An Automatic Timestamp Replanting Algorithm for Panorama Video Surveillance*

Xinguo Yu, Wu Song, Jun Cheng, Bo Qiu, and Bin He

National Engineering Research Center for E-Learning, Central China Normal University, Wuhan, China 430079
xgyu@mail.ccnu.edu.cn

Abstract. Timestamp replanting is required when we want to remove timestamps in individual videos and to plant a timestamp into their merged panorama video. This paper presents a preliminary automatic timestamp replanting algorithm for producing panorama surveillance video. Timestamp replanting is a challenge problem because localization, removal, and recognition of timestamp are three difficulty tasks. This paper develops methods to attack the difficulties to finish the tasks. First, it presents a novel localization procedure which first localizes second-digit by using a pixel secondly-periodicity method. And then it localizes timestamp via extracting all digits of timestamp. Second, it adopts a homography-based method to conduct timestamp removal. Third, it presents a digit-sequence recognition method to recognize second-digit and online template matching to recognize the other digits. Experimental results show that the algorithm can accurately localize timestamp in a very low computing cost and that the performances of replanting are visually acceptable.

Keywords: Video Surveillance, Timestamp Localization, Timestamp Replanting, Secondly-Periodicity, Second-Digit Localization.

1 Introduction

A timestamp is a sequence of characters or/and encoded information indicating when a certain event occurred, usually giving date and time of day, sometimes accurate to a small fraction of a second [11]. The information of video and image timestamp can be stored in the text channel and video/image players can choose whether the timestamp is overlaid on each frame/image according users' option. Alternatively, a timestamp is superimposed into each image [2,9,11]. In analog videos timestamps have to be superimposed into videos; in digital videos timestamps may purposely be superimposed into videos so that they cannot be easily changed, of course, videos may have both encoded the information and the superimposed timestamp. In the panorama video

* Partially supported by National Natural Science Foundation of China (No.61272206), National Key Technology R&D Program of China during "12th Five-Year Plan" (No.2011BAK14B01), and Collaboration Innovation Base on Educational Digital Media and Visualization.

R. Klette, M. Rivera, and S. Satoh (Eds.): PSIVT 2013, LNCS 8333, pp. 162–171, 2013.

surveillance scenario we need to replant superimposed timestamps in some types of applications. A panorama surveillance video is merged from multiple videos taken by individual cameras. Each individual video may have a superimposed timestamp. To produce a good quality of panorama surveillance video we need to remove the timestamp of each individual video. The timestamp replanting problem for panorama surveillance is a superposed timestamp replanting problem and it is a challenge problem because localization, removal, and recognition of timestamp are three challenging tasks.

The timestamp replanting also is a very interesting and its localization, removal and recognition are three special character (or text) processing problems in video analysis. The first thought of localizing timestamp is to adopt one of the text localization methods in the literature. However, the existing text localization algorithms cannot get the satisfactory results [3-5]. Two papers specifically addressed the timestamp localization. Yin et al [9] used a rule-based text localization method. This method mainly takes an image processing approach but it proposed a spatial–temporal suppression technique to enhance the timestamp localization performance. This algorithm probably cannot properly localize second-digit because it uses temporal suppression. Covavisaruch et al [2] also mainly takes an image processing approach but it proposed a non-timestamp edge elimination technique to enhance the timestamp localization performance. These two papers reported that they can achieve 96.1% and 85.6% of accuracies respectively for home videos. This level accuracy is not enough for the industrial standard algorithm. Li et al [6-7] first proposed another new algorithm to localize the digits of digital video clock (digital video clock can be considered as a type of timestamp). They adopted the image processing methods to get character candidates and they identified second-digit by finding the character candidate that secondly changes its colors. One of its demerits is that its image processing portion is error-prone and time consuming. A procedure was presented in [10] for localizing the four digits of digital video clock, using a pixel secondly-periodicity method to localize second-digit. This procedure was designed based on the fact that second-digit pixels secondly change their grey values in a certain degree.

