Video Fingerprinting System Using Wavelet and Error Correcting Code

Hyunho Kang[1], Brian Kurkoski[2], Youngran Park[3], Hyejoo Lee[4],
Sanguk Shin[3], Kazuhiko Yamaguchi[2], and Kingo Kobayashi[2]

[1] Graduate School of Information Systems, University of Electro-Communications,
1-5-1, Chofugaoka, Chofu-shi, Tokyo 182-8585, Japan
[2] Dept. of Inf. and Communications Eng., University of Electro-Communications,
1-5-1, Chofugaoka, Chofu-shi, Tokyo 182-8585, Japan
{kang, kurkoski, yama, kingo}@ice.uec.ac.jp
[3] Department of Information Security, Pukyong National University,
599-1 Daeyeon-3Dong, Nam-Gu, Busan 608-737, Republic of Korea
Podosongei@hanmail.net, shinsu@pknu.ac.kr
[4] Department of Computer Science, Kyungsung University, 324-79 Daeyeon-3Dong,
Nam-Gu, Busan 608-736, Republic of Korea
iamhj@paran.com

Abstract. In this paper, we present a video fingerprinting system to identify the source of illegal copies. Content is distributed along a specified tree, with the seller as the root of the tree, the legitimate users as the leaves, and the internal nodes as content buyer or seller. Because there is a limited number of user areas available in each tree, we propose to build sub-trees, where each sub-tree has a distinctive logo. In this paper, we will use logos which are bit mapped images of the tree number. The extracted logo shows better performance visually using ECC. The fingerprinting step is achieved by the insertion of a unique information in the video wavelet coefficients by temporal wavelet transform. Our fingerprinting system is able to detect unique fingerprinting information in video content even if it has been distorted. In addition, our method does not need original video frame for extraction step.

1 Introduction

The rapid development of the Internet and digital technologies in the past years have increased the availability of multimedia content. One of the great advantages of digital data is that it can be reproduced without loss of quality. However, it can also be modified easily. The question then arises about copyright protection.

Watermarking can be used for copyright protection or for identification of the receiver. Copyright protection watermarks embed some information in the data to identify the copyright holder or content provider, while receiver-identifying watermarking, commonly referred to as fingerprinting, embeds information to identify the receiver of that copy of the content. Thus, if an unauthorized copy of the content is recovered, extracting the fingerprint will show who the initial receiver was [1]. Namely, fingerprinting is a method of embedding a unique, inconspicuous serial

J. Song, T. Kwon, and M. Yung (Eds.): WISA 2005, LNCS 3786, pp. 150 – 164, 2006.
© Springer-Verlag Berlin Heidelberg 2006

number (fingerprint) into every copy of digital data that would be legally sold. The buyer of a legal copy is discouraged from distributing illegal copies, which can be traced back to the last legitimate owner via the fingerprint. In this sense, fingerprinting is a passive form of security, meaning that it is effective after an attack has been applied, as opposed to active forms of security, such as encryption, which is effective from the point it is applied to when decryption takes place[2].

Although a large number of studies have been made on cryptographic point of view [3-10], little is known about practical application. The purpose of this paper is to address the problem of implementation of video fingerprinting. In this paper, we use a tree as in Ref. [11] to distribute video content and wavelet transform as in Ref. [12-15] to embed data in video content and make it robust to attack.

Content is distributed along a specified tree, with the seller as the root of the tree, the legitimate users as the leaves, and the internal nodes as content buyer or seller according to circumstances. Because there are a limited number of user areas available in each tree, we propose to build sub-trees, where each sub-tree has a distinctive logo(Fig. 1). In this paper, we will use logos which are bit mapped images of the tree number. The extracted logo shows better performance visually using ECC(Error Correcting Code). We have used the technique in Ref. [16] to insert a logo in a fast and straightforward manner.

Our fingerprinting system is able to detect unique fingerprinting information of video content even if it has been distorted by an attack. In addition, our method does not need the original video content for fingerprint extraction. Experimental results are presented to demonstrate the ability of our system to trace unauthorized distribution of video content, and to show its robustness to various collusion attack operations and MPEG2 compression.

This paper is outlined as follows. In Section 2 we propose an embedding and detecting process that is based on temporal wavelet transforms. Section 3 presents analysis and simulation results. Section 4 shows experimental results for important attacks that are often considered in video fingerprinting. Finally, Section 5 gives the conclusion.

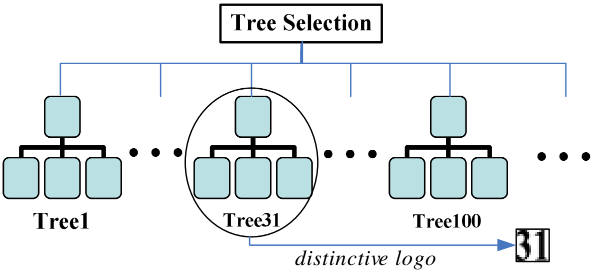

Fig. 1. Tree Selection – We can select the type of content distribution tree before sending to fingerprinting channel. If we are select the Tree 31, then 31 logo image will be embed into all video frames.

2 Proposed Method

Our system consists of four phases—embedding logo(Sect. 2.1), making content distribution tree(Sect. 2.2), embedding of fingerprinting information(Sect. 2.3), and extracting of fingerprinting information(Sect. 2.4).

