GENERAL AND DOMAIN-SPECIFIC TECHNIQUES FOR DETECTING AND RECOGNIZING SUPERIMPOSED TEXT IN VIDEO

Dongqing Zhang, Raj Kumar Rajendran, and Shih-Fu Chang
Department of Electrical Engineering, Columbia University
New York, NY 10027, USA. {dqzhang,kumar,sfchang}@ee.columbia.edu

ABSTRACT

We have developed generic and domain-specific video algorithms for caption text extraction and recognition in digital video. Our system includes several unique features: for caption box location, we combine the compressed-domain features derived from DCT coefficients and motion vectors. Long-term temporal consistency is employed to enhance localization performance. For character segmentation, we use a single-pass threshold free approach combining classification and projection to address noisy segmentation, text intensity variation, and algorithm complexity. In recognition, we use Zernike moments to achieve more accurate recognition performance. Finally, domain knowledge is explored and a statistical transition graph model is used to enhance recognition of domain-specific characters, such as ball counts and game score of baseball videos. The algorithms achieved real-time speed and significantly improved recognition accuracy. Furthermore, although the experiments were conducted in baseball videos only, these algorithms (except the transition model) are general and can be used in other applications, such as news and films.

1. INTRODUCTION

Detection and recognition of text captions embedded in image frames of videos is an important component for video retrieval and indexing. Such text captions may be further divided to two types: *superimposed text* (which are added during editing processes) and *scene text* (which exist in the real-world objects and scenes). This paper focuses on the former type. Detection and recognition of such captions are challenging due to: varied locations, font variations, low resolution, blurred/transparent characters etc.

In specific domains such as baseball videos, caption box provides important information such as score, inning, ball count etc., which are important for video indexing. Although such information may be obtained by manual logging (such as game statistics distributed over the Internet), recognizing such information directly from video signals provide unique benefits – (1) extracted text is exactly synchronized with the image data when the event occurs, and (2) manual logging may not be feasible for large collection of archived videos.

Many researchers have investigated the above problem. T.Sato *et al.* [1] developed a system for News videos. They use spatial filters to localize text regions, and use thresholding of projection profiles for character segmentation. In order to handle issues caused by fixed thresholding and problems in character segmentation, a unique iterative process was proposed to combine the segmentation step and the subsequent recognition step, at the cost of system speed. Lienhart *et al.* [2] proposed an approach to extract the superimposed text and scene text in video. They combine multiple features (texture, motion, contrast and color) to locate and extract the superimposed text and scene text. Commercial text recognition tools were used for recognition. H. Li *et al.* [3] detect text by a wavelet transform and neural network based method, and recognize the character by commercial tools. Y. Zhong [4] *et al*, used texture features derived from DCT coefficients to localize text regions in I frames of MPEG videos. The method may suffer from many false alarms from regions having similar textures. Bertini *et al.* [5] presented a text location method using salient point detection, which may result in poor performance in areas with cluttered background.

In this paper, we investigated the problem of extracting and recognizing the superimposed text in videos. We combine a set of novel, general methods, and a set of domain-specific methods exploring domain-specific knowledge. For caption box location, we combine compressed-domain features derived from DCT coefficients and motion vectors. Long-term temporal consistency is employed to eliminate false alarms and automatically adapt to the location variations among different sources. For character segmentation, we combine unsupervised cluster-based classification and local minima searching of projection profile to obtain more accurate segmentation. Zernike moment features are used in text recognition. Our system (when tested in sports video) demonstrated significant performance improvement, from 47%, 78% to 92%, compared to two prior approaches using different character segmentation and recognition methods. Our method utilizing domain knowledge improved accuracy from 78% to 95%. Real-time speed is also achieved by software.

2. SYSTEM DESCRIPTION

The system is to extract the caption box in video and recognize the superimposed text. Our system, as shown in

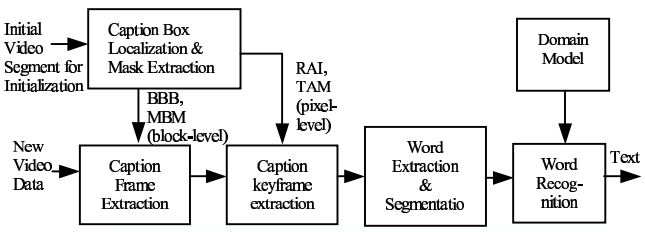

Fig. 1 System architecture

Figure 1, includes an initialization component to adapt to different video sources (e.g., different channels). Its output includes information about bounding box, representative feature mask etc of the text bearing area. Such information is used subsequently in detecting candidate image frames that may contain text, and extracting caption keyframes in which change in superimposed text occurs (compared to text shown on the previous text-bearing frame). Candidate text bearing frames are further processed to extract and segment words, which are then processed for recognition. Domain models are used for recognizing a subset of text to take advantage of the specific generation rules of such text.

