AN ACCURATE BILLING MECHANISM FOR MULTIMEDIA COMMUNICATIONS

José Gabriel Gomes, Mylène C. Q. de Farias, Sanjit K. Mitra

Department of Electrical and Computer Engineering
University of California at Santa Barbara
ECE Dept., Santa Barbara, CA 93106 USA
Email: (gabriel, mylene, mitra)@ece.ucsb.edu

Marco Carli

Department of Electronic Engineering
University of Roma TRE
via della Vasca Navale, 84, 00146, Rome, ITALY
Email: carli@ele.uniroma3.it

ABSTRACT

A novel billing mechanism is presented in this paper to provide telecommunication service users and advertisers with an effective billing mechanism based on the actual amount of advertisement data being transmitted/displayed. Experimental results show the effectiveness of the proposed system in terms of computational cost and performance.

1. INTRODUCTION

In many applications, the video communication data is usually composed of advertisement and the message itself. In the new generation of fast moving telecommunications service delivery environment, an adaptive service charging system is needed to provide an accurate billing for the personalized-user services offered over different networks. This paper presents a billing mechanism for multimedia communications by checking the effective time length of the service offered.

To test the proposed idea, two different scenarios have been examined: a television-based video communication system and a third generation mobile communication system. We choose to analyze also this latter environment since it is probable that the next generation of mobile communication will be based on fast, reliable, multimedia based communication.

2. WATERMARK EMBEDDING PROCEDURE

In this paper, we use a watermarking system for billing purposes. For this application, the frames of a video belonging to an advertiser are embedded with the same watermark, as shown in Fig. 1. The proposed watermarking procedure embeds a 2-D image into the DCT (discrete Cosine transform) of the blue component of each frame, since errors in this component are less visible. In MPEG applications, the C_b component is used, for the same reason. A spread-spectrum technique is then used to hide the marker, by multiplying the marker image by a pseudo-random noise before embedding it into the video [1]. Figure 2 depicts the black and white 24×24 marker images used in this work. The embedding procedure is summarized as follows.

First, the marker image is read and repeated 16 times (in a 4×4 matrix format). This repetition creates enough redundancy to

This work was supported in part by CAPES, Brazil, in part by a University of California MICRO grant with matching support from Conexant Systems, Lucent Technologies and Philips Research Labs and in part by Microsoft Corporation.

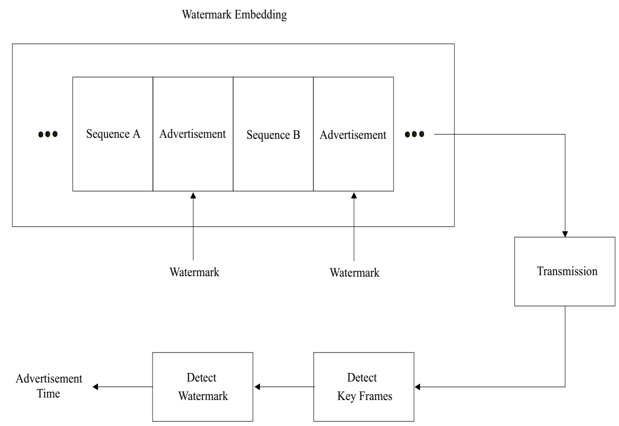

Fig. 1. System block diagram.

make it possible to recover the watermark image at the receiver within each frame. Second, a pseudo random algorithm is used to generate the pseudo-noise image $p(\mathbf{n})$, with values in the range $[-1, 1]$. The final watermark $\tilde{w}\left(\mathbf{n}\right)$ is obtained by multiplying the watermark image $w(\mathbf{n})$ by $p(\mathbf{n})$:

$$\tilde{w}\left(\mathbf{n}\right) = w\left(\mathbf{n}\right) p\left(\mathbf{n}\right) \quad (1)$$

Next, the DCT coefficients of the blue component of the frame are computed. The marker is added to the mid-frequencies of these DCT coefficients. The final marker is multiplied by a scaling factor α and is added to the DCT coefficients:

$$Y_i\left(\mathbf{n}\right) = \mathrm{DCT}\left(f_i(\mathbf{n})\right) + \alpha \cdot w\left(\mathbf{n}\right) \quad (2)$$

For all the experiments shown in Sec. 5, the scaling factor α is chosen so that the watermark strength (i.e. its power) is about 7% of the video signal variance, which is enough for good robustness to transmission noise.