2.1 Embedding Logo

The logo is a bit-mapped image of the tree number(we use 31 as an example). We have used the technique in Ref. [16] to insert a logo and briefly describe the method. With the logo embedding technique, a large number of logos, and thus a large number of end users can be supported with this proposed system.

After the binary logo image is permuted, the scrambled data sequence is then inserted into the frames in the spatial domain. Before insertion, the host video frame is first decomposed into blocks of size ***k***×***k***(we use 4×4 as an example). Let ***B*** be a selected block, the logo insertion method is described as follows:

Step 1. Sort the pixels in block ***B*** in an ascending order of pixel intensities.
Step 2. Compute the average intensity g_{mean}, maximal intensity g_{max}, and minimal intensity g_{min} of the block.

$$g_{mean}=\frac{1}{n^2}\sum_{i=0}^{n-1}\sum_{j=0}^{n-1} b_{ij}\ ,\ g_{max}=\max(b_{ij},\ 0\leq i,j<n),\ \text{and}\ g_{min}=\min(b_{ij},\ 0\leq i,j<n)$$

where b_{ij} represents the intensity of the (i,j)-th pixel in block ***B***.
Step 3. Classify every pixel in ***B*** according to:

$b_{ij} \in Z_H$ if $b_{ij} > g_{mean}$, $b_{ij} \in Z_L$ if $b_{ij} \leq g_{mean}$, where Z_H and Z_L represent high-intensity category and low-intensity category, respectively.

Step 4. Compute the mean values, m_H and m_L, of Z_H and Z_L.
Step 5. Define the contrast value of block ***B*** as $C_B=\max(C_{min}, \alpha(g_{max}-g_{min}))$, where α is a constant, and C_{min} is a constant value which determines the minimal value for pixel modification.
Step 6. Let $b_w \in \{0,1\}$ be the embedded value. Modify the pixel values in block ***B*** according to the following rules:

If $b_w=1$: $g'=g_{max}$ (if $g>m_H$), $g'=g_{mean}$ (if $m_L \leq g < g_{mean}$), $g'=g+\delta$ (otherwise)
If $b_w=0$: $g'=g_{min}$ (if $g<m_L$), $g'=g_{mean}$ (if $g_{mean} \leq g < m_H$), $g'=g-\delta$ (otherwise), where g is the original intensity, g' is the modified intensity and δ is a randomly generated value between 0 and C_B.

If the block is of larger contrast, the intensities of pixels will be changed greatly. Otherwise, the intensities are tuned slightly. The extraction of a logo is similar to the embedding process. Let block ***B*** and ***B'*** denote the original and modified blocks, respectively. The sum of pixel intensities of ***B'*** will be larger than that of ***B*** if the inserted logo pixel value b_w is 1. On the other hand, if the inserted logo pixel value b_w is 0, the sum of pixel intensities of B' will be smaller than that of ***B***.

In our method, ECC is integrated into Ref. [16] watermarking system. The convolutional error correcting code is easy to implement and fast, so we use this encoder to correct errors in the logo, which were introduced by attacks and compression. The resulting system is evaluated under our fingerprinting system channel with collusion attacks and MPEG compression.

To sum up(Fig. 2), first, tree number is encoded by the convolutional code. The encoded information is then embedded into each video frame using Ref. [16] watermarking system. Next, the resulting frames are fingerprinted according to the

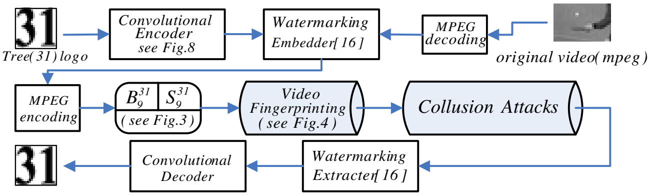

Fig. 2. The overall diagram

content distribution tree. In the experimental section, we show that the convolutional code enhances the robustness of Ref. [16] watermarking system.

2.2 Content Distribution Tree

Remark 1. Let γ ($\in$ ***Z***+) be the unique ID for seller ***S***, let ***k*** be the key expansion obtained from the seed ID γ, where ***k*** is a vector of real number from -1 to 1 of dimension $\boldsymbol{h}\times\boldsymbol{v}$ (as a video frame size).

Remark 2. Let δ ($\in$ ***Z***+) be the unique ID for buyer ***B***, let ***p*** be the pseudo-random number of the seed ID δ, where ***p*** is a vector of length $\boldsymbol{h}\times\boldsymbol{v}$ (as a video frame size).

The buyer(***B***) transmits pseudo-random number(***p***) to the seller(***S***). The seller then inserts fingerprinting information $\boldsymbol{I} = \boldsymbol{p}(\boldsymbol{k})$ into the appropriate user area of the wavelet transform.

When video content is distributed, fingerprinting information, ***I*** is inserted to each user's area of video content as described by the tree (31). Fig. 3 tells us each path has a unique fingerprint. There exists a unique path between the seller and buyer, and the unique fingerprint can be extracted to distinguish between the paths.

