SEMANTIC KEYWORD EXTRACTION VIA ADAPTIVE TEXT BINARIZATION OF UNSTRUCTURED UNSOURCED VIDEO

Michele Merler and John R. Kender

Department of Computer Science
Columbia University

ABSTRACT

We propose a fully automatic method for summarizing and indexing unstructured presentation videos based on text extracted from the projected slides. We use changes of text in the slides as a means to segment the video into semantic shots. Unlike precedent approaches, our method does not depend on availability of the electronic source of the slides, but rather extracts and recognizes the text directly from the video. Once text regions are detected within keyframes, a novel binarization algorithm, Local Adaptive Otsu (LOA), is employed to deal with the low quality of video scene text, before feeding the regions to the open source Tesseract[1] OCR engine for recognition. We tested our system on a corpus of 8 presentation videos for a total of 1 hour and 45 minutes, achieving 0.5343 Precision and 0.7446 Recall Character recognition rates, and 0.4947 Precision and 0.6651 Recall Word recognition rates. Besides being used for multimedia documents, topic indexing, and cross referencing, our system can be integrated into summarization and presentation tools such as the VAST MultiMedia Browser [1].

Index Terms— Semantic keywords,unstructured video

1. INTRODUCTION

Slides consist in a natural way of segmenting presentations, since the presenter thoughtfully prepared and conferred a specific semantic meaning to each of them. Studies have been conducted in order to assess the reliability of slides as a summarization tool [2].

Most of the methods proposed to summarize presentation videos with shots segmentations based on slides rely on the availability of electronic copies of the slides themselves, which is not always realistic. Our solution, on the other hand, works without any slide template to be compared, hence the definition "unsourced". Video text detection is a widely studied field, but most of the systems focus on recognizing artificial text which is superimposed to the video in a post processing step[3]. The text of the slides we focus on belongs instead to the category of scene text, which is embedded in the scene and captured with the rest of the data.

Directly applying an OCR engine such as Tesseract to the color or grayscale text regions extracted from video yields not reliable results, because of the low quality and low resolution of such regions. A common solution in literature consists in somehow enhancing and binarizing the text before feeding it to the OCR engine [4, 5]. Given the extremely large amount of data included in a video, the whole method should not only be reliable but also efficient, and the binarization step does not make any exceptions.

We propose a new Local Adaptive version of the Otsu [6] method (LAO), which is implemented with integral histograms. Thus it is more robust to illumination changes and background variations within the text areas, while still efficient thanks to the use of integral histograms. In particular, it allows the determination of an optimal threshold that maximizes the between-classes variance within a subwindow, with computational complexity independent from the size of the window itself.

2. SEMANTIC KEYWORD EXTRACTION

2.1. Domain Description

The videos we analyze were not captured by professionals or through an ad-hoc capture system such as in [7]. Furthermore, they were not edited in any way. They therefore present challenges for text recognition in that the camera is rarely steady, and the projected slides are often truncated out from the field of view or occluded by the speakers. In fact, in many cases the recording person focused his attention on the speaker rather than on the slide, zooming in on his or her face.

Many new recording systems are explicitly designed to synchronously capture all contents of the presentations (including audio, video, and presentation material) [7]. However, there already exist many large scale archives of raw videos, such as universitiy lectures recordings, for which no other information, including electronic copy of the slides, besides the video itself is available. In order to summarize, index, and cross reference presentation videos from such archives, a more robust system is required.

2.2. Text Regions Detection

Initial text detection is performed with a simple and fast approach. Usually power point presentations do not present text overlaid to particularly challenging backgrounds, therefore even a simple approach can perform reasonably well. Initially a Laplacian of Gaussian operator is applied to a frame in order to extract edges. Subsequently connected components are located within the edge map and textural and geometrical properties of the regions enclosing such connected components are extracted. The inspected properties are: coordinates of the center, area, width, height, width/height ratio, density of edges, vertical and horizontal alignment. Empirically validated thresholds are applied to each feature in order to prune non-text regions. The

This research was supported in part by NSF grant IIS-0713064.

[1]http://code.google.com/p/tesseract-ocr/

candidate text regions R within a frame F must satisfy:

$$F_{Area}/1000 \leq R_{area} \leq F_{Area}/10 \tag{1}$$
$$2 \leq R_{width} \leq F_{width}/3 \tag{2}$$
$$6 \leq R_{height} \leq F_{height}/5 \tag{3}$$
$$E_{density} \geq 0.2 \tag{4}$$

The candidate text regions are then passed to the recognition block to be finally confirmed, in the case one or more characters are recognized, or discarded when no character is recognized.