This paper develops a novel algorithm for replanting timestamp for panorama surveillance video, which comprises localization, removal, and recognition three main procedures. First we propose a procedure to localize the timestamp by revising the procedure that localizes the clock digits presented in [10]. Then we develop a homography-based procedure to remove timestamp. This procedure mainly adopts the techniques presented in [1]. Third, we form a procedure to recognize the digits on timestamp. This procedure first uses the digit-sequence method to recognize s-digit. And then it creates online templates of digits and it uses the online template matching method to recognize the digits on a timestamp. Based on the results of these three main steps we can plant a timestamp on the merged panorama video. Notice that this paper assumes that the procedure for forming panorama video is available already.

The rest of the paper is organized as follows. Section 2 takes an overview of the timestamp replanting algorithm proposed in this paper. Section 3 presents the procedure of timestamp localization. Section 4 presents the procedure to recognize timestamp. Section 5 presents the procedure of timestamp removal and Replanting. Section 6 gives the experimental results. We draw the conclusion of the paper in Section7.

2 Overview of Timestamp Replanting Algorithm

This section first gives the pseudocode of the timestamp replanting algorithm proposed in this paper and then briefly explains the main steps of the algorithm. Normally a timestamp contains a digital video clock and a date as the sample timestamps in Fig 4-6 show. The digital video clock has the four clock-digits representing second, ten second, minute, and ten minute, denoted as s-digit, ts-digit, m-digit, and tm-digit in the rest of the paper, respectively.

Algorithm I: Timestamp Replanting

Input: multiple videos with superimposed timestamps.

Output: a merged panorama video with a planted timestamp.

Step 1: timestamp localization for each individual video

1.1 s-digit localization

1.2 acquisition of the digit color of timestamp

1.3 timestamp bounding box determination

Step 2: timestamp recognition for each individual video

2.1 recognize s-digit in digit-sequence match method;

2.2 prepare online digit templates from s-digit instances;

2.3 recognize digits of timestamp;

Step 3: timestamp removal for each individual video and timestamp replanting for panorama video

3.1 identify a neighbor frame of the considering frame that has a proper camera motion.

3.2 find the homography between the considering frame and the identified neighbor frame.

3.3 remove the timestamp of the considering frame according to the obtained homography.

3.4 insert a timestamp into the merged panorama video.

Fig. 1. The pseudocode of the timestamp replanting algorithm proposed in this paper

As Fig 1 depicts our algorithm comprises three main components: localization, removal, and recognition of timestamp. Our localization is a novel one and significantly improves the existing technique for timestamp localization. Timestamp localization can be solved by the static region detection method for some kinds of videos such as sports video, home video, and news videos. But in the panorama video surveillance scenario, cameras may be static or have non-continuous camera motion. Thus most of objects in the videos are static. Hence static region detection method has difficulty in detecting timestamp. We propose a novel way to localize the timestamp, in which it first localizes s-digit. Then it learns the digit color of timestamp after it localizes s-digit region. It extracts the timestamp digits by using the learnt color and then obtains the bounding box of the timestamp. We conduct timestamp recognition in several steps. We first recognize the s-digit in digit-sequence method and we recognize other digits using online template matching. Timestamp removal component first identifies a neighbor frame of the considering frame that has a proper camera motion with respect to the considering frame. Then it finds their homography matrix. It finishes the timestamp removal by replacing the pixels inside the timestamp in the considering frame by the corresponding pixels under the obtained homography.

3 Timestamp Localization

This section aims to find the bounding boxes of the timestamp. We first localize s-digit and localize other digits by using the learnt color from s-digit instances. Finally, we form the bounding box of timestamp based on the digit segmentation results.

3.1 S-Digit Localization

Here we first present a method to get the bounding box of s-digit by using pixel second-periodicity method. This method bases on a piece of knowledge: the pixels in the s-digit region of a working video clock approximately change its value every second [10]. Let W and H be the width and the height of frame. Let F_i be the considered frame of a R frame-rate video. Then $F_{i-R}, F_{i-R+1}, \ldots, F_{i-1}$ and $F_i, F_{i+1}, \ldots, F_{i+R-1}$ are the R frames in the preceding and the succeeding second, respectively. Let $c(k,p)$ be the grey value of pixel p in frame F_k. With these notations we define the following two conditions D_1 and D_2.