For example, when node-S_0 and node-B_1 engage in a transaction, fingerprinting information(I_1)—generated by the buyer and seller exchanging keys—is inserted into

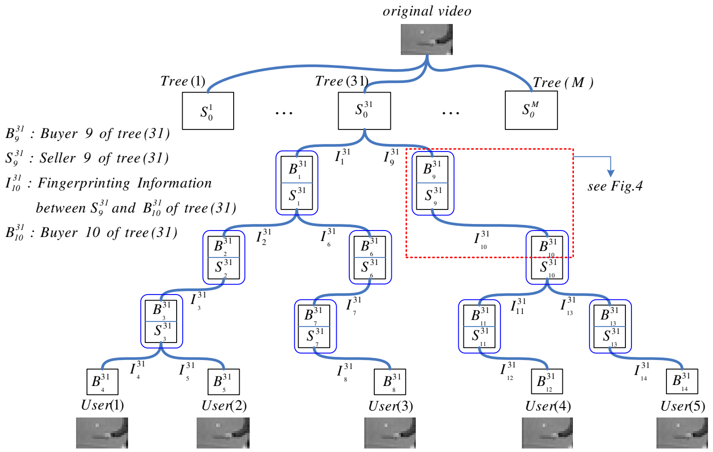

Fig. 3. Content Distribution Tree. Pay attention number of tree was omitted in the text. If we have M sub-trees(with M logos) and N users per sub-tree, then we can support M×N users.

user1, user2 and user3 area of the transmitted video. Because user1, user2 and user3 are the end users of the video in this transaction. The fingerprint is inserted into the frame by wavelet transform, and is described in Section 2.3. When node-S_1 and node-B_2 engage in a transaction, fingerprinting information(I_2) is inserted into user1 and user2 area of the transmitted video. When node-S_2 and node-B_3 engage in a transaction, fingerprinting information(I_3) is inserted into user1 and user2 area of the transmitted video. Finally, When node-S_3 and node-B_4 engage in a transaction, fingerprinting information(I_4) is inserted into user1 area of the end user1 video only.

Therefore, whenever a seller distributes content to a buyer, different fingerprinting information is inserted. Lastly, four different fingerprints are embedded into user1's video, placed in user1, user2 and user3 area. The fingerprinting information in user1's video are presented in Table 1. If it can detect the existence or nonexistence of fingerprinting information of illegal distributions, 70 correlation computations are required in Tree 31(See Fig. 3).

A 2-level temporal wavelet transform was performed on 32 frames of video, resulting in 4 types of frames(LL, LH, HL, HH where L and H stand for low and high

Table 1. Fingerprinting information of User1 video

Tree level	User1 Area	User2 Area	User3 Area	User4 Area	User5 Area
1	I_1	I_1	I_1		
2	I_2	I_2			
3	I_3	I_3			
4	I_4				

Table 2. Fingerprinting information of User2 video

Tree level	User1 Area	User2 Area	User3 Area	User4 Area	User5 Area
1	I_1	I_1	I_1		
2	I_2	I_2			
3	I_3	I_3			
4		I_5			

Table 3. Fingerprinting information of User3 video

Tree level	User1 Area	User2 Area	User3 Area	User4 Area	User5 Area
1	I_1	I_1	I_1		
2			I_6		
3			I_7		
4			I_8		

Table 4. Fingerprinting information of User4 video

Tree level	User1 Area	User2 Area	User3 Area	User4 Area	User5 Area
1				I_9	I_9
2				I_{10}	I_{10}
3				I_{11}	
4				I_{12}	

Table 5. Fingerprinting information of User5 video

Tree level	User1 Area	User2 Area	User3 Area	User4 Area	User5 Area
1				I_9	I_9
2				I_{10}	I_{10}
3					I_{13}
4					I_{14}

frequency respectively)[17]. In the experiment, user's areas were 5 sequential frequency frames among 8 frames of LH(low-high) areas. The LH means kind of intermediate frequency area(see Fig. 6). The LH region was selected because it was found to have the best detectability while not interfering with image quality.

Below is a series of five tables illustrating the information that is inserted in video content having five end users.

2.3 Embedding of Fingerprinting Information

Fig. 4 shows the embedding process using temporal wavelet transform, selection of the end user's area and insertion of fingerprinting information. The fingerprint is composed of information from seller and buyer. The seller information is a random sequence (-1~1) of h×v real numbers from a pseudo-random number generation, with a seed acting as an ID.

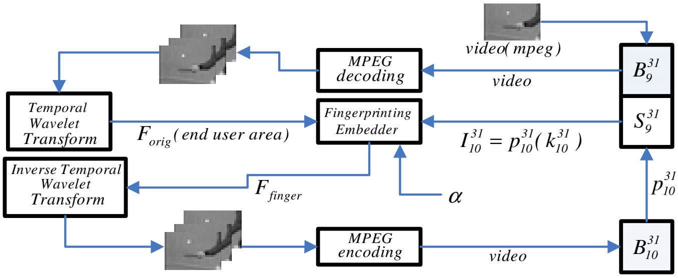

Fig. 4. Embedding Diagram (when node-S_9 and node-B_{10} engage in a transaction) (See Fig. 3)

Apply Eq. (1) to get fingerprinted video frames. In Fig. 3, when the content are distributed from the seller node-S to the buyer node-B, Eq. (1) is used once. The parameter α is the insertion strength; in this experiments, we choose α=0.5.