2.3. Binarization

Otsu [6] adaptive thresholding method is one of the oldest and most used binarization techniques in image processing. Assuming a bi-modal distribution within the gray scale histogram of an image, it aims at automatically selecting an optimal threshold T to minimize the within-class variance $\sigma^2_{within}(T)$ of the two modes, or equivalently to maximize their between-class variance $\sigma^2_{between}(T)$. Given an image with pixel values ranging in an interval of intensity levels $[0, L-1]$, the within- and between-class variance for a specific threshold T are computed as

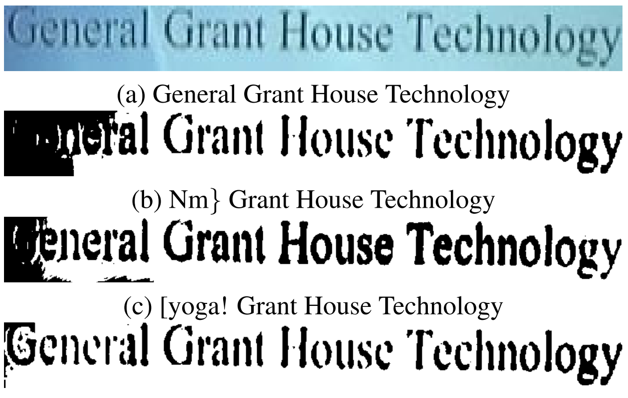
$$\sigma^2_{within}(T) = n_B(T)\,\sigma^2_B(T) + n_F(T)\,\sigma^2_F(T) \tag{5}$$
$$\sigma^2_{between}(T) = n_B(T)\,n_F(T)\,(\mu_B(T) - \mu_F(T))^2 \tag{6}$$

where $\mu_B(T)$ and $\mu_B(T)$, $\sigma^2_B(T)$ and $\sigma^2_F(T)$ are respectively the mean and variances of the background (below the threshold T) and foreground (above T) pixels clusters, while $n_B(T)$ and $n_F(T)$ represent the number of pixels belonging to each cluster. For efficiency reasons, the optimal T is computed as the one maximizing $\sigma^2_{between}(T)$.

Despite being very simple, parameter free, fast to compute and generally quite well performing, the classical Otsu method presents a main limitation in its globality. Computing an optimal threshold for the whole image makes it sensitive to shadows, shading and local noise, as shown in Figure 1(b). In order to overcome such limitation, local methods have been introduced. Those methods work by sliding a $W \times W$ window and select a threshold for the pixel where it is centered based on the statistics of its neighbors. One of the most popular among such algorithms is Sauvola's, in which the threshold $t(x, y)$ is computed as

$$t(x,y) = \mu(x,y,W)\left[1 + k\left(\frac{\sigma(x,y,W)}{R} - 1\right)\right] \tag{7}$$

were where R is the maximum value of the standard deviation within the window, and k is a parameter which takes positive values in the range $[0.2, 0.5]$. In Figure 1(c) we can see that this local approach tends to overcome the limitations of the global Otsu algorithm (in (b)). However, the algorithm is limited by the dependence of $t(x, y)$ from 2 parameters: the window size W, which also determines efficiency, and the value k. The computational complexity has been recently made independent from W thanks to the introduction of the integral images [8], which allow to compute mean and variance of subwindows of any size within an image with a constant amount of operations. However, the method is still chained to an ad hoc selection of k. The choice of $t(x, y)$ is not related to any optimization process, and remains quite arbitrary. We propose to eliminate the dependency of $t(x, y)$ from an ad hoc parameter by computing it as the threshold that optimizes the between-class variance within the window. In other words, our solution consists in a localized version of the Otsu algorithm, or an optimal version of Sauvola's one, which can combine the strengths of the two methods: locality and optimality. The Local Adaptive Otsu algorithm (LAO) slides a window of size W across the image and computes the threshold with the optimal Otsu criterion in each position. At each position one can choose whether to apply the threshold directly to the whole window (used in the rest of this paper) or simply to the pixel at which the window is centered (as in Sauvola's approach). In order to limit the computational complexity derived from the application of the window, we use the integral histogram [9], a structure consisting in L integral images, one per bin. Once paid the initial cost of building the integral histogram for the whole image, such structure allows to compute the values of $n_B(T,W)$, $n_F(T,W)$, $\mu_B(T,W)$ and $\mu_F(T,W)$ for equation 6 in any subwindow with a constant number of operations, independently from the window size. In Table 2 is reported the experimental gain in time achieved with the introduction of the integral histogram over a baseline implementation of the algorithm.


