Embedding Visible Video Watermarks in the Compressed Domain

Jianhao Meng and Shih-Fu Chang
Department of Electrical Engineering
Columbia University, New York, NY 10027, USA
{jmeng,sfchang}@ee.columbia.edu

Abstract

Digital visible or invisible watermarks are increasingly in demand for protecting or verifying the original image or video ownership. We propose a novel compressed-domain approach to embedding visible watermarks in MPEG-1 and MPEG-2 video streams. Our algorithms operate on the DCT coefficients which are obtained with minimal parsing of input video. The embedded watermarks adapt to the local video features such as brightness and complexity to achieve consistent perceptual visibility. The embedded watermarks are robust against attempts of removal since clear artifacts remain after the possible attacks.

1. Introduction

Conventional watermarks can be found on valuable paper documents such as cash, check or stock certificates. These watermarks may be viewed from a certain angle or under certain illumination. They are robust in the sense that they can not be easily removed without leaving evidence of tampering. Digital images/video, when displayed on the computer monitor, do not provide equivalent physical properties (such as the surface reflectance) other than the actual pixel values to modify. When pixel values are changed, the content could be changed as well. Therefore, for visible watermarks, a balance must be maintained so that the watermark is clearly visible yet difficult to remove, and not causing too much visual distraction.

Visible watermarks are different from invisible watermarks [1], although the objective of copyright protection is very similar. Visible watermarks prevent piracy by showing the ownership claims semi-transparently on top of the images. One robust way of inserting a visible image watermark has been proposed by Braudaway *et. al.* [2]. First a luminance scaler is selected to set the strength of the watermark. Then the scaler is used to scale the watermark mask with a non-linear function. Finally, the scaled watermark image is added to the original image in the luminance channel. This algorithm modifies images pixel by pixel in the spatial domain. Uniform perceptual visibility is achieved by controlling the constant change factor in a perceptually uniform color space.

In this paper, we propose an innovative watermarking algorithm for MPEG videos. Specifically, we propose a highly efficient algorithm to embed visible video watermarks in the compressed domain without full decoding of the compressed video streams. A DCT domain motion compensation technique is also used to handle special issues in B and P frames. One unique contribution of our work is to adjust the watermark strength dynamically depending on the local features derived in the DCT domain. With the compressed domain approach, the computational complexity is greatly reduced and real-time implementation is possible.

This paper is organized as follows: Section 2 describes the creation of the watermark mask; Section 3 shows the motion compensation module needed for B and P frames; Section 4 discusses the robustness of the algorithm; Section 5 shows the experimental results followed by the conclusion in Section 6.

2. Watermark Mask Generation Module

Ideally, a new watermark mask should be generated for each frame in the video in order to adapt the watermark according to the local content in each frame. But this will make real-time implementations difficult. As a compromise, we assume that video content will not change too dramatically within a Group of Pictures (GOP) which is usually 0.5 second. We generate a new watermark mask for each new GOP and use the same mask for the entire GOP. If there is a scene change in the middle of the GOP, the visual content will change greatly. In this case, we also generate a new mask which adapts to the new content of the scene. The scene change detection algorithm was described in [5].

First, we convert the input watermark image as shown in Figure 1, to a gray scale image because the watermark is only added to the luminance channel of the original

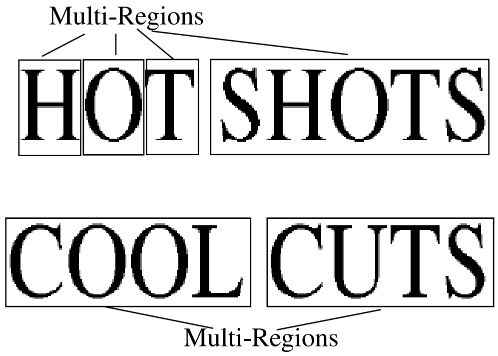

FIGURE 1. Watermark mask

image. A transparent color or background color is chosen. All pixels with the transparent color are set to 0. The non-transparent pixels are scaled based on the local image content so that the resulting watermark will have constant visibility. To increase the robustness, the watermark mask is randomly shifted in both x and y directions (in sub-pixel resolution) when embedded into the image.

2.1. Adaptive Watermark Scaling

The luminance of the mask will be scaled adaptively according to the input image content before adding to the input image. In the pixel domain, the following formulae have been proposed in [2],

$$w_{nm}' = w_{nm} \cdot \frac{y_w}{38.667}\left(\frac{y_{nm}}{y_w}\right)^{2/3} \cdot \Delta L \quad \text{for} \quad \frac{y_{nm}}{y_w} > 0.008856 \quad (1)$$

$$w_{nm}' = w_{nm} \cdot \frac{y_w}{903.3} \cdot \Delta L \quad \text{for} \quad \frac{y_{nm}}{y_w} \le 0.008856$$

where w_{nm}' is the scaled watermark mask that will be added to the original image, w_{nm} is the non-transparent watermark pixel value at (n,m), y_w is the scene white, y_{nm} is the luminance value of the input image at image coordinate (n,m) and ΔL is the scaling factor which controls the watermark strength.