3. SHOT BOUNDARY DETECTION ALGORITHM

An optimal vector quantizer is known to be a function of the statistics of the signal for which it was designed [2]. This suggests that, in the case of video signals, a vector quantizer specially designed

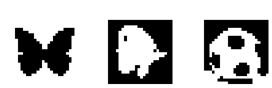

Fig. 2. Marker images used by the embedding algorithm.

0-7803-7965-9/03/$17.00 ©2003 IEEE

Table 1. Summary of VQ key frame extraction procedure

- **Codebook creation**
  - **Loop for first K frames after transition frame S.**
    Decimate frame by L on both directions;
    Partition decimated frame into 48 8x8 blocks;
    Calculate DC level of each block (optionally, calculate 8×8 DCT);
    Read 2×2 blocks of DC coefficients into 4×1 vectors;
    Store the 12 4-dimensional vectors $b(i_f, i_b)$;
  - **Set codebook to initial default $P/2 \times \Gamma_0$ according to average 2-norm P of training set vectors**, where Γ_0 is a 16×4 matrix with 16 4×1 initial centroids
  - **Loop for 12×K-vector training set (GLA)**
    Encode all vectors using nearest neighbor partition;
    Find new centroid positions by averaging clusters;
    Create new centroids if required;
    Calculate average distortion in vector quantization;
    If distortion variation is under threshold, finish loop;
- **Content change evaluation**
  - **Loop for all frames starting at S**
    Decimate frame by L on both directions;
    Partition decimated frame into 48 8x8 blocks;
    Calculate DC level of each block (optionally, calculate 8×8 DCT);
    Read 2×2 blocks of DC coefficients into 4×1 vectors;
    Encode 4-dimensional vector using current codebook;
    Calculate three change markers in Eqs. (7), (8) and (9);
  - **If markers define a shot boundary, store index S and go to Codebook creation procedure**

for segments of a video sequence can be used as a tool for fast analysis of the variations of the video content. This idea was developed and validated with experimental results in [3], and a summary of the algorithm is presented in Table 1.

The content change evaluation methods rely heavily on the codebook. It has been verified in [3] that using the first 20 frames from the video sequence immediately after the shot boundary detection will yield 16-centroid condebooks that are robust enough for shot-boundary detection. In the second part of the algorithm, i.e., when a frame is processed for content evaluation, the distances from each vector to its centroid in frame S are computed:

$$d(S, i_b) = ||b(S, i_b) - \Gamma(S, c(S, i_b))||, \quad (3)$$

where $||x||$ denotes the 2-norm of vector x. At frame $S + D$, where D is an arbitrary positive offset with respect to the shot boundary, the vectors $b(S + D, i_b)$ are also assigned to centroid indices $c(S + D, i_b)$, and the current distortion measurements are taken:

$$d(S + D, i_b) = ||b(S + D, i_b) - \Gamma(S, c(S + D, i_b))||, \quad (4)$$

If $D \geq 1$, then change measurements can be computed from Eqs. (5)-(9) for all 12 blocks within frame $S + D$:

$$\begin{aligned} \Delta d(S + D, i_b) &= d(S + D, i_b) - d(S + D - 1, i_b) \\ \Delta d_0 &= \Delta d(S + D, i_b) \\ \Delta d_1 &= \Delta d(S + D - 1, i_b) \end{aligned} \quad (5)$$

$$\begin{aligned} V_1 &= (\Delta d_0 > r_{min}(S, c(S + D, i_b))/2.5) \\ V_2 &= (\Delta d_1 > r_{min}(S, c(S + D - 1, i_b))/2.5) \\ V_3 &= (c(S + D, i_b) = c(S + D - 1, i_b)) \end{aligned} \quad (6)$$

$$M_B(S + D) = \begin{cases} M_B(S + D) + 1, \text{ if } V_1; \\ M_B(S + D), \text{ else.} \end{cases} \quad (7)$$

$$M_A(S + D) = \begin{cases} M_A(S + D) + 1, \text{ if } V_1 \oplus V_2 \\ M_A(S + D), \text{ else.} \end{cases} \quad (8)$$

$$M_C(S + D) = \begin{cases} M_C(S + D) + 1, \text{ if } V_3; \\ M_C(S + D), \text{ else.} \end{cases} \quad (9)$$

Equation (7) yields a marker that takes into account the number of blocks with a distortion change above 80% of the size of the codebook cell to which they are individually associated. Equation (8) defines a measurement of block change similar to M_B, except for the fact that a frame-memory of size $F = 1$ is included. For the detection of longer transitions such as dissolving effects, the memory size F can be increased. Extremely sharp transitions are handled by the marker M_C in Eq. (9). Three examples of completely reassigned vectors, representing blocks that are detected by M_C, are provided in Fig. 3.

