Journal of Robotics, Networking and Artificial Life, Vol. 3, No. 2 (September 2016), 107-110

An Authentication Method Using a Discrete Wavelet Transform for a Recaptured Video

Ren Fujii , Yasunari Yoshitomi, Taro Asada, and Masayoshi Tabuse
Graduate School of Life and Environmental Sciences, Kyoto Prefectural University,
1-5 Nakaragi-cho, Shimogamo, Sakyo-ku, Kyoto 606-8522, Japan
E-mail: { r_fujii, t_asada}@mei.kpu.ac.jp, {yoshitomi, tabuse}@kpu.ac.jp
http://www2.kpu.ac.jp/ningen/infsys/English_index.html

Abstract

Recently, several digital watermarking techniques have been proposed for protecting the copyright of moving image files by hiding data in the frequency domain. In the present study, we applied our method for authenticating a moving image, which uses a discrete wavelet transform for a static image and a method for selecting several frames from a moving image, to a recaptured video. In contrast to digital watermarking, no additional information is inserted into the original moving image by the proposed method.

Keywords: Authentication, Moving image, Copyright protection, Recaptured video, Wavelet transform

1. Introduction

Digital watermarking is a promising method for protecting the copyright of digital data. Several studies have developed methods in which a digital watermark (DW) can be extracted from data, even after compression, and the quality of the digital data remains high after the DW has been embedded. However, there is usually a tradeoff between these two properties. For a useful DW, any distortion it introduces must be imperceptible, and it must be robust to signal processing methods, such as compression or attempts to delete it. Both the processing rate and the complexity of DWs have adversely affected their performance.

In order to overcome these performance issues, we developed alternative authentication methods for digital audio[1] and static images.[2] These methods use a discrete wavelet transform (DWT), and in contrast to digital watermarking, our method for static images[2] does not insert additional information into the original digital data. The authentication is based on features extracted by a DWT and a characteristic code.

In the present study, we will review our method[3] for authenticating a digital moving image; it is a combination of our previously proposed method[2] for static images and a proposed method[3] for selecting several frames from a moving image. As in our method[2] for static images, no additional information is inserted into the original moving image, and it is authenticated by features extracted by a DWT and a characteristic code. We present the results of an experiment in which we evaluated the ability of our proposed authentication method[3] to recapture a moving image. We will describe this method[3] and present the results of the experiment in the following sections.

2. Observed Phenomenon Underpins the Authentication Method

It has been observed that when a DWT is applied to a natural image, in the histogram of the wavelet coefficients of the multi-resolution representation (MRR), the center of the distribution is very close to zero.[4] We exploited this phenomenon in order to

develop an authentication method for a static image.[2] For further information on the DWT, see Refs. 5 and 6.

3. Authentication Ratio

We set the authentication parameters as discussed below.[2]

In Fig. 1, $Th'(\text{minus})$ was chosen so that it divides the nonpositive wavelet coefficients (S'_m in total frequency) into two equal groups, and $Th'(\text{plus})$ was chosen so that it divides the positive wavelet coefficients (S'_p in total frequency) into two equal groups. Next, the values of the parameters $T1' - T4'$, which control the authentication precision, were chosen such that the following conditions were satisfied:

1) $T1' < Th'(\text{minus}) < T2' < 0 < T3' < Th'(\text{plus}) < T4'$.
2) The value of S'_{T1}, the number of wavelet coefficients in $(T1', Th'(\text{minus}))$, is equal to S'_{T2}, the number of wavelet coefficients in $[Th'(\text{minus}), T2')$, i.e., $S'_{T1} = S'_{T2}$.
3) The value of S'_{T3}, the number of wavelet coefficients in $(T3', Th'(\text{plus})]$, is equal to S'_{T4}, the number of wavelet coefficients in $(Th'(\text{plus}), T4')$, i.e., $S'_{T3} = S'_{T4}$.
4) $S'_{T1} / S'_m = S'_{T3} / S'_p$.

