Time and Date OCR in CCTV Video

Ginés García-Mateos[1], Andrés García-Meroño[1], Cristina Vicente-Chicote[3], Alberto Ruiz[1], and Pedro E. López-de-Teruel[2]

[1] Dept. de Informática y Sistemas
[2] Dept. de Ingeniería y Tecnología de Computadores,
Universidad de Murcia, 30.170 Espinardo, Murcia (Spain)
[3] Dept. Tecnologías de la Información y Comunicaciones,
Universidad Politécnica de Cartagena, 30.202 Cartagena, Murcia (Spain)
{ginesgm, aruiz, pedroe}@um.es, andres.garcia2@carm.es,
cristina.vicente@upct.es

Abstract. Automatic recognition of time and date stamps in CCTV video enables the inclusion of time-based queries in video indexing applications. Such ability needs to deal with problems of low character resolution, non-uniform background, multiplexed video format, and random access to the video file. In this paper, we address these problems and propose a technique that solves the difficult task of character segmentation, by means of a recognition-based process. Our method consists of three main steps: pattern matching, character location and syntactic analysis. The experiments prove the reliability and efficiency of the proposed method, obtaining an overall recognition rate over 80%.

1 Introduction and Related Research

Optical character recognition in digital video (in short, video OCR) has received an increasing interest in the last few years [1]. Some of its applications include video indexing and digital libraries [2,3,4], commercial segments detection [2], sign detection and recognition [6], MPEG-7 text descriptions and video summarization [5]. In this paper, we address the problem of time and date stamp recognition in CCTV (Closed Circuit TV) surveillance video. While the problem is still in the scope of video OCR, some peculiarities suggest the development of specific methods, as we will discuss.

Although digital CCTV systems will eventually supersede analogic CCTV, at present, analogic systems are far more commonly used in supermarkets, banks, traffic control, and similar applications. A typical CCTV system consists of several cameras, placed either indoors or outdoors. Their outputs are multiplexed and recorded in a single analogic videotape, i.e. successive frames correspond to different cameras, as shown in Fig. 1. Additionally, the multiplexer adds to each frame a time and date stamp, and optionally the camera number.

The video OCR presented here was designed to be integrated in a bigger system[1], which is able to digitize, demultiplex and process analogic tapes of this

[1] Owned by Vision Base Int. Ltd. We want to thank their support in this research.

F. Roli and S. Vitulano (Eds.): ICIAP 2005, LNCS 3617, pp. 703–710, 2005.

Fig. 1. Sample time and date stamps in consecutive frames. Image resolution is 704x286 pixels (1 image = 1 field). Input video frequency is 4 fps.

kind of systems. The aim of our work is to add time and date stamps recognition functionality, allowing time-based queries. These queries could be of the form: "select the frames between time t_1 and t_2", "move to instant t", or "show only the frames from camera number n".

It is widely accepted that the two main problems in video OCR are complex background and low resolution [1,2]. The former complicates segmentation, since character and background pixels have similar values, while the later means that recognition is very sensitive to noise. Time and date OCR suffers from both problems, and has to deal with two added difficulties. First, the video is multiplexed, which involves the lack of continuity between consecutive frames. Second, queries require a random access to any part of the video. Considering also the low input frequency –which in CCTV systems is usually from 1 to 5 fps–, we adopt the premise that each frame has to be recognized individually.

On the other hand, the text to recognize is not a mere sequence of characters, but a valid date and time. Different syntaxes of time and date have to be considered, which include: time and date in one or two lines; date above or below; numeric or alphabetic month representation; seconds with one decimal digit; order of day and month in the date; different separators, etc.

The rest of the paper is organized as follows. Section 2 gives a general overview of the method. Next, we describe our solution in sections 3, 4 and 5, detailing the three main steps: pattern matching, character location and syntactic analysis. Finally, we present some experimental results and conclusions.