4. WATERMARK DETECTION

After the key frames are obtained it is necessary to find out which of these frames belong to the same advertiser. This is done by detecting the advertiser's watermark for each of these key frames. The first and last frames belonging to the same advertiser will be used to determine how much time the advertisement has been transmitted/displayed.

The watermark detection method can be summarized as follows. First, the DCT of the blue component of each key frame $f'_i(\mathbf{n})$ is calculated:

$$G'_i(\mathbf{h}) = DCT\left(f'_i(\mathbf{n})\right), \quad (10)$$

Second, the DCT coefficients where the marker was inserted are multiplied by the corresponding pseudo-noise image. We assume that the receiver knows the initial position in the DCT domain where the marker is inserted. The result is averaged for the 16 repeated pixels (4×4 matrix). The extracted binary watermark is then obtained by taking the sign of this final result:

$$\tilde{w}_r(\mathbf{h}) = sign\left(\frac{1}{16}\sum_{i=1}^{16} G'_i(\mathbf{h})\, p_i(\mathbf{n})\right), \quad (11)$$

Equation (2) is always invertible, i.e., an inverse function can be computed and the original marker extracted. However, the marker will be degraded due to the DCT transformation, the pseudo-noise

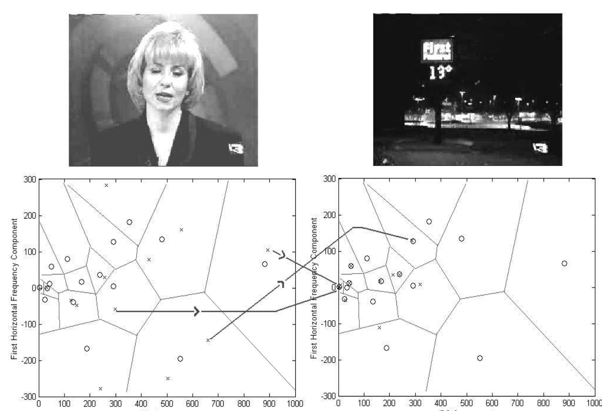

Fig. 3. Changes of centroid assignment detected by M_C between two consecutive video frames.

Table 2. Main routine for timing and billing simulation

- **Set advertisement count to** `i=0;`
- **Loop for all frames in a given video sequence**
  - Read frame S and calculate 4×1 vectors;
  - Shot boundary detection;
  - If shot boundary is detected, then look for watermark in previous and next frames;
    - ∗ If watermark found and advertisement not started, set: `InitialTime=CurrentTime;` Flag that advertisement has started to `true`;
    - ∗ If watermark not found and advertisement started, set: `FinalTime=CurrentTime;` Flag that advertisement has started to `false`; `Ad_Length(i)=FinalTime-InitialTime;` Advertisement index update: `i=i+1;`

image $p_i(\mathbf{n})$, and to any loss the video suffers, such as compression, channel noise, etc. A measure of the degradation of the recovered watermark is then obtained by calculating the mean square error of between $\tilde{w}(\mathbf{n})$ and $\tilde{w}_r(\mathbf{n})$ which are, respectively, the original and the extracted watermarks. The MSE is used as an indicator of the presence of the mark in the frame being tested.

5. SIMULATION EXAMPLE AND RESULTS

In this section, the computer simulations results of the proposed billing method are presented. The procedure for generating test sequences is shown in Subsection 5.1, and simulation results are provided in Subsections 5.2 and 5.3.

A summary of the billing algorithm is presented in Table 2. At the beginning of the video transmission, the number of times the advertisement was shown is set to zero, and the transmitted frames are analyzed progressively in search of the shot boundaries. If a boundary is found, then the beginning or end of the advertisement is decided from the presence or absence of the advertiser's watermark. The broadcast duration of every occurrence of the advertisement is stored in the successive entries of the vector `Ad_Length`.

5.1. Test Sequence Generation

The algorithm shown in Table 3 is used for creating a video sequence that represents two occurrences of a given advertisement, interleaved with two other sequences of no interest (Table 4). The basic watermarked video sequence is then corrupted in two different ways, to model the communication problems found in television broadcasting and in wireless channels with error correction.