$$F_{finger} = F_{orig} + \alpha \cdot F_{orig} \cdot I_j \tag{1}$$

F_{finger} : *Fingerprinted video*
F_{orig} : *Original video, LH frames which are the user areas*
α : *Insertion strength*
I_j : *Fingerprinting Information* (j: buyer's path index)

2.4 Extracting of Fingerprinting Information

In the extraction step, we can extract the embedded information with Eq. (2). Note that the original video frames are not needed for extraction step.

$$I_{extract} = F_{finger} - F_{\beta}^{\ any} \tag{2}$$

$I_{extract}$: *Extracted fingerprinting information, an estimate of fingerprint*
F_{finger} : *Fingerprinted frames*
$F_{\beta}^{\ any}$: *any one frame among* F_{β}
F_{β} : *frames except* $F_{finger[LL]}$, $F_{finger[LH[User1,\ User2,\ User3,\ User4,\ User5]]}$ (see Fig. 6)

Linear correlation is calculated by Eq. (3). The linear correlation is known to be an optimal method of detecting signals in the presence of additive, white Gaussian noise[18]. In our experiments, collusion attacks and MPEG compression appear to have AWGN characteristics. Therefore, linear correlation is suitable.

$$Cor = \frac{1}{N}\sum I_{original} \cdot I_{extract} \tag{3}$$

N : video frame size ($\boldsymbol{h}\times\boldsymbol{v}$)

In Figure 6, we show each user's area and F_{β} frames that are used in extracting fingerprinting information.

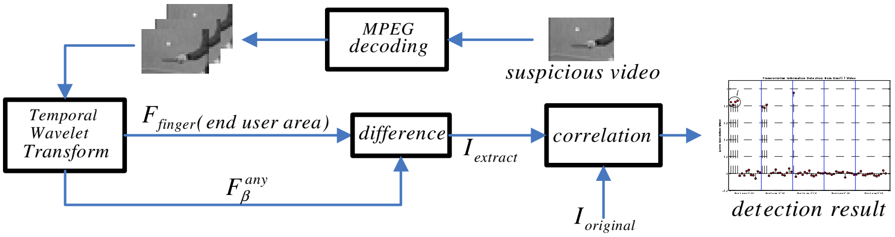

Fig. 5. Extracting Diagram

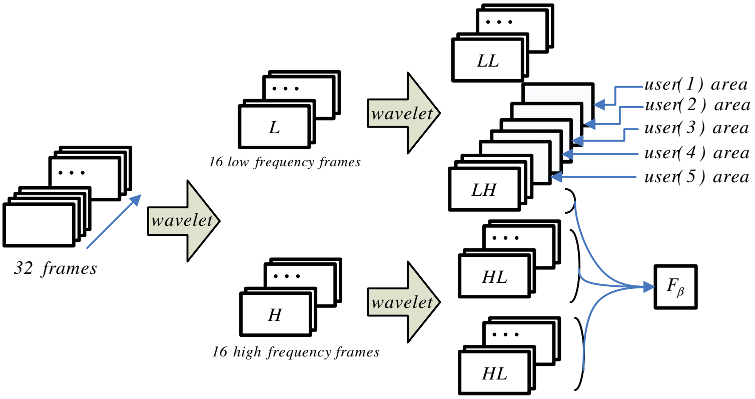

Fig. 6. User areas and F_{β} areas after temporal wavelet transform

3 Simulation Result

We have used the video sequence “table-tennis” with a frame size of 240×360 pixels and a total of 32 frames. We use convolutional codes to correct errors introduced by attacks and MPEG compression. A block diagram of the binary rate R≅ 1/2 nonsystematic feedforward convolutional encoder with memory order m=2 is shown in Figure 7(far right).

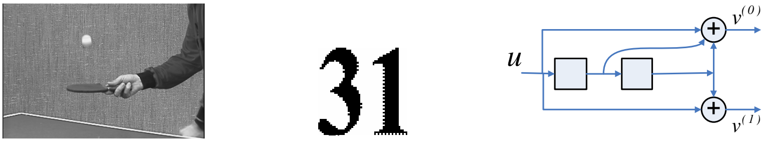

Fig. 7. (left) test video, (middle) bit mapped tree number logo(51*52), (right) convolutional encoder

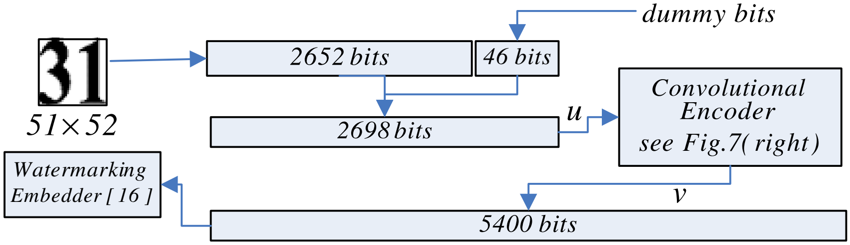
31
51×52
dummy bits
2652 bits
46 bits
2698 bits
u
Convolutional Encoder see Fig.7(right)
Watermarking Embedder [16]
v
5400 bits

Fig. 8. Detailed model of tree number embedding part of Fig. 2. We used a convolutional error correcting code which is easy to implement and fast.

