PROTECTION OF VIDEO LOGOS WITH RANDOMIZATION

Yongdong Wu

Institute for Infocomm Research ($\mathbf{I}^2\mathbf{R}$), Singapore
wydong@i2r.a-star.edu.sg

ABSTRACT

To announce the ownership of video such as TV programs, the owner usually embeds his logo into the programs in a visible way. Since the logo and its position are usually fixed in the video frames, an adversary can completely remove the logo from the video without video quality loss. In order to thwart this removal attack, the present paper randomly changes logo location and shape without obviously reducing the visibility and fidelity of the logo.

1. INTRODUCTION

Generally, there are two information hiding methods: invisible watermarking (e.g., [1]) and visible watermarking (e.g., [2]). Both watermarking methods have the following differences: (1) invisible watermark is usually detected with designated software with exceptional visual watermark method (e.g., [3]), while visible watermark is detected with human eyes directly. (2) invisible watermark has no oblivious perceptual distortion, but visible watermark decreases viewing pleasure; (3) an invisible watermark is usually used for copyright confirmation after a video is inspected such as traitor tracing, while a visible watermark is often used for copyright announcement in the process of consuming.

A typical application of visible watermarking is to embed a logo into the TV program. The video logo, as a perceptual trademark for digital videos, is unique in terms of shape and visible colors. Presently, a logo is popularly localized at a stable position (e.g., the screen corners) within a video so that it can draw the attention of viewers [4].

A video logo can be embedded in two distinct styles: overlapped and transparent. An overlapped logo directly overlaps a portion of a video frame so as to completely obscure the underlying contents of the video; while a transparently overwritten logo blends itself with the frame. Since the transparent logo allows the content of the visual media to remain partially visible, it results in less quality loss than overlapped logo does.

Since visible watermarking decreases the image quality, a viewer is interested in removing the logo from each frames. The process of removing a video logo can be thought of as an attack on a visible watermarking. Its aim is to erase the embedded watermark from its host video [5]. To this end, the attacker will detect the logo location first, then repair the video region where the logo is occupied.

Feature matching is one of the widely used methods in logo detection step. For example, Soffer *et al.* [6] matched logos based on shape features; and Seiden *et al.* [7] extracted a set of grayscale features so as to construct a suite of rules for classifying the segmentation of the logos.

Yan *et al.* [8] presented another method to detect logo by exploring the temporal correlation of video frames. First, they localized the logo boundary box using a distance threshold of video frames and further refined it by employing a comparison of edge lengths. Second, a Bayesian classifier framework locates fragments of logos with the prior knowledge about the location of the video logos and their intrinsic local features to achieve a robust detection result. Third, they developed an algorithm based on image inpainting [9] to erase video logos [10]. To restore the missing or damaged portion of a frame, they manually selected the logo region and chose the clearest logo from all video frames, and automatically erased the logo based on region filling by extrapolation.

In order to prevent watermark removal attack, Meng and Chang [2] suggested to embed watermark in moving foreground so as to defend against temporal filtering. However, this countermeasure may be make the viewer unhappy and hence is unsuitable for logo embedding. They also proposed to adaptively change the embedding strength. However, this method may be vulnerable to known-logo attack.

To enhance the robustness of visible embedding, this paper extended the protection method in [2]. Since the function of a video logo is used to draw attention of the viewer other than claim copyright of the logo, a distorted logo (e.g., logo animation) is acceptable if it still represents the original owner. Hence, in comparison with invisible watermarking, visible watermarking can tolerate much more distortion. Based on this observation, the present paper randomizes the logo in terms of location, shape, scalar, and a lot of distortion methods without reducing the logo function.

The remainder of this paper is organized as follows. Section 2 introduces the present protection method and its security analysis. In Section 3, the experiments are demon-

strated for the protection in terms of the security and quality. Section 4 summarizes the paper.

2. PROPOSED SCHEME

2.1. Attack model

Since the logo is visible, a patient adversary can check the video one by one and manually remove all the logo by brute force. That is to say, it is impossible to prevent logo removal theoretically. Thus, a practical protection method is to increase the cost of logo removal, or make the logo removal tool to be inefficient. Thus, this paper assumes: (1) Based on [11], a logo is located in the boundary of any frame to reduce the annoyance, but its location is not known; (2) the attacker may extract the logo from several frames or obtain the logo from some resource such as Internet. In addition, although there are other detecting rules, we simply assume that the adversary calculates correlation value

$$\lambda = \frac{\mathbf{L} \bullet \mathbf{I}}{\| \mathbf{L} \| \cdot \| \mathbf{I} \|} > \varepsilon \tag{1}$$

where $\mathbf{L}$ is the selected logo, and $\mathbf{I}$ is the inspected frame region. If the correlation value λ is greater than a predefined value ε, the adversary will regard that region $\mathbf{I}$ is watermarked, and remove the logo to recover region $\mathbf{I}$.