$$D_1:\begin{cases} |c(k,p)-C_1|<\beta_1 \quad for\ k=i-R+1\ to\ i-1, \quad where\ C_1=\frac{1}{R}\sum_{k=i-R}^{i-1}c(k,p), \\ |c(k,p)-C_1|>\beta_2 \quad for\ k=i+1\ to\ R-1. \end{cases}$$

$$D_2:\begin{cases} |c(k,p)-C_2|<\beta_1 \quad for\ k=i+1\ to\ i+R-1, \quad where\ C_2=\frac{1}{R}\sum_{k=i}^{i+R-1}c(k,p), \\ |c(k,p)-C_2|>\beta_2 \quad for\ k=i-R+1\ to\ i-1. \end{cases}$$

β_1 is the threshold of the variance of digit colors within one second of no digit change and β_2 is the threshold for the difference between font color and font-background color. These two thresholds link to the human sight ability to percept the digits from video so they have relatively constant values. Currently we set their values based on our observation for several hundreds of videos. In our future work we will learn their values from a big volume of videos.

For pixel p in frame Fi we define a function $T(i,p)$.

$$T(i,p)=\begin{cases} 1 & if\ D_1\ or\ D_2\ \ holds, \\ 0 & otherwise. \end{cases} \tag{1}$$

Pixel p has a second-change at F_i if $T(i,p)=1$. And we define the accumulator $S(i)$ below.

$$S(i)=\sum\nolimits_{p\in B}T(i,p). \tag{2}$$

By summing $S(i)$ for 10 seconds we have $\Omega(k)$ being defined on the domain $[0,R)$.

$$\Omega(k)=\sum\nolimits_{i=0}^{9}S(k+i*R) \qquad k\in[0,R) \tag{3}$$

Thus we can know the transit frame of digit second is F_t such that $\Omega(t)=\arg\max_k \Omega(k)$ and $\Omega(t)$ is larger than a threshold.

We define the pixel second-periodicity function $\phi(i,p)$ below. Here $\phi(i,p)=1$ means that p has a second-periodicity value change referring to F_t.

$$\phi(i,p)=\begin{cases}1 & \text{if } T(i,p)=1\ \&\ |i\%R-F_t|<2,\\ 0 & \text{otherwise.}\end{cases} \tag{4}$$

A pixel second-periodicity measure function is defined as.

$$A(i,p)=\sum\nolimits_{j=0}^{9*R}\phi(i+j\times R,p) \tag{5}$$

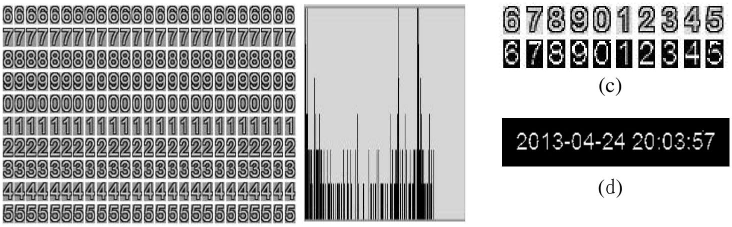
We claim that p is in the second area if $\aleph(p)=1$ and the function $\aleph(\bullet)$ is defined as follows.

$$\aleph(p)=\begin{cases}1 & \text{if } A(i,p)>\beta_3,\\ 0 & \text{otherwise.}\end{cases} \tag{6}$$

β_3 is the threshold for the second-periodicity measure function $A(i,p)$ and $A(i,p)\le\beta_3$ indicates that p is not enough second-periodicity index. β_4 is the threshold for the number of the found second-region pixels. Hence, $N=\sum\nolimits_{p=0}^{W*H}\aleph(p)$ is the number of the pixels that have a high pixel second-periodicity indexes. We claim there is no working clock when $N<\beta_4$; otherwise the bounding box of $\aleph(p)$ is the approximate place of s-digit. Then a local analysis can get the bounding box of s-digit.

3.2 Acquisition of the Digit Color of Timestamp

Since all timestamp digits have the same color so that we can learn the digit color of timestamp from the instance of s-digits. Here we describe the method for acquiring the digit color of timestamp. We collect all the instances of the digits on s-digit from a 10-second long clip as Fig 2(a) shows. Fig 2(b) is the histogram of these s-digit instances, called the instance histogram. From $\aleph(\bullet)$ in section 3.1 we can know some colors of timestamp digits. Hence we can know that a portion of the histogram belongs to the histogram of the digit color. Then we can identify the digit color histogram by searching the separate point between the background histogram and digit color histogram.