(a) General Grant House Technology

(b) Nm} Grant House Technology

(c) [yoga! Grant House Technology

(d) [(cncra1l Grant House Technology

Fig. 1. Example of the compared binarization methods. Original image (a) and its versions binarized with (b) original Otsu, (c) Sauvola and (d) adaptive Otsu. Under each image is reported the text recognized by the system. In this case adaptive Otsu outperformes the other methods in dealing with the shaded area around the word "General". In fact, it manages to identify 4 of its characters, against none of the original Otsu and 1 of Sauvola.

2.4. Recognition

Word recognition is implemented by the Tesseract OCR engine. We trained Tesseract with 15 character sets, using the most commmon fonts for PowerPoint presentations [10], with a text reflecting the frequencies of English letters[2]. Each font was represented in its regular, italic and bold version during training, with characters of height equal to 30pt.

2.5. Index Construction

Once the text for a frame has been recognized, it is stored to be compared to the text extracted from neighboring frames for indexing. The comparison is performed according to the edit distance between

[2]http://en.wikipedia.org/wiki/Letter_frequencies

the strings of extracted text, normalized by the length of the strings themselves. If such distance is lower than a predefined threshold τ, the frames are considered as belonging to the same slide and grouped together. The longest string and the frame from which it was extracted is kept as reference for the slide. Retrieval of a certain concept can be performed, as exemplified in Figure 2, by looking for a query word among the strings chosen to be representative of their slides.

3. PERFORMANCE ANALYSIS

We analyzed videos containing 8 student presentations, for a total of 1 hour and 45 minutes of video. Each presentation has on average 13 slides, for a total of 2276 words and 13804 characters. In this Section we present the performances of our system in terms of localization of text regions, binarization quality and finally semantic concepts recognition (in particular, the text extracted from the slides). All the experiments were carried on a Pentium 4 2.33 GHz machine.

3.1. Detection

Text localization performances were tested on a set of 500 frames randomly selected (100 of which did not contain text), based on precision and recall measures. For each frame, Precision and Recall are defined as the intersection between the ground truth text area TA_{GT} and the text area estimated by the system TA_E, divided respectively by TA_E and TA_{GT}.

$$Precision = \frac{TA_{GT} \cap TA_E}{TA_E}, Recall = \frac{TA_{GT} \cap TA_E}{TA_{GT}}$$

Table 1 presents how the average Precision and Recall rates obtained by the system change with the application of the character and word recognition step. The $simple$ performances refer to the regions found by the simple text detector and successively passed to the word recognition block. The $refined$ precision and recall rates are calculated after the recognition block has either rejected or confirmed such regions as text (containing at least one recognized character). Following intuition, at the $refined$ stage Precision increases and Recall diminishes, as some candidate text regions (including some relevant ones) are rejected by the recognition step.

$Prec_{simple}$	Rec_{simple}	$Prec_{refined}$	$Rec_{refined}$
0.71213	0.85914	0.88584	0.68046

Table 1. Text Precision and Recall localization rates.

3.2. Binarization

In this Section we present a quantitative comparison of the three binarization algorithms mentioned in Section 2.3. The evaluation was performed on a subset of 54 regions localized by the text detection block and manually segmented with the aid of a heuristic visualization tool, in order to generate ground truth, for a total of 2177154 pixels. Precision and Recall metrics are defined similarly to what was described in Section 2.3, by re-defining TA_{GT} to be the set of ground truth pixels representing a character (foreground of the region) and TA_E as the set of pixels labeled as foreground by the algorithm. Table 2 presents the results. The performances of the different algorithms are comparable. The original Otsu algorithm is the fastest but has the lowest Recall and Precision, because it looks at every region globally and works well only up to a limited detail. The others provide higher precision, thanks to their focus on locality. It must be noted that our algorithm's results are comparable to those of Sauvola's method, which is known to be one of the best performing binarizations methods, but without the need for a predetermined threshold. We show two versions of the Local Adaptive Otsu algorithm (LAO): one with and one without the use of the integral histogram. The $Time$ results demonstrate the utility of integral histogram, which allows us to compute the optimal thresholds in a time which is independent from the window size. For all the local adaptive algorithms used in the experiments, we used a window of size 25x25, as it optimizes character and word recognition.