2 System Overview

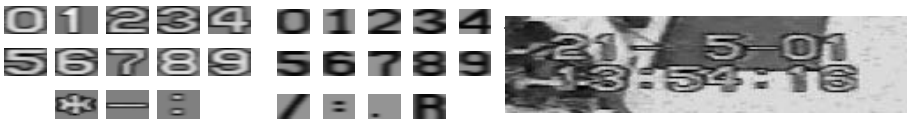
Time and date OCR in CCTV video has to cope with high variability in background, location of the stamp in the image, font type, and syntactic format. However, in a single video sequence all of them, except background, suffer no change. Thus, we decompose the problem into detection and updating. In the detection phase, a costly search through all possible locations, fonts and formats is applied to select the most likely combination[2]. After that, the updating phase performs an easy and efficient computation of the current time and date.

[2] By application requirement, a region of interest (ROI) is supposed to be manually selected by the user in the images.

Since segmentation is not feasible under the existing background conditions, we propose a method which does not require prior segmentation; it can be considered a case of recognition-based segmentation [2], where segmentation takes place only after a pattern matching process. The detection phase consists of three main steps. First, pattern matching is applied in order to detect the characters of the stamp. Second, characters are located and arranged in a string, according to the maximum values of matching. Third, syntactic analysis is performed, selecting the most likely representation of time and date among a predefined set of valid formats.

3 Pattern Matching

Time and date stamps are superimposed to the video signals using a reduced number of standard font types, specific of each manufacturer. Fig. 2 shows two of these types. Contrary to printed text OCR, neither rectification nor scale are needed, as font size does not change and the baseline is always horizontal. Under both conditions, we can apply a simple pattern matching on the ROI, to detect characters of the predefined font types.



a) Set 0 (24×16) b) Set 1 (18×14) c) Sample stamp (228×56)

Fig. 2. Two font sets and a sample time and date stamp. Fonts and stamp sizes (width×height) are indicated below.

Patterns of the characters are compared through all the ROI using a normalized coefficient matching, defined as follows. Let i and t be the image and the pattern, respectively, and let t be of size $w \times h$. The normalized patches of i and t are given by:

$$i'(x,y) = i(x,y) - \frac{1}{wh} \sum_{a=0}^{w-1} \sum_{b=0}^{h-1} i(x+a, y+b) \tag{1}$$

$$t'(x,y) = t(x,y) - \frac{1}{wh} \sum_{a=0}^{w-1} \sum_{b=0}^{h-1} t(a,b) \tag{2}$$

That is, both i' and t' have zero mean brightness. Matching value is a cross correlation of these normalized patches:

$$m_{i,t}(x,y) = \frac{\sum_{a=0}^{w-1} \sum_{b=0}^{h-1} t'(a,b) i'(x+a, y+b)}{\sqrt{\sum_{a=0}^{w-1} \sum_{b=0}^{h-1} t'(a,b)^2 \sum_{a=0}^{w-1} \sum_{b=0}^{h-1} i'(x+a, y+b)^2}} \tag{3}$$

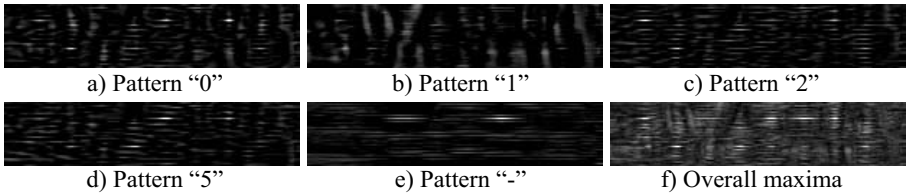


Fig. 3. Matching results of stamp in Fig. 2c) with some patterns of the set shown in Fig. 2a). Time and date digits are superimposed on their original locations, for visualization purposes. White means a better matching, and it should be located on the upper-left corner of the characters.

Fig. 3 presents some sample results of this step. In general, local maxima appear at the expected positions. However, we can observe that many spurious local maxima exists, thus complicating the task of character location.

Obviously, pattern matching is the most time-consuming step of the process. In the detection phase –when no information concerning time and date is assumed–, all patterns from all font types are matched with the image. In the following frames, only a few patterns are matched in small parts of the ROI, as far as font type and locations of characters are known.