3. INITIALIZATION

An unsupervised initialization process is used to automatically adapt to different sources (with different text locations, lighting, fonts etc) and find the location of text bearing areas and their representative masks. The process typically lasts for 30-60 seconds, after which the extracted information is used in continuous process of new video data.

During the initialization process we convert the motion vectors into a motion energy image, which is then binarized to a binary image. This is based on the assumption that superimposed text does not move over time (exceptions exist, such as rolling credits in films). We also use DCT coefficients on I frames to derive approximate texture features. A texture energy image is computed at the block level (e.g., 8 pixels by 8 pixels), and then binarized to another binary image. The above motion image and texture image are combined (over neighboring I and B/P frames) to form a joint motion-texture binary map, in which value '1' corresponds to candidate text locations with almost zero motion and text-like texture. Morphological filter and grouping methods are then applied to form contiguous candidate text regions.

Results of the above steps typically contain significant false alarms. To solve this, we explore the text box location consistency among different frames over time. An unsupervised incremental clustering method is used. Candidate regions in a new frame are mapped to region clusters based on the following area overlap metric:

$$S_r(R_1,R_2)=1-\frac{a(R_1-R_1\cap R_2)+a(R_2-R_2\cap R_1)}{a(R_1)+a(R_2)} \quad (1)$$

where R_1, R_2 are the two regions, and $a(R)$ is the area of the region R. A new region is mapped to an existing cluster if the above metric is less than a threshold, otherwise, a new cluster is formed. The clustering process stops when a dominant cluster is found. A cluster is said to be dominant if more than 40% of the frames in a continuous sliding window (e.g., 30 seconds) are mapped to the cluster.

The dominant cluster is used to compute the following outputs – the Median Binary Mask (MBM, by taking the median of the member binary images in the cluster), and the Block-Level Bounding Box (BBB) of the median binary mask. In addition, the candidate regions are also decoded to the pixel domain, from which an averaging operation is applied over the members of the cluster to obtain a Representative Average Image (RAI). The rationale for such averaging operation in the pixel domain is that text pixels are usually static while background pixels vary over time. Such averaging operation enforces the text pixels while smoothing out the temporal variations (i.e., high frequency components) of the background pixels. Finally, edge detection and connection methods are applied to extract the outermost contour of the text area and thus a Text Area Mask (TAM) is obtained. Figure 2 shows examples of the aforementioned outputs.

(a) Caption frame (b) Median Binary Mask (c) Representative Average Image (d) Text Area Mask

Fig 2. Outputs from the initialization process

Note the above outputs carry information about the location and representative values of the text areas in both the compressed domain (block level) and the uncompressed domain (pixel level), and will be used in the subsequent processing modules as references primarily for confining the search areas, verifying candidates, and detecting text keyframes.

4. DETECTING TEXT KEYFRAMES

After the initialization process, subsequent images are processed. All compressed-domain processing is confined within the BBB while all pixel-domain processing is confined within the TAM. The confinement helps achieve real-time implementation and avoid interference from background pixels outside the text area. First, similar to the first half of the initialization process, motion vectors and DCT coefficients are used to compute binary feature maps indicating the motion and texture features of each block. After morphological processing, each candidate region is verified by checking its area overlap correlation (metric defined in Eq 1) with the MBM of the dominant cluster. This is to ensure the region has sufficient similarity in shape and location compared to the representative values found in the initialization process. A frame is identified as caption frame if one of its detected regions has high correlation with the MBM. Then, the candidate area of the caption frame is decoded to the pixel domain, and a pixel-wise distance (Euclidean distance in RGB space) between the caption image and the RAI is computed. If the distance is larger than a threshold, it is also identified as false alarm (e.g., text during commercials or non-text areas). Duplicate caption images (i.e., text box showing the same information) are ruled out by distance computation between successive caption frames and distance thresholding. After these operations, we obtain a set of caption keyframes. Keyframes are the frames in which text information changes, e.g., the frame when the ball count is changed from 1-2 to 2-2.

5. WORD EXTRACTION AND CHARACTER SEGMENATION

We apply word extraction and text recognition on the above detected keyframes. Word extraction is to extract the word regions from the text area in the caption image. We use both spatial and temporal segmentation. Spatial segmentation is

applied on the caption image using intensity binarization and grouping to form candidate word regions. Size constrains are used to remove outlier regions. Sometimes, the graphic objects in the text area (such as line or boxes) may interfere with the spatial segmentation process and cause misses of word regions in the vicinity of the graphic objects. To correct this, we add a module to filter out regions with static values by thresholding temporal variance of the image. Details are skipped here due to space limit.

