Digital Watermarking of 3D Head Model Animation Parameters

Frank Hartung, Peter Eisert, and Bernd Girod

Telecommunications Institute I
University of Erlangen-Nuremberg
Cauerstrasse 7, 91058 Erlangen, Germany
Phone +49 9131 85 7116, Fax +49 9131 85 8849
{hartung,eisert,girod}@nt.e-technik.uni-erlangen.de

Abstract

The enforcement of intellectual property rights is difficult for digital data. One possibility to support the enforcement is the embedding of digital watermarks containing information about copyright owner and/or receiver of the data. Watermarking methods have already been presented for audio, images, video, and polygonal 3D models. In this paper, we present a method for digital watermarking of model animation parameter data sets, for example MPEG-4 facial animation parameters. The watermarks are additively embedded into the parameter values and can be retrieved either from the watermarked parameters or from video sequences rendered using the watermarked animation parameters for head animation. For retrieval from rendered sequences, the facial animation parameters have to be estimated first using appropriate estimation methods. Experimental results confirm the applicability of the presented watermarking technique.

1 Introduction

For rendering of video sequences using animated 3D models, usually 3 separate data sets are utilized: a data set describing the basic shape of the 3D object (e.g. a wireframe or a spline surface), a data set describing the surface brightness and color (the texture), and a data set describing the temporal variation of the model (the animation parameters). For the special case of animated head-and-shoulder sequences, the International Standardization Organization (ISO) has standardized a format for the representation and animation of such 3D models [1].

Each of the three mentioned data sets can be regarded as an own entity where own associated intellectual property rights may exist. As with all other digital multimedia data, copy protection is difficult to achieve for 3D model and animation data. Every copy of digital data has the same fidelity as the original. Thus, mechanisms are desirable to enforce intellectual property rights. One possible solution to the problem is digital watermarking [2]. Digital watermarking allows to robustly embed copyright information, as well as information about the receiver of data, into digital data. It does not prevent copying, but can help to identify and trace back illegal copies of multimedia data. Since textures of 3D models are often represented as a 2D image, most of the presented methods for images [9, 3, 10] are also applicable to watermarking of texture images. 3D polygonal models can be protected by watermarking using the methods suggested in [11].

We propose a method that allows to embed digital watermarks into MPEG-4 facial animation parameters (in the following called FAPs). The method derives from ideas of spread spectrum communications, and is applicable to other model animation parameters as well. The watermark can either be retrieved directly from the watermarked parameters, or from the video sequence rendered from the model using the watermarked parameters. In the latter case, the animation parameters have to be estimated from the sequence, and the watermark is then retrieved from the estimated parameters.

2 MPEG-4 Face Animation

The video compression standards MPEG-1 and MPEG-2 are widely known and used for video storage and distribution. The new standard MPEG-4 [1] does also supply tools for video and still image compression, but has additional functionalities and features beyond waveform-based compression, such as mechanisms for face animation. The tool of MPEG-4 face animation allows the compression of head-and-shoulder scenes down to bit-rates of very few kbit/s. This is beyond today's possibilities of waveform-based compression. A generic face is predefined in MPEG-4 and can be animated with facial animation parameters (FAPs). Additionally, a particular face can be transmitted if desired using so-called facial definition parameters (FDPs).

A set of 66 FAPs is defined in MPEG-4, Annex C [1]. It includes global head motion parameters (e.g.

head pitch and yaw angles) and local face motion parameters (e.g. opening of eyelids, opening of lips, movement of innerlip corners). In total, there are 16 FAPs controlling the jaw, chin, inner lips and cornerlips; 12 FAPs controlling the eyeballs, pupils and eyelids; 8 FAPs controlling the eyebrows; 4 FAPs controlling the cheeks; 5 FAPs controlling the tongue; 3 FAPs controlling the global head rotation; 10 FAPs controlling the outer lip positions; 4 FAPs controlling the nose; and 4 FAPs controlling the ears. It is valid to transmit only a subset of the defined FAPs in order to animate a head model. The output of the MPEG-4 decoding process is a stream of decoded FAPs. It is left to the implementation how to generate the FAPs at the transmitter side, and how to animate the 3D head model with the FAPs at the receiver side.

In the following, we regard the FAPs as a time-varying k_{max}-dimensional vector $\mathbf{FAP}(t)$ where k_{max} is the number of transmitted FAPs ($k_{max} \leq 66$), t is the integer time index of the video frame, and $FAP_k(t)$ is the k-th FAP at time t ($k \in \{1 \ldots k_{max}\}$). Thus, $FAP_{k=const}(t)$ denotes a sequence of a certain FAP value over time, while $\mathbf{FAP}(t = const.)$ denotes all FAPs at a given time instant.