(a) (b)

Fig. 2. The illustration for digit color acquisition and digit extraction as well as digit color conversion. The cropped 10-second s-digit instances are in (a); the histogram of the instances in (a) is given in (b); 10 s-digits and their converted version are given in (c); a converted timestamp is given in (d).

3.3 Timestamp Bounding Box Determination

After we acquired the Gaussian of the digit color of timestamp we can segment the digits of timestamp by color as Fig 2(d) shows. Then extended rectangle of the bounding box of these digits is considered as the bounding box of timestamp. To be safe our box is a little larger than the real bounding box of timestamp. And this is not harm to solve our problem.

4 Timestamp Recognition

4.1 S-Digit Recognition

Let Π_1 and Π_2 be two regions of image with the same dimension. Then we use M (Π_1, Π_2) to denote the matching result between them, defined as follows.

$$M(\Pi_1,\ \Pi_2) = \frac{<\Pi_1, \Pi_2>}{|\Pi_1| \times |\Pi_2|}. \tag{7}$$

where $<\bullet, \bullet>$ is the dot product of two vectors and $|\Pi|$ is the dimension of vector, i.e. the area of Π.

Since we have known s-digit location and s-digit transit frames in Section 3, now we can collect all image instances of “0” to “9” in 10 seconds at the s-digit place without knowing what digit is on each instance yet. But we know that the s-digits in the frames from t+k*R+1 to t+(k+1)*R are the same if frame t is s-digit transition frame. Thus, the s-digit in the frames t+k*R+1 to t+ (k+1)*R is number k if we assume that the s-digit in the frames from t to t+R is “0”. In other words, we know that the s-digits in the frames from t to t+10*R form a digit periodic increasing sequence according to the clock knowledge. But we do not know that the starting digit of this sequence yet. We select frame t+ (k+0.5)*R to represent the frames from t+k*R+1 to t+ (k+1)*R and denote this frame as F^k and the s-digit instance of F^k as S^k. Let D(j) be the standard template of digit “j”, j=0, 1,2,…,9 in s-digit dimension. Then U(x), the measurement of the sequence starting with x, is defined as follows.

$$U(x) = \sum_{k=0}^{9} M\,(D((x+k)\%10),\, S^{k}) \tag{8}$$

Thus U (x) is defined on {0, 1, 2, 3,…, 9}. Then we identify the maximum point of U(x), which tells us the s-digits on any frame.

4.2 Timestamp Recognition

After we recognize the seconds of digital video clock we prepare the digit online templates from the instances of s-digit. We use the usual color segmentation to extract digits of timestamp due to we have obtained the digit color in section 3.2 as Fig 2 shows. Based on this good digit binarilized map we can localize all the timestamp digits by using a routine procedure. Now we can use the online template matching method to recognize all timestamp digits.

5 Timestamp Removal and Replanting

In the proposed algorithm, we assume that the cameras never change their intrinsic parameters and that the camera only do slight tilt motion. The proposed algorithm comprises a procedure to acquire the homography matrixes between the considering frame and one of its neighbor frames. This procedure first obtains homographies between the considering frame and its neighbor frames using the method presented in [1] and then it chooses one of neighbor frames according to their homographies. Thus, we can remove the timestamp by recovering the pixels covered by timestamp with the corresponding pixels in the chosen neighbor image. We use the procedure presented in [1] to produce the panorama surveillance video. Once we obtain the panorama surveillance video we replant a superimposed digital timestamp into it according to the recognized time.

6 Experimental Results

We implemented our timestamp replanting algorithm in Visual C++ and tested the algorithm on 300 mepg2 video clips. We first experiment accuracy and computing time of s-digit localization because it is the main step of timestamp localization. Then we conduct the timestamp recognition, timestamp removal, and timestamp replanting experiments on a set of frames.