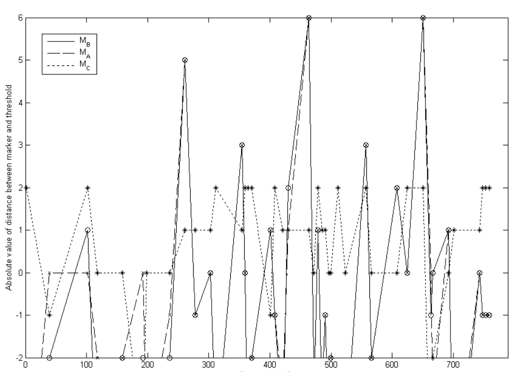

Fig. 4. Use of markers M_A, M_B and M_C to detected shot boundaries in Table 5. The curves plotted are $M_A - 6$, $M_B - 6$ and $2 - M_C$, and the detection threshold is at 0.0 for all markers.

Table 3. Creation of Test Sequences with Watermarks and Transmission Errors

- **Define watermark:** `mark_ad.bmp`
- **Define a basic sequence of watermarked bitmaps**
  - Sequence A - Marked with `mark1.bmp`
    Advertisement - Marked with `mark_ad.bmp`
    Sequence B - Marked with `mark2.bmp`
    Advertisement - Marked with `mark_ad.bmp`
- **In the case of signal for television (loop for all frames)**
  - For frame S, represented by a bitmap, choose E% of the pixels at random, and add random noise to these pixels by changing each of their R, G and B values by a random increment between -128 and 128. Values smaller than 0 or greater than 255 are truncated to 0 and 255, respectively;
  - Add frame S to video stream (`.avi` format);
- **In the case of a sequence for MPEG-2 Testing**
  - Convert the whole sequence to MPEG video stream;
  - According to the channel bit error rate (P_{error}), choose P bits of the MPEG stream at random, and invert the value of these bits. $P = \lfloor P_{Error} L \rfloor$, L is the stream length, in bits;
  - Write the modified bit sequence to a new MPEG stream;

5.2. Generic Television Testing

The sequence that was created as described in Section 5.1 has been read on a frame-by-frame basis. All shot boundaries have been detected and these frames are processed by a watermark detection routine so that the presence of the advertiser's marker can be verified. The results in Table 5 are obtained with random changes in 0.5% ($E = 0.5$) of the values in the components R, G and B of the bitmap sequence. The shot boundaries in this Table have been detected using the markers M_A, M_B and M_C (Eqs. (7)-(9)), as shown in Fig. 4. The results in Table 5 agree with those of Table 4, which means the advertisement has been detected and timed correctly in both the times it occurs. Even for values of E as high as 20, i.e., 20% of the RGB values are independently damaged by the transmission (left picture in Fig. 6), the presence of the marker in the advertisement can still be detected, as shown in Figure 5.

Table 4. Structure of the watermarked test sequence

Frame	Description
1	Beginning of video
236	**Advertisement starts**
401	**Advertisement finishes**
625	**Advertisement starts**
789	End of video

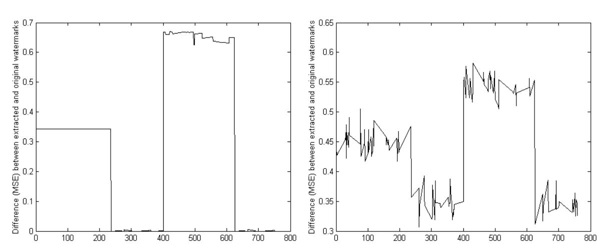

Fig. 5. Mean Squared Error (MSE) between watermarks extracted from shot boundaries and the original watermarks, for the cases of 0.5% (left) and 20% (right) error in the R, G and B fields.

Table 5. Shot boundaries detected and their classification into advertisement boundaries

S	T	S	T	S	T	S	T	S	T
1		278		421		511		692	
39		303		430		523		701	
102		312		464		557		743	
118		354		472		566		748	
158		359		478		608	$\surd$	753	
192		364		485		625		759	
198		370		490		650			
236	$\surd$	401	$\surd$	496		663			
261		408		499		666			

S: Frame index of detected shot boundary;
$T = \surd$ means shot boundary is also start/end of advertisement;

5.3. MPEG-2 Testing

The compressed sequence that was created as described in Section 5.1 is processed so that, for each frame, the 48 DC coefficients necessary for shot boundary detection are obtained. Since these coefficients are already available for the intra frames in the MPEG system, a fast processing using the motion vectors also available in the stream allows for the reconstruction of the required coefficients in real time. The results are very similar to those in Table 5, and are obtained with $P_{error} = 10^{-6}$ (right picture in Fig. 6), which is a value commonly assumed in the literature, for the analysis of wireless systems with transmission based on error correcting codes. Even for strong MPEG compression factors such as the 20:1 factor used for the results in this Section, which strongly impairs the watermark detection, the advertisement validation/billing is done correctly, as Fig. 7 shows.