3 Analysis of Facial Animation Parameters from Video Sequences

While the MPEG-4 standard does not specify how to generate FAPs, we need a method which determines FAPs from video sequences in order to retrieve embedded watermarks from rendered sequences. A suitable method has been presented in [4, 5] and was used to obtain the FAPs from rendered sequences in the section on experimental results.

The approach for the estimation of the facial parameters is model-based and combines a motion model of an explicit 3D textured wireframe with the optical flow constraint from the video data. This leads to a linear algorithm that is robustly solved in a hierarchical framework with low computational complexity.

The inputs to the estimation method are the video sequence and the 3D model of the head shown in the sequence, and the output are the estimated MPEG-4 animation parameters. The method estimates the FAPs with very satisfying accuracy. For details, please refer to [4, 5].

4 Watermarking of MPEG-4 Facial Animation Parameters

For embedding of watermark data into the FAPs, we adopt a spread spectrum approach [6] that has been applied similarly to image and video watermarking before [3, 7, 8]. The idea is to apply small changes to the FAPs that seem random and are not conspicuous but correlate with a secret pseudo-random key. The correlation can be exploited to retrieve the embedded information later on. The watermark information to be embedded may be any arbitrary 1D or 2D bit pattern. Here we regard the watermark as a 2D bit pattern

$$WM(m, n) \in \{-1, +1\} \quad (1)$$

with $m \in 0 \ldots m_{max}$ and $n \in 0 \ldots n_{max}$.

4.1 Watermark Embedding

One basic concept of watermarking is to distribute one bit of watermark information over more than one FAP. For ease of understanding, we embed one bit of watermark information into a $M \times N$ block of FAPs, that means into M consecutive FAPs over N consecutive time instants, but we could also distribute one bit of watermark information over $M \times N$ randomly chosen FAPs from the whole set of FAPs, $FAP_k(t)$.

Before embedding, the stretched bit pattern is modulated, i.e. multiplied, with a filtered pseudo noise pattern $PN(k, t)$. This pseudo noise pattern is the secret key used for embedding, and the modulation makes the modulation product difficult to detect and manipulate. The pseudo-noise pattern can be generated by any mean-free random number generator that produces binary output values -1 and $+1$. For a discussion about the choice of a good pseudo random generator, please refer to [3]. We use a filtered pseudo noise pattern that is lowpass filtered along t, e.g. with a 9-tap lowpass filter, in order to decrease the noisiness in the watermarked FAPs which could result in annoying noisy behaviour of the head model. Before finally adding the modulated bit pattern to the FAPs, we apply an amplitude adaptation in order to limit visible distortions. We found it a good choice to limit the maximum deviation of the watermarked FAPs from the unwatermarked FAPs to 3 % of the dynamic range for local FAPs like lip movement, and 1 % of the dynamic range for global FAPs like head rotation. We denote this amplitude adaptation by a factor of $\beta(k)$ which is an amplitude factor for FAP_k. Thus, the watermarked FAPs are

$$FAP_k^{WM}(t) = FAP_k(t) + [WM(\lfloor k/M \rfloor, \lfloor t/N \rfloor) \cdot PN(k, t) \cdot \beta(k)] \quad (2)$$

where $\lfloor \ldots \rfloor$ is the floor operator.

Since the FAPs are a highly compressed representation of human face expressions, and since their data volume is very low, the signal space which can be used to embed information is also very limited. This results in the fact that the information embedded into the FAPs is not robust enough to resist all possible kinds of attacks, unless $M \times N$ is increased very much, that is, the data rate for the watermark information is decreased very much. However, already for choices of M

and N like the ones that were used for the experiments in Section 5, namely $M = 2$ and $N = 76$, there is a certain robustness, and manipulation of single FAPs or DCT-based video compression of sequences rendered from the FAPs are not sufficient to impair the watermark.

The method according to (2) can easily be applied to streaming or fixed-length FAPs sequences in order to embed information in real-time, like the ID of a downloading client for download on the WWW. The data rate r_{WM} of the watermark (the number of bits per second) depends on the number k_{max} of transmitted FAPs, frame rate R (number of FAPs sets per second), and the number $M \times N$ of FAPs that one bit of watermark information is embedded into and is $r_{WM} = \frac{k_{max} \times R}{M \times N}$.

