Improved Video Verification Method Using Digital Watermarking

Isao Echizen, Takaaki Yamada, Satoru Tezuka
Hitachi Ltd. SDL, 890, Kashimada, Saiwai-ku
Kawasaki, 212-8567, Japan
isao.echizen.vs@hitachi.com

Stephan Singh
Swiss Institute of Technology - Eurécom
BP 193, 06904 Sophia Antipolis cedex, France
stephan.singh@eurecom.fr

Hiroshi Yoshiura
Faculty of Electro-Communication
University of Electro-Communications
1-5-1, Chofugaoka, Chofu, 182-8585, Japan
yoshiura@hc.uec.ac.jp

Abstract

A method is described that verifies video content integrity by checking the continuity of embedded timecodes used as digital watermarks. Conventional verification methods using digital signatures and fragile watermarking are unable to distinguish between attacks and regular modifications due and thus are unable to protect against threats to content. The proposed verification method distinguishes attacks against video content from regular modifications by extracting timecodes embedded in consecutive frames of the content and then checking their continuity. A prototype implementation showed that the method is more effective than conventional ones and that it can be used by a variety of applications using video content.

1. Introduction

Digital video content has become widely available through various media such as the Internet and digital broadcasting because of its advantages over analog video content. It requires less space, is easier to process, and does not degrade over time or with repeated use. A serious problem, however, is that the integrity of digital video content is easily violated because the content can be easily modified using software editing tools. Methods for verifying the integrity of video content by detecting changes in the content are thus becoming increasingly important. Since the video format is regularly encoded and transcoded in many ways, the verification methods should be able to distinguish between illegal modifications, i.e., attacks against the content, and regular modifications. Conventional verification methods using digital signatures and fragile watermarking schemes cannot do this.

We previously investigated the technical requirements for verifying and protecting the integrity of video and proposed a system concept [1]. We have now developed a method for implementing this concept. Testing of a prototype system using the proposed algorithms showed that the method can fully verify video content integrity

2. Conventional methods

There are two types of conventional methods for verifying the integrity of video content:
(a) Methods using digital signatures are widely used for content verification [2, 3]. Digital signatures are generated by calculating a hash value from data values of the content, encrypting the value, and adding it to the content header. Verification is done by recalculating the hash value from the content, decrypting the one in the header, and comparing them. If the values match, content integrity has been maintained. If they do not, it has been broken.

(b) Methods using fragile or semi-fragile watermarks are also widely used [4, 5, 6]. Watermarks are embedded in each frame and are easily broken by a change in the content. Semi-fragile watermarks are likely to survive against JPEG and MPEG compression at high bit rates, while fragile ones are not. Verification is done by checking for broken watermarks. If any are found, content integrity is assumed to have been broken.

The first type is well suited for small-sized content, such as text and document files, that are not modified by an application. The second type is well suited for still images that are not modified or restrictively modified. Neither type is well suited for video content because video content is regularly encoded, transcoded, and converted in various ways such as MPEG encoding, resizing, filtering, and D/A-A/D conversion depending on the application. A method for verifying video content should therefore be able to distinguish between regular modifications and irregular modifications, i.e., attacks. Our proposed method has this capability.

3. Proposed method

Our proposed integrity verification method can identify attacks against video content. It can also distinguish between attacks and regular modifications such as video encoding and transcoding.

3.1. Example target applications

Our verification method has many target applications. Here we describe two of them.

Medical operations
Operations in hospitals are now commonly video recorded. If the surgeon makes a mistake, he or she might be tempted to later edit the recording to excise any damaging evidence. An auditor checking the consistency of the altered recording using our method could determine that it had been changed.
Public Works
Public works projects often use video recording to create a visual record of the progress of construction. If progress falls behind schedule, the site manager can, using a simple PC editing tool, replace some of the content with a recording of work completed elsewhere. Again, an auditor checking the consistency of the altered recording using our method could determine that it had been changed.