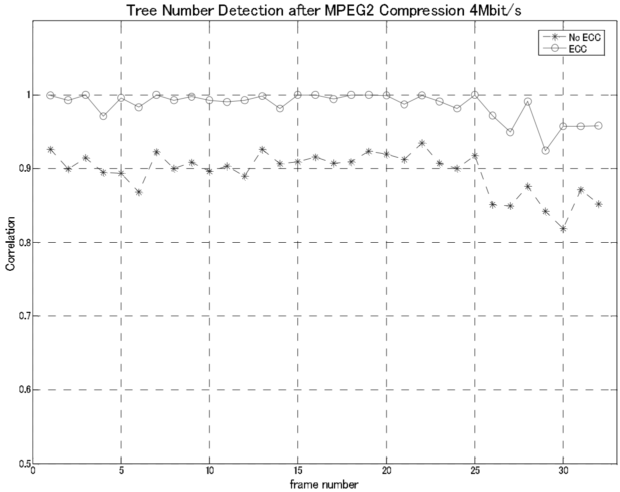

Fig. 9. Normalized correlation value of the detected logo after MPEG2 compression

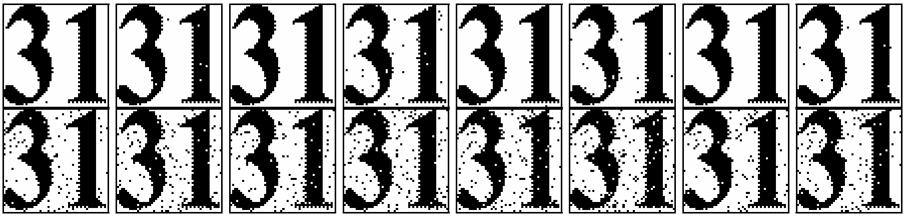

Fig. 10. Extracted tree number logo after MPEG2 compression, (above) with ECC, (below) without ECC. The 8 pairs show 8 of the 32 frames.

3.1 Tree Number

In this experiment, we show that the addition of ECC improves the correlation value of the system. As Figure 9 indicates, our system has good performance under MPEG2 compression.

3.2 Fingerprinting Information Detection

To analyze the detection result, consider the content distribution tree in Fig. 3. As Fig. 11 indicates, we see that fingerprints I_1, I_2, I_3 and I_4 were detected, corresponding to the path 1 → 2 → 3 → 4 for user1. In user2 area, fingerprints I_1, I_2 and I_3 were detected, but, this does not correspond to a path in the tree. Similarly for user3 area. Thus, we can conclude that this video was distributed to end user1.

Figs. 12~15 indicate a similar analysis for user2~5 video, respectively. This analysis showed similar results.

As Fig. 12 indicates, we see that fingerprints I_1, I_2, I_3 and I_5 were detected, corresponding to the path 1 → 2 → 3 → 5 for user2. In user1 area, fingerprints I_1, I_2 and I_3