4.2 Watermark Retrieval from Watermarked Parameters

Retrieval of the embedded watermark from the watermarked FAPs is done by subtraction of the unwatermarked FAPs from the watermarked FAPs, subsequent correlation with the same filtered PN sequence that has been used for embedding (and which is the secret key), summation over the window for each watermark bit, and thresholding as a bit decision. Put into an equation, the reconstructed watermark $\overline{WM}$ is

$$\overline{WM}(m,n) = sign\left[\sum_k \sum_t \left(FAP_k^{WM}(t) - FAP_k(t)\right) \cdot PN(k,t)\right] \quad (3)$$

4.3 Watermark Retrieval from Rendered Video Sequences

If the watermarked FAPs have been used to animate a head model and render a video sequence, and if the video sequence is available but the FAPs are not, the watermark cannot directly be retrieved. It is however possible to estimate the FAPs from the rendered sequence, for example with the method presented in [4, 5], and to use the estimated FAPs to retrieve the watermark. Of course this is only possible if the video sequence is rendered such that the FAPs can be estimated. For example, mouth and eye of the model should be visible. The needed steps are estimation of the FAPs from the rendered sequence, and subsequent retrieval of the watermark from the estimated FAPs as described in Section 4.2.

5 Experimental Results

In an experimental set-up, we have embedded a binary information of 3 bytes into two different FAP streams of size $k_{max} = 17, t = 228$ (since the frame rate is 25 frames per second, these FAP streams represent sequences of $\frac{228}{25} = 9.12$ seconds each). Thus, $M = 2$ and $N = 76$, and each bit of watermark information has been embedded into 2 parameters over 76 frames, i.e., into 152 FAPs. The first bit of watermark information has thus been embedded into FAPs 1 and 2 over the first 76 frames, the second bit into FAPs 3 and 4 over the first 76 frames; and the last bit into FAPs 15 and 16 over frames 153 to 228. The data rate r_{WM} of the watermark is 2.8 bit/s (the frame rate is 25 Hz). Please note that only a subset of 17 out of the 66 admissible FAPs has been used here. If all possible FAPs are transmitted, more information can be embedded.

In the first experiment, we tried to retrieve the watermark information directly from the watermarked FAPs according to (3). It should be noted that in this case the watermarked FAPs were known, and the FAPs did not have to be estimated, as for the following experiments. For both used FAP sequences, it was possible to retrieve the watermarks without errors, as one would expect.

In the second experiment, the two watermarked FAP streams have been used to render two different animated head models into video sequences. The background was static in both cases, and the used resolution was 352×288 pixels. Figure 1 shows representative frames from the two sequences. The FAPs

Figure 1: Representative frames from the two sequences rendered from animated 3D head models.

were then estimated from the video sequences as described [4, 5], and the watermarks were retrieved from the estimated watermarked FAPs. For both used FAP sequences, it was possible to retrieve the watermarks without errors.

In the third experiment, we have used the rendered sequences of the second experiment and have compressed/decompressed them using MPEG-2 video compression. The bit-rate was chosen to 250 kbit/s for the first sequence and 600 kbit/s for the second sequence. Figure 2 shows representative frames from the two compressed sequences. The FAPs were then estimated from the compressed/decompressed video sequences as described in [4, 5], and the watermarks were retrieved from the estimated watermarked FAPs. For both used FAP sequences, it was possible to retrieve the watermarks without errors for the chosen parameters. This successful retrieval from the MPEG-2 compressed sequences demonstrates that there is inherent ro-

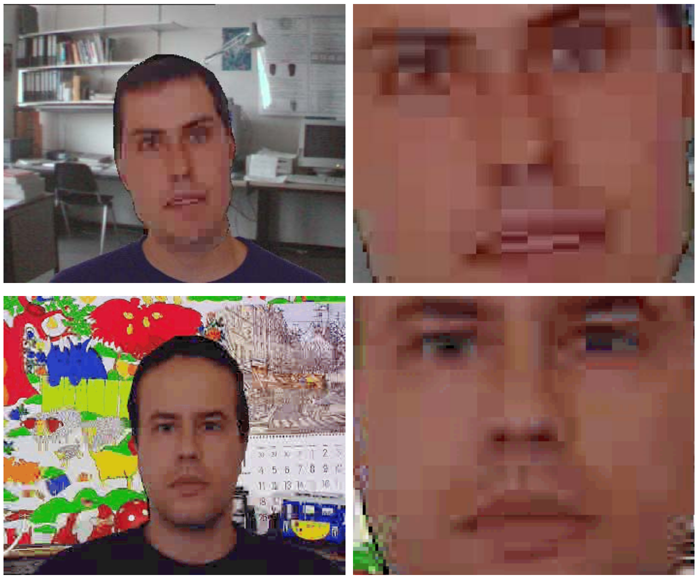

Figure 2: Example frames from the rendered and subsequently MPEG-2 encoded video sequences. The details demonstrate the compression artefacts.

