SEMANTIC INDEXING FOR INSTRUCTIONAL VIDEO VIA COMBINATION OF HANDWRITING RECOGNITION AND INFORMATION RETRIEVAL

Lijun Tang and John R. Kender

Columbia University
Department of Computer Science
New York, NY 10027, USA
{ljtang, jrk}@cs.columbia.edu

ABSTRACT

Efficient indexing and retrieval of digital videos are important needs within instructional video databases. Semantic indexing for instructional videos can be achieved by combining the analysis of the instructor's handwriting in the video with domain knowledge taken from course support materials such as the course textbook, syllabus, or slides. We propose such a semantic indexing method, by combining handwritten word recognition with information retrieval techniques. We first present a novel handwritten word segmentation and recognition approach for instructional videos. Then we construct a table-of-contents (TOC) structure from course materials. We use word recognition results to query the TOC, implemented as matrix operations, and spot the most likely discussed chapters and topic words for each video. We evaluate the overall approach on 12 videos of two courses, and the results are encouraging.

1. INTRODUCTION

Universities take digital videos of courses for the purpose of being reviewed by students, or being used in a distance learning system. These videos, however, can only be viewed in a sequential manner, and the semantic content of each video remains unclear to the user until the user samples and understands several segments from the video. Moreover, if a student intends to take notes when viewing the video, the user must often wait, first for the instructor to start, and then to complete, the writing of a significant phrase. It would be very convenient if there were a tool to assist the user to locate text-filled frames, to automatically extract text regions, to generate notes for each video, and to relate those notes to other course materials. Content-based browsing based on videotext segmentation and recognition is therefore important to make the use of these videos more efficient.

In our instructional videos, almost all text is "scene text", and it is usually the instructor's handwriting. Onishi et. al. presented their work on text segmentation on a blackboard in a lecture room using a video image captured by a static camera [1]. Similarly, Liu et. al. proposed a rule-based semantic summarization of videos with handwritten notes, by analysis of the density of "ink pixels" [2]. These are both examples of image-based indexing. To provide semantic indexing, one solution would be to use Optical Character Recognition (OCR) to convert word images into ASCII. Existing OCR technology works well with good machine printed fonts against good clean backgrounds, but poorly if the text is handwritten. However, the recognition of handwritten words is difficult; it suffers of the so-called "segmentation/recognition dilemma"—where characters need both to be segmented before they can be recognized, and recognized before they can be segmented.

We propose a novel handwritten word recognition method especially adopted for our instructional videos. Because of the low spatial resolution and occlusion by the instructor, we do not use "online" methods for recognition. We use the "stroke" as a primitive unit for character segmentation, and train a multi-level perceptron neural network for the character recognition. For each candidate vocabulary word, we use a dynamic programming algorithm on the stroke sequence to incorporate character segmentation and recognition into one procedure, where we find the optimal segmentation and similarity score between word and image simultaneously (See Fig. 1).

Since our fundamental goal is to find handwritten evidence of the topics of a single course for the purpose of video indexing, we limit our domain knowledge to the support materials of the course. Candidate vocabulary words for handwriting recognition are extracted from Table of Contents (TOC) sources of the textbooks for the course, and from other related course documents like electronic slides. Using the handwritten recognition results, we are able to match the similarity of each video against each chapter of the textbook, via document query techniques. We regard a chapter as a document with its topic words as its content, and a given video can be used as a query by using the word recognition results from the video.

0-7803-9332-5/05/$20.00 ©2005 IEEE

2. NOVEL HANDWRITTEN TEXT RECOGNITION METHOD FOR INSTRUCTIONAL VIDEO INDEXING

Traditional handwritten word recognition methods separate character segmentation and character recognition into two independent procedures. Usually the segmentation algorithm segments the characters from the handwritten word image, and then Handwritten character recognition (HCR) is carried out on the segmented handwritten character. However, segmentation is a hard problem without any prior knowledge of the character recognition, and HCR will fail if characters are segmented at the wrong places. To overcome this dilemma, we use a dynamic programming algorithm to incorporate segmentation and recognition into one procedure. While searching for the optimal segmentation positions for a candidate vocabulary word, the algorithm makes use of the intermediate character recognition results. Finally, the optimal segmentation position and similarity score are returned simultaneously. The work flow of the system is illustrated in Fig. 1.

