THE USE OF NONLINEAR FILTERING IN AUTOMATIC VIDEO TITLE CAPTURE

Sharmila Kannangara[†], *Eduardo Asbun*[†], *Robert X. Browning*[‡], *and Edward J. Delp*[†]

[†] Video and Image Processing Laboratory (*VIPER*)
[†] School of Electrical and Computer Engineering
[‡] Department of Political Science
[‡] Public Affairs Video Archives (PAVA)
Purdue University
West Lafayette, Indiana 47907-1285
USA

ABSTRACT

The recognition of text overlay information appearing in video frames can be used for classification and scene indexing for archival purposes. In this paper, an algorithm for text recognition in C-SPAN images is presented. A method for the segmentation of text blocks into individual letters is outlined. A recognition method using shape sensitive morphological operations is presented.

1. INTRODUCTION

The identification of overlay text information in a video frame can be very useful for indexing a frame in a sequence. [1, 2] Optical character recognition research has presented a plethora of algorithms varying in complexity, reliability and speed [3]. In this paper we describe a fast yet accurate morphological character recognition algorithm to recognize text appearing in C-SPAN video.

The Public Affairs Video Archives (PAVA) was established at Purdue University to record, index, and archive all C-SPAN programming. Over 70,000 hours of C-SPAN programming, every program aired since 1987, are contained in the Archives and immediately accessible through the database and electronic archival systems developed and maintained by PAVA.

The Archives records both C-SPAN networks seven days a week, 24-hours a day. Programs are extensively indexed making the database of C-SPAN programming an unparalleled chronological resource. Programs are indexed by subject, speaker names, titles, affiliations, sponsors, committees, categories, formats, policy groups, keywords, and location.

The goal of the work described here is to provide a method to automate some of the indexing tasks now done manually by the PAVA staff. Initial research focused on the text that was presented on a blue background in C-SPAN video.

2. TEXT EXTRACTION AND SEGMENTATION

An original C-SPAN video frame (640×480 pixels) is shown in Figure 1. The goal is to identify the overlayed characters so that this information can be used to index the frame. In order to obtain an image with the greatest contrast between text and background, the red component was used.

Empirical studies revealed that the text was located in a region bounded by the pixel locations from 369 to 392 (vertically) and from 58 to 470 (horizontally). These boundary values varied by approximately 2 pixels from frame to frame due to timing errors. This constancy of location was used to segment the text block from the overall image. Figure 2 is the segmented text block region of a sample image.

The segmentation of the individual letters from the text block was accomplished using the recursive algorithm described below. The text block image was thresholded at pixel intensity 102. Figure 3 shows the thresholded text block of the original image.

The vertical projection profile [3] of the resulting image was obtained by adding the pixel intensity va-

This work was supported by a grant from the Public Affairs Video Archives (PAVA) through the C-SPAN Educational Foundation. Address all correspondence to E. J. Delp, ace@ecn.purdue.edu, http://www.ece.purdue.edu/∼ace, or +1 765 494 1740. The Public Affairs Video Archives can be reached at http://www.pava.purdue.edu/.

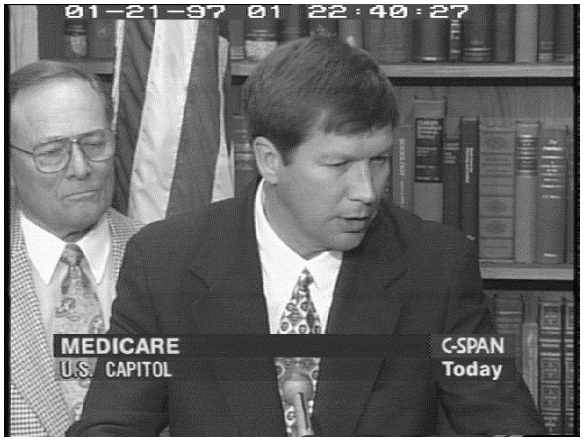

Figure 1: Original image

lues along each column of pixels (Figure 4). As can be seen from this plot, it is relatively easy to locate the boundaries between each letter in the text block. The boundaries correspond to the minima in the plot.

