{B_i + B_{i+1}}{2} \right| + w_3 |m_{\max} - m(t)|,$$

$$F_i = \arg\min_{t \in [B_i \ B_{i+1}]} (z(t)),$$

where F_i is the key frame within the shot boundary $[B_i \ B_{i+1}]$ and $z(t)$ is the objective function that is composed of three components. The first component $|\Delta m(t)|$ is a component that measures a stability of the scene. In this work,

the stable frames mean the frames with no change or very small change in comparison with the adjacent frames. Because of the accuracy of text detection and recognition processes are degraded in blurred images, the stable frame criterion is used to avoid blurred texts and other motion based degradations in the key frames. The second component $|t-(B_i+B_{i+1})/2|$ is called a centering criterion. This component is applied to avoid the key frames to locate too close to the shot boundaries since the frames near the shot boundaries may still have some transition of text contents. The last component $|m_{\max}-m(t)|$ is a component to measure the number of character candidates (MSER components) where $m_{\max}$ is the highest number of character candidates that can be extracted from the frames in $[B_i \ B_{i+1}]$. The number of character candidates reflects a possibility to discover all text contents in that shot. To create the objective function, three components are combined together using weights w_1, w_2, w_3. The optimal set of weights can be obtained by using training process.

C. Linkage Clustering

After the key frames have been detected, the next task is to locate the text regions in the key frames. Let M_i denote the i-th character candidate (MSER component) in the current key frame. According to X.C.Yin's method [20], the character candidates can be grouped by using a hierarchical clustering where a distance threshold should be specified to terminate the clustering process. A distance function is defined as a linear combination of individual feature distances where weights of combination and the distance threshold can be learned using a distance matric learning algorithm. In this work, we modify the feature set of X. C. Yin's method by removing text alignment depended spatial features to increase the robustness in perspective view. In addition, we use more color channels since, in practical situations, color quality in the videos are often lower than in the still images. The definitions of the individual features used in this process are listed as follows.

- Spatial distance:

$$D_1(i,j)=\frac{\sqrt{(x_i-x_j)^2+(y_i-y_j)^2}}{\max(w_i,w_j,h_i,h_j)}.$$

where $D_1(i,j)$ denotes a distance between the positions of character candidates M_i and M_j which is normalized by the size. (x_i, y_i) is a center of the bounding box of the character candidates M_i. w_i and h_i are the width and the height of bounding box of M_i, respectively.

- RGB color difference:

$$D_2(i,j)=\frac{\sqrt{(r_i-r_j)^2+(g_i-g_j)^2+(b_i-b_j)^2}}{255}.$$

where $D_2(i,j)$ denotes a color difference in RGB color space between M_i and M_j which is normalized by the upper bound of color value. r_i, g_i, b_i are medians of red, green and blue color components of the pixels in M_i, respectively.

- Hue difference:

$$D_3(i,j)=\frac{|h_i-h_j|}{360^\circ}.$$

where $D_3(i,j)$ denotes a hue difference in HSV color space between M_i and M_j. The values of hue are typically represented as a degree in color circle or $h\in[0^\circ \ 360^\circ]$; therefore, we normalize the hue difference by 360°. h_i denotes a median of hue of the pixels in M_i. $|h_i-h_j|$ is calculated based on 360° degree repetition, so the difference between 0° and 359° is equal to 1°.

- Width and height difference:

$$D_4(i,j)=\frac{|w_i-w_j|}{\max(w_i,w_j)}.$$

$$D_5(i,j)=\frac{|h_i-h_j|}{\max(h_i,h_j)}.$$

where $D_4(i,j)$ and $D_5(i,j)$ denote normalized differences in width and height between M_i and M_j, respectively.

- Stroke width difference:

$$D_6(i,j)=\frac{|sw_i-sw_j|}{\max(sw_i,sw_j)}.$$

where $D_6(i,j)$ denotes a normalized difference in stroke width between M_i and M_j. The stroke width [19] is a parameter to measure the line thickness of the characters. sw_i represents an average stroke width of M_i.