We extend the above pixel domain approach to the DCT domain by using simple stochastic approximation model. Consider y_{nm} and w_{mn} as independent random variable. We normalize y from [0, 255] to [16, 235], the luminance range used in MPEG, and let $y_w=235$. Then from Eq. (1) the expected values of w' are,

$$E[w'] = 0.1607 \cdot E[w] \cdot E[y^{2/3}] \cdot \Delta L \quad , \quad y > 17.9319 \quad (2)$$

$$E[w'] = 0.2602 \cdot E[w] \cdot \Delta L \quad , \quad y \le 17.9319$$

Assuminge that y has a normal distribution with mean α, and variance β^2, then the $E[y^{2/3}]$ term in Eq. (2) can be computed by

$$E[y^{2/3}] = \int_{17.9319}^{235} t^{2/3} \cdot \frac{1}{\sqrt{2\pi\beta^2}} e^{-\frac{(t-\alpha)^2}{2\beta^2}} dt = f(\alpha, \beta^2) \quad (3)$$

Thus $E[y^{2/3}]$ is a function of the mean and the variance of the pixel value.

Eq. (2) specifies the relationship between the moments of random variables w, w', and y. We extend this relationship to the deterministic case to simplify Eq. (2). The approximation result becomes a linear one and can be easily extended to the DCT domain.

For each 8x8 image block, we use the mean and variance of each block to approximate the factors α, β^2 in Eq. (3). Further we use the mean α to approximate y to switch which one of the formulae to use in Eq. (2),

$$w'_{ijk} = 0.1607 \cdot w_{ijk} \cdot f(\alpha, \beta^2) \cdot \Delta L , \quad \alpha > 17.9319 \quad , \quad (4)$$

$$w'_{ijk} = 0.2602 \cdot w_{ijk} \cdot \Delta L , \quad \alpha \le 17.9319$$

where $k=0,..,63$, w_{ijk} is the kth pixel of ijth 8x8 block in the watermark image. w'_{ijk} is for the scaled watermark.

Basically, Eq. (4) approximates a non-linear function in Eq. (2) by a linear one block by block. The scaled watermark strength depends on the mean and variance of the image block. This is an intuitive approach. The higher mean (*i.e.*, brighter) and higher variance (*i.e.*, more cluttered) the image block has, the stronger watermark it requires in order to maintain a consistent visibility of the watermark.

We take the DCT of Eq. (4) to get the DCT of watermark mask, which can be inserted into the video in the DCT domain. The mean and variance of the input video blocks may be derived from the DCT coefficients,

$$\alpha = (Y_{DC}/8) \text{ and} \quad (5)$$

$$\beta^2 = Var(y) = \frac{\sum_{l=0}^{63} Y_l^2}{64} - \frac{Y_{DC}^{\ 2}}{64} = \frac{\sum_{l=1}^{63} Y_l^2}{64} = \frac{(\sum Y_{AC}^{\ 2})}{64} \quad (6)$$

where Y_{DC} and Y_{AC} are DC and AC DCT coefficients of the image block Y.

2.2. Region-level Adaptation

The above block based scaling of watermark image changes the watermark mask block by block and may result in visual discontinuity. To solve this problem, we separate the input watermark image into multiple regions each of which contains one or more meaningful entities (*e.g.*, see Figure 2a). For each region, we calculate the average mean and variance $(\bar{\alpha}, \bar{\beta})$ and scale the watermark mask accordingly. We found this region-level adaptation approach produces the best perceptual quality.

2.3. Randomized Location

To enhance the security of the watermark, a randomized location shifting is applied to the mask. Sub-pixel randomized location shifting makes it hard for attackers to remove the watermark without leaving noticeable residue. After the scaling with Eq. (4) and before undergoing the DCT transform, the watermark mask is shifted in x, y directions by two random numbers which are normalized between [-1,1]. Bi-linear interpolation is used in shifting the watermark image by a sub-pixel distance.

3. Motion Compensation Module

Once the DCT coefficients of the watermark mask are computed after the above three steps, they are inserted into the DCT frames of the input video differently for each of three macroblock types.

For I frame or intracoded blocks in the B or P frames, the DCT of scaled watermark is added directly,

$$E'_{ij} = E_{ij} + W'_{ij} \quad (7)$$

where, E'_{ij} is the *ij*th DCT block of the watermarked frame, E_{ij} is the original DCT block, W'_{ij} is the DCT of the scaled watermark.

For blocks with forward motion vector in P frame (or backward motion vector only in B frame), the watermark added in the anchor frame needs to be subtracted before adding the current watermark in the current frame. The resulting DCT error residue is,

$$E'_{ij} = E_{ij} - MCDCT(W'_F, V_{Fij}) + W'_{ij} \quad (8)$$

where *MCDCT()* is the motion compensation function performed in the DCT domain as described in [3]. W'_F is the watermark DCT used in the forward anchor frame. V_{Fij} is the motion vector. E_{ij} and E'_{ij} and the original and new motion compensation residue errors.

