# A digital watermarking application: Watermarking low bit-rate MPEG video[*]

**Iwan Setyawan**

***Abstract***

This paper discusses the challenges of embedding digital watermark into MPEG video encoded in low bit-rate. The way MPEG compression works, namey by discarding visually irrelevant portions of the video, presents challenges for the digital watermark system designer particularly in fulfilling the capacity and imperceptibility requirements. This paper presents a digital watermarking scheme suitable for MPEG video encoded at low bit rates.

***Keywords***

Digital video watermarking, multimedia security, MPEG compression

## 1. Introduction

Digital video data distribution through the internet is becoming more common [1]. The rapid growth of the internet and the increasing number of internet users make it a very strong marketing medium with which to reach potential customers for various products. When Hollywood studios release new movies, it is now common for them to set up an official website for the movies in which they put multimedia materials, such as the movie trailers, interviews with the cast, etc. Most recording artists nowadays have their own official websites, where they can also put their video clips to promote their new albums. The same goes for big music publishing companies, because such promotion can also boost the sales of the albums of the artists under their label. The interactive entertainment industry, i.e. video and computer games industry, also sees the internet as a medium not only to distribute demos or preview versions of their games for potential customers to download, but also as a medium to distribute video materials of their games, such as in-game video sequences, the opening cinematics of their games or dedicated “game trailers” in which they show off the exciting parts of their game in a similar manner as that used in movie trailers. All these marketing efforts, especially for the last case, may make or break the sales of the products.

These multimedia materials share an important feature, namely they must be compressed at low bit rates to facilitate distribution through the internet. Furthermore, these materials need to be protected in order to prevent copyright infringement issues. Digital watermarking is one of the possible solutions for this copyright protection problem [2,5,7]. However, most of the existing video watermarking algorithms are more geared towards high bit rate environments suitable for DVD or television broadcasting [3,6]. Low bit rate (below 1 kbps) video watermarking utilizing MPEG-4 facial animation parameters has been investigated [4], and is suitable for video telephony application. However, low bit-rate watermarking for other applications, such as the one mentioned in the previous paragraph, has received little attention in the literature.

Low bit-rate environments present new challenges to the watermarking operation which are not found in watermarking operations at high bit-rate environments. Video encoded at low bit-rates inherently possesses low redundancy and small visual degradation tolerance. This brings forward three important issues. The first issue is the watermark capacity, i.e. the number of watermark bits we can embed into the data. The second issue is the visual impact.

---

[*] This paper is an expanded version of the work published *Low bit-rate video watermarking using temporally extended Differential Energy Watermarking (DEW) algorithm* by I. Setyawan & R.L. Lagendijk, in the Proceedings of SPIE, Security and Watermarking of Multimedia Contents III, Vol. 4314, pp. 73-84, San Jose, CA, 2001

The low visual degradation tolerance of the original video sequence/stream means that we must take special care before embedding the watermark, which essentially adds more distortion into the data. On the other hand, the coding artefacts are also more visible in streams encoded at low bitrates. Therefore, it is possible for the watermarking artefact to be dominated by the coding artefact. The third issue is the robustness of the watermark. These three issues are closely interrelated and adjusting one of these performance aspects will affect the performance of the others.

In our previous work, we developed a video watermarking scheme suitable for MPEG-1/-2 video streams encoded at high bit-rates (1.4 to 8 Mbps), called the Differential Energy Watermarking (DEW) algorithm [7,9]. This method has been shown to have relatively low complexity, high capacity and low visual impact. We consider this technique to have the potential to be extended for use in low bit-rate environments, and in this paper we present the extension scheme and an evaluation of its performance in order to investigate the behavior of this technique in low bit rate environments.

This paper is organized as follows. In Section 2, the DEW algorithm is briefly described and the extension scheme is explained in detail. In Section 3, the experiment setup and results are presented. In Section 4 we present the conclusions of our experiments. Finally, in Section 5 we present brief discussion on recent developments in low bit-rate video watermarking techniques

## 2. The Extended DEW algorithm

### 2.1. The DEW algorithm

The DEW algorithm embeds watermark bits into an MPEG stream (or any other block DCT based video compression system) by enforcing energy difference between certain groups of $8 \times 8$ DCT blocks of the I-frames to represent either a ‘1’ or a ‘0’ watermark bit. The energy difference is enforced by selectively removing high frequency components from the DCT blocks. The $8 \times 8$ DCT blocks of an I-frame are first randomly shuffled using a secret seed. This process serves two purposes. In the first place, the seed serves as a secret key without which one cannot extract the watermark properly. In the second place, the process is performed to avoid having a group of blocks in which there is an unbalanced energy content. If this happens, then the watermarking artefact may become visible. As mentioned above, the energy difference is enforced by removing high frequency DCT components. If too many high-frequency components are removed in order to enforce this difference, then the watermarking artefacts, in the form of blurred edges, will be visible. This may happen, for example, when one group of blocks has no high frequency component (i.e., contains only flat areas) while the other group of blocks contains edges. If the energy content of the second group of blocks has to be reduced to enforce the energy difference, then too many high frequency components would have to be removed and the edges will be blurred as a consequence. The fundamental terms of the DEW algorithm are illustrated in Figure 1.

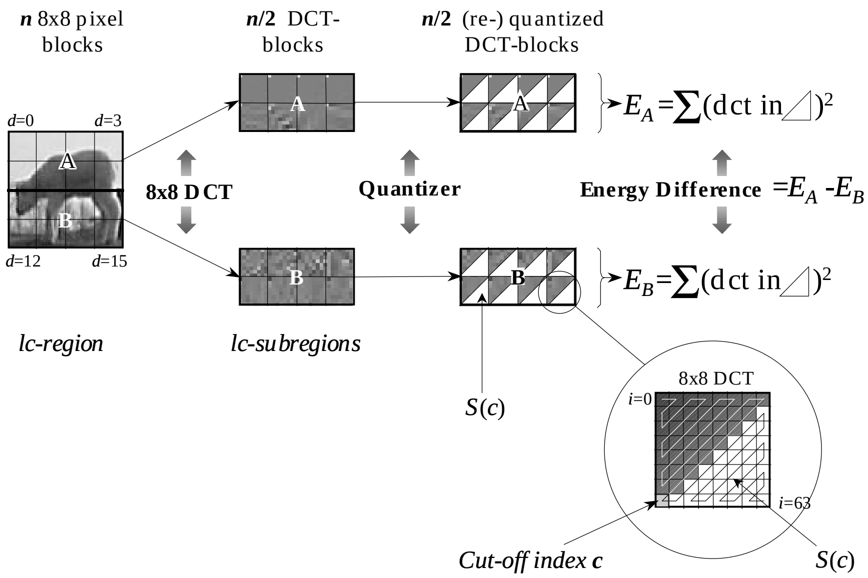



*Figure 1. The Differential Energy Watermarking*

The DEW algorithm has several adjustable parameters. By adjusting these parameters, we can adjust the watermarker to optimise it either for capacity, robustness or visual impact [9]. The parameters are as follows:

