SPREAD TRANSFORM WATERMARKING FOR VIDEO SOURCES

John Earl *Nick Kingsbury*

jwe21@cam.ac.uk ngk@cam.ac.uk

Cambridge University Engineering Department
Trumpington Street, Cambridge CB2 1PZ, UK

ABSTRACT

Spread Transform (ST) is a quantization watermarking algorithm in which vectors of the wavelet coefficients of a host work are quantized, using one of two dithered quantizers, to embed hidden information bits; Loo [1] had some success in applying such a scheme to still images. We extend ST to the video watermarking problem. Visibility considerations require that each spreading vector refer to corresponding pixels in each of several frames, that is, a multi-frame embedding approach. Use of the hierarchical complex wavelet transform (CWT) for a visual mask reduces computation and improves robustness to jitter and valumetric scaling. We present a method of recovering temporal synchronization at the detector, and give initial results demonstrating the robustness and capacity of the scheme.

1. INTRODUCTION

Video watermarking technology has applications in several areas including copy control, broadcast monitoring, and copyright protection. While many video watermarking systems have been proposed in the research literature, most are based on a spread-spectrum embedding principle similar to Cox.

Modelling the blind watermarking problem as communications with side information at the embedder offers the prospect of improved data-hiding capacity when compared with the older spread-spectrum approach. In the early 1980s, Costa [2] addressed a simplified version of this problem, communication over a channel with input X and output Y, characterized by $Y = X + S + Z$, where the channel noise components S and Z are zero-mean i.i.d. Gaussian; he showed that if host interference S is known to the embedder (though not to the detector), theoretically the capacity of the channel is independent of the variance of S. Efforts toward developing practical codes in this framework include the Scalar Costa Scheme (SCS) [3], Quantization Index Modulation (QIM) [4], and Spread Transform (ST) [1]; all have been devloped and tested on still image sources.

Here we bring the ST approach to video, so as to develop a robust watermarking technology that can reliably achieve higher capacity than the current state of the art. We embed in very coarse scale subbands in the CWT domain, to reduce computation, incorporate limited contrast masking, and improve robustness to compression. The structure of the ST embedder implies adopting a novel multi-frame watermarking scheme in which the watermark cannot be detected from a single frame alone, but only from a series of frames. We describe methods of recovering the temporal synchronization of such a watermark at the detector using the properties of an error-correcting code (we use turbocodes). Finally, we present initial results demonstrating the robustness of this technique to MPEG-2 compression.

2. PRELIMINARIES

2.1. Spread Transform

Spread Transform, first described by Chen [4], is a practical informed watermarking method. The embedder chooses a column vector $\mathbf{x}$ of coefficients from the host work and projects it in a key-dependent random direction $\mathbf{v}$, to compute a projection,

$$p = \frac{\mathbf{x}^T \mathbf{v}}{\mathbf{v}^T \mathbf{v}} \tag{1}$$

The embedder quantizes p using one of two quantizers $Q_i(\,)$, $i \in \{0, 1\}$ of identical step size Δ, where the choice of i encodes a bit. The bins of the quantizers $Q_i(\,)$ are offset from each other by $\frac{\Delta}{2}$ and shifted by a key-dependent dither for added security. The detector merely computes the same projection (1); the received bit is r such that if

$$\text{err}_i = \|Q_i(p) - p\|, \tag{2}$$

$$r = \begin{cases} 0 & \text{err}_0 < \text{err}_1 \\ 1 & \text{otherwise} \end{cases} \tag{3}$$

This work was supported by a generous grant from BBC Research and Development.