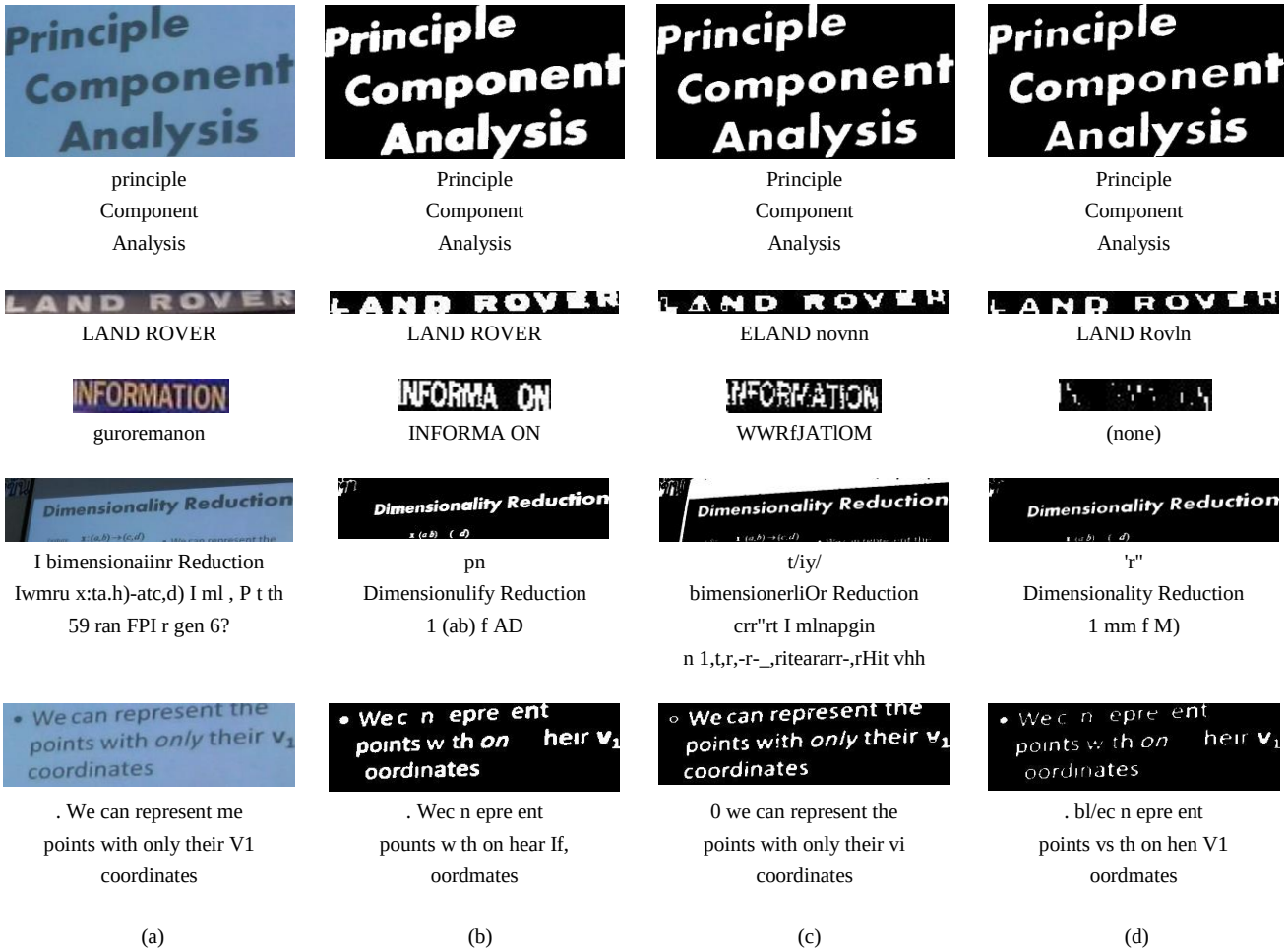

Figure 4. Examples of the text regions and the recognition results from Tesseract OCR using different types of input images, (a) Original color images, (b) MSER images, (c) Color thresholded images, (d) Eroded MSER image.