6.1 Experiments on S-Digit Localization

This paper adopts the novel s-digit localization method presented in our previous paper [10]. This method uses the pixel secondly-periodicity of s-digits. Thus this method can accurately localize s-digit in a very low cost of computing. We conduct experiments to evaluate the accuracy and computing time of s-digit localization and the results are given in Table 1. In Table 1, #yes and % indicate the numbers and the percent that our methods correctly localize the s-digit; μ and σ are the means and the variances of the computing times of localizing s-digit for a batch of videos; 1st-100, 2nd-100, and 3rd-100 means the first to the third 100 videos of the total 300 videos. From Table 1, we can conclude that our method can achieve an accuracy of 99% for localizing s-digits for mpeg2 video in 2.5 seconds.

Table 1. Accuracy and computing time in second of s-digit localization for 300 mpeg2 videos

Localization accuracy		*computing time of s-digit localization*					
		1st-100		2rd-100		3rd-100	
#yes	%	μ	σ	μ	σ	μ	σ
297	99.0%	2.445	0.0274	2.406	0.0273	2.440	0.0237

6.2 Experiments on Timestamp Replanting

Before replanting timestamp in panorama video we need to remove timestamp from each of individual videos and to recognize timestamp. So we do experiments on timestamp recognition, timestamp removal, and replanting. For timestamp recognition separately recognize s-digit, ts-digit, minute digits, and date digits for 300 mpeg2 (we first convert the surveillance videos into mpeg2). Table 2 shows that our algorithm can achieve a very high accuracy of timestamp recognition.

Table 2. Accuracy of timestamp recognition

name	*s-digit*		*ts-digit*		*minutes*		*date*	
method	digit-seq		online temp.		online temp.		online temp.	
#total	#yes	%	#yes	%	#yes	%	#yes	%
300	300	100%	299	99.7%	300	100%	299	99.7%

Fig. 3. The digit extracted images of four timestamps

Fig. 4. The four samples of various timestamps and their result images after timestamps are removed

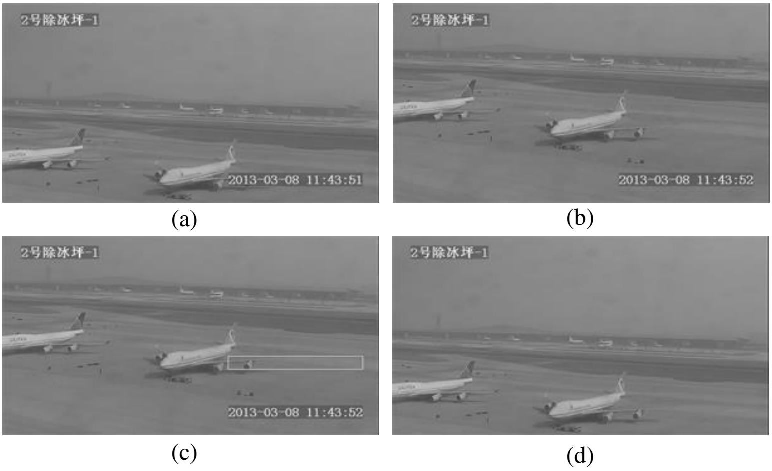


Fig. 5. A frame and its timestamp removal performance. The frame in (a) is the frame we want to remove its timestamp; the frame in (b) is one of the neighbor frames (a); The corresponding area in the frame in (b) of the timestamp in the frame in (b) is shown in (c); The frame in (a) changes into the frame in (d) after its timestamp is removed.