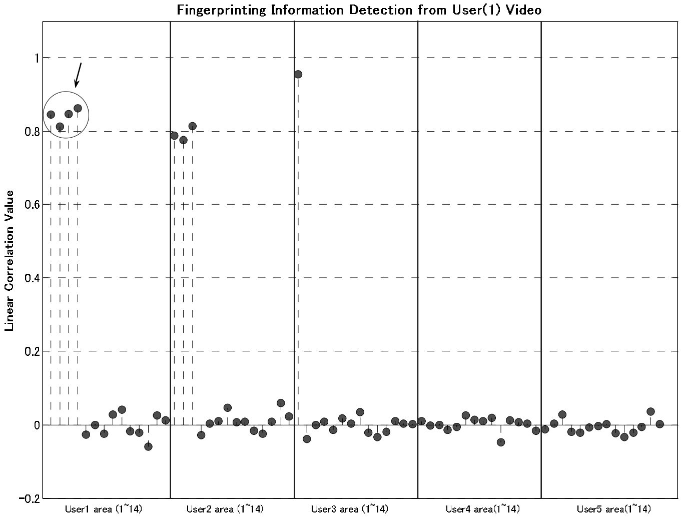

Fig. 11. Detection Result from User(1) Video

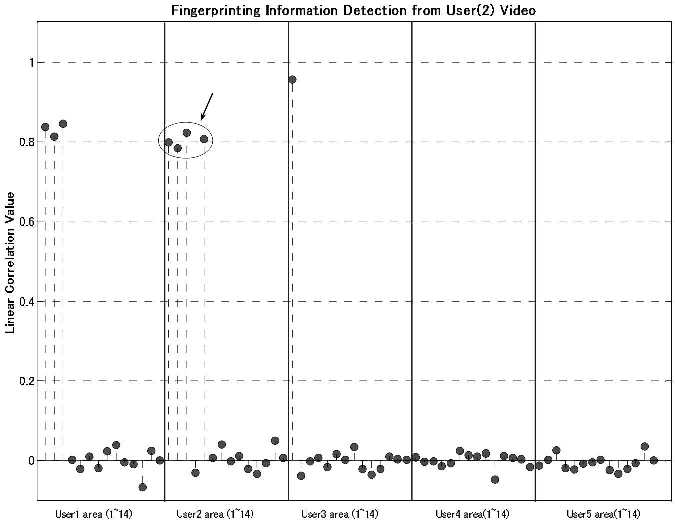

Fig. 12. Detection Result from User(2) Video

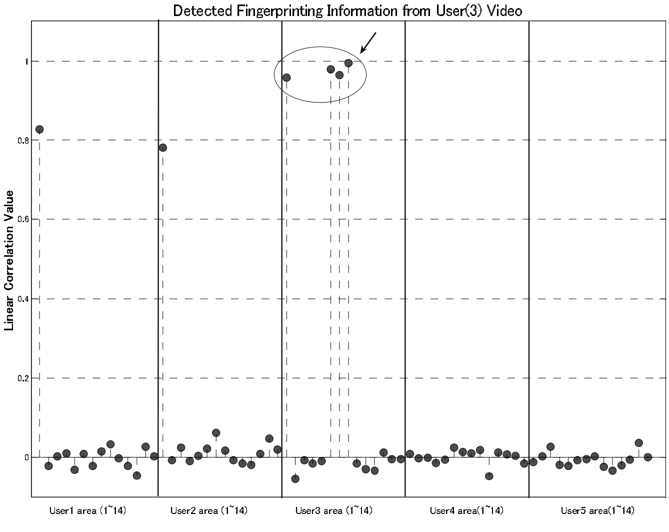

Fig. 13. Detection Result from User(3) Video

were detected, but, this does not correspond to a path in the tree. Similarly for user3 area. Thus, we can conclude that this video was distributed to end user2.

As Fig. 13 indicates, we see that fingerprints I_1, I_6, I_7 and I_8 were detected, corresponding to the path 1 → 6 → 7 → 8 for user3. In user1 area, fingerprints I_1 was detected, but, this does not correspond to a path in the tree. Similarly for user2 area. Thus, we can conclude that this video was distributed to end user3.

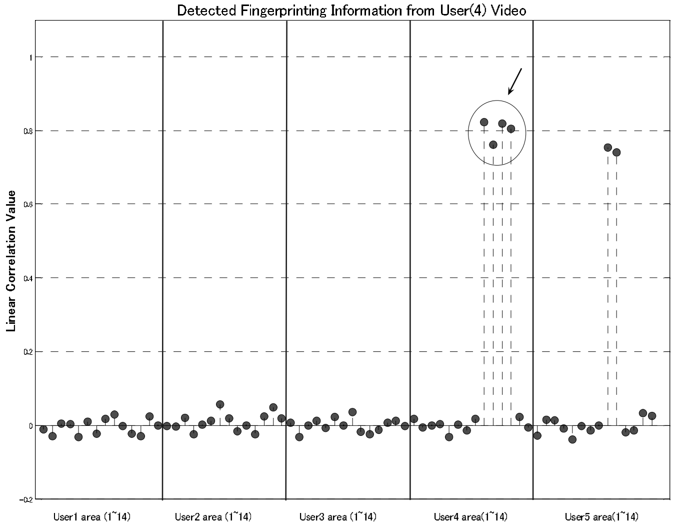

Fig. 14. Detection Result from User(4) Video

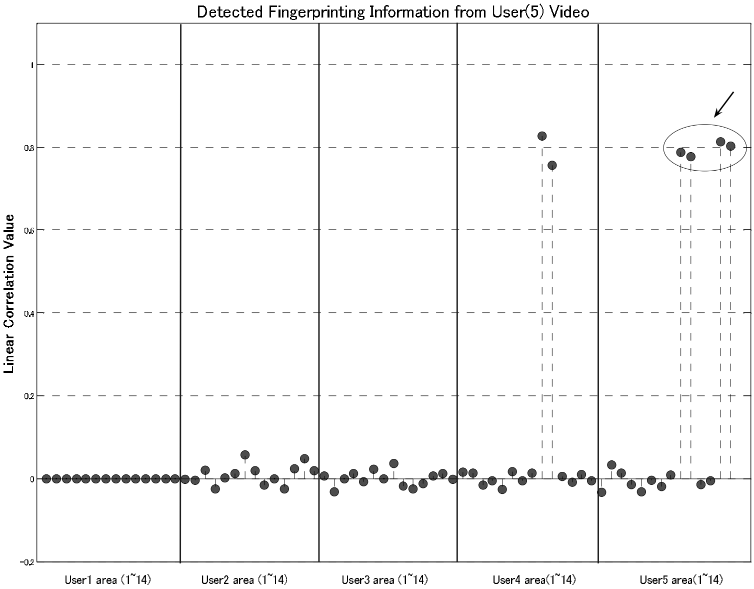

Fig. 15. Detection Result from User(5) Video

As Fig. 14 indicates, we see that fingerprints I_9, I_{10}, I_{11} and I_{12} were detected, corresponding to the path 9 → 10 → 11 → 12 for user4. In user5 area, fingerprints I_9 and I_{10} were detected, but, this does not correspond to a path in the tree. Thus, we can conclude that this video was distributed to end user4.

As Fig. 15 indicates, we see that fingerprints I_9, I_{10}, I_{13} and I_{14} were detected, corresponding to the path 9 → 10 → 13 → 14 for user5. In user4 area, fingerprints I_9 and I_{10} were detected, but, this does not correspond to a path in the tree. Thus, we can conclude that this video was distributed to end user5.