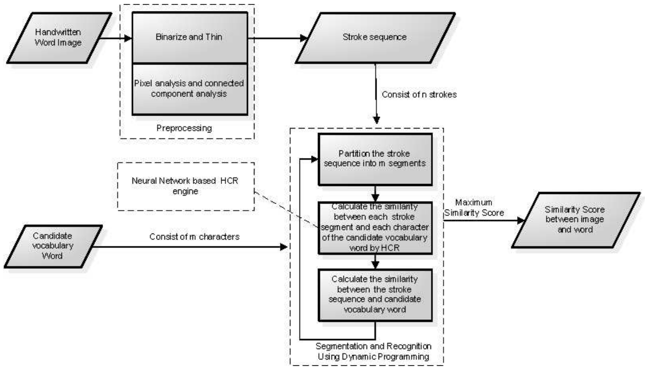
Handwritten Word Image
Binarize and Thin
Pixel analysis and connected component analysis
Preprocessing
Stroke sequence
Consist of n strokes
Neural Network based HCR engine
Candidate vocabulary Word
Consist of m characters
Partition the stroke sequence into m segments
Calculate the similarity between each stroke segment and each character of the candidate vocabulary word by HCR
Calculate the similarity between the stroke sequence and candidate vocabulary word
Segmentation and Recognition Using Dynamic Programming
Maximum Similarity Score
Similarity Score between image and word

Fig. 1. Handwritten word recognition

Feature	GSC	Zernike	KL Transform	Raw Binary
Size	512	115	256	256
Data Type	binary	float	float	binary
Upper case	92.4%	87.9%	86.7%	86.1%
Lower case	90.8%	89.4%	86.6%	86.4%

Table 1. Precisions of neural network-based handwritten character recognition engines with different feature extraction methods.

2.1. Neural Network-Based Handwritten Character Recognition

For character recognition, we first apply several popular feature extraction methods: Zernike [3], KL(Principal Component Analysis) transform [4], GSC [5], and binary raw data. Then we train and test a multi-layer perceptron neural networks (MLP NN) for each different feature extraction method. The best network and feature extraction method combination is finally chosen as the HCR module for our handwritten word recognition.

Neural networks have been used successfully in HCR field for years [6]. Since our training data is taken from the NIST Special Database 19 (SD19), we use a MLP NN as recommended in "Form-Based Handprint Recognition System" [6]. We also investigate feature extraction methods for HCR. Trier et al. discussed a number of methods in [7] and recommended Zernike moments [3] as features. Also, in the "Form-Based Handprint Recognition System", the coefficients of the Karhunen Loeve (KL) transform [4] were used as features. The GSC (Gradient, Structural and Concavity) features [5] are another feature set used extensively at SUNY Buffalo. We implemented these three feature extraction algorithms, together with using raw binary data as features directly (See Table 1).

We then trained and tested the MLP NN with each of the four feature extraction methods. The NN is a three layer back-propagation network with 128 hidden nodes. Its output units represent probabilities by training with $0 - 1$ target values. The training set we used are taken from Partitions HSF_4 and HSF_6 in SD19 (totaling 24,420 uppercase character samples and 24,205 lowercase character samples). The testing set is taken from Partition HSF_1 (totaling 61,303 uppercase character samples and 48,108 lowercase character samples). There are around 1,000 samples per character for training, and more than 2,000 samples per character for testing. The testing results are listed in Table 1. We trained two NNs, for 26 upper case characters and 26 lower case character separately. From Table 1 we observe that the GSC feature set has the best performance.

To classify a character image *without* prior knowledge of its lower or upper case, we can simply feed the character image into these two NNs and choose the maximum of the 52 resulting probabilities, return its corresponding character as recognized character. The precision is 64.7% for this method, which is comparable to a scheme that trains and tests one NN for all 52 characters. However, the top 2 precision is much higher, 91.9%. We found that this precision difference is due to the confusion of some essentially similar character pairs, such as 'o' and 'O', or 'i' and 'I', etc.