3.2. Attack Types

We consider three types of attacks against video recordings: deletion, addition, and replacement. A *deletion attack* removes some of the content, as described in the medical operations example. An *addition attack* adds content between frames. A *replacement attack* is a combination of deletion and addition, resulting in the same number of added and deleted frames at the same position. This is the type of attack described in the public works example.

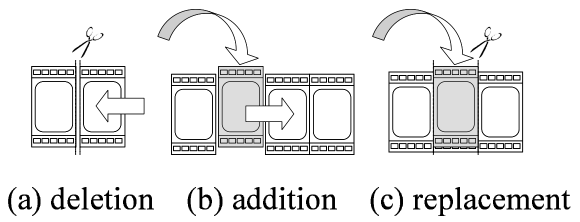


Figure 1. Attack types

3.3. Process flow

Our method uses an *encoder*, which embeds watermarks, and a *detector*, which extracts the watermarks and checks content integrity.

The encoder is implemented in a video camera system and embeds watermarks and encodes the frames at the same time the data is recorded. The watermarks are timecodes equal to the actual time (*hh:mm:ss*). The same watermark is embedded in N consecutive frames, as shown in Figure 2. The watermark for each N-frame segment is the timecode corresponding to the encoding time (beginning at $t = t_1$) of the segment's first frame.

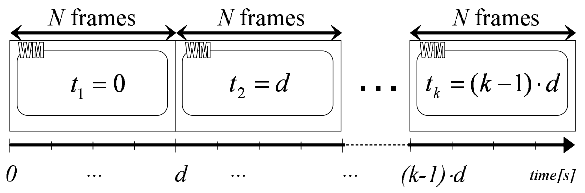


Figure 2. Watermark embedding

The detector's aim to locate and identify any attack on the video content. It works as follow: Using a detection window ($n = \frac{N}{2}$ frames sized), it first take the timecode extracting the watermark (WM) for each n-frame segment. It then checks the consistency between the timecodes by verifying their order. It also counts the number of windows where the WM has not been detected. In this way, it can detect an attack, determine the type of attack, and determine how many frames are affected (with n-frame precision). The parameters used are listed in Table 1.

Definition of parameters	
TC_{cur}	current detected timecode
TC_{pre}	previous detected timecode
TC_{old}	timecode detected before TC_{pre}
i	ordering number of the detection windows
nD	number of the detection windows where WMs are not detected
N_{max}	Maximal number of non-detected WM in succession

Table 1. Parameters used in detection

There are four steps in the detection process.

Step D1: Set the initial values of the parameters, TC_{cur}, TC_{pre}, TC_{old}, i, nD:

- $TC_{\text{cur}} = TC_{\text{pre}} = TC_{\text{old}} = i = nD = 0$,

Step D2: For the ith detection window, accumulate n frames and extract their WMs.

1. If $nD > N_{max}$, end the process.
2. If WMs are not detected, increment nD and i ($nD = nD + 1; i = i + 1$).
3. If WMs are detected, set $TC_{\rm old}$, $TC_{\rm pre}$, and $TC_{\rm cur}$:
   - $TC_{\rm old} = TC_{\rm pre}$
   - $TC_{\rm pre} = TC_{\rm cur}$
   - $TC_{\rm cur} =$ "detected timecode"
   - $nD = 0$

Step D3: Check the consistency between the timecodes $TC_{\rm old}$, $TC_{\rm pre}$ and $TC_{\rm cur}$, and check the value of nD. If the values satisfy the specified conditions, the content has been attacked.

Step D4: Increment i ($i = i + 1$) and retry Step D2.