4 Character Location and Segmentation

After the pattern matching step, we obtain a set of *matching maps*, $m_{i,t}$, indicating the likelihood that each character t is present at every possible location of the ROI. However, extracting a coherent string of characters is not a trivial task. As we have seen in Fig. 3, most of the existing local maxima do not correspond to real character locations. Moreover, some character patterns produce very low maxima even in the correct locations; see, for example, number "1" in Fig. 3b).

We propose a method to cope with both problems, producing a string of locations for a given font set. The resulting string is stored in a matrix[3], p, of size $\lfloor W/w \rfloor \times \lfloor H/h \rfloor$, where $W \times H$ is the ROI size, and $w \times h$ is the font size. Our method is based on the following greedy algorithm.

Let us consider a font type s consists of a set of character patterns, $s = \{s_0, s_1, ..., s_9, s_{10}, ..., s_k\}$, where s_0 to s_9 are the corresponding decimal digits, and s_{10} to s_k are separators. All these patterns have the same size, $w \times h$. We postulate that the maximum matching through all possible locations and patterns in s is a correct recognition. That is, the upper-left corner of the first segment is:

$$(x_0, y_0) = \arg \max_{\forall x,y} \{ \max_{\forall c} (m_{i,s_c}(x, y)) \} \tag{4}$$

The assumption that (x_0, y_0) is a correct segment, is the starting point of the algorithm. This way, cell $p(a, b)$ is set to (x_0, y_0), with $a = \lfloor x_0/w \rfloor$ and $b =$

[3] Recall that time and date can appear in one or two rows.

$\lfloor y_0/h \rfloor$. Since all characters are equally spaced, the left adjacent segment, $p(a-1,b)$, should be located in (x_0-w, y_0), and the right one in (x_0+w, y_0). However, this fixed jump of w pixels could accumulate small errors, producing a bad segmentation of distant characters. Thus, we compute the maximum matching in a certain tolerance region, of size $r_x \times r_y$:

$$(x_1, y_1) = \arg \max_{\forall x \in R_x, \forall y \in R_y} \{\max_{\forall c}(m_{i,s_c}(x,y))\} \tag{5}$$

with:

$$R_x = \{x_0 + w - r_x, ..., x_0 + w + r_x\}; R_y = \{y_0 - r_y, ..., y_0 + r_y\} \tag{6}$$

for the right adjacent character. Then, segment $p(a+1,b)$ is given by:

$$p(a+1,b) = \left\{ \begin{array}{cc} (x_1, y_1) & if \max_{\forall c}(m_{i,s_c}(x_1,y_1)) > threshold \\ (x_0+w, y_0) & otherwhise \end{array} \right\} \tag{7}$$

That is, the maximum is not considered if it is below a given threshold; instead, the predefined width w is used. This condition is necessary to avoid low spurious maximums that appear in the place of blank spaces (see an example in the month field in Fig. 2c). The process is repeated to the left, to the right, and then above and below, until the matrix of locations p is completed. Fig. 4 shows the results of this algorithm when applied on the stamp in Fig. 2c).

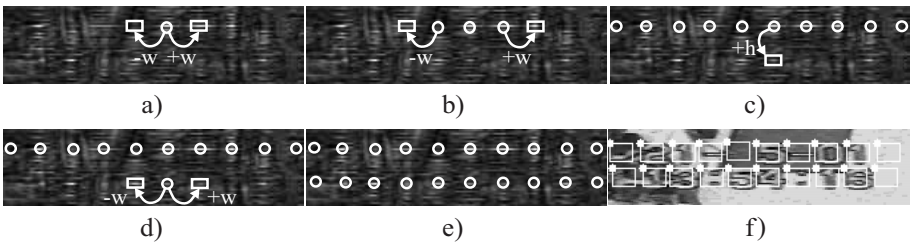


Fig. 4. Character location and segmentation algorithm. Starting from the overall maximum matching, a), the process moves left and right searching for local maxima, b) and c), until it reaches the extremes. Then, we move below and repeat the process, d) and e). The resulting segmentation is shown in f).