4 Attacks

A powerful attack against digital fingerprinting is the collusion attack. The results of our experiment show that the algorithm has some built-in resilience to collusion attacks, since the algorithm uses a long, uniformly distributed random number as fingerprinting information. In this attack, the following results were obtained.

4.1 Collusion Attack

(1) Averaging Collusion Attack

The averaging collusion attack was introduced by Cox, et al. [19]. The attacked video is created by averaging four fingerprinted videos such as user1, user2, user3 and user4's video. Fig. 16(left) shows the results of user1 colluding with user2, user3 and user4.

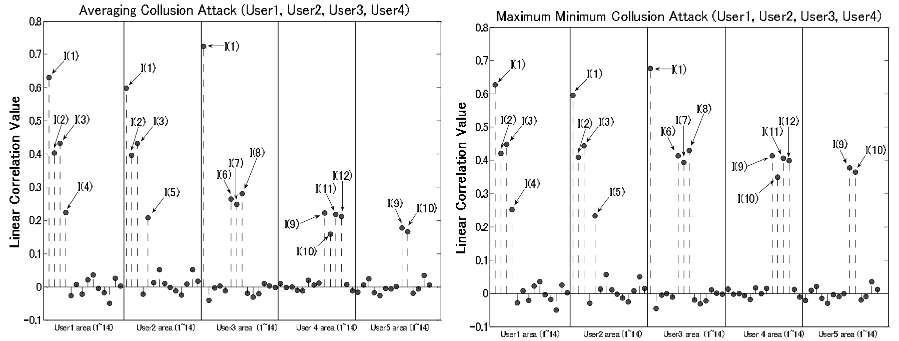

Fig. 16. Detection Result after Averaging Collusion(left), Maximum-Minimum Collusion Attack(right)

(2) Maximum-Minimum Collusion Attack

A more powerful collusion attack is the maximum-minimum collusion attack proposed by Stone [20]. The attacked video is created by taking the average of the maximum and minimum values across the components of the fingerprinted video. Fig. 16(right) shows the results of user1 colluding with user2, user3 and user4.

(3) Negative-Correlation Collusion Attack

This attack is drives the correlation coefficient to a negative value[20]. However, we can know that user1 colluded with user2, user3 and user4 in Fig. 17(left).

(4) Zero-Correlation Collusion Attack

This attack[21] is a modification method from Stone's collusion attack. This attack select a fingerprinted video from a number of available fingerprinted videos(user3

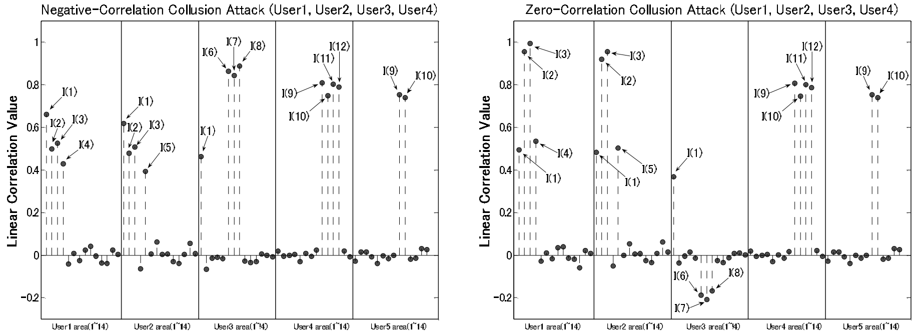

Fig. 17. Detection Result after Negative-Correlation Collusion(left), Zero-Correlation Collusion Attack(right)

(selected as an example). In user3 area, the correlation value has decreased perceptibly. However, we consider the case of user1 colluding with user2, user3 and user4 in Fig. 17(right).

4.2 Robustness to MPEG2 Compression

Robustness against MPEG2 compression, is an essential requirement of digital broadcasting content. This experiment result shows that there is possibility for practical use in broadcasting.

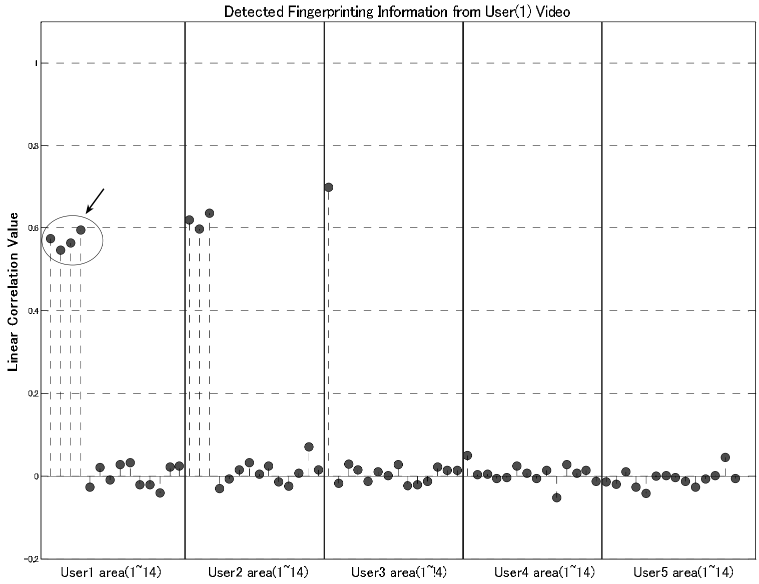

Fig. 18. Detection Result after MPEG2 Compression 4Mbit/s (User1)

In user1's video(Fig. 18), we can clearly see the points in distribution path 1 → 2 → 3 → 4 in user1 area, path 1 → 2 → 3 in user2 area, and path 1 in user3 area in 4Mbits/s video quality. That is, the end buyer is identified as user1 because this agrees with the content distribution path of user1 in Figure 3.