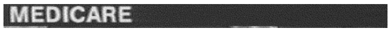

Figure 2: Text block of original image

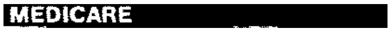

Figure 3: Thresholded text block

The first six values of the vertical projection profile were forced to zero. This was done in order to ensure that any non-uniform background included in the text block segmentation due to text block position variation from frame to frame did not affect the segmentation of the individual letters. Through empirical studies it was determined that the first letter in the text block started after the sixth pixel in the block. Therefore, no information from individual letters was lost due to this procedure.

The individual letters were segmented using the vertical projection profile values. The threshold for the boundary between letters was set at vertical projection profile value 2.6. (The area in the image that belonged to regions with vertical projection profile values lower than 2.6 was labeled as the boundaries between letters.)

From empirical studies, the maximum allowable width for a segmented letter was determined to be 28. A check was performed on the results of the segmentation and if any segmented letter exceeds the maximum width, that area of segmentation is re-thresholded (with threshold value incremented by pixel intensity 1.3), its vertical projection profile re-computed and the section re-scanned for new boundary positions. The procedure of re-thresholding serves to thin the letters and thereby enlarge the boundary widths between them. Two letters that may have merged together after the initial thresholding procedure will have a boundary between them after successive iterations of this procedure. The procedure of re-thresholding, re-computing the vertical projection profiles and segmenting is repeated until no segmented area is wider than 28 pixels.

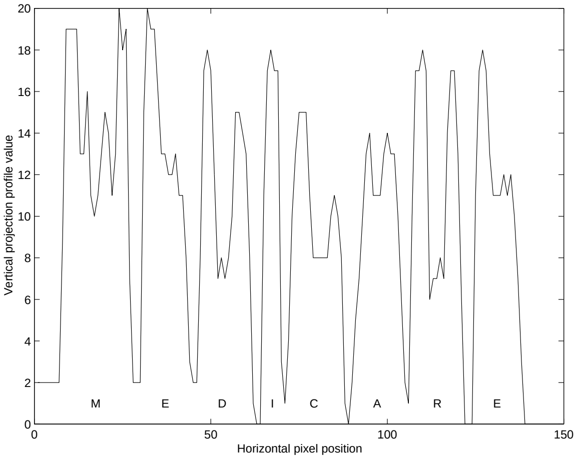

Figure 4: Vertical projection profile for "MEDICARE"

3. CHARACTER RECOGNITION

3.1. Correlation

Initial efforts at recognizing the segmented letter with respect to an alphabet was focused on using a correlation method. The text appearing in the area under consideration was composed using a fixed font. Utilizing this fact, a 26 letter alphabet of templates was created manually, based on sample images. On average the dimensions of an individual letter were 17×17 pixels. Each of the 26 templates was correlated with sample letters from images.

The results were inspected in order to determine whether there was sufficient difference in the correlation measure so that alphabet letters and segmented letters matched uniquely (either individually or in classes of letters). As Figure 5 shows, the results of the correlations were too close (for **H**, **E**, **N** and **R**) to use in a decision criteria with sufficient probability of success. This was due to the small size of the letters and blurring around the edges of the letters.

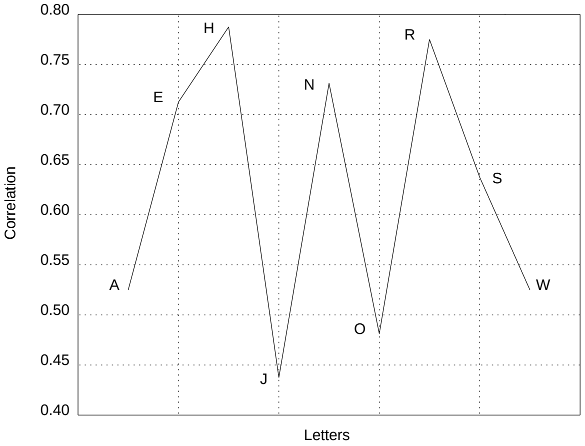

Figure 5: Correlation between "**H**" and other letters

3.2. Morphological Filtering and Detection

The failure of correlation methods lead to our focusing on using morphological methods [4, 5] to perform the character recognition.