Based on the assumption that the characters that belong to the same text components should be near and have similar properties of color, size, and line thickness, finally, the distance function is defined from six individual features as

$$D(i, j) = \sum_{k=1}^{6} w_k D_k(i, j).$$

The character candidates M_i and M_j are merged together if the distance $D(i, j)$ is less than threshold. In this work, we also apply the separated constraints for each individual feature in this linkage clustering process.

D. Text Recognition

The characters in the text candidate regions are recognized in this step. In this paper, we use Tesseract OCR engine (version 3.0.2) as the recognizer. Since Tesseract already includes processes of image binarization, text line and word finding, joined character chopping, broken character associating, and classification [20], we can input text bounding boxes of raw image to the Tesseract engine. However, recognition results are still affected by how we process the image before input to Tesseract.

As a result, we proposed to use four different types of the input images as follows.

1) Original color image: This image is obtained by cropping the original image by the bounding box of text candidate region. This type of image is a typical types of input that includes all characters and background component. Let $T^{(1)}$ denote the recognized text by using the original color image. Fig. 4(a) shows the examples of the original color images and their corresponding recognized texts from Tesseract.

2) MSER image: This image is obtained by creating a binary mask of the text candidate region. Let $T^{(2)}$ denote the recognized text by using the MSER image. Fig. 4(b) shows the examples of the MSER images and their corresponding recognized texts. The background is removed by using this type of image but some characters may be lost as shown in Fig. 4(b).