- ***Number of 8 ´ 8 blocks per watermark bit***: This parameter is represented by *n* in Figure 2.1. It influences the capacity and the robustness of the watermark. The more blocks are used to embed a single watermark bit, the less capacity is achieved, but the more robust the watermark would be and the less degradation would be introduced because the required energy difference is "spread" among the blocks, and the more blocks we use the less energy in the region *S(c)* has to be removed from each DCT block.
- ***Enforced energy difference*:** This is the minimum allowed value of $E_A$-$E_B$ in Figure 1. This parameter influences the robustness and visual impact of the watermark. The larger the energy difference enforced, the more robust the watermark would be. The disadvantage could be worse visual quality because more DCT coefficients have to be discarded. Furthermore, due to the limitation imposed by the next parameter, it is possible that some watermark bits cannot be correctly embedded.
- ***Minimal cut-off point*:** This parameter is represented by *c* in Figure 1, and denotes the index of the particular DCT coefficient number in an $8 \times 8$ block (zigzag scanned). Any coefficient with index $i < c$ may not be removed to enforce the energy difference. Thus it can be seen as a limiter to the previous parameter because this parameter determines how many DCT coefficients may be removed to enforce the energy difference. If this parameter is set too high, then the watermark robustness would suffer and there is a possibility that some watermark bits cannot be properly embedded because the proper energy difference could not be enforced. However, the visual quality degradation introduced by the watermarking would be lower than the degradation introduced when a lower minimal cut-off is set because fewer DCT coefficients are removed.

The DEW algorithm also has several interesting properties. It is relatively uncomplicated because it embeds the watermark at the DCT coefficient level and thus only VLC decoding is needed for the watermark embedding and detection process, and no full decoding and re-encoding of the stream is needed. This scheme also has sufficient robustness because a full decoding and re-encoding is needed to completely remove the watermark from the stream. It has been shown that even transcoding a watermarked 8 Mbps MPEG stream down to 6 Mbps only introduce a 7% Bit Error Rate (BER) [7]. The capacity of the watermarking scheme is also sufficiently high, up to 0.42 kbps for a stream encoded at 8 Mbps. The visual impact of the watermarking process is also negligible.

### 2.2. Extending the DEW algorithm

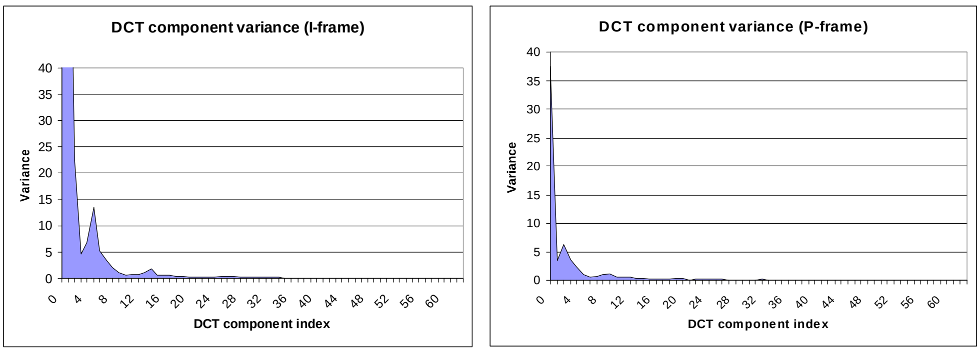
The primary motivation of extending the DEW algorithm is to “spread” the watermark bits more in the temporal dimension. Spreading the watermark data in the temporal dimension offers potential improvements to the original DEW algorithm, especially for implementation in low bit-rate environments. The potential improvements are discussed below:

- ***Improved capacity:*** One of the main issues when we move to a lower bit-rate environment is the watermark capacity. The spatial resolution of the frames plays a very important role for the watermark capacity of the DEW algorithm. Videos in the application scenarios mentioned in the introduction are usually of CIF or lower spatial resolution, while the video used in the applications for which the DEW algorithm was originally designed usually has as much as four times the spatial resolution (4CIF). Thus, one of the main problems here is to find a way to compensate this capacity limitation imposed by the video resolution. One possible solution to this problem is to reduce the number of $8 \times 8$ DCT blocks that are used to embed each watermark bit. However, reducing the number of blocks also reduces the robustness of the embedded watermark. The other solution we investigate here is to use more frames to embed the watermark, thus spreading the watermark in the temporal dimension. To achieve this, we will use not only the I-frames of the stream to embed the watermark, but also the P-frames. The challenges here are:
    - The much lower energy content of the P-frames compared to the I-frames means that this solution will require a more delicate approach in order to balance the capacity, robustness and visual impact requirements.
    - The drift effects [3]. This is the result of error accumulation because the watermarked frames are used to reconstruct the frames and they are also being used as a prediction reference for other frames. Over time, this error accumulation may become visible. Even worse, the error may spatially spread.
- ***Improved robustness:*** The next issue concerns the robustness of the watermark. In this respect, the extension offers two possible advantages. The first advantage is derived directly by the extra space we can use when we use the temporal dimension. By using this extra space, we can embed fewer bits in each frame (thus increasing the watermark robustness) but still achieve the same watermark payload. Another possible advantage is that if an attacker wants to remove the watermark in a video stream watermarked using the original DEW algorithm, he has only to deal with the I-frames, which are relatively few in a sequence. If the algorithm is extended so that the watermark data does not reside only in the I-frames, then the attacker would have to deal with more frames. This does not eliminate the possibility of an attacker successfully removing the watermark, but this will make the attack more cumbersome.
- ***Improved visual quality:*** The next possible improvement concerns the visual quality of the watermarked stream. This issue is related to the previous issues, because if we embed fewer watermark bits per frame then the degradation to the data is reduced. The extra space provided by the extension would compensate the decrease of watermark payload per frame.