In the present study, the value of both S'_{T1} / S'_m and S'_{T3} / S'_p is set to 0.25, which is the same setting used for creating the code for the original image file[2]. When preparing the authentication codes, the wavelet coefficients V' for each MRR sequence are divided into three sets (hereinafter referred to as F, G, and H), as shown in Fig. 1; these sets are defined as follows:

- $F = \{V' \mid V' \in V'^{AC}, V' < Th'(minus)\}$
- $G = \{V' \mid V' \in V'^{AC}, Th'(minus) \le V' \le Th'(plus)\}$
- $H = \{V' \mid V' \in V'^{AC}, Th'(plus) < V'\}$,

where V'^{AC} is the set of wavelet coefficients from the target image file that is used to create the authentication code.

The wavelet coefficients V'_i are then classified according to the following rules with the flags f_i used in creating the original code $\boldsymbol{C}$:

When $f_i = 1$ and $V'_i \in G$, b'_i is set to 0.

When $f_i = 1$ and $V'_i \in (F \cup H)$, b'_i is set to 1.

When $f_i = 0$, b'_i is set to 0.5.

Note that the value 0.5 can be chosen arbitrarily, since the value of b_i that is the bit for creating the code for the original image file[2] does not influence the method's performance. Finally, this sequence of b'_i values is used to form the authentication code C'.

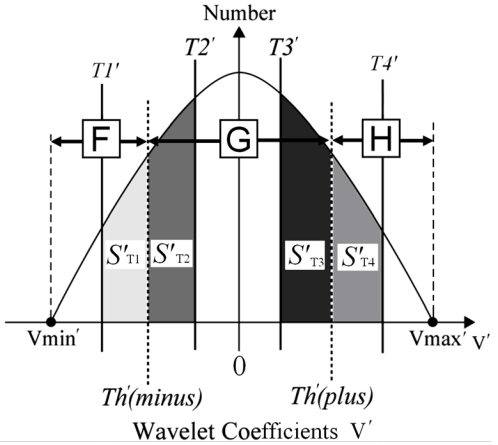

Fig. 1. Three sets (F, G, and H) of MRR wavelet coefficients used for authentication.[1]

The authentication ratio AR (%) is defined as follows:

$$AR = \frac{100 \sum_{i=1}^{N} f_i \left(1 - |b_i - b'_i|\right)}{\sum_{i=1}^{N} f_i}, \qquad (1)$$

where N is the number of wavelet coefficients chosen to create the authentication code for the original image file.[2] As can be seen in equation (1), neither b_i nor b'_i influence the value of AR when $f_i = 0$, which occurs when the corresponding V_i that is the wavelet coefficient of the original image is not selected for coding in the original image file.[2]

To use the proposed method, we need to store the flags f_i and the original code $\boldsymbol{C}$ for each copyrighted file that we want to protect. When calculating (1) in order to authenticate an image file, we do not use the original image file; instead, we use the flags f_i and the code $\boldsymbol{C}$ for that file.[2]

4. Authentication of a Moving Image

We applied our proposed authentication method[3] to a recaptured digital moving image. We explain the proposed method[3] in this section.

4.1. *Authentication code*

We obtained 10 codes for each moving image segment, each of which had.

Step 1: Let N_{total} be the total number of frames in the moving image. Calculate frame No. k_B, where k_B is the

smallest integer not less than $0.1\times N_{total}$, and calculate frame No. k_E, where k_E is the largest integer not greater than $0.9\times N_{total}$.

Step 2: Output the images of frames No. k_B and k_B+1 as BMP files. For each pixel, evaluate the difference between the grayscale level in frame No. k_B and that in frame No. k_B+1. Calculate the total sum S[1] of squares of these differences. Store the grayscale levels of frame No. k_B+1 in array A_1. Set the initial conditions as $C[1]:=k_B+1$, $count:=1$, and $k:=k_B$+2.

Step 3: If $k=k_E$+2, go to Step 8. Otherwise, go to Step 4.