2.2. Protection methods

To defense against logo removal attack, we must randomize the logo position and the logo given that the distorted logo is semantically the same as the original one for a human viewer. In this subsection, we enumerate the methods to randomly distort a logo. As an illustrative example, Fig.1 shows the watermarked images embedded with the randomized logos.

2.2.1. Pixel modification

It is known that reducing the color pixels randomly will reduce the logo visibly. But if the average value of the logo is fixed, then the content of the logo may be intelligible yet [8]. In order to detect a logo, the previous removal method depends on the relationship between two continuous frames. Thus, if we embed the logos in different frames, the correlation value may be decreased. For example, the logo in Fig.1(b) is negative to the original logo in Fig.1(a), although the viewer still recognizes the ownership, the correlation value λ will be small (see Table.1).

2.2.2. Linear transformation

Assume that a logo $\mathbf{L}$ includes n independent sub-logos as $\mathbf{L}_0, \mathbf{L}_1, \ldots, \mathbf{L}_n$, for each frame to be visibly watermarked, then, a new logo is

$$\tilde{\mathbf{L}}_i = R^t \mathbf{L}_i + T \tag{2}$$

where R is a rotation matrix, and T is a translation value as used in [2]. The linear transformation not only alleviates the temporal attack used in [8] so as to increase resistance capability, but also increases the attention of viewers due to logo animation. Fig.1.c-d illustrate the watermarked frames embedded with the linearly transformed logos. Please note, the logo can be rotated in 3-dimensional space (X-Y plane, in horizontal axis, and vertical axis).

2.2.3. De-synchronization

It is well-known that de-synchronization reduces the correlation value in information hiding community. To desynchronize the frame detecting, we can zoom in/out the logo so as to change the logo size randomly. Fig.1.e-f illustrate the watermarked frames embedded with the scaled logos. After a logo size is changed randomly, the detecting method is not valid. Other de-synchronization methods may be deleting/inserting lines/rows etc.

2.2.4. Filtering

There are many tools which can be used for filtering. For example, Adobe $\mathrm{Photoshop}^{TM}$ provides tools such as *Liquify*, *Artistic*, *Blur*, *Distort*, *Sharpen*, *Skeleton*, etc. Each tool will change the shape of the logo and hence reduce the correlation value λ. Fig.1(g)-(h) illustrate the watermarked frame by spherizing and swirling the logo with $\mathrm{Photoshop}^{TM}$.

2.2.5. Watermarking strength

Given a frame region $\mathbf{I}$ and a logo $\mathbf{L}$, the watermarking method [11] is represented as Eq.(3).

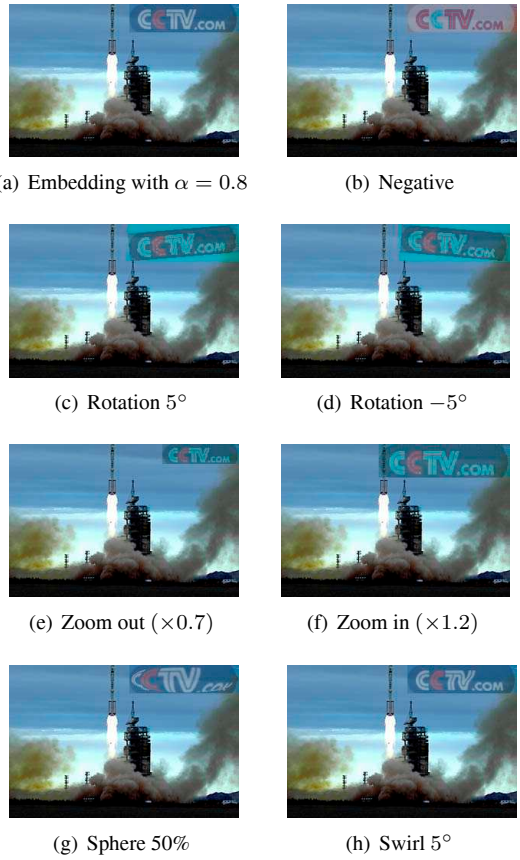
$$\tilde{\mathbf{I}} = (1 - \alpha)\mathbf{I} + \alpha\mathbf{L} \tag{3}$$

where α is watermarking strength, $\alpha = 1$ indicates overlapped watermarking, $\alpha = 0$ means that the frame is skipped for embedding, otherwise, transparent watermarking. Roughly speaking, when α is decreased, λ is decreased, the fidelity of the transparent watermarking is decreased in the images, but the quality of the frame is increased.

2.3. Security analysis

According to the attack model in Subsection 2.1, the security of logo-protection approaches is measured with the cost of the watermark removal process, especially the effort to accurately locate the logo.