The extension to the original DEW algorithm is achieved by modifying the watermark embedder so that it embeds the watermark not only in the I-frames, but also the P-frames. As noted above, the energy content of an I-frame and a P-frame is very much different. In an I-frame, the whole image data from that frame is intra-encoded. A P-frame, on the other hand, is predicted from a previous I-frame. Only the *prediction error* is encoded into a P-frame. This prediction error carries much less energy than an intra-coded frame. As an example, one I-frame of the Claire MPEG stream (CIF resolution, encoded at 700 kbps) has a total signal variance of 2057.7, while a P-frame predicted from this I-frame has a total signal variance of only 59.8. This value varies widely from P-frame to P-frame, depending on the amount of activity (movements) in the sequence. When there are a lot of movements in the sequence, the P-frame will contain more energy than when the amount of movements is small. For this particular sequence, the average variance of the P-frames is 40.223. The variance of the I-frames also varies from I-frame to I-frame, but the variation is less significant. The average variance of the I-frames in this particular sequence is 2034.6. In Figure 2, we show the variance of the DCT components of one I- and one P-frame predicted from the I-frame.

Figure 2(a) is clipped at variance values of 40 in order to show the variance of the higher DCT coefficients, and also to allow a somewhat easier comparison between the two figures. Other than the obvious difference in scale, the two figures show very similar behavior. Although the variance of each individual P-frame is different, depending on the level of activity found in the sequence, the behavior is similar to that observed in Figure 2(b). This behavior suggests that, with proper parameter adjustments, the DEW algorithm can be applied directly to the P-frames. The parameters that need to be adjusted are either the ***enforced energy difference***, which should be lower due to the lower energy content of the P-frame, the ***minimal cut-off point***, which should also be lower due to how the energy is distributed in the frame, or both.

On the average, the variance of the I-frames is larger by a factor of approximately 50 compared to the variance of the P-frames. Therefore, we will need approximately 50 times as many $8 \times 8$ DCT blocks to embed one watermark bit in one P-frame to be able to achieve the same level of performance as when we embed the watermark in an I-frame, all other parameters being equal. If we can properly embed watermarks into an I-frame using 32 DCT blocks per watermark bit, we would need approximately 1600 DCT blocks in a P-frame. Since a sequence with CIF resolution only has 1584 DCT blocks per frame, this means that there is a maximum of 1 watermark bit that can be reliably embedded in a P-frame on average. Therefore, instead of only increasing the number of blocks, we also reduce the level of energy difference that has to be enforced. In this way, we should not need 50 times as many blocks to be able to properly embed the watermark bits, which means that we should be able to embed more than one watermark bit per P-frame. The price we have to pay here is the lower robustness of the watermark.




*(a)* *(b)*

*Figure 2. DCT component variance of (a) an I-frame and (b) a P-frame taken from the Claire sequence, MPEG encoded at 700 kbps*

It is also possible to extend the algorithm by embedding the watermark into the B-frames. Most of the frames in an MPEG sequence are B-frames. This large number of frames offers even more capacity increase than if we extend the algorithm to use the P-frames. Furthermore, from a robustness point of view, this will make the attack even more cumbersome. The B-frames are not used to predict other frames, which means we do not have to deal with drift effects. However, the B-frames contain even less energy than the P-frames. We have discussed above that even the P-frames may not contain enough energy to properly accommodate the watermark, and thus we choose not to extend the algorithm to embed the watermark in the B-frames.

The watermark bits are embedded in the P-frames in the same manner as they are embedded in the I-frames. First, the $8 \times 8$ DCT blocks of the P-frames are shuffled pseudo-randomly using a random key. The same key is used for both the I-frames and the P-frames in our experiments. However, technically there is no problem if a different key (independent of the key used to shuffle the I-frames blocks) is used. After the blocks are shuffled, the DEW algorithm is performed to embed the watermark bits. Each watermark bit is embedded into a certain number of $8 \times 8$ DCT blocks. From the watermarking point of view, the operations on the I-frame and the P-frame are independent, i.e. the watermark in the I-frame can be detected independently from the watermark in the P-frame (and vice versa) and the BER of the watermark embedded in the I-frame is not affected by the BER of the watermark embedded in the P-frame (and vice versa).

The BER of the watermark embedded in either the I- or P-frames can be introduced either due to:

- Insufficient energy content in the I- or P-frames which means that certain energy differences cannot be enforced. This happens during embedding, and we will call this the ***e***BER.
- Distortion of the watermarked frame due to attacks, for example due to re-encoding. We will call this the ***a***BER.

Both BERs are calculated as follows:

$$BER = \frac{bit\,errors}{total\,embedded\,bits} \times 100\% \tag{1}$$

In Equation (1), the *total embedded bits* refers to the total amount of bits the watermarker software attempts to embed in the sequence. Due to the parameter settings chosen, it may not be able to properly embed some of these bits.

Furthermore, the watermark embedded in the P-frames also has the same adjustable parameters as the watermark embedded in the I-frames. These parameters are adjusted independently from the parameters of the I-frame watermark. This allows us to use either identical settings or different settings that are more appropriate for the P-frames due to their different characteristics.

## 3. Experiment setup and results

### 3.1. Experiment setup

We test the extended DEW algorithm to see its performance in terms of watermark capacity, watermark robustness and visual quality impact. We use MPEG-2 sequences encoded at 256, 384, 512 and 700 kbps with a frame rate of 25 fps. All sequences are encoded using the same *group of pictures* (GOP) structure. We use the commonly used GOP structure, i.e., IBBPBBPBBPBB. The sequences are encoded as a *progressive* sequence and therefore the use of MPEG-1 coded sequences would also have been possible. The extended DEW algorithm itself is compatible with MPEG-1 stream. The spatial resolution of the sequences is $352 \times 288$ pixels (CIF). The sequences used in our experiments are: “Claire” (78 frames), “Trevor” (150 frames) and “Akiyo” (250 frames). The sequences are watermarked using the new version of our DEW watermarker software that can operate in both “default” and “extended” mode. In default mode, the software essentially operates identically to the original watermarking software. In extended mode, the watermarker embeds the watermark using the extended DEW algorithm.