6. CONCLUDING REMARKS

The experimental results show that the proposed system can achieve a very good performance in the case of typical television broadcasting under noise. Both the timing and billing are exact. Although we present results obtained from a small database, the 100% detection accuracy is confirmed by tests run on larger databases, whose results were published in [3]. The performance of watermark detection for low bitrates is impaired to some extent since the watermark is sensitive to MPEG compression, but the advertisement is still correctly timed. On the other hand, the algorithm for billing from MPEG streams seems to be very robust to the transmission error rate, so that the performance remains acceptable as P_{error} is increased. Future work will involve the analysis of the trade-off between successful watermark extraction and P_{error} as it is increased to values around 10^{-2}.

The validation of both billing systems through MATLAB simulation results sets a completely generic framework for the implementation of such systems in a wide variety of contexts, including the current and next generation wireless standards. The television routines running at 1 frame/sec can be accelerated by a factor of 50

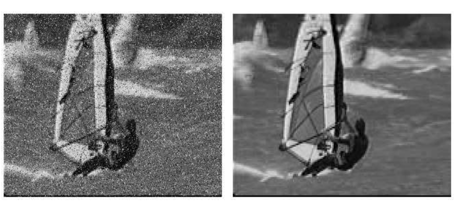

Fig. 6. Frames from sequence of bitmaps with 20% error rate in the R, G and B fields (left) and from MPEG video sequence transmitted with bit error rate $P_{error} = 10^{-6}$ (right).

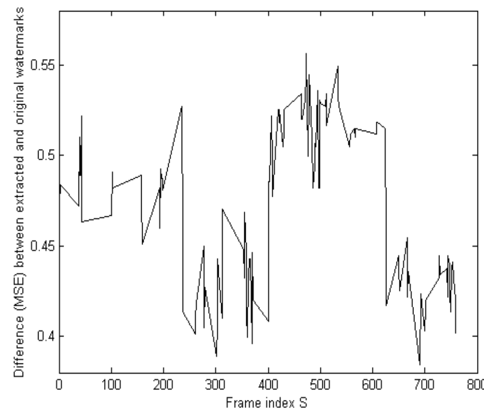

Fig. 7. Mean Squared Error (MSE) between watermarks extracted from shot boundaries and the original watermarks, for MPEG with compression 20:1 and 10^{-6} bit error rate.

in a C++ implementation, and the MPEG routines running at 0.5 frame/sec by a factor of 20, which makes both C++ implementations real-time. By using the designed system, telecommunication service users and advertiser have an effective billing mechanism based on the real amount of advertised data transmitted.

In both cases, the advantage of the proposed billing system comes from the fact that not every frame in the video sequence has to be analyzed (in the case of the generic television system) or decoded-and-analyzed (in the case of the MPEG system) for verification of the presence of the marker. Only 5% of all the video frames were processed to the marker extraction routine, while the remaining 95% were screened out by a computationally inexpensive shot boundary detection routine. We point out that the VQ detection system implemented by Eqs. (7) to (9) has a very simple implementation, and make use of MSE for watermark detection and repetition codes for robustness. We propose a thorough analysis of the specific compromise between 95% VQ and 5% watermark detection as part of our future work. The images in Fig. 2 would be eventually replaced by the logo of different companies, but the same system can be used for embedding arbitrary binary strings or bar code information to identify the advertiser. It should be pointed out that the system probability of error is likely to be improved by embedding binary messages using error correcting codes, instead of using the repetition of binary images.

7. REFERENCES

[1] I. Cox, J. Kilian, F. Leighton, T. Shamoon, "Secure spread spectrum watermarking for multimedia," *IEEE Trans. Image Processing*, vol. 6, no. 12, pp. 1673-1687, December 1997.

[2] A. Gersho, R. Gray, *Vector Quantization and Signal Processing*, Kluwer, 1991.

[3] J.G.R.C. Gomes, M. Carli and S.K. Mitra, "Vector quantization-based automatic shot boundary detection," *Proc. IEEE International Workshop on Multimedia Signal Processing*, St. Thomas, Virgin Islands, USA, December 2002.

[4] T. Lin, CW. Ngo, H. Zhang, Q.Y. Shi, "Integrating color and spatial features for content-based video retrieval," *Proc. 2001 Int. Conf. on Image Proc.*, (vol.3), Thessaloniki, Greece, pp. 592-595, Oct. 2001.

[5] R. Zabih, J. Miller, K. Mai, "A feature-based algorithm for detecting and classifying production effects," *Multimedia Systems*, vol. 7, (no.2), Springer-Verlag, p. 119-28, March 1999.