For bidirectional predicted blocks in B frame, both forward and backward motion compensation needs to be averaged and subtracted while adding the current watermark.

$$E'_{ij} = E_{ij} - (MCDCT(W'_F, V_{Fij}) + MCDCT(W'_B, V_{Bij}))/2 + W'_{ij} \quad (9)$$

where V_F and V_B are the forward and backward motion vector respectively.

For skipped blocks, which are those with 0 motion and 0 residue error blocks in B and P frames, no operations are necessary since the watermark inserted in the anchor frame are carried over.

3.1. Impact on the Bitrate

After the new DCT error residues are computed from the above steps, we provide three different quantization for output rate control. The first one is to maintain the original input stream's bitrate. We use the standard MPEG encoder virtual buffer simulator to control the bitrate at macroblock level, as shown in Figure 2a. The second option is to use the same quantization values of input stream. This approach causes the bitrate to increase by 10-12%. The third option is to use smaller quantizers to achieve comparable picture quality of the input video. This approach increases the bitrate by about 20% as shown in Figure 2b.

4. Robustness Discussion

In order to defeat or remove the watermark inserted with the above mentioned algorithm in MPEG video, one needs to recover the watermark mask, estimate the embedding locations by extensive sub pixel block matching, and then estimate the $(\bar{\alpha}, \bar{\beta})$ factors for each watermark region. We have tested several possible attacks with this approach and found they always left noticeable traces in the resulting video. The noticeable traces can be used to reject false claims of ownership and to deter piracy.

However, for video shots with simple camera motions (*e.g.*, smooth panning) and large static background, watermarks are equivalent to moving foreground objects with respect to the static background. In that case, the embedded watermarks can be possibly removed by sophisticated temporal filtering or motion stabilizing techniques. However, this type of attack will not work when there are also other moving foreground objects since the watermark will be embedded in the moving foreground objects as well.

Ongoing work includes the optimization in speed and quality. One way to reduce the bits introduced by the watermark is to take advantage of the motion information of the input MPEG video. Motion vectors are used to estimate affine parameters of the global motion introduced by camera panning and zoom. The watermark mask is then transformed with the same affine parameters. The error residue is greatly reduced after the affine transform. Visually, the watermark will appear static relative to the background.

5. Results

In Figure 2, we show the results using the adaptive watermarking techniques. We have tested the watermarking algorithm on the HP J210 workstation, where a speed of 6 frames/second on 352 pixels x 240 pixels video is achieved. The bottleneck is the MCDCT operation. There are many optimization procedures which can be further applied (such as pre-computing MCDCT of the watermark in Eq. (8) and (9)). With further optimization, software-based real-time implementation is possible. The speed is still faster than that of spatial domain approach using decoding and re-encoding approach: for 180 frames of MPEG-1 352x240 pixels, it takes 18 seconds to decode and insert watermark plus 230 seconds to re-encode. The equivalent frame rate for the spatial-domain approach is merely 0.73 frames/sec.

This algorithm has been used in the VideoQ project at Columbia University. VideoQ is an online web based video database equipped with object-oriented visual tools for retrieving high quality video footage. Demos of VideoQ and preliminary examples of the watermark techniques proposed in this are available at http://www.cnmtc.columbia.edu/videoq.

6. Conclusion

An efficient compressed-domain content-based algorithm for inserting visible watermarks in MPEG video is proposed. The proposed algorithm generates adaptive watermarks overlaying on MPEG video with consistent visibility. We have also shown that the proposed method is efficient and robust.

7. Acknowledgements

The software implementation is based on the mpeg2codec package from the public domain by MPEG Simulations Group.

The video samples are contributed by Hot Shots/Cool Cuts, Inc.

8. References

[1] I. Cox, J. Kilian, T. Leighton, and T. Shamoon, "Secure Spread Spectrum Watermarking for Multimedia," NEC Research Institute Technical Report, 95-10, 1995.

[2] G.W. Braudaway, K.A. Magerlein, and F. Mintzer, "Protecting Publicly-Available Images with a Visible Image Watermark", IBM Research Division, T.J. Watson Research Center, Technical Report 96A000248.

[3] S.-F. Chang and D.G. Messerschmitt, "Manipulation and Compositing of MC-DCT Compressed Video," *IEEE Journal of Selected Areas in Communications*, pp. 1-11, Jan. 1995.

[4] Information technology -- Generic coding of moving pictures and associated audio information," ISO/IEC 13818, International Standard (MPEG-2), 1995, http://drogo.cselt.stet.it/mpeg/standards/mpeg-2.htm.

[5] J. Meng and S.-F. Chang, "CVEPS: A Compressed Video Editing and Parsing System," *ACM 4th International Multimedia Conference*, Nov. 1996, Boston, MA.

(a) Original BitRate

(b) Smaller Quantizer 20% higher BitRate

FIGURE 2. Visible Watermark in MPEG