Two parameters are fixed during the experiments, i.e. the ***enforced energy difference*** and the ***minimal cut-off point***. For the I-frames (both in “default” and in “extended” modes) the ***enforced energy difference*** is set at 20 and the ***minimal cut-off point*** is set at 6. For the P-frames, the ***enforced energy difference*** is set at 4. This much lower value is chosen due to the lower energy content of the P-frames compared to the I-frames. We have pointed out in Section 2.2.1 that this parameter (for a fixed number of DCT blocks per watermark bit and a certain ***minimal cut-off point***) influences the probability that a watermark bit can be properly embedded. If this parameter is set at the same value as the one used for the I-frames, there is a high possibility that the watermark bits cannot be properly embedded because the P-frames contain much lower energy. Furthermore, as discussed in Section 2.2.2, choosing lower value for this parameter will allow us to use fewer DCT blocks per watermark bit and thus enable us to embed more watermark bits in the P-frame. Since the average variance of the P-frames in our test sequences is lower by a factor of approximately 50, by choosing an ***enforced energy difference*** of 4 (which is 20% of the value chosen for the I-frames), we expect that we would need as few as around ten times as many DCT blocks to embed one watermark bit in a P-frame, with similar performance as embedding the watermark in an I-frame, instead of 50. The ***minimal cut-off point*** is set at 6 for both cases, in order to avoid too much image degradation due to the watermarking process.

### 3.2. Watermark capacity

As noted in Section 2.2, the watermark capacity is determined by the number of $8 \times 8$ DCT blocks that are used to embed one watermark bit. The number of watermark bits that can be embedded in one frame can be computed as follows:

$$W_b = \left\lfloor \frac{F_p}{B \times 64} \right\rfloor \text{ bits} \qquad (2)$$

In the equation above, $W_b$ is the number of watermark bits in the frame, $F_p$ is the total number of pixels in the frame and $B$ is the number of $8 \times 8$ DCT blocks per watermark bit. The watermark bit-rate is computed simply as the total number of embedded watermark bits divided by the length of the sequence in seconds.

When 4CIF-sized sequences and the default parameter (64 blocks per bit) are used, a label rate of 0.21 kbps is achieved [7]. By reducing this number to 32 blocks per bit, a capacity of 0.42 kbps is achieved. The sequences used in our experiment are in CIF format, which is a quarter as large as the 4CIF sequence. Therefore, using the default parameter a label rate of 50 bps can be achieved. We call this watermark bit-rate the *base capacity*. In the following sub-sections, we investigate the performance of the original and the extended DEW algorithms in two areas, namely ***below*** and ***above*** this base capacity.

#### 3.2.1. Performance below base capacity

In order to investigate the behavior of both algorithms below base capacity, we use various settings yielding watermark bit-rates ranging from 2 bps to 50 bps. For the I-frames, we use various numbers of DCT blocks per watermark bit, ranging from 128 to 1024 blocks. For the P-frames, we use 512 or 1024 DCT blocks per watermark bit. The settings we use, the achieved watermark bit-rates and the achieved *e*BER are presented in Table 1.

*Table 1. Settings, Watermark Bit-rate and eBER of the original and extended DEW algorithm below base capacity (Claire, 256 kbps)*

| DCT blocks/water mark bit (I-frame) | P-frame settings | | | | | |
|---|---|---|---|---|---|---|
| | No watermark bits in P-frames (DEW) | | 1024 DCT blocks/ watermark bit (XDEW) | | 512 DCT blocks/ watermark bit (XDEW) | |
| | Watermark bitrate (bps) | *e*BER (%) | Watermark bitrate (bps) | *e*BER (%) | Watermark bitrate (bps) | *e*BER (%) |
| 1024 | 2.2 | 0 | 8.6 | 0 | 21.5 | 0 |
| 512 | 6.7 | 0 | 13.1 | 0 | 26 | 0 |
| 256 | 13.5 | 0 | 19.9 | 0 | 32.7 | 0 |
| 128 | 27 | 0 | 33.3 | 0 | 46.2 | 0 |

Table 1 only shows the *e*BER of a sequence encoded at 256 kbps, but the results for sequences encoded at other bit-rates are identical. We can observe from the results that both the original and the extended DEW algorithm performs well for watermark bit-rates below the base capacity. Furthermore, the results show that the P-frames contain enough energy to accommodate the watermark if we use 512 or 1024 DCT blocks to embed one watermark bit.

### 3.2.2. Performance above base capacity

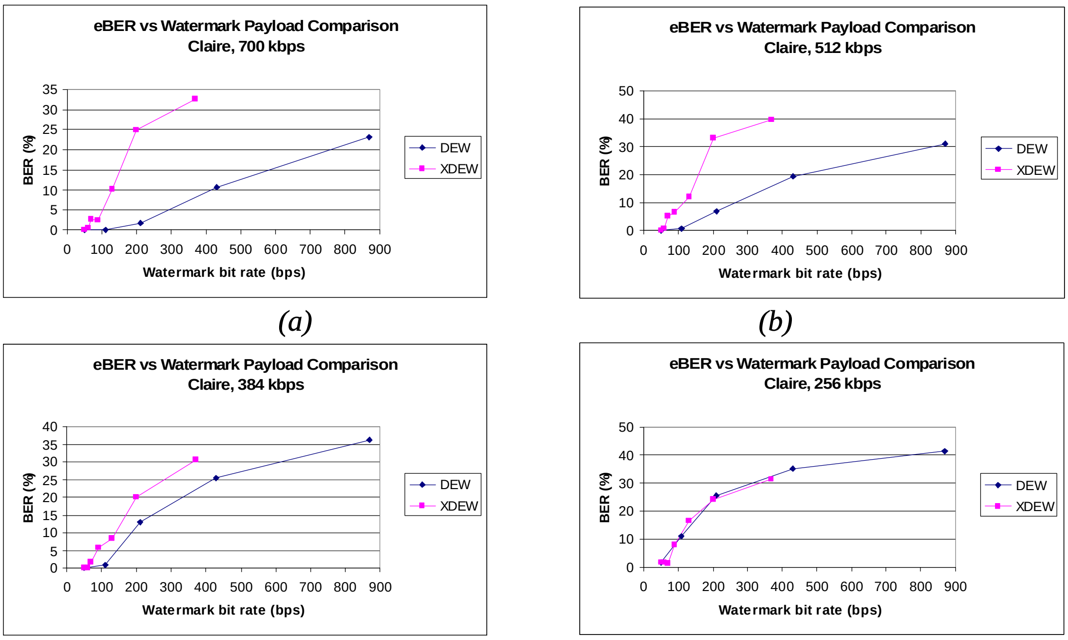
The base capacity of 50 bps may not be sufficient for all applications, as some applications may require that the watermark bit-rate is at least 70 bps [8]. We compare two approaches to increase the capacity. The first approach is to use the original DEW algorithm but reduce the number of blocks used to encode each watermark bit and the second approach uses the extended DEW algorithm (with various numbers of blocks to encode each watermark bit in the P-frame) using the default parameter for the I-frames (64 blocks/bit). In the first approach, we can achieve various bit-rates of 50 to 870 bps. In the second approach, we achieve bit-rates of 50 to 370 bps. The relation of the number of blocks per watermark bit and the achieved watermark bit-rate is presented in Table 2.