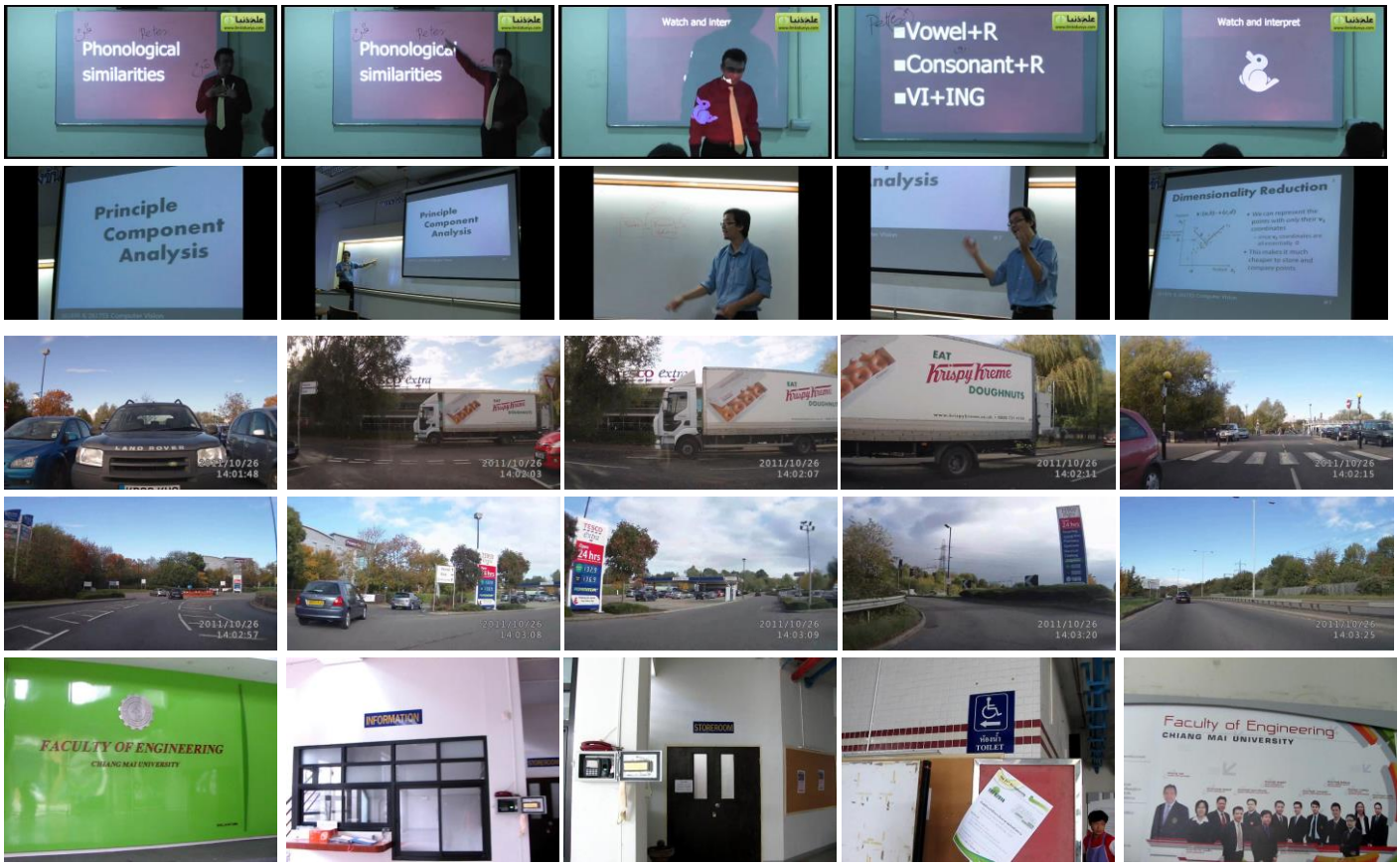

Figure 5. Screenshots of five videos used in the experiments.

TABLE I. DETAILS OF THE VIDEOS USED IN THE EXPERIMENTS

Video	Length	Resolution	Frame Rate	Source
Lecture 1	30 sec	854×468	25 fps	Youtube[1]
Lecture 2	1 min	640×480	30 fps	Canon TX1
Driving 1	30 sec	1920×1080	29 fps	Youtube[2]
Driving 2	30 sec	1920×1080	29 fps	Youtube[2]
Indoor	3 min	640×480	30 fps	Canon TX1

3) Color thresholded image: This image is a binary mask that is obtained by selecting the pixels in the original image that have the color within a range of text color. The text color is defined as a median of the every pixel color in the text candidate region. Let $T^{(3)}$ denote the recognized text by using the color thresholded image. Fig. 4(c) shows the examples of the color thresholded image images and their corresponding recognized texts. The advantage of this type of input is to recover some lost components in the MSER image as shown in the fifth row of Fig 4. However, this type of image may include some background elements that have similar color to the text as shown in the forth row.

4) Eroded MSER image: This image is obtained by using morphological erosion to the MSER image to chop the joined characters and removed some small non-text elements. Let $T^{(4)}$ denote the recognized text by using the eroded MSER image. Fig. 4(d) shows the examples of the eroded MSER image and their corresponding recognized texts.

The results obtained from this step are four versions of the recognized texts for every text candidate region in every key frame. These recognized texts will be matched with the keyword in the next step.

E. Keyword Matching

In this step, the distances from the target keyword K to every text region T are calculated. The distance function is defined based on an approximate string matching [26] using the Levenshtein distance. The approximate string matching is a technique for finding a pattern of string in the text. Let $d(K,T)$ denote a distance between the keyword K and the text region T which is defined as

$$d(K,T) = \min_{i\in[1,4]}(d(K,T^{(i)})).$$

The best match among four versions of the recognized texts $T^{(i)}$ is used to represent the distance to the text region T. The key frame is considered to be matched with keyword K if there is at least one text region T which $d(K,T) \le \varepsilon \cdot L_K$. L_K is a string length of the keyword K and ε is a threshold.

1: http://www.youtube.com/watch?v=JqBqftAVpyY
2: http://www.youtube.com/watch?v=2nD1v-UYoO8