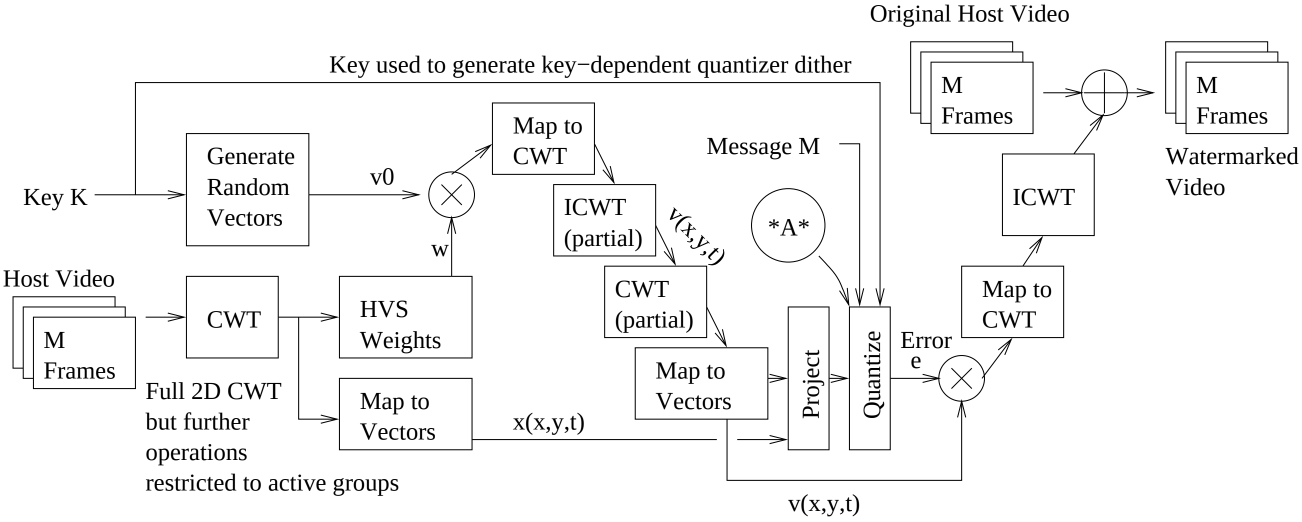

Fig. 1. Multi-frame CWT spread transform video embedding algorithm. The message m is in practice coded using an error-correcting code (here, a rate 1/3 turbocode) prior to embedding.

In our implementation, the vector $\mathbf{x}$ is composed of n wavelet coefficients of the host video which are located spatially and temporally near each other. We also take the same set of wavelet coefficients and compute a masking function, described in the next section, from which we create a similarly n-dimensional weighting vector, $\mathbf{w}$. We choose the component-wise product of weights $\mathbf{w}$ with a key-dependent pseudo-random direction $\mathbf{v}_0$ as the projection direction,

$$\mathbf{v} = \mathbf{w}.\mathbf{v}_0 \tag{4}$$

Because the projection direction is dependent on a function $\mathbf{w}$ which scales with activity in the host vector $\mathbf{x}$, this technique is not as vulnerable to valumetric scaling attacks as other quantization-based methods. We describe the details of our embedding algorithm below in section 3.1.

2.2. Complex Wavelets

Loo [1] proposed that the complex wavelet transform (CWT) domain described in [5] should have advantages over the DWT for visual masking of watermark patterns, due to the addition of shift invariance, improved directional selectivity, and the similarity of CWT filters to the Gabor filters used by Watson for the cortex transform [6]. He developed a model to support visual masking in the CWT domain:

$$g_{l,\theta}(u,v) = \beta\sqrt{k_{l,\theta}^2\|\bar{x}_{l,\theta}(u,v)\|^2 + \gamma_{l,\theta}^2} \tag{5}$$

where $g_{l,\theta}$ is the allowable watermark gain in the subband at scale l oriented in the direction θ; β represents absolute luminance effects modelled with a quadratic function; k controls the masking contrast for the subband; γ is the contrast masking threshold; and $\|\bar{x}(u,v)\|^2$ is a lowpass filtered version of the squared magnitude of local CWT coefficients in the subband centered at spatial coordinates (u,v). Spread spectrum (SS) and ST watermarks were embedded in still images using this mask at the finer scales ($l = 1$ to $l = 3$); coarser scales were ignored.

While this visual mask can be applied similarly to video sources, the complexity limitations of video processing and the 4:1 redundancy of the two-dimensional CWT, combined with the improved robustness to lossy compression available at lower frequencies, makes the choice of coarser scale wavelet coefficients ($l = 4$ or $l = 5$) more appropriate. When using this mask at coarser scales, we also find it helpful to replace the Gaussian low-pass filter mentioned above with a variant of a 3x3 median filter which selects the second-smallest coefficient from each 3x3 region instead of a 3x3 Gaussian, to avoid watermark artifacts spread around image features due to the large support of each wavelet coefficient.

We find that for coarser scales the parameter $k_{l,\theta}$, which controls the dependency of the mask on spatially local energy in the subband (l,θ), must be relatively weak compared to the fine-scale case, where local contrast masking is more significant. However, we retain this wavelet masking scheme, partly to aid robustness to valumetric scaling (section 2.1), and partly to optimise the tradeoff between watermark energy and visibility. In addition, the shift-invariance of the CWT filters improves resistance to jitter, i.e., translation of the whole picture by a few pixels vertically or horizontally. Coarse-scale CWT coefficients vary more smoothly under jitter than do those of alternative transforms such as Hadamard, DWT, and DCT.