*Table 2.*
*Relation between number of blocks/watermark bit and watermark bit-rate*

| Original DEW | | Extended DEW (64 DCT blocks/I-frame) | |
|---|---|---|---|
| DCT Blocks/bit (I-frame) | Watermark Bitrate (bps) | DCT Blocks/bit (P-frame) | Watermark Bitrate (bps) |
| 64 | 50 | 1024 | 60 |
| 32 | 110 | 512 | 70 |
| 16 | 210 | 256 | 90 |
| 8 | 430 | 128 | 130 |
| 4 | 870 | 64 | 200 |
| | | 32 | 370 |

We compare the two approaches by evaluating the watermark *e*BER produced by each approach as we increase the number of bits embedded in the streams. The results of this experiment are presented in Figure 3.

From Figures 3(a) to 3(d), we can see that for all encoded bit-rates, except 256 kbps, it is much more attractive to use the original DEW algorithm and reduce the number of blocks used to encode each watermark bit rather than using the P-frames to gain extra space to embed more watermark bits. For a 256 kbps encoded bit-rate, both approaches seem to yield the same performance. This is because the I-frames in such a stream have already lost much of the high-frequency DCT components due to the nature of MPEG quantization. As an example, a comparison of the variance of the DCT coefficients between two matching I-frames, one taken from a sequence encoded at 700 kbps and the other from a sequence encoded at 256 kbps, is shown in Figure 4.




*(c)* *(d)*

*Figure 3. Watermark eBER for sequences encoded at various bit rates, watermarked using the DEW algorithm and the Extended DEW (XDEW) algorithm with various payloads*

Figure 4 shows only the variances of the DCT coefficients with indices 6 up to 63, because the coefficients with lower index (indices lower than 5) have relatively similar variance and also these coefficients do not play an important role in the enforcement of the energy difference. As we can see, the energy content of the I-frame of the sequence encoded at 256 kbps is lower than the one of the sequence encoded at 700 kbps. As the discussion in Section 2.2 points out, this means that in order to enforce the same energy difference with the same number of DCT blocks per watermark bit, a lower *minimal cut-off point* should be chosen. And since in our experiments this parameter is fixed, there are some watermark bits that cannot be properly embedded. It should also be noted that the variances plotted in Figure 2.4 are from the collection of blocks and the condition of individual DCT blocks might be worse (i.e., there is less energy available to enforce the energy difference).

From Figure 3, we can also see that for the original DEW algorithm relatively low *e*BER (2.5% bit error or less, which is roughly equal to a BER of $10^{-3}$) is achieved only for watermark payloads below 110 bps, which means that the number of blocks used to embed each watermark bit is halved (from 64 blocks/watermark bit to 32 blocks/watermark bit). For a sequence encoded at 700 kbps, this number can still be achieved at a payload of 210 bps. Meanwhile, for the sequence encoded at a bit-rate of 256 kbps, even a watermark payload of 110 bps produces more than 10% *e*BER. For the Extended DEW algorithm, the numbers are even lower. For the sequence encoded at 700 kbps, a payload of up to 90 bps can be achieved, while for other bit-rates, this number drops to around 70 bps.

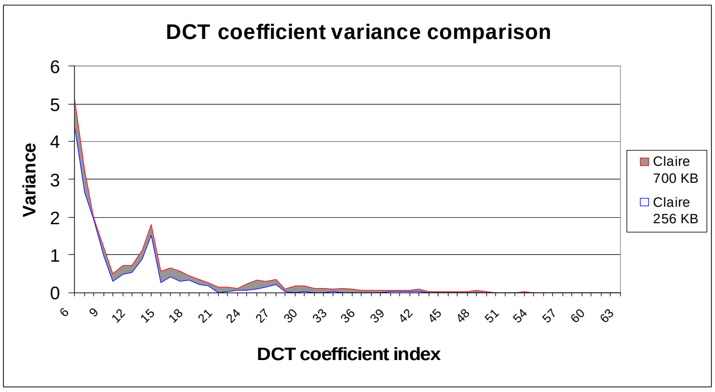



*Figure 4. Comparison of the variance of theDCT coefficients from 2 I-frames Taken from sequences encoded at different bit-rates*

These numbers show that for the I-frames a proper energy difference can no longer be enforced when less than 32 DCT blocks are used to encode one watermark bit (16 DCT blocks, in the case of sequences encoded at 700 kbps), while for the P-frames the minimum number of DCT blocks that should be used to embed one watermark bit is 512 blocks (256 DCT blocks for sequences encoded at 700 kbps).

### 3.3. Watermark robustness

We test the watermark robustness by re-encoding the watermarked stream at a lower bit-rate and then seeing whether the watermark survives the operation. Watermark survival is measured by the *a*BER of the watermark. The reencoding operation we performed is illustrated in Figure 5.

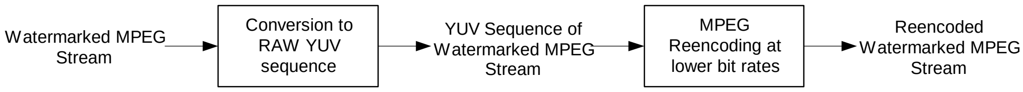
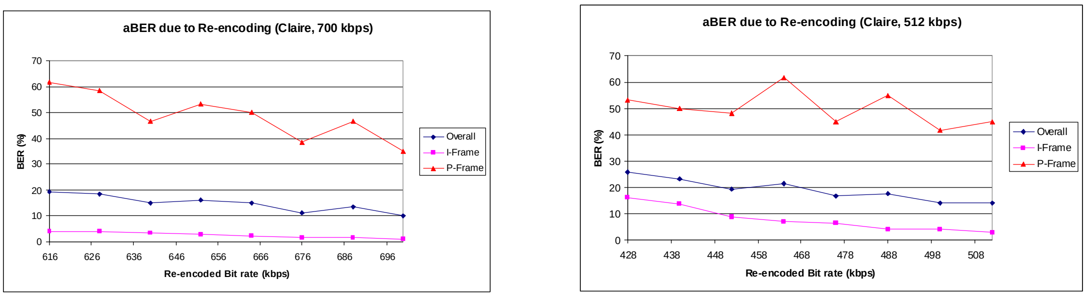



*Figure 5. Re-encoding procedure*

The *a*BER is computed from a re-encoded sequence previously watermarked using the Extended DEW algorithm, and is presented in Figure 6 for the I-frames and the P-frames separately. Since the watermarks embedded in the I-frames and the P-frames are detected independently, the watermark *a*BER for the I-frames can be interpreted as the watermark *a*BER of the original DEW algorithm after re-encoding. The behavior observed in Figure 6 is typical for all sequences in our experiments.