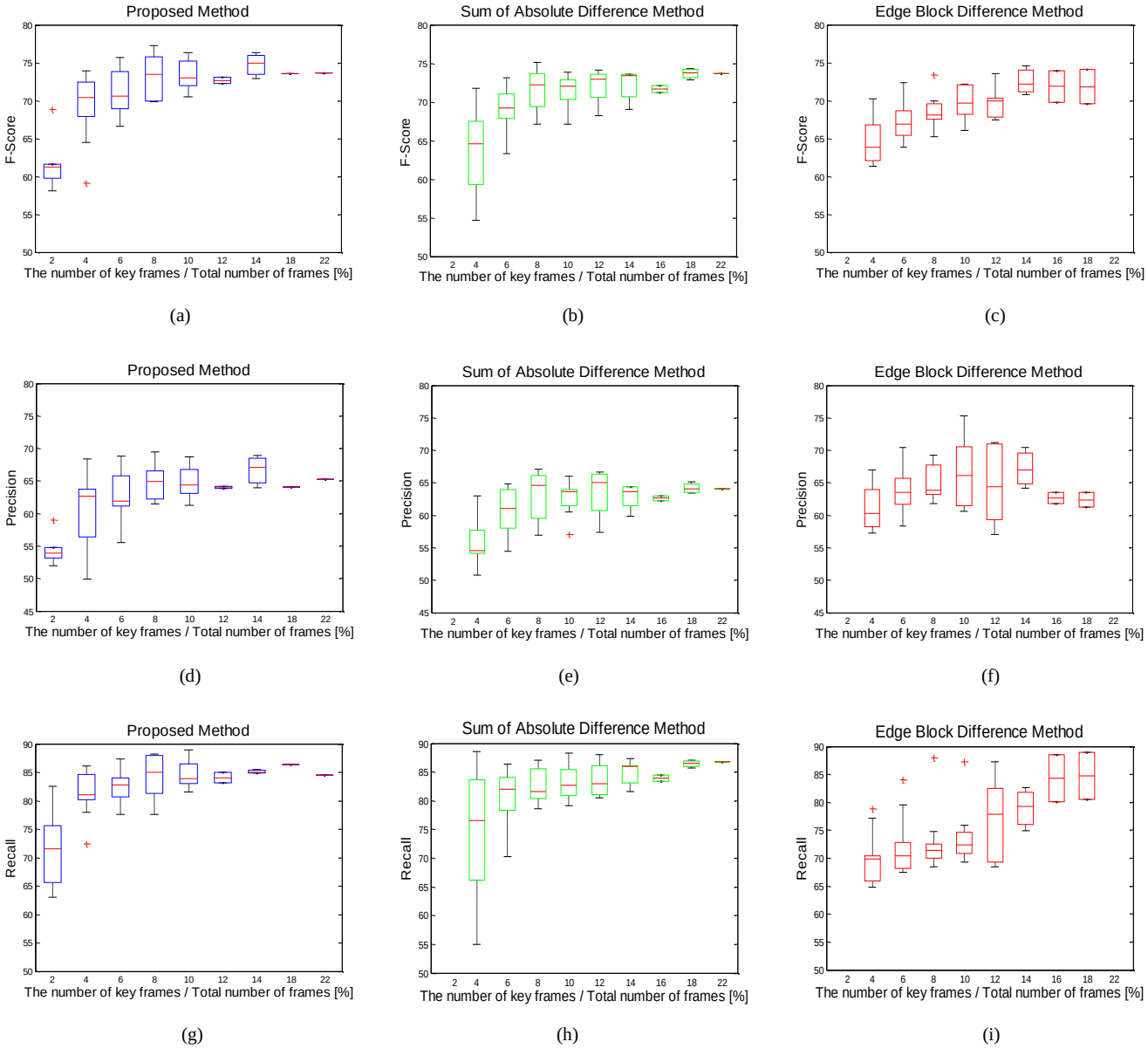

Figure 6. Performance of the proposed scheme in comparison with the baseline methods when the number of key frames is varying by thresholds and smooth operators, (a) F-Score of the proposed MSER feature, (b) F-Score of the SAD features, (c) F-Score of the EBD features, (d) Precision of the peoposed MSER features, (e) Precision of the SAD features, (f) Precision of the EBD features, (g) Recall of the proposed MSER features, (h) Recall of the SAD features, (i) Recall of the EBD features

Ranges of the frames which the keyword appears can be determined by using shot boundaries of the matched key frames.