5 Syntactic Analysis

The output of a generic OCR would basically consist of taking the characters that produce the highest matching for every segment in string p. But, in our case, syntactic analysis is required to enforce an interpretation of the result as a valid time and date. Two problems are involved: *where* are located time and date in p, and *what* time and date format is used. We consider a predefined set of

valid time and date formats. Syntactic analysis consists of searching the format that better fits in the computed string, according to the matching values.

Each time and date format is described as a string of format elements, where a format element represents a set of valid characters. We have also defined *double* format elements, that stand for sets of valid pairs of characters –e.g., any pair from "00" to "23"–. We distinguish 6 single and 6 double format elements, as shown in Table 1. Using these format elements, three possible time formats could be "[00-23]:59", "[00-23]:59:59" and "[00-23]:59:59.9". In the first case, only hour and minutes appear, while the third represents seconds with one decimal digit.

Table 1. Format elements used to represent dates and times. Left: a single element stands for a set of valid characters. Right: a double element stands for a valid pair of characters.

Single	Set of characters, $set(.)$										Double	Corresponding
element	blank	0	1	2	3	4	5	6	7	8	9	element	pairs of characters
"*"	x											"[0-23]"	*0, *1, *2, ..., 23
"1"		x	x									"[00-23]"	00, 01, 02, ..., 23
"2"		x	x	x								"[1-12]"	*1, *2, *3, ..., 12
"3"		x	x	x	x							"[01-12]"	01, 02, 03, ..., 12
"5"		x	x	x	x	x	x					"[1-31]"	*1, *2, *3, ..., 31
"9"		x	x	x	x	x	x	x	x	x	x	"[01-31]"	01, 02, 03, ..., 31

We define the likelihood that each character, single or double element is present at every segment –or pair of segments, in the last case– of a string. The likelihood of a particular character t in position a of the string is given by $l(t,a) = m_{i,t}(p(a,0))$, assuming the stamp occupies one single row. The likelihood of a format element e is given by $l(e,a) = max_{t \in set(e)} l(t,a)$, where $set(e)$ is the set of characters associated to e, as shown in Table 1. We pinpoint the special case of the blank space, whose likelihood is $l(\text{“}*\text{”},a) = 1 - max_{\forall t} l(t,a)$. Finally, the likelihood of a double format element accounts for all possible combinations of pairs; for example, the likelihood of "[0-23]" is given by $l(\text{“}[0-23]\text{”},a) = max\{l(\text{“}*\text{”},a) + l(\text{“9”},a+1), l(s_1,a) + l(\text{“9”},a+1), l(s_2,a) + l(\text{“3”},a+1)\}$.

Moveover, the likelihood that a time or date format, f, is present in p starting from position a can be easily defined. If f is composed of elements $f(1)$, $f(2)$, ..., $f(n)$, the likelihood of f is:

$$l(f,a) = \sum_{i=1}^{n} l(f(i), a+i-1) \tag{8}$$

In this way, syntactic analysis consists of finding the format f and the location a that maximize equation 8. Once a format is selected, we have semantic information of every segment in p, i.e., where the hour, minutes, seconds, etc., are. Finally, we simply take the characters with the highest likelihoods, and coherent with the format element in each position.

6 Experimental Results

We have evaluated our method using a set of 17 CCTV videos at a 704×286 resolution. The total number of images in these videos is 5337. Although the video frequency is 25 fps, the typical input frequency is about 5 fps, so more than 1000 different times and dates are available for testing. We have trained and used 9 font sets, applying a semi-automatic training process (using a temporal average of the characters and manual segmentation). On the other hand, 9 different time and date formats were defined and used in the experiments.

Table 2 summarizes the results of our method. For each video sequence, the three steps of the process –pattern matching, character location and syntactic analysis– are applied to the first frame in the selected ROIs, using all the font sets and formats. This is the detection phase, as described in Section 2. In the rest of frames, pattern matching is restricted to the found font type, in the previously selected segments, and with patterns that are admissible in each segment –according to the selected format–. This is what we call the updating phase. For clarity reasons, the results of the 17 videos are joined into 4 groups, from the most favorable (group 1) to the most difficult cases (group 4).