As can be observed in Figure 6, the watermark embedded in the I-frames performs quite well after re-encoding to a lower bit-rate. However, the watermark embedded in the P-frames is severely damaged by the re-encoding operation. The reason for this phenomenon is twofold. In the first place, the P-frames are predicted from a *different version* of the I-frame (i.e., an already watermarked I-frame) during MPEG compression which yields a *different* error signal from the one originally contained in the watermarked MPEG stream. Since this error signal is where the watermark is embedded, this difference will introduce errors in the watermark detection. Furthermore, the re-quantization process introduces further differences in the P-frames, which in turn introduce errors in the watermark detection process. We measure the difference between the original and the re-encoded watermarked P-frames by computing the average correlation value of the matching P-frames of the original watermarked sequence (Claire, encoded at 512 kbps) and the sequence re-encoded at 428 kbps. The average correlation value is 0.57, which is quite low and shows that there are indeed significant differences between the original and the re-encoded P-frames. The effects of re-encoding to the I-frames are caused by the re-quantization process, but the effects are not very significant due to the high energy content of the I-frames.




*(a)* *(b)*

*Figure 6. Watermark aBER due to reencoding at lower bit rates for sequences encoded at 700 and 512 kbps. The sequences are watermarked using the Extended DEW algorithm with a watermark payload of 70 bps*

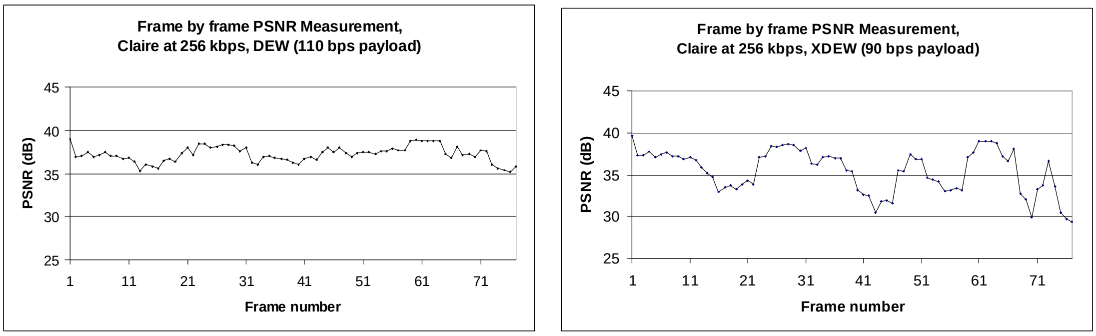
Applying pre-quantization to the data prior to watermarking has been shown to increase the watermark robustness against re-encoding [7]. Pre-quantization is done using a standard MPEG quantization procedure with a certain Q factor. The watermark embedded in the I-frames does indeed have higher robustness when pre-quantization is used. But pre-quantization does not seem to have any positive effect on the robustness of the watermark embedded in the P-frames. Actually, pre-quantization with low Q value seems to have a negative effect on the watermark embedded in the P-frames because in many cases the DCT blocks do not have enough energy to enforce the energy difference determined by the selected pre-quantization Q factor. The end result is an actually higher watermark *e*BER.

### 3.4. Visual quality impact

The visual quality is assessed objectively and subjectively. The objective assessment is done by measuring the PSNR value of the video stream watermarked using both the original and the extended DEW algorithm compared to the unwatermarked stream. The subjective assessment is done by visually judging the quality of the watermarked material. The behavior of the PSNR curves of the extended and the original DEW algorithm is different, because in the original algorithm we only process the I-frames directly while in the extended algorithm we process the I- and the P-frames. This is apparent at higher watermark bit-rates (i.e., when 256 DCT blocks or fewer are used to encode one watermark bit in the P-frames). An example of such behaviour of the PSNR curves is shown in Figure 7.

The different behavior may seem to indicate that it is unfair to simply compare the time-averaged PSNR values produced by the algorithms, and that it is more appropriate to compare only the quality of the processed (watermarked) frames. However, despite the difference in this curve behavior, we decide to use the time-averaged PSNR value to compare the performance of the two algorithms because we consider this to be a better representation of the overall quality of the watermarked material, and this will be the main concern of somebody who is viewing the watermarked material.

The results of the objective visual quality assessment are presented separately for watermark bit-rates below and above the base capacity. The results for watermark bit-rates below the base capacity are presented in Table 2.3. In this table, the values in the brackets below the PSNR values are the difference between the PSNR values of the watermarked sequence and the unwatermarked sequence, which is 37.57 dB. The results for watermark bit-rates above the base capacity are presented in Figure 8. The results in both cases are presented only for sequences encoded at 256 kbps, but the results for other encoded bit-rates show similar behavior, except as noted below.




*(a)* *(b)*

*Figure 7. Frame by frame PSNR measurements of a sequence encoded at 256 kbps, watermarked using: (a) the original algorithm (DEW, watermark payload=110 bps) and (b)the extended algorithm (XDEW, watermark payload=90 bps)*

Table 3 shows that the performances of both the original and extended DEW algorithms are very similar when the watermark bit-rate is below the base capacity. Furthermore, we can see from this table that both algorithms incur virtually no visual degradation to the sequence being watermarked. This means that for these watermark bit-rates, the required energy difference can be enforced without removing too many DCT coefficients.

However, Figure 8 shows the much sharper decrease in visual quality of the sequences watermarked using the extended algorithm compared to the one watermarked using the original algorithm as the watermark payload is increased. We can also observe that at 256 kbps and low payload (up to 70 bps), the performance of both algorithms is similar, but the visual quality of the sequence watermarked using the extended DEW algorithm rapidly deteriorates as the payload is increased above this level. This is not observed at the other encoded bit-rates, where the performance of the two algorithms is very different for all watermark bit-rates. In both cases, the visual degradation introduced by both the original and the extended DEW algorithms is much higher than the one introduced when the watermark bit rates are below base capacity. This means that more DCT coefficients have to be removed to enforce the energy difference for watermark bit-rates above the base capacity.