4. Attack identification method

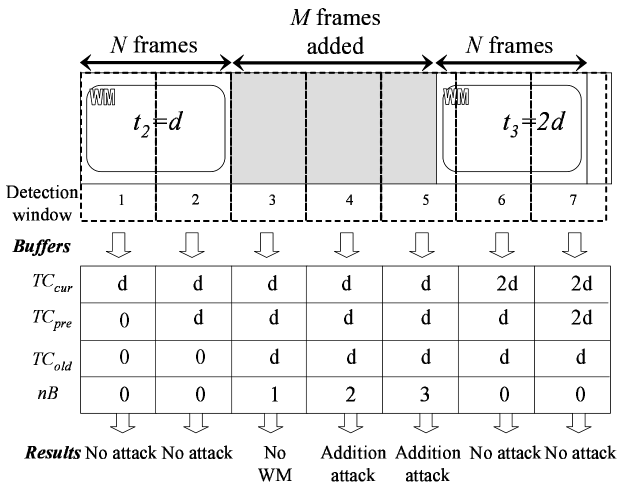
As described above, the detector uses $TC_{\rm cur}$, $TC_{\rm pre}$, and $TC_{\rm old}$ and the value of nD to identify the type of attack. The detector first determines whether three distinct timecodes following themselves appear ($TC_{old} + d = TC_{pre}$ and $TC_{pre} + d = TC_{cur}$). If they do, less than N frames have been deleted. If they do not ($TC_{old} = TC_{pre}$ or $TC_{cur} = TC_{pre}$), the type of attack is identified using the values of $TC_{\rm pre}$, $TC_{\rm cur}$, and nD ($\alpha, \beta > 1$), as shown in Table 2. A replacement attack is when addition and deletion attacks occur in succession. Such an attack has occurred if $\beta = \alpha$ or $\beta = \alpha + 1$, and it is detected after the timecode for the next window is extracted.

	$TC_{cur} - TC_{pre}$	nD
No attack	d or 0	0 or 1
Addition	d or 0	α
Deletion	$\alpha \cdot d$	0 or 1
Combination	$\beta \cdot d$	α

Table 2. Correspondence between parameter values and type of attack

Figure 3 illustrates how an addition attack is identified. By detecting consecutive windows without a WM ($nD = 3$ in the example shown) and detecting no gaps between the preceding and the current timecodes (t_2 and t_3), the detector identifies an addition attack.[1]

[1]Note that the 3rd window is not identified as an attacked one but simply as one without a WM.



Detection window	1	2	3	4	5	6	7
TC_{cur}	d	d	d	d	d	2d	2d
TC_{pre}	0	d	d	d	d	d	2d
TC_{old}	0	0	d	d	d	d	d
nB	0	0	1	2	3	0	0
Results	No attack	No attack	No WM	Addition attack	Addition attack	No attack	No attack

Figure 3. Addition attack

5. Prototype system

5.1. Description

We developed a prototype system of our proposed method for verifying video content integrity.

First, to measure the WM encoding strength, we watermarked a video sample by using the method described in paper [7] and encoded it in H.264 at a low bit rate. The detection rate of the highly compressed video content was 100% with 30 frames accumulation.

On the basis of this accumulation value, we set the window size, n, to 30 frames and the number of frames per segment, N, to 60 ($2 \cdot 30$). The timecodes used are the actual ones for the sample video file.

The encoder embed, as a WM, the timecode for the first frame in the first 60 frames; that is, they all have the same WM. The timecode for the 61st frame are embedded as a WM in frames 61-120 and so on until all the frames had been watermarked.

The detector's display is shown in Figure 4. 18 seconds of the video content is represented on the first line. Each character represents 15 frames (two characters per window). There were no attacks in the first six seconds (as shown by the $\star$ at the bottom). A deletion attack (`|`) then occurred, followed by seven clean seconds. A five-second addition attack (`A`) then occurred.