Step 4: Overwrite the image of frame No. k onto that of frame No. k_B–2 in the BMP file. Store the grayscale levels of frame No. k in array B. For each pixel, evaluate the difference between the grayscale level in frame No. $k-1$ and that in frame No. k. Calculate the total sum T of squares of these differences. Update the value of k to $k:=k+1$.

Step 5: If $count<10$, update the value of $count$ to $count:=count+1$, and go to Step 6. Otherwise, go to Step 7.

Step 6: Overwrite the array A_{count} with B, i.e., $A_{count}:=B$. Set $S[count]:=T$, and $C[count]:=k-1$. Go to Step 4.

Step 7: Calculate $l=\arg\max_{i=1,2,\cdots,10} S[i]$. If $S[l]>T$, overwrite the array A_l with B, i.e., $A_l:=B$. Set $S[l]:=T$, and set $C[l]:=k-1$. If $k=k_E$+1, go to Step 8. Otherwise, go to Step 4.

Step 8: Output the arrays $A_i(i=1,2,\cdots,10)$ as BMP files. Obtain the codes and flags used for the authentication codes by using the method described in Ref. 2. Output the codes, the flags, $C[i](i=1,2,\cdots,10)$, and $S[i](i=1,2,\cdots,10)$.

4.2. *Authentication*

We used the method described in Section 4.1 to select frames from moving images. The codes and the flags for the selected frames were obtained using the methods described in Ref. 2. For a given test moving image and a given moving image in the database, the authentication ratio of the test moving image to the moving image in the database is defined as the largest authentication ratio of any selected frame in the test moving image to any selected frame in the moving image in the database. This is determined for each moving image in the database, and the one with the highest authentication ratio is selected as the assumed original image.

5. Experiment

We evaluated the performance of the proposed method by conducting a computer experiment. In this section, we present the results.

5.1. *Method*

The experiment was performed in the following computational environment: Dell OptiPlex 3020; CPU: Intel Core i5-4570 3.2 GHz; 4.0 GB memory; OS: Microsoft Windows 7 Professional. The development language was Microsoft Visual C++ 6.0.

The experiment proceeded as follows. We obtained 77 moving image segments, provided as MPEG-1 files, from MUS-CLE-VCD-2007.[7] These were converted from RGB components into YCrCb components, and for each of these segments, we obtained ten codes for Y components. We numbered the segments in the order in which they were recorded and then divided them into 11 groups of 7 members each. We then chose the six segments (Nos. 24, 45, 27, 21, 93, and 91) that recorded the middle amount of time in each group that had the shortest to the sixth shortest recording time among 11 groups. Nos. 24 and 45 were monochrome, and so we replaced them with the two segments that had the most similar recording times, Nos. 100 and 34, respectively. These six segments were displayed on a liquid crystal television screen of Panasonic VIERA TH-19C305, and the moving images were recaptured by a digital video camera (Sony Handy-cam HDR-CX7); they were then saved as MPEG-2 files, followed by cutting useless margin and being saved as MPEG-4 files. Divide all frames into 10 groups for each file by the time from the beginning, and pick up 10 frames per a group by our previously proposed method[3] except the first and last groups in the sense of time from the beginning. This resulted in 80 frames for each recaptured moving image, and these were used for authentication of the 77 segments.

We used FFmpeg[8] to output a BMP file for each segment; 24 bits were used for the grayscale level and the image consisted of 256×256 pixels. For the DWT, we used Daubechies wavelets. Based on the results of preliminary experiments, we used the LH components obtained from the DWT up to level 4.[2]

Several consecutive frames in which most pixels are black (or most are white) are sometimes inserted into a moving image, because they can be useful for scene transitions. A frame in such a series is neither representative nor unique to that moving image, and if it is selected to be coded, it could damage the ability of the code to authenticate that moving image. Therefore, we used only frames in which the average grayscale level for the Y component was in the range of 5 to 250 (the full range is 0 to 255).