Structuring elements corresponding to the 25 letters of the alphabet (all letters except **I**) were created manually. These structuring elements were used to perform an opening operation on the segmented letters from the image. We call this operation "checking the segmented letter for a match with the structuring element." The result of the opening was thresholded (to zero and one) to remove false residual artifacts. The thresholded image was inspected for the number of surviving (non-zero) pixels. If the number of surviving pixels exceeded the number of non-zero pixels in the structuring element used, a match between the segmented letter and the structuring element was accepted.

The opening operation had to be performed in a specific order to minimize misclassification of certain letters. For example, there was a high rate of confusion between **O** and **C**. This was due to their similar structure. An **O** could be confused for a **C**, but rarely was a **C** confused for an **O**. Therefore, an order in performing the identification of letters was devised. The segmented letter was checked for a match with **O** first. If no match was determined, it was checked for a match with **C**. However, if a match for **O** was found, no check would be performed against **C**. This "path order" approach is shown by the matrix in Figure 6. Note that the column under **Q** is shaded gray as no structuring element was available for **Q** and thus no letter was checked against it.

The non-shaded areas of the matrix correspond to possible transitions from the letters on the rows to those on the columns. For example, if a match were found with **B**, we still perform checks for matches against all letters except **E** and **T**.

	A	B	C	D	E	F	G	H	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
B					■											▨			■						
R						■									■	▨			■						
P						■										▨			■						
E																▨									
O			■													▨									
D																▨									
G			■													▨									
M																▨									
W																▨						■			
N								■								▨									
H																▨									
A																▨						■			
S																▨									
C																▨									
U																▨									
F																▨			■						
K																▨									
X																▨									
Z																▨									
J																▨									
V																▨									
Y																▨									
T																▨									
L																▨									

Figure 6: Path order matrix

The match with the maximum number of surviving pixels is declared as the final match for a specific segmented letter and the ASCII version of that letter is stored in the appropriate position in the text string.

For the case of the letter **I** and punctuation marks (**.** **,** and **'**), the opening approach was not used. Instead if a segmented letter had a width of less than 10 pixels, it was declared to be either the letter **I** or a punctuation mark. Further inspection of the location of the maximum number of high valued pixels in the letter/punctuation was used to decide between the choices. Periods and commas have high value pixels located in the lower half of the segmented region, quotes have them located in the upper half and the letter **I** has them located throughout. Figures 7 – 9, show the original image, structuring element and the result of opening (with a successful match) for three letters in the string "MEDICARE". Figure 10 shows the result of an unsuccessful match.

Figure 7: Original segmented letters

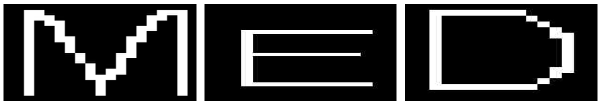

Figure 8: Structuring elements

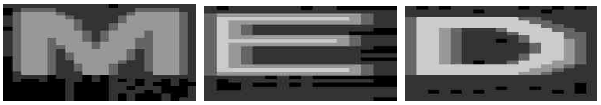

Figure 9: Result of opening

4. RESULTS AND IMPROVEMENTS

4.1. Results

Tests were run on 58 frames using the algorithm shown in Figure 11. The results are shown in Table 1. *Err Hits* corresponds to mistakes made in matching (for example, in picture No. 47, **J** was mistaken for a **U**). *Misses* refer to letters that were not matched as any letter (signified with **?** in the *result* column). *Seg* corresponds to errors due to faulty segmentation of letters (i.e., either letters were merged together or single letters were fragmented).