2.2. Stroke Based Segmentation and Handwritten Full Word Recognition

With the HCR engine trained, we use it as a separate module in a novel method that exploits the "stroke" as the primitive unit for segmenting handwritten full word images into discrete characters. A "stroke" is defined as sequence of pixels

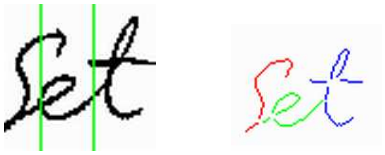

Fig. 2. Character segmentation for handwritten word “set”. Left image: Segmentation with vertical cut. Right image: Stroke-based segmentation, with different colors showing characters segmented at stroke boundaries.

representing one connected component in the skeleton image of handwritten word. We break the skeleton at all cross points to separate strokes from the skeleton. Segmentation based on strokes is a more precise way of segmenting characters from handwritten words, compared to segmentation by the standard vertical cut (see Figure 2). Additionally, segmentation based on strokes is much faster than the vertical cut method, since the number of strokes in each handwritten word image is usually much less than the number of possible vertical cut positions.

We use a dynamic programming algorithm to search for the optimal segmentation positions for a candidate vocabulary word, which makes use of intermediate character recognition results. We first sort the strokes extracted for each word image according to the x position of their centroid to construct a stroke sequence. We regard the segmentation and matching problem as an optimal partition problem of the stroke sequence. Suppose the length of the candidate word is m (i.e. sequence of characters $c_1, ..., c_m$) and the number of strokes in the stroke sequence is n. The problem can be described as: Given a sequence of strokes $s_1, ..., s_n$, partition the sequence into m segments $S_1 = (s_1, ..., s_{i_1 - 1}), S_2 = (s_{i_1}, ..., s_{i_2 - 1}), ..., S_m = (s_{i_m}, ..., s_n)$, such that $C[m][n] = \sum_{1 \leq j \leq m} d(c_j, S_j)$ is maximized, where $d(c_j, S_j)$ is the similarity score of matching stroke segment S_j against the character c_j of the candidate vocabulary word.

In order to use our NN-based HCR (in Section 2.1) to calculate $d(c_j, S_j)$, we recover a binary character image from the binary handwritten word image using the stroke segment S_j. We first dilate the stroke segment, and then logically “and” it with the binary handwritten word image. We feed this resulting binary character image into our NN-based HCR, and the probability for character c_j returned by HCR is the similarity score $d(c_j, S_j)$. Although we use a NN-based HCR module in the dynamic programming, any other HCR could be used to recognize the characters.

3. COMBINING HANDWRITTEN TEXT RECOGNITION AND INFORMATION RETRIEVAL

For domain knowledge, we construct a vocabulary of topics words for each course. Its size is essential to recognition performance. The larger the size, the lower the precision will be. But if it is too small, some handwritten word images will have no corresponding topics words in vocabulary.

3.1. Constructing the Topic Word Vocabulary

We extract topic words from the set of course documents available in electronic form: the table of contents (TOC) of the textbook, the online course syllabus, and any course electronic slides. We refine this raw data using techniques introduced by information retrieval. We first eliminate all function words, by applying a stop list of those words that have no meaning useful for indexing. In our experiment for an course in “Operating Systems”, this results in 481 topics words after this step. We tested this vocabulary by segmenting 1357 handwritten word images from 6 lecture videos of the same course, and obtained a recognition precision of 42.3%. This precision is imperfect in major part because there are many function words in the handwritten word images. However, as we now show, these results can be improved and are still useful for semantic indexing.

3.2. Constructing the “Word Stem-Frame” and “Word Stem-Chapter” Matrix

After the handwritten word recognition, we first construct an intermediate “word-frame” matrix, where each row of the matrix represents one topic word in the vocabulary, and each column represents one text key frame of a video. Ideally, each cell (i, j) should record a likelihood measure that topic word i appears in frame j, given by: $\sum_{\text{each word image k in frame j}} \Pr\{\text{image k recognized as word i}\}$.