5.2. *Results and discussion*

Table 1 shows some examples of the authentication ratios obtained in the experiment. In the six moving image segments (Nos. 100, 34, 27, 21, 93, and 91), the average authentication ratio was in the range 71.6% to 74.3% when the recaptured segment was from a different section of the original moving image, but it was in the range of 91.7% to 97.0% when the recaptured segment was from the same segment; see Table.1. When authenticating the recaptured segment No. 93 against the codes of the original segment No. 17, the authentication ratio was 90.1%. This unusually high authentication ratio was caused by a pair of similar structures, which can be seen in the frames shown in Fig. 2. An example of authentication by the proposed method is shown in Fig. 3.

Table 1. Authentication ratios for recaptured moving images.

	Original file No. for recaptured moving image					
	100	34	27	21	93	91
Average for others	73.9	73.1	74.3	73.5	71.6	72.4
For corresponding raw file	**95.5**	**94.7**	**96.2**	**97.0**	**94.7**	**91.7**
Maximum for others	87.1	82.6	86.4	84.8	90.1	88.6
Recording time (min:sec)	00:30	03:20	06:58	11:32	14:50	19:50

(%)

6. Conclusion

We applied our proposed authentication method[3] to a recaptured digital moving image in order to evaluate the ability of the method to recapture a moving image. In contrast to digital watermarking, in the proposed method, no additional information is inserted into the original moving image. The experimental results show that the proposed authentication method performs well.

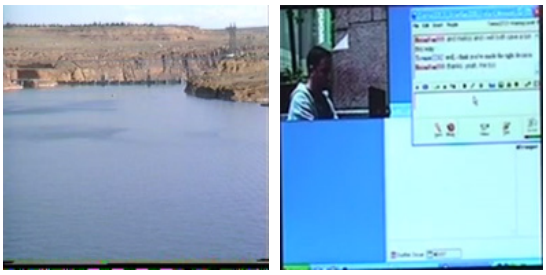

Fig. 2. Left: recaptured frame of moving image segment No. 93; right: frame of moving image segment No. 17 in the database.

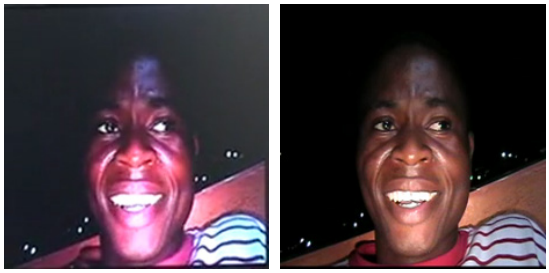

Fig. 3. Left: recaptured frame of moving image segment No. 27; right: frame in the database with an authentication ratio of 96.2% with segment No. 27.

References

1. Y.Yoshitomi, T.Asada, Y.Kinugawa, and M.Tabuse, An authentication method for digital audio using a discrete wavelet transform, *J. Inf. Sec*. **2**(2) (2011) 59-68.
2. T. Asada, Y. Yoshitomi, and M. Tabuse, A verification method for a digital image file using a discrete wavelet transform (in Japanese), *J. IIEEJ* **39**(6) (2010) 1088-1094.
3. R. Fujii, Y. Yoshitomi, T. Asada, and M. Tabuse, Authentication method using a discrete wavelet transform for a digital moving image, *J. Inf. Sec.* **7** (2016) 1-13.
4. M. Shino, Y. Choi, and K. Aizawa, Wavelet domain digital watermarking based on threshold-variable decision (in Japanese), *Technical Report of IEICE*, DSP2000-86, **100**(325) (2000) 29-34.
5. D. Inoue and Y. Yoshitomi, Watermarking using wavelet transform and genetic algorithm for realizing high tolerance to image compression, *J. IEEJ* **38**(2) (2009)136-144.
6. T. Taniguchi and Y. Yoshitomi, Method for character domain extraction from image using wavelet transform, *J. Robotics, Networking and Artif. Life* **2**(1) (2015) 103-106.
7. MUSCLE-VCD-2007, http://www.rocq.inria.fr/imedia/civrbench/index.html Accessed 9 September 2015.
8. FFmpeg, http://ffmpeg.org/ Accessed 9 September 2015.