Character segmentation is a tough task due to the low resolution of the characters, and thus character spacing. Vertical projection profile and thresholding has been used in previous system [1]. Iterative algorithm is used to improve accuracy at the cost of speed. However, different videos may have various character intensity, fixed threshold may result in partial segmentation or loss of strokes. To overcome this problem, we developed an algorithm that less sensitive to source variation, single pass, fast, and accurate. Instead of finding the crossing point in vertical profile projection, we first find the local minima points as the candidate character segmentation lines. Pixels within the text area are classified into three types, i.e. *character*, *background* and *neither*, based a histogram segmentation method. The candidate segmentation lines are confirmed to be segmentation boundary if the segmentation line does not contain a significant number of pixels that are classified to be 'character' type. (not more than 2 in our experiment). (See Fig 3) This method is less sensitive to intensity changing by using histogram clustering, also less likely to produce partial segmentation due to use of local minima (partly because of single line boundary). The computation is efficient and can be made real-time.

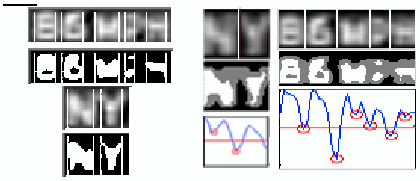

Fig 3. (a) inappropriate threshold may cause inaccurate segmentation (b) segmentation using local minima and classification. Note the partial segmentation errors in (a).

6. RECOGNITION OF CHARACTERS

Limited resolution significantly hampers text recognition in video. This problem is particularly acute in sports video, in which the character size may be as low as 7x7, much lower than news video.

In text recognition in document images, various schemes have been developed. The performance comparison in [6] showed that Zernike moments performed best. Zernike moments [7] are calculated based on a set of complex polynomials that form a complete orthogonal set inside the unit circle. The Zernike moments are the projection of the image onto these complex bases.

After the above character segmentation process, the image of each character is projected to complex bases to form a character feature vector. In order to calculate Zernike moments, the coordinate of each white pixel of the character image is normalized into the interior of unit circle, i.e. $x^2 + y^2 \le 1$. This normalization process provides robustness under uniform scaling of the image size. Both magnitudes and phases of Zernike moments are used. In our test, the maximum order of Zernike is *19*, and the dimension of character feature vector is 220.

The character image is first converted into three binary images using three thresholds. Two of them are produced by intensity segmentation used in section 4, i.e. the thresholds distinguishing *background* vs. *neither* and *neither* vs. *character*. The third threshold is the average value of these two thresholds. Zernike feature vectors are computed for each binary image. Such process is to enhance the robustness to the variation of stroke widths of different fonts. After thresholding, the black margin of the character image has to be trimmed to obtain the tightest rectangle box.

To enhance recognition, we further adopt some domain-specific approaches to explore the domain knowledge and production rules. For baseball video, in addition to the feature vectors corresponding to single characters, we add a word dictionary, which contains team names, speed numbers, digits, and other possible words. A word-level feature vector is generated for each word in the dictionary by concatenating the Zernike feature vectors of all characters contained in the word. The character feature vectors are created from each template character image given the pre-selected font bases (e.g., Gothic or Times). In the recognition stage, a word feature vector is generated for each extracted word region by concatenating all feature vectors of the segmented character images contained in the word region. A cosine distance metric is used to match the input word feature vector and feature vectors of all candidate words of the same length in the database. Finally, the highest match is taken as the recognition result.

7. INCORPORATING DOMAIN MODEL

False recognition is inevitable due to noise, font variation or inaccurate segmentation. For domain-specific text like scores, ball counts etc., there are special rules governing the permissible changes in time. Taking the example of ball count (strike-ball pair) in the baseball, only specific changes are allowed. The sequence pattern, like 0-1, 0-2, 1-1, is unlikely except very rare human editing errors occur. To take advantage of the specific transition rules, we use a transition graph to model such temporal relationship as shown in the Figure 4.

We define the node score as best correlation value between the input word with the template word in database (described in section 6). The transition matching score is defined as the conditional probability

$$S_t(n_{t-1}, n_t) = p(n_t \mid n_{t-1}) \qquad (2)$$

n_t, n_{t-1} are the nodes at the time t and time $t-1$. The

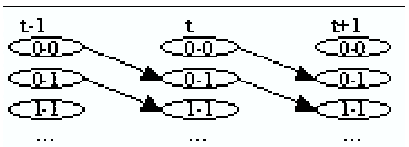

Fig 4. Transition graph of strike-ball sequence in baseball

transition conditional probability was estimated from the observation samples of strike-ball sequences in real baseball videos. A small probability is assigned initially to each conditional probability to handle possible miss or false alarm. Weighting factor λ is used to combine the node cost and transition cost as an overall cost at one node:

$$S(n_t) = \lambda S_n(n_t) + (1-\lambda) S_t(n_{t-1}, n_t) \quad (3)$$

Under the above model, character recognition problem is reduced to a longest path searching problem in the transition graph, which can be solved by dynamic programming. The weighting factor is to adjust the balance between the contribution from the image-based text recognition methods and the knowledge-based approach. Determination of λ depends on the characteristics of the sources and may be made adaptive by some supervised training process.