Since it is hard to estimate the security for different protection methods, we simply assume that the protected



(a) Embedding with $\alpha = 0.8$ (b) Negative

(c) Rotation $5°$ (d) Rotation $-5°$

(e) Zoom out $(\times 0.7)$ (f) Zoom in $(\times 1.2)$

(g) Sphere 50% (h) Swirl $5°$

Fig. 1. Transparent watermarking with example distorted logos given that $\alpha = 0.8$. The distorted logos are generated with Adobe Photoshop^{TM}. In order to make it easy to recognize the logo, each logo is enhanced adaptively with similar technique in [2].

method is described with tuple $(\alpha, \delta_x, \delta_y, \theta_x, \theta_y, \theta_z, s)$, where (δ_x, δ_y) indicate the translation vector, $(\theta_x, \theta_y, \theta_z)$ represent the rotation angles, s is the zoom factor. Define $\mid X \mid$ represents the number of possible values. As an illustrative selection method, let $\mid \alpha \mid = 32$, $\mid \delta_x \mid = \mid \delta_y \mid = 64$, $\mid \theta_x \mid = \mid \theta_y \mid = 256$, and $\mid \theta_z \mid = 16$, and $\mid s \mid = 16$, the possible distorted logo is

$$\mid \alpha \mid \times \mid \delta_x \mid^2 \times \mid \theta_x \mid^2 \times \mid \theta_z \mid \times \mid s \mid = 2^{41}.$$

Thus, the adversary has to check 2^{40} distorted logos on average so as to completely remove the logo from a frame. In practice, the above parameters should be changed based on the capacity of the target adversary.

Additionally, the owner can randomly employ filtering or distortion methods so as to harden the adversary detection. For example, Adobe Photoshop^{TM} introduces 14 filtering methods and each method has sub-methods, each method includes several parameters. As a result, the number of watermarked frames is significant, and hence it is not easy for an adversary to remove all the video logos.

3. EXPERIMENTS

3.1. Quality of transparent watermarking

Since a logo in video merely announces the copyright or ownership of video content, the quality of the logo is not very important, and the fidelity of the logo is decided by the viewers. Hence, the subjective decision may be more sound than the image fidelity measure such as PSNR (Peak-Signal-Noise-Rate). To show this observation, we test the quality of the logo in transparent mode as Eq.(3) given that the distorted logo is of acceptable shape. We calculate the PSNR of the watermarked region against the original logo. As shown in Fig.2, PSNR is very low in case of $\alpha = 0.5$, but the logo is still well recognized as a valid logo. That is to say, the objective measure PSNR may be not suitable for evaluating the acceptance of logo embedding.

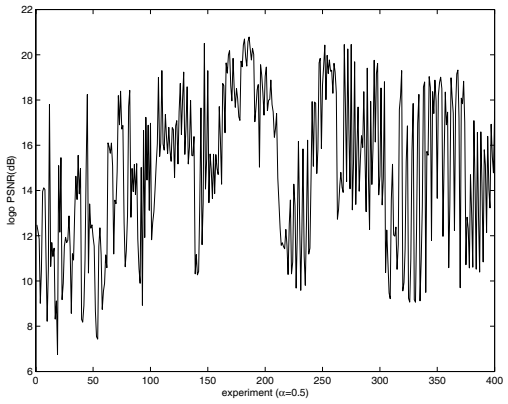


Fig. 2. PSNR in transparent watermarking $\alpha = 0.5$, mean $\mu = 14.9$ and standard variance $\sigma = 11$.

Meanwhile, Fig.3 shows that PSNR is increased with α, this observation is in consistent with the embedding Eq.(3).

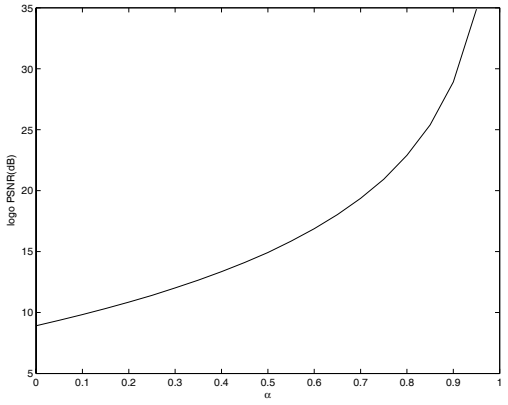


Fig. 3. logo PSNR *vs* embedding strength α. PSNR is calculated with watermarked region against the original logo.