The total error rate was 11.8%. Of that, 64% was due to mistakes made in matching, 28% was due to errors in segmentation and the remaining 8% was due to letters that were not matched at all. It is clear from the results that there is a low success rate in identifying the letter **A**. This could be due to the noise in the original image. Distinguishing **K** and **C**, **V** and **W**, and **J** and **U** is also error prone.

The average length of a processed text string was 12 characters. The running time on 75% of the CPU on a SUN workstation was on average 3 seconds per text string.

4.2. Improvements

This paper has presented a simple and fast morphological method for character recognition. A success rate of almost 90% has been obtained. The current metric for deciding a match, the number of surviving pixels, could be replaced by a metric that better reflects the accuracy of the matches.

The method of using structuring elements that correspond to the letters of the alphabet uniquely could be improved by using structuring elements that correspond to classes of letters. For instance: a round structuring element could be used to represent classes that include **O**, **D**, **C** and **G**, while a more linear one could correspond to a class containing **R**, **B**, **P** and **T**. In this way, the path order could be replaced by a decision tree.

Figure 10: Result of opening letter M with structuring element E

Furthermore, a probabilistic approach could also be used for the detection of letters. The *a priori* probability of a certain letter appearing could be computed via an empirical method. The result could then be applied to compute the likelihood of detections.

Once a text string is completely or even partially detected, it could be checked against a set of dictionary entries. This would serve to verify that the detection is a word or name, and also would serve to complete any partial detections.

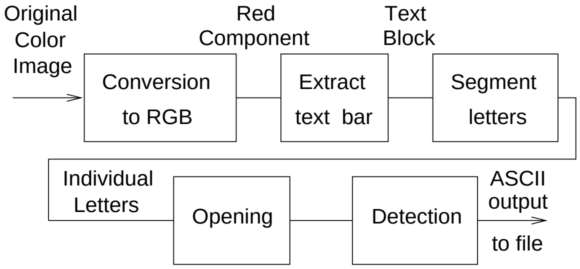

Figure 11: Block diagram of the algorithm

5. REFERENCES

[1] R. Lienhart, "Automatic text recognition for video indexing," *Proceedings of the Fourth ACM International Conference on Multimedia*, November 18–22 1996, Boston, Massachussetts, pp. 11–20.

[2] J. Ohya, A. Shio, and S. Akamatsu, "Recognizing characters in scene images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, no. 2, pp. 214–220, February 1994.

[3] L. O'Gorman and R. Kasturi, *Document Image Analysis.* Los Alamitos, California: IEEE Computer Society Press, 1995.

[4] R. M. Haralick and L. G. Shapiro, *Computer and Robot Vision*, vol. I. Reading, Massachussetts: Addison-Wesley, 1992.

[5] J. Song and E. J. Delp, "A study of the generalized morphological filter," *Circuits, Systems, and Signal Processing*, vol. 11, no. 1, pp. 229–252, 1992.