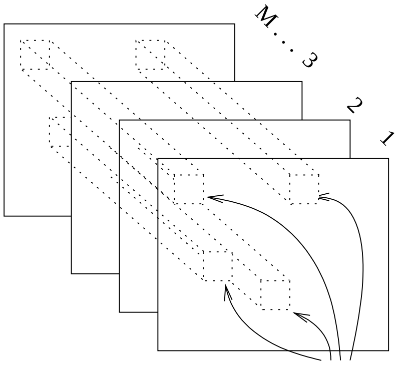

Fig. 2. Four component blocks from an active group, across M frames of wavelet coefficients, from which four spreading vectors are formed—each available for embedding one bit.

3. WATERMARKING ALGORITHM

3.1. Embedder

Applying quantization watermarking techniques directly to a sequence of frames, each quantized independently of the next, leads immediately to a flickering watermark pattern, easily perceptible to the eye. To avoid this flicker, we implement a multi-frame embedding scheme (figure 1) in which detection of any bit requires a frame group of length M frames.

Frames of the host work are divided into blocks of pixels (e.g., 64x64). The message to be transmitted, m, is coded using an error-correcting code; one coded bit is embedded in each block. The embedder computes the 2D CWT, to l levels, of each of a sequence of M frames. The vector $\mathbf{x}$ used to embed one bit is composed of CWT coefficients corresponding to a 3D block of pixels (M frames temporally plus two spatial dimensions) — as in figure 2. If $l = 5$, for instance, for a 704x576 PAL frame, there are six directional subbands of complex coefficients each of size 22x18. If blocks are size 64x64 in pixel space, there are 99 blocks and the block size is 2x2 complex coefficients. The six directional subbands and the real and imaginary parts of each wavelet coefficient yield 2x2x6x2 = 48 real coefficients per block per frame over M (say, 8) frames. This yields 384 real coefficients to form a single spreading vector $\mathbf{x}$, which is used to embed a single bit. Hence the bitrate before turbo decoding is 99 bits per M frames (giving 31 bits after decoding). If it is known that the same message will be transmitted continuously, then the detection results at the input to the soft-decision turbo decoder may be accumulated for improved detection.

In the absence of significant motion in the host video, the actual watermark values added to the video during embedding remain nearly constant across the group of M frames. We have found that the sharp change in embedded watermark following the Mth frame increases watermark visibility. Changes in the embedded pattern become less visible if only part of the watermark changes each frame. We therefore divide the available blocks into M block groups, and temporally stagger the embedding of each block group by one frame. Operating in this mode, full decoding of a message requires $2M - 1$ frames, although the overall data-hiding bitrate remains the same.

The partial inverse CWT and forward CWT in figure 1 deserve special explanation. Because the CWT is a redundant transform, any pseudo-randomly generated signal will be partly in the range space, and partly in the null space of the transform. Watermark patterns added to the host work are oriented in directions corresponding to the spreading vectors $\mathbf{v}$ after an ICWT step removes any parts in the null space; the detector (and the embedder, which must use the same vectors so that the quantization bins match) should therefore use spreading vectors $\mathbf{v}$ entirely in the range space. One cycle, inverting the CWT 2 or 3 levels, then applying a matching forward transform, accomplishes the removal of null space components.

3.2. Decoding and Detection

If the decoder knows on which frame the watermark began, that is, if the temporal synchronization is known, then the decoding process is identical to the embedding process of figure 1 up to the point labelled **A**. Here, rather than quantize the projection p according to a coded message bit, the detector follows equations (2) and (3), and decodes each message bit according to which of the two quantizers yields the smaller error. Thus, any host work, whether or not it contains a watermark, will decode into a bit sequence.

In the detection problem, we address whether a watermark is present in a given work. In this case, detection comes in two stages. First, we use the error correcting code as in [7]:

- Decode the received message r to produce a decoded message d
- Code the decoded message d to produce r'
- Compute the bit error rate of r' with respect to r and compare with a threshold.

If additional insurance against false positives is required, in some applications it may be appropriate to fix some number of message bits; if twenty such bits were fixed then a further improvement in false positive probability of 2^{-20} (10^{-6}) is achievable.