III. Experimental Results

This section shows the experiment and performance evaluation of the proposed scheme. In this paper, the proposed system is tested on five videos. Fig. 5 shows some examples of screenshots of the test videos. The details of each video are given in Table I. Three videos are selected from Youtube's videos and the other two videos are our original videos captured for this experiment by using Canon TX1 camera. Keywords used in this experiment are collected from all English words appearing in the videos that contain at least five characters. List of keywords is shown in Table II. The performance of retrieval process is measured by using three parameters as

$$Precision = \frac{TP}{TP + FP},$$

$$Recall = \frac{TP}{TP + FN},$$

$$F = \frac{2 \times Precision \times Recall}{Precision + Recall},$$

where F represents F-score. TP is the number of the retrieved frames from the system that have the keyword. FP is the number of the retrieved frames that have no target keyword. FN is the number of frames in the video that cannot be retrieved by the system.

In this experiments, we compare our propose MSER based key frame detection method with two baseline features that are sum of absolute difference (SAD) and edge block difference (EBD). The down-sampling interval is set as 0.33 second in every experiment for every feature. In the SAD method, the frame difference is measured by summing up the absolute value of pixel difference in the entire frame. The EBD method is one example of the edge-based approaches where each frame are segmented into several blocks and difference of the number of edge pixels in each block are compared. The number of blocks that have large different in the number of edge pixels is counted and used as the feature. In order to compare the performance of the proposed MSER feature to the baseline methods in the key frame extraction process, the same thresholding method is implemented. We vary the threshold level and size of smoothing operator and measure the performance.

The characteristics of precision, recall and F-score are observed. The results are presented by using box plots as shown in Fig. 6. The proposed MSER-based method can provide the better F-score while using significantly less number of key frames in comparison with the other two baseline methods. By using the MSER feature, videos can be segmented into the shots with the same text content by using efficient number of shots; therefore, the F-score of the proposed methods is quite stable over the number of shots that segmented. In term of precision, the proposed method provides not much difference in performance with SAD and EBD. However, the recalls of SAD and EBD features decrease faster by reducing the number of key frames. It means that the shot segmentation does not fit to text content in SAD and EBD features.

The query results from the proposed and baseline methods are demonstrated in Table II. The precisions and recalls at the best F-scores are presented. The results show that the average precision and recall from five videos by using the proposed method are 68.26% and 85.03%, respectively. There are five missing keywords that cannot be found in "Driving 2" video. The reason is that these five words appears together at the same frames in only three frames (after sampling) so the proposed method cannot extract the key frames within that three frames.

IV. CONCLUSIONS

In this paper, we presented the content-based video retrieval system for searching text information in the video databases. The new shot boundary and key frame extraction are introduced in this work. The number of MSER components is proposed as the feature for the text content oriented shot segmentation. The MSER-based text localization and Tesseract OCR engine are applied to locate and recognize texts in the key frames. To improve the recognition results, the images from four different pre-processing methods are used as inputs for the OCR engine. In the experiment, the performance of the proposed method is compared with two baseline features using five test videos with 38 query keywords. The results show that shot segmentation by the MSER feature is more suitable for the text content than other features. The proposed method can provide the better results and use less key frames.

ACKKNOWLEDGEMENTS

This research was supported by Graduate School, Chiang Mai University, Thailand.

REFERENCES

[1] Y.Yang, B.C.Lovell, and F.Dadgostar, "Content-Based Video Retrieval (CBVR) System for CCTV Surveillance Videos," Digital Image Computing: Techniques and Applications, pp. 183-187, December 2009.

[2] M.Bertini, A.D.Bimbo, and P.Pala. "Content-based indexing and retrieval of TV news." Pattern Recognition Letters vol.22, no.5, pp. 503-516, 2001.

[3] T.Kawashima, K.tateyama, T.Iijima and Y.Aoki, "Indexing of Baseball Telecast for Content-based Video Retrieval" , Image Processing, 1998. ICIP 98. Proceedings. 1998 International Conference vol.1, pp 871-874.

[4] D.Wazlwar, E.Oruklu and J.Saniie, "A Design Flow for Robust License Plate Localization and Recognition in Complex Scenes," Journal of Transportation Technologies , 2012

[5] C.H.Chen, T.Y.Chen, M.T.Wu, T.T.Tang, W.C.Hu "License Plate Recognition for Moving Vehicles Using a Moving Camera" Ninth International Conference on Intelligent Information Hiding and Multimedia Signal Processing, pp. 497-500, 2013.