bustness in the watermark. We noticed however that, depending on the bit-rate of video compression and the data-rate of the embedded watermark, bit errors may occur. This is shown in Fig. 3 where we have repeated the third experiment for the first FAP stream and for different choices of M and N (different tilings of the FAPs-time-plane) and, thus, for different watermark data rates r_{WM}. Shown here is the bit error rate of the watermark information vs. the data rate of the watermark information. Though there is clearly a tendency to lower bit error rates of the watermark information for low data rates, sometimes bit error rates up to 0.2 may still occur, due to estimation errors in the FAPs estimation. Thus, it is advisable to protect the watermark information with an error correcting code.

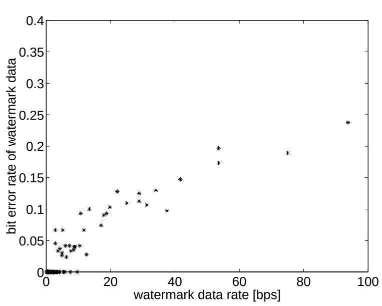

Figure 3: Bit error rates of watermarks retrieved from a rendered and MPEG-2 encoded video sequence for different choices of the watermark data-rate.

6 Conclusions

We have presented a spread spectrum watermarking method which allows to embed information such as copyright information into 3D head model animation parameters as they are used in MPEG-4 face animation. This allows to enforce intellectual property rights (IPR) for animation parameters in a similar way as it has previously been proposed for images, video [2], and 3D polygonal models [11]. The watermark is embedded into the FAPs and can be retrieved from the watermarked FAPs or from a video sequence rendered with the watermarked FAPs. For the latter case, the parameters have to be estimated from the rendered sequence. A suitable estimation method has been described in [4, 5]. We have shown results confirming the applicability of the scheme. Applications are for example marking and protection of MPEG-4 FAP data available on the internet or broadcasted in video delivery networks.

References

[1] ISO/IEC 14496-2, Coding of Audio-Visual Objects: Visual (MPEG-4 video), Committee Draft, October 1997.

[2] H. Berghel and L. O'Gorman. Protecting ownership rights through digital watermarking. *IEEE Computer*, pages 101–103, July 1996.

[3] I. Cox, J. Kilian, T. Leighton, and T. Shamoon. Secure spread spectrum watermarking for multimedia. Technical Report 95-10, NEC Research Institute, Princeton, NJ, USA, 1995.

[4] P. Eisert and B. Girod. Facial expression analysis for model-based coding of video sequences. *Picture Coding Symposium*, pages 33–38, Sep. 1997.

[5] P. Eisert and B. Girod. Model-based estimation of facial expression parameters from image sequences. *International Conference on Image Processing*, 2:418–421, Oct. 1997.

[6] Paul G. Flikkema. Spread-spectrum techniques for wireless communications. *IEEE Signal Processing*, 14(3):26–36, May 1997.

[7] F. Hartung and B. Girod. Digital watermarking of raw and compressed video. In N. Ohta, editor, *Digital Compression Technologies and Systems for Video Communications*, volume 2952 of *SPIE Proceedings Series*, pages 205–213, October 1996.

[8] F. Hartung and B. Girod. Digital watermarking of uncompressed and compressed video. *Signal Processing, Special issue on copyright protection and access control for multimedia services*, 66(3), 1998.

[9] E. Koch and J. Zhao. Towards robust and hidden image copyright labeling. In *Proceedings of 1995 IEEE Workshop on Nonlinear Signal and Image processing*, Neos Marmaras, Greece, June 1995.

[10] M. Kutter, F. Jordan, and F. Bossen. Digital signature of color images using amplitude modulation. In *Proceedings of Electronic Imaging 1997 (EI 97), San Jose, USA*, February 1997.

[11] R. Ohbuchi, H. Masuda, and M. Aono. Embedding data in three-dimensional polygonal models. In *Proceedings ACM Multimedia '97, Seattle, USA*, November 1997.