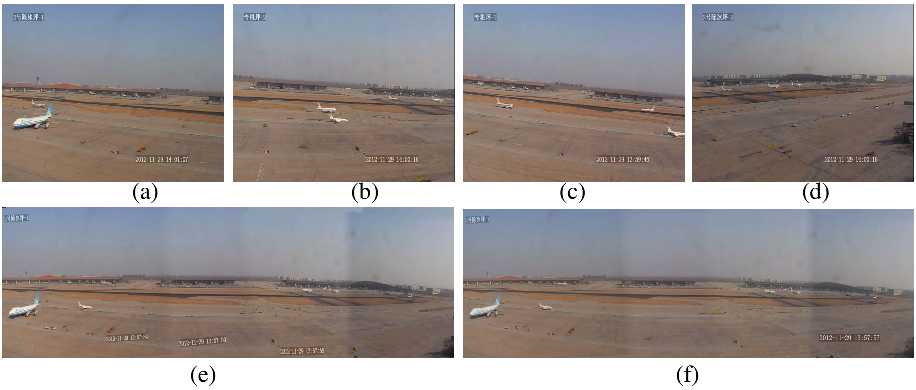


Fig. 6. Four concurrent frames from four different cameras and their panorama frame with replanting timestamp. The frames in (a) to (d) are from four different cameras. The frame in (e) is the panorama frame of four frames in (a) to (d) without timestamp replanting; the frame in (f) is the panorama frame of four frames in (a) to (d) with timestamp replanting.

Fig 3 gives the digit extraction results of four different timestamps. Fig 3 shows that noise points of digit extraction are near digits so we can get the proper timestamp bounding box even with some noise. Fig 4 gives five timestamps and their appearances after we conduct timestamp replanting step. Fig 5 shows a sample frame and its removing effect. From these two figures, we conclude that our algorithm has good performance in removing timestamp. Thus we can say that our timestamp replanting algorithm is promising to be developed into a tool for real application. Fig 6 shows the effect of timestamp replanting. When we do not do timestamp replant the three skew timestamps appear in the merged panorama frame; after we do the timestamp replanting a single timestamp is in the proper place of the panorama frame.

7 Conclusions and Future Work

We have presented an algorithm for replanting timestamp for panorama surveillance videos. This algorithm can accurately localize timestamp in a very low computing cost because it uses a very efficient timestamp localization procedure that first localizes s-digit and then segments all the digits of timestamp by using digit color. For timestamp removal, we use the corresponding pixels in one of neighbor frames of considering frame to replace the pixels covered by the timestamp. The experimental results show that the results of timestamp removal are very good.

In the near future we plan to develop an algorithm that can replant timestamp in real time robustly so that it can be used in real applications. Especially we need to improve our homography acquisition procedure in computing time. We will also need to improve our algorithm to cope with different types of timestamps.

References

1. Brown, M., Lowe, D.G.: Recognising panoramas. In: IEEE International Conference on Computer Vision (ICCV 2003), October 13-16, vol. 2, pp. 1218–1225 (2003)
2. Covavisaruch, N., Saengpanit, C.: Timestamp detection and recognition in video frames. In: Int'l Conference on Imaging Science, Algorithms and Technology (CISST 2004), Las Vegas, Nevada, USA, June 21-24, pp. 173–178 (2004)
3. Chugh, S., Jain, Y.K.: Character localization from natural images using nearest neighbours approach. Int'l Journal of Scientific & Engineering Research 2(12), 1–6 (2011)
4. Epshtein, B., Ofek, E., Wexler, Y.: Detecting text in natural scenes with stroke width transform. In: IEEE Int'l Conference on Computer Vision and Pattern Recognition (2010)
5. Jung, K., Kim, K.I., Jain, A.K.: Text information extraction in images and video: a survey. Pattern Recognition 37(5), 977–997 (2004)
6. Li, Y., Wan, K., Yan, X., Yu, X., Xu, C.: Video clock time recognition based on temporal periodic pattern. In: ICASSP 2006, vol. II, pp. 653–656 (2006)
7. Li, Y., Xu, C., Wan, K., Yan, X., Yu, X.: Reliable video clock time recognition. In: ICPR 2006, vol. 4, pp. 128–131 (2006)
8. Xu, C., Wang, J., Wan, K., Li, Y., Duan, L.: Live sports event detection based on broadcast video and web-casting text. ACM Multimedia, 226–730 (2006)
9. Yin, P., Hua, X.-S., Zhang, H.-J.: Automatic timestamp extraction algorithm for home videos. In: IEEE Int' l Symposium on Circuits and Algorithms (ISCAS 2002), vol. 2, pp. 73–76 (2002)
10. Yu, X.: Localization and extraction of the four clock-digits using the knowledge of the digital video clock. In: Int'l Conf. on Pattern Recognition (ICPR 2012), November 11-15, pp. 1217–1220 (2012)
11. http://www.2mcctv.com/blog/2012_08_09-time-stamping-on-security-cameras/