[6] H.Yang, C.Meinel, "Content Based Lecture Video Retrieval Using Speech and Video Text Information.", IEEE Trans. on Learning Technologies ,vol. 7-2, pp. 142 – 154, 2014

[7] K.Choros, "Automatic Detection of Headlines in Temporally Aggregated TV Sports News Videos" Image and Signal Processing and Analysis (ISPA), 2013 8th International Symposium on. IEEE, pp. 147-152, September, 2013.

[8] S.C.H.Hoi., L.L.S.Wong, and A.Lyu. "Chinese university of hongkong at trecvid 2006: Shot boundary detection and video search," TRECVid 2006 Workshop ,pp. 76-86 2006

[9] J.Mas, G.Fernandez, "Video shot boundary detection based on color histogram," Notebook Papers TRECVID2003, Gaithersburg, Maryland, NIST, 2003

[10] A.G.Hauptmann, R.Baron, M.Y.Chen, M.Christel, P.Duygulu, C.Huang, R.Jin, W.H.Lin, T.Ng, N.Moraveji, N.Papernick, C.Snoek, G.Tzanetakis, J.Yang, R.Yan, and H.Wactlar, "Infor-media at TRECVID 2003: Analyzing and searching broadcast news video," in Proc. TREC Video Retrieval Eval., Gaithersburg, MD,2003.

[11] H.-W.Yoo, H.-J. Ryoo, and D.-S.Jang, "Gradual shot boundary detectionusing localized edge blocks,"Multimedia Tools, vol. 28, no. 3, pp. 283–300, Mar. 2006.

[12] T.Barbu "Content–based Image Retrieval using Gabor Filtering", Database and Expert Systems Application. DEXA'09. 20th International Workshop pp. 236-240 , August 2009

[13] Z.Cernekova, I.Pitas, and C.Nikou, "Information theory-based shotcut/fade detection and video summarization,"IEEE Trans. Circuits Syst.Video Technol., vol. 16, no. 1, pp. 82–90, Jan. 2006.

[14] K.Matsumoto, M.Naito, K.Hoashi, and F.Sugaya, "SVM-based shotboundary detection with a novel feature," inProc. IEEE Int. Conf. Mul-timedia Expo., Jul. 2006, pp. 1837–1840.

[15] K.W.Sze, K.M.Lam, and G.P.Qiu, "A new key frame representation for video segment retrieval," IEEE Trans. Circuits Syst. Video Technol., vol. 15, no. 9, pp. 1148–1155, Sep. 2005.