5.2. Evaluation

First we compared the performance of our system with those of conventional ones using digital signatures, fragile watermarking, and semi-fragile watermarking (see Section 2). Table 3 shows the performances for three different

```
*: no attack
|: deletion attack
A: addition attack
R: replacement attack

      !!! VERIFICATION SYSTEM !!!
0---------o---------o---------o------18
|---------|---------|---------|------|
************|*************AAAAAAAAAA**
```

Figure 4. Portion of System Display Showing Deletion and Addition Attacks

video-content conditions: without modification, with regular modifications (e.g., MPEG encoding, resizing, filtering, D/A-A/D conversion), and attacked. It shows that conventional systems using digital signatures and fragile watermarking cannot differentiate regular modifications and attacks because they can determine only whether the video content has changed. The one using the semi-fragile watermarking can differentiate them but only for particular modifications and attacks. The proposed system can differentiate various regular modifications and the attacks described in Section 3.2. Moreover, it can identify the type of attack. The proposed system is thus more effective.

	Without modification	Regular modifications	Maliciously attacked
Digital signatures	OK	NG	NG
Fragile WM	OK	NG	NG
Semi-fragile WM	OK	OK for limited modifications	OK for limited attacks
Proposed	OK	OK	OK

Table 3. Performance of conventional and proposed systems

We then evaluated the performance of our system by using the following standard video samples [8] (450 frames of 720 × 480 pixels) having different motion properties: "Square" having little movement and "Whale" having a great deal of movement. To measure our system's detection reliability, we first watermarked a sample video file and compressed it in H.264 ($bitrate = 1Mbps$). We then applied deletion, addition, and replacement attacks to it. Next we checked the attacked file with our detection system. Each attack was detected and identified.

Then we applied two attacks on different part of the same content. Each arrangement has been tested: two deletions, additions, and replacements, deletion-addition, deletion-replacing, and finally replacing-addition. Our detector identified every attack, and also determined the position where they happened.

6. Conclusion

Conventional video content integrity verification systems using digital signatures and fragile watermarking schemes are unable to distinguish attacks from regular modifications and are thus not effective countermeasures against threats to video content. Moreover, they are unable to identify the type of attack because their output is simply Boolean (content changed or not changed). The proposed verification method distinguishes attacks and regular modifications by extracting the timecodes embedded as watermarks in consecutive frames of the content and checking their continuity. Evaluation using a prototype showed that the proposed method is more effective than conventional ones. It can detect and identify attacks on video content, even if the content has suffered multiple types of attacks. It is thus usable by various types of applications using video content as evidence.

References

[1] I. Echizen *et al.*, "Integrity Verification System For Video Content By Using Digital Watermarking", *International Conference on Service Systems and Service Management (ICSSM'06)*, October, 2006.

[2] M. Pramateftakis *et al.*, "Authentication of MPEG-4-based surveillance video", *Proceedings, International Conference on Image Processing (ICIP'04)*, vol. 1, pp. 33–37, 2004.

[3] H. Morito *et al.*, "Digital Camera for Taking Evidential Photographic Images", *Proceedings, IEEE International Conference on Consumer Electronics (ICCE'01)*, pp. 118–119, 2001.

[4] M. Wu *et al.*, "Watermarking for Image Authentication", *IEEE International Conference on Image Processing*, vol. 2, pp. 437–441, 1998.

[5] C-Y. Lin *et al.*, "Robust Image Authentication Method Surviving JPEG Lossy Compression", *Storage and Retrieval for Image and Video Databases (SPIE)*, vol. 3312, pp. 296-307, 1998.

[6] C-Y. Lin *et al.*, "Issues and Solutions for Authenticating MPEG Video", *Security and Watermarking of Multimedia Contents (SPIE)*, vol. 3657, pp. 54-65, 1999.

[7] I. Echizen *et al.*, "Perceptually Adaptive Video Watermarking Using Motion Estimation", *International Journal of Image and Graphics, World Scientific*, vol. 5(1), pp. 89-109, 2005.

[8] "Evaluation video sample (standard definition)", *The Institute of Image Information and Television Engineers*.