*Table 3. Visual quality impact assessment for the original and extended DEW algorithm, below base capacity*

| DCT blocks/watermark bit (I-frame) | P-frame settings | | | | | |
|---|---|---|---|---|---|---|
| | No watermark bits in P-frames (DEW) | | 1024 DCT blocks/ watermark bit (XDEW) | | 512 DCT blocks/ watermark bit (XDEW) | |
| | Watermark bitrate (bps) | PSNR (dB) | Watermark bitrate (bps) | PSNR (dB) | Watermark bitrate (bps) | PSNR (dB) |
| 1024 | 2.2 | 37.57 (0) | 8.6 | 37.56 (-0.01) | 21.5 | 37.56 (-0.01) |
| 512 | 6.7 | 37.56 (-0.01) | 13.1 | 37.56 (-0.01) | 26 | 37.55 (-0.02) |
| 256 | 13.5 | 37.54 (-0.03) | 19.9 | 37.54 (-0.03) | 32.7 | 37.54 (-0.03) |
| 128 | 27 | 37.52 (-0.05) | 33.3 | 37.52 (-0.05) | 46.2 | 37.52 (-0.05) |

Our experiments also show that the visual quality degradation is sharper at higher bit-rates. This is observed in sequences watermarked using both algorithms. This means that at lower bit-rates, the MPEG coding artefacts start to play a bigger role in the overall visual quality of the sequences, which is not the case in higher bit-rates. In other words, the watermarking artefacts are dominated by the coding artefacts at lower bit-rates.

The subjective quality assessment reveals that no watermarking artefacts are visible when the watermark bit-rate is below the base capacity. However, some artefacts become visible at high payloads, albeit only in some frames. The artefacts are visible as blurred edges in some blocks and blotches. The blotches appear due to the drift effect, and are sometimes visible in the sequences watermarked using the extended DEW algorithm, especially at higher watermark payloads and lower bit-rates. These blotches appear (and disappear again) gradually over time, except when the blotches appear in a frame directly preceding an I-frame, in which case the blotches disappear completely when the I-frame is displayed. These blotches are the reason why for some frames the PSNR of the sequence watermarked using the extended algorithm drops to a low value, as can be seen in Figure 7 (b).

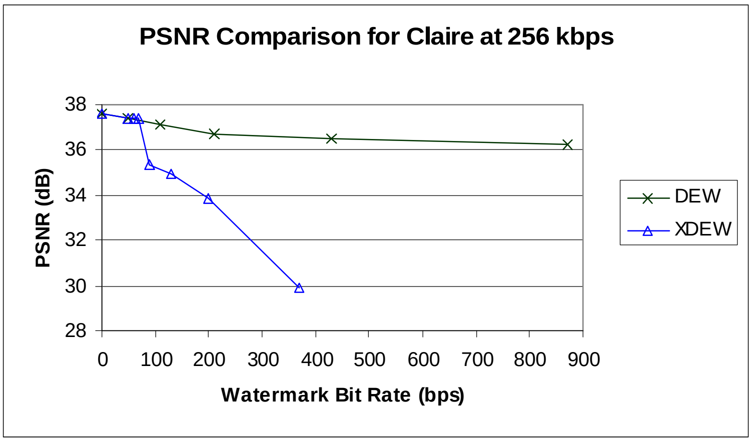



*Figure 8. Visual quality impact comparison between the DEW algorithm and the extended DEW (XDEW) algorithm for watermark bit rates above base capacity.*

## 4. Conclusions

We have presented in this paper the results of our investigation on the performance of the DEW and Extended DEW algorithm in a low bit-rate environment according to three performance criteria, namely payload, robustness and visual quality. The results can be summarised as follow:

1. From the payload point of view, both the original DEW and the extended DEW algorithms perform similarly when the watermark bit-rate is low (below base capacity) for all encoded video bit-rates. However, the extended algorithm generally performs worse than the original DEW as the watermark capacity is increased beyond the base capacity. This is observed in all encoded video bit-rates, except for the bit-rate of 256 kbps, where the performance of the two algorithms is comparable.
2. From the robustness point of view, the watermark embedded in the P-frames is very vulnerable to re-encoding. This is true for all sequences used in this experiment.
3. From the visual quality point of view, both the original and the extended DEW algorithms perform similarly when the watermark bit-rate is below base capacity, and both incur virtually no visual degradation to the data. However, the extended DEW algorithm shows rapid visual quality degradation as the watermark payload is increased. Furthermore, drift effects are visible in the sequences watermarked using the extended algorithm. We also observe that the visual quality curves for the DEW algorithm become less steep as the encoded bit-rates become lower. This is apparent from the fact that, as the encoded bit-rate decreases, encoding artefacts play a larger role to determine the overall quality of the sequence. In other words, the watermarking artefacts become less significant or dominated by the coding artefacts. For the Extended DEW algorithm, the same can be said, but only for lower watermark payloads (less than 70 bps). For higher payloads, the watermarking artefacts are still more significant than the coding artefacts.

Based on these results we draw the following conclusions:

1. The original DEW algorithm scales well to lower bit-rates/smaller spatial resolution for watermark payloads of up to 110 bps, except for the sequence encoded at 256 kbps.
2. The extension scheme to the DEW algorithm we have presented works reasonably for low payloads (up to watermark payload of 70 bps), especially when the watermark bit-rate is still below base capacity. For this payload level, the *e*BER is still relatively low while the introduced watermarking artefacts are either negligible or dominated by the coding artefacts.
3. From these two preceding statements, we can conclude that the Extended DEW algorithm should not be used to pursue higher payloads. To achieve higher watermark payload it is better and easier to adjust the number of blocks used to encode each watermark bit. Even then, the watermark capacity can not be pushed beyond 110 bps without incurring severe *e*BER.
4. Finally, at low bit-rates, the limitations of the MPEG-1/-2 encoder become more obvious. The coding artefacts become visible and at very low bit-rates the specified encoding bit-rate cannot be achieved due to the overhead associated with MPEG-1/-2 stream. Therefore, further developments in low bit-rate video watermarking should be focused on formats more suitable to such bit-rates, like MPEG4 or H.263.

## 5. Final remarks

Further research on watermarking techniques of low bit-rate video has been performed since the work presented in this paper was originally published in 2001. Two examples of recent works on low bit-rate video watermarking found in the literature are briefly discussed below.