TABLE II. PRECISION AND RECALL OF EACH QUERY KEYWORD

Videos	Keywords	#Frames (after down-sampling)	Proposed			SAD			EBD		
			#Key Frames	Precision	Recall	#Key Frames	Precision	Recall	#Key Frames	Precision	Recall
Lecture 1	Consonant	9	6	90%	100%	14	56%	100%	10	100%	78%
	Phonological	31		100%	100%		100%	81%		91%	100%
	Vowel	9		90%	100%		56%	100%		100%	78%
	Watch	49		94%	100%		94%	100%		84%	100%
	Interpret	43		83%	100%		84%	100%		86%	98%
	Similarities	31		100%	100%		100%	81%		91%	100%
	Average			92.47%	100.00%		86.49%	93.02%		88.36%	97.09%
Lecture 2	Analysis	162	42	36%	100%	55	35%	99%	25	35%	100%
	Component	160		44%	76%		43%	99%		36%	76%
	Computer Vision	418		99%	37%		100%	50%		97%	51%
	Dimensionality	286		87%	96%		99%	100%		87%	97%
	Feature	256		87%	82%		78%	82%		65%	83%
	Principle	160		100%	76%		98%	99%		98%	76%
	Reduction	284		86%	95%		98%	99%		85%	95%
	Cheaper	234		65%	97%		75%	94%		95%	96%
	Compares	234		62%	92%		54%	91%		51%	95%
	Coordinates	234		93%	97%		93%	94%		96%	95%
	Essentially	234		62%	91%		55%	91%		61%	92%
	Points	234		62%	97%		55%	95%		61%	93%
	Represent	234		93%	97%		92%	94%		95%	96%
	Average			70.26%	84.70%		69.37%	88.63%		68.43%	86.49%
Driving 1	Doughnuts	48	17	96%	52%	17	100%	60%	15	100%	60%
	Land Rover	24		63%	100%		61%	83%		65%	100%
	Extra	37		65%	100%		35%	65%		56%	51%
	Average			71.07%	78.90%		56.15%	66.97%		72.00%	66.06%
Driving 2	Clothing	3	13	0%	0%	21	0%	0%	20	0%	0%
	Electrical	3		0%	0%		0%	0%		0%	0%
	Extra	7		24%	57%		22%	57%		27%	100%
	Opticians	3		0%	0%		0%	0%		0%	0%
	Petrol	4		33%	75%		0%	0%		11%	75%
	Pharmacy	3		0%	0%		0%	0%		0%	0%
	Recycling	3		0%	0%		0%	0%		0%	0%
	Tesco	7		12%	57%		10%	14%		23%	100%
	Average			18.64%	33.33%		12.20%	15.15%		20.48%	51.52%
Indoor	Chiang Mai	49	44	81%	98%	43	76%	80%	52	86%	76%
	Engineering	49		81%	96%		98%	88%		87%	98%
	Faculty	49		71%	98%		52%	88%		49%	98%
	Floor	24		32%	63%		60%	100%		27%	58%
	Information	18		23%	78%		14%	39%		11%	39%
	Storeroom	18		100%	56%		79%	83%		82%	50%
	Toilet	8		5%	88%		6%	100%		5%	88%
	University	49		98%	90%		96%	45%		100%	41%
	Average			48.54%	88.26%		45.27%	76.14%		39.67%	71.97%
Overall Average			**122**	**68.26%**	**85.03%**	**150**	**66.95%**	**86.65%**	**122**	**65.59%**	**85.03%**

[16] D.Besiris, F.Fotopoulou, N.Laskaris, and G.Economou, "Key frame extraction in video sequences: A vantage points approach," inProc.IEEE Workshop Multimedia Signal Process., Athens, Greece, Oct. 2007,pp. 434–437.

[17] C.Gianluigi and S.Raimondo, "An innovative algorithm for key frameextraction in video summarization,"J. Real-Time Image Process.,vol.1,no. 1, pp. 69–88, Oct. 2006.

[18] W.Hu, N.Xie, L.Li, X.Zeng, S.Maybank "A Survey on Visual Content-Based Video Indexing and Retrieval," Systems, Man, and Cybernetics, Part C: Applications and Reviews, vol. 41, no.6, pp. 797-819, 2011.

[19] B.Epshtein, E.Ofek and Y.Wexler, "Detecting Text in Natural Scenes with Stroke Width Transform" Computer Vision and Pattern Recognition (CVPR) , pp. 2963-2970, 2010

[20] X.C.Yin, X.Yin, K.Huang and H.W.Hao, "Robust Text Detection in Natural Scene Images," IEEE Trans. on Pattern Analysis and Machine Intelligence, vol.36, No. 5, May 2014

[21] R.Smith, "An Overview of the Tesseract OCR Engine," ICDAR Vol. 7,pp. 629-633, 2007

[22] N.Arica and F.T.Yarman-Vural, "Optical Character Recognition for Cursive Handwriting" Pattern Analysis and Machine Intelligence, IEEE Transactions on 24.6 pp. 801-813, 2002.

[23] Y.Li, D.Lopresti, G.Nagy and A.Tomkins. "Validation of image defect models for optical character recognition." Pattern Analysis and Machine Intelligence, IEEE Transactions on 18.2 pp. 99-107 1996.

[24] L.Yuijian and L.bo, "A Normalized Levenshtein Distance Metric, " Pattern Analysis and Machine Intelligence, IEEE Transactions on 29.6 (2007): 1091-1095.

[25] J.Matas, O.Chum, M.Urban and T.Pajdla "Robust wide-baseline stereo from maximally stable extremal regions." Image and vision computing 22.10 (2004): 761-767.

[26] G.Navarro, "A Guided Tour to Approximate string matching, " ACM computing surveys (CSUR) pp. 33-38, 2001.