5 Conclusion

We have presented here an approach for video fingerprinting implementation using a watermarking technique. The embedding method in video frames is robust to various attacks because of the use of the temporal wavelet transform. We showed robustness of tree number insertion using ECC, which permits support of a large number of users in the proposed fingerprinting scheme. Future research will include improvement applying with cryptographic algorithm technique.

References

1. Judge, P., Ammar, M.: Security Issues and Solutions in Multicast Content Distribution: A Survey. IEEE Network, Vol. 17. (2003) 30–36
2. Furht, B., Kirovski, B.: Multimedia Security Handbook. CRC Press (2005)
3. Pfitzmann, B., Schunter, M.: Asymmetric Fingerprinting. EUROCRYPTO'96, Lecture Notes in Computer Science, Vol. 1070, Springer-Verlag, (1996) 84-95
4. Pfitzmann, B., Waidner, M.: Anonymous Fingerprinting EUROCRYPTO'97, Lecture Notes in Computer Science, Vol. 1233, Springer-Verlag, (1997) 88-102
5. Domingo-Ferrer, J.: Anonymous Fingerprinting Based on Committed Oblivious Transfer. PKC'99, Lecture Notes in Computer Science, Vol. 1560, Springer-Verlag, (1999) 43-52
6. Wang, Y., Lu, S., Liu, Z.: A Simple Anonymous Fingerprinting Scheme Based on Blind Signature. Information and Communications Security, Lecture Notes in Computer Science, Vol. 2836, Springer-Verlag, (2003) 260-268
7. Kuribayashi, M., Tanaka, H.: A Watermarking Scheme Applicable for Fingerprinting Protocol. IWDW'03, Lecture Notes in Computer Science, Vol. 2939, Springer-Verlag, (2004) 532-543
8. Ahmet, M.E.: Multimedia Security in Group Communication: Recent Progress in Key Management, Authentication, and Watermarking. Multimedia Systems, Vol. 9, No. 3, Springer-Verlag, (2003) 239-248
9. Emmanuel, S., Kankanhalli, M.S.: A Digital Rights Management Scheme for Broadcast Video. Multimedia Systems, Vol. 8, No. 6, Springer-Verlag, (2003)
10. Kundur, D., Karthik, K.: Video Fingerprinting and Encryption Principles for Digital Rights Management. Proceedings of the IEEE, Vol. 92, No. 6, (2004) 918-932
11. Wang, Y., Doherty, J., Dyck, R.V.: A Watermarking Algorithm for Fingerprinting Intelligence Images. Conference on Information Sciences and Systems, (2001)
12. Swanson, M.D., Zhu, B., Tewfik, A.H.: Multiresolution Scene-based Video Watermarking using Perceptual Models. IEEE Journal on Selected Areas in Comm., Vol. 16, No. 4, (1998) 540-550
13. Sagetong, P., Zhou, W.: Dynamic Wavelet Feature-based Watermarking for Copyright Tracking in digital Movie Distribution Systems. IEEE International Conference of Imaging Processing. Sep. (2002)

14. Yang, J., Lee, M.H, Liu, Q., Tan, G.Z., Wan, X.: Robust 3D Wavelet Video Watermarking. IEEE International Conference on Consumer Electronics. June. (2003)
15. Li, Y., Gao, X., Ji, H.: A 3D Wavelet Based Spatial-Temporal Approach for Video Watermarking. International Conference on Computational Intelligence and Multimedia Applications. Sept. (2003)
16. Lee, C.H., Lee, Y.K.: An Adaptive Digital Image Watermarking Technique For Copyright Protection. IEEE Trans. On Consumer Electronics, Vol. 45, No. 4, (1999) 1005-1015
17. Burrus, C.S., Gopinath, R.A., Guo, H.: Introduction to Wavelets and Wavelet Transforms. Prentice Hall (1997)
18. Cox, I.J., Miller, M.L., Bloom, J.A.: Digital Watermarking. Morgan Kaufmann, Academic Press (2002)
19. Cox, I.J., Kilian, J., Leighton, T., Shanmoon, T.: Secure Spread Spectrum Watermarking for Multimedia. IEEE Trans. On Image Processing, Vol. 6, No. 12, (1997) 1673-1687
20. Stone, H.: Analysis of Attacks on Image Watermarks with Randomized Coefficients. NEC Technical Report. (1996)
21. Wahadaniah, V., Guan, Y.L., Chua, H.C.: A New Collusion Attack and Its Performance Evaluation. IWDW'02, Lecture Notes in Computer Science, Vol. 2613, Springer-Verlag, (2003) 64-80