The first example, presented in [10], uses semi-fragile watermarks to assess the Quality of Service (QoS) of the communication link between cellular video communication devices. The QoS of the communication link is evaluated by measuring the distortion suffered by the embedded watermarks due to a noisy communication channel. Due to the inherent limitations of cellular devices, the size and bit-rate of the video are limited. In their experiments, the authors used QCIF sequences encoded in MPEG4 at 200 – 1000 kbps. The proposed scheme uses spread spectrum watermarking technique. The watermark message $w$ is embedded in each Video Object (VO) of the MPEG4-coded video. The VO is first transformed into DCT domain, then the watermark pattern is added to the mid-band frequency DCT coefficients of the VO. On the detection side, $w_i'$ is estimated from each VO, where the index $i = 1,\ldots, n$ is the number of VO's in one video frame. An estimate of the embedded watermark, $w'$, is then computed by averaging all $w_i'$ in one frame. By calculating the Mean Square Error (MSE) between the original watermark message $w$ and the estimated watermark message $w'$, the quality of the communication link can be estimated. In [10], the authors introduce random bit errors (with adjustable BER's) to simulate a noisy channel. Their experiments show that the increase of the measured MSE corresponds well to the increase of the introduced BER. Therefore, the proposed technique can be used to estimate the quality of the communication link with additive noise.

The second example is a spread-spectrum based watermarking scheme for low bit-rate (128 – 768 kbps) MPEG4 video proposed in [11]. The proposed scheme is an extension of the scheme proposed in [5]. The watermark is embedded in DCT domain, so that full decoding of the compressed MPEG4 bit stream is not necessary. The detection process, however, is performed in spatial domain. This gives the advantage of enabling watermark detection even when the watermarked MPEG4 video is re-encoded using another compression algorithm. However, the disadvantage is that the proposed scheme cannot use MPEG4-specific properties in the detection process.

Robustness against synchronization attacks (rotation, translation and scaling (RTS) transform of the video frame) is provided by using synchronization templates. The first template is constructed in an approach similar to the one used in [6], namely by tiling the message-carrying watermark pattern periodically over the video frame. In the autocorrelation domain these tiles will produce periodic peaks that can be used to recover watermark synchronization. The second template is a purely synchronization signal. The signal is constructed in a similar manner to that proposed in [12]. In the frequency domain, this signal is composed of peaks in the mid-frequency band. Each peak occupies one frequency coefficient with pseudo-random phase. This synchronization signal provides additional synchronization capability especially in low bit-rate environment where a large part of the watermark is lost due to the compression process [11]. Estimation of the RTS transform parameters is performed in log-polar domain. After the parameters are estimated, the RTS transform is reversed, thus re-synchronizing the watermark.

To improve watermark imperceptibility, the authors use an adaptive gain control mechanism that adjusts watermark signal strength based on the local "activity" of the original video. Small gain is applied to low-activity, i.e., smooth areas and larger gain is applied to high-activity (textured) areas. Furthermore, since the proposed scheme is similar to the one in [5], drift compensation is also needed to prevent error propagation when the watermarked frame is used to predict another frame. Finally, the authors also implement heuristic-based optimization approach to control bit-rate. In this approach, the bit-rate control tries to balance the increase in bit-rate due to the watermarking process and bit allocation based on the local gain factor. The details of this optimization approach is provided in [11]. Bit-rate control is needed to prevent the size of the watermarked video from consuming substantially more bits compared to the original video stream.

The results of the experiments show that the watermark detection rate is quite high even when the watermarked video is attacked using filtering, scaling, rotation and transcoding. The performance of the adaptive gain control is not yet optimal for video segments with a lot of movements since it has not taken the temporal properties of the watermarked video into account. The bit-rate control implemented is shown to be able to limit the increase of the video bitstream size to under than 5%. The complete description of the test setup and results are provided in [11].

## 6. References

1. H. Brynhi, H. Lovett, E. Maarmann-Moe, D. Solvoll, T. Sorensen, *On-demand Regional Television over the Internet*, ACM Multimedia '96, Boston, MA, 1996
2. J. Dittmann, T. Fiebig, R. Steinmetz, S. Fischer, I. Rimac, *Combined Video and Audio Watermarking: Embedding Content Information in Multimedia Data*, in Proceedings of SPIE, Security and Watermarking of Multimedia Content II, Vol. 3971, pp. 455 – 464, San Jose, CA, 2000
3. F. Hartung, B. Girod, *Watermarking of uncompressed and compressed video*, Signal Processing, Vol. 66, No. 3, pp. 283 – 301, May 1998
4. F. Hartung, P. Eisert, and B.Girod, *Digital Watermarking of MPEG-4 facial animation parameters*, Computer Graphics, Vol. 22, No. 3, pp. 425 – 435, 1998
5. F. Hartung, M. Kutter, *Multimedia Watermarking Techniques*, Proceedings of the IEEE, Vol. 87, No. 7, Special Issue: Identification & Protection of Multimedia Information, pp. 1079 – 1107, July 1999
6. T. Kalker, G. Depovere, J. Haitsma, M. Maes, *A video watermarking system for broadcast monitoring*, in Proceedings of SPIE, Security and Watermarking of Multimedia Contents, vol. 3657, pp. 103 – 112, San Jose, CA, January 1999
7. G.C. Langelaar, *Real-time watermarking techniques for compressed video data*, Ph.D. dissertation, Delft University of Technology, The Netherlands, January 2000

8. G.C. Langelaar, I. Setyawan, R.L. Lagendijk, *Watermarking Digital Image and Video Data*, IEEE Signal Processing Magazine, Vol. 17, No. 5, ISSN 1053-5888, pp. 20 – 46, September 2000
9. G.C. Langelaar, R. L. Lagendijk, *Optimal Differential Energy Watermarking of DCT Encoded Images and Video*, IEEE Transactions on Image Processing, Vol. 10, No. 1, January 2001
10. P. Campisi, G. Giunta, A. Neri, *Object-based Quality of Service Assessment using Semi-fragile Tracing Watermarking in MPEG4 Video Cellular Devices*, in Proceedings of IEEE, ICIP 2002, Vol. II, pp. 881 – 884, Rochester, NY, 2002
11. A.M. Alattar, E.T. Lin, M.U. Celik, *Digital Watermarking of Low Bit-Rate Advanced Simple Profile MPEG4 Compressed Video*, IEEE Trans. On Circuits and Systems for Video Technology, vol. 13, no. 8, pp. 787 – 800, August 2003
12. J.J.K. Ó Ruanaidh, T. Pun, *Rotation, scale and translation invariant digital image watermarking*, in Proceedings of IEEE, ICIP 1997, Vol. I, pp. 536 – 539, Santa Barbara, CA, 1997