Table 2. Recognition results of the proposed method. The experiments have been done on an AMD Athlon XP 2000+ with 256 Mbytes of RAM.

	Number of frames	% Correct characters	Correct stamps	% Correct stamps	Detection time (ms)	Updating time (ms)
Group 1	1522	99.9%	1512	99.3%	814.1	8.7
Group 2	1318	99.3%	1220	92.6%	1031.1	30.2
Group 3	1594	95.9%	1186	74.4%	794.6	17.8
Group 4	903	90.8%	380	42.1%	912.2	8.5
Total	5337	97.0%	4298	80.5%	889.5	16.6

As shown in Table 2, we achieve a very high character recognition rate of 97%, despite the problems described in Section 4. Most of the errors are due to two reasons: confusion between similar patterns, like "5" and "6", and low matching values for patterns with many background pixels, like "1" and "-".

Stamp recognition rates show a higher variability from one group to another, with an overall rate of 80.5%. A successful result does not necessarily involve a correct recognition of all characters –that could be corrected in the syntactic analysis–. However, a single error in a decimal digit is prone to cause an incorrect output. This explains the low ratio of 42.1% in group 4, which presents severe noise conditions. Fig. 5 shows some correct and wrong results.

7 Conclusions

Despite the increasing need for general text recognition in video [2], OCR in video is still a challenging problem. In this paper we have addressed the case

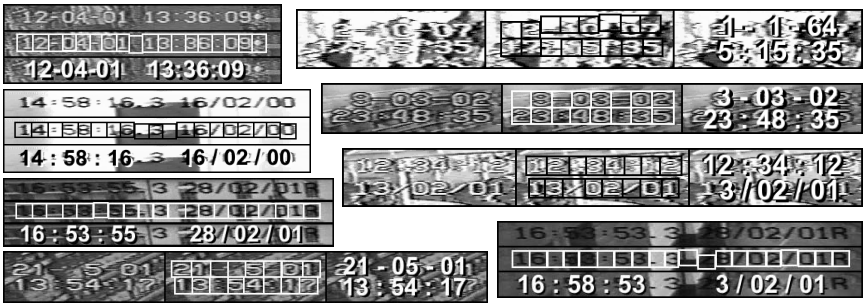


Fig. 5. Some sample results. Left: correct recognition of time and date. Right: incorrect results. For each stamp, character segmentation and system outputs are shown.

of time and date stamp recognition in CCTV video. While, in our case, some specific restrictions simplify the complexity of the problem, we have to deal with two additional difficulties: multiplexed video format and random file access.

We have developed a method which does not require character segmentation prior to recognition. First, pattern matching is applied in the ROI of the images, and then a greedy algorithm locates the characters using matching values. Finally, we perform a syntactic analysis step that selects the most likely presence of a valid time and date in the located segments.

Time and date OCR in CCTV video provides useful information that cannot be obtained by other means. The integration of our technique in a bigger application will offer the ability of performing time-based queries. The experimental results exhibit high recognition rates, showing the feasibility and computational efficiency of our approach.

References

1. Lienhart, R.: Video OCR: A Survey and Practitioner's Guide. Video Mining. Kluwer Academic Publisher (2003) 155–184
2. Sato, T., Kanade, T., Hughes, E.K., Smith, M.A., Satoh, S.: Video OCR: Indexing Digital News Libraries by Recognition of Superimposed Captions. ACM Multimedia Systems, Vol. 7, No. 5 (1999) 385–395
3. Jain, A.K., Yu, B.: Automatic Text Localication in Images and Video Frames. Pattern Recognition, 31(12) (1998) 2055–2076
4. Li, H., Doermann, D., Kia, O.: Automatic Text Detection and Tracking in Digital Video. IEEE Trans. on Image Processing, 9(1) (2000) 147–156
5. Lienhart, R., Pfeiffer, S., Effelsberg, W.: Video Abstracting. Communications of the ACM, Vol. 40, No. 12 (1997) 55–62
6. Yang, J., Chen, X., Zhang, J., Zhang, Y., Waibel, A.: Automatic Detection and Translation of Text from Natural Escenes. Proc. of ICASSP'02, vol. 2 (2002)