Table 1: Results of 58 trials

Pict No.	Original Text	Result	Err Hits	Misses	Seg
41	MEDICARE	MEDICTRE	1	0	0
42	REP. NANCY PELOSI	REP NANCY PELOSI	0	0	1
43	WILLIAM STRAUSS	Y.LL.TM STRAUSS	4	0	0
44	WASHINGTON, DC	WASHINGTON. DC	0	0	0
45	PAUL O'DAY	PAUL OIDCY	2	0	0
46	VAHID MOTEVALLI	VAHID MOTEV?LLI	0	1	0
47	JOHN HARRALD	UOHN HARRALD	1	0	0
48	ALEX SCHMID	ALEX SCHMID	0	0	0
49	RONALD CRELINSTEN	RONALD CRELNSTEN	0	0	1
50	SHELDON KRYS	SHE.. IION CRYS	1	0	3
51	VICTORIA CUMMOCK	VICTOHIA CUMMOCK	1	0	0
52	THE WHITE HOUSE	THE WHITE HOUSE	0	0	0
53	MIKE McCURRY	MIKE MTCURRY	0	0	0
54	KENNETH BACON	KENNETH BTCON	1	0	0
55	SANDY BERGER	SANDY GERKER	2	0	0
56	REP. VIC FAZIO	REP VIC FAZIO	0	0	1
57	ROBERT RUBIN	ROBERT PUBIN	1	0	0
58	PAUL-LOUIS ARSLANIAN	PAUL.LOUIS VRSLVNIAN	2	0	0
59	REP. BILL PAXON	REP BILL PCXON	1	0	1
60	U.S. CAPITOL	UIS. CAPFOL	2	0	1
61	FRANKLIN RAINES	FRANKLN RTINET	2	0	1
62	KATHLEEN FLYNN	CATHLEN FLNN	1	0	2
63	ARLINGTON, VA	ARLNGTON. WA	1	0	1
64	JOHN HAGAN	JOHN HAGAN	0	0	0
65	REP. STENY HOYER	REP SYEJY LOYER	3	0	1
66	THE PENTAGON	THE PENTAGON	0	0	0
67	DAVID CHANCE	DAWID CHANCE	1	0	0
68	LONDON	LONDON	0	0	0
69	LESLIE HILL	L?LE HILL	0	1	2
70	BARRY COX	BARRY CDX	1	0	0
71	SANDY BERGER	SANDY BERKER	1	0	0
72	DEBORAH ORIN	DEBORVH ORIN	1	0	0
73	ROB HOUSEMAN	PCE HOUSEMTN	4	0	0
74	DAVID NORCROSS	DTVID NORCROSS	1	0	0
75	BENNETT JOHNSTON	BENNETT JOHNSTON	0	0	0
76	CARL ROCHELL	C?RL ROCHELL	0	1	0
77	ROGER WILKINS	ROGER W.LCIN?	2	1	0
78	DIANA FURCHTGOTT-ROTH	DIANJ FURCHTGOTTROTH	1	0	1
79	ROGER WILKINS	ROGER W.LKINS	1	0	0
80	CYCLES OF HISTORY	CYCLS OF HISTOPY	1	0	1
81	NEIL HOWE	NEIL HOWE	0	0	0
82	WASHINGTON. DC	WCSHINGTON. DC	1	0	0
83	JAMES FALLOWS	UAWES FALLOWS	2	0	0
84	STATE OF THE NEWS MEDIA	STVTE OF THE NEWS MEDIA	1	0	0
85	JAMES FALLOWS	JAMES FAL. I IWS	0	0	2
86	CAMBRIDGE, MA	CAMBRIDGE. MA	0	0	0
87	WARREN CHRISTOPHER	W?RREN CHRISTOPHER	0	1	0
88	MEXICAN LOAN REPAYMENT	MEXICAN LOJN REPAYMENT	1	0	0
89	DAVID NORCROSS	D?VID NORCPOSS	1	1	0
90	LAWRENCE SUMMERS	LAWRENCE SUMMERS	0	0	0
91	SEN. JOSEPH BIDEN	SEN. UOSEPH GIDEN	2	0	0
92	SEN. JOHN BREAUX	SEN. JOHN BREAUX	0	0	0
93	SEN. PATRICK LEAHY	SEN. PATRICK LVHY	0	0	1
94	JFK SCHOOL OF GOVERNMENT	JFC TKHOOL OF GOERNMENT	3	0	1
95	HAZEL O'LEARY	H..ZEL OIETRY	2	0	2
96	HAROLD SMITH	HAROLD CMITH	1	0	0
97	TARA O'TOOLE	TVHA OTOOE	2	0	2
98	SEN. JOSEPH BIDEN	SEN. UOSEPH BIDEN	1	0	0