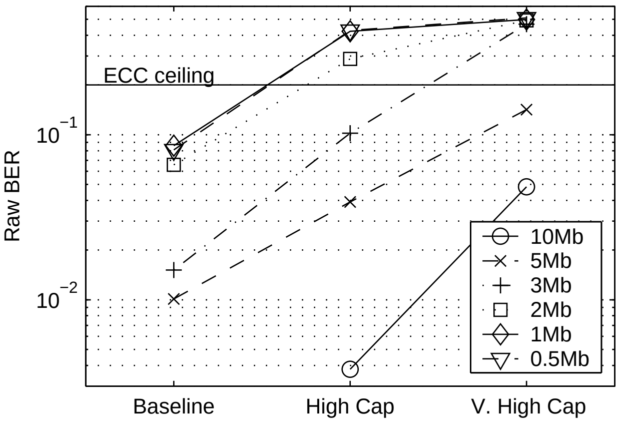

Fig. 3. Robustness of scheme to MPEG-2 attacks

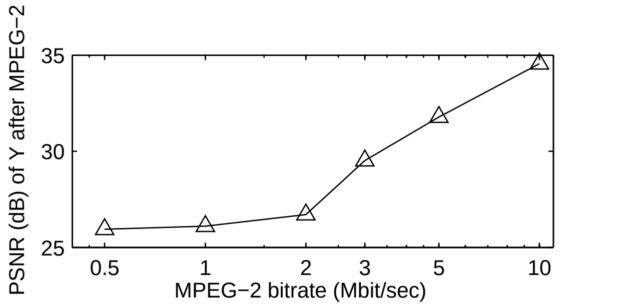

Fig. 4. Measured quality degradation from MPEG-2 coding

3.3. Temporal synchronization

Assume for the moment that the same message is transmitted repeatedly over consecutive groups of M frames. At any given frame, if the detector knows the frame group length, M, then, if the video contains a watermark, the last M frames inclusive must allow detection of exactly one block group; the following block group is expected to follow in the next M-frame group, and so on. Therefore, after $2M - 1$ frames, the detector accumulates M different possible messages. We then apply one or both of the detection schemes of section 3.2 to determine which is the correct group timing.

If the message is not known to remain constant, then the detector must search all M possible starting offsets. This means that a total of $3M - 1$ frames are required in order to determine whether a watermark is present.

4. RESULTS

We present results for three configurations with $M = 8$ and message kept constant over $4M$ frames (just over a second of PAL video). These are baseline robust (31 bits per $4M$ frames, $l = 5$, blocksize 64x64), high capacity robust (130 bits, $l = 4$, blocksize 32x32), and very high capacity (526 bits, $l = 4$, blocksize 16x16). The quantization step size Δ is set so as to achieve a watermark pattern with RMS 0.5 (half an 8-bit luminance quantization step), an imperceptible level. Figure 3 shows performance in terms of the raw bit error rate under various MPEG-2 attacks, for each configuration, when embedded in the standard table-tennis test sequence. When the raw bit error rate falls below the ECC ceiling of 0.2, the watermark is considered detectable and the message is correctly decoded with high reliability. Thus, the baseline robust system appears robust to MPEG-2 down to 0.5 Mb/sec. By contrast, the high capacity robust system survives down to 3 Mb/sec, while the very high capacity scheme survives only 10 Mb/sec compression.

For this work, we have used a software MPEG-2 codec (from the MPEG Software Simulation Group, MSSG). It is likely that a properly-configured hardware MPEG-2 coder using the same bit-rates would produce significantly lower distortion. For context, we report the PSNR values experimentally measured on this sequence at each bitrate (figure 4).

5. REFERENCES

[1] P. Loo, *Digital Watermarking with Complex Wavelets*, Ph.D. thesis, Cambridge University Engineering Department, Cambridge, UK, Feb. 2002.

[2] M. Costa, “Writing on dirty paper,” *IEEE Transactions on Information Theory*, vol. IT-29, no. 3, pp. 439–441, May 1983.

[3] J. Eggers and B. Girod, “Quantization watermarking,” in *Proceedings of SPIE: Security and Watermarking of Multimedia Contents II, Electronic Imaging 2000*, San Jose, CA, USA, Jan. 2000.

[4] B. Chen, *Design and analysis of digital watermarking, information embedding, and data hiding systems*, Ph.D. thesis, MIT, Cambridge, MA, June 2000.

[5] N. G. Kingsbury, “Image processing with complex wavelets,” *Phil. Trans. R. Soc. Lond. A*, vol. 357, pp. 2543–2560, 1999.

[6] A. Watson, “The Cortex Transform: Rapid computation of simulated neural images,” *Computer Vision, Graphics, and Image Processing*, vol. 39, pp. 311–327, 1987.

[7] P. Loo and N. G. Kingsbury, “Watermark detection based on the properties of error control codes,” in *IEE Proceedings - Vision, Image and Signal Processing*, 2002, (Submitted).