To improve performance further, we use another Information Retrieval technique. We note that many of our content words are plural nouns or inflected verbs, causing confusion with their root words. We use a “Porter stemmer” to remove common endings from words, leaving behind an invariant root form. After stemming, there are only 395 word stems extracted from 481 topic words in our vocabulary. We therefore collapse the intermediate “word-frame” matrix into a final “word stem-frame” matrix, by merging rows corresponding to topic words with same word stem:

```
Frame:        1   2   3    4   5   6   7   ...

word stem1    0   0   0   .2   0   0   0   ...
word stem2    0   0  .1    0   0   0   0   ...
...
word stemm    0   0   0    0   0   0  1.3...
```

Lecture Videos	Chapters	Chapters Spotted	Top 8 Topic Words Spotted
video 6	Ch. 5	Ch. 4	DFT inverse spatial fast high implement pad filter
video 7	Midterm review	Ch. 3	Wiener zoom sharpen enhancement filter histogram problem transform
video 8	Ch. 5	Ch. 5	watershed wavelet Wiener zoom order noise read filter
video 9	Ch. 10	Ch. 10	histogram segment detect hough threshold transform edge line
video 10	Ch. 10&Ch.11	Ch. 10	link point region transform boundary edge merge threshold
video 11	Ch. 9	Ch. 9	scale thin dilate thicken component erose extract connect

Table 2. The spotting results for 6 lecture videos of a course in Digital Image Processing

Additionally, we construct a "word stem-chapter" matrix with the assistance of the TOC, where each cell (i, j) indicates the number of occurrences of word stem i in chapter j:

```
Chapter:       1  2  3  4  5  6  7  ...

word stem1     0  0  0  2  0  0  0  ...
word stem2     0  0  1  0  0  0  0  ...
...
word stemm     0  0  0  0  0  0  3  ...
```

3.3. Executing the Query

Following the method of information retrieval, we denote the "word stem-frame" matrix as the traditional "term-document" co-occurrence matrix A, and the "word stem-chapter" matrix as the querying matrix Q. When one multiplies matrix A^T by matrix Q, the column vectors of the resulting matrix indicate the confidence of topics in each chapter being discussed in the video. The chapter with the maximum confidence value is regarded as spotted chapter for the video, and the topic words which appear in both the spotted chapter and the video are regarded as spotted topics.

4. EXPERIMENTAL RESULTS AND DISCUSSION

Table 2 illustrates the spotted chapters and topic words for 6 lecture videos of a course in "Digital Image Processing". (The full course has 13 videos and there are 12 chapters in TOC, but the mapping is clearly not one-to-one.) The ground truth chapter for the videos are listed in second column of Table 2, they are "chapter 5, midterm review, chapter 5, chapter 10, chapter 10 and 11, chapter 9" for 6 lecture videos respectively. Our spotted results for Video 6 and Video 7 are incorrect, but Video 7 is a review lecture with no given chapter, and the chapter spotted for Video 6 has many topic words in common with the correct chapter.

We believe we can significantly improve our results by normalizing large chapters with many keywords, which currently overwhelm smaller chapters in our result set. We also can apply more advanced term weighting and filtering approaches. We expect to be able to then determine not only the chapter, but also the section of the TOC is most likely discussed in each of several sections of each video. We are building novel graphic user interfaces to present these results, and will conduct user studies to evaluate them.

5. REFERENCES

[1] Masaki Onishi, Masao Izumi, and Kunio Fukunaga, "Blackboard segmentation using video image of lecture and its applications," in *Proc. 15th International Conference on Pattern Recognition(ICPR2000)*, September 2000, vol. 4.

[2] Tiecheng Liu and John R. Kender, "Rule-based semantic summarization of instructional videos," in *International Conference on Image Processing*, 2002.

[3] A. Khotanzad and Y.H. Hong, "Invariant image recognition by zernike moments," *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 12, no. 5, pp. 489–487, May 1990.

[4] P. J. Grother, "Karhunen loeve feature extraction for neural handwritten character recognition," in *Proceedings of Applications of Artificial Neural Networks III*, Orlando,FL, USA, April 1992, vol. 1709, pp. 155– 166.

[5] Stephen W. Lam Geetha Srikantan and Sargur N. Srihari, "Gradient-based contour encoding for character recognition," *Pattern Recognition*, vol. 29, no. 7, pp. 1147 – 1160, July 1996.

[6] M. D. Garris, C. L. Wilson, and J. L. Blue, "Neural network-based systems for handprint ocr applications," *IEEE Trans. Image Processing*, vol. 7, pp. 1097–1112, Aug 1998.

[7] D. Trier, A. K. Jain, and T. Taxt, "Feature extraction methods for character recognition - a survey," *Pattern Recognition*, vol. 29, no. 4, pp. 641–662, April 1996.