8. EXPERIMNETAL RESULTS

Three baseball videos and one NBA basketball video are taken as experimental subjects. They are from different TV channels and have different caption box patterns. They include commercials, and introductory segments on teams and players at the beginning of the video.

The initialization process correctly recognizes the bounding rectangles in all four videos. Among the 1134 ground truth keyframes, there are only 4 misses (0.3%) and 22 false alarms (1.9%). The keyframe miss is defined as those frames that contain new characters but are not extracted. It is usually caused by brief duration of some words, which are removed by the causal temporal morphological filtering used in our implementation. Such problem may be corrected by using a non-causal or bi-directional filtering method. False alarms are frames that contain duplicate text as previous frames or text in commercials. Note here we focus on text in the video programs and consider text in commercial periods as false alarms.

Character recognition is performed only on the detected keyframes. The word dictionary contains all team symbols of Major League Baseball, and NBA basketball; all possible two-digit numbers (for speed or time); all strike-ball numbers; single digits, and other characters (e.g., out, mph). The number of word templates in dictionary is 187. The character template database is constructed on a special "Gothic" font used in Windows 2000 system. We evaluate the performance of our system vs two other systems – one substituting the character segmentation module with the method using projection profile, thresholding, and crossing point detection (i.e., the baseline method used in [1] without complicated iterative process); another substituting the recognition module with a traditional image template matching method (also used in [1]). The characters include score, strike-ball, inning, quarters, team name, time and other words. Among the 13,657 extracted characters (from more than 4 hours of videos of various sources), the character recognition rates are 92% (our system), 78% (system with old character segmentation method), and 47% (system with image template matching method).

The performance comparison shows that our character segmentation method outperforms the old approach, which suffers from thresholding sensitive to intensity variation. Also, our recognition method using Zernike moment significantly outperforms the image template matching approach. This is confirmed by the significantly worse performance when the comparison system using image template matching method is applied to NBA videos (which use a slightly different font than baseball videos).

We evaluate the performance of domain knowledge based method, compared to the generic method. The strike-ball image sequences are extracted from those three baseball videos. "Arial" instead of "Gothic" font database is used to simulate font variation. Some strike-ball pairs have connected strokes, which make segmentation output inaccurate. Before recognition, one strike-ball sequence from a two-hour baseball video is taken to train the graph model and estimate the transition probabilities. The weighting factor λ is set to 0.02, which is selected empirically. The small value of λ may mean the variation of $S_n(n_t)$ is very low compared to $S_t(n_{t-1}, n_t)$, namely, the image feature based recognition is weak and the knowledge-based module plays a more important role. Within the ball-count characters, the knowledge-based model improves the performance from 78% to 95%. When λ is changed to 0.1, the improvement is from 78% to 87%.

9. CONCLUSION

We propose new methods to extract the caption box and recognize the superimposed text in videos. Our methods consist of several unique features, including robust localization, accurate segmentation, Zernike feature based recognition, and temporal transitional graph model, and real-time performance. We demonstrated significant performance improvement in experiments using videos from different sources. Future works include improving the domain-independent components to handle more severe conditions such as font variations, and systematic derivation of knowledge from new domains.

10. REFERENCES

[1] T. Sato, T. Kanade, E. Hughes, and M. Smith, "Video OCR: Indexing Digital News Libraries by Recognition of Superimposed Captions", Multimedia Systems, 7:385-394, 1999.

[2] R.Lienhart, W.Effelsberg,"Automatic text segmentation and text recognition for video indexing", Multimedia System, 2000.

[3] Huiping Li; Doermann, D.; Kia, O., "Automatic text detection and tracking in digital video", IEEE Trans. on Image processing, Vol 9, No. 1, January 2000.

[4] Y. Zhong, H, Zhang, and Anil K.Jain, "Automatic Caption Localization in Compressed Video", IEEE Trans on PAMI, Vol 22. No.4 April 2000.

[5] M.Bertini, C. Colombo and A. Del Bimbo, "Automatic caption localization in videos using salient points". Proceedings of IEEE International Conference on Multimedia and Expo ICME 2001, Tokio, Japan, August 2001.

[6] O. Trier and A. Jain and T. Taxt, "Feature extraction methods for character recognition - A survey", Pattern Recognition 29, pp. 641-662, 1996.

[7] A. Khotanzad and Y.H. Hong, "Invariant Image Recognition by Zernike Moments", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol 12, No 5, May 1990.