Algorithm	Precision	Recall	Time(sec)
Otsu	0.8611	0.8555	0.539
Sauvola ($k = 0.5$)	0.9003	0.8759	0.626
LAO	0.8831	0.9278	2.126
LAO + Integral Hist.	**0.8831**	**0.9278**	**1.29**

Table 2. Text binarization rates of the original Otsu algorithm, Sauvola's method as implemented in [8], and two implementations of our Local Adaptive Otsu (LAO) algorithm, with and without the use of integral histograms.

N_{gtc}	N_{raw}	N_{corc}	$Prec_c$	Rec_c	TCED
13804	3564	7376	0.5343	0.7446	6428
N_{gtw}		N_{corw}	$Prec_w$	Rec_w	
2276		1126	0.4947	0.6651	

Table 3. Character and Word Recognition rates per presentation. N_{gtc} : number of ground truth characters. N_{corc} : number of correctly recognized characters. N_{raw} : number of correctly recognized characters without adaptive binarization. TCED : Total Characters Edit Distance. N_{gtw} : number of ground truth words. N_{corw} : number of correctly recognized words. $Prec_c$, Rec_c, $Prec_w$, Rec_w refer respectively to Precision and Recall measures at character and word level.

3.3. Recognition

As explained in Section 2.4 the Tesseract OCR engine was used as a tool to recognize characters and words. We analyzed the performance of the system both at the word and character level. We defined the following metrics.

$$Precision_c = \frac{N_{corc}}{N_{gtc}}, \quad Recall_c = \frac{N_{corc}}{N_{rc}} \qquad (8)$$

with $N_{corc} = N_{gtc} - ED\,(s_g, s_r)$, N_{corc} is the number of correctly recognized characters, that is, the number N_{gtc} of ground truth characters minus the edit distance $ED\,(s_g, s_r)$ between the ground truth text s_g and the text output from the system s_r. N_{rc} is the number of recognized characters. Substituting the subscript c with w we obtain the same type of statistics at the word level, instead of character level ($Precision_w$ and $Recall_w$). N_{corw} is simply defined as the number of ground truth words correctly recognized by the system. We