3.2. Detection accuracy

In the attack model of Subsection 2.1, we assume that the adversary knows the logo in advance. Hence, the adversary is more powerful than that in the paper [8]. Based on the embedding method as Eq.(1), the detection value is

$$\lambda = \frac{\mathbf{L} \bullet ((1-\alpha)\mathbf{I} + \alpha\tilde{\mathbf{L}})}{\| \mathbf{L} \| \cdot \| \tilde{\mathbf{I}} \|} \approx \frac{\alpha \mathbf{L} \bullet \tilde{\mathbf{L}}}{\| \mathbf{L} \| \cdot \| \tilde{\mathbf{I}} \|} \qquad (4)$$

where $\mathbf{L}$ is the logo to be confirmed, and $\tilde{\mathbf{I}}$ is the inspected frame region. Fig.4 describes the correlation values with the watermarking strength α. Therefore, the bigger α is, it is easier for an attacker to detect the logo location.

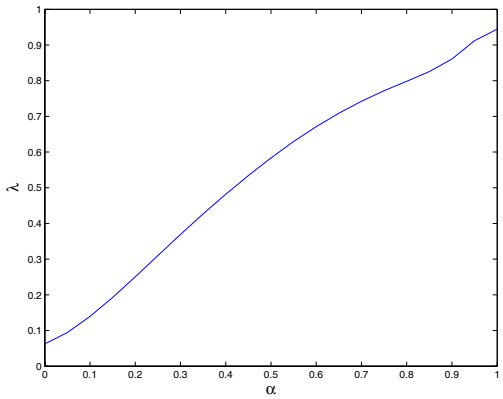


Fig. 4. Correlation value λ *vs* embedding strength α. In the experiments, the mean values of $\mathbf{L}$ and $\tilde{\mathbf{I}}$ are removed from Eq.(4) so as to reduce the noise.

Besides the embedding strength, the detection value is also variable with the protection methods. Table.1 illustrates the detection values, where α is changed in the interval [0,1] as shown in the first column. The other columns indicate the correlation values for the watermarking methods shown in Fig.1. Since values in columns 2-7 in Table.1 are small, the protection methods such as pixel modification, rotation and zoom in/out are useful. Nonetheless, swirling 5° is not large enough to defeat against logo removal attack according to $\lambda > 0.5$ (see Fig.4 $\alpha = 0.5$) in the last column.

4. CONCLUSION AND FUTURE WORKS

To protect logo from automatic removing, the present paper transforms a logo before it is embedded into the frame based on random transform. The experiments indicate the various protection methods greatly reduce the possibility of localizing logo. But more experiments for robustness to variant attacks are our future works, in particular to the number of distorted logos.

Table 1. Detection value λ for the methods in Fig.1. For simplicity, (b) in the 2nd column means the method in Fig.1.b, and so on.

α	(b)	(c)	(d)	(e)	(f)	(g)	(h)
0.4	0.08	0.08	0.05	0.07	0.08	0.22	0.65
0.5	0.08	0.08	0.05	0.08	0.08	0.25	0.75
0.8	0.08	0.08	0.03	0.08	0.09	0.31	0.89
1.0	0.17	0.19	0.03	0.13	0.75	0.32	0.90

5. REFERENCES

[1] I. J. Cox, J. Kilian, T. Leighton, and T. G. Shamoon, "Secure Spread Spectrum Watermarking for Multimedia," *IEEE Trans. on I. Proc.*, 6(12):1673-1687,1997

[2] J. Meng, S.F. Chang, "Embedding visible video watermarking in the compressed domain," ICIP, 474-477, 1998.

[3] Ming Sun Fu, Oscar C. Au, "Joint Visual Cryptography and Watermarking," ICME, 975-978, 2004.

[4] R.J. Hollander, A. Hanjalic, "Logo recognition in video stills by string matching," ICIP, 517-520, 2003.

[5] S.P. Mohanty, M.S. Kankanhalli, K.R. Ramakrishnan, "A DCT domain visible watermarking technique for images," ICME, 1029-1032, 2000.

[6] A. Soffer, H. Samet, "Using negative shape features for logo similarity matching," ICPR, 571-573, 1998.

[7] S. Seiden, M. Dillencourt, S. Irani, "Logo detection in document images," Int'l Conference on Imaging Science, Systems, and Technology, 446-449, 1997.

[8] Wei-Qi Yan, Jun Wang, Mohan S. Kankanhalli, "Automatic Video Logo Detection and Removal," *Multimedia Systems*, 10(5): 379- 391, 2005.

[9] M. Bertalmio, G. Sapiro, V. Caselles, C. Ballester, "Image inpainting," *ACM Siggraph*, 417-424, 2000.

[10] W.Q. Yan, M.S. Kankanhalli, "Erasing video logos based on image inpainting," ICME, 521-524, 2002.

[11] Chun-Hsiang Huang, Ja-Ling Wu, "Attacking visible watermarking schemes," *IEEE Transactions on Multimedia*, 6(1):16-30, 2004.

[12] Y. Hu, S. Kwong, J. Huang, "An Algorithm for Removable Visible Watermarking," *IEEE Trans. on Cir. and Sys. for Video Tech.*, 16(1):129-133, 2006.