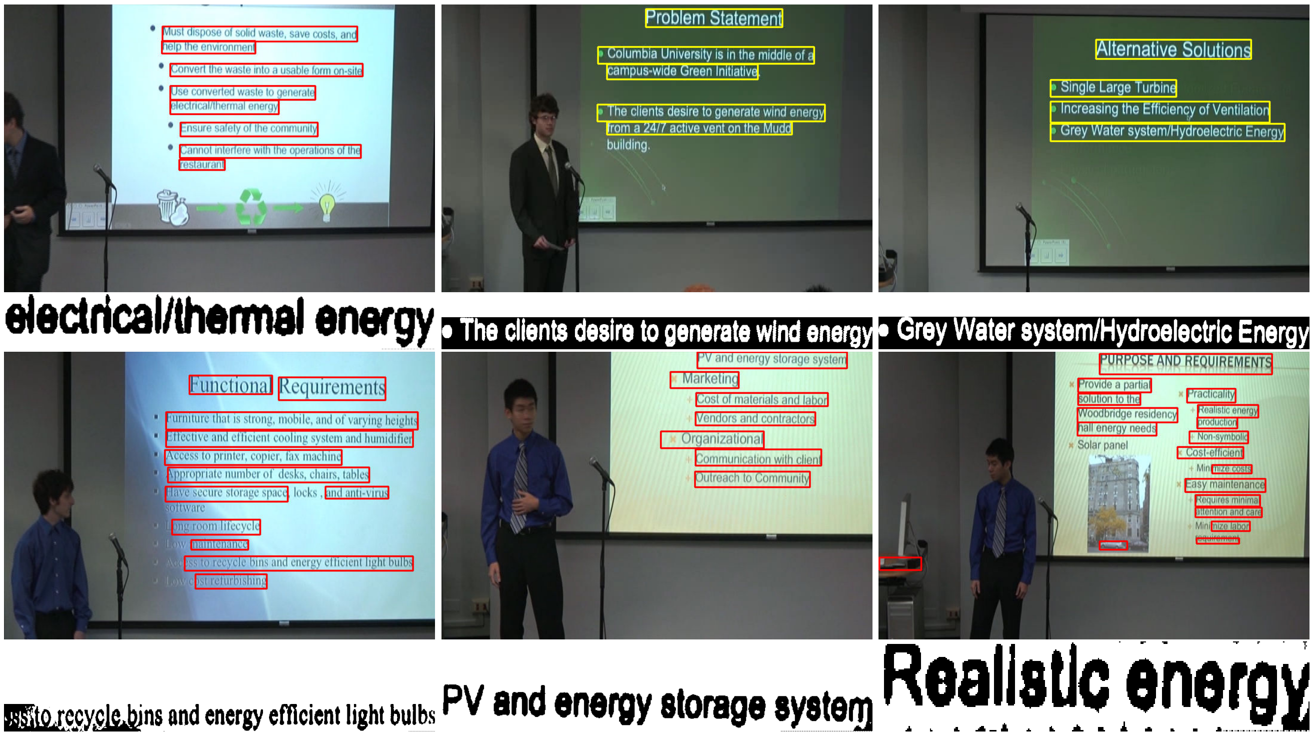

electrical/thermal energy
• The clients desire to generate wind energy
• Grey Water system/Hydroelectric Energy
ss to recycle bins and energy efficient light bulbs
PV and energy storage system
Realistic energy

Fig. 2. Example of indexing function: the word "Energy" is localized in different slides across 4 different presentations (top left, top center and right, bottom left, bottom center and right) and also within the same (top center and right, bottom center and right). In each frame are highlighted the localized text regions. Under every image the binarized version of the text region containing the word "Energy" (correctly recognized by the system) is presented.

compared the character and word recognition rates with and without our binarization preprocessing step. From Table 3, which reports the best performances obtained with the parameters set to the values reported in the previous Sections, can be appreciated how Tesseract, that has the Otsu binarization algorithm built in, benefit from our binarization process. It is interesting to notice that the word recognition rates are lower than their character equivalents. In fact, even if all its characters but one are correctly matched, a word is considered wrongly recognized. This suggests the use of ranking measures, such as the edit distance, which take into account also partial word matches in order to improve the quality of indexing and retrieval of semantic segments extracted from such videos. In such a way a system would be more flexible and robust to single or limited character recognition errors.

4. REFERENCES

[1] A. Haubold and J.R. Kender, "Analysis, user interface, and their evaluation for student presentation videos," *ICME*, pp. 863–866, July 2007.

[2] Liwei He, Elizabeth Sanocki, Anoop Gupta, and Jonathan Grudin, "Comparing presentation summaries: slides vs. reading vs. listening," in *SIGCHI conference on Human factors in computing systems*, New York, NY, USA, 2000, pp. 177–184, ACM.

[3] R. Lienhart and A. Wernicke, "Localizing and segmenting text in images and videos," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 12, no. 4, pp. 256–268, Apr 2002.

[4] Feng Wang, Chong-Wah Ngo, and Ting-Chuen Pong, "Structuring low-quality videotaped lectures for cross-reference browsing by video text analysis," *Pattern Recognition*, vol. 41, no. 10, pp. 3257–3269, 2008.

[5] Keechul Jung, Kwang In Kim, and Anil K. Jain, "Text information extraction in images and video: A survey," *Pattern Recognition*, vol. 37, no. 5, pp. 977–997, 2004.

[6] N. Otsu, "A threshold selection method from gray level histograms," *IEEE Trans. Systems, Man and Cybernetics*, vol. 9, pp. 62–66, mar 1979.

[7] Cha Zhang, Yong Rui, Jim Crawford, and Li-Wei He, "An automated end-to-end lecture capture and broadcasting system," *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 4, no. 1, pp. 1–23, 2008.

[8] Faisal Shafait, Daniel Keysers, and Thomas M. Breuel, "Efficient implementation of local adaptive thresholding techniques using integral images," in *Document Recognition and Retrieval XV*, San Jose, CA, Jan 2008.

[9] Fatih Porikli, "Integral histogram: A fast way to extract histograms in cartesian spaces," *CVPR Vol. 1*, pp. 829–836, 2005.

[10] Jo Mackiewicz, "Audience perceptions of fonts in projected powerpoint text slides," *Technical Communication*, vol. 54, pp. 295–307(13), August 2007.